No-Cost and Low-Cost Methods of Reducing Floating Point Error in Sums

Vanessa Job

HPC-DES/NMC

Laura Monroe, mentor

HPC-DES

August 12, 2020

Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

LA-UR-20-26095L

Goals for the talk

- Floating point sums are prone to rounding error
- Consider how you are doing your sums
  - And how much error there is likely to be
- We will provide heuristics to increase sum accuracy with low or no cost

Motivation and Background

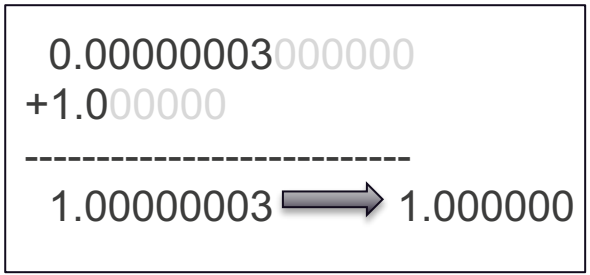
A little background

- IEEE 754 Standard for Floating-Point Arithmetic:
  - Used on most common platforms
  - Addition is commutative
  - Floats (single precision) have around **seven** decimal digits of precision
  - Doubles have around **fifteen** decimal digits of precision

- We target sums with a modest number of terms that are computed with few lines of code

Problem: Rounding error on a finite precision machine

- In general, finite precision arithmetic is not associative.
- This can lead to rounding error in summation!




```
float a = 0.00000003;
float b = 1.0;
float f = (a+b) -b;
printf("%.8lf\n", f);
float e = a+ (b-b);
printf("%.8lf\n", e);
```

Output:

```
0.00000000
0.00000003
```

- This problem is well-known, but sometimes disregarded

N. J. Higham. The accuracy of floating point summation. j-SISC, 14(4):783–799, July 1993.

Problem implications: Error that persists into the final results!

- This rounding error can be significant and can propagate

- This can cause significant error in the final result
- So it must be addressed

- Talk goals
  - Consider your summations, and how error-prone they might be
  - Low-overhead heuristics that mitigate error

Motivation: How should we add?

- You could assume it's a non-issue
- You could add variables in the mathematically common order, e.g. $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}$
- You could guess
- You could sort the variables before summing

Which method do **you** use?

Experiments

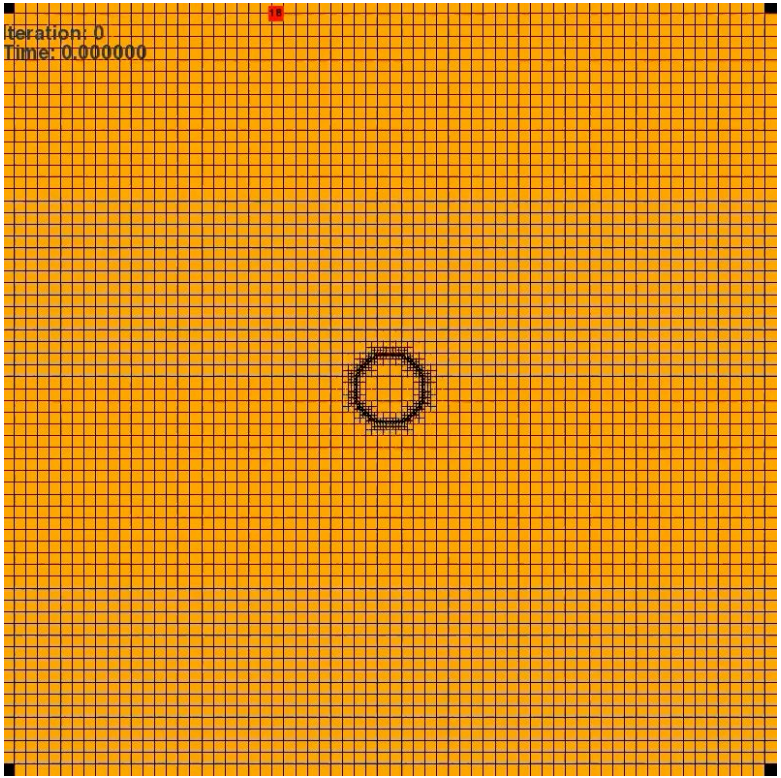
Experiments: Mini-apps

- Experimentation on two mini-apps (CLAMR and the Oregonator)
- We confirm that some orderings are much worse than others
- And developed some heuristics
  - These are based upon groupings and orderings that give good results
  - They have low runtime overhead

Vanessa Job, Terry Grové, Shane Fogerty, Chris Mauney, Brett Neuman, Laura Monroe, and Bob Robey. “Order matters: a case study on reducing floating point errors in sums through ordering and grouping”

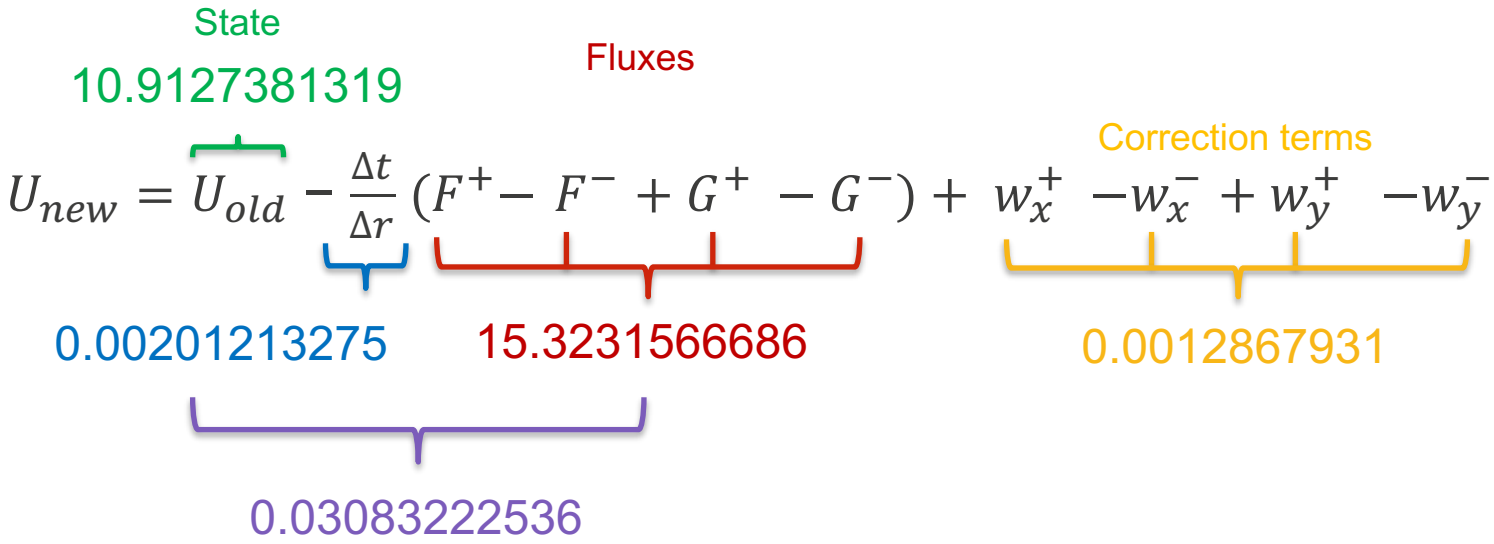
Experiment Setup: Case study CLAMR

- CLAMR is a cell-base Adaptive Mesh Refinement (AMR) mini-app
- We looked at one equation to see how changing the ordering and grouping affects the error
- Runs 56 million times



D. Nicholaeff, N. Davis, D. Trujillo, and R. W. Robey. Cell-based adaptive mesh refinement implemented with general purpose graphics processing units.
Technical Report LA-UR-11-07127, Los Alamos National Laboratory, 2012.

Experiment: State equation



Experiment Setup: Screening method

- U_{gold} is original ordering/grouping of state equation in double precision:

$$U_{gold} = U_{old} - \frac{\Delta t}{\Delta r}(F^{+} - F^{-} + G^{+} - G^{-}) + w_x^{+} - w_x^{-} + w_y^{+} - w_y^{-}$$

- U_{test} is test ordering/grouping of state equation in single precision, e.g.

$$U_{test} = (-\frac{\Delta t}{\Delta r}((F^{+}+G^{+}) + (-F^{-} - G^{-})) + (w_x^{+} + ((-w_x^{-} + w_y^{+}) - w_y^{-}))) + U_{old}$$

- For each time step, compute the relative difference (the error in double vs. single): $\left| \frac{U_{gold} - U_{test}}{U_{gold}} \right| \geq 1 * 10^{-7}$
- The relative error rate for the state equation in double vs. single is 2.109%

Experiment: Orderings and groupings

- There are a lot of these: >= 2 million
  - All of these look different to the compiler
  - All of these may have a different computational result
  - L. Monroe and V. Job. Computationally inequivalent summations and their parenthetic forms. http://arxiv.org/abs/2005.05387, 2020

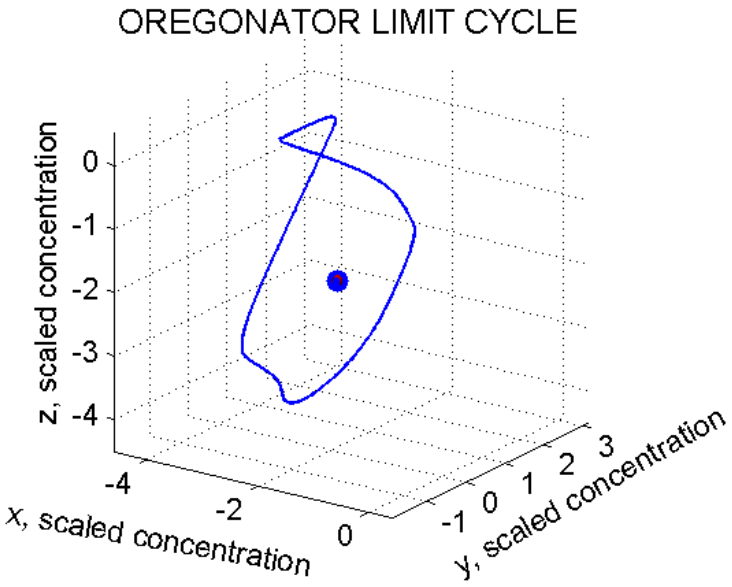
- So we sampled, rather than test all
  - Grouping by mean magnitude of the variable
  - Grouping pairwise
  - Random sample of 10000 of all orderings and groupings
- Then assessed error rate and symmetry

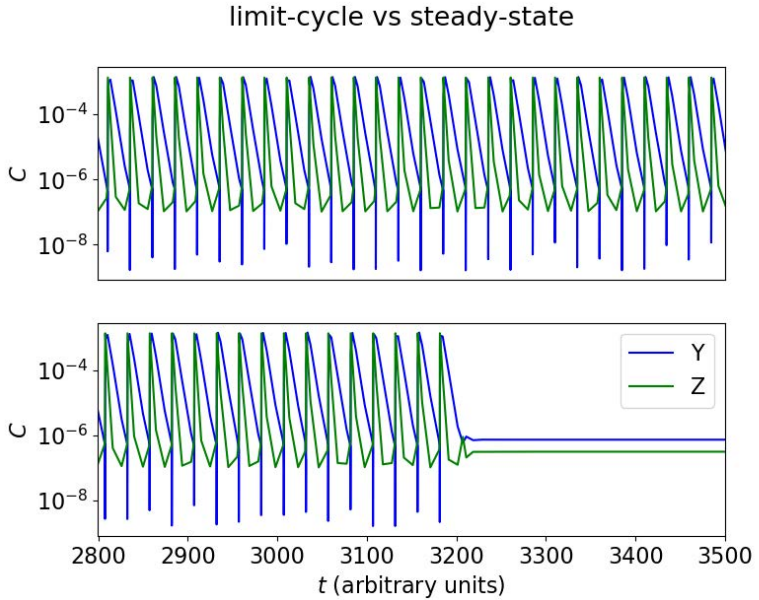
Experiment: Other summation methods

- Also tested traditional methods
  - Kahan (a known accurate method)
  - Sorting
  - Several others
  - More computationally expensive

Experiment: Second case study, the Oregonator

- Models a complex chemistry with oscillating components at equilibrium
- After developing heuristics on CLAMR, we tested on this application
  - The heuristics were confirmed



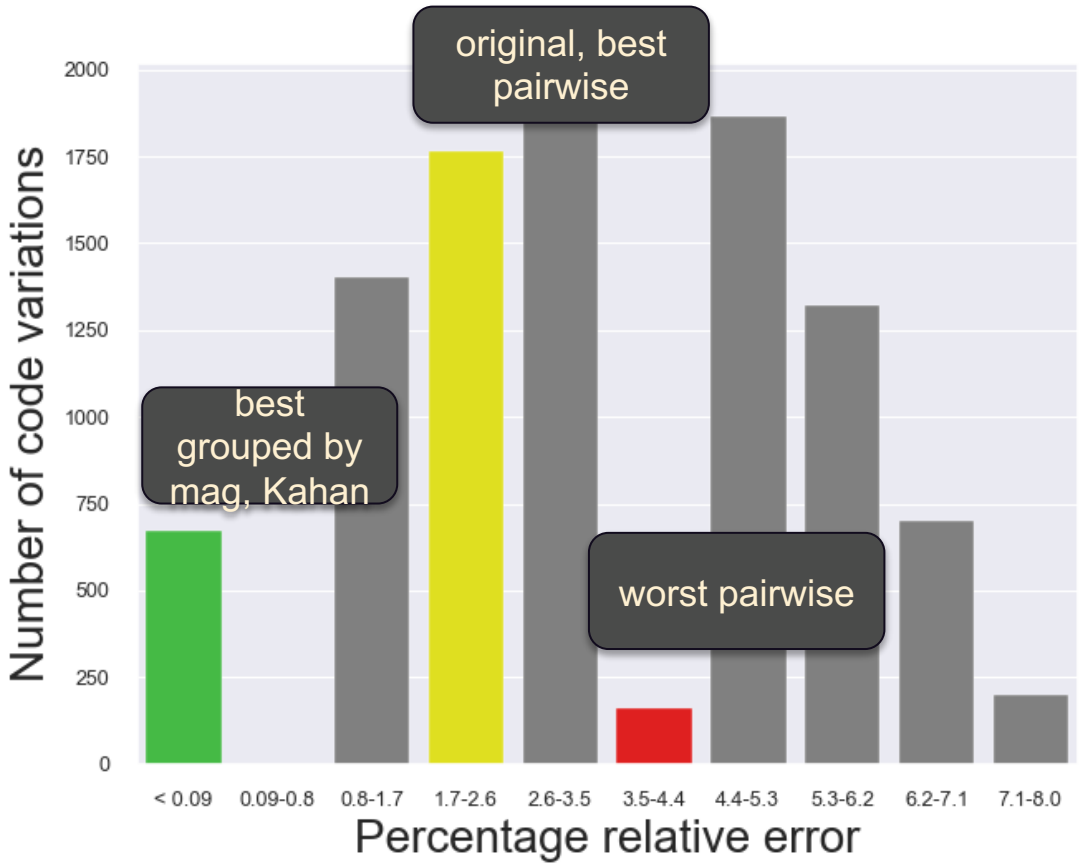


Results

Results for CLAMR

Method	Relative Error Rate
Best ordering/grouping (group by mean magnitude)	0.082 %
Best random sample of 10000 ordering and grouping, all nine	0.083 %
Kahan	0.083 %
PSum	0.084 %
Ascending	0.084 %
Best of group by mean mag, with U_{old} added first	1.015%
Original parenthesization	2.109 %
Best of pairwise	2.109 %
Worst of pairwise	3.517 %
Worst of randomly selected	7.909%

Results: ~8% of randomly chosen orderings and groupings were good



Tool under construction!

Results: Best of grouped by magnitude from random selection

$$U_{group\ by\ mag} = U_{old} + \left[\frac{\Delta t}{\Delta r}\left((-F^{+} + F^{-}) + (-G^{+} + G^{-})\right) + \left(\left(\left(w_x^{+} + w_y^{+}\right) - w_x^{-}\right) - w_y^{-}\right)\right]$$

Error rate 0.082%

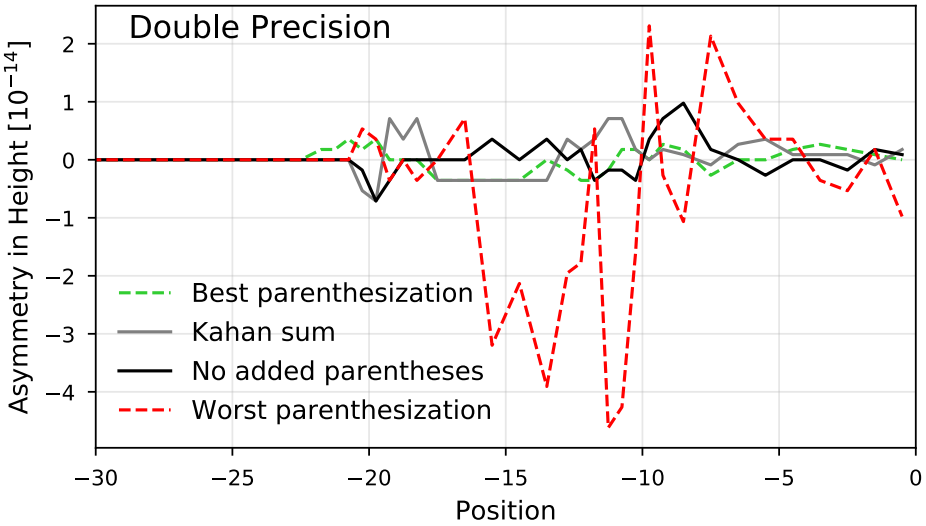
$$U_{rand\ from\ 9} = U_{old} + \left[\left((-f^{+} + f^{-}) + (-g^{+} + g^{-})\right) + \left(\left(\left(w_x^{+} + w_y^{+}\right) - w_x^{-}\right) - w_y^{-}\right)\right]$$

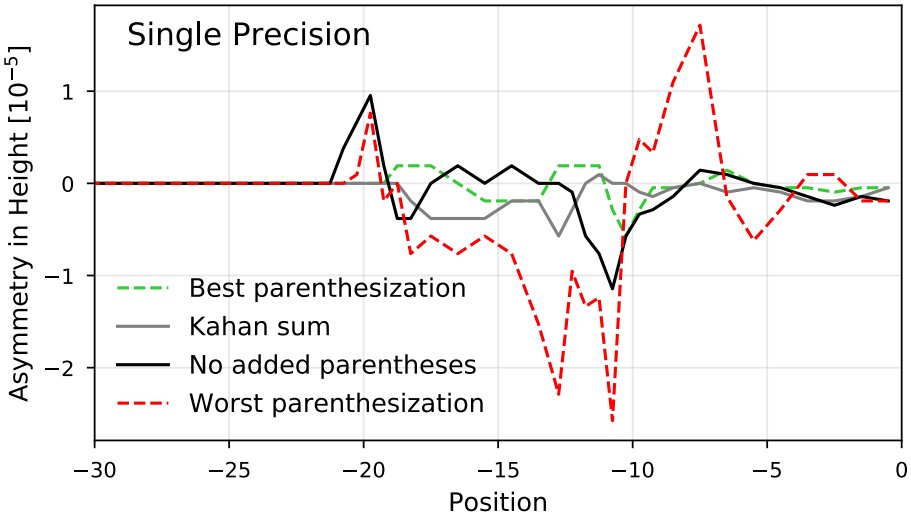
Error rate 0.083%

Note: F's grouped together and G's grouped together in all three. They are numerators of derivatives

Analysis

Analysis: Increased accuracy improved symmetry





Recommendations

Recommendations: Heuristics

- Always parenthesize sums completely
- Group variables of the same magnitude together
- Add groups smallest to largest
- Group pairs of terms that represent the numerator of a numerical derivative together
- Add pairwise within groups

Recommendations: Blueprint for investigation

- Target a sum from the code whose accuracy is critical
- Determine a benchmark for accuracy for the sum
  - Relative error to identify candidates could be a gating criterion
  - Symmetry of the simulation in our study
- Can the application can trade more time for more accuracy?
  - If so, use a method like Kahan or FastAccSum
- Is exhaustive testing infeasible?
  - Form groups according to the heuristics
  - Test a random sample
- Test the best candidates against the benchmark

Conclusions

Takeaways

- Naively ordered sums can be error prone
  - Don’t leave ordering up to the compiler
- Informed ordering and grouping can provide additional accuracy without any performance cost
  - Can rival gold standard methods such as Kahan
  - Profiling or domain expertise can be used
- Certain important cases can be error prone, need attention
  - Parallelization of sums
  - Reduced precision sums

Thank you

ASC BML Inexact Computing Project

Laura Monroe, Project Leader
lmonroe@lanl.gov

Andy Dubois
ajd@lanl.gov

Shane Fogerty
sfogerty@lanl.gov

Terry Grové
tagrove@lanl.gov

Vanessa Job
vjob@lanl.gov

Chris Mauney
mauneyc@lanl.gov

Brett Neuman
bneuman@lanl.gov

Bob Robey
brobey@lanl.gov

Supplemental Slides

Experiments: Ordering and Grouping

Type of equation	Distinct Computationally Inequivalent Examples
Ordered and grouped clumps, with permuted clumps	675
Distributed, added pairwise	11,340
Distributed, order and group all nine terms	2,027,025

L. Monroe and V. Job. Computationally inequivalent summations and their parenthetic forms. http://arxiv.org/abs/2005.05387, 2020.

Recommendations: Blueprint for investigation

- Target a sum from the code whose accuracy is critical
- Determine a benchmark for accuracy for the sum
  - Symmetry of the simulation in our study
- Decide whether the application can trade more time for more accuracy
  - If so, use a method like Kahan or FastAccSum
- Decide whether to use a screening tool to identify candidates

Recommendations: Blueprint for investigation

- Form groups according to the heuristics if exhaustive testing is not practical
- Test a random sample if exhaustive testing is not practical
- Test the best candidates against the benchmark

Recommendations: Testing our heuristics on Oregonator

- The Oregonator models a complex chemistry of a system with oscillating components at equilibrium
- Contains two differential equations

$$\frac{dX}{dt} = k_1 AY - k_2 XY + k_3 AX - 2k_4 X^2$$

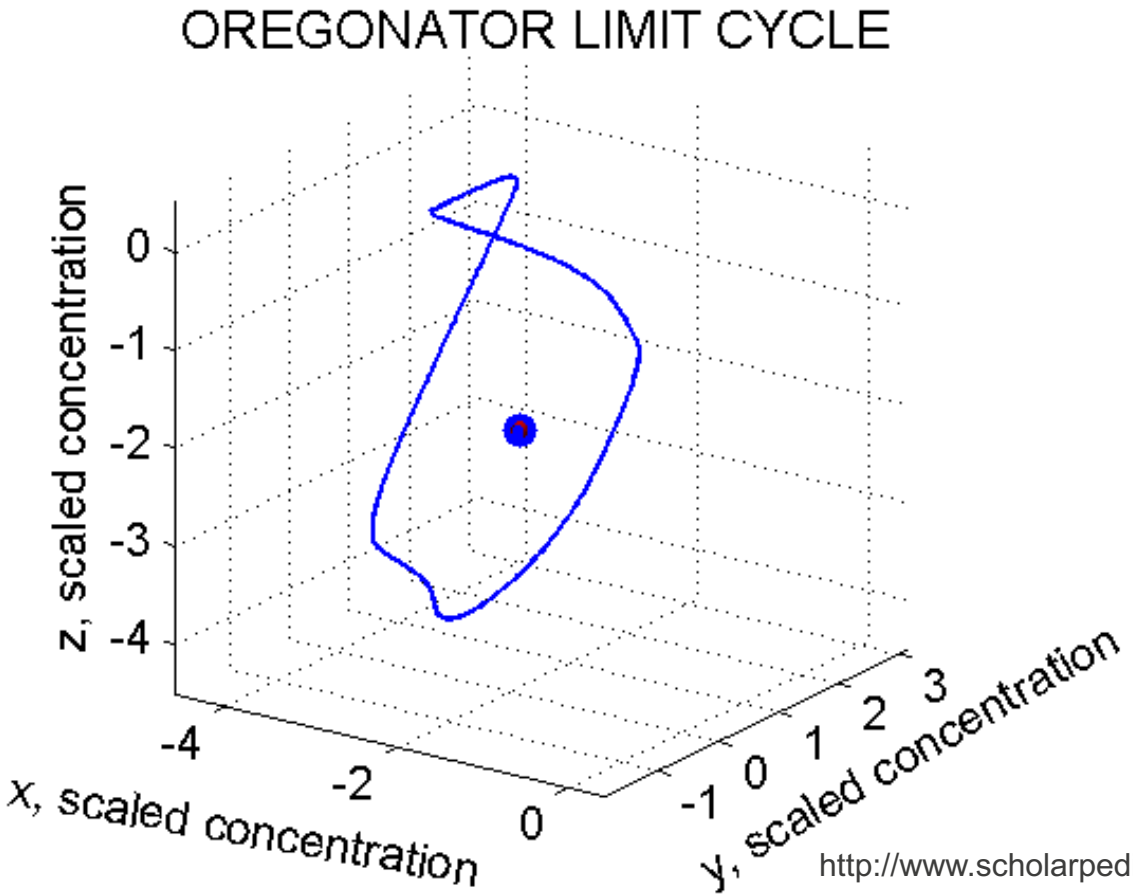
$$\frac{dY}{dt} = -k_2 AY - k_2 XY + k_5 fZ$$

CONSTANTS AND PARAMETERS USED IN INTEGRATING (10)

$k_1 A$	k_2	$k_3 A$	k_4	k_5
0.02	2×10^9	10^7	5×10^7	1.0
t_0	t_1	ϵ_A	ϵ_R	$\vec{X}_0$
0.0	10^5	10^{-8}	10^{-8}	$(10^3, 1.0, 1.0)$

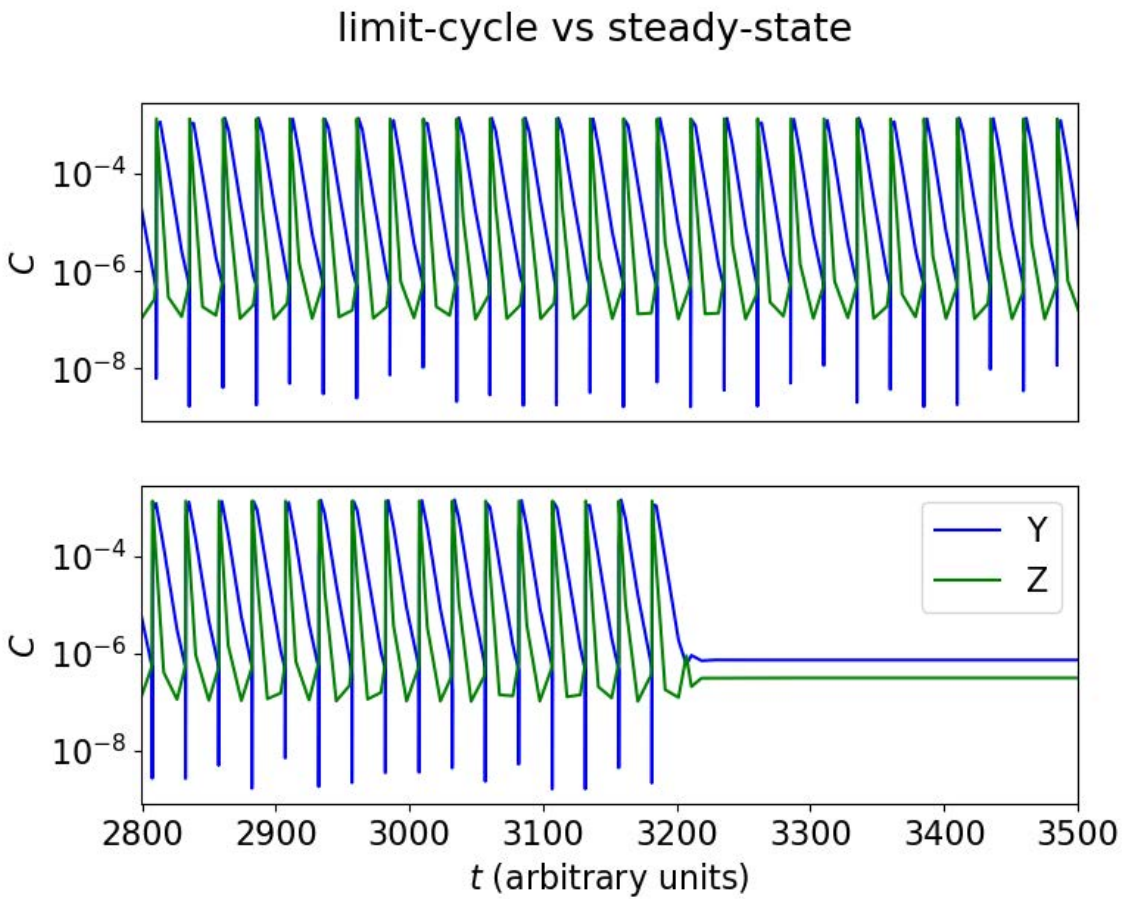
There would be $(15)(3) = 45$ combos to check

Recommendations: Testing our heuristics on Oregonator



http://www.scholarpedia.org/article/Oregonator

Recommendations: Testing our heuristics on Oregonator



Recommendations: Testing our heuristics on Oregonator

10^{-2} Y - $10^{7}X^{2}$

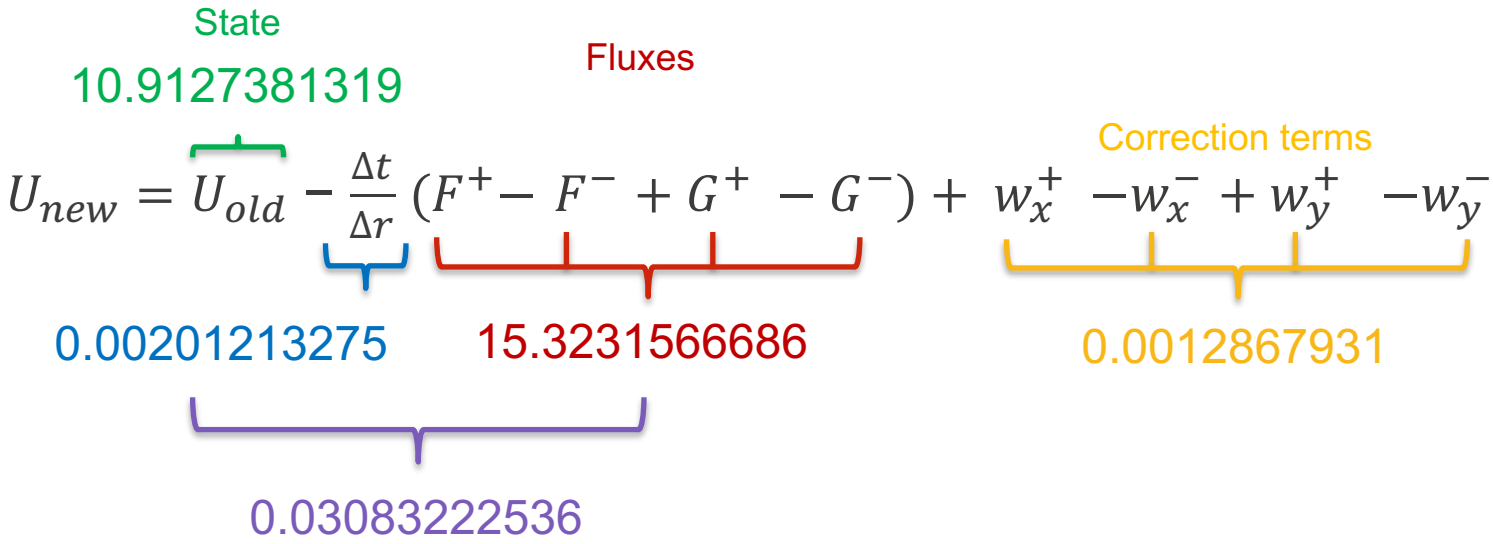
$$\frac{dX}{dt} = (((k_1 \text{AY} - 2k_4X^2) + k_3 \text{AX}) - k_2XY)$$

Fails due to large difference in magnitude between first two terms

$$\frac{dX}{dt} = (-2k_4X^2 + (k_1 \text{AY} + (k_3 \text{AX} - k_2XY)))$$

Success – grouped terms of comparable magnitude together.
This supports our heuristic

Experiment Setup: State equation



Experiment Setup: Screening method

- Let U_{gold} be the original ordering and grouping of the state equation in double precision:

$$U_{gold} = U_{old} - \frac{\Delta t}{\Delta r}\left(F^{+} - F^{-} + G^{+} - G^{-}\right) + w_x^{+} - w_x^{-} + w_y^{+} - w_y^{-}$$

- Let U_{test} be a test ordering and grouping of the state equation in single precision

Example:

$$U_{test} = \left(-\frac{\Delta t}{\Delta r}\left(\left(F^{+}+G^{+}\right) + \left(-F^{-} - G^{-}\right)\right) + \left(w_x^{+} + \left(\left(-w_x^{-} + w_y^{+}\right) - w_y^{-}\right)\right)\right) + U_{old}$$

Experiment Setup: Relative difference

For each time step, compute the relative difference:

relative difference $= \left| \frac{U_{gold} - U_{test}}{U_{gold}} \right|$ (error in double vs. single)

U_{gold} U_{test} *relative difference*

$$\left| \frac{0.443948170945829 - 0.4439482}{0.443948170945829} \right| = 0.0000000654449616$$

If relative difference >= error threshold, the error is not tolerable

Experiment Setup: Relative difference

- State equation runs 56 million times
- Compute relative difference every time the state equation runs
- Compute the error rate for the entire CLAMR run

$$E = \frac{\text{\# times relative difference exceeds the threshold}}{\text{\# times state equation executes}}$$

The error rate for the original version of the state equation, when moving from **double** precision to **single** precision is **2.1%**

- Smaller relative errors may not guarantee a “better” simulation output
- We don’t know until we test using goodness metric

Experiment Setup: Relative difference

For each time step, compute the relative difference:

relative difference $= \left| \frac{U_{gold} - U_{test}}{U_{gold}} \right|$ (error in double vs. single)

U_{gold} U_{test} *relative difference*

$$\left| \frac{0.443948170945829 - 0.4439482}{0.443948170945829} \right| = 0.0000000654449616$$

If relative difference >= error threshold, the error is not tolerable

Experiments: Ordering and grouping by magnitude (clumps)

$$U_{new} = U_{old} - \frac{\Delta t}{\Delta r}(F^+ - F^- + G^+ - G^-)(w_x^+ - w_x^- + w_y^+ - w_y^-)$$

$$U_{test} = \left[\blacktriangle \; \blacktriangle \; \blacktriangle \; \blacktriangle + U_{old} \right] + \pentagon \; \pentagon \; \pentagon \; \pentagon$$

Experiments: Ordering and grouping by magnitude (clumps)

$$U_{new} = U_{old} - \frac{\Delta t}{\Delta r}(F^+ - F^- + G^+ - G^-)(w_x^+ - w_x^- + w_y^+ - w_y^-)$$

$$U_{test} = \left[\blacktriangle + ((\blacktriangle + \blacktriangle) + \blacktriangle) + U_{old} \right] + ((\pentagon + \pentagon) + (\pentagon + \pentagon))$$

675 of these

Experiments: Pairwise grouping and ordering and grouping all nine terms

$$U_{new} = U_{old} - \frac{\Delta t}{\Delta r}\left(F^{+} - F^{-} + G^{+} - G^{-}\right) + w_x^{+} - w_x^{-} + w_y^{+} - w_y^{-}$$

$U_{test} =$ U_{old} + 

Experiments: Pairwise grouping and ordering and grouping all nine terms

$$U_{new} = U_{old} - \frac{\Delta t}{\Delta r}(F^{+} - F^{-} + G^{+} - G^{-}) + w_x^{+} - w_x^{-} + w_y^{+} - w_y^{-}$$

$U_{test}=$  $+ U_{old} +$

Experiments: Pairwise grouping and ordering and grouping all nine terms

$$U_{new} = U_{old} - \frac{\Delta t}{\Delta r}\left(F^{+} - F^{-} + G^{+} - G^{-}\right) + w_x^{+} - w_x^{-} + w_y^{+} - w_y^{-}$$

2,027,025 of these.
We'll sample 10,000

$U_{test,all\ nine}$ = 

$U_{test,pairwise}$ = 

11,340 of these

L. Monroe and V. Job. Computationally inequivalent summations and their parenthetic forms. http://arxiv.org/abs/2005.05387, 2020.

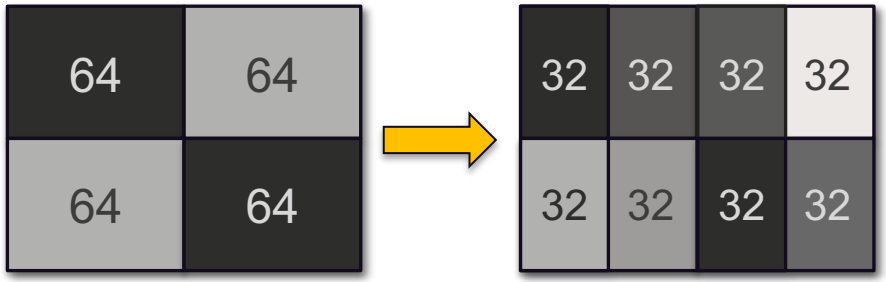
Experiments: Traditional methods

- Kahan
- Sorting
- Several others

N. J. Higham. The accuracy of floating point summation. *j-SISC*, 14(4):783–799, July 1993.

Motivation

- With increased accuracy, simulations may be able to run with reduced precision
- With lower precision, finer grained resolution may be possible for the same storage footprint
  - This may lead to overall increased accuracy



Experiment Setup: Screening method

- Let U_{gold} be the original ordering and grouping of the state equation in double precision:

$$U_{gold} = U_{old} - \frac{\Delta t}{\Delta r}\left(F^{+} - F^{-} + G^{+} - G^{-}\right) + w_x^{+} - w_x^{-} + w_y^{+} - w_y^{-}$$

- Let U_{test} be a test ordering and grouping of the state equation in single precision

Example:

$$U_{test} = \left(-\frac{\Delta t}{\Delta r}\left(\left(F^{+}+G^{+}\right) + \left(-F^{-} - G^{-}\right)\right) + \left(w_x^{+} + \left(\left(-w_x^{-} + w_y^{+}\right) - w_y^{-}\right)\right)\right) + U_{old}$$