Evolving Black Holes with Fixed Mesh Refinement

Scott Hawley
Center for Relativity
University of Texas at Austin
`shawley@physics.utexas.edu`

18 April 2003

With:
Erik Schnetter
University of Tübingen

AMR Overview

- Many problems in physics solvable via finite different 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.

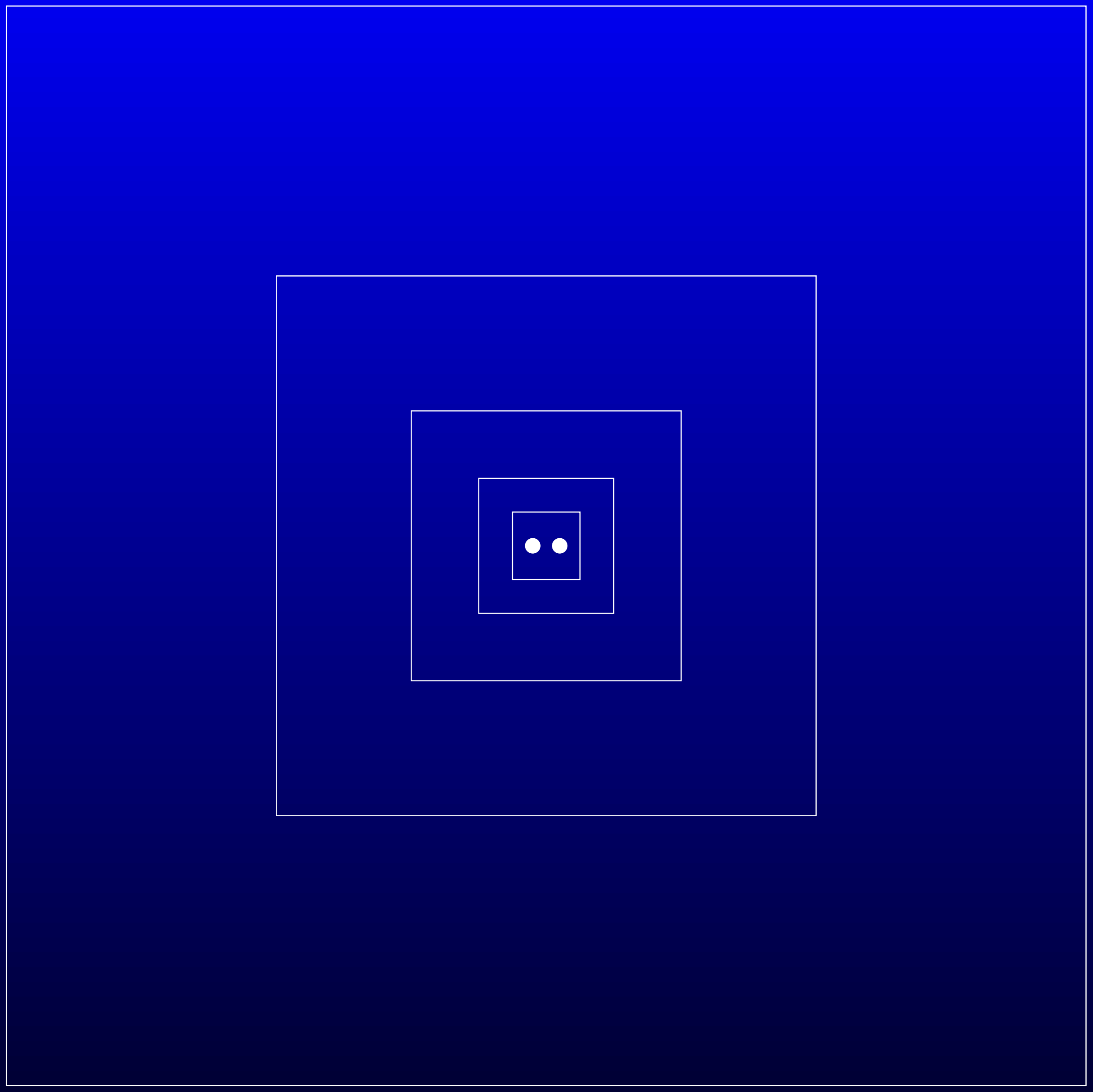
- The development of "adaptive mesh refinement" (AMR) computer codes poses difficulties for many researchers because the codes tend to be quite complicated. This complexity is intensified when one considers that, to take advantage of modern "high-performance" computing machinery, a simulation code should 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".

- Even without a truly ‘adaptive’ mesh code, it may still be desirable to have a fixed mesh refinement (FMR) code, with which to concentrate computational resources where they are most needed.

- The principle challenge is the same in FMR & AMR: handling inter-grid boundaries well. If one can do this with FMR, then AMR development presents no significant mathematical challenges.

- Thus FMR is a natural goal along the way to AMR, and may even be all that is required to study some interesting systems.



r~32M

e.g., 40^3 (Comparable to 1280^3)

Some AMR/FMR Work in NR To Date

- Matthew Choptuik; Steve Liebling; Frans Pretorius
- Grand Challenge/GrACE - WashU Group (Wai-Mo Suen, Ed Evans, Sai Iyer)
- Lee Wild
- Simon Hearn
- Bernd Brügmann
- Gerd Lanfermann
- NASA Goddard (Joan Centrella's Group)
- Peter Diener (Khohklov, et al)

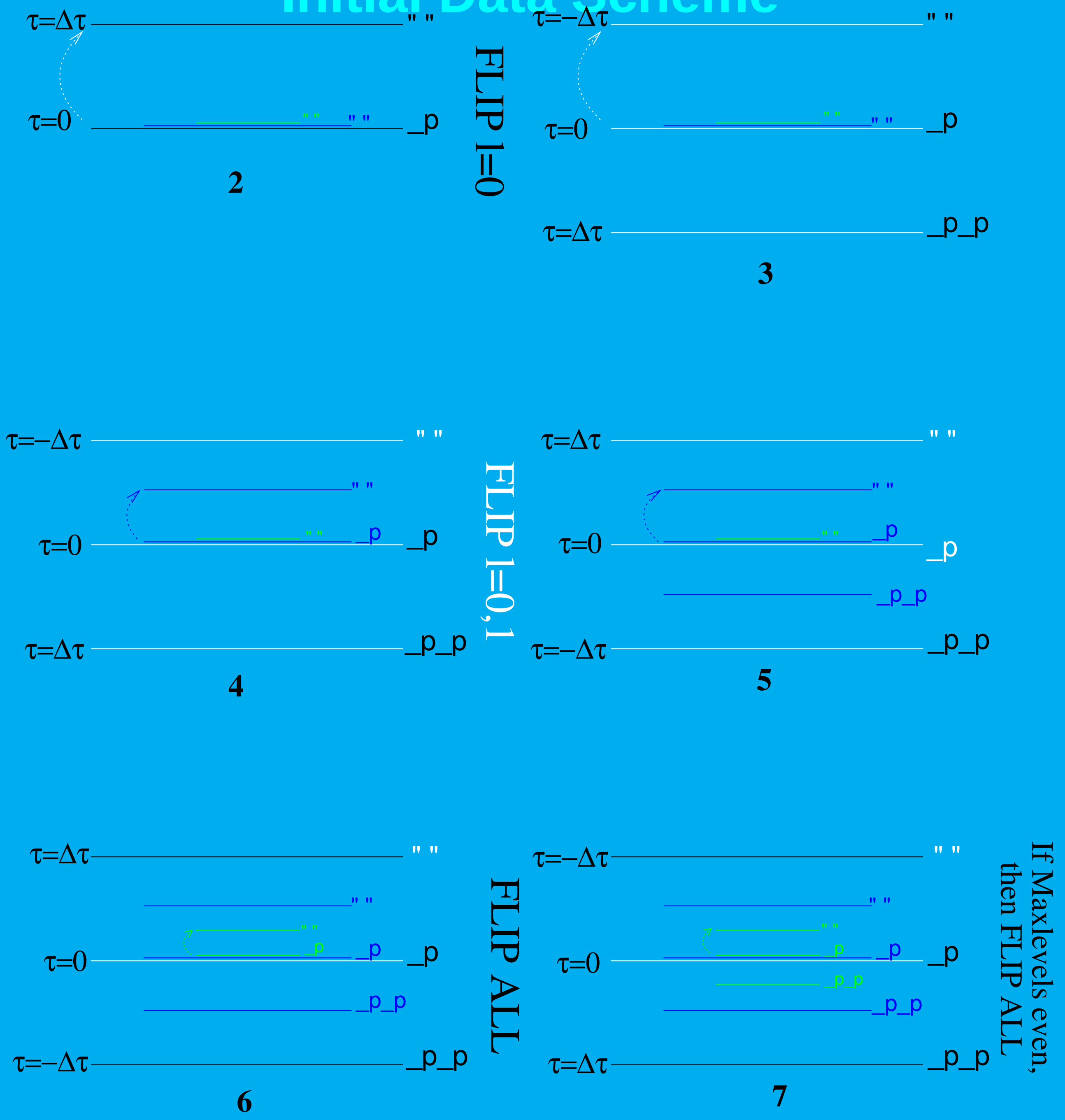
Our Method

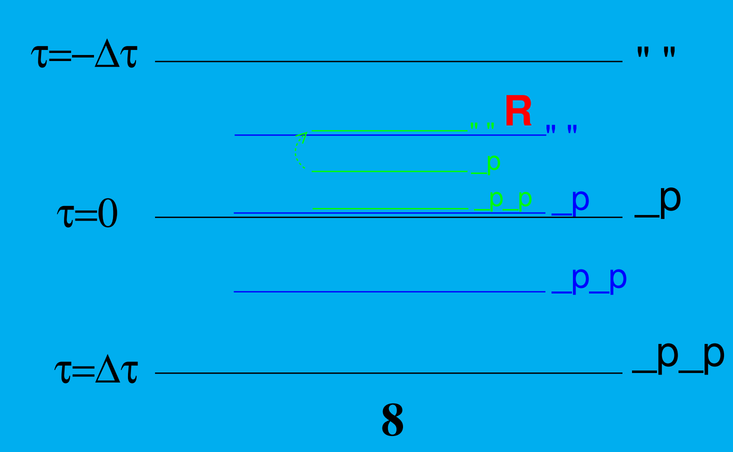
- The MR Method
  - ⋆ FMR (Can also do “progressive” MR)
  - ⋆ Vertex Centered
  - ⋆ Interpolation: Cubic (4th order) in space, Parabolic (3rd order) in time
  - ⋆ Initial Data Scheme, enabling parabolic time interp even at startup
- The Code itself
  - ⋆ Uses Carpet (by Erik Schnetter), a Cactus Driver for (parallel) FMR
  - ⋆ “Black box”
  - ⋆ Performance?
- The “Physics”
  - ⋆ Brandt-Bruegmann “puncture” initial data (Brill-Lindquist topology; Bowen-York K_{ij}; Conformally flat, solve for Ψ numerically)
  - ⋆ BSSN Evolution
  - ⋆ 1+log slicing
  - ⋆ Gamma-freezing shift
  - ⋆ Robin boundary conditions

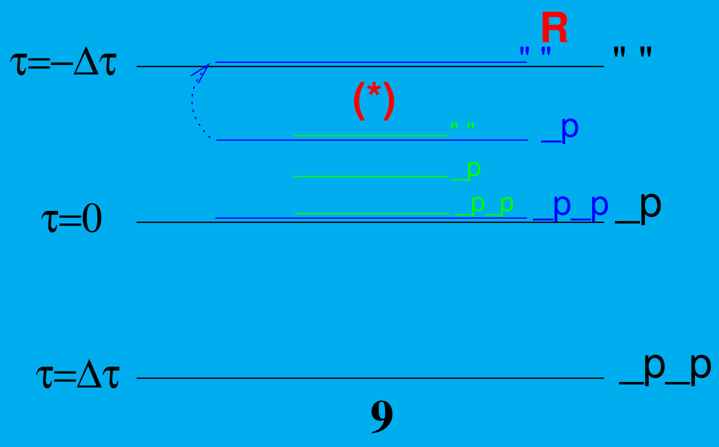
Initial Data Scheme

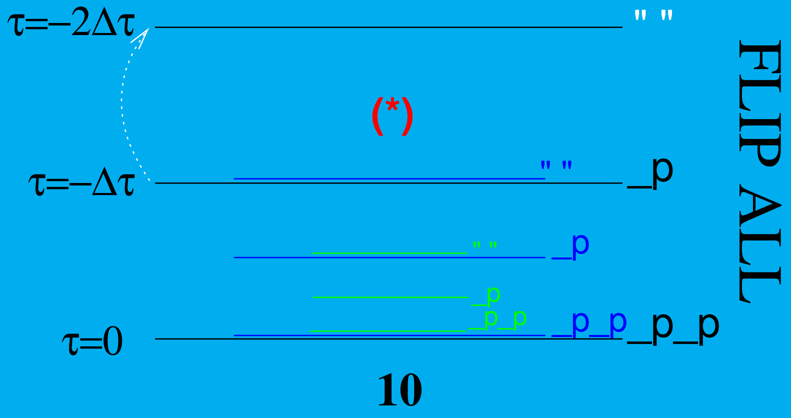
- Boundary data for fine grids are obtained by interpolating coarse grid data in time & space.
- To preserve convergence properties, we want to do parabolic (3rd-order) interpolation in time (even at the get-go)
- Need *three* time levels of coarse grid data to do this
- How to get these levels of coarse grid data at the initial time?
- Simple: Evolve forward *and* backward on coarse grid, then (along with $t = 0$ data) we have three time levels with which to interpolate.

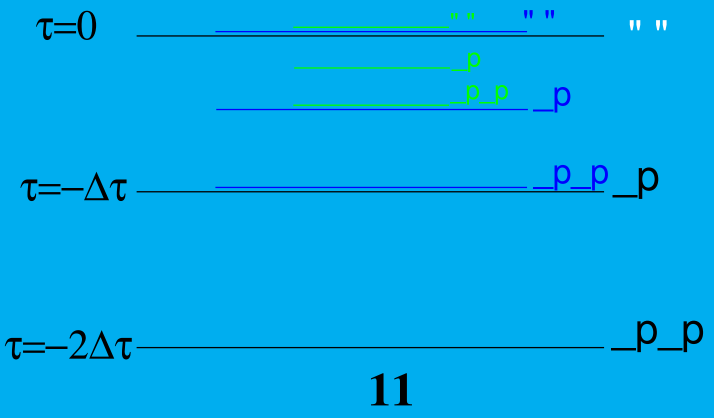
Initial Data Scheme







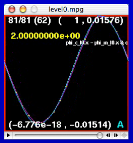


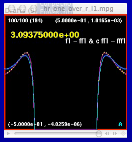


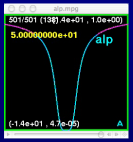
R = Restriction

(*) = Can evolve finer levels further, if feeling "anal"

Tests - Simple Systems

- Wave Equation
  - ⋆ Periodic waves: Convergence movie 

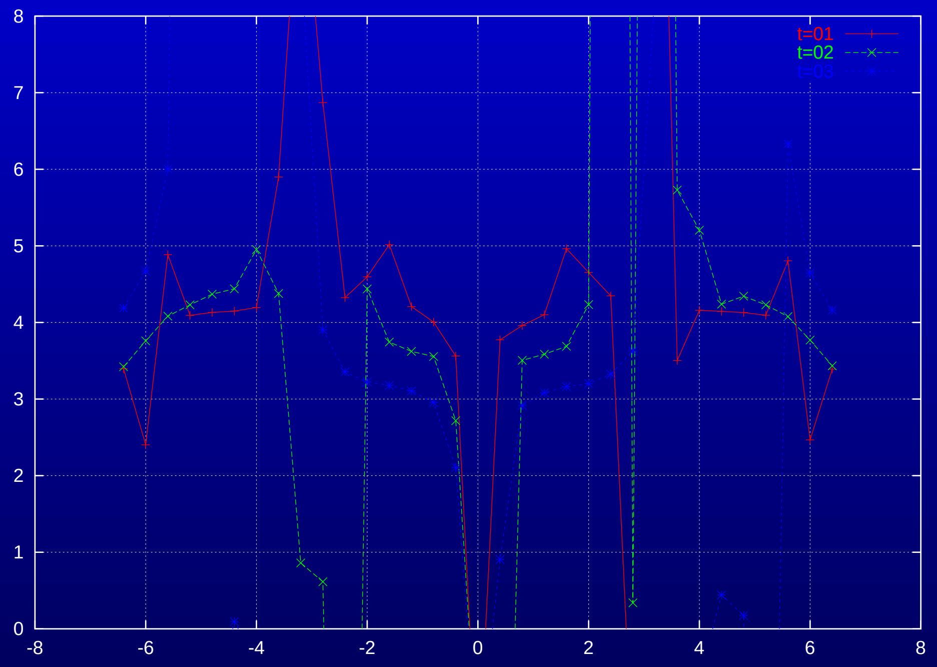
  - ⋆ 1/r data: Convergence movie 

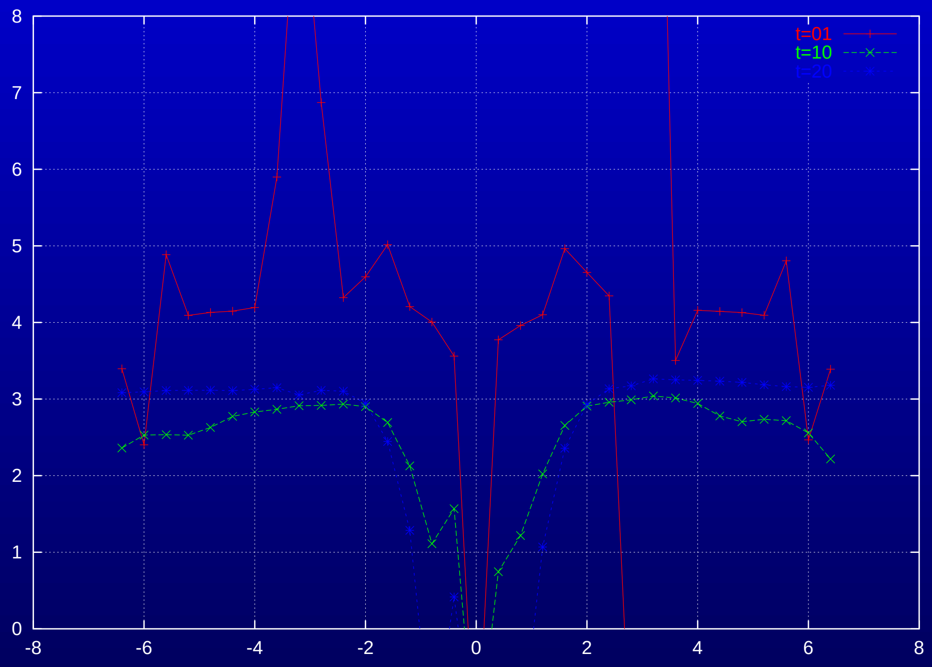
- Schwarzschild:
  - ⋆ Lapse movie 

  - ⋆ Only first-order convergent near puncture.
- FMR vs unigrid
  - ⋆ Good news: FMR runs crap out on the *same* timescale as comparable unigrid runs.
  - ⋆ FMR runs more sensitive to outer boundary location
  - ⋆ FMR Needs puncture *between* grid points; Unigrid doesn't care
  - ⋆ Not big problems practically, but would like to figure out why

Schwarzschild Convergence Tests

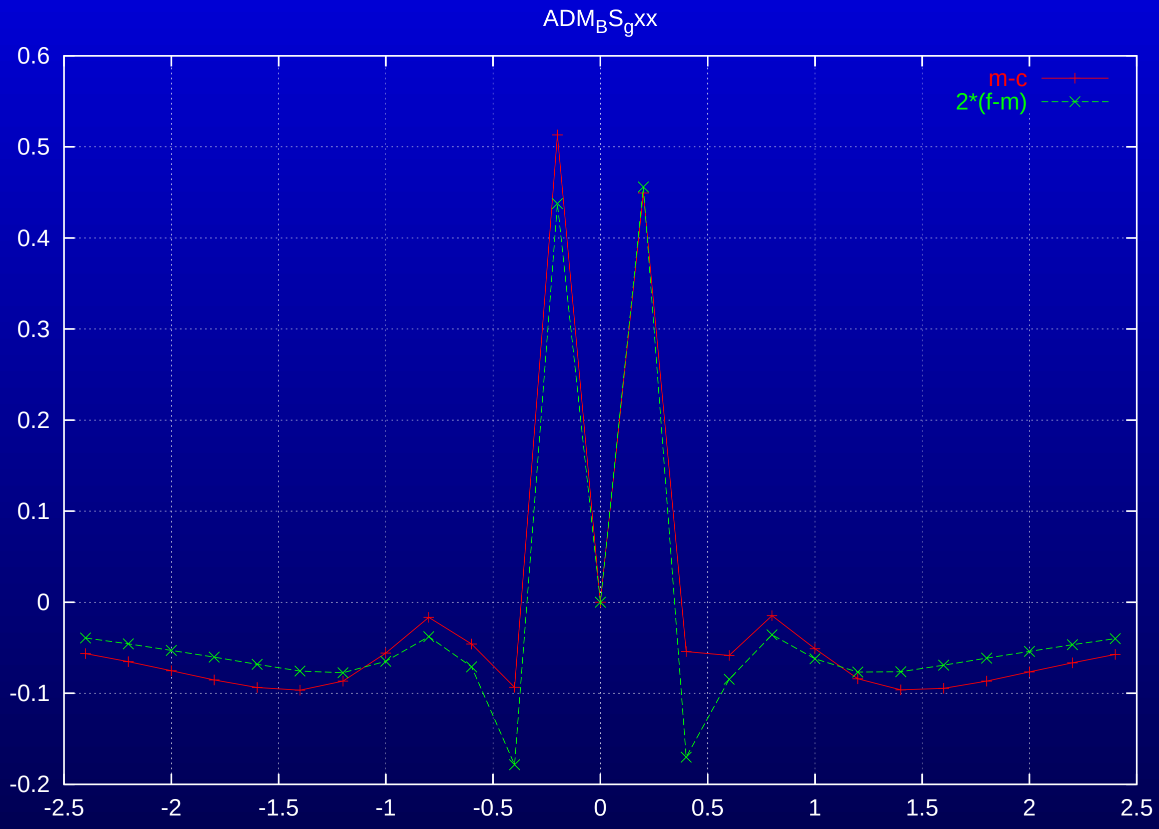
Q(gxx) along x-axis, for $t = 1, 2, 3$, and $t = 1, 10, 20$:





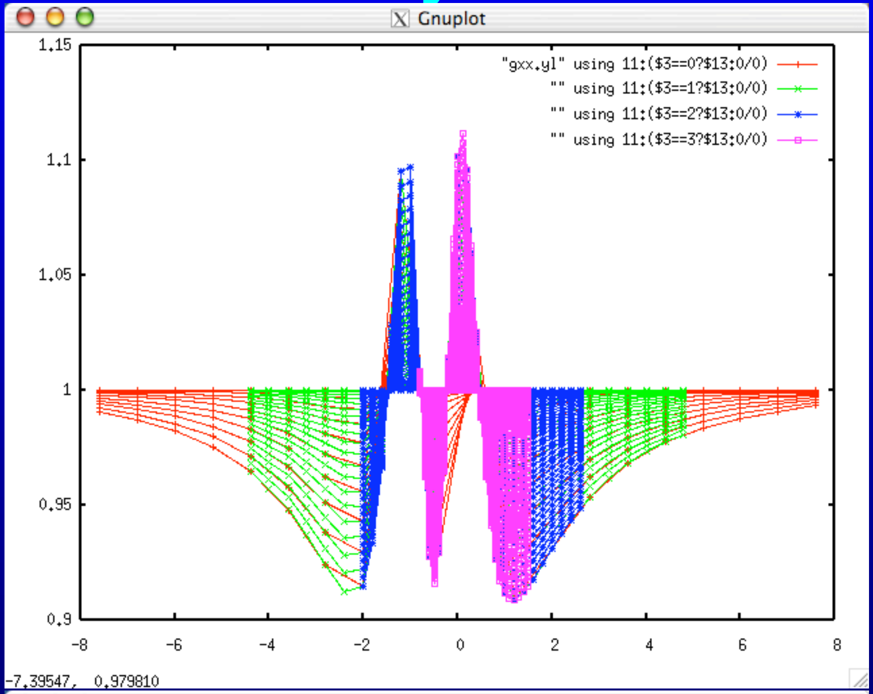
...logψ and α look similar (but worse!)

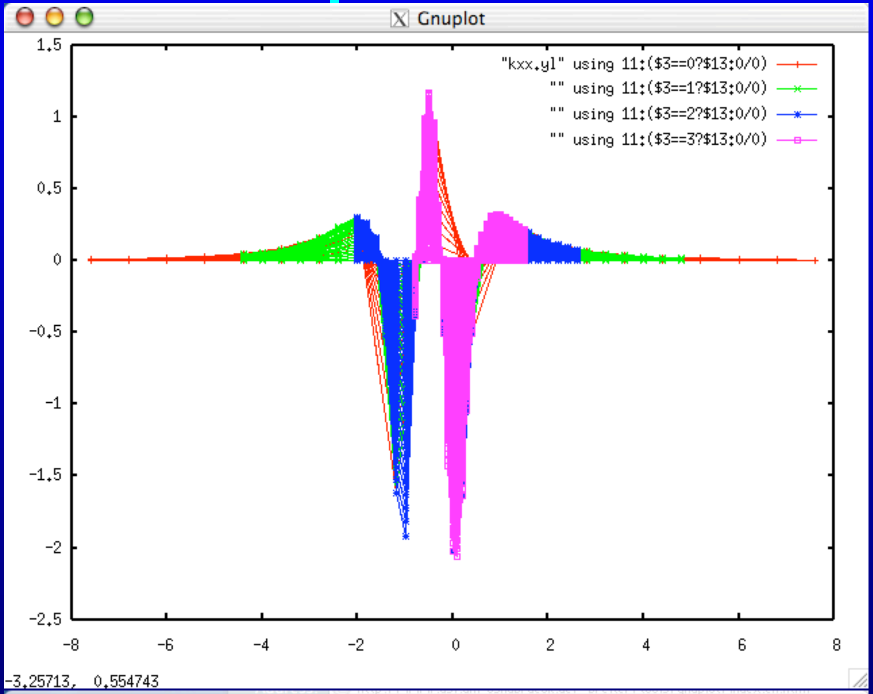
LTE[gxx] $\sim \mathcal{O}(h)$:

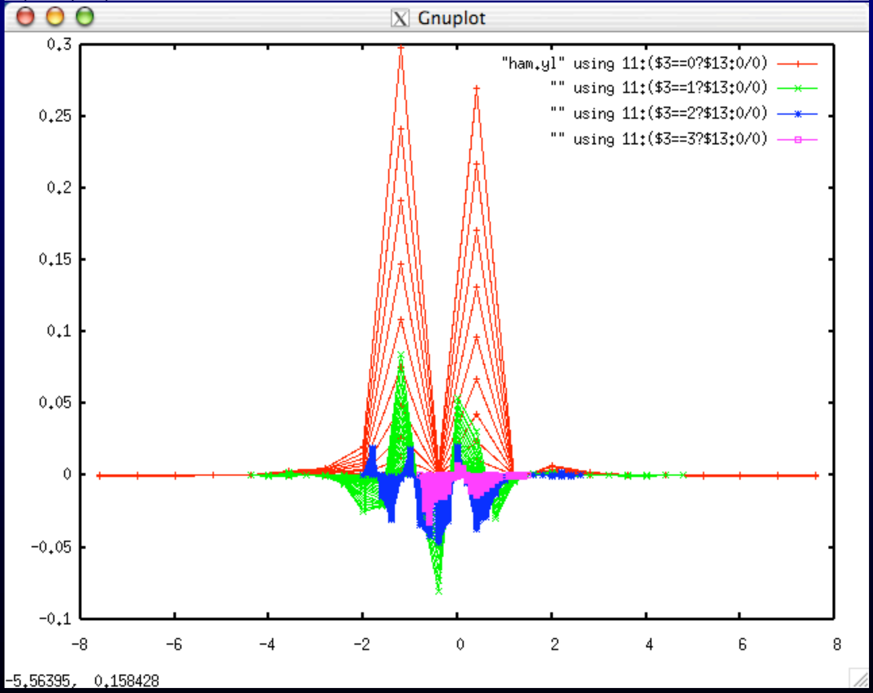


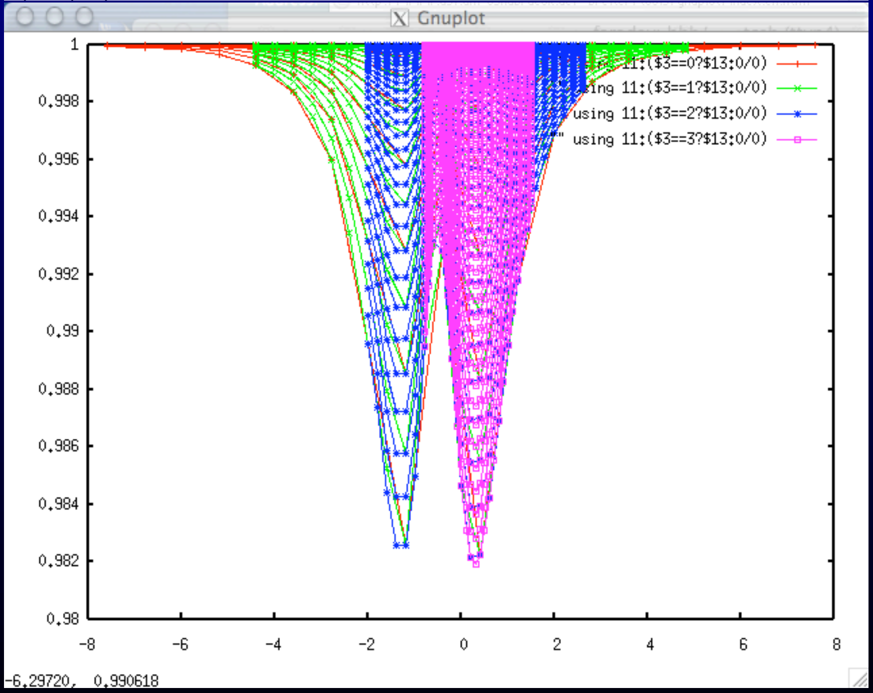
Observation: But unigrid run doesn't converge either!

Binary Black Holes: Brill-Lindquist Data









Conclusions

- *So far*, it *seems* the FMR driver is basically working as the “black box” we hoped for.
- That is, we’ve taken some existing evolution systems, swapped out the unigrid driver for an FMR driver, which...
- Handles scalar waves in a second-order convergent manner – even 1/r data, which is the way ψ goes near punctures
- GR evolutions are...not *obviously* wrong; functions smooth near inter-grid boundaries, no “funny business”.
- ...but more testing is needed.

Future Work

- Further tests:
  - ⋆ 2nd-O convergence for GR system - Linear (standing?) waves
  - ⋆ Further checks to make sure multi-level FMR run gives comparable results to fine-unigrid run (& in less time?)
  - ⋆ “Putting boundaries far out”: Repeat a unigrid run, but with a coarser grid around it, extending outer boundary.
  - ⋆ Parallel performance?
- Use it more! *Current* capabilities of code have hardly been tapped.
  - ⋆ Linear gravitational waves
  - ⋆ Binary Black Holes - Misner & Brill-Lindquist
  - ⋆ Many levels of refinement
  - ⋆ ”Progressive Mesh Refinement” - predefine grids and “turn them on” over time, e.g. to follow gravitational collapse.
- Need to write MR elliptic solver to handle more general initial data
- Soon, FMR simulations become mundane!

Aside: 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)