Increasing the multiscale/multiphysics capability of CAM-SE using implicit time integration and GPU accelerators

Rick Archibald*, Katherine J. Evans*, Patrick Worley*, Matthew Norman*, P. Aaron Lott†, Andy Salinger‡, Mark A. Taylor‡, Carol Woodward§

*Oak Ridge National Laboratory, ‡Sandia National Laboratory, and §Lawrence Livermore National Laboratory

Summary

The inclusion of new physics, chemistry, and grid refinement of the recently released Community Atmosphere Model (CAM5) creates new algorithmic challenges, including coupled nonlinear multiscale processes and enhanced scalability requirements. These finer and more complex model configurations have led to recent work to utilize GPU processors within a supercomputer as well as numerical methods that can handle a variety of time scales and maintain acceptable accuracy and efficiency. Efforts to port the scalable spectral element dynamical core to incorporate these developments is presented, with early results, challenges, and next steps outlined in detail. The current implicit solver and preconditioner implementations utilize a Fortran interface package within the Trilinos project, third party software that allows fully tested, optimized, and robust code with a suite of parameter options to be included a priori. Merging this coding strategy with GPU libraries has been accomplished for a few targeted kernels. A full port of the implicit method with pre-conditioning is a priority

Scalable Preconditioning and Fast Solutions

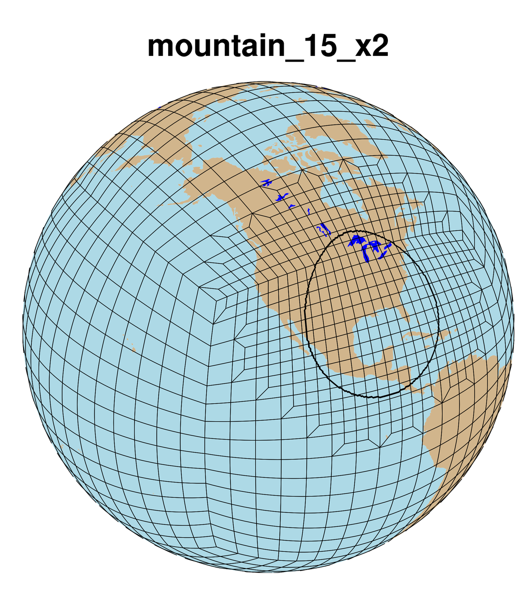
Benifits of Block Preconditioning Magnified with Scale

	Preconditioned				Unpreconditioned			
NP	TC1	TC5	TC6	SJTC1	TC1	TC5	TC6	SJTC1
4	1	1.6	2.1	2.5	2.0	5.5	6.1	7.5
8	1	2.6	2.4	2.7	2.5	17.5	14.6	21.7
12	1	3.2	2.5	3.1	3.3	38.3	30.2	41.5
16	1	3.3	2.6	4.3	4.9	49.9	36.4	49.9

Average preconditioned linear iterations per Newton step using various choices of NP for each test case

	Preconditioned				Unpreconditioned			
NE	TC1	TC5	TC6	SJTC1	TC1	TC5	TC6	SJTC1
12	1	1.6	2.1	2.5	2.0	5.6	6.2	7.5
24	1	1.8	2.1	2.8	2.5	10.4	11.7	13.6
48	1	2.6	2.1	2.7	2.5	20.8	21.4	25.9
96	1	3.1	2.3	2.9	3.4	42.1	35.4	43.0

Average preconditioned linear iterations per Newton step using various choices of NE for each test case



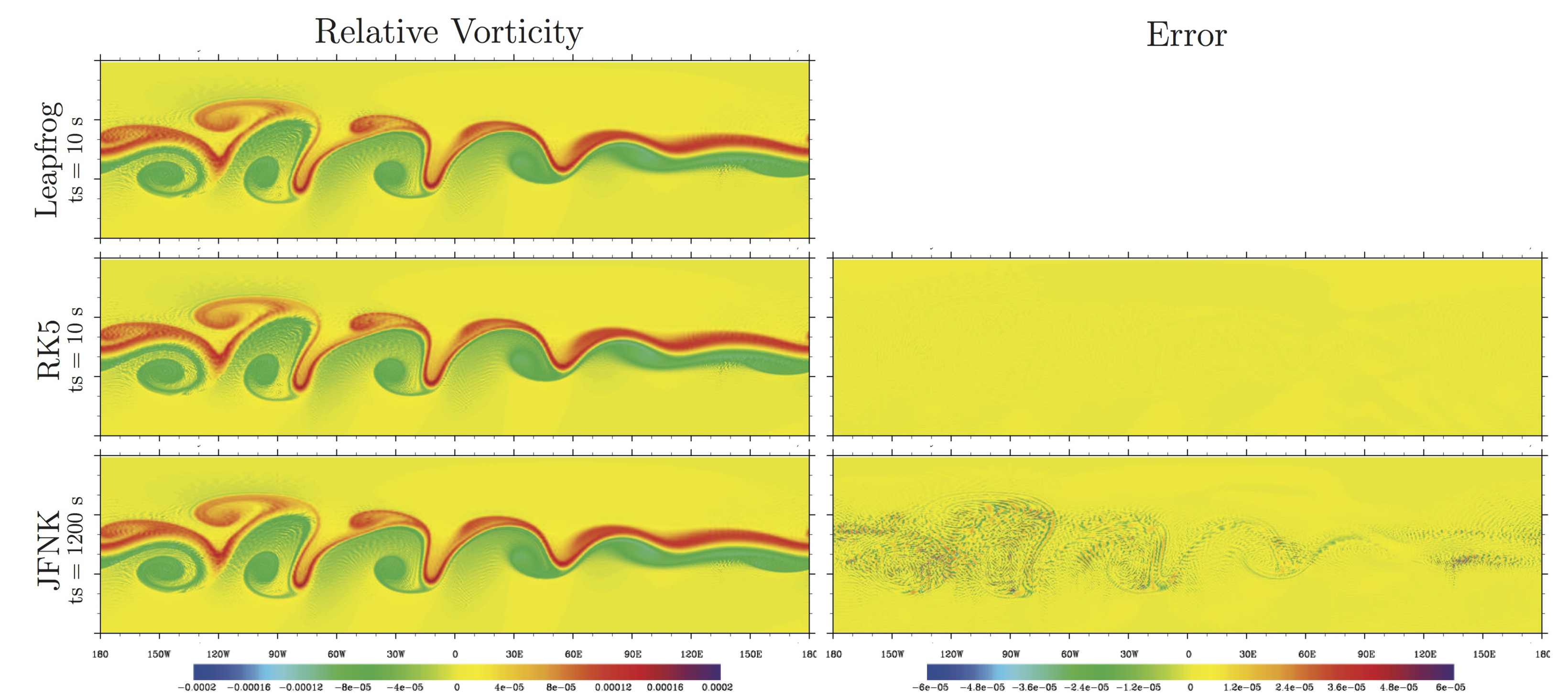
Matintained Time-step with local grid refinement

- -Grid refinement from 2° globally to 1° local refinement.
- -Implicit timestep ten times large than explicit CFL restriction.
- -Convergenved successfully to set tolerance of 1e-4.

Next Steps: Extend scalable preconditioner to full dycore and connect to tracers and physics

Efficient Fully Implicit (FI) Time Integration

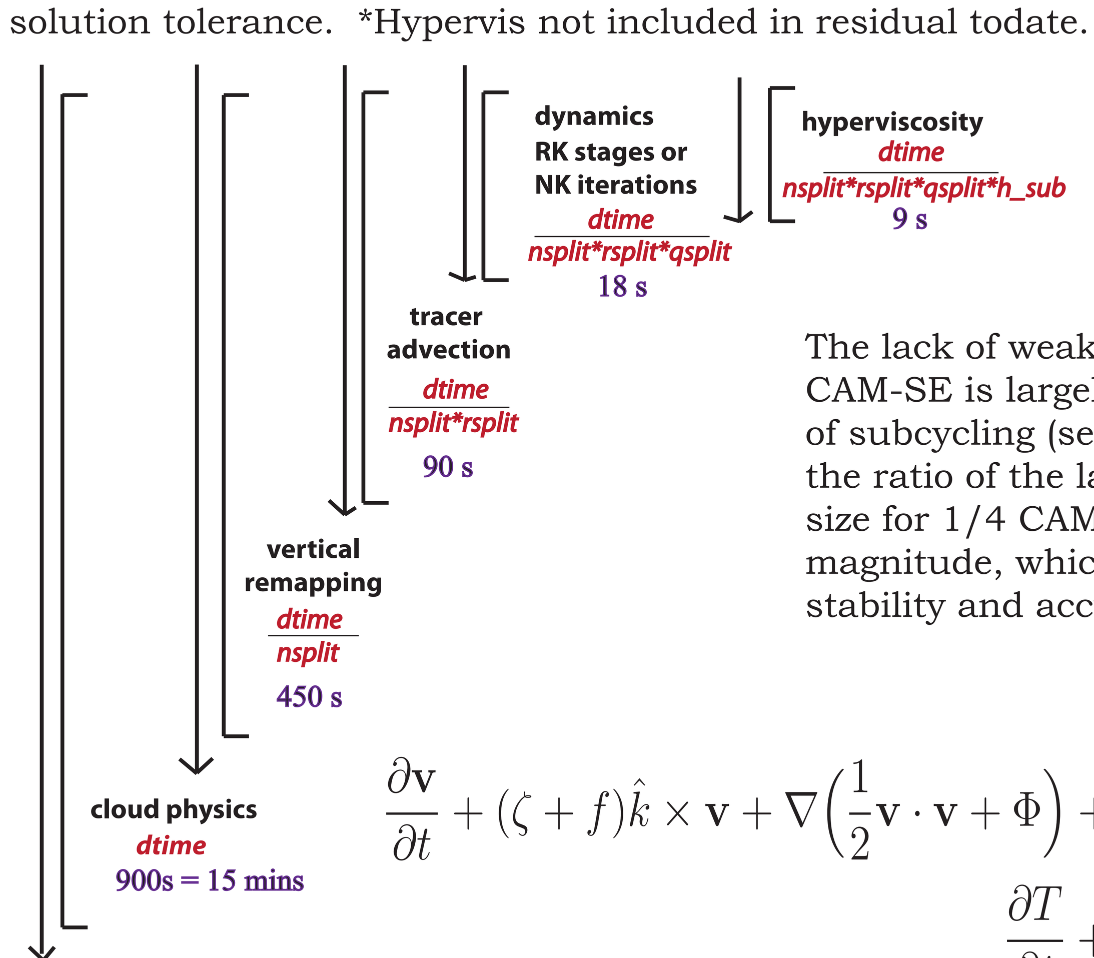
Nonlinearly consistent solution methods for 3D



The fully implicit Jacobian-Free Newton-Krylov method has been implemented in both the 2D shallow water dycore and the 3D primitive equations dycore within CAM-SE.

Method	δ T	Sim Time	qsplit	Hypervis	L_2 Error
Exp RK (2D)	10s	256s	NA	NA	NA
BDF2	1200s	222s	NA	NA	$1.3e^{-3}$
BDF2	1800s	210s	NA	NA	$1.1e^{-3}$
Exp RK (3D)	75s	85s	8	1	$5.9e^{-6}$
Exp RK	150s	67s	4	2	$1.7e^{-5}$
BE*	600s	259s	1	8	$1.4e^{-3}$
BDF2	150s	692s	4	1	$3.2e^{-4}$
BDF2	600s	210s	1	1	$7.2e^{-3}$

Displaying performance for 2D and 3D baroclinic instability test case (2D:np8,ne24,12D & 3D:np4,ne15,9D). For 3D, dynamics and hyperviscosity subcycling can be removed and the model can be run with the same time step as the tracers. For 2D, with significant scale separation, and efficiency gain is realized by using a bigger time step size and a 1e-3 solution tolerance. *Hypervis not included in residual todate.



The lack of weak scaling with the explicit CAM-SE is largely mitigated with many layers of subcycling (see figure to the left). However the ratio of the largest to smallest time step size for 1/4 CAM-SE covers 2 orders of magnitude, which can create issues with stability and accuracy.

$$\frac{\partial \mathbf{v}}{\partial t} + (\zeta + f)\hat{k} \times \mathbf{v} + \nabla\left(\frac{1}{2}\mathbf{v}\cdot\mathbf{v} + \Phi\right) + \frac{RT}{p}\nabla p + \dot{\eta}\frac{\partial \mathbf{v}}{\partial \eta} + \nu\Delta^2\mathbf{v} = 0$$

$$\frac{\partial T}{\partial t} + \mathbf{v}\cdot\nabla T + \dot{\eta}\frac{\partial T}{\partial \eta} - \frac{RT}{c_p}\frac{\omega}{p} = 0$$

$$\frac{\partial}{\partial t}\left(\frac{\partial p}{\partial \eta}\right) + \nabla\cdot\left(\frac{\partial p}{\partial \eta}\mathbf{v}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right) = 0$$

$$\frac{\partial}{\partial t}\left(\frac{\partial p}{\partial \eta}q\right) + \nabla\cdot\left(\frac{\partial p}{\partial \eta}q\mathbf{v}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}q\right) = 0.$$

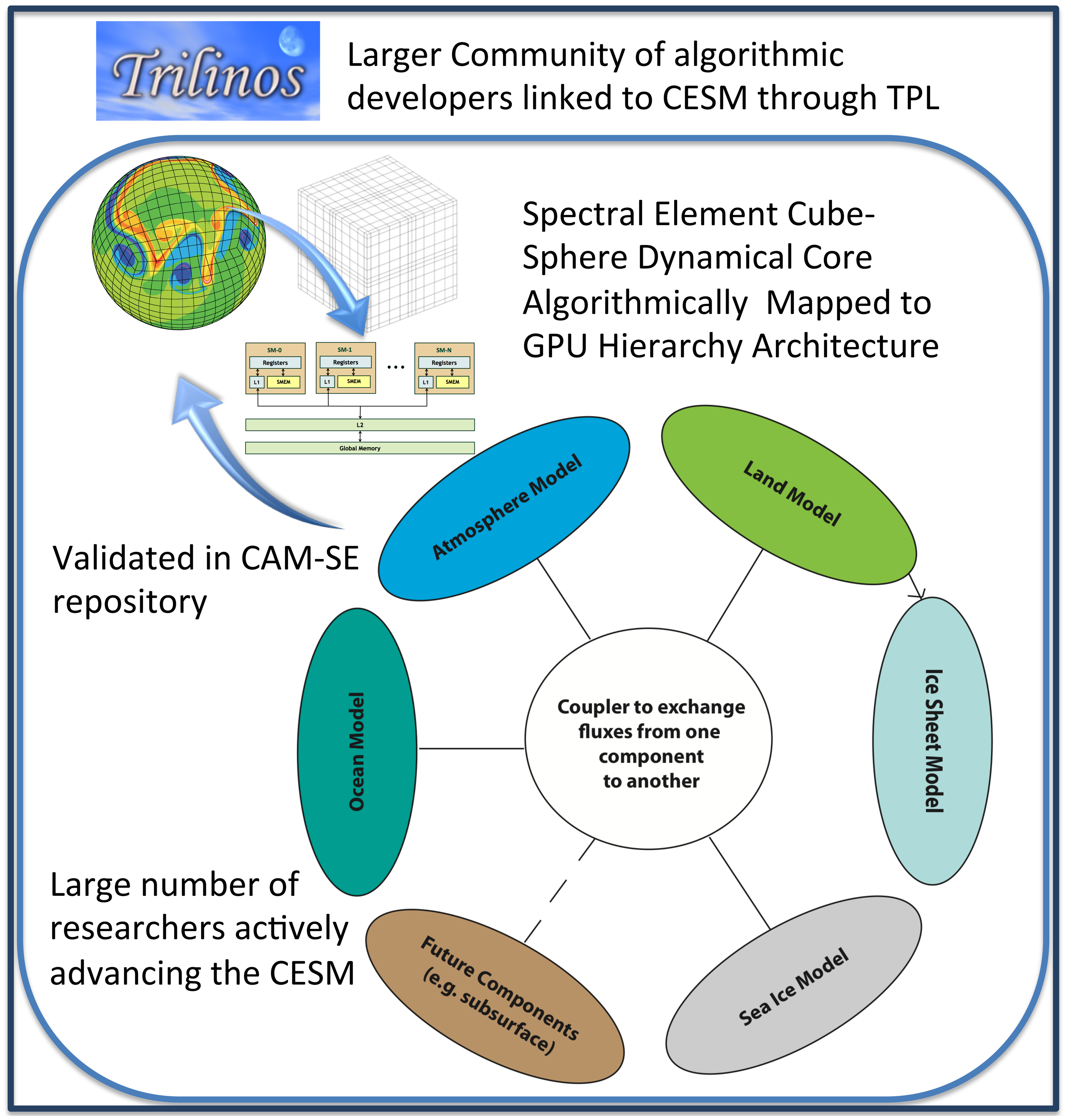
The primitive equations within CAM-SE are displayed to the left. With the hyperviscosity term within the residual (last term on the RHS of the velocity residual), the second order time integration scheme converges consistently.

Next Steps: Optimize performance, include a scalable preconditioner, extend to full dycore and connect to tracers and physics

GPU acceleration and time integration

Framework to accelerate high resolution CAM-SE with many tracers and implicit time stepping: utilize GPU effectively

Common build systm that connects Trilinos third party library (part of the FastMATH institute), used for the implicit solver, and the latest development for GPU kernels



Cost of data movement reduced by framework - *Need new CESM data structure*

Implicit Framework

```
do ie,q,k,j,i
     xstate(:) = elem(ie)%state%var(i,j,k,q)
end
call nxsolve(xstate,..)
do ie,q,k,j,i
      elem(ie)%state%var(i,j,k,q)=xstate(:)
end
```

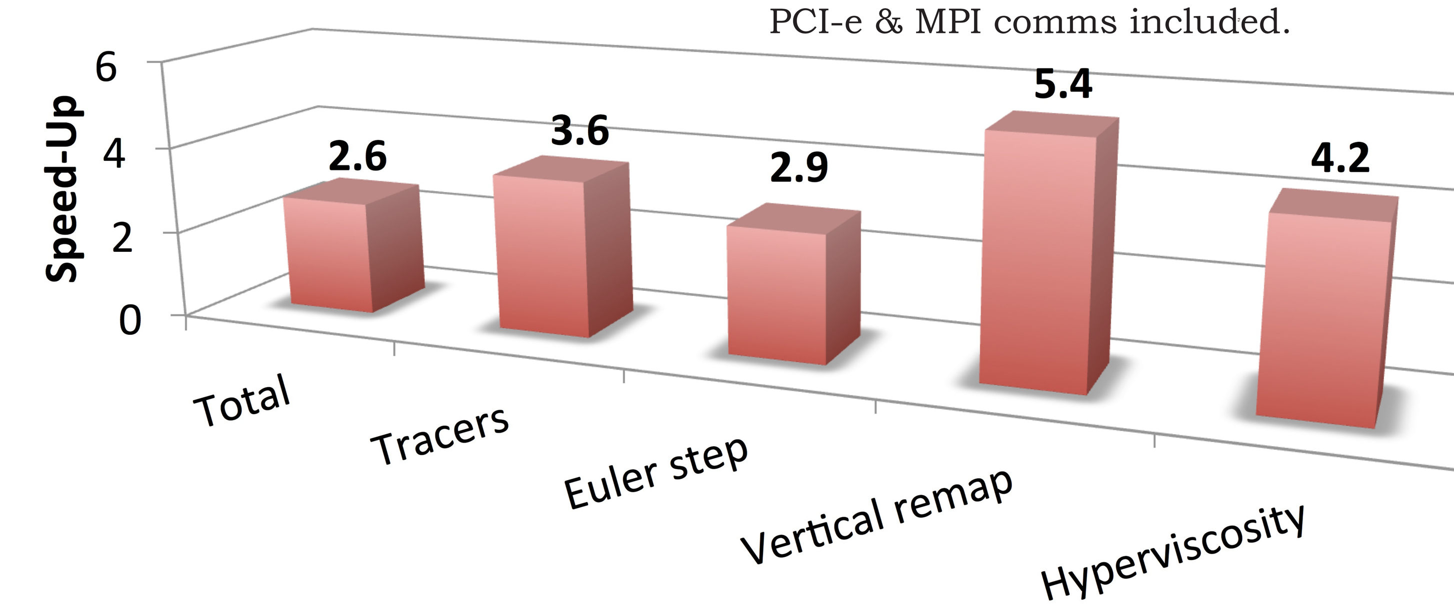
GPU Framework

```
do ie,q,k,j,i
     gpu_array(:) = elem(ie)%state%var(i,j,k,q)
end
call GPU_Kernel(gpu_array,..)
do ie,q,k,j,i
      elem(ie)%state%var(i,j,k,q)=gpu_array(:)
end
```

Implicit-GPU Framework

```
do ie,q,k,j,i
    xstate(:) = elem(ie)%state%var(i,j,k,q)
end
call nxsolve(xstate,..) ➔ call GPU_Kernel(xstate,..)
do ie,q,k,j,i
    elem(ie)%state%var(i,j,k,q)=xstate(:)
end
```

Accleration of 14km, full chemistry (110 tracers), base on Fermi GPU vs Interlagos XK6. PCI-e & MPI comms included.



Next steps: Complete the port of all implicit solver kernels & optimization of CAM-SE for hybrid architectures

Citations:

Carpenter, I., R.K. Archibald, K.J. Evans, J. Larkin, P. Micikevicius, J. Rosinski, J. Schwarzmeier, M.A. Taylor, M. Norman (2013). Progress towards accelerating HOMME on hybrid multi-core systems, Int. J. High Perf. Comp. App., 27(3):335-347.

Dennis, J., J. Edwards, K. J. Evans, O. Guba, P.H. Lauritzen, A. Mirin, A. St.-Cyr, M.A. Taylor, and P. H. Worley (2012). “A scalable spectral element dynamical core for the Community Atmosphere Model," Int. J. High Perf. Comp. App., 26: 74-89.

Evans, K. J., P. Lauritzen, S. K. Mishra, R. Neale, M. A. Taylor, and J. J. Tribbia (2012). “AMIP Simulations with the CAM4 Spectral Element Dynamical Core.” J. Climate, in press, doi: http://dx.doi.org/10.1175/JCLI-D-11-00448.1.

Evans K. J., M. A. Taylor and J. B. Drake (2010). Accuracy analysis of a spectral element atmospheric model using a fully implicit solution framework, Mon. Wea. Rev.**138**:3333-3341.

Lott P. A., Woodward, C. S., and Evans, K. J. “Algorithmically scalable block preconditioner for fully implicit shallow-water equations in CAM-SE.” submitted.

Polvani, L. M., R. K. Scott, S. J. Thomas (2004). Numerically converged solutions of the global primitive equations for testing the dynamical core of atmospheric GCMs, Mon. Wea. Rev.**132**:2539-2540.

Williamson, D.L., and J. B. Drake, J. J. Hack, R. Jakob, and P.L. Swarztrauber, (1992). “A standard test set for numerical approximations to the shallow water equations in spherical geometry.” J. Comp. Phys. **102**: 211-224.

Acknowledgements:

Climate Change Science Institute at Oak Ridge National Laboratory · Oak Ridge National Laboratory, Managed by UT-Battelle for the Department of Energy · Sandia National Laboratories

Research sponsored by the DOE/BER SciDAC Project: “Multiscale Methods for Accurate, Effcient, and Scale-Aware Models of the Earth System,” William Collins, PI with assistance from the DOE/ASCR SciDAC FastMATH Institute (Salinger and Woodward). This project used the resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.