Finite Difference Stencils Robust to Silent Data Corruption

Jaideep Ray, Jackson R. Mayo, and Robert C. Armstrong
Sandia National Laboratories, Livermore, CA 94551, USA
{jairay, jmayo, rob}@sandia.gov
In extreme-scale algorithm development, need to account for likely “silent errors”

- Silent errors (undetected hardware glitches) will become more commonplace at extreme scale
  - Mere “completion” of jobs no longer constitutes success — also need to ensure that results are sufficiently reliable

- Empirical information on silent-error models for extreme-scale platforms is limited, but:
  - We can seek ways of correcting at least some likely errors at the application level via algorithm-based fault tolerance (ABFT)
  - Understanding what errors can be mitigated will expand resilient co-design options such as more efficient but “lossy” hardware
Physics simulation, a dominant HPC application, has promising ABFT options

- Any **predictive simulation** requires that the physical quantities of interest are not sensitive to unavoidable perturbations (noise) in the real physical system.
  - For behaviors that are physically unstable to perturbations (e.g., turbulence), the quantities of interest are stable **statistics**.

- Already today’s solver algorithms must ensure that numerical **roundoff and discretization** errors are damped similarly to physical noise.
  - Good algorithms render these errors negligible by construction.

- Likewise, seek to **damp hardware glitches** by leveraging stability & smoothness properties of the physical system being simulated.
  - In general, add phase-space redundancy to the simulation to improve response to digital errors.
Memory bit flips in floating-point data are a useful silent-error model to consider.

- Even at commodity scale, ECC memory shows the rising need for error correction, and potential savings if it could be omitted.
- In extreme-scale scientific computing, floating-point (FP) data are an obvious concern for silent data corruption:
  - FP data often constitute the bulk of memory usage.
  - Data corruption can also be a proxy for processor FP glitches.
  - Corruption in other places (logic, pointers) is more likely to cause outright crashes, which will be mitigated by other means.
  - Relaxing FP correctness can benefit accelerators such as GPUs.

- Our error-injection framework for solvers: Asynchronously perform raw memory bit flips in the FP solution array.
We seek joint digital-physical resilience designs for solvers

- A "purely digital" example is **triple modular redundancy**: Run each step 3 times and vote frequently to correct glitches.
  - Requires $3 \times$ memory overhead and $> 3 \times$ processing overhead (including voting).
  - Damps silent errors by brute force without taking advantage of any properties of the simulation.

- A "purely physical" example is **artificial viscosity**: Augment PDE with diffusive term that eventually damps any perturbation.
  - But this leaves an imprint that is not physically meaningful.
  - Minimizing this effect requires a very fine grid, another form of costly redundancy.

- Instead, seek expanded discretizations of **physical dynamics** specifically targeted at damping **digital perturbations**.
Robust stencils discard outliers to mitigate silent data corruption

- Example: Finite difference scheme for 1D linear advection equation $\frac{\partial u}{\partial t} = -a \frac{\partial u}{\partial x}$; CFL number $c = \frac{a \Delta t}{\Delta x}$

- Standard Lax–Wendroff (LW) stencil is stable and second-order: $u_j^{n+1} = (\frac{1}{2} c + \frac{1}{2} c^2) u_{j-1}^n + (1 - c^2) u_j^n + (\frac{1}{2} c - \frac{1}{2} c^2) u_{j+1}^n$

- An isolated bit flip will affect one of the inputs to the stencil and often will cause a large numerical deviation

- Strategy: Use a large enough window that a single “most suspect” point can be discarded and a stencil is always available with no dependence on that point

- Here, we can derive a stable second-order update at $j$ from $\{j-2, j, j+2\}$ or $\{j-3, j-1, j+1, j+3\}$ as alternatives to $\{j-1, j, j+1\}$, so a 7-point robust stencil exists

- Note: These are stencils for the updated value, rather than for the increment, so it is okay for $j$ itself to be discarded
Simple robust stencil implementation offers proof of concept

- Compute outlier-insensitive "central" value – e.g., median of \( \{j-1, j, j+1\} \) – and discard point with largest absolute deviation of the 7, including a NaN or Inf (no explicit threshold needed)

- This tends to discard points on periphery when the solution is smooth, enabling use of most compact (LW) stencil

- Barring two bit flips in same window, undetected glitches are numerically small and behave similarly to discretization errors

**Simple demo in Mathematica**

(synthetic glitches)
Bit-flip injection at machine level confirms effectiveness of robust stencil

- 1D linear advection solver implemented in C++ with asynchronous memory bit flips
- Start with short single-core runs on a workstation
- Compare standard (LW), robust, and triple modular redundancy (TMR: 3 LW solutions with voting at each time step)

![Graphs showing relative memory use and runtime comparison between Standard, Robust, and TMR]

Here, the robust stencil provides similar bit-flip tolerance at lower cost than TMR.
Preliminary weak-scaling experiments show favorable trends for robust stencils

- Because robust stencils (similar to TMR) should fail only when two bit flips occur in proximity, we expect that failure of robust stencils occurs at second order in the local error rate.

- Thus, robust stencils should impose much weaker requirements for reduction of local error rates as scale increases.

- As a research tool, we have implemented a modular C++/MPI framework for explicit Cartesian PDE solvers.

- Favorable trends confirmed by initial 1D linear advection results from short runs on a capacity cluster, $10^6$ grid cells per core.

---

![Graphs showing relative wall runtime and max tolerated error probability per bit per standard time step.](chart.png)

- **Manageable overhead:**
  - Relative wall runtime:
    - Robust: $\sim 3\times$ runtime
    - Standard

- **Increasing resilience advantage:**
  - Max tolerated error probability per bit per standard time step:
    - Robust: $\sim 5000\times$ increasing resilience advantage
    - Standard
The inviscid Burgers equation idealizes nonlinear shock dynamics

- Inviscid Burgers equation describes fluid motion without pressure or viscosity, here in 1D for velocity $u$

$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x} = -\frac{\partial}{\partial x} \left( \frac{1}{2} u^2 \right)$$

- Even without an explicit viscosity term, this equation is dissipative – velocity gradients steepen to form shocks where fluid elements (characteristics) disappear.

- It is equivalent to the inviscid KPZ equation for interface growth

$$\frac{\partial h}{\partial t} = \frac{1}{2} \left( \frac{\partial h}{\partial x} \right)^2, \quad u \equiv -\frac{\partial h}{\partial x}$$

- Note that $h$ (unlike $u$) is continuous, with “cusps” instead of “jumps” at shocks.
We consider a Burgers/KPZ solution that illustrates formation of shocks.
The physics of the Burgers equation informs stable discretizations

- With Burgers equation in form of “hyperbolic conservation law”, standard second-order MacCormack stencil applies
  
  - Remarkably, MacCormack stencil can be written directly for KPZ field $h$ and we implement it in this way

- Seeking robust stencil, we note exact KPZ maximum principle

\[
h(t, x) = \max_y \left( h(0, y) - \frac{(x - y)^2}{2t} \right)
\]

- Because this principle reflects bounds on the solution that hold even around shocks, a stencil based on this principle should be highly stable:
  
  - Interesting as a Burgers discretization in general
  
  - And particularly suitable for creating a robust stencil
We create a second-order KPZ stencil from the maximum principle

- **Strategy:** Spatially interpolate $h$ field and analytically maximize

- **First**, a 5-point KPZ stencil that automatically captures shocks:
  Find maxima of interpolations on 5 intervals around cell $j$
  - $\{j-2, j-1, j\}$ (second-order)
  - $\{j-1, j\}$ (first-order)
  - $\{j-1, j, j+1\}$ (second-order)
  - $\{j, j+1\}$ (first-order)
  - $\{j, j+1, j+2\}$ (second-order)

- **Take second highest** of these maxima as the $h$ update – why?
  - In smooth regions, the 2 first-order updates (upwind and downwind) bracket the second-order updates, so we get second-order
  - At shocks, the 2 highest updates are first- and second-order in the direction with greater $|u|$, so we get upwind
Thanks to maximum principle, we intrinsically avoid oscillations seen with MacCormack and similar stencils.

- We are investigating generalizations to other hyperbolic conservation equations.

- Readily extend to a 7-point robust stencil for Burgers/KPZ.
  - Discard the largest outlier as before.
  - Use 5 available (not necessarily consecutive) points around $j$ for the interpolations.
Favorable weak-scaling trends also observed for Burgers robust stencil

- We use a norm for solution accuracy based on velocity field $u$
  - This is a very stringent metric since slight deviations in shock position cause jumps in $u$
- Compare standard (MacCormack) and robust KPZ-based stencils
- Behavior similar to linear case seen in initial 1D Burgers results from short runs on a capacity cluster, $10^5$ grid cells per core
Robust stencils show promise for mitigating silent data corruption at extreme scale

- Confronting silent errors suggests further research needs in applied mathematics and computer science.
  - Variability in simulation results due to hardware glitches calls for a new type of **uncertainty quantification** in scientific computing.
  - Because hardware and software jointly affect the reliability of results, a **resilient co-design** process is needed for future extreme-scale systems.
- “What matters in floating-point computation is how closely a web of mathematical relationships can be maintained in the face of roundoff, and whether that web connects the program’s output strongly enough to its input no matter how far the web sags in between” (Kahan & Darcy, 2004)
Robust stencils show promise for mitigating silent data corruption at extreme scale

- Confronting silent errors suggests further research needs in applied mathematics and computer science
  - Variability in simulation results due to hardware glitches calls for a new type of uncertainty quantification in scientific computing
  - Because hardware and software jointly affect the reliability of results, a resilient co-design process is needed for future extreme-scale systems
- “What matters in floating-point computation is how closely a web of mathematical relationships can be maintained in the face of silent errors, and whether that web connects the program’s output strongly enough to its input no matter how far the web sags in between” (Kahan & Darcy, 2004)