A Scalable Robust Solver for the Diffusion Limited Radiation Transport Equations

Dinesh Kaushik
Argonne National Laboratory
David Keyes
Columbia University
William Gropp
Barry Smith
Argonne National Laboratory

Radiation Diffusion Equations

- Govern the evolution of photon radiation in an optically thick medium
- Derived by integrating over all energy frequencies, assuming
  - Isotropy (angle dependence averaged out)
  - Small mean-free photon paths
- Very important in the simulation of
  - forest fires
  - inertial confinement fusion (http://fusion.gat.com/icf)
  - Astrophysical phenomenon
- Suitable for analyzing the effectiveness of a nonlinear solution algorithm
  - We use them to simulate the radiation penetration from hot source to cold medium

Organization of the Presentation

- Introduction
- Discretization of 2T Radiation Diffusion Equations
- Solution Algorithm
- Performance Issues
- Conclusions and Future Work

Two Equation Model

- Based on Mousseau, Knoll, and Rider (LA-UR-99-4230)
- Photon Energy Equation

$$\frac{\partial E}{\partial t} - \nabla.(D_E \nabla E) = \sigma_a (T^4 - E)$$

- Material Energy Equation

$$\frac{\partial T}{\partial t} - \nabla.(D_T \nabla T) = -\sigma_a (T^4 - E)$$

- where

$$\sigma_a = \frac{z^3}{T^3}, D_E(T, E) = \frac{1}{3\sigma_a + \frac{1}{E}|\nabla E|}, D_T(T) = \kappa T^{\frac{5}{2}}$$

- Atomic number (z) depends on the location
- Solve the nonlinear equation, R = 0 at every time step

Space Discretization

- Linear P1 elements on triangles (2D) and tetrahedrons (3D) giving second order spatial accuracy
- Various ways to evaluate the diffusion coefficients
  - We express them in terms of basis functions

$$D_e = \sum_j^{n_e} D_j \phi_j \qquad \frac{1}{V_e}\int_{\Omega_e} D_e dV_e = \frac{1}{n_e}\sum_j^{n_e} D_j$$

  - Preserves the second order accuracy
  - Care is needed not to overestimate the diffusion coefficients

Time Discretization- BE

- Backward Euler

$$R_E^{n+1} = [M]\left\{\frac{E^{n+1} - E^n}{\Delta t^n}\right\} - \left\{\nabla \cdot (D_r \nabla E) - \sigma_a (T^4 - E)\right\}^{n+1}$$

with

$$\max\left(\frac{\left|E^{n+1} - E^n\right|}{\left|E^{n+1}\right|}\right) \leq \varepsilon$$

- May permit large time step size and nonlinear solver will not converge – cut time step size drastically when this happens
- For robustness, have minimum specified value of time step size

Time Discretization - CVODE

- General purpose ODE solver for stiff and nonstiff ODEs
- Uses variable order multi-step backward differentiation (BDF) methods
- Time step sizes are based on the error estimates

$$\|E_n\|_{rms,w} = \left[\sum_{i=1}^{N} \frac{1}{N} (\omega_i E_{n,i})^2 \right]^{1/2}$$

with

$$\omega_i = \frac{1}{rtol\,|y_i| + atol_i}$$

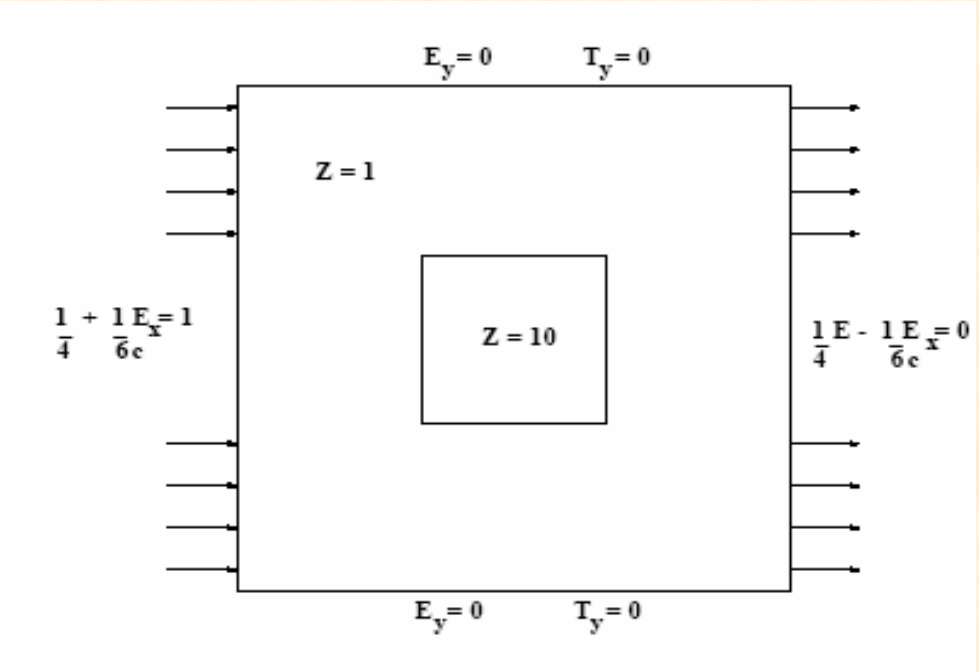
- We have used PETSc interface to CVODE

Time-Implicit Newton-Krylov-Schwarz

For nonlinear robustness, NKS iteration is wrapped in time-stepping:

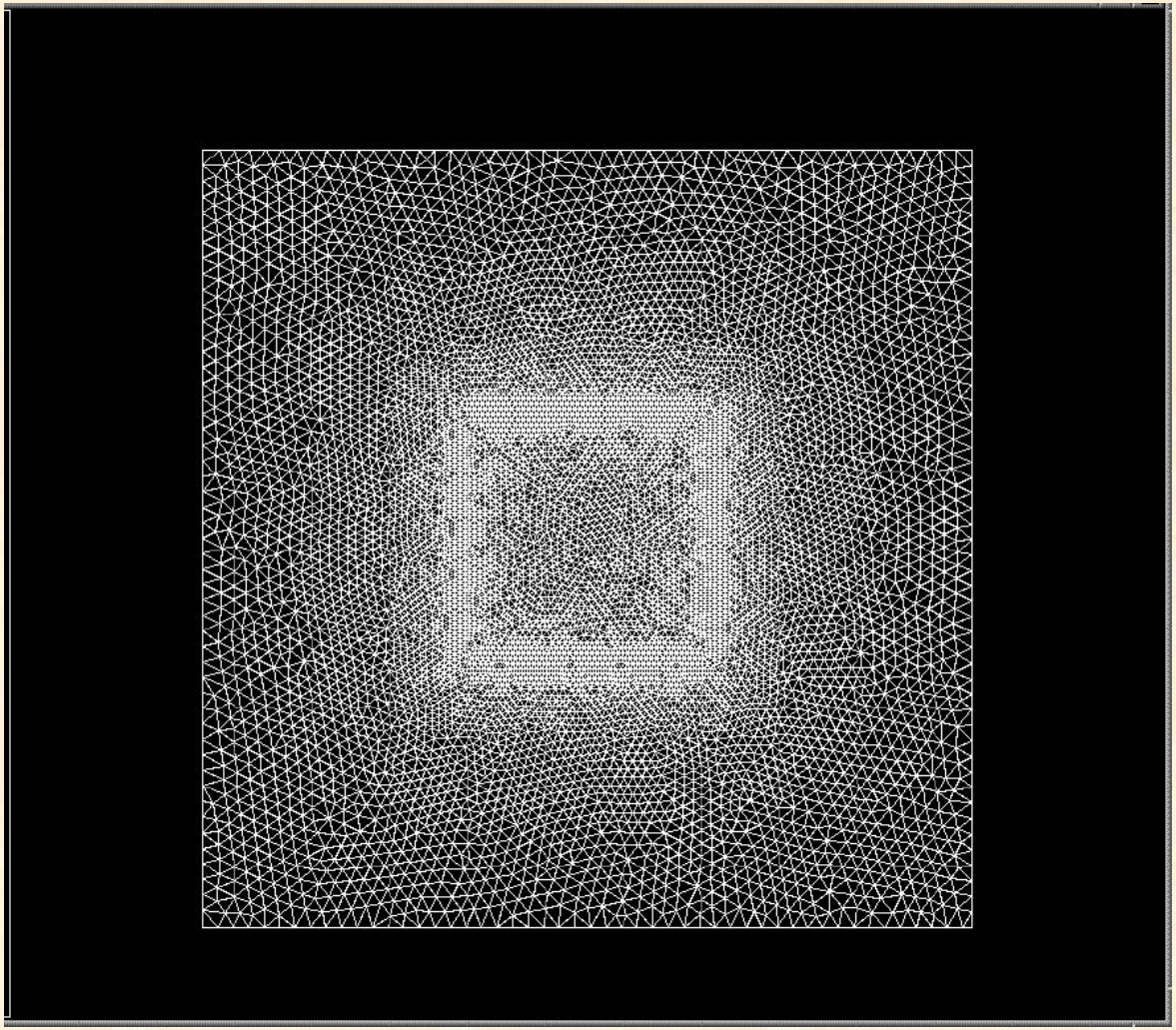
```
for (l = 0; l < n_time; l++) {                          # n_time ~ 2500
    select time step
    for (k = 0; k < n_Newton; k++) {                    # n_Newton ~ 5
        compute nonlinear residual and Jacobian
        for (j = 0; j < n_Krylov; j++) {                # n_Krylov ~ 20
            forall (i = 0; i < n_Precon ; i++) {
                solve subdomain problems concurrently
            } // End of loop over subdomains
            perform Jacobian-vector product
            enforce Krylov basis conditions
            update optimal coefficients
            check linear convergence
        } // End of linear solver
        perform DAXPY update
        check nonlinear convergence
    } // End of nonlinear loop
} // End of time-step loop
```

Sample 2D Test Problem



- Square region of inhomogeneous material
- Robin boundary condition (influx) applied at t = 0, and x=0 to the initially cold material

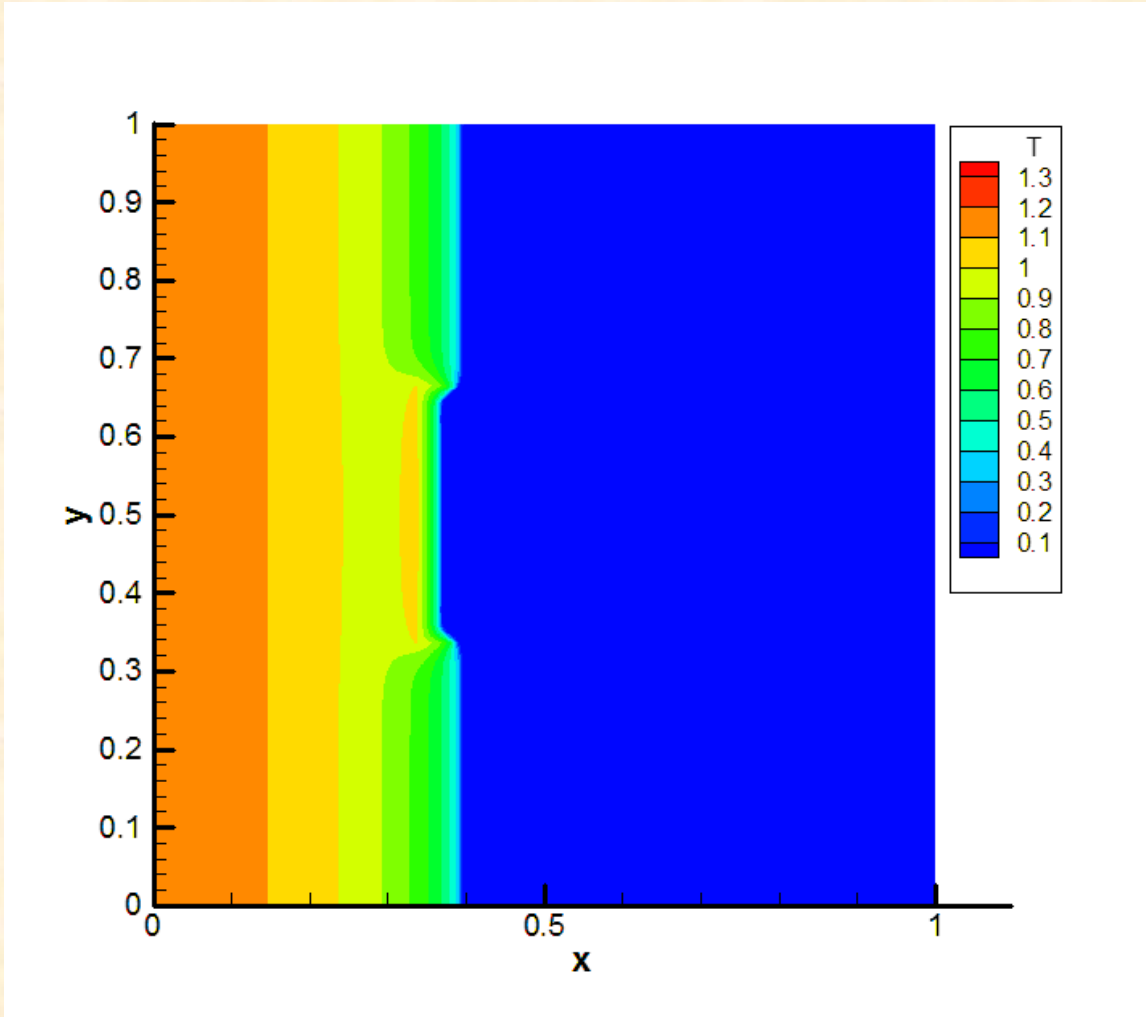
Grid for Marshak Wave Problem



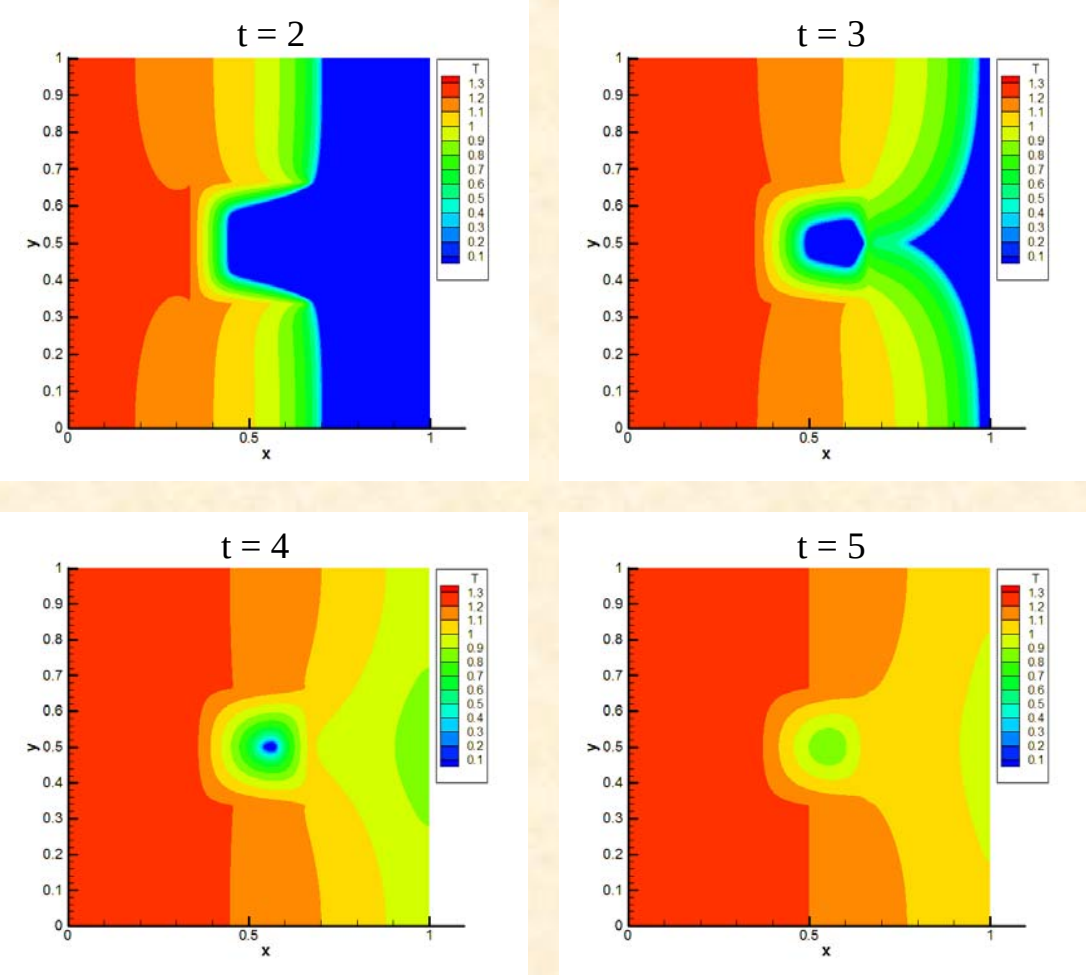
Algorithmic Setting

- The time integration is carried out from t=0 to t=5 using backward Euler (BE) or CVODE
  - Time step is cut in half if nonlinear convergence is not obtained in 50 iterations for BE
- Newton's Method solves the nonlinear problem at each time step with relative tolerance of 10^{-8}
  - Analytical Jacobian
    - matrix-free slower for this case when derived from the same Jacobian matrix
    - we need to evaluate preconditioners based on other approximations to Jacobian matrix
- GMRES (80) solves the linear system
  - Usually converges in less than 20 iterations with relative tolerance of 10^{-3}
- Block Jacobi with ILU(1) in each subdomain

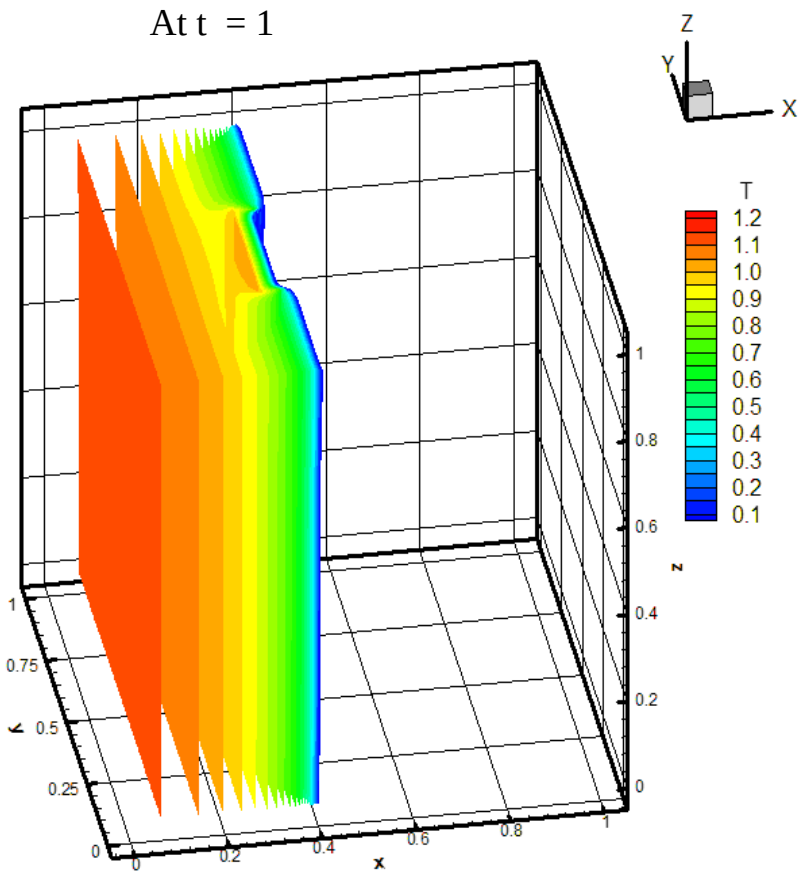
Material Temperature Contours at t = 1.0
2D Problem

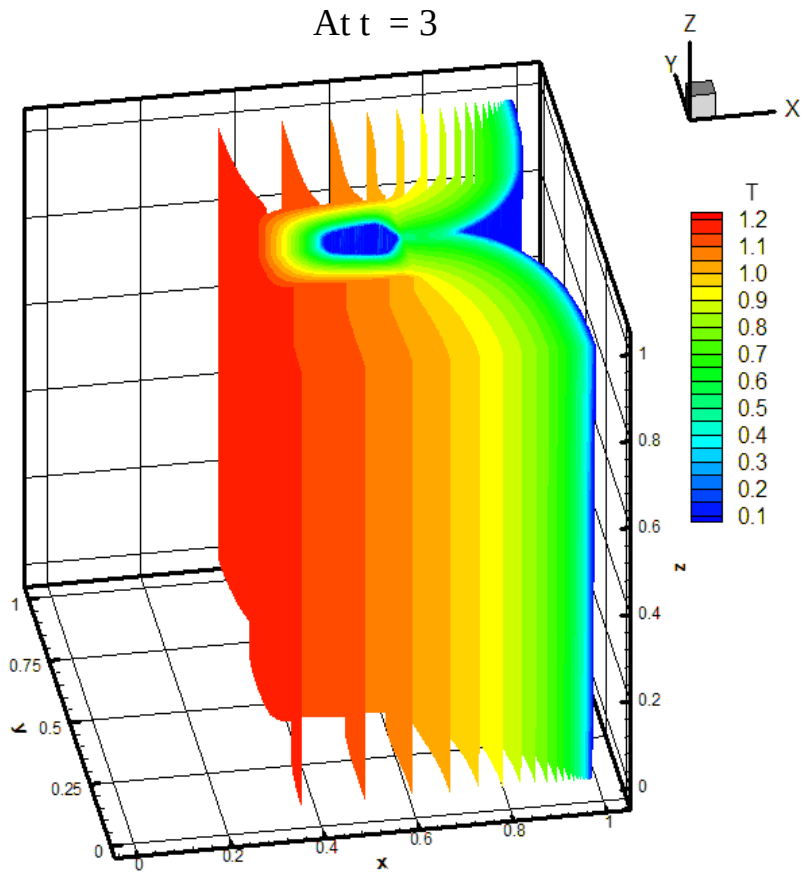


Material Temperature Contours
2D Problem

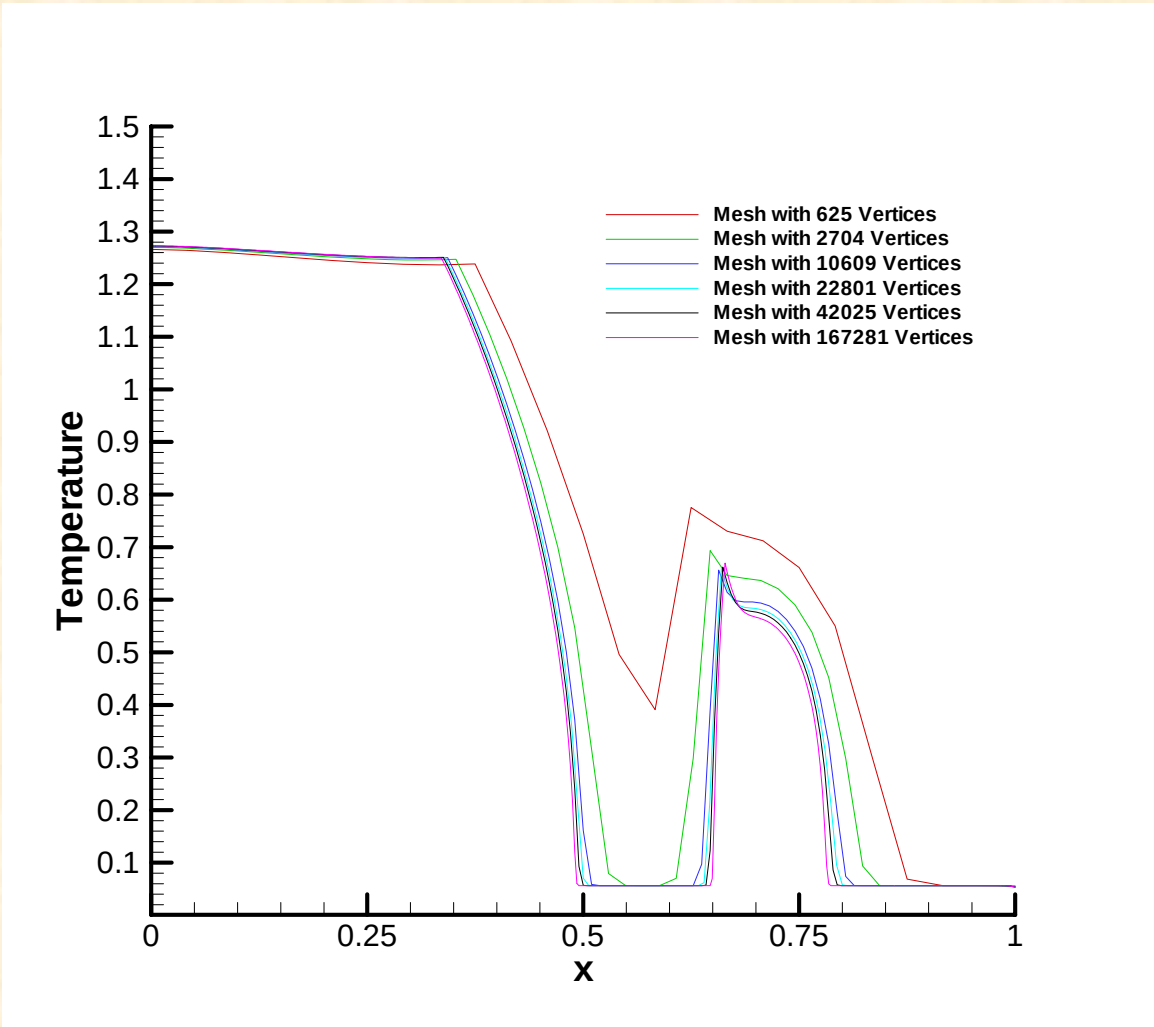


Material Temperature Contours for 3D Problem
Atomic number constant along z-axis
1,264,086 Elements and 237,160 Vertices

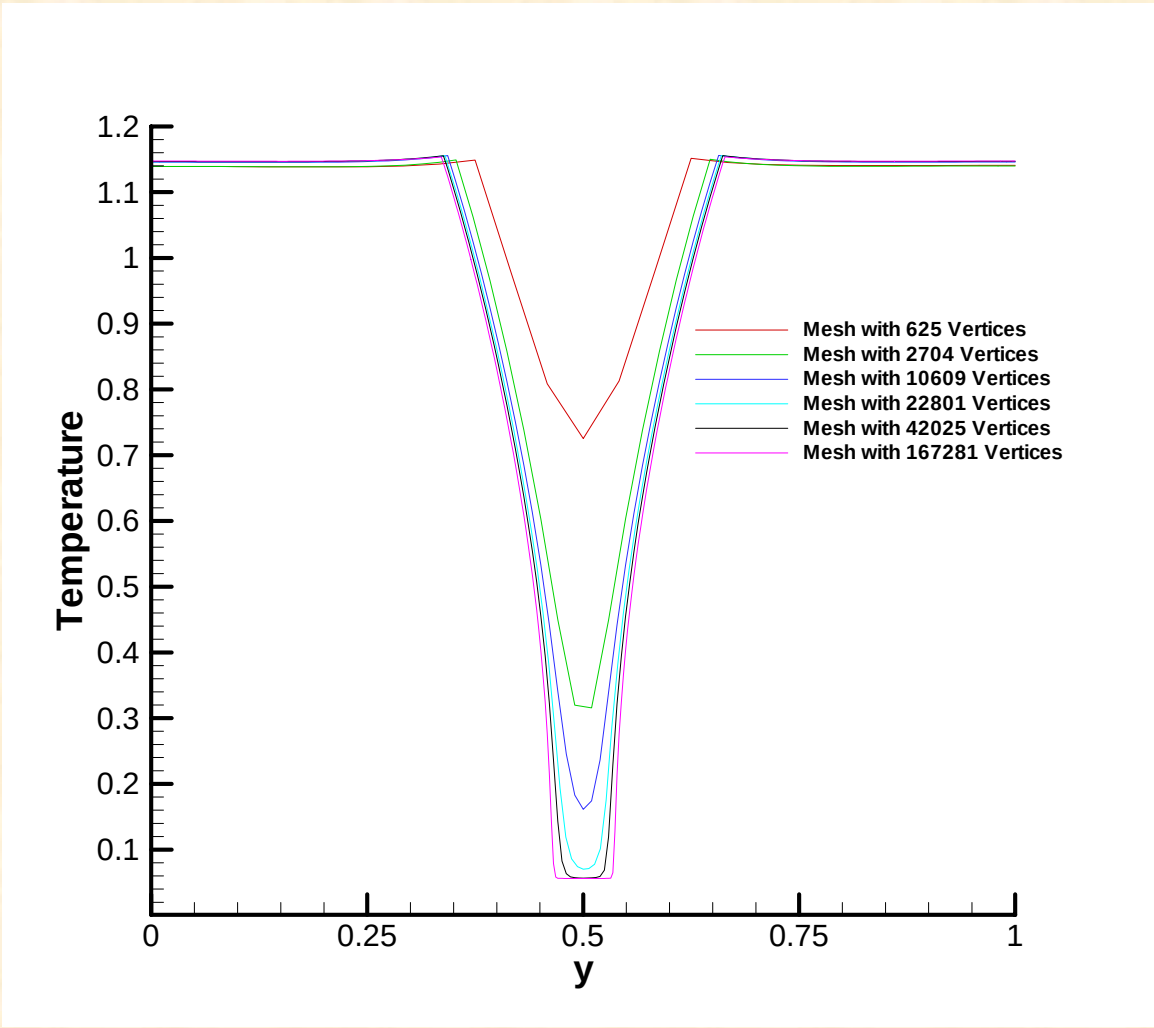




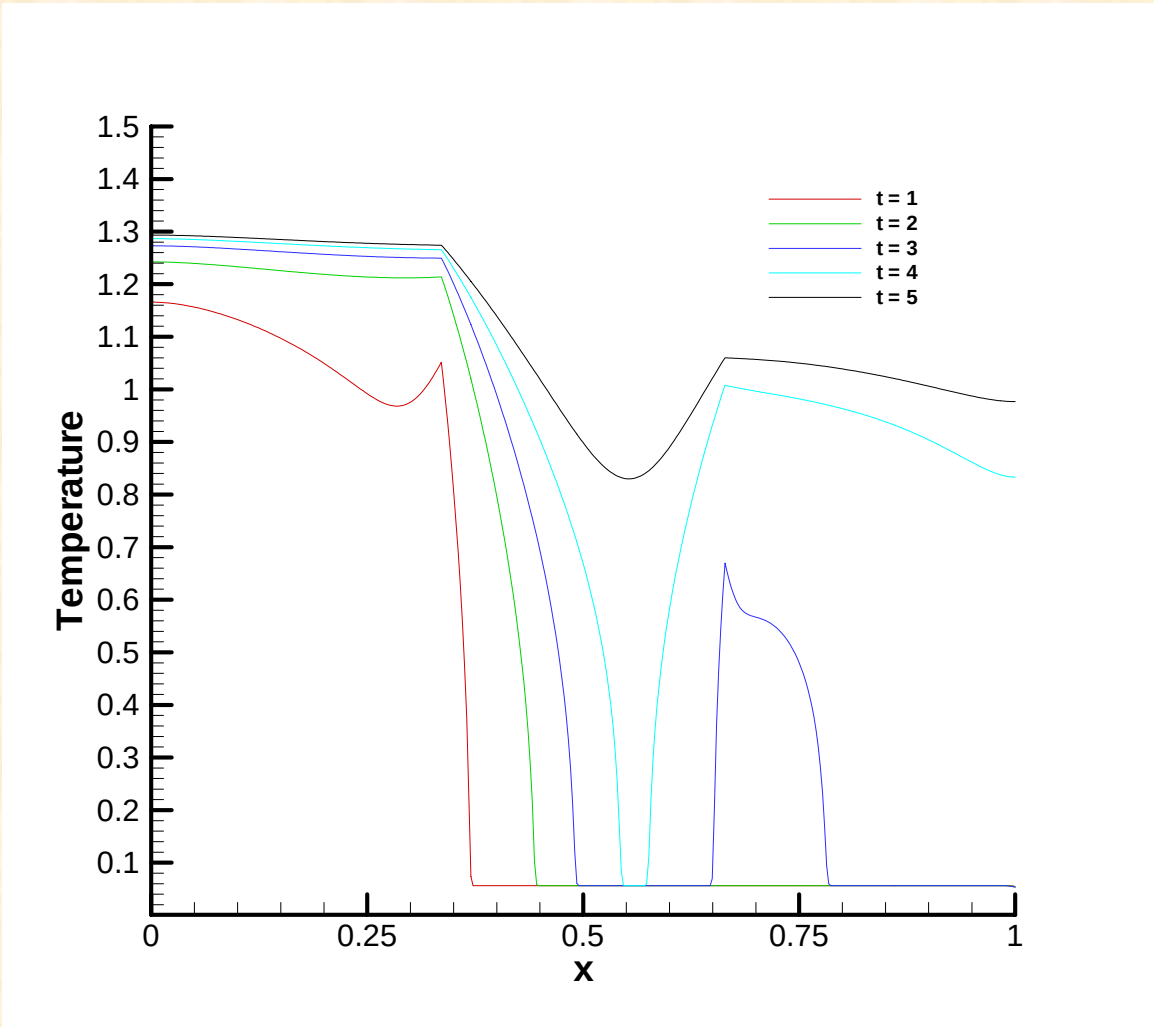
Mesh Independence (2D code)
Material Temperature Along y=0.5 at t=3



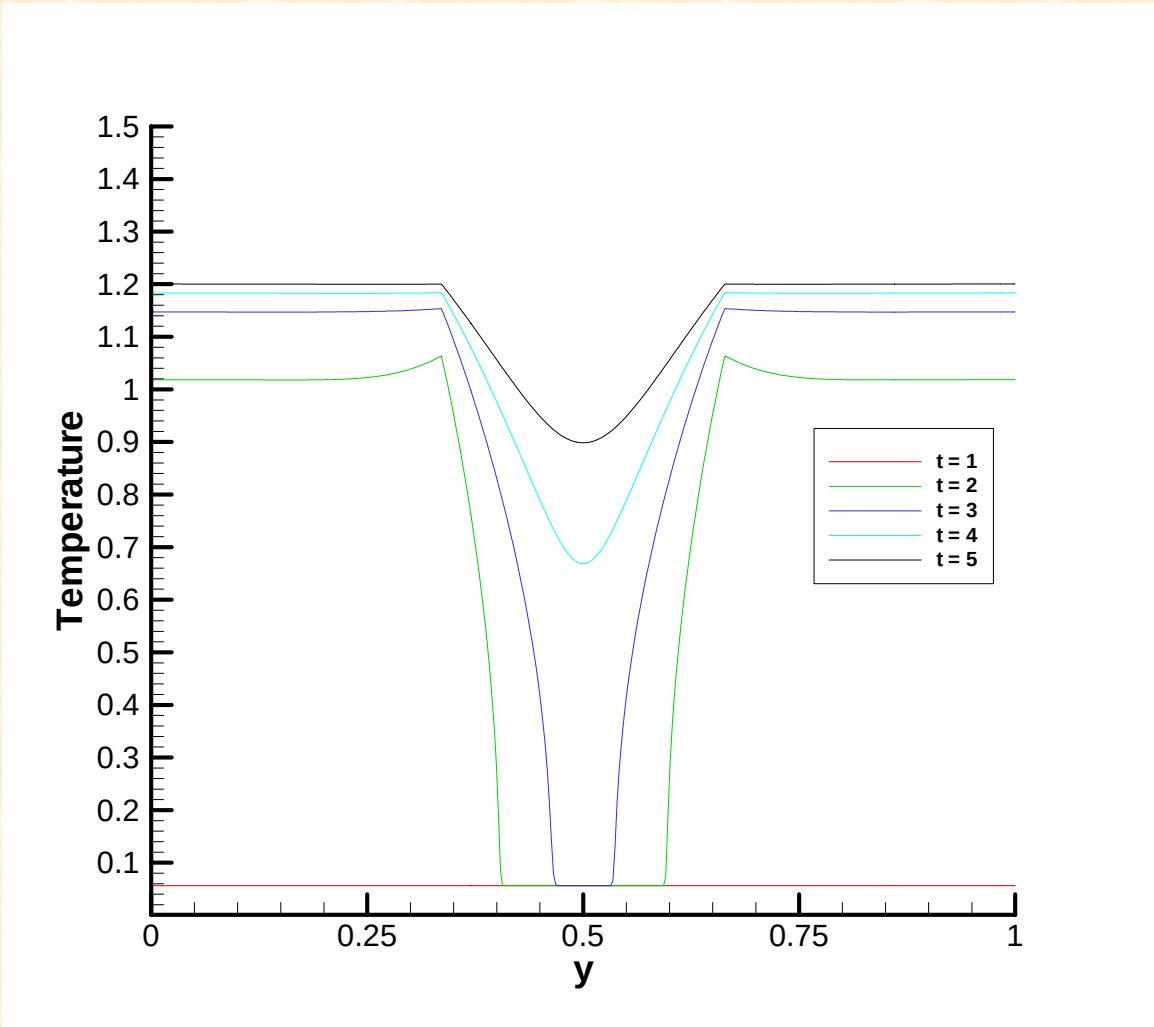
Mesh Independence (2D code)
Material Temperature Along x=0.5 at t=3



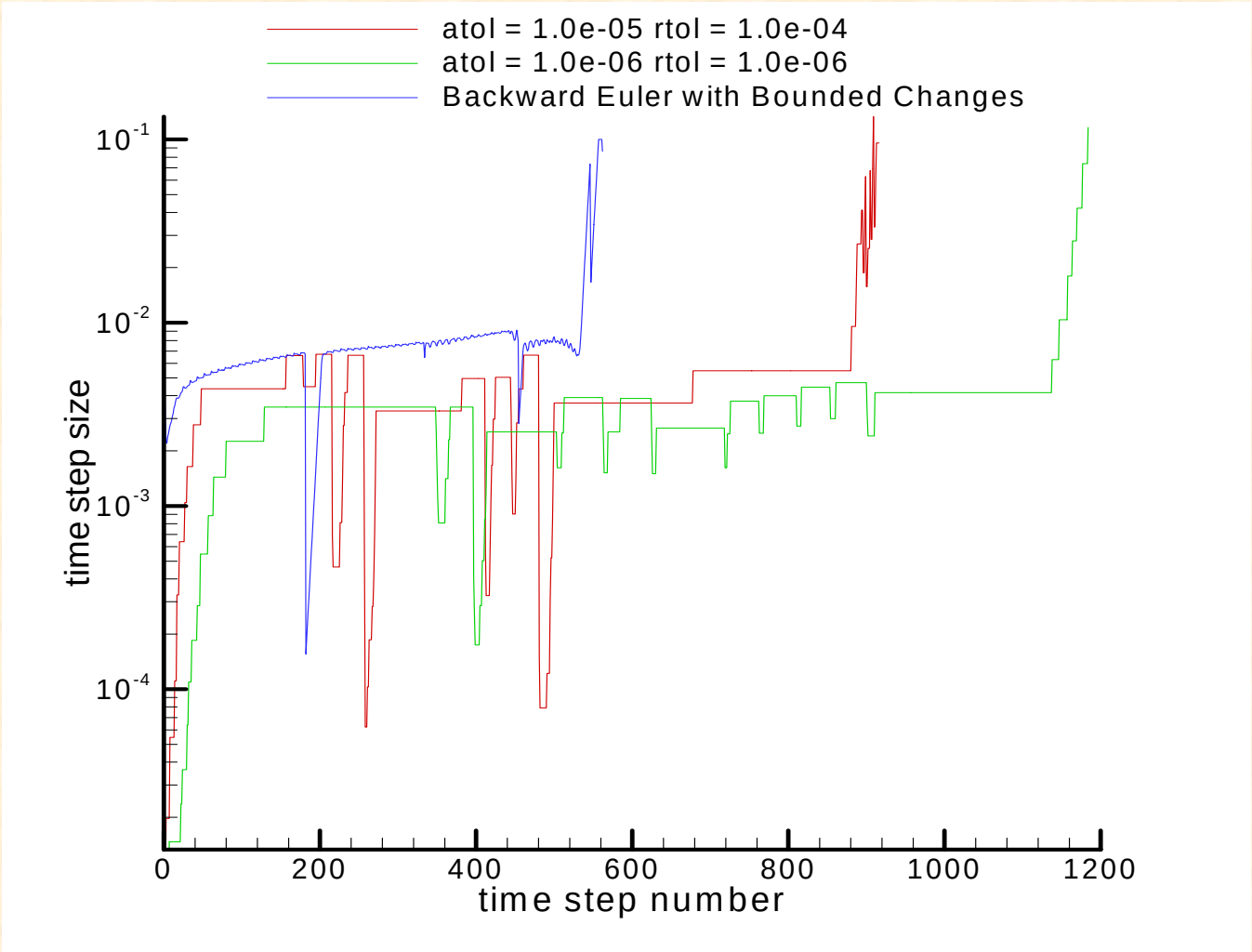
Time Evolution (2D code)
Material Temperature Along y = 0.5



Time Evolution (2D code)
Material Temperature Along x = 0.5



Time Integration Efficiency
2D Problem: 14,802 Elements and 7,502 Vertices



Rosenbrock Methods

- s-stage Rosenbrock method for

$$\frac{dE}{dt} = f(E)$$

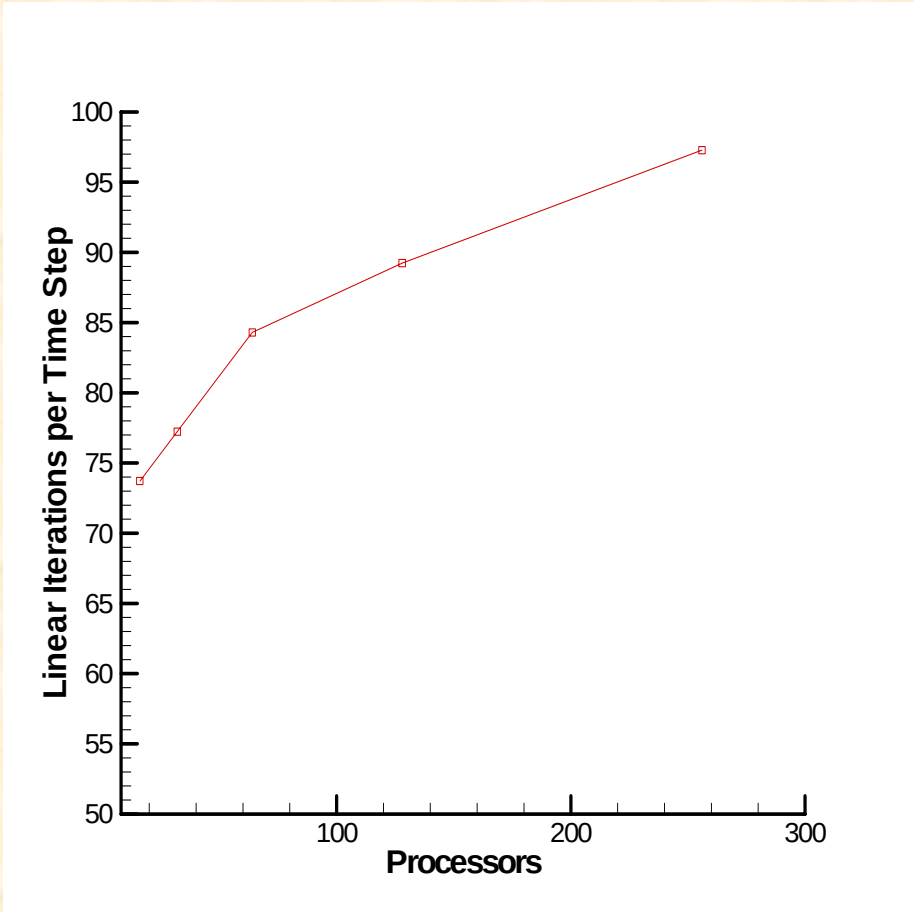
$$E^{n+1} = E^n + \sum_{i=1}^{s} b_i k_i$$

$$k_i = \Delta t^n f(E^n + \sum_{j=1}^{i-1} \alpha_{ij} k_j) + \Delta t^n J \sum_{j=1}^{i} \gamma_{ij} k_j$$

- Many studies find that Rosenbrock method can be significantly more efficient than BDF methods
  - Sandu, et al, “Benchmarking Stiff ODE Solvers for Atmospheric Chemistry Problems II: Rosenbrock Solvers”, Report No. 90/1996, University of Iowa
  - Bijl, et al., “Time Integration Schmes for the Unsteady Navier-Stokes Equations”, AIAA-2001-2612 (**ESDIRK**: Explicit first stage, Single diagonal coefficient, Diagonally Implicit Runge Kutta)

Algorithmic Performance
3D Mesh: 1,264,086 Elements and 237,160 Vertices

- Linear iteration count per time step rises slowly from 16 to 256 processors
- Nonlinear iteration count per time stays steady at about 6.3



Parallel Performance on TeraGrid

Dual 1.5 GHz Intel Madison Processors with 6 MB cache
3D Mesh: 1,264,086 Elements and 237,160 Vertices

Processors	Time in minutes	Speedup	Parallel Efficiency
16	196.9	1.00	1.00
32	97.1	2.01	1.00
64	45.6	4.32	1.08
128	22.3	7.74	1.26
256	13.3	14.8	0.93

Note that we have used both processor per node for this case

Implications

- Super linear speedup shows that more work needs to be done on optimizing the code for memory hierarchy
  - This will help when the per processor problem is relatively large (small number of processors)
- This is supported by the observation that this code achieves only a small fraction of machine peak

Primary PDE Solution Kernels

- Element-based loops
  - residual evaluation
  - approximate Jacobian evaluation
  - Jacobian-vector product (often replaced with matrix-free form, involving residual evaluation)
- Vertex-based loops
  - state vector and auxiliary vector updates
- Sparse, narrow-band recurrences
  - approximate factorization and back substitution
- Vector inner products and norms
  - orthogonalization/conjugation
  - convergence progress and stability checks

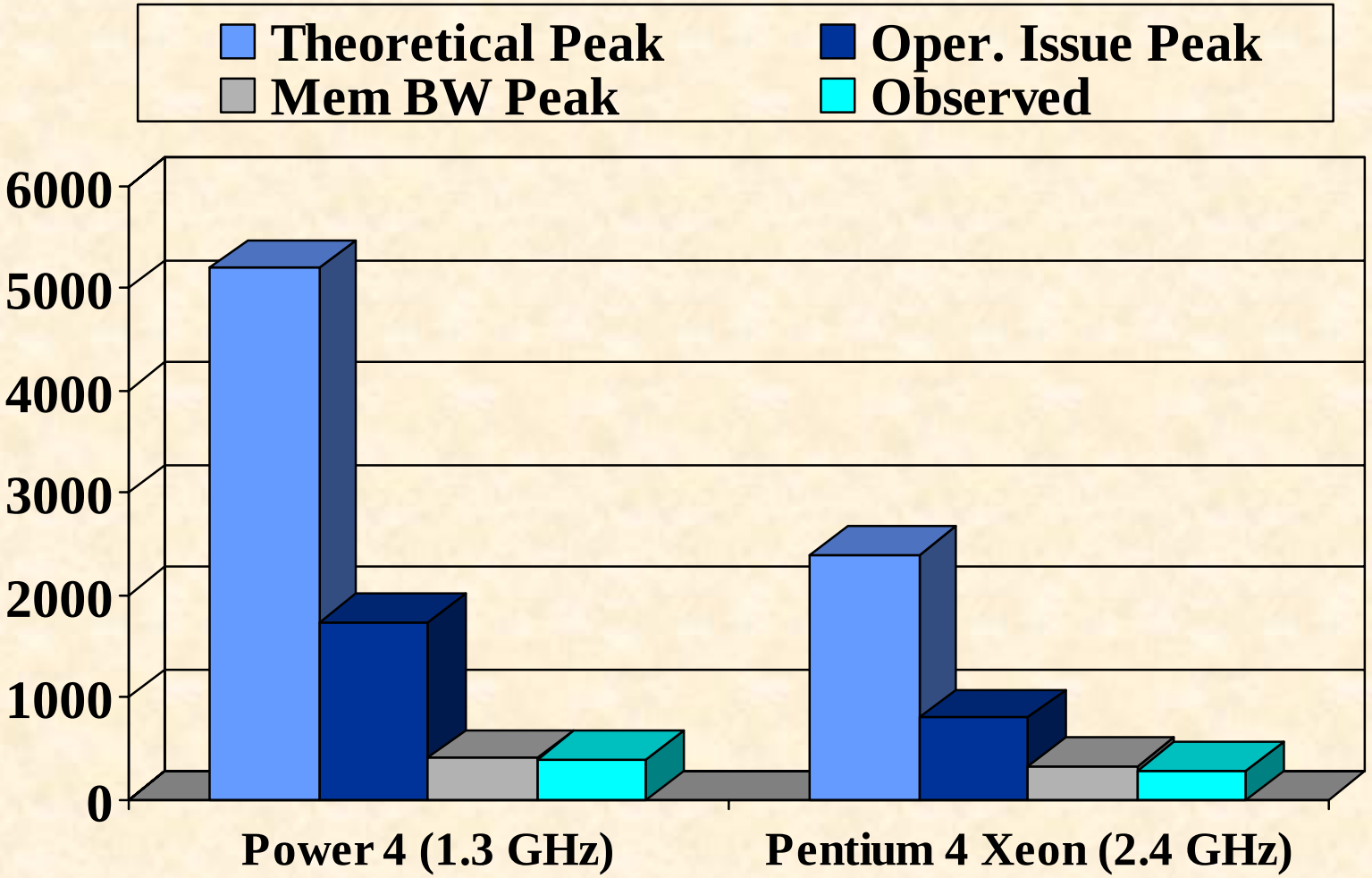
Three Fundamental Limiting Factors to Peak Performance

- Memory Bandwidth
  - Processor does not get data at the rate it requires
- Instruction Issue Rate
  - If the loops are load/store bound, we will not be able to do a floating point operation in every cycle even if the operands are available in primary cache
  - Several constraints (like primary cache latency, latency of floating point units etc.) are to be observed while coming up with an optimal schedule
- Fraction of Floating Point Operations
  - Not every instruction is a floating point instruction

Realistic Measures of Peak Performance

Sparse Matrix Vector Product

One vector, matrix size, m = 90,708, nonzero entries nz = 5,047,120



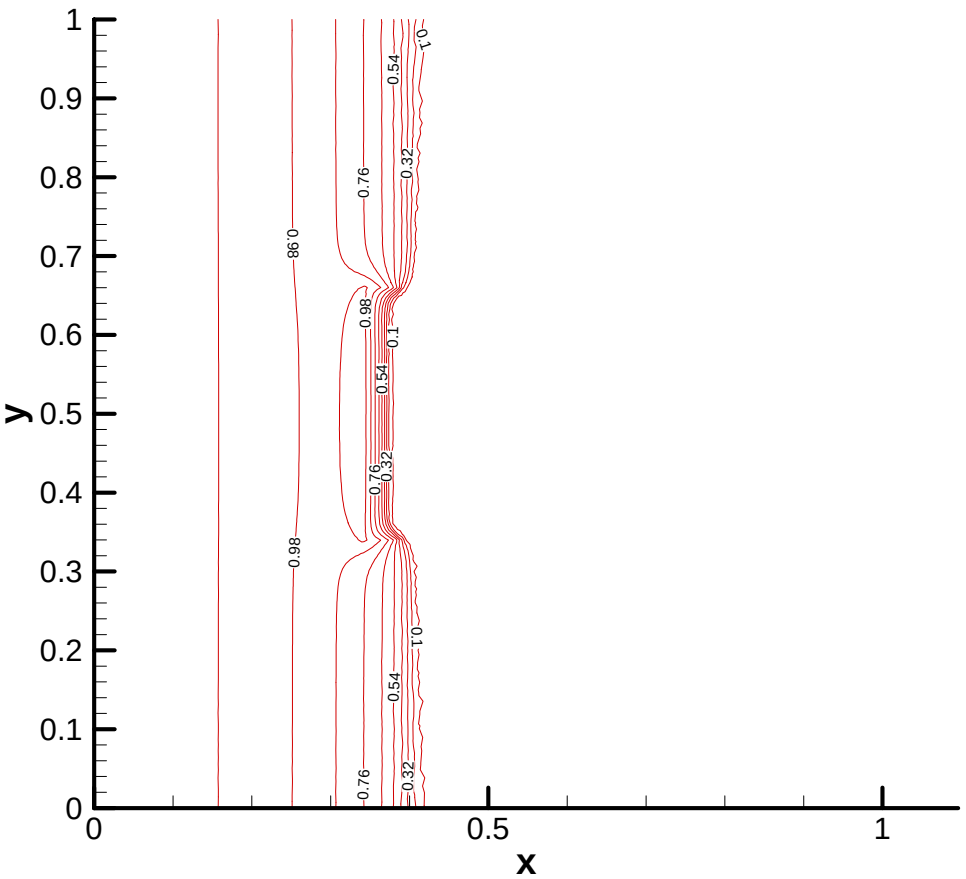
Conclusions and Future Work

- Efficiency and robustness of time integration methods is critical to solve RT problems in reasonable amount of time
  - Many recent studies show that Rosenbrock methods are more efficient than BDF for the stiff ODEs they addressed
  - We plan to investigate these methods in the context of RT problems in future
- More work is needed on optimizing the per-processor performance of this code
  - Most expensive phase is the Jacobian evaluation (~50-60 % of execution time)
  - Only 5% of the execution time is spent on sparse matrix vector products on 128 processors while it is about 20% on 16 processors

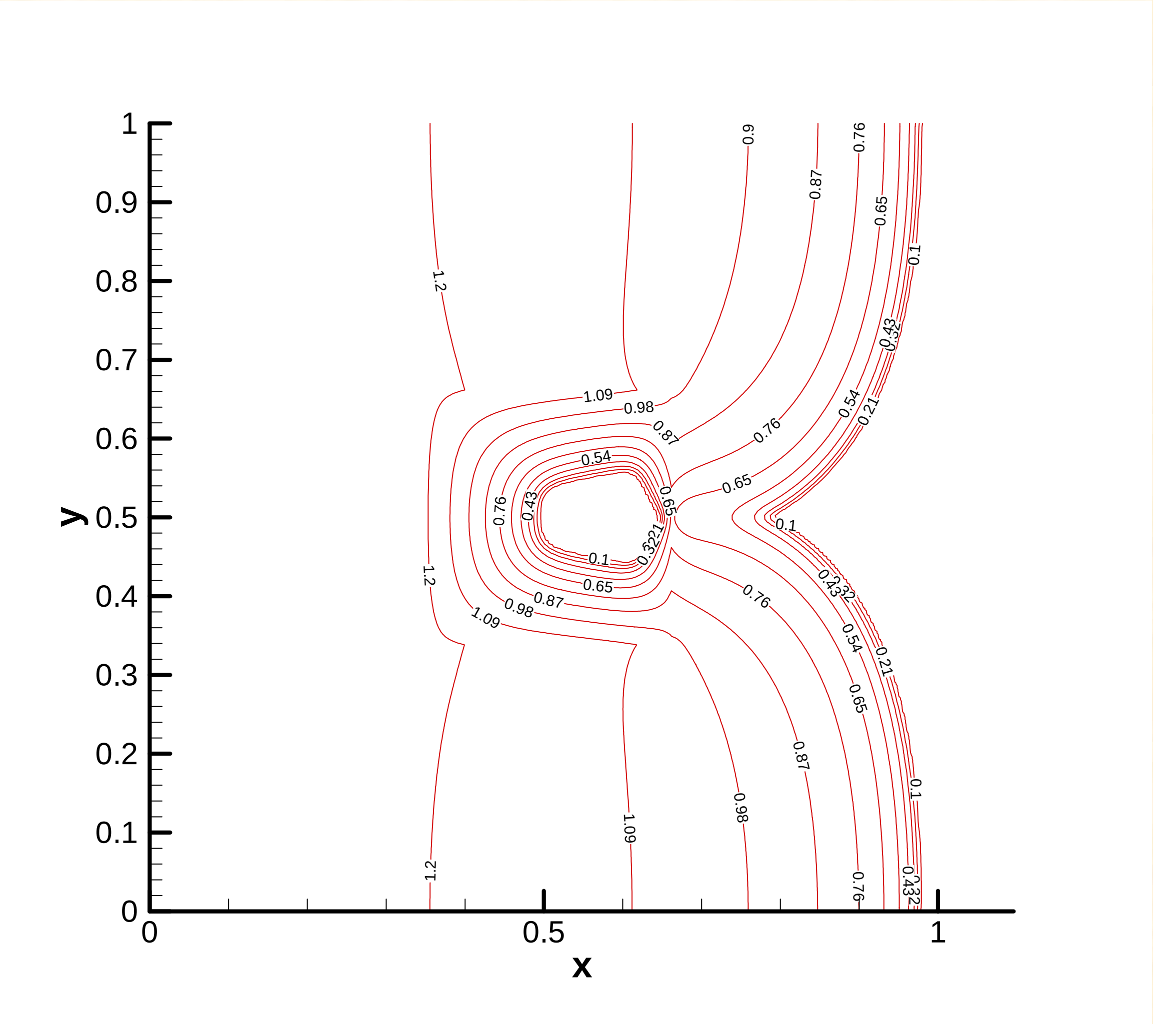
Acknowledgements

- Dimitri Mavriplis of University Of Wyoming for many helpful discussions
- DOE ASCI Project
- NSF for computer time on Teragrid
- Argonne National Laboratory for computer time on Jazz cluster

Material Temperature Contours at t = 1.0
2D Problem: 14,802 Elements and 7,502 Vertices



Material Temperature Contours at t = 3.0
2D Problem: 14,802 Elements and 7,502 Vertices



Material Temperature Contours for 2D Problem 14,802 Elements and 7,502 Vertices