Algorithms, architectures, and community for high-resolution climate modeling

Jed Brown jed.brown@colorado.edu (CU Boulder and ANL)

Latsis Symposium, 2019-08-22

This talk: https://jedbrown.org/files/20190822-Latsis.pdf

Accelerated Climate Modeling for Energy

ACME 2014 (now E3SM)

5.3 Computational Performance Improvement

During the first six months of the project, the performance engineering effort will assess ACME v0 application performance and identify the target metrics for each component in terms of its throughput and scaling behavior in the coupled system. Then improvements in on-node (using OpenMP or OpenACC to expose more parallelism and leverage features of the different LCF architectures) and between-node (with communication-hiding implemented over MPI) can be focused on improving the coupled model throughput. Our performance strategy for the v1 model was chosen to increase performance on the existing LCFs and position us to exploit new codesign-inspired approaches in the v2 model. We will focus on two key tasks: exposing increased concurrency throughout the model and increasing the on-core performance of key computational kernels. In v1, we will be using conventional approaches, such as threading and MPI for the first task and increased use of accelerators for the second task. Increases in concurrency and much of the work needed to refactor and modularize kernels for accelerators will be beneficial for most any possible exascale architecture. The throughput-based priority performance metric is:

- **(Perf1)** Maximum simulated years per wall-clock day of the coupled system running without I/O.

This includes the ACME target of five simulated years per wall-clock day (SYPD), and the required performance by each component that is needed to achieve that target. Because I/O performance is simulation-dependent (based on science objectives), it will be dealt with in a separate task that focuses on the most critical issues.

architecture. The throughput-based priority performance metric is:

only one objective!

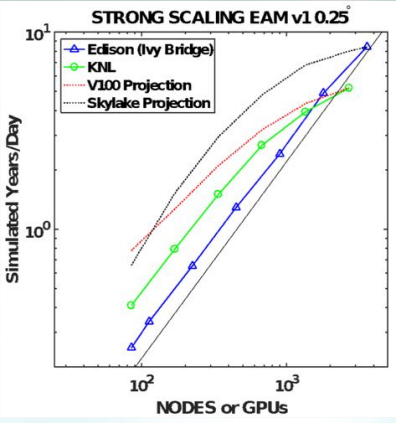
- **(Perf1)** Maximum simulated years per wall-clock day of the coupled system running without I/O.

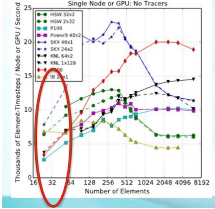
This includes the ACME target of five simulated years per wall-clock day (SYPD), and the required performance by each component that is needed to achieve that target. Because I/O performance is simulation-

“No DOE facility through 2020 will run ACME faster than Edison” – 2013

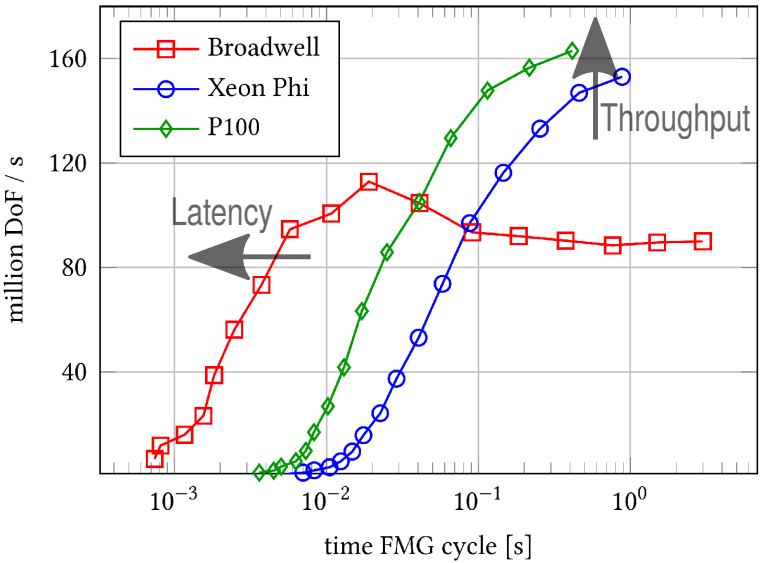
Atmosphere Performance

- Full EAM model, v1 physics
- 28km,72L, 40 tracers
- Only with sufficient work per node can KNL outperform Edison (Xeon Ivy Bridge)
- Skylake and GPU (V100) Projections based on benchmarked dycore single node performance



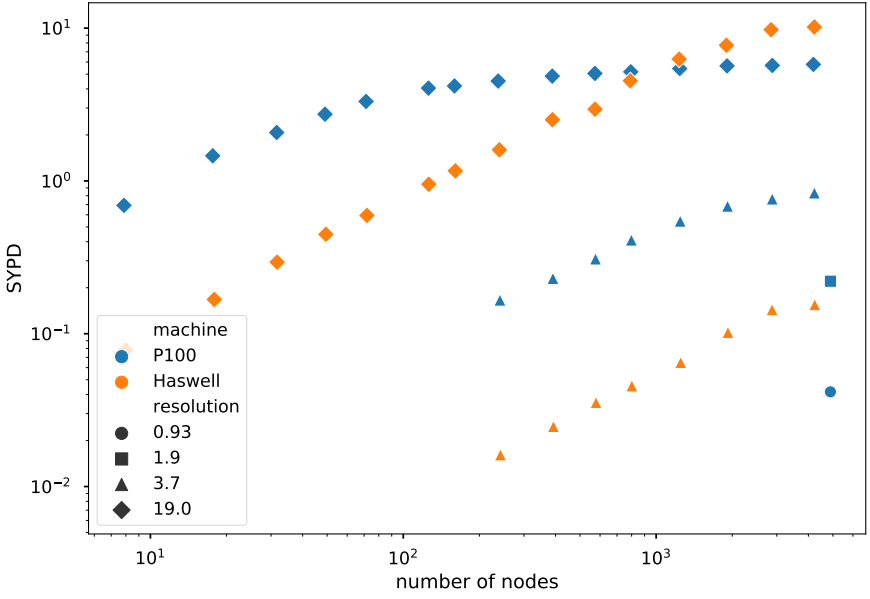


Latency versus Throughput

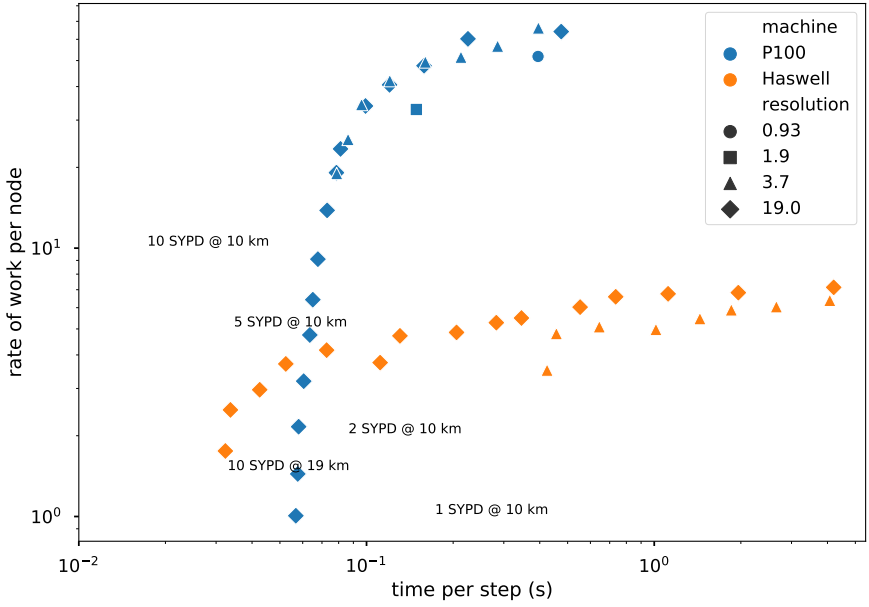


Adapted from Kronbichler and Ljungkvist (2019)

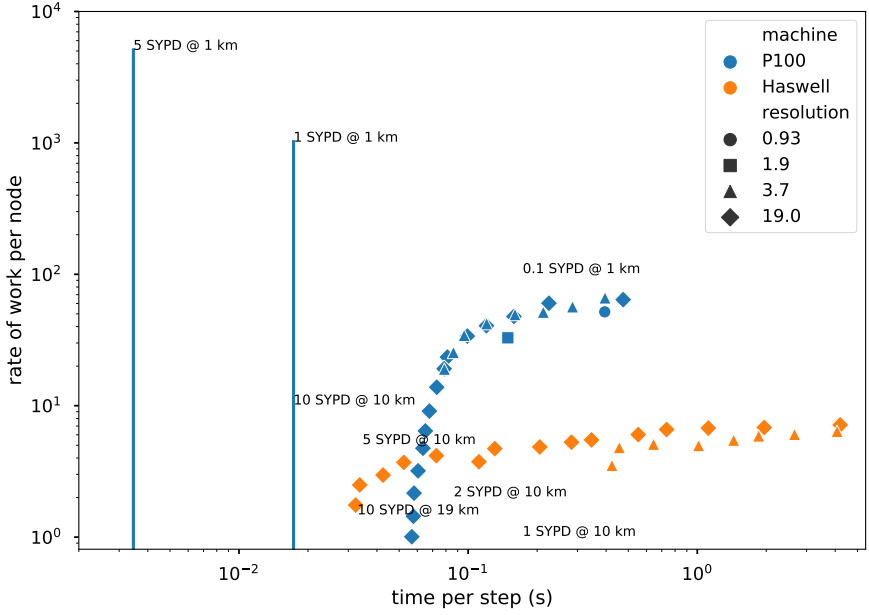
Fuhrer at al, 2018



Fuhrer at al, 2018: Work-Time spectrum



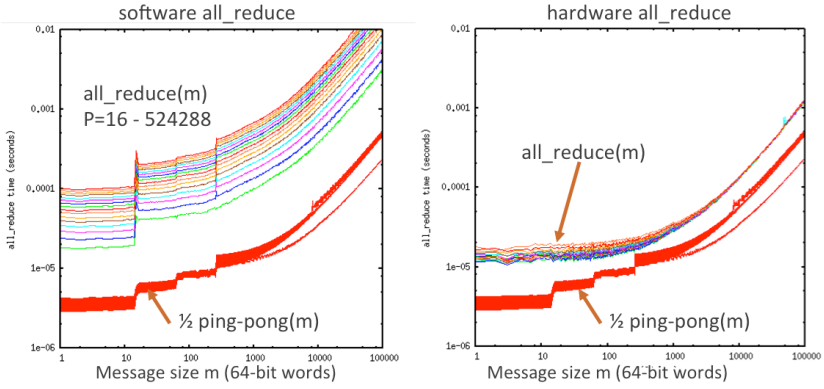
Fuhrer at al, 2018: Work-Time spectrum



MPI_Allreduce performance, c/o Paul Fischer

Eliminating log P term in CG

- On BG/L, /P, /Q, all_reduce is nearly ***P-independent***.
- For P=524288, all_reduce(1) is only 4α !

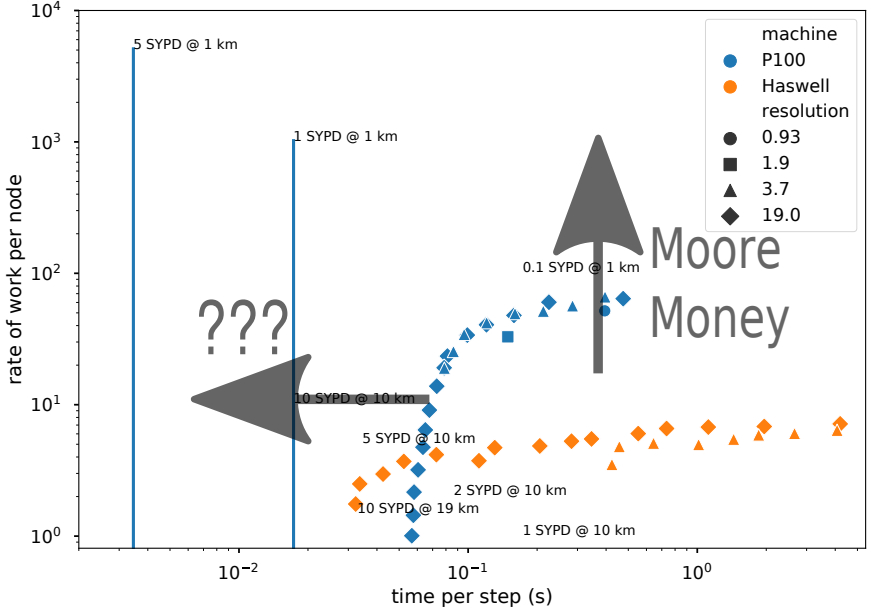


Latency hasn't improved much in 15 years

Year	Latency (μs)	1/Bandwidth (μs/word)	Machine
1986	5960	64	Intel iPSC-1 (286)
1988	938	2.8	Intel iPSC-2/ (386)
1990	80	2.8	Intel iPSC-i860
1991	60	0.8	Intel Delta
1992	50	0.15	Intel Paragon
1995	60	0.27	IBM SP2 (BU96)
1996	30	0.02	ASCI Red 333
1999	20	0.04	Cray T3E/450
2005	4	0.026	BGL/ANL
2008	3.5	0.022	BGP/ANL
2011	2.5	0.002	Cray XE6 (KTH)
2012	3.8	0.0045	BGQ/ANL
2015	2.2	0.0015	Cray XK7

Measured machine-dependent parameters from Fischer, Heisey, Min (2015)

Fuhrer at al, 2018: Work-Time spectrum



What won't save us?

$$\text{Time} = \text{Latency} + \frac{\text{Data volume}}{\text{bandwidth}} + \frac{\text{Work}}{\text{Compute rate}}$$

- *s*-step methods (high overhead in strong scaling regime)
- Adaptivity (doesn't reduce latency)
- Reduced precision (doesn't reduce latency)
- Parallel-in-time integrators
  - Poor efficiency
  - Lack of stable coarse integrator
  - Slow convergence with positive Lyapunov exponent

Algorithmic themes

- Do more work each time you pay for latency
  - Communicate
  - GPU kernel launch
  - `#pragma omp barrier`, etc.
- Combine operations via implicitness
  - Only possible with very fast iterative convergence!
- Control time-splitting errors

Runge-Kutta methods

$$\dot{u} = F(u)$$

$$\underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_s \end{pmatrix}}_{Y} = u^n + h \underbrace{\begin{bmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{s1} & \cdots & a_{ss} \end{bmatrix}}_{A} F \begin{pmatrix} y_1 \\ \vdots \\ y_s \end{pmatrix}$$

$$u^{n+1} = u^n + hb^T F(Y)$$

- General framework for one-step methods
- Diagonally implicit: A lower triangular, stage order 1 (or 2 with explicit first stage)
- Singly diagonally implicit: all A_{ii} equal, reuse solver setup, stage order 1
- If A is a general full matrix, all stages are coupled, “implicit RK”

Method of Butcher (1976) and Bickart (1977)

- Newton linearize Runge-Kutta system at u^*

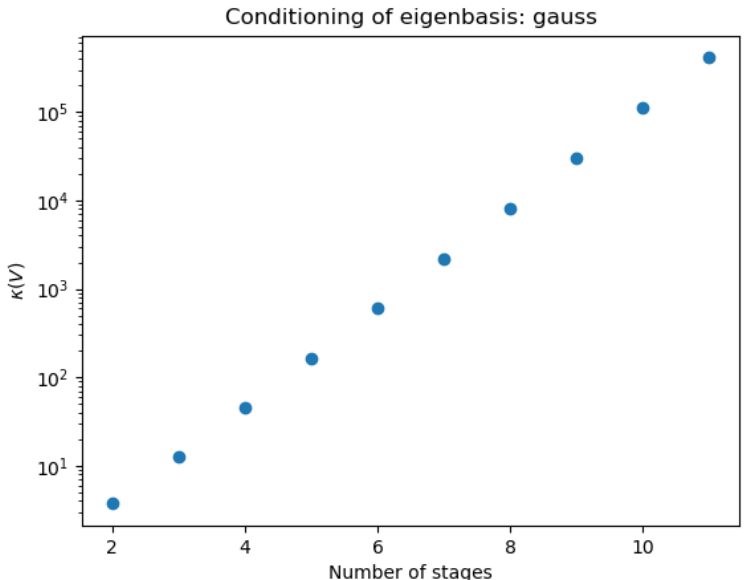
$$Y = u^n + hAF(Y) \qquad \left[I_s \otimes I_n + hA \otimes J(u^*)\right] \delta Y = RHS$$

- Solve linear system with tensor product operator

$$\hat{G} = S \otimes I_n + I_s \otimes J$$

  where $S = (hA)^{-1}$ is $s \times s$ dense, $J = -\partial F(u)/\partial u$ sparse
- SDC (2000) is Gauss-Seidel with low-order corrector
- Butcher/Bickart method: diagonalize $S = V \Lambda V^{-1}$
  - $\Lambda \otimes I_n + I_s \otimes J$
  - s decoupled solves
  - Complex eigenvalues (overhead for real problem)

Eigenbasis ill conditioning $A = V\Lambda V^{-1}$



Why implicit is silly for waves

- Implicit methods require an implicit solve in each stage.
- Time step size proportional to CFL for accuracy reasons.
- Methods higher than first order are not unconditionally strong stability preserving (SSP; Spijker 1983).
  - Empirically, $c_{\text{eff}} \leq 2$, Ketcheson, Macdonald, Gottlieb (2008) and others
  - Downwind methods offer to bypass, but so far not practical
- Time step size chosen for stability
  - Increase order if more accuracy needed
  - Large errors from spatial discretization, modest accuracy
- My goal: need less data motion *per stage*
  - Better accuracy, symplecticity nice bonus only
  - Cannot sell method without efficiency

Implicit Runge-Kutta for advection

Table: Total number of iterations (communications or accesses of J) to solve linear advection to $t = 1$ on a 1024-point grid using point-block Jacobi preconditioning of implicit Runge-Kutta matrix. The relative algebraic solver tolerance is 10^{-8}.

Method	order	nsteps	Krylov its.	(Average)
Gauss 1	2	1024	3627	(3.5)
Gauss 2	4	512	2560	(5)
Gauss 4	8	256	1735	(6.8)
Gauss 8	16	128	1442	(11.2)

- Naive centered-difference discretization
- Leapfrog requires 1024 iterations at CFL=1
- This is A-stable (can handle dissipation)

Diagonalization revisited

$$(I \otimes I - hA \otimes L)Y = (\mathbf{1} \otimes I)u_n \tag{1}$$

$$u_{n+1} = u_n + h(b^T \otimes L)Y \tag{2}$$

- eigendecomposition $A = V\Lambda V^{-1}$

$$(V \otimes I)(I \otimes I - h\Lambda \otimes L)(V^{-1} \otimes I)Y = (\mathbf{1} \otimes I)u_n.$$

- Find diagonal W such that $W^{-1}\mathbf{1} = V^{-1}\mathbf{1}$
- Commute diagonal matrices

$$(I \otimes I - h\Lambda \otimes L)\underbrace{(WV^{-1} \otimes I)Y}_{Z} = (\mathbf{1} \otimes I)u_n.$$

- Using $\tilde{b}^T = b^T VW^{-1}$, we have the completion formula

$$u_{n+1} = u_n + h(\tilde{b}^T \otimes L)Z.$$

- $\Lambda, \tilde{b}$ is new diagonal Butcher table

REXI: Rational approximation of exponential

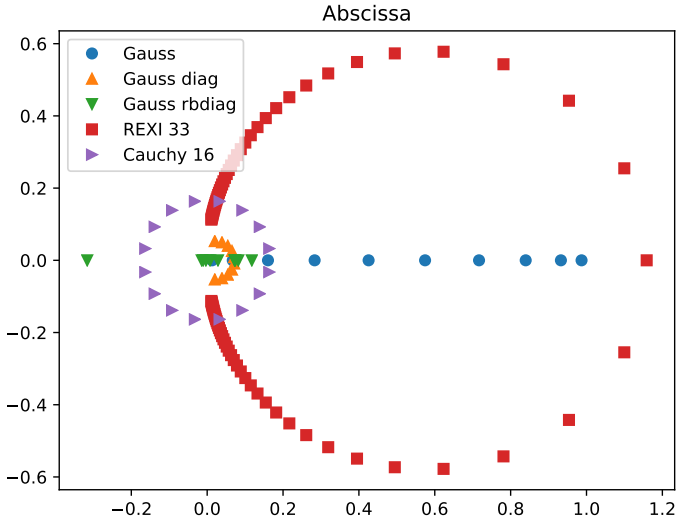
$$u(t) = e^{Lt} u(0)$$

- Haut, Babb, Martinsson, Wingate; Schreiber and Loft

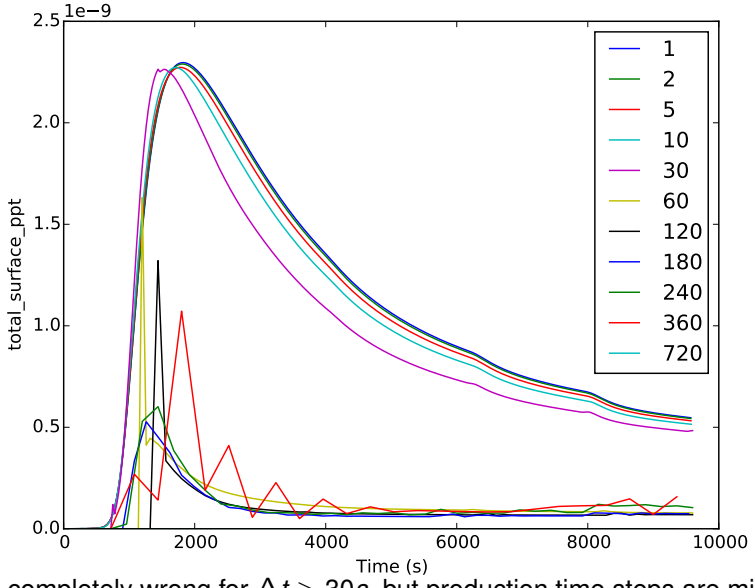
$$(\alpha \otimes I + hI \otimes L) Y = (\mathbf{1} \otimes I) u_n$$
$$u_{n+1} = (\beta^T \otimes I) Y.$$

- α is complex-valued diagonal, β is complex
- Constructs rational approximations of Gaussian basis functions, target (real part of) e^{it}
- REXI is a Runge-Kutta method: can convert via "modified Shu-Osher form"
  - Developed for SSP (strong stability preserving) methods
  - Ferracina, Spijker (2005), Higueras (2005)
  - Yields diagonal Butcher table $A = -\alpha^{-1}, b = -\alpha^{-2}\beta$

Abscissa for RK and REXI methods

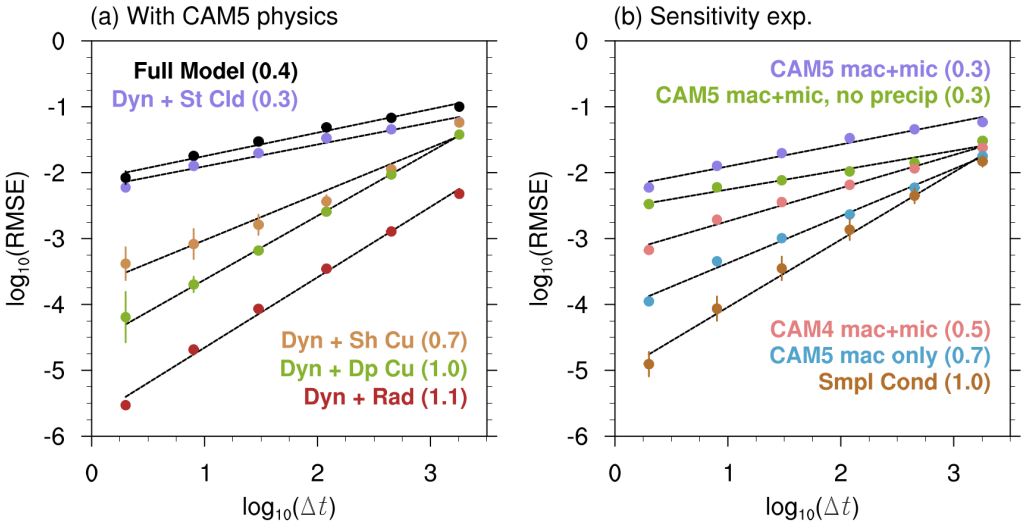


KiD: accuracy of time integrator



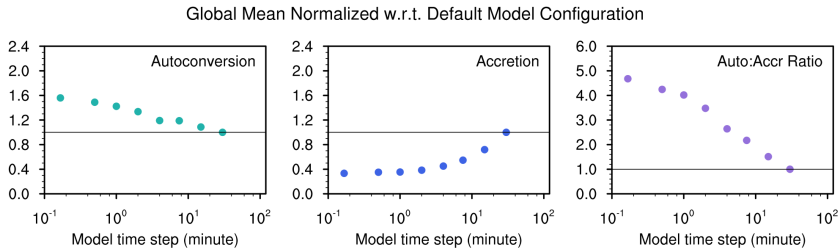
- Solution completely wrong for $\Delta t > 30s$, but production time steps are minutes

Slows convergence of global model



[Wan et al. 2015]

Impact of time step on autoconversion vs accretion partitioning (from Hui)



- Parameters calibrated for systematic discretization error

Parameter tuning

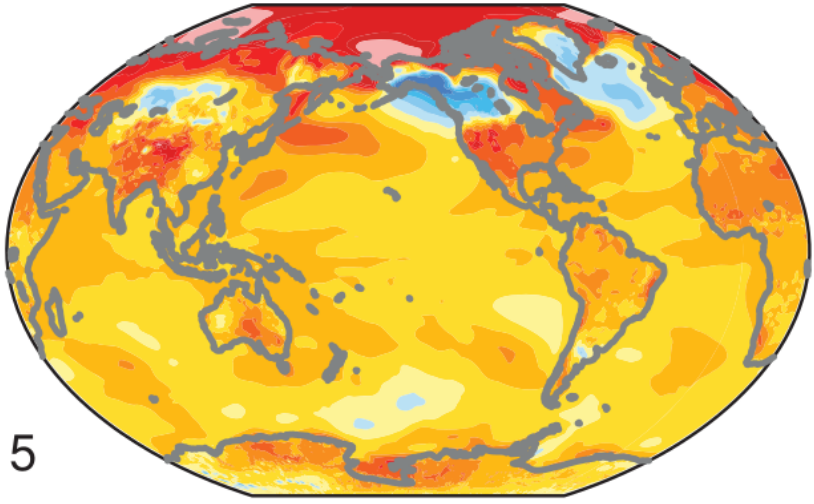
With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.
— John von Neumann

- Over-fitting is a pathology
- *Good* subgrid models do not require (much) re-tuning parameters when Δt or Δx change
- Experimenting with new discretizations requires expensive, ad-hoc parameter re-calibration.

Are we solving the right problem?

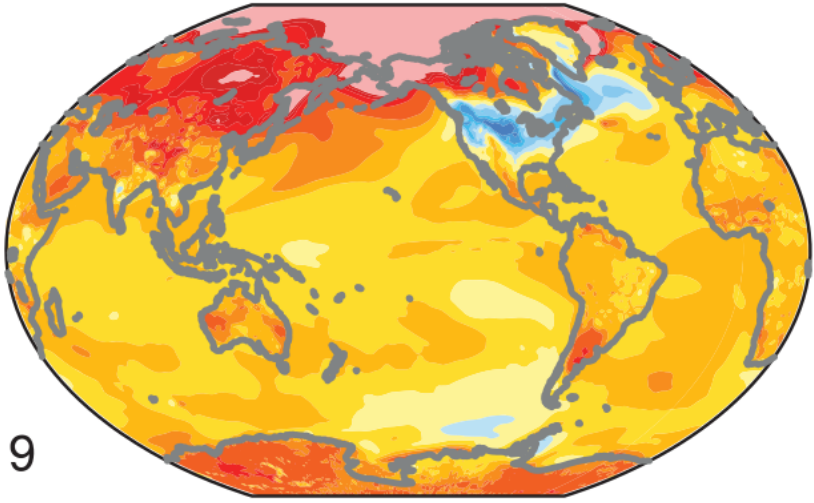
- CESM Large Ensmeble Project (Kay et al., 2015)
- 30-member ensemble
- Identical initial conditions except for $10^{-14}K$ perturbation in initial temperature
- CESM(CAM5) at $\sim 1^{\circ}$ resolution

Cliff Mass projected warming in the PNW



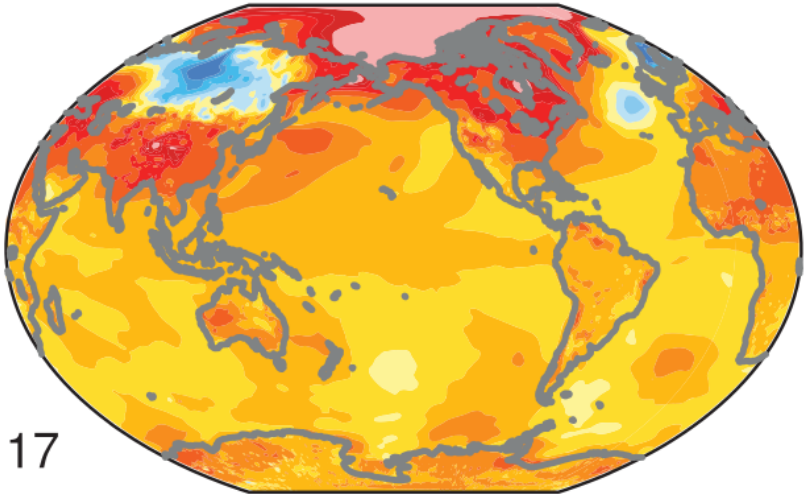
5

Cliff Mass projected warming in the PNW



9

Cliff Mass projected warming in the PNW



17

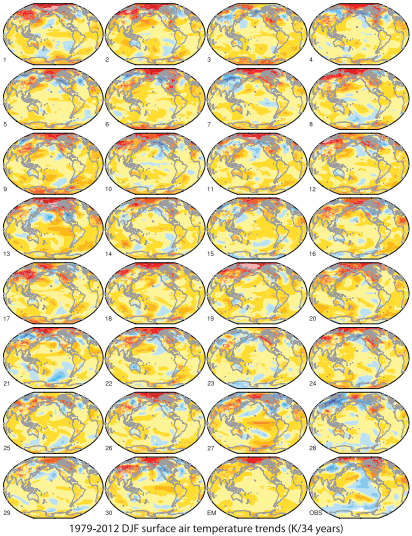


Fig. 4. Global maps of historical (1979–2012) boreal winter (DJF) surface air temperature trends for each of the 30 individual CESM-LE members, the CESM-LE ensemble mean (denoted EM), and observations (denoted OBS based on GISTEMP; Hansen et al. 2010).

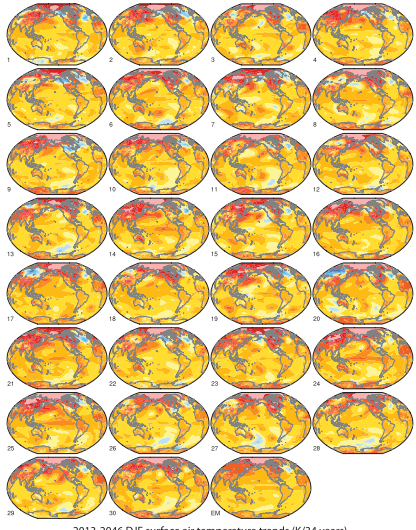


Fig. 5. Global maps of near-future (2013–46) boreal winter (DJF) surface air temperature trends for each of the 30 individual CESM-LE members and the CESM-LE ensemble mean (denoted EM).

Outlook

- Latency is a killer for high resolution. Needs to be confronted directly.
- Fascinating applied math/CS questions
- High inertia to address, especially when recalibration needed
- Need explicit support for career paths in methods development
- Thanks to DOE ASCR, BER, and ECP