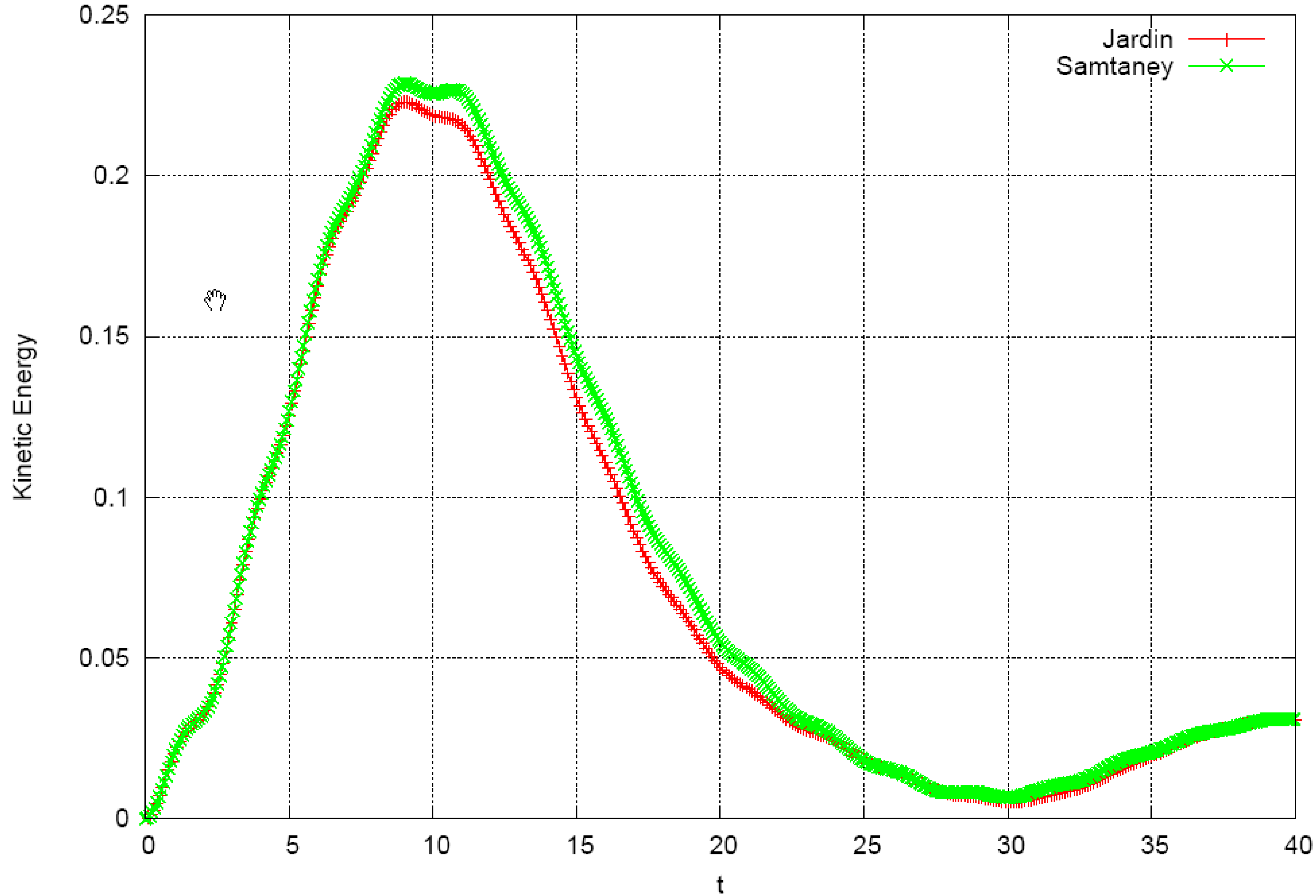
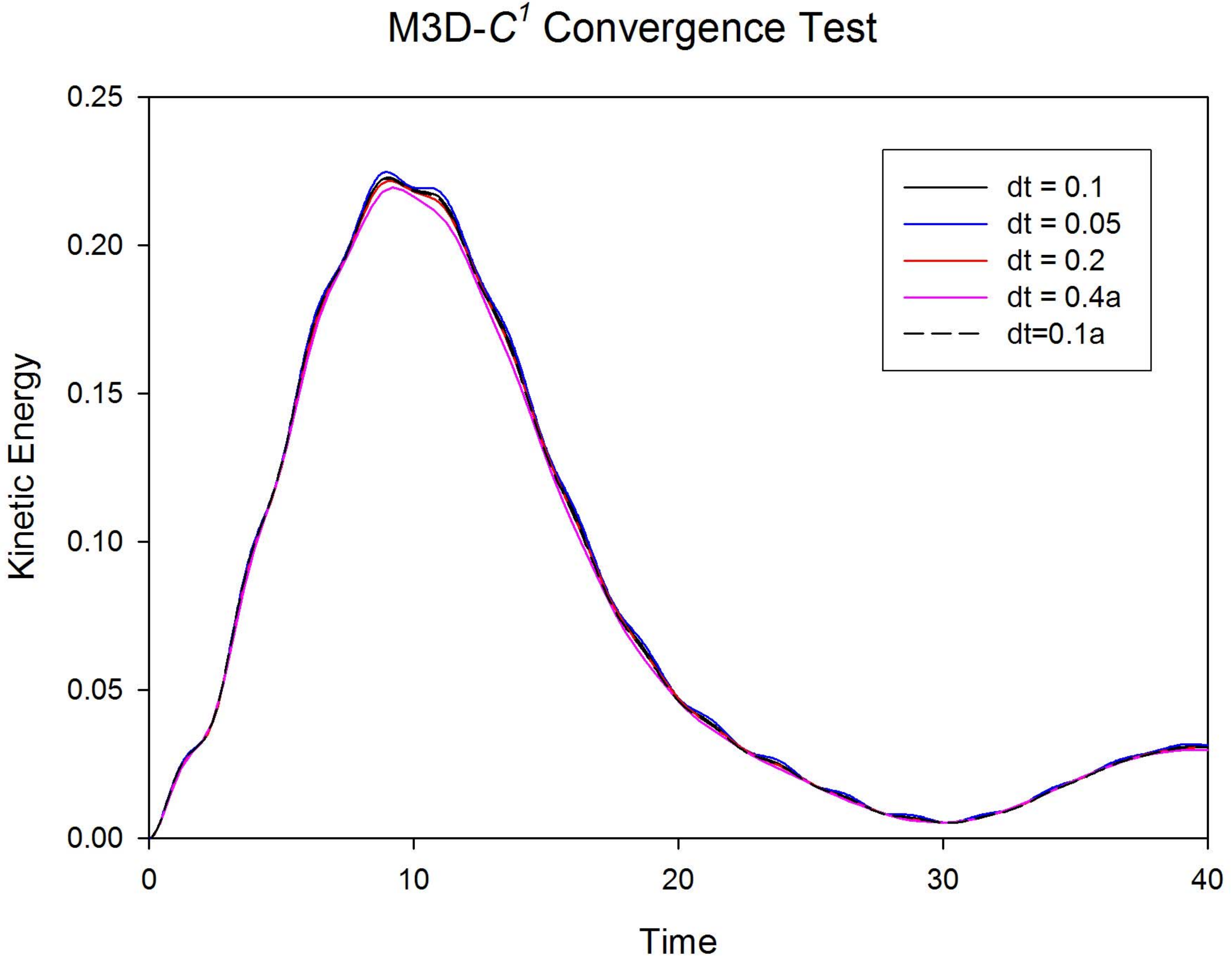
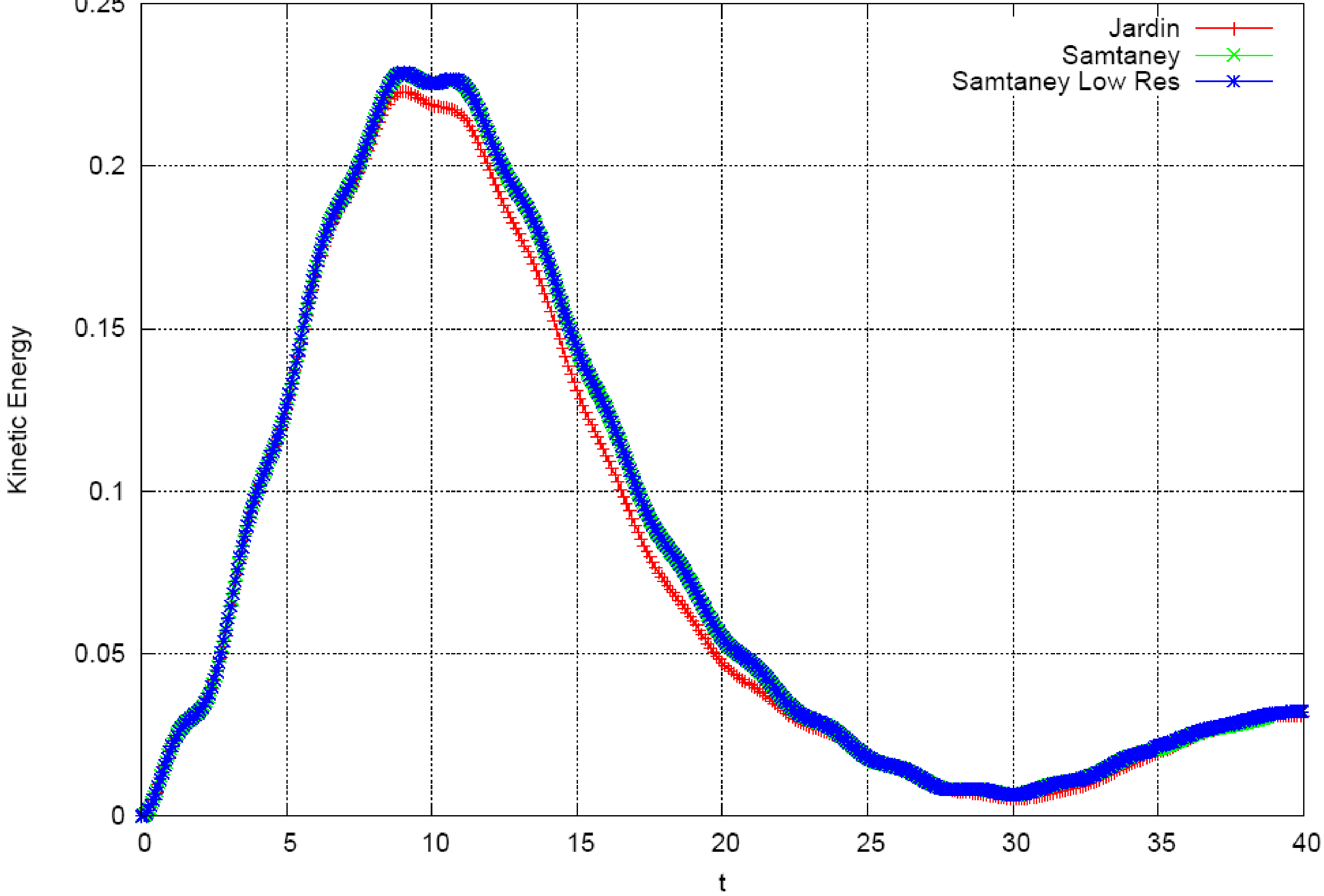
M3D-C^1 Code Developments

- Comparison with Samtaney Code for GEM base case
  - Ohmic and Viscous heating terms added
  - Equilibrium subtracted out or not
- Time step convergence test
- New formulation of compressible term
  - allows substantially larger stable timestep and lower viscosity
- Gravitational instability and Gyroviscosity
  - Nate Ferraro
- Parallel Scaling
  - Josh looking into
- Adaptivity
  - Andy Bauer, RPI
- Future activities



Comparison of kinetic energy vs time for M3D-C1 and Samtaney code for GEM base case: η=0.005 , μ=0.05


M3D-C^1 Convergence Test
Kinetic Energy
Time
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
dt = 0.1
dt = 0.05
dt = 0.2
dt = 0.4a
dt=0.1a



So, both the Samtaney and the M3D-C1 results appear to be converged, but they differ by about 3% RMS over the time period.

It was suggested we repeat this test at lower viscosity to see if the difference is larger.

M3D-C1 code became numerically unstable at lower viscosity when guide field B_0= 0

In looking for the reason for this, we saw you could do another integration by parts in the χ equation (divergence of momentum)

$$-\theta\delta t \nu_i \left\{\left(\nabla^2\dot{\psi},\psi\right)+\left(\nabla^2\psi,\dot{\psi}\right)+2\left(\nabla^2\psi\right)\left(\nabla^2\dot{\psi}\right)\right\} = -\theta\delta t\left\{-\nabla^2\dot{\psi}\left(\nu_i,\psi\right)-\nabla^2\psi\left(\nu_i,\dot{\psi}\right)\right\}$$

$$= -\theta\delta t\left\{\left(\dot{\psi},\left(\nu_i,\psi\right)\right)-\nabla^2\psi\left(\nu_i,\dot{\psi}\right)\right\}$$

New way

$$\left\{J\left(\nu_i,\left[\psi,\dot{U}\right]\right)-\left(\left(\nu_i,\psi\right),\left[\psi,\dot{U}\right]\right)\right\} \equiv C^{10}_{i,j,k,l}\Psi_l\Psi_k\dot{U}_j + C^{11}_{i,j,k,l}\Psi_l J_k\dot{U}_j$$

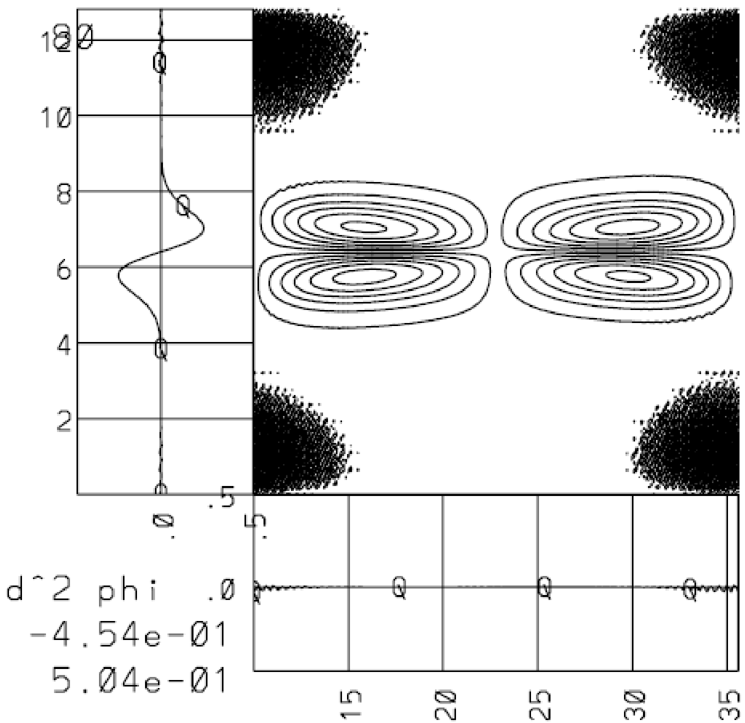
$$\left\{J\left(\nu_i,\left(\psi,\dot{\chi}\right)\right)-\left(\left(\nu_i,\psi\right),\left(\psi,\dot{\chi}\right)\right)\right\} = C^{12}_{i,j,k,l}\dot{\mathrm{X}}_j\Psi_k J_l + C^{13}_{i,j,k,l}\dot{\mathrm{X}}_j\Psi_k\Psi_l$$

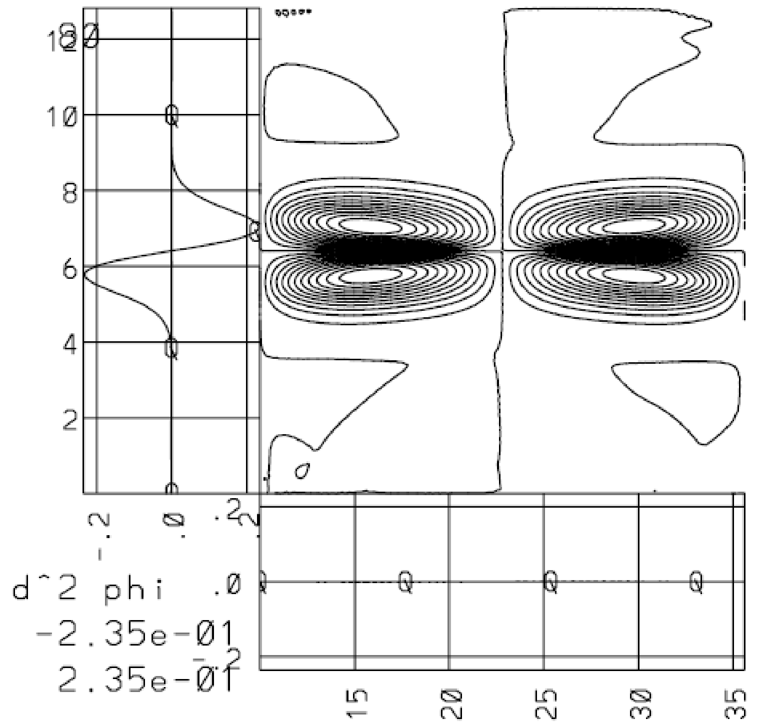
Old way

$$\left\{\begin{array}{l}\nabla^2\dot{U}\left[\left(\nu_i,\psi\right),\psi\right]-2\left\{\left[\dot{U}_x,\psi_x\right]+\left[\dot{U}_y,\psi_y\right]\right\}\left(\nu_i,\psi\right)\\ +J\left\{\left(\nu_i,\left[\psi,\dot{U}\right]\right)+\left[\dot{U},\left(\nu_i,\psi\right)\right]\right\}\end{array}\right\} \equiv C^{10}_{i,j,k,l}\Psi_l\Psi_k\dot{U}_j + C^{11}_{i,j,k,l}\Psi_l J_k\dot{U}_j$$

$$\left\{\begin{array}{l}+J\left\{\left(\nu_i,\left(\psi,\dot{\chi}\right)\right)-\left(\left(\nu_i,\psi\right),\dot{\chi}\right)-2\left(\nu_i,\psi\right)\nabla^2\dot{\chi}\right\}\\ -\left(\psi,\left(\nu_i,\psi\right)\right)\nabla^2\dot{\chi}+2\left\{\left(\psi_x,\dot{\chi}_x\right)+\left(\psi_y,\dot{\chi}_y\right)\right\}\left(\nu_i,\psi\right)\end{array}\right\} = +C^{12}_{i,j,k,l}\dot{\mathrm{X}}_j\Psi_k J_l + C^{13}_{i,j,k,l}\dot{\mathrm{X}}_j\Psi_k\Psi_l$$

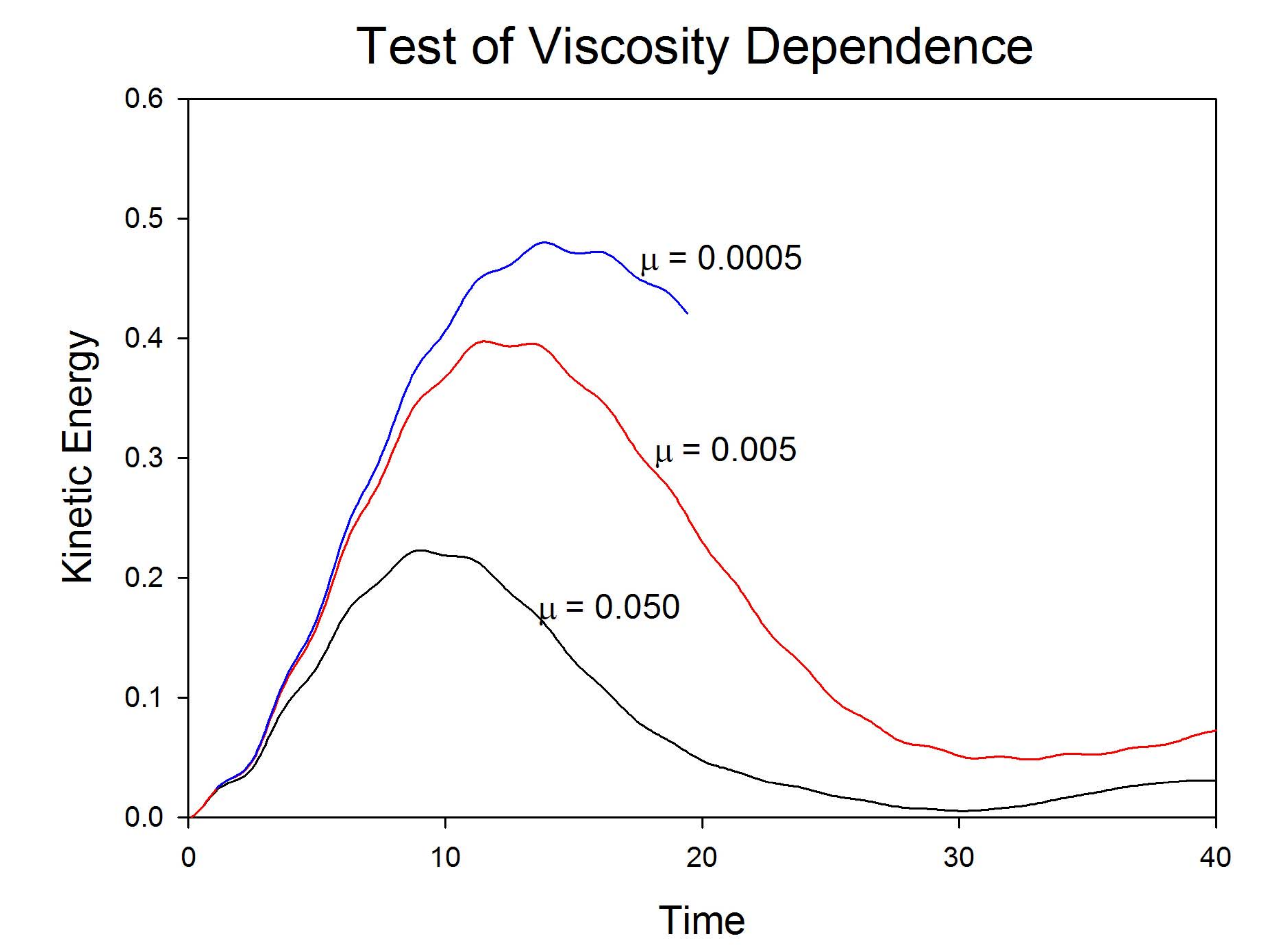
t=8





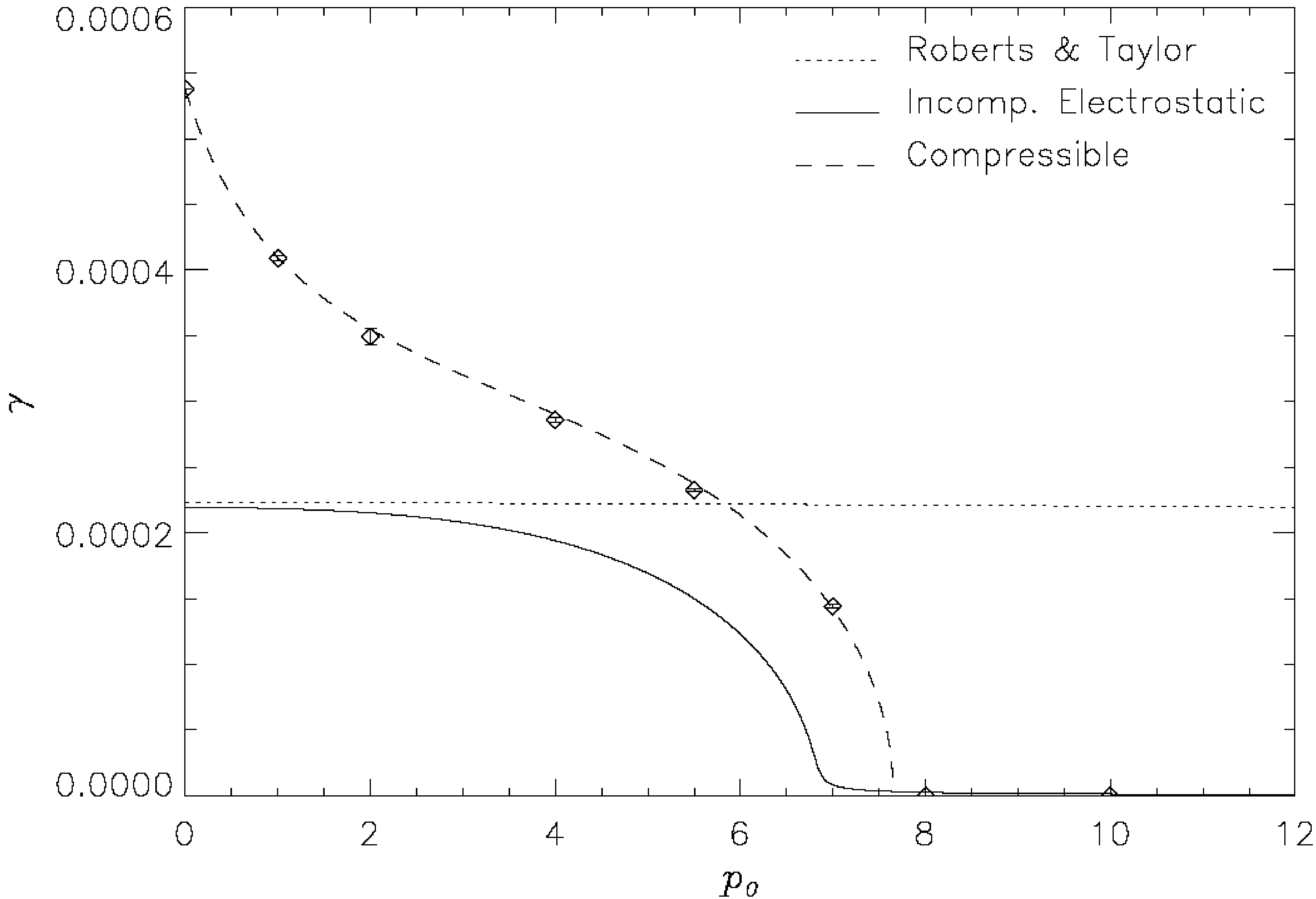
Old way: goes unstable for
μ = 0.005, Δt = 0.10

NEW way: remains stable
for μ = 0.0005, Δt = 0.20 !!



Test of Viscosity Dependence
Kinetic Energy
Time
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30
40
μ = 0.0005
μ = 0.005
μ = 0.050

Gravitational Instability stabilized by Gyroviscosity



Nate Ferraro

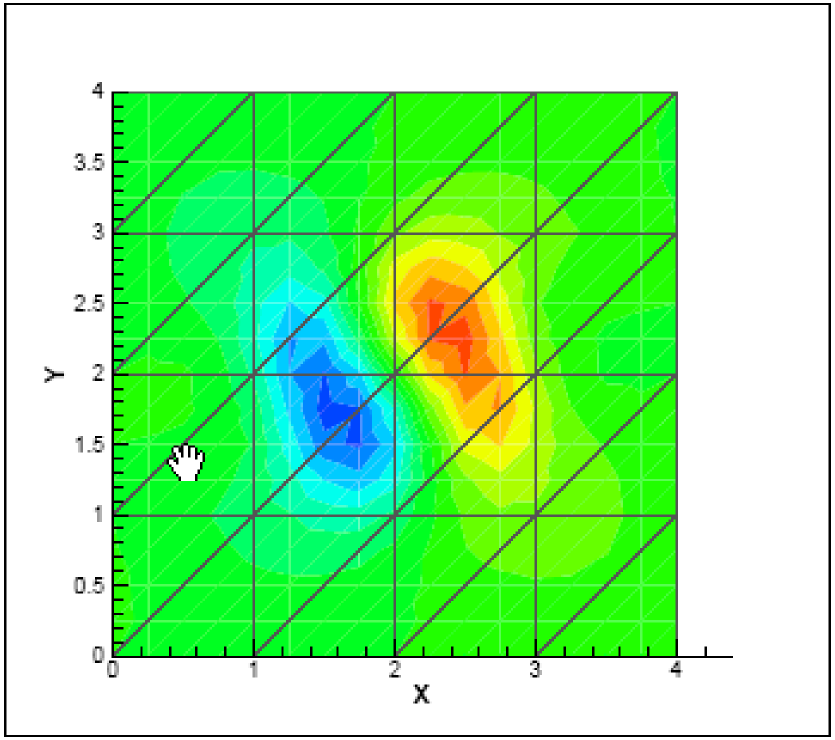
Parallel Scaling

Primary parts of the time loop should be:

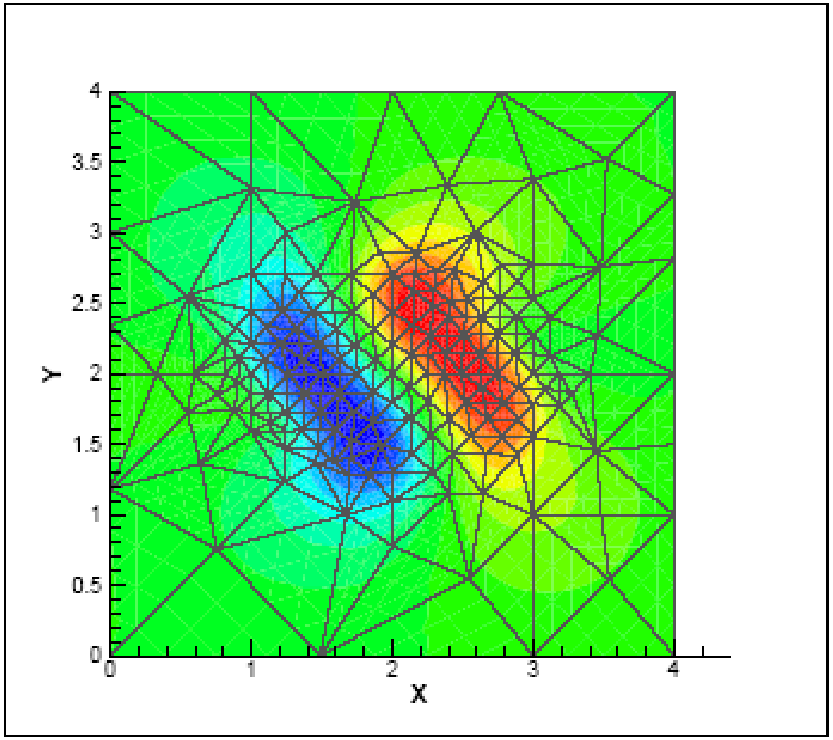
- Define the matrix elements
- Solve the linear equation

Both of these scale with processor number. However, some setup and initialization tasks do not scale and are taking most of the time.   Josh looking into.

Adaptive Meshing



(a) Initial mesh



(b) Adapted mesh

Andy Bauer (RPI) has implemented an arbitrary Adapted Mesh in the M3D-C1 code and is exploring different adaptive strategies

Future Activities

- Continue to verify code
  - Benchmarks with other codes…(why difference with Samtaney code)
  - Wave propagation tests…seems ok
  - New test problems
- Better document, understand, and improve stability limits
  - Goal is stability for all Δt and μ
- Further improve parallel scaling
- Add separate pressure equation in mainline code version (in addition to electron pressure) Ferraro version already has this.
- Add anisotropic heat conduction
- Toroidal axisymmetric equations….This could be a thesis topic
  - 2-fluid equilibrium with flow
- Incorporate unstructured, adaptive mesh into mainline code version
- Toroidal non-axisymmetric (linear)
- Free boundary boundary-conditions
- Toroidal non-axisymmetric (non-linear)