3D Structured Adaptive Mesh Refinement and Multilevel Preconditioning for Non-Equilibrium Radiation Diffusion

Bobby Philip
Computational Engineering and Energy Sciences Group
Oak Ridge National Laboratory
ORNL-UTK Numerical Day

April 30, 2012

Collaborators

- Zhen Wang, ORNL
- Manuel Rodriguez, ORNL
- Mark Berrill, ORNL
- Michael Pernice, INL

Non-Equilibrium Radiation-Diffusion

- Coupled set of time dependent nonlinear diffusion equations for energy density and material temperature
- Applications:
  - Diffusion approximation to neutral particle transport
  - Astrophysics
  - Inertial confinement fusion
  - Atmospheric radiation
- Representative application for implicit time integration on dynamic adaptively refined grids in ORNL OLCF3 project

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned} \frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\ \frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \end{aligned}$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned}\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a(T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\ \frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a(T^4 - E) \qquad \text{in } \Omega = [0,1]^d\end{aligned}$$

Constitutive law: $\sigma_a = \frac{z^3}{T^3}$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned} \frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\ \frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \end{aligned}$$

Constitutive law: $\sigma_a = \frac{z^3}{T^3}$
Diffusion coefficients:

$$\begin{aligned} D_r &= \frac{1}{\left(3\sigma_a + \frac{\|\nabla E\|}{E}\right)} \\ D_t &= kT^{5/2} \end{aligned}$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned}
\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\
\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d
\end{aligned}$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned} \frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\ \frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \end{aligned}$$

Initial conditions:

$$E = E_0, \quad T = (E_0)^{1/4} \qquad \text{at } t = 0$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$
\begin{aligned}
\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\
\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d
\end{aligned}
$$

Initial conditions:

$$
E = E_0, \quad T = (E_0)^{1/4} \qquad \text{at } t = 0
$$

Boundary conditions:

$$
\begin{aligned}
\frac{1}{2} \mathrm{n} \cdot D_r \nabla E + \frac{E}{4} = R &\quad \text{on } \partial\Omega_{\mathcal{R}}, \; t \geq 0 \\
\mathrm{n} \cdot D_r \nabla E = 0 &\quad \text{on } \partial\Omega_{\mathcal{N}}, \; t \geq 0 \\
\mathrm{n} \cdot \nabla T = 0 &\quad \text{on } \partial\Omega, \; t \geq 0
\end{aligned}
$$

Previous Work

- Rider, Knoll and Olson (JQSRT, **63**, 1999; JCP, **152**, 1999) introduced the idea of physics based preconditioning in 1D
- Mousseau, Knoll, Rider (JCP, 2000) and Mousseau, Knoll (JCP, 2003) demonstrated effectiveness for 2D problems
- Mavriplis (JCP, **175**, 2002) compared Newton-Multigrid and FAS using agglomeration ideas on unstructured grids.
- Stals (ETNA, **15**, 2003), Newton-Multigrid and FAS, local refinement on *unstructured* grids for equilibrium radiation diffusion.
- Lowrie (JCP, 2004) compares different time integration methods for non-equilibrium radiation diffusion

Previous Work

- Brown, Shumaker, Woodward (JCP, 2005) consider fully implicit methods and high order time integration.
- Shestakov, Greenough, and Howell (JQSRT, 2005) consider pseudo-transient continuation on AMR grids using an alternative formulation.
- Glowinski, Toivanen (JCP, 2005) consider using automatic differentiation and system multigrid.
- Pernice, Philip (SISC, 2006), use JFNK with FAC preconditioners on AMR grids for equilibrium radiation-diffusion on SAMR grids.

Time Discretization: BDF2

$$\frac{1+2\alpha_n}{1+\alpha_n}\mathbf{u}^{n+1} - (1+\alpha_n)\mathbf{u}^n + \frac{\alpha_n^2}{1+\alpha_n}\mathbf{u}^{n-1} = \Delta t_n f(\mathbf{u}^{n+1})$$

Time Discretization: BDF2

$$\frac{1+2\alpha_n}{1+\alpha_n}\mathbf{u}^{n+1} - (1+\alpha_n)\mathbf{u}^n + \frac{\alpha_n^2}{1+\alpha_n}\mathbf{u}^{n-1} = \Delta t_n f(\mathbf{u}^{n+1})$$

with

$$\begin{aligned} \alpha_n &= \frac{\Delta t_n}{\Delta t_{n-1}} \\ \mathbf{u}^n &= \begin{pmatrix} E^n \\ T^n \end{pmatrix} \\ f(\mathbf{u}) &= \begin{pmatrix} \nabla\cdot(D_r\nabla E) + \sigma_a(T^4 - E) \\ \nabla\cdot(D_t\nabla T) - \sigma_a(T^4 - E) \end{pmatrix} \end{aligned}$$

Time Discretization: Choice of timestep

- First step: Backward Euler with user selected initial guess for timestep

Time Discretization: Choice of timestep

- First step: Backward Euler with user selected initial guess for timestep
- Subsequent steps
  - constant fixed timestep
  - constant final timestep
  - limiting relative change in energy
  - predictor-corrector with adaptive timestep selection

Time Discretization: Predictor-Corrector[1]

[1] ‘Incompressible Flow and the Finite Element Method, Volume 2, Isothermal Laminar Flow‘, Gresho and Sani

Time Discretization: Predictor-Corrector[1]

- Predict using generalized leapfrog:

$$\mathbf{u}_P^{n+1} = \mathbf{u}^n + (1+\alpha_n)\,\Delta t_n \dot{\mathbf{u}}^n - \alpha_n^2\left(\mathbf{u}^n - \mathbf{u}^{n-1}\right)$$

[1] 'Incompressible Flow and the Finite Element Method, Volume 2, Isothermal Laminar Flow', Gresho and Sani

Time Discretization: Predictor-Corrector[1]

- Predict using generalized leapfrog:

$$\mathbf{u}_P^{n+1} = \mathbf{u}^n + (1+\alpha_n)\,\Delta t_n \dot{\mathbf{u}}^n - \alpha_n^2\left(\mathbf{u}^n - \mathbf{u}^{n-1}\right)$$

- Solve for $\mathbf{u}^{n+1}$ using BDF2

$$\frac{1+2\alpha_n}{1+\alpha_n}\mathbf{u}^{n+1} - (1+\alpha_n)\mathbf{u}^n + \frac{\alpha_n^2}{1+\alpha_n}\mathbf{u}^{n-1} - \Delta t_n f(\mathbf{u}^{n+1}) = 0$$

[1] ‘Incompressible Flow and the Finite Element Method, Volume 2, Isothermal Laminar Flow‘, Gresho and Sani

Time Discretization: Predictor-Corrector

- Estimate truncation error

$$d^n \equiv \mathbf{u}^{n+1} - \mathbf{u}(t_{n+1}) \approx \frac{1}{(1+\alpha_n)\left(1+\left(\frac{\alpha_n}{1+\alpha_n}\right)^2\right)} \left(\mathbf{u}^{n+1} - \mathbf{u}_P^{n+1}\right)$$

Time Discretization: Predictor-Corrector

- Estimate truncation error

$$d^n \equiv \mathbf{u}^{n+1} - \mathbf{u}(t_{n+1}) \approx \frac{1}{(1+\alpha_n)\left(1+\left(\frac{\alpha_n}{1+\alpha_n}\right)^2\right)}\left(\mathbf{u}^{n+1} - \mathbf{u}_P^{n+1}\right)$$

- Compute time derivative approximation at current step

$$\dot{\mathbf{u}}^{n+1} = \frac{1}{\Delta t_n}\left(\frac{1+2\alpha_n}{1+\alpha_n}\mathbf{u}^{n+1} - (1+\alpha_n)\mathbf{u}^n + \frac{\alpha_n^2}{1+\alpha_n}\mathbf{u}^{n-1}\right)$$

Time Discretization: Predictor-Corrector

- Estimate truncation error

$$d^n \equiv \mathbf{u}^{n+1} - \mathbf{u}(t_{n+1}) \approx \frac{1}{(1+\alpha_n)\left(1+\left(\frac{\alpha_n}{1+\alpha_n}\right)^2\right)}\left(\mathbf{u}^{n+1} - \mathbf{u}_P^{n+1}\right)$$

- Compute time derivative approximation at current step

$$\dot{\mathbf{u}}^{n+1} = \frac{1}{\Delta t_n}\left(\frac{1+2\alpha_n}{1+\alpha_n}\mathbf{u}^{n+1} - (1+\alpha_n)\mathbf{u}^n + \frac{\alpha_n^2}{1+\alpha_n}\mathbf{u}^{n-1}\right)$$

Note: Using $f(\mathbf{u}^{n+1})$ to approximate $\dot{\mathbf{u}}^{n+1}$ accumulates round off error, is costly, and leads to smaller timesteps

Time Discretization: Predictor-Corrector

- Estimate truncation error

$$d^n \equiv \mathbf{u}^{n+1} - \mathbf{u}(t_{n+1}) \approx \frac{1}{(1+\alpha_n)\left(1+\left(\frac{\alpha_n}{1+\alpha_n}\right)^2\right)}\left(\mathbf{u}^{n+1} - \mathbf{u}_P^{n+1}\right)$$

- Compute time derivative approximation at current step

$$\dot{\mathbf{u}}^{n+1} = \frac{1}{\Delta t_n}\left(\frac{1+2\alpha_n}{1+\alpha_n}\mathbf{u}^{n+1} - (1+\alpha_n)\mathbf{u}^n + \frac{\alpha_n^2}{1+\alpha_n}\mathbf{u}^{n-1}\right)$$

Note: Using $f(\mathbf{u}^{n+1})$ to approximate $\dot{\mathbf{u}}^{n+1}$ accumulates round off error, is costly, and leads to smaller timesteps

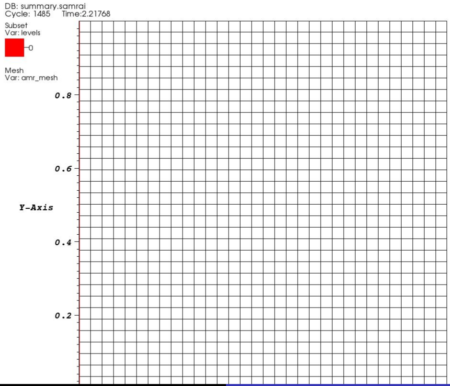
- $\Delta t_{n+1} = \Delta t_n \left(\frac{\epsilon_r ||\mathbf{u}^{n+1}|| + \epsilon_a}{||d_n||}\right)^{1/3}$

Space Discretization:

- Cell Centered Finite Volume Discretization for E, T
- Cell centered diffusion coefficients averaged to cell faces
- Fluxes are computed at cell faces
- Material discontinuities are aligned with cell faces for simplicity
- Linear interpolation at coarse-fine interfaces to provide centered ghost cell data
- Coarse-fine interpolation is programming intensive to account for all special cases

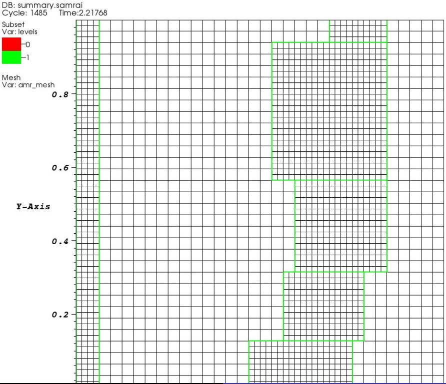
Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



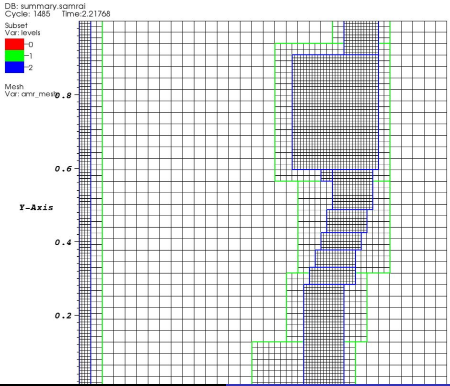
Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



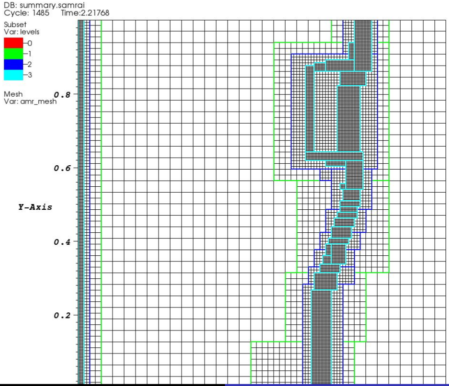
Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



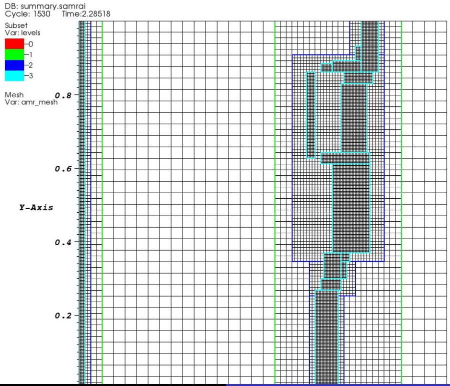
Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



Nonlinear systems

Implicit time discretizations lead to a nonlinear system of equations that needs to be solved at each timestep

$$F(\mathbf{u}^{n+1}) = 0$$

where

$$F(\mathbf{u}^{n+1}) \equiv \frac{1+2\alpha_n}{1+\alpha_n}\mathbf{u}^{n+1} - (1+\alpha_n)\mathbf{u}^n + \frac{\alpha_n^2}{1+\alpha_n}\mathbf{u}^{n-1} - \Delta t_n f(\mathbf{u}^{n+1})$$

with $\mathbf{u}^{n+1}$ a cell centered vector over an AMR mesh.

Inexact Newton Methods

- Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and consider solving $F(\mathbf{u}) = 0$.

Inexact Newton Methods

- Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and consider solving $F(\mathbf{u}) = 0$.
- The k^{th} step of classical Newton's method requires solution of the Newton equations:

$$F'(\mathbf{u}_k)\mathbf{s}_k = -F(\mathbf{u}_k).$$

Inexact Newton Methods

- Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and consider solving $F(\mathbf{u}) = 0$.
- The k^{th} step of classical Newton's method requires solution of the Newton equations:

$$F'(\mathbf{u}_k)\mathbf{s}_k = -F(\mathbf{u}_k).$$

- With *inexact Newton methods*, we only require

$$\|F(\mathbf{u}_k) + F'(\mathbf{u}_k)\mathbf{s}_k\| \leq \eta_k \|F(\mathbf{u}_k)\|, \qquad \eta_k > 0.$$

This can be done with any iterative method.

Iterative Linear Solvers

- System multigrid could be used directly, or
- Krylov subspace methods - need Jacobian-vector products, which can be approximated by

$$F'(\mathbf{u}_k)\mathbf{v} \approx \frac{F(\mathbf{u}_k + \varepsilon \mathbf{v}) - F(\mathbf{u}_k)}{\varepsilon}, \qquad \varepsilon \approx \mathcal{O}(\sqrt{\epsilon_{\text{mach}}}).$$

- The resulting *Jacobian-free Newton-Krylov* (JFNK) method is easier to implement because only function evaluation and preconditioning setup/apply is required.
- ε must take into account accuracy, efficiency, and non-negativity considerations

Preconditioned Krylov Methods

- Right-preconditioning of the Newton equations is used, i.e., we solve

$$(J(\mathbf{u}_k)P^{-1})P\mathbf{s}_k = -F(\mathbf{u}_k).$$

where P is the preconditioner.

Preconditioned Krylov Methods

- Right-preconditioning of the Newton equations is used, i.e., we solve

$$(J(\mathbf{u}_k)P^{-1})P\mathbf{s}_k = -F(\mathbf{u}_k).$$

  where P is the preconditioner.

- For JFNK this requires the Jacobian-vector products:

$$(J(\mathbf{u}_k)P^{-1})\mathbf{v} \approx \frac{F(\mathbf{u}_k + \varepsilon P^{-1}\mathbf{v}) - F(\mathbf{u}_k)}{\varepsilon}.$$

Preconditioned Krylov Methods

- Right-preconditioning of the Newton equations is used, i.e., we solve
$$(J(\mathbf{u}_k)P^{-1})P\mathbf{s}_k = -F(\mathbf{u}_k).$$
where P is the preconditioner.
- For JFNK this requires the Jacobian-vector products:
$$(J(\mathbf{u}_k)P^{-1})\mathbf{v} \approx \frac{F(\mathbf{u}_k + \varepsilon P^{-1}\mathbf{v}) - F(\mathbf{u}_k)}{\varepsilon}.$$
- The approximate Jacobian-vector is computed in two steps:
  - Solve $\mathbf{y} = P^{-1}\mathbf{v}$ approximately
  - Compute $\frac{F(\mathbf{u}_k+\varepsilon\mathbf{y})-F(\mathbf{u}_k)}{\varepsilon}$.

Linear Systems

JFNK allows us to focus on developing effective preconditioners. The Jacobian systems at each Newton step are of the form:

$$\mathcal{L}\begin{pmatrix} \delta E \\ \delta T \end{pmatrix} = \begin{pmatrix} -r_E \\ -r_T \end{pmatrix}$$

where

$$\mathcal{L} \approx \begin{pmatrix} \frac{I}{\Delta t} - \nabla \cdot D_r^k \nabla + \sigma_a I & -\sigma_a (T^k)^3 \\ -\sigma_a I & \frac{I}{\Delta t} - \nabla \cdot D_t^k \nabla + \sigma_a (T^k)^3 \end{pmatrix}$$

Operator Split Preconditioner

We use a splitting of the form shown in our preconditioner

$$\mathcal{L} \approx \mathcal{P}_1 \mathcal{P}_2$$

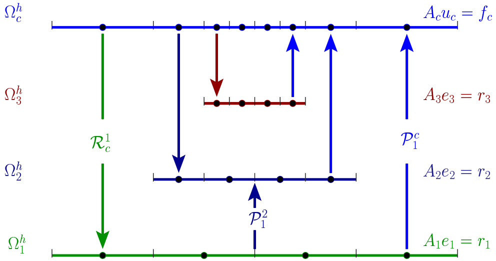
where

$$\mathcal{P}_1 = \begin{pmatrix} \frac{I}{\Delta t} - \nabla \cdot D_r^k \nabla & 0 \\ 0 & \frac{I}{\Delta t} - \nabla \cdot D_t^k \nabla \end{pmatrix}$$

and

$$\mathcal{P}_2 = \begin{pmatrix} (1 + \Delta t \sigma_a) I & -\Delta t \sigma_a (T^k)^3 \\ -\Delta t \sigma_a I & I + \Delta t \sigma_a (T^k)^3 \end{pmatrix}$$

Grids and Notation for FAC



$$r_k = \mathcal{R}_c^k(f_c - A_c u_c), \quad k = 1, 2, 3$$

Preconditioner: FAC

Compute composite grid residual $r_c = f_c - \mathcal{A}_c u_c$.

$u_c = 0$

For $k = J, \ldots 2$ {

  Smooth : $\mathcal{A}_k e_k = r_k$

  Correct: $u_c \leftarrow u_c + \mathcal{P}_k^c e_k$.

  Update : $r_{k-1} = \mathcal{R}_c^{k-1}(f_c - \mathcal{A}_c u_c)$

  }

Solve: $\mathcal{A}_1 e_1 = r_1$

For $k = 2, \ldots J$ {

  Correct: $u_c \leftarrow u_c + \mathcal{P}_{k-1}^c e_{k-1}$

  Update : $r_k = \mathcal{R}_c^k(f_c - \mathcal{A}_c u_c)$

  Smooth : $\mathcal{A}_k e_k = r_k$

  }

AMR Regridding: Data Transfer

- Error estimation based on gradient and curvature estimators
- Berger-Rigoutsos algorithm to determine new patches
- Linear interpolation of data from old to new grid hierarchy
- Old grid solution not a solution on new grid hierarchy
- Higher order interpolation helps minimize this[2]
- High solution gradients prevent high order interpolation
- Leads to high nonlinear residuals after regridding

[2] Philip, Chacon, Pernice, JCP, 2008

Regridding: Time Integration

- Linear interpolation causes jump in time derivative
- Warm restart to minimize timestep changes
- Col restart results in time step cuts
- Resolve at existing step to minimize perturbation to solution
- Regrid can lead to non-positive values

Simulation Software

- SAMRAI package for AMR
- PETSc SNES package for inexact Newton
- PETSc Krylov solver - GMRES
- MLSolvers package for multilevel preconditioners and operators - FAC, AFACx, MDS
- NRDF application code with implicit time integration

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d$$
$$\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) = -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned}
\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\
\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d
\end{aligned}$$

Constitutive law: $\sigma_a = \frac{z^3}{T^3}$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned} \frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\ \frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \end{aligned}$$

Constitutive law: $\sigma_a = \frac{z^3}{T^3}$

Diffusion coefficients:

$$\begin{aligned} D_r &= \frac{1}{\left(3\sigma_a + \frac{\|\nabla E\|}{E}\right)} \\ D_t &= 0.1\, T^{5/2} \end{aligned}$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned} \frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\ \frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \end{aligned}$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$
\begin{aligned}
\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\
\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d
\end{aligned}
$$

Initial conditions:

$$
E = 1.0e-05, \quad T = (1.0e-05)^{1/4} \qquad \text{at } t = 0
$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\begin{aligned}
\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d \\
\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E) \qquad \text{in } \Omega = [0,1]^d
\end{aligned}$$

Initial conditions:

$$E = 1.0e-05, \quad T = (1.0e-05)^{1/4} \qquad \text{at } t = 0$$

Boundary conditions:

$$\begin{aligned}
\frac{1}{2}\mathrm{n} \cdot D_r \nabla E + \frac{E}{4} &= 1 \qquad \text{at } x = 0,\ t \geq 0 \\
\frac{1}{2}\mathrm{n} \cdot D_r \nabla E + \frac{E}{4} &= 0 \qquad \text{at } x = 1,\ t \geq 0 \\
\mathrm{n} \cdot D_r \nabla E &= 0 \qquad \text{on } \partial\Omega_{\mathcal{N}},\ t \geq 0
\end{aligned}$$

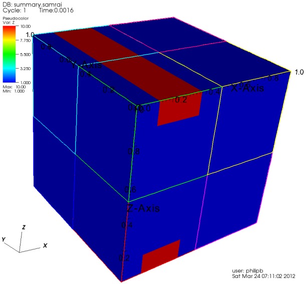
Solver parameters

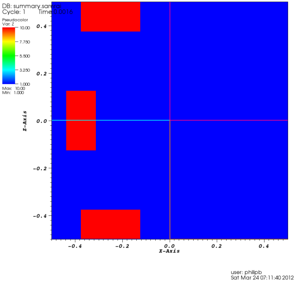
- timestepper absolute tolerance: $1.0e-7$
- timestepper relative tolerance: $1.0e-7$
- nonlinear solver absolute tolerance: $1.0e-15$
- nonlinear solver relative tolerance: $1.0e-12$
- step tolerance: $1.0e-10$
- forcing term: $\eta_k = 0.01$
- max. gmres subspace dimension: 50
- max. linear iterations: 100
- FAC V-cycle, R-B Gauss Seidel (2 pre and post smooths)
- final time: 1.0

AMR parameters

- refinement ratio: 2
- combine efficiency: 0.85
- error tagging: based on curvature and gradient of E
- Berger-Rigoutsos regridding algorithm
- linear interpolation at coarse-fine boundaries
- volume linear interpolation during regridding

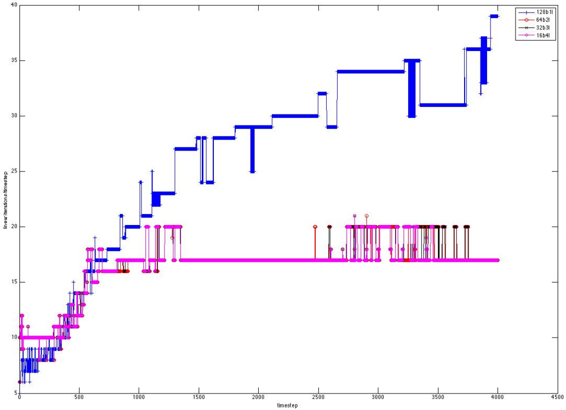
Numerics: Material Properties



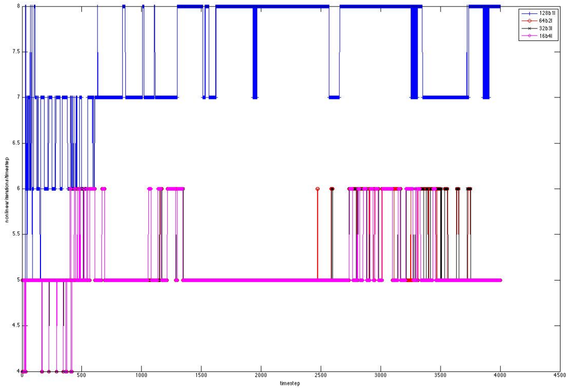


Movies

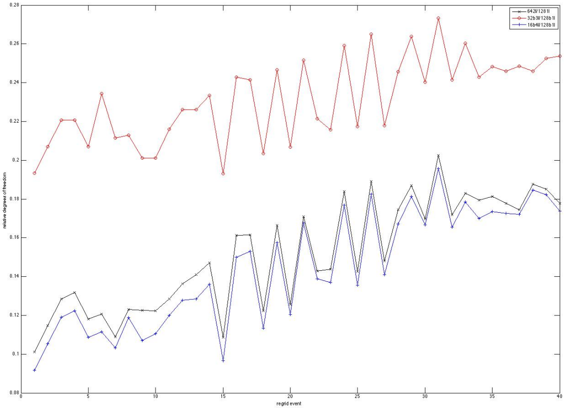
Performance: Linear Iterations



Performance: Nonlinear Iterations



Performance: Degrees of Freedom



Conclusions and Future Work

- Conclusions
  - Developed an efficient solver for non-equilibrium radiation diffusion on AMR grids
- Future work/Possible improvements
  - Better discretizations for AMR grids for problems with discontinous coefficients
  - Improved performance of preconditioners
  - Error estimation for finite volume discretizations
  - AMR Grid alignment
  - GPU acceleration (ongoing work)
  - Massively parallel simulations (ongoing work)