

Update on Using FMR/AMR for GR

Scott Hawley
Albert Einstein Institute
shawley@aei.mpg.de

Workshop on Formulations of Einstein's Equations
Mexico City, 23 May 2002

With:

Here at the meeting: Erik Schnetter, Ian Hawke, Tom Goodale
Back at AEI: Nils Dorband, Bernd Brügmann

AMR Overview

- Many problems in physics solvable via finite difference approx's to PDEs of interest. Divide the computational domain into a mesh, and in the limit where the mesh spacing goes to zero we expect to recover the exact solutions to the PDEs.
- For a given degree of numerical accuracy, it is often the case that the minimum required mesh spacing may not be known a priori, so it may be desirable for the finite difference algorithm to adaptively refine the mesh in various parts of the domain.
- The development of “adaptive mesh refinement” (AMR) computer codes poses difficulties for many researchers because the codes tend to be quite complicated. This complexity is intensified when one considers that, to take advantage of modern “high-performance” computing machinery, a simulation code should be able to run efficiently in a distributed computing environment.
- Thus it is desirable to have a publicly-available system in which parallelism and AMR are provided “automatically”.

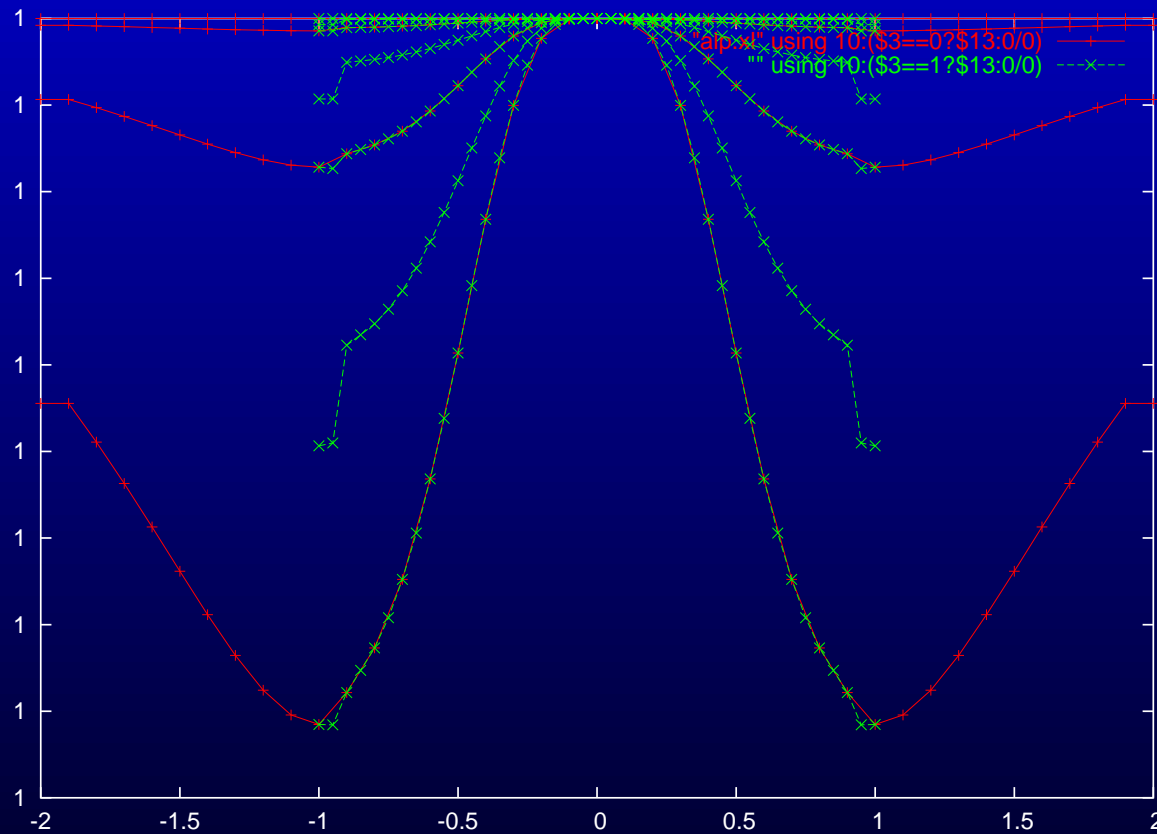
Recent Work among AEI/KDI

- Codes:
 - ★ Erik Schnetter's Carpet (FMR package) works with Cactus.
 - ★ Manish Parashar's GrACE (AMR package)...does not yet work with Cactus.
- Convergence Tests using wave equation
 - ★ Sinusoidal waves
 - ★ 1/r data
- Over the past month:
 - ★ Revamp of Einstein thorn allowed for timelevels
 - ★ Revamp of CartGrid3D thorn allowed Carpet to work with our thorns
 - ★ MoL does not yet work with FMR/AMR. Awaiting change to Flesh
 - ★ Still needed timelevels in ADM_BSSN...

Ian & I Added Timelevels to ADM_BSSN, and Look!

Schwarzschild, 1+log slicing, ICN

$\alpha(t)$ vs. x

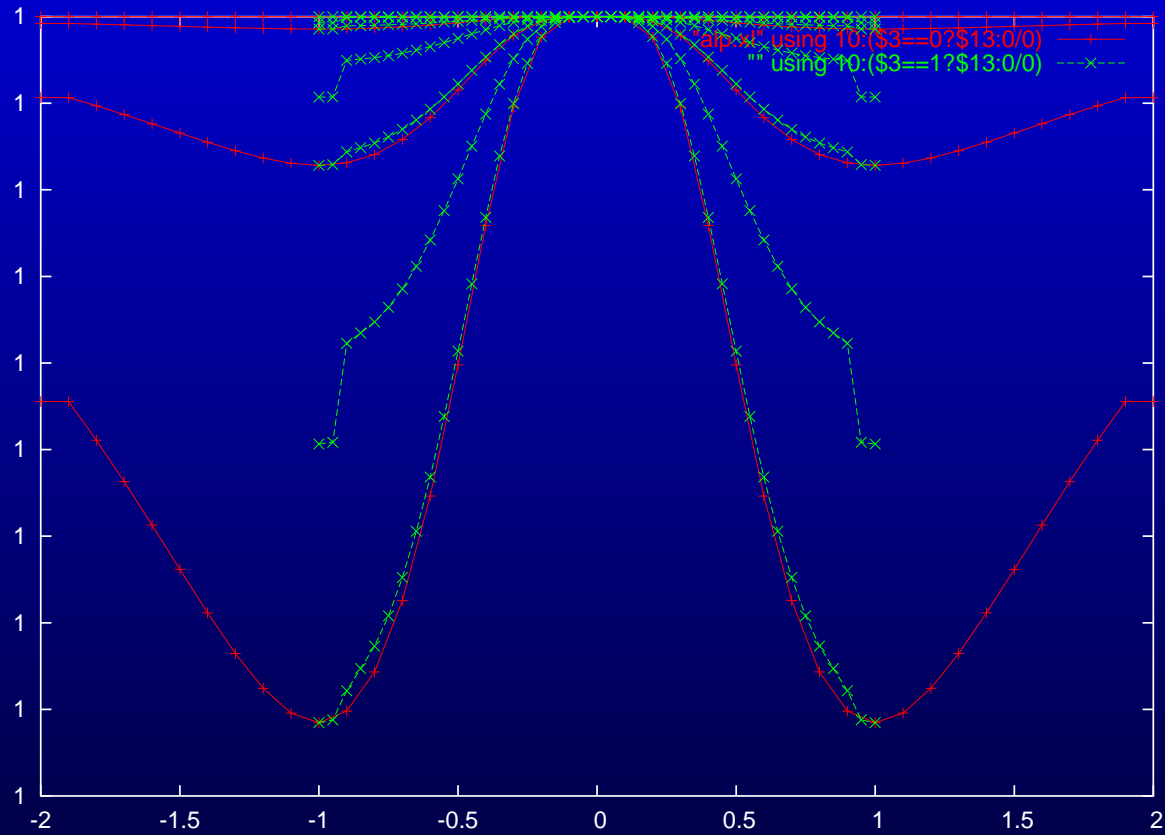


Using Carpet package (Schnetter):
 2 Refinement Levels, Linear Interpolation in Time, Cubic in Space

Immediate Questions About That

- What was the vertical scale on that plot?
 - ★ Roughly $1 - 10^{-8} < y < 1$
- When can we start using this in our GR code?!
 - ★ *Conserve la calma*. The point is merely that we didn't get NaNs or seg faults! There's still work to do...
- What are those funny 'discontinuities' at the "adaptive ghosts"?
 - ★ Adaptive ghosts are updated using coarse-grid data. Finer-grid data evolves (in this case) more slowly than coarse-grid data. Can see this by looking at evolution without restriction (next slide)
 - ★ Time interpolation is working OK (Space interp is probably also OK.)
- Does the solution converge?
 - ★ Doubt it. For linear interp. in time, we may not get 2nd-order conv.
 - ★ Other tests indicate that it's **possible** to obtain 2nd-order convergence for wave eq, for both periodic and $1/r$ data, using parabolic interp in time

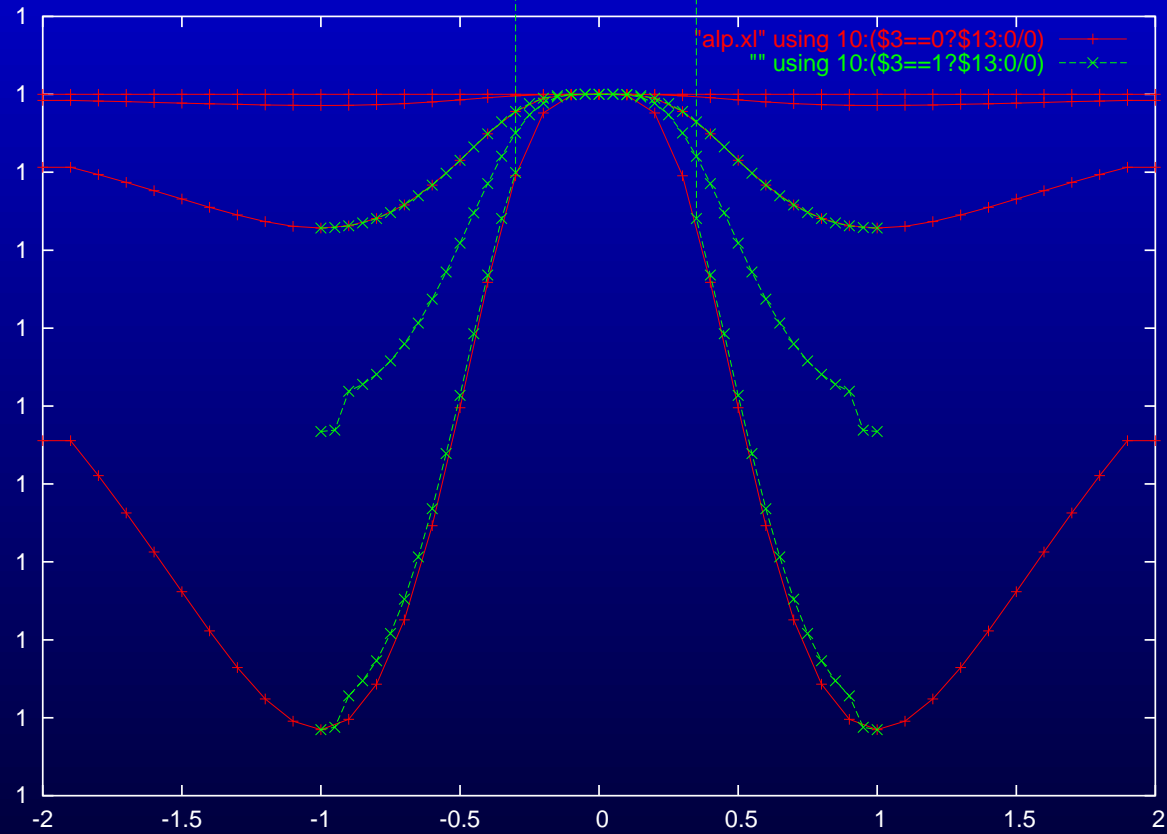
Interpolation is Working OK



Linear Interpolation in Time, Cubic in Space
 NO RESTRICTION

Interpolation is Working OK (2)

(More on these NaNs later)



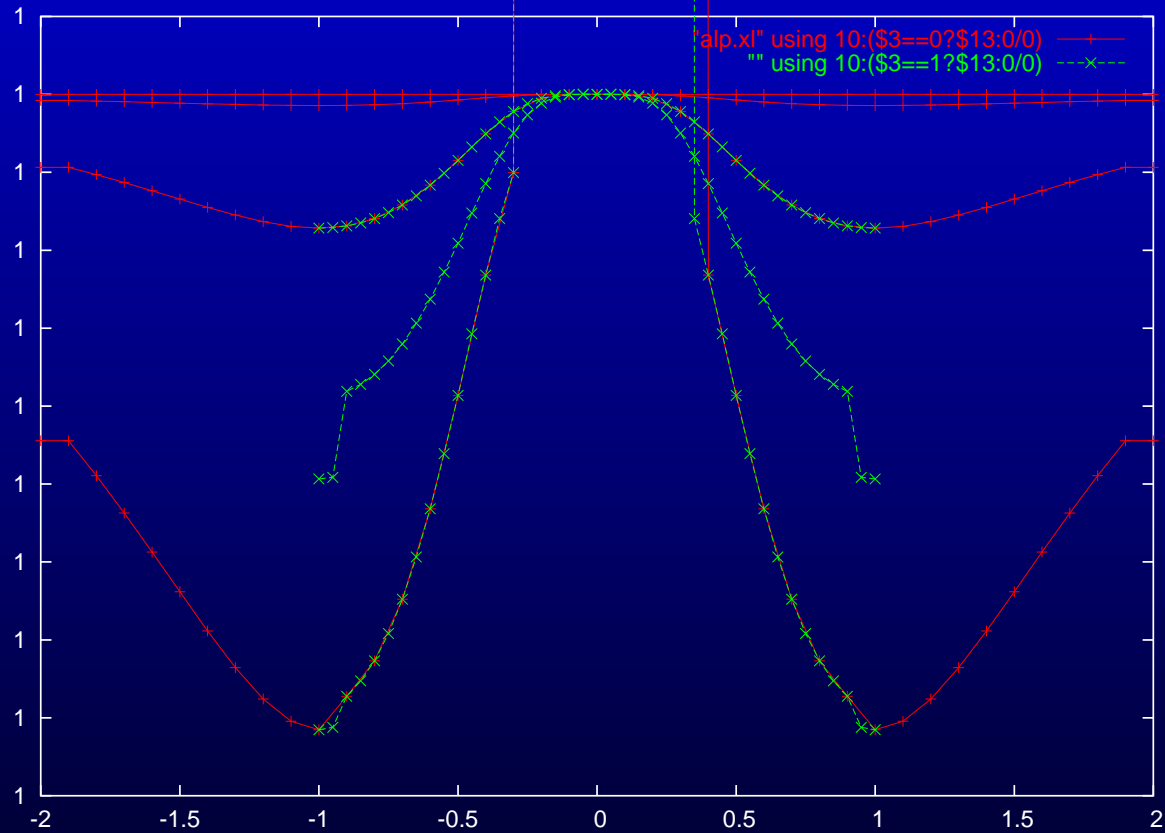
Parabolic Interpolation in Time, Cubic in Space
 NO RESTRICTION

How Do We Do Parabolic Interpolation in Time?

- Need 3 timelevels worth of data, for all but the finest level. **This includes initial data!**
- Can come up with fancy schemes for getting 3 timelevels of initial data (Scott can provide if you're interested)
- Ian's idea:
 - ★ Just start evolving with only the coarse grid.
 - ★ Then turn on refinement after a few steps.
 - ★ If we do that...

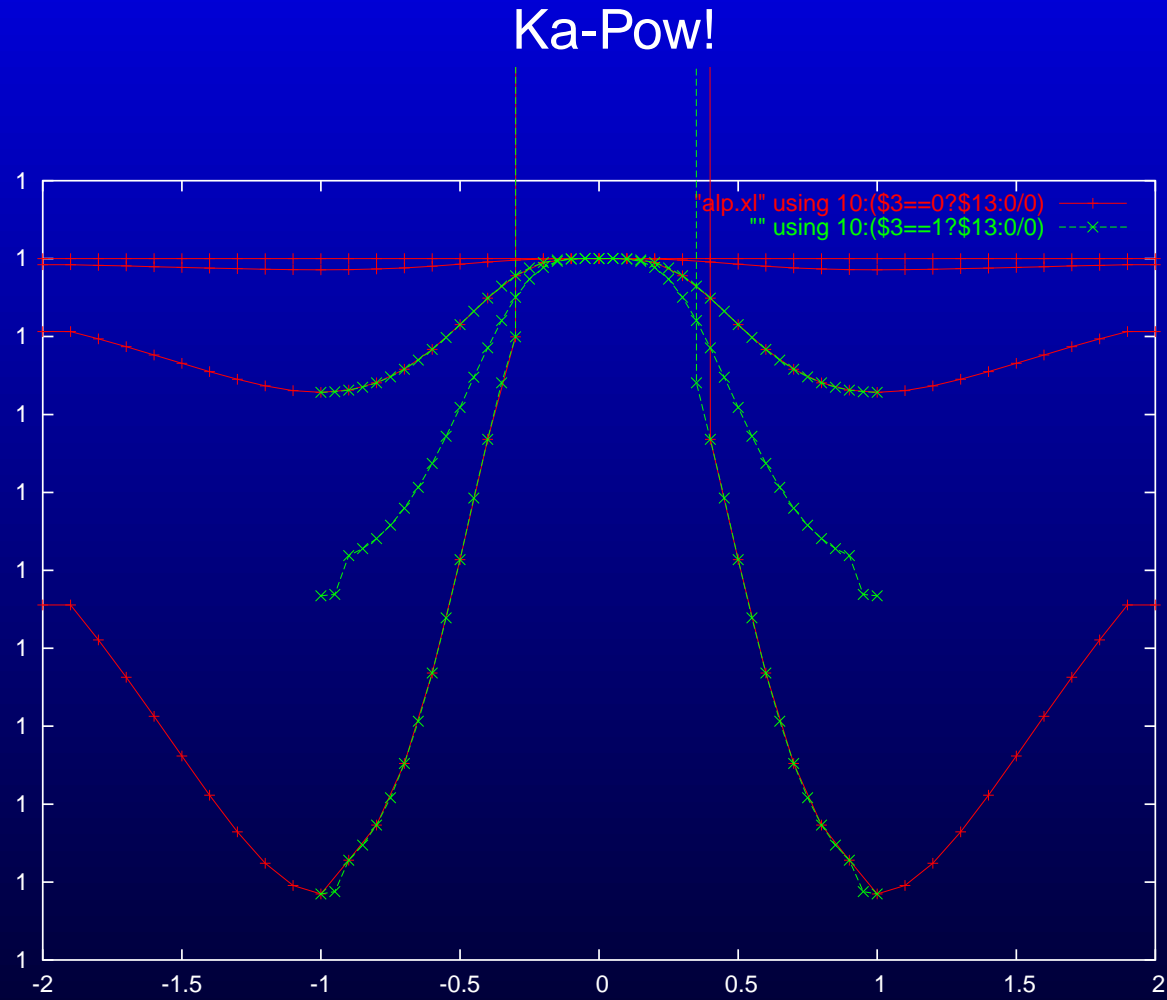
We Get NaNs

Ka-Pow!



Linear Interpolation in Time, Cubic in Space

We get NaNs (2)



Parabolic Interpolation in Time, Cubic in Space

Why Do We Get NaNs?

- Erik's idea: We're effectively supplying initial data via interpolation of coarse-grid data — near the puncture!
 - ★ Thus we do OK when we have all levels active at initial time, because levels are initialized using analytic data
 - ★ Couldn't we try a simpler problem, without punctures? You betcha!
- But we need a fancy scheme if we want parabolic interp in time from the get-go.
 - ★ Scott has fancy schemes in case they're needed.
- Ian's other idea: Give as high an order of time interp as possible
 - ★ Thus we start out with only linear interp in time, but get parabolic when enough time levels of data are present.
 - ★ Currently "Carpet don' play dat," but not a problem to implement.

Preparing Your Thorns for the Advent of FMR/AMR in Cactus

In general: **Make sure your thorn is ‘suitably ignorant’** about what’s happening in time & space.

For anything ‘clever’, don’t do it in the C or Fortran code!
Instead, use Cactus infrastructure:

- Use Cactus timelevels
- Use CCTK_DELTA_SPACE(?) macros for Δx , Δy , Δz
- Do all your Sync’s in the Scheduler
- Use “grid variables” for all your data, i.e. don’t store static data (e.g. temporary arrays) which you think is data for the whole grid
- Use thorn Time if you’re going to be doing funny things to the time (e.g. May-White codes)

Concert Tonight

- At Hotel San Francisco (on Luis Moya street)
- 9pm - 10pm/10:30ish
- Room 1105
- Cover? 40-50 Pesos