

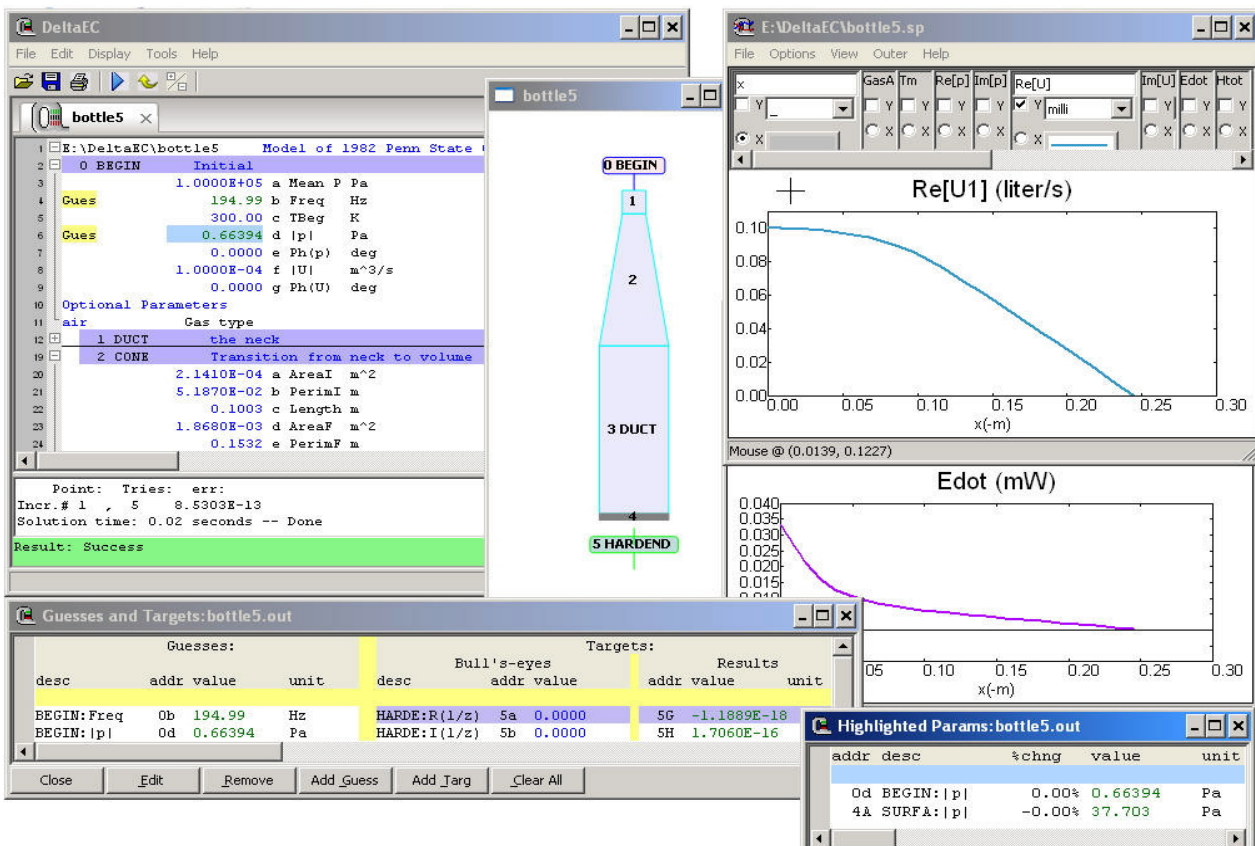
# Design Environment for Low-amplitude Thermoacoustic Energy Conversion

## DELTAEC Version 6.4b2.7 Users Guide

Bill Ward, John Clark, and Greg Swift  
Los Alamos National Laboratory

LA-CC-01-13, LA-CC-16-053  
*December 4, 2017*

For the latest version, visit [www.lanl.gov/thermoacoustics](http://www.lanl.gov/thermoacoustics)  
(moving to [www.lanl.gov/org/padste/adepts/materials-physics-applications/condensed-matter-magnet-science/thermoacoustics/](http://www.lanl.gov/org/padste/adepts/materials-physics-applications/condensed-matter-magnet-science/thermoacoustics/))



# CONTENTS

- Contents** **4**
  
- 1 Introduction** **1**
  - 1.1 What DELTAEC does 1
  - 1.2 The Users Guide 6
  - 1.3 Obtaining DELTAEC 7
  - 1.4 Installing DELTAEC 7
  - 1.5 Acknowledgments 8
  - 1.6 Copyright 8
  
- PART I: TUTORIAL** **9**
  
- 2 Acoustics & user interface** **10**
  - 2.1 Acoustics in DELTAEC 10
  - 2.2 I/o file structure 12
  - 2.3 Running DELTAEC 14
    - 2.3.1 Font augmentation 17
  - 2.4 State plots 17
  - 2.5 Guesses and targets 18
  - 2.6 Other acoustics features 25
    - 2.6.1 Simple segment types 25
    - 2.6.2 The zero of time 27
    - 2.6.3 Gases and solids 27
    - 2.6.4 Turbulence 29
  - 2.7 Other useful features 29
    - 2.7.1 Keeping parameters equal 29
    - 2.7.2 Master–slave links 30
    - 2.7.3 Schematic view 33
    - 2.7.4 Highlights display 34
    - 2.7.5 Structured branches, loops, and networks 35
    - 2.7.6 Manipulating entire segments 39
    - 2.7.7 Thermophysical properties 39
  
- 3 More about plotting** **40**
  - 3.1 State plots 40
    - 3.1.1 Cloning 41

3.1.2	Legs	42
3.1.3	Phasor plotting options	42
3.2	Incremental plotting	44
3.2.1	One-dimensional incremental plots	45
3.2.2	Two-dimensional incremental plots	47
3.2.3	Rewinding incremental plots	48
3.2.4	Incremental plots when more than one model is run	49
<b>4</b>	<b>RPN math</b>	<b>50</b>
4.1	Summary	50
4.2	Non-standard results	51
4.3	List Linkages	53
4.4	Non-standard inputs	53
4.5	Non-standard targets	54
4.6	Changing an integration variable	56
4.7	Two examples	56
4.8	Non-standard guesses	58
4.9	Linking multiple models	58
4.10	Other math segments	60
<b>5</b>	<b>Engines &amp; refrigerators</b>	<b>61</b>
5.1	Principles of calculation	61
5.2	Hofler's refrigerator	63
5.3	Choosing guesses and targets	71
5.4	Meeting targets in an elaborate model	77
5.4.1	Basic rules	77
5.4.2	Developing a substantially new model	78
5.4.3	Modifying an existing model	80
5.5	Standing-wave engine	81
5.6	More thermoacoustic segments	84
5.7	More user-interface features	86
5.8	Stirling and pulse-tube refrigerators	88
5.8.1	Alpha Stirling cryocooler	88
5.8.2	Beta or gamma cryocooler	91
5.8.3	Pulse-tube refrigerator	95
5.9	Exergy flow	99
<b>6</b>	<b>Steady flow, too</b>	<b>102</b>
6.1	Principles of calculation	102
6.2	Reid's refrigerator	104
6.3	TASHE	108
6.4	Self-circulating heat exchanger	114

<b>7</b>	<b>Mixture separation</b>	<b>125</b>
7.1	Principles of calculation . . . . .	125
7.2	Boundary conditions . . . . .	126
7.3	Continuous He–Ar separator . . . . .	129
<b>PART II: REFERENCE</b>		<b>133</b>
<b>8</b>	<b>General principles</b>	<b>134</b>
8.1	Integration from BEGIN to **END . . . . .	134
8.1.1	Additional details for $\dot{N}$ and $p_{2,0,HL}$ . . . . .	136
8.2	Shooting method . . . . .	136
8.3	Numerical options . . . . .	138
8.3.1	Integration options . . . . .	139
8.3.2	Shooting options . . . . .	139
8.4	Error messages . . . . .	140
8.5	Inherent limitations . . . . .	141
<b>9</b>	<b>Good design habits</b>	<b>144</b>
9.1	Attention to details . . . . .	144
9.1.1	Imperfect external heat exchange . . . . .	144
9.1.2	Internal 2-D effects from imperfect external heat exchange . . . . .	145
9.1.3	Heat leaks . . . . .	145
9.1.4	Geometry . . . . .	146
9.1.5	Actual operating conditions . . . . .	146
9.1.6	Thermophysical properties . . . . .	146
9.2	Component conservatism . . . . .	146
9.2.1	Turbulence . . . . .	146
9.2.2	Adiabatic–isothermal mixing . . . . .	147
9.2.3	Stacked screen regenerators . . . . .	147
9.2.4	Pulse tubes and thermal buffer tubes . . . . .	147
9.2.5	Standing-wave engine stacks . . . . .	148
<b>10</b>	<b>Physical segments</b>	<b>150</b>
10.1	Ducts and cones . . . . .	150
10.1.1	DUCT . . . . .	150
10.1.2	CONE . . . . .	156
10.2	Lumped elements . . . . .	159
10.2.1	COMPLIANCE . . . . .	159
10.2.2	SURFACE . . . . .	160
10.2.3	IMPEDANCE . . . . .	161
10.2.4	MINOR . . . . .	163
10.3	Series transducers . . . . .	167
10.3.1	VESPEAKER, IESPEAKER, VEDUCER, and IEDUCER . . . . .	167
10.4	Side-branch transducers and side-branch impedances . . . . .	171
10.4.1	BRANCH, OPNBRANCH, and PISTBRANCH . . . . .	171

10.4.2	VSPEAKER, ISPEAKER, VDUCER, and IDUCER	173
10.5	Stacks and regenerators	176
10.5.1	STKSLAB, STKCIRC, STKRECT, and STKPIN	177
10.5.2	STKSCREEN and STKPOWERLW	182
10.6	Pulse tubes and thermal buffer tubes	187
10.6.1	STKDUCT and STKCONE	187
10.7	Heat exchangers	190
10.7.1	HX and TX	190
10.7.2	SX and PX	193
10.7.3	VXQ1, VXQ2, VXT1, and VXT2	199
10.8	Adiabatic–isothermal interface loss	212
10.8.1	JOIN	212
10.9	Mixture separation	214
10.9.1	MIXBL, MIXSLAB, and MIXCIRC	214
10.9.2	MIXTBL, MIXTSLAB, and MIXTCIRC	217
<b>11</b>	<b>Logistical segments</b>	<b>223</b>
11.1	Starting and ending	223
11.1.1	TITLE	223
11.1.2	BEGIN	223
11.1.3	HARDEND and SOFTEND	227
11.2	Structured branches and unions	229
11.2.1	TBRANCH	229
11.2.2	UNION	231
11.3	Insulation	233
11.3.1	ANCHOR and INSULATE	233
11.4	Math segments	234
11.4.1	RPN	235
11.4.2	VOLUME	242
11.4.3	CONSTANTS	243
11.5	Interfaces to external programs and files	244
11.5.1	BLKDATA	244
11.5.2	SYSEXEC	245
<b>12</b>	<b>Gases (and liquids)</b>	<b>248</b>
12.1	Helium (helium)	248
12.2	Helium–argon mixtures (HeAr)	248
12.3	Helium–xenon mixtures (HeXe)	249
12.4	Neon (neon)	249
12.5	Air (air)	249
12.6	Humid air and fog (HumidAir)	249
12.7	Nitrogen (nitrogen)	250
12.8	Hydrogen (hydrogen)	250
12.9	Deuterium (deuterium)	250

12.10	Carbon dioxide ( <b>C02</b> )	251
12.11	Neon–xenon mixtures ( <b>NeXe</b> )	251
12.12	Natural-gas combustion products ( <b>NGCbProd</b> )	251
12.13	Liquid sodium ( <b>sodium</b> )	252
12.14	Liquid sodium–potassium eutectic ( <b>NaK-78</b> )	252
12.15	User-defined gases	253
12.15.1	User-defined pure gases	253
12.15.2	User-defined gas mixtures	254
<b>13</b>	<b>Solids</b>	<b>256</b>
13.1	Ideal solid ( <b>ideal</b> )	256
13.2	Copper ( <b>copper</b> )	256
13.3	Nickel ( <b>nickel</b> )	256
13.4	Stainless steel ( <b>stainless</b> )	256
13.5	Molybdenum ( <b>molybdenum</b> )	257
13.6	Tungsten ( <b>tungsten</b> )	257
13.7	Kapton ( <b>kapton</b> )	257
13.8	Mylar ( <b>mylar</b> )	257
13.9	Celcor ( <b>celcor</b> )	257
13.10	User-defined solids	257
<b>BACK MATTER</b>		<b>261</b>
<b>14</b>	<b>Derivations</b>	<b>261</b>
<b>15</b>	<b>Symbols</b>	<b>270</b>
<b>Bibliography</b>		<b>274</b>
<b>Alphabetical indexes</b>		<b>280</b>
Segment index		281
Subject index		285

# 1. INTRODUCTION

DELTAEC—Design Environment for Low-Amplitude ThermoAcoustic Energy Conversion—is a computer program that can calculate details of how thermoacoustic equipment performs, or can help the user to design equipment to achieve desired performance. Input data can be modified or entered via DELTAEC’s user interface (or, very awkwardly, by using any text editor). Results can be examined via the user interface, built-in graphics displays, the operating system’s text utilities, or any spreadsheet or graphics software.

For good portability, the computational core of DELTAEC is compiled from GNU Fortran and the user interface is built on Python. All calculations are performed in double precision. Version 6 is currently running on Windows-based PCs and Macintosh. We hope that our use of GNU and Python will soon yield true operating-system independence, so we can provide Linux and Unix variants, too.

The transition from DELTAEC version 5 to DELTAEC version 6 marked the inclusion of thermoacoustic mixture separation and steady-flow effects, as well as the transition from a DOS-keyboard user interface to a Python graphical, keyboard-and-mouse user interface with a built-in plotter.

## 1.1. What DeltaEC does

DELTAEC numerically integrates in one spatial dimension using a low-amplitude, “acoustic” approximation and sinusoidal time dependence. It integrates the wave equation and sometimes other equations such as the energy equation, in a gas (or a very compressible, thermodynamically active liquid), in a geometry given by the user as a sequence of segments (no more than 200) such as ducts, compliances, transducers, and thermoacoustic stacks and regenerators. A glance through the figures below will orient the new user to a broad range of situations that DELTAEC can handle.

DELTAEC always assumes a time dependence of  $e^{i\omega t}$ , so its “wave” equation is essentially the second-order Helmholtz differential equation for the complex pressure amplitude  $p_1(x)$ , which can be regarded as two coupled first-order differential equations for  $p_1(x)$  and the complex volume flow rate amplitude  $U_1(x)$ . An integration of these differential equations is performed for each segment, with pressures, volume flow rates, and some other variables matched at the junctions between segments. In stacks and regenerators, the acoustic solution for pressures and volume flow rates is found simultaneously with the solution of the energy-flow equation to obtain the mean-temperature profile as well. The energy flow through stacks and regenerators is controlled by temperatures and/or heat flows at adjacent heat exchangers. With binary gas mixtures in mixture-separation channels, the solution of the wave equation is found simultaneously with that of the equation describing the mole fluxes



Figure 1.1: Driven, lossy plane-wave resonator.

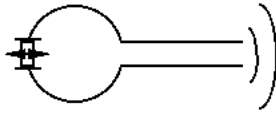


Figure 1.2: Driven, radiating Helmholtz resonator.

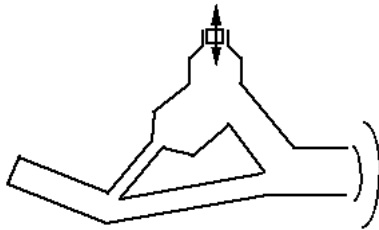


Figure 1.3: Driven duct network.

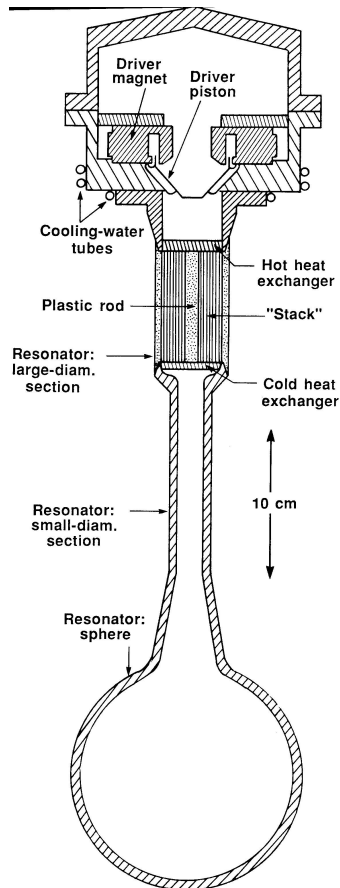


Figure 1.4: Standing-wave thermoacoustic refrigerator (Hofler style [1, 2]).



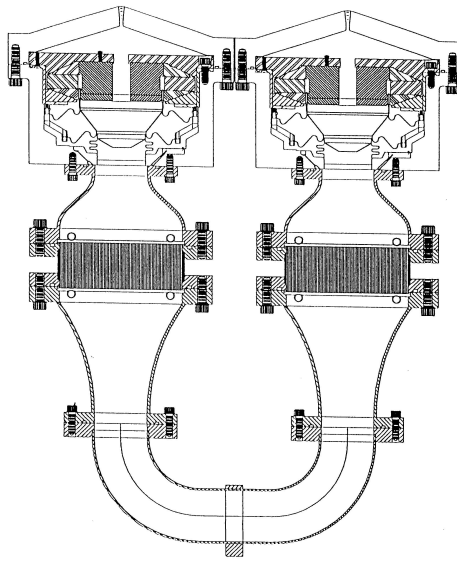


Figure 1.5: Standing-wave thermoacoustic refrigerator (TALSR style [3]).

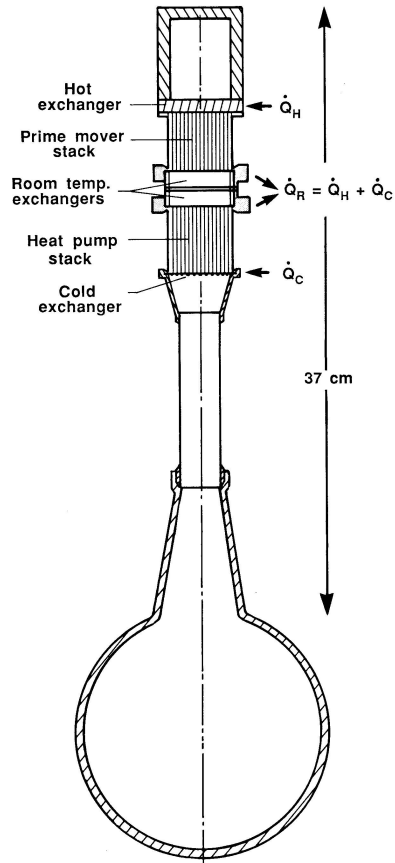


Figure 1.6: John Wheatley's heat-driven refrigerator, affectionately called "the beer cooler" in the early 1980s [4].

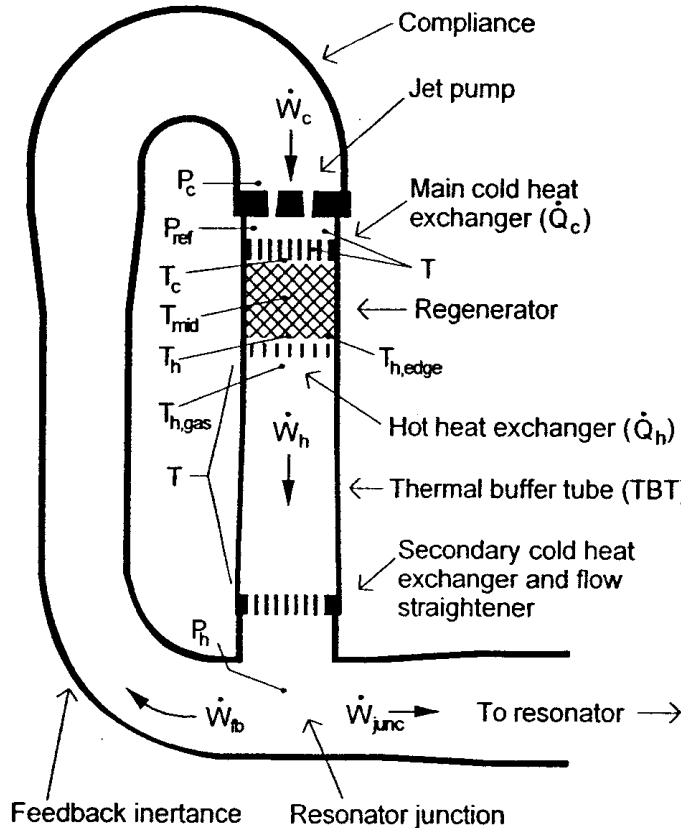


Figure 1.7: Thermoacoustic-Stirling hybrid engine [5].

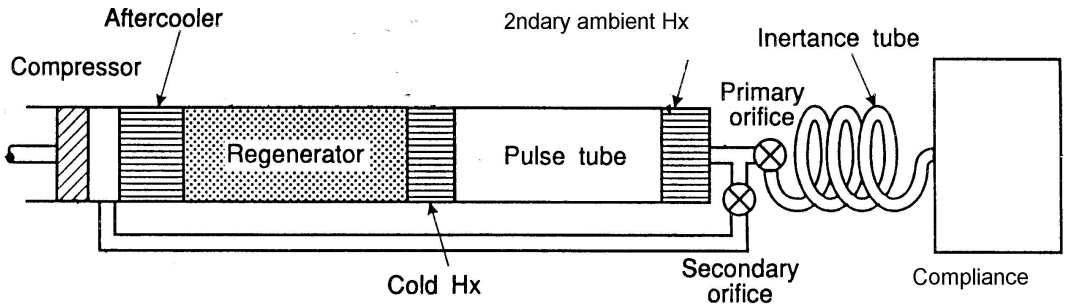


Figure 1.8: Double-inlet pulse-tube refrigerator [6, 7].

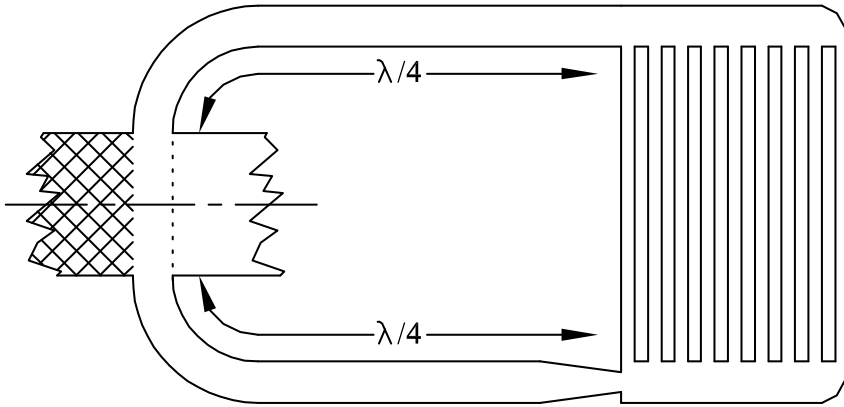


Figure 1.9: Resonant self-circulating heat exchanger with one gas diode. [8, 9]

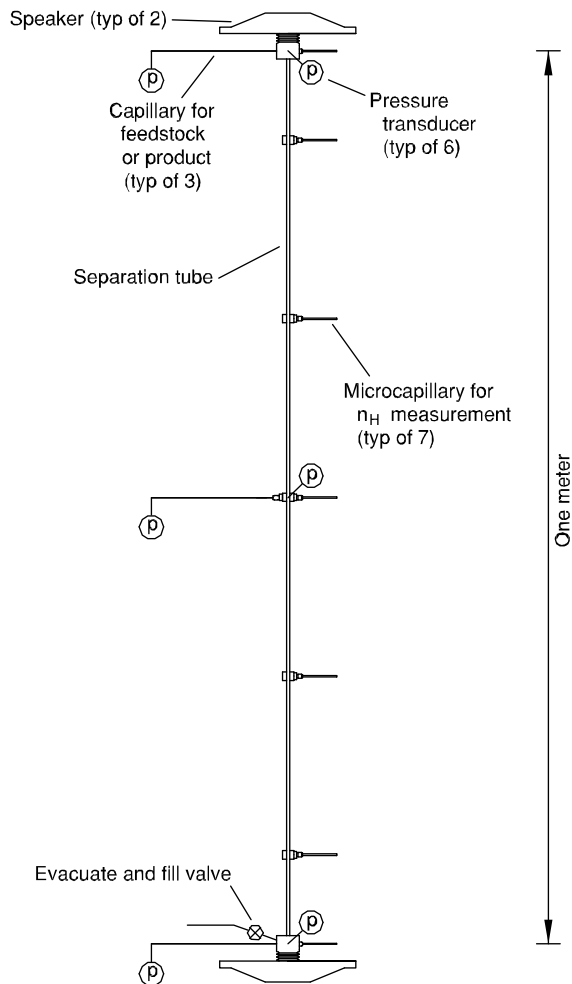


Figure 1.10: Half-wavelength helium-argon mixture separator [10].

of the components of the mixture, inverted to find the light mole fraction as a function of position. Modeling of acoustics superimposed on steady flows is also supported.

With its multi-parameter shooting method to satisfy a variety of mixed boundary conditions, DELTAEC gives the user considerable freedom in choosing which variables are computed as “solutions.” For example, in a simple acoustic resonator (the first example below), DELTAEC can compute the input impedance as a function of frequency, or the resonance frequency for a given geometry and gas, or the length required to give a desired resonance frequency, or even the concentration in a binary gas mixture required to give a desired resonance frequency in a given geometry.

DELTAEC includes few of the nonlinear effects that arise at high amplitudes, so be cautious using it when Mach numbers or Reynolds numbers are high. The principal exceptions to this rule are the turbulence algorithm in ducts and cones (introduced briefly in Chapter 2 and described in detail in Section 10.1.1) and the minor-loss segment, which models lumped-element dissipation where pressure difference is proportional to the square of the velocity (introduced briefly in Chapter 6 and described in detail in Section 10.2.4).

## 1.2. The Users Guide

This Users Guide is a `<.pdf>` document, compatible with Adobe Acrobat 6.0 and later versions. Its internal hyperlinking is convenient for online navigating. (Its overall organization also makes it suitable for studying a printed copy. If some of the figures are garbled or missing from your printed version, your printer may not be postscript-compatible, and you should find the Advanced options button in Adobe Acrobat’s print dialog, click it, and select “Print as image” at 300 dpi.)

In Part I, “Tutorial,” we teach the use of DELTAEC by increasingly complicated examples in Chapters 2–7. Chapter 2 is just acoustics, without thermoacoustics. It serves to introduce DELTAEC’s input/output formats and editing, running, and plotting features. Chapter 5 gives the most complete discussion of the overall principles behind the thermoacoustics computations, and the simplest thermoacoustic engine and refrigerator examples. The agreement of such examples with published experimental data serves as validation of the code. Chapter 6 introduces the effects of superimposed steady flows, and Chapter 7 introduces separation mole fluxes and concentration gradients in binary-gas mixtures.

Part II, “Reference,” includes segment-by-segment reference sections for the experienced user, documenting the assumptions built into the computations for each segment and the data format for each segment. “Reference” also includes chapters with calculation formulas for thermophysical properties. It is our hope that experienced users can quickly find the information they need in Part II, especially using the index and the internal hyperlinks, while new users will find the wordier explanations of the Tutorial chapters helpful.

The examples we’ve included are simpler than DELTAEC files we use in our own research. We’ve maintained this simplicity in the Users Guide to avoid clutter. Experienced users will find that the number of segments in their DELTAEC files grows and grows, as small effects are included and non-standard features are implemented with RPN segments.

Some of the examples here were run many years ago in MS-DOS, others on a Mac, others with Windows. Many of the examples were created using earlier versions of DELTAEC and

DELTAEC, so there may be some formatting irregularities. In Chapter 2, screen shots of the user interface are often shown, but in later chapters most examples are displayed as they appear in a text editor after DELTAEC has saved them to the disk. This keeps the Users Guide `<.pdf>` file size smaller.

We assume that the reader of the Users Guide is very comfortable with basic linear acoustics [11, 12] and reasonably familiar with thermoacoustics. The choices of which variables to regard as independent and integrable reflects the philosophy of Ref. [13], extended here to accommodate mixture separation and superimposed steady flow. We use variables as defined, for example, in the lists of symbols in Refs. [13] and [4], and Chapter 15.

### 1.3. Obtaining DeltaEC

The latest version of DELTAEC is freely available from [www.lanl.gov/thermoacoustics/](http://www.lanl.gov/thermoacoustics/), which should redirect to [www.lanl.gov/org/padste/adepts/materials-physics-applications/condensed-matter-magnet-science/thermoacoustics/](http://www.lanl.gov/org/padste/adepts/materials-physics-applications/condensed-matter-magnet-science/thermoacoustics/). DELTAEC is under continual development, so regular users should update their copies occasionally.

There is no formal registration for DELTAEC, no fee, and no formal support or warranty of any kind (please read the copyright notice and disclaimer in Section 1.6). We are interested in hearing from users so we can fix any bugs that are found. Please report any bugs related to platform compatibility and the user interface to Bill Ward ([ww@lanl.gov](mailto:ww@lanl.gov)), with a cc to Greg Swift ([swift@lanl.gov](mailto:swift@lanl.gov)); please report any bugs related to thermoacoustic computations or the Users Guide to Greg with a cc to Bill.

News of your successes using this code will encourage us and our sponsors to consider this effort worthwhile and will enable us to continue DELTAEC’s development and freely-shared status. We are especially grateful when you acknowledge DELTAEC in publications and reports and when you mention it to individuals at agencies that support acoustics research. This improves our chances to create and distribute improvements to DELTAEC in the future.

### 1.4. Installing DeltaEC

Download the installation file from the Los Alamos thermoacoustics website, and run it. DELTAEC should automatically install itself in `<d:\Program Files(x86)\DeltaEC>` or some other sensible default folder. This Users Guide (`<UsersGuide.pdf>`) appears in the same folder. In Windows, this all appears as a program group accessible in the usual way through the Start button, and a desktop icon for launching the program should appear. The file extension `<.out>` should be automatically associated with DELTAEC, so double clicking any `<.out>` file should also launch the program. (If not, consider the standard rebooting of the computer that Windows usually requires for establishing such file association after program installation.) During installation, choose a location without write protection, such as `<MyDocuments>`, for installing the Examples discussed in the rest of the Users Guide; or put an archival copy of the Examples into the write-protected installation folder, and later make an additional “working” copy elsewhere.

For a second or two, launching DELTAEC version 6.4 can display a black window with OMP warning 178 and System Error 126. Don’t worry—this can be ignored.

DELTAEC can be uninstalled via the Windows Control Panel, Add or Remove Programs; or by running `<d:\Program Files(x86)\DeltaEC\D-uninst.exe>`. To install a new version of DELTAEC, it should not be necessary to un-install a previous version. If a re-installation of DELTAEC does not work as it should, even after uninstalling the previous version, look for ghost processes named `<DeltaEC.exe>` in the Windows task manager, and kill them; or reboot your computer to kill such ghosts. (If you can reproducibly create such ghosts, please let us know how, so we can prevent them in future versions. We have never yet seen a ghost in versions 6.3 or 6.4, so we hope we've exorcised them forever.)

## 1.5. Acknowledgments

The development of DELTAEC and DELTAEC has been supported in part by many agencies and entities: Tektronix Corporation, SPAWAR, the Naval Postgraduate School, ONR, Praxair, and, most importantly, by several offices of the Department of Energy: Advanced Industrial Concepts, Materials Science (a part of Basic Energy Sciences in the Office of Science), the Technology Transfer Initiative Office, and the Office of Fossil Energy. Local support at Los Alamos has also been provided, via our Industrial Partnership Office (later called the Technology Transfer Division; now the Feynman Center for Innovation), our Locally Directed R&D program, and our local management in the Materials Physics and Applications Division and the Advanced Engineering Technology Division here at Los Alamos National Laboratory.

In the 1990s, a long discussion with Pat Arnott helped us define the initial scope of this work, and comparisons with the results of parallel-plate-stack codes (written by Al Migliori and Dick Martin) were useful in the early stages. Suggestions by Kim Godshalk, Charles Jin, Tom Hoffer, Jeff Olson, Scott Backhaus, Vince Kotsubo, and Jalal Zia led to significant improvements in capability and usability through the years. The TRITON project-kickoff meeting at Penn State motivated the development of the RPN segment. Charles Jin, Ray Radebaugh, and the code REGEN3.1 were indispensable in development of the stacked-screen algorithm. Scott Backhaus, David Gardner, Matt Poese, and Steve Garrett have helped shape the DELTAEC user interface.

## 1.6. Copyright

This Software was produced under a U.S. Government contract (DE-AC52-06NA25396) by Los Alamos National Laboratory, which is operated by the Los Alamos National Security, LLC (LANS) for the U.S. Department of Energy's National Nuclear Security Administration. The U.S. Government is licensed to use, reproduce, and distribute this Software. Permission is granted to the public to copy and use this Software without charge, provided that this Notice and any statement of authorship are reproduced on all copies. Neither the Government nor the LANS makes any warranty, express or implied, or assumes any liability or responsibility for the user of this Software.

# Part I: Tutorial

## 2. ACOUSTICS & USER INTERFACE

In this Chapter we use the simplest acoustic segments, especially ducts and cones, to introduce DELTAEC's most basic acoustics and user-interface features.

### 2.1. Acoustics in DeltaEC

DELTAEC deals with one-dimensional sequences of acoustic and thermoacoustic elements, called *segments*, so DELTAEC's "wave" equation is one-dimensional. We always assume a time dependence of  $\text{Re}[e^{i\omega t}]$ , so the "wave" equation can be taken as the second-order Helmholtz differential equation for the complex pressure amplitude  $p_1(x)$ . In its most familiar form, for an  $x$ -independent cross-sectional area  $A$ , without viscous or thermal-hysteresis losses, it is

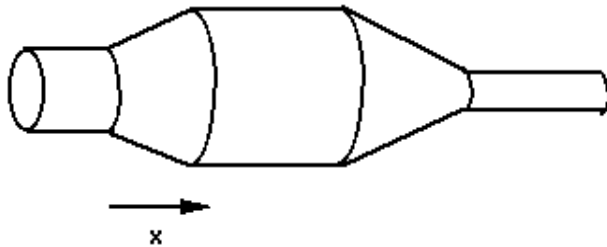
$$p_1 + \frac{a^2}{\omega^2} \frac{d^2 p_1}{dx^2} = 0. \quad (2.1)$$

It is often easier to think of this second-order equation as two coupled first-order equations in pressure  $p_1$  and volume flow rate  $U_1$  :

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{A}U_1, \quad (2.2)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A}{\rho_m a^2}p_1. \quad (2.3)$$

This point of view is taken in Ref. [13]. The  $dp_1/dx$  equation is derived from the momentum equation of fluid mechanics and the  $dU_1/dx$  equation is derived from the continuity equation of fluid mechanics. In this form, the equations are ready for simultaneous numerical integration along the axial position coordinate  $x$  to generate solutions  $p_1(x)$  and  $U_1(x)$ .



DELTAEC uses more complicated momentum and continuity equations that include additional effects such as dissipation of acoustic power along the sides of ducts. It uses different equations in different segments to suit local circumstances. For example, in boundary-layer



approximation in large-diameter ducts and shallow cones, the governing equations are

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{A} \left[ 1 - \frac{1-i}{2} \frac{\Pi}{A} \delta_\nu \right]^{-1} U_1, \quad (2.4)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A}{\rho_m a^2} \left[ 1 + \frac{1-i}{2} \frac{\Pi}{A} \frac{\gamma-1}{1+\epsilon_s} \delta_\kappa \right] p_1. \quad (2.5)$$

where  $A$  is the cross-sectional area,  $\Pi$  is the perimeter,  $\delta_\nu$  and  $\delta_\kappa$  are the viscous and thermal penetration depths, respectively,  $\gamma$  is the ratio of isobaric and isochoric specific heats, and  $\epsilon_s$  is a correction for thermal properties of the solid wall that is usually negligible. The equations for each DELTAEC segment are given in Chapters 10 and 11.

DELTAEC uses continuity of  $p_1$  and  $U_1$  to pass from the end of one segment to the beginning of the next. Within each segment, wave propagation depends on local parameters such as area and perimeter as well as on global parameters such as frequency. Although DELTAEC uses analytic solutions to the governing equations for some of the simplest segment types, it usually must integrate the equations numerically, so it is generally correct to imagine DELTAEC beginning at the beginning of a series of segments and numerically integrating “a momentum equation”  $\mathcal{F}_{\text{momentum}}$ , such as Eq. (2.2) or (2.4), and “a continuity equation”  $\mathcal{F}_{\text{continuity}}$ , such as Eq. (2.3) or (2.5), through each segment, sequentially, to the end of the series of segments, always using the local values of variables such as area, perimeter, and  $p_1$  and  $U_1$  themselves:

$$\frac{dp_1}{dx} = \mathcal{F}_{\text{momentum}}(p_1, U_1, T_m, p_m, \omega, \text{geometry, gas properties, solid properties, etc.}), \quad (2.6)$$

$$\frac{dU_1}{dx} = \mathcal{F}_{\text{continuity}}(p_1, U_1, T_m, p_m, \omega, \text{geometry, gas properties, solid properties, etc.}), \quad (2.7)$$

where most variables on the right-hand sides of these equations are functions of  $x$ .

The solution  $p_1(x)$ ,  $U_1(x)$  is only determined uniquely if four real boundary conditions are imposed, because the governing equations can be expressed as two coupled first-order equations in two complex variables or as four coupled first-order equations in four real variables. This is true whether considering a single segment or a one-dimensional series of segments with each joined to its neighbor(s) by continuity of  $p_1$  and  $U_1$ . If all four boundary conditions are given at the initial end of the apparatus (i.e., if we know the magnitude and phase of  $p_1$  and the magnitude and phase of  $U_1$  at the initial end) then the integration is straightforward, proceeding from the initial end to the final end. But usually one or more boundary conditions are instead given at the final end. In such circumstances DELTAEC uses a shooting method,<sup>1</sup> by *guessing* any unknowns among the four numbers defining  $p_1$

---

<sup>1</sup>Precisely speaking, DELTAEC forms a system of nonlinear equations from the model using the guesses and targets that the user selects, and manipulates the guesses to make the integration results match the targets. The routine incorporated in the code is called DNSQ, and it is part of the SLATEC Common Mathematical Library, which is freely available through the internet software repository at “<http://www.netlib.org>.” Its algorithm is a modification of the Powell hybrid method. See Chapter 8 for details.

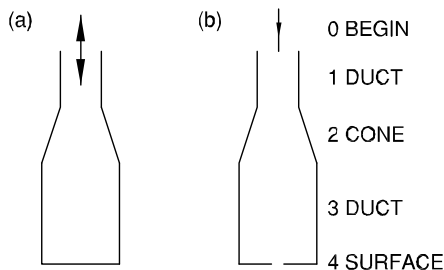


Figure 2.1: A simple open-topped bottle for the first example, <bottle0.out>. (a) Conventional representation. (b) DELTAEC representation.

and  $U_1$  at the initial end of the integration, integrating to the other end, comparing the results with the *target* boundary conditions imposed at that other end or elsewhere, and adjusting its guesses until the integration results meet the targets.

One of DELTAEC’s most powerful features is that the guesses are not limited to the conventional choices consisting of real and imaginary parts of  $p_1$  and  $U_1$  (or, equivalently, magnitudes and phases) at the initial end of the numerical integration along  $x$ . Any variables that have an effect on the downstream target variables can be used. This enables DELTAEC to calculate a resonance frequency, a geometrical dimension, a temperature, or even the concentration in a binary gas mixture to satisfy given boundary conditions.

## 2.2. I/o file structure

We begin this tutorial with an open-topped bottle. This example introduces the basic input/output file structure of DELTAEC and how to “run” the numerical integration.

DELTAEC recognizes all of its own output files as valid input files. In practice, users rarely make a new file from scratch; it is much more common to start with an existing file and modify it as needed. Thus, we begin with a file called <bottle0.out> (included in the <Examples\Acoustics> directory or folder that DELTAEC installs, usually in <d:\ProgramFiles(x86)\DeltaEC\Examples\Acoustics>), which we created earlier from DELTAEC’s user interface. Figure 2.1 shows the acoustician’s usual cartoon of this bottle and a pictorial representation of how we modeled it for DELTAEC. For now, examine this file in a text editor. A DELTAEC model file must always be a plain text file, in the native text format of the machine on which it is running.

```

TITLE      Model of 1982 Penn State Championship Bottle
!->E:\deltaec\TEXTFILES\examples\acoust\BOTTLE0.OUT
!Created@10:24:32 01-Jun-2007 with DeltaEC Vers. 6.0a0 under win32
!Using Win 5.1.2600 (Service Pack 2) under Python DeltaEC.
!----- 0 -----
BEGIN      The mouth
1.0000E+05 a Mean P Pa
300.00    b Freq Hz
300.00    c TBeg K
1.0000    d |p| Pa
0.0000    e Ph(p) deg
1.0000E-04 f |U| m^3/s
0.0000    g Ph(U) deg
air       Gas type
!----- 1 -----
DUCT      the neck
2.1410E-04 a Area m^2
5.1870E-02 b Perim m
0.0000    A |p| Pa
0.0000    B Ph(p) deg

```

```

1.7780E-02 c Length m          0.0000 C |U|      m^3/s
                                0.0000 D Ph(U)    deg
                                0.0000 E Htot    W
                                0.0000 F Edot   W
ideal                          Solid type
!----- 2 -----
CONE      Transition from neck to volume
2.1410E-04 a AreaI m^2        0.0000 A |p|      Pa
5.1870E-02 b PerimI m        0.0000 B Ph(p)   deg
0.1003 c Length m           0.0000 C |U|      m^3/s
1.8680E-03 d AreaF m^2       0.0000 D Ph(U)    deg
0.1532 e PerimF m           0.0000 E Htot    W
ideal                          Solid type
!----- 3 -----
DUCT      Bottle volume
1.8680E-03 a Area m^2        0.0000 A |p|      Pa
0.1532 b Perim m            0.0000 B Ph(p)   deg
0.1270 c Length m           0.0000 C |U|      m^3/s
                                0.0000 D Ph(U)    deg
                                0.0000 E Htot    W
                                0.0000 F Edot   W
ideal                          Solid type
!----- 4 -----
SURFACE   Bottom End
1.8680E-03 a Area m^2        0.0000 A |p|      Pa
                                0.0000 B Ph(p)   deg
                                0.0000 C |U|      m^3/s
                                0.0000 D Ph(U)    deg
                                0.0000 E Htot    W
                                0.0000 F Edot   W
ideal                          Solid type

```

Several features of DELTAEC model files are illustrated here. Model files should be named `<something.out>`. These files consist of a set of “segments” whose order and format are important. After a `TITLE` (with an optional heading where the user can type notes), the initial (or “zeroth”) segment is always a `BEGIN` segment. This is one of the “logistical” segments of DELTAEC, usually having no actual geometrical correspondence to parts of the hardware being modeled. Subsequent segments describe the geometry and other properties of the “physical” hardware parts of the acoustic system—here, `DUCTs`, a `CONE`, and a `SURFACE`. All units are MKS. The number and order of data in each segment is crucial. Within each line, the first number (e.g., “1.e5” or “100.”) or word (e.g., “helium” or “BEGIN”) is the most important to DELTAEC as input.<sup>2</sup> Subsequent fields on each line give the units or name of the variable whose value appears. Lines that begin with “!” are comments with no impact on the integrations. Numbers can be in fixed or exponential format.<sup>3</sup> Segment names are stored uppercase, and only the first five characters are interpreted.

The column of zeroes on the right side shows where DELTAEC’s calculated results will appear. At the moment, they are all zero because DELTAEC has not yet calculated anything.

`BEGIN` sets the stage, in this case, with 1-bar room-temperature air being driven at 300 Hz with a pressure amplitude of 1 Pa and a volume flow rate amplitude of 0.0001 m<sup>3</sup>/s, in phase with the pressure oscillation. Some of these variables, such as frequency, extend unchanged through all subsequent segments. Other variables, such as pressure amplitude, are initial conditions for DELTAEC’s numerical integration through subsequent segments.

A lossy duct segment, `DUCT`, comes next. Here, we have made the perimeter equal to  $\sqrt{4\pi \times \text{area}}$ , to make this a circular duct. A `CONE` and another `DUCT` come next, completing the shape of the bottle. The file ends with a `SURFACE` to account for oscillatory-pressure thermal-hysteresis dissipation at the bottom.

<sup>2</sup>Exceptions: The file reader looks beyond the first field of characters to resolve ambiguity about optional parameters, described later.

<sup>3</sup>In some computers, integers must be followed by a decimal point. Also, some computers require the last line in the file to be followed by an end-of-line character, before the end-of-file character occurs.

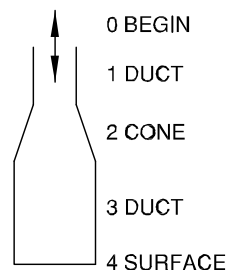
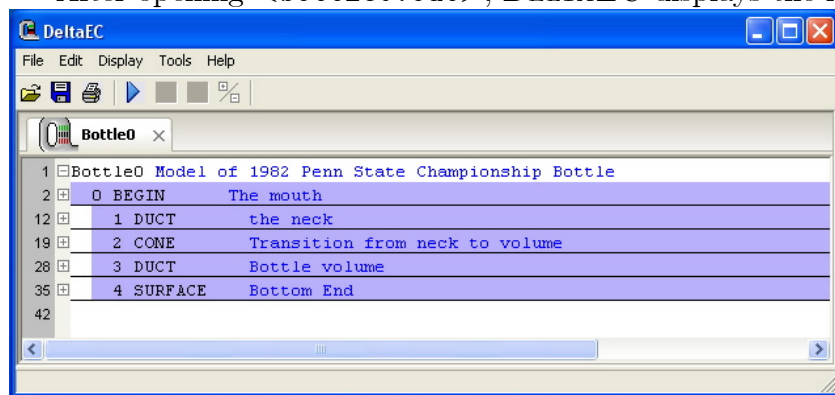
## 2.3. Running DeltaEC

As children, most of us learn that blowing across the mouth of a bottle produces a pleasant audible tone. With this example, the new user of DELTAEC will obtain the frequency of that tone, while exploring the user interface and learning to “run” the numerical integration.

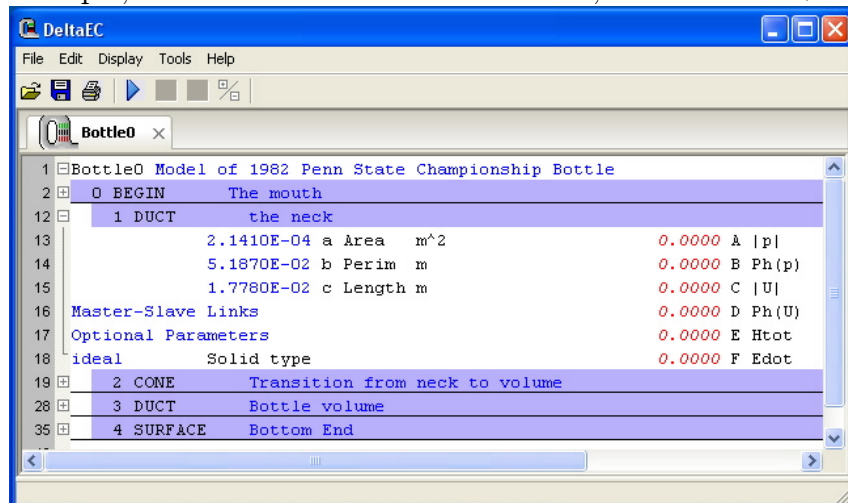
Open the file <bottle0.out>, using whichever of these three methods you prefer:

1. Open the folder <...\\Examples\\Acoustics> wherever your installation has created it, and double click on the file <bottle0.out>. If this folder is write-protected from changes, SaveAs to an unprotected folder immediately.
2. In Windows, find DELTAEC through the Start button at the bottom-left corner of your desktop, via Programs and the DELTAEC group. Double click on the program to launch it. Then open the file <bottle0.out> via the DELTAEC Files pulldown menu or by clicking the icon that looks like a manila folder. In either case, find your way to the <Examples\\Acoustics> folder in the usual way.
3. Double click on the Desktop DELTAEC icon to launch DELTAEC, and open the file <bottle0.out> as described in method 2.

After opening <bottle0.out>, DELTAEC displays the list of segments, like this:



Clicking on the “+” next to any segment’s name exposes that segment’s details. For example, to see the dimensions of the neck, click on the “+” next to 1 DUCT:



Blue text (numbers or words) means you can double click or right click to make a change or see more options. Italic red numbers indicate what will be a result, as soon as the calculation is run, but right now the red color indicates that these numbers are probably inconsistent with the blue input numbers. Green numbers that you will see shortly indicate good results: calculated by DELTAEC and consistent with all the other numbers in the model.

Now show all the segments in the model, by clicking on the “+” for each segment or on the big “+/-” on the toolbar. What you see should look like this:<sup>4</sup>

```

TITLE      Model of 1982 Penn State Championship Bottle
!----- 0 -----
BEGIN      The mouth
1.0000E+05 a Mean P      Pa
300.00    b Freq         Hz
300.00    c TBeg        K
1.0000    d |p|         Pa
0.0000    e Ph(p)       deg
1.0000E-04 f |U|        m^3/s
0.0000    g Ph(U)       deg
air        Gas type
!----- 1 -----
DUCT      the neck
2.1410E-04 a Area        m^2      0.0000    A |p|         Pa
5.1870E-02 b Perim      m          0.0000    B Ph(p)       deg
1.7780E-02 c Length    m          0.0000    C |U|         m^3/s
0.0000    D Ph(U)       deg
0.0000    E Hdot        W
0.0000    F Edot        W
ideal     Solid type
!----- 2 -----
CONE      Transition from neck to volume
2.1410E-04 a Areal      m^2      0.0000    A |p|         Pa
5.1870E-02 b PerimI    m          0.0000    B Ph(p)       deg
0.1003    c Length    m          0.0000    C |U|         m^3/s
1.8680E-03 d AreaF     m^2      0.0000    D Ph(U)       deg
0.1532    e PerimF    m          0.0000    E Hdot        W
ideal     Solid type
!----- 3 -----
DUCT      Bottle volume
1.8680E-03 a Area      m^2      0.0000    A |p|         Pa
0.1532    b Perim    m          0.0000    B Ph(p)       deg
0.1270    c Length    m          0.0000    C |U|         m^3/s
0.0000    D Ph(U)       deg
0.0000    E Hdot        W
0.0000    F Edot        W
ideal     Solid type
(etc.)

```

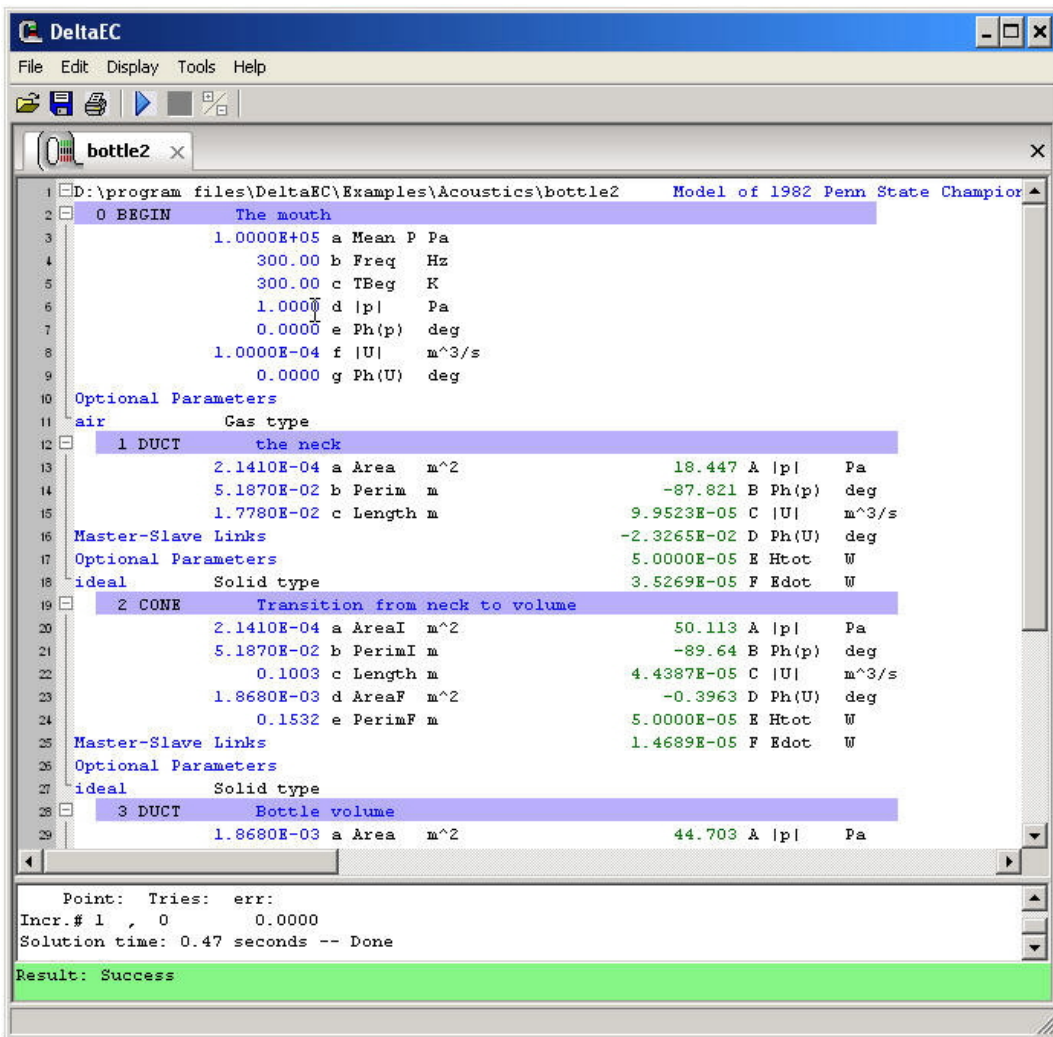
The BEGIN segment establishes that the bottle is full of air at atmospheric pressure and room temperature, and it launches a 300-Hz wave into the bottle with a pressure amplitude of 1 Pa and a volume-flow-rate amplitude of  $10^{-4}$  m<sup>3</sup>/s. (We will examine its “optional” parameters in later chapters, when we encounter them.)

Now, run the file, by typing ‘r’, by clicking the blue “Run” button, or by accessing “Run” in the Tools pulldown menu. (To encourage users to get in the habit of using the keyboard shortcuts, the Users Guide summarizes all such options in the briefest possible way, with the shortcut letter underlined, e.g., run. Infrequently used operations whose shortcuts the user has not memorized can be found by browsing the pulldown menus.)

Upon running, the Run Monitor appears at the bottom of the display, indicating with its cheerful green color and the word “Success” that the run has succeeded, in this case

<sup>4</sup>Please ignore minor formatting differences between what you see on your computer screen and what is present in the Users Guide. Much of the Users Guide was written before the user interface was complete, and it’s much easier for us to let text-formatted results page-break automatically in the Users Guide than to manage large figures that show the actual computer-screen display. Text-formatted results also minimize the Users Guide file size.

taking only a fraction of a second. (Typing “b” toggles the Run Monitor display off and on, and resets it to default size.) Now the results in the right column of the model display are nonzero, and they have changed from red italic to green, indicating that they are consistent with one another and with the blue input numbers:



DELTAEC has integrated through the model and calculated some results, in particular  $p_1(x)$  and  $U_1(x)$ . The results in the right column of the `<.out>` file show values for  $p_1(x)$  and  $U_1(x)$  and other results at the *end* of each segment. If a few numbers are all you want to see, you are done!

One of the other results is the acoustic power  $\dot{E}$ ,

$$\dot{E} = \frac{1}{2} \text{Re} \left[ p_1 \tilde{U}_1 \right], \quad (2.8)$$

which is always listed at line F.

The total power  $H_{tot}$ , listed at line E, is introduced in Chapter 5; it can safely be ignored in this chapter, and for all purely acoustic problems in which heat exchangers and time-averaged temperature gradients in the  $x$  direction are of no concern.

### 2.3.1. Font augmentation

You can change the size of the font in DELTAEC’s main display and many secondary windows by spinning your mouse wheel while holding down the Ctrl key, or by using Ctrl n, Ctrl b, and Ctrl /. (Be sure the computer’s focus is on the DELTAEC window—click it if necessary.) Resizing the DELTAEC window by dragging its perimeter with the mouse, or making DELTAEC fill the entire screen by clicking the usual Windows “Maximize” button in the top-right corner, makes more room for a bigger font.

The shades of blue, green, and red used to indicate different types of variables seem to show up well for color-blind users.

## 2.4. State plots

State plots allow the user to view graphs showing the distribution of temperature, pressure, volume flow rate, energy, and many other variables as functions of  $x$  along the entire length of a model or any subset of that length. After each successful run, DELTAEC creates a state-plot file, `<.sp>`, with this information in tabular format. You can open this file from your hard disk folder with any text editor or spreadsheet program, or display it with DELTAEC’s plotter, using “Plot SP file” in the Display pulldown menu. The example below, `<bottle2.sp>` (`<bottle2>` is identical to `<bottle0>`), except that it includes the `<.ssv>` file, described below), does not show all of the available columns of data; a text editor or spreadsheet would let you scroll across to view more columns to the right.

```
->bottle2.sp
!Created@15:28:56 7-Apr-07 with DeltaE Vers. 6.0g7 for the IBM/PC-Compatible
-- Model of 1982 Penn State Championship Bottle ==
Leg:Seg x(m) GasA(m^2) T(K) Re[p] (Pa) Im[p] (Pa) Re[U] (m3/s) Im[U] (m3/s) Edot(W) Htot(W) Xdot(W)
0: 1 0.000 2.1410E-04 300.0 1.0000 0.0000 1.0000E-04 0.0000 5.0000E-05 5.0000E-05 5.0000E-05
0: 1 0.002 2.1410E-04 300.0 0.9704 -1.8463 9.9995E-05 -5.0520E-09 4.8524E-05 5.0000E-05 4.8524E-05
0: 1 0.004 2.1410E-04 300.0 0.9408 -3.6924 9.9981E-05 -9.8803E-09 4.7049E-05 5.0000E-05 4.7049E-05
0: 1 0.005 2.1410E-04 300.0 0.9111 -5.5381 9.9957E-05 -1.4485E-08 4.5574E-05 5.0000E-05 4.5574E-05
0: 1 0.007 2.1410E-04 300.0 0.8813 -7.3833 9.9924E-05 -1.8864E-08 4.4099E-05 5.0000E-05 4.4099E-05
0: 1 0.009 2.1410E-04 300.0 0.8514 -9.2278 9.9881E-05 -2.3019E-08 4.2625E-05 5.0000E-05 4.2625E-05
0: 1 0.011 2.1410E-04 300.0 0.8215 -11.071 9.9828E-05 -2.6949E-08 4.1152E-05 5.0000E-05 4.1152E-05
0: 1 0.012 2.1410E-04 300.0 0.7915 -12.914 9.9766E-05 -3.0653E-08 3.9680E-05 5.0000E-05 3.9680E-05
0: 1 0.014 2.1410E-04 300.0 0.7615 -14.755 9.9695E-05 -3.4132E-08 3.8208E-05 5.0000E-05 3.8208E-05
0: 1 0.016 2.1410E-04 300.0 0.7314 -16.595 9.9614E-05 -3.7385E-08 3.6738E-05 5.0000E-05 3.6738E-05
0: 1 0.018 2.1410E-04 300.0 0.7013 -18.434 9.9523E-05 -4.0412E-08 3.5269E-05 5.0000E-05 3.5269E-05
0: 2 0.018 2.1410E-04 300.0 0.7013 -18.434 9.9523E-05 -4.0412E-08 3.5269E-05 5.0000E-05 3.5269E-05
0: 2 0.028 3.0593E-04 300.0 0.5705 -27.065 9.8705E-05 -5.6940E-08 2.8928E-05 5.0000E-05 2.8928E-05
0: 2 0.038 4.1411E-04 300.0 0.4906 -33.188 9.7222E-05 -7.4046E-08 2.5078E-05 5.0000E-05 2.5078E-05
0: 2 0.048 5.3864E-04 300.0 0.4382 -37.707 9.4923E-05 -9.2334E-08 2.2540E-05 5.0000E-05 2.2540E-05
0: 2 0.058 6.7951E-04 300.0 0.4021 -41.129 9.1662E-05 -1.1242E-07 2.0741E-05 5.0000E-05 2.0741E-05
0: 2 0.068 8.3673E-04 300.0 0.3762 -43.761 8.7296E-05 -1.3494E-07 1.9373E-05 5.0000E-05 1.9373E-05
0: 2 0.078 1.0103E-03 300.0 0.3570 -45.799 8.1690E-05 -1.6053E-07 1.8260E-05 5.0000E-05 1.8260E-05
0: 2 0.088 1.2002E-03 300.0 0.3425 -47.373 7.4714E-05 -1.8986E-07 1.7293E-05 5.0000E-05 1.7293E-05
0: 2 0.098 1.4065E-03 300.0 0.3313 -48.576 6.6246E-05 -2.2360E-07 1.6404E-05 5.0000E-05 1.6404E-05
(etc., etc.)
```

State-plot files have the following features:

- Numbers are lined up in columns. The initial columns are leg number (relevant when TBRANCH and/or multiple BEGIN segments are used; described in Section 11.2.1), segment number, and  $x$ .
- DELTAEC writes `Nint + 1` lines of data for each segment that it integrates (e.g., ducts, cones, and stacks). `Nint` is the number of Runge-Kutta integration steps—see Section 8.1 for details.

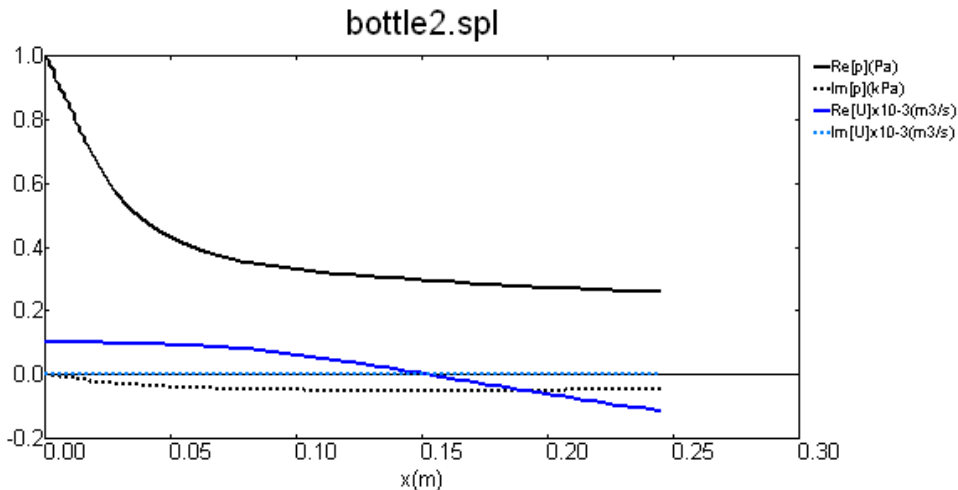


Figure 2.2: State plots of  $p_1$  and  $U_1$  for `<bottle2>`.

- DELTAEC writes two lines of data for lumped elements (e.g., `SURFACE`) and other short segments that do direct end-to-end calculations (e.g., most heat exchangers): one line before and one line after the segment is computed.
- Logistical segments (e.g., `BEGIN`) that do not correspond to any hardware parts generate nothing in the `<.sp>` file.

The file `<bottle2.ssv>` instructed DELTAEC's plotter that we would like to see graphical plots of  $p_1(x)$  and  $U_1(x)$  after each run. These graphs, shown in Fig. 2.2, will automatically refresh after each new run, as the user makes changes to the model.

The real parts of  $p_1$  and  $U_1$  show the values of oscillating pressure and oscillating volume flow at times  $\omega t = 0, 2\pi, 4\pi, \dots$  and the imaginary parts show their values at times  $\omega t = -\pi/2, 2\pi - \pi/2, 4\pi - \pi/2, \dots$

DELTAEC's plotter lets the user select other plots easily. The time-averaged product of  $p_1$  and  $U_1$  is acoustic power  $\dot{E}$ , which is also interesting. To change the display for this model to  $\dot{E}$  vs  $x$ , check the y box under `Edot`, and uncheck the `Re[p]`, `Im[p]`, `Re[U]`, and `Im[U]` boxes, to obtain Fig. 2.3. Chapter 3 explains much more about DELTAEC's plot capabilities.

Now use `Save Copy As` to save this file with a new name, `<bottle3.out>`. Examine the resulting file with a text editor. Note that the output results giving  $p_1$  and  $U_1$  at the ends of the `DUCT`, `CONE`, `DUCT`, and `SURFACE` are now part of the file. The fact that we want to plot  $\dot{E}$  vs  $x$  from now on has also been included in the associated `<bottle3.ssv>` file.

## 2.5. Guesses and targets

The astute reader may have noticed that  $U_1$  was not zero at the `SURFACE` above. (In fact, we left a hole in the bottom of Fig. 2.1(b) to indicate the possibility of this leakage.) This shows that our choice of initial values for  $p_1$  and  $U_1$  at the `BEGIN` was incompatible with a



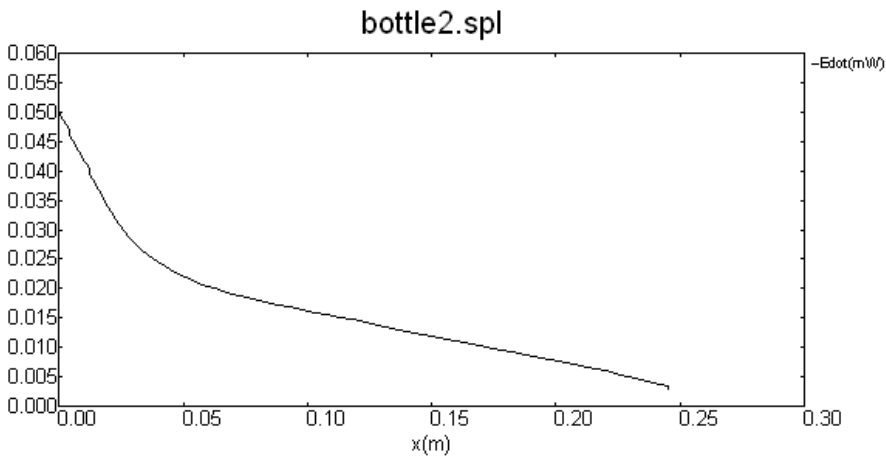
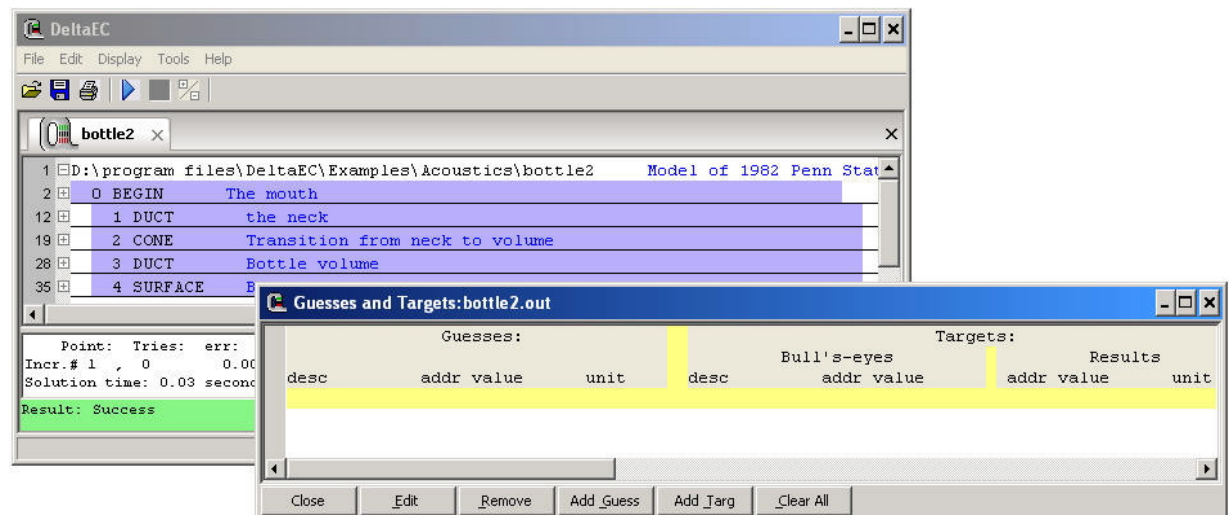


Figure 2.3:  $\dot{E}$  vs  $x$  for <bottle2>.

bottle that is supposed to be sealed at the bottom. In this section we introduce DELTAEC’s shooting method to resolve this issue, allowing boundary conditions at the end of a model (or anywhere else in a model) to be “targeted” to desired values.

Start DELTAEC again, opening file <bottle2.out>. As a matter of good habit, run it, and check some plots for sanity (your own sanity, as well as DELTAEC’s). Now open the “Guesses and Targets” summary display box. You can open this box through the pull-down menus by clicking on Display, then Guesses Targets. Alternatively, just type “g”(the “g” stands for “guesses”). The guess–target summary should look like this:



The table is blank: At this point DELTAEC has no guesses and no targets. Each time we have run DELTAEC thus far in the tutorial, it has integrated along  $x$  only once, starting from the values of  $p_1$  and  $U_1$  that we gave it in the **BEGIN** segment.

To set up this model with appropriate targets, we need to insert an additional segment at the end of the model. To insert a segment, pull down the Edit menu and click on **Insert Seg**, or type “i”, or right click below the last segment and select **Append**. Whichever way

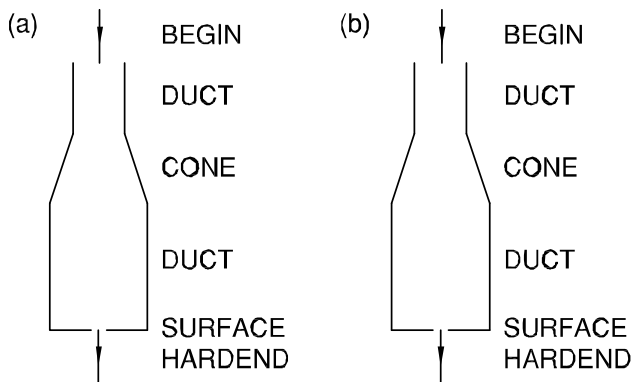
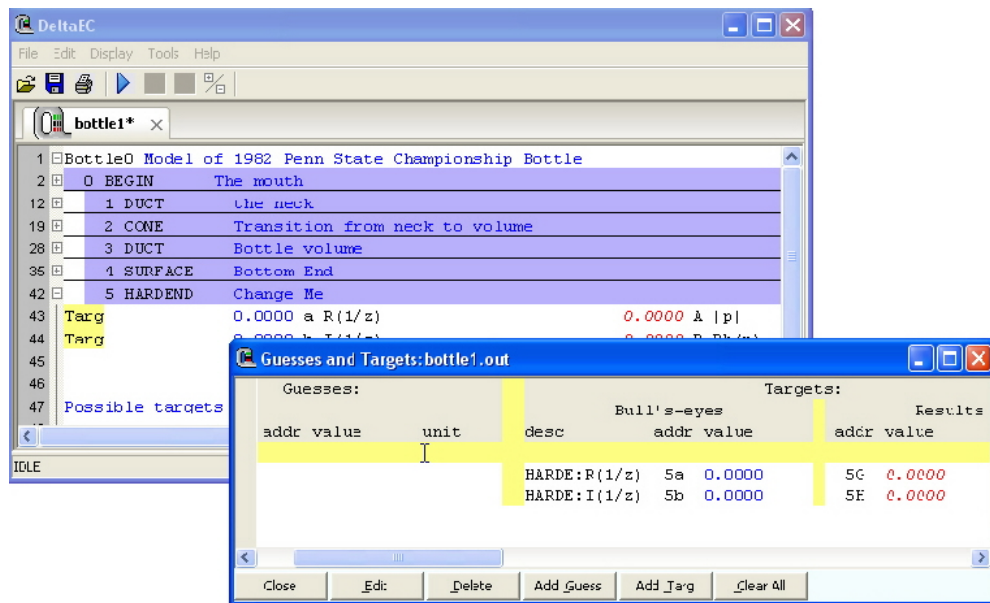


Figure 2.4: The bottle. (a) **HARDEND** untargeted, so  $U_1$  can be nonzero at the bottom. (b) **HARDEND** targeted to  $(0.00, 0.00)$ , to seal the bottom.

you choose to do it, insert a **HARDEND**, at the end of the model, after the **SURFACE**.

Now establish **5a** and **5b** in the **HARDEND** as targets by clicking on **Possible targets** in the **HARDEND** itself or through **Add Targ** in the **guess-target** summary. Inspect your work thus far by examining the **guess-target** summary:



Also notice how the targets are flagged with yellow markers in the main display.

(We could have programmed **HARDEND** segments with the real and imaginary parts of  $U_1$  as our targets, but experience has shown that the DELTAEC shooting-method algorithm usually works without extra attention from us if we normalize the targeted  $U_1$  by dividing by area and  $p_1$ . The rare occasions when more attention is needed are described in Section 8.3.2.)

We have two targets, which will soon enforce complex  $U_1 = 0$  at the bottom end of the model, appropriate for the sealed end of the bottle. We must choose two guesses that

DELTAEC can adjust to meet those targets.<sup>5</sup> Basically, DELTAEC integrates from BEGIN to HARDEND. DELTAEC will refine whatever two guess variables we allow it to vary, to find a solution to this acoustics problem that arrives at the HARDEND with zero complex volume flow rate. What to choose as these guesses depends on exactly what situation you are trying to model. Do you have in mind a resonator driven by a fixed volume flow rate amplitude, or one that is driven by a fixed pressure amplitude? Neither answer is wrong; it depends on the circumstances.

In the present example, we arbitrarily choose to keep  $|U_1|$  fixed at  $0.0001 \text{ m}^3/\text{s}$ , and let DELTAEC adjust  $p_1$ . Begin to accomplish this by adding 0d and 0e as guesses, by double clicking or right clicking on their values and selecting “guess,” or through Add Guess in the guess–target summary. Inspect the resulting situation by looking at the summary.

The screenshot shows the DeltaEC software interface. The main window displays a model setup for a bottle neck, with parameters and guesses listed. The summary window, titled "Guesses and Targets:BOTTLE1.OUT", shows the following data:

Guesses:				Targets:			
desc	addr	val	unit	desc	addr	val	Resu
BEGIN: p	0d	1.0000	Pa	HARDE:R(1/z)	5a	0.0000	5G 0.0000
BEGIN:Ph(p)	0e	0.0000	deg	HARDE:I(1/z)	5b	0.0000	5H 0.0000

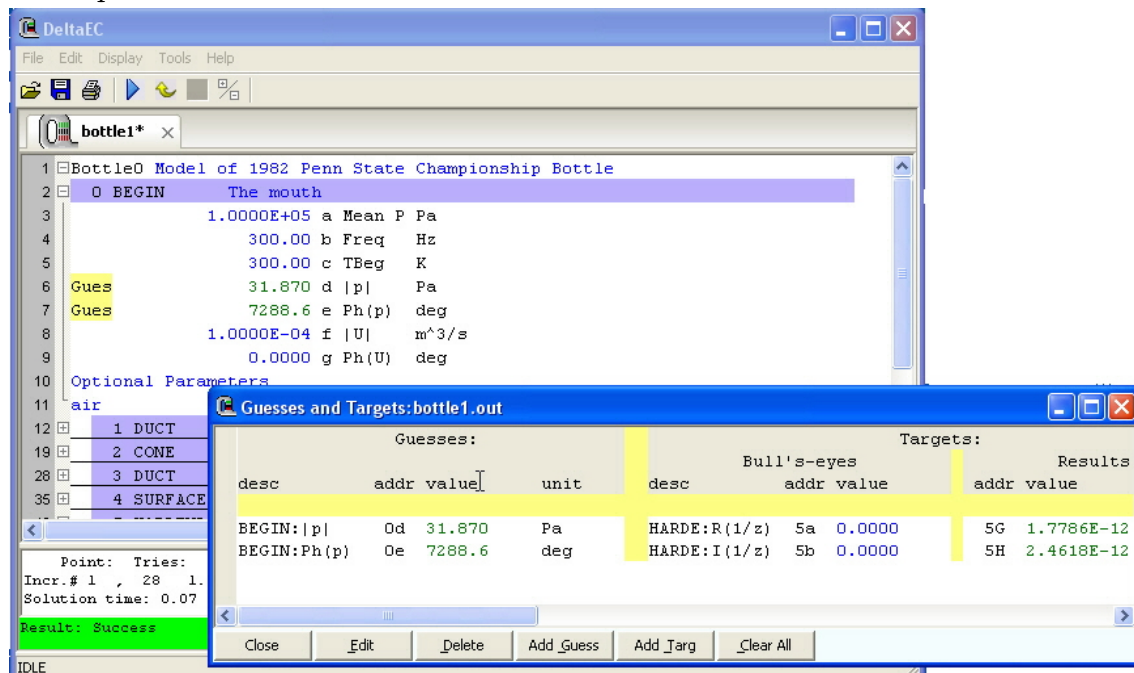
The target values are red italic, indicating that DELTAEC does not know whether results are consistent with input variables; it knows that it has not yet run with the most recent changes of input variables. The guesses (1 Pa, 0 degrees) are still just the values that are left over from before. DELTAEC has not yet adjusted the guesses to try to hit the targets, but it is ready to do so, because there are equal numbers of guesses and targets: two each

<sup>5</sup>Users sometimes ask why we need *two* targets and *two* guesses when we are really only trying to impose *one* constraint at the HARDEND, namely  $|U_1| = 0$ . The answer to this question can be appreciated by imagining a blindfolded archer trying to hit a bull’s-eye. If an observer tells the archer “Your last attempt missed by 2.5 meters,” which way is the archer supposed to revise her aim for her next attempt to hit the target? More useful information is conveyed if the observer tells the archer “Your last attempt struck 1.5 meters above the bull’s-eye and 2.0 meters to the left of the bull’s-eye.” Like the blindfolded archer’s target, DELTAEC’s HARDEND is a point target in a *two*-dimensional space, namely the complex plane of  $U_1$ . DELTAEC needs to adjust its aim (its guesses) in *two* dimensions to hit the target, and DELTAEC can hit the target in a small number of attempts if it gets *two*-dimensional information about how far off-target it is.

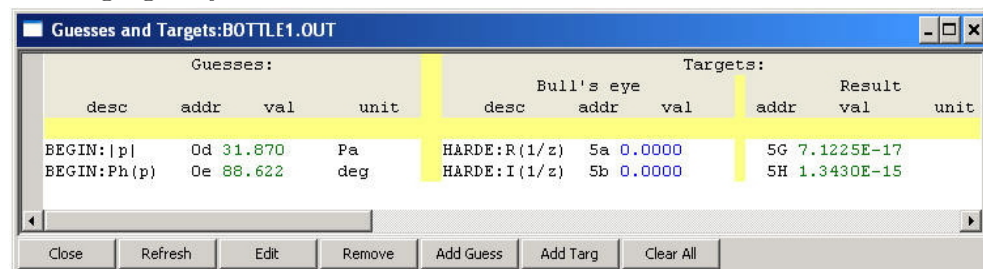
in this case.

With an equal number of guesses and targets, let's run. After running, DELTAEC's Run Monitor at the bottom of the display reports that it integrated along  $x$  28 times while adjusting guesses and coming ever closer to the targets, and it spent 0.08 seconds doing so. (The numbers that are reported by the Run Monitor may vary from computer to computer and version to version. How close to the targets is judged to be "close enough" is described in Section 8.3.2.)

Inspect the results:



The value of the phase shown in 0e, 7288.6°, might seem strange until we realize that DELTAEC does not understand that integer multiples of 360° can be subtracted from any phase angle without consequence. With the initial guessed value of 0d so far from reality, DELTAEC's shooting method wandered far, looking for a solution; under normal circumstances, we might have given it a more plausible starting point (such as the result of a previous run). Subtracting 7200° from the value of 0e for the sake of our human sanity and running again yields



The targets are accurately met, because DELTAEC has adjusted the two guesses to be 31.870 Pa, 88.622 degrees. The state plots in Fig. 2.5 also show that the targets have been met, because they show that  $|U_1| = 0$  at the end.

Save the `<.out>` file with a new filename of your choosing, and examine it with a text

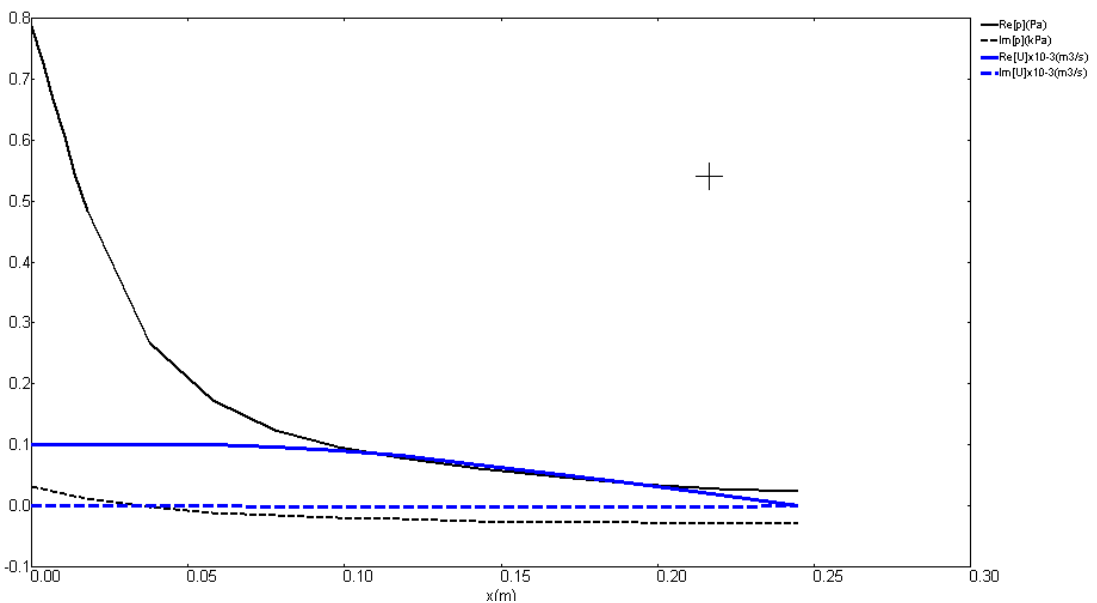


Figure 2.5: State plots for the bottle, with **HARDEND** targets met so  $U_1 = 0$  at the bottom.

editor to see how DELTAEC has indicated which variables are guesses and targets, next to the guesses and targets themselves and in the restart table at the bottom of the file.

Most of the thought required to successfully use DELTAEC occurs while trying to figure out which of the variables are appropriate targets and guesses, i.e., what *constraints* must be imposed on *results* of a single DELTAEC integration pass, and what variables are needed as *inputs* by a single DELTAEC integration pass but are in fact *unknown* to us when we ask DELTAEC to run. While many choices are possible, as long as the number of guesses equals the number of targets, some choices are physically nonsensical and cannot succeed. For example, asking DELTAEC to try to achieve resonance in a given bottle by guessing the mean pressure of the air would be futile, because the speed of sound in an ideal gas is independent of pressure. In the examples in subsequent Chapters, and in advanced use of DELTAEC, the choice of good target and guess variables is seldom as easy as it is in this simple example.

Notice the use of columnar location and color in the display: Columns organize variables according to DELTAEC’s most short-sighted internal needs, while colors organize variables according to users’ needs. The numbers in the left column (ordinary blue numbers, and the guesses, and the bull’s-eye target values) are variables needed by DELTAEC in each pass of its integration, as it integrates repeatedly, adjusting the guesses to try to hit the bull’s-eyes. Variables in the right column are computed as results in each pass, and are displayed green when targets are met. We users, however, often wish to forget that DELTAEC must adjust guesses and hit targets. From our perspective, blue numbers are what we give DELTAEC as known quantities, and green numbers are the results that we want DELTAEC to give us. If the green numbers are indeed green, DELTAEC feels confident of its results, but when those numbers are red DELTAEC is not displaying a consistent set of results, for any of a number of reasons—usually because we have not yet run it or because it has missed the targets.

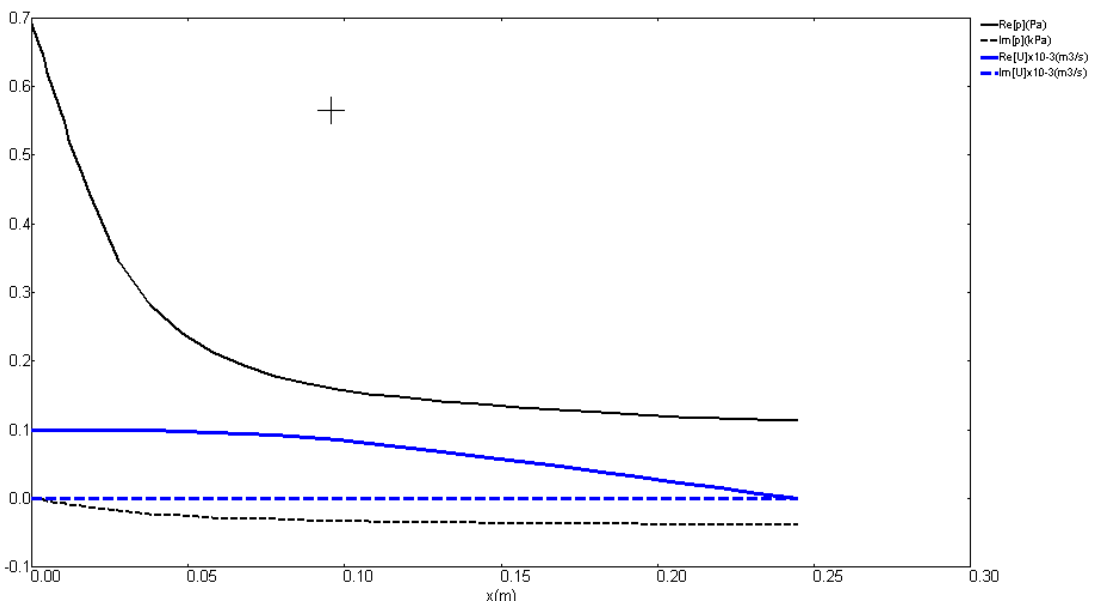


Figure 2.6: State plot with `HARDEND` targeted and resonance achieved, so  $U_1 = 0$  at the bottom and  $p_1$  and  $U_1$  are in phase at the mouth.

Now let’s find the resonance frequency  $f$ , which is the frequency at which the bottle sings when we blow across its top. Look at the `guess-target` summary to get oriented.

```

Guess-Target Summary:
GUESS      0d          0e
name BEGIN: |p|  BEGIN: Ph(p)
units      Pa          deg
value     31.870      88.587
TARGET     5a          5b
name HARDE:R(1/z HARDE:I(1/z
value      0.0000     0.0000
result    -5.6504E-13 4.1801E-14
    
```

First, think about the physics long enough to realize that having  $p_1$  and  $U_1$  in phase at the mouth is a suitable, approximate condition for resonance.<sup>6</sup> Now make the corresponding changes in the `DELTAEC` model: `modify 0e` to be  $0^\circ$  instead of  $41.5^\circ$  and remove `0e` from the guess list so `DELTAEC` cannot change it any more. Next, add `0b`, the frequency, in the guess list. Inspect the `guess-target` summary again. Now, “`run`” the calculation. Inspect the results in the main display or the `guess-target` summary, finding that the resonance frequency is 194.99 Hz. Inspect the state plots in Fig. 2.6, to see that the desired boundary conditions at both ends are met. Save this version as `bottle5`, for use later in this Chapter.

As acousticians, we know that this resonator has more than one resonance mode. To find another one, `modify 0b` to be 600 Hz, and `run` again. With this starting guess, `DELTAEC` converges on the 2nd mode, resembling  $3/4$  of a wavelength from end to end, at 1029.6 Hz.

<sup>6</sup>That short sentence is an example of where novice users of `DELTAEC` who have insufficient acoustics knowledge can have a great deal of trouble. Please do not expect to learn “everything you need to know” about acoustics from `DELTAEC`. Take a university acoustics course or study an acoustics textbook! Remember that `DELTAEC` cannot “understand” physical acoustics better than you do. It merely calculates, accurately and quickly. N.b.: we said “approximate condition for resonance” because some users might want to include radiation impedance. See Section 10.4.1.

DELTAEC can use any physically appropriate input variable as a guess. We could have learned what temperature makes the system resonate at 180.00 Hz, by using 0c instead of 0b as a guess. (The answer is 255 Kelvin.) Or, by using 1c as a guess, we could have found out what length the neck should be to put the resonance at 180 Hz at 300 K.

DELTAEC’s gas library contains three binary mixtures of gases: He-Xe, He-Ar, and Ne-Xe. These introduce another real variable in **BEGIN**: the mole fraction of helium in the mixture (for example, HeAr with  $n_L = 0.889$  indicates 88.9% helium and 11.1% argon). The helium fraction in one of these mixtures can be used as a guess variable for resonance. Simply add 0j as a guess. Thus, we could learn the helium concentration that would have to be added to argon in the bottle to make the resonance equal 180.00 Hz.

## 2.6. Other acoustics features

Here we briefly describe some additional DELTAEC features that are relevant to a simple acoustic apparatus for which  $\mathcal{F}_{\text{momentum}}$  and  $\mathcal{F}_{\text{continuity}}$  in Eqs. (2.6) and (2.7) are the only equations that we want DELTAEC to integrate.

### 2.6.1. Simple segment types

A list of the most commonly used purely acoustic segment types (including those introduced above) is given below. More detailed descriptions and the calculations performed by each are given in Chapter 10 and Chapter 11.

**TITLE** Required at the top.

**BEGIN** Required immediately after **TITLE**. The “zeroth” segment must be **BEGIN**. It contains global variables such as mean pressure, gas type, and frequency, and initial conditions for variables that can vary with  $x$ , including  $p_1$ ,  $U_1$ , and mean temperature (whose  $x$  dependence is introduced in Chapter 5). See details in Section 11.1.2.

**SURFACE** A surface area with pressure-hysteresis loss in its thermal penetration depth. Usually used at ends of ducts. See details in Section 10.2.2.

**DUCT** A duct, with viscous and thermal losses at the wall if the perimeter is realistically nonzero. Separate entry of area and perimeter accommodates ducts of any cross-sectional shape. Laminar or turbulent. See details in Section 10.1.1.

**CONE** A cone, often used to adapt between ducts of different sizes. Uses lossy Webster horn equation. Laminar or turbulent. See details in Section 10.1.2.

**COMPLIANCE** An acoustic compliance, i.e., a volume full of gas. Pressure-hysteresis losses on surface. See details in Section 10.2.1.

**IMPEDANCE** A lumped-parameter, series impedance. Resistance and inertial reactance are specified. See details in Section 10.2.3.

**IDUCER** and **VDUCER** Current-driven and voltage-driven transducers, with parameters independent of frequency, connected “on the side” of the **DUCTS** or other segments. See Fig. 2.7. See details in Section 10.4.2.

**ISPEAKER** and **VSPEAKER** Current-driven and voltage-driven electrodynamic transducers, parameterized by mass,  $B\text{-}l$  product, etc., so impedance coefficients depend on frequency. Connected “on the side” of a sequence of **DUCTS** or other segments. See Fig. 2.7. See details in Section 10.4.2.

**IEDUCER** and **VEDUCER**, **IESPEAKER** and **VESPEAKER** The four transducers that were described above, which do not have “E” as the second letter in their names, are connected in *side-branched* configurations, where the “back side” of the transducer hangs outside of **DELTAEC**’s computation space. The four **Enclosed** versions are their *series* counterparts, with one side of the transducer facing the previous segment and the other side facing the subsequent segment, so the volume flow rate remains constant across the segment (except for thermal-hysteresis effects on the surfaces) and  $p_1$  usually changes significantly. See Fig. 2.8. See details in Section 10.3.1.

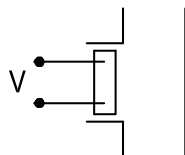


Figure 2.7: **IDUCER**, **VDUCER**, **ISPEAKER**, and **VSPEAKER** are attached as side branches to the series of **DELTAEC** segments.

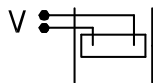
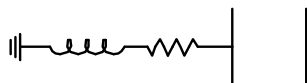


Figure 2.8: **IEDUCER**, **VEDUCER**, **IESPEAKER**, and **VESPEAKER** are acoustically in series with the adjacent **DELTAEC** segments.

**BRANCH** A frequency-independent side-branch impedance. See details in Section 10.4.1.



**OPNBRANCH** and **PISTBRANCH** Frequency-dependent side-branch impedances with the frequency characteristics of the radiation impedance of an opening into  $4\pi$  or  $2\pi$  solid angle, respectively. See details in Section 10.4.1.



**HARDEND** A logistical final segment, used to enforce  $U_1 = 0$  through use of the inverse of the acoustic impedance in the target list. See details in Section 11.1.3.

**SOFTEND** A logistical final segment. Can be used to set  $p_1 = 0$  through use of the acoustic impedance in the target list. Useful for defining a mirror-image plane in symmetric apparatus with a pressure node at a center of symmetry. Also used for loops and networks of segments, to mark a location that will be reconnected elsewhere in the model. See details in Section 11.1.3.

### 2.6.2. The zero of time

The phases of the complex functions  $p_1$  and  $U_1$  encode the time phases of the sinusoidally oscillating parts of  $p(t)$  and  $U(t)$ . Within a DELTAEC model, phase *differences* are vital, but the overall “zero” of phase is arbitrary, for the same reason that the “zero” of time is arbitrary in classical physics. For example, in the <bottle> considered above, any arbitrary phase  $\theta$  can be added to **0e** and **0g**, and the only consequence is that  $\theta$  is added to all **Ph(p)** and **Ph(U)** results, shifting the zero of time by  $-(\theta/360^\circ)(1/f)$ .

The skilled user of DELTAEC should be conscious of where the overall, arbitrary time phase is set in a model. Usually this occurs in **BEGIN**, where **Ph(p)** or **Ph(U)** is fixed at zero or some other value. However, in some models, the phase of a transducer’s electric current or voltage is fixed, and the phase of whichever phasor in **BEGIN** has nonzero amplitude must be guessed. (If one of the phasors has zero amplitude, its time phase is meaningless.)

Human convenience or convention often suggests a wise choice for the zero of phase in a model. In the <bottle>’s <.sp> plots, the real and imaginary parts of  $p_1$  and  $U_1$  are easiest to interpret if the phases of  $p_1$  and  $U_1$  are mostly near  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and/or  $270^\circ$ , rather than near  $45^\circ$ ,  $135^\circ$ , etc. In pulse-tube-refrigerator models, some of us find it most convenient to set **Ph(p) = 0** in the pulse tube itself, or at a nearby pressure sensor, which requires guessing one or both of the **BEGIN** phases and using an RPN target (see Chapter 4) to enforce the desired condition. When a phase must be targeted, avoid doing so near  $180^\circ$ , because target results must be smooth functions of guesses, as explained at the end of Section 8.5.

### 2.6.3. Gases and solids

DELTAEC recognizes many gases<sup>7</sup> and solids, and routinely calculates their properties such as density, specific heat, and viscosity. Gases are specified in **BEGIN** segments, and solids are specified in each physically realistic segment. As described in detail in Chapter 12, DELTAEC includes the following “gases”: dry air, humid air and fog, helium, neon, He-Xe mixtures, He-Ar mixtures, Ne-Xe mixtures, hydrogen, deuterium, nitrogen, carbon dioxide,

---

<sup>7</sup>A “fluid” is a gas or a liquid. Gases are the most commonly used fluids in thermoacoustics and DELTAEC. However, liquid sodium and eutectic liquid sodium-potassium are included in DELTAEC because they are thermodynamically powerful, with non-negligible thermal expansion coefficients and non-negligible  $(\partial s/\partial p)_T$ , when close enough to their critical points. For further details on the use of these liquids in thermoacoustics, see Ref. [14]. Perhaps we should refer to the “gases” used in DELTAEC as “fluids,” but for simplicity we always call them gases despite the fact that some liquids are included.

natural-gas combustion products (i.e., flue gas), liquid sodium, and eutectic liquid sodium-potassium. The ideal-gas equation of state is used for most of the gases, while carbon dioxide, sodium, and eutectic sodium-potassium have their own equations of state.

Solids, described in detail in Chapter 13, include Kapton, Mylar, stainless steel, molybdenum, tungsten, copper, nickel, Celcor, and ideal. An ideal solid has extremely high heat capacity, density, and thermal conductivity.

DELTAEC also allows users to specify “external,” user-defined gases or solids that are not part of its own internal library. Properties are calculated, according to local  $T_m$  and  $p_m$ , from coefficients read from a user-written text file. Up to five user-defined gases and five user-defined solids can be used at one time. Details of user-defined pure gases and binary mixtures are described in Sections 12.15.1 and 12.15.2, respectively. Details of user-defined solids are described in Section 13.10.

To illustrate the use of a single-component user-defined gas, consider the example below. To replace DELTAEC’s internal “ideal” helium gas (see Section 12.1) with a more accurate representation that calculates density and sound speed including the first coefficient of the virial expansion for helium, we can create the following file in a text editor, call it `<helium.tpf>`, and put it in the same folder as our model:

```
! user-defined gas; He with first virial coeff (B=12cc/mole)
! Equation is in UsersGuide Section 12.15:
! C0 + C1*pm/(T+C2*pm) + C3*T + C4*T^2 + C5*T^C6 + pm^2 *C7*T^C8 + pm*C9
! Density, rho (m^3):
0. 4.814e-4 1.44e-6
! isobaric heat capacity, cp (J/kg-K):
5192.
! Thermal conductivity, k (W/m-K):
0. 0. 0. 0. 0. 0.0025672 0.716
! Square of sound speed, a^2 (m^2/s^2):
0. 0. 0. 3461.92 0. 0. 0. 0. 0. .0100
! Viscosity, mu (kg/m-s):
0. 0. 0. 0. 0. 0.412e-6 0.68014
```

The coefficients for density in this file are used by DELTAEC in Eq. (12.1) and were determined by a user who knew that

$$\rho_m = \frac{p_m m}{R_{\text{univ}}(T_m + B p_m / R_{\text{univ}})}, \quad (2.9)$$

where  $R_{\text{univ}} = 8.314 \text{ J/mole-K}$ , the molar mass  $m = 0.0040026 \text{ kg/mole}$ , and the first virial coefficient  $B = 1.2 \times 10^{-5} \text{ m}^3/\text{mole}$ . Thus, the user set  $C_1 = m/R_{\text{univ}}$  and  $C_2 = B/R_{\text{univ}}$  in the density line in the `<.tpf>` file. For squared sound speed, the user knew that

$$a^2 = \frac{\gamma R_{\text{univ}} T}{m} \left( 1 + 2 \frac{B p_m}{R_{\text{univ}} T_m} \right), \quad (2.10)$$

yielding  $C_3 = \gamma R_{\text{univ}}/m$ , and  $C_9 = 2B\gamma/m$  in the square-of-sound-speed line in the `<.tpf>` file, where  $\gamma = 5/3$ .

User-defined gas mixtures (described in Section 12.15.2) and user-defined solids (described in Section 13.10) follow similar formats.

It is a good idea to check your implementation of a new user-defined gas or solid very carefully by using one of the two ways of displaying or plotting thermophysical properties described in Section 2.7.7.

## 2.6.4. Turbulence

A turbulence algorithm can be enabled in `DUCT` and `CONE`, by double clicking on “Optional parameters.” Checking the Turbulence box brings up parameter `d` in `DUCT` and parameter `f` in `CONE`, which specify the relative roughness (roughness height divided by pipe diameter). Set the roughness equal to a small value greater than zero for rough walls. (We usually use  $5 \times 10^{-4}$ , even if this is larger than the actual relative roughness of the channel.) To ensure a laminar calculation, check the laminar box for the segment (which causes the roughness parameter to be hidden from view in the `<.out>` file).

The turbulence algorithm, which is described in detail in Section 10.1.1, follows the quasi-steady approximation, (the spirit of the assumptions of Iguchi et al., Ref. [15]), assuming that oscillatory-flow losses can be calculated by using the steady-flow Moody friction factor at each instant of time during the oscillatory flow, with the surface roughness used as an experimentally determined fitting parameter. The quasi-steady assumption has only a little experimental validation for `DUCTs` and `CONEs` in the range of Reynolds number and  $R/\delta_\nu$  of interest in thermoacoustics (where  $R$  is radius), but we believe it provides a useful estimate, much better than no estimate at all.

## 2.7. Other useful features

### 2.7.1. Keeping parameters equal

The `sameas nl` feature ties an input parameter in one segment to a value elsewhere. This helps prevent typographical errors in the `<.out>` file, and is especially useful in linking dimensions of adjacent segments that you might want to vary all together while modifying or plotting, such as areas of adjacent segments when increasing the size of the apparatus. Specify the segment number and line letter (e.g., “`sameas 3a`”).

The following example is for the bottle we modeled above:

```
!----- 1 -----
DUCT      The neck
2.141e-4  a Area      m
5.187E-2  b Perim     m
1.778E-2  c Length   m
ideal     Solid type
!----- 2 -----
CONE      transition from neck to volume
sameas 1a a AreaI    m^2
sameas 1b b PerimI   m
          c Length   m
          .100
sameas 3a d AreaF    m^2
sameas 3b e PerimF   m
ideal     Solid type
!----- 3 -----
DUCT      Bottle volume
1.868e-3  a Area      m^2
0.1532    b Perim     m
0.1270    c Length   m
ideal     Solid type
!----- 4 -----
SURFACE   Second End
sameas 3a a Area      m^2
ideal     Solid type
```



When you add a parameter specified by `sameas` to the guess list, you must first sever the `sameas` relationship. This is required because the value at this point will be controlled by DELTAEC’s shooting method. But if a variable that is the root of one or several `sameas`

references is caused to change by DELTAEC itself in any way, e.g., as a guess, all `sameas` references to this root within the model will change with it.

### 2.7.2. Master–slave links

When geometric variables are changed, whether by the user, by DELTAEC’s shooting method, or by DELTAEC’s incremental plot routine, there are often geometric relationships with other parameters that should be maintained. For example, if the area of a duct increases, we should usually increase the associated perimeter as well. Another common wish is to lengthen one segment while simultaneously shortening another segment to keep overall length constant. Master–slave links are available in some segments for just these purposes; a Master-Slave Links option appears in a segment whenever a built-in parameter-linking capability is possible for variables in that segment.

For example, consider a DUCT whose cross section is a half circle attached to a square, as shown in Fig. 2.9. The cross-sectional area is

$$A = \frac{1}{2} \frac{\pi}{4} d^2 + d^2, \quad (2.11)$$

and the perimeter is

$$\Pi = \frac{\pi}{2} d + 3d. \quad (2.12)$$

Notice that

$$c = \frac{\Pi^2}{A} = \frac{(\pi + 6)^2}{\pi/2 + 4} \simeq 15.001 \quad (2.13)$$

is a constant, independent of the size  $d$  of the DUCT, but depending on the half-circle-plus-square character of its shape. Additional examples are  $c = 4\pi$  for a circular duct,  $c = 16$  for a square duct. When perimeter–area linking is established in a DUCT, DELTAEC remembers the value of this constant  $c$ , and uses it thereafter in

$$\Pi = \sqrt{cA} \quad (2.14)$$

to update the perimeter based on the area, thereby maintaining the cross-sectional shape. Thus, as area is changed, this link keeps a circular duct circular and maintains the aspect ratio of a rectangular duct.

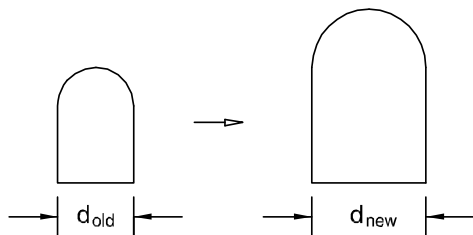


Figure 2.9: When the area of a DUCT is changed and perimeter–area linking is in use, the perimeter is automatically recalculated to maintain the DUCT’s shape.

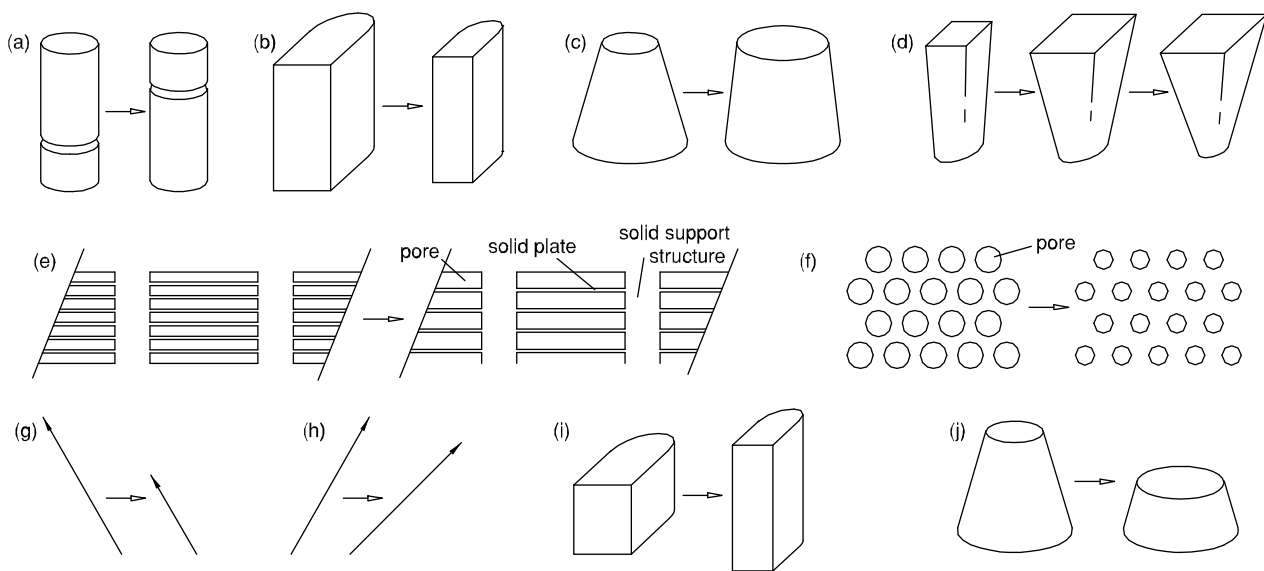


Figure 2.10: Illustration of some master–slave linking. (a) Two lengths can be linked so the total length remains constant when one length is changed. (b), (c), (d) Perimeters can be linked to areas so shapes remain constant when areas are changed. (e) In **STKSLAB**, porosity can be linked to pore size and solid thickness, to keep the fraction of the stack that’s devoted to support structure constant. The plane of this illustration is perpendicular to  $x$ . (f) In **STKCIRC** and **TX**, porosity can be linked to hole radius, to keep the number of holes constant. The plane of this illustration is perpendicular to  $x$ . (g), (h) In segments with impedances, the imaginary part can be linked to the real part so either phase or magnitude remains constant when the real part is changed. (i) In **STKDUCT**, maintain shape and volume when length is changed. (j) In **CONE**, adjust length to maintain taper angle when areas are changed.

Below is a list of all parameter-link options and the segment types for which they are available, including some for segments that have not yet been introduced in this tutorial. The numbers in parentheses at the end indicate how the link is recorded in the restart table at the bottom of the `<.out>` file. Figure 2.10 illustrates these links.

Adjust length in segment  $n$  when the length here changes, to keep their sum constant: All segments with length. (In the restart table, this is indicated as  $n > 0$ .)

Adjust porosity (i.e., **GasA/A**) when gas gap or solid thickness changes in **STKSLAB**, keeping constant area and constant fraction of the area devoted to support ribs. Adjust porosity (i.e., **GasA/A**) when hole radius or area changes in **STKCIRC** or **TX**, keeping constant number of holes. Set porosity (i.e., **GasA/A**) equal to  $ab/(a+l)(b+l)$  in **STKRECT**. For details, see Section 10.5.1. (In the restart table, this is indicated as  $-1$ .)

Adjust perimeter when area changes, to maintain shape: **DUCT** and **STKDUCT**. Adjust initial perimeter when initial area changes, to maintain initial end’s shape: **CONE** and **STKCONE**. (In the restart table, this is indicated as  $-2$ .)

Adjust final perimeter when final area changes, to maintain shape of final end: **CONE** and **STKCONE**. (In the restart table, this is indicated as  $-3$ .)

Adjust length and both perimeters, to maintain end shapes and taper angle: **CONE** and **STKCONE**. (In the restart table, this is indicated as  $-4$ .)

Adjust surface area when volume changes, to maintain shape: **COMPLIANCE**. (In the restart table, this is indicated as  $-5$ .)

Adjust area and perimeter when length changes, to maintain constant volume and cross-sectional shape: **STKDUCT**. (In the restart table, this is indicated as  $-6$ .)

Adjust imaginary part when real part changes, to maintain constant magnitude (when possible): **IMPEDANCE**, **BRANCH**, and **TBRANCH**. (In the restart table, this is indicated as  $-7$ .)

Adjust imaginary part when real part changes, to maintain constant phase angle: **IMPEDANCE**, **BRANCH**, and **TBRANCH**. (In the restart table, this is indicated as  $-8$ .)

Adjust initial and final perimeters in **CONE** and **STKCONE** as initial and final areas change, maintaining the shapes on both ends. (In the restart table, this is indicated as  $-9$ .)

Master-slave links can be tracked by status fields in the center of the `<.out>` file, between the input and output columns. The master parameter—the one through which the mode is controlled—is labeled **Mstr**. The slave parameter, i.e., the one that cannot be modified independently when it is controlled by a link, has a status indicator of the form *nnp* to indicate its master's address (e.g., 2a).

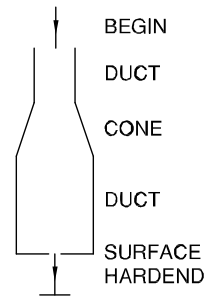
Newly inserted **DUCT** and **CONE** segments automatically initialize with their perimeters linked to their areas on the basis of a circular shape, with a radius of 1 meter.

With each of the `<bottle>` model's perimeters slaved to its local area, and adjacent areas that must be identical connected by **sameas**, the `<bottle>` model looks like this:

```

2 0 BEGIN Initial
12 1 DUCT the neck
13      2.1410E-04 a Area m^2 Mstr 12.044 A |p| Pa
14      5.1870E-02 b Perim m 1a -87.986 B Ph(p) deg
15      1.7780E-02 c Length m 9.9797E-05 C |U| m^3/s
16 Master-Slave Links -9.4266E-03 D Ph(U) deg
17 Optional Parameters 3.3197E-05 E Htot W
18 ideal Solid type 2.1219E-05 F Edot W
19 2 CONE Transition from neck to volume
20 Same 1a 2.1410E-04 a AreaI m^2 33.945 A |p| Pa
21 Same 1b 5.1870E-02 b PerimI m -89.794 B Ph(p) deg
22      0.1003 c Length m 7.5895E-05 C |U| m^3/s
23 Same 3a 1.8680E-03 d AreaF m^2 -3.4905E-02 D Ph(U) deg
24 Same 3b 0.1532 e PerimF m 3.3197E-05 E Htot W
25 Master-Slave Links 5.4209E-06 F Edot W
26 Optional Parameters
27 ideal Solid type
28 3 DUCT Bottle volume
29      1.8680E-03 a Area m^2 Mstr 37.703 A |p| Pa
30      0.1532 b Perim m 3a -89.859 B Ph(p) deg
31      0.1270 c Length m 2.3620E-08 C |U| m^3/s
32 Master-Slave Links -89.859 D Ph(U) deg
33 Optional Parameters 3.3197E-05 E Htot W
34 ideal Solid type 4.4527E-07 F Edot W
35 4 SURFACE Bottom End
36 Same 3a 1.8680E-03 a Area m^2 37.703 A |p| Pa
37      -89.859 B Ph(p) deg

```



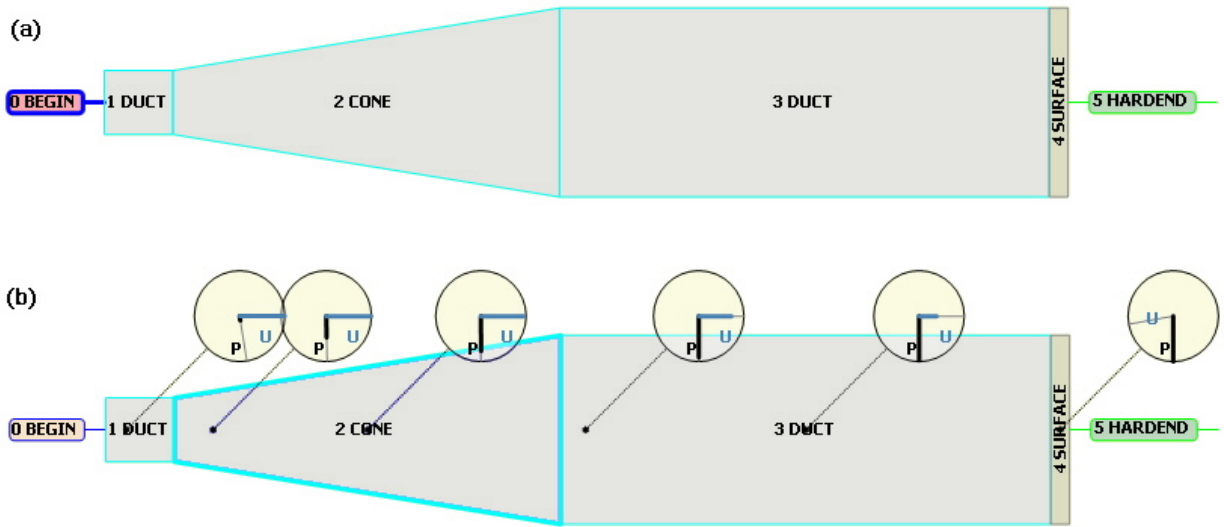


Figure 2.11: (a) The schematic display of the `<bottle>` model. Dimensions are to scale, assuming that areas are circular. Right-clicking on any segment brings up a menu of options, similar to right-clicking on a segment in the main model display. (b) Some phasor diagrams added to the schematic.

so it is apparent that the bottle has only five independent geometrical dimensions—neck area, bulb area, and the lengths of the neck, cone, and bulb—whose independence is indicated by their blue color. All other dimensions depend on those five. The user who is interested in studying the resonance frequency as a function of geometry can now change those five dimensions at will, confident that self-consistent geometry will be maintained by the master-slave and `sameas` links.

### 2.7.3. Schematic view

A schematic view of the geometry of a DELTAEC model can be turned on by clicking on “view schematic” in the Display pulldown menu. As an example, the schematic of the `<bottle>` model is shown in Fig. 2.11. The schematic is to scale: Dimensions in the  $x$  direction are proportional to the lengths of segments with length, and dimensions perpendicular to  $x$  are proportional to  $\sqrt{4A/\pi}$  in segments having area and length. A glance at this display can help you catch order-of-magnitude typographical errors in a model and keep track of the topology of large models. Left click and drag up to zoom in for a closer view, and right click to bring up a menu of segment operations similar to the right-click segment menu in the main model.

When both a schematic view and a state plot are active, phasor information from the state plot can be displayed on the schematic, based on the position of the mouse cursor within the schematic. Click Alt when the cursor is over the desired position in a highlighted physical segment in the schematic, to create a small phasor plot showing both  $p_1$  and  $U_1$  at that position. The magnitude of each phasor is scaled by the maximum magnitude found in the model. The colors are the same as the default colors in the state plot: black for  $p_1$ , blue

for  $U_1$ . Drag the phasor circle or its anchor point with the mouse to reposition them. In integrable segments like DUCTs, Nint unique  $x$  locations are available; nonintegrable segments like COMPLIANCE have only one location.

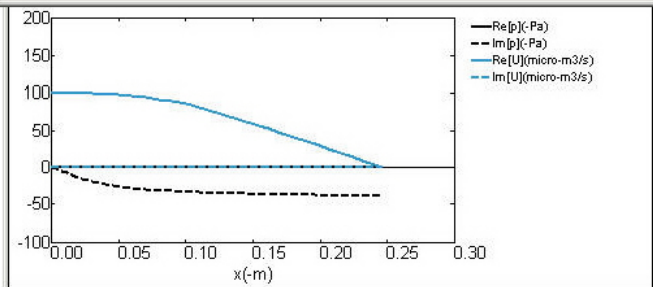
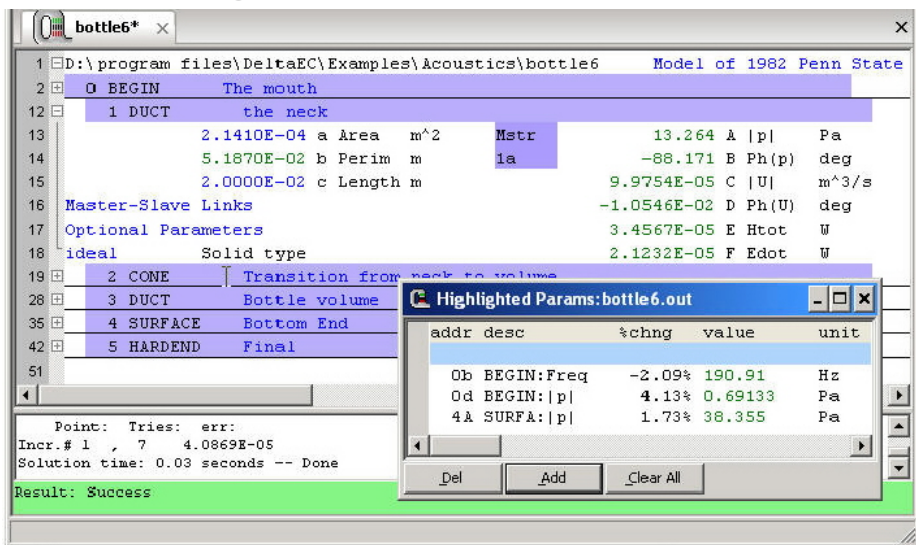
To delete a phasor circle, move the mouse over the anchor point and press the Alt or Delete key. To delete all phasor circles, close and reopen the schematic window.

### 2.7.4. Highlights display

DELTAEC models often have many dozens of segments. The user is usually not interested in examining *every* numerical result after every run—typically, we examine the guesses (and perhaps targets) to ensure that DELTAEC has not gotten completely lost, and we examine a small number of other results of particular interest. The highlights display shows a compact, user-customized list of such variables of particular interest, which is updated after every run.

Any input or output variables can be added to the highlight list, by double clicking or right clicking.

Returning to the bottle as an example, we might be particularly interested in the pressure amplitude at the mouth of the bottle and at the bottom of the bottle, and the resonance frequency, as we explore different dimensions of the neck by modifying its area and/or length and running. A convenient setup for such an investigation looks something like the figure below, just after lengthening the neck and running. In ordinary use, we would type ‘m’ to modify one or more parameters, then ‘r’ to run, examine the results, and repeat.





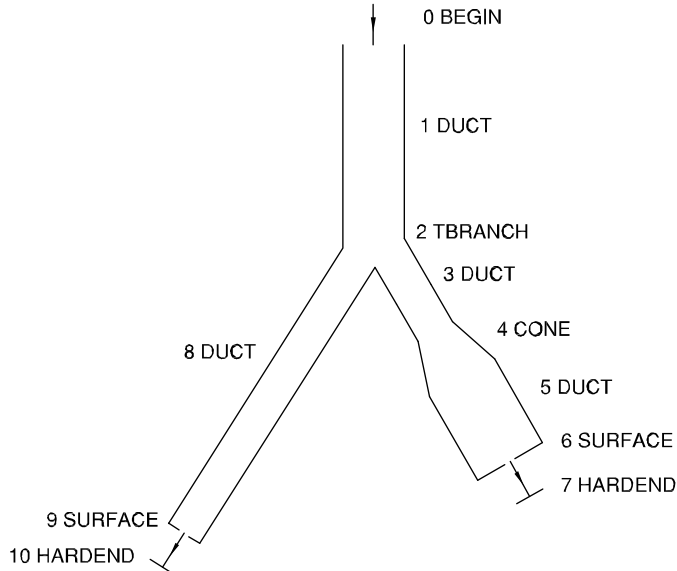


Figure 2.12: TBRANCH allows DELTAEC to integrate in branched geometries like this.

### 2.7.5. Structured branches, loops, and networks

Although BRANCH and OPNBRANCH can be useful, they are often inadequate for describing the variations in branch impedance with operating conditions. For example, the branch might be a Helmholtz resonator whose impedance changes significantly with frequency. Further, BRANCH and OPNBRANCH are wholly inadequate when networks include reconnecting topology, as in Fig. 1.3. The TBRANCH and UNION segments overcome these inadequacies.

When it encounters a TBRANCH, DELTAEC treats subsequent segments as the sequential members of the branch, integrating until it reaches a HARDEND or SOFTEND. It then “returns to the trunk,” treating further segments as trunk members. The logic of this is shown in Fig. 2.12. At the TBRANCH, the branch impedance determines how the (complex) volume flow rate splits up. Usually we use the branch impedance as a pair of guesses that DELTAEC adjusts in its usual way to hit two targets elsewhere. TBRANCHED models tend to have many guesses and targets, since every \*\*END contributes two targets (and a few more targets are almost always needed for temperatures and other variables). Of course, branches can have sub-branches of their own.

Networks with loops, like Fig. 1.3, can also be handled by DELTAEC, through use of TBRANCH and UNION. The UNION segment is used to tell DELTAEC to “connect” a TBRANCH’s SOFTEND back to the trunk at the location of the UNION segment. The branch’s SOFTEND potential targets are not used. However, at least two input variables (b and c) of the UNION segment should always be used as targets. As DELTAEC processes the UNION segment, it assigns the current values of the complex pressure at the SOFTEND referenced by the number in parameter a of the UNION segment. These values are compared to the local complex pressure result, at this UNION, in the trunk, and DELTAEC’s shooting-method iteration should run until their difference is zero. This assures that the end of the branch and the trunk share the same complex  $p_1$  where they meet at the UNION. As before, a guessed branch

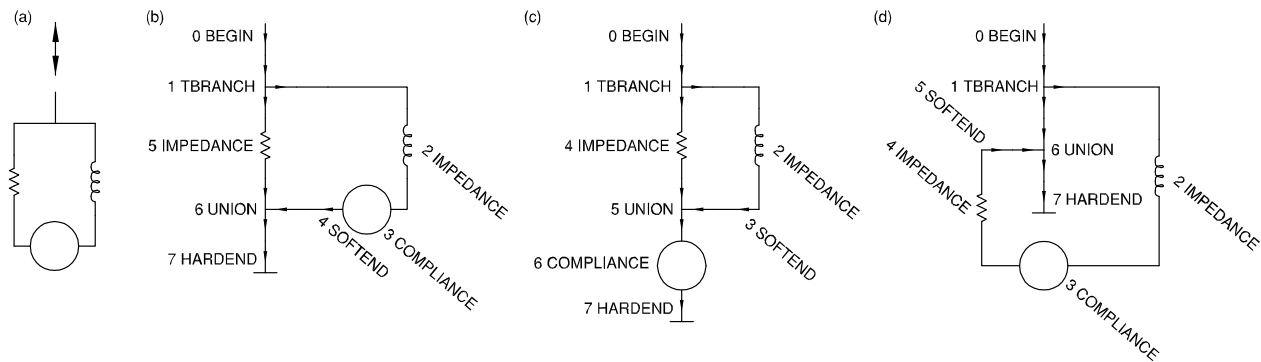


Figure 2.13: A driven, parallel LRC network. (a) Schematic. (b) One way to model the LRC network in DELTAEC, with the L and C in the TBRANCH and the R in the trunk, rejoined at the UNION, whose downstream end must be sealed by a targeted HARDEND. (c) Another way to model the LRC, with the L in the TBRANCH and the R in the trunk, rejoined at the UNION, whose downstream end feeds the C, whose exit must be sealed by a targeted HARDEND. (d) A third way, with the L, R, and C all in the TBRANCH. Returning to the trunk, DELTAEC encounters no segment before the UNION, so the UNION effectively connects the initial and final ends of the branch to each other. The UNION’s exit must be sealed by a targeted HARDEND.

impedance usually determines how the (complex) volume flow rate splits up at the TBRANCH. Volume flow rates are summed at the UNION. (Note that the UNION targets are somewhat different from HARDEND targets discussed above, because “target” numerical values in UNION are not specified by the user—they vary from run to run, depending on what is happening at the associated SOFTEND.)

Figure 2.13 and the DELTAEC models below show three ways to use TBRANCH and UNION to model a parallel LRC network. The reader is encouraged to invent one or more additional ways to model this network, and try them.

Here is the DELTAEC model corresponding to Fig. 2.13(b):

```

TITLE      Example LRC1
!_lrc1.out
!Created@19:30:01  9-Jul-06 with DeltaE Vers. 5.5b6 for the IBM/PC-Compatible
----- 0 -----
BEGIN      Initialize things
1.0000E+05 a Mean P      Pa      6.1460E-03 A |U|      G( 0f)      P
60.000    b Freq      Hz      80.940    B Ph(U)      G( 0g)      P
300.00    c TBeg      K      3.7260E+05 C Re(Zb)      G( 1a)      P
2000.0    d |p|      Pa      -2.7012E+05 D Im(Zb)      G( 1b)      P
0.0000    e Ph(p)      deg
6.1460E-03 f |U|      m^3/s      G
80.940    g Ph(U)      deg      G
air      Gas type
----- 1 -----
TBRANCH    the split
3.7260E+05 a Re(Zb) Pa-s/m^3      G      2000.0    A |p|      Pa
-2.7012E+05 b Im(Zb) Pa-s/m^3      G      0.0000    B Ph(p)      deg
4.3459E-03 C |U|      m^3/s
35.940    D Ph(U)      deg
3.5185    E Hdot      W
3.5185    F Edot      W
-2.5508    G EdotTr      W
----- 2 -----
IMPEDANCE  pure inertance
0.0000    a Re(Zs) Pa-s/m^3      2282.4    A |p|      Pa
1.0000E+05 b Im(Zs) Pa-s/m^3      -8.8682   B Ph(p)      deg
4.3459E-03 C |U|      m^3/s
35.940    D Ph(U)      deg
3.5185    E Hdot      W

```

```

----- 3 -----
COMPLIANCE      a compliance tank
4.8360E-02 a SurfAr      m^2 Fn( 3b)  2282.4      A |p|      Pa
1.0000E-03 b Volume      m^3      S= -5      -8.8682    B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           -54.060    D Ph(U)    deg
                                           3.4951     E Hdot     W
ideal          Solid type
                                           3.4951     F Edot     W
----- 4 -----
SOFTEND         this will reconnect at the UNION
0.0000 a Re(z)          (t)      2282.4      A |p|      Pa
0.0000 b Im(z)          (t)      -8.8682    B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           -54.060    D Ph(U)    deg
                                           3.4951     E Hdot     W
                                           3.4951     F Edot     W
                                           44.392     G Re(z)
                                           44.690     H Im(z)
----- 5 -----
IMPEDANCE      pure resistance
1.0000E+05 a Re(Zs) Pa-s/m^3      2282.4      A |p|      Pa
0.0000 b Im(Zs) Pa-s/m^3      -8.8682    B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           125.94     D Ph(U)    deg
                                           -3.4951    E Hdot     W
                                           -3.4951    F Edot     W
----- 6 -----
UNION          close the loop here
4.0000 a SegNum          2282.4      A |p|      Pa
2282.4 b |p|End Pa      = 6A?      -8.8682    B Ph(p)    deg
-8.8682 c Ph(p)E deg    = 6B?      7.0901E-17 C |U|      m^3/s
                                           -82.266    D Ph(U)    deg
                                           2.3118E-14 E Hdot     W
----- 7 -----
HARDEND        seal the final end of the resonator
0.0000 a R(1/z)          = 7G?      2282.4      A |p|      Pa
0.0000 b I(1/z)          = 7H?      -8.8682    B Ph(p)    deg
                                           7.0901E-17 C |U|      m^3/s
                                           -82.266    D Ph(U)    deg
                                           2.3118E-14 E Hdot     W
                                           2.3118E-14 F Edot     W
                                           7.4001E-17 G R(1/z)

```

And here is the DELTAEC model corresponding to Fig. 2.13(c):

```

TITLE          Example LRC2
lrc2.out
!Created@19:31:13 9-Jul-06 with DeltaE Vers. 5.5b6 for the IBM/PC-Compatible
----- 0 -----
BEGIN          Initialize things
1.0000E+05 a Mean P      Pa      6.1460E-03 A |U| G( 0f) P
60.000 b Freq          Hz      80.940 B Ph(U) G( 0g) P
300.00 c TBeg          K      3.7260E+05 C Re(Zb) G( 1a) P
2000.0 d |p|          Pa      -2.7012E+05 D Im(Zb) G( 1b) P
0.0000 e Ph(p)          deg
6.1460E-03 f |U|          m^3/s G
80.940 g Ph(U)          deg G
air           Gas type
----- 1 -----
TBRANCH        the split
3.7260E+05 a Re(Zb) Pa-s/m^3 G      2000.0      A |p|      Pa
-2.7012E+05 b Im(Zb) Pa-s/m^3 G      0.0000     B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           35.940    D Ph(U)    deg
                                           3.5185    E Hdot     W
                                           3.5185    F Edot     W
                                           -2.5508   G EdotTr  W
----- 2 -----
IMPEDANCE      pure inertance
0.0000 a Re(Zs) Pa-s/m^3      2282.4      A |p|      Pa
1.0000E+05 b Im(Zs) Pa-s/m^3      -8.8682    B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           35.940    D Ph(U)    deg
                                           3.5185    E Hdot     W
                                           3.5185    F Edot     W
----- 3 -----
SOFTEND        this will reconnect at the UNION
0.0000 a Re(z)          (t)      2282.4      A |p|      Pa
0.0000 b Im(z)          (t)      -8.8682    B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           35.940    D Ph(U)    deg
                                           3.5185    E Hdot     W
                                           3.5185    F Edot     W

```

```

----- 4 -----
                                44.690  G Re(z)
IMPEDANCE pure resistance
1.0000E+05 a Re(Zs) Pa-s/m^3      2282.4  A |p|      Pa
0.0000  b Im(Zs) Pa-s/m^3      -8.8682 B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           125.94  D Ph(U)    deg
                                           -3.4951 E Hdot     W
                                           -3.4951 F Edot     W
----- 5 -----
UNION close the loop here
3.0000 a SegNum      2282.4  A |p|      Pa
2282.4 b |p|End      Pa = 5A?    -8.8682 B Ph(p)    deg
-8.8682 c Ph(p)E    deg = 5B?    6.1460E-03 C |U|      m^3/s
                                           80.940  D Ph(U)    deg
                                           2.3429E-02 E Hdot     W
                                           2.3429E-02 F Edot     W
                                           300.00  G End-T     K
----- 6 -----
COMPLIANCE a compliance tank
4.8360E-02 a SurfAr  m^2  Fn( 6b) 2282.4  A |p|      Pa
1.0000E-03 b Volume  m^3  S= -5   -8.8682 B Ph(p)    deg
                                           2.8686E-14 C |U|      m^3/s
                                           119.89  D Ph(U)    deg
                                           -2.0495E-11 E Hdot     W
ideal Solid type      -2.0495E-11 F Edot     W
----- 7 -----
HARDEND seal the final end of the resonator
0.0000 a R(1/z)      = 7G?    2282.4  A |p|      Pa
0.0000 b I(1/z)      = 7H?    -8.8682 B Ph(p)    deg
                                           2.8686E-14 C |U|      m^3/s
                                           119.89  D Ph(U)    deg
                                           -2.0495E-11 E Hdot     W
                                           -2.0495E-11 F Edot     W
                                           -6.5606E-14 G R(1/z)

```

And finally, here is the DELTAEC model corresponding to Fig. 2.13(d):

```

TITLE Example 1: Plane-wave resonator
lrc3.out
!Created@19:32:21 9-Jul-06 with DeltaE Vers. 5.5b6 for the IBM/PC-Compatible
----- 0 -----
BEGIN Initialize things
1.0000E+05 a Mean P Pa      6.1460E-03 A |U| G( 0f) P
60.000  b Freq Hz      80.940  B Ph(U) G( 0g) P
300.00  c TBeg K      3.7260E+05 C Re(Zb) G( 1a) P
2000.0  d |p| Pa      -2.7012E+05 D Im(Zb) G( 1b) P
0.0000  e Ph(p) deg
6.1460E-03 f |U| m^3/s G
80.940  g Ph(U) deg G
air Gas type
----- 1 -----
TBRANCH the split
3.7260E+05 a Re(Zb) Pa-s/m^3 G 2000.0  A |p|      Pa
-2.7012E+05 b Im(Zb) Pa-s/m^3 G 0.0000  B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           35.940  D Ph(U)    deg
                                           3.5185  E Hdot     W
                                           3.5185  F Edot     W
                                           -2.5508 G EdotTr   W
----- 2 -----
IMPEDANCE pure inertance
0.0000 a Re(Zs) Pa-s/m^3      2282.4  A |p|      Pa
1.0000E+05 b Im(Zs) Pa-s/m^3      -8.8682 B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           35.940  D Ph(U)    deg
                                           3.5185  E Hdot     W
                                           3.5185  F Edot     W
----- 3 -----
COMPL a compliance tank
4.8360E-02 a SurfAr  m^2  Fn( 3b) 2282.4  A |p|      Pa
1.0000E-03 b Volume  m^3  S= -5   -8.8682 B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           -54.060  D Ph(U)    deg
                                           3.4951  E Hdot     W
ideal Solid type      3.4951  F Edot     W
----- 4 -----
IMPEDANCE pure resistance
1.0000E+05 a Re(Zs) Pa-s/m^3      2000.0  A |p|      Pa
0.0000  b Im(Zs) Pa-s/m^3      -3.4849E-13 B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           -54.060  D Ph(U)    deg
                                           2.5508  E Hdot     W

```

```

----- 5 -----
!----- 5 -----
SOFTEEND      this will reconnect at the UNION
0.0000      a Re(z)      (t)      2000.0      A |p|      Pa
0.0000      b Im(z)      (t)      -3.4849E-13 B Ph(p)    deg
                                           4.3459E-03 C |U|      m^3/s
                                           -54.060    D Ph(U)    deg
                                           2.5508    E Hdot    W
                                           2.5508    F Edot    W
                                           32.398    G Re(z)
!----- 6 -----
UNION         close the loop here
5.0000      a SegNum      2000.0      A |p|      Pa
2000.0      b |p|End      Pa = 6A?    0.0000    B Ph(p)    deg
-3.4849E-13 c Ph(p)E      deg = 6B?    1.3999E-16 C |U|      m^3/s
                                           -73.811   D Ph(U)    deg
                                           3.9031E-14 E Hdot    W
!----- 7 -----
HARDEND      seal the final end of the resonator
0.0000      a R(1/z)      = 7G?      2000.0      A |p|      Pa
0.0000      b I(1/z)      = 7H?      0.0000    B Ph(p)    deg
                                           1.3999E-16 C |U|      m^3/s
                                           -73.811   D Ph(U)    deg
                                           3.9031E-14 E Hdot    W
                                           3.9031E-14 F Edot    W

```

A topologically similar model of a complicated thermoacoustic system using TBRANCH and UNION is given in the traveling-wave-engine example in Section 6.3. Some more complicated examples of heirarchically nested structured branches and loops are shown in Figs. 6.9 and 11.1.

### 2.7.6. Manipulating entire segments

An entire segment, or a block of segments, can be copied, cut, pasted, and deleted, much like copying and pasting an entire paragraph or a contiguous block of paragraphs in a text editor. Dragging the cursor over the segment or segments indicates their selection via a background color change, and right clicking brings up the cut/copy/paste menu. (For flipping a block of segments top-to-bottom, see Section 5.7.) DELTAEC retains `sameas` addresses, RPN addresses, and master–slave links inside the copied block, while replacing any such addresses that point outside the block with appropriate numerical values. Segments and blocks of segments can be moved from one model to another using this feature, if both models are open in the same DELTAEC window.

### 2.7.7. Thermophysical properties

The Tools / `thermophysical properties` pulldown-menu selection gives the user keyboard access to DELTAEC’s library of gas and solid properties for a given mean pressure and temperature. This feature is so convenient that we often start DELTAEC simply to look up the transport properties of air, helium, and other common gases.

RPN segments, introduced in Chapter 4 and described fully in Section 11.4.1, can also be used to access thermophysical properties. These segments can be inserted anywhere in a model where the user wants to know the gas or solid properties at the local temperature and pressure. By using the incremental plot feature described in Section 3.2, tables and graphs of properties can be generated over ranges of temperature and pressure, with  $T_m$  and  $p_m$  as independent plot variables in a BEGIN segment upstream of such RPN segments.

## 3. MORE ABOUT PLOTTING

DELTAEC's built-in plotter, which is accessible in DELTAEC's Display pulldown menu, provides convenient graphical displays of DELTAEC results.

In *state plots*, the state of the wave and other thermoacoustic variables such as temperature and energy flux are displayed vs location  $x$  in the model whenever a run has converged properly. Data are tabulated in a file with the extension `<.sp>`.

In *incremental plots*, the relationships among a number of DELTAEC variables are displayed for a set of many converged runs, as one or two independent variables are automatically incremented between runs. Data are tabulated in a file with the extension `<.ip>`. Incremental plotting is also used to let DELTAEC automatically accomplish a large change in some variable via a series of many smaller changes, to avoid target-convergence problems, as described in Section 5.4.3.

DELTAEC's plotter has many built-in, rudimentary formatting features. Curves can be identified by color, line width, or line type. Default axis labels can be overwritten, and the units on axes can be changed by factors of 10. The view can be changed by zooming, and logarithmic scales can be used. Curves can be examined closely by moving the mouse cursor onto them and reading the horizontal and vertical coordinates. Segment numbers can be displayed on graphs, showing where each segment ends. The graphical results can be exported as `<.jpg>`, `<.bmp>`, or other graphics files.

DELTAEC saves the user's most recent plotting and formatting preferences for each DELTAEC output file in auxiliary files with the extensions `<.ssv>` and `<.isv>`, for the `<.sp>` and `<.ip>` files, respectively. (If the plotter is not working, close DELTAEC, delete the `<.ssv>` and `<.isv>` files associated with your model, and try again. This is also recommended when updating from one version of DELTAEC to another.)

### 3.1. State plots

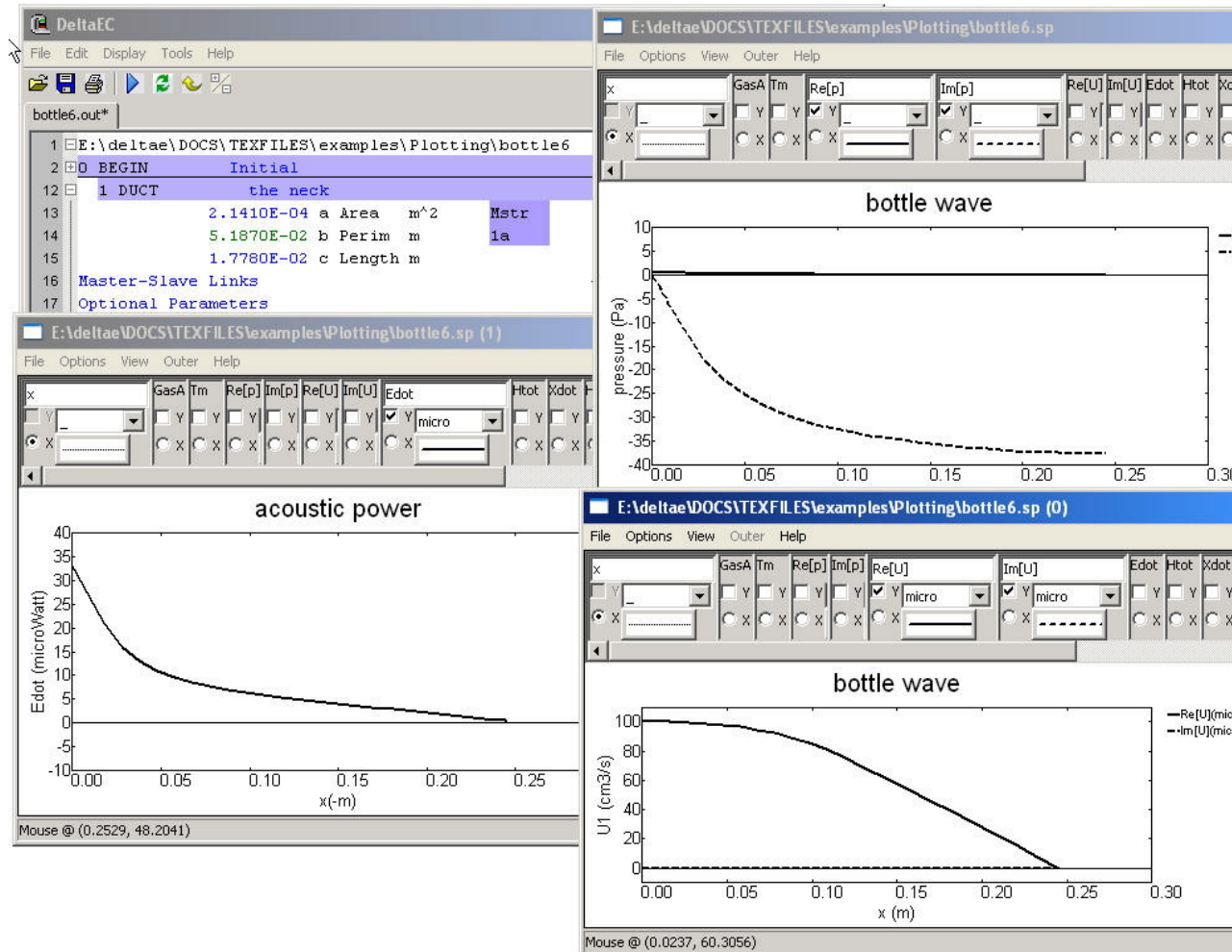
State plots show the current state of the DELTAEC model. DELTAEC's plotter makes state plots from the `<.sp>` file that is created by DELTAEC after every successful run. If the user modifies a parameter in the model, and re-runs it, DELTAEC quickly updates its plot displays to reflect the new reality, just like it updates its numerical displays. This feature was introduced in Section 2.4, where the `<bottle>` model awoke with a state plot showing  $p_1$  and  $U_1$  vs  $x$ , and we changed it to display  $\dot{E}$  vs  $x$ .

DELTAEC subdivides integrable segments like DUCTs and STK\*\*s into `Nint` pieces, writing `Nint + 1` lines of numbers to the `<.sp>` file for each segment. On the `<.sp>` plot display, each segment is then represented by `Nint` short, straight lines connecting those `Nint + 1` points. The accuracy of the underlying calculation is generally better than what is indicated

by the kinkiness of the plot, because the fourth-order Runge-Kutta integration technique [16] essentially simulates the physics on a finer length scale than the `<.sp>` file actually tabulates.

### 3.1.1. Cloning

The experienced user will soon want to display more than one plot simultaneously. “Cloning” DELTAEC plots lets this occur. Return to `<bottle6>` of Chapter 2, which awakens displaying  $p_1$  vs  $x$ . Format this curve for visual appeal, and then “clone” it twice from the plotter’s File pulldown menu, editing each clone to arrive at something that looks like this layout:



All five curves displayed here (on three graphs) are updated every time the model runs successfully, and all our effort of formatting and cloning is saved in the `<.ssv>` file if we save when closing the model, so these formatted graphs will reappear if we return to the `<bottle6.out>` model weeks later.

Sometimes it is useful to include  $A_{\text{gas}}$  vs  $x$  in one or more of the state plots, to have a sort of “drawing” of the apparatus lined up with the data graphs. The SegEndMarks option in the plotter’s Options menu puts segment numbers on the graphs, at the end of each segment, as another indication of apparatus locations on the data graphs. The schematic

view, available on DELTAEC’s Display pulldown menu, also gives a picture of the geometry (see Fig. 2.11 and the Users Guide’s title page for examples).

### 3.1.2. Legs

Models with one or more TBRANCHes and/or two or more BEGINs are topologically complicated, so DELTAEC’s `<.sp>` file breaks them down into “legs,” with each leg starting over at  $x = 0$  for purposes of displaying graphs of data. For example, Fig. 3.2 shows a branched resonator, which we modeled with `<legs.out>` (available in the `<Examples\Plotting>` folder). Figure 3.3 shows the standing-wave part of the pressure in that resonator. The side-branch parts of the data form one curve, and the trunk parts of the data form another curve.

Using clone plots and the Legs pulldown menu, the user can show one variable per graph, with all legs plotted in each graph; or can show one leg per graph with all variables of interest plotted in each graph.

### 3.1.3. Phasor plotting options

The complex variables  $p_1$  and  $U_1$  are usually of the greatest interest in thermoacoustic systems, so DELTAEC offers several ways of displaying them graphically.

State plots awaken by default with  $\text{Re}[p_1]$  and  $\text{Im}[p_1]$  displayed, and  $\text{Re}[U_1]$  and  $\text{Im}[U_1]$  easily accessible by checking the appropriate boxes at the top of the state-plot display. The real and imaginary parts of the ratio  $Z = p_1/U_1$  are also easily accessible. All three of these complex functions can also be displayed in terms of magnitude and phase, through the small pulldown menus associated with each real or imaginary variable, as illustrated in Fig. 3.4.

In standing-wave systems, the state plots that display variables such as  $p_1$  and  $U_1$  as functions of  $x$  are usually the most useful. However, for traveling-wave systems, phasor diagrams often offer vital insight, so they are included among DELTAEC’s standard state-plot options. To enable a phasor plot, check  $\text{Re}[p_1]$  or  $\text{Re}[U_1]$  instead of  $x$  as the “x axis” of the plot, and then use  $\text{Im}[p_1]$  or  $\text{Im}[U_1]$  as the “y axis.” Alternatively, check “Phasor Mode” in the plotter’s Options pulldown menu, to ensure equal scaling on the real and imaginary axes and to allow both  $p_1$  and  $U_1$  phasors on the same graph.

Phasor diagrams are only used for complex variables, here  $p_1$  and  $U_1$ . (Other examples include transducer current and voltage, but these are not available in DELTAEC’s `<.sp>` files.) In a phasor diagram, these complex variables are plotted on the complex plane, with

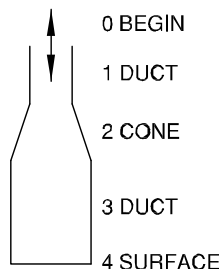


Figure 3.1: The bottle, used as an example in both Chapter 2 and the present chapter.



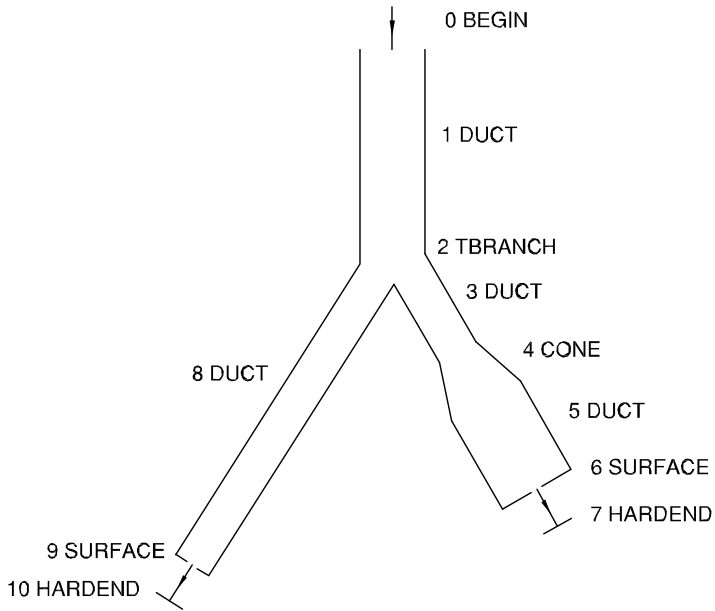


Figure 3.2: A branched resonator. In the model of this resonator,  $p_1$  and  $U_1$  have the same phase at BEGIN, and the frequency is a guess. The magnitude of  $p_1$  in BEGIN is fixed and the magnitude of  $U_1$  in BEGIN is a guess. Both TBRANCH impedances are guesses, and the four targets in the two HARDENDs are used.

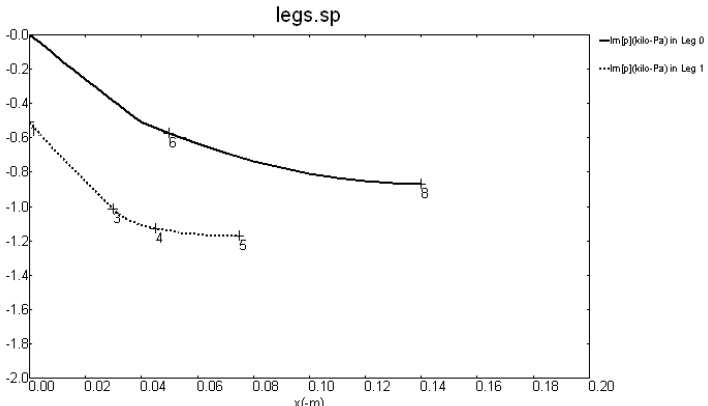


Figure 3.3: The standing-wave part of the pressure oscillation in the branched resonator of Fig. 3.2. There are two curves because the resonator has two legs and both legs have been selected in the Legs pulldown menu. The kink in the upper curve occurs at the same pressure as the start of the lower curve, because these are the same location: the TBRANCH.

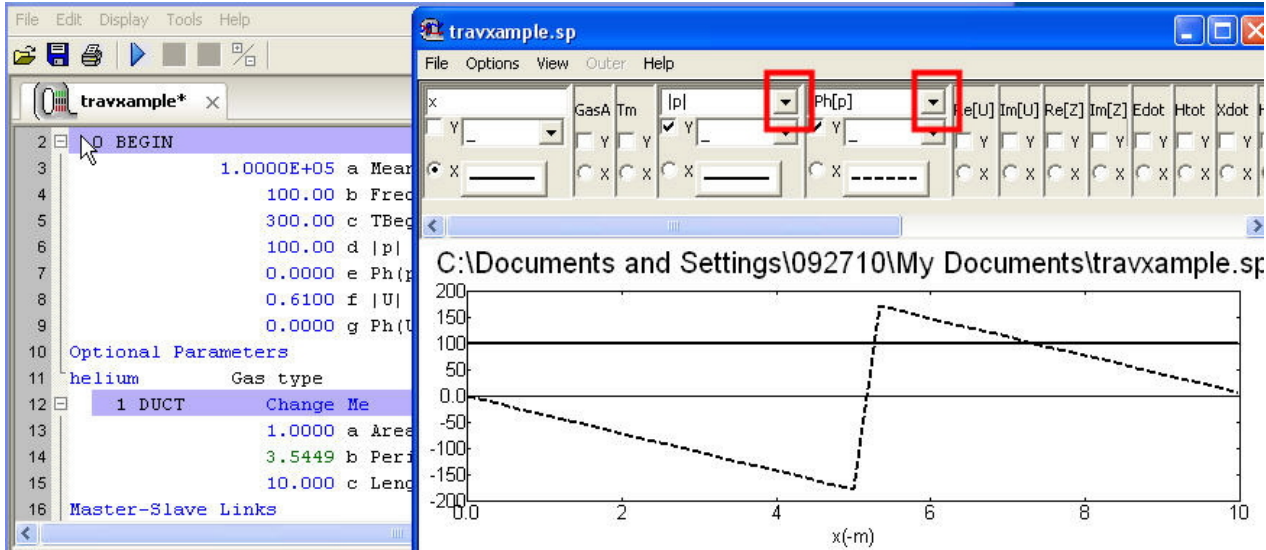


Figure 3.4: A traveling wave with  $|p_1|$  nearly constant at 100 Pa and the phase of  $p_1$  starting at zero and going down by  $360^\circ$ , resetting from  $-180^\circ$  to  $+180^\circ$  near  $x = 5$  m. The little menus to convert the real and imaginary parts of  $p_1$  to the magnitude and phase are indicated here by the red boxes. Similar features are available for  $U_1$  and  $Z$ .

the real part plotted along the horizontal axis and the imaginary part plotted along the vertical axis. Conventionally, arrows from the origin point to each such complex value. One can imagine all of the phasors on a phasor diagram rotating counterclockwise around the origin at radian frequency  $\omega$ . Then the actual variables represented by the phasors are the time-dependent projections of the phasors on the horizontal axis. Thus a phasor diagram shows the relative magnitudes and time-phase relationships of a number of phasors. The difference between phasors at adjacent places in an apparatus also offers insight into the impedance of what lies between. The difference between pressure phasors is related to the volume flow rate phasor via the momentum equation, which can include resistance and inertance; the difference between volume-flow-rate phasors is related to the pressure phasor via the continuity equation and the compressibility of the intervening gas [13].

Figure 3.5 shows the  $U_1$  phasors for the bottle.

Another way to look at phasors, based on mouse cursor position in the schematic view, is described in Section 2.7.3.

### 3.2. Incremental plotting

DELTAEC allows for incremental “plotting” by automatically incrementing (or decrementing) an independent variable (or two independent variables) and running repeatedly. This can be used either to keep DELTAEC’s shooting method happily well-targeted while changing a single independent variable by many small, “safe” steps when a single leap to a desired new value would fail to meet targets (see Section 5.4.3), or to tabulate the changing independent variable(s) together with one or more changing output variables in a file named

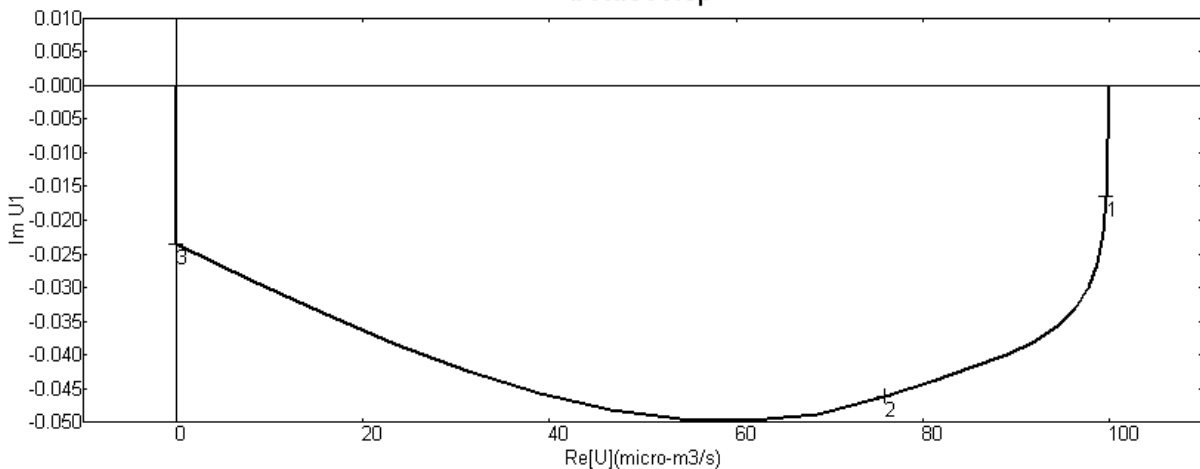
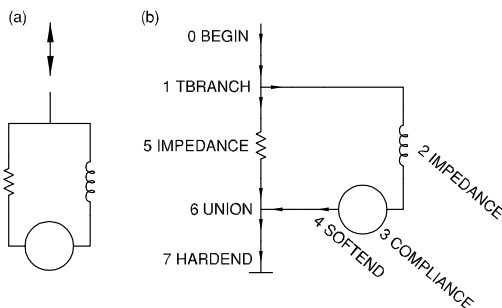


Figure 3.5: The phasor plot displays complex  $p_1$  and/or  $U_1$  on the real–imaginary plane. With SegEndMarks selected in the Options pulldown menu, segment numbers are displayed at the end of each segment.

`<.ip>`. Users can then manipulate and/or plot the `<.ip>` file with DELTAEC’s built-in plotter (or with their favorite graphics or spreadsheet software). We illustrate these features by continuing two previous examples.

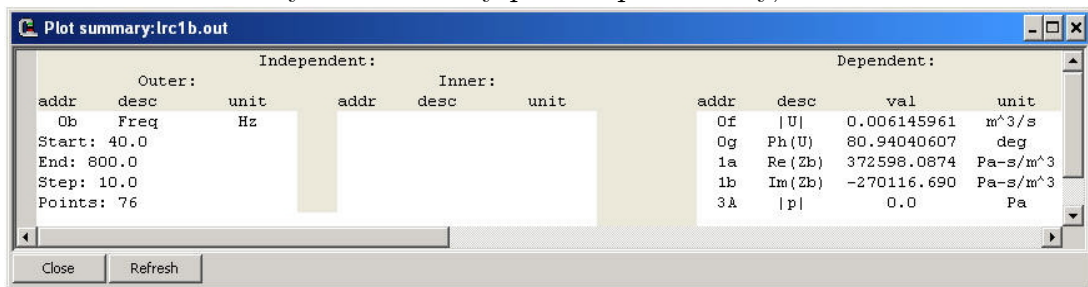
### 3.2.1. One-dimensional incremental plots

A driven, parallel LRC network was used as an example of structured branches and unions in Section 2.7.5. Here, we use the same example to illustrate a simple incremental plot in DELTAEC. The system is illustrated in the figure below, and the file we start with is `<lrc1.out>` from the `<Examples\Acoustics>` folder. Open that file, and save it as `<lrc1b.out>` so subsequent changes don’t affect the original. Review the model, to understand that this LRC network is driven at 60 Hz and 2 kPa. Under these circumstances, the pressure amplitude in the COMPLIANCE is 2.28 kPa—higher than at the driver.



Let’s see how the pressure amplitude in the COMPLIANCE depends on frequency, using an incremental plot. Unroll the BEGIN segment, double click on the frequency, and check the Plot box. Let the frequency incrementing start from 40 Hz, end at 800 Hz, and do so in 77 steps so the increments are 10 Hz. Now unroll the COMPLIANCE segment, double click on

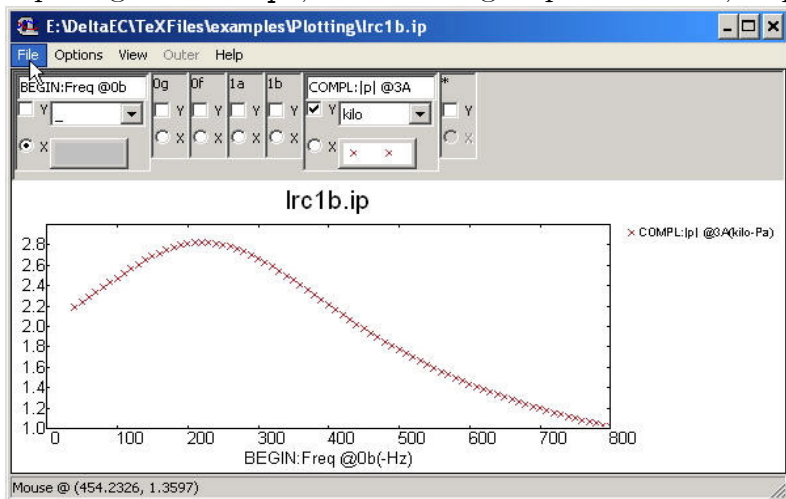
output **A**—the pressure amplitude—and select it as a dependent plot variable. Note that **0b** and **3A** are flagged in the main display as plot variables now. Open the Incremental Plot Summary from the Tools pulldown menu, and see what information is presented. (Note that, in addition to our selection of **3A**, all four guesses are listed as dependent plot variables—to let the user return to any incrementally plotted point easily, as described in Section 3.2.3.)



Now start the repetitive `run`. DELTAEC begins at the initial value of the frequency, 40 Hz, and performs an ordinary run, adjusting the guesses until the targets are met. It writes a line to the incremental-plot file `<lrc1b.ip>`, recording on one line the independent variable ( $f$ ), the guesses, and the chosen dependent variables ( $|p_1|$  in the COMPLIANCE in the present case). Then DELTAEC increments the independent variable by its step (10 Hz in this case), performs another ordinary run, and records the results on the next line in the `<.ip>` file...repeating until the final value of the independent variable is reached. In a text editor, the `<.ip>` file shows this:

```
BEGIN:Freq BEGIN:|U| BEGIN:Ph(U) TBRAN:Re(Zb) TBRAN:Im(Zb) COMPL:|p|
 Hz m^3/s deg Pa-s/m^3 Pa-s/m^3 Pa
 0b 0A 0B 0C 0D 3A
 40.00 3.9237E-03 84.11 5.5931E+05 -4.5475E+05 2186.
 50.00 5.0127E-03 82.56 4.4726E+05 -3.4400E+05 2234.
 60.00 6.1460E-03 80.94 3.7260E+05 -2.7012E+05 2282.
 ...
```

and opening `<lrc1b.ip>`, and choosing to plot **3A** vs **0b**, displays this:



The pressure amplitude starts at 2 kPa at low frequency, and rises with frequency at first, peaking near 200 Hz. Could it be a resonance? The user could now change a parameter such as the resistance in segment 5, repeat the repetitive `run`, and examine the effects of such a change on the resonance frequency and quality factor. (Don't forget that a frequency-independent  $\text{Im}[Z_s]$  in segment 2 is not the same as a frequency-independent inertance!)

### 3.2.2. Two-dimensional incremental plots

We use the bottle example from Chapter 2 again, starting from <bottle6.out> in the <Examples\Acoustics> folder. Make the guess–target summary look like this:

```
Guess-Target Summary:
GUESS      Of          Og
name BEGIN:|U|  BEGIN:Ph(U)
value      1.00E+03      90.
units      m^3/s          deg
TARGET     5a          5b
name HARDEND  :R(1/z)  HARDEND  :I(1/z)
value      .00          .00
```

Now display the Incremental Plot Summary:

```
Dependent Variables (outputs):
PLOTS      Of          Og
name BEGIN:|U|  BEGIN:Ph(U)
units      m^3/s          deg
```

Keep these parameters as two of the dependent variables to be plotted (they are copies of the guesses).

We will watch these variables as functions of two independent variables: frequency  $f$  and mean temperature  $T_m$ . To set up these two independent variables to be incremented, modify each of them, checking the Plot box in the Parameter Edit dialog and then entering initial and final values, and the number of steps. Use only three values of mean temperature in the “outer” loop—300, 400, and 500 K—by letting  $T_m$  go from 300 to 500 K in three steps of 100 K, and let  $f$  go from 100 Hz to 1200 Hz in 551 steps to get exactly 2-Hz increments in the “inner” loop. Also add the pressure in the bottom of the bottle, 5A, as another dependent plot variable, by double clicking on it and selecting it for plotting. After these actions, check the Incremental Plot Summary again:

Outer:			Inner:			Dependent:		
addr	desc	unit	addr	desc	unit	addr	desc	unit
0c	TBeg	K	0b	Freq	Hz	0f	U	m^3/s
Start:	300.0		Start:	100.0		0g	Ph(U)	deg
End:	500.0		End:	1200.0		5A	p	Pa
Step:	100.0		Step:	2.0				
Points:	3		Points:	551				

Now start a repetitive run. DELTAEC steps through the frequency and mean temperature, taking several seconds. For each value of frequency and mean temperature, it adjusts guesses until targets are met, and then writes one line to the <.ip> file, recording the values of the independent and dependent plot variables, including all guesses. When it has finished, use the pc’s operating system to find the new file, <bottle6.ip>:

```
BEGIN:TBeg  BEGIN:Freq  BEGIN:|U|  BEGIN:Ph(U)  HARDEN:|p|
      K          Hz          m^3/s          deg          Pa
      0c          0b          0f          Og          5A
300.000    100.000    2.0250E-06  -8.8471E+04    1.38690
300.000    102.000    2.0953E-06  -8.8471E+04    1.40830
300.000    104.000    2.1682E-06  -8.8471E+04    1.43090
.
.
500.000    1198.00    1.6259E-07  -8.8481E+04    0.61493
500.000    1200.00    1.7641E-07  -8.8480E+04    0.62388
```

The results are displayed by DELTAEC’s plotter as shown in Fig. 3.6. The plotter’s pulldown menu must be used to get all three curves to show on the plot.

### bottle at three temperatures

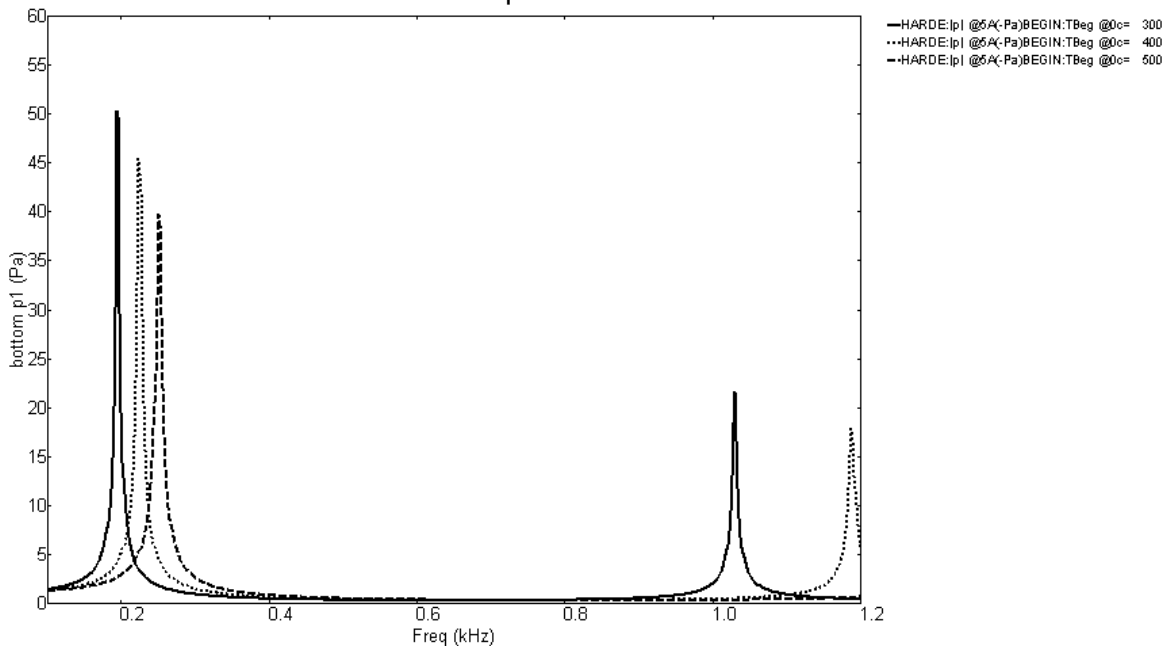


Figure 3.6: Pressure amplitude at the bottom of the bottle as a function of drive frequency, with the pressure amplitude at the mouth held constant at 1 Pa, for three different temperatures.

### 3.2.3. Rewinding incremental plots

The rewind feature takes DELTAEC quickly back to a user-selected, previously calculated state that has been saved in the `<.ip>` file. This is useful for:

- Selecting a new operating point after using incremental plotting to explore a broad range of behaviors. For example, the user could select a point at the top of one of the peaks in Fig. 3.6.

- Selecting a recent well-converged operating point from an incremental plot that contains both converged and unconverged operating points and/or from a long incremental plot that seemed to go awry and was canceled by the user.

The rewind feature is available whenever an `<.ip>` file exists. It is accessible in the Tools pulldown menu or through a button on the toolbar near the top of the main display. When a point is selected, DELTAEC restores the independent plot variable(s) and the values of guesses at that point. It also restores any output parameters that it has in the `<.ip>` file, which it might have selected in addition to the users choices because it caught the user inadvertently or boldly using a `sameas` or `RPN` pointer to an output parameter in a segment downstream of that `sameas` or `RPN`, despite DELTAEC’s “Forward Reference Condition” warning about the unstable shooting-method behavior that can result from using such downstream results. (See Section 5.4 for an example.)

### 3.2.4. Incremental plots when more than one model is run

Two or more models can be open in DELTAEC at the same time. If these models are run separately, any incremental plots in any of those models will run separately, just as if they were open in separate DELTAEC windows.

Linking two or more models together in a single DELTAEC window and running them together with Run All is described in Section 4.9. The interaction between incremental plots and multi-model running is described near the end of that section.

Rewinding from an incremental plot of inter-linked models behaves similarly to a single-model plot rewinding as described above in Section 3.2.3.

## 4. RPN MATH

### 4.1. Summary

The RPN segment lets the user create nonstandard guesses, targets, and simple algebraic calculations anywhere in a DELTAEC model. The format of the segment is shown in this example:

```
!----- 10 -----  
! The result(s) of the second line's calculations show up as output(s) in the right column.  
RPN      Dimensionless magnitude of specific acoustic impedance  
0.000    a                                7.8540E-05 A Znorm  
p1 U1 / mag 9a * rho / a /
```

Input “a” is a real number, which can be a guess, a target, and/or a number used subsequently in calculations via `sameas` links that point to it. The second line (line “b”) is a formula expressing a user-defined algebraic procedure in Reverse Polish Notation (RPN), a parenthesis-free algebra encoding technique. Variables such as density `rho` are re-evaluated at this RPN segment’s location in the model (using, for example,  $T_m$  at this location) during each numerical integration through the model.

A complete lesson in Reverse Polish Notation (also called postfix) can be found in instruction manuals for most HP (Hewlett-Packard) pocket calculators and on many websites, including Refs. [17] and [18]. DELTAEC users who are unfamiliar with RPN notation are encouraged to consult such a resource. Briefly, as numbers (including pointers to numbers elsewhere in the model, and including variables introduced below) are encountered in the instruction line, these numbers are pushed down onto a stack. Any numbers already present on the stack are pushed deeper into the stack. When a unary operator such as `cos`, `log`, or `sqrt` is encountered in the formula, it takes a single number off the stack (“pops” it), acts on it, and pushes the result back onto the stack. When a binary operator such as `+` or `*` is encountered, it pops two numbers off the stack, combines them appropriately, and pushes the result back onto the stack. Other numbers that may be on the stack but are not involved in the current operation simply move up and down during the operation so no gaps appear in the stack and no uninvolved numbers get lost. (Some of us find it helpful to visualize a spring-loaded stack of dishes at a cafeteria, which adjusts itself so the top dish is always at the most convenient height no matter how many dishes are hidden underneath it.)

After a simple formula has been processed, there may be only one number remaining on the stack; if so, this number is the “result” of the RPN segment and is displayed as a single output, “A.” However, it is often desirable to leave additional results on the stack to be accessed elsewhere or viewed by the user. Whatever is left on the stack after the instruction line has been processed is displayed as outputs A, B, C,... For example, if the instruction line is “1 2 3 4 + 45 cos”, the segment’s output values will be A = 0.7071, B = 7, C = 2, D = 1. To assist in human readability, DELTAEC recognizes the semicolon as a punctuation mark



that the user can insert anywhere without affecting the calculation. Thus, the instruction line given above could also be written “1 ; 2 ; 3 4 + ; 45 cos”. (Note that blanks must both precede and follow the semicolon.) Three shades of colored highlighting in the main display of RPN segments also help human readability.

As a further aid to human readability, clicking on an output parameter in an RPN segment and then clicking on “List Linkages” brings up a display showing how that parameter is calculated, among other things. There, an RPN calculation can be displayed with parentheses if desired. For example, outputs A and B in the last paragraph can be displayed as cos(45) and 3+4, respectively.

DELTAEC’s RPN segment recognizes the common trigonometric, logarithmic, exponential, and algebraic functions present on most pocket calculators, as well as hyperbolic and Bessel functions that are not available on most calculators. Most of these functions accept complex arguments. Other functions convert between complex numbers and real numbers in the usual ways (e.g., real, imaginary, magnitude, phase). For complete lists and short descriptions of DELTAEC’s RPN operations, see the tables in Section 11.4.1.

Whenever the command string of an RPN is being modified, clicking the ShowOps button displays a reminder of available operations and variables similar to the following:

```
Valid RPN commands listed below.
(It is OK to type lower case, except Bessel functions.)

Basic: +      -      *      /      ^      ~      inv
        log     log10  exp     tenx   sqrt   sqrd   abs
Trig:  cos     acos   sin     asin   tan     atan  atanr  pi
        cosr   acosr  sinr    asinr  tanr   atan2  atan2r
        sinh   cosh   tanh
Bessel: J0     J1     Y0     Y1
RPN stack: #   lstx  a<b   sto   rcl   min   max   avg   inp
Complex: mag   real  imag  arg   argr  conj  i     cmplx cmplxr

Gas:  gamma  a      rho   cp   k     mu   beta  enth  dk   dn   m
Mixture: nL   kT     D12   mL   mH   NLdot nLzro
Solid: rhos  cs     ks     ds

State: Tm     w      f      pm   p1   U1
Thermodyn: Edot Htot  H2k   Xdot Tzero
Mean flow: Ndot p20HL p20tot
Cumulative: F1   vol   x
Change:  =Tm   =p1   =U1   =H2k =Ndot =NLdot
         =pm   =f     =nL   =F1  =vol  =p20HL
         =Tzero =nLzro
```

Clicking on one of these reminders will save the user a little typing.

Where capital letters appear in the preferred spelling of an RPN operation, DELTAEC will usually accept lower-case typing.

The rest of this chapter introduces many ways that RPN segments can be used.

## 4.2. Non-standard results

The RPN segment is often used to display interesting results for the user.

For example, to calculate and display the gas displacement amplitude

$$|\xi_1| = \frac{|U_1|}{\omega A} \tag{4.1}$$

at the end of, for example, a DUCT at segment 5, an RPN after segment 5 could be written

```
!----- 6 -----
RPN magU1 over omega A (units: meters)
0.00
5C 2 / pi / 0b / 5a /
```

The “List Linkages, Show with parentheses” display shows this as  $\{[(5C/2)/\text{pi}]/0b\}/5a/$ . The “0.00” in line **a** is not used in this example. **5C** points to  $|U_1|$  in segment 5, **0b** points to the frequency in segment 0, and **5a** points to the duct area in segment 5. Alternative ways to arrive at the same result in DELTAEC RPN formulas include

```
5C 2 pi * 0b * 5a * /
```

OR

```
5C 2 pi 0b 5a * * * /
```

OR

```
U1 mag w / 5a /
```

These examples show how DELTAEC recognizes numbers (e.g., “2”), pointers to input and output numbers (e.g., “0b” and “5C”), special constants (here “pi”), and local complex and real variables (here  $U_1$  and  $\omega$ ).

As a second example, the user could display the local values of several thermophysical properties of the gas, namely  $\rho_m$ ,  $c_p$ ,  $a$ , and  $\mu$ , by inserting this RPN segment into a model:

```
RPN  A=rho, B=cp, C=sound speed, D=viscosity (MKS units)
0.00
mu a cp rho
```

DELTAEC can perform complex arithmetic in RPNs. For example, it can calculate

$$f_k = \frac{\tanh [(1+i) y_0/\delta_\kappa]}{(1+i) y_0/\delta_\kappa} \quad (4.2)$$

for a parallel-plate geometry [e.g., see the first of Eqs. (10.106)]. We let the input parameter, line “a”, be equal to the plate half-spacing  $y_0$  so we can change it easily later:

```
RPN  Thermal function for parallel plate geom. "a"=y0 (meters)
2.0000E-04 a (t) ( 0.9244 , -0.2392 ) A
(1,1) inp * dk / tanh lstx /
```

Here (1,1) illustrates the entry of a complex constant,  $(1+i)$ . This complex constant could just as well have been made by entering  $1 \text{ i} + .$  The input variable of the segment, line **a**, is referenced via **inp** in the RPN command string, and the thermal penetration depth  $\delta_\kappa$  is accessed via **dk**. The “List Linkages, Show with parentheses” display shows this result as  $\tanh([(1,1)*2a]/dk)/\{[(1,1)*2a]/dk\}$ .

For complicated algebra that won’t fit in one RPN, the user can cascade two or more RPNs, each relying on the output(s) of the previous one(s).

It can be dangerous to link the master of a master–slave set to the result of an RPN segment through a **sameas** link. Such an arrangement works fine until the model is saved to the disk and later re-opened by DELTAEC. While opening such a model, DELTAEC calculates the constant associated with the master–slave link during file parsing, before the model is run and thus before the numerical value of the master is calculated with Fortran double-precision accuracy. Thus, the constant might be set inaccurately. If the  $\langle .\text{out} \rangle$  file has been modified in a text editor, values of the *output* parameters read from the disk, which are usually irrelevant to subsequent model behavior, can affect such master–slave linkages severely.

### 4.3. List Linkages

When cascaded RPNs also involve parameters that are slaved to, or `sameas`, other parameters, and input parameters in other, non-RPN segments are `sameas` to the RPN results, the dependencies can become very complicated. The List Linkages display, available in the modify-parameter dialog for both input and output variables (with RPN parameter `b` being the only exception—click on the associated output variable(s) instead) can be very useful for avoiding mistakes, diagnosing infinite loops of pointers, diagnosing Forward Reference Conditions, etc. At Los Alamos, we often find ourselves with complicated situations, illustrated with this example for the Linkages List from variable `8A`, segment 8 being an RPN:

```

8A k(m-1) is an RPN result, calculated by 8b, via
   the formula (sqrt[1+[(gamma-1)*7B]}/(1-7A)]*w/a,
   which needs 7B and 7A.
7A fnu is an RPN result, calculated by 7b, via
   the formula ({[(1-i)*8a]*dn}/2)/7a,
   which needs 8a and 7a.
7B fkappa is an RPN result, calculated by 7b, via
   the formula ({[(1-i)*8a]*dk}/2)/7a,
   which needs 8a and 7a.
8a G_or_T does not need other parameters.
7a G_or_T does not need other parameters.

9A needs 8A via RPN calc.
9B needs 8A via RPN calc.
9C needs 8A via RPN calc.
10a needs 9A via sameas.
10b needs 9B via sameas.

```

If you were planning to delete this segment, the final lines in the List Linkages display would inform you to look first at segments 9 and 10 for unintended consequences, and it would suggest that you could investigate whether any other segments need segment 7, which might then become another candidate for deletion.

### 4.4. Non-standard inputs

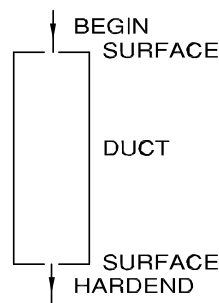
Line `a` of an RPN segment can be used as an input variable that is unavailable in any other segment. This is convenient for a variable that the user wants to manipulate in non-MKS units, a variable that is frequently changed by the user, or a variable that is changed by `DELTAEC` as a guess or an independent incremental-plot variable.

For example, suppose that a user is concerned with the volume of a cylindrical resonator, and wants to find out what shape of cylinder requires the lowest drive power for a given pressure amplitude. The user also prefers to work with the resonator's diameter instead of its area.

```

TITLE      Example 1: Plane-wave resonator
!Created@13:21:32 8-Jul-06 with DeltaE Vers. 5.5b6 for the IBM/PC-Compatible
----- 0 -----
BEGIN      Initialize things
1.0000E+05 a Mean P      Pa          7903.3      A Freq    G( 0b)    P
7903.3     b Freq        Hz          G          3.9873E-05 B |U|    G( 0f)    P
300.00     c TBeg       K
1000.0     d |p|        Pa
0.0000     e Ph(p)     deg
3.9873E-05 f |U|      m^3/s      G
0.0000     g Ph(U)     deg
helium     Gas type
----- 1 -----
RPN      a=diam (cm), A=cross-sectional area (square meters)
1.0000   a G or T      (t)          7.8540E-05 A Area

```



```

3.1372E-02 B Perim
inp 100 / sto pi * rcl sqrd pi * 4 /
!----- 2 -----
RPN      a=volume (cubic cm).  A=length (meters)
10.000   a G or T              (t)          0.1273  A length
inp 1E6 / 1A /
!----- 3 -----
SURFACE  top end cap of resonator (imagine it's a moving driver)
sameas  1A a Area              m^2          1000.0   A |p|      Pa
                                                0.0000   B Ph(p)    deg
                                                3.9203E-05 C |U|    m^3/s
                                                0.0000   D Ph(U)    deg
                                                1.9602E-02 E Hdot   W
ideal    Solid type              1.9602E-02 F Edot   W
!----- 4 -----
DUCT     the most important piece of this resonator
sameas  1A a Area              m^2          996.65   A |p|      Pa
sameas  1B b Perim            m           5.8368E-04 B Ph(p)    deg
sameas  2A c Length           m           6.6737E-07 C |U|    m^3/s
                                                5.8368E-04 D Ph(U)    deg
ideal    Solid type              3.3257E-04 E Hdot   W
                                                3.3257E-04 F Edot   W
!----- 5 -----
SURFACE  End of the pipe
sameas  1A a Area              m^2          996.65   A |p|      Pa
                                                5.8368E-04 B Ph(p)    deg
                                                2.2261E-19 C |U|    m^3/s
                                                -104.32   D Ph(U)    deg
ideal    Solid type              -2.7438E-17 E Hdot   W
                                                -2.7438E-17 F Edot   W
!----- 6 -----
HARDEND  seal the final end of the resonator
0.0000   a R(1/z)              = 6G?     996.65   A |p|      Pa
0.0000   b I(1/z)              = 6H?     5.8368E-04 B Ph(p)    deg
                                                2.2261E-19 C |U|    m^3/s
                                                -104.32   D Ph(U)    deg
                                                -2.7438E-17 E Hdot   W
                                                -2.7438E-17 F Edot   W
                                                -1.1503E-16 G R(1/z)
                                                -4.5062E-16 H I(1/z)

```

With this file, the user can modify the diameter (in cm) and the volume (in cm<sup>3</sup>) conveniently in 1a and 2a, and could use those as independent plot variables or as guesses. RPN instruction 1b calculates the area (in m<sup>2</sup>) and puts it in output 1A for use in segments 3, 4, and 5. RPN instruction 2b calculates the length from the given diameter and volume, putting it in output 2A for use in segment 4.

Note that segments 3, 4, and 5 show how **sameas** can point to DELTAEC outputs (capital letters) as well as inputs (small letters).

## 4.5. Non-standard targets

DELTAEC has a few specially assigned input parameters that hold target values in segment types that have outputs very commonly used in targets. Some of these were introduced in Chapter 2: the complex normalized impedances  $z_n$  in **HARDEND** and **SOFTEND**, and the complex pressure  $p_1$  in **UNION**. Other routine, built-in targets are the solid temperatures in heat exchangers, which will be introduced in Chapter 5. DELTAEC knows which output variables in these segments should be associated with their “default” target values when the shooting-method algorithm is using them.

The experienced user will soon hunger for more possibilities. An application may call for specifying piston stroke, acoustic power, or pressure or velocity (magnitude or phase) somewhere other than **BEGIN**. The RPN segment is used to establish such variables as targets. The first of the RPN outputs, “A,” is recognized by DELTAEC as paired with the RPN input “a” to comprise a potential target/result.

For example, the stroke of a linear motor is often limited by the flexure characteristics of its suspension. Here is an example using the RPN segment to find out how much current must be delivered to a motor to keep the motor's stroke targeted at 2 mm while driving a resonator. The usual HARDEND targets are also used, with complex  $U_1$  in BEGIN being the corresponding guesses. The RPN segment's input, 2a, which is set by the user at 2.000, is the user's desired target, and the RPN formula yields the motor stroke, which is written in 2A. DELTAEC has adjusted guess 1h, the motor current, until the motor stroke met its target.

```
TITLE      Example 1: Plane-wave resonator
!Created@14:22:59 8-Jul-06 with DeltaE Vers. 5.5b6 for the IBM/PC-Compatible
!----- 0 -----
```

```
BEGIN      Initialize things
1.0000E+05 a Mean P      Pa
150.00    b Freq        Hz
300.00    c TBeg        K
0.0000    d |p|        Pa
0.0000    e Ph(p)      deg
9.4483E-03 f |U|        m^3/s    G
147.39    g Ph(U)      deg    G
helium    Gas type
```

```
!----- 1 -----
IESPEAKER a linear motor and piston
sameas 3a a Area        m^2      2443.2    A |p|        Pa
1.0000 b R            ohms     -177.67   B Ph(p)      deg
3.0000E-03 c L        H        9.4248E-03 C |U|        m^3/s
5.0000 d BLProd      T-m      147.29    D Ph(U)      deg
1.0000E-02 e M        kg       9.4267    E Hdot       W
1.0000E+04 f K        N/m      9.4267    F Edot       W
0.1000 g Rm          N-s/m    20.919    G WorkIn     W
4.7776 h |I|        A          14.031    H Volts      V
0.0000 i Ph(I)      deg        4.7776    I Amps       A
51.381    J Ph(V/I)   deg
2443.2    K |Px|      Pa
-177.67   L Ph(Px)   deg
ideal     Solid type
```

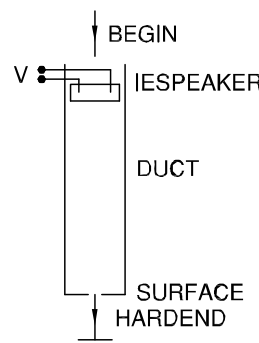
```
!----- 2 -----
RPN      peak to peak stroke of the motor (mm)
2.0000  a G or T      = 2A?      2.0000  A stroke
U1 mag 1a / w / 2 * 1000 *
```

```
!----- 3 -----
DUCT     the most important piece of this resonator
1.0000E-02 a Area      m^2      S= -2    2441.6    A |p|        Pa
0.3540 b Perim        m        Fn( 3a)  -177.56   B Ph(p)      deg
6.7000 c Length       m        2.8677E-05 C |U|        m^3/s
-177.56   D Ph(U)      deg
3.5009E-02 E Hdot      W
3.5009E-02 F Edot      W
ideal     Solid type
```

```
!----- 4 -----
SURFACE  End of the pipe
sameas 3a a Area      m^2      2441.6    A |p|        Pa
-177.56   B Ph(p)      deg
8.8927E-15 C |U|        m^3/s
-36.711   D Ph(U)      deg
-8.4187E-12 E Hdot      W
-8.4187E-12 F Edot      W
ideal     Solid type
```

```
!----- 5 -----
HARDEND  seal the final end of the resonator
0.0000 a R(1/z)      = 5G?      2441.6    A |p|        Pa
0.0000 b I(1/z)      = 5H?      -177.56   B Ph(p)      deg
8.8927E-15 C |U|        m^3/s
-36.711   D Ph(U)      deg
-8.4187E-12 E Hdot      W
-8.4187E-12 F Edot      W
-4.6190E-14 G R(1/z)
3.7606E-14 H I(1/z)
```

```
! The restart information below was generated by a previous run
! You may wish to delete this information before starting a run
! where you will (interactively) specify a different iteration
! mode. Edit this table only if you really know your model!
guessz   Of Og 1h
xprecn   -1.1737E-08 4.2373E-03 -2.9847E-05
targs    2a 5a 5b
Mstr-slave 1 3 -2
```



The target parameter of an RPN, like any other target, can be used as the independent variable in a plotting loop.

An RPN segment should usually be placed downstream of all the results and guesses that it depends on, so the RPN will have current values during each DELTAEC integration sequentially through the segments—especially if the RPN is used as a target or if its results are used elsewhere in the model.

## 4.6. Changing an integration variable

It is occasionally necessary to abruptly change the value of one or more of DELTAEC’s integration variables in the middle of its integration. The RPN segment lets the user change these in response to a simple algebraic calculation, by using an RPN operator of the form `=variable`, where `variable` is one of the integration variables. For example, one might have a resonator whose top half is wrapped in a water bath at one temperature and whose bottom half is bathed at another temperature. Splitting the resonator duct into two sequential DUCTs and inserting an RPN with an appropriate command string between the two DUCTs will change the gas temperature  $T_m$  halfway along the resonator. To change the temperature from 300 Kelvin (set in the BEGIN) to 320 Kelvin, the segment between the DUCTs could be

```
RPN
0.00
320 =Tm
```

OR

```
RPN
320.0
inp =Tm
```

To increment the temperature by 20 Kelvin, use either

```
RPN
0.00
Tm 20 + =Tm
```

OR

```
RPN
20.0
Tm inp + =Tm
```

Used in this way, RPNs can essentially act like a user-created lumped-element segment, changing the integration variables in ways not available in DELTAEC’s built-in segments. For a complicated example of a “custom segment” implemented with RPNs that change three integration variables, see “pulsed combustion” in Chapter 14.

## 4.7. Two examples

The `<Examples\RPN>` folder of DELTAEC’s installation has more-elaborate examples of using RPN segments in the ways described above.

`<LinearAlternator.out>` is a 24-segment model in which most of the segments are RPNs. The model simulates a linear alternator, including leakage through an imperfect piston seal. One RPN input specifies the leakage gap around the piston, and another RPN calculates the flow impedance of that gap, for use in a subsequent IMPEDANCE segment,

verifying that the Reynolds number is low enough that the laminar calculation of the flow impedance should be accurate. Another RPN calculates the piston stroke. A few more RPNs allow the IESPEAKER's complex current to be guessed, to target a specified electrical impedance: A 100-Ohm resistor in series with a 39- $\mu$ F capacitor, chosen to cancel the effect of the IESPEAKER's internal inductance. Final RPNs and other segments confirm that all incoming acoustic power is accounted for as the sum of electrical power delivered, electrical power dissipated in the IESPEAKER's internal resistance, mechanical power dissipated in the IESPEAKER's  $R_m$ , and thermal-hysteresis losses on all the surface areas in the model.

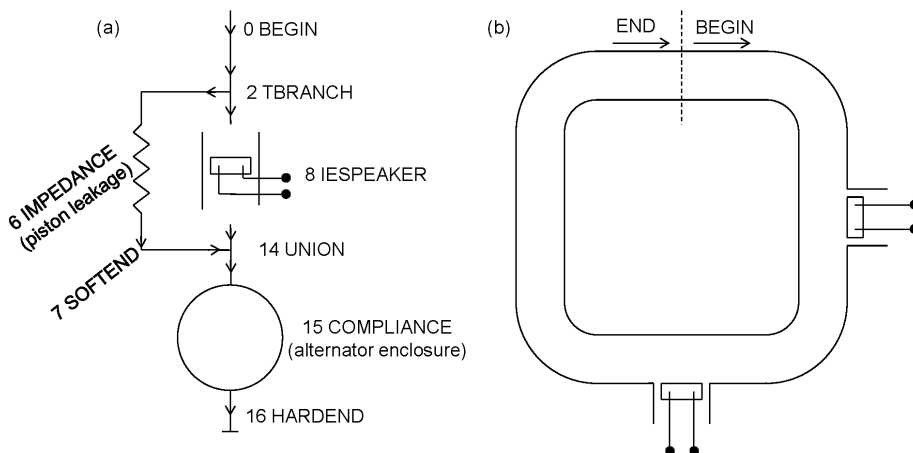


Figure 4.1: Two examples using many RPN segments. (a) A linear alternator with specified load impedance and piston leakage. (b) A loop of tubing filled with air and driven at full-wave resonance by two speakers separated by a quarter wavelength.

`<closedLoop.out>` shows a loop of tubing 2 meters long, with two side-branch drivers a half meter apart being electrically driven 90° out of phase at a frequency intended to create a full-wavelength traveling wave in the air in the loop. This example illustrates the use of four RPNs as targets, to ensure that DELTAEC “connects” the end of the model back to the beginning, by making  $p_1$  and  $U_1$  identical at the end and at the beginning.

```

TITLE      A loop of tubing, with two side-branch drivers
0  BEGIN   Start in the middle of nowhere, and integrate around a loop
1.0000E+5  a Mean P Pa
178.50     b Freq Hz
300.00     c TBeg K
688.67     d |p| Pa G
-0.92599   e Ph(p) deg G
1.7189E-3  f |U| m^3/s G
2.3837     g Ph(U) deg G
air
1  RPN      Edot passing this point
2  DUCT     FIRST QUARTER WAVELENGTH
3  TBRANCH
-2.5622E+5  a Re(Zb) Pa-s/m^3 G
1.7433E+6   b Im(Zb) Pa-s/m^3 G
4  VESPEAKER FIRST DRIVER
5  RPN      driver resonance frequency
6  SOFTEND  backside of driver open to the room
           0.0000 a Re(z) =6G
           0.0000 b Im(z) =6H
7  DUCT     SECOND QUARTER WAVELENGTH
8  RPN      miscellaneous calculations, some needed later
9  TBRANCH
-4.5837E+5  a Re(Zb) Pa-s/m^3 G           700.64  A |p| Pa
1.7440E+6   b Im(Zb) Pa-s/m^3 G          -174.43  B Ph(p) deg
10 VESPEAKER SECOND DRIVER

```

```

11 SOFTEND    backside of driver open to the room
           0.0000 a Re(z)           =11G
           0.0000 b Im(z)           =11H
12 DUCT      THIRD AND FOURTH QUARTER WAVELENGTH
13 RPN       close the loop: |p1|
sameas      Od a =A?                688.67          A Pa
p1 mag
14 RPN       close the loop: phase(p1)
sameas      Oe a =A?                -0.92599          A deg
p1 arg
15 RPN       close the loop: |U1|
sameas      Of a =A?                1.7189E-3         A m3/s
U1 mag
16 RPN       close the loop: phase(U1)
sameas      Og a =A?                2.3837           A deg
U1 arg

```

## 4.8. Non-standard guesses

Like most other DELTAEC inputs, parameter **a** in an RPN segment can be used as a guess. For example, to let DELTAEC guess the temperature increment needed at the present location to hit a target downstream, guess input **a** in an RPN segment like this:

```

RPN
20.0    G
Tm inp + =Tm

```

As a contrived example, suppose that the magnetic field  $B$  in a speaker depends on temperature  $T_m$  according to a strange equation such as

$$B^3 - 5B^2 + 3 = T_m, \quad (4.3)$$

and we want DELTAEC to correctly incorporate a temperature-dependent  $Bl$  product in a SPEAKER segment. We don't know how to solve Eq. (4.3) for  $B$  analytically, but we can make DELTAEC find the root of the cubic equation numerically by guessing  $B$  and targeting Eq. (4.3):

```

!-----14-----
RPN (a)=magnetic field (tesla), guessed. 1.6 meters of wire. (A)= BL product.
12.863 a G or T A 20.581 B x L
inp 1.6 *
!-----15-----
RPN Target the strange equation in the Users Guide:
0.00 a G or T A 0.000 A zero
14a 3 ^ 14a 2 ^ 5 * - 3 + Tm -
!-----16-----
IESPEAKER a linear motor and piston
sameas 3a a Area m^2 2443.2 A |p| Pa
1.0000 b R ohms -177.67 B Ph(p) deg
3.0000E-03 c L H 9.4248E-03 C |U| m^3/s
sameas 14A d BLProd T-m 147.29 D Ph(U) deg
1.0000E-02 e M kg 9.4267 E Hdot W
1.0000E+04 f K N/m 9.4267 F Edot W
0.1000 g Rm N-s/m 20.919 G WorkIn W
4.7776 h |I| A G 14.031 H Volts V
0.0000 i Ph(I) deg 4.7776 I Amps A
51.381 J Ph(V/I) deg
2443.2 K |Px| Pa
ideal Solid type -177.67 L Ph(Px) deg

```

## 4.9. Linking multiple models

Two or more models can be open simultaneously in a single DELTAEC window, and it is sometimes useful to bring a number from one model into another model. For example, we have sometimes been interested in a few unequal engines and many unequal refrigerators



sharing a common resonator, requiring so many segments and so many guesses and targets that several models are needed to hold them all. How can such multiple models communicate with one another automatically? This is accomplished with an extended RPN pointer of the form `filename:addr` in an RPN formula.

For example, suppose that three models, named `<engine1.out>`, `<engine2.out>`, and `<engine3.out>`, which represent a cascade engine [19], are simultaneously open in DELTAEC. Suppose that hot-heat-exchanger heats are given at address `12e` in all three models. An RPN segment near the end of `<engine3.out>` could calculate the total heat required by the set of three engines with the following formula which brings results from `<engine1.out>` and `<engine2.out>` into `<engine3.out>`:

```
engine1:12e engine2:12e + 12e +
```

Extending this example, if the beginning of the `<engine2.out>` model is supposed to be fed by the end of the `<engine1.out>` model, where segment 87 is a `STK**` and 88 is a `HX**`, the key variables can be shared if the beginning of `<engine2.out>` looks like

```
! engine2.out
!----- 0 -----
BEGIN This segment is required, even though these variables are re-initialized below!
999. Pa a Mean P
999. Hz b Freq
999. K c TBeg
999. Pa d |p|
999. deg e Ph(p)
999. m3/s f |U|
999. deg g Ph(U)
helium Gas type
!----- 1 -----
RPN bring pm and f from the other model
engine1:0a =pm %:0b =f
!----- 2 -----
RPN bring Tm from the other model
engine1:87H =Tm
!----- 3 -----
RPN bring p1 and U1 from the other model
engine1:88B %:88A cplx =p1 %:88D %:88C cplx =U1
```

(The notation “%” in an interlinked RPN formula signifies “the previously named model.”)

To run all of the models open in DELTAEC together, use Run All, with capital-R as its keyboard shortcut. The order of running will be left to right in the sequence in which the model tabs are laid out. Use the mouse to drag tabs from side to side to change that order. (Drag a tab off the side, top, or bottom of the panel to split the panel into two or more displays so two or more models can be seen simultaneously.) Choose the order of running carefully, so each model has up-to-date values of any numbers that it gets from other models. For convenience, choose file names so the default alphabetical opening of multiple models puts them in the desired tab order.

If two or more models are run separately, any incremental plots in any of those models will run separately, just as if they were open in separate DELTAEC windows. However, Run All can optionally change this behavior dramatically, disabling any incremental plots in the separate models and following the choices established by the user in the Multi-Model Incremental Plot Control dialog, which is available in the Display pulldown menu. The Multi-Model Plot Control lets the user choose one or two of the independent plot variables that exist in the individual models as the overall independent plot variable (1-dimensional plot) or variables (2-dimensional plot), and choose from among the established dependent plot variables (including guesses) in the individual models too. A Run All will then run all

the models, increment an independent plot variable, run all the models again, etc., creating a table of results in a file called `<MultiMod.ip>`. For example, a set of linked models often share the same frequency, linked together via RPN formulas similar to `firstmodelname:Ob=f`. If parameter `Ob` in model `firstmodelname` is chosen as the independent Multi-Model Plot variable, then a Run All will sequentially increment the frequency in all of the linked models.

To view graphs of `<MultiMod.ip>` after such a Run All, click the appropriate button in the Multi-Model Plot Control. DELTAEC tries to include variables in the `<MultiMod.ip>` file that assist Rewinding when the user has ignored Forward-Reference-Condition warnings, so don't be surprised to find a few variables there that you don't remember including yourself.

Please do not create more than one set of such inter-linked models with Multi-Model Plotting in a single folder, because DELTAEC can have only one `<MultiMod.ip>` file per folder.

## 4.10. Other math segments

The `CONSTANTS` segment allows the user to insert up to a dozen real constants in one segment. These can then be referred to by other segments (via `sameas`) and can be modified, used as guesses, or used as independent incremental-plot variables. See details in Section 11.4.3.

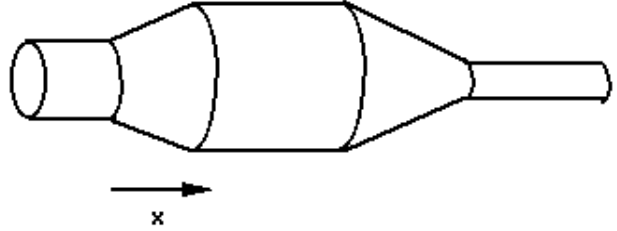
The `VOLUME` segment automatically adds up the volume in a series of segments. See details in Section 11.4.2. `VOLUME` adds up the volume of gas plus the volume of the solids in porous structures such as heat exchangers, so it gives one estimate of the size of an apparatus. (The RPN variable `vol`, described in Section 11.4.1, is similar to the segment `VOLUME`, but it accesses the cumulative *gas* volume in the model, without involving the solid volume in the porous segments.)

# 5. ENGINES & REFRIGERATORS

The examples given in the previous Chapters were relatively simple cases of linear acoustics. In this Chapter, we introduce the full thermoacoustic computing power of DELTAEC with some simple engines and refrigerators. After discussing the principles and assumptions that are built into the computation, we present example calculations.

## 5.1. Principles of calculation

DELTAEC processes one-dimensional strings of acoustic and thermoacoustic elements (or networks of such strings—see Section 2.7.5 and Chapter 11 for TBRANCHs and UNIONS), so the one-dimensional “wave” equation is of the greatest importance. We always assume a time dependence of  $e^{i\omega t}$ , so the “wave” equation is the second-order Helmholtz differential equation for the complex pressure amplitude  $p_1(x)$ . As was described at the beginning of Chapter 2, we find it easiest to think of this as two coupled first-order equations in pressure  $p_1$  and volume flow rate  $U_1$ . This point of view is taken in Refs. [13] and [20]. In coupled, first-order form, the equations are ready for numerical integration along  $x$ .



In Chapter 2, we introduced such equations for DUCTs and CONEs, in which  $T_m$  did not vary with  $x$ . Nicholas Rott [21] derived the corresponding equation when  $dT_m/dx \neq 0$ :

$$\left(1 + \frac{(\gamma - 1)f_\kappa}{1 + \epsilon_s}\right)p_1 + \frac{\rho_m a^2}{\omega^2} \frac{d}{dx} \left(\frac{1 - f_\nu}{\rho_m} \frac{dp_1}{dx}\right) - \beta \frac{a^2}{\omega^2} \frac{(f_\kappa - f_\nu)}{(1 - \sigma)(1 + \epsilon_s)} \frac{dT_m}{dx} \frac{dp_1}{dx} = 0, \quad (5.1)$$

which can be rewritten as two coupled first-order equations:

$$\begin{aligned} \frac{dp_1}{dx} &= -\frac{i\omega\rho_m}{A(1 - f_\nu)}U_1, \\ \frac{dU_1}{dx} &= -\frac{i\omega A}{\rho_m a^2} \left(1 + \frac{(\gamma - 1)f_\kappa}{1 + \epsilon_s}\right)p_1 + \frac{(f_\kappa - f_\nu)}{(1 - \sigma)(1 + \epsilon_s)}\beta \frac{dT_m}{dx}U_1. \end{aligned} \quad (5.2)$$

These equations are used in most of the STK\*\* segments introduced below, which accommodate  $dT_m/dx \neq 0$  in engines and refrigerators. The complex variables  $f_\nu$  and  $f_\kappa$  depend on the size and shape of the pores and on the gas properties. Of course the right sides of these

equations depend implicitly on the local value of  $T_m$ , because gas properties such as  $\rho_m$  and  $a$  depend on  $T_m$ .

To complete this picture for many thermoacoustic engines and refrigerators, only two more equations are required, i.e., those for the temperature  $T_m(x)$  and the total power  $\dot{H}_{\text{tot}}(x)$ . As for  $p_1(x)$  and  $U_1(x)$ , the equations for  $T_m(x)$  and  $\dot{H}_{\text{tot}}(x)$  depend on the type of segment, and  $T_m$  and  $\dot{H}_{\text{tot}}$  are continuous across the junctions between segments.

In DELTAEC, the total power is treated as the sum of two terms,

$$\dot{H}_{\text{tot}} = \dot{H}_{2,k} + \dot{H}_{\dot{N}}. \quad (5.3)$$

The first term,  $\dot{H}_{2,k}$ , represents power carried in the  $x$  direction by second-order thermoacoustic effects, generally proportional to the lateral spatial average of  $\text{Re}[T_1\tilde{u}_1]$ , and by zeroth-order thermal conduction. The second term,  $\dot{H}_{\dot{N}}$ , represents power carried by nonzero time-averaged flow of gas in the  $x$  direction. In the present chapter, we consider only zero time-averaged flow, so  $\dot{H}_{\text{tot}} = \dot{H}_{2,k}$ . (Nonzero  $\dot{H}_{\dot{N}}$  is introduced in Chapter 6.)

Most segments, such as DUCTs and CONEs described in Chapter 2, obey simply  $dT_m/dx = 0$  and  $d\dot{H}_{\text{tot}}/dx = 0$ .

STK\*\* segments have a more complicated differential equation for  $T_m(x)$ , which is derived from Rott's energy equation [22],

$$\begin{aligned} \dot{H}_{2,k} &= \frac{1}{2} \text{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{f_\kappa - \tilde{f}_\nu}{(1 + \epsilon_s)(1 + \sigma)(1 - \tilde{f}_\nu)} \right) \right] \\ &+ \frac{\rho_m c_p |U_1|^2}{2A_{\text{gas}} \omega (1 - \sigma) |1 - f_\nu|^2} \frac{dT_m}{dx} \text{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1 + \epsilon_s f_\nu / f_\kappa)}{(1 + \epsilon_s)(1 + \sigma)} \right] \\ &- (A_{\text{gas}} k + A_{\text{solid}} k_s) \frac{dT_m}{dx}, \end{aligned} \quad (5.4)$$

by solving it algebraically for  $dT_m/dx$ .

The total power flow  $\dot{H}_{\text{tot}}$  through a STK\*\* segment is assumed to be independent of  $x$ , because the side walls are assumed to be rigid and thermally insulated. The total power  $\dot{H}_{\text{tot}}$  changes with  $x$  in heat exchanger segments, according to

$$\frac{d\dot{H}_{\text{tot}}}{dx} = \dot{q}, \quad (5.5)$$

where  $\dot{q}$  is the heat added per unit length in the heat exchanger.

(There is also a ‘‘thermally anchored’’ mode in which  $d\dot{H}_{2,k}/dx = d\dot{E}/dx$  in DUCTs, CONEs, etc. to reflect the assumption that such segments are immersed in heat sinks; this exception to the normal circumstances will be described in Section 11.3.1.)

So, for thermoacoustic calculations, DELTAEC integrates from the BEGINning through the sequence of subsequent segments, with respect to two complex and two real variables:  $p_1(x)$ ,  $U_1(x)$ ,  $T_m(x)$ , and  $\dot{H}_{\text{tot}}(x)$ . DELTAEC uses continuity of  $p_1$ ,  $U_1$ ,  $T_m$ , and  $\dot{H}_{\text{tot}}$  to pass from the end of one segment to the beginning of the next. Within each segment, it uses equations appropriate to that segment type, governed by local parameters, such as area and perimeter, by global parameters, such as frequency and mean pressure, and by evolving

parameters such as  $p_1$ ,  $U_1$ ,  $T_m$ , and  $\dot{H}_{\text{tot}}$  themselves and the properties of the gas and solid. Mathematically, the DELTAEC integration can be regarded as

$$\frac{dp_1}{dx} = \mathcal{F}_{\text{momentum}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega; \text{geometry, gas properties, etc.}), \quad (5.6)$$

$$\frac{dU_1}{dx} = \mathcal{F}_{\text{continuity}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \text{geometry, gas properties, etc.}), \quad (5.7)$$

$$\frac{dT_m}{dx} = \mathcal{F}_{x\text{-energy}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \text{geometry, gas properties, etc.}), \quad (5.8)$$

$$\frac{d\dot{H}_{\text{tot}}}{dx} = \mathcal{F}_{\text{lateral energy}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \text{geometry, gas properties, etc.}). \quad (5.9)$$

In reality, as documented here and in Chapter 10, most of the equations for specific segments are simpler than what’s suggested by the fully general forms of Eqs. (5.6)–(5.9). For example, the right-hand side of the momentum equation never depends explicitly on  $\dot{H}_{\text{tot}}$ ; many segment types have simply  $dU_1/dx = 0$  or  $dT_m/dx = 0$  or can calculate  $\Delta p_1$  algebraically instead of with a numerical integration; and  $d\dot{H}_{\text{tot}}/dx$  is usually zero or as simple as Eq. (5.5). Nevertheless, it is useful to keep in mind the fully general picture of Eqs. (5.6)–(5.9), to help remember what DELTAEC needs to know at each location  $x$  as it proceeds with its numerical integration.

## 5.2. Hofler’s refrigerator

Tom Hofler’s thermoacoustic refrigerator was described in detail in his Ph.D. thesis [1] and was briefly summarized in Ref. [2]. We use this case to further illustrate capabilities of DELTAEC, generating curves similar to Figs. 16 and 17 in Ref. [1] (Figs. 5 and 6 in Ref. [2]).

The apparatus is shown here in Fig. 5.1. We begin with a file (`<hofler1.out>`, in the `<Examples\EnginesAndRefr>` folder) whose geometry is that of Hofler’s “long” apparatus. (We originally typed this file in a Notepad text editor while studying Ref. [1].) The list of segments serves as an outline of the model:

```
0 BEGIN:
1 SURFACE: driver end
2 DUCT : ambient temperature duct
3 HX : ambient temperature heat exchanger
4 STKSLAB: Stack
5 HX : Cold heat exchanger
6 DUCT : Cold Duct
7 CONE :
8 COMPLIANCE: end bulb
```

and clicking the toolbar “+” shows details:

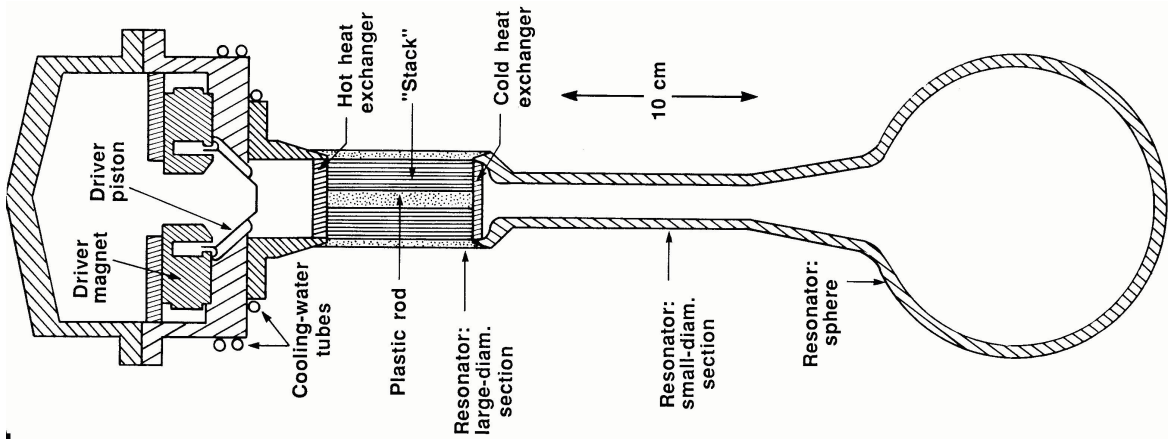


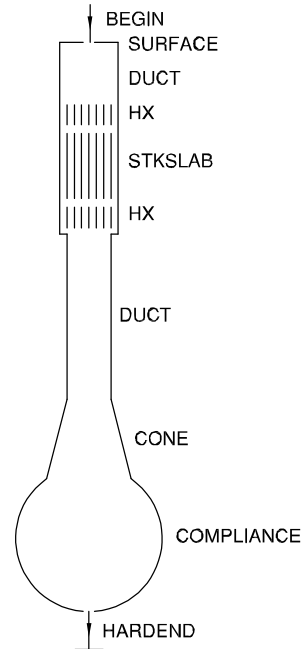
Figure 5.1: Hofler's thermoacoustic refrigerator.

TITLE Hofler's 1986 thermoacoustic refrigerator, hofler1.out  
 !Created@16:29:03 1-Oct-06 with DeltaE Vers. 6.0g2 for the IBM/PC-Compatible

```

----- 0 -----
BEGIN
1.0000E+06 a Mean P Pa
500.00 b Freq Hz
300.00 c TBeg K
3.0000E+04 d |p| Pa
0.0000 e Ph(p) deg
5.0000E-04 f |U| m^3/s
0.0000 g Ph(U) deg
helium Gas type
----- 1 -----
SURFACE driver end
1.1340E-03 a Area m^2
0.0000 A |p| Pa
0.0000 B Ph(p) deg
0.0000 C |U| m^3/s
0.0000 D Ph(U) deg
0.0000 E Htot W
0.0000 F Edot W
ideal Solid type
----- 2 -----
DUCT ambient temperature duct
sameas 1a a Area m^2
0.1190 b Perim m
4.2600E-02 c Length m
0.0000 A |p| Pa
0.0000 B Ph(p) deg
0.0000 C |U| m^3/s
0.0000 D Ph(U) deg
0.0000 E Htot W
0.0000 F Edot W
ideal Solid type
----- 3 -----
HX ambient temperature heat exchanger
sameas 1a a Area m^2
0.6000 b GasA/A
6.3500E-03 c Length m
1.9000E-04 d y0 m
-9.6400 e HeatIn W
300.00 f Est-T K (t)
0.0000 A |p| Pa
0.0000 B Ph(p) deg
0.0000 C |U| m^3/s
0.0000 D Ph(U) deg
0.0000 E Htot W
0.0000 F Edot W
0.0000 G GasT K
0.0000 H SolidT K
copper Solid type
----- 4 -----
STKSLAB Stack
sameas 1a a Area m^2
0.7240 b GasA/A
7.8500E-02 c Length m
1.8000E-04 d y0 m
4.0000E-05 e Lplate m
0.0000 A |p| Pa
0.0000 B Ph(p) deg
0.0000 C |U| m^3/s
0.0000 D Ph(U) deg
0.0000 E Htot W
0.0000 F Edot W
0.0000 G TBeg K
0.0000 H TEnd K
kapton Solid type
----- 5 -----
HX Cold heat exchanger
sameas 1a a Area m^2
0.6700 b GasA/A
2.5400E-03 c Length m
2.5500E-04 d y0 m
3.0000 e HeatIn W
200.00 f Est-T K (t)
0.0000 A |p| Pa
0.0000 B Ph(p) deg
0.0000 C |U| m^3/s
0.0000 D Ph(U) deg
0.0000 E Htot W
0.0000 F Edot W
0.0000 G GasT K
0.0000 H SolidT K
copper Solid type

```



```

!----- 6 -----
DUCT      Cold Duct
3.8400E-04 a Area      m^2      0.0000 A |p|      Pa
6.9400E-02 b Perim    m          0.0000 B Ph(p)    deg
0.1670   c Length     m          0.0000 C |U|      m^3/s
          0.0000 D Ph(U)    deg
          0.0000 E Htot     W
ideal     Solid type
!----- 7 -----
CONE
sameas    6a a AreaI    m^2      0.0000 A |p|      Pa
sameas    6b b PerimI   m          0.0000 B Ph(p)    deg
6.6800E-02 c Length    m          0.0000 C |U|      m^3/s
1.1600E-03 d AreaF     m^2      0.0000 D Ph(U)    deg
0.1210   e PerimF     m          0.0000 E Htot     W
ideal     Solid type
!----- 8 -----
COMPLIANCE end bulb
4.9000E-02 a SurfAr    m^2      0.0000 A |p|      Pa
1.0600E-03 b Volume    m^3      0.0000 B Ph(p)    deg
          0.0000 C |U|      m^3/s
          0.0000 D Ph(U)    deg
          0.0000 E Htot     W
ideal     Solid type

```

The segments that were not introduced in Chapter 2 are **HX** and **STKSLAB**.

**HX**'s have parallel-plate geometry, with plate spacing  $2y_0$ . Wave propagation through heat exchangers is computed using the momentum and continuity equations, including both viscous and thermal dissipation, much like the calculations in **DUCTS**. There is one additional input: the heat added to the thermoacoustic wave from an external thermal reservoir (or, if negative, the heat rejected to the external reservoir). The heat flow determines the change in total energy flow  $\dot{H}_{\text{tot}}$  in the heat exchanger. As an input variable, heat flow can either be specified by the user (and, optionally, an independent plot variable) or it can be a guess. There is also a potential target: the temperature of the solid in the heat exchanger, which is assumed to be spatially uniform and which differs from the temperature in the gas in proportion to the heat, as described in Chapter 10. The temperature difference between gas and solid can presently be computed only with an accuracy of about a factor of 2, even in the acoustic approximation; nevertheless, it is included in **DELTAEC**, to prevent naive users from being led to designs with heat exchangers of negligible surface area that have negligible losses and that would appear to have no disadvantages if this temperature difference were not included. We hope that future revisions of **DELTAEC** will have a more accurate calculation algorithm for this effect. Meanwhile, however, if you prefer not to use this feature, watch or target the gas mean temperature instead of the solid temperature by using **RPN** (see Chapter 4) or observing the gas temperature in the adjacent stack segment (parameter **G** or **H**).

**STKSLAB** also has parallel-plate geometry. This is the first segment we have encountered in the Tutorial in which  $T_m$  can depend on  $x$ . **DELTAEC** obtains the  $x$  dependence by integrating the thermoacoustic energy equation, schematically given by Eq. (5.8) and in detail in Eqs. (5.4) and the equations given in Section 10.5.1, with  $\dot{H}_{\text{tot}}$  independent of  $x$ .

Using the lessons we learned in Chapter 2, we recognize that we do not know all the initial conditions at **BEGIN**, so we must introduce some necessary guesses and targets. We know that zero  $U_1$  must flow through the bottom of the refrigerator, so we insert a **HARDEND** there and use its two impedance targets. Corresponding to those targets, we'll guess two variables in **BEGIN**. These might be complex  $U_1$ , but Hofler always ran his refrigerator on resonance, so we'll guess  $|U_1|$  and frequency  $f$  as guesses to get started, leaving the phases of  $p_1$  and  $U_1$  equal (both zero) as the condition for resonance at the driver. After adding these variables, the guess–target list looks like

GUESS	Ob	Of
name	BEGIN:Freq	BEGIN: U
units	Hz	m <sup>3</sup> /s
value	5.00E+02	5.00E-04
TARGET	9a	9b
name	HARDE:R(1/z	HARDE:I(1/z
value	.00	.00

Running this case yields  $f = 498$  Hz and  $|U_1| = 5.50 \times 10^{-4}$  m<sup>3</sup>/s. We are gratified that none of the results thus far appear crazy. If you are running this file for yourself, note also that the solid is cooler than the gas in the ambient heat exchanger, where the (negative) heat flows from gas to solid, and that the solid is warmer than the gas in the cold heat exchanger, where the (positive) heat flow is from solid to gas.

Now consider what guesses and targets might be required by the heat and temperature aspects of this model. We know that the bottom of the apparatus was thermally insulated, so no energy can flow in or out there. We also note from Ref. [1] that Hofler maintained the solid temperature in the ambient heat exchanger at 300 K by vigorous flow of temperature-regulated water through it. These two constraints, not at the BEGINning, call for the use of two targets: 9c and 3f. Hofler applied a known electrical heater power to the cold heat exchanger, so we need not guess that; we leave its value at 3.00 W for now. However, he did not control the heat removed at the ambient heat exchanger, nor the temperature of the *gas* at the ambient end; so DELTAEC must guess those two variables: 0c and 3e. After making the necessary changes, the guess–target summary looks like

GUESS	Ob	0c	Of	3e
name	BEGIN:Freq	BEGIN:TBeg	BEGIN: U	HX:HeatI
units	Hz	K	m <sup>3</sup> /s	W
value	498.02	300.0	5.054E-04	-9.640
TARGET	3f	9a	9b	9c
name	HX:Est-T	HARDE:R(1/z	HARDE:I(1/z	HARDE:Hdot
units	K			W
value	300.0	0.00	0.00	0.00

before running, and

GUESS	Ob	0c	Of	3e
name	BEGIN:Freq	BEGIN:TBeg	BEGIN: U	HX:HeatI
units	Hz	K	m <sup>3</sup> /s	W
value	504.22	300.80	5.4172E-04	-11.126
TARGET	3f	9a	9b	9c
name	HX:Est-T	HARDE:R(1/z	HARDE:I(1/z	HARDE:Htot
units	K			W
value	300.00	0.0000	0.0000	0.0000
result	300.00	-6.0493E-12	-1.0355E-10	1.7764E-15

after running. By examining segment 5 in the `<.out>` file, we see that DELTAEC predicts a cold temperature of 223.76 K with this 3.00 W heat load on the cold heat exchanger.

All of these results are reasonable, so now we examine Ref. [1] very closely to consider what else we should do to compare DELTAEC results directly to the published results of Hofler’s experiments. Hofler added the *calculated* dissipation in the DUCT, CONE, and COMPLIANCE to his electrically applied heat load, to plot the gross cooling power instead of the net cooling power. So we do the same, with an RPN segment at the end of the model. We add another RPN to display another of his plotted variables, the temperature ratio. On the basis of the gross cooling power and the measured acoustic power delivered to the resonator by the driver, Hofler also plotted the coefficient of performance,  $COP = \dot{Q}_{C,gross} / \dot{E}$ , relative to Carnot’s COP,  $COP_{Carnot} = T_C / (T_0 - T_C)$ , so we calculate  $COP / COP_{Carnot}$  with a third RPN at the end. These additions look like



```

+ 0 BEGIN
+ 1 SURFACE      driver end
+ 2 DUCT        ambient temperature duct
+ 3 HX          ambient temperature heat exchanger
+ 4 STKSLAB     Stack
+ 5 HX          Cold heat exchanger
+ 6 DUCT        Cold Duct
+ 7 CONE
+ 8 COMPLIANCE end bulb
+ 9 HARDEND     target this to seal the end
!----- 10 -----
RPN      gross cooling power is net power plus tail dissipation
0.0000  a G or T      (t)      3.8487  A Qg (W)
5e 5F +
!----- 11 -----
RPN      temperature ratio, Tc/T0
0.0000  a G or T      (t)      0.7459  A ratio
5H 3H /
!----- 12 -----
RPN      COP / Carnot's COP
0.0000  a G or T      (t)      0.1614  A COP rel
10A 1F / 5H 3H 5H - / /

```

We let the net cooling power, i.e., heat **5e** at the cold heat exchanger, be the independent plot variable in DELTAEC, ranging from 1 to 7 W in 0.5 W steps. To obtain the variables we want to display, we add **10A**, **11A**, and **12A** to the list of dependent plot variables. After running this case, we changed  $|p_1|$  to 0.015 of  $p_m$ , and changed **5e** to 0.5 W, making those changes by hand in a couple of steps to make sure DELTAEC did not get lost. Then we again used **5e** as the independent plot variable, over the range of  $\dot{Q}_C$  to 0.5–3.5 W in steps of 0.25 W, and ran it again, appending the results to the previous plot run. We imported the resulting `<.ip>` file into a spreadsheet so we could display it together with Hofler’s experimental data. The results are shown in Figs. 5.2. The DELTAEC calculations come reasonably close to the measurements presented in Figs. 16 and 17 of Hofler’s thesis [1].

Let’s modify this example a little more, using it to illustrate some points about side-branch and series transducers and about DELTAEC’s consistency with the first law of thermodynamics. First, let’s add a more realistic driver to the system, using **VSPEAKER**. We insert **VSPEAKER** at the beginning, using Ref. [1] for approximate numerical values for the mass, resistance, and force constant for the speaker, and we delete the **SURFACE** segment that was previously near the beginning because **VSPEAKER** accounts for the oscillatory pressurization losses on its surface area. We use the drive voltage as a guess, so we are asking DELTAEC what voltage we must supply to maintain  $|p_1| = 30$  kPa in front of the speaker; we estimate it will take about 20 V to drive it, so we use that as our starting value for the guess. We’ll also add an RPN target after the **VSPEAKER**, to enforce resonance by ensuring that the phases of  $p_1$  and  $U_1$  are equal at the driver. We do this by forcing their difference, computed by subtracting the values appearing as results **1B** and **1L**, to be zero, which is the value given in line 2a.

With this change, **BEGIN** is no longer at a position of nonzero  $|U_1|$ ; **BEGIN** is now at a place where  $U_1 = 0$  and  $p_1$  is nonzero, and the subsequent side-branch **VSPEAKER** adds all the volume flow rate that flows into the subsequent segments. We change guesses and initial values accordingly. With these changes, but before running, the guess–target list becomes

GUESS	0b	0c	0e	1h	4e
name	BEGIN:Freq	BEGIN:TBeg	BEGIN:Ph(p)	VSPEA:AplVo	HX:HeatI
units	Hz	K	deg	V	W
value	504.22	300.80	100.00	20.000	-11.126
TARGET	2a	4f	10a	10b	10c
name	RPN:Target	HX:Est-T	HARDE:R(1/z)	HARDE:I(1/z)	HARDE:Htot
units		K			W
value	0.0000	300.00	0.0000	0.0000	0.0000
result	0.0000	300.00	-7.8635E-15	5.8006E-14	-1.7764E-15

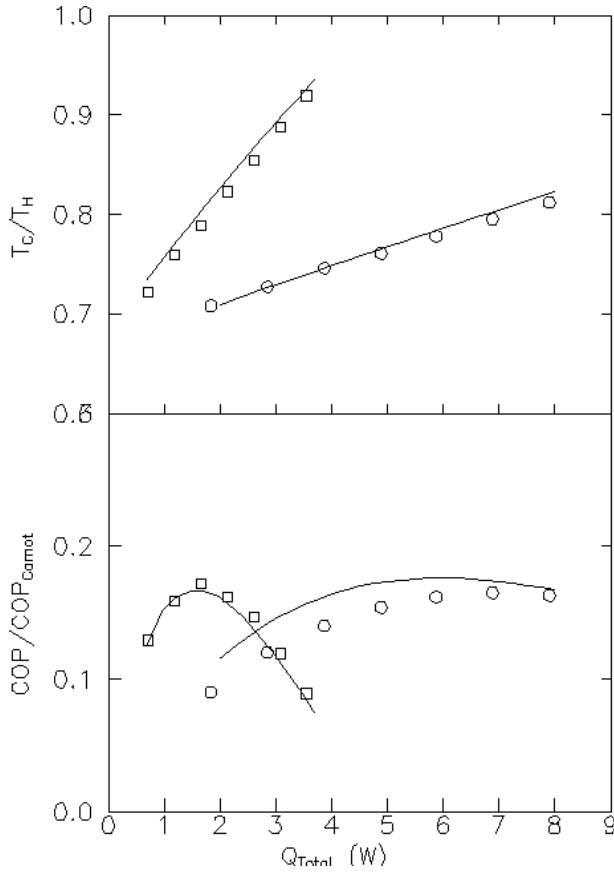


Figure 5.2: Hofler refrigerator results. Lines are DELTAEC results; points are from experimental data presented in Hofler's thesis. Squares,  $p_1 = 0.015p_m$ . Circles,  $p_1 = 0.03p_m$ .

and when we run it the guess-target list becomes

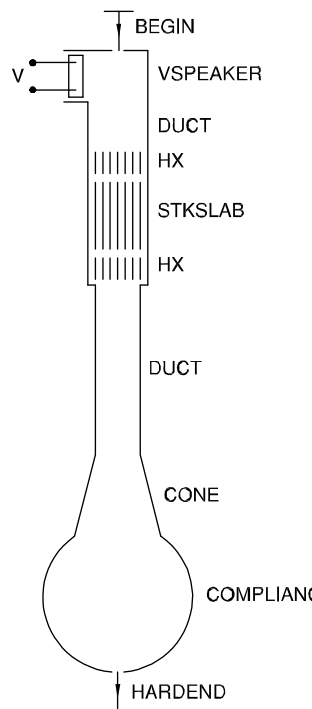
GUESS	0b	0c	0e	1h	4e
name	BEGIN:Freq	BEGIN:TBeg	BEGIN:Ph(p)	VSPEA:AplVo	HX:HeatI
units	Hz	K	deg	V	W
value	505.59	302.60	152.52	23.385	-36.043
TARGET	2a	4f	10a	10b	10c
name	RPN:Targe	HX:Est-T	HARDE:R(1/z)	HARDE:I(1/z)	HARDE:Htot
units		K			W
value	0.0000	300.00	0.0000	0.0000	0.0000
result	1.7536E-11	300.00	2.5373E-12	-2.5260E-11	-4.2633E-12

and the <.out> file becomes

```

TITLE      Hofler's 1986 thermoacoustic refrigerator
!Created@21:47:02 2-Oct-06 with DeltaE Vers. 6.0g2 for the IBM/PC-Compatible
!----- 0 -----
BEGIN
1.0000E+06 a Mean P      Pa
505.59    b Freq        Hz      G
302.60    c TBeg       K        G
3.0000E+04 d |p|         Pa
152.52    e Ph(p)      deg     G
0.0000    f |U|       m^3/s
0.0000    g Ph(U)     deg
helium    Gas type
!----- 1 -----
VSPEAKER  electrodynamic driver
6.0000E-04 a Area      m^2      3.0000E+04 A |p|      Pa
6.0000    b R         ohms     152.52    B Ph(p)    deg
0.0000    c L         H        5.4225E-04 C |U|      m^3/s
8.0000    d BLProd   T-m      152.52    D Ph(U)    deg
5.0000E-03 e M         kg
0.0000    f K         N/m
0.0000    g Rm       N-s/m
23.385    h AplVol   V        G
-11.156   J Ph(V/I)  deg
5.4348E-04 K |Ux|    m^3/s
152.52    L Ph(-Ux) deg
ideal     Solid type
!----- 2 -----
RPN       enforce resonance
0.0000   a G or T      = 2A?    1.7536E-11 A null
1B 1L -
+ 3 DUCT      ambient temperature duct
+ 4 HX        ambient temperature heat exchanger
+ 5 STKSLAB   Stack
+ 6 HX        Cold heat exchanger
+ 7 DUCT      Cold Duct
+ 8 CONE
+ 9 COMPLIANCE end bulb
+ 10 HARDEND  target this to seal the end
+ 11 RPN      gross cooling power is net power plus tail dissipation
+ 12 RPN      temperature ratio, Tc/TO
+ 13 RPN      COP / Carnot's COP

```



Comparing the values of the guesses here with those before we inserted the **VSPEAKER** shows two big differences: the phase of the pressure and the heat rejected at the ambient heat exchanger.

The change in the phase of the pressure is essentially trivial: The **VSPEAKER** segment sets the overall time phase of everything in the model, because the phase of its voltage is fixed at zero. Previously, the phase of  $U_1$  at the top of the model was arbitrarily set to zero by the **BEGIN** segment. Now, the **VSPEAKER** controls the overall time phase of the model, and the nonzero imaginary part of the speaker's mechanical impedance causes a large phase shift between  $V_1$  and  $U_1$ . This simply shifts all  $p_1$  and  $U_1$  phasors by about  $30^\circ$ . A further  $180^\circ$  phase shift enters because of the sign convention for positive voltage built into the segment.

The increase in the heat rejected at the ambient heat exchanger from 11.1 W to 36.0 W is meaningful. The extra heat comes from the **VSPEAKER**, and illustrates how **DELTAEC** handles energy for **\*\*SPEAKERS** and **\*\*DUCERS**: They are treated as thermally insulated, so all of the electrical energy going into them shows up either as acoustic power or as heat. This transducer is not remarkably efficient at converting electrical power into acoustic power; most

of its electrical power becomes heat, which is rejected at the ambient heat exchanger in this model. The extra load on the ambient heat exchanger increases the temperature difference in that heat exchanger a little, which leads to the small increase in gas temperature seen in 0c, which in turn leads to the small increase in resonance frequency seen in 0b.

The speaker can also be included in the model “in series,” using a **VESPEAKER** segment, with yet another reconfiguration of the **BEGIN** and associated guesses and targets to reflect new circumstances. As shown below, we include the cavity above the speaker, which appears from Ref. [1] to have a volume of approximately a liter. Above that volume, the **BEGIN** starts with  $U_1 = 0$  and with a guessed  $p_1$ . We arbitrarily set the time phase of the entire model by setting the phase of  $p_1$  in that volume to zero in the **BEGIN**. We note that Hofler had cooling-water tubes attached to the driver housing, to try to remove as much of the driver’s electrical and other dissipation as possible. To crudely model the effect of these tubes, we insert an **RPN** below the driver, which we craft so it removes 75% of the driver’s heat, to illustrate the use of **RPN** for such purposes. With these changes, and after running, the guess–target list becomes

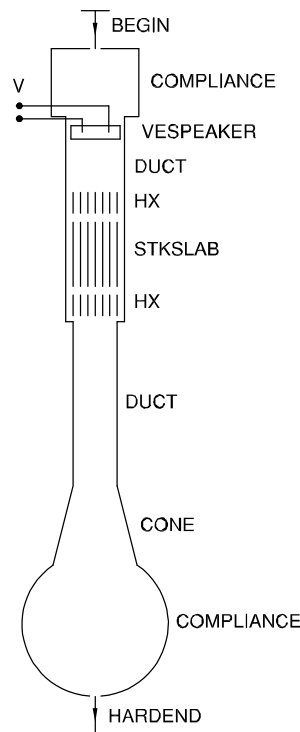
GUESS	0b	0c	0d	2h	2i	7e
name	BEGIN:Freq	BEGIN:TBeg	BEGIN: p	VESPE: V	VESPE:Ph(V)	HX:HeatI
units	Hz	K	Pa	V	deg	W
value	504.56	301.24	284.59	23.271	116.88	-17.243
TARGET	3a	4a	7f	13a	13b	13c
name	RPN:Target	RPN:Target	HX:Est-T	HARDE:R(1/z)	HARDE:I(1/z)	HARDE:Htot
units			K			W
value	0.0000	3.0000E+04	300.00	0.0000	0.0000	0.0000
result	-5.5422E-13	3.0000E+04	300.00	1.7565E-14	1.8398E-13	2.4869E-14

and the `<.out>` file becomes

```

TITLE      Hofler's 1986 thermoacoustic refrigerator
!Created@18:52:37 3-Oct-06 with DeltaE Vers. 6.0g2 for the IBM/PC-Compatible
!----- 0 -----
BEGIN
1.0000E+06 a Mean P      Pa
504.56   b Freq          Hz      G
301.24   c TBeg         K        G
284.59   d |p|          Pa      G
0.0000   e Ph(p)        deg
0.0000   f |U|          m^3/s
0.0000   g Ph(U)        deg
helium   Gas type
!----- 1 -----
COMPLIANCE the space above the speaker
0.1000   a SurfAr      m^2      284.59   A |p|      Pa
1.0000E-03 b Volume      m^3      0.0000   B Ph(p)    deg
5.4133E-04 c |U|        m^3/s    5.4133E-04 C |U|    m^3/s
-90.206   d Ph(U)      deg      -90.206   D Ph(U)    deg
0.0000   e Htot       W        0.0000   E Htot     W
-2.7680E-04 F Edot      W        -2.7680E-04 F Edot     W
ideal    Solid type
!----- 2 -----
VESPEAKER electrodynamic driver
6.0000E-04 a Area      m^2      3.0000E+04 A |p|      Pa
6.0000   b R          ohms     -90.207   B Ph(p)    deg
0.0000   c L          H        5.4010E-04 C |U|    m^3/s
8.0000   d BLProd    T-m      -90.207   D Ph(U)    deg
5.0000E-03 e M        kg       32.668   E Htot     W
0.0000   f K          N/m      8.1015   F Edot     W
0.0000   g Rm        N-s/m    32.668   G WorkIn   W
23.271   h |V|        V          23.271   H Volts    V
116.88   i Ph(V)     deg      2.8605   I Amps     A
-11.041   J Ph(V/I)  deg     3.0002E+04 K |Px|    Pa
-90.751   L Ph(Px)  deg     -90.751   L Ph(Px)  deg
!----- 3 -----
RPN      enforce resonance
0.0000   a G or T      = 3A?   -5.5422E-13 A null
2B 2D -
!----- 4 -----
RPN      enforce desired pressure amplitude
3.0000E+04 a G or T      = 4A?   3.0000E+04 A |p1|
p1 mag
!----- 5 -----

```



```

RPN      (a)= % of driver heat removed by water tubes on driver box
0.7500  a G or T      (t)      14.243  A Watt
                                18.425  B Watt
Htot Edot - inp * sto Htot rcl - =H2k
!----- 6 -----
DUCT      ambient temperature duct
1.1340E-03 a Area      m^2      2.9736E+04 A |p|      Pa
0.1190  b Perim      m      -90.401  B Ph(p)      deg
4.2600E-02 c Length  m      2.8084E-03 C |U|      m^3/s
                                -169.43  D Ph(U)      deg
                                14.243  E Htot      W
                                7.9452  F Edot      W
ideal      Solid type
+ 7 HX      ambient temperature heat exchanger
+ 8 STKSLAB Stack
+ 9 HX      Cold heat exchanger
+ 10 DUCT   Cold Duct
+ 11 CONE
+ 12 COMPLIANCE end bulb
+ 13 HARDEND target this to seal the end
+ 14 RPN    gross cooling power is net power plus tail dissipation
+ 15 RPN    temperature ratio, Tc/TO
+ 16 RPN    COP / Carnot's COP

```

Notice how the DELTAEC RPN operator =H2k has been used to override DELTAEC's normal continuity of  $\dot{H}$  between segments, to simulate the removal of 18.425 W of heat by the water-cooled tubing on Hoffer's driver housing.

The standing-wave components of the pressure and volume-flow-rate waves in this refrigerator model are shown in Fig. 5.3—it is essentially a quarter-wave device. In DELTAEC models of thermoacoustic engines and refrigerators, it is often also worthwhile to examine graphs of  $\dot{H}$  and  $\dot{E}$  and to think about their consistency with the laws of thermodynamics, e.g., to reveal possible fuzzy thinking about how BEGIN and \*\*END are set up in the model. In the present case, shown in Fig. 5.4, the acoustic power delivered by the driver drops most rapidly in the stack and more slowly in the bottom part of the resonator, and the constancy of  $\dot{H}_{\text{tot}}$  in each insulated segment (i.e., everywhere but in heat exchangers) is evident. The positive signs of these power flows indicate flows in the positive- $x$  direction, i.e., from the BEGIN toward the HARDEND in the model.

Another instructive graph displays gas temperature  $T_m$  and solid temperature  $T_{\text{solid}}$  as functions of  $x$ , as illustrated for the present example in Fig. 5.5. In heat exchangers, the difference  $T_{\text{solid}} - T_m$  is due to the time-averaged heat flowing from solid to gas. In other segments, where there is no time-averaged heat flowing from solid to gas, these two temperatures are equal.

### 5.3. Choosing guesses and targets

The final example in the previous section used six targets and six guesses—the most complicated guess–target summary table we have yet encountered in the Users Guide. In this section, we discuss how we chose its members, and explain some general guidelines for guess–target choices.

Recall the purpose of guesses and targets. DELTAEC is numerically integrating equations

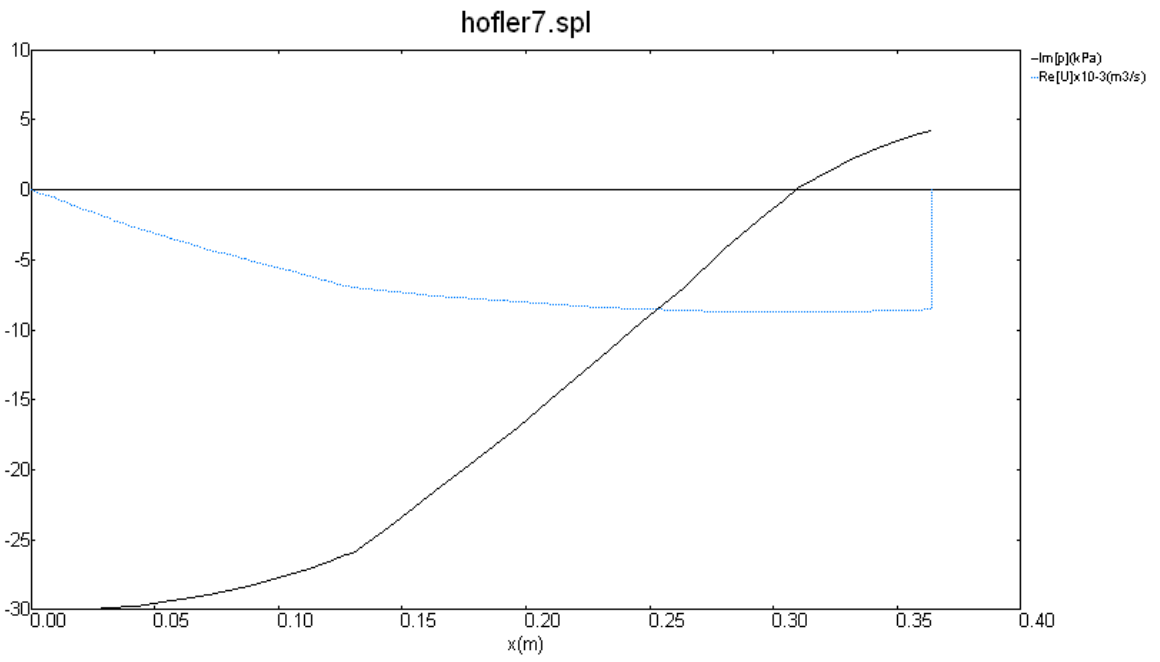


Figure 5.3: Standing-wave parts of the  $p_1$  and  $U_1$  wave in Hoffer's refrigerator.

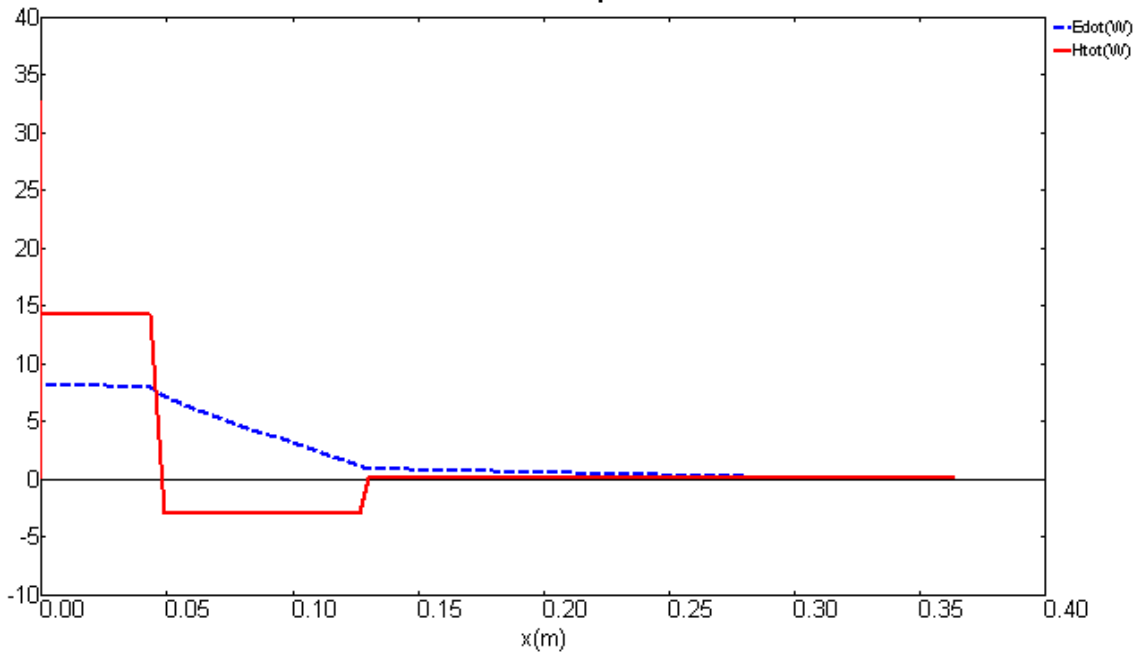


Figure 5.4: Acoustic power  $\dot{E}$  (dashed blue) and total power  $\dot{H}$  (solid red) as a function of  $x$  in the refrigerator model. The BEGIN, COMPLIANCE, and VSPEAKER are all crowded into  $x = 0$ . The drop in  $\dot{H}$  marks the ambient heat exchanger and the smaller rise in  $\dot{H}$  marks the cold heat exchanger.

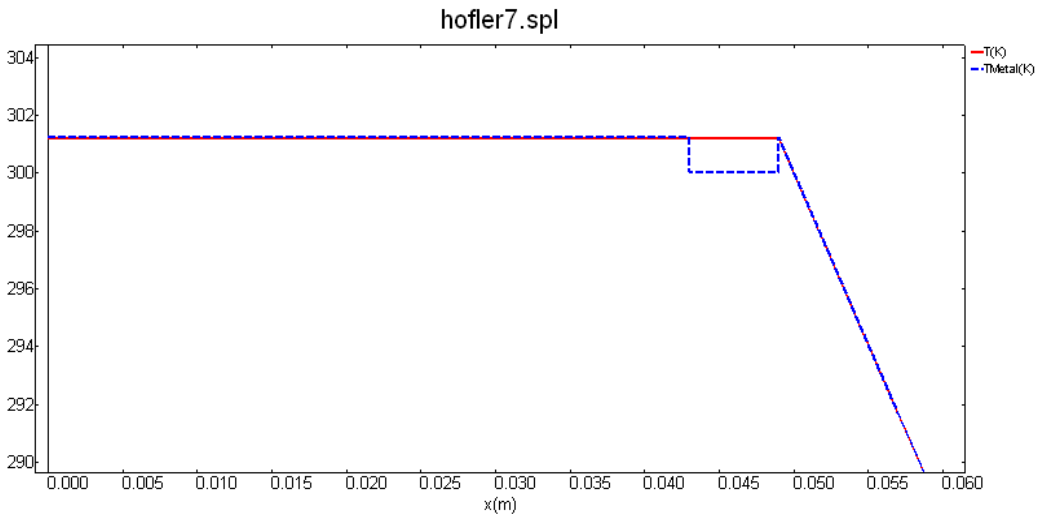


Figure 5.5: DELTAEC results for gas temperature  $T_m$  and solid temperature  $T_{\text{solid}}$  as functions of  $x$  in the uppermost part of Hofler’s refrigerator. These temperatures differ only in the heat exchanger.

for  $p_1$ ,  $U_1$ , and  $T_m$ , which can generally be regarded as

$$\frac{dp_1}{dx} = \mathcal{F}_{\text{momentum}}(\text{local and global parameters and variables}), \quad (5.10)$$

$$\frac{dU_1}{dx} = \mathcal{F}_{\text{continuity}}(\text{local and global parameters and variables}), \quad (5.11)$$

$$\frac{dT_m}{dx} = \mathcal{F}_{x\text{-energy}}(\text{local and global parameters and variables}), \quad (5.12)$$

$$\frac{d\dot{H}_{\text{tot}}}{dx} = \dot{q} \text{ in heat exchangers; } = 0 \text{ elsewhere.} \quad (5.13)$$

from the **BEGIN** segment to the end of the model. Some of the results of that integration are actually known *a priori* by the user, and some of the variables needed by DELTAEC to perform that integration are not known *a priori* by the user. These exceptional variables are specified by the user as targets and guesses, respectively, and DELTAEC uses a shooting-method algorithm, attempting to meet the targets by finding suitable values for the guesses. Initial values for guesses are usually left over from a previous run of DELTAEC, but sometimes they are provided by the user.

For the refrigerator example of the previous section, we chose targets and guesses by considering what integration results were known by us *a priori* and what input variables were not known by us *a priori*. We looked first near the end of the model for possible targets. We definitely needed the three **HARDEND** variables  $\text{Re}[1/z_n]$ ,  $\text{Im}[1/z_n]$ , and  $\dot{H}_{\text{tot}}$  in the target list, all targeted to zero, because there is no hole in the end of the apparatus that would allow nonzero  $U_1$  or  $\dot{H}_{\text{tot}}$  to escape—these are integration results that are known to us *a priori*. Experimentally, Hofler maintained the hot heat exchanger at 300 Kelvin; but DELTAEC computes that as a result of each integration pass, so it also had to be a target.

We looked first at the **BEGIN** segment for candidate guesses. The beginning gas temperature was not known *a priori* by us: DELTAEC had to adjust the beginning  $T_m$  to arrive at the first HX  $T_{\text{solid}}$  correctly, so **0c** had to be a guess. Next, we did not know the frequency needed to maintain resonance *a priori*, so it too had to be a guess. But how was resonance determined experimentally? By comparing the phases of  $p_1$  and  $U_1$  at the driver. These phases are computed by DELTAEC as results. Hence, we added their difference = 0 as a fifth target. While we were thinking about the driver, we realized that we wanted to imagine that  $|p_1|$  just below the driver was fixed, because Hofler adjusted his driver voltage to achieve a desired amplitude. Thus  $|p_1|$  at that location—a calculated result in DELTAEC—had to be a target, and the driver voltage had to be a guess. The heat rejected in the first heat exchanger had to be guessed because we didn't know it or control it experimentally, yet it is required by DELTAEC for each numerical integration. By now we had six targets and four guesses, so we needed two more guesses. One guess must be  $|p_1|$  at the beginning, which is in the cavity above the loudspeaker, where we had no *a priori* knowledge of pressure. One more guess was needed. The overall time phase of a DELTAEC model is meaningless; somewhere there is always one time-phase variable that can be chosen arbitrarily. However, here we still had two phases set arbitrarily: the phase of  $p_1$  at the **BEGIN** and the phase of  $V_1$  in the driver. One of these had to be used as a guess. We chose to use the phase of  $V_1$  as that guess.

We chose a point of view where we wanted to learn the cold solid temperature **9H** for a given cooling power **9e**. Had we wanted to have DELTAEC tell us the cooling power at a known cold temperature, we would have used an additional guess–target pair, targeting  $T_C$  and guessing  $\dot{Q}_C$ .

Although some situations that we model in DELTAEC are not this confusing, choosing the guesses and targets is never easy for a complicated thermoacoustic system. To choose them wisely, two things are essential: careful thought about the particular thermoacoustic system itself, and an appreciation of what variables are results of a single DELTAEC integration from **BEGIN** to **\*\*END** and what variables are needed by DELTAEC for each such integration. We introduce a few general guidelines for this thought process here.

To understand which variables are candidate targets, you must know which variables are *computed* as results by DELTAEC during each pass. To understand which variables are candidate guesses, you must know which variables are *needed* by DELTAEC for each pass of its integration. To aid this understanding, look at Eqs. (5.10)–(5.13), where computed results are the variables on the left-hand sides, and variables needed are on the right-hand sides. Print these equations (or the more complete versions at the beginning of Chapter 10) and post them near your workstation, so you can look at them for inspiration when you are trying to figure out guesses and targets for a new or modified model. Also look at the columnar structure of the `<.out>` file, where variables needed by DELTAEC for each pass of its integration are in the left column (as are potential targets) and variables that DELTAEC computes as results during each pass are in the right column. For a perfect understanding, there is no good substitute for studying the summaries of the computation algorithms for each segment, as discussed briefly in the tutorial chapters of the Users Guide, and more fully in Chapter 10.

To understand what variables should be regarded as known *a priori* by the user, it is



Table 5.1: Choosing guesses and targets.

	Variables <i>we</i> think of as known	Variables <i>we</i> think of as results
Variables that DELTAEC needs as inputs for <i>each pass</i> of integration. Includes $T_m$ -begin, $p_1$ -begin, $U_1$ -begin, $p_m$ , $f$ , all dimensions, 'ducer coefficients, heat at heat exchangers, gas-mixture mole fraction.	simply fixed	guess
Variables that DELTAEC calculates as results of <i>each pass</i> of integration. Includes all $T_m$ , $p_1$ , $U_1$ , $\dot{H}_{\text{tot}}$ except in <b>BEGIN</b> ; combinations of above such as $\dot{E}$ , RPN results, and $z_n$ at ends.	target	simply results in <.out> and <.sp> files

sometimes helpful to think about what variables are (or could be, in principle) experimentally controlled, in contrast to what variables are experimentally observed. These must be compared with the variables that DELTAEC computes as results during each integration pass and those that DELTAEC needs as inputs during each integration pass through the system. Note that our definition of an experimental result is more general than usual. In the Hofler refrigerator case, we considered the drive voltage an experimental result because it is determined experimentally by the condition that the pressure amplitude have the desired value. The viewpoint expressed in Table 5.2 is appropriate for comparison of DELTAEC and experimental data. In this case, geometrical parameters are simply fixed. Targets are experimentally fixed or controlled variables that are results of a single pass of numerical integration, chosen from among  $T_m$ ,  $p_1$ , and  $U_1$  (everywhere but in **BEGIN**); current magnitudes and phases in **VDUCERS** and voltage magnitudes and phases in **IDUCERS**; etc. Guesses may be unknown parameters chosen from among  $f$ , the magnitude and phase of  $U_1$ -**BEGIN** and  $p_1$ -**BEGIN**,  $T_m$ -**BEGIN**, heats at heat exchangers, the magnitude and phase of voltage at **VEDUCERS**, etc.

When designing hardware instead of analyzing it, a different viewpoint may be adopted. In this case, many geometrical parameters are not yet fixed, but desired operating temperatures, powers, frequency, etc. have been chosen. Often, several geometrical parameters are included as guesses, and more temperatures and other numerical results are included as targets. Hence, another useful way to think about guesses and targets is represented by Table 5.3.

As we did in our description of the choices of guesses and targets for the refrigerator example above, it is useful to look for targets at the end or middle of the model, then

Table 5.2: Interpreting input and result variables: experimental viewpoint.

	Experimentally controlled variable	Experimental result
Variable needed as input for <i>each pass</i> of DELTAEC's integration	simply fixed	guess
Variable computed as a result of <i>each pass</i> of DELTAEC's integration	target	simply a result in <code>&lt;.out&gt;</code> and <code>&lt;.sp&gt;</code> files

Table 5.3: Interpreting input and result variables: design viewpoint.

	Variable <i>we</i> want to think of as fixed	Variable <i>we</i> want to think of as a result
Variable needed as input for <i>each pass</i> of DELTAEC's integration	simply fixed	guess
Variable computed as a result of <i>each pass</i> of DELTAEC's integration	target	simply a result in <code>&lt;.out&gt;</code> and <code>&lt;.sp&gt;</code> files

look for guesses in the beginning or middle of the model. Sometimes it is easy to mentally associate each target with a corresponding guess, as we did for some of them in the discussion above. A few targets, such as the **HARDEND** targets in the example above, may be utterly obvious and necessary, depending only on the model but not on the user's point of view. However, many targets depend on the user's point of view. Does the user want to consider an electrically driven refrigerator to be driven at fixed frequency, or to be kept on resonance by active control? Is the user interested in cooling power at a given cold temperature, or in the cold temperature that is reached for a given cooling load? Thus, it is essential to think carefully about what we want to think of as "known" and what we want to think of as "unknown" for every model and for every use of that model.

It is also important to remember that the overall time phase of a DELTAEC model is arbitrary, to find the one and only one location in the model where that time phase is set (usually in the **BEGIN**, but sometimes in a transducer or an RPN target), and to make sure there are no other conflicting time-phase input variables (multiple side-branch transducers often being problematic).

## 5.4. Meeting targets in an elaborate model

### 5.4.1. Basic rules

Guesses must precede targets: At least one guess must precede the first target; at least two guesses must precede the first two targets; etc. DELTAEC warns if this rule is violated.

Avoid a **sameas** or RPN that points to a downstream result, if it affects an integration result that is compared with a target. This would confuse DELTAEC by making the targeted result depend on a result left over from the *previous* integration instead of only on the upstream results of the *present* integration. (When several **sameas**'s and RPNs point to one another, it can be difficult to see whether they are all independent of downstream results. DELTAEC should warn if this rule is violated, and the List Linkages feature described in Section 4.3 is also useful for diagnosis.) For example, this will not work, but changing the RPN formula from 3A to 1A will fix it:

```

!----- 0 -----
BEGIN      the setup
5.5400E+06 a Mean P Pa
  94.000   b Freq   Hz
  300.00   c TBeg   K
2.9938E+05 d |p|    Pa
-514.8    e Ph(p)  deg
1.5400E-03 f |U|    m^3/s   G
  125.00   g Ph(U)  deg
    1.0000 j nL
HeAr      Gas type
!----- 1 -----
DUCT      bypass tube
1.6417E-05 a Area   m^2           6.1161E+05 A |p|    Pa
1.4363E-02 b Perim  m             18.320   B Ph(p)  deg
  1.8290   c Length  m             1.2280E-03 C |U|    m^3/s
  etc. ...
!----- 2 -----
RPN p1 magnitude
6.1800E+05 a G or T           =2A      6.1161E+05           A Pa
3A
!----- 3 -----
COMPLIANCE next
8.3000E-04 a SurfAr m^2           6.1161E+05 A |p|    Pa

```

### 5.4.2. Developing a substantially new model

Occasionally the user can create a completely new model, establish several targets and guesses, hit the blue run button, and enjoy instant success. However, it is much more common to find that this approach yields no more than a message from DELTAEC explaining that it has failed to meet targets. Several strategies can be used to overcome this situation:

Understand approximately what to expect from your model before you even begin working with DELTAEC, so you can provide reasonable initial estimates for guesses. For example, use your knowledge of acoustics to make reasonable initial estimates of values for TBRANCH impedances to give DELTAEC a plausible starting point.

Proofread your model carefully. One of the most common problems in DELTAEC models is an order-of-magnitude typographical error in an input variable. Use the schematic view for a quick check of dimensions.

clear all guesses and targets, then run the model and examine the results for approximate sanity. (Better yet, do this sanity check before you establish any guesses and targets in the first place.) If you see a result that is wildly unrealistic, such as a temperature of 10,000 Kelvin, modify something upstream to fix it, so DELTAEC's shooting-method algorithm begins in the correct ballpark.

Even though you may have chosen guesses and targets by thinking about the end of the model first, introduce the guesses and targets starting near the beginning, one target and one guess at a time. To be sure that your first guess will be a suitable partner for your first target, think about the physics of cause and effect. Modify the candidate guess's value by hand before you establish it as a guess, running to confirm that changing the candidate guess has a significant effect on the intended target. Then add that guess and target, run, and confirm that the target is met. If successful, repeat this process for the next candidate guess and target.

Save your model frequently, so you can backtrack if things go awry. You can save a file with any filename extension, and we frequently find ourselves naming saved files to indicate the order they are created, e.g., <bottle01.out>, <bottle02.out>, <bottle03.out>, or to indicate what they represent to us, e.g., <bottle1targ.out>, <bottle2targs.out>, etc. We keep notes to show what each saved file represents.

As the number of guesses and targets increases, DELTAEC's shooting method has more and more work to do. Be kind. Before adding the Nth target and guess, modify the candidate guess by hand to make the intended target variable *very* close to the desired value before asking DELTAEC to take over responsibility for that target and guess.

If an overly bold change has gotten DELTAEC lost, use Restore all guessez, described in Section 5.7, to recover the previous set of guesses.

If DELTAEC seems to be missing targets by only a little, reduce the convergence tolerance or adjust one of the other shooting-method tuneup parameters, which are accessible through the Edit pulldown menu as Options, described in Section 8.3.2.

Consider whether the number of guesses and targets could be reduced by beginning at the **\*\*END** and ending at the **BEGINning**. If so, “flip” the model as described in Section 5.7.

Consider other ways to reduce the number of guesses and targets. For example use **VSPEAKER** instead of **VESPEAKER** if possible. Or avoid a transducer by using a nonzero  $U_1$  in **BEGIN**.

Change the guess-list membership, if you can think of an appropriate alternative to one or more existing guesses.

In a model with a **STK\*\*** segment, it is common for a heat upstream of that segment to be a guess, corresponding to a temperature target downstream of that segment. With regenerators, pulse tubes, and thermal buffer tubes, small changes in the guessed heat tend to cause large changes in the downstream temperature while the shooting method is exploring the landscape. If long **DUCTs** or other temperature-sensitive segments are downstream of the temperature-target segment, any targets downstream of the long segments can be wildly affected by the shooting method’s exploratory changes in the heat guess. Shooting-method stability can be improved in such cases by putting an **RPN** segment that fixes the temperature (e.g., “300 =Tm”) immediately upstream of the long segment(s).

If convergence seems strangely problematical, consider changing the relative importance of different targets. If some targets (such as heat-exchanger temperatures) are aimed at a nonzero number while others (such as **\*\*END** impedance) are aimed at zero, DELTAEC sometimes has trouble deciding how close is close enough. The relative weight can be changed via the Normalization Mode parameter described in Section 8.3.2, but it is sometimes more productive to change the weight of an individual target by incorporating it into an **RPN** segment that is aimed at zero and multiplying (or dividing) the calculated target result by a large number to emphasize (or de-emphasize) it; or adding 1 to what would otherwise be a zero target, to de-emphasize it.

If convergence seems strangely problematical, also consider whether some guesses might essentially have large fixed offsets. The shooting method begins by calculating the sensitivities of the results to changes in the guesses by changing each guess by a fixed percentage (described in Section 8.3.2). If a part-per-million change in one guess causes part-per-million changes in targeted results, but a part-per-million change in a second guess causes an order of magnitude change in one or more targeted results, the shooting method will fail. In such cases, it helps to use an **RPN** input parameter as small guess, add a large offset to it in the **RPN**’s line **b** to make a number that’s the size of the original guess, and **sameas** the original guess to the **RPN** result. This will ensure that ppm changes in all guesses cause ppm changes in all targeted results.

### 5.4.3. Modifying an existing model

Purely acoustic models with only a few guesses and targets, like the `bottle` in Chapter 2, are usually robust: From a state with all targets met, the user can often change any variable by a factor of two or so, and DELTAEC's shooting method will once again hit the targets. However, DELTAEC's shooting method gets lost more easily for thermoacoustic models with more than about five targets and five guesses, especially with `STKDUCT` and `STKCONE` segments. Some strategies for successful routine modifications of complex models, starting from a state in which targets are met, include:

If an overly bold change has gotten DELTAEC lost, use `Restore all guesses`, described in Section 5.7, to recover the previous set of guesses. Then use the incremental plot feature to change a variable by small steps. In practice, this usually means: Observe that a big change led to failure to converge; `Restore guesses` and un-do the big change manually to return to the secure starting condition; Try a smaller change (such as 1/10 of the original change); If the smaller change still leads to failure to converge, then restore and undo again, and try an even smaller change; When a small-enough change has been found, use that change as the step in the incremental plot feature to let DELTAEC accomplish the desired big change in 10, 100, or more small steps. (If DELTAEC gets part way through these small steps and then loses its ability to hit targets, use the plot rewind feature, described in Section 3.2.3, to recover a meaningful operating point that is at least part way toward the desired change.)

Insert a new segment gradually, by first inserting it with near-zero volume and near-zero impedance, then gradually increasing these with the incremental plot feature. Often this entails inserting a new segment with near-zero length and gradually increasing the length. Similarly, the heat of a new heat exchanger can be increased gradually from zero; the impedance of a new side-branch impedance can be reduced gradually from a very large value; the impedance of a series impedance can be increased gradually from zero.

Kill an undesired segment gradually by reducing its length with the incremental plot feature before Killing it.

Save your model frequently, so you can recover from the unexpected.

The shooting method within DELTAEC can sometimes get stuck around an inappropriate local minimum. Try manually changing one of the guesses slightly and see if DELTAEC will loose its fondness for this particular sticky point.

Always check results carefully for reasonableness, particularly when modeling complicated systems or using any of DELTAEC's more elaborate features. While DELTAEC is a useful tool, it is far from foolproof—for example, the shooting method can easily end up generating devices that are several wavelengths long or wander into the completely unrealistic realm of negative values for guessed frequency or length.

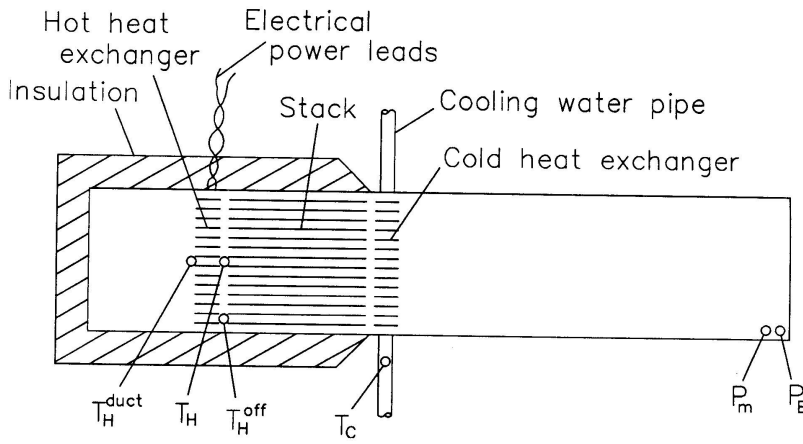


Figure 5.6: Five-inch-diameter standing-wave engine.

## 5.5. Standing-wave engine

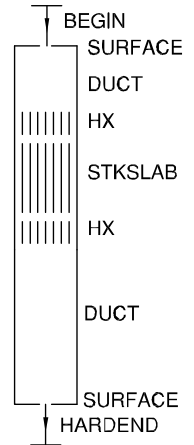
This standing-wave-engine example is described in detail in Ref. [23]. The device described there is used here to illustrate the thermoacoustic capabilities of DELTAEC in the context of an engine. We will reproduce some of the figures in that paper here.

The apparatus is shown in Fig. 5.6. After running the file <5inch.out>, the result is:

```

TITLE      Five-Inch Standing-wave Thermoacoustic Engine
!Created@16:56:43 5-Oct-06 with DeltaE Vers. 6.0g2 for the IBM/PC-Compatible
-----0-----
BEGIN      Initial
1.3800E+06 a Mean P      Pa
121.16    b Freq        Hz      G
557.92    c TBeg        K        G
7.3549E+04 d |p|        Pa      G
0.0000    e Ph(p)       deg
0.0000    f |U|        m^3/s
0.0000    g Ph(U)       deg
helium    Gas type
-----1-----
SURFACE    Hot End
1.2920E-02 a Area      m^2
7.3549E+04 A |p|        Pa
0.0000    B Ph(p)       deg
3.3316E-05 C |U|        m^3/s
180.00    D Ph(U)       deg
0.0000    E Htot        W
-1.2252   F Edot        W
ideal      Solid type
-----2-----
DUCT       Hot Duct
sameas    1a a Area      m^2
0.4030    b Perim      m
0.2790    c Length     m
7.2686E+04 A |p|        Pa
5.5059E-03 B Ph(p)       deg
8.7694E-02 C |U|        m^3/s
-90.209   D Ph(U)       deg
0.0000    E Htot        W
-11.907   F Edot        W
ideal      Solid type
-----3-----
HX         Hot HX
sameas    1a a Area      m^2
0.3930    b GasA/A
6.0000E-02 c Length     m
4.8300E-04 d y0        m
2210.0    e HeatIn      W
999.00    f Est-T       K      (t)
7.1519E+04 A |p|        Pa
0.3876    B Ph(p)       deg
9.6763E-02 C |U|        m^3/s
-91.200   D Ph(U)       deg
2210.0    E Htot        W
-95.846   F Edot        W
557.92    G GasT        K
563.81    H SolidT      K
ideal      Solid type
-----4-----
STKCIRC    Honey Stack
sameas    1a a Area      m^2
0.8100    b GasA/A
0.2790    c Length     m
6.5700E+04 A |p|        Pa
2.7530    B Ph(p)       deg
0.1599    C |U|        m^3/s

```



```

5.0000E-04 d radius      m      -85.386  D Ph(U)    deg
5.0000E-05 e Lplate     m      2210.0   E Htot     W
                    170.63  F Edot     W
                    557.92  G TBeg     K
                    306.39  H TEnd     K
stainless Solid type
!----- 5 -----
HX Ambient HX
1.2670E-02 a Area      m^2    6.3081E+04 A |p|      Pa
0.4860 b GasA/A      m      3.2510 B Ph(p)    deg
5.0800E-02 c Length   m      0.1676 C |U|      m^3/s
4.0600E-04 d y0      m      -85.829 D Ph(U)    deg
-2125.1 e HeatIn     W      84.898 E Htot     W
303.00 f Est-T       K      = 5H?    84.898 F Edot     W
                    306.39 G GasT     K
ideal Solid type      303.00 H SolidT   K
!----- 6 -----
ANCHOR subsequent duct is water jacketed. dHdot/dx = dEdot/dx
!----- 7 -----
DUCT Ambient Duct
sameas 5a a Area      m^2    6.9652E+04 A |p|      Pa
0.3990 b Perim      m      -176.59 B Ph(p)    deg
3.6500 c Length     m      1.8500E-05 C |U|      m^3/s
                    -176.59 D Ph(U)    deg
                    0.6443 E Htot     W
ideal Solid type      0.6443 F Edot     W
!----- 8 -----
SURFACE Ambient End
sameas 5a a Area      m^2    6.9652E+04 A |p|      Pa
                    -176.59 B Ph(p)    deg
                    1.7385E-16 C |U|      m^3/s
                    95.222 D Ph(U)    deg
ideal Solid type      1.9100E-13 E Htot     W
                    1.9100E-13 F Edot     W
!----- 9 -----
HARDEND target this to seal the end
0.0000 a R(1/z)      = 9G?    6.9652E+04 A |p|      Pa
0.0000 b I(1/z)      = 9H?    -176.59 B Ph(p)    deg
                    1.7385E-16 C |U|      m^3/s
                    95.222 D Ph(U)    deg
                    1.9100E-13 E Htot     W
                    1.9100E-13 F Edot     W
                    1.3878E-17 G R(1/z)
                    -4.3970E-16 H I(1/z)
!----- 10 -----
RPN ensure all ambient-duct dissipation goes to water jacket
0.0000 a G or T      = 10A?   -5.6843E-14 A zero
5E 5F -

```

Examining the guess–target summary, we find

```

GUESS      0b      0c      0d      5e
name BEGIN:Freq BEGIN:TBeg BEGIN: |p| HX:HeatI
units Hz K Pa W
value 121.16 557.92 7.3549E+04 -2125.1
TARGET
name HX:Est-T HARDE:R(1/z) HARDE:I(1/z) RPN:Targe
units K
value 303.00 0.0000 0.0000 0.0000
result 303.00 1.3878E-17 -4.3970E-16 -5.6843E-14

```

and some `<.sp>` graphs show the wave, power, and temperature profiles in Fig. 5.7.

Targets **9a** and **9b**, the real and imaginary parts of  $1/z_n$ , are of course necessary to impose  $U_1 = 0$  at the closed ambient end of the resonator. Our choices of other targets and guesses were made to reflect the perspective of Ref. [23], in which the as-built hardware was driven by a fixed heat input (ordinary input **3e**) and the ambient heat exchanger was held at fixed temperature (target **5f**). Frequency **0b** is a guess, because we want DELTAEC to tell us the frequency at which it thinks the system should spontaneously oscillate. Initial temperature **0c** and initial pressure amplitude **0d** are also guesses, because we want DELTAEC to tell us the hot temperature and the pressure amplitude at which it thinks the system should oscillate for the given hot heat input. The final target and guess, **10a** and **5e**, allow DELTAEC to tell us the heat rejected at the ambient temperature, under the constraint that  $\dot{E} = \dot{H}_{\text{tot}}$  at the interface between the ambient heat exchanger and the long ambient resonator duct. We judge this to be a reasonable energy-closure condition for the model because the resonator duct was



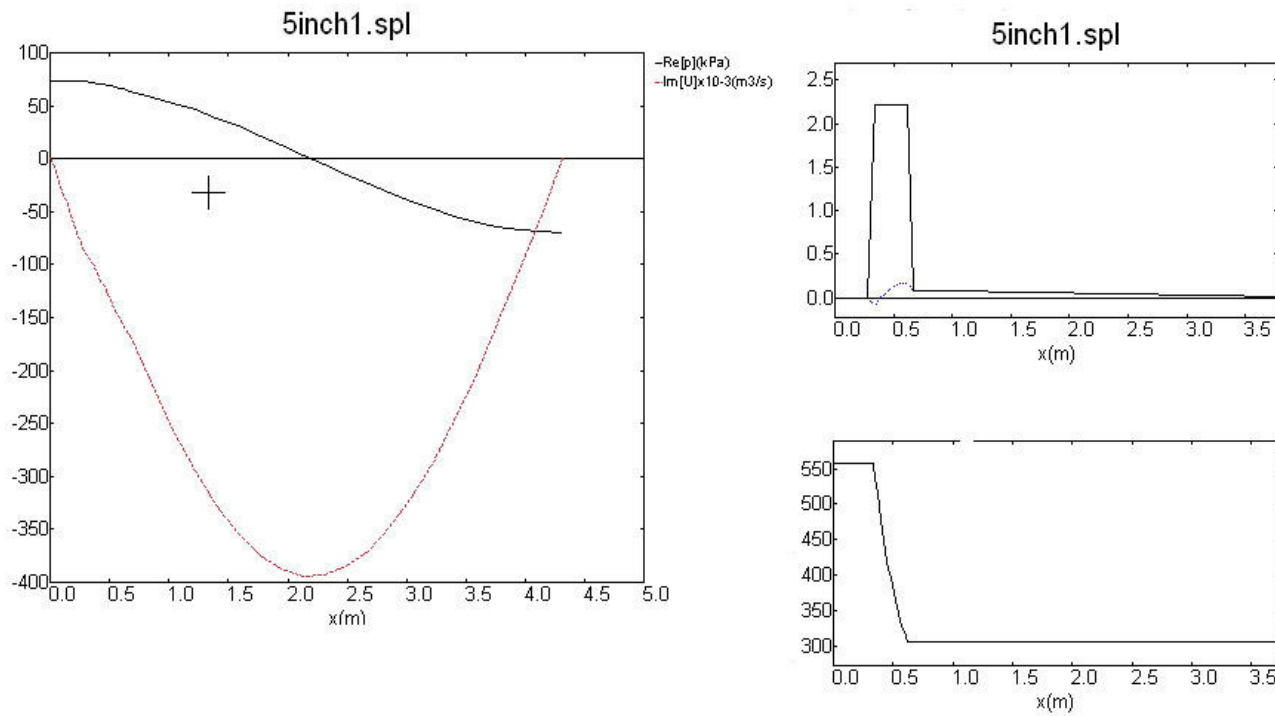


Figure 5.7: Graphs of the wave (left), total power and acoustic power (top right), and temperature (bottom right) as functions of  $x$  in `<5inch.out>`. Changing temperature and constant  $\dot{H}_{\text{tot}}$  show the location of the stack. The heat exchanger locations are indicated by the step changes in  $\dot{H}_{\text{tot}}$ .

water-cooled in this apparatus, so the heat arising from  $\dot{E}$  dissipated in it might be expected to be carried away locally by water. (This is of course not strictly true; boundary-layer thermoacoustic processes redistribute energy along  $x$ , as was dramatically demonstrated in Ref. [24] long ago.) Consistent with this energy-closure condition, the ANCHOR segment just before the final DUCT overrides the default behavior of the subsequent segments (thermally insulated, except in heat exchangers), ensuring that  $\dot{H}_{2,k} = \dot{E}$  thereafter.

Other choices could be made for the guesses and targets. For instance, the cold-duct length could be substituted for the frequency as one of the guesses, if we were designing an apparatus to operate at a desired frequency. The hot-heat-exchanger temperature **3f** could be added to the target list and, simultaneously, the hot-heat-exchanger heat **3e** could be added to the guess list, if we were designing an apparatus to operate at a desired hot temperature. For now, however, the guesses and targets will be left as they are, since they reflect the point of view adopted in Ref. [23].

To plot some results for this 5-inch engine case, we started DELTAEC with this file again and prepared parameters so the `plot` summary became

```
Dependent Variables (outputs):
PLDTS      OA      OB      OC      OD      3H      8A
name BEGIN:Freq BEGIN:TBeg BEGIN:|p| BEGIN:HeatI HX:Solid HARDE:|p|
units      Hz      K      Pa      W      K      Pa
Independent Variables (inputs):
Outer loop: 3e HX:HeatI Beg= 9.50E+02 End= 50. Step= -33.
```

by adding **3H** and **8A** to the dependent-plot-variable list and establishing **3e** as an independent plot variable, setting its initial, final, and step values. (Some other variables such as **TBeg** are of minor interest now, but cannot be deleted from the list of plot variables because guesses appear here by default.)

Next, we modified mean pressure to be 19.2 bar, and ran the code. When completed, we modified mean pressure to 13.8 bar, and ran it again, appending the new results to the `<.ip>` file. Three more runs with mean pressures of 9.6, 6.9, and 5.2 bar completed the data set. We exited from DELTAEC, and read the `<.ip>` file into a spreadsheet/graphics program for a little massaging: converting pressure amplitude at the cold end from Pascals to bar, and then squaring that number; subtracting 303 Kelvin from  $T_H$ , and adding the measured heat leak to the room to  $\dot{Q}_H$ . Plotting these results then yields the curves shown in Fig. 5.8, resembling Figs. 5, 6, and 7 in Ref. [23]. These curves differ slightly from those in the reference, because of DELTAEC's inclusion of the small gas-to-solid temperature differences in the heat exchangers.

More detailed comparison between DELTAEC computations and measurements with this apparatus can be found in Ref. [25]. The agreement is excellent at low amplitude, but disagreement grows large at high amplitude—typical of our experience with standing-wave engines up to  $|p_1|/p_m \simeq 0.1$ . See Section 9.2.5 for details about such disagreement.

## 5.6. More thermoacoustic segments

In this section we list thermal segments that are commonly used for engines and refrigerators: the several types of heat exchangers, stacks, and regenerators. More details on each can be found in Chapter 10 and Chapter 11.

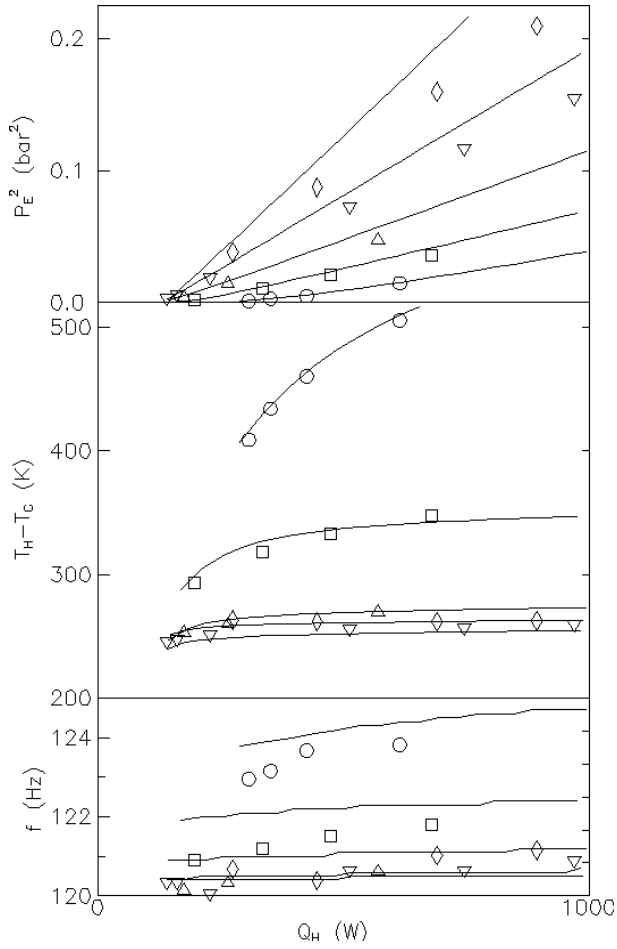


Figure 5.8: Standing-wave engine results. Lines are DELTAEC results; points are from experimental data in Ref. [23].

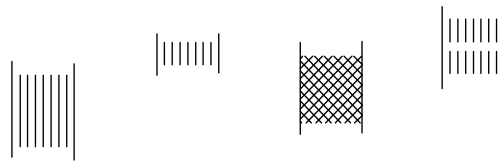


Figure 5.9: More thermoacoustic segments: stacks, regenerators, and heat exchangers.

**STKCIRC** A stack or regenerator with circular pores. We also use this to model hexagonal honeycomb stacks.

**STKSLAB** A stack or regenerator with parallel-plate geometry.

**STKRECT** A stack or regenerator with rectangular (or square) pore geometry.

**STKPIN** A stack or regenerator comprised of an array of pins aligned parallel to  $x$ .

**STKDUCT** A “stack” with lateral dimensions much larger than  $\delta_\kappa$ , computed in boundary-layer approximation. (Used for pulse tubes and thermal buffer tubes.)

**STKCONE** A tapered “stack” with lateral dimensions much larger than  $\delta_\kappa$ , computed in boundary-layer approximation. (Used for pulse tubes and thermal buffer tubes.)

**STKSCREEN** A screen regenerator for Stirling systems.

**STKPOWERLW** A porous medium with friction factor and heat-transfer coefficient specified as power-law functions of Reynolds number.

**HX** A parallel-plate heat exchanger.

**TX** Tube-array heat exchanger, with the thermoacoustic working gas inside the tubes.

**SX** Stacked-screen heat exchangers, valid only for  $\delta_\kappa$  greater than hydraulic radius.

**PX** Power-law heat exchangers, with friction factor and heat-transfer coefficient characterized by power laws in Reynolds number.

**VXT1 and VXT2** Heat exchangers with fixed solid temperature, in which gas temperature can vary with  $x$ .

**VXQ1 and VXQ2** Heat exchangers with fixed heat per unit length, in which gas temperature can vary with  $x$ .

**ANCHOR** This segment overrides what would otherwise be default thermal-insulation mode in subsequent **DUCTs**, **CONES**, **COMPLIANCEs**, **IMPEDANCEs**, **SURFACEs**, etc., so **DELTAEC** treats those segments as if they are immersed in a thermal bath at  $T = T_m$ .

**INSULATE** This segment is used to re-establish the default thermal-insulation condition if it was previously overridden by **ANCHOR**.

## 5.7. More user-interface features

These options offer substantial power and convenience for experienced users.

**Restore all guesses.** Before beginning iterations during a run, DELTAEC saves copies of the guess values. It also saves copies of a few output parameters if it caught the user inadvertently or boldly using a `sameas` or RPN pointer to an output parameter in a segment downstream of that `sameas` or RPN, despite DELTAEC’s “Forward Reference Condition” warning about the unstable shooting-method behavior that can result from using such downstream results. (See Section 5.4 for an explanation of this danger.) Whenever an unsuccessful run overwrites the guesses (leaving you and DELTAEC lost), you can use this option to restore all of these variables to their earlier values. Simply Restore guesses, then modify one or more variables yourself (e.g., returning anything you recently changed to earlier, more successful values), and try again.

**Comment lines.** If the heading on the first line of a segment does not provide enough space for the user’s notes, additional comment lines can be added below it. Examples are shown at the beginning of the next section, Section 5.8.1.

**Split segment.** This option, available by right clicking on the top line of a segment or in the Edit pulldown menu, automates the process of splitting any segment with length (e.g., a DUCT, CONE, or STK\*\*) into two segments, each having half of the original length, correcting the `sameas` and RPN references everywhere in the model, and correcting the guess, target, and plot lists. (To partition the lengths unequally while keeping the correct total length, you can use master–slave linking (Section 2.7.2) after splitting a segment in equal halves: slave the first length to the second, then modify the first length, then clear the length link to allow the two pieces to be modified independently.)

**AutoLayout.** Sometimes auxiliary windows such as plots and the highlight display become lost off the edge of the computer’s screen. (For example, undocking a laptop from a larger screen can cause this.) The AutoLayout option, available in the Display pulldown menu, brings all of DELTAEC’s auxiliary displays close to DELTAEC itself. (If this fails for a plot, close it, delete the associated `<.ssv>` or `<.isv>` from your model’s folder, and try again.)

**Flip a block of segments.** (Implemented in version 6.4.) The number of guesses and targets can sometimes be reduced by beginning the integration of a model from what you had previously considered the “end.” Orifice pulse-tube refrigerators (described in Section 5.8) are a particularly good example because we naturally think of them as beginning at the driver, but they “end” with an accurately known impedance in the COMPLIANCE tank, while the driver impedance is usually less well known; BEGINning in the tank can often reduce the number of guesses and targets. Similarly, shooting-method stability can sometimes be improved by putting regenerators upstream of thermal buffer tubes, instead of the other way around. The “Flip Selected Segs” operation automates switching back and forth between these two directions of integration, simplifying an otherwise very tedious and error-prone process. Highlight a block of segments, such as an entire model or the segments between a TBRANCH and its SOFTEND, and right-click to access the “Flip Selected Segs” feature. Flipping reverses the order of the physical segments in the block, and interchanges parameters such as AreaI and AreaF within unsymmetrical segments such as CONES. DELTAEC tries to relocate any

RPN segments intelligently in the block, but sometimes gives up and puts them at the end of the block. “Flip Selected Segs” takes care of `sameas` pointers, RPN-segment pointers, and plot references, and preserves the guess and target lists. Always proof-read very carefully after flipping; in complicated models there is always something that DELTAEC handles incorrectly, and intelligent human intervention will be needed to fix things like TBRANCH impedances feeding into flipped segment blocks and targets getting moved upstream of the corresponding guesses.

## 5.8. Stirling and pulse-tube refrigerators

Rott’s equations implemented in DELTAEC are valid for any phase difference between  $p_1$  and  $U_1$ , and any degree of thermal contact in the “stack.” Hence, DELTAEC can be used to model Stirling thermodynamic systems, in which  $p_1$  and  $U_1$  are substantially in phase, as well as standing-wave thermoacoustic devices, in which the phases  $p_1$  and  $U_1$  differ by nearly  $90^\circ$  as described in Sections 5.2 and 5.5. The principal additional DELTAEC segment that is often needed for Stirling systems is one for stacked screen beds, STKSCREEN, because stacked-screen regenerators are more common than parallel-plate, circular-pore, or rectangular-pore regenerators in Stirling systems. In our opinion, the principal shortcomings of DELTAEC for Stirling applications are DELTAEC’s acoustic approximation (which leads to reduced accuracy at high pressure amplitudes) and its inability to predict two-dimensional effects and streaming-driven convective heat transport in pulse tubes (shortcomings shared by many other design programs). DELTAEC’s main virtues are speed and easy inclusion of some auxiliary components such as ducts, dead volumes, and linear motors in the model.

Harmonic analysis of Stirling systems is introduced by I. Urieli and D. M. Berchowitz in Ref. [26] and by A. J. Organ in Ref. [27].

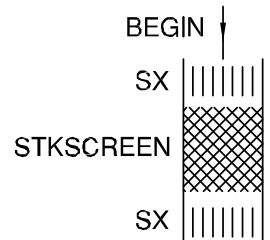
### 5.8.1. Alpha Stirling cryocooler

The example file <Stirling.out> represents a simple 55-Hz, 2-MPa helium Stirling cryocooler with stacked-screen regenerator and heat exchangers. This apparatus is illustrated in Fig. 5.10. (It does not resemble any real cryocooler known to us; we created it only to illustrate DELTAEC’s capabilities.) First, we examine <Stirling.out>:

```

TITLE      Bare bones Stirling cryocooler
! The user can type a few lines of notes here, describing the model.
!----- 0 -----
BEGIN      Initialize things.
! I will use values that seem reasonable, but are not drawn from
! anything specific in my experience.
2.0000E+06 a Mean P      Pa
55.000    b Freq        Hz
300.14    c TBeg        K          G
2.8459E+05 d |p|        Pa          G
-43.119   e Ph(p)       deg         G
3.6500E-04 f |U|        m^3/s
0.0000    g Ph(U)       deg
helium    Gas type
!----- 1 -----
SX        aftercooler
! SX should be ok because this is low enough power that copper screens
! will be work ok.
sameas   2a a Area      m^2          2.8085E+05 A |p|      Pa
0.6000   b VolPor      m            -43.835   B Ph(p)    deg
1.0000E-03 c Length    m            3.6265E-04 C |U|      m^3/s
sameas   2d d rh        m            -0.3900  D Ph(U)    deg

```



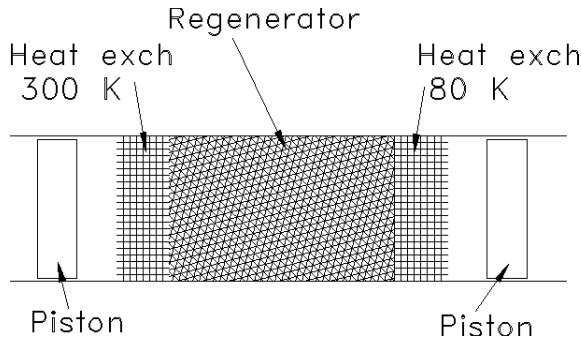


Figure 5.10: A Stirling cryocooler.

-35.833	e HeatIn	W	G	2.0781	E Htot	W
300.00	f Est-T	K	= 1H?	36.973	F Edot	W
				300.14	G GasT	K
				300.00	H SolidT	K
----- 2 -----						
copper	Solid type					
!-----						
STKSCREEN	regenerator					
1.1670E-04	a Area	m^2		2.2845E+05	A  p	Pa
0.6860	b VolPor			-53.179	B Ph(p)	deg
5.0000E-02	c Length	m		6.2225E-05	C  U	m^3/s
1.3900E-05	d rh	m		-49.487	D Ph(U)	deg
0.3000	e ksFrac			2.0781	E Htot	W
				7.0929	F Edot	W
				300.14	G TBeg	K
				79.960	H TEnd	K
----- 3 -----						
SX	cold heat exch					
! See the file Stirling.xls in this folder for an estimate of						
! the temperature difference from the edge to the center of this						
! heat exchanger, due to finite conductivity of copper.						
sameas	2a a Area	m^2		2.2804E+05	A  p	Pa
0.6000	b VolPor			-53.184	B Ph(p)	deg
1.0000E-03	c Length	m		6.2000E-05	C  U	m^3/s
sameas	2d d rh	m		-52.000	D Ph(U)	deg
4.9896	e HeatIn	W	G	7.0677	E Htot	W
80.000	f Est-T	K	= 3H?	7.0677	F Edot	W
				79.960	G GasT	K
				80.000	H SolidT	K
----- 4 -----						
RPN	enforce Htot = Edot here					
0.0000	a G or T		= 4A?	-4.5359E-12	A	
	Htot Edot	-				
!-----						
RPN	U sub 1 at cold end					
6.2000E-05	a G or T		= 5A?	6.2000E-05	A  U1	
	3C					
!-----						
RPN	phase(U) at cold end					
-52.000	a G or T		= 6A?	-52.000	A ph(U1)	
	3D					

The physical segments consist of a first heat exchanger, at 300 K, the regenerator, and a second heat exchanger at 80 K. All three are stacked screens. The other segments—BEGIN and the RPNs—define the end conditions.

The hydraulic radius  $r_h$  and volumetric porosity  $\phi$  for the screens can be calculated from the screen manufacturer's specs, using expressions from Ref. [27]:

$$\phi \simeq 1 - \frac{\pi m d}{4}, \quad (5.14)$$

$$r_h = \frac{d}{4} \frac{\phi}{1 - \phi}, \quad (5.15)$$

where  $d$  is wire diameter and  $m$  is mesh number (i.e., number of wires per unit length). The second equation results from the conventional definition of hydraulic radius: Gas volume

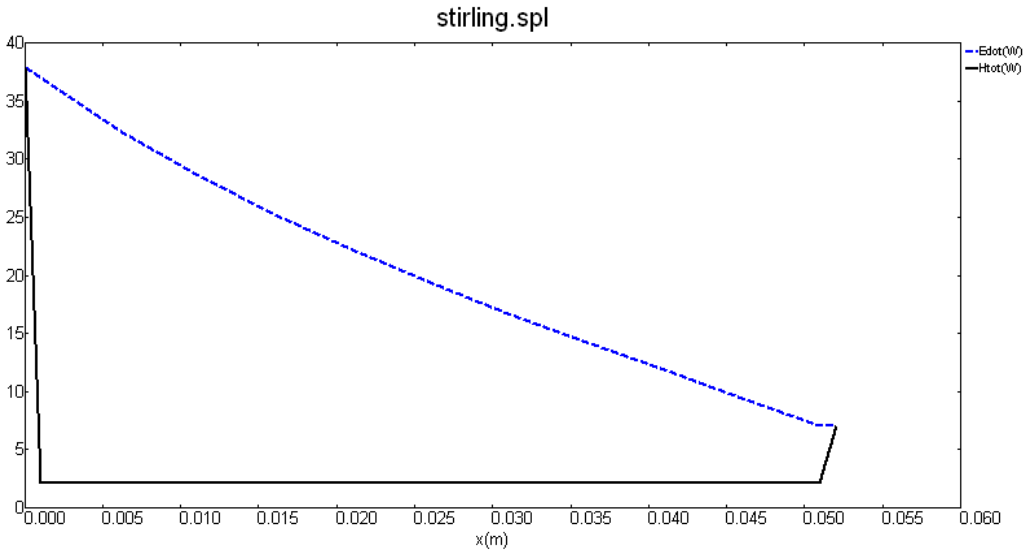


Figure 5.11: Total power (solid black) and acoustic power (dashed blue) vs position in the Stirling cryocooler example. The total power  $\dot{H}_{\text{tot}}$  is constant and small in the regenerator, and changes in the heat exchangers.

divided by gas–solid interface area. This regenerator is a little over 1 cm in diameter and is 5 cm long. The heat exchangers are the same diameter but only 1 mm long.

DELTAEC estimates the temperature difference between the helium gas and the copper screen wires in the heat exchangers, but it has no provision for estimating the temperature difference between the screen wires and the “housing” in which they are mounted (a temperature difference caused by the finite thermal conductance of the screen wires themselves). This is not a serious concern for small machines, but should be checked by hand on a case-by-case basis.

Line **e** in the regenerator segment, “**ksFrac**,” is the fudge factor by which longitudinal conduction through the regenerator is adjusted due to the spatially intermittent thermal contact between adjacent screens and due to the conduction of the pressure-vessel wall. Following Ref. [28], we often set **ksFrac** somewhere between 0.1 and 0.3, the lower end of the range being appropriate if the conduction of the pressure-vessel wall is negligible.

Our point of view with respect to boundary conditions in this example is most easily displayed by running DELTAEC on this file and examining the guess–target summary,

GUESS	0c	0d	0e	1e	3e
name	BEGIN:TBeg	BEGIN: p	BEGIN:Ph(p)	SX:HeatI	SX:HeatI
units	K	Pa	deg	W	W
value	300.14	2.8459E+05	-43.119	-35.833	4.9896
TARGET	1f	3f	4a	5a	6a
name	SX:Est-T	SX:Est-T	RPN:Targe	RPN:Targe	RPN:Targe
units	K	K			
value	300.00	80.000	0.0000	6.2000E-05	-52.000
result	300.00	80.000	-2.2675E-12	6.2000E-05	-52.000

and the **BEGIN** segment. We are considering the complex volume flow rates at the two ends to be given, as if we have in mind an “alpha” Stirling machine, with the motions of the two pistons determining the volumes of the compression and expansion spaces. The volume flow rate at the hot end is set by lines **0f** and **0g** in the **BEGIN** segment. The  $0^\circ$  phase of



line **0g** essentially determines the zero of phase for the entire system. The volume flow rate  $3.65 \times 10^{-4} \text{ m}^3/\text{s}$  of line **0f** (together with the frequency set in line **0b**) implies a volumetric stroke of  $2.1 \text{ cm}^3$  peak-to-peak at the hot end. The **RPNs** at the cold end ensure that DELTAEC’s shooting method arrives there with the desired cold piston stroke and phase. To arrive at these two targets, DELTAEC adjusts two guesses: the pressure amplitude and phase in the **BEGIN** segment (and hence throughout the cooler). We also insist that the solid temperatures in the two heat exchangers be 300 K and 80 K; DELTAEC achieves these two targets by adjusting two more guesses: the heat extracted at the hot heat exchanger, and the temperature in the **BEGIN** segment. We also insist that the acoustic power and total power leaving the cold heat exchanger are equal, implying that the cold piston is thermally insulated. (Acoustic power and total power are also equal at the ambient piston, because the **BEGIN** segment assigns  $\dot{H}_{2,k} = \dot{E}$  by default.)

DELTAEC predicts that, under these circumstances, the cooler will reject 36 W at the hot heat exchanger and will have a cooling power of 5 W. This cooling power accounts for heat conduction and enthalpy flow through the regenerator, but does not account for any heat load imposed by frictional irreversibilities in the cold piston, nor any heat load imposed by the regenerator *case* conduction except to the extent that it is included in **ksFrac**.

We now make or suggest a few simple modifications to this file to illustrate additional features of DELTAEC.

To discover what temperature the cooler would maintain with a heat load of 10 W instead of 5 W, we remove **3f**—the cold heat exchanger temperature—from the target list, and we remove **3e**—the cooling power—from the guess list, and modify it to 10 W. Running DELTAEC shows that the shooting method is lost with such a dramatic change, so we Restore **guessez**, modify **3e** back to 5 W, and use an incremental plot of **3e** from 5 W to 10 W, step 0.2 W, to reach the desired heat load. Under these circumstances the cold temperature will be 232 K.

Using **3e** as an independent plot variable running from 10 W to 2 W with steps of, say, 0.2 W, and using **3H** (cold solid temperature) as a dependent plot variable will generate a table and plot of cold temperature (and other defaults) vs heat load.

Insertion of two **\*ESPEAKER** segments, before the aftercooler and after the cold heat exchanger, would model use of linear motors and pistons there.

In the next subsection, we will use **TBRANCH** and **UNION** to change this model from an “alpha” Stirling machine to a “beta” or “gamma” machine, with one power piston on the hot end and a displacer piston in parallel with the heat exchange elements.

### 5.8.2. Beta or gamma cryocooler

The use of **TBRANCH** and **UNION** for hardware with branches and loops was introduced in Section 2.7.5. Recall the method of computation: At a **TBRANCH**, the branch impedance determines how the (complex) volume flow rate splits up; volume flow rates are added at the **UNION**. In purely acoustic models, the complex branch impedance is most often used as a pair of guesses that DELTAEC adjusts in its usual way to get the complex pressure at the **SOFTEND** of the **TBRANCH** to come out to be the same as it is at the **UNION**. In models with thermoacoustic components, a third guess–target pair is also employed: The fraction

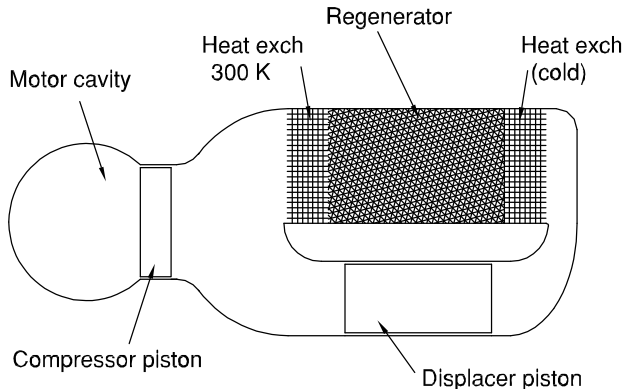


Figure 5.12: “Gamma”-style Stirling machine.

of the energy flow  $\dot{H}_{\text{tot}}$  going down the TBRANCH is guessed, and the temperature at the SOFTEND of the TBRANCH is targeted to equal the temperature at the UNION. Models with TBRANCHs and UNIONS tend to have many guesses and targets, since every SOFTEND and UNION contributes two or three targets (and a few more targets are almost always needed for temperatures, heats, etc.). Keep in mind that UNION targets differ from the usual RPN, \*\*END and heat-exchanger-temperature targets because their actual values are not specified by the user—they vary from one integration to the next, depending on what is happening at the associated SOFTEND segment, as the shooting method varies the guesses.

As an example of use of TBRANCH and UNION in a thermoacoustic setting, we return to the Stirling cryocooler example above, and convert it to a “displacer”-style [26] Stirling machine, with a compressor piston at the hot end and a driven displacer piston connecting the hot and cold ends. In the previous example,  $PU$  power (i.e.,  $\dot{E}$  in DELTAEC vocabulary) flowed in at the BEGIN and out at the end; no explicit segments indicated the pistons there. Here, with a displacer piston, the cold-end  $PU$  power is returned automatically to the hot end, reducing the hot-end  $PU$  power requirement.

The apparatus layout is illustrated in Fig. 5.12. The corresponding display of segment headings is

```

0 BEGIN:G P Initialize things
1 COMPLIANCE: space around power-piston motor
2 IESPEAKER: the power piston
3 COMPLIANCE: a little space in front of the power piston
4 TBRANCH:G branch to displacer
5 IESPEAKER: a spring-mounted, driven moving mass
6 RPN:G reassign Tm (Displacer is "axially insulating.")
7 COMPLIANCE: a little volume for the connection
8 SOFTEND: reconnect at UNION
9 SX :GT aftercooler
10 STKSCREEN: regenerator
11 SX :GT cold heat exch
12 UNION: T displacer cold end
13 HARDE: T close the end
14 RPN : T simulate insulation in the displacer piston

```

and the guess–target summary is the largest we have yet encountered in the Users Guide:

GUESS	0c	0d	0e	4a	4b	4c	6a	9e	11e
name	BEGIN:TBeg	BEGIN: p	BEGIN:Ph(p)	TBRAN:Re(Zb)	TBRAN:Im(Zb)	TBRAN:HtotB	RPN:Guess	SX:HeatI	SX:HeatI
units	K	Pa	deg	Pa-s/m <sup>3</sup>	Pa-s/m <sup>3</sup>	W		W	W
value	300.22	6373.3	114.43	-3.0131E+09	-3.2836E+09	-6.7651	79.962	-45.267	4.5794

TARGET	9f	11f	12b	12c	12d	13a	13b	13c	14a
name	SX:Est-T	SX:Est-T	UNION: p SO	UNION:Ph(p)	UNION:Tm	HARDE:R(1/z	HARDE:I(1/z	HARDE:Htot	RPN:Target
units	K	K	Pa	deg	K			W	
value	300.00	80.000	2.5375E+05	-44.127	79.962	0.0000	0.0000	0.0000	0.0000
result	300.00	80.000	2.5375E+05	-44.127	79.962	8.9507E-14	2.8896E-13	-2.499E-11	-4.838E-11

One can think of these guesses and targets as paired up in the following way. The TBeg guess lets DELTAEC hit the T-ambient target. (These two are so nearly equal, and so trivially related, that they could easily be omitted.) The two branch-impedance guesses let DELTAEC reach the two  $p_1$  targets at the UNION. The branch-Htot guess lets DELTAEC hit the RPN target at segment 13, which enforces thermal insulation on the cold end of the displacer piston. The temperature-reassignment guess, which lets us imagine the displacer piston to be axially insulating, trivially targets the UNION temperature. The heat removed at the hot heat exchanger lets DELTAEC reach the target temperature at the cold heat exchanger. Finally, the two  $p_1$  guesses in the BEGIN segment and the heat in the cold heat exchanger allow DELTAEC to achieve  $U_1 = 0$  and  $\dot{H}_{tot} = 0$  at the HARDEND at the end of the apparatus.

The corresponding <.out> file is

TITLE Stirling cooler w displacer piston, illustrating TBRANCH--UNION

!->C:\Users\092710\Documents\DeltaEC\PySource\Examples\EnginesAndRefr\gammaamma2.out

!Created@11:48:32 21-Mar-2016 with DeltaEC version 6.4b2.1!under win32,

using Win 6.1.7601 (Service Pack 1) under Python 2.7.9.

!----- 0 -----

BEGIN Initialize things

```

2.0000E+6 a Mean P Pa
55.000 b Freq Hz
300.22 c TBeg K G
6373.3 d |p| Pa G
114.43 e Ph(p) deg G
0.0000 f |U| m^3/s
0.0000 g Ph(U) deg
helium Gas type
!----- 1 -----

```

COMPLIANCE space around power-piston motor

```

0.1000 a SurfAr m^2 6373.3 A |p| Pa
5.0000E-4 b Volume m^3 114.43 B Ph(p) deg
3.3041E-4 c |U| m^3/s 3.3041E-4 C |U| m^3/s
23.551 D Ph(U) deg
0.0000 E Htot W
-1.6157E-2 F Edot W
ideal Solid type
!----- 2 -----

```

IESPEAKER the power piston

```

2.0000E-4 a Area m^2 2.9239E+5 A |p| Pa
1.0000 b R ohms -39.277 B Ph(p) deg
0.0000 c L H 3.3020E-4 C |U| m^3/s
10.000 d BLProd T-m 23.621 D Ph(U) deg
8.3000E-2 e M kg 40.076 E Htot W
1.0000E+4 f K N/m 21.992 F Edot W
0.0000 g Rm N-s/m 40.076 G WorkIn W
6.0000 h |I| A 19.931 H Volts V
140.00 i Ph(I) deg 6.0000 I Amps A
-312.09 J Ph(V/I) deg
2.9812E+5 K |Px| Pa
-39.82 L Ph(Px) deg
ideal Solid type
!----- 3 -----

```

COMPLIANCE a little space in front of the power piston

```

1.0000E-3 a SurfAr m^2 2.9239E+5 A |p| Pa
2.0000E-6 b Volume m^3 -39.277 B Ph(p) deg
2.7635E-4 c |U| m^3/s 2.7635E-4 C |U| m^3/s
18.316 D Ph(U) deg
40.076 E Htot W
21.652 F Edot W
ideal Solid type
!----- 4 -----

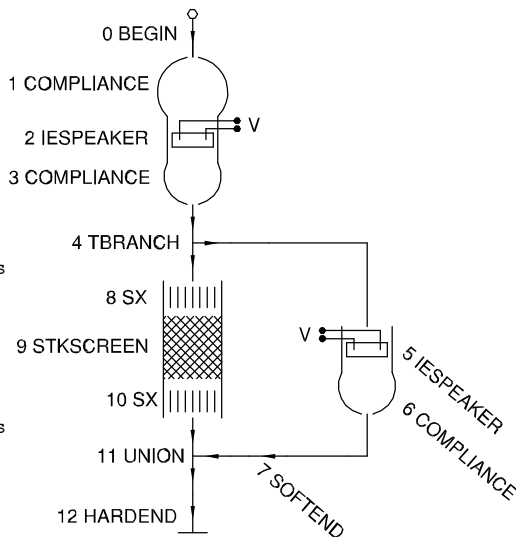
```

TBRANCH branch to displacer

```

-3.0131E+9 a Re(Zb) Pa-s/m^3 G 2.9239E+5 A |p| Pa
-3.2836E+9 b Im(Zb) Pa-s/m^3 G -39.277 B Ph(p) deg
-6.765 c HtotBr W 6.5609E-5 C |U| m^3/s
93.263 D Ph(U) deg
-6.765 E HtotBr W
-6.4851 F EdotBr W
28.137 G EdotTr W
!----- 5 -----

```



```

IESPEAKER a spring-mounted, driven moving mass
5.0000E-5 a Area m^2 2.5375E+5 A |p| Pa
0.5000 b R ohms -44.127 B Ph(p) deg
0.0000 c L H 6.5763E-5 C |U| m^3/s
1.0000 d BLProd T-m 93.397 D Ph(U) deg
7.0000E-3 e M kg -6.154 E Htot W
1190.0 f K N/m -6.1539 F Edot W
0.0000 g Rm N-s/m 0.6110 G WorkIn W
1.0000 h |I| A 1.6426 H Volts V
-30.00 i Ph(I) deg 1.0000 I Amps A
-41.929 J Ph(V/I) deg
4.4991E+4 K |Px| Pa
169.20 L Ph(Px) deg
ideal Solid type
!----- 6 -----
RPN Reassign Tm. (Displacer is "axially insulating.")
79.962 a G or T G 79.962 A Kelvin
inp =Tm
!----- 7 -----
COMPLIANCE a little volume for the connection
1.0000E-6 a SurfAr m^2 2.5375E+5 A |p| Pa
5.0000E-7 b Volume m^3 -44.127 B Ph(p) deg
5.7702E-5 C |U| m^3/s
103.08 D Ph(U) deg
-6.154 E Htot W
-6.154 F Edot W
ideal Solid type
!----- 8 -----
SOFTEND reconnect at UNION
0.0000 a Re(z) 2.5375E+5 A |p| Pa
0.0000 b Im(z) -44.127 B Ph(p) deg
0.0000 c Htot W 5.7702E-5 C |U| m^3/s
103.08 D Ph(U) deg
-6.154 E Htot W
-6.154 F Edot W
-29.176 G Re(z)
-18.80 H Im(z)
79.962 I T K
!----- 9 -----
SX aftercooler
sameas 10a a Area m^2 2.8988E+5 A |p| Pa
0.6000 b VolPor -39.755 B Ph(p) deg
1.0000E-3 c Length m 2.6449E-4 C |U| m^3/s
sameas 10d d rh m 4.0412 D Ph(U) deg
-45.267 e HeatIn W G 1.5747 E Htot W
300.00 f SolidT K =9H 27.670 F Edot W
300.22 G GasT K
300.00 H SolidT K
copper Solid type
!----- 10 -----
STKSCREEN regenerator
1.1670E-4 a Area m^2 2.5407E+5 A |p| Pa
0.6860 b VolPor -44.169 B Ph(p) deg
5.0000E-2 c Length m 5.6226E-5 C |U| m^3/s
1.3900E-5 d rh m -74.284 D Ph(U) deg
0.3000 e ksFrac 1.5747 E Htot W
6.1784 F Edot W
300.22 G TBeg K
79.962 H TEnd K
stainless Solid type
!----- 11 -----
SX cold heat exch
sameas 10a a Area m^2 2.5375E+5 A |p| Pa
0.6000 b VolPor -44.127 B Ph(p) deg
1.0000E-3 c Length m 5.7702E-5 C |U| m^3/s
sameas 10d d rh m -76.923 D Ph(U) deg
4.5794 e HeatIn W G 6.1540 E Htot W
80.000 f SolidT K =11H 6.1540 F Edot W
79.962 G GasT K
80.000 H SolidT K
copper Solid type
!----- 12 -----
UNION displacer cold end
8 a SegNum 2.5375E+5 A |p| Pa
sameas 8A b |p|Sft Pa =12A -44.127 B Ph(p) deg
sameas 8B c Ph(p)S deg =12B 8.9952E-15 C |U| m^3/s
sameas 8I d TSoft K =12G 99.432 D Ph(U) deg
-5.4001E-13 E Htot W
-9.1812E-10 F Edot W
79.962 G T K
!----- 13 -----
HARDEND close the end
0.0000 a R(1/z) =13G 2.5375E+5 A |p| Pa
0.0000 b I(1/z) =13H -44.127 B Ph(p) deg
0.0000 c Htot W =13E 8.9952E-15 C |U| m^3/s
99.432 D Ph(U) deg
-5.4001E-13 E Htot W
-9.1812E-10 F Edot W
-1.5481E-12 G R(1/z)
1.1431E-12 H I(1/z)
!----- 14 -----

```

```

RPN          simulate insulation in the displacer piston
0.0000 a =A?          1.0729E-12          A ChngeMe
7E 7F -
! The restart information below was generated by a previous run
! and will be used by DeltaEC the next time it opens this file.
guessz 0c 0d 0e 4a 4b 4c 6a 9e 11e
xprecn -3.8026E-3 7.5295E-3 -4.8956E-5 -2.2194E+4 1.4047E+4 -3.6186E-5 -1.5964E-4 4.3401E-4 -3.0683E-5
targs 9f 11f 12b 12c 12d 13a 13b 13c 14a

```

We are using IESPEAKER segments for both the displacer and the driver, because linear motors and loudspeakers share the same physical transduction mechanism.

The user will soon discover that this is a surprisingly robust model, considering the large number of guesses and targets: The model tolerates steps in many independent variables of several percent without getting lost. The user might next generate cooling-power curves by using the cold-temperature target as an independent plot variable and the cooling power as dependent plot variable; or the user might explore the frequency dependence of the cooler, by using frequency as an independent plot variable.

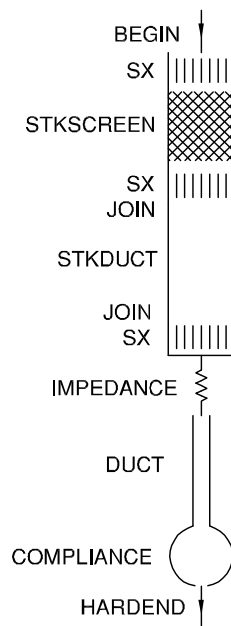
### 5.8.3. Pulse-tube refrigerator

Changing a Stirling cryocooler into an orifice pulse-tube refrigerator (OPTR) is a simple matter of replacing the cold piston with a pulse tube, heat exchanger, orifice, inertance (optional), and compliance volume in series. Figure 5.13 represents such a cooler. The example file <optr.out> represents a 300 Hz, 3 MPa helium orifice pulse-tube refrigerator. (It is not representative of any real hardware known to us. In fact, it is deliberately inefficient, to serve as a lesson in the next section.) The <.out> file is:

```

TITLE      a crude cooler design, not optimal
!----- 0 -----
BEGIN      Start with 8% p osc
3.0000E+6  a Mean P Pa
300.00     b Freq Hz
300.10     c TBeg K          G
2.4000E+5  d |p| Pa
0.0000     e Ph(p) deg
6.9646E-3  f |U| m^3/s      G
52.754     g Ph(U) deg      G
helium      Gas type
!----- 1 -----
SX         Aftercooler
1.0290E-3  a Area m^2          2.3017E+5  A |p| Pa
0.6900     b VolPor          -3.488    B Ph(p) deg
1.2500E-2  c Length m          5.8385E-3 C |U| m^3/s
6.4500E-5  d rh m              46.716   D Ph(U) deg
-455.16    e HeatIn W          50.679   E Htot W
300.00     f SolidT K          430.08   F Edot W
copper      Solid type          300.10   G GasT K
300.00     H SolidT K
!----- 2 -----
STKSCREEN  Regenerator
sameas 1a  a Area m^2          1.7212E+5 A |p| Pa
0.7300     b VolPor          -19.522   B Ph(p) deg
5.5000E-2  c Length m          1.2330E-3 C |U| m^3/s
2.4000E-5  d rh m              -29.70   D Ph(U) deg
0.3000     e ksFrac          50.679   E Htot W
104.44     F Edot W
300.10     G TBeg K
149.91     H TEND K
stainless   Solid type
!----- 3 -----
SX         Cold heat exchanger
sameas 5a  a Area m^2          1.1381E+5 A |p| Pa
0.6900     b VolPor          -14.781   B Ph(p) deg
2.0000E-3  c Length m          1.2333E-3 C |U| m^3/s
6.4500E-5  d rh m              -30.009   D Ph(U) deg
13.312     e HeatIn W          63.991   E Htot W
150.00     f SolidT K          67.712   F Edot W
copper      Solid type          149.91   G GasT K
150.00     H SolidT K
!----- 4 -----
JOIN      pulse-tube end effects

```



			1.1381E+5	A	p	Pa
			-14.781	B	Ph(p)	deg
			1.2179E-3	C	U	m <sup>3</sup> /s
			-30.009	D	Ph(U)	deg
			63.991	E	Htot	W
			66.871	F	Edot	W
			149.91	G	TBeg	K
			149.89	H	TEnd	K
!-----						
5	-----					
STKDUCT	Pulse tube					
5.6870E-5	a	Area	m <sup>2</sup>	Mstr	1.0284E+5	A  p  Pa
2.6740E-2	b	Perim	m	5a	-43.845	B Ph(p) deg
0.2000	c	Length	m		1.3033E-3	C  U  m <sup>3</sup> /s
1.0000E-5	d	WallA	m <sup>2</sup>		-51.003	D Ph(U) deg
					63.991	E Htot W
					66.491	F Edot W
					149.89	G TBeg K
					300.23	H TEnd K
!-----						
6	-----					
stainless	Solid type					
!-----						
JOIN	pulse-tube end effects					
			1.0284E+5	A	p	Pa
			-43.845	B	Ph(p)	deg
			1.2882E-3	C	U	m <sup>3</sup> /s
			-51.003	D	Ph(U)	deg
			63.991	E	Htot	W
			65.724	F	Edot	W
			300.23	G	TBeg	K
			300.20	H	TEnd	K
!-----						
7	-----					
SX	Hot heat exchanger					
sameas	5a	a	Area	m <sup>2</sup>	2.2011E+4	A  p  Pa
0.6900	b	VolPor			-19.722	B Ph(p) deg
5.0000E-3	c	Length	m		1.2889E-3	C  U  m <sup>3</sup> /s
6.4500E-5	d	rh	m		-51.331	D Ph(U) deg
-63.991	e	HeatIn	W	G	2.8422E-13	E Htot W
300.00	f	SolidT	K	=7H	12.081	F Edot W
					300.20	G GasT K
					300.00	H SolidT K
!-----						
8	-----					
IMPEDANCE	The orifice					
1.0000E+7	a	Re(Zs)	Pa-s/m <sup>3</sup>		1.2938E+4	A  p  Pa
0.0000	b	Im(Zs)	Pa-s/m <sup>3</sup>		11.755	B Ph(p) deg
					1.2889E-3	C  U  m <sup>3</sup> /s
					-51.331	D Ph(U) deg
					2.8422E-13	E Htot W
					3.7742	F Edot W
!-----						
9	-----					
DUCT	inertance					
1.0000E-5	a	Area	m <sup>2</sup>	Mstr	2.2780E+4	A  p  Pa
1.1210E-2	b	Perim	m	9a	-141.21	B Ph(p) deg
3.0000E-2	c	Length	m		1.2882E-3	C  U  m <sup>3</sup> /s
5.0000E-4	d	Strough			-51.345	D Ph(U) deg
					2.8422E-13	E Htot W
					3.3272E-2	F Edot W
!-----						
10	-----					
ideal	Solid type					
!-----						
COMPLIANCE	Reservoir volume					
1.2680E-2	a	SurfAr	m <sup>2</sup>	10b	2.2780E+4	A  p  Pa
1.5000E-4	b	Volume	m <sup>3</sup>	Mstr	-141.21	B Ph(p) deg
					1.6700E-12	C  U  m <sup>3</sup> /s
					2.0161	D Ph(U) deg
					2.8422E-13	E Htot W
					-1.5237E-8	F Edot W
!-----						
11	-----					
HARDEND	The end					
0.0000	a	R(1/z)		=11G	2.2780E+4	A  p  Pa
0.0000	b	I(1/z)		=11H	-141.21	B Ph(p) deg
0.0000	c	Htot	W	=11E	1.6700E-12	C  U  m <sup>3</sup> /s
					2.0161	D Ph(U) deg
					2.8422E-13	E Htot W
					-1.5237E-8	F Edot W
					-2.8802E-8	G R(1/z)
					2.1522E-8	H I(1/z)

The Stirling part of the system is modeled as a stacked-screen regenerator **STKSCREEN** and two stacked-screen heat exchangers **SX**. We model the pulse tube itself as a **STKDUCT**, using Rott's wave equation and enthalpy flux equation in boundary-layer approximation, because the pulse-tube diameter is  $\gg \delta_\kappa$ . The **JOIN** segments are used at the ends of the pulse tube to model the temperature overshoots and adiabatic-isothermal interface losses there, as explained in Section 10.8. The heat exchanger at the hot end of the pulse tube is the **HX**. The orifice, inertance, and compliance are easily modeled as **DELTAEC IMPEDANCE**,

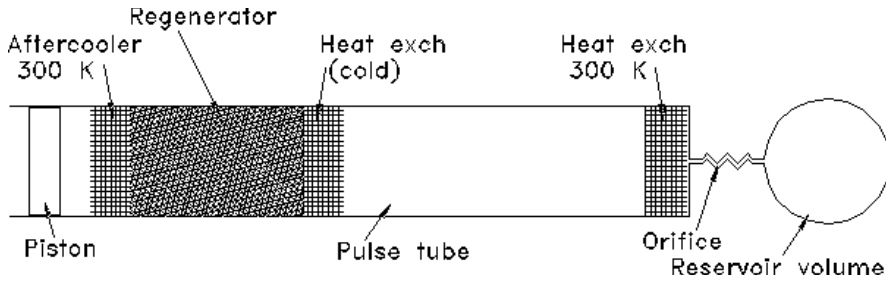


Figure 5.13: An orifice pulse-tube refrigerator (OPTR). The orifice can be purely resistive, as shown here, but adding inertance in series with the resistive orifice improves performance.

DUCT, and COMPLIANCE, respectively. Our use of zero for the imaginary part of the IMPEDANCE reflects our intention that this orifice should truly be resistive, with its pressure drop in phase with mass flux.

For purposes of illustration here, we will regard the geometry of the apparatus as given, and will explore its performance for a given, fixed input  $|p_1|$ . The guess–target summary indicates our point of view:

GUESS	0c	0f	0g	1e	3e	7e
name	BEGIN:TBeg	BEGIN: U	BEGIN:Ph(U)	SX:HeatI	SX:HeatI	SX:HeatI
units	K	m <sup>3</sup> /s	deg	W	W	W
value	300.10	6.9646E-03	52.754	-455.16	13.312	-63.991
TARGET	1f	3f	7f	11a	11b	11c
name	SX:Est-T	SX:Est-T	SX:Est-T	HARDE:R(1/z	HARDE:I(1/z	HARDE:Htot
units	K	K	K			W
value	300.00	150.00	300.00	0.0000	0.0000	0.0000
result	300.00	150.00	300.00	3.6105E-09	-2.7741E-09	-2.8422E-14

Three of the six targets fix the hot and cold temperatures at 300 K and 150 K. We leave the amplitude of the oscillatory pressure at the BEGINning at 8% of mean pressure, and leave the frequency fixed at 300 Hz. Hence, we are asking: What is the cooling power at 150 K, and how much input power, volume flow rate, etc. are required, for fixed frequency and pressure amplitude, and for this geometry? The result, given in the file listings above: 13.3 W of cooling power, requiring over 500 W of input power from the compressor.

As with most OPTR models we have worked with in DELTAEC, this one is not particularly “robust.” An abrupt change of a typical variable by 20% or 30% will likely cause DELTAEC to get hopelessly lost, unable to adjust its guesses to hit the targets. Much of the shooting-method “fragility” of OPTR models (and other models with a STKDUCT or STKCONE) in DELTAEC is due to the fact that small changes in variables near the BEGINning, such as  $p_1$ ,  $U_1$ , and the heat removed at a heat exchanger, have a large effect on temperatures at the end of the pulse tube. Some of the fragility is due to the fact that OPTR models typically have a large number of guesses and targets. When you encounter a fragile DELTAEC model, try to reduce the number of guesses and targets as much as possible (particularly in initial design explorations when you are changing things dramatically) and, once you have a convergent model, make only small changes in variables. Try negative values of the Normalization Mode,<sup>1</sup> accessible in the Edit, Options pulldown menu; we often find that Mode

<sup>1</sup>As described in Section 8.3.2, the value of Mode is stored in the `<.ini>` file, and it is often convenient to rename a model’s `<.ini>` file to be `<default.ini>` so all models in a given folder can automatically benefit from a good value of Mode.

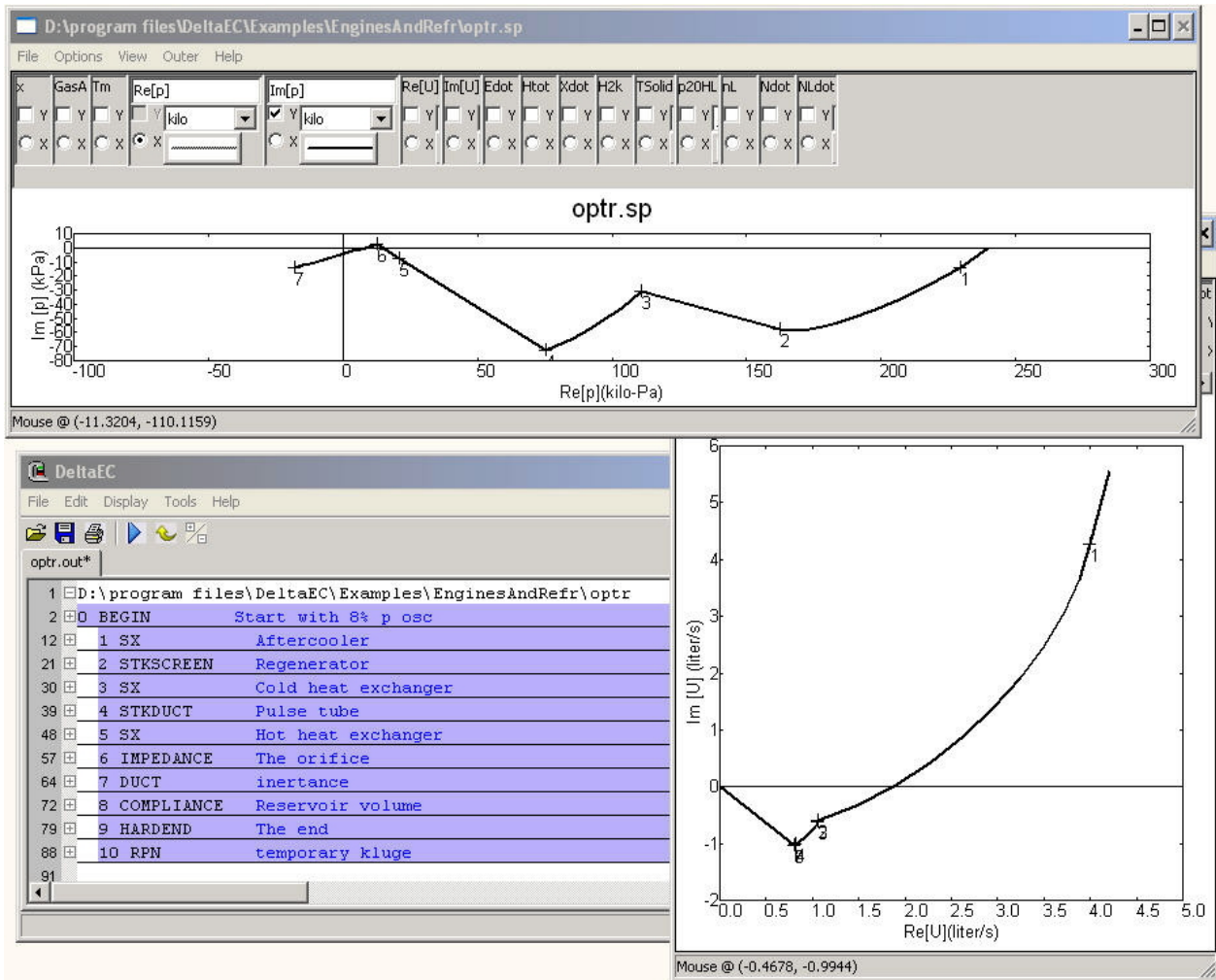


Figure 5.14: Plotting <optr.sp> using  $Re[p_1]$  as the horizontal axis and  $Im[p_1]$  as the vertical axis yields a pressure-phasor plot. The SegEndMarks option tags the curve with numbers indicating the ends of the segments. Cloning the plot and switching  $p_1$  to  $U_1$  yields a volume-flow-rate phasor plot. THESE can also be created with Phasor Mode in the plotter's Options pulldown menu.



$= -4$  works much better than DELTAEC’s default Mode = 1. Try reducing DELTAEC’s convergence tolerance (Section 8.3.2) if you have to use more than 5 guesses and 5 targets. Remember to subdivide a large change in a variable into a series of small changes, using an incremental plot. A fast computer and frequent hard-disk saving of satisfactory converged models will minimize frustration. See Section 5.4 for additional tips.

Examination of the pulse tube segment STKDUCT above shows a possible problem: The pulse tube figure of merit  $\dot{H}_{\text{tot}}/\dot{E}$ , which can theoretically be as high as 1.000 but is usually more like 0.8, may be unrealistically high:  $\dot{H}_{\text{tot}}/\dot{E} \simeq 64.0 \text{ W}/66.5 \text{ W} \simeq 0.96$ . DELTAEC knows nothing about jet- or streaming-driven convection, and pulse-tube experimentalists are only beginning to learn how to reliably avoid such convection. (For a discussion of streaming-driven convection, see Refs. [29] or [30], and [31] and [13].) To force DELTAEC to accommodate a reduced pulse-tube figure of merit, the user can introduce an RPN and an additional guess–target pair. The RPN instruction line can compute the ratio of  $\dot{H}_{\text{tot}}$  to  $\dot{E}$  in the pulse tube, and the target value in line a can be set to something like 0.8. The user can simulate the thermal loading of streaming-driven convection, etc. by letting DELTAEC guess an unphysically large value for the cross section of the pulse tube solid wall (line 4d), which then conducts significant heat from hot to cold, allowing DELTAEC to meet its target of 0.8. We will not do so here.

Our choices for nearly all input variables in this model were random, not guided by good design sense or any real cryocooler. In particular, our choice of  $\text{Re}[Z_s] = 1 \times 10^7$  for the orifice impedance above was random. To find a better orifice setting, use  $\text{Re}[Z_s]$  as an independent plot variable, letting it range from  $1 \times 10^7$  to  $1 \times 10^8$ . The cooling power peaks at 13.6 W for  $\text{Re}[Z_s] = 2.3 \times 10^7$ . This is not much improvement. For guidance on what to try changing next to improve the performance, we consider *exergy* in the next section.

## 5.9. Exergy flow

Among the many important thermodynamic variables having units of power (e.g., Watts) is *exergy* flow. In thermodynamics, exergy represents the ability to do useful work, in principle, assuming unlimited access to an infinitely large thermal reservoir at an “ambient” temperature  $T_0$  at which heat has no value [32].

Novice thermoacousticians sometimes suppose that the acoustic power  $\dot{E}$  represents the ability to do useful work, because the treatment of intensity and power in adiabatic waves in introductory acoustics courses implies that the face of a moving piston is a perfect interface between mechanical power ( $Fu$ ) and acoustic power ( $pU$ ). However, thermoacoustic waves are generally not adiabatic. The “heat” carried near solid surfaces by the  $s_1\tilde{u}_1$  term in the second-order enthalpy flow represents an ability to do useful work that is separate from  $\dot{E}$ , because this term in the enthalpy flow can be usefully coupled to the assumed thermal reservoir at  $T_0$  through a heat engine if  $s_1\tilde{u}_1$  is flowing at a temperature different from  $T_0$ . Taking this into account [13], the exergy flow associated with a thermoacoustic wave is

$$\dot{X} = \frac{T_0}{T_m} \dot{E} + \left(1 - \frac{T_0}{T_m}\right) \dot{H}_{2,k}. \quad (5.16)$$

(For an adiabatic sound wave,  $\dot{H}_{2,k} = \dot{E}$ , so this reduces to  $\dot{X} = \dot{E}$ , independent of  $T_m$ ,

as expected.) DELTAEC's ambient temperature  $T_0$  is set to 300 K by default in the `BEGIN` segment; this default can be overridden by using `=Tzero` in a subsequent `RPN` segment.

The usefulness of the concept of exergy arises from its relationship to irreversible entropy generation, expressed by the Guoy-Stodola theorem [32],

$$\Delta\dot{X} = T_0\dot{S}_{\text{gen}}. \quad (5.17)$$

In this simplified expression of the theorem,  $\Delta\dot{X}$  represents the difference in exergy flow between two locations in an apparatus between which exergy cannot flow in or out laterally, and the theorem says that this difference is the product of  $T_0$  and the rate of irreversible entropy generation between those two locations. In a perfectly efficient apparatus, there is no irreversible entropy generation, so  $\Delta\dot{X} = 0$  (except where  $\dot{X}$  flows in or out, e.g., at a heat exchanger or a `**DUCER` or `**SPEAKER`). Thus, a graph of  $\dot{X}$  vs  $x$  can help show the user what parts of an apparatus are most responsible for its inefficiency.

For example, consider the OPTR described in Section 5.8.3. The `<.sp>` plot of  $\dot{E}$ ,  $\dot{H}_{\text{tot}}$ , and  $\dot{X}$  is shown in Fig. 5.15. Most of the exergy is lost in the regenerator, and the rate of exergy loss is steepest near the ambient end of the regenerator. This suggests that viscous losses in the regenerator may be severe. Indeed, the `<.out>` file shows a large pressure drop in the regenerator. The user might try changing the dimensions of the regenerator to improve the performance. Figure 5.15 also shows significant drops in  $\dot{X}$  in the ambient and cold heat exchangers. There, the analysis of the situation is a little more complicated, because the heats flowing into a heat exchanger contribute exergy according to [13]

$$\Delta\dot{X} = \dot{Q}(1 - T_0/T_m), \quad (5.18)$$

where  $T_0$  is ambient temperature and  $T_m$  is the local temperature. At the ambient heat exchanger,  $T_m = T_0$ , so the drop in exergy there is entirely lossy. At the cold heat exchanger,  $T_m \neq T_0$ , but the loss in thermoacoustic exergy due to  $\dot{Q}$  is only 15 W, so most of the drop in  $\dot{X}$  in Fig. 5.15 at the cold heat exchanger is lossy. Thus, the heat exchangers are also contributing significantly to the poor performance in this OPTR. At this point, the user might suspect that the poor performance of this OPTR arises from trying to force too much flow through pores that are too tight or an area that is too small.

Subsequent chapters describe additional steady-flow and mixture-separation features in DELTAEC. These phenomena also contribute to  $\dot{X}$ , and the terms that they add to Eq. (5.16) are given in Eq. (8.9).

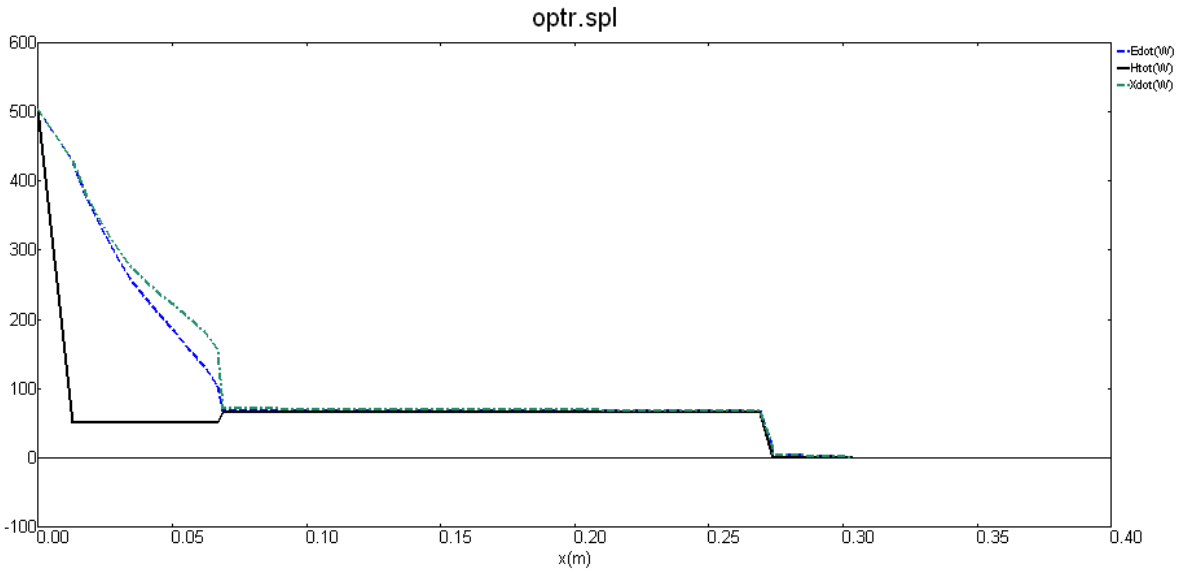


Figure 5.15: Graph of total power (black solid line), acoustic power (blue dashed line), and exergy flow (green dot-dashed line) in the OPTR example. The locations where  $\dot{H}_{\text{tot}}$  changes are the three heat exchangers.

## 6. STEADY FLOW, TOO

The superposition of steady flow and oscillating flow creates many interesting and important effects that can be modeled with DELTAEC. A nonzero steady flow profoundly affects  $T_m(x)$  in stacks and regenerators, creates Doppler shifts [33] and increases the turbulent dissipation of acoustic power in DUCTs, and causes a small  $x$  dependence in the time-averaged pressure. In systems such as Reid’s refrigerator (shown below in Fig. 6.1), externally forced steady flow provides the ability to deliver cold gas from a thermoacoustic refrigerator without a cold heat exchanger. In more conceptually subtle situations, the acoustic oscillations themselves cause the steady flow, which is then called “streaming” by acousticians [34]. In some circumstances, such as the thermoacoustic-Stirling hybrid engine shown in Fig. 1.7 or 6.4, we try to understand the streaming well enough to eliminate it. In other circumstances, such as the self-circulating heat exchanger shown in Fig. 1.9, we try to make streaming as large as possible without undue dissipation of acoustic power.

### 6.1. Principles of calculation

DELTAEC accounts for steady flow as  $\dot{N}$ , the moles per second<sup>1</sup> flowing through the segments in the positive- $x$  direction. It can be specified or guessed in BEGIN, and it splits up at TBRANCHs, adds up at UNIONs, and can be abruptly changed in RPNs. Otherwise,  $\dot{N}$  flows unchanged through segments. (But note that it seldom makes sense to imagine that nonzero  $\dot{N}$  could flow *through* one of the *enclosed* \*EDUCER or \*ESPEAKER segments.)

Steady flow carries power

$$\dot{H}_{\dot{N}} = \dot{N}mw_m, \quad (6.1)$$

where  $m$  is the molar mass and  $w_m$  is the mean enthalpy per unit mass. The total power

$$\dot{H}_{\text{tot}} = \dot{H}_{2,k} + \dot{H}_{\dot{N}} \quad (6.2)$$

is the power that obeys the first law of thermodynamics for open control volumes. It is the sum of the second-order time-averaged thermoacoustic enthalpy flux, the ordinary thermal conduction in the  $x$  direction, and  $\dot{H}_{\dot{N}} = \dot{N}mw_m$ .

In DELTAEC, mean enthalpy is referenced to zero at temperature  $T_0$ , which is set to 300 Kelvin by default in the first BEGIN segment of a model. (This default can be overridden

---

<sup>1</sup>DELTAEC’s moles are gram-moles:  $6.022 \times 10^{23}$  particles (e.g., helium’s molar mass is 0.004002 kg/mol).

We might have chosen to use mass flow or volume flow instead of molar flow as DELTAEC’s steady-flow parameter. We rejected volume flow because it changes with  $x$  when  $T_m$  depends on  $x$ , which seemed unnecessarily confusing. We chose molar flow instead of mass flow for convenience in the context of mixture separation, which is introduced in Chapter 7.

by using the =Tzero operation in an RPN segment.) For an ideal gas, Eq. (6.1) can then be regarded as

$$\dot{H}_{\dot{N}} = \dot{N}mc_p(T_m - T_0). \quad (6.3)$$

Nonzero  $\dot{N}$  also contributes a term

$$\dot{N}(mw_m - T_0ms_m) \quad (6.4)$$

to exergy flow  $\dot{X}$ . Like the enthalpy per unit mass  $w_m$ , the entropy per unit mass  $s_m$  is referenced<sup>2</sup> to zero at  $T_0$ .

DELTAEC also introduces a new integration variable, the second-order time-averaged head-loss pressure  $p_{2,0,HL}$ , to keep track of the small but important effects of steady flow on the time-averaged pressure. The second-order head-loss pressure  $p_{2,0,HL}$  is initialized at zero in BEGIN and evolves with  $x$  in most other segments. When  $\dot{N}$  and  $p_{2,0,HL}$  are involved in a DELTAEC calculation, the numerical-integration picture that is summarized in Eqs. (5.6)–(5.9) for thermoacoustics with  $\dot{N} = 0$ , namely

$$\begin{aligned} \frac{dp_1}{dx} &= \mathcal{F}_{\text{momentum}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega; \text{geometry, gas properties, etc.}), \\ \frac{dU_1}{dx} &= \mathcal{F}_{\text{continuity}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \text{geometry, gas properties, etc.}), \\ \frac{dT_m}{dx} &= \mathcal{F}_{x\text{-energy}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \text{geometry, gas properties, etc.}), \\ \frac{d\dot{H}_{\text{tot}}}{dx} &= \mathcal{F}_{\text{lateral energy}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \text{geometry, gas properties, etc.}). \end{aligned}$$

must be replaced by this slightly more complicated picture:

$$\frac{dp_1}{dx} = \mathcal{F}_{\text{momentum}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \dot{N}; \text{geometry, gas properties, etc.}), \quad (6.5)$$

$$\frac{dU_1}{dx} = \mathcal{F}_{\text{continuity}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \dot{N}, \text{geometry, gas properties, etc.}), \quad (6.6)$$

$$\frac{dT_m}{dx} = \mathcal{F}_{x\text{-energy}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \dot{N}, \text{geometry, gas properties, etc.}), \quad (6.7)$$

$$\frac{d\dot{H}_{\text{tot}}}{dx} = \mathcal{F}_{\text{lateral energy}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \dot{N}, \text{geometry, gas properties, etc.}), \quad (6.8)$$

$$\frac{dp_{2,0,HL}}{dx} = \mathcal{F}_{\text{head loss}}(p_1, U_1, T_m, \dot{H}_{\text{tot}}; p_m, \omega, \dot{N}, \text{geometry, gas properties, etc.}). \quad (6.9)$$

---

<sup>2</sup>Entropy per unit mass is referenced to zero at  $T = T_0$  and  $p = p_m$ . Since all parts of a thermoacoustic apparatus share the same  $p_m$ , this unconventional reference pressure does not affect any entropy or exergy differences [e.g., Eq. (5.17)] within a DELTAEC model.

Please note that  $dp_{2,0,HL}/dx$  is usually nonzero even when  $\dot{N} = 0$ , because  $dp_{2,0,HL}/dx$  arises from

$$\rho_m U_{2,0} \simeq \dot{N} m - \frac{1}{2} \text{Re} \int [\rho_1 \tilde{u}_1] dA_{\text{gas}}, \quad (6.10)$$

where  $m$  is the molar mass, as explained in more detail near Eq. (8.15).

Our confidence in the accuracy of the DELTAEC implementation of these equations is lower than in the equations described in Chapters 2 and 5. We are very confident of the effect of nonzero  $\dot{N}$  on  $\mathcal{F}_{x\text{-energy}}$  and  $\mathcal{F}_{\text{lateral energy}}$  under all circumstances we have encountered, and confident of the effect of nonzero  $\dot{N}$  on all the equations when the steady and oscillating flows are both laminar, but for turbulent flow our understanding of  $\mathcal{F}_{\text{head loss}}$  and of the effects of nonzero  $\dot{N}$  on  $\mathcal{F}_{\text{momentum}}$  and  $\mathcal{F}_{\text{continuity}}$  are still evolving, and are mostly based on the quasi-steady approximation, namely that the behavior at any instant of time  $t$  is the same as it would be for steady flow with volume flow rate  $U = \dot{N}m/\rho_m + \text{Re}[U_1 e^{i\omega t}]$ , an approximation with obvious shortcomings in acoustics where inertial effects of gas acceleration are usually as important as anything else. The complicated details of the current steady-flow DELTAEC implementation are given in Chapter 10.

One other variable is sometimes useful in the context of nonzero  $\dot{N}$ . The (total) second-order, time-averaged pressure  $p_{2,0,\text{tot}}$  corresponds to pressure changes as measured by side-wall pressure transducers [35], because  $p_{2,0,\text{tot}}$  is given by

$$p_{2,0,\text{tot}} = p_{2,0,\text{rev}} + p_{2,0,HL} \quad (6.11)$$

where the reversible part of the second-order time-averaged pressure [36] is given by

$$p_{2,0,\text{rev}}(x) = \frac{|p_1|^2}{4\rho_m a^2} - \frac{\rho_m |U_1|^2}{4A_{\text{gas}}^2}. \quad (6.12)$$

The variable  $p_{2,0,\text{tot}}$  is available in RPN segments. Note that it depends on area, which may not have a sensible value after segments without area, such as BEGIN, TBRANCH, UNION, or IMPEDANCE.

User-defined gases specified by `<.tpf>` files cannot be used with  $\dot{N}$  enabled, because Eq. (6.1) requires  $m$ , which is not available in `<.tpf>` format. To circumvent this shortcoming when a user-defined gas must be used with nonzero  $\dot{N}$ , use a `<.tpm>` file and set the mole fraction of one component to zero.

In the rest of this chapter, we illustrate DELTAEC's steady-flow features with three examples. In the first example, nonzero  $\dot{N}$  is externally driven and  $p_{2,0,HL}$  is of little interest. In the second example,  $\dot{N} = 0$  is desired, but must be accomplished through careful balancing of small  $p_{2,0,HL}$  effects. In the third example, nonzero  $\dot{N}$  is driven by large  $p_{2,0,HL}$  effects.

## 6.2. Reid's refrigerator

Bob Reid's standing-wave refrigerator, shown in Fig. 6.1 and described in Refs. [37, 38, 39], was built for the study of large nonzero  $\dot{N}$  through the stacks. The standing wave in the resonator had pressure nodes at the top-center and bottom-center locations, and out-of-phase pressure antinodes near the left and right speaker pairs. Steady flow of ambient-temperature

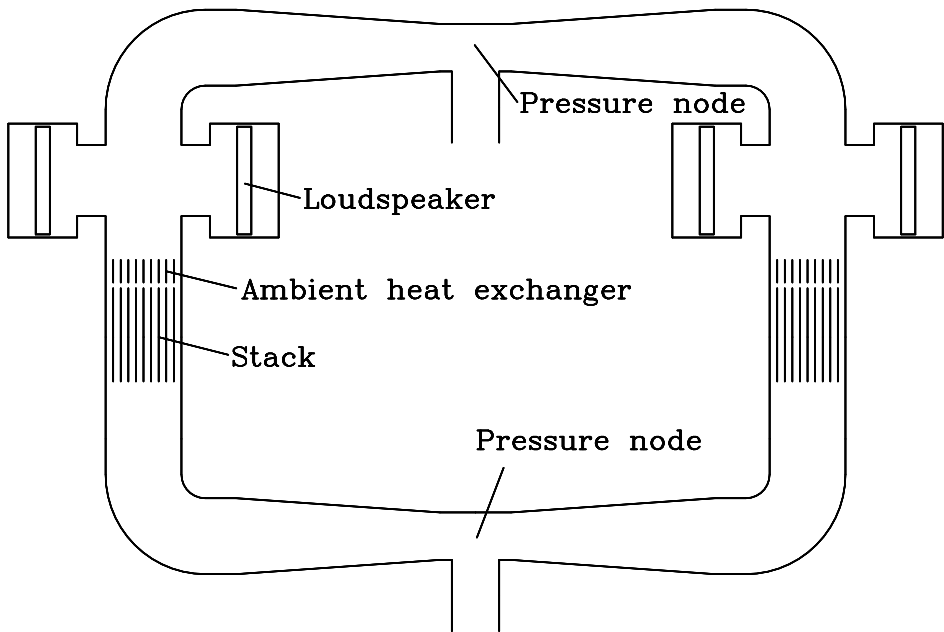


Figure 6.1: Standing-wave refrigerator with nonzero steady flow through the stacks, studied by Reid [37, 38, 39]. Steady flow was injected at the upper pressure node at ambient temperature and flowed out at the lower pressure node at a cold temperature.

working gas was injected into the resonator at the upper pressure node. This flow was cooled as it passed through the stacks, so cold gas flowed out of the resonator at the lower pressure node. There was no concern that acoustic power might flow out into the steady-flow piping because the steady-flow connections were made at pressure nodes.

To model this apparatus in DELTAEC, we will consider only one half of the apparatus, because the other half is identical (though shifted in time phase by  $180^\circ$ ). Focusing only on the thermal aspects, we begin the model just below the speakers in the right half, and end it where the right half joins the left half, at the exit port at the bottom. The BEGIN and the guess–target summary reflect the following perspective: The refrigerator is driven at a known frequency and with a known pressure amplitude near the drivers; a known steady flow of gas at a known temperature goes down through the apparatus; the ambient heat exchanger solid is held at a known temperature by cooling water. We want to learn the temperature of the exiting steady flow of cold gas, among other things.

```

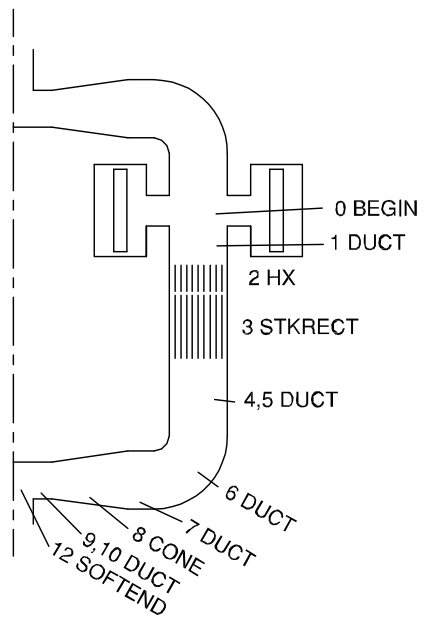
GUESS      Of          Og          3a          5e
name BEGIN:|U|    BEGIN:Ph(U)    RPN:G or T    HX:HeatI
units      m^3/s      deg              W
value     5.5034E-02  8.7817          6.3995E-02   -135.8
TARGET     5f          15a             15b          15c
name      HX:Est-T    SOFTE:Re(z)    SOFTE:Im(z)  SOFTE:Htot
units      K
value     300.00      0.0000          0.0000       -66.713
result    300.00      -5.3646E-17    3.9259E-16   -66.713
TITLE     segments 1-7 < Bob 9/97; segs 8-14 < Hiller ntbk, pg one, 10/97
!->reid4.out
!----- 0 -----
BEGIN      Initialize in main duct where drivers are attached
3.2388E+05 a Mean P Pa
92.000    b Freq Hz
303.00    c T-beg K

```

```

6500.0 d |p| Pa
90.000 e Ph(p) deg
5.5034E-02 f |U| m^3/s G
8.7817 g Ph(U) deg G
0.6700 i Ndot mol/s
0.9200 j nL
HeAr Gas type
!----- 1 -----
DUCT Pre-stack Duct
1.8430E-2 a Area m^2 6234.3 A |p| Pa
0.4813 b Perim m 89.692 B Ph(p) deg
0.1494 c Length m 7.3716E-2 C |U| m^3/s
ideal Solid type 6.4552 D Ph(U) deg
69.087 E Htot W
27.060 F Edot W
!----- 2 -----
RPN display energies
0.0000 a G or T 27.307 A W H2k
69.087 B W Htot
Htot H2k
!----- 3 -----
RPN Fix Tm. Inp = frac of heat that should go to mean flow
6.3995E-2 a G or T G 302.38 A K
35.997 B W h2k
H2k inp 5e * - =H2k ; inp 5e * Ndot / m / cp / Tm + =Tm
!----- 4 -----
RPN verify energy conservation
0.0000 a G or T 35.997 A W H2k
69.087 B W Htot
Htot H2k
!----- 5 -----
HX Ambient Heat exchanger
1.7211E-2 a Area m^2 6169.2 A |p| Pa
0.5227 b GasA/A 89.798 B Ph(p) deg
1.2700E-2 c Length m 7.4628E-2 C |U| m^3/s
6.3500E-4 d y0 m 6.2352 D Ph(U) deg
-135.8 e HeatIn W G -66.713 E Htot W
300.00 f SolidT K =5H 25.809 F Edot W
302.38 G GasT K
300.00 H SolidT K
-99.802 I H2k W
copper Solid type
!----- 6 -----
STKRECT rectangular-pore stack
1.8824E-2 a Area m^2 5547.8 A |p| Pa
0.7050 b GasA/A 93.314 B Ph(p) deg
0.1524 c Length m 9.1544E-2 C |U| m^3/s
4.0640E-4 d aa m 3.8385 D Ph(U) deg
1.1811E-4 e Lplate m -66.713 E Htot W
6.3500E-3 f bb m 2.3232 F Edot W
302.38 G TBeg K
295.21 H TEnd K
8.7823E-12 I H2k W
kapton Solid type
+ 7 DUCT From cold hx to bolt flange, + a half inch
+ 8 DUCT Duct, beginning 1/2 in below big bolt flange
+ 9 DUCT Elbow, 6 inch diam, 5 inch radius of curvature on centerline
+ 10 DUCT straight part of big black cone
+ 11 CONE The long black plastic cone
+ 12 DUCT little straight section of black cone
+ 13 DUCT half the white "tee", up to the symmetry midpoint
!----- 14 -----
RPN Expect Htot = Edot + Ndot*m*enth
0.0000 a Target (t) -66.713 A RPNval
Edot Ndot m * enth * +
!----- 15 -----
SOFTEND pressure node
0.0000 a Re(z) = 12G? 4.5144E-12 A |p| Pa
0.0000 b Im(z) = 12H? -74.05 B Ph(p) deg
sameas 14A c Htot W = 12E? 0.1597 C |U| m^3/s
3.5289 D Ph(U) deg
-66.713 E Htot W
-4.8796E-14 F Edot W
-5.3646E-17 G Re(z)
3.9259E-16 H Im(z)
295.21 I T K
-23.967 J p20h1 Pa
!----- 16 -----
RPN U mean here (liter/sec)
999.00 a Target (t) 5.0776 A RPNval
Ndot m * rho / 1000 *

```



Output 3H tells us that the refrigerator cools the gas to 295 Kelvin. The `<.sp>` plots shown in Figs. 6.2 and 6.3 show details of the temperature profile in the stack and the heat exchanger upstream of it, and how the energy flow is independent of  $x$  in the stack



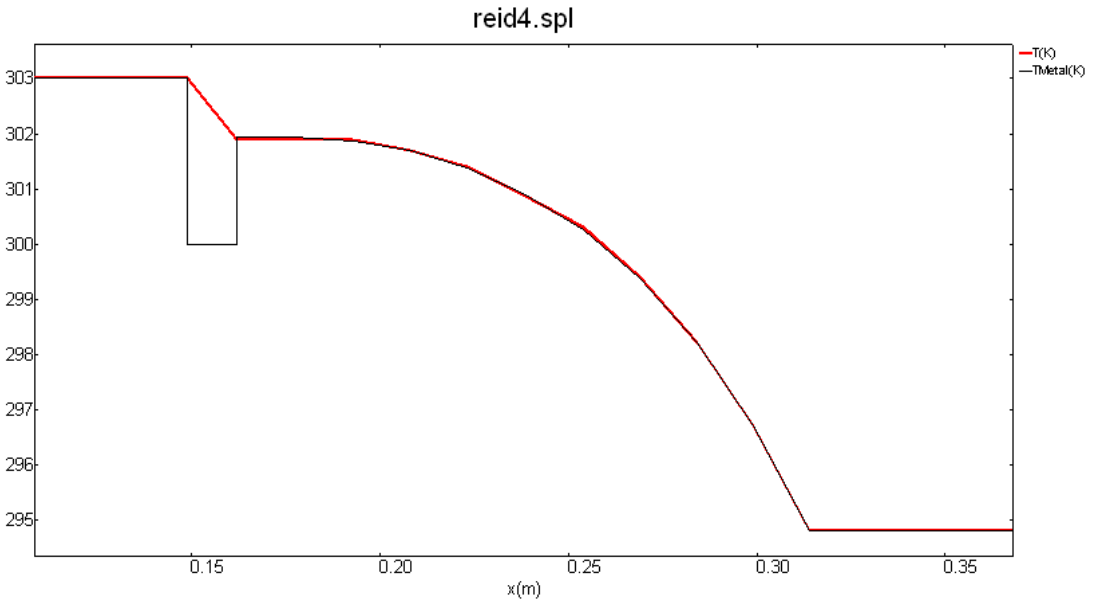


Figure 6.2: Temperatures vs  $x$  in Reid's refrigerator. Red line, gas temperature  $T_m(x)$ . Black line, solid temperature  $T_{\text{solid}}(x)$ .

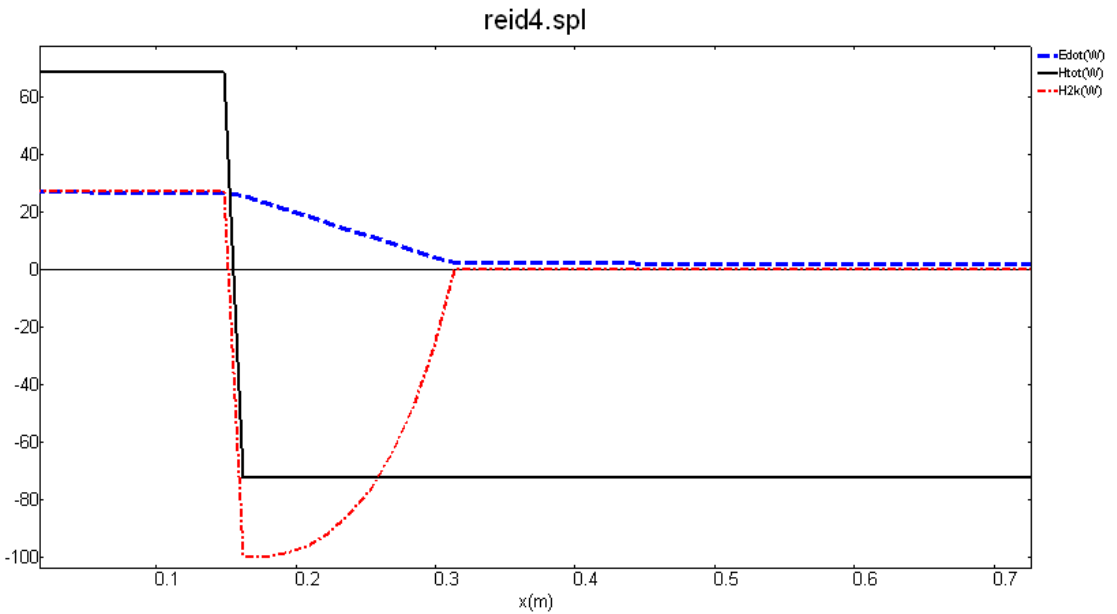


Figure 6.3: Power flows in Reid's refrigerator. Blue dashed line is  $\dot{E}$ . Solid black line is  $\dot{H}_{\text{tot}}$ . Dot-dashed red line is  $\dot{H}_{2,k}$ .

even as energy shifts from  $\dot{N}mc_pT_m$  to  $\dot{H}_{2,k}$ . The temperature profile is so severely curved—essentially horizontal near the ambient end—that an increase in flow above the present value of 0.67 mol/s (segment 0), which is about 5 liters per second (segment 13), would probably be undesirable. Note that the solid temperature equals the gas temperature everywhere except in the heat exchanger, where the solid is cooler than the gas so it can remove ambient heat from the gas.

When  $\dot{N} \neq 0$ , the energy boundary conditions at the **BEGIN**ning and **END** of a model often require careful thought. The most commonly desired boundary condition is

$$\dot{H}_{\text{tot}} = \dot{E} + \dot{N}mw_m. \quad (6.13)$$

On its face, this condition sets total power equal to the sum of acoustic power and steady-flow enthalpy. Comparison of this expression with Eqs. (6.1) and (6.2) here and with Eq. (5.29) in Ref. [13] shows what this condition deliberately omits: axial thermal conduction in the gas and solid, and boundary-layer, bucket-brigade thermoacoustic entropy transport in the gas. Thus, this boundary condition is appropriate whenever a long, large-diameter, unremarkable **DUCT** or other segment starts or ends a model with  $\dot{N} \neq 0$ . This condition is encountered so frequently that it is taken to be the default in **BEGIN**, which can be overridden by accessing the Default Htot / Other Htot choice in the Optional Parameters in **BEGIN**. The Default Htot choice sets the initial value of  $\dot{H}_{\text{tot}}$  according to Eq. (6.13). This condition must also sometimes be enforced in subsequent **SOFTENDS**, where an **RPN** that expresses Eq. (6.13) can be used, as illustrated above in segments 14 and 15.

Another new issue is addressed in segment 3 now that  $\dot{N} \neq 0$ . This variable,  $F_{\dot{Q}\dot{N}}$ , tells **DELTAEC** what fraction of the heat exchanger's heat  $\dot{Q}$  should be allocated to changing energy in the steady flow through the heat exchanger; the remainder is allocated to  $\dot{H}_{2,k}$ . When  $\dot{N} \neq 0$ , we often use  $F_{\dot{Q}\dot{N}}$  as a guess, to let **DELTAEC** meet temperature and energy-flow targets.

Output J in segment 12 gives  $p_{2,0,HL} = -24$  Pa. Since  $p_{2,0,HL}$  is initialized to zero in **BEGIN**, this shows that only 24 Pa of steady pressure is needed to push the 5 liters per second through the stack and heat exchanger. To learn how much of that falls across the heat exchanger and how much falls across the stack, we could insert an **RPN** segment between the heat exchanger and the stack, with **p20HL** in line **b**, or display  $p_{2,0,HL}$  in a `<.sp>` plot.

### 6.3. TASHE

The thermoacoustic-Stirling hybrid engine (**TASHE**), shown in Fig. 6.4 and described in detail in Refs. [5, 40], will be used to illustrate the use of **DELTAEC** to design a toroidal thermoacoustic apparatus without Gedeon streaming [41]. Without intelligent design, Gedeon streaming around the loop of the **TASHE** can convect an enormous amount of heat away from the hot heat exchanger.

We created this **DELTAEC** model starting from Backhaus's 1998 **DELTAEC** model (Appendix B4 of Ref. [13]; also [www.lanl.gov/thermoacoustics/Tashe.out](http://www.lanl.gov/thermoacoustics/Tashe.out)). We removed some inessential segments to reduce the total number, in an effort to save room in the Users Guide. We modified some segment types to bring the model up to date in **DELTAEC** version

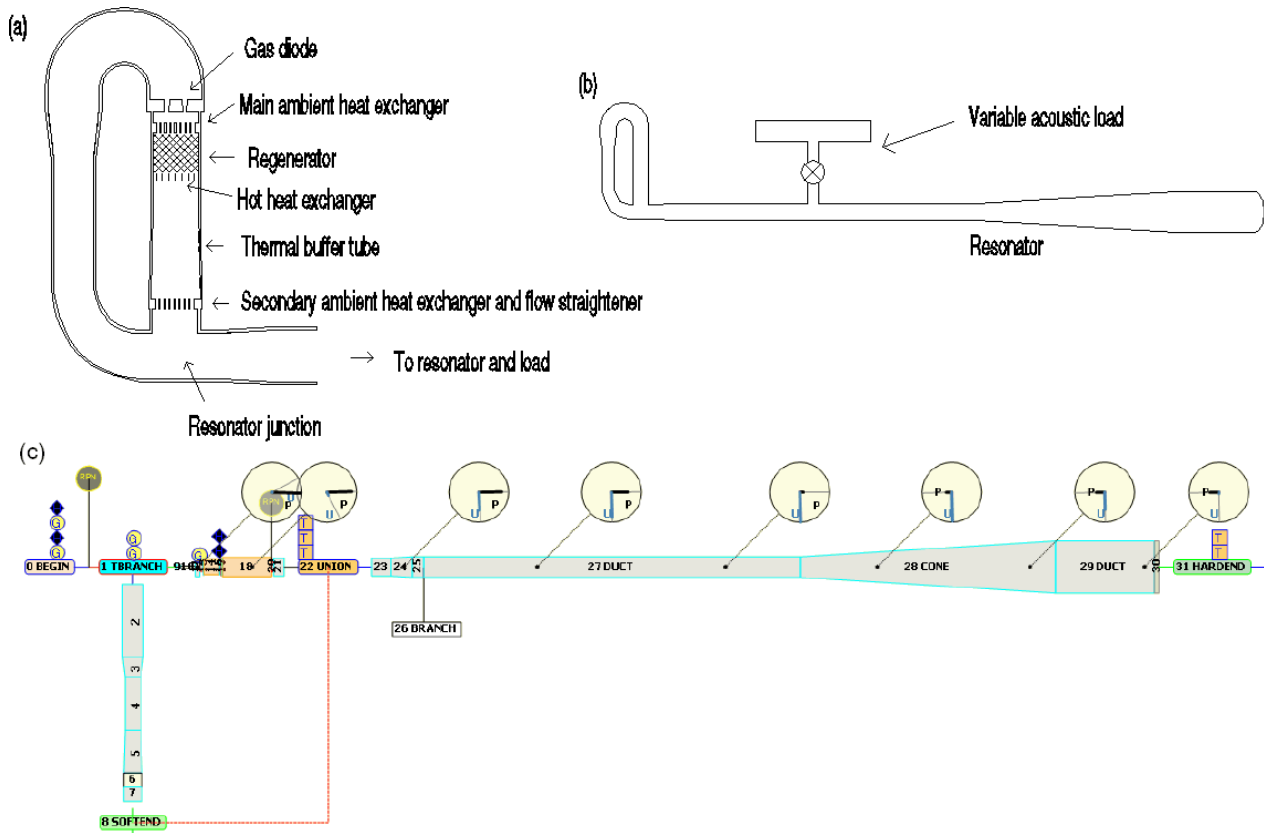


Figure 6.4: Illustrations of the TASHE. (a) Scale drawing of the toroidal part of the TASHE, containing the heat exchangers. (b) Overall scale drawing, with the toroidal part at the left end. (c) DELTAEC's Schematic view of the TASHE, with some phasor diagrams inserted to illustrate the wave.

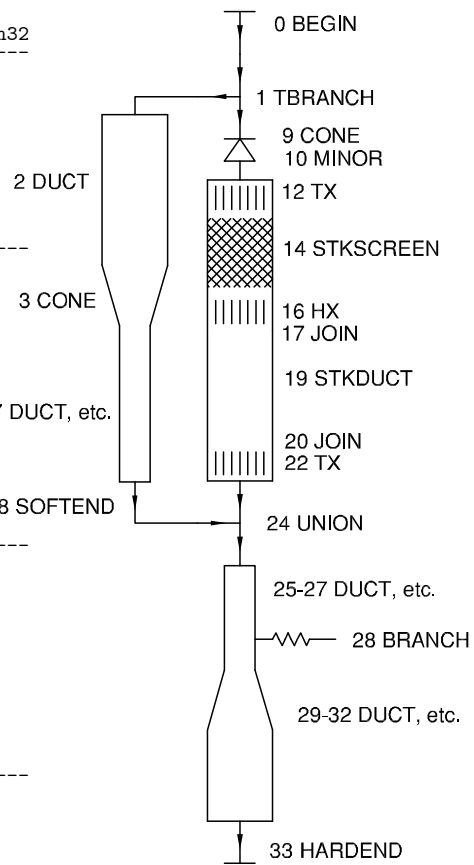
6. For selecting guesses and targets, we have adopted the following perspective: At what frequency, amplitude, and hot temperature will this engine run, for a given  $\dot{Q}_H$  applied at the hot heat exchanger? Ambient temperature and gas mean pressure are given, and hardware dimensions are fixed. In anticipation of adding more targets and guesses later, we make a special effort to minimize their numbers here. Thus, we do not use gas temperature 0c as a guess and ambient heat-exchanger solid temperature 12f as a target; we simply set 0c manually to a value that brings 12H near 25°C. We also avoid guessing heat-exchanger heat 20e and targeting  $\text{HARDEND } \dot{H}_{\text{tot}} = 0$  by using an RPN calculation at segment 19 to anticipate what the heat-exchanger heat must be based on global first-law considerations. With those simplifications, we still have 31 segments, 5 targets, and 5 guesses. This file is in the <Examples\SteadyFlow> folder, as <tashe1.out>.

!Created@20:43:11 01-Dec-2017 with DeltaEC version 6.4b2.7!under win32

```

-----
BEGIN      the setup
3.1030E+6  a Mean P Pa
85.815    b Freq Hz      G
325.00    c TBeg K
3.1675E+5 d |p| Pa      G
0.0000    e Ph(p) deg
0.0000    f |U| m^3/s
0.0000    g Ph(U) deg
helium     Gas type
-----
!
TBRANCH    Split up the flow
-1.8366E+7 a Re(Zb) Pa-s/m^3 G      3.1675E+5 A |p| Pa
1.4082E+7  b Im(Zb) Pa-s/m^3 G      0.0000 B Ph(p) deg
-1263.0    c HtotBr W                1.3686E-2 C |U| m^3/s
                                           -142.52 D Ph(U) deg
                                           -1263.0 E HtotBr W
                                           -1720.1 F EdotBr W
                                           1720.1 G EdotTr W
+ 2 DUCT    180 bend plus brass connecting flange (pg 27 book 4)
+ 3 CONE    4" to 3" Concentric reducer (pg 36 book 4)
+ 4 DUCT    3" FB Duct - Length given in concept.skf
+ 5 CONE    3.5" to 3" Long radius reducing elbow (pg 36 book 4)
+ 6 MINOR   minor loss, small end of elbow
+ 7 DUCT    FB connector/part of tee (Pg 55 book 4 concept.skf)
-----
!
SOFTEND    End of feedback branch
0.0000    a Re(z)                2.6323E+5 A |p| Pa
0.0000    b Im(z)                1.1006 B Ph(p) deg
0.0000    c Htot W                0.20533 C |U| m^3/s
                                           -92.847 D Ph(U) deg
                                           -1263.0 E Htot W
                                           -1860.7 F Edot W
                                           -0.11237 G Re(z)
                                           1.6282 H Im(z)
                                           325.00 I T K
+ 9 CONE    gas diode plate
-----
!
MINOR      minor loss here
sameas    9d a Area m^2          3.0830E+5 A |p| Pa
0.8000    b K+                  -1.9149 B Ph(p) deg
7.0000E-2 c K-                  1.3473E-2 C |U| m^3/s
                                           36.516 D Ph(U) deg
                                           1263.0 E Htot W
                                           1626.8 F Edot W
+ 11 DUCT   jetting space
-----
!
TX         Main room temp water HX (pg 90 book 3)
6.6580E-3 a Area m^2          3.0843E+5 A |p| Pa
0.2275    b GasA/A            -2.003 B Ph(p) deg
2.0400E-2 c Length m         1.0922E-2 C |U| m^3/s
1.2700E-3 d radius m         14.689 D Ph(U) deg
-1606.0   e HeatIn W          -342.95 E Htot W
300.00    f SolidT K          1613.3 F Edot W
stainless  Solid type          325.00 G GasT K
292.62    H SolidT K
+ 13 DUCT   Regen cold end dead space due to ribs (pg 91 book 3)
-----
!
STKSCREEN   Regenerator (pg 92 book 3) (Ks frac est:pg 20 book 4)
6.2070E-3 a Area m^2          2.6856E+5 A |p| Pa
0.7190    b VolPor            1.5666 B Ph(p) deg
7.3000E-2 c Length m         3.2898E-2 C |U| m^3/s
4.2200E-5 d rh m              -38.064 D Ph(U) deg
-----

```



```

0.3000 e ksFrac -342.95 E Htot W
3402.2 F Edot W
325.00 G TBeg K
stainless Solid type 857.45 H TEnd K
+ 15 DUCT All regen hot end dead space (pg 92 book 3)(area is avg)
!----- 16 -----
HX HHX (pg 93 book 4) heat xfer area used/not acoustic area
5.6970E-3 a Area m^2 2.6848E+5 A |p| Pa
0.9867 b GasA/A 1.5517 B Ph(p) deg
6.3500E-3 c Length m 3.4276E-2 C |U| m^3/s
7.9400E-4 d y0 m -41.297 D Ph(U) deg
3700.0 e HeatIn W 3357.0 E Htot W
0.0000 f SolidT K 3373.4 F Edot W
857.45 G GasT K
stainless Solid type 950.13 H SolidT K
+ 17 JOIN thermal buffer tube end effects
+ 18 DUCT hhx dead space (pg 94 book 3) stainless used for Qdot
+ 19 STKDUCT Straight section of pulse tube (pg 101 bk 4)
+ 20 JOIN thermal buffer tube end effects
!----- 21 -----
RPN how much heat must be rejected here? Resonator is insulated
0.0000 a G or T -2094.0 A W
16e 12e + ~
!----- 22 -----
TX Small water Xger
6.6580E-3 a Area m^2 2.6475E+5 A |p| Pa
0.2690 b GasA/A 1.2086 B Ph(p) deg
1.0160E-2 c Length m 7.4090E-2 C |U| m^3/s
2.2860E-3 d radius m -69.45 D Ph(U) deg
sameas 21A e HeatIn W 1263.0 E Htot W
290.00 f SolidT K 3248.2 F Edot W
325.00 G GasT K
195.94 H SolidT K
ideal Solid type
+ 23 DUCT PT connector (see pg 55 book 4 and concept.skf)
+ 24 UNION Rejoin
+ 25 DUCT Section of 3.5" tee
+ 26 CONE 3.5" to 4" cone to adapt to resonator (pg 93 book 4)
+ 27 DUCT Initial section of resonator
+ 28 BRANCH RC dissipator
+ 29 DUCT Continuation of resonator
+ 30 CONE 7 degree cone - 10.02" final diameter
+ 31 DUCT 10" duct P8
+ 32 SURFACE end cap of resonator
!----- 33 -----
HARDEND end of duct
0.0000 a R(1/z) =33G 9.4878E+4 A |p| Pa
0.0000 b I(1/z) =33H -178.63 B Ph(p) deg
0.0000 c Htot W 4.3937E-12 C |U| m^3/s
-88.08 D Ph(U) deg
0.0000 E Htot W
-1.9968E-9 F Edot W
-4.2520E-14 G R(1/z)
4.4380E-12 H I(1/z)

GUESS 0b 0d 1a 1b 12e
name BEGIN:Freq BEGIN: |p| TBRAN:Re(Zb) TBRAN:Im(Zb) TX:HeatI
units Hz Pa Pa-s/m^3 Pa-s/m^3 W
value 85.815 3.1675E+05 -1.8366E+07 1.4802E+07 -1606.0
TARGET 24b 24c 24d 33a 33b
name UNION:|p|SO UNION:Ph(p) UNION:T_SOF HARDE:R(1/z) HARDE:I(1/z)
units Pa deg K
value 2.6323E+05 1.1006 325.00 0.0000 0.0000
result 2.6323E+05 1.1006 325.00 -1.1441E-16 1.0380E-15

```

With 3700 W applied to the hot heat exchanger, the apparatus resonates at 85.8 Hz with an amplitude of 317 kPa at BEGIN. The hot gas temperature is 857 K and the hot solid temperature is 950 K.

One strange-looking number appears above: Segment 1, TBRANCH, input c,  $-1263$  W,  $H_{totBr}$ . DELTAEC arrives at the TBRANCH with some value for  $\dot{H}_{tot}$ . (It is zero in the present example, but not always.)  $\dot{H}_{tot,br}$  tells DELTAEC how much  $\dot{H}_{tot}$  to send into the branch; the balance goes into the trunk. Neither we nor DELTAEC may know the right value for this number in the present model, because the contribution of Rayleigh- and jet-driven streaming to heat convection in the part of the torus that connects the main ambient heat exchanger (segment 12) and the secondary ambient heat exchanger (segment 22) is unknown. Such streaming helps determine how much of the heat that comes from dissipation of  $\dot{E}$  in the load and resonator (assumed insulated here) shows up in each of those heat exchangers.

Fortunately, the performance of the apparatus is insensitive to  $\dot{H}_{\text{tot,br}}$ : If we change its value from  $-1263$  W to  $-1763$ , the resonance frequency, pressure amplitude, and hot temperature change by less than 0.1%, even though the heat rejected at the main ambient heat exchanger increases by 500 W and that in the secondary ambient heat exchanger decreases by the same amount. In the absence of any evidence to the contrary, it is sometimes reasonable to set

$$\dot{H}_{\text{tot,br}} = \dot{E}_{\text{br}} + \dot{N}_{\text{br}} m w_m, \quad (6.14)$$

and this is available as a check-box option in the TBRANCH segment. Another reasonable option here would be to add  $\dot{H}_{\text{tot,br}}$  to the guess list, using it to allow DELTAEC to maintain equal solid temperatures in the two ambient-temperature heat exchangers. Since we are trying to minimize the number of guesses and targets right now, we choose the first option.

Now we let DELTAEC begin to consider Gedeon streaming, by checking the “Enable Ndot” box in BEGIN. This enables nonzero  $\dot{N}$  and the calculation and display of  $p_{2,0,HL}$  throughout the model. For now, we leave  $\dot{N} = 0$  and examine the UNION at segment 24:

```

!----- 24 -----
UNION      Rejoin
  8.0000   a SegNum          2.6331E+05 A |p|      Pa
  2.6331E+05 b |p|Sft      Pa = 22A?      1.0991 B Ph(p)      deg
  1.0991   c Ph(p)S        deg = 22B?      0.2837 C |U|      m^3/s
  325.00   d TSoft         K = 22G?      -86.775 D Ph(U)      deg
  -0.6445  e p20HLS        Pa (t)          2.2737E-13 E Htot      W
                                           1385.9 F Edot        W
                                           325.00 G T            K
                                           -4038.2 H p20HL      Pa

```

“Allowing steady flow” has turned on the calculation of  $p_{2,0,HL}$ . Parameter **e** here shows that  $p_{2,0,HL} = -0.6$  Pa at SOFTEND segment 8, which is the end of the TBRANCH and is supposedly reconnected to the trunk here at segment 22. However, parameter **H** shows that  $p_{2,0,HL} = -4038$  Pa here in the trunk.

There is a mismatch of 4 kPa in second-order time-averaged pressure in this UNION segment. We have already targeted  $p_1$  and  $T_m$  in the UNION, to satisfy acoustic and thermodynamic boundary conditions; now we must add  $p_{2,0,HL}$  in the UNION to the target list. Our corresponding guess will be  $\dot{N}_{\text{br}}$  in segment 1: We ask DELTAEC how much Gedeon streaming will flow around the torus, and what effects that will have on operation of the apparatus.

We make a small increase in  $\dot{N}_{\text{br}}$  and watch segment 22, to make sure  $p_{2,0,HL}$  changes in the right direction, to make sure we understand what we’re doing. We note that DELTAEC gets lost if we change  $\dot{N}_{\text{br}}$  by 0.1 mol/s, so we make smaller changes of 0.01 mol/s until  $p_{2,0,HL}$  in segments 8 and 22 are nearly matched. Then we add **1d** and **22e** to the guess and target lists, respectively.

Making those changes and running, we notice that the state plot of  $T_m(x)$  is not smooth in the thermal buffer tube. We increase the number of steps in the numerical integration of STKDUCT (and the other integrated segments), by increasing Nint. The irregularities in  $T_m(x)$  go away above Nint = 30. Running with Nint = 40, the guess–target summary becomes

GUESS	0b	0d	1a	1b	1d	12e
name	BEGIN:Freq	BEGIN:  p	TBRAN:Re(Zb)	TBRAN:Im(Zb)	TBRAN:Ndotb	TX:HeatI
units	Hz	Pa	Pa-s/m <sup>3</sup>	Pa-s/m <sup>3</sup>	mol/s	W
value	86.157	1.9974E+05	-2.0031E+07	1.3845E+07	0.28991	-3950.0
TARGET	22b	22c	22d	22e	31a	31b

## tashe4.spl

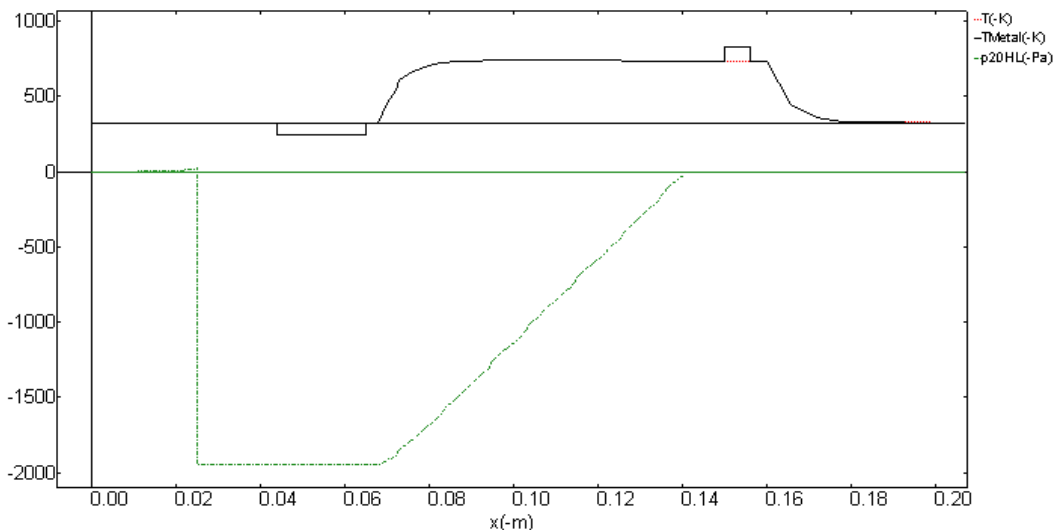


Figure 6.5: State plot of the TASHE example, with steady-flow calculations enabled but Gedeon streaming not properly eliminated. Solid black curve is  $T_{\text{solid}}$ , dashed red curve is  $T_m$ , and dot-dashed green curve is  $p_{2,0,HL}$ . The locations of the three heat exchangers are evident as short plateaus in  $T_{\text{solid}}$ .

```

name  UNION:|p|SO  UNION:Ph(p)  UNION:T_SOF  UNION:p20HL  HARDE:R(1/z  HARDE:I(1/z
units  Pa          deg         K           Pa          0.0000      0.0000
value  1.6592E+05  1.0458      325.00      -0.4932      0.0000      0.0000
result 1.6592E+05  1.0458      325.00      -0.4932     -1.2258E-13 -1.4742E-13

```

and the interesting characteristics of the engine become  $f = 86$  Hz,  $|p_1| = 200$  kPa,  $T_{m,H} = 743$  K, and  $T_{\text{solid},H} = 834$  K. This estimate of performance is probably much more realistic than the values of 317 kPa, 857 K, and 950 K we found earlier, because now we have included the Gedeon streaming, which is carrying away a lot of heat from the hot heat exchanger without contributing to any thermoacoustic power generation.

To decide what to do about the Gedeon streaming, we can either look at the sign of  $\dot{N}_{\text{br}}$  or examine temperatures and  $p_{2,0,HL}$  in the state plot shown in Fig. 6.5. The regenerator is running far on the hot side of a linear temperature profile, and  $\dot{N}$  is streaming through it from hot to cold. Apparently the gas diode is too tight, overcompensating for the tendency of the regenerator to stream in the direction of  $\vec{E}$ . No other segments contribute significantly to  $p_{2,0,HL}$ . Examination of the gas-diode segments,

```

!----- 9 -----
CONE      gas diode plate
5.8880E-04 a AreaI      m^2      2.0040E+05 A |p|      Pa
0.4210   b PerimI      m          -0.5402 B Ph(p)     deg
2.5399E-02 c Length    m          8.0839E-03 C |U|      m^3/s
1.1400E-04 d AreaF      m^2      33.617  D Ph(U)      deg
0.1850   e PerimF      m          1263.0   E Htot      W
5.0000E-04 f Strough    m          670.28  F Edot      W
ideal      Solid type
!----- 10 -----
MINOR     minor loss here
sameas    9d a Area      m^2      1.9709E+05 A |p|      Pa
0.8000    b K+          -1.1947 B Ph(p)     deg
7.0000E-02 c K-          8.0839E-03 C |U|      m^3/s
33.617    D Ph(U)      deg
1263.0    E Htot      W

```

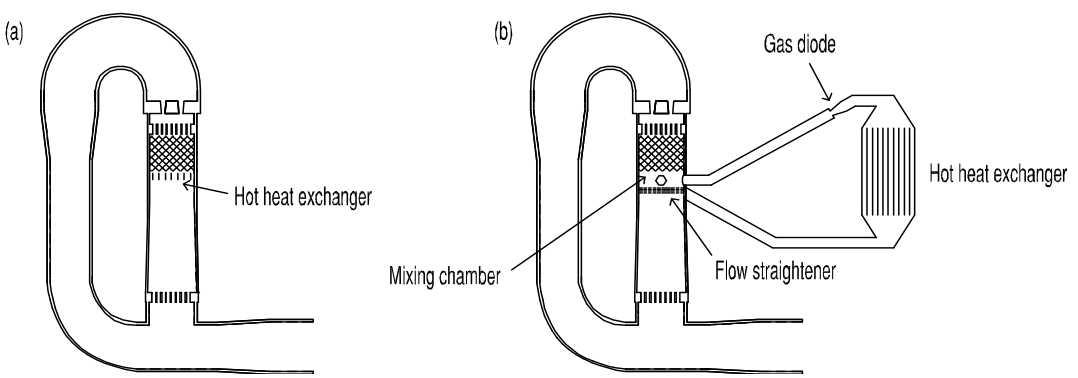


Figure 6.6: (a) The TASHE of the previous section, with a traditional hot heat exchanger between the regenerator and the thermal buffer tube. (b) The same TASHE, with the hot heat exchanger replaced by a mixing chamber, self-circulating loop, and flow straightener.

654.07 F Edot W

shows that they are qualitatively in order, with the minor-loss coefficient larger in the  $+x$  direction than in the  $-x$  direction, as necessary to fight Gedeon streaming. We surmise that the area at the small end of the gas diode,  $1.14 \text{ cm}^2$ , may be too small. We tentatively increase that area about 1% (leaving its perimeter constant, since this was an adjustable slit in the experimental hardware), and we are pleased to see that DELTAEC converges on a solution with reduced streaming and higher amplitude. We use the incremental plot to continue increasing that area, watching  $\dot{N}_{\text{br}}$  and stopping the sequence when it gets close to zero. We then clear  $\dot{N}_{\text{br}}$  from the guess list and set it equal to zero, and add the gas-diode's small-end area to the guess list. A run then yields `<tashe5.out>` in the `<Examples\SteadyFlow>` folder, which shows that an area of  $2.47 \text{ cm}^2$  stops Gedeon streaming, and with this area the apparatus runs with  $f = 86 \text{ Hz}$ ,  $|p_1| = 300 \text{ kPa}$ ,  $T_{m,H} = 731 \text{ K}$ , and  $T_{\text{solid},H} = 822 \text{ K}$ —better than before. A quick glance at the state plot shows near-linear temperature profiles in the stack and thermal buffer tube, even with `Nint` back down at its default value of 10.

## 6.4. Self-circulating heat exchanger

References [8] and [42] describe the use of a gas diode in a resonant loop to induce a large  $\dot{N}$  for deliberate heat transfer. This gives us the freedom to design heat exchangers that are not constrained by geometrical proximity to a regenerator or stack. In the present section, we will continue to explore the use of nonzero  $\dot{N}$  in DELTAEC by replacing the hot HX in the TASHE example of the previous section with a self-circulating hot heat-transfer loop. This replacement is illustrated in Fig. 6.6, which builds on Fig. 6.4. The heat-transfer loop here bears a qualitative resemblance to the system described in Ref. [8], but the quantitative details here are entirely made up and no details have been optimized or even thoughtfully chosen. The purpose of this section is only to illustrate the development of a topologically complicated DELTAEC model.

We began with the realization that this is going to be a complicated model, so initial simplification is important. To begin that simplification, we stripped `<tashe5.out>`, de-



scribed at the end of the previous section, down to only a few segments in the vicinity of the hot heat exchanger:

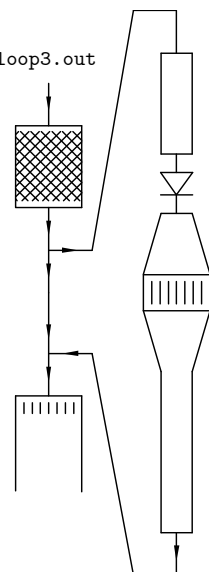
```
!----- 0 -----
BEGIN the setup
3.1030E+06 a Mean P Pa
85.895 b Freq Hz
325.00 c TBeg K
2.9837E+05 d |p| Pa
-0.9353 e Ph(p) deg
1.3300E-02 f |U| m^3/s
13.797 g Ph(U) deg
-316.07 h Htot W
0.0000 i Ndot mol/s
helium Gas type
+ 1 STKSCREEN Regenerator (pg 92 book 3) (Ks frac est:pg 20 book 4)
+ 2 DUCT All regen hot end dead space (pg 92 book 3)(area is avg)
+ 3 HX HHX (pg 93 book 4) heat xfer area used/not acoustic area
+ 4 DUCT hhx dead space (pg 94 book 3) stainless used for Qdot
+ 5 JOIN thermal buffer tube end effects
+ 6 STKDUCT Straight section of pulse tube (pg 101 bk 4)
```

We set the initial conditions in BEGIN to match the conditions of the segment just upstream of the STKSCREEN in <tashe5.out>, so the integration proceeds through the rest of the model as it did in <tashe5.out>, with no need for guesses and targets.

Next, we install a hot self-circulating loop instead of the hot heat exchanger. In place of the HX, we insert, as a side-branch loop, in sequence: a quarter wavelength of duct, a gas diode, an adapter cone, a VXQ1 heat exchanger, another adapter, and a quarter-wavelength return duct. We also insert a flow straightener above the thermal buffer tube. We keep guesses and targets at a minimum: those that are necessary at the UNION and TBRANCH, and one additional guess-target pair, guessing the input heat and targeting the thermal buffer tube's final temperature.

```
!->C:\Users\092710\Documents\DeltaEC\PySource\Examples\SteadyFlow - withJoin\hotloop3.out
!Created@08:04:05 02-Dec-2017 with DeltaEC version 6.4b2.7!under win32,
using Win 6.1.7601 (Service Pack 1) under Python 2.7.9.
```

```
!----- 0 -----
BEGIN the setup
3.1030E+6 a Mean P Pa
85.895 b Freq Hz
325.00 c TBeg K
2.9837E+5 d |p| Pa
-0.9353 e Ph(p) deg
1.3300E-2 f |U| m^3/s
13.797 g Ph(U) deg
-316.07 h Htot W
0.0000 i Ndot mol/s
helium Gas type
!----- 1 -----
STKSCREEN Regenerator (pg 92 book 3) (Ks frac est:pg 20 book 4)
6.2070E-3 a Area m^2 2.5369E+5 A |p| Pa
0.7190 b VolPor 1.9218 B Ph(p) deg
7.3000E-2 c Length m 3.2324E-2 C |U| m^3/s
4.2200E-5 d rh m -30.836 D Ph(U) deg
0.3000 e ksFrac -316.07 E Htot W
3448.0 F Edot W
325.00 G TBeg K
731.79 H TEnd K
-316.07 I H2k W
stainless Solid type
!----- 2 -----
DUCT top half of mixing chamber
6.2000E-3 a Area m^2 Mstr 2.5365E+5 A |p| Pa
0.2791 b Perim m 2a 1.9108 B Ph(p) deg
1.0000E-2 c Length m 3.3238E-2 C |U| m^3/s
-33.235 D Ph(U) deg
-316.07 E Htot W
3447.0 F Edot W
ideal Solid type
!----- 3 -----
TBRANCH branch into hot loop
5.8567E+7 a Re(Zb) Pa-s/m^3 G 2.5365E+5 A |p| Pa
3.9622E+7 b Im(Zb) Pa-s/m^3 G 1.9108 B Ph(p) deg
0.7468 d NdotBr mol/s G 3.5872E-3 C |U| m^3/s
-32.169 D Ph(U) deg
```



```

7079.6 E HtotBr W
376.82 F EdotBr W
3070.2 G EdotTr W
!----- 4 -----
DUCT quarter-wave duct from engine to diode
7.0000E-4 a Area m^2 Mstr 1.0477E+4 A |p| Pa
9.3748E-2 b Perim m 4a -163.77 B Ph(p) deg
4.6000 c Length m 5.5089E-2 C |U| m^3/s
5.0000E-4 d Srough -87.716 D Ph(U) deg
7079.6 E Htot W
ideal Solid type 69.557 F Edot W
!----- 5 -----
RPN calculate Borda-Carnot minor loss here
0.0000 a G or T 0.18367 A ChngMe
1 6a 4a / - sqrd
!----- 6 -----
MINOR asymmetric flow at gas diode
4.0000E-4 a Area m^2 1.0190E+4 A |p| Pa
5.0000E-2 b K+ -173.93 B Ph(p) deg
sameas 5A c K- 5.5089E-2 C |U| m^3/s
-87.716 D Ph(U) deg
7079.6 E Htot W
18.515 F Edot W
!----- 7 -----
CONE expansion to plenum
sameas 6a a AreaI m^2 Mstr 2.0026E+4 A |p| Pa
7.0899E-2 b PerimI m 7a -176.65 B Ph(p) deg
0.4000 c Length m 5.0311E-2 C |U| m^3/s
sameas 9a d AreaF m^2 Mstr -87.796 D Ph(U) deg
0.4342 e PerimF m 7d 7079.6 E Htot W
5.0000E-4 f Srough 10.101 F Edot W
ideal Solid type
!----- 8 -----
RPN watch that angle
5.0000 a G or T 16.125 A ChngMe
7e 7b - pi / 7c / atan
!----- 9 -----
VXQ1 plenty of room for heat transfer
1.5000E-2 a Area m^2 1.7465E+4 A |p| Pa
0.4000 b GasA/A -176.04 B Ph(p) deg
0.1000 c SolA/A 5.7564E-2 C |U| m^3/s
1.0000E-2 d rh m 92.120 D Ph(U) deg
1.0000E-2 e LenTS1 m 9990.3 E Htot W
6.0000 f Len P1 m -16.133 F Edot W
1.0000E-2 g LenTS2 m 734.64 G TSolBeg P1 K
2910.7 h HeatP1 W G 922.27 H TSolEnd P1 K
1.0000 i FracQN 376.82 I H2k W
stainless Solid type
!----- 10 -----
CONE adapt between heat exchanger and quarter-wave tube
sameas 7d a AreaI m^2 1.0775E+4 A |p| Pa
sameas 7e b PerimI m -174.31 B Ph(p) deg
sameas 7c c Length m 6.1978E-2 C |U| m^3/s
sameas 11a d AreaF m^2 92.238 D Ph(U) deg
sameas 11b e PerimF m 9990.3 E Htot W
5.0000E-4 f Srough -20.081 F Edot W
ideal Solid type
!----- 11 -----
DUCT quarter wave connection tube
sameas 4a a Area m^2 2.5365E+5 A |p| Pa
sameas 4b b Perim m 1.9108 B Ph(p) deg
5.0000 c Length m 6.1093E-3 C |U| m^3/s
sameas 4d d Srough 124.84 D Ph(U) deg
9990.3 E Htot W
ideal Solid type -421.19 F Edot W
!----- 12 -----
SOFTEND end of the branch; this will reconnect to trunk at UNION
0.0000 a Re(z) 2.5365E+5 A |p| Pa
0.0000 b Im(z) 1.9108 B Ph(p) deg
0.0000 c Htot W 6.1093E-3 C |U| m^3/s
124.84 D Ph(U) deg
9990.3 E Htot W
-421.19 F Edot W
-5.4502 G Re(z)
-8.4153 H Im(z)
919.30 I T K
1623.7 J p2OHL Pa
!----- 13 -----
UNION reconnect the "final" end of the loop here
12 a SegNum 2.5365E+5 A |p| Pa
sameas 12A b |p|Sft Pa =13A 1.9108 B Ph(p) deg
sameas 12B c Ph(p)S deg =13B 2.4086E-2 C |U| m^3/s
sameas 12I d TSoft K -27.96 D Ph(U) deg
sameas 12J e p2OHL Pa =13H 2594.7 E Htot W
2649.0 F Edot W

```

```

              731.79   G T   K
              1623.7   H p20HL Pa
!----- 14 -----
DUCT      bottom half of mixing chamber
sameas 2a a Area   m^2   2.5363E+5   A |p|   Pa
sameas 2b b Perim  m     1.9024   B Ph(p) deg
sameas 2c c Length m    2.4942E-2 C |U|   m^3/s
              -31.255   D Ph(U) deg
              2594.7   E Htot  W
              2648.0   F Edot  W
ideal      Solid type
!----- 15 -----
SX        flow straightener (no heat input)
sameas 14a a Area   m^2   2.5345E+5   A |p|   Pa
              0.6500 b VolPor 1.9185   B Ph(p) deg
5.0000E-3 c Length m    2.5100E-2 C |U|   m^3/s
5.0000E-4 d rh      m    -32.614   D Ph(U) deg
0.0000 e HeatIn  W     2594.7   E Htot  W
0.0000 f SolidT  K     2620.4   F Edot  W
              731.79   G GasT  K
              731.79   H SolidT K
ideal      Solid type
              2594.7   I H2k   W
!----- 16 -----
JOIN      thermal buffer tube end effects
              2.5345E+5   A |p|   Pa
              1.9185   B Ph(p) deg
              2.4527E-2 C |U|   m^3/s
              -32.614   D Ph(U) deg
              2594.7   E Htot  W
              2560.5   F Edot  W
              731.79   G TBeg  K
              731.28   H TEnd  K
!----- 17 -----
STKDUCT   Straight section of pulse tube (pg 101 bk 4)
7.0000E-3 a Area   m^2   2.5115E+5   A |p|   Pa
0.2963 b Perim  m     1.6538   B Ph(p) deg
0.2400 c Length m    6.1562E-2 C |U|   m^3/s
1.1600E-3 d WallA m^2   -69.113   D Ph(U) deg
              2594.7   E Htot  W
              2546.7   F Edot  W
              731.28   G TBeg  K
              300.00   H TEnd  K
stainless Solid type
              2594.7   I H2k   W
!----- 18 -----
RPN      temperature target
300.00 a =A?          300.00          A K
Tm
GUESS    3a          3b          3d          9h
name TBRAN:Re(Zb) TBRAN:Im(Zb) TBRAN:Ndotb VXQ1:HeatP
units Pa-s/m^3 Pa-s/m^3 mol/s W
value 5.8567E+07 3.9622E+07 0.7468 2910.7
TARGET 13b          13c          13e          18a
name UNION:|p|SO UNION:Ph(p) UNION:p20HL RPN:Targe
units Pa deg Pa
value 2.5365E+05 1.9108 1623.7 300.00
result 2.5365E+05 1.9108 1623.7 300.00

```

Several features that have not been previously encountered or discussed in the Users Guide are displayed in this file. Details about each of these features are given in Chapter 10.

Segment MINOR is employed here with unequal minor-loss coefficients, as appropriate for an asymmetrical geometry and as necessary to create a pumping effect to move  $\dot{N} \simeq 0.8$  mole/sec around the loop [42]. Figure 6.7 shows the state plot of  $p_{2,0,HL}$  around the loop. The rise near  $x = 4$  m is the abrupt effect of the gas diode's MINOR segment. The gradual decreases elsewhere are the pressure gradients in the DUCTs (and, much smaller, in the VXQ1), which the gas-diode pump is overcoming. The gradient  $dp_{2,0,HL}/dx$  varies along each DUCT, despite the constancy of the area and flow rate, because of the nonlinear interaction between  $U_1$  and  $\dot{N}$ , which increases  $dp_{2,0,HL}/dx$  where  $|U_1|$  is largest. The nature of that nonlinear interaction is controlled in part by the value of the roughness parameter in DUCT segments 4 and 11. As described in detail in Section 10.1.1, the interaction between turbulent steady flow and turbulent oscillating flow is also governed by which of the Optimistic, Average, or Pessimistic boxes are checked in a DUCT. The p20HL target in the UNION ensures that

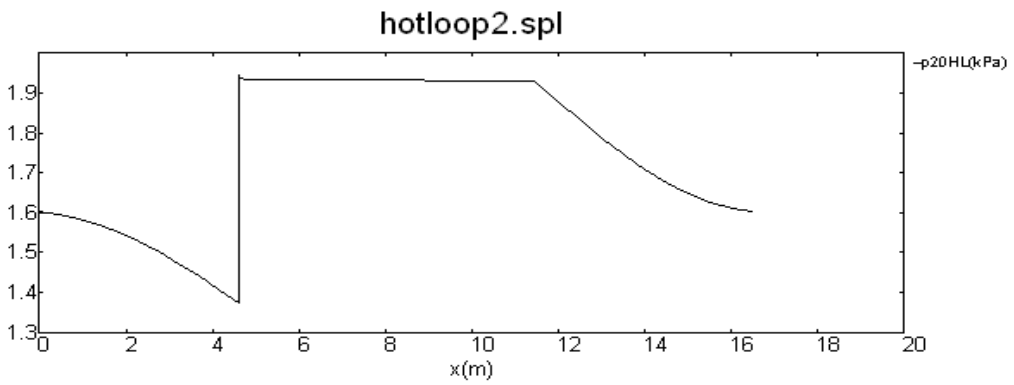


Figure 6.7: Second-order head-loss pressure vs  $x$  in the heat-exchange loop. The abrupt rise marks the location of the gas diode.

$p_{2,0,HL}(x = 16.5 \text{ m}) = p_{2,0,HL}(x = 0)$ . Recall that the zero of  $p_{2,0,HL}$  is set arbitrarily in the **BEGIN** segment, so the vertical offset of this graph is meaningless for the present discussion; it only indicates  $\Delta p_{2,0,HL}$  across the regenerator when  $\dot{N} = 0$  in the regenerator.

Segment 9, **VXQ1**, is one of the family of “variable” heat exchanger, **VX**, segments. The other types of heat exchanger segments—**HX**, **TX**, **SX**, and **PX**—are assumed to be short, to have  $x$ -independent solid temperatures, and to have constant  $dT_m/dx$ . The **VX\*\*** segments are numerically integrated with **Nint** steps, so they can be long. **VXT1** and **VXT2** are intended to model long heat exchangers with shell-side heat transfer such that the solid surfaces are held essentially isothermal. The gas temperature  $T_m(x)$  and heat transfer per unit length  $\dot{q}(x)$  are obtained numerically in the **VXT\*\*** segments. **VXQ1** and **VXQ2** are intended to model long heat exchangers with shell-side heat transfer such that the heat input per unit length is independent of  $x$ . The gas temperature  $T_m(x)$  and solid temperature  $T_{\text{solid}}(x)$  are obtained numerically in the **VXQ\*\*** segments. **VXQ2** and **VXT2** allow for two-part heat exchangers, such as heat exchangers with two passes on the shell side. Additional lengths for tubesheets, where no heat transfer occurs, are built into both ends of the **VX\*\*** segments. For more details, see Section 10.7.3. In the present example, Fig. 6.8 shows temperatures around the loop. Outside of the heat exchanger, gas and solid temperatures are equal, as usual. Inside the heat exchanger,  $T_m$  increases linearly with  $x$  because this is an ideal gas flowing through a **VXQ1** segment, with  $c_p$  and  $\dot{q}$  independent of  $x$ . The difference  $T_{\text{solid}} - T_m$  peaks in the center of the **VX**, where  $|U_1|$  is the smallest and hence turbulent heat transfer is the least effective at transferring heat from the solid to the gas.

The logistical segments in `<hotloop3.out>` also have a few peculiar features. In the **BEGIN**,  $\dot{H}_{\text{tot}}$  is set to the value that it had just upstream of the regenerator in `<tashe5.out>`, and  $\dot{N}$  is enabled and set to zero, to turn on  $p_{2,0,HL}$  and steady-flow calculations in the rest of the model.  $\dot{N}$  is, of course, zero or negligible in the regenerator, since `<tashe5.out>` properly suppressed Gedeon streaming with its gas diode. However,  $\dot{N}_{\text{br}}$  in the **TBRANCH** is large. **DELTAEC** adjusts this guess to hit the  $p_{2,0,HL}$  target at the **UNION**. We don’t know exactly what  $\dot{H}_{\text{tot,br}}$  should be, because we don’t know exactly how much heat might be convected into the loop by  $\dot{N}$  here, but the measurements reported in Ref. [8] indicate that any such convected heat is much smaller than what is convected *out* of such a loop at the

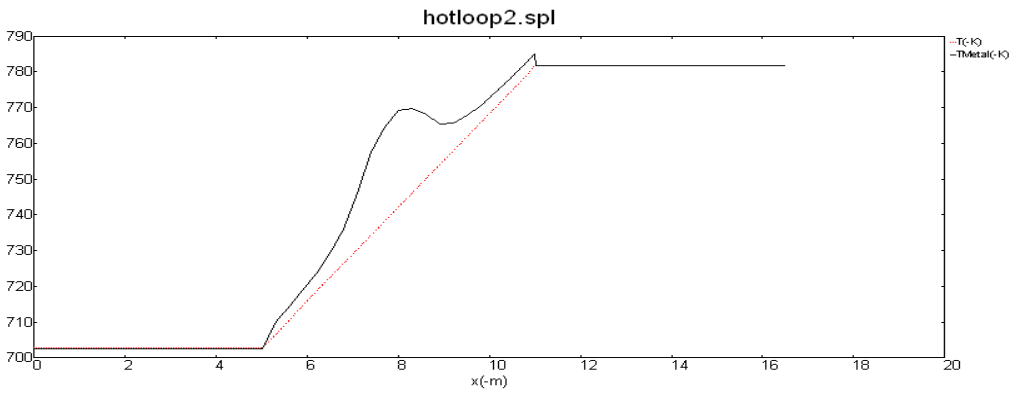


Figure 6.8: Temperatures around the hot loop. Solid black line, solid temperature. Dashed red line, gas temperature. The velocity-dependent gas–solid heat-transfer coefficient in `VXQ1` is evident in the difference between the two curves. (`Nint` = 20 for this plot.)

other end, so we simply set  $\dot{H}_{\text{tot,br}} = \dot{E}_{\text{br}}$  in the `TBRANCH`. At the `UNION`, we do not use the temperature target, because this would force the gas temperature at both ends of the loop to be equal, which would contradict the entire purpose of the circulating loop. Instead, we essentially let the 919-K gas coming out of the `SOFTEND` of the loop pour into the `UNION` where  $T_m = 732$  K. We rely on the energy-conserving feature of `DELTAEC`'s `UNION` to add the heat picked up in the loop into the  $H_{\text{tot}}$  leaving the `UNION`.

Next, as shown in Fig. 6.9, we combine `<hotloop3.out>` and `<tashe5.out>`, to make a complete `DELTAEC` model of the `TASHE`, the hot self-circulating heat-exchange loop, and the resonator. We run `<tashe5.out>`, `Kill` its traditional hot heat exchanger, and then copy and paste the appropriate segments from `<hotloop3.out>`. We fix a few things that `DELTAEC` did not do correctly. A `run` fails to converge, so we `Restore guessez` and then `clear` all guesses and targets, intending to introduce them one pair at a time. With no guesses and targets, a `run` shows mostly sensible results, except at the `HARDEND` at the end of the model, where energy is flowing out but acoustic power is flowing in, in the  $-x$  direction. We introduce guesses and targets beginning with the four each that were involved in the hot loop, since they are closest to being on-target already. Then we add three of the four `UNION` targets that were used in `<tashe5.out>` (skipping the temperature target for the moment), and three corresponding guesses, two  $Z_{\text{br}}$  in the segment-1 `TBRANCH` and the third being the slot width in the gas diode in segment 9. At this point, we have 7 targets and 7 guesses, and each change we make is requiring small, gentle steps. We remove the temporary `RPN` target we had been using in `<hotloop3.out>` to keep the temperature near ambient at the end of `<hotloop3.out>`, adding instead the segment-35 `UNION` temperature target. We add frequency as a guess, and  $\text{Im}(1/z_n)$  at the `HARDEND` as a target. Pause to think. In `<tashe5.out>`, our perspective was to give the engine a known hot heat input, and observe performance. Here, we are currently guessing the hot heat input, because that is what we did in `<hotloop3.out>`. Clear that guess, and try guessing the main ambient heat instead. With that change, `DELTAEC` still tolerates changes to variables reasonably well, given the fact that we have 8 targets and 8 guesses. In `<tashe5.out>`, we injected 3700 W of hot

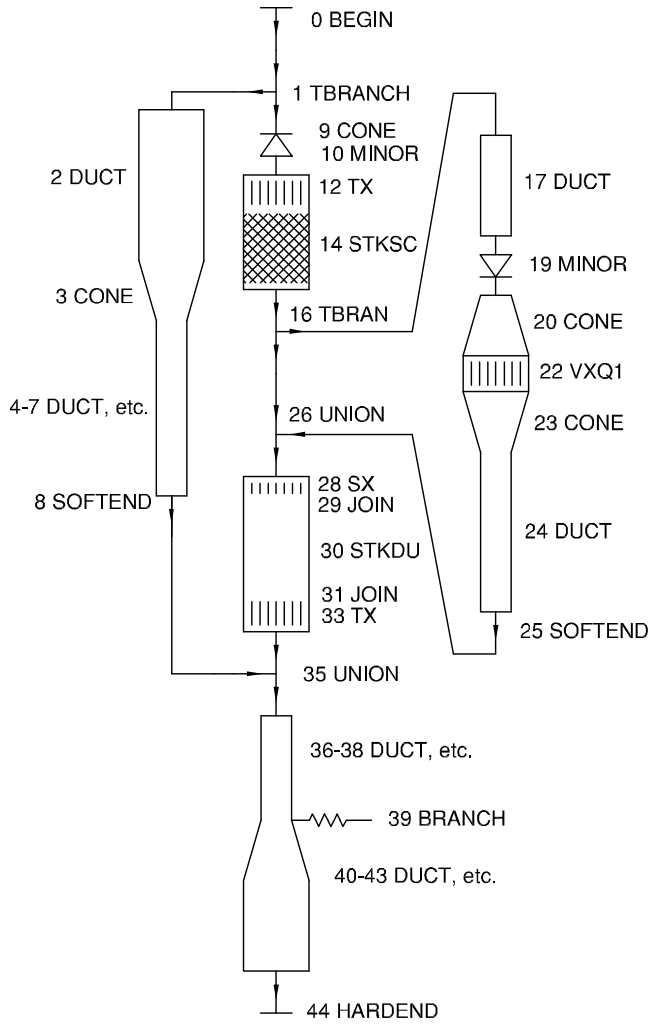


Figure 6.9: Sketch of the `<combo.out>` model. The TASHE feedback loop hangs to the left of the trunk, and the hot self-circulating heat exchanger loop hangs to the right.

heat; let's increase the heat in our hot loop to the same value here, for a fair comparison to the earlier model. We begin incremental plotting to higher hot heat, in 5-W steps which are small enough that DELTAEC stays converged at each step. Along the way, we notice that the  $\dot{E}$  flowing in at the as-yet-incompletely-targeted HARDEND is steadily decreasing, approaching zero at about 3500 W. We stop there, and add the HARDEND's  $\text{Re}[1/z_n]$  to the target list and the BEGIN's  $|p_1|$  to the guess list, and resume our increase of  $\dot{Q}_H$  up to 3700.

The resulting file, <combo2.out>, appears below. Whereas <tashe5.out> ran at an amplitude (in segment 0) of 300 kPa, <combo2.out> runs at 311 kPa. However, <tashe5.out> required only  $T_{\text{solid},H} = 822$  K; <combo2.out> requires solid temperatures from 1197 K to 1286 K.

This example may not be practical, but it does demonstrate the creation of a DELTAEC model for a topologically complicated apparatus with 9 targets and 9 guesses.

```
GUESS      0b      0d      1a      1b      9d      12e
name BEGIN:Freq BEGIN:|p| TBRAN:Re(Zb) TBRAN:Im(Zb) CONE:AreaF TX:HeatI
units Hz Pa Pa-s/m^3 Pa-s/m^3 m^2 W
value 89.111 3.1114E+05 -1.4792E+07 1.4654E+07 2.0008E-04 -2153.4
GUESS      16a      16b      16d
name TBRAN:Re(Zb) TBRAN:Im(Zb) TBRAN:Ndotb
units Pa-s/m^3 Pa-s/m^3 mol/s
value 1.9797E+06 9.5090E+06 2.0228
TARGET      26b      26c      35b      35c      35d
name UNION:|p|SO UNION:Ph(p) UNION:p20hl UNION:|p|SO UNION:Ph(p) UNION:T_SOF
units Pa deg Pa Pa deg K
value 2.5486E+05 1.6159 -0.72692 2.5372E+05 1.1616 325.00
result 2.5486E+05 1.6159 -0.72692 2.5372E+05 1.1616 325.00
TARGET      35e      44a      44b
name UNION:p20hl HARDE:R(1/z) HARDE:I(1/z)
units Pa
value -0.48008 0.0000 0.0000
result -0.48008 -1.3244E-14 3.5147E-12
```

```
!->C:\Users\092710\Documents\DeltaEC\PySource\Examples\SteadyFlow - withJoin\combo2.out
!Created@08:42:24 02-Dec-2017 with DeltaEC version 6.4b2.7!under win32,
using Win 6.1.7601 (Service Pack 1) under Python 2.7.9.
```

```
!----- 0 -----
BEGIN the setup
3.1030E+6 a Mean P Pa
89.111 b Freq Hz G
325.00 c TBeg K
3.1114E+5 d |p| Pa G
0.0000 e Ph(p) deg
0.0000 f |U| m^3/s
0.0000 g Ph(U) deg
0.0000 i Ndot mol/s
helium Gas type
!----- 1 -----
TBRANCH Split up the flow
-1.4792E+7 a Re(Zb) Pa-s/m^3 G 3.1114E+5 A |p| Pa
1.4654E+7 b Im(Zb) Pa-s/m^3 G 0.0000 B Ph(p) deg
0.0000 d NdotBr mol/s 1.4944E-2 C |U| m^3/s
-135.27 D Ph(U) deg
-1651.6 E HtotBr W
-1651.6 F EdotBr W
1651.6 G EdotTr W
+ 2 DUCT 180 bend plus brass connecting flange (pg 27 book 4)
+ 3 CONE 4" to 3" Concentric reducer (pg 36 book 4)
+ 4 DUCT 3" FB Duct - Length given in concept.skf
+ 5 CONE 3.5" to 3" Long radius reducing elbow (pg 36 book 4)
+ 6 MINOR minor loss, small end of elbow
+ 7 DUCT FB connector/part of tee (Pg 55 book 4 concept.skf)
!----- 8 -----
SOFTEND End of feedback branch
0.0000 a Re(z) 2.5372E+5 A |p| Pa
0.0000 b Im(z) 1.1616 B Ph(p) deg
0.0000 c Htot W 0.21036 C |U| m^3/s
-92.693 D Ph(U) deg
-1651.6 E Htot W
-1794.1 F Edot W
-0.10324 G Re(z)
1.5321 H Im(z)
325.00 I T K
-0.48008 J p20HL Pa
!----- 9 -----
CONE gas diode plate
```

5.8880E-4	a	AreaI	m <sup>2</sup>	3.1270E+5	A	p	Pa
0.4210	b	PerimI	m	-0.42704	B	Ph(p)	deg
2.5399E-2	c	Length	m	1.4669E-2	C	U	m <sup>3</sup> /s
2.0008E-4	d	AreaF	m <sup>2</sup>	43.826	D	Ph(U)	deg
0.1850	e	PerimF	m	1651.6	E	Htot	W
5.0000E-4	f	Srough		1642.7	F	Edot	W
ideal Solid type							
!----- 10 -----							
MINOR minor loss here							
sameas	9d	a	Area m <sup>2</sup>	3.0945E+5	A	p	Pa
0.8000	b	K+		-1.0163	B	Ph(p)	deg
7.0000E-2	c	K-		1.4669E-2	C	U	m <sup>3</sup> /s
				43.826	D	Ph(U)	deg
				1651.6	E	Htot	W
				1609.3	F	Edot	W
!----- 11 -----							
DUCT jetting space							
6.8250E-3	a	Area	m <sup>2</sup>	3.0951E+5	A	p	Pa
0.2930	b	Perim	m	-1.0302	B	Ph(p)	deg
1.9000E-2	c	Length	m	1.1993E-2	C	U	m <sup>3</sup> /s
5.0000E-4	d	Srough		28.944	D	Ph(U)	deg
				1651.6	E	Htot	W
				1607.8	F	Edot	W
!----- 12 -----							
TX Main room temp water HX (pg 90 book 3)							
6.6580E-3	a	Area	m <sup>2</sup>	3.0968E+5	A	p	Pa
0.2275	b	GasA/A		-1.1073	B	Ph(p)	deg
2.0400E-2	c	Length	m	1.1403E-2	C	U	m <sup>3</sup> /s
1.2700E-3	d	radius	m	24.265	D	Ph(U)	deg
-2153.4	e	HeatIn	W	-501.81	E	Htot	W
300.00	f	SolidT	K	1595.4	F	Edot	W
				325.00	G	GasT	K
				282.39	H	SolidT	K
				-501.81	I	H2k	W
stainless Solid type							
!----- 13 -----							
DUCT Regen cold end dead space due to ribs (pg 91 book 3)							
4.9700E-3	a	Area	m <sup>2</sup>	3.0968E+5	A	p	Pa
0.7400	b	Perim	m	-1.1104	B	Ph(p)	deg
3.1750E-3	c	Length	m	1.1182E-2	C	U	m <sup>3</sup> /s
				21.805	D	Ph(U)	deg
				-501.81	E	Htot	W
				1594.7	F	Edot	W
stainless Solid type							
!----- 14 -----							
STKSCREEN Regenerator (pg 92 book 3) (Ks frac est:pg 20 book 4)							
6.2070E-3	a	Area	m <sup>2</sup>	2.5488E+5	A	p	Pa
0.7190	b	VolPor		1.6250	B	Ph(p)	deg
7.3000E-2	c	Length	m	4.0190E-2	C	U	m <sup>3</sup> /s
4.2200E-5	d	rh	m	-25.452	D	Ph(U)	deg
0.3000	e	ksFrac		-501.81	E	Htot	W
				4560.5	F	Edot	W
				325.00	G	TBeg	K
				1192.4	H	TEnd	K
				-501.81	I	H2k	W
stainless Solid type							
!----- 15 -----							
DUCT top half of mixing chamber							
6.2000E-3	a	Area	m <sup>2</sup>	2.5486E+5	A	p	Pa
0.2791	b	Perim	m	1.6159	B	Ph(p)	deg
1.0000E-2	c	Length	m	4.0993E-2	C	U	m <sup>3</sup> /s
6.0000E-4	d	Srough		-27.604	D	Ph(U)	deg
				-501.81	E	Htot	W
				4559.0	F	Edot	W
stainless Solid type							
!----- 16 -----							
TBRANCH branch into hot loop							
1.9797E+6	a	Re(Zb)	Pa-s/m <sup>3</sup>	2.5486E+5	A	p	Pa
9.5090E+6	b	Im(Zb)	Pa-s/m <sup>3</sup>	1.6159	B	Ph(p)	deg
606.94	c	HtotBr	W	2.6239E-2	C	U	m <sup>3</sup> /s
2.0228	d	ndotBr	mol/s	-76.623	D	Ph(U)	deg
				606.94	E	HtotBr	W
				681.51	F	EdotBr	W
				3877.4	G	EdotTr	W
+ 17 DUCT quarter-wave duct from engine to diode							
+ 18 RPN calculate Borda-Carnot minor loss here							
+ 19 MINOR asymmetric flow at gas diode							
+ 20 CONE expansion to plenum							
+ 21 RPN watch that angle							
!----- 22 -----							
VXQ1 plenty of room for heat transfer							
1.5000E-2	a	Area	m <sup>2</sup>	2.6634E+4	A	p	Pa
0.4000	b	GasA/A		-174.71	B	Ph(p)	deg
0.1000	c	SoIA/A		6.9480E-2	C	U	m <sup>3</sup> /s
1.0000E-2	d	rh	m	92.426	D	Ph(U)	deg
1.0000E-2	e	LenTS1	m	4306.9	E	Htot	W
6.0000	f	LenP1	m	-46.285	F	Edot	W
1.0000E-2	g	LenTS2	m	1197.0	G	TSolBeg	P1 K
3700.0	h	HeatP1	W	1285.5	H	TSolEnd	P1 K



```

1.0000 i FracQN -3.6915E+4 I H2k W
stainless Solid type
+ 23 CONE adapt between heat exchanger and quarter-wave tube
+ 24 DUCT quarter wave connection tube
!----- 25 -----
SOFTEND end of the branch; this will reconnect to trunk at UNION
0.0000 a Re(z) 2.5486E+5 A |p| Pa
0.0000 b Im(z) 1.6159 B Ph(p) deg
0.0000 c Htot W 2.3017E-2 C |U| m^3/s
105.05 D Ph(U) deg
4306.9 E Htot W
-681.45 F Edot W
-0.73312 G Re(z)
-3.0692 H Im(z)
1280.4 I T K
-0.72692 J p20HL Pa
!----- 26 -----
UNION reconnect the "final" end of the loop here
25 a SegNum 2.5486E+5 A |p| Pa
sameas 25A b |p|Sft Pa =26A 1.6159 B Ph(p) deg
sameas 25B c Ph(p)S deg =26B 3.7639E-2 C |U| m^3/s
sameas 25I d TSoft K 49.830 D Ph(U) deg
sameas 25J e p20HLS Pa =26H 3198.2 E Htot W
3196.0 F Edot W
1192.4 G T K
-0.72692 H p20HL Pa
!----- 27 -----
DUCT bottom half of mixing chamber
sameas 15a a Area m^2 2.5489E+5 A |p| Pa
sameas 15b b Perim m 1.6094 B Ph(p) deg
sameas 15c c Length m 3.6364E-2 C |U| m^3/s
48.035 D Ph(U) deg
3198.2 E Htot W
3194.4 F Edot W
ideal Solid type
!----- 28 -----
SX flow straightener (no heat input)
sameas 27a a Area m^2 2.5474E+5 A |p| Pa
0.6500 b VolPor 1.5619 B Ph(p) deg
5.0000E-3 c Length m 3.5627E-2 C |U| m^3/s
5.0000E-4 d rh m 47.337 D Ph(U) deg
0.0000 e HeatIn W 3198.2 E Htot W
0.0000 f SolidT K 3165.1 F Edot W
1192.4 G GasT K
1192.4 H SolidT K
3198.2 I H2k W
ideal Solid type
!----- 29 -----
JOIN thermal buffer tube end effects
2.5474E+5 A |p| Pa
1.5619 B Ph(p) deg
3.4935E-2 C |U| m^3/s
47.337 D Ph(U) deg
3198.2 E Htot W
3103.5 F Edot W
1192.4 G TBeg K
1200.5 H TEnd K
!----- 30 -----
STKDUCT Straight section of pulse tube (pg 101 bk 4)
7.0000E-3 a Area m^2 Mstr 2.5461E+5 A |p| Pa
0.2963 b Perim m 30a 1.3516 B Ph(p) deg
0.2400 c Length m 3.2156E-2 C |U| m^3/s
1.1600E-3 d WallA m^2 -40.217 D Ph(U) deg
3198.2 E Htot W
3062.7 F Edot W
1200.5 G TBeg K
322.59 H TEnd K
3198.2 I H2k W
stainless Solid type
!----- 31 -----
JOIN thermal buffer tube end effects
2.5461E+5 A |p| Pa
1.3516 B Ph(p) deg
3.1485E-2 C |U| m^3/s
-40.217 D Ph(U) deg
3198.2 E Htot W
2998.9 F Edot W
322.59 G TBeg K
325.00 H TEnd K
!----- 32 -----
RPN how much heat must be rejected here? Resonator is insulated
0.0000 a G or T -1546.6 A ChngMe
22h 12e + ~
!----- 33 -----
TX Small water Xger
6.6580E-3 a Area m^2 2.5427E+5 A |p| Pa
0.2690 b GasA/A 1.2740 B Ph(p) deg
1.0160E-2 c Length m 3.1819E-2 C |U| m^3/s
2.2860E-3 d radius m -40.949 D Ph(U) deg

```

sameas	32A	e	HeatIn	W	1651.6	E	Htot	W
	290.00	f	SolidT	K	2995.7	F	Edot	W
					325.00	G	GasT	K
					231.46	H	SolidT	K
ideal			Solid type		1651.6	I	H2k	W
!----- 34 -----								
DUCT	PT	connector	(see pg 55	book 4 and concept.skf)				
6.2070E-3	a	Area	m^2		2.5372E+5	A	p	Pa
0.2790	b	Perim	m		1.1616	B	Ph(p)	deg
5.1000E-2	c	Length	m		3.8215E-2	C	U	m^3/s
6.0000E-4	d	Strough			-50.713	D	Ph(U)	deg
					1651.6	E	Htot	W
stainless			Solid type		2993.1	F	Edot	W
!----- 35 -----								
UNION			Rejoin					
8	a	SegNum			2.5372E+5	A	p	Pa
sameas	8A	b	p Sft Pa	=35A	1.1616	B	Ph(p)	deg
sameas	8B	c	Ph(p)S deg	=35B	0.24013	C	U	m^3/s
sameas	8I	d	TSoft K	=35G	-86.583	D	Ph(U)	deg
sameas	8J	e	p20HLS Pa	=35H	2.3874E-11	E	Htot	W
					1198.9	F	Edot	W
					325.00	G	T	K
					-0.48008	H	p20HL	Pa
+ 36	DUCT		Section of 3.5" tee					
+ 37	CONE		3.5" to 4" cone to adapt to resonator	(pg 93 book 4)				
+ 38	DUCT		Initial section of resonator					
+ 39	BRANCH		RC dissipator					
+ 40	DUCT		Continuation of resonator					
+ 41	CONE		7 degree cone - 10.02" final diameter					
+ 42	DUCT		10" duct	P8				
+ 43	SURFACE		end cap of resonator					
!----- 44 -----								
HARDEND			end of duct					
0.0000	a	R(1/z)		=44G	8.5902E+4	A	p	Pa
0.0000	b	I(1/z)		=44H	-178.64	B	Ph(p)	deg
0.0000	c	Htot	W		3.5950E-13	C	U	m^3/s
0.0000	d	Ndot	mol/s		91.562	D	Ph(U)	deg
					2.3874E-11	E	Htot	W
					5.5388E-11	F	Edot	W
					1.4388E-15	G	R(1/z)	
					-4.0109E-13	H	I(1/z)	
					0.0000	I	Ndot	mol/s

# 7. MIXTURE SEPARATION

A sound wave in a binary gas mixture in a channel causes time-averaged molar fluxes of the two components of the mixture in opposite directions, and can create a significant mole-fraction gradient along the channel. This occurs because oscillating lateral thermal diffusion causes the light and heavy components of the mixture to take turns enriching the central region of the channel, where the oscillating motion is the greatest; the time phasing is best when the oscillating enrichment and oscillating velocity are in phase, causing the velocity oscillations to carry light-enriched gas in one direction and heavy-enriched gas in the other direction [43, 44, 45]. The light component usually flows in the direction of  $\vec{E}$ .

The thermoacoustic mixture-separation mole flux and mole-fraction gradient are in some ways analogous to the time-averaged enthalpy flux and temperature gradient that a sound wave creates in a stack in a standing-wave thermoacoustic refrigerator.

DELTAEC can model this effect accurately [10].

## 7.1. Principles of calculation

Consider a binary gas mixture flowing steadily in the  $x$  direction through a channel, with a molar flow rate  $\dot{N}$ . Let the mole fraction of the lighter component be  $n_L$  and let its molar flow rate be  $\dot{N}_L$ ; similarly  $n_H$  and  $\dot{N}_H$  for the heavy component. It is always true that

$$n_L + n_H = 1 \tag{7.1}$$

and

$$\dot{N}_L + \dot{N}_H = \dot{N}. \tag{7.2}$$

In a simple “bulk” flow of the gas mixture (i.e., flow that does not involve mass diffusion, mole-fraction gradients, or thermoacoustic mixture separation), it is also true that

$$\dot{N}_L = n_L \dot{N}. \tag{7.3}$$

Combining Eqs. (7.1)–(7.3) algebraically shows that

$$\dot{N}_H = n_H \dot{N} \tag{7.4}$$

is also true in a bulk flow, as expected from the symmetry of the equations. Thus, bulk flow in a binary mixture can be uniquely described by assigning values to just *two* variables, which can be chosen as  $\dot{N}$  and  $n_L$ .

Nontrivial flows arise in binary mixtures whenever mass diffusion or thermoacoustic mixture separation occurs. For example, for  $\dot{N} = 0$  and no thermoacoustics, a nonzero  $\dot{N}_L$

and an equal and opposite  $\dot{N}_H$  arise from ordinary diffusion in the  $x$  direction whenever a mole-fraction gradient  $dn_L/dx \equiv -dn_H/dx$  exists; Eq. (7.3) is obviously not satisfied. Thus, DELTAEC must routinely handle nontrivial mixture flows, which must be specified by three independent variables. DELTAEC takes those three variables to be  $\dot{N}$ ,  $\dot{N}_L$ , and  $n_L$ .<sup>1</sup>

For  $dT_m/dx = 0$ , DELTAEC allows the mole fraction to evolve with  $x$  in segment types MIXBL (boundary-layer approximation), MIXSLAB (between parallel plates), and MIXCIRC (circular channel), according to the theory described in Ref. [10] and references therein. Nonzero  $dn_L/dx$  is supported for three of DELTAEC’s gas mixtures: He-Ar, He-Xe, and Ne-Xe. The user can create an external file to specify the properties of other gas mixtures, as described in Section 12.15.2. The equations being integrated in these MIX\*\* segments are

$$\frac{dp_1}{dx} = \mathcal{F}_{\text{momentum}}(p_1, U_1, T_m; p_m, \omega; \text{geometry, gas properties, etc.}), \quad (7.5)$$

$$\frac{dU_1}{dx} = \mathcal{F}_{\text{continuity}}(p_1, U_1, T_m; p_m, \omega; \text{geometry, gas properties, etc.}), \quad (7.6)$$

$$\frac{dT_m}{dx} = 0, \quad (7.7)$$

$$\frac{dn_L}{dx} = \mathcal{F}_{\text{mix sep}}(p_1, U_1, T_m, \dot{N}, \dot{N}_L; p_m, \omega; \text{geometry, gas properties, etc.}). \quad (7.8)$$

The similar segment types MIXTBL, MIXTSLAB, and MIXTCIRC allow  $dT_m/dx \neq 0$ , so the equations being integrated in these segments are

$$\frac{dp_1}{dx} = \mathcal{F}_{\text{momentum}}(p_1, U_1, T_m; p_m, \omega; \text{geometry, gas properties, etc.}), \quad (7.9)$$

$$\frac{dU_1}{dx} = \mathcal{F}_{\text{continuity}}(p_1, U_1, T_m; p_m, \omega; \text{geometry, gas properties, etc.}), \quad (7.10)$$

$$\frac{dT_m}{dx} = \mathcal{F}_{\text{energy}}(p_1, U_1, T_m; p_m, \omega; \text{geometry, gas properties, etc.}), \quad (7.11)$$

$$\frac{dn_L}{dx} = \mathcal{F}_{\text{mix sep}}(p_1, U_1, T_m, \dot{N}, \dot{N}_L; p_m, \omega; \text{geometry, gas properties, etc.}). \quad (7.12)$$

In DELTAEC’s BEGIN segment, we specify  $\dot{N}$  and  $n_L$ . If the “bulk NLdot” box is checked, DELTAEC calculates  $\dot{N}_L$  using Eq. (7.3); if the bulk NLdot box is not checked, the user must specify  $\dot{N}_L$  (perhaps as a guess).

Continuity of  $n_L$  between segments is enforced.  $\dot{N}$  and  $\dot{N}_L$  split up at TBRANCHs, add at UNIONs, and are independent of  $x$  in all physical segments.

## 7.2. Boundary conditions

Deciding how to handle  $\dot{N}$ ,  $\dot{N}_L$ , and  $n_L$  at BEGINS, SOFTENDs, etc. is subtle and depends on exactly what the user wants to model. A feedstock with a given  $n_L$  might be injected into

---

<sup>1</sup>DELTAEC focuses on the light component because DELTAEC’s first mixture, He–Ar, was set up many years ago with its input parameter representing the helium fraction. This convention causes a minor bookkeeping nuisance now, since Refs. [10, 43, 44, 45] focus on the heavy component.

the middle or end of a mixture separator whose vigorous thermoacoustic processes maintain a locally different  $n_L$ ; in this case,  $n_L$  can be regarded as discontinuous [10], and Eq. (7.3) would not apply if the local thermoacoustic channel's  $n_L$  were used, even while the feedstock flow itself is bulk [i.e., the feedstock mole fraction and feedstock flows satisfy Eq. (7.3)]. In other circumstances, one location of the mixture separator might be firmly connected to an essentially infinite bath of feedstock, so  $n_L$  would be maintained by *net* molar or partial-molar flows to or from the bath that might not satisfy Eq. (7.3). If a **\*\*END** represents a location where the acoustics stops and an enriched product flows out of the system, Eq. (7.3) must apply to the outflow.

Here are several possible situations, with discussions of how to initialize variables in **BEGIN** and how to choose guesses and targets for each of them in **DELTAEC**. We only discuss the mixture-specific guesses and targets here; realistic mixture-separation models typically also include guesses and targets related to  $p_1$  and  $U_1$ .

## Sealed system

The steady-state conditions of Refs. [43, 44] represent sealed systems with  $\dot{N} = 0$  everywhere. Those experiments had no steady inflow or outflow, but were constrained by the mole fractions present in the original charge of gas mixture, so getting the right answer from **DELTAEC** depends on constraining **DELTAEC**'s solution with an integral of  $n_L$  over the volume of the apparatus to match the total number of moles of light component that was present in the original charge. Getting the correct answer to that integral would be the target corresponding to the guessed value of  $n_L$  in the **BEGIN**. In practice, we might approximate such an integral in **DELTAEC** with a simple, volumetrically weighted average of values at a small number of locations. Each such value could be captured with an **RPN** segment. The **BEGIN** should have  $\dot{N} = 0$  (and, by default,  $\dot{N}_L = 0$ ), because nothing flows in or out there.

## Firm connection to feedstock bath, no steady flow

If one end of a mixture-separation channel is firmly connected to a feedstock bath, then the system could be modeled starting at that end with a **BEGIN** having a known mole fraction and a guessed  $\dot{N}_L$ , and integrating along the channel. If the **BEGIN** is not at a location where the system is firmly connected to a feedstock bath, then  $n_L$  could be guessed at the **BEGIN** and the mole fraction could be targeted wherever the system is firmly connected to a feedstock bath.

## Inject feedstock at one end, withdraw product at the other end

Now consider a slow injection of feedstock at one end of the acoustic separation channel, and a removal of product at the other end, through capillaries [9]. When the system reaches steady state, the flow variables at the exit— $\dot{N}_L$  and  $\dot{N}$ —must equal their incoming values. Yet, if the flow is weak enough, we feel confident that a mole-fraction gradient must exist from end to end. Thus, the flow cannot satisfy  $\dot{N}_L = n_L \dot{N}$  at both ends. What happens? The exit flow is bulk, the mole fraction in the exit flow must be the same as the mole fraction

in the feedstock, and the feedstock mixes abruptly into a locally different mole fraction at the feedstock-end of the apparatus.

To model that situation with DELTAEC, the easiest approach is to put the **BEGIN** at the exit end, because the exit flow is bulk so we have fewer unknowns to guess there. Suppose we know the incoming  $\dot{N}$  and  $n_L$ . We use these values in the **BEGIN** (with a minus sign for the molar flux), check the “bulk flow” box to set  $\dot{N}_L$  appropriately, and integrate straightforwardly to the other end of the channel, where we learn what the mole fraction in the channel is where the feedstock mixes into it. This approach is the easiest because it involves only two segments—**BEGIN** and **MIXBL**—and it involves no targets or guesses.

A seemingly more logical and readable approach to this situation is to begin at the feedstock end, but then we need more DELTAEC baggage, to work around the fact that we know less information *a priori* at that end:

0. **BEGIN** with the known  $p_1$ ,  $U_1$ , feedstock  $\dot{N}$ , and feedstock  $n_L$ . Since the feedstock capillary flow is bulk, check the “bulk” box in the **BEGIN** segment to establish the correct  $\dot{N}_L$ .

1. **RPN** with **1a** a guessed mole fraction at the feedstock end of the separation channel and **1b** having `inp =nL .`

2. **MIXCIRC**.

3. **RPN** expressing the bulk condition, Eq. (7.3), for the exit flow, and target it. This target is associated with the guess in segment 1. For example, put `0.` in **3a** and `“nL Ndot * NLdot - ”` in **3b**.

## Feedstock in the middle of a long tube, products extracted at both ends

Let’s assume that we know the molar flow rate and mole fraction of the feedstock injected into the middle of a long separation tube, and that external control of the capillary flow impedances forces a known fraction of that molar flow out the top and the remainder out the bottom. We’ll begin at one of the places where the steady flow goes out, because we can count on bulk flow there. This situation is modeled in Ref. [10] and shown in the example in Section 7.3 below.

0. **BEGIN** with the acoustic conditions at one end of the channel, with guessed  $n_L$ . Set  $\dot{N}$  equal to the known rate of product extraction here. Check the “bulk” box in the **BEGIN** segment so  $\dot{N}_L$  is automatically assigned the appropriate, bulk value at this product-outflow location.

1. **MIXBL** for the first part of the channel.

2. **RPN** with the helium mole fraction of the injected feedstock in **2a**. (not targeted!)

3. **RPN** with the injected molar flow rate in **3a** and with the following instructions in **3b** to add the correct increment to both  $\dot{N}$  and  $\dot{N}_L$ : `3a Ndot + =Ndot 3a 2a * NLdot + =NLdot`

4. **MIXBL** for the second part of the channel.

At this point, we have the correct  $\dot{N}$  and  $\dot{N}_L$ , but we are not necessarily guaranteed the bulk condition needed for outflow into a capillary. Hence, we need the next segment:

5. **RPN** expressing the bulk condition, Eq. (7.3), and target it. For example, put `0.` in **5a** and `“ nL Ndot * NLdot - Ndot / ”` in **5b**. (We often multiply the bulk-flow-enforcement

expression  $0 = n_L \dot{N} - \dot{N}_L$  by a large number, or divide it by  $\dot{N}$ , to give DELTAEC a weighty enough target.) This target corresponds to the guessed value of  $n_L$  in segment 0.

## Firm connection to feedstock bath in the middle of a long tube, products extracted at both ends

This time, let's assume that we know the two exiting molar flow rates and the mole fraction of the feedstock bath in the middle. The feedstock bath must supply  $\dot{N}$  and  $\dot{N}_L$  in the center, but not necessarily in bulk proportions.

In hardware, how might we make a firm connection to a fixed mole fraction in the middle of a tube? We could have two capillaries there, and pump a huge mole flux into one of them and a nearly equal mole flux at a slightly different mole fraction out of the other one. The net flow—the sum of these two—is not a bulk flow.

0. BEGIN at one end of the channel with a guessed  $n_L$  and the known  $\dot{N}$ . Check the “bulk flow” box to get the known  $\dot{N}_L$ , because this is a location of product outflow.

1. MIXBL for the first part of the channel.

2. RPN Target the feedstock bath's mole fraction here, with the target value in 2a and nL in 2b.

3. RPN Guess the incremental  $\dot{N}$  in 3a. Add the incremental  $\dot{N}$  and  $\dot{N}_L$  to the separation tube in 3b using `inp Ndot + =Ndot inp 2a * NLdot + =NLdot` .

4. MIXBL for the second part of the channel.

5. RPN expressing the bulk condition, Eq. (7.3), and target it. For example, put 0. in 5a and `nL Ndot * NLdot - Ndot /` in 5b.

We have 2 targets (segments 2, 5) and 2 guesses (segments 0, 3).

## 7.3. Continuous He–Ar separator

Guidance for choosing the guesses and targets related specifically to mixture separation was given above. Here is an extended example from Ref. [10] including the targets and guesses relating to the acoustic wave. The apparatus is shown in Fig. 7.1.

First, we use DELTAEC to predict the shape of  $n_H(x)$  for steady flow injected at the tube midpoint and extracted at the ends, for one particular operating point, namely

$$\dot{N}_{\text{feedstock}} = 1.59 \mu\text{mol/s of 50–50 He–Ar at } x = 0.5 \text{ m,}$$

$$\dot{N}_{\text{product}} = 20\% \times 1.59 \mu\text{mol/s at } x = 0.0 \text{ m,}$$

$$\dot{N}_{\text{product}} = 80\% \times 1.59 \mu\text{mol/s at } x = 1.0 \text{ m,}$$

$$|p_1| = 3.00 \text{ kPa at } x = 0.50 \text{ m,}$$

$$z = Ap_1/U_1 = \rho_m a, \text{ purely real, at } x = 0.5 \text{ m.}$$

The DELTAEC model for the hardware dimensions given in Ref. [10] and the conditions listed above uses the mixture-specific guess and target described above under the heading “Feedstock in the middle of a long tube, products extracted at both ends,” plus three additional guesses and targets to enforce the acoustic conditions specified in that paper. The `<.out>` file is

```
TITLE      planned first expt in fy 06
!----05081220.out
!Created@14:57:50 28-Oct-06 with DeltaE Vers. 6.0g2 for the IBM/PC-Compatible
!-----0-----
```

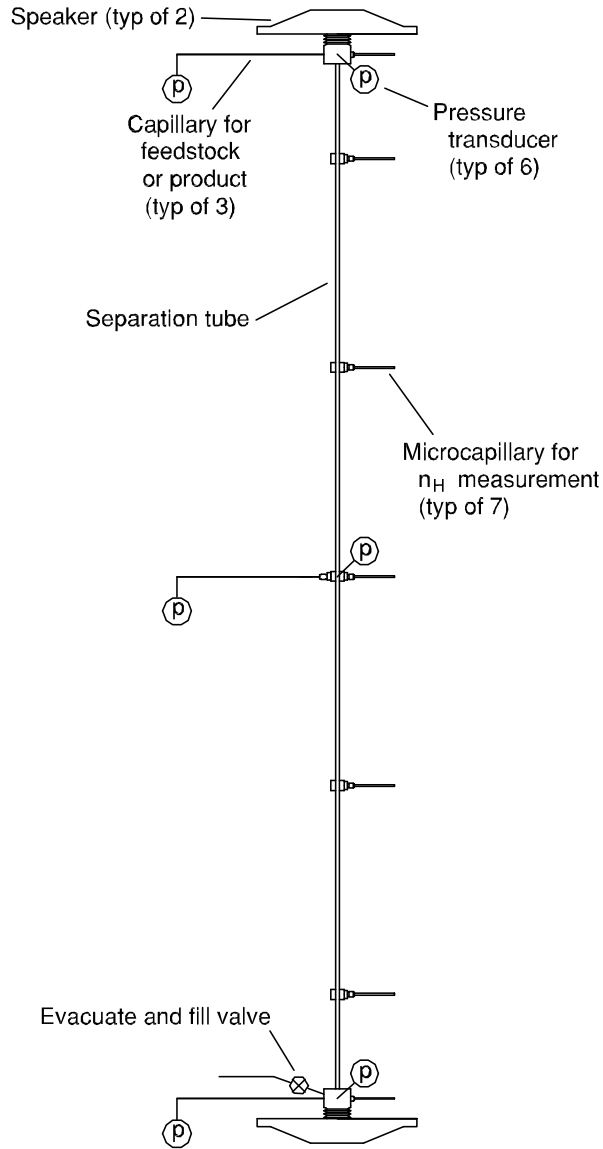


Figure 7.1: Apparatus for helium–argon separation used in Ref. [10] and in this example.



```

BEGIN      in front of top driver
7.9907E+04 a Mean P      Pa
200.00    b Freq        Hz
298.15    c TBeg        K
3882.6    d |p|         Pa      G
0.0000    e Ph(p)       deg
9.6980E-05 f |U|        m^3/s   G
-0.9501   g Ph(U)       deg      G
-2.6300E-08 h Htot      W
-3.1800E-07 i Ndot     mol/s
0.3406    j nL          G
HeAr      Gas type

!----- 1 -----
MIXCIRC   tube #1 (bottom)
8.5630E-06 a Area      m^2          3000.0    A |p|      Pa
1.0000    b GasA/A     -101.23  B Ph(p)    deg
0.5000    c Length     m           8.1411E-05 C |U|      m^3/s
1.6510E-03 d radius    m           -101.23  D Ph(U)    deg
-2.6300E-08 E Htot     W
0.1221    F Edot      W
0.3406    G nL Beg
0.4661    H nL End

!----- 2 -----
RPN       pressure amplitude in center
3000.0    a G or T      = 2A?    3000.0    A |p1|
p1 mag

!----- 3 -----
RPN       set Re[z]= rho a
0.0000    a G or T      = 3A?    3.6028E-16 A null
rho a * p1 U1 / 1a * real - rho a * /

!----- 4 -----
RPN       set Im[z] = 0
0.0000    a G or T      = 4A?    1.8585E-16 A Im z
p1 U1 / 1a * imag rho a * /

!----- 5 -----
RPN       feedstock concentration
0.5000    a G or T      (t)          0.0000    A nL Feed
0

!----- 6 -----
RPN       [a]=injection (micromol/sec)
1.5900    a G or T      (t)          6.8668E-07 A ChngeMe
1.2720E-06 B ChngeMe
6a 1e-6 * Ndot + =Ndot 6a 1e-6 * 5a * NLdot + =NLdot

!----- 7 -----
MIXCIRC   tube #2 (bottom)
sameas    1a a Area      m^2          2350.2    A |p|      Pa
sameas    1b b GasA/A     163.69    B Ph(p)    deg
sameas    1c c Length     m           6.7249E-05 C |U|      m^3/s
sameas    1d d radius    m           163.84    D Ph(U)    deg
-6.1169E-05 E Htot     W
7.9026E-02 F Edot      W
0.4661    G nL Beg
0.5398    H nL End

!----- 8 -----
RPN       exit bulk condition
0.0000    a G or T      = 8A?    -3.3295E-16 A null
nL Ndot * NLdot - Ndot /

```

The `<.sp>` file from that run was imported into a spreadsheet to convert  $n_L$  to  $n_H = 1 - n_L$  for plotting in Ref. [10]. Figure 7.2 shows the result, compared with measurements of the heavy mole fraction  $n_H$  that were made at 7 locations along the separation tube.

Next, an incremental plot was made, with feedstock  $\dot{N}$  as the independent variable and with an RPN segment maintaining the molar flow rate in the BEGIN segment equal to  $-20\%$  of the feedstock flow. The result is shown in Fig. 7.3.

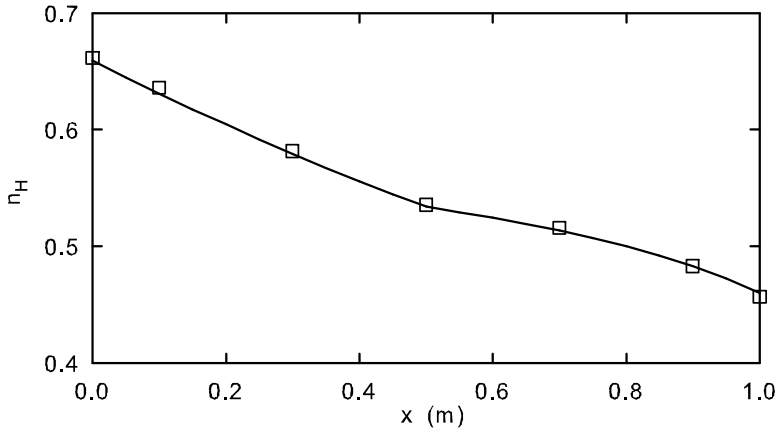


Figure 7.2: Heavy mole fraction  $n_H$  vs position in tube for one particular operating condition that is described in the text. The points are measurements from Ref. [10] and the curve is the result of DELTAEC's calculation as described in the text.

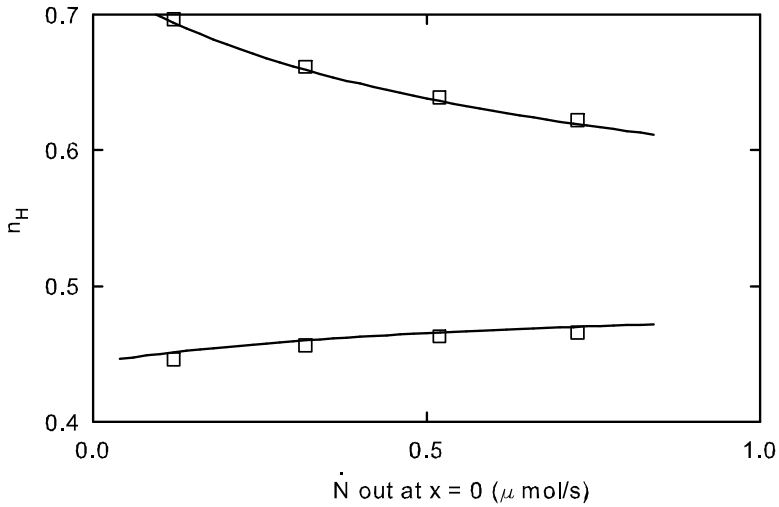


Figure 7.3: Heavy mole fractions  $n_H$  at  $x = 0$  and  $x = 1$  m, at various feedstock molar flows, with one product flow equal to 20% of the feedstock flow and the other product flow equal to 80% of the feedstock flow. The points are measurements from Ref. [10] and the curves are DELTAEC calculations as described in the text.

## Part II: Reference

# 8. GENERAL PRINCIPLES

## 8.1. Integration from BEGIN to \*\*END

Except for trivial cases (e.g., **IMPEDANCE**) that are handled algebraically, each of **DELTAEC**'s physical segments numerically integrates two or more of the following differential equations from its initial end to its final end:

$$\frac{dp_1}{dx} = \mathcal{F}_{\text{momentum}}(\text{local and global parameters and variables}), \quad (8.1)$$

$$\frac{dU_1}{dx} = \mathcal{F}_{\text{continuity}}(\text{local and global parameters and variables}), \quad (8.2)$$

$$\frac{dT_m}{dx} = \mathcal{F}_{x\text{-energy}}(\text{local and global parameters and variables}), \quad (8.3)$$

$$\frac{dT_{\text{solid}}}{dx} = \mathcal{F}_{\text{solid}}(\text{local and global parameters and variables}), \quad (8.4)$$

$$\frac{d\dot{H}_{\text{tot}}}{dx} = \mathcal{F}_{\text{lateral energy}}(\text{local and global parameters and variables}), \quad (8.5)$$

$$\frac{dp_{2,0,HL}}{dx} = \mathcal{F}_{\text{head loss}}(\text{local and global parameters and variables}), \quad (8.6)$$

$$\frac{dn_L}{dx} = \mathcal{F}_{\text{mix sep}}(\text{local and global parameters and variables}). \quad (8.7)$$

The first of these equations, involving  $p_1$  and  $U_1$ , are sufficient for users who are interested only in acoustics, without thermoacoustics. For these users, the other equations can be ignored. Users who are interested in ordinary thermoacoustic and Stirling engines and refrigerators must also concern themselves with  $T_m$ ,  $T_{\text{solid}}$ , and  $\dot{H}_{\text{tot}}$ . Very few users have to consider all seven of these equations simultaneously.

In most segments that employ numerical integration of these equations, **Nint** steps of fourth-order Runge-Kutta integration [16] are employed. However, **HX**, **TX**, **SX**, and **PX**, which are most often used for heat exchangers whose lengths are only of the order of  $|\xi_1|$ , employ only one step of fourth-order Runge-Kutta integration.

For each integration through the segments in the model, the initial values of the integration variables,  $p_1$ ,  $U_1$ ,  $T_m$ ,  $\dot{H}_{\text{tot}}$ ,  $p_{2,0,HL}$ , and  $n_L$ , are set in the **BEGIN** segment. Except for  $T_{\text{solid}}$ , all of these variables are taken to be continuous between segments; but  $T_{\text{solid}}$  is usually discontinuous at the ends of heat exchangers.

A few additional variables, such as  $\dot{N}$ , are also initialized in the **BEGIN** segment but are independent of  $x$  unless they are changed abruptly at a logistical segment such as a **TBRANCH** or via a variable-reassignment operator such as **=Ndot** in an **RPN** segment.

The total power flow  $\dot{H}_{\text{tot}}$  is often independent of  $x$ , but there are important exceptions. In heat exchangers,  $\dot{H}_{\text{tot}}$  changes by the amount of heat added (or removed), and in transducers it changes by the amount of power added (or removed). If the **ANCHOR** logistical segment (described in Section 11.3.1) is used upstream,  $\dot{H}_{\text{tot}}$  also changes in every component except **STK\*\*s**, because **ANCHOR** means that acoustic power dissipation is removed locally as heat in each such component.

Usually the initial values of at least some of these integration variables are unknown at the **BEGIN**, so **DELTAEC** uses a shooting method to find initial values that are consistent with targeted results that are identified by the user elsewhere in the model. The shooting method is described in the next section.

After the shooting method has met its targets, **DELTAEC** knows the complete solution  $p_1(x)$ ,  $U_1(x)$ ,  $T_m(x)$ ,  $T_{\text{solid}}(x)$ ,  $\dot{H}_{\text{tot}}(x)$ ,  $p_{2,0,HL}(x)$ , and  $n_L(x)$ . Three additional useful functions of  $x$  are calculated straightforwardly from that solution: the acoustic power  $\dot{E}$ , the exergy flow  $\dot{X}$ , and the total second-order time-averaged pressure  $p_{2,0,\text{tot}}$ :

$$\dot{E}(x) = \frac{1}{2} \text{Re} \left[ p_1 \tilde{U}_1 \right], \quad (8.8)$$

$$\begin{aligned} \dot{X}(x) = & \frac{T_0}{T_m} \dot{E} + \left( 1 - \frac{T_0}{T_m} \right) \dot{H}_{2,k} + \dot{N} (mw_m - T_0 ms_m) \\ & + \dot{N}_L R_{\text{univ}} T_0 \ln \left( \frac{n_L}{n_{L,0}} \right) + \left( \dot{N} - \dot{N}_L \right) R_{\text{univ}} T_0 \ln \left( \frac{1 - n_L}{1 - n_{L,0}} \right), \end{aligned} \quad (8.9)$$

$$p_{2,0,\text{tot}}(x) = p_{2,0,\text{rev}} + p_{2,0,HL} \quad (8.10)$$

$$= \frac{|p_1|^2}{4\rho_m a^2} - \frac{\rho_m |U_1|^2}{4A_{\text{gas}}^2} + p_{2,0,HL}. \quad (8.11)$$

The expression for acoustic power is discussed in Ref. [13] and should be familiar to most acousticians.

The expression for  $\dot{X}$  is an amalgam of the basic thermoacoustic exergy flow derived in Ref. [13], a steady-flow component derived in Ref. [32], and the  $\dot{N} \neq 0$  extension of the thermoacoustic mixture-separation expression derived in Ref. [44]. The reference temperature  $T_0$ , which is the ambient temperature at which heat has no value, is set to 300 K by default in **BEGIN** segments and can be changed to a different value by using the operation **=Tzero** in an **RPN** segment soon after the **BEGIN**. Steady pressure makes no contribution to the steady-flow component because we assume that the reference pressure for exergy is  $p_m$ . The second line of Eq. (8.9) is relevant only for mixtures. The zero-exergy mixture mole fraction,  $n_{L,0}$ , is the mole fraction for which the mixture has no value, typically the feedstock's mole fraction. It is set to 0.500 by default in **BEGIN** segments, and can be changed to another value by using **=nLzro** in an **RPN** segment soon after the **BEGIN**.

The reversible part of the second-order time-averaged pressure,  $p_{2,0,\text{rev}}$ , is derived in Ref. [36], and measurements of  $p_{2,0,\text{tot}}$  are described in Ref. [35].

These ten functions of  $x$ , as well as many others (e.g., the mean density  $\rho_m$ , the vibration force  $F_1$  described in Section 11.4.1) are available to the user in tabular and graphical forms via the **<.sp>** file, at the end of each segment in the **<.out>** file, and/or at any location via **RPN** segments.

### 8.1.1. Additional details for $\dot{N}$ and $p_{2,0,HL}$

DELTAEC's input parameter  $\dot{N}$ , denoted in the Users Guide by  $\dot{N}$ , represents the true time-averaged molar flow, which is the steady-flow quantity under the most direct experimental control and the quantity that carries steady enthalpy flux

$$\dot{H}_{\dot{N}} = \dot{N}mw_m, \quad (8.12)$$

where  $m$  is the molar mass and  $w_m$  is the mean enthalpy per unit mass. The total power

$$\dot{H}_{\text{tot}} = \dot{H}_{2,k} + \dot{H}_{\dot{N}} \quad (8.13)$$

is the power that obeys the first law of thermodynamics for open control volumes. The total power  $\dot{H}_{\text{tot}}$  is the sum of the second-order time-averaged thermoacoustic enthalpy flux, the ordinary thermal conduction in the  $x$  direction, and  $\dot{H}_{\dot{N}} = \dot{N}mw_m$ .

In DELTAEC, mean enthalpy is referenced to zero at temperature  $T_0$ , which is set to 300 Kelvin by default in the first **BEGIN** segment of a model. (This default can be overridden by using the **=Tzero** operator in an **RPN** segment.) For an ideal gas, Eq. (8.12) can then be regarded as

$$\dot{H}_{\dot{N}} = \dot{N}mc_p(T_m - T_0). \quad (8.14)$$

To second order,

$$\dot{N}m \simeq \frac{1}{2}\text{Re} \int [\rho_1 \tilde{u}_1] dA_{\text{gas}} + \rho_m U_{2,0}, \quad (8.15)$$

where  $m$  is the molar mass. However, the second-order time-averaged momentum equation shows that second-order time-averaged head-loss gradients are generated by  $U_{2,0}$ , with no contribution from  $\rho_1 \tilde{u}_1$ . Defining  $\dot{N}_{2,0} \equiv \rho_m U_{2,0}/m$ , substituting  $\rho_1 = (\gamma/a^2)p_1 - \rho_m \beta T_1$  in Eq. (8.15), and solving for  $\frac{1}{2}\text{Re} \int [T_1 \tilde{u}_1] dA_{\text{gas}}$  in the standard thermoacoustic expression for  $\dot{H}_{2,k}$ , it is easy to show that Eq. (8.15) is equivalent to

$$\dot{N}_{2,0} = \dot{N} + \frac{\beta}{mc_p} \left[ \dot{H}_{2,k} + (A_{\text{gas}}k + A_{\text{solid}}k_s) \frac{dT_m}{dx} \right] - \left( \frac{1}{ma^2} + \frac{\beta}{mc_p} \right) \dot{E}, \quad (8.16)$$

which is the molar flow that is used in most of DELTAEC's  $p_{2,0,HL}$  calculations.<sup>1</sup> Note that  $\dot{N}_{2,0}$  is often nonzero when  $\dot{N} = 0$ , which means that interesting second-order, time-averaged head losses arise in many circumstances having  $\dot{N} = 0$ , such as in the regenerators of TASHEs and pulse-tube refrigerators. Click the **Enable Ndot** button in the **BEGIN** segment's **Optional parameters** to “turn on” the display of **p20HL** even when  $\dot{N} = 0$ .

## 8.2. Shooting method

If initial values of all of the integration variables of interest [chosen from among  $p_1(x)$ ,  $U_1(x)$ ,  $T_m(x)$ ,  $T_{\text{solid}}(x)$ ,  $\dot{H}_{\text{tot}}(x)$ ,  $p_{2,0,HL}(x)$ , and  $n_L(x)$ ] are known at the **BEGINning** end

---

<sup>1</sup>In a few expressions for the dissipative parts of  $dp_1/dx$  below, our choice of whether to use  $\dot{N}_{2,0}$  or  $\dot{N}$  is based as much on history or convenience as on careful thought. In those cases,  $\dot{N} - \dot{N}_{2,0}$  is of second order and affects  $dp_1/dx$  only to third order.

of the thermoacoustic apparatus, then DELTAEC’s numerical integration is straightforward, proceeding from the initial end to the final end. But usually one or more boundary conditions are instead specified at the final end or somewhere in the middle. In such circumstances DELTAEC uses a shooting method [16], by *guessing* any unknowns at the **BEGIN**, integrating to the **\*\*END**, comparing the results with the *target* boundary conditions imposed at that end and/or elsewhere, and adjusting its guesses until the targets are met. Up to eighteen guesses and eighteen targets are allowed. One of DELTAEC’s most powerful features is that the guesses and targets are not limited to the conventional choices, which would be the initial values of integration variables at the **BEGIN** as guesses and calculated values of integration variables at the **\*\*END** as targets. *Any* results can be used for targets, and *any* upstream variables that have effects on downstream target variables can be used as guesses. This enables DELTAEC to calculate a resonance frequency, a geometrical dimension, a temperature, or even the concentration in a binary gas mixture to satisfy desired boundary conditions.

DELTAEC treats its collection of  $n$  guesses and  $n$  targets as a system of  $n$  nonlinear equations,

$$F_1 = r_1(g_1, g_2, g_3 \dots g_n) - t_1, \quad (8.17)$$

$$F_2 = r_2(g_1, g_2, g_3 \dots g_n) - t_2, \quad (8.18)$$

$$F_3 = r_3(g_1, g_2, g_3 \dots g_n) - t_3, \quad (8.19)$$

$$\dots \quad (8.20)$$

$$F_n = r_n(g_1, g_2, g_3 \dots g_n) - t_n, \quad (8.21)$$

where  $g_i$  is the  $i$ th guess,  $t_i$  is the  $i$ th target value, and  $r_i$  is the  $i$ th calculated result, which we hope can be made equal to the  $i$ th target  $t_i$  by suitable choices of the  $n$  guesses  $g_i$ . Thus, DELTAEC must seek values of the  $n$  guesses  $g_i$  that make  $F_i = 0$  for all  $i$ . Several methods for finding the simultaneous zeroes of complicated, simultaneous functions exist. The routine incorporated in DELTAEC is called DNSQ, and it is part of the SLATEC Common Mathematical Library, which is freely available through the internet software repository at <http://www.netlib.org>. The algorithm used is a modification of the Powell hybrid method [46, 47] (a.k.a. the Powell dogleg method). The ordinary DELTAEC user can just as well imagine that DNSQ is the Newton–Raphson method, which is easy to understand and is described in Ref. [16] and many tutorial websites. The description below is based on this oversimplification.<sup>2</sup>

A Newton–Raphson algorithm begins with initial “guessed” values of the guesses, and calculates the  $n$  values of  $F_i$ . If  $\sum (F_i)^2$  is not close enough to zero, the algorithm estimates the  $n^2$  local slopes

$$\frac{\partial F_i}{\partial g_j} \simeq \frac{F_i(g_1, g_2, \dots, g_j + \delta g_j, \dots, g_n) - F_i(g_1, g_2, \dots, g_j, \dots, g_n)}{\delta g_j}. \quad (8.22)$$

---

<sup>2</sup>Sometimes Newton–Raphson gets hung up because the matrix  $\partial F_i / \partial g_j$  is singular or nearly so. Then the “hybrid” aspect of Powell’s method is invoked, blending Newton–Raphson with a gradient descent toward a minimum of Eq. (8.23).

The algorithm forms a “better” set of guesses based on the assumption that these functions  $F_i$  are linear, and repeats the process, until it judges the results to be close enough to the targets, according to

$$\sqrt{\sum (w_i F_i)^2} \leq \text{Tolerance} \times \sqrt{\sum (w_i t_i)^2}, \quad (8.23)$$

where the weighting factors  $w_i$  passed from DELTAEC to DNSQ are given by

$$w_i = \frac{1}{360} \text{ if the target is a phase angle,} \quad (8.24)$$

$$= \frac{1}{|t_i|} \text{ if } |t_i| \geq 10^{-4} \text{ and the target is not a phase angle,} \quad (8.25)$$

$$= 10^4 \text{ if } |t_i| \leq 10^{-4} \text{ and the target is not a phase angle} \quad (8.26)$$

in the default “Normalization Mode = 1.” When  $\text{Tolerance} = 10^{-N}$ , this should ensure  $N$  digits of accuracy for most targets like temperatures and pressure amplitudes, and values  $< 10^{-(N+4)}$  for “zero” targets such as normalized \*\*END impedances. Small negative values (e.g.,  $-4$ ) seem to work best for pulse-tube refrigerators and many other models.

This only works if the functions  $F_i$  are reasonably smooth. To ensure that each target result is a smooth function of all guesses, we have deliberately made DELTAEC’s built-in features be continuous functions. However, some have discontinuous changes in slope [e.g., Eqs. (10.115) and (10.116)]. For an example of a pathological discontinuity that could easily overwhelm the shooting method, see the end of Section 8.5.

In complicated models, choosing appropriate guesses and targets and getting DELTAEC to hit the targets are the most challenging aspects of using DELTAEC. Guidance for choosing guesses and targets and for meeting targets appear in Secs. 5.3 and 5.4, respectively.

### 8.3. Numerical options

DELTAEC has many internal parameters that can be altered by the experienced user to control the behavior of DELTAEC’s numerical integration and shooting method. These parameters are accessible via Options in the Edit pulldown menu.

If any of the parameters described below are modified from their default values, it is usually desirable to keep the new values for all subsequent runs, even after exiting from DELTAEC and restarting. Therefore, whenever the default settings are changed, all of the tunable parameters are written to a special file when the model is saved. This file has the same base filename as the model, ending with `<.ini>`. Whenever a new model is loaded, DELTAEC checks in the same folder for a `<.ini>` file with a matching base filename and loads these settings if it is found. This file is written in NAMELIST format which makes it easy to examine and modify using any text editor.

Frequently, a collection of similar models will reside in a single folder, and these files will share identical custom settings. For these situations, any `<.ini>` file can be copied (or renamed) to `<default.ini>` and DELTAEC will use the settings within `<default.ini>` for any model run from that folder. If a model has its own individual `<.ini>`, the individual settings take precedence over `<default.ini>`.



### 8.3.1. Integration options

**Runge-Kutta steps.** Default: 10. Recommended range: 1 – 100. This is an integer `Nint` that determines the number of integration steps used to span each `STK**`, `DUCT`, `CONE`, `MIX**`, and `VX**` segment. It does not affect other segment types. It also determines the resolution in state plots: `Nint + 1` points per integrable segment. Larger values will cause a slower, more accurate computation. Larger values are beneficial for long `VX**` segments and when large nonzero  $\dot{N}$  flows through `STK**` segments. Small values will increase speed at the price of integration accuracy and possible convergence problems.

**Minimum Temperature.** Default: 10 K. This is a temperature floor to prevent DELTAEC’s shooting method from exploring unphysical temperatures such as negative temperatures. Brave users with special needs at lower temperatures (and with their own, user-defined thermophysical properties files!) can set this floor to a lower value. (Some of DELTAEC’s internal gases use a higher temperature floor. See Chapter 12.)

### 8.3.2. Shooting options

Most of these parameters are described more fully in the online documentation for DNSQ at the SLATEC software repository. More tips on successful shooting in complicated DELTAEC models are in Section 5.4.

**Convergence Tolerance.** Default:  $10^{-8}$ . Recommended range:  $10^{-8}$  or smaller. This parameter governs the accuracy at which DELTAEC considers its shooting-method iterations finished. See Eq. (8.23). At  $10^{-N}$ , this should ensure  $N$  digits of accuracy for most targets, and values  $< 10^{-(N+4)}$  for “zero” targets if the Normalization Mode is 1.

**Normalization Mode.** Default: 1. In a numerical problem in which the output variables used in the target list differ greatly in magnitudes, a difficulty arises in choosing how much to weigh the errors between each target and corresponding result. This particularly affects `**END` segments. A 0.001 K error in a heat-exchanger temperature is fairly benign to us, but an error of 0.001 in the normalized complex end impedance  $z_n$  could leave us with tens of watts of power flow where there must be zero. In the standard Normalization Mode (1), DELTAEC uses the target weighting described in Eqs. (8.24)–(8.26) to normalize targets, which often does a reasonable job. For unusual cases, set Normalization Mode to an integer  $\neq 1$ . Then Eqs. (8.24)–(8.26) become

$$\begin{aligned} w_i &= \min\left(\frac{1}{360}, 10^{3+\text{Mode}}\right) \text{ if the target is a phase angle,} \\ &= \min\left(\frac{1}{|t_i|}, 10^{3+\text{Mode}}\right) \text{ if the target is not a phase angle,} \end{aligned}$$

so a higher Mode assigns greater weight to targets  $|t_i| \leq 10^{-4}$ . We often find that Mode =  $-3$ ,  $-4$ , or  $-5$  works well for pulse-tube refrigerators and other traveling-wave systems with large numbers of guesses and targets. See also the strategy for changing the weights of individual targets described near the end of Section 5.4.2.

**EpsFcn.** Default:  $10^{-10}$ . Recommended range:  $> 5 \times 10^{-15}$ . To estimate  $\partial F_i / \partial g_j$  using Eq. (8.22), DELTAEC sets  $\delta g_j = g_j \sqrt{\text{EpsFcn}}$ . (But if  $g_j = 0$ , then  $\delta g_j = \sqrt{\text{EpsFcn}}$ .) A large value of **EpsFcn** can sometimes speed iterations when initial values of guesses are very inaccurate. Too small a value can cause the solver to lose its way completely, as roundoff error in the computer ruins the accuracy with which  $\partial F_i / \partial g_j$  is calculated. The optimal value is computer-dependent and model-dependent. To guard against too-small values, **EpsFcn** has a built-in floor of  $2 \times 10^{-16}$ , twice the double-precision roundoff bound (see Wikipedia, “Machine epsilon”).

**Step Bound Factor.** Default: 100. Recommended range: 0.01–100. This parameter controls the boldness with which DELTAEC first tries to adjust the  $g_i$ , after the  $\partial F_i / \partial g_j$  have been calculated for the first time. Some difficult cases can benefit from a smaller Step Bound Factor.

## 8.4. Error messages

Launching DELTAEC version 6.4 can momentarily display a black window with an OMP warning and an OMP System Error. This can be ignored.

If you find a reproducible problem with DELTAEC, please email [ww@lanl.gov](mailto:ww@lanl.gov) and [swift@lanl.gov](mailto:swift@lanl.gov), with an explanation of what you were doing, including a model that triggers the problem. DELTAEC saves some error-diagnostic messages in `<DeltaEC.exe.log>`, whose contents can be seen in the Help pulldown menu. This file might be stored in a folder such as `<\Users\...\AppData\Local\VirtualStore\DeltaEC>`, `<\MyDocuments>`, or `<\ProgramFiles(x86)\DeltaEC>`. (To deliberately trigger the addition of a few lines to that file, modify a parameter, but type a letter instead of a number.) Clear the log file, reproduce the problem you plan to report, and include the contents of the log file in your email if your problem causes any messages there.

Occasionally DELTAEC will produce a “wxWidgets Debug Alert.” If you see this alert, please copy the cryptic message in the top of the box and send it to [ww@lanl.gov](mailto:ww@lanl.gov) and [swift@lanl.gov](mailto:swift@lanl.gov), with a brief explanation of what you were doing. Please also check for a `<DeltaEC.exe.log>` file as described in the previous paragraph.

Most of DELTAEC’s other diagnostics are meant to be self-explanatory, but some may require additional explanation:

**Warnings were generated while parsing...** This may indicate a problem with the `<.out>` file. This can occur when a version-5 model is opened in version 6 for the first time, or when the savvy user has made a minor mistake when editing a `<.out>` file with a text editor instead of using DELTAEC’s built-in parameter- and segment-modification methods. Examine the `<.err>` file for specific information. When translating a model from version 5 to version 6, DELTAEC gives warnings whenever it is unsure of itself, even though it usually makes the right assumptions about the translation.

**Success, Partial convergence, or FAILED.** The Run Monitor at the bottom of the main display gives one of these three messages after every run, on a panel of green, yellow, or red. “Success” indicates that targets were met with the accuracy implied

by **Convergence Tolerance** (or no targets and guesses exist at all). “**FAILED**” and **Iteration is complete**, but some results may not be near their targets indicate that targets have definitely not been met to DELTAEC’s satisfaction. If the error occurs on a model that has displayed good convergence under other conditions, this may simply be one pathological point. The user may be able to jump-start it by manually (somewhat intelligently) changing the value of one or two members of the guess list to put the solver on the right track. Otherwise, consider the tactics described in Section 5.4. “**Partial convergence**” indicates that targets are almost met to DELTAEC’s satisfaction, and in most cases the user can also be satisfied. Usually “**Partial convergence**” can quickly be changed to “**Success**” simply by `run` again. If the frequent appearance of “**Partial success**” becomes a nuisance, try changing the **Tolerance** and/or **Normalization Mode**, described in Section 8.3.2.

**Plot non-convergence indicators.** When a convergence error occurs during an incremental plotting operation, a ‘\*’ is prepended to the plot line for that point, followed by a single digit representing the “info” variable at that point. A typical line in the `<.ip>` file might then look like this:

```
*4 2.990      98.61      528.2      45.36      559.3      3219.
```

The plotter indicates such suspicious and failed convergence by plotting the point in red. Below is a key to interpreting the “info” codes:

info=	Significance
1	Partial convergence: The solver considers the iteration successful, but the residual error is suspiciously large.
2	The solver was making progress, but the maximum number of iterations has been reached. Another <code>run</code> might make further progress.
4	FAILED: The solver was unable to converge.

Sometimes the user may want to delete the prepended asterisk and number. For example, if the code is “info=2,” and the remaining error that the solver was working on was already quite small, it might be quite valid to use this plot point instead of discarding it.

## 8.5. Inherent limitations

Users should be aware of fundamental, inherent limitations in DELTAEC’s ability to model real hardware:

DELTAEC assumes that  $p_1$  depends only on  $x$ , that  $u_1$  depends only on  $x$  except within the viscous boundary layer, and that  $T_m$  depends only on  $x$ . Thus, DELTAEC cannot accurately model flow in short-radius elbows, flow and pressure at abrupt transitions between DUCTs of different areas (as shown in Fig. 8.1), or inertial or thermal effects in thermal buffer tubes and pulse tubes with significant Rayleigh streaming. (For example, in `STKDUCT`,  $T_m(x)$  and  $T_{\text{solid}}(x)$  are equal for all  $x$ , which is a very bad approximation if significant Gedeon streaming, Rayleigh streaming, or jet-driven streaming

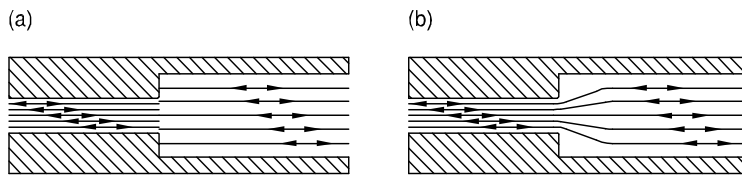


Figure 8.1: (a) Naive and (b) more-realistic sketches of streamlines where two large-diameter DUCTS join. DELTAEC performs inherently one-dimensional calculations, so it naively imagines that the streamlines are as shown in (a). In reality, some high-velocity flow extends into the large-diameter duct as shown in (b), adding inertance that DELTAEC does not automatically account for, and resistive minor losses also occur at high Reynolds numbers. Even (b) may not be sufficiently realistic, because the flow at such a transition is seldom left-right symmetrical. These effects can and usually should be modeled with RPN calculations and MINOR and/or IMPEDANCE segments.

is present.) We use MINOR segments, RPN segments, IMPEDANCE segments, and off-line hand calculations to try to account for 2- and 3-dimensional effects like these.

The derivations of most thermoacoustic equations rely on the assumption that the gas displacement amplitude  $|U_1|/\omega A$  is much smaller than all other relevant dimensions in the  $x$  direction. This assumption is often violated in thermoacoustic components, especially heat exchangers, whose length in the  $x$  direction is seldom larger than  $2|U_1|/\omega A$ .

The turbulence algorithm in DUCT, CONE, and VX\*\* is based on the quasi-steady approximation, namely that the pressure gradient at any instant of time in time-dependent flow is the same as that which would be present for steady flow at the same rate. This is known to be naive [48, 49, 50, 51].

Similarly, the turbulence algorithm in STKSCREEN and STKPOWERLW is based on the quasi-steady approximation, which is a poor approximation when gas displacement amplitude is smaller than, or comparable to, the screen pore size. Thus, this aspect of DELTAEC works poorly at *low* amplitudes.

Harmonics are often present in  $p(t)$  and  $U(t)$ , but DELTAEC works exclusively with the fundamentals  $p_1$  and  $U_1$ . This limitation usually has little impact on the accuracy of power calculations [13], but experiments being compared to DELTAEC calculations must use instruments that measure the fundamental components of the wave, not rms or peak values.

DELTAEC's shooting method is efficient at converging to solutions for complicated systems; however, it knows nothing about acoustics, or any other aspect of physical reality. DELTAEC doesn't know that negative frequencies, negative lengths, or pressure amplitudes greater than the mean pressure are improper; it simply does the math. Thus, the reasonableness of the answers produced by DELTAEC's shooting method often depends on the quality of the initial guesses, which in turn depends on the thermoacoustics knowledge of the user.

Be alert for situations where DELTAEC cannot be expected to understand the user's intentions. DELTAEC does not warn the user of all such possible inconsistencies and ambiguities. For example, DELTAEC can't guess why a user would allow nonzero  $\dot{N}$  to flow through an enclosed **\*\*SPEAKER** or **\*\*DUCER**, so no change in  $p_{2,0,HL}$  is calculated. The velocity term in Eq. (6.12) for  $p_{2,0,tot}$  depends on area, so don't be surprised if you disagree with its calculation at an **IMPEDANCE** segment, since **IMPEDANCES** have no area. Allowing significant nonzero  $\dot{H}_{2,k}$  (when  $\dot{H}_{2,k} \neq \dot{E}$ ) to flow through an **IMPEDANCE** or a tight-gapped **HX** or **TX** is physically implausible, and leads to errors in the calculation of  $\dot{N}_{2,0}$  via Eq. (8.16) and thus to erroneous calculation of  $p_{2,0,HL}$ .

**VXQ1** and **VXQ2** rely on assumptions that may be invalid in some circumstances. See Section 10.7.3 for details.

The shooting method requires that target results change *smoothly* as guesses are varied, so the near-linearity assumed in Eqs. (8.22) can be a good-enough approximation for small variations in the guesses. So, be wary of building step-function or absolute-value calculations into **RPN** segments if downstream target results depend on them. For example, if a model is operating near  $|\xi_1| = D$  at the location of an orifice plate in a **DUCT**, abruptly switching on a minor loss as shown here may be very problematic if the model is operating close enough to the transition that DELTAEC's guess variations carry it back and forth between  $|\xi_1| > D$  and  $|\xi_1| < D$ .

```

5      RPN      inp = orifice diameter (m);  result = area (m2)
1.0000E-2  a G or T      7.8540E-5      A m2
inp 2 / sqrd pi *
6      RPN      Turn on minor loss only if |xi_1| > D. Result is 0 or 1.
0.0000  a G or T      1.0000      A K+, K-
U1 mag 5A / w / 5a - sto abs rcl / 1 + 2 /
7      MINOR      small orifice plate in big duct
sameas  5A a Area  m^2
sameas  6A b K+
sameas  7b c K-

```

A more subtle abrupt change in results can arise from DELTAEC's treatment of phase angles. When displaying its internal complex numbers as magnitude and phase, DELTAEC selects the phase to be  $-180^\circ < \theta \leq 180^\circ$ . Targeting phase angles near  $0^\circ$ ,  $90^\circ$ , or  $90^\circ$  is unproblematic. However, when trying to target a phase to be  $180^\circ$ , DELTAEC's shooting method explores guess values that yield results both slightly above and slightly below  $180^\circ$ . The results that are slightly above  $180^\circ$  will be seen by the shooting method as slightly above  $-180^\circ$  (e.g.,  $181^\circ$  becomes  $-179^\circ$ ), and the shooting method will have no chance to hit this target. The simplest way to avoid this problem is to shift the time phase of the entire model by a large, convenient amount, such as  $180^\circ$  or  $\pm 90^\circ$ . (See Section 2.6.2.) Another simple work-around is to target  $\sin \theta = 0$  instead of  $\theta = 180^\circ$ , although this technique can sometimes let the shooting method "jump" from  $\theta = 180^\circ$  to  $\theta = 0^\circ$ . More awkwardly, to convert a standard DELTAEC phase angle to a function that changes smoothly as it passes through  $180^\circ$ , use the following trick, which yields  $\text{ph}(U)$  if  $\text{ph}(U) > 0$ , and adds  $360^\circ$  to  $\text{ph}(U)$  if  $\text{ph}(U) < 0$ , thus shifting the troublesome discontinuity from  $180^\circ$  to  $0^\circ$ .

```

7      RPN      target phase(U1) here to be 180
180.00
U1 arg U1 arg abs - U1 arg / 180 * U1 arg +

```

## 9. GOOD DESIGN HABITS

When we study fundamental thermoacoustics issues, we try to design experiments to minimize unrelated effects so the issues being studied stand out clearly. In contrast, when we apply what we know about thermoacoustics to design a practical device for a goal-oriented sponsor, we often find ourselves forced by the sponsor’s goals into operating conditions with great complexity and significant uncertainty, and yet we must account for *everything* as well as possible. This chapter outlines some habits that we use at Los Alamos, as of 2008, to make complex DELTAEC design models as realistic as possible, so we and our goal-oriented sponsors are not disappointed when hardware is built and tested.

Our philosophy: (1) Attention to details, because there are many details and their effects add up. (2) Component-by-component conservatism, based on experimental evidence when it differs from “raw” DELTAEC results. Ignoring these considerations can lead to factor-of-two overprediction of performance.

We try to be honest with ourselves about these issues: If there are enough such issues present in a given design, then uncertainties should tend to average out if each issue is treated honestly, so we’ll end up with hardware that works approximately like we predict, with equal probability of working a little better or a little worse. In contrast, neglecting these issues early in a design process can produce misleading “optimization” that does not represent reality, and neglecting them in the final stages of a design usually leads to disappointment when hardware is tested.

Note that some of these details can be neglected at low amplitudes.

(Thanks to Scott Backhaus for contributions to this chapter.)

### 9.1. Attention to details

#### 9.1.1. Imperfect external heat exchange

We pay attention to the interfaces between DELTAEC-modeled heat exchangers and their environment. Even though DELTAEC is not set up to model anything outside the “skin” of the solid that is in contact with the thermoacoustic gas in heat exchangers, many effects beyond that boundary are important, and should be modeled, either using RPN segments within DELTAEC or using calculations elsewhere (e.g., spreadsheets). For example, in a water-cooled heat exchanger, non-negligible temperature differences usually exist across the water boundary layer (due to the finite thermal conductivity of the water) and between the inlet and outlet streams (due to the finite specific heat and flow rate of the water). Our sponsors usually care about the water inlet temperature, not DELTAEC’s heat-exchanger internal temperature. Similar considerations apply in a drilled metal-block heat exchanger

that relies on conductivity of metal to transfer heat between the thermoacoustic gas in the drilled channels and an external heat source or sink at the perimeter of the block.

### 9.1.2. Internal 2-D effects from imperfect external heat exchange

DELTAEC assumes that  $p_1$ ,  $U_1$ , and  $T_m$  are independent of coordinates perpendicular to  $x$ . Sometimes the low heat capacity and/or low flow rate of an external heat-transfer fluid (especially combustion products from a burner flowing through an engine’s hot heat exchanger) ruin this laterally-uniform- $T_m$  assumption in a cross-flow heat exchanger, thereby ruining the same assumption in an adjacent stack or regenerator, which in turn can cause lateral nonuniformity in  $U_1$  and/or internal streaming in the stack or regenerator. DELTAEC is fundamentally ill suited to modeling such issues, yet we must consider them. To get a rough idea of the impact of such nonuniformity, we have sometimes used TBRANCH and UNION to split a hx-regen-hx (or hx-stack-hx) package into two (or even three) parallel subsets, each with half (or a third) of the total area, allowing different  $T_m(x)$  in each. RPN calculations can be used to keep the heat-exchanger temperatures consistent with the heat-transfer fluid’s flow rate and heat capacity. (Such split models usually require so many guesses and targets that other guesses and targets in the model must be disabled while this issue is being explored.)

### 9.1.3. Heat leaks

We include axial thermal conduction of the pressure vessel surrounding any STK\*\* components that have nonzero axial temperature gradients. This is built in to STKDUCT and STKCONE segments, but it is not built in to the other STK\*\* segments used for stacks and regenerators. In STKSCREEN regenerators, we often tie ksFrac to an RPN calculation that knows the vessel wall geometry, so the vessel wall conduction shows up directly in HX or other \*X segments’ results. If the screens and the pressure vessel are made of the same material, the expression is

$$k_{s,\text{Frac,effective}} = k_{s,\text{Frac,Lewis}} + \frac{A_{\text{wall}}}{(1 - \phi) A_{\text{screen}}}, \quad (9.1)$$

where  $k_{s,\text{Frac,Lewis}} = 0.1$  is recommended by Lewis et al. [28] for stacked screens. In other cases, we use RPN segments to add and subtract appropriate conduction estimates to HX and other \*X heats, yielding “net” heats as RPN results. In the early stages of a design, such RPNs can keep wall thickness proportional to STK\*\* diameter, making this heat leak track the reality of ASME pressure-vessel design.

We model black-body radiation heat leaks along pulse tubes and thermal buffer tubes using RPNs, adding them to DELTAEC’s built-in \*X calculations to obtain “net” heats. We assume that the walls of such tubes are fairly good infrared reflectors and the flow straighteners at the ends are fairly good black-body surfaces, so we use  $\sigma_{\text{Stefan-Boltzmann}} A (T_H^4 - T_C^4) / 2$  where  $A$  is the inside area of the STKDUCT and the factor of 2 accounts very roughly for the “fairly good” surfaces. We neglect black-body radiation within STKSCREENs, which are optically opaque. We still debate whether and how to account for black-body radiation in standing-wave stacks, where the long aspect ratio of the channels gives radiation ample opportunity for absorption and re-emission at intermediate temperatures.

We also account for unavoidable heat flow through insulation layers wrapped around components, with RPNs or separate spreadsheets.

In systems with toroidal topology (e.g., TASHE [40, 5]), we suppress Gedeon streaming, or account for it with nonzero  $\dot{N}$ .

#### 9.1.4. Geometry

As a design evolves towards its final state, we add more and more geometrical detail to the DELTAEC model. Even including the short gaps between regenerators and heat exchangers, or accounting for weld shrinkage that shortens such gaps, is not too much attention to detail.<sup>1</sup> When shop drawings are being prepared, we modify the DELTAEC model to reflect machine-shop realities (e.g., inch-based dimensions; available drill sizes). While assembling parts, we often check measured dimensions against the DELTAEC model one more time.

#### 9.1.5. Actual operating conditions

Actual operating conditions are seldom identical to the final “design” condition, especially for ambient temperature. When comparing experimental results with DELTAEC calculations to decide whether hardware is working acceptably well, we always use actual, experimental conditions for parameters like ambient temperature. (See also Section 5.3 in this Users Guide, and Section 9.4 in Ref. [13], to consider what variables to regard as “given.”)

In high-power systems, different parts of an “ambient”-temperature resonator can have significantly different temperatures. We often use RPN expressions like “inp =Tm” or “Tm inp + =Tm” to make our DELTAEC models follow this reality.

#### 9.1.6. Thermophysical properties

DELTAEC’s built-in thermophysical properties may not be accurate in the temperature (or pressure) range of interest for a new design. If we are going to work far from room temperature and/or with a new gas or solid, we check DELTAEC’s gas and solid properties against some reliable reference, and create our own `<.tpf>` file if necessary.

### 9.2. Component conservatism

We rely on best-estimate “headroom” associated with every segment in the DELTAEC models wherever our previous experience shows that experimental performance falls a little short of the built-in first-principles calculations.

#### 9.2.1. Turbulence

We habitually use a value of  $5 \times 10^{-4}$  for the relative surface roughness in DUCTs and CONEs, even when it is greater than the actual roughness, because this gives results that fit a lot of

---

<sup>1</sup>If a heat exchanger’s `GasA/A` is small and its pores are large compared with those of the adjacent `STK**`, a gap is absolutely necessary to prevent  $U_1$  from getting crowded into a reduced area in the beginning of the `STK**`. DELTAEC’s naive one-dimensional calculations do not show the user the need for such gaps.



$\dot{N} = 0$  data over the past 15 years. When steady flow is superimposed, we use the “average” turbulence setting, which is consistent with halfway between the two lines in Fig. 9 in Ref. [42].

We use `MINOR` segments generously wherever appropriate, e.g., corner turning, `TBRANCHs` and `UNIONs`, and area changes (except very gently tapered cones), with Idelchik’s book [52] for estimates of the  $K$  coefficients. We even check the need for `MINOR` segments at transitions between heat exchangers and open ducts, where abrupt changes in `GasA/A` occur. In toroidal `TASHEs` [40, 5], we might include almost a dozen `MINOR` segments.

### 9.2.2. Adiabatic–isothermal mixing

We use `JOIN` segments wherever appropriate, e.g., on both ends of pulse tubes, thermal buffer tubes, and hx-regen-hx packages. `JOIN` segments are usually negligible around hx-stack-hx packages in standing-wave systems, but we always check anyway.

We know that some of the same dissipative physics is present if the gap between a heat exchanger and an adjacent regenerator or stack is an appreciable fraction of the volume-flow-rate amplitude and is spatially stratified, so we strive to keep such gaps small and/or well-mixed by jets.

### 9.2.3. Stacked screen regenerators

We determine the “nominal” hydraulic radius  $r_h$  of a stacked-screen regenerator using the manufacturer’s specified wire diameter, the volumetric porosity determined by weighing the complete regenerator, and Eq. (10.118). We have never seen a pressure drop across a stacked-screen regenerator that is smaller than `DELTAEC` predicts using this nominal  $r_h$ . Our measured pressure drops have ranged from 0% to 15% above `DELTAEC` predictions using the nominal value. Conservatively taking the upper half of that range, and noting that the pressure drop goes like  $r_h^{-2}$ , we usually set the `DELTAEC`  $r_h$  equal to 0.93–0.96 times the nominal value.

We set `ksFrac` equal to 0.1 to account for screen-to-screen conduction, as recommended by Ref. [28], unless we are using this parameter to account for pressure-vessel-wall conduction [see Eq. (9.1)] as well as the screen-to-screen conduction.

### 9.2.4. Pulse tubes and thermal buffer tubes

We know that jet-driven streaming in pulse tubes and thermal buffer tubes can carry a lot of heat, so we always use screen flow straighteners on both ends, using `SX` segments to model their acoustic characteristics in `DELTAEC` and using the guidance near Eq. (7.86) in Ref. [13] to choose the screen size and number of layers. Since this design process does not fill us with confidence, we sometimes double the number of layers, because it is often easier to remove unnecessary layers later than to add layers later.

Rayleigh streaming can also carry a lot of heat in these tubes. In the experiment described in Ref. [31], very careful tuning of the phase of  $Z$  in a pulse tube could bring that streaming heat leak to zero, but we generally don’t expect to keep pulse tubes so accurately tuned at all times. Thus, we try to work as close as possible to the zero-Rayleigh-streaming condition

described in Ref. [29] while avoiding cone angles that might create flow separation, and even then we usually deduct 10% of  $\dot{E}$  at the cold heat exchanger from DELTAEC’s raw prediction of cooling power in pulse-tube refrigerators, and add 10% of  $\dot{E}$  at the hot heat exchanger to the heater power in engines having thermal buffer tubes. (This correction also affects the ambient heat exchangers at the other ends of the tubes.) It is not unreasonable to encode the math described in Ref. [29] in RPN segments (it takes more than one!) to keep an eye on this issue in a model with a pulse tube or thermal buffer tube.

Large-aspect-ratio refrigerator regenerators can suffer from regenerator internal streaming, as described in Ref. [53], which must be accounted for, even if only approximately.

(See also “Adiabatic–isothermal mixing” and “Heat leaks—black-body radiation” above.)

### 9.2.5. Standing-wave engine stacks

At low amplitudes, standing-wave engines have worked in agreement with DELTAEC predictions, so we trust that the calculations are correct to lowest order (i.e., energy expressions are correct to second order in acoustic amplitude). However, at higher amplitude, measurements on standing-wave engines are always worse than calculations: higher hot temperature  $T_H$  and higher heater power  $\dot{Q}_H$  for a given  $p_1$  and  $Z$ . Thus, until research brings understanding to this situation, we build some “headroom” into our DELTAEC designs of standing-wave engines.

Figures 2 and 3 in Ref. [25] show that the extra, unaccounted-for heater power is proportional to  $|p_1|^3$  and the extra, unaccounted-for hot temperature is proportional to  $|p_1|^2$ . In the absence of any evidence to the contrary, we assume that these dependences on  $|p_1|$  are applicable to all standing-wave engines.

Figure 9.1 summarizes our experience with several standing-wave engines, with all data extrapolated to  $|p_1|/p_m = 0.10$  at the hot end of the stack using the pressure dependences described in the previous paragraph. The honeycomb data (small symbols) are more reliable than the parallel-plate data (large triangles), so we trust the honeycomb fit (dashed curves) more than the parallel-plate fit (solid curves). We believe that putting some headroom in standing-wave designs as suggested by Fig. 9.1 and the pressure dependences described above is much better than putting in no headroom at all.

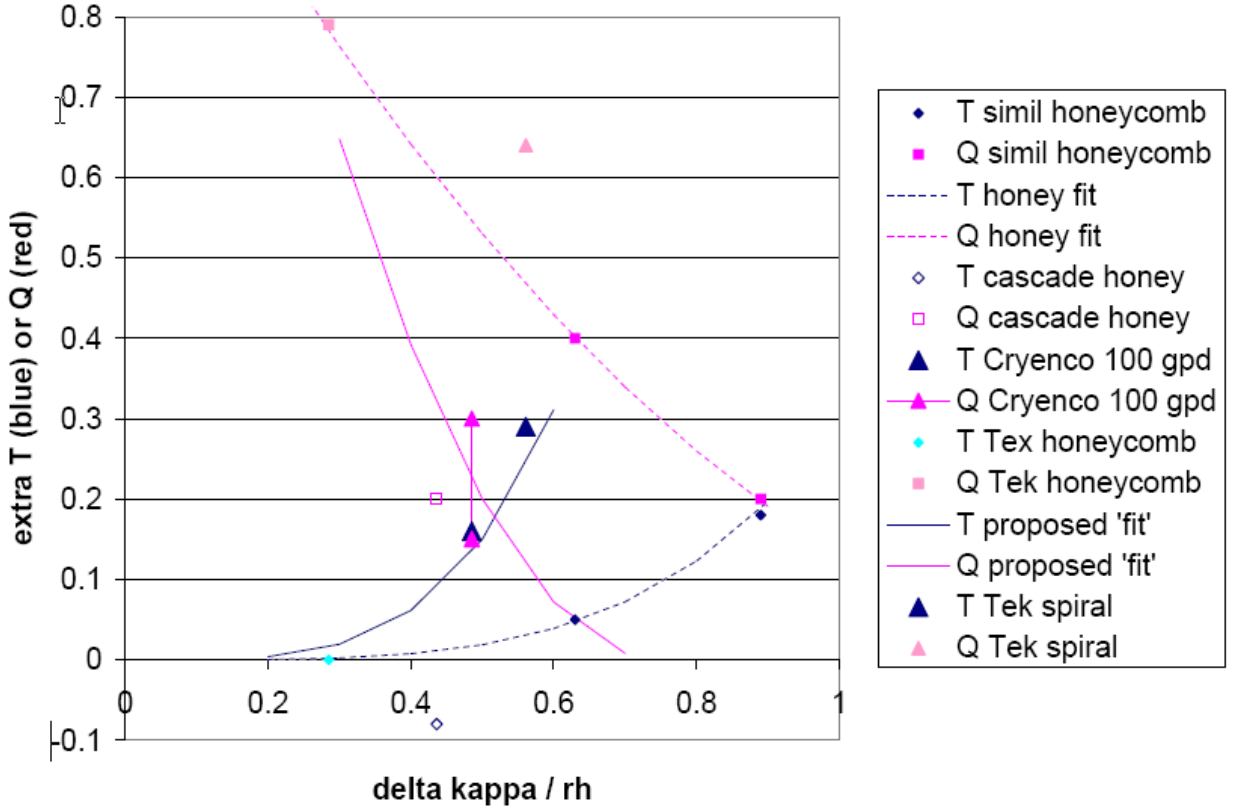


Figure 9.1: The vertical axis shows the experimentally observed extra  $T_H$  (blue) or  $\dot{Q}_H$  (pink), in excess of DELTAEC’s calculations, divided by DELTAEC’s values, for standing-wave engines that we have used. The horizontal axis is  $\delta_\kappa$  at the ambient end of the stack, divided by  $r_h$ . All data are scaled to  $|p_1|/p_m = 0.10$  at the hot end of the stack, using  $|p_1|^3$  scaling for extra  $\dot{Q}_H$  and  $|p_1|^2$  scaling for extra  $T_H$ . Data from three honeycomb (STKCIRC) stacks are shown: “simil honeycomb” is Ref. [25], “cascade honey” is Ref. [19], and “Tex honeycomb” is Ref. [54]. Unpublished data from two spiral (STKSLAB) stacks are shown: “Cryenco 100 gpd” is from the engine shown in Fig. 4 of Ref. [55], corroborated by measurements on a half-scale model at Los Alamos, and with the range of data for heater power indicating low acoustic load (more excess heat) vs. heavy acoustic load (less excess heat), and “Tek spiral” is from the engine described in Ref. [54] but with the honeycomb stacks replaced by spiral stacks as shown in Fig. 1.11 in Ref. [13].

# 10. PHYSICAL SEGMENTS

All of DELTAEC's physically meaningful segment types are listed in this chapter, organized by function. Logistical and management segments appear in the next chapter. An alphabetical index of all segment types is in the Segment Index at the end of the Users Guide.

## 10.1. Ducts and cones

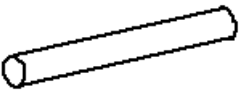
### 10.1.1. DUCT

#### General description:

Use DUCT for circular ducts of any diameter, and for large ducts (i.e., with  $r_h \gg \delta_\nu, \delta_\kappa$ ) of any cross-sectional shape by giving suitable area and perimeter. Mean temperature is independent of  $x$ , while  $p_1$  and  $U_1$  evolve with  $x$ .

DELTAEC's turbulence algorithm is controlled through optional parameter **d**, the relative roughness  $\varepsilon$ , whose definition for steady flow can be found in fluid-mechanics textbooks [56]: roughness height divided by pipe diameter. We have found that  $\varepsilon = 0.0005$  works well in high-amplitude acoustics, even if the geometrical roughness is smaller than this value. Omitting this parameter forces the calculation to be laminar at all Reynolds numbers, no matter how large.

(See also **STKDUCT** in Section 10.6.1, which allows a temperature gradient along a duct in boundary-layer approximation.)



#### Input variables:

**Area** (m<sup>2</sup>) Cross-sectional area  $A$  available to the gas, based on inside dimensions of the DUCT.

**Perim** (m) Perimeter  $\Pi$  of the inside cross section described above.

**Length** (m) The length  $\Delta x$  of the DUCT.

**Laminar\turbulent** Whenever the laminar box is checked, the calculations of  $dp_1/dx$  and  $dp_{2,0,HL}/dx$  in the DUCT are based on noninteracting flow, using laminar flow for  $dp_1/dx$  and either laminar or turbulent flow for  $dp_{2,0,HL}/dx$ , depending on the steady-flow Reynolds number. The laminar condition is identified in the `<.out>` file as a blank line where input **d** would otherwise be.

**Srough** This input variable appears if the turbulent box is checked in the Optional Parameters dialog. **Srough** is  $\varepsilon$ , the surface roughness inside the DUCT, relative to the diameter. This is called “relative roughness” in many fluid-mechanics books, for steady flow. However, for oscillating flow, we usually regard **Srough** as a fitting parameter, with a typical value of  $\varepsilon = 5 \times 10^{-4}$ . Larger values yield higher turbulent dissipation of  $\dot{E}$ .

**Optimistic/Normal/Pessimistic** These choices appear if the turbulence box is checked in the Optional Parameters dialog and if the “Steady flow” box is checked in the **BEGIN** segment. These choices control the nonlinear interaction between oscillating flow and steady flow in the DUCT. “Optimistic” and “Pessimistic” refer to the two calculation schemes described in Ref. [42], and “Normal” is the average of the two. For the optimistic case, the turbulent steady and turbulent oscillating flows are assumed to be independent, simply superimposed. For the pessimistic case, the turbulent steady flow and turbulent oscillating flow are assumed to interact with each other and to increase each other’s resistive effects according to the quasi-steady approximation. In the `<.out>` file, Optimistic/Normal/Pessimistic is recorded with **SrOptm**, **Srough**, or **SrPess** to indicate the name of  $\varepsilon$ .

**Solid** The solid material on the inside surface of the DUCT. (This is used only in the calculation of  $\epsilon_s$ , which is seldom significant in DUCTs.)

### Potential targets:

None.

### Master–slave links:

The perimeter of a DUCT can be slaved to its area to keep the cross-sectional shape the same when area is varied.

The length of a DUCT can be slaved to that of another segment to keep the total length constant.

(For an introduction to master–slave links, and a complete listing of linking options for all segments, see Section 2.7.2.)

### Format and examples:

Many examples of DUCT occur throughout Chapters 2, 5, and 6.

### Calculation of $p_1(x)$ and $U_1(x)$ with $\dot{N} = 0$ :

In DUCTs with no superimposed steady flow,  $p_1$  and  $U_1$  evolve according to

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A}U_1, \quad (10.1)$$

$$\frac{dU_1}{dx} = -\frac{iA\omega}{\rho_m a^2} \left( 1 + \frac{\gamma-1}{1+\epsilon_s} f_\kappa \right) p_1, \quad (10.2)$$

which are equivalent to

$$\left(1 + \frac{\gamma - 1}{1 + \epsilon_s} f_\kappa\right) p_1 + \frac{a^2}{\omega^2} (1 - f_\nu) \frac{d^2 p_1}{dx^2} = 0. \quad (10.3)$$

In laminar DUCTs of length  $\Delta x$ , these are equivalent to

$$\begin{aligned} p_{1,\text{out}} &= p_{1,\text{in}} \cos k\Delta x - \frac{i\omega\rho_m}{(1 - f_\nu)kA} U_{1,\text{in}} \sin k\Delta x, \\ U_{1,\text{out}} &= U_{1,\text{in}} \cos k\Delta x - \frac{i(1 - f_\nu)kA}{\omega\rho_m} p_{1,\text{in}} \sin k\Delta x, \end{aligned} \quad (10.4)$$

with complex wave number  $k$  given by

$$k = \frac{\omega}{a} \sqrt{\frac{1 + (\gamma - 1)f_\kappa/(1 + \epsilon_s)}{1 - f_\nu}}. \quad (10.5)$$

In DUCTs, we define the effective radius to be  $r_0 = 2A/\Pi$ . (For circular cross sections, this is indeed the radius of the circle. For other cross sections, it is twice the hydraulic radius.) In narrow DUCTs, for  $r_0/\delta_j < 25$ ,  $f_\kappa$  and  $f_\nu$  are calculated using complex Bessel functions

$$f_j = \frac{2J_1[(i - 1)r_0/\delta_j]}{(i - 1)(r_0/\delta_j)J_0[(i - 1)r_0/\delta_j]} \quad \text{for } j = \kappa \text{ or } \nu. \quad (10.6)$$

Where  $r_0/\delta_j > 30$ , the boundary-layer approximation is used:

$$f_j = (1 - i)\Pi\delta_j/2A \quad \text{for } j = \kappa \text{ or } \nu. \quad (10.7)$$

For intermediate values, linear interpolation is used to make a smooth match between the two regimes. While the narrow-duct solution assumes a circular cross section, the shape of the duct is irrelevant in the boundary-layer approximation. A square duct with dimensions much larger than the penetration depth can be modeled simply by choosing  $\Pi = 4\sqrt{A}$ , for example.

When turbulent,  $p_1$  and  $U_1$  are numerically integrated according to Eqs. (10.1) and (10.2) above, with  $f_\nu$  and  $f_\kappa$  modified as described below to account for the turbulence. DELTAEC's turbulence algorithm is described in detail in Chapter 7.2 of Ref. [13] and is based on the quasi-steady approximation: that turbulent oscillatory flow is described by the Moody friction factor at each instant of time during the oscillatory flow. (See any engineering fluid mechanics textbook, such as Ref. [56], to review the Moody friction factor as a function of Reynolds number and relative roughness of the pipe wall.) This assumption must be excellent in the low-frequency limit, in which  $r_0/\delta_\nu \rightarrow 0$ . This limit is approached in the inertances of the lowest-power pulse-tube refrigerators. We have learned by experience with a few large systems that the assumption is reasonable for large  $r_0/\delta_\nu$ , which is of interest in the resonators of standing-wave thermoacoustic systems. For experimental validation of the assumption for intermediate  $r_0/\delta_\nu$ , see Ref. [15].

To account for turbulence, DELTAEC increases the resistive component of the pressure gradient, and hence the viscous power dissipation, by

$$m_\nu = \frac{\delta_\nu^2 N_R [f_M - (1 - 9\pi/32) N_R df_M/dN_R]}{24\pi r_h^2 \operatorname{Re} [i / (1 - f_\nu)]}. \quad (10.8)$$

DELTAEC evaluates the Moody friction factor  $f_M$  and its derivative with respect to Reynolds number,  $df_M/dN_R$ , as functions of Reynolds number and  $\varepsilon$  using the iterative expression

$$\frac{1}{\sqrt{f_M}} = 1.7385 - 2 \log_{10} \left( 2\varepsilon + \frac{18.574}{N_R \sqrt{f_M}} \right), \quad (10.9)$$

which is a remarkably good approximation [56, 57]. (In our own work at Los Alamos, we have set roughness equal to  $5 \times 10^{-4}$  for many years, as representative of most turbulent acoustic losses in ducts, even when the actual surface roughness over pipe diameter is smaller than this value.) DELTAEC decreases the inertial pressure gradient by

$$m'_\nu = \left( \frac{1 - \delta_\nu/r_0}{1 - \delta_\nu/m_\nu r_0} \right)^2 \quad (10.10)$$

to correct approximately for the steeper velocity gradient at the wall, which increases the effective area open to gas that contributes to inertial effects. The factors  $m_\nu$  and  $m'_\nu$  are applied using

$$\operatorname{Im} \left[ \frac{1}{1 - f_\nu} \right] = m_\nu \operatorname{Im} \left[ \frac{1}{1 - f_{\nu, \text{laminar}}} \right], \quad (10.11)$$

$$\operatorname{Re} \left[ \frac{1}{1 - f_\nu} \right] = m'_\nu \operatorname{Re} \left[ \frac{1}{1 - f_{\nu, \text{laminar}}} \right], \quad (10.12)$$

$$f_\nu = 1 - \left( \operatorname{Re} \left[ \frac{1}{1 - f_\nu} \right] + i \operatorname{Im} \left[ \frac{1}{1 - f_\nu} \right] \right)^{-1} \quad (10.13)$$

DELTAEC also multiplies  $\delta_\kappa$  by  $m_\nu$  before using  $\delta_\kappa$  in Eq. (10.6) or (10.7), in an attempt to account very approximately for changes in thermal relaxation losses due to increased heat transfer.

At low enough velocities,  $m_\nu \rightarrow 1$  and DELTAEC reverts to a laminar calculation. The  $m_\nu = 1$  boundary between the laminar and turbulent zones in DELTAEC occurs roughly at

$$N_R \simeq 2000 \text{ for } r_0/\delta_\nu < 2, \quad (10.14)$$

$$\frac{N_R}{2r_h/\delta_\nu} \simeq 1000 \text{ for } r_0/\delta_\nu > 2. \quad (10.15)$$

Whether laminar or turbulent, and in both small and large DUCTs, the boundary-layer approximation is used for the effect of the solid on thermal-hysteresis loss:

$$\epsilon_s = \left( \frac{k\rho_m c_p}{k_s \rho_s c_s} \right)^{1/2}. \quad (10.16)$$

## Calculation of $p_1(x)$ and $U_1(x)$ with $\dot{N} \neq 0$ :

Nonzero  $\dot{N}$  in a DUCT creates a Doppler-shift effect on  $p_1$  and  $U_1$ , an increased resistive effect on  $p_1$  if the flows are turbulent, and a time-averaged resistive head loss. Whether the flows are treated as laminar or as one of three varieties of turbulent is controlled by the user. The pressure and volume flow rate change with  $x$  according to

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{A(1-f_\nu)}U_1 + \frac{i\omega m\dot{N}}{\rho_m a^2 A} \frac{1 + (\gamma - 1)f_\kappa / (1 + \epsilon_s)}{(1 - f_\nu)} p_1, \quad (10.17)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A}{\rho_m a^2} \frac{1 + (\gamma - 1)f_\kappa}{1 + \epsilon_s} p_1 + \frac{i\omega m\dot{N}}{\rho_m a^2 A(1 - f_\nu)} U_1, \quad (10.18)$$

which can be derived using standard results of thermoacoustics combined with Eqs. (11.1.3) in Morse and Ingard's *Theoretical Acoustics*, Ref. [33]. (In CONEs,  $A$  is a function of  $x$ .)

Reference [42] describes two approaches to the calculation of the resistive effects on  $f_\nu$ ,  $f_\kappa$ , and  $dp_{2,0,HL}/dx$  if the flow is turbulent. These two approaches, which we call "optimistic" and "pessimistic," bracket the experimental results in that paper. In DELTAEC the user can select either of these two approaches, or their average. Until more physics is learned, we recommend selecting the average by checking the "Normal" box, because the experimental results in Ref. [42] were roughly the average of the optimistic and pessimistic calculations.

For the optimistic case, the turbulent steady and turbulent oscillating flows are assumed to be independent, simply superimposed. Turbulent  $f_\nu$  and  $f_\kappa$  are calculated with Eqs. (10.8)–(10.10).

For the pessimistic case, the turbulent steady flow and turbulent oscillating flow are assumed to interact with each other and to increase each other's resistive effects, as described in the appendix of Ref. [42]. In this case,  $f_\nu$  and  $f_\kappa$  are calculated according to the procedure described near Eqs. (10.8)–(10.10), but with the turbulent multiplier  $m_\nu$  enhanced by the steady flow according to

$$m_\nu = \frac{\delta_\nu N_{R,\max}}{16r_h} \left[ |\varepsilon_{2,0}| f_{M,\max} + \frac{(3 - 2|\varepsilon_{2,0}|)(1 - 2|\varepsilon_{2,0}|)}{8(1 + |\varepsilon_{2,0}|)} \frac{df_M}{dN_R} N_{R,\max} \right], \quad |\varepsilon_{2,0}| \geq 1, \quad (10.19)$$

$$\begin{aligned} &= \frac{\delta_\nu N_{R,\max}}{12\pi r_h} \left\{ \left[ \left(1 + \frac{\varepsilon_{2,0}^2}{2}\right) \sqrt{1 - \varepsilon_{2,0}^2} + \frac{3}{2} \varepsilon_{2,0} \sin^{-1} \varepsilon_{2,0} \right] f_{M,\max} \right. \\ &\left. - \left[ \left(1 + \frac{\varepsilon_{2,0}^2}{2}\right) \sqrt{1 - \varepsilon_{2,0}^2} + \frac{3}{2} \varepsilon_{2,0} \sin^{-1} \varepsilon_{2,0} - \frac{9\pi}{32} \frac{1 + 4\varepsilon_{2,0}^2}{1 + |\varepsilon_{2,0}|} \right] \frac{df_M}{dN_R} N_{R,\max} \right\}, \quad |\varepsilon_{2,0}| \leq 1, \end{aligned} \quad (10.20)$$

where the friction factor and its slope are based on the peak Reynolds number

$$N_{R,\max} = \frac{\left(m|\dot{N}_{2,0}|/\rho_m + |U_1|\right) 4r_h\rho_m}{A\mu} \quad (10.21)$$

and  $\varepsilon_{2,0} = m\dot{N}_{2,0}/\rho_m |U_1|$ . The derivations of these expressions can be found in the appendix of Ref. [42].



## Calculation of $p_{2,0,HL}(x)$ with $\dot{N} \neq 0$ :

How the flow is treated depends on the roughness parameter in the DUCT. There are three cases: optimistic, pessimistic, and “normal,” which is the average of the other two.

For the optimistic case, the turbulent steady and turbulent oscillating flows are assumed to be independent, simply superimposed. The steady-flow pressure drop is calculated using

$$\frac{dp_{2,0,HL}}{dx} = -\text{sign}(\dot{N}_{2,0}) \frac{1}{2} \rho_m \left( \frac{m \dot{N}_{2,0}}{\rho_m A} \right)^2 \frac{f_{\text{Moody,steady}}}{4r_h}. \quad (10.22)$$

The steady-flow Moody friction factor  $f_{\text{Moody,steady}}$  is calculated with a Reynolds number based on the steady velocity:

$$N_{R,2,0} = \frac{m |\dot{N}_{2,0}| 4r_h}{A \mu}. \quad (10.23)$$

For the pessimistic case, the turbulent steady flow and turbulent oscillating flow are assumed to interact with each other and to increase each other’s resistive effects, as described in the appendix of Ref. [42] ignoring any dependence on  $p_1$  and in Chapter 14 including the lowest-order dependence on  $p_1$ . From Chapter 14, the steady-flow pressure drop is calculated using

$$\begin{aligned} \frac{dp_{2,0}}{dx} &= -\text{sign}(\dot{N}_{2,0}) \frac{m^2 \dot{N}_{2,0}^2}{8 \rho_m r_h A^2} \left[ \left( 1 + \frac{1}{2 \varepsilon_{2,0}^2} + \frac{|p_1| \cos \phi}{\gamma p_m \varepsilon_{2,0}} \right) f_{M,\max} \right. \\ &\quad \left. - \frac{2 \varepsilon_{2,0}^2 - 2 |\varepsilon_{2,0}| + 1}{2 \varepsilon_{2,0}^2 (1 + |\varepsilon_{2,0}|)} \frac{df_M}{dN_R} N_{R,\max} \right], \quad |\varepsilon_{2,0}| \geq 1, \quad (10.24) \\ &= -\frac{\rho_m |U_1|^2}{8 \pi r_h A^2} \left\{ \left[ (1 + 2 \varepsilon_{2,0}^2) \sin^{-1} \varepsilon_{2,0} + 3 \varepsilon_{2,0} \sqrt{1 - \varepsilon_{2,0}^2} \right. \right. \\ &\quad \left. \left. + \frac{2 |p_1| \cos \phi}{\gamma p_m} \left( \varepsilon_{2,0} \sin^{-1} \varepsilon_{2,0} + \frac{2 + \varepsilon_{2,0}^2}{3} \sqrt{1 - \varepsilon_{2,0}^2} \right) \right] f_{M,\max} \right. \\ &\quad \left. + \left[ \pi \varepsilon_{2,0} \frac{3/2 + \varepsilon_{2,0}^2}{1 + |\varepsilon_{2,0}|} - 3 \varepsilon_{2,0} \sqrt{1 - \varepsilon_{2,0}^2} - (1 + 2 \varepsilon_{2,0}^2) \sin^{-1} \varepsilon_{2,0} \right] \frac{df_M}{dN_R} N_{R,\max} \right\}, \\ &\quad |\varepsilon_{2,0}| \leq 1, \quad (10.25) \end{aligned}$$

where  $\varepsilon_{2,0} = m \dot{N}_{2,0} / \rho_m |U_1|$ .

## Calculation of $T_m(x)$ and $\dot{H}$ :

$T_m$  is not affected by DUCTs.

The exiting energy flow is left as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}}$  in insulated mode, described in Section 11.1.2. This essentially assumes that the side walls are thermally insulated, so the heat generated by acoustic power dissipation finds its way to a heat exchanger somewhere else. If several DUCTs (and/or CONEs) are strung together in insulated mode, the power dissipated in all of them must show up at one or two nearby heat exchangers.

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$  in “anchored” mode, described in Section 11.1.2. This essentially assumes that the duct wall is thermally anchored, e.g., by a water bath, so heat generated by acoustic-power dissipation is carried away locally. Thermoacoustic heat transport in the boundary layer along the perimeter, which in fact contributes a small difference between  $\dot{H}_{2,k}$  and  $\dot{E}$  in ducts, is neglected in “anchored” mode.

### 10.1.2. CONE

#### General description:

Use **CONE** to find the evolution of  $p_1$  and  $U_1$  with  $x$  in tapered channels of any cross-sectional shape (e.g., square, circular) by giving suitable areas and perimeters. The perimeter is linearly interpolated between its initial and final values, and the area is quadratically interpolated between its initial and final values. Use this segment only when  $T_m$  is independent of  $x$ . (See **STKCONE** in Section 10.6.1, to allow nonzero  $dT_m/dx$ .) The calculation is identical to that for a **DUCT** if the initial and final areas are equal and the initial and final perimeters are equal. As discussed below, the laminar calculation is accurate for circular cones of any diameter, and for large cones (i.e., with  $r_0 \gg \delta_\nu, \delta_\kappa$ ) of any cross-sectional shape.

In a **CONE**, **DELTAEC**’s turbulence algorithm is controlled through optional parameter **f**, the relative roughness  $\varepsilon$ , whose definition for steady flow can be found in fluid-mechanics textbooks [56]: roughness height divided by pipe diameter. We have found that  $\varepsilon = 0.0005$  works well in high-amplitude acoustics, even if the geometrical roughness is smaller than this value. Omitting this parameter forces the oscillatory calculation to be laminar at all Reynolds numbers.

At high enough amplitudes, we believe that a steeply tapered **CONE** experiences separation during the half of the cycle when the flow goes from small toward large radius, just like the separation at high Reynolds number in a too-steeply-tapered diffuser. **DELTAEC** does not warn of this condition. To account for such separation, the user could include minor loss, described in Section 10.2.4.

#### Input variables:

**AreaI** (m<sup>2</sup>) Cross-sectional area  $A_I$  available to the gas at the initial end of the **CONE**, based on inside dimensions.

**PerimI** (m) Perimeter  $\Pi_I$  of the initial inside cross section described above.

**Length** (m) The length  $\Delta x$  of the **CONE**.

**AreaF** (m<sup>2</sup>) Cross-sectional area  $A_F$  available to the gas at the final end of the **CONE**, based on inside dimensions.

**PerimF** (m) Perimeter  $\Pi_F$  of the final inside cross section described above.

**Laminar\turbulent** Whenever the laminar box is checked, the calculations of  $dp_1/dx$  and  $dp_{2,0,HL}/dx$  in the **CONE** are based on noninteracting, laminar flow. The laminar condition is identified in the `<.out>` file as a blank line where parameter **f** would otherwise be.

**Srough** This input parameter appears if the turbulent box is checked in the Optional Parameters dialog. Srough is  $\varepsilon$ , the surface roughness inside the **CONE**, relative to the diameter. This is called “relative roughness” in many fluid-mechanics books, for steady flow. However, for oscillating flow, we often regard Srough as a fitting parameter, with a typical value of  $5 \times 10^{-4}$ . Larger values yield higher turbulent dissipation of  $\dot{E}$ .

**Optimistic/Normal/Pessimistic** These choices appear if the turbulence box is checked in the Optional Parameters dialog and if the “Steady flow” box is checked in the **BEGIN** segment. These choices control the nonlinear interaction between oscillating flow and steady flow in the **CONE**. “Optimistic” and “Pessimistic” refer to the two calculation schemes described in Ref. [42], and “Normal” is the average of the two. For the optimistic case, the turbulent steady and turbulent oscillating flows are assumed to be independent, simply superimposed. For the pessimistic case, the turbulent steady flow and turbulent oscillating flow are assumed to interact with each other and to increase each other’s resistive effects according to the quasi-steady approximation. In the `<.out>` file, Optimistic/Normal/Pessimistic is recorded with **SrOptm**, **Srough**, or **SrPess** to indicate the name of  $\varepsilon$ .

**Solid** The solid material on the inside surface of the **CONE**. (This is used only in the calculation of  $\varepsilon_s$ , which is seldom significant in **CONES**.)

### Potential targets:

None.

### Master–slave links:

The initial perimeter of a **CONE** can be slaved to the initial area, to keep the cross-sectional shape the same when area is changed.

The final perimeter of a **CONE** can be slaved to the final area to keep the cross-sectional shape the same when area is changed.

Both perimeters of a **CONE** can be slaved to their respective areas to keep the cross-sectional shapes the same when either or both areas are changed.

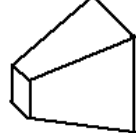
The length and both perimeters of a **CONE** can be slaved to its two areas, to keep the cross-sectional shapes of the ends *and* the wall taper angle constant when either or both areas are changed.

The length of a **CONE** can be slaved to that of another segment to keep the total length constant.

(For an introduction to master–slave links, and a complete listing of possible links for all segments, see Section 2.7.2.)

## Format and examples:

Examples of the use of CONE appear throughout Chapters 2, 5, and 6.



## Calculation of $p_1(x)$ and $U_1(x)$ with $\dot{N} = 0$ :

In CONES,  $p_1$  and  $U_1$  evolve according to

$$\begin{aligned} \frac{dp_1}{dx} &= -\frac{i\omega\rho_m}{(1-f_\nu)A}U_1, \\ \frac{dU_1}{dx} &= -\frac{iA\omega}{\rho_m a^2} \left(1 + \frac{\gamma-1}{1+\epsilon_s}f_\kappa\right)p_1, \end{aligned} \quad (10.26)$$

which are equivalent to the Webster horn equation [58] with losses included:

$$\left(1 + \frac{\gamma-1}{1+\epsilon_s}f_\kappa\right)p_1 + \frac{a^2}{\omega^2} \frac{1}{A} \frac{d}{dx} \left[(1-f_\nu)A \frac{dp_1}{dx}\right] = 0. \quad (10.27)$$

The perimeter varies linearly from its initial value to its final value, and area varies quadratically:

$$\Pi(x) = \Pi_I + (\Pi_F - \Pi_I) \frac{x - x_o}{\Delta x}, \quad (10.28)$$

$$A(x) = \left[ \sqrt{A_I} + \left( \sqrt{A_F} - \sqrt{A_I} \right) \frac{x - x_o}{\Delta x} \right]^2, \quad (10.29)$$

where  $x_o$  is the  $x$  coordinate of the initial end of the CONE. Hence, CONES that are circular at both ends have circular cross sections everywhere in between, with diameter varying linearly with axial position.

In CONES, we define the effective radius to be  $r_o(x) = 2A/\Pi$ . (For circular cross sections, this is indeed the radius of the circle. For other cross sections, it is twice the hydraulic radius.) For  $r_o/\delta_j < 25$ ,  $f_\kappa$  and  $f_\nu$  are calculated using complex Bessel functions:

$$f_j = \frac{2J_1[(i-1)r_o/\delta_j]}{(i-1)(r_o/\delta_j)J_0[(i-1)r_o/\delta_j]} \text{ for } j = \kappa \text{ or } \nu. \quad (10.30)$$

Where  $r_o/\delta_j > 30$ , the boundary-layer approximation is used:

$$f_j = (1-i)\delta_j/r_o. \quad (10.31)$$

For intermediate values, linear interpolation is used to make a smooth match between the two regimes. While the narrow solution assumes a circular cross section, the cross-sectional shape of the cone is irrelevant in the boundary-layer approximation.

In CONE, the effects of turbulence are calculated exactly the same as given above for DUCT.

In all cases, boundary-layer approximation is used for the effect of the solid:

$$\epsilon_s = \left( \frac{k\rho_m c_p}{k_s \rho_s c_s} \right)^{1/2}. \quad (10.32)$$

## Calculation of $p_1(x)$ , $U_1(x)$ , and $p_{2,0,HL}(x)$ with $\dot{N} \neq 0$ :

In CONE, these calculations are exactly the same as given above for DUCT, except that area  $A$  is a function of  $x$  in the equations for CONE.

## Calculation of $T_m$ and $\dot{H}$ :

$T_m$  is not affected by CONE. (But see also STKCONE in Section 10.6.1, which allows nonzero  $dT_m/dx$ .)

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}}$  in insulated mode, described in Section 11.1.2. This essentially assumes that the side walls are thermally insulated, so the heat generated by acoustic power dissipation is deposited in an adjacent heat exchanger. If several CONEs (and/or DUCTs) are strung together in insulated mode, the power dissipated in all of them should show up in a nearby heat exchanger.

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$  in anchored mode, described in Section 11.1.2. This mode essentially assumes that the cone wall is thermally anchored, e.g., by a water bath, so heat generated by acoustic-power dissipation is carried away externally. Thermoacoustic heat transport along the perimeter, which in fact contributes a small difference between  $\dot{H}_{2,k}$  and  $\dot{E}$  in CONEs, is neglected.

## 10.2. Lumped elements

### 10.2.1. COMPLIANCE

#### General description:

A COMPLIANCE is a lumped acoustic volume element, with adiabatic volumetric compressibility and surface thermal-hysteresis dissipation.

#### Input variables:

**Area** ( $\text{m}^2$ ) The surface area  $S$  inside the COMPLIANCE.

**Volume** ( $\text{m}^3$ ) The volume  $V$  inside the COMPLIANCE.

**Solid** The solid material on the inside surface of the COMPLIANCE. (This is used only in the calculation of  $\epsilon_s$ , which seldom differs significantly from its value for a solid with infinite conductivity and specific heat in COMPLIANCES.)

#### Potential targets:

None.

#### Master–slave links:

The surface area of a compliance can be slaved to its volume, so the shape remains the same when the volume is changed.

(For an introduction to master-slave links, and a complete listing of link options for all segments, see Section 2.7.2.)

### Format and examples:

```
COMPLIANCE  this one is a sphere, R = 0.01 m
0.1257     m2  Area
4.189e-3   m3  Volume
nickel     solid
```



Other examples of COMPLIANCE can be found in Sections 5.2 and 5.8.2.

### Calculations:

Pressure amplitude and mean temperature are unchanged by COMPLIANCE. Volume flow changes according to

$$U_{1,\text{out}} = U_{1,\text{in}} - i \frac{\omega p_1}{\rho_m a^2} \left[ V - i \frac{\gamma - 1}{1 + \epsilon_s} S \frac{\delta_\kappa}{2} \right], \quad (10.33)$$

with the thermal-hysteresis contribution of the solid calculated in boundary-layer approximation:

$$\epsilon_s = \left( \frac{k \rho_m c_p}{k_s \rho_s c_s} \right)^{1/2}. \quad (10.34)$$

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}}$  in insulated mode. This essentially assumes that the component is thermally insulated, so the heat generated by acoustic power dissipation must somehow be deposited elsewhere, either upstream or downstream, such as in an adjacent heat exchanger (or flowing out through a BEGIN or \*\*END segment).

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{E}$  in anchored mode. This is intended to model a component that is thermally anchored, e.g., by a water bath, so heat generated by acoustic power dissipation is carried away locally.

COMPLIANCE behaves the same whether  $\dot{N} = 0$  or not. Since  $T_m$  is unchanged in COMPLIANCE,  $\dot{H}_{\dot{N},\text{out}} = \dot{H}_{\dot{N},\text{in}}$ .

#### 10.2.2. SURFACE

### General description:

A SURFACE is a surface area with thermal-hysteresis dissipation. It always absorbs acoustic power. It has no volume. It is often used at the end of a DUCT, before a HARDEND. (See Fig. 10.1 for two other typical uses for SURFACE.)

### Input variables:

**Area** (m<sup>2</sup>) The surface area  $S$  exposed to oscillating pressure in the SURFACE segment, not accounted for in other segments nearby.

**Solid** The solid material on the exposed surface. (This is used only in the calculation of  $\epsilon_s$ , which seldom differs significantly from its value for a solid with infinite conductivity and specific heat on SURFACEs.)

## Potential targets; Master–slave links:

None.

## Format and examples:

SURFACE a surface with thermal hysteresis dissipation  
1.134e-3 m2 Area  
ideal solid



Examples of SURFACE also occur throughout Chapter 2 and Sections 5.2 and 5.5.

## Calculations:

A SURFACE does not affect temperature or pressure amplitude. Volume flow changes according to

$$U_{1,\text{out}} = U_{1,\text{in}} - \frac{\omega p_1}{\rho_m a^2} \frac{\gamma - 1}{1 + \epsilon_s} S \frac{\delta_\kappa}{2}. \quad (10.35)$$

(This is the same calculation as would occur in a COMPLIANCE with zero volume.) The influence of the solid on thermal hysteresis is calculated in boundary-layer approximation:

$$\epsilon_s = \left( \frac{k \rho_m c_p}{k_s \rho_s c_s} \right)^{1/2}. \quad (10.36)$$

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}}$  in insulated mode. This essentially assumes that the component is thermally insulated, so the heat generated by acoustic power dissipation must somehow be deposited elsewhere, either upstream or downstream, such as in an adjacent heat exchanger (or flow out through a BEGIN or \*\*END segment).

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$  in anchored mode. Thus, in a series of such anchored segments, heat generated by acoustic power dissipation is carried away locally.

SURFACE behaves the same whether  $\dot{N} = 0$  or not. Since  $T_m$  is unchanged in SURFACE,  $\dot{H}_{\dot{N},\text{out}} = \dot{H}_{\dot{N},\text{in}}$ .

### 10.2.3. IMPEDANCE

## General description:

An IMPEDANCE is a lumped series complex impedance.

## Input variables:

**Re(Z)** (Pa·s/m<sup>3</sup>) The real part of the series acoustic impedance  $Z$ . If the impedance has resistance, this number is positive.

**Im(Z)** (Pa·s/m<sup>3</sup>) The imaginary part of the series acoustic impedance  $Z$ . If the impedance has inductance  $L$ , this number is equal to  $\omega L$ .

## Potential targets:

None.

## Master–slave links:

In an IMPEDANCE,  $\text{Im}[Z]$  can be slaved to  $\text{Re}[Z]$  so  $|Z|$  remains constant when  $\text{Re}[Z]$  is changed. Note, however, that this becomes impossible if  $\text{Re}[Z]$  is changed to a value that is too large.

Alternatively,  $\text{Im}[Z]$  can be slaved to  $\text{Re}[Z]$  so  $\text{phase}(Z)$  remains constant when  $\text{Re}[Z]$  is changed.

(For an introduction to master–slave links and a complete listing of linking options for all segments, see Section 2.7.2.)

## Format and examples:

IMPEDANCE just a lumped series impedance, i.e., resistance plus inertance  
1.0 Pa-s/m<sup>3</sup> Re(Z)  
0.2 Pa-s/m<sup>3</sup> Im(Z)



Linked to one or more RPN calculations, IMPEDANCE gives the user the freedom to create nonlinear impedances. The following example shows an impedance with its resistance proportional to the square of the flow and its inertance proportional to the flow:

```
!-----2
RPN real part of impedance
12.0
U1 mag sqrd inp *
!-----3
RPN imaginary part of impedance
3.0E-4
U1 mag inp *
!-----4
IMPEDANCE
sameas 2A
sameas 3A
```

Another example of IMPEDANCE, in the context of the orifice of a pulse-tube refrigerator, is in Section 5.8.3.

## Calculations:

At an IMPEDANCE, volume flow rate and temperature are unchanged. Pressure changes according to

$$p_{1,\text{out}} = p_{1,\text{in}} - ZU_1. \quad (10.37)$$

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}}$  in insulated mode, described in Section 11.1.2. This assumes that the component is thermally insulated, so the heat generated by acoustic power dissipation must somehow be deposited elsewhere, either upstream or downstream, such as in a nearby heat exchanger (or flow out through a BEGIN or \*\*END segment).

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$  for anchored mode, described in Section 11.1.2. This essentially assumes that the component is thermally anchored, such as by a water bath, so heat generated by acoustic power dissipation is carried away locally.



If  $\dot{N} \neq 0$  in IMPEDANCE, the steady flow and the oscillating flow are assumed to be independent, simply superimposed, as if the flows are laminar. Thus, the difference in  $p_1$  across an IMPEDANCE is unaffected by the presence of the steady flow, and the change in  $p_{2,0,HL}$  is given by

$$p_{2,0,HL,out} - p_{2,0,HL,in} = \text{Re}[Z]\dot{N}_{2,0}m/\rho_m. \quad (10.38)$$

Since an IMPEDANCE has no defined area,  $p_{2,0,rev}$  is calculated based on the area of the most recent upstream segment having area.

#### 10.2.4. MINOR

##### General description:

Segment type MINOR accounts conveniently for minor-loss effects. The basic equation for a minor loss for steady flow [56, 52] is simply

$$\Delta p = -\frac{1}{2}K\rho u^2, \quad (10.39)$$

where  $K$  is the minor-loss coefficient and  $u$  is the velocity. DELTAEC's treatment of minor loss in MINOR is based on this equation.

Equations (10.40)–(10.47) were derived using the quasi-steady approximation, in which it is assumed that Eq. (10.39) is obeyed at each instant of time as  $u$  varies in time, with  $K$  independent of time except for its dependence on the direction of flow. This approximation is very good when  $|\xi_1| \gg r_h$ . For  $|\xi_1| \sim r_h$  or  $|\xi_1| \ll r_h$ , actual pressure drops may differ significantly from values calculated in MINOR.

Since DELTAEC uses  $U_1$  to characterize flow, MINOR has an area for converting  $U_1$  to velocity. Because DELTAEC treats oscillating flow, MINOR has two inputs for  $K$ s:  $K_+$  for the minor-loss coefficient in the  $+x$  direction, and  $K_-$  for the minor-loss coefficient in the  $-x$  direction. The coefficient in the  $\dot{N} = 0$  equations below differs from the 1/2 in Eq. (10.39) because of the sinusoidal time dependence of the velocity.

As shown in the second example below, the gas diodes in Ref. [42] can be modeled as a MINOR and a CONE in series.

We often obtain  $K_+$  and  $K_-$  from Ref. [52], even though that reference is for steady flow. This approximation should be fairly accurate when the Reynolds number is large and  $|\xi_1|$  is much greater than any dimensions of the component.

##### Input variables:

**Area** ( $\text{m}^2$ ) The area  $A$  used in the equations below. This is the area on which the minor-loss coefficients are based. Conventionally, this is usually the smallest area for area-changing geometries such as diffusers, but it is sometimes the nominal pipe area for valves and fittings.

**K+** Minor-loss coefficient  $K_+$  for flow in the positive- $x$  direction.

**K–** Minor-loss coefficient  $K_-$  for flow in the negative- $x$  direction.

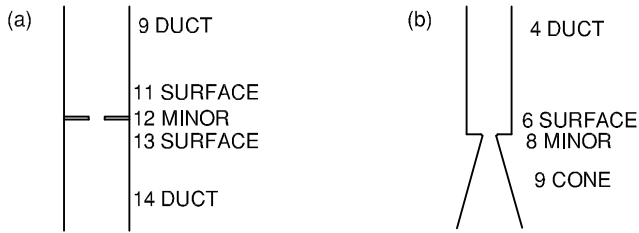


Figure 10.1: Examples of the use of MINOR. (a) Orifice plate in a channel. (b) Conical gas diode.

**Potential targets:**

None.

**Master-slave links:**

None. (Use sameas if you want to keep  $K_+$  and  $K_-$  equal to each other.)

**Format and examples:**

This is an example of a small orifice plate mounted in a pipe, as illustrated in Fig. 10.1(a):

```

!----- 9 -----
DUCT      Pre-orifice pipe
1.8000E-02 a Area      m^2      S= -2      2.2139E+04 A |p|      Pa
0.4756   b Perim      m        Fn( 9a)  -141.30   B Ph(p)     deg
0.1500   c Length     m                2.3148E-02 C |U|      m^3/s
                                                128.66   D Ph(U)     deg
                                                4.1756E-09 E Htot    W
ideal     Solid type
                                                -0.1923  F Edot     W
!----- 10 -----
RPN       calculate area difference (m2)
0.0000   a G or T      (t)      1.6000E-02 A ChngeMe
9a 12a -
!----- 11 -----
SURFACE   surface area on top of orifice plate
sameas 10A a Area      m^2      2.2139E+04 A |p|      Pa
                                                -141.30   B Ph(p)     deg
                                                2.3148E-02 C |U|      m^3/s
                                                128.65   D Ph(U)     deg
                                                4.1756E-09 E Htot    W
ideal     Solid type
                                                -0.2320  F Edot     W
!----- 12 -----
MINOR     minor loss in orifice plate
2.0000E-03 a Area      m^2      2.2140E+04 A |p|      Pa
1.0000   b K+          -140.59   B Ph(p)     deg
1.0000   c K-          2.3148E-02 C |U|      m^3/s
                                                128.65   D Ph(U)     deg
                                                4.1756E-09 E Htot    W
                                                -3.3973  F Edot     W
!----- 13 -----
SURFACE   surface area on bottom of orifice plate
sameas 11a a Area      m^2      2.2140E+04 A |p|      Pa
                                                -140.59   B Ph(p)     deg
                                                2.3148E-02 C |U|      m^3/s
                                                128.64   D Ph(U)     deg
                                                4.1756E-09 E Htot    W
ideal     Solid type
                                                -3.4370  F Edot     W
!----- 14 -----
DUCT      Post-orifice pipe
sameas 9a a Area      m^2      1.9565E+04 A |p|      Pa
sameas 9b b Perim      m        -140.52   B Ph(p)     deg
sameas 9c c Length     m        4.4525E-02 C |U|      m^3/s
                                                129.01   D Ph(U)     deg
                                                4.1756E-09 E Htot    W
ideal     Solid type
                                                -3.6358  F Edot     W

```

This is an example of a conical gas diode, such as is described in Ref. [42] and illustrated in Fig. 10.1(b).

```

!----- 4 -----
DUCT      quarter-wave duct from engine to diode
7.0000E-04 a Area      m^2      S= -2      7705.2      A |p|      Pa
9.3748E-02 b Perim    m      Fn( 4a)  -161.41    B Ph(p)    deg
4.6000    c Length    m                        5.4080E-02 C |U|      m^3/s
9.0500E-02 d Srough   m                        -87.804    D Ph(U)    deg
ideal     Solid type
          354.15      E Htot      W
          58.812     F Edot      W
!----- 5 -----
RPN       surface area of adapter plate
0.0000    a G or T      (t)      3.0000E-04 A ChngeMe
4a 8a -
!----- 6 -----
SURFACE   surface area of adapter plate
sameas    5A a Area      m^2      7705.2      A |p|      Pa
          -161.41    B Ph(p)    deg
          5.4080E-02 C |U|      m^3/s
          -87.804    D Ph(U)    deg
          354.15      E Htot      W
ideal     Solid type
          58.812     F Edot      W
!----- 7 -----
RPN       calculate Borda-Carnot minor loss here
0.0000    a G or T      (t)      0.1837      A ChngeMe
1 8a 4a / - sqrd
!----- 8 -----
MINOR     asymmetric flow at gas diode
4.0000E-04 a Area      m^2      7402.7      A |p|      Pa
5.0000E-02 b K+
sameas    7A c K-
          5.4080E-02 C |U|      m^3/s
          -87.804    D Ph(U)    deg
          354.15      E Htot      W
          10.829     F Edot      W
!----- 9 -----
CONE      expansion to plenum
sameas    8a a AreaI    m^2      S= -9      1.7471E+04 A |p|      Pa
7.0899E-02 b PerimI   m      Fn( 9b)  -177.49    B Ph(p)    deg
0.4000    c Length    m                        4.9949E-02 C |U|      m^3/s
1.5000E-02 d AreaF    m^2      S= -9      -87.809    D Ph(U)    deg
0.4342    e PerimF    m      Fn( 9d)  354.15     E Htot      W
5.0000E-04 f Srough   m                        2.4219     F Edot      W
ideal     Solid type

```

Other examples of MINOR are found in Sections 6.3 and 6.4.

### Pressure calculations for $\dot{N} = 0$ :

In the  $\dot{N} \rightarrow 0$  limit, changes in  $p_1$  across segment type MINOR are given by

$$p_{1,\text{out}} - p_{1,\text{in}} = - (K_- + K_+) \frac{2\rho_m |U_1|}{3\pi A^2} U_1, \quad (10.40)$$

and changes in p2OHL are given by

$$p_{2,0,HL,\text{out}} - p_{2,0,HL,\text{in}} = (K_- - K_+) \frac{\rho_m |U_1|^2}{8A^2}. \quad (10.41)$$

## Pressure calculations for $\dot{N} \neq 0$ :

For  $\dot{N} \neq 0$ , changes in  $p_1$  across segment type MINOR are given by

$$p_{1,\text{out}} - p_{1,\text{in}} = \varepsilon K_- \frac{\rho_m |U_1|}{A^2} U_1 \text{ for } \varepsilon \leq -1, \quad (10.42)$$

$$= - \left[ \left( 1 + \frac{\varepsilon^2}{2} \right) \sqrt{1 - \varepsilon^2} + \frac{3}{2} \varepsilon \sin^{-1} \varepsilon - \frac{K_- - K_+}{K_- + K_+} \frac{3\pi\varepsilon}{4} \right] (K_- + K_+) \\ \times \frac{2\rho_m |U_1|}{3\pi A^2} U_1 \text{ for } |\varepsilon| \leq 1, \quad (10.43)$$

$$= -\varepsilon K_+ \frac{\rho_m |U_1|}{A^2} U_1 \text{ for } \varepsilon \geq 1, \quad (10.44)$$

where  $\varepsilon = m\dot{N}/\rho_m |U_1|$ . Changes in p20HL are given by

$$p_{2,0,HL,\text{out}} - p_{2,0,HL,\text{in}} = K_- \frac{m^2 \dot{N}_{2,0}^2}{2\rho_m A^2} \left( 1 + \frac{1}{2\varepsilon_{2,0}^2} + \frac{|p_1| \cos \phi}{\gamma p_m \varepsilon_{2,0}} \right) \text{ for } \varepsilon_{2,0} \leq -1, \quad (10.45)$$

$$= \frac{\rho_m |U_1|^2}{8A^2} \left\{ (K_- - K_+) \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|p_1| \cos \phi}{\gamma p_m} \right) \right. \\ \left. - (K_+ + K_-) \frac{2}{\pi} \left[ \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|p_1| \cos \phi}{\gamma p_m} \right) \sin^{-1} \varepsilon_{2,0} \right. \right. \\ \left. \left. + \left( 3\varepsilon_{2,0} + \frac{4}{3} [1 + \varepsilon_{2,0}^2/2] \frac{|p_1| \cos \phi}{\gamma p_m} \right) \sqrt{1 - \varepsilon_{2,0}^2} \right] \right\} \\ \text{for } |\varepsilon_{2,0}| \leq 1, \quad (10.46)$$

$$= -K_+ \frac{m^2 \dot{N}_{2,0}^2}{2\rho_m A^2} \left( 1 + \frac{1}{2\varepsilon_{2,0}^2} + \frac{|p_1| \cos \phi}{\gamma p_m \varepsilon_{2,0}} \right) \text{ for } \varepsilon_{2,0} \geq 1, \quad (10.47)$$

where  $\varepsilon_{2,0} = m\dot{N}_{2,0}/\rho_m |U_1|$  and  $\phi$  is the phase angle between  $p_1$  and  $U_1$ . Derivations of these equations neglecting the terms in  $p_1$  are given in Ref. [42]; and derivations including the  $p_1$  terms appear in Chapter 14.

## Other calculations

Segment MINOR leaves  $T_m$  and  $U_1$  unchanged.

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}}$  in insulated mode, described in Section 11.1.2. This assumes that the component is thermally insulated, so the heat generated by acoustic power dissipation must somehow be deposited elsewhere, either upstream or downstream, such as in a nearby heat exchanger (or can flow out through a BEGIN or \*\*END segment). The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$  for anchored mode, described in Section 11.1.2. This essentially assumes that the component is thermally anchored, such as by a water bath, so heat generated by acoustic power dissipation is carried away locally.

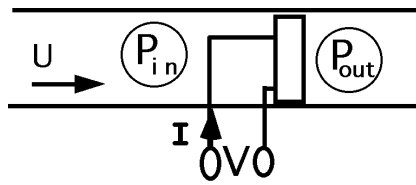


Figure 10.2: Enclosed \*EDUCER or \*ESPEAKER.

## 10.3. Series transducers

### 10.3.1. VESPEAKER, IESPEAKER, VEDUCER, and IEDUCER

#### General description:

The \*EDUCERS and \*ESPEAKERS are electroacoustic transducers with both ends “Enclosed” in the DELTAEC model, so they are in series between the previous segment and the subsequent segment.

\*ESPEAKERS let the user specify mass, spring constant, force constant, resistance, and inductance, so frequency-dependent (even resonant) electrodynamic transducers (e.g., linear motors and linear alternators, as well as loudspeakers) can be modeled easily. These \*ESPEAKER segments can be used to model moving-magnet transducers as well as moving-coil transducers; in the former case, set the  $Bl$  product equal to the moving-magnet transducer’s force constant, expressed in newtons per amp.

\*EDUCERS have specified transduction coefficients, which can be linked to the results of RPN calculations upstream to create the frequency or amplitude dependences of piezoelectric or any other type of transducer.

With IEDUCER and IESPEAKER, the user specifies the complex current, and each pass of DELTAEC calculates the complex voltage; with VEDUCER and VESPEAKER, the user specifies complex voltage, and DELTAEC computes complex current. VEDUCER and VESPEAKER cannot be used with zero electrical impedance because this would lead to a division by zero (see below). Hence, use IEDUCER or IESPEAKER for “electrically ideal” transducers.

\*ESPEAKERS incorporate thermal-hysteresis losses on their upstream and downstream areas, as if they included two SURFACES of equal areas but generally different pressure amplitudes. The \*EDUCER segments, which have no area parameter, have no built-in thermal-hysteresis losses, so use SURFACES upstream and downstream if needed.

See also Chapter 14 for a discussion of the saturation of magnetic flux in \*ESPEAKERS.

See also the side-branch transducers described in Section 10.4.2 below, which have their backsides hanging outside the DELTAEC model.

#### Input variables, VESPEAKER and IESPEAKER:

**Area** ( $\text{m}^2$ ) The surface area  $A$  of the moving part of the \*ESPEAKER. An area  $A$  is exposed to oscillating pressures on both the back side and the front side. The velocity of the moving element is  $U_1/A$ .

**R** (ohm) The electrical resistance  $R_e$  of the coil.

**L** (henry) The electrical inductance  $L$  of the coil when it is prevented from moving.

**BLProd** (tesla-m, which is equivalent to N/amp) For a traditional moving-coil loudspeaker, the product of the magnetic field  $B$  and the wire length  $l$  that is immersed in  $B$ . For other electrodynamic transducers, the force in newtons exerted when one amp of current is applied and nothing is allowed to move.

**M** (kg) The moving mass  $M$  of the transducer.

**K** (N/m) The spring constant  $K$  of the transducer. This should include zero-current magnetic forces (present in moving-magnet, stationary-iron transducers) and forces from mechanical springs, but not the gas-pressure forces on the front and back of area  $A$ .

**Rm** (N·s/m) The mechanical resistance  $R_m$  of the transducer.

**|V|** or **|I|** (volts or amps) In **VESPEAKER**, the amplitude of the oscillating voltage,  $|V_1|$ . In **IESPEAKER**, the amplitude of the oscillating current,  $|I_1|$ .

**ph(V)** or **ph(I)** (degrees) In **VESPEAKER**, the phase of the oscillating voltage  $V_1$ . Similarly, in **IESPEAKER**, the phase of the oscillating current  $I_1$ .

**Solid** The solid material on the leading and trailing surfaces  $A$ . (This is used only in the calculation of  $\epsilon_s$ , which seldom differs significantly from its value for a solid with infinite conductivity and infinite specific heat in **\*SPEAKERS**.)

#### Input variables, VEDUCER and IEDUCER:

**Re(Ze)** (ohm) The real part of the transducer's electrical impedance  $Z_e$  when it is prevented from moving.

**Im(Ze)** (ohm) The imaginary part of the transducer's electrical impedance  $Z_e$  when it is prevented from moving.

**Re(T1)** (V·s/m<sup>3</sup>) The real part of the complex transduction coefficient  $\tau$  in the equations below.

**Im(T1)** (V·s/m<sup>3</sup>) The imaginary part of the complex transduction coefficient  $\tau$  in the equations below.

**Re(T2)** (Pa/A) The real part of the complex transduction coefficient  $\tau'$  in the equations below.

**Im(T2)** (Pa/A) The imaginary part of the complex transduction coefficient  $\tau'$  in the equations below.

**Re(Zm)** (Pa·s/m<sup>3</sup>) The real part of the mechanical impedance  $Z_m$  of the transducer.

**Im(Zm)** (Pa·s/m<sup>3</sup>) The imaginary part of the mechanical impedance  $Z_m$  of the transducer.

**|V| or |I|** (volts or amps) In **VEDUCER**, the amplitude of the oscillating voltage  $V_1$ . In **IEDUCER**, the amplitude of the oscillating current  $I_1$ .

**ph(V) or ph(I)** (degrees) In **VEDUCER**, the phase of the oscillating voltage  $V_1$ . In **IEDUCER**, the phase of the oscillating current  $I_1$ .

### Potential targets:

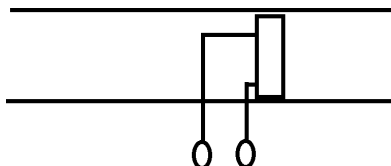
None.

### Master–slave links:

None. The use of **sameas** is sometimes convenient in **\*EDUCERs**, to keep  $\tau' = \tau$ .

### Format and examples:

```
VEDUCER      Enclosed driver
1.000E-09    a Re(Ze)   ohms
.000         b Im(Ze)   ohms
1.000E+04    c Re(T1)  V-s/m^3
.000         d Im(T1)  V-s/m^3
-1.000E+04   e Re(T2)  Pa/A
.000         f Im(T2)  Pa/A
1.000E-09    g Re(Zm)  Pa-s/m^3
1.000E-09    h Im(Zm)  Pa-s/m^3
10.0         i Vin     V
45.0         j Ph(Vin) deg
```



**IEDUCER**: same as **VEDUCER**, except that current appears in lines **i** and **j** instead of voltage.

```
VESPEAKER
6.000E-04    a Area     m^2
6.00         b R        ohms
.000         c L        H
8.00         d BLProd   T-m
5.000E-03    e M        kg
1.000E+04    f K        N/m
.000         g Rm       N-s/m
62.0         h Vin     V
-37.2        i Ph(Vin) deg
kapton       Solid type
```

**IESPEAKER**: same as **VESPEAKER**, except that current appears in line **h** and **i** instead of voltage.

Other examples of series **\*ESPEAKERs** are in Sections 5.2 and 5.8.2.

### Calculations:

An Enclosed transducer **IEDUCER**, **VEDUCER**, **IESPEAKER**, **VESPEAKER** is an object attached in series with other segments, as shown in Fig. 10.2. Volume flow rate  $U_1$  is nearly unchanged (except for surface thermal-hysteresis losses—see below), but pressure is changed by the force exerted by the transducer, obeying the complex canonical equations

$$V_1 = Z_e I_1 - \tau U_1, \quad (10.48)$$

$$p_{1,\text{out}} - p_{1,\text{in}} = \tau' I_1 - Z_m U_1. \quad (10.49)$$

- If current  $I_1$  is specified, then use **IESPEAKER** or **IEDUCER**, and find  $p_{1,\text{out}} = p_{1,\text{in}} + \tau' I_1 - Z_m U_1$  and  $V_1 = Z_e I_1 - \tau U_1$ .

- If voltage  $V_1$  is specified, then use **VESPEAKER** or **VEDUCER**, and find  $I_1 = (V_1 + \tau U_1)/Z_e$  and  $p_{1,\text{out}} = p_{1,\text{in}} + \tau' I_1 - Z_m U_1$ .
- If a fixed electrical load impedance  $Z_{\text{ext}}$  is attached to the transducer, it can be modeled using an **IMPEDANCE** segment *instead of* a **\*ESPEAKER** or **\*EDUCER** segment, with  $Z_{\text{imp}} = Z_m - \tau\tau'/(Z_e + Z_{\text{ext}})$ ; or use one of these Enclosed transducer segments with one or more RPN targets to maintain the impedance, as shown in the example in Section 4.7.

For **\*EDUCERS**, the complex electrical impedance  $Z_e = \text{Re}[Z_e] + i\text{Im}[Z_e]$  and the complex mechanical impedance  $Z_m = \text{Re}[Z_m] + i\text{Im}[Z_m]$  are given explicitly by the user.

In the case of **\*ESPEAKERS**,  $Z_e = R_e + i\omega L$ ;  $\tau = -\tau' = Bl/A$ ;  $Z_m = R_m/A^2 + i(\omega M - K/\omega)/A^2$ . Thermal surface losses are computed for area  $A$  using the same approach as for a **SURFACE**, but Enclosed speakers have area  $A$  exposed to  $p_{1,\text{in}}$  and area  $A$  exposed to  $p_{1,\text{out}}$ , because typically both sides of the speaker experience oscillatory pressure. This thermal hysteresis on those two surfaces manifests itself as a small change in volume flow rate, according to

$$U_{1,\text{out}} = U_{1,\text{in}} - \frac{\omega}{\rho_m a^2} \frac{\gamma - 1}{1 + \epsilon_s} \frac{\delta_\kappa}{2} A (p_{1,\text{in}} + p_{1,\text{out}}), \quad (10.50)$$

and  $U_1$  in Eqs. (10.48) and (10.49) is actually the value after the *first* thermal-hysteresis loss:

$$U_1 = U_{1,\text{in}} - \frac{\omega}{\rho_m a^2} \frac{\gamma - 1}{1 + \epsilon_s} \frac{\delta_\kappa}{2} A p_{1,\text{in}}. \quad (10.51)$$

The **Px** outputs for these segments give the complex pressure difference across the transducer.

None of these segments affects  $T_m$ .

The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}} + \dot{W}_{\text{elec}}$  in insulated mode, described in Section 11.1.2. This assumes that the component is thermally insulated, so the electrical energy added at the transducer must flow out of the segment, either upstream or downstream. The exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$  for thermally anchored mode, described in Section 11.1.2.

The **WorkIn** output variable displays the electrical power  $\text{Re}[V_1 \tilde{I}_1]/2$  that must be supplied to the transducer. If negative, this is electrical power produced by the transducer.

The **Ph(V/I)** variable is especially useful for constructing a target related to the electrical load impedance attached to a **\*SPEAKER** that is used as an alternator; for example, **Ph(V/I) = 180°** means that the load impedance is purely resistive. See Fig. 4.1a for a more complicated example with both resistance and capacitance.

It is hard for us to anticipate what a user might intend if nonzero  $\dot{N}$  were to flow through a series transducer, so these segments behave the same whether  $\dot{N} = 0$  or not.



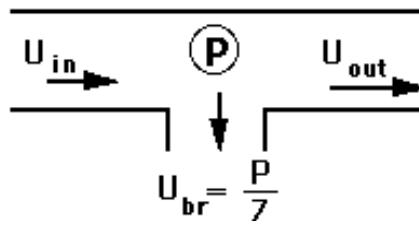


Figure 10.3: BRANCH, OPNBRANCH, or PISTBRANCH.

## 10.4. Side-branch transducers and side-branch impedances

### 10.4.1. BRANCH, OPNBRANCH, and PISTBRANCH

#### General description:

BRANCH, OPNBRANCH, and PISTBRANCH are side branches with specified impedances. With BRANCH, the user specifies the real and imaginary parts of the impedance, assumed independent of frequency (unless linked by `sameas` to results of frequency-dependent RPN calculations). OPNBRANCH and PISTBRANCH incorporate the frequency dependence of radiation impedances. PISTBRANCH approximates the radiation impedance of a flanged piston of the given radius, while radiation impedance at an opening radiating into  $4\pi$  solid angle can be modeled as an OPNBRANCH. Long ago, OPNBRANCH was added to DELTAE before RPN segments were available. Today, we find OPNBRANCH very awkward; it seems much easier to model the end of an open tube radiating into  $4\pi$  solid angle with RPN calculations and a SOFTEND segment. An example is shown in Section 11.1.3.

(If a side branch has nonzero  $\dot{N}$  flowing past it, the user might consider what fraction of the energy added by the transducer goes into raising the temperature of the steady flow passing by and what fraction is added to the thermoacoustic wave. In a VX\* heat-exchanger segment,  $F_{\dot{Q}\dot{N}}$ , (FracQN) determines these fractions. With a side-branch transducer, an RPN segment can be used, with user-specified assignment of these fractions using =Tm and =H2k.)

#### Input variables, BRANCH:

**Re(Z)** (Pa·s/m<sup>3</sup>) The real part of the branch impedance  $Z_{br}$ .

**Im(Z)** (Pa·s/m<sup>3</sup>) The imaginary part of the branch impedance  $Z_{br}$ .

**HtotBr** (W) How much power  $\dot{H}$  flows out of the trunk into the branch.

#### Input variables, OPNBRANCH:

**Re(Z)/k<sup>2</sup>** (Pa·s/m) The real part of the branch impedance  $Z_{br}$ , divided by the square of the wave number  $k = \omega/a$ .

**Im(Z)/k** (Pa·s/m<sup>2</sup>) The imaginary part of the branch impedance  $Z_{br}$ , divided by the wave number  $k = \omega/a$ .

**HtotBr** (W) How much power  $\dot{H}_{\text{tot}}$  flows out of the trunk into the branch.

**Input variables, PISTBRANCH:**

**Radius** (m) The radius  $r$  of the circular, flanged radiating surface.

**Potential targets:**

None.

**Master–slave links:**

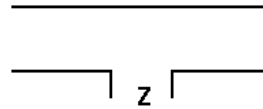
The imaginary part of a **BRANCH** impedance can be slaved to the real part, to hold the magnitude constant or to hold the phase constant when the real part is changed.

No master–slave options are available for **OPNBRANCH** and **PISTBRANCH**.

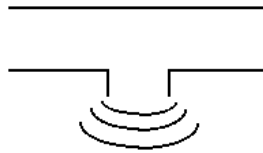
(For an introduction to master–slave links, and a complete listing of linking options for all segments, see Section 2.7.2.)

**Format and examples:**

```
BRANCH
1 Pa-s/m3 Re(Z)
1. Pa-s/m3 Im(Z)
12.5 W HtotBr
```



```
OPNBRANCH
0.05 Pa-s/m Re(Z)/k^2
0.20 Pa-s/m2 Im(Z)/k
12.5 W HtotBr
```



```
PISTBRANCH Flanged Piston
0.05 Radius m
```

**Calculations:**

A **BRANCH** is a side branch with complex impedance  $Z_{\text{br}}$ . Pressure is unchanged in the trunk, but volume flow rate in the trunk changes according to  $U_{1,\text{out}} = U_{1,\text{in}} - p_1/Z_{\text{br}}$ .

For an **OPNBRANCH**, the numbers in lines **a** and **b** are multiplied by  $(\omega/a)^2$  and  $\omega/a$ , respectively, to obtain the impedance  $Z_{\text{br}}$ . These dependences on frequency [12] are correct in the low-frequency limit,  $\omega r/a \ll 1$ . As for **BRANCH**, pressure is unchanged in the trunk, but volume flow rate in the trunk changes according to  $U_{1,\text{out}} = U_{1,\text{in}} - p_1/Z_{\text{br}}$ .

Similarly, for a flanged piston of radius  $r$  where the wave number is  $k = \omega/a$  locally, the **PISTBRANCH** radiation impedance is given by [59]

$$Z_{\text{br}} = \frac{\rho_m a}{A} \left( 1 - \frac{2J_1(2kr)}{2kr} + i \left\{ \begin{array}{ll} \left( \frac{4/\pi}{2kr} + \frac{\sqrt{8/\pi} \sin(2kr - 3\pi/4)}{(2kr)^{3/2}} \right) & \text{if } 2kr > 2.68, \\ \frac{(4/\pi)2kr}{3} \left( 1 - \frac{(2kr)^2}{15} \right) & \text{otherwise.} \end{array} \right. \right) \quad (10.52)$$

Note the implicit assumption that the outside and inside gases have the same density and sound speed.

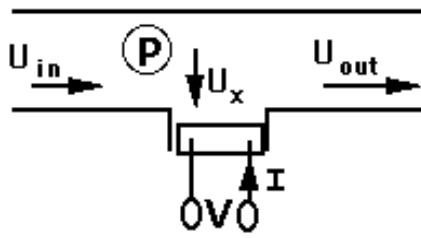


Figure 10.4: The side-branched transducers: IDUCER, ISPEAKER, VDUCER, or VSPEAKER.

Thus, the usual method for modeling a flanged piston or open tube radiating into  $2\pi$  solid angle is DUCT, PISTBRANCH, HARDEND, with the HARDEND's impedance targets targeted to zero to enforce  $U_{1,\text{out}} = 0$ , ensuring that  $U_{1,\text{in}} = U_{1,\text{br}} = p_1/Z_{\text{br}}$  at the PISTBRANCH.

Output  $\dot{E}_{\text{br}}$  gives the acoustic power flowing out of the branch.

None of these segments affects  $T_m$ . In BRANCH and OPNBRANCH, the trunk exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}} - \dot{H}_{\text{br}}$  in insulated mode, described in Section 11.1.2. The trunk exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}} - \dot{E}_{\text{br}}$  in PISTBRANCH in insulated mode. In thermally anchored mode, described in Section 11.1.2,  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$  in all three of these segments.

The behavior of these segments is independent of  $\dot{N}$ .

#### 10.4.2. VSPEAKER, ISPEAKER, VDUCER, and IDUCER

##### General description:

The \*DUCERs and \*SPEAKERs without “E” as their second character are electroacoustic transducers mounted as side branches to the apparatus. \*DUCERs have frequency-independent parameters; \*SPEAKERs let the user specify mass, spring constant, force constant, resistance, and inductance, so frequency-dependent (even resonant) transducers can be modeled. With IDUCER and ISPEAKER, the user specifies the (real) current, and each pass of DELTAEC calculates the complex voltage; with VDUCER and VSPEAKER, the user specifies the (real) voltage, and DELTAEC computes complex current. IDUCER and ISPEAKER cannot be used with zero mechanical impedance because this would lead to a division by zero (see below). Hence, use VDUCER or VSPEAKER for lossless resonant or massless-and-springless transducers.

\*SPEAKER segments incorporate thermal-hysteresis losses on their area as if they included a SURFACE, but \*DUCER segments, which have no area parameter, do not.

Side-branched transducer elements, which accept only the magnitude of voltage or current applied as an input, but not the phase, effectively fix the overall time phase of the entire DELTAEC model by enforcing zero time phase for that voltage or current. The phase of pressure or volume flow rate (or both, if both  $|p_1|$  and  $|U_1|$  are nonzero) in the BEGIN statement must usually be allowed to vary (i.e., guessed) to accommodate this condition. This effectively limits a model to only one V or ISPEAKER (or \*DUCER), unless they are wired exactly in phase (or  $180^\circ$  out of phase, which can be accommodated with a minus sign).

(If a side-branch transducer has nonzero  $\dot{N}$  flowing past it, the user might consider what fraction of the energy added by the transducer goes into raising the temperature of the

steady flow passing by and what fraction is added to the thermoacoustic wave. In a **VX\*** heat-exchanger segment,  $F_{\dot{Q}\dot{N}}$ , (**FracQN**) determines these fractions. With a side-branch transducer, an **RPN** segment can be used, with user-specified assignment of these fractions using **=Tm** and **=H2k**.)

See also Chapter 14 for a discussion of the saturation of magnetic flux in **\*SPEAKERS**.

See also the series transducers described in Section 10.3.1 above.

### Input variables, VSPEAKER and ISPEAKER:

**Area** ( $\text{m}^2$ ) The surface area  $S$  of the moving part of the **\*SPEAKER**. An area  $S$  is exposed to oscillating pressure on the front side. The velocity of the moving element is  $U_1/S$ .

**R** (ohm) The electrical resistance  $R_e$  of the coil.

**L** (henry) The electrical inductance  $L$  of the coil when it is prevented from moving.

**BLProd** (tesla-meter, which is equivalent to newton/amp) For a traditional moving-coil loudspeaker, the product of the magnetic field  $B$  and the wire length  $l$  immersed in  $B$ . For other electrodynamic transducers, the force in newtons exerted when one amp of current is applied and nothing is allowed to move.

**M** (kg) The moving mass  $M$  of the transducer.

**K** (N/m) The spring constant  $K$  of the transducer. This should include mechanical-spring forces and zero-current magnetic forces, but not gas-pressure forces exerted on area  $S$ .

**Rm** (N·s/m) The mechanical resistance  $R_m$  of the transducer.

**|V|** or **|I|** (volts or amps) In **VSPEAKER**, the amplitude of the oscillating voltage  $V_1$ , which is taken to be real. In **ISPEAKER**, the amplitude of the oscillating current  $I_1$ , which is taken to be real.

**Solid** The solid material on the surface  $S$ . (This is used only in the calculation of  $\epsilon_s$ , which seldom differs significantly from its value for a solid with infinite conductivity and infinite specific heat in **\*SPEAKERS**.)

### Input variables, VDUCER and IDUCER:

**Re(Ze)** (ohm) The real part of the transducer's electrical impedance  $Z_e$  when it is prevented from moving.

**Im(Ze)** (ohm) The imaginary part of the transducer's electrical impedance  $Z_e$  when it is prevented from moving.

**Re(T1)** ( $\text{V}\cdot\text{s}/\text{m}^3$ ) The real part of the complex transduction coefficient  $\tau$  in the equations below.

**Im(T1)** ( $\text{V}\cdot\text{s}/\text{m}^3$ ) The imaginary part of the complex transduction coefficient  $\tau$  in the equations below.

**Re(T2)** (Pa/A) The real part of the complex transduction coefficient  $\tau'$  in the equations below.

**Im(T2)** (Pa/A) The imaginary part of the complex transduction coefficient  $\tau'$  in the equations below.

**Re(Zm)** (Pa·s/m<sup>3</sup>) The real part of the mechanical impedance  $Z_m$  of the transducer.

**Im(Zm)** (Pa·s/m<sup>3</sup>) The imaginary part of the mechanical impedance  $Z_m$  of the transducer.

**|V|** or **|I|** (volts or amps) In VDUCER, the amplitude of the oscillating voltage  $V_1$ , which is taken to be real. In IDUCER, the amplitude of the oscillating current  $I_1$ , which is taken to be real.

### Potential targets:

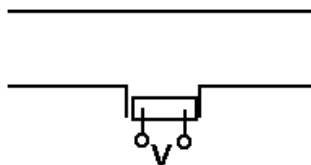
None.

### Master–slave links:

None. The use of **sameas** is sometimes convenient in \*DUCERS, to keep  $\tau' = \tau$ .

### Format and examples:

```
VDUCER
1.000E-09 a Re(Ze) ohms
.000 b Im(Ze) ohms
1.000E+04 c Re(T1) V-s/m^3
.000 d Im(T1) V-s/m^3
-1.000E+04 e Re(T2) Pa/A
.000 f Im(T2) Pa/A
1.000E-09 g Re(Zm) Pa-s/m^3
1.000E-09 h Im(Zm) Pa-s/m^3
10.0 i AplVol V
```



```
VSPEAKER
6.000E-04 a Area m^2
6.00 b R ohms
.000 c L H
8.00 d BLProd T-m
5.000E-03 e M kg
.000 f K N/m
.000 g Rm N-s/m
-22.5 h AplVol V
ideal Solid type
```

Another example of VSPEAKER is in Section 5.2.

### Calculations:

A side-branch transducer IDUCER, VDUCER, ISPEAKER, VSPEAKER is an object attached as shown Fig. 10.4 like a branch impedance, but obeying the complex equations  $V_1 = Z_e I_1 + \tau U_x$ ,  $p_1 = \tau' I_1 + Z_m U_x$ . Pressure is unchanged, but volume flow rate changes according to  $U_{1,\text{out}} = U_{1,\text{in}} - U_x$ . The volume flow rate of the transducer,  $U_x$ , is displayed in the output column.

- If current  $I_1$  is specified (note! its phase must be real), then use ISPEAKER or IDUCER, and find  $U_x = (p_1 - \tau' I_1) / Z_m$  and  $V_1 = Z_e I_1 + \tau U_x$ .

- If voltage  $V_1$  is specified (note! its phase must be real), then use **VSPEAKER** or **VDUCER**, and find  $I_1 = (Z_m V_1 - \tau p_1)/(Z_e Z_m - \tau \tau')$  and  $U_x = (V_1 - Z_e I_1)/\tau$ .
- If a fixed electrical load impedance  $Z_{\text{ext}}$  is attached to the transducer, it can be modeled using a **BRANCH** segment *instead of* a **\*SPEAKER** or **\*DUCER** segment, with  $Z_{\text{br}} = p_1/U_x = Z_m - \tau \tau'/(Z_e + Z_{\text{ext}})$ .

In the case of speakers,  $Z_e = R_e + i\omega L$ ;  $\tau = -\tau' = Bl/S$ ;  $Z_m = R_m/S^2 + i(\omega M - K/\omega)/S^2$ . Side-branch speakers are assumed to have area  $S$  exposed to the oscillating pressure. Thermal-hysteresis surface losses are computed for that area  $S$  using the same formula as for a **SURFACE**, Eqs. (10.35) and (10.36). As described in Section 10.2.2, thermal-hysteresis losses manifest themselves as a small change in volume flow rate.

Note that **IDUCER** and **ISPEAKER** will crash if  $Z_m$  is zero, so it is best to use **VDUCER** or **VSPEAKER** for mechanically ideal or lossless resonant transducers.

None of these segments affects  $T_m$ .

The trunk's exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}} + \dot{W}_{\text{elec}}$  in insulated mode, described in Section 11.1.2. This assumes that the component is thermally insulated, so the electrical energy added at the transducer must flow out of the segment along the trunk, either upstream or downstream. The trunk's exiting energy flow is computed as  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$  for thermally anchored mode, described in Section 11.1.2.

The **WorkIn** output variable displays the electrical power  $\text{Re}[V_1 \tilde{I}_1]/2$  that must be supplied to the transducer. If negative, this is electrical power produced by the transducer.

The **Ph(V/I)** variable is especially useful for constructing a target related to the electrical load impedance attached to a **\*SPEAKER** that is used as an alternator; for example, **Ph(V/I) = 180°** means that the load impedance is purely resistive. See Fig. 4.1 for a more complicated example with both resistance and capacitance.

These segments behave the same whether  $\dot{N} = 0$  or not.

## 10.5. Stacks and regenerators

The first three letters **STK** indicate a segment<sup>1</sup> that can develop a nonzero  $dT_m/dx$ . These segments are used to model the stacks of standing-wave thermoacoustic engines and refrigerators, the regenerators of Stirling engines and refrigerators (including pulse-tube refrigerators), the thermal buffer tubes of thermoacoustic-Stirling hybrid engines and refrigerators, and the pulse tubes of pulse-tube refrigerators. They are collected into three groups here: the smooth porous media **STKSLAB**, **STKRECT**, **STKCIRC**, and **STKPIN**, which can be used for either stacks or regenerators and are described in Section 10.5.1, the tortuous porous media **STKSCREEN** and **STKPOWERLW** for modeling regenerators, described in Section 10.5.2, and the boundary-layer **STKDUCT** and **STKCONE** segments, mostly used for pulse tubes and thermal buffer tubes and described in Section 10.6.1. If pore size or plate separation is much greater than thermal and viscous penetration depths, use **STKDUCT** or **STKCONE**.

<sup>1</sup>Segments **MIXT\*\***, described in Section 10.9.2, also develop nonzero  $dT_m/dx$ .

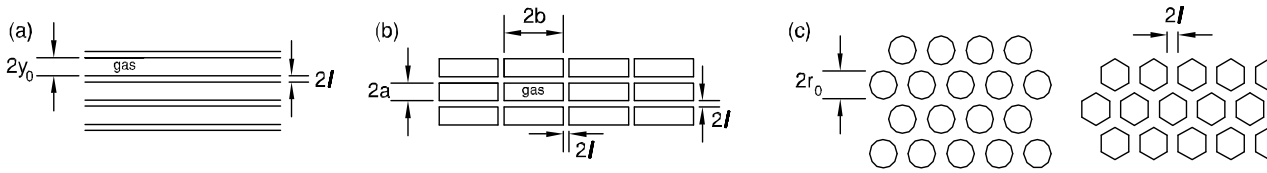


Figure 10.5: Common smooth-stack geometries, with the  $x$  axis perpendicular to the page. (a) STKSLAB. (b) STKRECT. (c) STKCIRC.

### 10.5.1. STKSLAB, STKCIRC, STKRECT, and STKPIN

#### General description of smooth stacks and regenerators:

Use STKSLAB for parallel-plate or spiral stacks or regenerators. Use STKRECT for square pores or rectangular pores whose aspect ratio is not large enough to justify the use of STKSLAB (see Ref. [60]). Use STKCIRC for circular or hexagonal pores. Use STKPIN for stacks comprised of pin arrays (see Ref. [61] for geometrical details of pin arrays).

In STKSLABs, STKRECTs, STKCIRCS, and STKPINs, the “Area” (the first line of the segment) is the total cross sectional area of the stack assembly, including both gas cross section and solid cross section.

In STKSLABs, STKRECTs, and STKCIRCS,  $A_{\text{gas}} = (\text{Area}) \times (\text{GasA}/A)$  and  $A_{\text{solid}} = (\text{Area}) \times (1 - \text{GasA}/A)$ . Plate half thickness (the 4th line of the segment) is used only for computing  $\epsilon_s$ , not for computing heat conduction along  $x$  or what fraction of the Area is available to the gas. This allows separate accounting for area blocked by “ideal” fins and by support struts or other structure as shown in Fig. 2.10(e). In most cases,  $\epsilon_s$  is near 0, so plate thickness need not be specified with much accuracy;  $\text{GasA}/A$  is far more important.

Because of the need to compute specialized functions, STKCIRCS compute more slowly than STKSLABs or STKDUCTs; STKPINs are slower still, and STKRECTs are very slow, especially for large aspect ratios. Hence, in the latter case, it may be convenient to use STKSLABs, at least until a design is almost finalized.

#### Input variables, STKSLAB:

**Area** ( $\text{m}^2$ ) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ .

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A = y_0/(y_0 + l)$ , where  $A_{\text{gas}}$  is the thermoacoustic area. The solid fraction  $1 - \phi$  is used for the calculation of axial thermal conduction by the solid.

**Length** (m) The length  $\Delta x$  of the stack or regenerator.

**y0** (m) Half the plate spacing is  $y_0$ . In other words,  $2y_0$  is the gap available to the gas—not the center-to-center spacing of the plates. See Fig. 10.5(a).

**Lplate** (m) Half the thickness of the solid plate is  $l$ . Thus,  $2l$  is the plate thickness. See Fig. 10.5(a). This variable is used in the calculation of  $\epsilon_s$  to account for any imper-

fection in the imposition of the isothermal boundary condition imposed on the gas by the solid.

**Solid** The solid of which the plates are made.

### Input variables, STKRECT:

**Area** ( $\text{m}^2$ ) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ .

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A$ . The solid fraction  $1 - \phi$  is used for the calculation of axial thermal conduction by the solid. Often, the user intends  $\phi = ab/(ab + al + bl + l^2)$ ; this can be enforced with a master–slave link (see below).

**Length** (m) The length  $\Delta x$  of the stack or regenerator.

**a** (m) Half of pore width. Thus,  $2a$  is the small dimension of the rectangular pore available to the gas. See Fig. 10.5(b).

**Lplate** (m) Half the thickness of the solid plate. Thus,  $2l$  is the plate thickness. This variable is used for the calculation of  $\epsilon_s$  to account for any imperfection in the imposition of the isothermal boundary condition imposed on the gas by the solid. See Fig. 10.5(b).

**b** (m) Half of pore breadth  $2b$ . Thus,  $2a \times 2b$  is the cross-sectional area available to the gas in a single pore. See Fig. 10.5(b).

**Solid** The solid of which the stack is made.

### Input variables, STKCIRC:

**Area** ( $\text{m}^2$ ) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ .

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A$ . The solid fraction  $1 - \phi$  is used for the calculation of axial thermal conduction by the solid.

**Length** (m) The length  $\Delta x$  of the stack or regenerator.

**radius** (m) The radius  $r_0$  of the circular pore. Thus,  $\pi r_0^2$  is the cross-sectional area available to the gas in a single pore. (When we use STKCIRC to model hexagonal “honeycomb” pores, we choose  $r_0$  so the pore has the correct cross-sectional area:  $r_0 = (\sqrt[4]{3}/\sqrt{2\pi}) b \simeq 0.525038b$ , where  $b$  is the flat-to-flat distance across the hexagonal pore.)

**Lplate** (m) Half the thickness of the solid sheet between pores. Thus,  $2l$  is the plate thickness. This variable is used in the calculation of  $\epsilon_s$  to account for any imperfection in the imposition of the isothermal boundary condition imposed on the gas by the solid.

**Solid** The solid of which the sheets are made.



### Input variables, STKPIN:

**Area** (m<sup>2</sup>) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ , but usually not intended to include the cross-sectional area of the pressure-vessel wall.

**2y0** (m)  $2y_0$  is the center-to-center, nearest-neighbor pin-to-pin distance in the hexagonal lattice of pins. See Ref. [61].

**Length** (m) The length  $\Delta x$  of the stack or regenerator.

**R pin** (m) The radius  $r_i$  of each pin.

**Solid** The solid of which the pins are made.

### Potential targets:

None.

### Master–slave links:

The porosity  $\phi$  ( $= \text{GasA}/A$ ) of some of these segments can be slaved to other parameters:

In STKSLAB, the porosity  $\phi$  can be slaved to other parameters through the equation

$$\phi = c \frac{y_0}{y_0 + l}, \quad (10.53)$$

where  $c$  is calculated when the link is first enabled, using

$$c = \phi \frac{y_0 + l}{y_0}. \quad (10.54)$$

If  $c = 1$ , this simply means that all of the cross section is full of gas and plate, with no struts, support members, or pressure-vessel wall. Otherwise,  $c \neq 1$  indicates that a fraction of the area is occupied by such structure; the link keeps that fraction a constant.

In STKRECT, the porosity  $\phi$  can be slaved to other parameters through the equation

$$\phi = \frac{ab}{(a + l)(b + l)}, \quad (10.55)$$

so the geometry looks exactly as shown in Fig. 10.5. No constant  $c$  is used.

In STKCIRC, the porosity  $\phi$  can be slaved to other parameters through the equation  $\phi = \min(cr_0^2/A, 1)$ , with  $c$  calculated when the link is first enabled using  $c = \phi A/r_0^2$ . This link maintains a constant number of holes, but with the porosity clipped at a maximum value of 1.

The length of any of these segments can be slaved to the length of another segment to keep the total length constant.

(For an introduction to master–slave links, and a complete listing of linking options for all segments, see Section 2.7.2.)

## Format and examples:

```

STKSLAB  parallel-plate stack
sameas 1  Area
0.724    GasA/A
7.85e-2 m Length
1.8e-4 m y0 (half the plate spacing)
4.0e-5 m Lplate (half the plate thickness)
kapton   Solid

```



```

STKRECT  rectangular-pore stack
sameas 1  Area
0.694    GasA/A
7.85e-2 m Length
2.0e-4 m  a (half of pore width)
4.0e-5 m  Lplate (half the plate thickness)
4.0e-4 m  b (pore area is 2a times 2b)
stainless Solid

```

```

STKCIRC  approximates hexagonal honeycomb stack
sameas 1  (m^2) total area
0.81     gas area/total area
0.279    (m) length
0.50e-3  (m) radius of circular pore
0.05e-3  (m) L:half of sheet thickness
stainless stack material

```

```

STKPIN   Muller/Keolian pinstack invention
sameas 2a a area      m^2
3.2e-4   b 2y0      m   2y0 = nearest-neighbor center-to-center distance
!        in the hexagonal lattice
0.1      c Length    m
4.e-5    d R pin     m   pin radius
stainless

```

Another example of STKSLAB is in Section 5.2. STKCIRC is used in Section 5.5, and STKRECT is used in Section 6.2.

## Calculations in smooth stacks and regenerators for $\dot{N} = 0$ :

Pressure propagates according to Rott's wave equation, written in the form

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}}U_1, \quad (10.56)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \left(1 + \frac{(\gamma-1)f_\kappa}{1+\epsilon_s}\right) p_1 + \frac{\beta(f_\kappa - f_\nu)}{(1-f_\nu)(1-\sigma)(1+\epsilon_s)} \frac{dT_m}{dx} U_1, \quad (10.57)$$

subject to the condition that the total energy flow  $\dot{H}_{2,k}$  is independent of  $x$ , which imposes the following condition on  $T_m(x)$ :

$$\frac{dT_m}{dx} = \frac{\dot{H}_{2,k} - \frac{1}{2}\text{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{T_m \beta (f_\kappa - \tilde{f}_\nu)}{(1+\epsilon_s)(1+\sigma)(1-f_\nu)} \right) \right]}{\frac{\rho_m c_p |U_1|^2}{2\omega A_{\text{gas}}(1-\sigma)|1-f_\nu|^2} \text{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1+\epsilon_s f_\nu / f_\kappa)}{(1+\epsilon_s)(1+\sigma)} \right] - A_{\text{gas}} k - A_{\text{solid}} k_s}. \quad (10.58)$$

For STKSLAB,

$$f_j = \frac{\tanh[(1+i)y_0/\delta_j]}{(1+i)y_0/\delta_j}, \quad \text{for } j = \kappa \text{ or } \nu, \quad (10.59)$$

$$\epsilon_s = \left( \frac{k\rho_m c_p}{k_s \rho_s c_s} \right)^{1/2} \frac{\tanh[(1+i)y_0/\delta_\kappa]}{\tanh[(1+i)\ell/\delta_s]}. \quad (10.60)$$

For STKRECT,

$$f_j = 1 - \frac{64}{\pi^4} \sum_{\substack{m,n \\ \text{odd}}} \frac{1}{m^2 n^2 Y_{mn}(\delta_j)}, \quad \text{for } j = \kappa \text{ or } \nu, \quad (10.61)$$

$$\epsilon_s = \left( \frac{k \rho_m c_p}{k_s \rho_s c_s} \right)^{1/2} \frac{f_\kappa (1+i) ab / \delta_\kappa (a+b)}{\tanh[(1+i)\ell/\delta_s]}, \quad (10.62)$$

$$\text{where } Y_{mn}(\delta) = 1 - i \frac{\pi^2 \delta^2}{8a^2 b^2} (b^2 m^2 + a^2 n^2). \quad (10.63)$$

For STKCIRC,

$$f_j = \frac{2J_1[(i-1)r_0/\delta_j]}{(i-1)(r_0/\delta_j)J_0[(i-1)r_0/\delta_j]}, \quad \text{for } j = \kappa \text{ or } \nu, \quad (10.64)$$

$$\epsilon_s = \left( \frac{k \rho_m c_p}{k_s \rho_s c_s} \right)^{1/2} \frac{f_\kappa (1+i)r_0/2\delta_\kappa}{\tanh[(1+i)\ell/\delta_s]}. \quad (10.65)$$

For STKPIN,

$$f_j = -\frac{\delta_j}{(i-1)} \frac{2r_i}{r_o^2 - r_i^2} \frac{Y_1[(i-1)r_o/\delta_j]J_1[(i-1)r_i/\delta_j] - J_1[(i-1)r_o/\delta_j]Y_1[(i-1)r_i/\delta_j]}{Y_1[(i-1)r_o/\delta_j]J_0[(i-1)r_i/\delta_j] - J_1[(i-1)r_o/\delta_j]Y_0[(i-1)r_i/\delta_j]} \quad \text{for } j = \kappa \text{ or } \nu, \quad (10.66)$$

$$\epsilon_s = \left( \frac{k \rho_m c_p}{k_s \rho_s c_s} \right)^{1/2} \frac{J_0(\sqrt{-i\omega/\kappa_s} r_i)}{J_1(\sqrt{-i\omega/\kappa_s} r_i)} f_\kappa \sqrt{-i\omega/\kappa} \frac{r_o^2 - r_i^2}{2r_i}, \quad (10.67)$$

where  $\kappa = k/\rho_m c_p$ ,  $\kappa_s = k_s/\rho_s c_s$ ,  $r_i$  is pin radius, and  $r_o^2 = (\sqrt{3}/2\pi) (2y_o)^2$  with  $2y_o$  being the center-to-center distance between nearest-neighbor pins in the hexagonal array of pins.

### Calculations in smooth stacks and regenerators for $\dot{N} \neq 0$ :

To turn on steady-flow calculations, check the appropriate box in the Optional Parameters dialog in the BEGIN segment. In the affected STK\*\* segments, the dependence of  $T_m$  on  $x$  can be changed significantly by the nonzero  $\dot{N}$ .

The display of information is slightly changed. H2k is the “acoustic”-plus-longitudinal-conduction part of the total energy flux—not including the steady-flow part of the total energy flux,  $\dot{N} m w_m$ , where  $w_m$  is the enthalpy per unit mass. Htot is the sum of H2k and  $\dot{N} m w_m$ .

Nonzero  $\dot{N}$  changes the temperature profile in smooth stacks according to Refs. [38, 39, 13]: The temperature gradient in stacks is computed using constancy of total energy flux

$\dot{H}_{\text{tot}}$ . In other words, we solve this equation for  $dT_m/dx$  :

$$\begin{aligned} \dot{H}_{\text{tot}} &= \frac{1}{2} \text{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{T_m \beta (f_\kappa - \tilde{f}_\nu)}{(1 + \epsilon_s)(1 + \sigma) (1 - \tilde{f}_\nu)} \right) \right] \\ &+ \frac{\rho_m c_p}{2A_{\text{gas}} \omega (1 - \sigma) |1 - f_\nu|^2} \frac{dT_m}{dx} |U_1|^2 \text{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1 + \epsilon_s f_\nu / f_\kappa)}{(1 + \epsilon_s)(1 + \sigma)} \right] \\ &- (A_{\text{gas}} k + A_{\text{solid}} k_s) \frac{dT_m}{dx} + \dot{N} m w_m \end{aligned} \quad (10.68)$$

and DELTAEC uses  $\dot{H}_{\text{tot}} = \text{constant}$  to integrate with respect to  $x$  through a stack. The only  $\dot{N} \neq 0$  feature in this equation is the final term,  $\dot{N} m w_m$ , where  $w_m$  is the mean enthalpy per unit mass. The mean enthalpy is referenced to zero at the environment temperature  $T_0$ , which is 300 K by default and can be set to another value via `=Tzero` in an RPN segment. As usual, the value of  $\dot{H}_{\text{tot}}$  is determined by conditions in segments other than the stack, such as the value of  $\dot{N}$  set in the `BEGIN` segment and the heat flows in adjacent heat exchangers.

The stack energy equation, Eq. (10.68), is reasonable only if  $\dot{N}$  is essentially of second order, and hence only if  $\dot{N}$  is formally negligible in the first-order momentum and energy equations [38, 39, 13]. Hence, we have not implemented any Doppler effect on  $dp_1$  and  $dU_1$  in stacks.

In the smooth stacks,  $dp_{2,0,HL}/dx$  is calculated with laminar expressions based on  $\dot{N}_{2,0}$  :

$$\frac{dp_{2,0,HL}}{dx} = -\frac{8\mu m \dot{N}_{2,0}}{\rho_m A_{\text{gas}} r_0^2} \text{ for STKCIRC}, \quad (10.69)$$

$$= -\frac{3\mu m \dot{N}_{2,0}}{\rho_m A_{\text{gas}} y_0^2} \text{ for STKSLAB}, \quad (10.70)$$

$$= -\frac{3\mu m \dot{N}_{2,0}}{\rho_m A_{\text{gas}} a^2 F_{\text{shape}}} \text{ for STKRECT}, \quad (10.71)$$

$$= -\frac{8\mu m \dot{N}_{2,0}}{\rho_m A_{\text{total}} r_o^2 [3 + r_i^4/r_o^4 - 4r_i^2/r_o^2 - 4 \ln(r_i/r_o)]} \text{ for STKPIN}, \quad (10.72)$$

where the `STKRECT`'s shape factor in Eq. (10.71) is given by

$$F_{\text{shape}} = 1 - \frac{192 a}{\pi^5 b} \sum_{n=1,3,5,\dots} \frac{\tanh(n\pi b/2a)}{n^5} \quad (10.73)$$

$$\simeq 1 - \frac{192 a}{\pi^5 b} \left[ \tanh \frac{\pi b}{2a} + \frac{\tanh(3\pi b/2a)}{3^5} + \frac{\tanh(5\pi b/2a)}{5^5} + 0.000088536 \right] \text{ for } a \leq b. \quad (10.74)$$

### 10.5.2. STKSCREEN and STKPOWERLW

(For a general description of all of the `STK**` segments, see the first paragraph of Section 10.5.)

## General description of tortuous regenerators:

Use **STKSCREEN** for stacked-screen regenerators (see Ref. [62]). Use **STKPOWERLW** for any other regenerator for which friction factor and heat-transfer coefficients follow power laws in Reynolds number. The algorithms for these segments assume reasonably good internal thermal contact, so do not trust these segments for  $r_h \sim \delta_k$  and certainly not for  $r_h > \delta_k$ . **DELTAEC** has no capability to model the physics of oscillating flow through tortuous porous media except in the small-pore, nearly isothermal limit.

### Input variables, STKSCREEN:

**Area** (m<sup>2</sup>) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ , including the screens but not including the cross-sectional area of the pressure-vessel wall.

**VolPor** The volumetric porosity  $\phi$ . This is the ratio of the volume of gas in the regenerator to the total volume of the regenerator. For plain square-weave screen with  $m$  wires per unit length and wire diameter  $d_{\text{wire}}$ , we use

$$\phi \simeq 1 - \frac{\pi m d_{\text{wire}}}{4}, \quad (10.75)$$

as recommended by Organ [27], or we weigh the regenerator and use its  $A$ ,  $\Delta x$ , and the known density of its wire material to calculate  $\phi$ . [See also Eq. (5.14).]

**Length** (m) The length  $\Delta x$  of the regenerator.

**rh** (m) The hydraulic radius  $r_h$  of the regenerator. This is the ratio of the gas volume to the gas–solid contact area. For plain square-weave screen with wire diameter  $d_{\text{wire}}$ , we use

$$r_h \simeq d_{\text{wire}} \frac{\phi}{4(1 - \phi)}, \quad (10.76)$$

as recommended by Organ [27].

**ksFrac** If the solid material in the regenerator were arranged in smooth columns parallel with the  $x$  axis, as in the smooth stacks described above in Section 10.5.1, then the contribution of ordinary thermal conduction in the solid to  $\dot{H}_{2,k}$  would be  $(1 - \phi) A k_s dT_m/dx$ . However, in **STKSCREEN** the solid conduction is reduced by the random point contacts between wires. **ksFrac** is the reduction factor. Reference [28] recommends a value of 0.1. We often use a larger value, to include the thermal conductance of the pressure-vessel wall around the regenerator. [See Eq. (9.1).]

**Solid** The solid of which the wires are made.

### Input variables, STKPOWERLW:

**Area** (m<sup>2</sup>) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ , not including the cross-sectional area of the pressure-vessel wall.

**VolPor** The volumetric porosity  $\phi$ . This is the ratio of the volume of gas in the regenerator to the total volume of the regenerator.

**Length** (m) The length  $\Delta x$  of the regenerator.

**rh** (m) The hydraulic radius  $r_h$  of the regenerator. This is the ratio of the gas volume to the gas–solid contact area.

**ksFrac** If the solid material in the regenerator were arranged in smooth columns parallel with the  $x$  axis, as in the smooth stacks described above in Section 10.5.1, then the contribution of ordinary thermal conduction in the solid to  $\dot{H}_{2,k}$  would be  $(1 - \phi) Ak_s dT_m/dx$ . However, in porous media such as balls the solid conduction is reduced by the random point contacts in the  $x$  direction between pieces of the solid. **ksFrac** is the reduction factor. Reference [28] recommends a value of 0.1 for woven screens.

**f con** The friction-factor parameter  $f_{\text{con}}$  in the equations below.

**f exp** The friction-factor parameter  $f_{\text{exp}}$  in the equations below.

**h con** The heat-transfer parameter  $h_{\text{con}}$  in the equations below.

**h exp** The heat-transfer parameter  $h_{\text{exp}}$  in the equations below.

**Solid** The solid with which the regenerator is made.

### Potential targets:

None.

### Master–slave links:

The length of either of these segments can be slaved to the length of another segment to keep the sum of the two lengths constant.

(For an introduction to master–slave links, and a complete listing of linking options for all segments, see Section 2.7.2.)

### Format and examples of tortuous regenerators:

```

STKSCREEN          a screen regenerator
sameas 1a  a Area      m^2  cross section of regenerator
          b VolPor     volumetric porosity
          0.673
          5.500E-02 c Length    m
          1.830E-05 d rh       m  hydraulic radius
          0.300      e ksFrac   fudge factor F for solid conduction
stainless Solid type

```

```

STKPOWERLW        an etched foil regenerator
sameas 1a  a Area      m^2  cross section of regenerator
          b VolPor     volumetric porosity
          0.700
          0.04        c Length    m
          40.e-6      d rh       m  hydraulic radius
          0.300      e ksFrac   fudge factor for solid conduction
          36.        f f_con
          1.0        g f_exp
          24.        h h_con
          0.8        i h_exp
stainless Solid type

```

Other examples of the use of STKSCREEN are in Sections 5.8 and 6.3.

### Calculations in tortuous regenerators for $\dot{N} = 0$ :

In STKSCREEN regenerators, the pressure, volume flow rate, and mean temperature evolve according to

$$\frac{dp_1}{dx} = -i\omega\rho_m \left[ 1 + \frac{(1-\phi)^2}{2(2\phi-1)} \right] \langle u_1 \rangle - \frac{\mu}{r_h^2} \left( \frac{c_1(\phi)}{8} + \frac{c_2(\phi)N_{R,1}}{3\pi} \right) \langle u_1 \rangle, \quad (10.77)$$

$$\begin{aligned} \frac{d\langle u_1 \rangle}{dx} = & -\frac{i\omega\gamma}{\rho_m a^2} p_1 + \beta \frac{dT_m}{dx} \langle u_1 \rangle + \\ & i\omega\beta \left[ \frac{T_m\beta}{\rho_m c_p} \frac{\epsilon_s + (g_c + e^{2i\theta_p} g_v)\epsilon_h}{1 + \epsilon_s + (g_c + e^{2i\theta_T} g_v)\epsilon_h} p_1 - \frac{1}{i\omega} \frac{dT_m}{dx} \frac{\epsilon_s + (g_c - g_v)\epsilon_h}{1 + \epsilon_s + (g_c + e^{2i\theta_T} g_v)\epsilon_h} \langle u_1 \rangle \right], \end{aligned} \quad (10.78)$$

$$\begin{aligned} \frac{dT_m}{dx} = & \left\{ \text{Re} \left[ \left( T_m\beta \frac{\epsilon_s + \epsilon_h(g_c + e^{2i\theta_p} g_v)}{1 + \epsilon_s + \epsilon_h(g_c + e^{2i\theta_T} g_v)} + 1 - T_m\beta \right) p_1 \widetilde{\langle u_1 \rangle} \right] - \frac{2\dot{H}_{2,k}}{\phi A} \right\} \\ & / \left\{ \frac{\rho_m c_p}{\omega} \text{Im} \left[ \frac{\epsilon_s + \epsilon_h(g_c - g_v)}{1 + \epsilon_s + \epsilon_h(g_c + e^{2i\theta_T} g_v)} \right] \langle u_1 \rangle \widetilde{\langle u_1 \rangle} + 2 \left[ k_{s,\text{eff}} \frac{1-\phi}{\phi} + k \right] \right\}, \end{aligned} \quad (10.79)$$

using

$$c_1(\phi) = 1268 - 3545\phi + 2544\phi^2, \quad c_2(\phi) = -2.82 + 10.7\phi - 8.6\phi^2, \quad (10.80)$$

$$b(\phi) = 3.81 - 11.29\phi + 9.47\phi^2, \quad (10.81)$$

$$N_{R,1} = 4 |\langle u_1 \rangle| r_h \rho_m / \mu, \quad (10.82)$$

$$\epsilon_s = \phi \rho_m c_p / (1-\phi) \rho_s c_s, \quad \epsilon_h = 8ir_h^2/b(\phi)\sigma^{1/3}\delta_\kappa^2, \quad (10.83)$$

$$\delta_\kappa^2 = 2k/\omega\rho_m c_p, \quad (10.84)$$

$$\theta_p = \text{phase}(\langle u_1 \rangle) - \text{phase}(p_1), \quad \theta_T = \text{phase}(\langle u_1 \rangle) - \text{phase}(\langle T \rangle_{u,1}), \quad (10.85)$$

$$g_c = \frac{2}{\pi} \int_0^{\pi/2} \frac{dz}{1 + N_{R,1}^{3/5} \cos^{3/5}(z)}, \quad g_v = -\frac{2}{\pi} \int_0^{\pi/2} \frac{\cos(2z) dz}{1 + N_{R,1}^{3/5} \cos^{3/5}(z)}. \quad (10.86)$$

These expressions were derived [62] with the assumption that viscous and thermal penetration depths are much larger than  $r_h$ . The spatial average oscillatory velocity  $\langle u_1 \rangle = \langle U_1 \rangle / \phi A$ , where  $\phi$  is volumetric porosity and  $A$  is regenerator cross-sectional area; and  $k_{\text{eff}} = (\text{ksFrac}) k_s$  where **ksFrac** is a factor to reduce thermal conduction along  $x$  due to the poor thermal contact between adjacent screen layers (Reference [28] recommends **ksFrac**  $\approx 0.1$ ). **ksFrac** can also be used to account for conduction in the case surrounding the regenerator. The trigonometric integrals in Eqs. (10.86) are not evaluated by DELTAEC;

these integrals were performed once, off-line, to obtain simple functional fits that DELTAEC uses during computation.

STKPOWERLW segments are calculated in the same manner as STKSCREEN, with a few exceptions. The steady-flow friction factor and heat transfer coefficients are parameterized by

$$f = f_{\text{con}} N_R^{-f_{\text{exp}}}, \quad (10.87)$$

$$N_{St} \sigma^{2/3} = h_{\text{con}} N_R^{-h_{\text{exp}}}, \quad (10.88)$$

where  $N_{St}$  is the Stanton number and the Reynolds number  $N_R$  is defined in the usual way as

$$N_R = \frac{4Ur_h\rho}{\phi A\mu}. \quad (10.89)$$

[Note: Equation (10.87) is Fanning friction factor, the friction factor used by Kays and London [63], so instantaneously  $dp/dx = -(f/r_h)\frac{1}{2}\rho u^2$ .] The pressure equation is replaced by

$$\frac{dp_1}{dx} = -i\omega\rho_m \langle u_1 \rangle - I_f \frac{\mu}{8r_h^2} f_{\text{con}} N_{R,1}^{1-f_{\text{exp}}} \langle u_1 \rangle, \quad (10.90)$$

where

$$I_f = \frac{2}{\pi} \int_0^\pi \sin^{3-f_{\text{exp}}}(z) dz. \quad (10.91)$$

In the volume-flow-rate and mean-temperature equations, these parameters are redefined for the power-law stack:

$$g_c = N_{R,1}^{h_{\text{exp}}-1} \frac{2}{\pi} \int_0^{\pi/2} \cos^{h_{\text{exp}}-1}(z) dz, \quad (10.92)$$

$$g_v = -N_{R,1}^{h_{\text{exp}}-1} \frac{2}{\pi} \int_0^{\pi/2} \cos 2z \cos^{h_{\text{exp}}-1}(z) dz, \quad (10.93)$$

$$b(\phi) = h_{\text{con}}. \quad (10.94)$$

The trigonometric integrals in Eqs. (10.92) and (10.93) are not evaluated by DELTAEC; these integrals were performed once, off-line, to obtain simple functional fits that DELTAEC uses during computation.

### Calculations in tortuous regenerators for $\dot{N} \neq 0$ :

The temperature gradient in STKSCREEN and STKPOWERLW is computed using constancy of total energy flux  $\dot{H}_{\text{tot}} = \dot{H}_{2,k} + \dot{N}mw_m$ . In other words, we solve this equation for  $dT_m/dx$ :

$$\begin{aligned} \dot{H}_{\text{tot}} &= \frac{1}{2} \text{Re} \left[ p_1 \widetilde{U}_1 \left( 1 - \frac{T_m \beta (f_\kappa - \widetilde{f}_\nu)}{(1 + \epsilon_s)(1 + \sigma)(1 - \widetilde{f}_\nu)} \right) \right] \\ &+ \frac{\rho_m c_p}{2A_{\text{gas}}\omega(1 - \sigma)|1 - f_\nu|^2} \frac{dT_m}{dx} |U_1|^2 \text{Im} \left[ \widetilde{f}_\nu + \frac{(f_\kappa - \widetilde{f}_\nu)(1 + \epsilon_s f_\nu / f_\kappa)}{(1 + \epsilon_s)(1 + \sigma)} \right] \\ &- (A_{\text{gas}}k + A_{\text{solid}}k_{s,\text{eff}}) \frac{dT_m}{dx} + \dot{N}mw_m, \end{aligned} \quad (10.95)$$



and DELTAEC uses  $\dot{H}_{\text{tot}} = \text{constant}$  to find  $T_m(x)$  through the regenerator. The only term with nonzero  $\dot{N}$  in this equation is the final one,  $\dot{N}mw_m$ . As usual, the value of  $\dot{H}_{\text{tot}}$  is determined by conditions in segments other than the regenerator, such as the heat flows in adjacent heat exchangers.

The first-order pressure gradient  $dp_1/dx$  is unchanged by nonzero  $\dot{N}$ , and  $d\langle u_1 \rangle/dx$  is changed only indirectly through its dependence on  $dT_m/dx$ .

In STKSCREEN, the expression for head-loss gradient is a straightforward result of using the quasi-steady approximation, as outlined in Chapter 14:

$$\frac{dp_{2,0}}{dx} = -\frac{\mu m \dot{N}_{2,0}}{8r_h^2 \rho_m \phi A} \left[ c_1(\phi) + c_2(\phi) N_{R,2,0} \left( 1 + \frac{1}{2\varepsilon_{2,0}^2} + \frac{|p_1| \cos \phi}{p_m \varepsilon_{2,0}} \right) \right], \quad |\varepsilon_{2,0}| \geq 1, \quad (10.96)$$

$$= -\frac{\mu m \dot{N}_{2,0}}{8r_h^2 \rho_m \phi A} \left\{ c_1(\phi) + \frac{c_2(\phi) N_{R,2,0}}{\pi \varepsilon_{2,0}^2} \left[ \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|p_1| \cos \phi}{p_m} \right) \sin^{-1} |\varepsilon_{2,0}| \right. \right. \\ \left. \left. + \text{sign}(\varepsilon_{2,0}) \left( 3\varepsilon_{2,0} + \frac{2}{3} (2 + \varepsilon_{2,0}^2) \frac{|p_1| \cos \phi}{p_m} \right) \right] \sqrt{1 - \varepsilon_{2,0}^2} \right\}, \quad |\varepsilon_{2,0}| \leq 1, \quad (10.97)$$

where the steady-flow Reynolds number is

$$N_{R,2,0} = \frac{m |\dot{N}_{2,0}| 4r_h}{\phi A \mu} \quad (10.98)$$

and  $\varepsilon_{2,0} = m \dot{N}_{2,0} / \rho_m |U_1|$ . See Section 8.1.1 for explanation of the use of  $\dot{N}_{2,0}$  instead of  $\dot{N}$  here.

In STKPOWERLW, the expression for head-loss gradient is the same as the pessimistic case in turbulent DUCTs, but with  $f_M$  and  $df_M/dN_R$  given by the STKPOWERLW expression, Eq. (10.87), and with  $|p_1|/\gamma p_m$  replaced by  $|p_1|/p_m$  to model isothermal oscillations.

## 10.6. Pulse tubes and thermal buffer tubes

(For a general description of all of the STK\*\* segments, see the first paragraph of Section 10.5.)

### 10.6.1. STKDUCT and STKCONE

#### General description:

If  $dT_m/dx \neq 0$  and pore size or plate separation is much greater than thermal and viscous penetration depths, use STKDUCT or STKCONE. These segments are used mostly for pulse tubes and thermal buffer tubes.

In a STKCONE, the variable **fwall** (parameter **f**) specifies the wall thickness. If **fwall** < 1, the wall thickness is constant, but if **fwall** > 1, the wall thickness is proportional to the local cone radius (so the wall stress can be constant, which is the minimum-weight, minimum-thermal-conductance design).

The one-dimensional nature of DELTAEC calculations can be particularly unrealistic in STKDUCT and STKCONE, because DELTAEC assumes that  $T_m(x)$  and  $T_{\text{solid}}(x)$  are equal at all  $x$ , despite the fact that these are likely to be very different if any type of streaming occurs.

### Input variables, STKDUCT:

**Area** ( $\text{m}^2$ ) Cross-sectional area  $A_{\text{gas}}$  available to the gas, based on inside dimensions of the STKDUCT.

**Perim** (m) Perimeter  $\Pi$  of the inside cross section described above.

**Length** (m) The length  $\Delta x$  of the STKDUCT.

**WallA** ( $\text{m}^2$ ) The cross-sectional area  $A_{\text{solid}}$  of the wall material.

**Solid** The solid material comprising the walls.

### Input variables, STKCONE:

**AreaI** ( $\text{m}^2$ ) Cross-sectional area  $A_{\text{gas},I}$  available to the gas at the initial end of the STKCONE, based on inside dimensions.

**PerimI** (m) Perimeter  $\Pi_I$  of the initial inside cross section described above.

**Length** (m) The length  $\Delta x$  of the STKCONE.

**AreaF** ( $\text{m}^2$ ) Cross-sectional area  $A_{\text{gas},F}$  available to the gas at the final end of the STKCONE, based on inside dimensions.

**PerimF** (m) Perimeter  $\Pi_F$  of the final inside cross section described above.

**f\_wall** For  $\text{fwall} < 1$ , wall thickness =  $\text{fwall} \times$  initial perimeter, and is independent of  $x$ . For  $\text{fwall} > 1$ , wall thickness = local perimeter/ $\text{fwall}$ , so local wall cross-sectional area = (local perimeter) $^2$ / $\text{fwall}$ .

**Solid** The solid material comprising the walls.

### Potential targets:

None.

### Master–slave links:

In STKDUCT, the perimeter can be slaved to the area to maintain the cross-sectional shape when the area is changed.

In STKDUCT, the perimeter *and* the area can also be slaved to the length, to maintain both constant volume and constant cross-sectional shape when the length is changed.

In **STKDUCT**, the initial perimeter can be slaved to the initial area, the final perimeter can be slaved to the final area, or both perimeters can be slaved to both areas, so shapes are maintained when areas are changed.

The length and both perimeters of a **STKCONE** can be slaved to its two areas, to keep the cross-sectional shapes of the ends *and* the wall taper angle constant when either or both areas are changed.

The length of either of these segments can be slaved to the length of another segment to keep the sum of the two lengths constant.

(For an introduction to master–slave links, and a complete listing of linking options for all segments, see Section 2.7.2.)

### Format and examples:

```
STKDUCT boundary-layer approx
0.01 m2 area of gas
0.4 m perimeter (this duct is square)
1. m length
0.001 m2 wall material's cross-sectional area
stainless
```

```
STKCONE boundary-layer w/ taper
0.01 m2 area of gas
0.35 m perimeter
1. m length
sameas 8a
sameas 8b
0.001 fwall
stainless
```

Other examples of **STKDUCT** are in Sections 5.8.3 and 6.3.

### Calculations for $\dot{N} = 0$ :

In **STKDUCT** and **STKCONE**, pressure propagates according to Rott's wave equation, written in the form

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}}U_1, \quad (10.99)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \left(1 + \frac{(\gamma-1)f_\kappa}{1+\epsilon_s}\right) p_1 + \frac{\beta(f_\kappa - f_\nu)}{(1-f_\nu)(1-\sigma)(1+\epsilon_s)} \frac{dT_m}{dx} U_1, \quad (10.100)$$

subject to the condition that energy flow  $\dot{H}_{2,k}$  is independent of  $x$ , which imposes the following condition on  $T_m(x)$ :

$$\frac{dT_m}{dx} = \frac{\dot{H}_{2,k} - \frac{1}{2}\text{Re} \left[ p_1 \tilde{U}_1 \left(1 - \frac{T_m \beta (f_\kappa - \tilde{f}_\nu)}{(1+\epsilon_s)(1+\sigma)(1-\tilde{f}_\nu)}\right) \right]}{\frac{\rho_m c_p |U_1|^2}{2\omega A_{\text{gas}}(1-\sigma)|1-f_\nu|^2} \text{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1+\epsilon_s f_\nu/f_\kappa)}{(1+\epsilon_s)(1+\sigma)} \right] - A_{\text{gas}}k - A_{\text{solid}}k_s}. \quad (10.101)$$

For **STKDUCT** and **STKCONE**,

$$f_j = (1-i)\Pi\delta_j/2A, \quad \text{for } j = \kappa \text{ or } \nu, \quad (10.102)$$

$$\epsilon_s = \left(\frac{k\rho_m c_p}{k_s \rho_s c_s}\right)^{1/2} \frac{1}{\tanh[(1+i)\ell/\delta_s]}, \quad \text{where } \ell = \frac{\text{wall cross-sect area}}{\text{perimeter}}, \quad (10.103)$$

so long as  $2A/\Pi\delta_j > 30$ . For  $2A/\Pi\delta_j < 25$ , the functions are the same as for **STKCIRC**, with  $r_0(x) = 2A(x)/\Pi(x)$ . In between, a linear combination is used.

## Calculation with $\dot{N} \neq 0$ :

STKDUCT treats nonzero  $\dot{N}$  in the same way as STKCIRC. STKCONE is similar, with the additional variation of  $A$  with  $x$ . Remember to increase `Nint` if  $T_m(x)$  curves significantly.

## 10.7. Heat exchangers

DELTAEC heat exchangers are used to inject or remove heat. The first law of thermodynamics insists that this heat must equal the difference between upstream and downstream  $\dot{H}_{\text{tot}}$ . DELTAEC heat exchangers necessarily have surface area, so they experience both viscous and thermal dissipation of acoustic power.

Heat exchangers are collected into three groups here. The smooth, short HX and TX have parallel-plate and tubular geometry, respectively, and are described in Section 10.7.1. The short, tortuous porous heat exchangers SX and PX, for screens and power-law structures, respectively, are described in Section 10.7.2. The long, smooth, variable-temperature and variable-heat-density exchangers VXT1, VXT2, VXQ1, and VXQ2 are described in Section 10.7.3.

### 10.7.1. HX and TX

#### General description:

For a general description of all DELTAEC heat exchangers, see the initial paragraphs of the heat-exchanger documentation, Section 10.7.

In HX the thermoacoustic working gas is between parallel plates; in TX it is inside cylindrical tubes. Heat  $\dot{Q}$  is positive when it flows from the solid into the thermoacoustic gas, adding to  $\dot{H}_{\text{tot}}$ . The temperature difference between the solid temperature  $T_{\text{solid}}$  and the gas temperature  $T_m$  is proportional to  $\dot{Q}$ . The proportionality constant is not well validated, either theoretically or experimentally; we find that it usually seems to be accurate within a factor of 2.

HX and TX are assumed to be so short that the numerical integration from one end to the other is performed with a single fourth-order Runge-Kutta step. (Most other integrated segments, i.e., DUCT, CONE, MIX\*\*, STK\*\*, and VX\*\*, are integrated with `Nint` fourth-order Runge-Kutta steps.) See VX\*\* in Section 10.7.3 below for longer tubular heat exchangers.

#### Input variables, HX:

**Area** (m<sup>2</sup>) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ .

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A$ .

**Length** (m) The length  $\Delta x$  of the heat exchanger.

**y0** (m) The plate spacing is  $2y_0$ . In other words,  $2y_0$  is the gap available to the gas—not the center-to-center spacing of the plates.

**HeatIn** The heat  $\dot{Q}$  added to the thermoacoustic gas in the heat exchanger. Positive  $\dot{Q}$  indicates heat flow from solid to gas.

**Solid** The solid material on the inside surface of the HX. This is used only in the calculation of  $\epsilon_s$ , which seldom differs significantly from its value for a solid with infinite conductivity and specific heat.

### Input variables, TX:

**Area** (m<sup>2</sup>) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ .

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A$ .

**Length** (m) The length  $\Delta x$  of the heat exchanger.

**radius** (m) The radius of each tube,  $r_0$ .

**HeatIn** The heat  $\dot{Q}$  added to the thermoacoustic gas in the heat exchanger. Positive  $\dot{Q}$  indicates heat flow from solid to gas.

**Solid** The solid material on the inside surface of the TX. This is used only in the calculation of  $\epsilon_s$ , which seldom differs significantly from its value for a solid with infinite conductivity and specific heat.

### Potential targets:

**TSolid** If used as a target, DELTAEC's shooting method compares  $T_{\text{solid}}$  to this value.

### Master-slave links:

In TX, the porosity  $\phi$  can be slaved to other parameters via the equation  $\phi = \min(cr_0^2/A, 1)$ , with  $c$  calculated when the link is first enabled via  $c = \phi A/r_0^2$ . This link maintains a constant number of holes, up to a maximum porosity of 1.

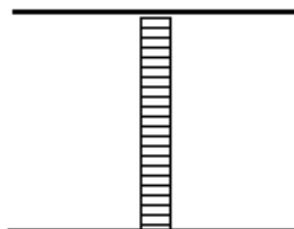
The length of HX or TX can be slaved to the length of another segment to keep the sum of the two lengths constant.

(For an introduction to master-slave links, and a complete list of linking options for all segments, see Section 2.7.2.)

### Format and examples:

```
HX      parallel-plate heat exchanger
sameas 1 Area
0.600   GasA/A
6.35e-3 m Length
1.9e-4 m y0 = half of plate spacing
-20.0   W HeatIn
300.    K Est-T Target
copper   solid
```

```
TX      tube-in-shell heat exchanger
0.2     a Area      m^2
.188    b GasA/A
.400    c Length    m
6.350E-03 d radius  m (radius of each tube)
1.818E+05 e HeatIn  W
possible target
nickel   Solid type
```



Other examples of HX appear in Chapter 5.

## Calculations:

In HX and TX, wave propagation is calculated using

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}}U_1, \quad (10.104)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \left(1 + \frac{(\gamma-1)f_\kappa}{1+\epsilon_s}\right) p_1. \quad (10.105)$$

HX uses parallel plate geometry in computing  $f_\kappa$ ,  $f_\nu$ , and  $\epsilon_s$ :

$$f_j = \frac{\tanh[(1+i)y_0/\delta_j]}{(1+i)y_0/\delta_j}, \quad \text{for } j = \kappa \text{ or } \nu, \quad (10.106)$$

$$\epsilon_s = \left(\frac{k\rho_m c_p}{k_s\rho_s c_s}\right)^{1/2}. \quad (10.107)$$

Similarly, TX uses cylindrical geometry in computing  $f_\kappa$ ,  $f_\nu$ , and  $\epsilon_s$ : For  $r_0/\delta_j < 25$ ,  $f_\kappa$  and  $f_\nu$  are calculated using complex Bessel functions

$$f_j = \frac{2J_1[(i-1)r_0/\delta_j]}{(i-1)(r_0/\delta_j)J_0[(i-1)r_0/\delta_j]} \quad \text{for } j = \kappa \text{ or } \nu. \quad (10.108)$$

For  $r_0/\delta_j > 30$ , the boundary-layer approximation is used:

$$f_j = (1-i)\delta_j/r_0 \quad \text{for } j = \kappa \text{ or } \nu. \quad (10.109)$$

For intermediate values, linear interpolation is used to make a smooth match between the two regimes. In both cases,  $\epsilon_s$  is calculated using

$$\epsilon_s = \left(\frac{k\rho_m c_p}{k_s\rho_s c_s}\right)^{1/2}. \quad (10.110)$$

In TX, the radius  $r_0$  is that of one circular pore, so for a heat exchanger comprised of  $N$  circular pores, the total cross-sectional area available to the working gas is  $N\pi r_0^2 = (\text{Area})(\text{GasA/A})$ .

HX and TX have no turbulence algorithms.

For  $\dot{N} = 0$ , temperature and power change according to

$$\frac{dT_m}{dx} = 0, \quad (10.111)$$

$$\frac{d\dot{H}_{\text{tot}}}{dx} = \frac{d\dot{H}_{2,k}}{dx} = \frac{\dot{Q}}{\Delta x}, \quad (10.112)$$

$$T_{\text{solid}} = T_m + \frac{\dot{Q}}{h A_{\text{gas}} x_{\text{eff}}/r_h}, \quad (10.113)$$

where  $\Delta x$  is the heat exchanger length and  $\dot{Q}$  is the heat added to the heat exchanger. Positive heat  $\dot{Q}$  (parameter **e**) flows into the apparatus. The heat flow  $\dot{Q}$  is an input for

each pass of DELTAEC, and it is often used as a guess for the shooting method described in Section 2.5. The heat-transfer coefficient  $h$  is estimated with

$$h = \frac{k}{y_{\text{eff}}}, \quad (10.114)$$

and

$$x_{\text{eff}} = \min[2|\xi_1|, \Delta x] \quad (10.115)$$

$$y_{\text{eff}} = \min[\delta_\kappa, r_h], \quad (10.116)$$

with hydraulic radius  $r_h$  equal to  $y_0$  for HX and equal to  $r_0/2$  for TX. These expressions may be quite inaccurate, but we believe they are more useful than nothing. A little experimental evidence for  $T_{\text{solid}} - T_m$  is presented in Ref. [23]. A user who mistrusts these expressions can use the gas temperature (available through RPN, and as an output in the adjacent stack segment) instead of the solid temperature for plotting or targeting, and/or use RPNs to construct custom expressions instead of Eqs. (10.113)–(10.116).

For  $\dot{N} \neq 0$ , the wave and energy calculations are unchanged, so all of the heat goes into  $\dot{H}_{2,k}$  and none into the steady flow. This seems fine for weak steady flows, such as weak Gedeon streaming when a TASHE is operated slightly away from the conditions at which its gas diode enforces  $\dot{N} = 0$ . For strong steady flow, the user can override the built-in calculations with an RPN such as

```
13 RPN Fix Tm. Inp = frac of heat that should go to mean flow
0.25
H2k inp 12e * - =H2k ; inp 12e * Ndot / m / cp / Tm + =Tm
```

following the heat exchanger, where 12e is the heat in the upstream heat exchanger and 13a is the steady-flow heat fraction. That fraction is often guessed, to give DELTAEC enough flexibility to meet nearby temperature and energy boundary conditions. Alternatively, the fraction can also be set equal to a simple expression like  $(1 + \rho_m |U_1| / |\dot{N}|m)^{-1}$  in an RPN, supplying an experimentally unsupported but plausible estimate of how power divides itself between the steady flow and the acoustic flow.

For  $\dot{N} \neq 0$ , the gradients in  $p_{2,0,HL}$  are calculated using Eq. (10.70) for HX and Eq. (10.69) for TX.

### 10.7.2. SX and PX

#### General description:

For a general description of all DELTAEC heat exchangers, see the initial paragraphs of the heat-exchanger documentation, Section 10.7.

In SX, the thermoacoustic working gas flows through a pile of woven screens. The PX segment allows the user to specify heat-exchanger parameters for which the steady-flow friction factor and heat transfer coefficient are known power laws in Reynolds number. The SX calculations assume fairly good thermal contact between gas and solid, so these calculations are valid only for hydraulic radius smaller than thermal and viscous penetration depths. There is no warning if this condition is not met.

Heat  $\dot{Q}$  is positive when it flows from the solid into the thermoacoustic gas, adding to  $\dot{H}_{\text{tot}}$ . In **SX**, typically the screen material has high thermal conductivity and the area is small, because typically the heat must flow radially through the screen wires, from the pressure vessel around the screens to the thermoacoustic gas.

The temperature difference between the solid temperature  $T_{\text{solid}}$  and the gas temperature  $T_m$  is proportional to  $\dot{Q}$ . The proportionality constant is not well supported, either theoretically or experimentally; we find that it seems to be accurate within a factor of 2.

**SX** and **PX** are assumed to be so short that the numerical integration from one end to the other is performed with a single fourth-order Runge-Kutta step. (Most other integrated segments, i.e., **DUCT**, **CONE**, **MIX\*\***, **STK\*\***, and **VX\*\***, are integrated with  $N_{\text{int}}$  fourth-order Runge-Kutta steps.)

### Input variables, SX:

**Area** (m<sup>2</sup>) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ .

**VolPor** The volumetric porosity  $\phi$ . This is the ratio of the volume of gas in the **SX** to the total volume of the heat exchanger. For plain square-weave screen with  $m$  wires per unit length and wire diameter  $d_{\text{wire}}$ , we sometimes use

$$\phi \simeq 1 - \frac{\pi m d_{\text{wire}}}{4}, \quad (10.117)$$

as recommended by Organ [27], or we weigh the pile of screens, measure its overall dimensions  $A$  and  $\Delta x$ , and use the known density of its wire material to calculate  $\phi$ .

**Length** (m) The length  $\Delta x$  of the heat exchanger.

**rh** (m) The hydraulic radius  $r_h$  of the **SX**. This is the ratio of the gas volume to the gas–solid contact area. For plain square-weave screen with wire diameter  $d_{\text{wire}}$ , we sometimes use

$$r_h \simeq d_{\text{wire}} \frac{\phi}{4(1 - \phi)}, \quad (10.118)$$

as recommended by Organ [27].

**HeatIn** The heat  $\dot{Q}$  added to the thermoacoustic gas in the heat exchanger. Positive  $\dot{Q}$  indicates heat flow from solid to gas.

**Solid** The solid material of the wires. This is used only in the calculation of  $\epsilon_s$ , which seldom differs significantly from its value for a solid with infinite conductivity and specific heat in **SXs**.

### Input variables, PX:

**Area** (m<sup>2</sup>) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}}$ .

**VolPor** The volumetric porosity  $\phi$ . This is the ratio of the volume of gas in the heat exchanger to the total volume of the heat exchanger.



**Length** (m) The length  $\Delta x$  of the heat exchanger.

**rh** (m) The hydraulic radius  $r_h$  of the heat exchanger. This is the ratio of the gas volume to the gas–solid contact area.

**HeatIn** The heat  $\dot{Q}$  added to the thermoacoustic gas in the heat exchanger. Positive  $\dot{Q}$  indicates heat flow from solid to gas.

**f con** The friction-factor parameter  $f_{con}$  in the equations below.

**f exp** The friction-factor parameter  $f_{exp}$  in the equations below.

**h con** The heat-transfer parameter  $h_{con}$  in the equations below.

**h exp** The heat-transfer parameter  $h_{exp}$  in the equations below.

**Solid** The solid with which the PX is made. This is used only in the calculation of  $\epsilon_s$ , which seldom differs significantly from its value for a solid with infinite conductivity and specific heat in heat exchangers.

**Potential targets:**

**TSolid** If used as a target, DELTAEC’s shooting method compares  $T_{solid}$  to this value.

**Master–slave links:**

The length of either of these segments can be slaved to the length of another segment to keep the sum of the two lengths constant.

(For an introduction to master–slave links, and a complete listing of linking options for all segments, see Section 2.7.2.)

**Format and examples:**

```

SX      Hot heat exchanger
1.029E-03 a Area      m^2  total cross sectional area
0.690   b VolPor     volumetric porosity
2.000E-02 c Length   m
6.450E-05 d rh       m    hydraulic radius
-284.   e HeatIn     W
300.   f Est-T      K      (t)
copper  Solid type

PX
1.0E-4  a Area      m^2  total cross-sectional area
0.70   b GasA/A    volumetric porosity
0.04   c Length   m
4.000E-05 d rh       m    hydraulic radius = gas volume / gas-solid contact area
900.   e HeatIn     W
0.07   f f_con
0.22   g f_exp
0.035  h h_con
0.22   i h_exp
copper  Solid type

```



Examples of SX occur throughout Sections 5.8 and 6.3.

## Calculations:

In **SX**,  $dp_1/dx$  and  $d\langle u_1 \rangle/dx$  are calculated in the same way as in **STKSCREEN**, as described in [62]:

$$\frac{dp_1}{dx} = -i\omega\rho_m \left[ 1 + \frac{(1-\phi)^2}{2(2\phi-1)} \right] \langle u_1 \rangle - \frac{\mu}{r_h^2} \left( \frac{c_1(\phi)}{8} + \frac{c_2(\phi)N_{R,1}}{3\pi} \right) \langle u_1 \rangle, \quad (10.119)$$

$$\frac{d\langle u_1 \rangle}{dx} = -\frac{i\omega\gamma}{\rho_m a^2} p_1 + \frac{i\omega T_m \beta^2}{\rho_m c_p} \frac{\epsilon_s + (g_c + e^{2i\theta_p} g_v)\epsilon_h}{1 + \epsilon_s + (g_c + e^{2i\theta_T} g_v)\epsilon_h} p_1, \quad (10.120)$$

using

$$c_1(\phi) = 1268 - 3545\phi + 2544\phi^2, \quad c_2(\phi) = -2.82 + 10.7\phi - 8.6\phi^2, \quad (10.121)$$

$$b(\phi) = 3.81 - 11.29\phi + 9.47\phi^2, \quad (10.122)$$

$$N_{R,1} = 4|\langle u_1 \rangle| r_h \rho_m / \mu, \quad (10.123)$$

$$\epsilon_s = \phi \rho_m c_p / (1-\phi) \rho_s c_s, \quad \epsilon_h = 8ir_h^2/b(\phi)\sigma^{1/3}\delta_\kappa^2, \quad (10.124)$$

$$\delta_\kappa^2 = 2k/\omega\rho_m c_p, \quad (10.125)$$

$$\theta_p = \text{phase}(\langle u_1 \rangle) - \text{phase}(p_1), \quad \theta_T = \text{phase}(\langle u_1 \rangle) - \text{phase}(\langle T \rangle_{u,1}), \quad (10.126)$$

$$g_c = \frac{2}{\pi} \int_0^{\pi/2} \frac{dz}{1 + N_{R,1}^{3/5} \cos^{3/5}(z)}, \quad g_v = -\frac{2}{\pi} \int_0^{\pi/2} \frac{\cos(2z) dz}{1 + N_{R,1}^{3/5} \cos^{3/5}(z)}. \quad (10.127)$$

The spatial average oscillatory velocity  $\langle u_1 \rangle = \langle U_1 \rangle / \phi A$ , where  $\phi$  is volumetric porosity and  $A$  is heat-exchanger cross-sectional area. These expressions were derived with the assumption that the thermal and viscous penetration depths are much larger than  $r_h$ .

The **PX** segment can be used when the steady-flow friction factor and heat-transfer coefficient are known power laws in Reynolds number  $N_R$ . The derivation assumes fairly good thermal contact between gas and solid. Area  $A$  is the total cross-sectional area of the heat exchanger, **VolPor**  $\phi$  is its volumetric porosity, and  $\Delta x$  is its length, so  $\phi A \Delta x$  is the total volume of gas in the heat exchanger. The steady-state friction factor and heat-transfer coefficients must be known by the user in power-law forms:

$$f = f_{\text{con}} (N_R)^{-f_{\text{exp}}}, \quad (10.128)$$

$$N_{St} \sigma^{2/3} = h_{\text{con}} (N_R)^{-h_{\text{exp}}}, \quad (10.129)$$

where  $N_{St} = N_{Nu}/\sigma N_R$  is the Stanton number, and the Reynolds number  $N_R$  is defined in the usual way as

$$N_R = \frac{4U r_h \rho_m}{\phi A \mu} = \frac{4\langle u \rangle r_h \rho_m}{\mu}. \quad (10.130)$$

Note: this is Fanning friction factor, the friction factor used by Kays and London, so instantaneously

$$\frac{dp}{dx} = \frac{f}{r_h} \frac{1}{2} \rho \langle u \rangle^2 = \frac{\mu}{8r_h^2} f N_R \langle u \rangle. \quad (10.131)$$

The complex pressure gradient is computed using

$$\frac{dp_1}{dx} = -i\omega\rho_m \langle u_1 \rangle - I_f \frac{\mu}{8r_h^2} f_{\text{con}} |N_{R,1}|^{1-f_{\text{exp}}} \langle u_1 \rangle \quad (10.132)$$

where

$$I_f = \frac{2}{\pi} \int_0^\pi \sin^{3-f_{\text{exp}}} \omega t d(\omega t). \quad (10.133)$$

Thermal-relaxation effects due to oscillating pressure are computed using the same equations for  $d\langle u_1 \rangle/dx$  as for screen heat exchangers but with

$$g_c = |N_{R,1}|^{h_{\text{exp}}-1} \frac{2}{\pi} \int_0^{\pi/2} \cos^{h_{\text{exp}}-1} \omega t d(\omega t), \quad (10.134)$$

$$g_v = -|N_{R,1}|^{h_{\text{exp}}-1} \frac{2}{\pi} \int_0^{\pi/2} \cos 2\omega t \cos^{h_{\text{exp}}-1} \omega t d(\omega t), \quad (10.135)$$

$$b(\phi) = h_{\text{con}}. \quad (10.136)$$

The solid temperature is computed relative to the gas mean temperature using

$$\Delta T = I_h \frac{\dot{Q}}{k h_{\text{con}} |N_{R,1}|^{1-h_{\text{exp}}}} \frac{r_h^2}{\phi A x_{\text{eff}}}, \quad (10.137)$$

where  $x_{\text{eff}} = \min\{2|\langle u_1 \rangle|/\omega, \Delta x\}$ ,  $N_{R,1}$  is the Reynolds number amplitude (based on the *amplitude* of the velocity), and

$$I_h = \frac{4}{\pi} \int_0^{\pi/2} \cos^{h_{\text{exp}}+1} \omega t d(\omega t) = \frac{h_{\text{exp}}}{h_{\text{exp}}+1} \frac{4}{\pi} \int_0^{\pi/2} \cos^{h_{\text{exp}}-1} \omega t d(\omega t). \quad (10.138)$$

(The second form of  $I_h$ , obtained from the first via integration by parts, expresses  $I_h$  in terms of  $g_c$  above.) To maintain DELTAEC's high speed, the trigonometric integrals are not evaluated by DELTAEC; we use simple functional fits to these integrals in DELTAEC.

The head-loss pressure gradient  $dp_{2,0}/dx$  in **SX** is calculated in the same way as for **STKSCREENS**. The nonlinear nature of the screen friction factor  $f = c_1(\phi)/N_R + c_2(\phi)$  also causes the mean flow to affect  $dp_1/dx$  in **SX**, according to

$$\frac{dp_1}{dx} = -\frac{\mu U_1}{8r_h^2 \phi A} [c_1(\phi) + 2c_2(\phi)N_{R,2,0}], \quad |\varepsilon_{2,0}| \geq 1, \quad (10.139)$$

$$= -\frac{\mu U_1}{r_h^2 \phi A} \left\{ \frac{c_1(\phi)}{8} + \frac{c_2(\phi)N_{R,1}}{3\pi} \left[ \left( 1 + \frac{\varepsilon_{2,0}^2}{2} \right) \sqrt{1 - \varepsilon_{2,0}^2} + \frac{3}{2} \varepsilon_{2,0} \sin^{-1} \varepsilon_{2,0} \right] \right\}, \quad |\varepsilon_{2,0}| \leq 1, \quad (10.140)$$

where the acoustic Reynolds-number amplitude is

$$N_{R,1} = \frac{|U_1| 4r_h \rho_m}{\phi A \mu}. \quad (10.141)$$

[Note  $N_{R,2,0}$  in Eq. (10.139) but  $N_{R,1}$  in Eq. (10.140) — *not* a typographical error.] We did not incorporate a mean flow effect on  $dp_1/dx$  in **STKSCREEN** because the energy equation restricts the magnitude of  $\dot{N}$  in stacks. No such restriction exists in heat exchangers.

In **PX**,  $dp_{2,0}/dx$  is calculated just as in **STKPOWERLW**. The mean flow affects  $dp_1/dx$  in **PX** according to

$$\begin{aligned} \frac{dp_1}{dx} &= -i\omega\rho_m \langle u_1 \rangle \\ &\quad - \frac{\rho_m |\langle u_1 \rangle| \langle u_1 \rangle}{4r_h} \left[ |\varepsilon_{2,0}| f_{M,\max} + \frac{(3 - 2|\varepsilon_{2,0}|)(1 - 2|\varepsilon_{2,0}|)}{8(1 + |\varepsilon_{2,0}|)} \frac{df_M}{dN_R} N_{R,\max} \right], \quad |\varepsilon_{2,0}| \geq 1, \end{aligned} \quad (10.142)$$

$$\begin{aligned} &= -i\omega\rho_m \langle u_1 \rangle - \frac{\rho_m |\langle u_1 \rangle| \langle u_1 \rangle}{3\pi r_h} \left\{ \left[ \left(1 + \frac{\varepsilon_{2,0}^2}{2}\right) \sqrt{1 - \varepsilon_{2,0}^2} + \frac{3}{2} \varepsilon_{2,0} \sin^{-1} \varepsilon_{2,0} \right] f_{M,\max} \right. \\ &\quad \left. - \left[ \left(1 + \frac{\varepsilon_{2,0}^2}{2}\right) \sqrt{1 - \varepsilon_{2,0}^2} + \frac{3}{2} \varepsilon_{2,0} \sin^{-1} \varepsilon_{2,0} - \frac{9\pi}{32} \frac{1 + 4\varepsilon_{2,0}^2}{1 + |\varepsilon_{2,0}|} \right] \frac{df_M}{dN_R} N_{R,\max} \right\}, \quad |\varepsilon_{2,0}| \leq 1. \end{aligned} \quad (10.143)$$

which is a hybrid of Eq. (10.132) here and Eqs. (A13) and (A14) in Ref. [42].

For  $\dot{N} = 0$ , temperature and power change according to

$$\frac{dT_m}{dx} = 0, \quad (10.144)$$

$$\frac{d\dot{H}_{\text{tot}}}{dx} = \frac{\dot{Q}}{\Delta x}, \quad (10.145)$$

where  $\Delta x$  is the heat exchanger length and  $\dot{Q}$  is the heat added to the heat exchanger. Positive heat  $\dot{Q}$  (parameter “e”) flows into the apparatus. The heat flow  $\dot{Q}$  is an input for each pass of **DELTAEC**, and it is often used as a guess for the shooting method described in Section 2.5.

For  $\dot{N} \neq 0$ , the wave and energy calculations are unchanged, so all of the heat goes into  $\dot{H}_{2,k}$  and none into the steady flow. This seems fine for weak steady flows, such as weak Gedeon streaming when a **TASHE** is operated slightly away from the conditions at which its gas diode enforces  $\dot{N} = 0$ . For strong steady flow, the user can override the built-in calculations with an **RPN** such as

```
13 RPN Fix Tm. Inp = frac of heat that should go to mean flow
0.25
H2k inp 12e * - =H2k ; inp 12e * Ndot / m / cp / Tm + =Tm
```

following the heat exchanger, where **12e** is the heat in the upstream heat exchanger and **13a** is the steady-flow heat fraction. That fraction is often guessed, to give **DELTAEC** enough flexibility to meet nearby temperature and energy boundary conditions. Alternatively, the fraction can also be set equal to a simple expression like  $\left(1 + \rho_m |U_1| / |\dot{N}|m\right)^{-1}$  in an **RPN**, supplying an experimentally unsupported but plausible estimate of how power divides itself between the steady flow and the acoustic flow.

In **SXs**, the solid temperature is computed relative to gas mean temperature using

$$\Delta T = \frac{\dot{Q} r_h^2 (g_c - g_v)}{k b(\phi) \phi A x_{\text{eff}}} \quad (10.146)$$

where  $x_{\text{eff}} = \min\{2|\langle \xi_1 \rangle|, \Delta x\}$ . In **PXs**, the solid temperature is computed relative to the gas mean temperature using

$$\Delta T = I_h \frac{\dot{Q}}{k h_{\text{con}} N_{R,1}^{1-h_{\text{exp}}}} \frac{r_h^2}{\phi A x_{\text{eff}}}, \quad (10.147)$$

where  $x_{\text{eff}} = \min\{2|\langle \xi_1 \rangle|, \Delta x\}$ ,  $N_{R,1}$  is the Reynolds number based on the amplitude of the velocity, and  $I_h$  is defined above. Then

$$T_{\text{solid}} = \max [T_m(0), T_m(\Delta x)] + \Delta T \quad \text{if } \dot{Q} > 0, \quad (10.148)$$

$$= \min [T_m(0), T_m(\Delta x)] + \Delta T \quad \text{if } \dot{Q} < 0, \quad (10.149)$$

These expressions may be very inaccurate, but we believe they are more useful than nothing. The user who mistrusts them can use the gas temperature (available as an output in the adjacent stack segment, and through **RPN**) instead of the solid temperature for plotting or targeting, and/or use **RPNs** to construct customized expressions for Eqs. (10.146)–(10.149).

### 10.7.3. **VXQ1, VXQ2, VXT1, and VXT2**

#### General description:

For a general description of all **DELTAEC** heat exchangers, see the initial paragraphs of the heat-exchanger documentation, Section 10.7.

Beginning in **DELTAEC** version 6.0, the “variable” heat exchanger segments **VXT1**, **VXT2**, **VXQ1**, and **VXQ2** are included, allowing gas temperature  $T_m$  and/or heat flux per unit length  $\dot{q}$  to vary with  $x$ . The two **VXQ\*** segments further allow  $T_{\text{solid}}$  to depend on  $x$ . Hence, these segments combine some features of previous heat-exchanger segments with some features of previous stack segments.

The third character in the segment name, **T** or **Q**, determines which of two extreme, limiting heat-transfer conditions is assumed for the non-thermoacoustic side of the heat exchanger. In **VXT\***,  $T_{\text{solid}}$  is independent of  $x$ . Thinking of a shell-and-tube heat exchanger in which the thermoacoustic processes take place inside the tubes while a heat-transfer liquid passes through the shell, the idea is that **VXT\*** should be used when the shell-side flow has such a high heat-transfer coefficient (e.g., vigorous flow of water) that the solid surfaces of the heat exchanger tubes are fixed in temperature, independent of  $x$ . In **VXQ\***, the heat flux delivered to/from the heat exchanger per unit length is independent of  $x$ . In the shell-and-tube situation, the idea is that **VXQ\*** should be used when the shell-side gas is weakly coupled to the solid (e.g., combustion products at low Reynolds number) so the temperature difference between the shell-side gas and the solid surface can be much larger than any changes in temperature along the solid surface. To decide which case is most appropriate for individual circumstances, use a standard heat-transfer book to estimate

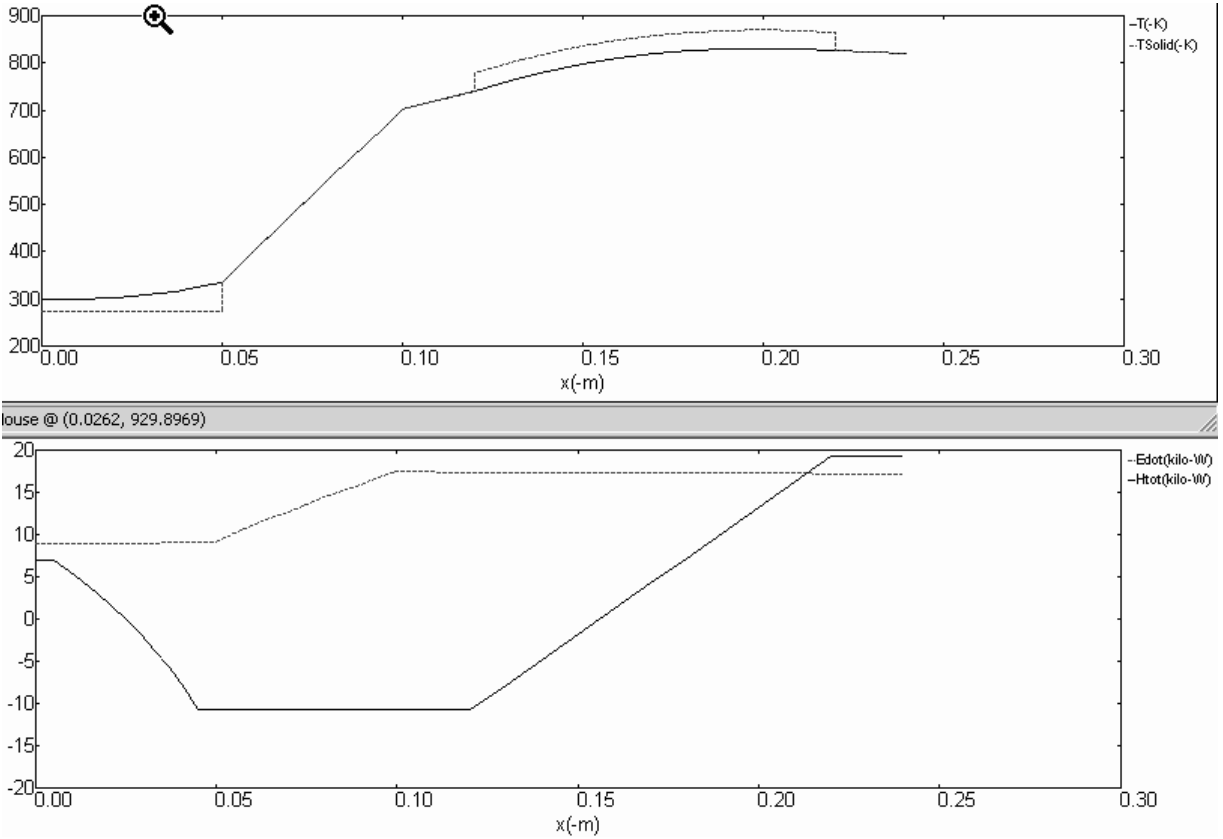


Figure 10.6: Example results from VXT1, STKSCREEN, VXQ1 segments in series in a Stirling engine. The VXT1 extends from  $x = 0$  to  $x = 0.05$  m, with tubesheets 0.005 m long at each end. The STKSCREEN extends from  $x = 0.05$  m to  $x = 0.10$  m. The VXQ2 extends from  $x = 0.10$  m to  $x = 0.24$  m, with tubesheets 0.02 m long on each end. Temperatures in upper panel. The gas temperature (solid line) varies with  $x$  in all three segments. The solid temperature (dashed line) does not vary with  $x$  in the VXT1 segment. Lower panel shows  $\dot{H}_{\text{tot}}$  (solid line) and  $\dot{E}$  (dashed line).

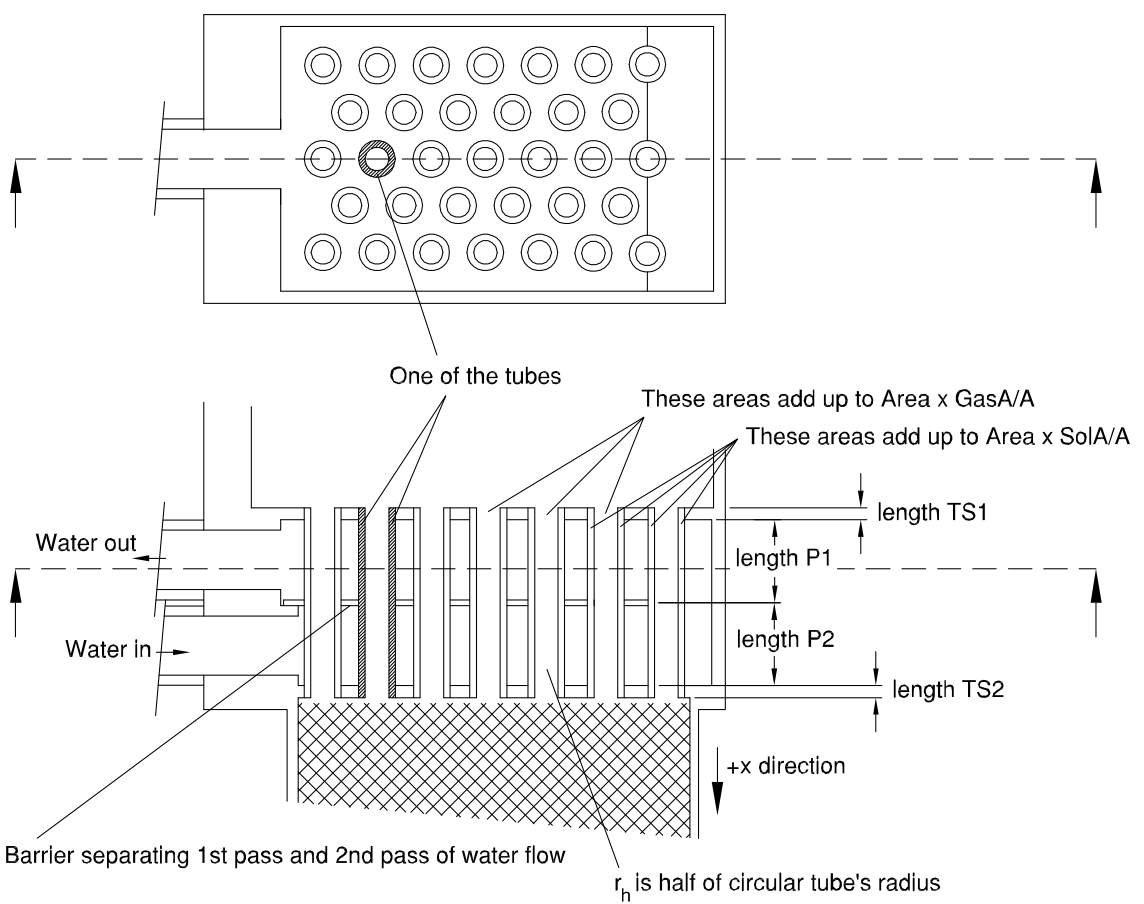


Figure 10.7: Two perpendicular cross sections through a VXT2, sandwiched between a duct above and a regenerator below. Thermoacoustic oscillations in the vertical direction pass through the insides of 33 tubes, of which 7 are exposed in the lower part of this figure. Cooling water passes through the “shell” in two passes: rightward across the lower half, and then leftward across the upper half.

the heat-transfer coefficients on the shell side and on the thermoacoustic side. If the heat transfer is much better on the shell side, use VXT\*. If the heat transfer is much better on the thermoacoustic side, use VXQ\*.

The fourth character in the segment name, 1 or 2, allows the segment to be formed of 1 or 2 separate heat-transfer pieces along  $x$  (in addition to the two “tubesheets” at either end). Thus, “2” is intended for heat exchangers with two “passes” on the shell side. Figure 10.7 shows some of the geometry parameters for a VXT2 (or VXQ2).

To understand the philosophy behind the VX\*\* calculations, consider Fig. 10.8, which shows a length  $dx$  of this segment. In contrast with the STK\*\* segments, where the mean temperature of the solid is assumed to be the same as the mean temperature of the gas, here those time-mean temperatures can be different, so large time-averaged heat flows can be driven from one to the other. Let  $T_m(x)$  be the mean temperature of the gas and  $T_{\text{solid}}(x)$  be the mean temperature of the solid. Both may be functions of  $x$ . Artificially, assume that  $T_m$  in the gas is independent of the coordinate perpendicular to  $x$ . This is equivalent to

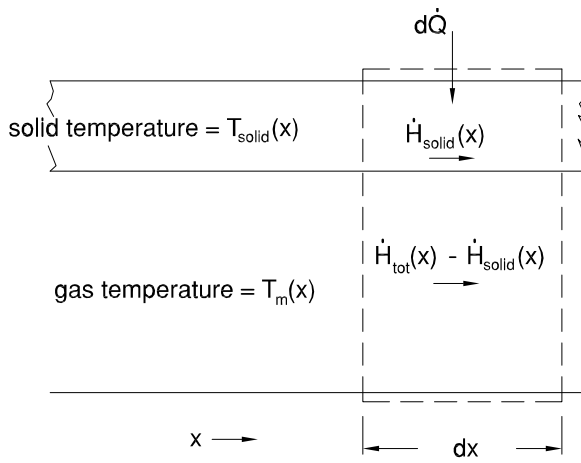


Figure 10.8: Schematic of a short section of a VX\*\* segment. The solid lines show the edges of the solid and the gas. The dashed box is the short control volume,  $dx$  long, used to write the conservation laws for this situation. The space above the solid represents the external heat-transfer gas such as water or combustion gases, which transfer heat at rate  $d\dot{Q}$  to the control volume.

assuming that the heat-transfer bottleneck for the time-averaged heat transfer occurs in a thin layer adjacent to the surface.

### Input variables, VXQ1 and VXQ2:

**Area** ( $\text{m}^2$ ) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}} + A_{\text{shell-side}}$ .

**GasA/A** The fraction of the total area that is available to gas,  $A_{\text{gas}}/A$ .

**SolA/A** The fraction of the total area occupied by solid that contributes to solid thermal conduction in the  $x$  direction,  $A_{\text{solid}}/A$ .

**rh** (m) The hydraulic radius  $r_h$  of the gas channels. Hydraulic radius is defined as the ratio of the gas volume to the gas–solid contact area. For the present case of an  $x$ -independent channel cross section,  $r_h = A_{\text{gas}}/\Pi$  where  $\Pi$  is the inside perimeter of the channel.

**LenTS1** (m) The length of the first tubesheet in the heat exchanger. This region experiences viscous and thermal-hysteresis loss, but not time-averaged heat transfer.

**Len P1** (m) The length of the heat-transfer region in VXQ1; the length of the first of two heat-transfer regions in VXQ2.

**Len P2** (m) The length of the second heat-transfer region in VXQ2. (Not present in VXQ1.)

**LenTS2** (m) The length of the second tubesheet in the heat exchanger. This region experiences viscous and thermal-hysteresis loss, but not time-averaged heat transfer.



**HeatP1** The heat added to the thermoacoustic gas in VXQ1. The heat added to the thermoacoustic gas in the first heat-transfer region in VXQ2. Positive heat indicates heat flow from solid to gas.

**HeatP2** The heat added to the thermoacoustic gas in the second heat-transfer region in VXQ2. (Not present in VXQ1.) Positive heat indicates heat flow from solid to gas.

**FracQN** If “Enable Ndot” is checked in an upstream BEGIN, this variable,  $F_{\dot{Q}\dot{N}}$ , determines what fraction of heat is added to  $\dot{H}_{\dot{N}} = \dot{N}mw_m$ . (The remainder is added to  $\dot{H}_{2,k}$ .) In VXQ2, the same value of  $F_{\dot{Q}\dot{N}}$  is applied to both passes.

**Solid** The solid material in the heat exchanger.

### Input variables, VXT1 and VXT2:

**Area** ( $\text{m}^2$ ) The total cross-sectional area  $A = A_{\text{gas}} + A_{\text{solid}} + A_{\text{shell-side}}$ .

**GasA/A** The fraction of the total area that is available to gas,  $A_{\text{gas}}/A$ .

**SolA/A** The fraction of the total area occupied by solid that contributes to solid thermal conduction in the  $x$  direction,  $A_{\text{solid}}/A$ .

**rh** (m) The hydraulic radius  $r_h$  of the gas channels. Hydraulic radius is defined as the ratio of the gas volume to the gas–solid contact area. For the present case of an  $x$ -independent channel cross section,  $r_h = A_{\text{gas}}/\Pi$  where  $\Pi$  is the inside perimeter of the channel.

**LenTS1** (m) The length of the first tubesheet in the heat exchanger. This region experiences viscous and thermal-hysteresis loss, but not time-averaged heat transfer.

**Len P1** (m) The length of the heat-transfer region in VXT1; the length of the first of two heat-transfer regions in VXT2.

**Len P2** (m) The length of the second heat-transfer region in VXT2. (Not present in VXT1.)

**LenTS2** (m) The length of the second tubesheet in the heat exchanger. This region experiences viscous and thermal-hysteresis loss, but not time-averaged heat transfer.

**TSolP1** The solid temperature in the heat-transfer region in VXT1. The solid temperature in the first heat-transfer region in VXT2.

**TSolP2** The solid temperature in the second heat-transfer region in VXT2. (Not present in VXT1.)

**FracQN** If “Enable Ndot” is checked in an upstream BEGIN, this variable,  $F_{\dot{Q}\dot{N}}$ , determines what fraction of heat is added to  $\dot{H}_{\dot{N}} = \dot{N}mw_m$ . (The remainder is added to  $\dot{H}_{2,k}$ .) In VXT2, the same value of  $F_{\dot{Q}\dot{N}}$  is applied to both passes.

**Solid** The solid material in the heat exchanger.

## Potential targets:

**HeatP1 or HeatP2** If used as a target in VXT1 or VXT2, DELTAEC's shooting method compares its calculated total heat for the corresponding heat-transfer region to this value.

There are no built-in potential targets in VXQ1 and VXQ2. Use RPN targets as needed.

## Master-slave links:

None.

## Format and examples:

```
VXT2          compare to segment 13
sameas 13a a Area          m^2          1.9116 A |p|          Pa
sameas 13b b GasA/A        -0.1565 B Ph(p)         deg
sameas 13c c SolA/A        9.9874E-07 C |U|        m^3/s
sameas 13d d rh            m            90.003 D Ph(U)         deg
sameas 13e e LenTS1        m            7.0105 E Htot          W
sameas 13f f Len P1        m            -2.6557E-09 F Edot       W
sameas 13g g Len P2        m            1.0009 G HeatP1            W
sameas 13h h LenTS2        m            5.0097 H HeatP2            W
0.00        i HeatP1        W            (t) 1605.5 I qmaxP1            W/m^2
0.00        j HeatP2        W            (t) 8034.8 J qmaxP2            W/m^2
321.70      k TSolP1        K            0.0000 K H2k          W
408.00      l TSolP2        K
1.0000      m FracQN
stainless   Solid type
```

Section 6.4 gives an example of the use of VXQ1.

## Calculations for $\dot{N} = 0$ :

The first-order continuity and momentum equations are taken to be unchanged by the time-averaged heat transfer. Hence,  $p_1$  and  $U_1$  evolve according to

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}}U_1, \quad (10.150)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \left(1 + \frac{(\gamma-1)f_\kappa}{1+\epsilon_s}\right) p_1 + \frac{\beta(f_\kappa - f_\nu)}{(1-f_\nu)(1-\sigma)(1+\epsilon_s)} \frac{dT_m}{dx} U_1, \quad (10.151)$$

in all four VX\*\* segments, just the same as in stacks. However, the energy equation used in stacks does not suffice here:  $\dot{H}_{2,k,\text{gas}}$  and  $\dot{H}_{\text{solid}}$  should be considered separately instead of only considering their sum  $\dot{H}_{2,k}$ , and the gas mean temperature  $T_m$  and solid temperature  $T_{\text{solid}}$  might sometimes evolve separately with  $x$ . So, here there are four equations related to

energy flux and energy conservation:

$$\begin{aligned} \dot{H}_{2,k} &= \frac{1}{2} \operatorname{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{T_m \beta (f_\kappa - \tilde{f}_\nu)}{(1 + \epsilon_s)(1 + \sigma)(1 - \tilde{f}_\nu)} \right) \right] \\ &\quad + \frac{\rho_m c_p |U_1|^2}{2\omega A_{\text{gas}}(1 - \sigma) |1 - f_\nu|^2} \frac{dT_m}{dx} \operatorname{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1 + \epsilon_s f_\nu / f_\kappa)}{(1 + \epsilon_s)(1 + \sigma)} \right] \\ &\quad - A_{\text{gas}} k \frac{dT_m}{dx} + \dot{H}_{\text{solid}}, \end{aligned} \quad (10.152)$$

$$\dot{H}_{\text{solid}} = -A_{\text{solid}} k_s \frac{dT_{\text{solid}}}{dx}, \quad (10.153)$$

$$\dot{q}(x) = \frac{d\dot{H}_{2,k}}{dx}, \quad (10.154)$$

$$\dot{q}(x) = \frac{d\dot{H}_{\text{solid}}}{dx} + \frac{A_{\text{gas}} k}{4r_h^2} N_{Nu} (T_{\text{solid}} - T_m), \quad (10.155)$$

where  $\dot{q}$  is the heat per unit length externally delivered to the heat exchanger (if negative, it's the heat removed).

The first of these four equations describes the total energy transport in the  $x$  direction, and the second describes the energy transport (via heat conduction) in the  $x$  direction in the solid.

The third equation expresses conservation of energy for the dashed-line control volume in Fig. 10.8, showing that the heat input  $d\dot{Q} = \dot{q}(x) dx$  equals the difference  $(d\dot{H}_{2,k}/dx) dx$  between energy transport in and out along  $x$ .

The fourth equation similarly expresses conservation of energy for the solid separately within the control volume.

Equations (10.152)–(10.155) are used to obtain the evolution of the temperatures  $T_m(x)$  and  $T_{\text{solid}}(x)$ , and the energy flows  $\dot{H}_2(x)$ ,  $\dot{H}_{\text{solid}}(x)$ , and  $\dot{q}(x)$ , as functions of  $x$ , simultaneously with the use of Eqs. (10.150) and (10.151) to obtain  $p_1(x)$  and  $U_1(x)$ . Some details are different in VXT\* segments and VXQ\* segments:

In the VXT\* segments,  $T_{\text{solid}}$  is independent of  $x$  and is a user's input. Then  $\dot{H}_{\text{solid}} = 0$ , Eq. (10.152) is solved for  $dT_m/dx$  to yield the evolution of  $T_m(x)$ , Eqs. (10.154) and (10.155) together yield  $d\dot{H}_{2,k}/dx$ , and Eq. (10.154) yields  $\dot{q}(x) = d\dot{H}_{2,k}/dx$ . The heat transfer per unit length  $\dot{q}(x)$  is integrated with respect to  $x$  to obtain the total heat transfer rate  $\dot{Q}$ . (In VXT2, the two “passes” can have different values of  $T_{\text{solid}}$ .) Thus, Eqs. (10.150)–(10.155) are regarded as first-order differential equations for  $dp_1/dx$ ,  $dU_1/dx$ ,  $dT_m/dx$ , and  $d\dot{H}_{2,k}/dx$  (and

$\dot{q}(x)$  is obtained trivially from  $d\dot{H}_{2,k}/dx$  :

$$\frac{dT_m}{dx} = \frac{\dot{H}_{2,k} - \frac{1}{2} \operatorname{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{T_m \beta (f_\kappa - \tilde{f}_\nu)}{(1+\epsilon_s)(1+\sigma)(1-\tilde{f}_\nu)} \right) \right]}{\frac{\rho_m c_p |U_1|^2}{2\omega A_{\text{gas}} (1-\sigma) |1-\tilde{f}_\nu|^2} \operatorname{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1+\epsilon_s f_\nu / f_\kappa)}{(1+\epsilon_s)(1+\sigma)} \right] - A_{\text{gas}} k}, \quad (10.156)$$

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}} U_1, \quad (10.157)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \left( 1 + \frac{(\gamma-1)f_\kappa}{1+\epsilon_s} \right) p_1 + \frac{\beta(f_\kappa - f_\nu)}{(1-f_\nu)(1-\sigma)(1+\epsilon_s)} \frac{dT_m}{dx} U_1, \quad (10.158)$$

with  $\frac{dT_m}{dx}$  given by Eq. (10.156),

$$\frac{d\dot{H}_{2,k}}{dx} = \frac{A_{\text{gas}} k}{4r_h^2} 3.7 m_k (T_{\text{solid}} - T_m). \quad (10.159)$$

The factor “3.7” in Eq. (10.159) is the laminar, isothermal-wall Nusselt number for a circular channel (Ref. [63], Table 6-1), so except for the presence of  $m_k$  this is just the simplest laminar, circular heat-transfer equation. The multiplier  $m_k$  accounts for the enhancement of the Nusselt number due to turbulence, as is described below.

Starting conditions are straightforward for VXT\* segments: The starting  $p_1$ ,  $U_1$ ,  $T_m$ , and  $\dot{H}_{2,k}$  are obtained directly from the end of the previous segment, and  $T_{\text{solid}}$  is given by the user (or as a guess, of course).

In the VXQ\* segments, the heat exchange per unit length  $\dot{q}$  is independent of  $x$  and is set by the user, as the segment’s heat parameter divided by the appropriate length parameter. In VXQ2, the two “passes” can, of course, have different values of heat and length, and hence different values of  $\dot{q}$ . To make progress, it is necessary to make a couple of assumptions to simplify the starting equations. Solving Eqs. (10.152)–(10.155) for  $dT_m/dx$ ,  $dT_{\text{solid}}/dx$ ,  $d\dot{H}_{2,k}/dx$ , and  $d\dot{H}_{\text{solid}}/dx$  and integrating them would require initial values for  $T_{\text{solid}}$  and  $\dot{H}_{\text{solid}}$ , in addition to the initial values of  $T_m$  and  $\dot{H}_{2,k}$  that are passed into the VXQ\* segment from the end of the previous segment. To avoid having to specify or guess the initial values of  $T_{\text{solid}}$  and  $\dot{H}_{\text{solid}}$ , we assume that most of  $\dot{q}$  goes into the thermoacoustic gas in cases of interest to us, so the first term on the right side of Eq. (10.155) can be neglected there. This assumption probably leads only to a slight overestimation of the temperature difference  $|T_{\text{solid}} - T_m|$  in cases of interest to us. Second, we assume that  $dT_{\text{solid}}/dx = dT_m/dx$ . Examination of Eq. (10.155) (with the aforementioned first assumption, and with  $\dot{q}$  independent of  $x$  here for a VXQ\* segment) shows that this assumption is equivalent to  $d(kN_{Nu})/dx = 0$ , which seems reasonable enough (until someone figures out a better way to

do all this simply enough for DELTAEC). Then the starting equations can be rearranged as

$$\frac{dT_m}{dx} = \frac{\dot{H}_{2,k} - \frac{1}{2} \operatorname{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{T_m \beta (f_\kappa - \tilde{f}_\nu)}{(1+\epsilon_s)(1+\sigma)(1-\tilde{f}_\nu)} \right) \right]}{\frac{\rho_m c_p |U_1|^2}{2\omega A_{\text{gas}}(1-\sigma)|1-\tilde{f}_\nu|^2} \operatorname{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1+\epsilon_s f_\nu / f_\kappa)}{(1+\epsilon_s)(1+\sigma)} \right] - A_{\text{gas}} k - A_{\text{solid}} k_s}, \quad (10.160)$$

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}} U_1, \quad (10.161)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \left( 1 + \frac{(\gamma-1)f_\kappa}{1+\epsilon_s} \right) p_1 + \frac{\beta(f_\kappa - f_\nu)}{(1-f_\nu)(1-\sigma)(1+\epsilon_s)} \frac{dT_m}{dx} U_1, \quad (10.162)$$

with  $\frac{dT_m}{dx}$  given by Eq. (10.160),

$$\frac{d\dot{H}_{2,k}}{dx} = \dot{q}, \quad (10.163)$$

$$T_{\text{solid}} = T_m + \frac{4r_h^2}{4.4 m_k A_{\text{gas}} k} \dot{q}, \quad (10.164)$$

The factor “4.4” in Eq. (10.164) is the laminar, constant-heat-flux-density Nusselt number for a circular channel (Ref. [63], Table 6-1), so except for the presence of  $m_k$  this is just the simplest laminar, circular heat-transfer equation. The multiplier  $m_k$  accounts for the enhancement of the Nusselt number due to turbulence, as is described below.

Because of the assumptions in the previous paragraph, the starting conditions for the numerical integration are unambiguous: the four integrated variables  $T_m$ ,  $p_1$ ,  $U_1$ , and  $\dot{H}_{2,k}$  all start with the values left over from the previous segment of the DELTAEC model.

Tubesheets are treated exactly like stacks, with  $\dot{H}_{2,k}$  constant and

$$\frac{dT_m}{dx} = \frac{\dot{H}_{2,k} - \frac{1}{2} \operatorname{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{T_m \beta (f_\kappa - \tilde{f}_\nu)}{(1+\epsilon_s)(1+\sigma)(1-\tilde{f}_\nu)} \right) \right]}{\frac{\rho_m c_p |U_1|^2}{2\omega A_{\text{gas}}(1-\sigma)|1-\tilde{f}_\nu|^2} \operatorname{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1+\epsilon_s f_\nu / f_\kappa)}{(1+\epsilon_s)(1+\sigma)} \right] - A_{\text{gas}} k - A_{\text{solid}} k_s}, \quad (10.165)$$

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}} U_1, \quad (10.166)$$

$$\frac{dU_1}{dx} = -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \left( 1 + \frac{(\gamma-1)f_\kappa}{1+\epsilon_s} \right) p_1 + \frac{\beta(f_\kappa - f_\nu)}{(1-f_\nu)(1-\sigma)(1+\epsilon_s)} \frac{dT_m}{dx} U_1, \quad (10.167)$$

with  $\frac{dT_m}{dx}$  given by Eq. (10.165),

but with  $A_{\text{solid}} = A_{\text{total}} - A_{\text{gas}}$ .

The important viscous and thermal functions  $f_\nu$  and  $f_\kappa$  needed in Eqs. (10.150)–(10.152) and their variants are calculated as described here. For large displacement amplitudes, the effective length of the tube is its geometrical length (the *total* geometrical length, i.e., the sum of tubesheet and heat-transfer-region lengths). For small displacement amplitudes, the effective length is infinite. In between, this interpolation is used:

$$\Delta x_{\text{eff}} = \frac{4\Delta x_{\text{geom}}}{10|\xi_1|/\Delta x_{\text{geom}} - 1} \text{ when } \frac{\Delta x_{\text{geom}}}{10} < |\xi_1| < \frac{\Delta x_{\text{geom}}}{2}. \quad (10.168)$$

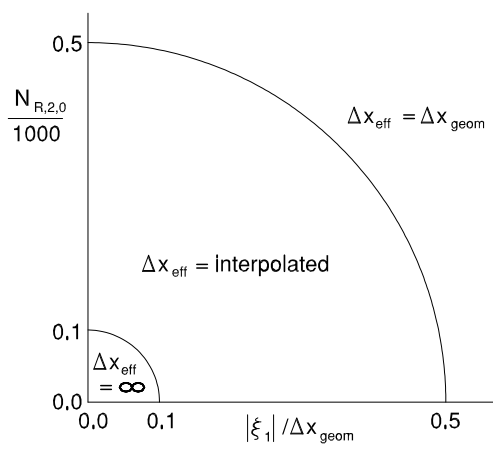


Figure 10.9: Effective length as a function of displacement amplitude  $|\xi_1|$ . For  $\dot{N}$  disabled, only the horizontal axis is relevant. For  $\dot{N}$  enabled, this plot displays Eqs. (10.185)–(10.187).

(The choices of this functional form, and of 1/10 and 1/2 as transitions, are arbitrary and without justification, but these details have little effect on the final results.) With this effective length, shown along the  $x$  axis in Fig. 10.9, the top half of Fig. 7-1 in Kays and London is used to estimate the effect of turbulence on  $f_\nu$  and  $f_\kappa$  in the tube. (That figure shows friction factor as a function of Reynolds number, for a few values of  $\Delta x/D$ .) DELTAEC’s internal approximations to the Kays and London data are shown here in Fig. 10.10. Power laws

$$F = g10^{-BN_R} \quad (10.169)$$

are fit to Kays and London’s high- $N_R$  straight lines, and similar power laws with offsets are fit to their low- $N_R$  straight lines, with interpolations for  $2450 < N_R < 10,000$  using a cubic polynomial matched to the inner ends of the two straight lines in both slope and value. (Here  $F$  is the Moody friction factor, four times the Fanning friction factor that Kays and London call  $f$ .) Then, using the peak Reynolds number, a multiplier

$$m_\nu = \max \left[ 1, \frac{\delta_\nu^2 N_R}{24\pi r_h^2 \text{Re} [i/(1 - f_\nu)]} \right] \quad (10.170)$$

gives the estimate of how turbulence (including entrance effects) affects what would otherwise be a laminar boundary layer. This expression is easily derived from the discussion about turbulence in Section 10.1.1.

Bessel-function (or, for large  $R$ , boundary-layer) expressions are used for “starting values” of  $f_\nu$  and  $f_\kappa$ . Then  $m_\nu$  as described above is used to generate modified  $f_\nu$  and  $f_\kappa$  for the momentum and continuity equations above, exactly as described in Section 10.1.1: DELTAEC increases the resistive component of the pressure gradient, and hence the viscous power dissipation, by  $m_\nu$ . It decreases the inertial pressure gradient by

$$m'_\nu = \left( \frac{1 - \delta_\nu/R}{1 - \delta_\nu/m_\nu R} \right)^2 \quad (10.171)$$

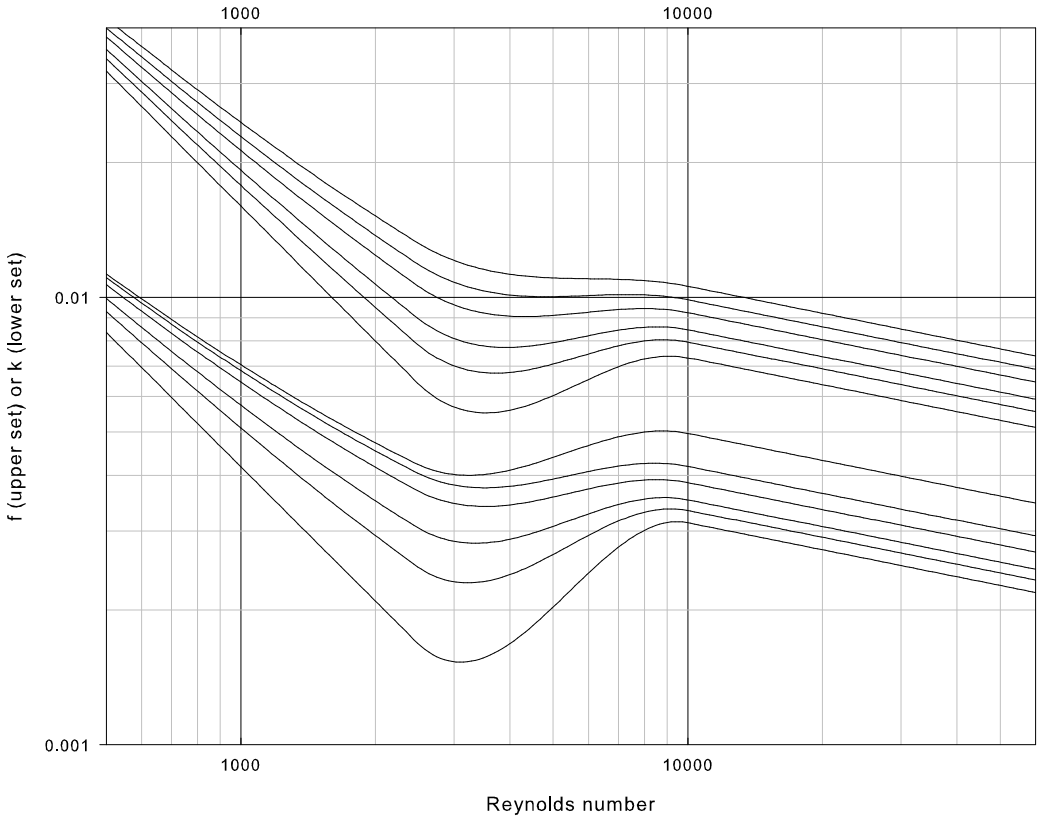


Figure 10.10: DELTAEC's fits for Kays and London's Fig. 7-1. The upper set of curves is the Fanning friction factor [Fanning = Moody /4 =  $F/4$  in Eqs. (10.169) and (10.170)], and the lower set is for the heat-transfer coefficient. Within each set, the different curves are for different  $L/D$  : infinity, 100, 50, 25, 15, and 1 from bottom to top.

to correct approximately for the steeper velocity gradient at the wall, which increases the effective area open to gas contributing to inertial effects. It applies the same turbulence correction to  $f_\kappa$  as to  $f_\nu$ , with no justification other than the idea that turbulent eddies carry heat and momentum similarly. For both  $f$ 's,

$$\text{Im} \left[ \frac{1}{1 - f_j} \right] = m_\nu \text{Im} \left[ \frac{1}{1 - f_{j, \text{laminar}}} \right], \quad (10.172)$$

$$\text{Re} \left[ \frac{1}{1 - f_j} \right] = m'_\nu \text{Re} \left[ \frac{1}{1 - f_{j, \text{laminar}}} \right], \quad (10.173)$$

$$f_j = 1 - \left( \text{Re} \left[ \frac{1}{1 - f_j} \right] + i \text{Im} \left[ \frac{1}{1 - f_j} \right] \right)^{-1} \quad (10.174)$$

[Note that, perhaps somewhat inconsistently,  $m_\nu$  enhances heat transfer in  $f_\kappa$  in Eqs. (10.151) and (10.152) while  $m_k$  is used in the time-averaged heat-transfer Eqs. (10.159) and (10.164).]

The value of  $m_k$  needed in Eq. (10.155) is obtained as follows. Power laws

$$\Sigma = A10^{-BN_R} \quad (10.175)$$

are fit to the low- $N_R$  and high- $N_R$  straight lines in the lower half of Kays and London Fig. 7-1, with interpolations for  $2450 < N_R < 10,000$  using a cubic polynomial matched to the inner ends of the two straight lines in both slope and value. (That figure shows heat-transfer factor as a function of Reynolds number, for a few values of  $\Delta x/D$ . Here  $\Sigma$  is the heat-transfer factor that Ref. [63] calls  $N_{St}\sigma^{2/3}$ .) Figure 10.10 shows these fits. Since time-averaged heat transfer is desired, these equations are evaluated using the time-averaged Reynolds number  $(2/\pi) |N_{R,1}|$ . Finally, a multiplier

$$m_k = \frac{\Sigma}{\Sigma_{\text{laminar}}(\Delta x_{\text{eff}} = \infty)} \quad (10.176)$$

gives the turbulence correction factor in the otherwise laminar Eqs. (10.159) and (10.164).

### Modifications to calculations for $\dot{N} \neq 0$ :

In the presence of nonzero steady flow  $\dot{N}$ , the starting equations are modified and augmented as follows:

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1 - f_\nu)A_{\text{gas}}}U_1 + \frac{i\omega m\dot{N}}{\rho_m a^2 A_{\text{gas}}} \frac{1 + (\gamma - 1)f_\kappa/(1 + \epsilon_s)}{1 - f_\nu} p_1, \quad (10.177)$$

$$\begin{aligned} \frac{dU_1}{dx} = & -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \left( 1 + \frac{(\gamma - 1)f_\kappa}{1 + \epsilon_s} \right) p_1 + \frac{\beta(f_\kappa - f_\nu)}{(1 - f_\nu)(1 - \sigma)(1 + \epsilon_s)} \frac{dT_m}{dx} U_1 \\ & + \frac{i\omega m\dot{N}}{\rho_m a^2 A_{\text{gas}}(1 - f_\nu)} U_1, \end{aligned} \quad (10.178)$$



$$\begin{aligned}\dot{H}_{\text{tot}} &= \frac{1}{2} \text{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{T_m \beta (f_\kappa - \tilde{f}_\nu)}{(1 + \epsilon_s)(1 + \sigma)(1 - \tilde{f}_\nu)} \right) \right] \\ &+ \frac{\rho_m c_p |U_1|^2}{2\omega A_{\text{gas}}(1 - \sigma) |1 - f_\nu|^2} \frac{dT_m}{dx} \text{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)(1 + \epsilon_s f_\nu / f_\kappa)}{(1 + \epsilon_s)(1 + \sigma)} \right] \\ &- A_{\text{gas}} k \frac{dT_m}{dx} + \dot{H}_{\text{solid}} + \dot{N} m w_m,\end{aligned}\quad (10.179)$$

$$\dot{H}_{\text{solid}} = -A_{\text{solid}} k_s \frac{dT_{\text{solid}}}{dx}, \quad (10.180)$$

$$\dot{q}(x) = \frac{d\dot{H}_{\text{tot}}}{dx} \cong \frac{\dot{N} m c_p}{F \dot{Q} \dot{N}} \frac{dT_m}{dx} \text{ in a pass, but } = 0 \text{ in a tubesheet,} \quad (10.181)$$

$$\dot{q}(x) = \frac{A_{\text{gas}} k}{4r_h^2} N_{Nu} (T_{\text{solid}} - T_m), \quad (10.182)$$

$$\frac{dp_{2,0,\text{HL}}}{dx} = -\text{sign}(\dot{N}_{2,0}) \frac{1}{2} \rho_m \left( \frac{m \dot{N}_{2,0}}{\rho_m A_{\text{gas}}} \right)^2 \frac{F}{4r_h} \sqrt{\frac{1}{2} + \left( \frac{\rho_m |U_1|}{m \dot{N}_{2,0}} \right)^2} \quad (10.183)$$

where  $\dot{N}_{2,0}$  and  $\dot{N}$  are related by

$$\dot{N}_{2,0} = \dot{N} + \frac{\beta}{m c_p} \left[ \dot{H}_{\text{tot}} - \dot{N} m w_m \right] - \left( \frac{1}{m a^2} + \frac{\beta}{m c_p} \right) \dot{E}_2. \quad (10.184)$$

**Important:** Equation (10.184) omits the  $(A_{\text{gas}} k + A_{\text{solid}} k_s) dT_m/dx$  term that is present in Eq. (8.16) because we expect it to be negligible most of the time and because DELTAEC needs  $\dot{N}_{2,0}$  before it knows  $dT_m/dx$  in each pass of its numerical integration. In Eqs. (10.177)–(10.183), DELTAEC evaluates  $m_\nu$  and  $F$  on the basis of  $\sqrt{|N_{R,1}|^2 + N_{R,2,0}^2}$ , and it evaluates  $m_\kappa$  on the basis of  $\sqrt{4|N_{R,1}|^2/\pi^2 + N_{R,2,0}^2}$ . The square-root factor at the end of the equation for  $dp_{2,0,\text{HL}}/dx$  is a rough empirical match to steady-flow DUCT results; this factor helps counteract the fact that high  $|U_1|$  biases the friction factor  $F$  toward values that are unrealistically low for use in this equation. The friction and heat-transfer factors are based on an effective tube length  $\Delta x_{\text{eff}}$  plotted in Fig. 10.9 and given by

$$\Delta x_{\text{eff}} = \infty \quad \text{when} \quad \left( \frac{|\xi_1|}{\Delta x_{\text{geom}}} \right)^2 + \left( \frac{N_{R,2,0}}{1000} \right)^2 \leq \frac{1}{100}, \quad (10.185)$$

$$= \frac{4\Delta x_{\text{geom}}}{10\sqrt{\left( \frac{|\xi_1|}{\Delta x_{\text{geom}}} \right)^2 + \left( \frac{N_{R,2,0}}{1000} \right)^2} - 1} \quad \text{when} \quad \frac{1}{100} \leq \left( \frac{|\xi_1|}{\Delta x_{\text{geom}}} \right)^2 + \left( \frac{N_{R,2,0}}{1000} \right)^2 \leq \frac{1}{4}, \quad (10.186)$$

$$= \Delta x_{\text{geom}} \quad \text{when} \quad \left( \frac{|\xi_1|}{\Delta x_{\text{geom}}} \right)^2 + \left( \frac{N_{R,2,0}}{1000} \right)^2 \geq \frac{1}{4}. \quad (10.187)$$

# 10.8. Adiabatic–isothermal interface loss

## 10.8.1. JOIN

### General description:

The JOIN segment accounts for “small” discontinuities in thermoacoustic variables at the interface between a heat exchanger or other isothermal segment and an unmixed, thermally stratified, adiabatic region such as a pulse tube. The discontinuity  $\delta T_m$  in mean temperatures is first order in the acoustic amplitude, the discontinuity  $\delta U_1$  in volume flow rate is second order, and there is no discontinuity in pressure. See Refs. [64, 65] for  $\delta T_m$ , Ref. [66] for  $\delta U_1$ , and Ref. [13] and references therein for both  $\delta T_m$  and  $\delta U_1$ .

When it encounters a JOIN segment, DELTAEC looks both upstream and downstream (ignoring merely logistical segments such as RPN) to figure out whether the “heat exchanger” is upstream and the “pulse tube” is downstream, or vice versa. Both  $A$  and  $dT_m/dx$  are obtained from the “pulse tube” or other open duct, with  $dT_m/dx = 0$  for ordinary DUCTs and CONEs, and  $dT_m/dx$  nonzero for STKDUCTs or STKCONEs. (If the relevant duct is a STKDUCT or STKCONE *downstream* of the JOIN segment, DELTAEC actually integrates ahead momentarily to evaluate  $dT_m/dx$ .)

The JOIN feature does not yet do anything different for nonzero  $\dot{N}$ .

### Input variables; Potential targets; Master–slave links:

None.

### Format and examples:

```

!----- 8 -----
SX      p.t. hot h.x.
1.1675E-04 a Area      m^2      1.3507E+05 A |p|      Pa
0.6470  b VolPor      -17.664 B Ph(p)      deg
3.0000E-03 c Length    m      2.9183E-04 C |U|      m^3/s
1.2000E-05 d rh        m      138.51  D Ph(U)      deg
2.8164  e HeatIn      W      G      -13.711 E Hdot      W
Possible targets      -18.030 F Edot      W W
copper Solid type      300.00  H SolidT      K
!----- 9 -----
JOIN    first join example
1.3507E+05 A |p|      Pa
-17.664 B Ph(p)      deg
2.9863E-04 C |U|      m^3/s
138.51  D Ph(U)      deg
-13.711 E Hdot      W
-18.450 F Edot      W
300.00  G TBeg      K
324.67  H TEnd      K
!----- 10 -----
STKDUCT the pulse tube
1.1675E-04 a Area      m^2      S=-2 1.3511E+05 A |p|      Pa
3.8307E-02 b Perim    m      Fnc(10a) -17.530 B Ph(p)      deg
7.0000E-02 c Length    m      2.7164E-04 C |U|      m^3/s
5.8678E-05 d WallA    m^2      166.83  D Ph(U)      deg
-13.711 E Hdot      W
-18.298 F Edot      W
324.67  G TBeg      K
57.932  H TEnd      K
stainless Solid type      0.1518 I StkEdt      W
!----- 11 -----
RPN    ratio of cold end displacement to p.t. length
0.1000 a G or T      (t)      9.2000E-02 A ChngeMe
10C 2 / pi / 0b / 10a / 10c /
!----- 12 -----
JOIN    second example; note that RPNs can go between

```

			1.3511E+05	A	p	Pa
			-17.530	B	Ph(p)	deg
			2.7854E-04	C	U	m <sup>3</sup> /s
			166.83	D	Ph(U)	deg
			-13.711	E	Hdot	W
			-18.763	F	Edot	W
			57.932	G	TBeg	K
			89.997	H	TEnd	K
----- 13 -----						
RPN	ratio of warm end displacement to p.t. length					
	6.0000	a G or T	(t)	9.8838E-02	A	ChngeMe
	8C 2 / pi / Ob / 10a / 10c /					
----- 14 -----						
SX	cold heat exchanger adj rh					
	1.1675E-04	a Area	m <sup>2</sup>	1.5473E+05	A	p  Pa
	0.6470	b VolPor		-16.785	B	Ph(p) deg
	5.7150E-03	c Length	m	2.7985E-04	C	U  m <sup>3</sup> /s
	1.2000E-05	d rh	m	169.37	D	Ph(U) deg
	7.1278	e HeatIn	W	-6.5828	E	Hdot W
	Possible targets			-21.525	F	Edot W
				7.1278	G	Heat W
copper		Solid type		90.000	H	SolidT K

## Calculations:

The discontinuity in temperature at JOIN is:

$$T_{m,\text{out}} - T_{m,\text{in}} = \left( -\frac{T_m \beta}{\rho_m c_p} |p_1| \sin \theta - \frac{|U_1|_{\text{in}}}{\omega A_{\text{gas}}} \frac{dT_m}{dx} \right) F, \quad (10.188)$$

where  $\theta$  is the angle by which  $p_1$  leads  $U_1$ ,  $dT_m/dx$  is evaluated on the thermally stratified, adiabatic side of the JOIN, and the factor  $F$  is given by

$$F = \sqrt{\frac{2 (A_{\text{gas}} k dT_m/dx)^2}{(A_{\text{gas}} k dT_m/dx)^2 + (\dot{H}_{2,k} - \dot{E})^2}}. \quad (10.189)$$

The factor  $F$  is a very crude attempt to account for the two-dimensional nature of the JOIN problem within DELTAEC's inherently one-dimensional character. The derivation of the  $T_m$  discontinuity joining condition in the references cited above neglects the thermal conductivity of the wall of the tube and the boundary-layer shuttle entropy flux. If these are zero while  $dT_m/dx \neq 0$ , then  $F = 1$  and Eq. (10.188) represents the joining condition derived in the references cited above. On the other hand, if the solid wall of the duct or the boundary layer is effectively a thermal short circuit compared to bulk gas conductivity, then  $F = 0$  and no temperature discontinuity appears at the JOIN segment. We arbitrarily chose a simple smooth function that passes through these two limiting cases. (This function was changed between DELTAEC version 6.3 and version 6.4, to cure a rare situation that had nonsensically yielded  $F \gg 1$ .)

The discontinuity in  $|U_1|$  at JOIN is

$$\begin{aligned} |U_1|_{\text{out}} &= |U_1|_{\text{in}} - \frac{8}{3\pi} \frac{(\gamma - 1)}{\rho_m a^2} |p_1| |U_1|_{\text{in}} \cos \theta \\ &= |U_1|_{\text{in}} - \frac{16}{3\pi} \frac{\gamma - 1}{\rho_m a^2} \dot{E}. \end{aligned} \quad (10.190)$$

There is no discontinuity in  $p_1$ , in the phase of  $U_1$ , or in  $\dot{H}_{2,k}$ .

## 10.9. Mixture separation

### 10.9.1. MIXBL, MIXSLAB, and MIXCIRC

#### General description:

The MIXBL, MIXSLAB, and MIXCIRC segments calculate  $dp_1/dx$ ,  $dU_1/dx$ , and  $dn_L/dx$  for sound propagation in gas mixtures, with or without steady flow, according to the theory described in Ref. [10], which builds on the theory described in Refs. [44, 67]. MIXBL performs the calculations in boundary-layer approximation, MIXSLAB in parallel-plate geometry of arbitrary gap  $2y_0$ , and MIXCIRC in circular pores of arbitrary radius  $R$ . The solid is assumed to be ideal, so it imposes a perfect isothermal boundary condition on the gas.

#### Input variables, MIXBL:

**Area** (m<sup>2</sup>) Cross-sectional area  $A_{\text{gas}}$  available to the gas, based on inside dimensions.

**Perim** (m) Perimeter  $\Pi$  of the inside cross section described above.

**Length** (m) The length  $\Delta x$  of the segment.

#### Input variables, MIXSLAB:

**Area** (m<sup>2</sup>) Total cross-sectional area  $A$  including gas and solid.

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A$ . (We often fix this at 1.000 and set **Area** equal to the gas area.)

**Length** (m) The length  $\Delta x$  of the segment.

**y0** (m) Half the plate spacing is  $y_0$ . Thus,  $2y_0$  is the gap available to the gas.

#### Input variables, MIXCIRC:

**Area** (m<sup>2</sup>) Total cross-sectional area  $A$  including gas and solid.

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A$ . (We often fix this at 1.000 and set **Area** equal to the gas area.)

**Length** (m) The length  $\Delta x$  of the segment.

**radius** (m) The radius  $R$  of the pores.

#### Potential targets:

None.

## Master–slave links:

The length of any of these segments can be slaved to the length of another segment to keep the sum of the two lengths constant.

(For an introduction to master–slave links, and a complete listing of linking options for all segments, see Section 2.7.2.)

## Format and examples:

```

!----- 1 -----
MIXBL      boundary-layer approx
4.0000E-04 a Area      m^2      3670.4      A |p|      Pa
8.0000E-02 b Perim    m          -0.2884     B Ph(p)    deg
0.1000     c Length   m          1.4715E-03 C |U|      m^3/s
                        -87.558     D Ph(U)    deg
                        -2.6300E-08 E Htot     W
                        0.1286      F Edot     W
                        0.3406      G nL Beg
                        0.3463      H nL End
!----- 2 -----
MIXCIRC    an array of circular pores
2.2000E-05 a Area      m^2      5637.2      A |p|      Pa
1.0000     b GasA/A    m          162.98      B Ph(p)    deg
0.1000     c Length   m          1.4540E-03 C |U|      m^3/s
1.0000E-03 d radius   m          -86.581     D Ph(U)    deg
                        -2.6300E-08 E Htot     W
                        -1.4313     F Edot     W
                        0.3463      G nL Beg
                        0.3452      H nL End
!----- 3 -----
MIXSLAB    an array of parallel plates
1.0000E-03 a Area      m^2      5460.3      A |p|      Pa
0.8000     b GasA/A    m          165.32      B Ph(p)    deg
0.1000     c Length   m          4.0475E-03 C |U|      m^3/s
5.0000E-04 d y0       m          53.191      D Ph(U)    deg
                        -2.6300E-08 E Htot     W
                        -4.1618     F Edot     W
                        0.3452      G nL Beg
                        0.2560      H nL End

```

Section 7.3 has an example DELTAEC file using MIXCIRC.

## Calculations:

The evolution of  $p_1$  with  $x$  in MIXBL, MIXSLAB, and MIXCIRC is given by the same integration as in DELTAEC's other ducts,

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}}U_1, \quad (10.191)$$

because Ref. [68] showed that the first-order thermoacoustic momentum equation is unchanged by mixture-separation effects. The evolution of  $U_1$  with  $x$  is given by Eq. (8) in Ref. [68], but they didn't write the result explicitly [10]. The explicit result, in DELTAEC's notation, is

$$\frac{dU_1}{dx} = -\frac{i\omega A_{\text{gas}}}{\rho_m a^2} \{1 + (\gamma - 1)[Bf_\nu + Cf_{\kappa D} + (1 - B - C)f_{D\kappa}]\} p_1, \quad (10.192)$$

where  $B$  and  $C$  are given in Ref. [44], and the other variables carry their usual DELTAEC meanings or are defined below.

The evolution of the light fraction  $n_L$  with  $x$  is given by solving Eq. (7) in Ref. [10] for  $dn_H/dx$ , and integrating numerically for fixed  $\dot{N}_L$ , which is equivalent to fixed  $\dot{N}_H$  and fixed  $\dot{N} = \dot{N}_H + \dot{N}_L$ . Solving Eq. (7) for  $dn_H/dx$  yields

$$\frac{dn_H}{dx} = \frac{\dot{N}_H - n_H \dot{N} - \frac{\delta_\kappa}{4r_h} \frac{\gamma-1}{\gamma} \frac{k_T}{R_{\text{univ}} T_m} |p_1| |U_1| (F_{\text{trav}} \cos \theta + F_{\text{stand}} \sin \theta)}{\frac{\delta_\kappa}{4r_h} \frac{\rho_m |U_1|^2}{m_{\text{avg}} \omega A_{\text{gas}}} F_{\text{grad}} - \frac{\rho_m}{m_{\text{avg}}} A_{\text{gas}} D_{12}}, \quad (10.193)$$

and then DELTAEC uses  $\bar{dn}_L/dx = -dn_H/dx$  to obtain the desired result.

In addition to the usual definitions of the viscous and thermal penetration depths  $\delta_\nu = \sqrt{2\mu/\omega\rho_m}$  and  $\delta_\kappa = \sqrt{2k/\omega\rho_m c_p}$ , two new penetration depths are defined for mixtures:

$$\delta_{\kappa D}^2 = \frac{1}{2} \delta_\kappa^2 \left[ 1 + (1 + \varepsilon) / L + \sqrt{[1 + (1 + \varepsilon) / L]^2 - 4/L} \right], \quad (10.194)$$

$$\delta_{D\kappa}^2 = \frac{1}{2} \delta_\kappa^2 \left[ 1 + (1 + \varepsilon) / L - \sqrt{[1 + (1 + \varepsilon) / L]^2 - 4/L} \right], \quad (10.195)$$

where

$$\varepsilon = \frac{\gamma - 1}{\gamma} \frac{k_T^2}{n_L (1 - n_L)}, \quad (10.196)$$

$k_T$  is the thermal diffusion ratio,  $L = k/\rho_m c_p D_{12}$ , and  $D_{12}$  is the binary mass-diffusion coefficient.

In MIXBL, DELTAEC uses the boundary-layer results of Ref. [44]. For  $f_\nu$ ,  $f_{\kappa D}$ , and  $f_{D\kappa}$ , the boundary layer result is

$$f_j = (1 - i) \frac{\delta_j}{2r_h}, \quad (10.197)$$

where  $j$  is  $\nu$ ,  $\kappa D$ , or  $D\kappa$ . For the three  $F$ 's, the boundary-layer results are

$$F_{\text{trav}} = \frac{\sigma \sqrt{\sigma L} - \sqrt{\sigma} - \sigma \sqrt{L} (\delta_{\kappa D} / \delta_\kappa + \delta_{D\kappa} / \delta_\kappa)}{(1 + \sqrt{L}) [(1 + \sigma)(1 + \sigma L) + \varepsilon \sigma]}, \quad (10.198)$$

$$F_{\text{stand}} = \frac{-\sigma \sqrt{\sigma L} + \sqrt{\sigma} - \sigma \sqrt{L} (\delta_{\kappa D} / \delta_\kappa + \delta_{D\kappa} / \delta_\kappa)}{(1 + \sqrt{L}) [(1 + \sigma)(1 + \sigma L) + \varepsilon \sigma]}, \quad (10.199)$$

$$F_{\text{grad}} = \frac{\sqrt{\sigma L} (1 - \sigma^2) (1 + \sqrt{L}) + \varepsilon \sqrt{\sigma} (\sqrt{L} - 1) + [(\sigma^2 - 1) L + \varepsilon \sigma \sqrt{L}] \left( \frac{\delta_{\kappa D}}{\delta_\kappa} + \frac{\delta_{D\kappa}}{\delta_\kappa} \right)}{(1 + \sqrt{L}) [(1 + \sigma)(1 + \sigma L) + \varepsilon \sigma] [(1 - \sigma)(1 - \sigma L) - \varepsilon \sigma] / \sigma}. \quad (10.200)$$

In MIXSLAB and MIXCIRC, DELTAEC uses the following expressions:

$$F_{\text{trav}} = -\frac{2r_h}{\delta_\kappa} \text{Re} \left[ \frac{G}{(1 - \tilde{f}_\nu)} \right], \quad (10.201)$$

$$F_{\text{stand}} = \frac{2r_h}{\delta_\kappa} \text{Im} \left[ \frac{G}{(1 - \tilde{f}_\nu)} \right], \quad (10.202)$$

$$F_{\text{grad}} = -\frac{2r_h}{\delta_\kappa} \frac{1}{|1 - \tilde{f}_\nu|^2} \frac{\sigma}{(1 - \sigma)(1 - \sigma L) - \varepsilon\sigma} \cdot \text{Im} \left[ \frac{(1 - \sigma)}{\sigma S} f_\nu \left( \frac{\delta_\kappa^2 - \delta_{\kappa D}^2}{\delta_\nu^2 + \delta_{\kappa D}^2} f_{\kappa D} - \frac{\delta_\kappa^2 - \delta_{D\kappa}^2}{\delta_\nu^2 + \delta_{D\kappa}^2} f_{D\kappa} + \frac{(1 + \sigma)LQ}{M} \tilde{f}_\nu + S \right) + \varepsilon G \right], \quad (10.203)$$

where  $r_h$  is the hydraulic radius and

$$S = \left( \frac{\delta_\kappa^2}{\delta_{D\kappa}^2} - 1 \right) f_{D\kappa} - \left( \frac{\delta_\kappa^2}{\delta_{\kappa D}^2} - 1 \right) f_{\kappa D}, \quad (10.204)$$

$$Q = \frac{\delta_{\kappa D}^2 - \delta_{D\kappa}^2}{\delta_\kappa^2}, \quad (10.205)$$

$$M = (1 + \sigma)(1 + \sigma L) + \varepsilon\sigma, \quad (10.206)$$

$$G = \frac{\sigma LQ}{MS} f_{\kappa D} f_{D\kappa} + \frac{\tilde{f}_\nu}{S} \left( \frac{f_{\kappa D}}{1 + \delta_\nu^2/\delta_{D\kappa}^2} - \frac{f_{D\kappa}}{1 + \delta_\nu^2/\delta_{\kappa D}^2} \right). \quad (10.207)$$

These expressions appear in Ref. [10] in the context of a circular tube.<sup>2</sup> In MIXCIRC the three  $f$ 's are given by

$$f_j = \frac{2J_1[(i - 1)R/\delta_j]}{(i - 1)(R/\delta_j)J_0[(i - 1)R/\delta_j]}, \quad (10.208)$$

where  $R$  is the circle radius,  $r_h = R/2$ , and  $j$  is  $\nu$ ,  $\kappa D$ , or  $D\kappa$ . In MIXSLAB, the three  $f$ 's are given by

$$f_j = \frac{\tanh[(1 + i)y_0/\delta_j]}{(1 + i)y_0/\delta_j}, \quad (10.209)$$

where the plate separation is  $2y_0 = 2r_h$  and  $j$  is  $\nu$ ,  $\kappa D$ , or  $D\kappa$ .

The head-loss pressure gradient  $dp_{2,0,HL}/dx$  is calculated on the basis of laminar flow. See Eqs. (10.69) and (10.70).

### 10.9.2. MIXTBL, MIXTSLAB, and MIXTCIRC

#### General description:

The MIXTBL, MIXTSLAB, and MIXTCIRC segments calculate  $dp_1/dx$ ,  $dU_1/dx$ ,  $dn_L/dx$ , and  $dT_m/dx$  for sound propagation in gas mixtures, with or without steady flow, according to

<sup>2</sup>There was a typographical error in Eq. (10.203) in the Version 6.0 Users Guide. The same typographical error appears in Eq. (A12), Ref. [10], for which an erratum was published in J. Acoust. Soc. Am. in 2008.

the theory described in Ref. [69], which builds on the theory described in Refs. [44, 67, 10]. MIXTBL performs the calculations in boundary-layer approximation, MIXTSLAB in parallel-plate geometry of arbitrary gap  $2y_0$ , and MIXTCIRC in circular pores of arbitrary radius  $R$ .

The solid is assumed to be laterally ideal, so it imposes a perfect isothermal boundary condition on the oscillating temperature of the gas. This assumption would correspond to  $\epsilon_s = 0$  in a **STK\*\*** segment. However, the solid's finite thermal conductivity in the  $x$  direction is included in the calculation of  $dT_m/dx$  in **MIXT\*\*** segments, in the same way as in **STK\*\*** segments.

### Input variables, MIXTBL:

**Area** ( $\text{m}^2$ ) Cross-sectional area  $A_{\text{gas}}$  available to the gas, based on inside dimensions.

**Perim** (m) Perimeter  $\Pi$  of the inside cross section described above.

**Length** (m) The length  $\Delta x$  of the segment.

**WallA** ( $\text{m}^2$ ) The cross-sectional area  $A_{\text{solid}}$  of the wall material.

**Solid** The material comprising the walls.

### Input variables, MIXTSLAB:

**Area** ( $\text{m}^2$ ) Total cross-sectional area  $A$  including gas and solid.

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A$ .

**Length** (m) The length  $\Delta x$  of the segment.

**y0** (m) Half the plate spacing is  $y_0$ . Thus,  $2y_0$  is the gap available to the gas between the plates.

**Solid** The material comprising the walls.

### Input variables, MIXTCIRC:

**Area** ( $\text{m}^2$ ) Total cross-sectional area  $A$  including gas and solid.

**GasA/A** The areal porosity  $\phi = A_{\text{gas}}/A$ .

**Length** (m) The length  $\Delta x$  of the segment.

**radius** (m) The radius  $R$  of the pores.

**Solid** The material comprising the walls.

### Potential targets:

None.



## Master–slave links:

The length of any of these segments can be slaved to the length of another segment to keep the sum of the two lengths constant.

(For an introduction to master–slave links, and a complete listing of linking options for all segments, see Section 2.7.2.)

## Format and examples:

```

!----- 1 -----
MIXTCIRC  an array of circular pores
2.2000E-05 a Area      m^2      5637.2    A |p|      Pa
0.9100    b GasA/A    162.98    B Ph(p)    deg
0.1000    c Length    m          1.4540E-03 C |U|      m^3/s
1.0000E-03 d radius   m          -86.581   D Ph(U)    deg
-2.6300E-08 E Htot     W          -1.4313   F Edot     W
0.3463    G nL Beg     0.3463    G nL Beg
0.3452    H nL End     0.3452    H nL End
310.00    I TBeg      310.00    I TBeg
451.23    J TEnd      451.23    J TEnd

stainless  Solid type

```

## Calculations:

In MIXTBL, MIXTSLAB, and MIXTCIRC, the integration variables evolve according to

$$\frac{dT_m}{dx} = \frac{\dot{H}_{\text{tot}} - \dot{N}mw_m - \frac{1}{2}\text{Re} \left[ p_1 \tilde{U}_1 \left( 1 - \frac{f_\kappa - \tilde{f}_\nu}{(1+\sigma)(1-f_\nu)} \right) \right]}{\frac{\rho_m c_p |U_1|^2}{2\omega A_{\text{gas}}(1-\sigma)|1-f_\nu|^2} \text{Im} \left[ \tilde{f}_\nu + \frac{(f_\kappa - \tilde{f}_\nu)}{(1+\sigma)} \right] - A_{\text{gas}}k - A_{\text{solid}}k_s}, \quad (10.210)$$

$$\frac{dp_1}{dx} = -\frac{i\omega\rho_m}{(1-f_\nu)A_{\text{gas}}}U_1, \quad (10.211)$$

$$\frac{dn_H}{dx} = \frac{\dot{N}_H - n_H\dot{N} - \frac{\delta_\kappa}{4r_h} \frac{(\gamma-1)k_T}{\gamma R_{\text{univ}} T_m} |p_1| |U_1| (F_{\text{trav}} \cos \theta + F_{\text{stand}} \sin \theta) + \frac{\rho_m}{m} \frac{k_T}{T_m} \frac{dT_m}{dx} \left( A_{\text{gas}}D_{12} - \frac{\delta_\kappa}{4r_h} \frac{|U_1|^2 F_{\nabla T}}{|1-f_\nu|^2 \omega A_{\text{gas}}} \right)}{\frac{\rho_m}{m} \left( \frac{\delta_\kappa}{4r_h} |U_1|^2 \frac{F_{\text{grad}}}{\omega A_{\text{gas}}} - A_{\text{gas}}D_{12} \right)}, \quad (10.212)$$

$$\frac{dU_1}{dx} = -i\omega A_{\text{gas}} \left( \frac{p_1}{p_m} - \frac{\langle T_1 \rangle}{T_m} + \frac{m_H - m_L}{m_{\text{avg}}} \langle n_{H,1} \rangle \right) + \frac{1}{T_m} \frac{dT_m}{dx} U_1 - \frac{m_H - m_L}{m_{\text{avg}}} \frac{dn_H}{dx} U_1, \quad (10.213)$$

where

$$\frac{\langle T_1 \rangle}{T_m} = \frac{\gamma - 1}{\gamma} [1 - B f_\nu - C f_{\kappa D} - (1 - B - C) f_{D\kappa}] \frac{p_1}{p_m} - \frac{(1 - f_{D\kappa})}{i\omega A_{\text{gas}} (1 - f_\nu)} \frac{1}{T_m} \frac{dT_m}{dx} U_1, \quad (10.214)$$

$$\begin{aligned} \langle n_{H,1} \rangle &= -\frac{\gamma - 1}{\gamma} \frac{m_L}{m_{\text{avg}}} \frac{k_T}{\varepsilon} \left[ B \left( 1 - \frac{1}{\sigma} \right) f_\nu + C \left( 1 - \frac{\delta_\kappa^2}{\delta_{\kappa D}^2} \right) f_{\kappa D} + (1 - B - C) \left( 1 - \frac{\delta_\kappa^2}{\delta_{D\kappa}^2} \right) f_{D\kappa} \right] \frac{p_1}{p_m} \\ &\quad - \frac{1}{i\omega A_{\text{gas}}} \frac{dn_H}{dx} U_1 + \frac{m_L}{m_{\text{avg}}} \frac{k_T}{\varepsilon} \frac{1}{i\omega A_{\text{gas}} (1 - f_\nu)} \left[ \left( 1 - \frac{\delta_\kappa^2}{\delta_{D\kappa}^2} \right) f_{D\kappa} - f_\nu \right] \frac{1}{T_m} \frac{dT_m}{dx} U_1, \end{aligned} \quad (10.215)$$

and

$$B = \frac{ie^{-i\theta}}{1 - f_\nu} \frac{\sigma}{(1 - \sigma)(1 - \sigma L) - \varepsilon\sigma} \left[ (\sigma L - 1) \frac{dT_m/dx}{\nabla T_{\text{crit}}} + \varepsilon \frac{dn_H/dx}{\nabla n_{\text{crit}}} \right], \quad (10.216)$$

$$\begin{aligned} C &= \frac{\sqrt{L}\delta_{\kappa D} - \delta_{D\kappa}}{(1 + \sqrt{L})(\delta_{\kappa D} - \delta_{D\kappa})} \left[ 1 - B \left( 1 - \frac{1 - \sigma}{\sqrt{\sigma}} \frac{\delta_\kappa}{\sqrt{L}\delta_{\kappa D} - \delta_{D\kappa}} \right) + \right. \\ &\quad \left. + \frac{ie^{-i\theta}}{1 - f_\nu} \left( 1 + \sqrt{\sigma} \frac{\delta_\kappa}{\sqrt{L}\delta_{\kappa D} - \delta_{D\kappa}} \right) \frac{dT_m/dx}{\nabla T_{\text{crit}}} \right], \end{aligned} \quad (10.217)$$

$$\nabla T_{\text{crit}} = \frac{\gamma - 1}{\gamma} \frac{|p_1|}{p_m} \frac{\omega A_{\text{gas}}}{|U_1|} T_m, \quad (10.218)$$

$$\nabla n_{\text{crit}} = \frac{\gamma - 1}{\gamma} \frac{|p_1|}{p_m} \frac{\omega A_{\text{gas}}}{|U_1|} k_T. \quad (10.219)$$

In addition to the usual definitions of the viscous and thermal penetration depths  $\delta_\nu = \sqrt{2\mu/\omega\rho_m}$  and  $\delta_\kappa = \sqrt{2k/\omega\rho_m c_p}$ , two new penetration depths are defined for mixtures:

$$\delta_{\kappa D}^2 = \frac{1}{2} \delta_\kappa^2 \left[ 1 + (1 + \varepsilon)/L + \sqrt{[1 + (1 + \varepsilon)/L]^2 - 4/L} \right], \quad (10.220)$$

$$\delta_{D\kappa}^2 = \frac{1}{2} \delta_\kappa^2 \left[ 1 + (1 + \varepsilon)/L - \sqrt{[1 + (1 + \varepsilon)/L]^2 - 4/L} \right], \quad (10.221)$$

where

$$\varepsilon = \frac{\gamma - 1}{\gamma} \frac{k_T^2}{n_L (1 - n_L)}, \quad (10.222)$$

$k_T$  is the thermal diffusion ratio,  $L = k/\rho_m c_p D_{12}$ , and  $D_{12}$  is the binary mass-diffusion coefficient.

For MIXTBL,

$$f_j = (1 - i) \frac{\delta_j}{2r_h}, \quad (10.223)$$

where  $j$  is  $\nu$ ,  $\kappa$ ,  $\kappa D$ , or  $D\kappa$  and  $r_h$  is the hydraulic radius, and the four  $F$ 's are given by

$$F_{\text{trav}} = \frac{\sigma\sqrt{\sigma L} - \sqrt{\sigma} - \sigma\sqrt{L}(\delta_{\kappa D}/\delta_{\kappa} + \delta_{D\kappa}/\delta_{\kappa})}{(1 + \sqrt{L})[(1 + \sigma)(1 + \sigma L) + \varepsilon\sigma]}, \quad (10.224)$$

$$F_{\text{stand}} = \frac{-\sigma\sqrt{\sigma L} + \sqrt{\sigma} - \sigma\sqrt{L}(\delta_{\kappa D}/\delta_{\kappa} + \delta_{D\kappa}/\delta_{\kappa})}{(1 + \sqrt{L})[(1 + \sigma)(1 + \sigma L) + \varepsilon\sigma]}, \quad (10.225)$$

$$F_{\text{grad}} = \frac{\sqrt{\sigma L}(1 - \sigma^2)(1 + \sqrt{L}) + \varepsilon\sqrt{\sigma}(\sqrt{L} - 1) + [(\sigma^2 - 1)L + \varepsilon\sigma\sqrt{L}](\delta_{\kappa D}/\delta_{\kappa} + \delta_{D\kappa}/\delta_{\kappa})}{(1 + \sqrt{L})[(1 + \sigma)(1 + \sigma L) + \varepsilon\sigma][(1 - \sigma)(1 - \sigma L) - \varepsilon\sigma]/\sigma}, \quad (10.226)$$

$$-F_{\nabla T} = \frac{\sqrt{\sigma L}(1 + \sqrt{L})(1 + \sigma^2\sqrt{L}) + \sigma^{\frac{3}{2}}L + \varepsilon\sqrt{\sigma} - \frac{1}{\sqrt{\sigma}} + [\sqrt{L}(1 - \sigma L - \varepsilon\sigma) - \sigma L(1 + \sigma)]\left(\frac{\delta_{\kappa D}}{\delta_{\kappa}} + \frac{\delta_{D\kappa}}{\delta_{\kappa}}\right)}{(1 + \sqrt{L})[(1 + \sigma)(1 + \sigma L) + \varepsilon\sigma][(1 - \sigma)(1 - \sigma L) - \varepsilon\sigma]/\sigma}. \quad (10.227)$$

In MIXTSLAB and MIXTCIRC, DELTAEC uses the following expressions:

$$F_{\text{trav}} = -\frac{2r_h}{\delta_{\kappa}} \text{Re} \left[ \frac{G}{(1 - \tilde{f}_{\nu})} \right], \quad (10.228)$$

$$F_{\text{stand}} = \frac{2r_h}{\delta_{\kappa}} \text{Im} \left[ \frac{G}{(1 - \tilde{f}_{\nu})} \right], \quad (10.229)$$

$$F_{\text{grad}} = -\frac{2r_h}{\delta_{\kappa}} \frac{1}{|1 - f_{\nu}|^2} \frac{\sigma}{(1 - \sigma)(1 - \sigma L) - \varepsilon\sigma} \cdot \text{Im} \left[ \frac{(1 - \sigma)}{\sigma S} f_{\nu} \left( \frac{\delta_{\kappa}^2 - \delta_{\kappa D}^2}{\delta_{\nu}^2 + \delta_{\kappa D}^2} f_{\kappa D} - \frac{\delta_{\kappa}^2 - \delta_{D\kappa}^2}{\delta_{\nu}^2 + \delta_{D\kappa}^2} f_{D\kappa} + \frac{(1 + \sigma)LQ}{M} \tilde{f}_{\nu} + S \right) + \varepsilon G \right], \quad (10.230)$$

$$F_{\nabla T} = \frac{2r_h}{\delta_{\kappa}} \frac{1}{|1 - f_{\nu}|^2} \frac{\sigma}{(1 - \sigma)(1 - \sigma L) - \varepsilon\sigma} \cdot \text{Im} \left[ \frac{f_{\nu}}{S} \left( \frac{\delta_{\kappa}^2 - \delta_{\kappa D}^2}{\delta_{\nu}^2 + \delta_{\kappa D}^2} f_{\kappa D} - \frac{\delta_{\kappa}^2 - \delta_{D\kappa}^2}{\delta_{\nu}^2 + \delta_{D\kappa}^2} f_{D\kappa} + \frac{(1 + \sigma)LQ}{M} \tilde{f}_{\nu} + S \right) + \frac{1 - \sigma L - \varepsilon\sigma}{\sigma} G \right], \quad (10.231)$$

where  $r_h$  is the hydraulic radius and

$$S = \left( \frac{\delta_\kappa^2}{\delta_{D\kappa}^2} - 1 \right) f_{D\kappa} - \left( \frac{\delta_\kappa^2}{\delta_{\kappa D}^2} - 1 \right) f_{\kappa D}, \quad (10.232)$$

$$Q = \frac{\delta_{\kappa D}^2 - \delta_{D\kappa}^2}{\delta_\kappa^2}, \quad (10.233)$$

$$M = (1 + \sigma)(1 + \sigma L) + \varepsilon\sigma, \quad (10.234)$$

$$G = \frac{\sigma L Q}{MS} f_{\kappa D} f_{D\kappa} + \frac{\tilde{f}_\nu}{S} \left( \frac{f_{\kappa D}}{1 + \delta_\nu^2 / \delta_{D\kappa}^2} - \frac{f_{D\kappa}}{1 + \delta_\nu^2 / \delta_{\kappa D}^2} \right). \quad (10.235)$$

These expressions appear in Refs. [10, 69] in the context of a circular tube.

In MIXTCIRC the three  $f$ 's are given by

$$f_j = \frac{2J_1[(i-1)R/\delta_j]}{(i-1)(R/\delta_j)J_0[(i-1)R/\delta_j]}, \quad (10.236)$$

where  $R$  is the circle radius,  $r_h = R/2$ , and  $j$  is  $\nu$ ,  $\kappa D$ , or  $D\kappa$ .

In MIXTSLAB, the three  $f$ 's are given by

$$f_j = \frac{\tanh[(1+i)y_0/\delta_j]}{(1+i)y_0/\delta_j}, \quad (10.237)$$

where the plate separation is  $2y_0 = 2r_h$  and  $j$  is  $\nu$ ,  $\kappa D$ , or  $D\kappa$ .

The head-loss pressure gradient  $dp_{2,0,HL}/dx$  is calculated on the basis of laminar flow in MIXT\*\* segments.

# 11. LOGISTICAL SEGMENTS

Details about DELTAEC’s physically realistic thermoacoustics segments were described in Chapter 10. Here, we detail the logistical segments, which perform only trivial computations or no computations but are used to coordinate other segments, impose boundary conditions, and display non-standard results.

The behavior of some of these segments depends on whether the DELTAEC model using the segments is pure acoustics or thermoacoustics or includes nonzero  $\dot{N}$  or includes separation effects in gas mixtures. The discussion of  $\dot{N}_L$  and  $n_L$  only applies when MIX\*\* segments and gas mixtures are in use. The discussion of  $\dot{N}$  and  $p_{2,0,HL}$  only applies when nonzero time-averaged flow is involved. Users who are interested only in acoustics can focus entirely on  $p_1$  and  $U_1$  here, and can safely ignore all discussion of  $\dot{H}$ ,  $T_m$ ,  $\dot{N}$ ,  $p_{2,0,HL}$ ,  $\dot{N}_L$ , and  $n_L$ .

An alphabetical listing of segments is in the Segment Index at the end of the Users Guide.

## 11.1. Starting and ending

### 11.1.1. TITLE

#### General description:

TITLE has no numerical inputs. It must come at the very beginning of a DELTAEC file, and its heading and (optional) notes should be descriptive, to remind the user what the file is about.

#### Format and examples:

```
engine2006-d   July 2006 engine with Karen's modifications.
```

### 11.1.2. BEGIN

#### General description:

After the TITLE segment, BEGIN is counted as the zeroth segment of the file. It is used to initialize “global” variables that are shared among subsequent segments (e.g., frequency and mean pressure) and the integration variables required by each pass of DELTAEC to get started (e.g., real and imaginary parts of  $p_1$  and  $U_1$ ).

BEGIN segments can be used anywhere in a DELTAEC file, to set any or all of DELTAEC’s integration variables or global variables to new values. This can be useful for packing two or more related DELTAEC models into a single file or for changing from one gas to another in the middle of a model. (Do not use such a BEGIN within a teebranch; instead, use RPN

“change” assignments to set or reset variables in a teebranch.) When more than one **BEGIN** is used in a model, each **BEGIN** starts a new leg. (Legs are described in Section 11.2.1.)

Starting with DELTAEC version 6.0, **BEGIN** causes subsequent **DUCTs**, **CONEs**, **SURFACEs**, and **IMPEDANCEs** to set  $\dot{H}_{\text{tot,out}} = \dot{H}_{\text{tot,in}}$ , so the user should imagine such segments to be thermally insulated. **ANCHOR**, described in Section 11.3.1, can be used to override this default, and the user should imagine segments downstream of **ANCHOR** to be attached to a heat sink at the local temperature  $T_m$ . (**ANCHOR** is useful for very low power systems and simple systems where total energy flow is irrelevant, and of course for systems with **DUCTs** and other components that actually are well-attached to a heat sink.)

### Input variables:

**Mean P** (Pa) The mean pressure of the gas,  $p_m$ .

**Freq** (Hz) The frequency  $f$  of the sinusoidal oscillations in subsequent segments. Most equations in the Users Guide use  $\omega = 2\pi f$ .

**TBeg** (K) The initial temperature  $T_m$  of the gas.

**|p|** (Pa) The initial value of  $|p_1|$ .

**Ph(p)** (deg) The initial phase of  $p_1$ .

**|U|** (m<sup>3</sup>/s) The initial value of  $|U_1|$ .

**Ph(U)** (deg) The initial phase of  $U_1$ .

**Gas** The gas type, e.g., **helium** or **air**.

**Default Htot / Other Htot** (W) With this choice in the Optional Parameters dialog, the user can let DELTAEC assign the default value to the total energy flow  $\dot{H}_{\text{tot}}$  or can assign a numerical value or let it be a guess.

**Enable Ndot / Disable Ndot** This check box in the Optional Parameters dialog controls some calculations and results in many subsequent segments.

**Ndot** (mol/s) If “Enable Ndot” is checked, the user can specify or guess the steady molar flow rate  $\dot{N}$ .

**nL** If the gas type is a gas mixture, this parameter initializes the light-component mole fraction  $n_L$ .

**Bulk NLdot / Other NLdot** (mol/s) If the gas type is a gas mixture, this check box in the Optional Parameters dialog controls whether DELTAEC assigns the default, bulk value to the light component’s steady molar flow rate  $\dot{N}_L$  or lets the user assign a different value.

# Potential targets; Master-slave links:

None.

## Format and examples:

BEGIN This example is the minimal format, as saved in an out file.

```

1.0e6 Pa      a Mean P
500. Hz      b Freq
300. K       c TBeg
3.0e4 Pa     d |p|
0.00 deg    e Ph(p)
0.00 m3/s   f |U|
0.00 deg    g Ph(U)
helium      Gas type

```

BEGIN This example is the most complicated possible format, as saved in an out file.

```

1.0e6 Pa      a Mean P
500. Hz      b Freq
300. K       c TBeg
3.0e4 Pa     d |p|
0.00 deg    e Ph(p)
0.00 m3/s   f |U|
0.00 deg    g Ph(U)
2000.       h Htot
1.00E-8     i Ndot
0.800      j nL
1.00E-9     k NLdot
HeAr       Gas type

```



TITLE Argon cover on top of sodium-potassium column, illustrating two BEGINS in one model  
!->2beg.out

!Created@10:20:31 29-Mar-07 with DeltaE Vers. 6.0g7 for the IBM/PC-Compatible

```

!----- 0 -----
BEGIN      Initialize in the argon
1.0000E+05 a Mean P      Pa      3.9676E-05 A |U|      G( Of)      P
100.00    b Freq        Hz      87.799    B Ph(U)    G( Og)      P
400.00    c TBeg        K
1000.0    d |p|         Pa
0.0000    e Ph(p)      deg
3.9676E-05 f |U|      m^3/s    G
87.799    g Ph(U)      deg      G
0.0000    h Htot        W
0.0000    j nL
HeAr      Gas type

```

```

!----- 1 -----
DUCT      the argon on top
1.0000E-04 a Area      m^2      S= -2    1015.7    A |p|      Pa
3.5449E-02 b Perim    m      Fn( 1a) -7.9065E-02 B Ph(p)    deg
0.1000    c Length    m      1.9924E-08 C |U|      m^3/s
89.527    D Ph(U)      deg
0.0000    E Htot      W
6.9504E-08 F Edot     W
ideal     Solid type

```

```

!----- 2 -----
BEGIN      transition from argon to Na-K
sameas    0a a Mean P      Pa
sameas    0b b Freq        Hz
sameas    0c c TBeg        K
sameas    1A d |p|         Pa
sameas    1B e Ph(p)      deg
sameas    1C f |U|      m^3/s
sameas    1D g Ph(U)      deg
sameas    1E h Htot      W
NaK-78    Gas type

```

```

!----- 3 -----
DUCT      The sodium-potassium column
sameas    1a a Area      m^2      1069.7    A |p|      Pa
sameas    1b b Perim    m      -0.1210  B Ph(p)    deg
1.0000    c Length    m      3.7365E-13 C |U|      m^3/s
-0.1156  D Ph(U)      deg
0.0000    E Htot      W
1.9984E-10 F Edot     W
ideal     Solid type

```

```

!----- 4 -----
SURFACE   Bottom end
sameas    1a a Area      m^2      1069.7    A |p|      Pa
-0.1210  B Ph(p)    deg
3.5753E-17 C |U|      m^3/s
83.367   D Ph(U)    deg

```

ideal	Solid type		0.0000	E Htot	W
			2.1686E-15	F Edot	W
HARDEND terminate in the liquid metal			5		
0.0000	a R(1/z)	= 5G?	1069.7	A  p	Pa
0.0000	b I(1/z)	= 5H?	-0.1210	B Ph(p)	deg
			3.5753E-17	C  U	m <sup>3</sup> /s
			83.367	D Ph(U)	deg
			0.0000	E Htot	W
			2.1686E-15	F Edot	W
			6.3787E-11	G R(1/z)	
			5.5883E-10	H I(1/z)	

## Calculations:

BEGIN establishes initial values of  $p_m$ ,  $f$ ,  $T_m$ ,  $p_1$ ,  $U_1$ ,  $\dot{H}_{2,k}$ ,  $\dot{N}$ , and (if applicable)  $n_L$  and  $\dot{N}_L$ . It establishes the gas type. Unless subsequently overridden by an ANCHOR segment, it establishes “insulate” mode, effectively putting thermal insulation around all subsequent segments except heat exchangers.

If Htot is omitted from the <.out> file, or if the “Default Htot” box is checked in the Optional Parameters dialog in the user interface, BEGIN sets an initial value of

$$\dot{H}_{\text{tot}} = \dot{E} + \dot{N}mw_m, \quad (11.1)$$

which implies that  $\dot{H}_{2,k} = \dot{E}$ . Otherwise, Htot is an ordinary input parameter, which can be set by the user or guessed.

If the “Disable Ndot” box is checked in the Optional Parameters dialog or the Ndot input parameter is missing from the <.out> file, DELTAEC assigns  $\dot{N} = 0$ , skips all subsequent  $p_{2,0,HL}$  calculations, and ignores any possible effect of nonzero  $\dot{N}_{2,0}$  on  $dp_1/dx$ . If the “Enable Ndot” box is checked,  $p_{2,0,HL}$  is initialized to zero and its dependence on  $x$  is calculated in all subsequent segments. This also allows input parameter  $\dot{N}$  to appear in BEGIN, and allows other input and output parameters that are relevant to steady flow to appear in subsequent segments. When this box is checked, nonnegligible  $dp_{2,0,HL}/dx$  and nonlinear effects on  $dp_1/dx$  can occur in subsequent segments even if  $\dot{N} \simeq 0$ , because of the difference between  $\dot{N}$  and  $\dot{N}_{2,0}$  described in Eq. (8.16).

If the gas type is a binary mixture, the initial mole fraction of the light component,  $n_L$ , appears as an input parameter, which can only change in MIX\*\* segments or be reassigned using =nL in an RPN segment downstream.

Other, less commonly used variables are invisibly initialized in BEGIN:

vol = 0, described in Section 11.4.1.

$F_1 = 0$ , described in Section 11.4.1.

Tzero = 300 Kelvin, described above Eq. (6.3) in the context of  $\dot{H}$  and below Eq. (8.9) in the context of  $\dot{X}$ .

nLzero = 0.5, described below Eq. (8.9) in the context of  $\dot{X}$ .

$\dot{N}_L = n_L \dot{N}$ , if the gas is a binary gas mixture and the “Bulk NLdot” box is checked.

$T_{\text{solid}} = T_m$ .



### 11.1.3. HARDEND and SOFTEND

#### General description:

Often, the final segment (except possibly RPN segments) will be either **HARDEND** or **SOFTEND**. These contain three or four potential targets.

**HARDEND** targets are appropriate when the user wants the complex volume flow rate somewhere to be zero. This is the usual case at the end of a closed thermoacoustic system. **HARDEND** is also often used at the end of a closed, sealed **TBRANCH**.

**SOFTEND** targets are appropriate when the user wants the complex pressure amplitude somewhere to be zero. This is often useful for symmetrical systems, where **SOFTEND** indicates that the remainder of the apparatus is a mirror image of what is in the `<.out>` file and forces a complex-pressure node. Also use the **SOFTEND** segment, but without using its targets, to signal the end of a **TBRANCHED** sequence of segments that will be reattached elsewhere at a **UNION**, or anywhere that a leg must end with circumstances that can best be defined using custom-made RPN targets.

Either **\*\*END** segment can also be targeted to enforce a non-trivial impedance at the end of a branch or a model. An example of **SOFTEND** to model the radiation impedance from an open-ended tube is shown on the next page.

The third default target in these segments,  $\dot{H}_{\text{tot}}$ , can be targeted to zero to represent thermal insulation at a **HARDEND** or the energy aspect of a mirror-image symmetry location in **SOFTEND**.

In both **\*\*ENDs**, the complex specific impedances are made dimensionless by dividing by  $\rho_m a$  evaluated at the local temperature.

Disabling the targets in **\*\*ENDs** is often useful in debugging a new model that doesn't readily converge. Set these targets nonzero to model a known, nonzero end impedance—or use **BRANCH** (followed by a zero-targeted **HARDEND**) to work with  $Z$  instead of  $z/\rho a$ .

#### Input variables:

None.

#### Potential targets, HARDEND:

**R(1/z)** If targeted, this number is compared with DELTAEC's calculation of the real part of the inverse of the normalized specific impedance,  $\text{Re}[1/z_n]$ , at the **HARDEND**.

**I(1/z)** If targeted, this number is compared with DELTAEC's calculation of the imaginary part of the inverse of the normalized specific impedance,  $\text{Im}[1/z_n]$ , at the **HARDEND**.

**Htot** (W) Total energy flow  $\dot{H}_{\text{tot}}$ . This potential target appears in insulated mode. If targeted, this number is compared with DELTAEC's calculation of  $\dot{H}_{\text{tot}}$  at the **HARDEND**.

**Ndot** (mol/s) Steady molar flow rate  $\dot{N}$ . This potential target appears when "Enable Ndot" is checked in an upstream **BEGIN**. If targeted, this number is compared with DELTAEC's calculation of  $\dot{N}$  at the **HARDEND**. Useful only in rare circumstances.

## Potential targets, SOFTEND:

**R(z)** If targeted, this number is compared with DELTAEC's calculation of the real part of the normalized specific impedance,  $\text{Re}[z_n]$ , at the SOFTEND.

**I(z)** If targeted, this number is compared with DELTAEC's calculation of the imaginary part of the normalized specific impedance,  $\text{Im}[z_n]$ , at the SOFTEND.

**Htot (W)** Total energy flow  $\dot{H}_{\text{tot}}$ . This potential target appears in insulated mode. If targeted, this number is compared with DELTAEC's calculation of  $\dot{H}_{\text{tot}}$  at the SOFTEND.

## Master-slave links:

None.

## Format and examples:

```
!----- 97 -----
HARDEND looks like this if steady-flow calculations are turned off
0.000 R(1/z)
0.000 I(1/z)
0.000 Htot
```



```
!----- 40 -----
DUCT final resonator tube, open to the room
8.0000E-4 a Area m^2 Mstr 1023.8 A |p| Pa
0.10027 b Perim m 40a -0.95235 B Ph(p) deg
3.1783E-2 c Length m G 2.2366E-2 C |U| m^3/s
5.0000E-4 d Srough -87.544 D Ph(U) deg
0.68065 E Htot W
0.68065 F Edot W
```

```
ideal Solid type
!----- 41 -----
RPN calc radius from area
0.0000 a G or T 0.1461 A kr
1.5958E-2 B m rad
```

```
40a pi / sqrt ; # w * a /
!----- 42 -----
RPN radiation impedance from open tube / (rho a / A)
! Levine and Schwinger, Phys Rev 73, 383-406 (1948)
! Z = rho a / A times z ;
! z = (kr)^2 / 4 + j 0.6133 kr
0.0000 a G or T 5.3360E-3 A Re(z)
8.9601E-2 B Im(z)
```

```
0.6133 41A * ; 41A sqrd 4 /
!----- 43 -----
SOFTEND radiate sound into 4 pi solid angle in the room
! Note: z calc here uses closest upstream area (40a) internally AND via RPN 41A
sameas 42A a Re(z) =43G 1023.8 A |p| Pa
sameas 42B b Im(z) =43H -0.95235 B Ph(p) deg
0.0000 c Htot W 2.2366E-2 C |U| m^3/s
-87.544 D Ph(U) deg
0.68065 E Htot W
0.68065 F Edot W
5.3360E-3 G Re(z)
8.9601E-2 H Im(z)
293.00 I T K
```

## Calculations:

The normalized impedance in HARDEND and SOFTEND is calculated according to

$$z_n = \frac{Ap_1}{\rho_m a U_1}, \quad (11.2)$$

with  $A$  from the most recent segment having an area (or  $A = 10^{-6} \text{ m}^2$  if no such segment exists), and with  $\rho_m$  and  $a$  evaluated at the local  $T_m$ .

## 11.2. Structured branches and unions

Figure 11.1 shows how **TBRANCH** and **UNION** can be used to model complicated branched and multiply-connected thermoacoustic equipment.

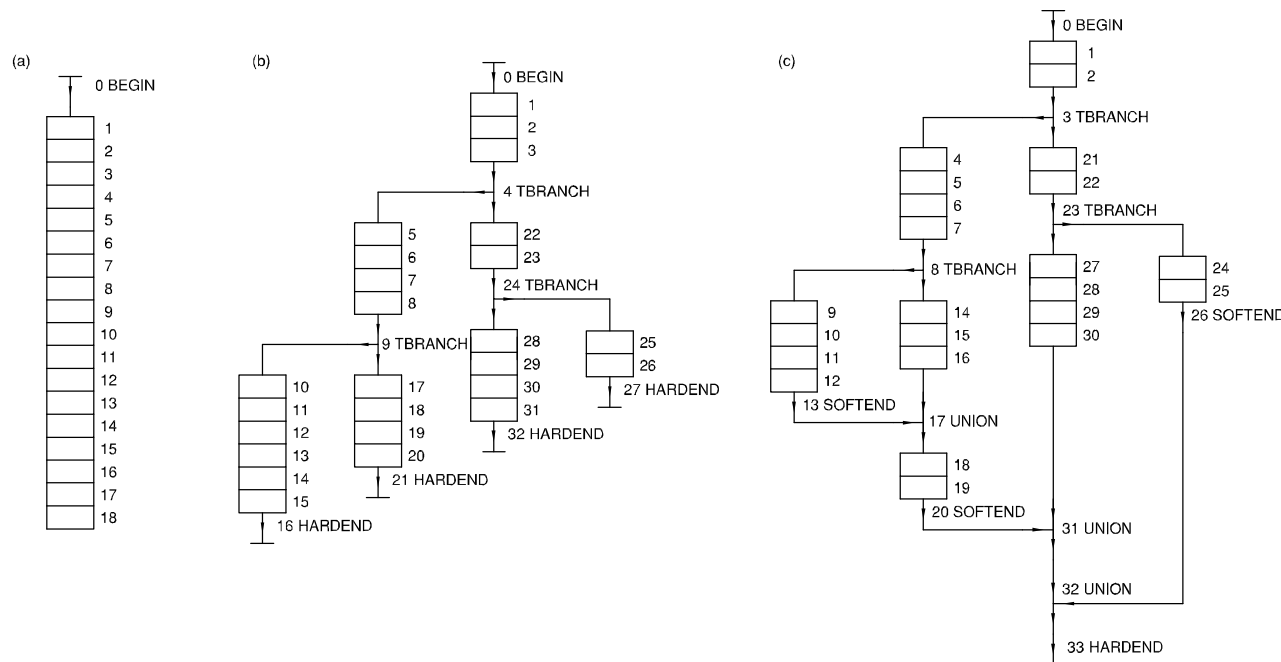


Figure 11.1: (a) Example of an 18-segment DELTAEC model of a linear array of thermoacoustic components. (b) Example of a model of a multiply-branched apparatus. (c) Example of a model of a multiply-connected apparatus. Line a of **UNION** 17 must point to **SOFTEND** 13, line a of **UNION** 31 must point to **SOFTEND** 20, and line a of **UNION** 32 must point to **SOFTEND** 26.

### 11.2.1. TBRANCH

#### General description:

Use **TBRANCH** for branched systems that are too complicated for **BRANCH** or **OPNBRANCH**.

When it encounters a **TBRANCH**, DELTAEC treats subsequent segments as the sequential members of a side branch (which we call a “teebranch” in the Users Guide), until it reaches a **HARDEND** or **SOFTEND**. Then it “returns to the trunk,” treating the rest of the segments as trunk members. The **HARDEND** termination of a teebranch is appropriate for a closed-end leg. The **SOFTEND** termination of a teebranch is appropriate for later reconnection to the trunk at a **UNION** in the trunk.

To manage the plotter display of results from models with one or more **TBRANCH**s, the `<.sp>` file’s initial column, “leg number,” is incremented each time a **TBRANCH** or **BEGIN** is encountered, and decremented when a **SOFTEND** or **HARDEND** is encountered. The user can display plots of one or more selected legs, suppressing results from uninteresting legs.

The complex impedance in the TBRANCH segment tells DELTAEC how to split up the volume flow rate between the teebranch and the trunk. Usually this complex impedance is used as two guesses to hit two downstream targets, often either complex  $1/z_n$  in a HARDEND at the end of a dead-ended teebranch or complex  $p_1$  at a subsequent UNION for a multiply-connected teebranch. Unless ANCHOR appears upstream, HtotBr tells DELTAEC how to split up the total power  $\dot{H}_{\text{tot}}$  between the branch and the trunk. This is often used as a guess to hit a downstream temperature target, such as at a UNION. The parameter NdotBr tells DELTAEC how to split any nonzero time-averaged molar flow  $\dot{N}$  between the branch and the trunk. Similarly, NLdotB specifies the portion of the time-averaged mole flux of the lighter component of a mixture,  $\dot{N}_L$ , that goes into the branch.

For duct networks in which temperature is constant and  $p_1$  and  $U_1$  are the only integration variables of interest, use ANCHOR after the BEGIN segment, so TBRANCH will not worry about how to split  $\dot{H}_{\text{tot}}$  and UNION will not think of temperature as a potential default target.

Nested teebranches are allowed, as shown in Figs. 11.1(b) and (c). However, never use a BEGIN segment within a teebranch to re-initialize variables.

The incoming values of the RPN variables F1 and vol are retained for the trunk, and these variables are initialized to zero for the teebranch.

### Input variables:

**Re(Zb)** (Pa·s/m<sup>3</sup>) The real part of the branch impedance  $Z_{\text{br}}$ .

**Im(Zb)** (Pa·s/m<sup>3</sup>) The imaginary part of the branch impedance  $Z_{\text{br}}$ .

**HtotBr** (W) This optional input variable,  $\dot{H}_{\text{br}}$ , can be used in insulated mode. It specifies how much total power flows into the teebranch. The default value is equal to  $\dot{E}_{\text{br}} + \dot{N}_{\text{br}}mw_m$ , which is simply  $\dot{E}_{\text{br}}$  when there is no steady flow.

**NdotBr** (mol/s) This input variable,  $\dot{N}_{\text{br}}$ , appears when “Enable Ndot” is checked in an upstream BEGIN. It specifies how much steady flow goes into the teebranch.

**NLdotB** (mol/s) This input variable,  $\dot{N}_{L,\text{br}}$ , appears when “Enable Ndot” is checked in an upstream BEGIN and when a gas mixture and MIX\*\* segments are in use. It specifies how much steady flow of the light component of the mixture goes into the teebranch. The default value represents bulk flow into the branch,  $\dot{N}_{L,\text{br}} = n_L \dot{N}_{\text{br}}$ .

### Potential targets:

None.

### Master–slave links:

The imaginary part of the branching impedance can be linked to the real part in two possible ways: to preserve either the magnitude or the phase while the real part is changed.

(For an introduction to master–slave links, and a complete listing of these link options for all segments, see Section 2.7.2.)

## Format and examples:

```
TBRANCH      the fork in the road, in the default "insulated" mode
4.412E+07 a  Re(Zb) Pa-s/m^3 G
-3.528E+06 b  Im(Zb) Pa-s/m^3 G
0.49        c  HtotBr          G
```

```
TBRANCH      the fork in the road, if "ANCHOR" segment appears somewhere upstream
4.412E+07 a  Re(Zb) Pa-s/m^3 G
-3.528E+06 b  Im(Zb) Pa-s/m^3 G
```

## Calculations:

The TBRANCH segment leaves  $T_m$ ,  $p_1$ , and  $p_{20HL}$  unchanged. Other variables change according to

$$U_{1,br} = p_1/Z_{br}, \quad (11.3)$$

$$U_{1,trunk} = U_{1,in} - U_{1,br}, \quad (11.4)$$

$$\dot{H}_{br} = \dot{H}_{in}, \quad (11.5)$$

$$\dot{H}_{trunk} = \dot{H}_{in} - \dot{H}_{br}, \quad (11.6)$$

$$\dot{N}_{br} = \dot{N}_{in}, \quad (11.7)$$

$$\dot{N}_{trunk} = \dot{N}_{in} - \dot{N}_{br}, \quad (11.8)$$

$$\dot{N}_{L,br} = \dot{N}_{L,in}, \quad (11.9)$$

$$\dot{N}_{L,trunk} = \dot{N}_{L,in} - \dot{N}_{L,br}, \quad (11.10)$$

$$F_{1,br} = 0, \quad (11.11)$$

$$F_{1,trunk} = F_{1,in}, \quad (11.12)$$

$$\text{vol}_{br} = 0, \quad (11.13)$$

$$\text{vol}_{trunk} = \text{vol}_{in}. \quad (11.14)$$

TBRANCH has an extra output, `EdotTr`, to display the acoustic power flowing past the branch in the trunk.

### 11.2.2. UNION

#### General description:

A UNION is used in concert with a TBRANCH and a SOFTEND somewhere upstream, to create loops of segments as illustrated in Fig. 11.1(c). In this arrangement, when it encounters a TBRANCH, DELTAEC treats subsequent segments as the sequential members of a branch until it reaches a SOFTEND; then it “returns to the trunk,” treating the rest of the segments as trunk members. Subsequently, a UNION segment in the trunk tells DELTAEC where to connect the teebranch’s SOFTEND back to the trunk.

When reconnected with a UNION, the teebranch’s SOFTEND potential impedance targets should not be used; instead, enable the UNION’s targets to ensure that  $p_1$  (complex) at the UNION is equal to  $p_1$  (complex) at the SOFTEND of the teebranch. Often, guessing the teebranch impedance determines how the (complex) volume flow rate splits up at the TBRANCH and allows the complex  $p_1$  target at the UNION to be met.

In all but the simplest acoustic duct network, additional potential targets can be used at a UNION to properly tie the teebranch's SOFTEND to the trunk at the UNION with continuity of relevant variables. These include  $T_m$ ,  $p_{2,0,HL}$ , and  $n_L$ .

UNION targets are unusual, because their target values are dynamically rewritten by DELTAEC during each iteration, so they always reflect the current result at the referenced SOFTEND. (When the user enables a UNION's target, the target input parameters, e.g., magnitude and phase of pressure, can be initialized to any values, because DELTAEC will overwrite them during each integration pass with the current magnitude and phase of pressure at the referenced SOFTEND via automatic sameas links from the UNION to the SOFTEND.)

Oscillating volume flow rates are added at the UNION, as are time-averaged energy flows and mole fluxes. The RPN variables F1 and vol are also added.

When TBRANCH and UNION are used for duct networks, where temperature is constant and hence  $p_1$  and  $U_1$  are the only variables of interest, an ANCHOR segment upstream of the TBRANCH will eliminate confusion that might arise from the display of uninteresting, irrelevant variables.

### Input variables:

**SegNum** The segment number of a SOFTEND, upstream of the UNION, which terminates a teebranch and which should be connected to the trunk here. See Fig. 11.1(c).

### Potential targets:

**|p|Sft** (Pa) If used as a target, DELTAEC's shooting method compares its calculation of  $|p_1|$  here in the trunk with its calculation of  $|p_1|$  at the SOFTEND identified by SegNum. DELTAEC updates the value of this variable continuously.

**Ph(p)S** (deg) If used as a target, DELTAEC's shooting method compares its calculation of the phase of  $p_1$  here in the trunk with its calculation of the phase of  $p_1$  at the SOFTEND identified by SegNum. DELTAEC updates the value of this variable continuously.

**TSoft** (K) This potential target appears in insulated mode. If used as a target, DELTAEC's shooting method compares its calculation of  $T_m$  here in the trunk with its calculation of  $T_m$  at the SOFTEND identified by SegNum. DELTAEC updates the value of this variable continuously.

**p20HLS** (Pa) This potential target appears when "Enable Ndot" is checked in an upstream BEGIN. If used as a target, DELTAEC's shooting method compares its calculation of  $p_{2,0,HL}$  here in the trunk with its calculation of  $p_{2,0,HL}$  at the SOFTEND identified by SegNum. DELTAEC updates the value of this variable continuously.

**nLSoft** This potential target appears when a gas mixture is in use and any MIX\*\* segment appears in the model. If used as a target, DELTAEC's shooting method compares its calculation of  $n_L$  here in the trunk with its calculation of  $n_L$  at the SOFTEND identified by SegNum. DELTAEC updates the value of this variable continuously.

## Master-slave links:

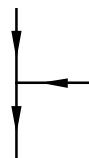
None.

## Format and examples:

```

!----- 13 -----
UNION      reconnect the "final" end of the loop, segment 8, here
  8.000    a SegNum          2.5420E+05 A |p|      Pa
  2.5420E+05 b |p|Sft      Pa      = 13A?      1.9444 B Ph(p)      deg
  1.9444    c Ph(p)S      deg      = 13B?      3.0515E-02 C |U|      m^3/s
  781.66    d TSoft      K          (t)      -44.624 D Ph(U)      deg
  1603.6    e p20HLS      Pa          = 13H?      2674.7 E Htot      W
                                          2666.4 F Edot      W
                                          702.66 G T          K
                                          1603.6 H p20HL      Pa

```



```

UNION      connect two paths together, matching complex pressure without worrying about any other variables
  4.000    a segment number of SOFTEND of the TBRANCH somewhere upstream
  3.e4     b |p|Sft      Pa
  1.234    c ph(p)S      deg

```

## Calculations:

Calculations at UNION are simple arithmetic:

$$T_{m,out} = T_{m,in}, \quad (11.15)$$

$$p_{1,out} = p_{1,in}, \quad (11.16)$$

$$U_{1,out} = U_{1,in} + U_{1,end}, \quad (11.17)$$

$$\dot{H}_{tot,out} = \dot{H}_{tot,in} + \dot{H}_{tot,end}, \quad (11.18)$$

$$\dot{N}_{out} = \dot{N}_{in} + \dot{N}_{end}, \quad (11.19)$$

$$\dot{N}_{L,out} = \dot{N}_{L,in} + \dot{N}_{L,end}, \quad (11.20)$$

$$F_{1,out} = F_{1,in} + F_{1,end}, \quad (11.21)$$

$$vol_{out} = vol_{in} + vol_{end}, \quad (11.22)$$

where the subscript “in” refers to the closest trunk segment upstream of the UNION (i.e., almost always simply the segment immediately preceding the UNION) and the subscript “end” refers to the SOFTEND that is identified by **SegNum** in the first input parameter of the UNION, i.e., the SOFTEND that terminates the teebranch that is being joined to the trunk at the UNION. Note that Eq. (11.18) implies  $\dot{H}_{2,k,out} = \dot{H}_{2,k,in} + \dot{H}_{2,k,end}$  only if  $T_m$  is successfully targeted.

In thermally anchored mode, Eq. (11.18) is replaced by  $\dot{H}_{tot,out} = \dot{E}$ .

## 11.3. Insulation

### 11.3.1. ANCHOR and INSULATE

#### General description:

By default, DELTAEC assumes that the only places total power can flow into or out of a model are

- heat exchangers, where heat can be added to the total power (or, if heat is negative, subtracted from the total power),
- electroacoustic transducers, where electrical power can be added to or subtracted from the total power,
- `BRANCH`, `OPNBRANCH`, and `PISTBRANCH`,
- `BEGIN`, unless  $\dot{H}_{\text{tot}}$  is explicitly set equal to zero (or is set equal to  $\dot{E} + mw_m\dot{N}$  which in turn happens to equal zero),
- `SOFTEND` and `HARDEND`, unless  $\dot{H}_{\text{tot}}$  is targeted to zero.

Other segments—stacks and regenerators, ducts and cones, and compliances, surfaces, and impedances—are assumed to have insulation wrapped around their side-wall boundaries by default, so whatever total power  $\dot{H}_{\text{tot}}$  flows into them from the previous segment must flow out of them to the subsequent segment.

However, sometimes it is useful to change the behavior of ducts, cones, compliances, surfaces, and impedances so the total power flowing out of them is equal to the acoustic power out of them. In a series of such segments, acoustic power dissipation in each segment “disappears” from the model, as it might if the segment were thermally anchored to a heat sink at the same temperature as the local gas temperature  $T_m$ . This behavior can be started by an `ANCHOR` segment before the first of a series of segments that are supposed to be thermally anchored. Thereafter, such segments obey  $\dot{H}_{2,k,\text{out}} = \dot{E}_{\text{out}}$ . The complementary `INSULATE` segment, or a `BEGIN` segment, can be used to return the computation to the default, thermally insulated mode in which  $\dot{H}_{2,k,\text{out}} = \dot{H}_{2,k,\text{in}}$ . `ANCHOR` is most useful for very low power systems and simple systems where total energy flow is irrelevant, and of course for components that actually are immersed in a heat sink.

Stacks and regenerators remain insulated when `ANCHOR` is in effect.

Some users who are interested only in acoustics, not thermoacoustics, habitually put `ANCHOR` immediately after `BEGIN`, to reduce irrelevant clutter.

Use the default, insulated mode for ducts, compliances, or impedances that come between stacks and heat exchangers, and for modeling insulated acoustic networks attached to thermoacoustic devices.

## Format and examples:

```
ANCHOR  this segment has no inputs or outputs
INSULATE  this segment also has no inputs or outputs
```

## 11.4. Math segments

Use this class of segments to access or display variables that DELTAEC would not otherwise display, to perform simple arithmetic operations, and to create targets other than DELTAEC default targets. See Chapter 4 for many examples.



Table 11.1: List of RPN number-entry methods. Each number is piled on top of the stack.

Item	Description	Comment or example
<Constant>	Real or complex number	8.3; 6.02e23; (-0.1, 1)
pi	3.1415926536...	
i	$\sqrt{-1}$	complex
<Input address>	Segment number and l.c. letter	3c
<Output address>	Segment number and cap letter	3F
<filename:addr>	Filename:segment number and letter (linking two or more models together)	bottle:3F
<%:addr>	% = shortcut for previous filename	bottle:3A %:3a * %:3C /
inp	Parameter “a” of this RPN segment	
rcl	Recall from Storage Register	see also sto
#	Put a copy of the stack’s top number on top of the stack.	
lstx	Recall what was on top of the stack before the most recent math operation, and put it on top of the stack.	

### 11.4.1. RPN

#### General description:

Results of the RPN segment are computed by interpreting the formula (line “b”) in Reverse Polish Notation, as described in [17, 18] and especially in all of Chapter 4. The formula is a sequence of numbers, addresses, internal DELTAEC variables, and algebraic operators. Variables such as Htot can be typed as lower case; DELTAEC will automatically convert to its preferred case. The formula in an RPN should not be longer than 80 characters. As with RPN calculators, when operators do not consume all numbers on the “stack,” more than one output is generated—a feature that can be exploited to display multiple results. The stack, whose members may be complex numbers, grows downward, from A–J. Valid operators and inputs are summarized in Tables 11.1–11.6. To display an RPN formula with parentheses, click on the output parameter, and then click on List Linkages.

#### Format and examples of RPN:

```

!-----6-----
RPN  magU1 over 2 pi freq Area (meters)
! gas displacement amplitude after duct in segment five
0.00                                0.0153  A ChngeMe
5C 2 / pi / 0b / 5a /

! -----23 -----
RPN  efficiency divided by Carnot's efficiency
! electrical power out of alternator is 19G
! heat into engine is 6e
! hot temperature is 6f
! ambient temperature is 8f
1.00                                0.2438  A eta_rel
19G 6e / 1 8f 6f / - /

! ----- 21 -----
RPN  B=diam.  A= Reynolds number amplitude here

```

Table 11.2: RPN variable-entry methods. Each number is piled on top of the stack.

Item	Description	Comment or units
<b>Thermophysical properties</b>		
a	$a$ ; sound speed	m/s
beta	$\beta$ ; expansion coefficient	1/K
cp	$c_p$ ; heat capacity	J/kg·K
cs	$c_s$ ; solid heat capacity	J/kg·K
dk	$\delta_\kappa$ ; thermal penetration depth	m
dn	$\delta_\nu$ ; viscous penetration depth	m
ds	$\delta_s$ ; solid thermal penetration depth	m
D12	$D_{12}$ ; binary mass diffusion coefficient	m <sup>2</sup> /s, for gas mixtures
enth	$w_m$ ; enthalpy per unit mass	J/kg
gamma	$\gamma = c_p/c_v$	
ks	$k_s$ ; solid thermal conductivity	W/m·K
kT	$k_T$ ; thermal diffusion ratio	for gas mixtures
k	$k$ ; thermal conductivity	W/m·K
m	$m$ ; molar mass (avg. mass of mixture)	kg/mol
mu	$\mu$ ; viscosity	kg/m·s
mL	$m_L$ ; molar mass of light comp. of mixture	kg/mol
mH	$m_H$ ; molar mass of heavy comp. of mixture	kg/mol
nLzero	$n_{L,0}$ ; reference mole fraction for $\dot{X}$	Eq. (8.9)
rho	$\rho_m$ ; density	kg/m <sup>3</sup>
rhos	$\rho_s$ ; solid density	kg/m <sup>3</sup>
Tzero	$T_0$ ; reference temperature for $w_m$ and $\dot{X}$	K
<b>State variables</b>		
Edot	$\dot{E}$ ; acoustic power flow	W
f	$f$ ; frequency	Hz
F1	$F_1$ ; cumulative force phasor	N; complex; <b>see text</b>
H2k	$\dot{H}_{2,k}$ ; thermoacoustic plus conduction power flow	W
Htot	$\dot{H}_{tot}$ ; total power flow, incl. steady flow	W
Ndot	$\dot{N}$ ; time-averaged molar flow	mol/s
nL	molar fraction of light gas in mixture	
NLdot	$\dot{N}_L$ ; time-avg. molar flow, light comp. of mixt.	mol/s
pm	$p_m$ ; mean pressure	Pa
p1	$p_1$ ; oscillatory pressure	Pa; complex
p20HL	$p_{2,0,HL}$ ; time-averaged head-loss pressure	Pa
p20tot	$p_{2,0,tot}$ ; time-averaged pressure [Eq. (8.11)]	Pa
Tm	$T_m$ ; mean temperature	K
U1	$U_1$ ; oscillatory volume flow rate	m <sup>3</sup> /s; complex
vol	cumulative gas volume	m <sup>3</sup> ; <b>see text</b>
w	$\omega$ ; radian frequency	rad/s
x	integration coordinate	m
Xdot	$\dot{X}$ ; exergy flow	W

Table 11.3: List of RPN unary functions. Each function removes a number from the top of the stack, acts on it, and puts the result on top of the stack. These functions accept and return complex numbers, unless otherwise noted. Whether phase angles are treated as radians or degrees is noted.

Item	Description	Comment or explanation
<b>Algebra</b>		
sqrt	$x = \sqrt{x}$	
sqrđ	$x = x * x$	
~	$x = -x$	change sign
abs	$x =  x $	absolute value
inv	$x = 1/x$	
real	$x = \text{real}(x)$	
imag	$x = \text{imag}(x)$	
mag	$x =  x $	magnitude of complex $x$
conj	$x = \tilde{x}$	complex conjugate of $x$
arg		phase of complex $x$ (degrees)
argr		phase of complex $x$ (radians)
<b>Trigonometry in degrees</b>		
sin; asin	$x = \sin(x); x = \sin^{-1}(x)$	real argument only
cos; acos	$x = \cos(x); x = \cos^{-1}(x)$	real argument only
tan; atan	$x = \tan(x); x = \tan^{-1}(x)$	real argument only
<b>Trigonometry in radians</b>		
sinr	$x = \sin(x)$	
asinr	$x = \sin^{-1}(x)$	real argument only
cosr	$x = \cos(x)$	
acosr	$x = \cos^{-1}(x)$	real argument only
tanr	$x = \tan(x)$	
atanr	$x = \tan^{-1}(x)$	real argument only
<b>Hyperbolic functions</b>		
sinh	$x = \sinh(x)$	
cosh	$x = \cosh(x)$	
tanh	$x = \tanh(x)$	
<b>Bessel functions</b>		
J0	$x = J_0(x)$	Bessel function of zero order
J1	$x = J_1(x)$	Bessel function of first order
Y0	$x = Y_0(x)$	Neumann function of zero order
Y1	$x = Y_1(x)$	Neumann function of first order
<b>Logarithms and exponentials</b>		
log	$x = \log_e(x)$	
exp	$x = e^x$	
log10	$x = \log_{10}(x)$	real argument only
tenx	$x = 10^x$	

Table 11.4: Ordinary RPN binary operations and functions. Each operation or function removes two numbers from the top of the stack, combines them, and puts the result on top of the stack. In this table, “ $x$ ” refers to the number on top of the stack, and “ $y$ ” refers to the number second from the top of the stack.

Item	Description	Comment or example
+	$x = y + x$	
-	$x = y - x$	
*	$x = y * x$	
/	$x = y / x$	
^	$x = y^x$	
min; max	$x = \min[x, y]; x = \max[x, y]$	real arguments only
avg	$x = (x + y) / 2$	
cmplx	$x = x(\cos y + i \sin y)$	$y$ in degrees
cmplxr	$x = x(\cos y + i \sin y)$	$y$ in radians
atan2	$x = \tan^{-1}(x/y);$ 2-argument, 4-quadrant arctangent	degrees
atan2r	$x = \tan^{-1}(x/y);$ 2-argument, 4-quadrant arctangent	radians

Table 11.5: RPN actions that have no effect on the RPN stack, but have an effect elsewhere.

Item	Description	Comment
sto	$S = x$	Store the number that’s on top of the stack in the Storage Register. (See also rcl)
=f	$f = x$	Set $f$ equal to the number that’s on top of the stack.
=F1	$F_1 = x$	Set the accumulated-force phasor equal to the number that’s on top of the stack.
=H2k	$\dot{H}_{2,k} = x$	Set $\dot{H}_{2,k}$ equal to the number that’s on top of the stack.
=Ndot	$\dot{N} = x$	Set $\dot{N}$ equal to the number that’s on top of the stack.
=nL	$n_L = x$	Set $n_L$ equal to the number that’s on top of the stack.
=NLdot	$\dot{N}_L = x$	Set $\dot{N}_L$ equal to the number that’s on top of the stack.
=NLzro	$n_{L,0} = x$	Set $n_{L,0}$ equal to the number that’s on top of the stack.
=pm	$p_m = x$	Set $p_m$ equal to the number that’s on top of the stack.
=p1	$p_1 = x$	Set $p_1$ phasor equal to the number that’s on top of the stack.
=p2OHL	$p_{2,0,HL} = x$	Set $p_{2,0,HL}$ equal to the number that’s on top of the stack.
=Tm	$T_m = x$	Set $T_m$ equal to the number that’s on top of the stack.
=Tzero	$T_0 = x$	Set $T_0$ equal to the number that’s on top of the stack.
=U1	$U_1 = x$	Set $U_1$ phasor equal to the number that’s on top of the stack.
=vol	vol = $x$	Set the accumulated volume equal to the number that’s on top of stack.

Table 11.6: RPN actions that affect the top two numbers on the RPN stack.

Item	Description	Comment or example
a<>b	$x = y; y = x$	Interchange top two numbers on the stack.

Table 11.7: RPN tokens that have no effect.

Item	Description	Comment or example
;	No effect	Punctuation mark for human readability.

```

! 20a is the area in segment 20
0.0000                                2838.5    A NR1
                                           0.123    B D(m)
20a 4 * pi / sqrt # U1 mag * rho * 20a / mu /

! ----- 128 -----
RPN some gas properties here
  0.0000  a G or T                                1.6667  A gamma
                                           519.22  B rho c
                                           0.6625  C Prandtl
mu cp * k / ; rho a * ; gamma

```

### Potential targets in RPN:

Input line **a** is the potential target to which output line **A** can be compared. If line **a** is used as a target, **A** should be real.

### Details of F1 calculation in RPN:

Using the **F1** variable in RPN segments, DELTAEC can calculate the net oscillating force exerted by the oscillating gas (and oscillating parts of speakers) on the surrounding pressure vessel, which is assumed to be rigid. As far as we know, the approach is not published anywhere, so the theory behind it is derived in Chapter 14. Here, we describe the implementation in DELTAEC, and we give two examples of its use.

DELTAEC always keeps track of the integral on the right-hand side of

$$F_1 = -i\omega \int (\rho_m u_1) dV = -i\omega \int \rho_m(x) U_1(x) dx \quad (11.23)$$

during the course of its other integrations. The derivation in Chapter 14 shows that this integral gives the total force amplitude exerted by the gas on the pressure vessel. The result is accessible to the user at any location by putting “**F1**” in line **b** of an RPN segment, yielding the complex number  $F_1$  and making it available for further RPN calculations. (As usual, lower-case typing is accepted.)

DELTAEC initializes **F1** to zero at **BEGIN** and at the beginning of the teebranch of every **TBRANCH**. The trunk’s **F1** and the referenced-**SOFTEND**’s **F1** are added in **UNION**. Any other segment adds its own **F1** contribution [according to Eq. (11.23)] to whatever total **F1** has accumulated before it.

These defaults can be overridden via RPN segments that include the change command “=**F1**”. The defaults must often be overridden for thermoacoustic systems with elbows or side

branches sticking out at different angles, to resolve the components of the force-amplitude vector in different directions and to take care of minus signs in toroidal acoustic paths.

For a 90° miter elbow, split the elbow into two segments. Displaying and then rezeroing the force variable between those segments (with “F1” and “=F1”, respectively) is a useful approach. For a smoothly radiused 90° elbow, strictly speaking we ought to do an integral of a vector velocity around the corner to obtain a vector force. This is beyond DELTAEC’s current capability. However to the extent that  $U_1$  does not vary from one end of such an elbow to the other, that integral yields simply  $2/\pi$  times DELTAEC’s full-elbow DUCT integral for the vertical direction and  $2/\pi$  times DELTAEC’s full-elbow DUCT integral for the horizontal direction. The user who requires  $F_1$  in such an elbow can use RPN calculations to implement the factor of  $2/\pi$ , as shown in the second example below.

For IESPEAKER and VESPEAKER segments, DELTAEC evaluates  $\int \rho_m U_1 dx$  in Eq. (11.23) as simply the speaker’s moving mass times its velocity phasor.

### Example 1

To find the oscillating velocity of a freely suspended, 14-kg pressure vessel enclosing a half-wavelength resonance driven by a noncompliant transducer at the far end, use this series of segments.

```

!----- 0 -----
BEGIN no transducer at this end
2.000E+06 a Mean P Pa
173.53 b Freq Hz
300.00 c TBeg K
1.0000E+05 d |p| Pa
0.0000 e Ph(p) deg
0.0000 f |U| m/s
0.0000 g Ph(U) deg
!----- 1 -----
SURFACE first end
sameas 2a a Area m^2
!----- 2 -----
DUCT
1.0000E-02 a Area m^2
0.3545 b Perim m
1.0000 c Length m
!----- 3 -----
SURFACE noncompliant transducer at this end
sameas 2a a Area m^2
!----- 4 -----
HARDEND
0.0000 a R(1/z) (t)
0.0000 b I(1/z) = 4H?
!----- 5 -----
RPN (a)=mass of pressure vessel (kg). [A]= velocity phasor of pressure vessel (m/s).
14.000 a G or T (t) (-8.8680E-05, 0.1309 ) m/s
F1 5a / i / w /
!----- 6 -----
RPN viscous drag force phasor (newtons)
0.0000 a G or T (t) ( 1.3536 , -1.3536 ) N
F1 p1 3a * Od 1a * - -
guessz 0b
targs 4b

```

We arbitrarily defined zero phase to be that of the pressure at segments 0 and 1. Segment 5 shows that the phase of the velocity is almost 90°, which is consistent with the phases of the pressure on the two ends of the half-wavelength resonator at segments 1 and 3. Segment 6 subtracts the normal end-cap forces from the total force to display what’s left over, which must be the integrated viscous shear force on the side walls.

### Example 2

To find oscillating forces and torques on the half-wavelength refrigerator shown in Fig. 11.2, we use a DELTAEC model like this:

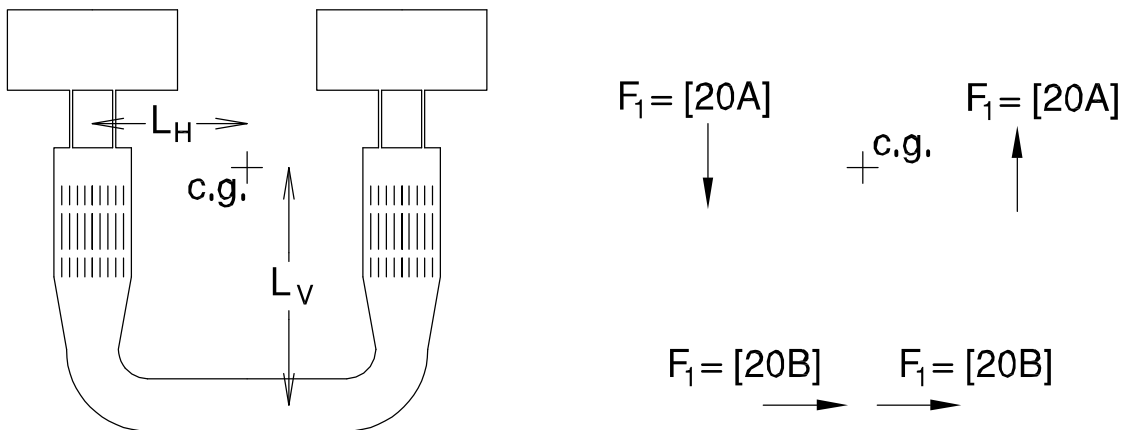


Figure 11.2: Example 2. Left, a sketch of a half-wavelength, two-speaker, standing-wave refrigerator. Right, force diagram, in which the directions of the arrows indicate the “positive” direction of calculated forces. In both parts of the figure, “c.g.” indicates the center of gravity.

```

TITLE symmetric speaker-driven refrigerator, example2.out
00 BEGIN in the motor enclosure. Guess temperature and complex p1.
01 INSULATE
02 SURFACE top lid
03 DUCT backside of piston = motor enclosure
04 SURFACE extra surface area for motor parts
05 IESPEAKER motor-driven piston. Guess phase of current.
06 RPN annular surface area around piston
07 SURFACE annular surface area around piston
08 DUCT short duct
09 HX ambient heat exchanger. Guess heat rejected and target temperature.
10 STKSLAB the stack
11 HX cold heat exchanger
12 CONE adapter to small-diam resonator
13 RPN capture F1 here, and reset it to zero
    0.0000 a G or T (t)                0.0000 A Newton
                                       ( 9.2386 , -113.62 ) B Newton
F1 0 =F1
14 DUCT 90 degree elbow
15 RPN vertical and horizontal forces on elbow; then set F1=0
    0.0000 a G or T (t)                0.0000 A Newton
                                       ( 3.0179E-03, -3.6644 ) B Newton
                                       ( 3.0179E-03, -3.6644 ) C Newton

F1 2 * pi / # 0 =F1
16 DUCT horizontal run of pipe, to mirror-image location
17 SOFTEND mirror-image plane. Target zero complex impedance and zero Hdot.
18 RPN for aesthetics, enforce phase(U1)=0 here. Target this.
    0.0000 a G or T = 18A?            -2.5791E-14 A deg
    U1 arg
19 RPN force due to horizontal segment
    0.0000 a G or T (t)                ( 6.9004E-04, -6.0857 ) A Newton
    F1
20 RPN [A]=net vertical force; [B]=net horizontal force (newtons)
    0.0000 a G or T (t)                ( 9.2416 , -117.29 ) A Newton
                                       ( 3.7079E-03, -9.7501 ) B Newton
15C 19A + 13B 15B +

```

Most of the input and output have been eliminated from the listing for clarity, leaving only the relevant RPN segments fully displayed. Segment 13 captures  $F_1$  for the vertical components, segment 15 captures  $F_1$  for the elbow using the approximation that  $U_1$  is uniform in the elbow, and segment 19 captures  $F_1$  for the horizontal part of the resonator. Then segment 20 adds up the previous results to yield the net horizontal and vertical  $F_1$ s for this half of the full refrigerator.

Considering the symmetry of the full refrigerator as shown in the figure (only half of which is modeled above with DELTAEC ) to get the minus signs and factors of two correct, we see that the total horizontal force exerted on the pressure vessel is  $2 \times [20B]$ , the total

vertical force is zero, and the total counter-clockwise torque around the center of gravity “c.g.” is  $2 \times ([20A]L_H + [20B]L_V)$  where  $L_H$  and  $L_V$  are distances between the center of gravity and the axes of the parts of the refrigerator as shown in the figure. Both the total horizontal force and the torque have time phases of about  $-90^\circ$ , relative to the time phase of the velocity at the mirror-image plane, which is set to  $0^\circ$  with segment 18.

### Details of vol calculation in RPN:

Beginning with version 6, DELTAEC keeps track of the cumulative gas volume in a series of segments while performing its other integrations. The result is accessible to the user at any location by putting “vol” in line b of an RPN segment. (A similar result is accessible through the VOLUME segment, described in Section 11.4.2, which calculates the volume in a range of segments, including both gas volume and the volume of fins, plates, and struts in STK\*\*s and heat exchangers.)

By default, DELTAEC initializes “vol” to zero at BEGIN and at the beginning of a TBRANCH’s teebranch, and adds the branch “vol” and trunk “vol” at UNIONS. Any other segment adds its own contribution to whatever total “vol” has accumulated before it. These defaults can be overridden via RPN segments that include the change command “=vol”.

The volume of a cone uses its average area  $(\text{AreaI} + \sqrt{\text{AreaI} \cdot \text{AreaF}} + \text{AreaF})/3$  and its length.

#### *Example*

```

TITLE symmetric speaker-driven refrigerator, example2v.out
00 BEGIN in the motor enclosure. Guess temperature and complex p1.
01 INSULATE
02 SURFACE top lid
03 DUCT backside of piston = motor enclosure
04 SURFACE extra surface area for motor parts
05 IESPEAKER motor-driven piston. Guess phase of current.
06 SURFACE annular surface area around piston
07 DUCT short duct
08 HX ambient heat exchanger. Guess heat rejected and target temperature.
09 STKSLAB the stack
10 HX cold heat exchanger
11 CONE adapter to small-diam resonator
12 DUCT 90 degree elbow
13 DUCT horizontal run of pipe, to mirror-image location
14 SOFTEND mirror-image plane. Target zero complex impedance and zero Hdot.
15 RPN for aesthetics, enforce phase(U1)=0 here. Target this.
16 RPN volume of gas (cubic meters) in segments 01 through 14
    0.0000 a G or T (t)                6.857E-02 A m3
vol
```

The vol operator in segment 16 causes the cumulative volume to be displayed: The total volume of gas in the DUCTs, HXs, STKSLAB, and CONE. (The other segments in this example have no gas volume.)

### 11.4.2. VOLUME

#### General description:

Result A of a VOLUME segment is the sum of the volumes in all duct, cone, stack, compliance, and heat exchanger segments beginning with BegAddr and ending with EndAddr (parameter letters are inconsequential). Porosity is ignored in calculating this volume, so the result is the volume of the gas plus the volume of internal fins, plates, etc. in heat exchangers and stacks. (In contrast, the vol variable, accessible in RPN segments, gives the cumulative



gas volume, not including internal fins and plates.) This segment is intended to give an indication of the overall size of an apparatus for doing trade-off analysis during design.

### Input variables:

**BegSeg** The first of the segments whose volumes are to be summed.

**EndSeg** The last of the segments whose volumes are to be summed.

### Potential targets:

Line “a” is the target at which VOLUME output line “A” can be aimed.

### Master–slave links:

None.

### Format and examples:

```
VOLUME
0.50  a targeted volume (cubic meters)
1     b BegSeg
10    c EndSeg
```

#### 11.4.3. CONSTANTS

### General description:

The **CONSTANTS** segment allows the user to input a dozen real numbers in one segment. This is useful when a subsequent **RPN** segment depends on a large number of nonstandard parameters that the user must control or that **DELTAEC** must guess. (Without **CONSTANTS**, a dozen **RPN** segments would be required.) This segment performs no calculations and has no potential targets.

### Format and examples:

```
CONSTANTS      parameters used elsewhere
1.55           a
2.00           b user-
3.22           c editable
4.00           d descriptors
5.00           e are here
6.00           f
7.00           g height(ft)
6.02E23        h
-0.048         i
0.00           j
0.00           k
0.00           l
```

## 11.5. Interfaces to external programs and files

### 11.5.1. BLKDATA

#### General description:

BLKDATA is a specialized segment designed for users who have a quantity of tabular data that they want DELTAEC to follow. The data may represent experimental results, the results from other computational methods, or simply an irregular parameter space to be explored that is not well served by DELTAEC's incremental-plot capability (e.g., irregular step size or parametric plotting). The data file is in text format, delimited with spaces or tabs, and it may contain up to 14 columns and any number of lines, all with the same number of entries. The first one or two lines can optionally contain descriptive strings that are read into the parameters' description and unit fields, respectively. Only one BLKDATA segment per model is allowed. The data file must have the suffix `<.blk>` and its root name is supplied in the heading of the BLKDATA segment. When the file is opened, the number of lines and columns are counted, and initial column headings are read in, if present. The number of input parameters for the segment is equal to the number of columns in the `<.blk>` file. During a `run`, each numeric line of the `<.blk>` file is read sequentially, and DELTAEC `runs` the model once for each line. Each line of the file is inserted into the input parameters of the BLKDATA segment, one line per data point. The use of `sameas` provides a direct means to tie the values of DELTAEC *input* parameters to the tabulated data.

Enforcing values that normally appear as DELTAEC *outputs* is a little more subtle. This requires an RPN target to be `sameas`'ed to an input parameter in the BLKDATA segment, so the RPN's calculated output value can be forced to hit the targeted value. This target must be used, of course, and an appropriately related input parameter must be added to the guess list. Segment 5 in the example below illustrates such a target.

#### Input variables:

The heading field, which is to the right of "BLKDATA," must contain the root name of the `<.blk>` file. For example, if the user's data is in `<mydata.blk>`, the BLKDATA segment looks like the example below.

#### Potential targets; Master-slave links:

None.

#### Format and examples:

If the BLKDATA segment looks like this:

```
BLKDATA    mydata
```

and the file `<mydata.blk>` looks like this:

Freq	p @0	SkrAmp
Hz	Pa	A
511.4	6653.	0.031
511.0	14401.	0.076
511.9	9872.	0.055

then DELTAEC will immediately transform the BLKDATA segment to this:

```

!----- 1 -----
BLKDATA mydata
511.40 a Freq Hz
6653.0 b |p|@0 Pa
3.1000E-02 c SkrAmp A

```

and will read subsequent lines of <mydata.blk> into place in subsequent iterations of the run. If the file <mydata.blk> is a log of experimental data, we can ask DELTAEC to emulate the experimental conditions by inserting appropriate **sameas** statements, as in the following fragment that matches the experimental applied frequencies, pressure amplitudes, and speaker currents:

```

!----- 0 -----
BEGIN the setup
7.8000E+04 a Mean P Pa
sameas 1a b Freq Hz
300.00 c TBeg K
sameas 1b d |p|@0 Pa
0.0000 e Ph(p)0 deg
0.0000 f |U|@0 m^3/s
0.0000 g Ph(U)0 deg
air Gas type
ideal Solid type
!----- 1 -----
BLKDATA mydata
511.40 a Freq Hz
6653.0 b |p|@0 Pa
3.1000E-02 c SkrAmp A
+ 2 SURFACE cavity back surface
+ 3 DUCT back duct
!----- 4 -----
VESPEAKER ...using an IESPEAKER would be simpler,
! but would not illustrate BLKDATA plus target.
1.5000E-04 a Area m^2 68.286 A |p| Hz
4.6000 b R ohms -91.445 B Ph(p) deg
5.5000E-05 c L H 5.2965E-06 C |U| m^3/s
3.3000 d BLProd T-m -90.455 D Ph(U) deg
4.0000E-04 e M kg 1.8081E-04 E Hdot W
8560.0 f K N/m 1.8081E-04 F Edot W
0.8000 g Rm N-s/m 1.8597E-03 G EdotIn W P
0.1945 h |V| V 0.1945 H Volts V
243.00 i Ph(V) deg 2.2615E-02 I Amps V
32.263 J Ph(V/I) deg
sameas 0 Gas type 121.99 K |Px| Pa
ideal Solid type -145.97 L Ph(Px) deg
!----- 5 -----
RPN use to match speaker current output to measured value
sameas 1c a = 5A? 2.2615E-02 A
41

```

## 11.5.2. SYSEXEC

### General description:

The SYSEXEC segment allows DELTAEC to send data to an external program via the computer's file-handling system, execute that external program, and bring any results of that external program back into DELTAEC. The use of **sameas** allows any input or output numbers in DELTAEC to be brought into the SYSEXEC segment's input parameters so they can be sent to the external program, and subsequently allows any DELTAEC input parameters or RPN tokens to be set equal to results of the external program.

The first action taken by SYSEXEC is to write all of its 14 real input parameters to a text file called <ToSys.dat> in single-column format. SYSEXEC then pauses DELTAEC execution and runs an external program using the entire text string specified in the heading of the SYSEXEC segment. The external program might or might not make use of the numbers in <ToSys.dat>. If the external program's results should be brought back in to DELTAEC, the

external program must put those results in a file named <FromSys.dat> in the same ASCII, single-column format. (If more than 14 values are required, the leftovers might be read in through a BLKDATA segment or a second SYSEXEC segment.) When the external program has finished running, control returns to DELTAEC, which then reads up to 14 parameters from the file <FromSys.dat> if it exists, putting those numbers in the output list of the SYSEXEC segment, and resumes its numerical integrations.

### Format and examples:

The following is a trivial example showing how a set of integers assigned to SYSEXEC input parameters can be sorted by the Windows CMD sort command. The SYSEXEC segment was initially prepared like this:

```

!-----1-----
RPN  a silly example
87.0  a                                13.000 A
5 8 +
!-----2-----
SYSEXEC  sort tosys.dat > fromsys.dat
3.0000  a In 1
4.0000  b In 2
6.0000  c In 3
7.0000  d In 4
2.0000  e In 5
sameas 1a f In 6
2.0000  g In 7
1.0000  h In 8
10.000  i In 9
sameas 1A j In 10
5.0000  k In 11
4.0000  l In 12
11.000  m In 13
9.0000  n In 14

```

When the SYSEXEC segment is encountered by DELTAEC during a run, DELTAEC creates the file <ToSys.dat> containing the parameters a through n:

```

3.0000
4.0000
6.0000
7.0000
2.0000
87.0000
2.0000
1.0000
10.0000
13.0000
5.0000
4.0000
11.0000
9.0000

```

and passes the segment's heading field, `sort tosys.dat > fromsys.dat`, to the operating system as a command string. Next, the operating system sends the newly created file <tosys.dat> to the sort function. The sort command sorts the numbers into ascending order. The results of the sort are piped into the <fromsys.dat> file using the 'greater than' character, i.e., the '>' character, in the command string. Finally, the SYSEXEC segment reads the <fromsys.dat> file and puts the results in the output column of the segment, so the segment ends up looking like this:

```

! Sort input parameters into output parameters (Windows version)
!-----1-----
RPN  a silly example
87.0  a                                13.000 A
5 8 +
!-----2-----
SYSEXEC  sort tosys.dat > fromsys.dat
3.0000  a In 1                                1.0000 A Out 1

```

4.0000	b	In	2	2.0000	B	Out	2
6.0000	c	In	3	2.0000	C	Out	3
7.0000	d	In	4	3.0000	D	Out	4
2.0000	e	In	5	4.0000	E	Out	5
sameas 1a	f	In	6	4.0000	F	Out	6
2.0000	g	In	7	5.0000	G	Out	7
1.0000	h	In	8	6.0000	H	Out	8
10.0000	i	In	9	7.0000	I	Out	9
sameas 1A	j	In	10	9.0000	J	Out	10
5.0000	k	In	11	10.0000	K	Out	11
4.0000	l	In	12	11.0000	L	Out	12
11.0000	m	In	13	13.0000	M	Out	13
9.0000	n	In	14	87.0000	N	Out	14

Then an RPN or **sameas** in a subsequent segment that points to output 2N will access the largest number in the original list, which was 87.000 in this example.

As an entertaining exercise, put a **SYSEXEC** segment at the end of a model, **sameas** some of the **SYSEXEC** inputs to all of the guesses in the model, **sameas** other **SYSEXEC** inputs to all of the targeted results in the model, and write a little batch file or python script (if python exists on your computer) to append the numbers in `<ToSys.dat>` as a single line in another file. Put the name of that batch file or python script in the **SYSEXEC**'s header. Ignore `<FromSys.dat>`. Running the model will then produce a nice table of the values of all the guesses and targeted results during each pass of the shooting method, showing the details of what DELTAEC's shooting method is actually doing.

The **SYSEXEC** segment can also be used to “insert” a user-created segment in a model. Use `<ToSys.dat>` to send  $p_m$  and  $f$  (both from segment 0) and integration variables such as  $p_1$ ,  $U_1$ , and  $T_m$  (from the preceding segment) to the operating system, run your own program to change the values of integration variables such as  $p_1$ ,  $U_1$ , and  $T_m$ , return your new values of  $p_1$ ,  $U_1$ , and  $T_m$  to DELTAEC in `<FromSys.dat>`, and finally use a subsequent RPN segment with tokens such as `=Tm` to pass your custom-calculated values to the next segment in DELTAEC.

## 12. GASES (AND LIQUIDS)

We provide an artificial temperature floor of 10 Kelvin to prevent DELTAEC from trying to use negative temperatures when it is really lost. Consequently, no temperature below 10 Kelvin can be used. (Most of the equations for the gases are very inaccurate when this limit is reached anyway.) This floor can be modified within Options in the Edit pulldown menu.

In what follows, **ta** is temperature in Kelvin, **t1** is temperature in Celsius.

DELTAEC looks for a 10-character field in the **BEGIN** segment to determine gas type. If you are modifying a model with a text editor, be sure to use plenty of trailing spaces after short gas names like “air” to get “gas-type” out of the 10-character field.

A “fluid” is a gas or a liquid. Gases are the most commonly used fluids in thermoacoustics and DELTAEC. However, liquid sodium and eutectic liquid sodium-potassium are included in DELTAEC because they are thermodynamically powerful, with non-negligible thermal expansion coefficients and non-negligible  $(\partial s/\partial p)_T$ , when close enough to their critical points. For further details on the use of these liquids in thermoacoustics, see Ref. [14] and references therein. Perhaps we should refer to the “gases” used in DELTAEC as “fluids,” but for simplicity we always call them gases despite the fact that some liquids are included.

Equations used here accurately cover ranges of  $T$  and  $p$  that have been of interest to the Los Alamos thermoacoustics team. DELTAEC issues no warning if it goes outside the accurate range. The user must study the equations here, or examine the results of these equations via the properties tokens in an **RPN** segment or the thermophysical-properties dialog, to decide if these equations are accurate enough or if a user-defined gas is called for.

To change gases in the middle of a model (e.g., after a flexible membrane), insert a **BEGIN** segment with the new gas, linking most or all other variables in the **BEGIN** to the previous **BEGIN** (to get  $p_m$  and  $f$ ) and to the results of the previous segment (e.g., to get  $p_1$  and  $U_1$ ) using **sameas**. See example in Section 11.1.2.

### 12.1. Helium (helium)

Ideal gas approximation for equation of state (including sound speed and expansion coefficient) and specific heat. Transport from Ref. [70].

```
k0=0.0025672*ta**0.716
mu=0.412e-6*ta**0.68014
```

### 12.2. Helium–argon mixtures (HeAr)

Number in **0j** is helium fraction. Ideal gas approximation for equation of state and specific heat. Transport from Refs. [71, 72].

```

k0he=0.0025672*ta**0.716
amuhe=0.412e-6*ta**0.68014
k0ar=(1.39e-4*ta**0.852-1.5e-8*(ta-300.)*(ta-300.))*(1.+2.e-8*pm)
amuar=(1.77e-7*ta**0.852-25.e-12*(ta-300.)*(ta-300.))*(1.+2.e-8*pm)
k0=x1*k0ar+x2*k0he-(k0ar+k0he)*x1*x2**1.5
mu=x1*amuar+x2*amuhe+0.2*(amuar+amuhe)*x1*x2
diff12= 7.49d0*(1.d0 - 0.05d0*x2)*(ta/295.d0)**1.66d0/pm
ktherm= 0.38d0 * x2**1.2d0 * x1**0.8d0

```

## 12.3. Helium–xenon mixtures (HeXe)

Number in 0j is helium fraction. Ideal gas approximation for equation of state and specific heat. Transport from Refs. [71, 72]. Our fits to the published transport data are only accurate for `frxe` < 0.5 or for `frxe` = 1.000.

```

k0he=0.0025672*ta**0.716
amuhe=0.412e-6*ta**0.68014
k0xe=4.75e-5*ta**0.84*(1.+1.e-7*pm)
amuxe=0.187e-6*ta**0.85*(1.+25.e-9*pm)
frxe=1.-fhe
k0=k0he*fhe+k0xe*frxe-2.*(k0he+k0xe)*frxe*fhe*fhe
mu=amuhe*fhe+amuxe*frxe+(amuhe+amuxe)*frxe*fhe*fhe*(0.8+3.7*fhe*fhe*(0.25-frxe))
diff12=5.9d0*(1.d0-0.074d0*litefrac)*(ta/300.d0)**1.72d0/0.96d0/pm
ktherm= 0.4d0 * litefrac**1.3d0 * frxe**0.7d0

```

## 12.4. Neon (neon)

Ideal gas approximation for equation of state and specific heat. Transport from Ref. [70].

```

k0=0.001149*ta**0.65907
mu=0.735e-6*ta**0.66065

```

## 12.5. Air (air)

Ideal gas approximation for equation of state and specific heat. Transport from Ref. [58].

```

parameter (tps=110.4, tpa=245.4, tpb=27.6, tp0=300., tpep=223.8306)
k0=2.624e-2*(ta/tp0)**1.5*(tp0+tpep)/(ta+tpa*exp(-tpb/ta))
mu=1.846e-5*(ta/tp0)**1.5*(tp0+tps)/(ta+tps)

```

## 12.6. Humid air and fog (HumidAir)

The number in 0j is the water mole fraction in a mixture of air and water. As with other gas mixtures, the water mole fraction can be modified or used as an independent plot variable. However, unlike other gas mixtures, the `MIX**` segments do not act on humid air. We worry that future improvements to understanding will bring dramatic change to this part of DELTAEC. Nevertheless, for the present:

In DELTAEC, we think of humid and wet air as a sort of single gas, having two or three interpenetrating components: dry air, water vapor, and sometimes liquid water.

The molar volume  $v$  and molar enthalpy  $w$  are DELTAEC's primary dependent variables, as functions of three independent variables: mean pressure  $p_m$ , mean temperature  $T_m$ , and mole fraction of water  $n_{\text{wat}}$ . To determine whether a given mixture is wet or merely humid, DELTAEC compares  $n_{\text{wat}}$  with  $p_{\text{sat}}/p_m$ , where  $p_{\text{sat}}(T_m)$  is the saturated vapor pressure at temperature  $T_m$ .

For humid air, DELTAEC uses the ideal-gas equation of state for  $v$  and a slightly non-linear temperature dependence for  $w$  (due to the  $T$  dependence of  $c_p$  for water vapor in the ASHRAE tables). Also in accordance with ASHRAE recommendations, we set  $\mu$  and  $k$  equal to their dry-air values.

In wet air, in accordance with Hiller's measurements [73], the *oscillating* thermodynamics in DELTAEC is that of humid air at saturation, while the *mean flow* thermodynamics includes the enthalpy of the liquid when  $\dot{N} \neq 0$ . In other words, if there is no mean flow, the calculation proceeds exactly as for an ideal-gas mixture and ignores the condensate, but if mean flow is nonzero DELTAEC performs stack integrations with ideal-gas-mixture properties in the momentum and continuity equations but with the heat of condensation/evaporation (plus the small enthalpy of the condensate itself) included in the energy equation as it is integrated to find  $dT_m/dx$ . In all cases, we ignore the dynamic effects of oscillating diffusion of the water vapor through the air, which have been described by Raspert [74] and Slaton [75].

Although the enthalpy of wet air calculated in DELTAEC's gas-properties algorithm includes the latent heat of freezing and melting as the condensate passes through  $0^\circ\text{C}$ , we have not yet incorporated this latent heat, of the liquid-to-solid phase transition, into the numerical integrations in STK\*\* segments, so integrating through  $T_m = 0^\circ\text{C}$  in wet air is of dubious value. (Anyway, we have not thought of any good reason to use wet air in DELTAEC below  $0^\circ\text{C}$ , because we believe the stack would simply plug up with ice.)

The saturated vapor pressure is accurate from  $-100^\circ\text{C}$  to  $370^\circ\text{C}$ ; other properties are reasonably accurate from  $-50^\circ\text{C}$  to  $150^\circ\text{C}$ .

## 12.7. Nitrogen (nitrogen)

Ideal gas approximation for equation of state and specific heat. Transport from Ref. [70].

$$k_0 = 0.0003609 * ta^{**0.7512}$$

$$\mu = 0.3577e-6 * ta^{**0.6885}$$

## 12.8. Hydrogen (hydrogen)

Ideal gas approximation for equation of state and specific heat. Transport from Ref. [70].

$$k_0 = 0.002627 * ta^{**0.744}$$

$$\mu = 0.19361e-6 * ta^{**0.6723}$$

## 12.9. Deuterium (deuterium)

Ideal gas approximation for equation of state and specific heat. Transport from Ref. [70].

$$k_0 = 0.002795 * ta^{**0.686}$$

$$\mu = 0.2726e-6 * ta^{**0.6721}$$



## 12.10. Carbon dioxide (CO2)

These equations provide a 1% match to the webbook.nist.gov/chemistry/fluid/ data from 300 K to 1100 K at 1 bar, 5 bar, and 10 bar.

```
DATA cpco2/ 560.0d0, 1.13d0, -0.00045d0, 4.0d-9, 1.5d0, 100.0d0 /
DATA aco2/ -0.015d0, 7.0d0, 1.0d-5, 400000.0d0 /
DATA rhoco2/ 4.3d-9, 1.7, 70.0 /
DATA kco2/ 2.8d-5, 1.1395d0, 4.0d-4, 4.0d-10, -2.2d-8, 625.0d0 /
DATA muco2/ 1.4187d-7, 0.8216d0, 1.5416d-7, 0.8094d0, -8.0d-12, 650.0d0, 8.0d-7 /
oldgam=1.2857d0
massavg=0.04401d0
r=runiv/massavg
cp=cpco2(1)+cpco2(2)*ta+cpco2(3)*ta**2+cpco2(4)*pm**cpco2(5)*ta*dexp(-ta/cpco2(6))
enth=cpco2(1)*(ta-tzero)+cpco2(2)/2.*(ta**2-tzero**2)+cpco2(3)/3.*(ta**3-tzero**3)
& -cpco2(4)*pm**cpco2(5)*cpco2(6)
& *(dexp(-ta/cpco2(6))*(1+ta/cpco2(6))-dexp(-tzero/cpco2(6))*(1+tzero/cpco2(6)))
entro=cpco2(1)*dlog(ta/tzero)+cpco2(2)*(ta-tzero)
& +cpco2(3)/2.*(ta**2-tzero**2)-cpco2(4)*pm**cpco2(5)*cpco2(6)
& *(dexp(-ta/cpco2(6))-dexp(-tzero/cpco2(6)))
a=dsqrt(oldgam*r*ta)+aco2(1)*(1+dexp(-pm/aco2(4)))*ta+aco2(2)-aco2(3)*pm
rho=pm/r/ta+rhoco2(1)*pm**rhoco2(2)*dexp(-ta/rhoco2(3))
beta=(pm/r/ta**2+rhoco2(1)/rhoco2(3)*pm**rhoco2(2)*dexp(-ta/rhoco2(3)))/rho
gamma=ta*beta**2 * a**2/cp + 1.0d0
k0=kco2(1)*ta**kco2(2)+kco2(3)+kco2(4)*pm+ kco2(5)*(ta-kco2(6))**2
mu=muco2(1)*ta**muco2(2)+(pm/1.0d6-1.0d0)*(muco2(3)*ta**muco2(4)
& -muco2(1)*ta**muco2(2))+muco2(5)*(ta-muco2(6))**2+muco2(7)
```

## 12.11. Neon-xenon mixtures (NeXe)

Number in 0j is neon fraction. Ideal gas approximation for equation of state and specific heat. Transport from Refs. [71, 72]. (Thermal conductivity not very accurate for high xenon concentrations.)

```
k0he=0.001149*ta**0.65907
amuhe=0.735e-6*ta**0.66065
k0xe=4.75e-5*ta**0.84*(1.+1.e-7*pm)
amuxe=0.187e-6*ta**0.85*(1.+25.e-9*pm)
frxe=1.-fhe(ns)
k0=k0he*fhe(ns)+k0xe*frxe-1.3*(k0he+k0xe)*frxe*fhe(ns)**2.5
mu=amuhe*fhe(ns)+amuxe*frxe+0.12*(amuhe+amuxe)*frxe*fhe(ns)**4
diff12=2.21d0*(1.d0+0.0326d0*frxe/(1.d0+0.531d0*frxe))*(ta/300.d0)**1.71d0/pm
ktherm=0.258d0 * litfrac**1.31d0 * frxe**0.69d0
```

## 12.12. Natural-gas combustion products (NGCbProd)

Natural Gas combustion products with 5% excess air. Use around 1 atm only. Our 1993 notes say “Data supplied by British Gas (private communication from R. J. Tucker to Bill Ward) for typical North-Sea gas. Molar weight is a fit to data in Ref. [76] between 288 K and 4000 K.”

```
gamma=1.4
cp=gasprop(ta,1392.02d0,39.3769d0,-3.89819d0,-0.0317961d0,
& 0.0327554d0,-1.44149d-3)
if (ta.gt.2000.) then
  mass=27.9495-7.81175*dexp(-((ta-4151.85)/1047.42)**2)
else
  mass=27.84
endif
r=8314.
a=dsqrt(gamma*r*ta/mass)
rho=pm*mass/(r*ta)
beta=1./ta
k0=gasprop(ta,0.0997279d0,0.0125516d0,6.73728d-5,4.22761d-4,
& 1.43198d-4,1.35508d-5)
mu=gasprop(ta,50.2973d0,4.68523d0,-0.12061d0,0.0140082d0,
& -0.001488951d0,4.97968d-5)*1.d-6
```

```

goto 900
real*8 function gasprop(ta,a,b,c,d,e,f)
real*8 z,ta,a,b,c,d,e,f
z=(ta-1400)/200
gasprop=a+z*(b+ z*(c + z*(d +z*(e+f*z))))
return
end

```

## 12.13. Liquid sodium (sodium)

Data for sodium from Ref. [77].

```

a0=2578.
at1=-.52
ap=6.1e-7
r0=950.1
rt1=-2.2976e-1
rt2=-1.46e-5
rt3=5.638e-9
c0=1.4361e3
ct1=-5.8024e-1
ct2=4.6208e-4
k0=.918e2-4.9e-2*t1
if (t1.le.500.) then
  e1=.697
  e2=1.235e-5
else
  e1=1.04
  e2=8.51e-6
endif
a=a0+at1*t1
rho=r0+rt1*t1+rt2*t1**2+rt3*t1**3
beta=(-rt1-2.*rt2*t1-3.*rt3*t1**2)/rho
bt=beta**2-(2.*rt2+6.*rt3*t1)/rho
cp=c0+ct1*t1+ct2*t1**2
rp=1./a/a+ta*beta**2/cp
bp=-beta/(rho*a**2)+2.*at1/(rho*a**3)-beta**2/rho/cp
bp=bp-2.*ta*beta*bt/rho/cp-ta*beta**3/rho/cp
bp=bp+ta*beta**2*(ct1+2.*ct2*t1)/rho/cp/cp
cpp=-ta*(beta**2+bt)/rho
c So far, everything is evaluated at p=0.
a=a+ap*pm
rho=rho+rp*pm
beta=beta+bp*pm
cp=cp+cpp*pm
gamma=1.+ta*beta**2*a**2/cp
mu=e2*rho**(1./3.)*exp (e1*rho/ta)

```

## 12.14. Liquid sodium–potassium eutectic (NaK-78)

This is for eutectic sodium–potassium; data from Ref. [77].

```

a0=2051.
at1=-.53
ap=0.
r0=876.4
rt1=-2.183e-1
rt2=-2.982e-5
rt3=0.
c0=970.69
ct1=-.36903
ct2=3.4309e-4
k0=21.4+2.07e-2*t1-2.2e-5*t1**2
if (t1.le.400.) then
  e1=.688
  e2=1.16e-5
else
  e1=.979
  e2=8.2e-6
endif
a=a0+at1*t1
rho=r0+rt1*t1+rt2*t1**2+rt3*t1**3
beta=(-rt1-2.*rt2*t1-3.*rt3*t1**2)/rho
bt=beta**2-(2.*rt2+6.*rt3*t1)/rho
cp=c0+ct1*t1+ct2*t1**2
rp=1./a/a+ta*beta**2/cp
bp=-beta/(rho*a**2)+2.*at1/(rho*a**3)-beta**2/rho/cp
bp=bp-2.*ta*beta*bt/rho/cp-ta*beta**3/rho/cp

```

```

bp=bp+ta*beta**2*(ct1+2.*ct2*t1)/rho/cp/cp
cpp=-ta*(beta**2+bt)/rho
c So far, everything is evaluated at p=0.
a=a+ap*pm
rho=rho+rp*pm
beta=beta+bp*pm
cp=cp+cpp*pm
gamma=1.+ta*beta**2*a**2/cp
mu=e2*rho**(1./3.)*exp (e1*rho/ta)

```

## 12.15. User-defined gases

### 12.15.1. User-defined pure gases

Files that define non-standard gases can have any name valid under the operating system under which DELTAEC is running, and should end with the extension `<.tpf>`. If the root filename is the same as any pre-defined gases, DELTAEC replaces its internal calculations for that gas with those given in the user's file. The `<.tpf>` file should be in the same directory or folder as the model file. The name of the gas is set to the root filename of the user-defined gas file. Up to five distinct user-defined gases can be active at one time. Please limit the root of the file name to eight characters.

Each of five gas properties is specified by a line containing 1–10 real coefficients to be read in as  $C_0 C_1 \dots C_9$ , where trailing unspecified parameters are taken to be zero. The order of the property lines is  $\rho_m$ ,  $c_p$ ,  $k$ ,  $a^2$ , and  $\mu$ . Comment lines can be inserted anywhere in the `<.tpf>` file with an initial `!`, and blank lines are not allowed.

It is critical that the properties be arranged in this order:

- line 1:  $\rho_m$
- line 2:  $c_p$
- line 3:  $k$
- line 4:  $a^2$
- line 5:  $\mu$

Each of the five properties is derived from its 10 coefficients using the following equation:

$$\text{gas property} = C_0 + C_1 \frac{p_m}{T_m + C_2 p_m} + C_3 T_m + C_4 T_m^2 + C_5 T_m^{C_6} + C_7 p_m^2 T_m^{C_8} + C_9 p_m, \quad (12.1)$$

where  $T_m$  and  $p_m$  are the absolute temperature (K) and mean pressure (Pa). An example `<.tpf>` file is shown in Section 2.6.3.

Equation (12.1) is a compromise between simplicity and flexibility; it is intended for use with a variety of simple expressions for gases and liquids and has a uniform syntax for specifying all five properties. There is only a limited mean-pressure dependence, suitable for nearly ideal gases; for more complicated mean-pressure dependence, multiple `<.tpf>` files should be written for each range of mean pressure used.

DELTAEC also needs the ratio of specific heats,  $\gamma$ , and the expansion coefficient  $\beta$ , but these are calculated internally from

$$\beta = -\frac{1}{\rho} \frac{\partial \rho}{\partial T} \quad \text{and} \quad \gamma = 1 + \frac{T \beta^2 a^2}{c_p}. \quad (12.2)$$

To employ a user-defined gas, select its filename as the gas type in a `BEGIN` segment.

User-defined gases specified by `<.tpf>` files cannot be used with nonzero  $\dot{N}$ , because Eq. (6.1) requires  $m$ , which is not available in `<.tpf>` format. To circumvent this shortcoming when a user-defined gas must be used with nonzero  $\dot{N}$ , use a `<.tpm>` file as described in the next Section, and set the mole fraction of the unused component to zero in the `BEGIN` segment of the model.

### 12.15.2. User-defined gas mixtures

The user can define the properties of his/her own binary ideal-gas mixture that is not a member of DELTAEC's internal library of mixtures, by using a file `<filename.tpm>` in the folder in which the DELTAEC model is running. If `filename` is one of DELTAEC's known gases, DELTAEC will replace its own properties with the user's properties. Please limit `filename` to eight characters.

The file format, shown below, is similar to that of `<.tpf>` files for pure gases. Blank lines are not allowed, and extra lines beginning with “!” are regarded as comments, which are very useful for keeping track of which line is which.

Properties are functions of mean pressure  $p_m$  (Pa), absolute temperature  $T_m$  (K), and mole fraction  $n_L$ . Each function is specified by a line containing a sequence of real coefficients (separated by blanks) which are read sequentially. Unused trailing parameters are set to zero. The properties must be arranged in this order:

line 1:  $m_L m_H \gamma_L \gamma_H$

line 2:  $k_{\text{pure } L}$

line 3:  $k_{\text{pure } H}$

line 4:  $\mu_{\text{pure } L}$

line 5:  $\mu_{\text{pure } H}$

line 6:  $k_{\text{mix}}$

line 7:  $\mu_{\text{mix}}$

line 8:  $D_{12}$

line 9:  $k_T$

The four numbers in line 1, which are  $m_L$ ,  $m_H$ ,  $\gamma_L$ , and  $\gamma_H$ , are used to calculate state properties based on ideal-gas behavior as follows:

$$m_{\text{avg}} = m_L n_L + m_H (1 - n_L), \quad (12.3)$$

$$\rho_m = \frac{p_m m_{\text{avg}}}{R_{\text{univ}} T_m}, \quad (12.4)$$

$$\frac{1}{\gamma - 1} = \frac{n_L}{\gamma_L - 1} + \frac{1 - n_L}{\gamma_H - 1}, \quad (12.5)$$

$$a^2 = \frac{\gamma R_{\text{univ}} T_m}{m_{\text{avg}}}, \quad (12.6)$$

$$c_p = \frac{\gamma R_{\text{univ}}}{\gamma - 1 m_{\text{avg}}}. \quad (12.7)$$

The pure-gas transport properties,  $k_{\text{pure } L}$ ,  $k_{\text{pure } H}$ ,  $\mu_{\text{pure } L}$ , and  $\mu_{\text{pure } H}$ , are calculated from numbers  $C_0 C_1 \dots C_9$  in each of lines 2 through 5 using the following equation:

$$\text{property} = C_0 + C_1 \frac{p_m}{T + C_2 p_m} + C_3 T + C_4 T^2 + C_5 T^{C_6} + C_7 p_m^2 T^{C_8} + p_m C_9. \quad (12.8)$$

Equation (12.8) is a compromise between simplicity and flexibility. It is intended for use with most coefficients set to zero in a variety of simple expressions. Finally, the numbers in lines 6 through 9 are used in these equations:

$$\text{line 6: } k_{\text{mix}} = n_L k_{\text{pure } L} + (1 - n_L) k_{\text{pure } H} + (C_0 + C_1 T_m^{C_2}) n_L^{C_3} (1 - n_L)^{C_4}, \quad (12.9)$$

$$\text{line 7: } \mu_{\text{mix}} = n_L \mu_{\text{pure } L} + (1 - n_L) \mu_{\text{pure } H} + (C_0 + C_1 T_m^{C_2}) n_L^{C_3} (1 - n_L)^{C_4}, \quad (12.10)$$

$$\text{line 8: } D_{12} = \frac{C_0}{(p_m/[10^5 \text{ Pa}])} (1 - C_1 n_L) \left( \frac{T_m}{[300 \text{ K}]} \right)^{C_2} + C_3, \quad (12.11)$$

$$\text{line 9: } k_T = C_0 n_L^{C_1} (1 - n_L)^{C_2} \left( \frac{T_m}{[300 \text{ K}]} \right)^{C_3} + C_4. \quad (12.12)$$

As an example of the use of these coefficients, consider the example below for neon, a mixture of two isotopes.

```
! user-defined mixture: neon, a mixture of two isotopes, 20 and 22.
! Natural abundance is nL = 0.91
! mL(kg/mole)  mH(kg/mole)  gammaL  gammaH:
0.01999 0.02199 1.6667 1.6667
! k pure L (W/m-K):
0. 0. 0. 0. 0. 0.00122 0.65
! k pure H (W/m-K):
0. 0. 0. 0. 0. 0.00116 0.65
! mu pure L (kg/m-s):
0. 0. 0. 0. 0. 7.41E-7 0.66
! mu pure H (kg/m-s):
0. 0. 0. 0. 0. 7.70E-7 0.66
! k mixture (W/m-K):
0.
! mu mixture (kg/m-s):
0.
! D12 (m2/s):
0.53E-4 0. 1.72
! kT:
0.0267 1.0 1.0
```

To employ a user-defined mixture, select its filename as the gas type in a BEGIN segment.

## 13. SOLIDS

We provide an artificial temperature floor of 10 Kelvin to prevent DELTAEC from trying to use negative temperatures when it is really lost. Consequently no temperature below 10 Kelvin can be used.

In what follows, `ta` is temperature in Kelvin, `t1` is temperature in Celsius.

DELTAEC looks for a 10-character field to determine solid type. When editing a `<.out>` file with a text editor, be sure to use plenty of trailing spaces after short solid names like “mylar” to get “solid-type” out of the 10-character field.

Equations used here accurately cover ranges of  $T$  and  $p$  that have been of interest to the Los Alamos thermoacoustics team. No warning is issued by DELTAEC if it goes outside the accurate range. It is up to the user to study the equations here, or examine the results of these equations via the properties tokens in an RPN segment or the thermophysical-properties dialog, and decide if these equations are accurate enough or if a user-defined solid is necessary.

### 13.1. Ideal solid (ideal)

`ks`, `rhos`, and `cs` are effectively infinite, so  $\epsilon_s = 0$ .

```
ks=1.d8
rhos=1.d8
cs=1.d8
```

### 13.2. Copper (copper)

```
ks=398.-.0567*(ta-300.)
rhos=9000.
cs=420.
```

### 13.3. Nickel (nickel)

```
if (ta.lt.631) then
  ks=63.8+.08066*(631.-ta)
else
  ks=63.8+.02156*(ta-631.)
endif
rhos=8700.
cs=530.
```

### 13.4. Stainless steel (stainless)

```
rhos=8274.55 -1055.23 *dexp(-((T1-2171.05)/2058.08)**2)
ks=(266800*ta**(-5.2)+0.21416*ta**(-1.6))**(-0.25)
cs=(1.7054e-6*ta**(-0.88962)+23324/ta**6)**(-1/3) + 15/ta
```

Prior to version 3.5b2, DELTAE's stainless steel properties were very inaccurate at cryogenic temperatures.

### 13.5. Molybdenum (molybdenum)

```
rhos= 10868.6 -2637.52 * exp (-((T1-11383.7)/9701.36)**2)
cs= 253.791 +0.0583812 *T1-2.73919e-06*T1**2
ks= (33.9616 -0.00947953 *T1-4.12809e-08*T1**2)*4.186
```

### 13.6. Tungsten (tungsten)

```
cs=.13576e3*(1.-4805./ta**2)+.0091159*ta+2.31341e-9*ta**3
ks=135.5+1.05e4/ta-.023*ta
rhos=19254*(1.-3.*(-8.69e-5+3.83e-6*t1+7.92e-10*t1**2))
```

### 13.7. Kapton (kapton)

```
ks=0.2*(1.-exp(-ta/100.))
rhos=1445.-0.085*ta
cs=3.64*ta
```

### 13.8. Mylar (mylar)

```
ks=0.11+1.7e-4*ta
rhos=1400.-0.175*ta
cs=3.7*ta
```

### 13.9. Celcor (celcor)

Properties from Steve Garrett, Penn State University (private communication).

```
ks=2.5
rhos=2510.
cs=262.5 + 1.864*ta - 0.001011*ta**2
```

### 13.10. User-defined solids

User-defined solids, like user-defined gases, are derived from coefficients in user-written text files that must be named <filename.tpf>. Please limit `filename` to eight characters. Up to five user-defined solids can be active at once. Each of three properties is specified by a line containing 1–10 real coefficients to be read in as  $C_0 C_1 C_2 \dots C_9$ , where unspecified parameters are taken to be zero. The order of the property lines is

- line 1:  $\rho_s$
- line 2:  $c_s$
- line 3:  $k_s$

Comment lines can be added with an initial '!', and blank lines are not allowed.

Each of the three properties is derived from its 10 coefficients using the following equation:

$$\text{property} = C_0 + C_1 \exp(-C_2 T_m) + C_3 T_m + C_4 T_m^2 + C_5 T_m^{C_6} + C_7 p_m^2 T_m^{C_8} + C_9 p_m. \quad (13.1)$$

To request a user-defined solid, simply use the root file name as the solid type. The `<.tpf>` file should be in the same directory or folder as the model file. If the name matches any pre-defined solid name, the (constant) user-defined properties will replace DELTAEC's internal calculations.

User-defined gases and user-defined solids share the `<.tpf>` file extension. DELTAEC can tell the difference between a solid and a fluid via the number of non-comment lines in the file.



# Back Matter

# 14. DERIVATIONS

Here we present our derivations of a few equations that are used in DELTAEC but are not derived in the archival thermoacoustics literature.

## Force $F_1$ on pressure vessel

Consider a rigid, uniaxial pressure vessel containing a thermoacoustic gas oscillation at angular frequency  $\omega$ . Newton's laws of motion show that the force exerted *by* the gas (and by moving parts of transducers, if present) *on* the rigid pressure vessel equals the force exerted *on* the gas (and optional transducers) *by* the rigid vessel, and this force equals the rate of change of momentum  $\rho\mathbf{v}$  of the gas (and optional transducers). Consider only forces and motion at  $\omega$ . Then the total first-order oscillating force exerted on the rigid pressure vessel by the oscillations inside it, in the  $x$  direction, is

$$F_1 = -i\omega \int (\rho_m u_1) dV = -i\omega \int \rho_m(x) U_1(x) dx. \quad (14.1)$$

The right-hand side of this equation is what DELTAEC's RPN force operator,  $\mathbf{F1}$ , calculates; this operator is described in Section 11.4.1. If  $F_1 = 0$ , the vessel does not vibrate. Otherwise, if the vessel is freely suspended and we want to learn its acceleration amplitude, we simply divide by its mass  $M$ ; if we want its velocity amplitude, we divide by  $i\omega M$ ; its displacement amplitude, we divide by  $-\omega^2 M$ .

Equation (14.1) includes both the normal force exerted by gas pressure on the vessel and the shear force exerted by viscosity and gas velocity on the vessel. Newton's laws say that this must be true, but it can also be verified directly, as follows. Calculus shows that

$$\frac{\partial}{\partial t} (\rho\mathbf{v}) = \rho \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \frac{\partial \rho}{\partial t}. \quad (14.2)$$

Switching to "sum over repeated indices" notation (as in Ref. [36]), the continuity equation is

$$\frac{\partial \rho}{\partial t} = -\frac{\partial}{\partial x_k} (\rho v_k) \quad (14.3)$$

and the momentum equation is

$$\rho \frac{\partial v_i}{\partial t} = -\rho \left( v_k \frac{\partial v_i}{\partial x_k} \right) - \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_k} \sigma'_{ik}, \quad (14.4)$$

where  $\sigma'_{ik}$  is the viscous stress tensor. Substituting Eqs. (14.3) and (14.4) into (14.2) and doing a little algebra to combine terms yields

$$\frac{\partial}{\partial t} (\rho v_i) = -\frac{\partial}{\partial x_k} (\rho v_i v_k + p \delta_{ik} - \sigma'_{ik}). \quad (14.5)$$

Now integrate over the whole volume of gas, and use Green's formula to convert the volume integral of the divergence on the right side into a surface integral:

$$\frac{\partial}{\partial t} \int (\rho v_i) dV = - \oint (\rho v_i v_k + p \delta_{ik} - \sigma'_{ik}) dS_k, \quad (14.6)$$

where  $dV$  is a differential element of the volume and  $dS_k$  is a vector normal to an element of surface with area  $dS$ . The velocity is zero on the unmoving surface, so

$$\frac{\partial}{\partial t} \int (\rho v_i) dV = - \oint (p \delta_{ik} - \sigma'_{ik}) dS_k. \quad (14.7)$$

The right side is the total force exerted on the gas by the solid vessel bounding the gas. It is the sum of the pressure forces and the viscous forces. The  $x$  component of the left side is Eq. (14.1), which is what DELTAEC's F1 operator calculates.

### Third-order $p_1$ dependence in $dp_{2,0,HL}/dx$

Derivations of  $dp_{2,0,HL}/dx$  in publications of the Los Alamos thermoacoustics team through 2006, such as Refs. [42, 13], kept terms only through second order. In 2006, Scott Backhaus found one example (a stacked-screen regenerator with a small  $u_{2,0}$ ) in which a third-order term, proportional to  $|\rho_1| |u_1|^2$ , contributed significantly to  $dp_{2,0,HL}/dx$ . This led us to include that term in DELTAEC's STKSCREEN calculations, and for consistency we have included similar third-order contributions in other segments, including MINOR, DUCT, and CONE. This section outlines the derivations of these expressions for  $dp_{2,0,HL}/dx$ , which include these small terms proportional to  $p_1$ . Scott Backhaus's contributions to these derivations are gratefully acknowledged.

#### MINOR

The time-averaged pressure drop across a minor-loss component is, by definition, given by

$$\overline{\Delta p} = -\frac{1}{2} \overline{K \rho u |u|}, \quad (14.8)$$

where  $u$  is the spatially averaged velocity,  $\rho$  is the density,  $K$  is the minor-loss coefficient, and all three of these variables can be time dependent. In previous derivations, such as in Ref. [13], we have taken  $\rho$  to be independent of time, because we were satisfied with results to second order. Here we retain the fundamental component of  $\rho$ 's time dependence.

Write Eq. (14.8) as

$$\overline{\Delta p_{\text{minor}}} = -\frac{1}{2} \overline{K(t) [\rho_m + |\rho_1| \cos(\omega t + \phi)] (|u_1| \cos \omega t + u_{2,0}) ||u_1| \cos \omega t + u_{2,0}|} \quad (14.9)$$

$$= -\frac{1}{4\pi} \int_0^{2\pi} K(t) [\rho_m + |\rho_1| \cos(\omega t + \phi)] (|u_1| \cos \omega t + u_{2,0}) ||u_1| \cos \omega t + u_{2,0}| d(\omega t). \quad (14.10)$$

In the interval  $0 < \omega t < 2\pi$ , the argument of the absolute value changes sign at  $\omega t = \text{Arccos}(-u_{2,0}/|u_1|) \equiv \tau_0$  and  $\omega t = 2\pi - \text{Arccos}(-u_{2,0}/|u_1|) = 2\pi - \tau_0$  if  $|u_{2,0}/u_1| < 1$ ; at these zero crossings, the value of  $K$  changes from  $K_+$  to  $K_-$  and back again. Thus

$$\overline{\Delta p_{\text{minor}}} = -\frac{1}{4\pi} \left( K_+ \int_{-\tau_0}^{\tau_0} - K_- \int_{\tau_0}^{2\pi-\tau_0} \right) [\rho_m + |\rho_1| \cos(\omega t + \phi)] (|u_1| \cos \omega t + u_{2,0})^2 d(\omega t) \quad (14.11)$$

if  $u_{2,0}/|u_1| < 1$ . Evaluating this (e.g., with the symbolic-mathematics software MATHEMATICA) yields

$$\begin{aligned} \overline{\Delta p_{\text{minor}}} = & \frac{\rho_m |u_1|^2}{4\pi} \left\{ (K_- - K_+) \frac{\pi}{2} \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|\rho_1|}{\rho_m} \cos \phi \right) \right. \\ & - (K_+ + K_-) \left[ \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|\rho_1|}{\rho_m} \cos \phi \right) \sin^{-1} \varepsilon_{2,0} \right. \\ & \left. \left. + \left( 3\varepsilon_{2,0} + \frac{4}{3} [1 + \varepsilon_{2,0}^2/2] \frac{|\rho_1|}{\rho_m} \cos \phi \right) \sqrt{1 - \varepsilon_{2,0}^2} \right] \right\} \quad (14.12) \end{aligned}$$

if  $|\varepsilon_{2,0}| \leq 1$ , where  $\varepsilon_{2,0} = u_{2,0}/|u_1|$ .

If  $|\varepsilon_{2,0}| \geq 1$ , the integral is less complicated because the argument of the absolute value never crosses zero; the results are

$$\overline{\Delta p_{\text{minor}}} = -\frac{\rho_m |u_1|^2}{4} K_+ \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|\rho_1|}{\rho_m} \cos \phi \right) \text{ for } \varepsilon_{2,0} \geq 1, \quad (14.13)$$

$$= \frac{\rho_m |u_1|^2}{4} K_- \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|\rho_1|}{\rho_m} \cos \phi \right) \text{ for } \varepsilon_{2,0} \leq -1. \quad (14.14)$$

We take the density oscillations to be adiabatic, so  $\rho_1/\rho_m = p_1/\gamma p_m$ . These results appear in Section 10.2.4.

### STKSCREEN and SX

The momentum equation in a screen bed is [62, 13]

$$\frac{dp}{dx} = -\frac{c_1}{8r_h^2} \mu u - \frac{c_2}{2r_h} \rho u |u|, \quad (14.15)$$

where  $u$  is the spatially averaged velocity,  $\rho$  is the density,  $\mu$  is the viscosity, and  $r_h$  is the hydraulic radius. We take

$$u = |u_1| \cos \omega t + u_{2,0}, \quad (14.16)$$

$$\rho = \rho_m + |\rho_1| \cos(\omega t + \phi). \quad (14.17)$$

We might also write  $\mu = \mu_m + |\mu_1| \cos(\omega t + \theta)$ , but the oscillations in a regenerator are nearly isothermal and viscosity is independent of pressure, so we neglect  $\mu_1$  in the first term in Eq. (14.15).

The time average of the left-hand side, to second order, is  $dp_{2,0,HL}/dx$ .

The time average of the first term on the right-hand side, to second order, is

$$-\frac{c_1}{8r_h^2}\mu_m u_{2,0}. \quad (14.18)$$

Taking the time average of the second term involves the integral

$$I_{\text{screen}} = \overline{[\rho_m + |\rho_1| \cos(\omega t + \phi)] (|u_1| \cos \omega t + u_{2,0}) ||u_1| \cos \omega t + u_{2,0}|} \quad (14.19)$$

$$= \frac{1}{2\pi} \int_0^{2\pi} [\rho_m + |\rho_1| \cos(\omega t + \phi)] (|u_1| \cos \omega t + u_{2,0}) ||u_1| \cos \omega t + u_{2,0}| d(\omega t). \quad (14.20)$$

In the interval  $0 < \omega t < 2\pi$ , the argument of the absolute value near the end of the equation changes sign at  $\omega t = \text{Arccos}(-u_{2,0}/|u_1|) \equiv \tau_0$  and  $\omega t = 2\pi - \text{Arccos}(-u_{2,0}/|u_1|) = 2\pi - \tau_0$  if  $|u_{2,0}/u_1| < 1$ , so we can write

$$I_{\text{screen}} = \frac{1}{2\pi} \left( \int_{-\tau_0}^{\tau_0} - \int_{\tau_0}^{2\pi-\tau_0} \right) [\rho_m + |\rho_1| \cos(\omega t + \phi)] (|u_1| \cos \omega t + u_{2,0})^2 d(\omega t). \quad (14.21)$$

Evaluating this with the symbolic-mathematics software MATHEMATICA yields

$$I_{\text{screen}} = \frac{\rho_m |u_1|^2}{\pi} \left\{ \left[ (1 + 2\varepsilon_{2,0}^2) \sin^{-1} \varepsilon_{2,0} + 3\varepsilon_{2,0} \sqrt{1 - \varepsilon_{2,0}^2} \right] + 2 \frac{|\rho_1|}{\rho_m} \cos \phi \left[ \varepsilon_{2,0} \sin^{-1} \varepsilon_{2,0} + \frac{2 + \varepsilon_{2,0}^2}{3} \sqrt{1 - \varepsilon_{2,0}^2} \right] \right\}. \quad (14.22)$$

where  $\varepsilon_{2,0} = u_{2,0}/|u_1|$ .

Combining all of these intermediate results and taking  $\rho_1/\rho_m = p_1/p_m$  for approximately isothermal oscillations yields

$$\begin{aligned} \frac{dp_{2,0,HL}}{dx} = & -\frac{c_1}{8r_h^2}\mu_m u_{2,0} - \frac{c_2}{2\pi r_h}\rho_m |u_1|^2 \left\{ \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|p_1|}{p_m} \cos \phi \right) \sin^{-1} \varepsilon_{2,0} \right. \\ & \left. + \left[ 3\varepsilon_{2,0} + \frac{2}{3} (2 + \varepsilon_{2,0}^2) \frac{|p_1|}{p_m} \cos \phi \right] \sqrt{1 - \varepsilon_{2,0}^2} \right\} \end{aligned} \quad (14.23)$$

for  $|\varepsilon_{2,0}| \leq 1$ .

In **SX**,  $|\varepsilon_{2,0}| \geq 1$  may sometimes be of interest. In that case,

$$\frac{dp_{2,0,HL}}{dx} = -\frac{c_1}{8r_h^2}\mu_m u_{2,0} - \frac{c_2}{4r_h}\rho_m |u_1|^2 \text{sign}(\varepsilon_{2,0}) \left( 1 + 2\varepsilon_{2,0}^2 + 2\varepsilon_{2,0} \frac{|p_1|}{p_m} \cos \phi \right). \quad (14.24)$$

These results appear in Sections [10.5.2](#) and [10.7.2](#).

DUCT and CONE

We follow the outline in the appendix of Ref. [42], but we change the time dependence from sine to cosine, to be consistent with other results in this section, and we keep some dependence on  $\rho_1$  by writing Eq. (A6) in Ref. [42] as

$$\frac{dp}{dx} = -f_M(t) \frac{\rho_m + |\rho_1| \cos(\omega t + \phi)}{2D} u(t) |u(t)|, \quad (14.25)$$

where  $u = U/S$  in Ref. [42]. Using Eq. (A4) in Ref. [42] for  $f_M$  yields

$$\begin{aligned} \frac{dp_{2,0,HL}}{dx} &= -\frac{\rho_m |u_1|^2}{2D} \frac{1}{2\pi} \left( \int_{-\tau_0}^{\tau_0} - \int_{\tau_0}^{2\pi-\tau_0} \right) \left[ f_{M,\max} + \frac{df_M}{dN_R} N_{R,\max} \left( \frac{|\cos \omega t + \varepsilon_{2,0}|}{1 + |\varepsilon_{2,0}|} - 1 \right) \right] \\ &\quad \times \left[ 1 + \frac{|\rho_1|}{\rho_m} \cos(\omega t + \phi) \right] (\cos \omega t + \varepsilon_{2,0})^2 d(\omega t), \end{aligned} \quad (14.26)$$

where  $\varepsilon_{2,0} = u_{2,0}/|u_1|$  and  $\tau_0 = \text{Arccos}(-u_{2,0}/|u_1|)$  as for the **STKSCREEN** derivation above. We neglect  $|u_1|^2 N_{R,\max} |\rho_1|$  because it is fourth order. Then

$$\begin{aligned} \frac{dp_{2,0,HL}}{dx} &= -\frac{\rho_m |u_1|^2}{2D} \frac{f_{M,\max}}{2\pi} \left( \int_{-\tau_0}^{\tau_0} - \int_{\tau_0}^{2\pi-\tau_0} \right) \left[ 1 + \frac{|\rho_1|}{\rho_m} \cos(\omega t + \phi) \right] (\cos \omega t + \varepsilon_{2,0})^2 d(\omega t) \\ &\quad - \frac{\rho_m |u_1|^2}{2D} \frac{df_M}{dN_R} \frac{N_{R,\max}}{2\pi} \left( \int_{-\tau_0}^{\tau_0} - \int_{\tau_0}^{2\pi-\tau_0} \right) \left( \frac{|\cos \omega t + \varepsilon_{2,0}|}{1 + |\varepsilon_{2,0}|} - 1 \right) (\cos \omega t + \varepsilon_{2,0})^2 d(\omega t). \end{aligned} \quad (14.27)$$

The integral in the first line was evaluated above, in Eqs. (14.21)–(14.22). The integral in the second line was evaluated to obtain the bottom line in Eq. (A10) in Ref. [42]. Pulling all the pieces together, and using  $\rho_1/\rho_m = p_1/\gamma p_m$  for adiabatic oscillations, yields

$$\begin{aligned} \frac{dp_{2,0}}{dx} &= -\frac{\rho_m |u_1|^2}{2\pi D} \left\{ \left[ (1 + 2\varepsilon_{2,0}^2) \sin^{-1} \varepsilon_{2,0} + 3\varepsilon_{2,0} \sqrt{1 - \varepsilon_{2,0}^2} f_{M,\max} \right. \right. \\ &\quad \left. \left. + \frac{2|p_1|}{\gamma p_m} \cos \phi \left( \varepsilon_{2,0} \sin^{-1} \varepsilon_{2,0} + \frac{2 + \varepsilon_{2,0}^2}{3} \sqrt{1 - \varepsilon_{2,0}^2} \right) \right] \right. \\ &\quad \left. + \left[ \pi \varepsilon_{2,0} \frac{3/2 + \varepsilon_{2,0}^2}{1 + |\varepsilon_{2,0}|} - 3\varepsilon_{2,0} \sqrt{1 - \varepsilon_{2,0}^2} - (1 + 2\varepsilon_{2,0}^2) \sin^{-1} \varepsilon_{2,0} \right] \frac{df_M}{dN_R} N_{R,\max} \right\}, \end{aligned} \quad |\varepsilon_{2,0}| \leq 1, \quad (14.28)$$

$$\begin{aligned} \frac{dp_{2,0}}{dx} &= -\text{sign}(u_{2,0}) \frac{\rho_m u_{2,0}^2}{2D} \left[ \left( 1 + \frac{1}{2\varepsilon_{2,0}^2} + \frac{|p_1| \cos \phi}{\gamma p_m \varepsilon_{2,0}} \right) f_{M,\max} \right. \\ &\quad \left. - \frac{2\varepsilon_{2,0}^2 - 2|\varepsilon_{2,0}| + 1}{2\varepsilon_{2,0}^2(1 + |\varepsilon_{2,0}|)} \frac{df_M}{dN_R} N_{R,\max} \right], \quad |\varepsilon_{2,0}| \geq 1. \end{aligned} \quad (14.29)$$

These results appear in Section 10.1.1 for DUCT and 10.1.2 for CONE.

### STKPOWERLW and PX

The result for  $dp_{2,0,HL}/dx$  in STKPOWERLW and PX is the same as for DUCT and CONE above, except that we replace  $\gamma p_m$  with  $p_m$  to model isothermal oscillations.

### Pole-tip flux in linear motor or alternator, \*\*SPEAKER

The pole-tip magnetic-flux phasor is the sum of the flux phasor due to current and the flux phasor due to motion:

$$\Phi_{pt,1} = \Phi_{I,1} + \Phi_{bEMF,1} \quad (14.30)$$

where the subscript “1” indicates a complex phasor as usual. Obviously

$$\Phi_{I,1} = -LI_1. \quad (14.31)$$

We use “-” instead of “+” here because we always write  $|V| = |L dI/dt|$  and basic-physics books write  $V = -d\Phi/dt$  and  $\oint E \cdot dl = -L dI/dt$  with minus signs expressing Lenz’s law (induced EMF *opposes* changes in current). Similarly, we know that the unmeasurable internal voltage  $V_{bEMF,1} = -d\Phi_{bEMF,1}/dt = -i\omega \Phi_{bEMF,1}$ , and in the linear operating regime we express this as  $V_{bEMF,1} = -(Bl)v_1 = -(Bl)U_1/A$  in DELTAEC’s language of “Bl” product and velocity. (Note DELTAEC’s electromechanical sign convention, described in Sections 10.3.1 and 10.4.2, differs from some others’ conventions, e.g., Ref. [12], via  $V_1 \rightarrow -V_1$ ,  $I_1 \rightarrow -I_1$ , and  $Bl \rightarrow -Bl$ . Basically, this amounts to a rotation of all electrical and magnetic phasors by 180° relative to all mechanical and acoustic phasors.) Hence

$$\Phi_{bEMF,1} = -i(Bl)U_1/\omega A. \quad (14.32)$$

Combining these three equations gives us

$$\Phi_{pt,1} = -LI_1 - i(Bl)U_1/\omega A. \quad (14.33)$$

Using the law of cosines for triangles, we obtain the magnitude of the pole-tip flux

$$|\Phi_{pt,1}| = \sqrt{[L|I_1|]^2 + [(Bl)|U_1|/\omega A]^2 - 2[L|I_1|][(Bl)|U_1|/\omega A] \sin \alpha} \quad (14.34)$$

where  $\alpha$  is the phase angle by which  $U_1$  leads  $I_1$ . (Note:  $\cos(\alpha - 90^\circ) = \sin \alpha$ .)

(Thanks to John Corey for help with this section.)

### Example

Here’s a DELTAEC fragment giving  $|\Phi_{pt,1}|$  as the result of RPN segment 17, while segment 16 displays  $|\Phi_{I,1}| = L|I_1|$  and  $|\Phi_{bEMF,1}| = (Bl)|U_1|/\omega A$ .

```

!----- 15 -----
IESPEAKER the driver 7.5 diam
2.8500E-02 a Area m^2          9.7598E+04 A |p| Pa
0.5200    b R      ohms        154.96 B Ph(p) deg
3.2200E-02 c L      H           7.3587E-02 C |U| m^3/s
84.700    d BLProd T-m        -115.16 D Ph(U) deg
15.000    e M      kg           7.4617 E Hdot W
7.3300E+04 f K      N/m        7.4617 F Edot W
105.00    g Rm     N-s/m       6602.7 G WorkIn W
59.466    h |I|    A           654.43 H Volts V

```

```

93.911      i Ph(I) deg G      59.466 I Amps A
                                70.164 J Ph(V/I)deg
                                4.3376E+05 K |Px| Pa
                                174.36 L Ph(Px) deg
ideal      Solid type      -1273.9 M HeatIn W
!----- 16 -----
RPN A = IL flux [weber]; B = bEMF flux [weber]; C = alpha [degree]
0.0000 a Target (t)      1.9148 A weber
                                0.5801 B weber
                                150.93 C deg
15D 15i - 15d 15C * 2 / pi / 0b / 15a / 15c 15h *
!----- 17 -----
RPN peak pole-tip flux. [weber = tesla.m^2]
0.0000 a Target (t)      1.7099 A weber
16A sqrd 16B sqrd + 2 16A * 16B * 16C sin * - sqrt

```

Dividing by pole-tip effective area would turn 17A into the magnetic field, in tesla.

## Pulsed combustion

Suppose fuel is injected into an air stream in a duct, and combustion occurs. If the injection and combustion are steady, then conservation of matter applied to a small region around the combustion zone implies that

$$\dot{N}_{\text{out}} = \dot{N}_{\text{in}} + \Delta\dot{N}, \quad (14.35)$$

where  $\dot{N}_{\text{in}}$  and  $\dot{N}_{\text{out}}$  are the molar flow rates of the incoming fresh air and the outgoing combustion products, respectively, and  $\Delta\dot{N}$  is the additional molar flow rate due to the injection of the fuel and the change in mole numbers accompanying the combustion. Similarly,

$$T_{\text{out}} = T_{\text{in}} + \frac{\dot{Q}}{mc_p\dot{N}_{\text{out}}}, \quad (14.36)$$

where  $T_{\text{in}}$  and  $T_{\text{out}}$  are the temperatures of the incoming and outgoing streams, respectively,  $\dot{Q}$  is the rate at which the combustion releases heat, and  $m$  and  $c_p$  are the average molar mass and the heat capacity per unit mass of the outgoing stream.

In a thermoacoustic system, the combustion might be time dependent, because of the pressure dependence of the chemical reaction rate and/or a time dependence in the fuel injection. Suppose that the region of interest in the duct is small enough that the pressure is spatially uniform at every instant of time. Let the instantaneous heating rate be  $\dot{q}(t)$ , so the average heating rate must be

$$\dot{Q} = \frac{\omega}{2\pi} \int_0^{2\pi/\omega} \dot{q}(t) dt. \quad (14.37)$$

Similarly, let the instantaneous rate of increase of moles in the region of interest, due to fuel injection and combustion, be  $\dot{n}(t)$ , whose average must be

$$\Delta\dot{N} = \frac{\omega}{2\pi} \int_0^{2\pi/\omega} \dot{n}(t) dt. \quad (14.38)$$



Assuming ideal gases, the local rate of change of temperature due to  $\dot{q}(t)$  is

$$\dot{T}(t) = \frac{\dot{q}(t)}{V\rho_m c_p} = \frac{\gamma - 1}{\gamma} \frac{\dot{q}(t)}{Vp_m/T_m}, \quad (14.39)$$

where  $V$  is the volume in which the combustion occurs, and the rate of change of that volume due to this heating is

$$\dot{V}_q(t) = V \dot{T}(t)/T_m \quad (14.40)$$

$$= \frac{\gamma - 1}{\gamma} \frac{\dot{q}(t)}{p_m}. \quad (14.41)$$

Similarly,  $\dot{n}(t)$  contributes a time-dependent volume change

$$\dot{V}_n(t) = m \dot{n}(t)/\rho. \quad (14.42)$$

The fundamental Fourier component of  $\dot{V}_q + \dot{V}_n$  contributes to the acoustic power  $\dot{E}$  if the time phasing relative to  $p_1$  is favorable. In the DELTAEC paradigm, it causes a local increment in  $U_1$  which in turn causes a local increment in  $\dot{E}$  —or a decrement, depending on the time phasing. The real function  $\dot{V}_q(t) + \dot{V}_n(t)$  can be written as a Fourier series of sin's and cos's at  $\omega$ ,  $2\omega$ ,  $3\omega$ , etc. and in the DELTAEC paradigm we are only interested in the  $\omega$  components, which we can express in dimensionless form as  $\Lambda_1$  :

$$\lambda(t) = \frac{\dot{V}_q(t) + \dot{V}_n(t)}{(\gamma - 1)\dot{Q}/\gamma p_m}, \quad (14.43)$$

$$\Lambda_1 = \frac{1}{\pi} \int_0^{2\pi} \lambda(\omega t) e^{i\omega t} d(\omega t). \quad (14.44)$$

Depending on the circumstances, the phase of the complex variable  $\Lambda$  might be related to the local phase of either  $p_1$  or  $U_1$ .

Thus, the change in DELTAEC's complex volumetric-flow-rate amplitude across the region of interest is

$$\Delta U_1 = \frac{\gamma - 1}{\gamma} \frac{\dot{Q}}{p_m} \Lambda_1, \quad (14.45)$$

and the local increment in acoustic power at the fundamental frequency is

$$\Delta \dot{E} = \frac{\gamma - 1}{\gamma} \frac{\dot{Q}}{p_m} \frac{1}{2} \text{Re} \left[ p_1 \widetilde{\Lambda}_1 \right]. \quad (14.46)$$

For air,  $(\gamma - 1)/\gamma = 2/7$ , and we might hope for  $|p_1|/p_m \sim 0.2$ . If  $|\Lambda_1| \sim 1$ , then the efficiency of conversion of  $\dot{Q}$  to acoustic power could be roughly 3%. This is small, but not negligible, compared with normal thermoacoustic-engine efficiencies.

(This derivation, new in the 6.3 Users Guide, might benefit from critical analysis by users! Feedback will be appreciated.)

To implement this in DELTAEC via RPN segments, the user will have to know  $\dot{Q}$  and  $\Delta \dot{N}$ , make an estimate of the time dependence for  $\dot{q}(t)$  and  $\dot{n}(t)$  (including the time phase relative to  $p_1$  or  $U_1$ ), and do the math (or eyeball it graphically) to obtain a numerical value for  $\Lambda_1$ . Then put all those values into RPN segments, and use Eqs. (14.35), (14.36), and (14.45) above to impose local changes in DELTAEC's integration variables  $\dot{N}$ ,  $T_m$ , and  $U_1$ . Here is one way to implement it:

```

!----- 0 -----
BEGIN      at a pressure node, where fresh air is easily injected.
1.0000E+05 a Mean P Pa
  60.000   b Freq   Hz
  900.00   c TBeg   K
    0.0000 d |p|    Pa
    0.0000 e Ph(p)  deg
    2.0000 f |U|    m^3/s
    0.0000 g Ph(U)  deg
2.0000E-02 i Ndot  mol/s
air        Gas type
!----- 1 -----
DUCT      pipe from pressure node to somewhere more interesting
3.0000E-02 a Area   m^2      Mstr      1.5548E+04 A |p|    Pa
  0.6140   b Perim  m        1a       -90.098  B Ph(p)  deg
  2.5000   c Length m        2.0572E-02 C |U|    m^3/s
5.0000E-04 d Strough
ideal      Solid type
          -138.43 F Edot  W
!----- 2 -----
RPN      (a) is time-average combustion power (Watt)
! Verify (A) > 1, to ensure enough fresh air. The fuel is methane,
! which has 50 MJ/kg LHV and 0.016 kg/mole. We need 2 moles of O2,
! and so 10 moles of air, per mole of CH4.
1000.0    a G or T      1.6000          A Verify
Ndot inp 50e6 / 0.016 // 10 /
!----- 3 -----
BEGIN      Change from air to natural-gas combustion products.
sameas    Oa a Mean P Pa
sameas    Ob b Freq   Hz
sameas    Oc c TBeg   K
sameas    1A d |p|    Pa
sameas    1B e Ph(p)  deg
sameas    1C f |U|    m^3/s
sameas    1D g Ph(U)  deg
sameas    Oi i Ndot  mol/s
NGcbProd  Gas type
!----- 4 -----
RPN      (a) is methane's ratio of DeltaNdot to Qdot (mole/Joule)
! This is the value for methane, assuming H2O stays vapor:
! 3 moles combustion products / moleCH4.
! This segment increments Ndot appropriately.
3.7500E-06 a G or T      2.3750E-02          A mol/s
          3.7500E-03          B mol/s
2a inp * # Ndot + =Ndot
!----- 5 -----
RPN      guess Tm increment
1105.8    a G or T      G      2005.8          A K
Tm inp + =Tm
!----- 6 -----
RPN      target increment in Htot
sameas    2a a G or T      =6A      1000.0          A W
Htot 1E -
!----- 7 -----
RPN      (a) is |Lambda_d1|, assuming sinusoidal combustion and
! an extra 10% for Vdot sub n. Set phase of Lambda_d1 same as that of p1.
! (A) is the new U1. (B) is the increment in U1 due to pulsed combustion.
1.1000    a G or T      (-1.0278E-02, 1.4681E-02) A m3/s
          (-5.3554E-06, -3.1429E-03) B m3/s
gamma 1 - gamma / 2a * pm / p1 arg inp cmplx * # U1 + =U1
!----- 8 -----
RPN      (A) is the acoustic power created by pulsed combustion
0.0000    a G or T      24.433          A W
Edot 1F -
...

```

# 15. SYMBOLS

Variables not found here are intermediate factors, constants, or functions defined in the text close to where they are used.

## English letters

$A$	area, $\text{m}^2$
$a$	sound speed, $\text{m/s}$
$a$	half of rectangular pore's width, $\text{m}$
$B$	magnetic field, tesla
$B$	first virial coefficient, $\text{m}^3/\text{mol}$
$b$	a constant or function
$b$	half of rectangular pore's breadth, $\text{m}$
$C$	compliance, $\text{m}^3/\text{Pa}$
$C$	a constant or function
$c$	a constant or function
$c$	heat capacity per unit mass, $\text{J/kg}\cdot\text{K}$
$c_p$	isobaric heat capacity per unit mass, $\text{J/kg}\cdot\text{K}$
COP	coefficient of performance, $\dot{Q}/\dot{W}$ or $\dot{Q}/\dot{E}$
$D_{12}$	binary mass diffusion coefficient, $\text{m}^2/\text{s}$
$D$	diameter, $\text{m}$
$d$	diameter, $\text{m}$
$d$	differential
$\dot{E}$	acoustic power, $\text{W}$
$e$	2.71828...
$F$	force, $\text{N}$
$F$	a factor, fraction, friction factor, constant, or function
$\mathcal{F}$	a function
$f$	frequency, $\text{Hz}$
$f$	spatially averaged diffusion function
$f$	friction factor
$f_M$	Moody friction factor
$\dot{H}$	rate at which energy flows, $\text{W}$
$h$	heat transfer coefficient per unit area, $\text{W}/\text{m}^2\cdot\text{K}$

$I$	electric current, amp
$i$	$\sqrt{-1}$
$J$	Bessel function
$K$	spring constant, N
$K$	minor-loss coefficient
<b>ksFrac</b>	fudge factor by which $(1 - \phi)k_s$ is adjusted in <b>STKSCREEN</b>
$k$	thermal conductivity, W/m·K
$k_T$	thermal diffusion ratio
$k$	real or complex wave number, $m^{-1}$
$L$	$= k/\rho_m c_p D_{12}$
$L$	electric inductance, henry
$L$	length, m
$l$	length, m
$l$	half of plate thickness, m
$M$	mass, kg
$m$	molar mass, kg/mol (e.g., 0.004002 kg/mol for helium)
$m$	screen mesh number, wires/m
$m_\nu$	multiplier for turbulent effects
$m'_\nu$	turbulence multiplier for inertial pressure gradient
$N_{Nu}$	Nusselt number
$N_R$	Reynolds number
$N_{St}$	Stanton number, $= N_{Nu}/\sigma N_R$
$\dot{N}$	molar flow rate, mol/s (One mole = $6.022 \times 10^{23}$ molecules)
$n$	mole fraction
$p$	pressure, Pa
$Q$	heat, J
$\dot{Q}$	rate at which heat is transferred (thermal power), W
$\dot{q}$	rate at which heat is transferred per unit length, W/m
$R$	acoustic resistance, Pa·s/m <sup>3</sup>
$R$	electric resistance, ohm
$R$	mechanical resistance, kg/s
$R$	radius, m
$R_{univ}$	universal gas constant, J/mol·K
$r$	radius, m
$r$	radial coordinate, m
$S$	entropy, J/K
$S$	surface area, m <sup>2</sup>
$\dot{S}$	rate of entropy generation, J/K·s
$s$	entropy per unit mass, J/kg·K
$T$	temperature, K
$t$	time, s
$U$	volume flow rate, m <sup>3</sup> /s
$u$	$x$ component of velocity, m/s
$V$	volume, m <sup>3</sup>
$V$	voltage, volt

$v$	molar volume, $\text{m}^3/\text{mol}$
$\mathbf{v}$	vector velocity, $\text{m/s}$
$W$	work, J
$\dot{W}$	rate at which work is done (mechanical or electrical power), W
$w$	enthalpy per unit mass, $\text{J/kg}$
$X$	exergy, J
$\dot{X}$	rate at which exergy flows, W
$x$	coordinate along sound-propagation direction, m
$Y$	Neumann function
$y$	coordinate perpendicular to $x$ direction, m
$Z$	acoustic impedance, $\text{Pa}\cdot\text{s}/\text{m}^3$
$Z$	electric impedance, ohm
$z$	coordinate perpendicular to $x$ and $y$ , m
$z$	specific acoustic impedance, $\text{kg}/\text{m}^2\cdot\text{s} = \text{Pa}\cdot\text{s}/\text{m}$
$z_n$	$= z/\rho a$

### Greek letters

$\beta$	$= -(1/\rho)(\partial\rho/\partial T)_p$ , thermal expansion coefficient, $\text{K}^{-1}$
$\gamma$	ratio of isobaric to isochoric specific heats
$\Delta$	big difference
$\delta$	penetration depth, m
$\delta$	small difference
$\delta_{ik}$	the unit tensor, equal to unity for $i = k$ and zero for $i \neq k$
$\varepsilon$	surface roughness
$\varepsilon$	in mixtures, function defined by Eq. (10.196)
$\varepsilon_{20}$	$= m\dot{N}_{20}/\rho_m  U_1 $
$\varepsilon$	$= m\dot{N}/\rho_m  U_1 $
$\varepsilon_s$	correction factor for finite solid heat capacity
$\eta$	efficiency, $\dot{W}/\dot{Q}_H$ or $\dot{E}/\dot{Q}_H$
$\theta$	phase angle
$\kappa$	thermal diffusivity, $\text{m}^2/\text{s}$
$\mu$	dynamic viscosity, $\text{kg}/\text{m}\cdot\text{s}$
$\nu$	kinematic viscosity, $\text{m}^2/\text{s}$
$\xi$	displacement of gas in $x$ direction, m
$\Pi$	perimeter, m
$\pi$	3.14159...
$\rho$	density, $\text{kg}/\text{m}^3$
$\sigma$	Prandtl number
$\sigma_{ik}$	nine-component viscous stress tensor, Pa
$\tau$	transduction coefficient, $\text{N}/\text{amp} = \text{volt}/(\text{m/s})$
$\tau$	$\omega t$
$\phi$	volumetric porosity
$\omega$	$= 2\pi f$ , radian frequency, $\text{s}^{-1}$

## Subscripts

$bEMF$	back EMF
br	branch
$C$	cold
$e$	electric
eff	effective
ext	external
fric	friction
gen	generated
geom	geometrical
grad	gradient
$H$	hot
$H$	heavy component of mixture
$h$	hydraulic (i.e., $r_h$ is hydraulic radius)
$HL$	head loss
$i$	inner
imp	impedance
ini	initial
$k$	thermal
$L$	light component of mixture
$M$	Moody
mix sep	mixture separation
$m$	mean (or mechanical)
max	maximum
$\dot{N}$	steady flow
$Nu$	Nusselt (i.e., $N_{Nu}$ is Nusselt number)
$o$	outer
$p$	pressure
$\dot{Q}$	heat
$R$	Reynolds (i.e., $N_R$ is Reynolds number)
rad	radiation
rev	reversible
$s$	series
$s$	solid
sat	saturation
stand	standing
$T$	temperature
tot	total
trav	traveling
wat	water
$D\kappa$	coupled mass and thermal diffusion
$\kappa D$	coupled mass and thermal diffusion

$\kappa$	thermal
$\nu$	viscous
0	“environment” or “ambient” or independent of time
1	first order, usually a complex amplitude
2	second order
2, $k$	sum of second-order enthalpy and ordinary conduction
+	positive- $x$ direction
-	negative- $x$ direction
univ	universal (i.e., $R_{\text{univ}} \simeq 8.314 \text{ J/mol}\cdot\text{K}$ is the molar gas constant)

## Special symbols

Im[ ]	Imaginary part of
Re[ ]	Real part of,
$\langle \rangle$	spatial average perpendicular to $x$
$  $	magnitude of complex number
overbar	time average
overdot	time derivative, time rate
tilde	complex conjugate

# BIBLIOGRAPHY

- [1] T. J. Hoffer. *Thermoacoustic refrigerator design and performance*. PhD thesis, Physics department, University of California, San Diego, 1986.
- [2] T. J. Hoffer. Concepts for thermoacoustic refrigeration and a practical device. In Chairman Paul Lindquist, editor, *Proceedings of the 5th International Cryocoolers Conference*, pages 93–101, Wright-Patterson AFB, OH, August 1988. Wright-Patterson Air Force Base.
- [3] S. L. Garrett, D. K. Perkins, and A. Gopinath. Thermoacoustic refrigerator heat exchangers: Design, analysis, and fabrication. In G. F. Hewitt, editor, *Heat Transfer 1994: Proceedings of the 10th International Heat Transfer Conference. Vol. 4*, pages 375–380, Rugby, England, 1994. Institution of Chemical Engineers.
- [4] G. W. Swift. Thermoacoustic engines. *J. Acoust. Soc. Am.*, 84:1145–1180, 1988.
- [5] S. Backhaus and G. W. Swift. A thermoacoustic-Stirling heat engine: Detailed study. *J. Acoust. Soc. Am.*, 107:3148–3166, 2000.
- [6] R. Radebaugh. A review of pulse tube refrigeration. *Adv. Cryogenic Eng.*, 35:1191–1205, 1990.
- [7] R. Radebaugh. Development of the pulse tube refrigerator as an efficient and reliable cryocooler. *Proc. Institute of Refrigeration (London)*, pages 11–29, 2000. [http://cryogenics.nist.gov/Papers/Institute\\_of\\_Refrig.pdf](http://cryogenics.nist.gov/Papers/Institute_of_Refrig.pdf), last viewed March, 2016.
- [8] S. Backhaus, G. W. Swift, and R. S. Reid. High-temperature self-circulating thermoacoustic heat exchanger. *Appl. Phys. Lett.*, 87:014102, 2005.
- [9] G. W. Swift and S. Backhaus. Gas diodes for thermoacoustic self-circulating heat exchangers. In *CP838, Innovations in Nonlinear Acoustics: 17th International Symposium on Nonlinear Acoustics*, edited by A. A. Atchley, V. W. Sparrow, and R. M. Keolian, pages 399–406, Melville, New York, 2001. American Institute of Physics.
- [10] G. W. Swift and D. A. Geller. Continuous thermoacoustic mixture separation. *J. Acoust. Soc. Am.*, 120:2648–2657, 2006. Erratum *J. Acoust. Soc. Am.* 124:2421, 2008.
- [11] S. L. Garrett. *Understanding Acoustics: An Experimentalist’s View of Acoustics and Vibration*. ASA Press, 2017.
- [12] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders. *Fundamentals of Acoustics*. John Wiley and Sons, 4th edition, 1999.



- [13] G. W. Swift. *Thermoacoustics: A Unifying Perspective for Some Engines and Refrigerators*. First Edition: Acoustical Society of America Publications, Sewickley PA, 2002. Second Edition: ASA Press and Springer, 2017.
- [14] A. Migliori and G. W. Swift. Liquid sodium thermoacoustic engine. *Appl. Phys. Lett.*, 53:355–357, 1988.
- [15] M. Iguchi, M. Ohmi, and K. Maegawa. Analysis of free oscillating flow in a U-shaped tube. *Bull. JSME*, 25:1398–1405, 1982.
- [16] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing, Second Edition*. Cambridge University Press, 1992.
- [17] Reverse Polish notation. In Wikipedia. Retrieved on March 26, 2016, from [https://en.wikipedia.org/wiki/Reverse\\_Polish\\_notation](https://en.wikipedia.org/wiki/Reverse_Polish_notation).
- [18] B. Brown, 2001. Postfix Notation Mini-Lecture. Retrieved on 3/26/2016 from [http://bbrown.kennesaw.edu/web\\_lectures/postfix/](http://bbrown.kennesaw.edu/web_lectures/postfix/).
- [19] D. L. Gardner and G. W. Swift. A cascade thermoacoustic engine. *J. Acoust. Soc. Am.*, 114:1905–1919, 2003.
- [20] G. W. Swift. *Encyclopedia of Applied Physics*, volume 21, chapter Thermoacoustic engines and refrigerators, pages 245–264. Wiley, for American Institute of Physics, 1997.
- [21] N. Rott. Damped and thermally driven acoustic oscillations in wide and narrow tubes. *Z. Angew. Math. Phys.*, 20:230–243, 1969.
- [22] N. Rott. Thermally driven acoustic oscillations, part III: Second-order heat flux. *Z. Angew. Math. Phys.*, 26:43–49, 1975.
- [23] G. W. Swift. Analysis and performance of a large thermoacoustic engine. *J. Acoust. Soc. Am.*, 92:1551–1563, 1992.
- [24] P. Merkli and H. Thomann. Thermoacoustic effects in a resonance tube. *J. Fluid Mech.*, 70:161–177, 1975.
- [25] J. R. Olson and G. W. Swift. Similitude in thermoacoustics. *J. Acoust. Soc. Am.*, 95:1405–1412, 1994.
- [26] I. Urieli and D. M. Berchowitz. *Stirling Cycle Engine Analysis*. Adam Hilger, Bristol UK, 1984.
- [27] A. J. Organ. *Thermodynamics and Gas Dynamics of the Stirling Cycle Machine*. Cambridge University Press, 1992.
- [28] M. A. Lewis, T. Kuriyama, F. Kuriyama, and R. Radebaugh. Measurement of heat conduction through stacked screens. *Adv. Cryogenic Eng.*, 43:1611–1618, 1998.

- [29] J. R. Olson and G. W. Swift. Acoustic streaming in pulse tube refrigerators: Tapered pulse tubes. *Cryogenics*, 37:769–776, 1997.
- [30] J. R. Olson and G. W. Swift. Suppression of acoustic streaming in tapered pulse tubes. In R. G. Ross, Jr., editor, *Cryocoolers 10*, pages 307–313. Plenum, New York, 1999.
- [31] G. W. Swift, M. S. Allen, and J. J. Wollan. Performance of a tapered pulse tube. In R. G. Ross, Jr., editor, *Cryocoolers 10*, pages 315–320. Plenum, New York, 1999.
- [32] A. Bejan. *Advanced Engineering Thermodynamics*. Wiley, New York, 2nd edition, 1997.
- [33] P. M. Morse and K. U. Ingard. *Theoretical Acoustics*. McGraw-Hill, 1968.
- [34] W. L. M. Nyborg. Acoustic streaming. In W. P. Mason, editor, *Physical Acoustics, Volume IIB*, pages 265–331. Academic Press, 1965.
- [35] B. L. Smith and G. W. Swift. Measuring second-order time-average pressure. *J. Acoust. Soc. Am.*, 110:717–723, 2001.
- [36] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics*. Pergamon, 1982.
- [37] R. S. Reid, W. C. Ward, and G. W. Swift. Cyclic thermodynamics with open flow. *Phys. Rev. Lett.*, 80:4617–4620, 1998.
- [38] R. S. Reid and G. W. Swift. Experiments with a flow-through thermoacoustic refrigerator. *J. Acoust. Soc. Am.*, 108:2835–2842, 2000.
- [39] R. S. Reid. *Open cycle thermoacoustics*. PhD thesis, Georgia Institute of Technology, School of Mechanical Engineering, 1999.
- [40] S. Backhaus and G. W. Swift. A thermoacoustic-Stirling heat engine. *Nature*, 399:335–338, 1999.
- [41] David Gedeon. DC gas flows in Stirling and pulse-tube cryocoolers. In R. G. Ross, Jr., editor, *Cryocoolers 9*, pages 385–392. Plenum, New York, 1997.
- [42] G. W. Swift and S. Backhaus. A resonant, self-pumped, circulating thermoacoustic heat exchanger. *J. Acoust. Soc. Am.*, 116:2923–2938, 2004.
- [43] P. S. Spoor and G. W. Swift. Thermoacoustic separation of a He–Ar mixture. *Phys. Rev. Lett.*, 85:1646–1649, 2000.
- [44] D. A. Geller and G. W. Swift. Saturation of thermoacoustic mixture separation. *J. Acoust. Soc. Am.*, 111:1675–1684, 2002.
- [45] D. A. Geller and G. W. Swift. Thermoacoustic enrichment of the isotopes of neon. *J. Acoust. Soc. Am.*, 115:2059–2070, 2004.

- [46] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114. Gordon and Breach, London, 1970.
- [47] M. J. D. Powell. A FORTRAN subroutine for solving systems of nonlinear algebraic equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 115–161. Gordon and Breach, London, 1970.
- [48] S. M. Hino, M. Sawamoto, and S. Takasu. Experiments on transition to turbulence in an oscillatory pipe flow. *J. Fluid Mech.*, 75:193–207, 1976.
- [49] M. Ohmi, M. Iguchi, K. Kakehashi, and M. Tetsuya. Transition to turbulence and velocity distribution in an oscillating pipe flow. *Bull. JSME*, 25:365–371, 1982.
- [50] M. Ohmi and M. Iguchi. Critical Reynolds number in an oscillating pipe flow. *Bull. JSME*, 25:165–172, 1982.
- [51] M. Ohmi, M. Iguchi, and I. Urahata. Flow patterns and frictional losses in an oscillating pipe flow. *Bull. JSME*, 25:536–543, 1982.
- [52] I. E. Idelchik. *Handbook of Hydraulic Resistance*. Begell House, New York, 3rd edition, 1994.
- [53] J. H. So, G. W. Swift, and S. Backhaus. An internal streaming instability in regenerators. *J. Acoust. Soc. Am.*, 120:1898–1909, 2006.
- [54] J. R. Olson and G. W. Swift. A loaded thermoacoustic engine. *J. Acoust. Soc. Am.*, 98:2690–2693, 1995.
- [55] G. W. Swift and J. J. Wollan. Thermoacoustics for liquefaction of natural gas. *NETL GasTIPS*, 8(4):21–26, 2002. Also available at [www.lanl.gov/thermoacoustics/Pubs/GasTIPS.pdf](http://www.lanl.gov/thermoacoustics/Pubs/GasTIPS.pdf).
- [56] R. W. Fox and A. T. McDonald. *Introduction to Fluid Mechanics*. Wiley, 1985.
- [57] R. M. Olson. *Essentials of Engineering Fluid Mechanics*. Intext Educational Publishers, New York, 1973.
- [58] A. D. Pierce. *Acoustics: An Introduction to Its Physical Principles and Applications*. Acoustical Society of America, Woodbury, New York, 1989.
- [59] A. D. Pierce. *Acoustics: An Introduction to Its Physical Principles and Applications*. Acoustical Society of America, Woodbury, New York, 1989. Eqs. (5-4.8), (5-4.9), (5-4.11b), (5-4.13b).
- [60] W. P. Arnott, H. E. Bass, and R. Raspet. General formulation of thermoacoustics for stacks having arbitrarily shaped pore cross sections. *J. Acoust. Soc. Am.*, 90:3228–3237, 1991.

- [61] G. W. Swift and R. M. Keolian. Thermoacoustics in pin-array stacks. *J. Acoust. Soc. Am.*, 94:941–943, 1993.
- [62] G. W. Swift and W. C. Ward. Simple harmonic analysis of regenerators. *J. Thermophysics and Heat Transfer*, 10:652–662, 1996.
- [63] W. M. Kays and A. L. London. *Compact Heat Exchangers*. McGraw-Hill, New York, 1964.
- [64] P. J. Storch, R. Radebaugh, and J. E. Zimmerman. Analytical model for the refrigeration power of the orifice pulse tube refrigerator. Technical note 1343, National Institute of Standards and Technology, 1990.
- [65] N. T. Weiland and B. T. Zinn. Open cycle traveling wave thermoacoustics: Mean temperature difference at the regenerator interface. *J. Acoust. Soc. Am.*, 114:2791–2798, 2003.
- [66] J. L. Smith and M. Romm. Thermodynamic loss at component interfaces in Stirling cycles. In *Proceedings of the 27th Intersociety Energy Conversion Engineering Conference*, pages 5.529–5.532. Society of Automotive Engineers, 1992.
- [67] G. W. Swift and P. S. Spoor. Thermal diffusion and mixture separation in the acoustic boundary layer. *J. Acoust. Soc. Am.*, 106:1794–1800, 1999. Errata *J. Acoust. Soc. Am.* 107:2299, 2000; 109:1261, 2001.
- [68] D. A. Geller and G. W. Swift. Thermodynamic efficiency of thermoacoustic mixture separation. *J. Acoust. Soc. Am.*, 112:504–510, 2002.
- [69] D. A. Geller and G. W. Swift. Thermoacoustic mixture separation with an axial temperature gradient. *J. Acoust. Soc. Am.*, 125:2937–2945, 2009.
- [70] Y. S. Touloukian and C. Y. Ho. *Properties of Nonmetallic Fluid Elements*. McGraw-Hill, 1981.
- [71] Y. S. Touloukian, S. C. Saxena, and P. Hestermans. *Viscosity: Thermophysical Properties of Matter, Volume 11*. Plenum, 1975.
- [72] Y. S. Touloukian, P. E. Liley, and S. C. Saxena. *Thermal Conductivity: Nonmetallic Liquids and Gases: Thermophysical Properties of Matter, Volume 3*. Plenum, 1970.
- [73] R. A. Hiller and G. W. Swift. Condensation in a steady-flow thermoacoustic refrigerator. *J. Acoust. Soc. Am.*, 108:1521–1527, 2000.
- [74] Richard Raspet, William V. Slaton, Craig J. Hickey, and Robert A. Hiller. Theory of inert gas-condensing vapor thermoacoustics: propagation equation. *J. Acoust. Soc. Am.*, 112:1414–1422, 2002.

- [75] William V. Slaton, Richard Raspet, Craig J. Hickey, and Robert A. Hiller. Theory of inert gas-condensing vapor thermoacoustics: Transport equations. *J. Acoust. Soc. Am.*, 112:1423–1430, 2002.
- [76] R. Pritchard, J. J. Guy, and N. E. Connor. *Industrial Gas Utilization—Engineering Principles and Practice*. Bowker Publishing Company, Epping, Essex, 1977.
- [77] O. J. Foust, editor. *Sodium-NaK Engineering Handbook*. Gordon and Breach, New York, 1972.

# SEGMENT INDEX

## ANCHOR

- examples, [81](#)
- reference summary, [233](#)
- tutorial discussion, [82](#)
- with TBRANCH, [230](#)

## BEGIN

- examples, [65](#), [67](#), [70](#), [104](#), [118](#)
- reference summary, [223](#)
- twice in one model, [225](#)

## BLKDATA

- reference summary, [244](#)

## BRANCH

- examples, [172](#)
- reference summary, [171](#)

## COMPLIANCE

- examples, [63](#)
- reference summary, [159](#)

## CONE

- examples, [12](#), [63](#), [158](#)
- reference summary, [156](#)
- tutorial discussion
  - ignoring steady flow, [25](#)

## CONSTANTS

- reference summary, [243](#)

## DUCT

- examples, [12](#), [63](#), [81](#), [151](#)
- reference summary, [150](#)
- tutorial discussion
  - ignoring steady flow, [12](#), [25](#)

## HARDEND

- examples, [63](#), [81](#), [108](#)
- reference summary, [227](#)
- tutorial discussion
  - ignoring steady flow, [20](#)

- with steady flow, [108](#)

## HX

- examples, [63](#), [81](#), [104](#)
- reference summary, [190](#)

## IDUCER

- examples, [175](#)
- reference summary, [173](#)
- tutorial discussion, [26](#)

## IEDUCER

- examples, [169](#)
- reference summary, [167](#)
- tutorial discussion, [26](#)

## IESPEAKER

- as alternator, [56](#)
- examples, [91](#), [169](#)
- magnetic flux saturation, [266](#)
- reference summary, [167](#)
- with piston blowby , [56](#)

## IMPEDANCE

- reference summary, [161](#)

## INSULATE

- reference summary, [233](#)

## ISPEAKER

- examples, [175](#)
- magnetic flux saturation, [266](#)
- reference summary, [173](#)
- tutorial discussion, [26](#)

## JOIN

- examples, [95](#), [108](#)
- reference summary, [212](#)

## MINOR

- examples, [108](#), [115](#)
- reference summary, [163](#)
- tutorial discussion
  - with steady flow, [114](#)

MIXBL  
 reference summary, [214](#)  
 tutorial discussion, [125](#)

MIXCIRC  
 examples, [129](#)  
 reference summary, [214](#)  
 tutorial discussion, [125](#)

MIXSLAB  
 reference summary, [214](#)  
 tutorial discussion, [125](#)

MIXTBL  
 reference summary, [217](#)

MIXTCIRC  
 reference summary, [217](#)

MIXTSLAB  
 reference summary, [217](#)

OPNBRANCH  
 examples, [172](#)  
 reference summary, [171](#)

PISTBRANCH  
 examples, [172](#)  
 radiation impedance from flanged tube,  
[172](#)  
 reference summary, [171](#)

PX  
 reference summary, [193](#)

RPN  
 examples, [56](#), [88](#)  
 reference summary, [235](#)  
 tutorial discussion, [50](#)

SOFTEND  
 as open-tube radiation impedance, [228](#)  
 examples, [104](#), [108](#), [228](#)  
 reference summary, [227](#)  
 tutorial discussion  
 ignoring steady flow, [35](#)  
 with steady flow, [104](#), [108](#)

STKCIRC  
 examples, [81](#), [180](#)  
 reference summary, [177](#)

STKCONE  
 reference summary, [187](#)

STKDUCT  
 examples, [95](#)  
 reference summary, [187](#)

STKPIN  
 examples, [180](#)  
 reference summary, [177](#)

STKPOWERLW  
 reference summary, [183](#)

STKRECT  
 examples, [104](#), [180](#)  
 reference summary, [177](#)  
 tutorial discussion  
 with steady flow, [104](#)

STKSCREEN  
 examples, [88](#), [108](#)  
 reference summary, [183](#)  
 tutorial discussion  
 ignoring steady flow, [88](#)  
 with steady flow, [108](#)

STKSLAB  
 examples, [63](#), [180](#)  
 reference summary, [177](#)

SURFACE  
 examples, [63](#), [81](#)  
 reference summary, [160](#)

SX  
 examples, [88](#)  
 reference summary, [193](#)

SYSEXEC  
 examples, [246](#)  
 reference summary, [245](#)

TBRANCH  
 examples, [56](#), [91](#), [108](#), [118](#)  
 reference summary, [229](#)  
 tutorial discussion  
 ignoring steady flow, [35](#)  
 with steady flow, [108](#)

TITLE  
 reference summary, [223](#)

TX  
 reference summary, [190](#)

UNION  
 examples, [56](#), [91](#), [108](#), [118](#)

- reference summary, [231](#)
- tutorial discussion
  - ignoring steady flow, [35](#)
  - with steady flow, [108](#)

## VDUCER

- examples, [175](#)
- reference summary, [173](#)
- tutorial discussion, [26](#)

## VEDUCER

- examples, [169](#)
- reference summary, [167](#)
- tutorial discussion, [26](#)

## VESPEAKER

- as alternator, *see* IESPEAKER
- examples, [70](#), [169](#)
- magnetic flux saturation, [266](#)
- never use with  $R = \omega L = 0$ , [167](#)
- reference summary, [167](#)

## VOLUME

- reference summary, [242](#)

## VSPEAKER

- examples, [67](#), [175](#)
- magnetic flux saturation, [266](#)
- never use with  $R = \omega L = 0$ , [167](#)
- reference summary, [173](#)
- tutorial discussion, [26](#)

## VXQ1

- examples, [114](#)
- reference summary, [199](#)
- tutorial discussion
  - with steady flow, [114](#)

## VXQ2

- reference summary, [199](#)

## VXT1

- reference summary, [199](#)

## VXT2

- reference summary, [199](#)



# SUBJECT INDEX

- Autolayout, [87](#)
- Black-body radiation, [145](#)
- Branches, structured, [35](#), [229](#)
- Bugs, reporting, [7](#), [140](#)
- Colors, meanings of, [23](#)
- Combustion, pulsed, [267](#)
- Comment lines, [87](#)
  - example, [88](#)
- Continuity equation, [10](#), [61](#), [134](#)
- Convergence problems
  - useful strategies, [78](#), [80](#)
- Convergence tolerance, [139](#)
- Copyright, [8](#)
- Debug alerts, [140](#)
- <default.ini>, [138](#)
- <DeltaEC.exe.log>, [140](#)
- Energy equation
  - axial, [62](#), [134](#)
  - lateral, [103](#), [134](#)
- Energy notation, [136](#)
- Error messages, [140](#)
  - reporting, [140](#)
- Examples
  - alternator load impedance, [56](#)
  - electrical alternator, [56](#)
  - force phasor, [240](#)
  - gas diode, [108](#), [115](#)
  - gas, changing mid-model, [225](#)
  - incremental plots, [44](#), [63](#), [81](#)
  - looped model, [36](#), [56](#), [92](#), [108](#), [115](#)
  - mixture separation, [129](#)
  - piston blowby, [56](#)
  - pulse-tube refrigerator, [95](#)
  - pulsed combustion, [267](#)
  - radiation impedance, [172](#), [228](#)
  - self-circulating heat exchanger, [114](#)
  - standing-wave engine, [81](#)
  - standing-wave refrigerator, [63](#), [104](#)
  - state plots, [17](#), [63](#), [81](#), [88](#), [95](#), [104](#), [108](#), [114](#), [129](#)
  - Stirling engine, [108](#)
  - Stirling refrigerator, [88](#)
  - structured branches, [35](#), [108](#), [114](#), [229](#)
  - traveling-wave engine, [108](#)
  - traveling-wave refrigerator, [88](#)
  - with nonzero  $\dot{N}$ , [104](#), [108](#), [114](#), [129](#)
- Exergy, [99](#), [103](#)
- File types
  - <.blk>, [244](#)
  - <.err>, [140](#)
  - <.ini>, [138](#)
  - <.ip>, [44](#)
  - <.isv>, [40](#)
  - <.log>, [140](#)
  - <.out>, [12](#)
  - <.sp>, [40](#)
  - <.ssv>, [40](#)
  - <.tpf>, [28](#), [253](#), [257](#)
  - <.tpm>, [254](#)
- Flipping segments, [87](#)
- Font, changing size, [17](#)
- Force-phasor calculation, [239](#), [261](#)
- Forward reference condition
  - avoiding, [77](#)
- FRC, *see* Forward reference condition
- Gas diode, examples, [108](#), [115](#), [165](#)
- Gases, [27](#), [248](#)
  - including liquids, [27](#)
  - mid-model change of, [225](#), [248](#)

- temperature floor, 139
- user-defined, 28, 253, 254
- Graphs, *see* State plots, Incremental plots,
  - Phasor plots, Schematic view
- Guesses and targets
  - convergence difficulty, 78, 80
  - examples, 19, 71, 90, 92, 119
  - general guidance, 19, 71, 77, 136
  - in RPN, 54, 58
  - maximum number, 136
  - meeting targets, 78, 80
  - mixture separation, 126
  - non-standard guesses, 58
  - non-standard targets, 54, 239
  - phase target near  $\pm 180^\circ$ , 143
  - restoring, 86
  - rewinding incremental plots, 48
  - shooting-method parameters, 139
  - troubleshooting, 78, 80
  - with nonzero  $\dot{N}$ , 108, 111
- Harmonic content, 141
- Head-loss equation, 103, 134
- Heat exchange, external  $\Delta T$ , 144
- Highlights display, 34
- Hydraulic radius of screens, 89, 147, 183
- Incremental plots, 44
  - cloning, 41
  - examples, 67, 84
  - multiple models, 58
  - rewinding, 48
- Inherent limitations, 141
- Installation, 7
- Integration variables, 134
  - acoustics only, 11
  - thermoacoustics, 63
  - with mixture separation, 126
  - with steady flow, 103
- ksFrac
  - including vessel-wall conduction, 145
  - recommended value, 145
- Legs, 42, 229
- License, 8
- Limitations, 141, 145, 146
- Linking multiple models, 58, 235
- List linkages, 53
- Looped models, examples, 36, 56, 92, 108, 115
- Magnetic flux saturation, 266
- Master-slave links, 30, 52
- Messages, error and warning, 140
- Mixture separation, typical end conditions, 126
- Mixture-separation equation, 126, 134
- Mode, *see* Normalization mode
- Momentum equation, 10, 61, 134
- Multi-model plots, 58
- Multiple models, *see* Linking multiple models
- Nint, 139
- Nonlinear effects, 6
- Normalization mode, 139
- Notation, 7, 270
  - energy, 136
- Notes, user-specified, 87
- Numerical integration method, 136
- Numerical options
  - integration steps, 139
  - shooting method, 139
  - temperature floor, 139
- Open tube, radiation impedance, 171, 228
- Options, numerical, 139
- Overview of DELTAEC, 1, 10, 61, 134
- Phases near  $\pm 180^\circ$ , 143
- Phasor plots, 33, 42, 97
- Plots, *see* State plots, Incremental plots,
  - Phasor plots, Schematic view
- Power notation, 136
- Pressure, second-order, 103
  - examples, *see* Looped models
  - inherent limitations, 142
  - subtleties in calculation, 136
- Pulsed combustion, 267
- Radiation impedance

- open end of flanged tube, [172](#)
- open end of unflanged tube, [228](#)
- Rayleigh streaming, [147](#)
- Reporting bugs, [7](#), [140](#)
- Restoring guesses, [86](#)
- Rewinding incremental plots, [48](#)
- Run monitor, [15](#)
  
- sameas, [29](#)
- Schematic view, [33](#)
- Screens, hydraulic-radius and porosity calculation, [89](#), [183](#)
- Shooting method, *see* Guesses and targets
- Size of font, [17](#)
- Solids, [27](#), [256](#)
  - temperature floor, [139](#)
  - user-defined, [257](#)
- Splitting a segment, [87](#)
- Standing-wave engine headroom, [148](#)
- State plots, [17](#), [40](#)
  - cloning, [41](#)
  - examples, [71](#), [82](#), [106](#)
  - phasor, [42](#), [97](#)
  - with legs, [42](#)
- Steady flow, [102](#), [136](#)
  - and user-defined gases, [254](#)
- Structured branches, [35](#), [229](#)
- Symbols, [270](#)
  
- Targets, *see* Guesses and targets
- Temperature floor, [139](#)
- Thermophysical properties
  - formulas for each gas, [248](#)
  - formulas for each solid, [256](#)
  - from the keyboard, [39](#)
  - in RPN, [52](#), [235](#)
- Time phase, [27](#), [69](#), [74](#)
  - examples, [67](#), [70](#)
  - in side-branch transducers, [173](#)
- Tolerance, convergence, [139](#)
- Toroidal models, *see* Looped models
- Total Power, typical end condition, [108](#), [111](#)
- Turbulence, [29](#), [141](#)
  - in CONE, [29](#), [142](#), [146](#), [156](#)
  - in DUCT, [29](#), [142](#), [146](#), [150](#), [152](#)
  - in MINOR, [147](#), [163](#)
  - in PX, [196](#)
  - in STKPOWERLW, [142](#), [187](#)
  - in STKSCREEN, [142](#), [187](#)
  - in SX, [196](#)
  - in VX\*\*, [142](#), [204](#)
  
- Users Guide, [6](#)
  
- Variables, [270](#)
- Visually impaired users, features for, [17](#)
  
- Warning messages, *see* Error messages