AMR techniques for Numerical Relativity

Gravitational Interaction of Compact Objects
KITP – 14 May 2003

Frans Pretorius
Caltech

Outline

- Overview of the problem
  - why grid-based solution of Einstein's equations need some form of adaptive mesh refinement (AMR)
  - Berger and Oliger style AMR
- Brief history of AMR in numerical relativity
- Select techniques of possible interest
  - *using a "self-shadow hierarchy" to compute truncation error estimates*
  - incorporating elliptic equations for fully constrained evolution
  - AMR for characteristic codes
- Conclusions

2

Overview of the problem

- Numerical solution of the Einstein field equations in 3D requires significant computational power
  - a system of coupled, non-linear partial differential equations containing hundreds of individual terms
  - relevant length scales span several orders of magnitude for systems of interest to gravitational wave detection. For example, equal mass binary black hole merger:
    - 1) individual sources, ~ M
    - 2) orbit, ~10M
    - 3) 'wave-zone', where resultant waveforms can accurately be measured, ~100M
  - temporal scales span a similar range of magnitudes, e.g.
    - at late stages of inspiral, expect significant dynamical changes on characteristic timescale ~M
    - need to evolve several orbits at least, ~100M to 1000M

3

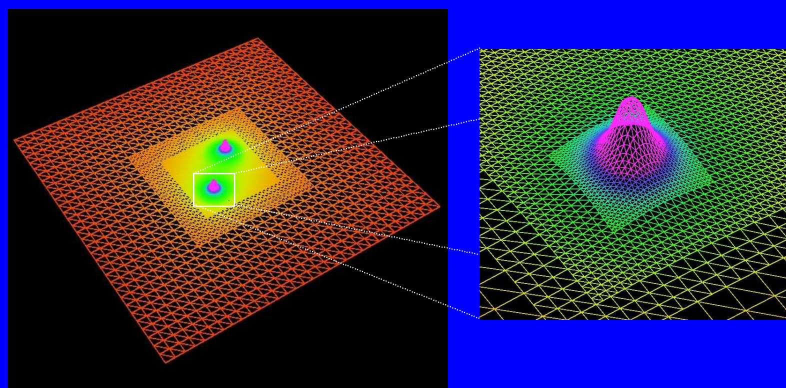
Why is AMR a possible solution?

- For many of the systems of interest to gravitational wave detection (i.e. systems containing black holes and/or neutron stars), the shorter length scales occupy smaller volumes of the computational domain
- We do not know enough about these systems to want to bet that an a-priori choice of a "body fitting" coordinate system will allow us to solve all problem with sufficient resolution

4

Berger & Oliger style AMR

- covers the computational domain with a hierarchy of uniform meshes, with smaller, higher resolution meshes entirely contained within the larger, coarse meshes



5

Berger & Oliger style AMR

- The hierarchy is generated via local truncation error (TE) estimates to provide sufficient but not excess resolution
- A recursive evolution algorithm allows "optimal" use of resources in time as well as space, within limits allowed by the CFL condition
  - the recursive rule is that 1 evolution step is taken on a coarse level before ρ_t steps are taken on the next finer level
    - ρ_t is an integer, typically equal to the spatial refinement ratio ρ_s in order to satisfy the CFL condition
    - the solution obtain on the coarse grid is used to set interior boundary conditions on the fine grid
    - a side benefit of the recursive nature of the algorithm is that TE estimates can be computed essentially for "free"

6

Berger & Oliger style AMR

- Example of possible speed-up versus a unigrid code in an equal mass black hole merger simulation. Assume:
  - finest grid fully covers each black hole (smallest scale, of width $2M$)
  - a linear filling factor of 1/2
  - a refinement-ratio of 2:1
  - an outer boundary at $L_x M$ (longest scale, of width $2L_x M$)
  - evolution to $t = L_t M$
  - CFL factor = 1/2
  - finest level $L_f = \log_2(L_x)$ is a positive integer, sufficiently large such that $\sum_{n=0}^{L_f-1} (1/2)^n \sim 2$

7

Berger & Oliger style AMR

- Then, the speed-up factor of AMR/unigrid is

$$\frac{T_u}{T_{AMR}} \approx \frac{C_u}{2C_{AMR}} (L_x)^3$$

- T_{AMR} / T_u is the total run time for the AMR/unigrid simulation
- C_{AMR} / C_u is the average CPU time per grid point needed to solve the equations in the AMR/unigrid code — the ratio will be O(1)

8

A couple of Caveats

- Interior grid-boundary "noise" can be problematic when trying to extract a small signal, such as gravitational waves, from the simulation
  - find a set of interpolation/injection/dissipation operators that work for a given problem
  - choose variables adapted to the signal of interest, so that it is not a "small" feature
  - filter out the noise in a post-processing operation
- AMR is not a cure-all
  - even with the most optimistic speed up from AMR, and estimated increase in computer power, it is not reasonable to expect to be able to simulate a binary inspiral for thousands (or even hundreds) of orbits … analytic approaches essential then
  - can only be efficient in problems with small filling factors
  - does not eliminate the need to choose good coordinates … grid stretching and other coordinate "singularities" will kill an AMR code just as easily as a unigrid one.

9

Brief history of AMR in NR

- Choptuik (1993) - critical phenomena
  - a handful of critical studies since then have also used AMR
- Brügmann - first 3+1 AMR simulation of Schwarzschild spacetime (1996) and binary black hole merger (1999)
- Papadopoulos et. al. (1998) - 3+1 simulations of linearized gravitational wave-black hole interactions
- Hern (1999) - 2+1 evolution of inhomogeneous cosmologies
- Diener et. al. (2000) -computation of initial data for black hole collisions
- New et. al. (2000) - 3+1 simulations of weak gravitational waves (recently extended to strong waves and black hole spacetimes)

10

Coupled Elliptic-Hyperbolic Systems with AMR

- An evolution technique for the Einstein equations that may be worth pursuing is a *fully constrained evolution scheme*
  - solve the 4 constraint equations in lieu of 4 evolution equations
  - *a priori* eliminate all gauge degrees of freedom from the variables
- With an appropriate choice of coordinates and/or decomposition of metric variables, the constraint equations are elliptic in nature; hence, must solve coupled elliptic-hyperbolic equations
- Potential advantages
  - eliminate "constraint violating modes"
  - minimize the number of variables/equations and hence additional constraints that need to be imposed
- Possible disadvantages
  - no guarantees that there will not be "evolution violating modes"
  - more difficult (impractical ?) to impose inner boundary conditions consistent with the full set of Einstein equations

11

Coupled Elliptic-Hyperbolic Systems with AMR

- Standard B&O time-stepping procedure works for hyperbolic equations because of finite propagation speeds and TE-driven hierarchy generation
  - Calculating the hierarchy based on TE estimates guarantees that the solution on the parent level in the vicinity of child boundaries is sufficiently accurate that interpolation is reasonable for setting child boundaries
  - Finite propagation speed prevents the poorly known solution at the interior of a coarse grid from polluting the parent-child boundary region between injection steps.
- Arguments for why B&O works for hyperbolic equations suggests it will not work for elliptic equations in general.

12

Coupled Elliptic-Hyperbolic Systems with AMR

- Modify the B&O algorithm to incorporate elliptic equations via "extrapolation and delayed solution":
  - solution of coupled hyperbolic/elliptic equations is split into two phases within the B&O recursive evolution scheme:
    - 1) hyperbolic equations are solved in the usual order, when descending the hierarchy from coarse to fine levels. During this phase, all elliptic variables are "evolved" via extrapolation from past time levels.
    - 2) the elliptic equations are solved when ascending the hierarchy from fine to coarse levels, after the injection step. Furthermore, after injection at level L, the elliptic equations are solved over the entire sub-hierarchy from levels L to L_f (finest level).

13

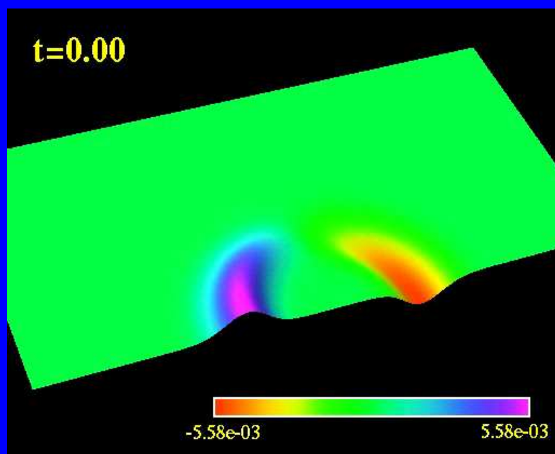
Coupled Elliptic-Hyperbolic Systems with AMR

- Empirically, the following extrapolation scheme was found to be stable
  - for level L, linear extrapolation using the solution at the two previous times when level L was in sync with level L-1 (one level coarser).
  - when a new (semi) global solution is obtained from level L up to (coarser) level L-d, causing the local solution at level L to change, one adjusts the past time variables used for extrapolation at level L to preserve extrapolation velocities, and add a "correction" to account for global shifts in the solution

14

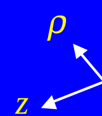
Sample fully constrained adaptive evolution:

Scalar field critical collapse in axisymmetry
(with M. Choptuik, S. Liebling & E. Hirschmann)



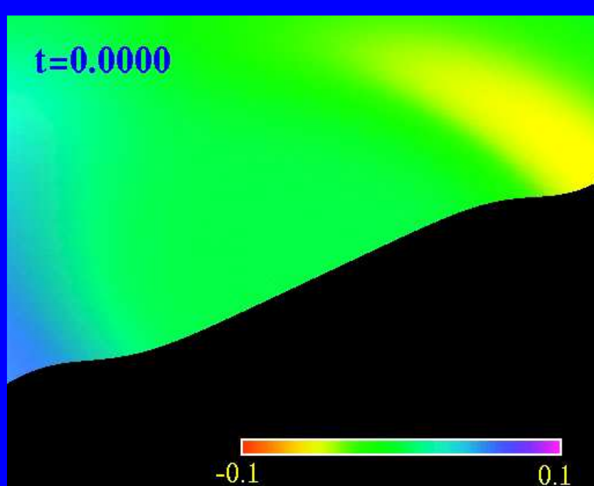
Scalar field, "anti-symmetric" initial data

Weak field evolution



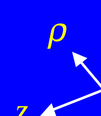
15

Scalar field critical collapse in axisymmetry

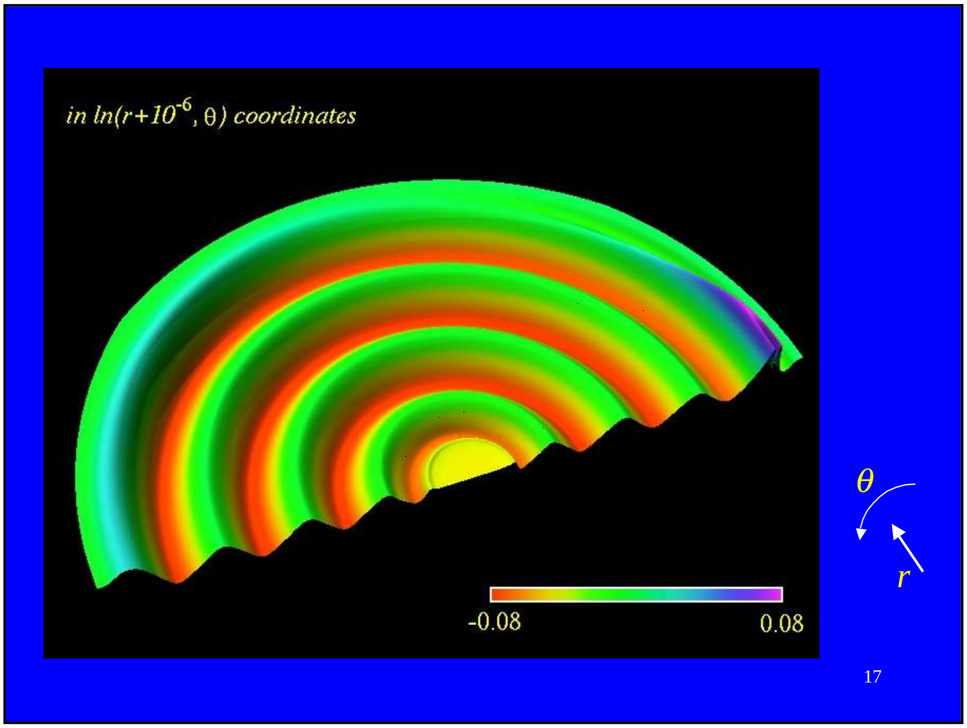


Scalar field,

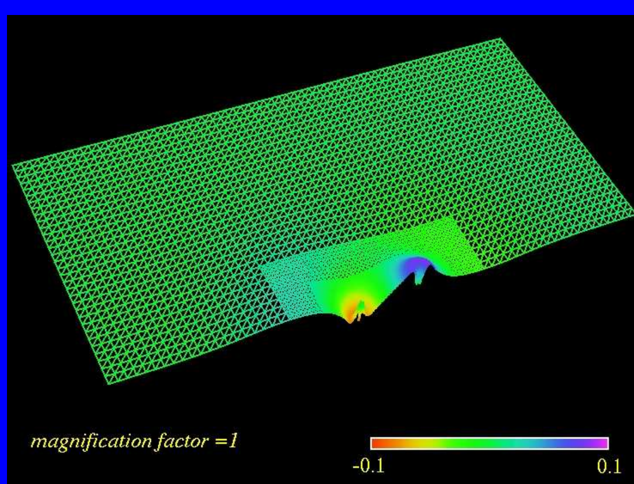
Near-critical collapse



16

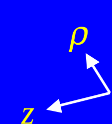


AMR grid hierarchy, last frame



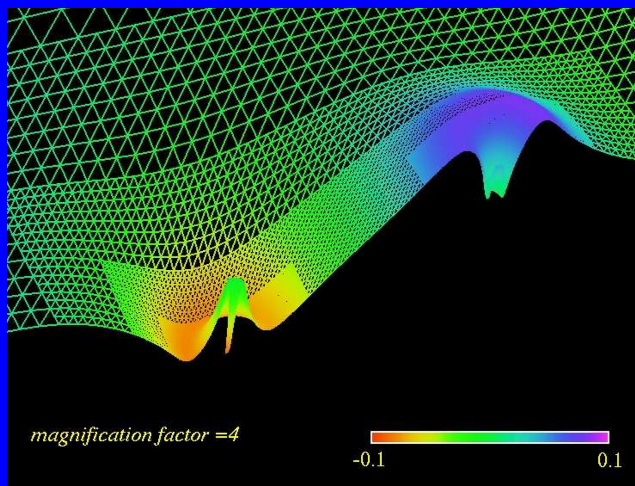
24(+1), 2:1 refined levels

(2:1 coarsened in figure)



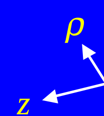
18

AMR grid hierarchy, last frame



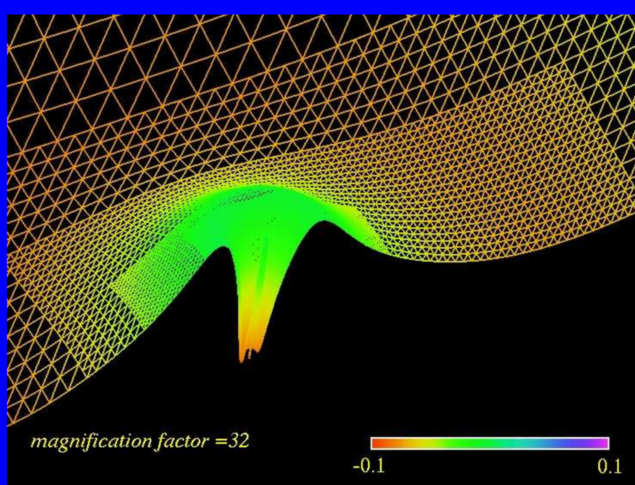
24(+1), 2:1 refined levels

(2:1 coarsened in figure)



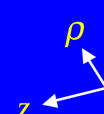
19

AMR grid hierarchy, last frame



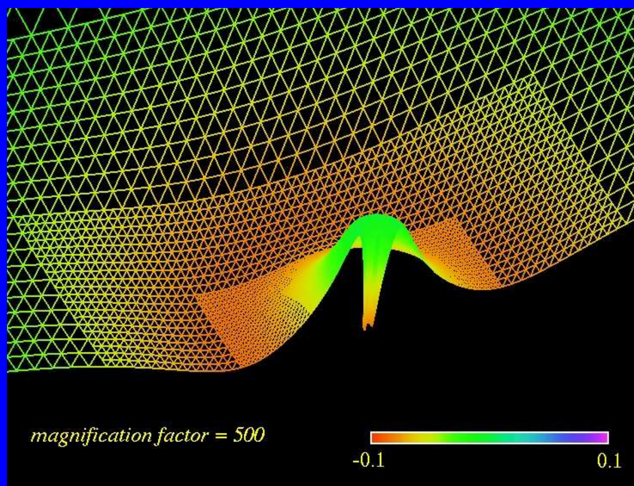
24(+1), 2:1 refined levels

(2:1 coarsened in figure)



20

AMR grid hierarchy, last frame



24(+1), 2:1 refined levels

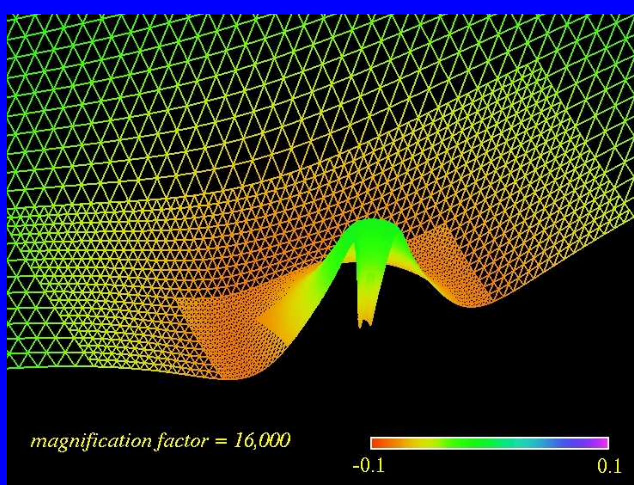
(2:1 coarsened in figure)

ρ

z

21

AMR grid hierarchy, last frame



24(+1), 2:1 refined levels

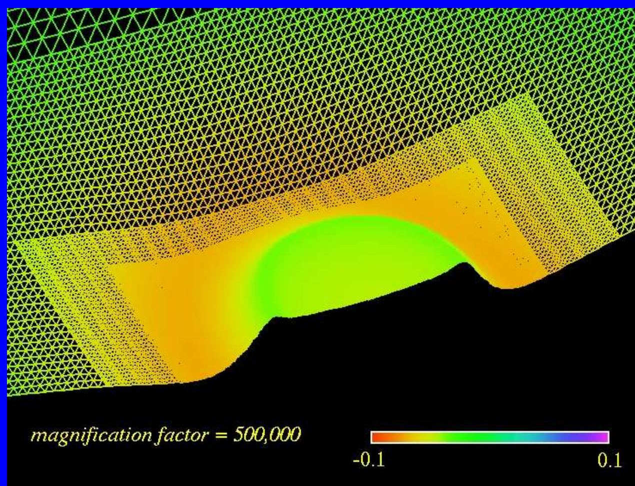
(2:1 coarsened in figure)

ρ

z

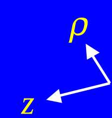
22

AMR grid hierarchy, last frame



24(+1), 2:1 refined levels

(2:1 coarsened in figure)



23

Sample fully constrained adaptive evolution:

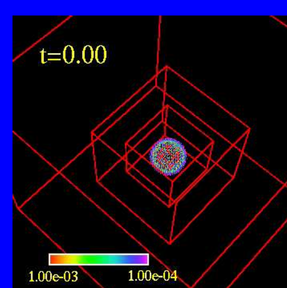
weak-field scalar field evolution in 3D (with M. Choptuik)

base grid= 32^3 ; 4, 2:1 levels (effective finest res. = 256^3)

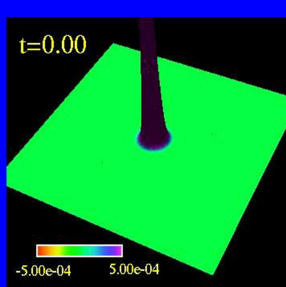
using compactified Cartesian coordinates

run-time: ~ several hours on a single 2.4 Ghz P4 (~ 25 "dynamical crossing times"),

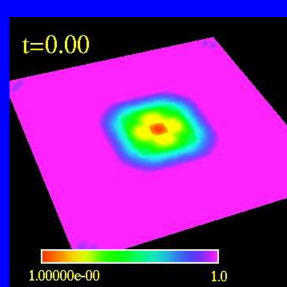
maximum memory usage ~ 60MB



Scalar field showing grid hierarchy



z=0 slice of scalar field



z=0 slice of lapse function (reaches minimum ~ 0.92)

24

AMR for characteristic codes

(with L. Lehner)

- In a characteristic evolution, one or two of the coordinates are null.
- We want an evolution algorithm for such a code that shares the desirable features of B&O AMR:
  - a grid-hierarchy composed of unigrid building blocks
  - dynamical regridding via local truncation error estimates
  - efficient use of resources along all coordinate dimensions

25

AMR for characteristic codes

- In certain restricted situations, the B&O scheme can be applied to a characteristic evolution by effectively treating one of the null coordinates as "space", the other as "time" (as in Garfinkle's algorithm)
- Cannot work in general, as there will be situations where effects are propagated "instantaneously" along the "space" coordinate
  - causality arguments justifying the use of interpolation to set fine-grid boundary conditions fail
  - will not be able to efficiently track fine-features propagating along the "space"-null direction (doing so correctly could cause an entire level to refine, i.e. get a filling factor of 1)

26

AMR for characteristic codes

- Solution is to consider both null directions as "time" in the B&O sense
  - causality then forces a particular recursive evolution scheme through the hierarchy
  - additional spacelike dimensions are treated in the same manner as B&O for Cauchy codes

27

Characteristic AMR example

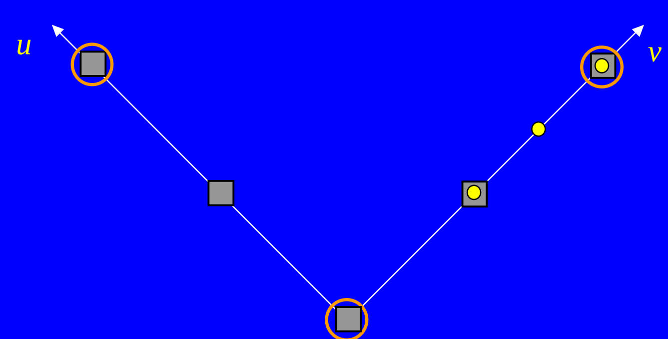
3 Levels ρ = 2:1

gridpoint symbols:

L=1

L=2

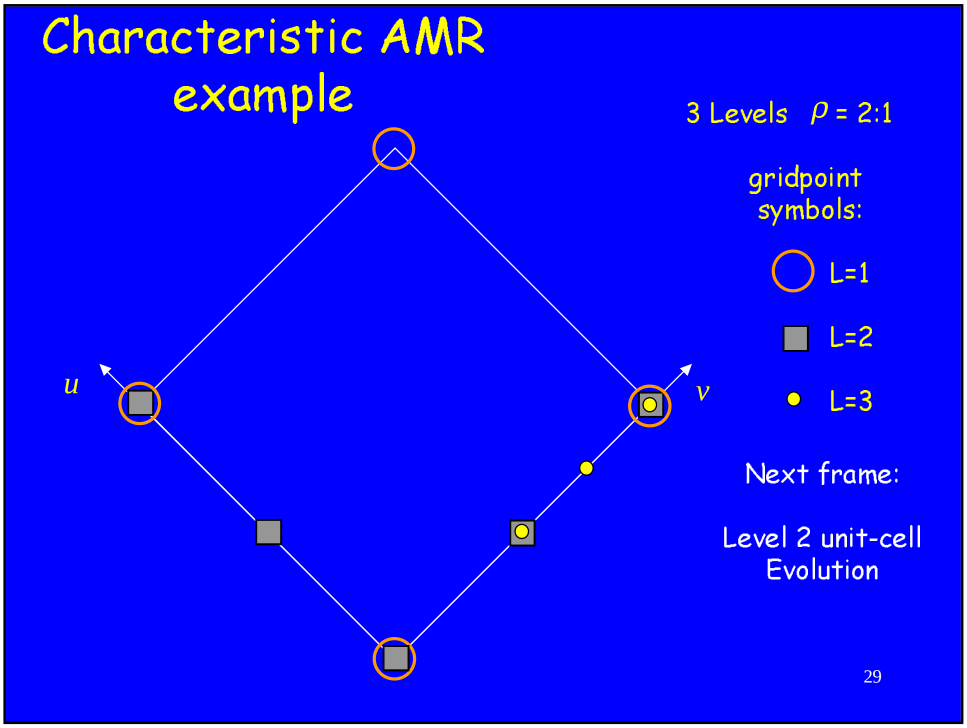
L=3

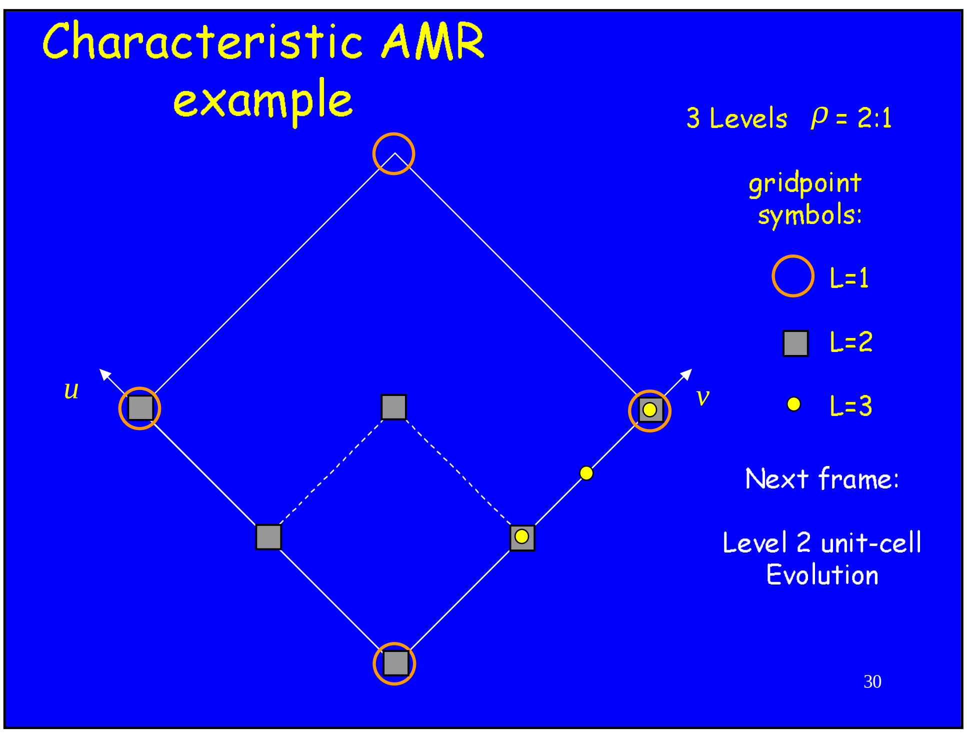


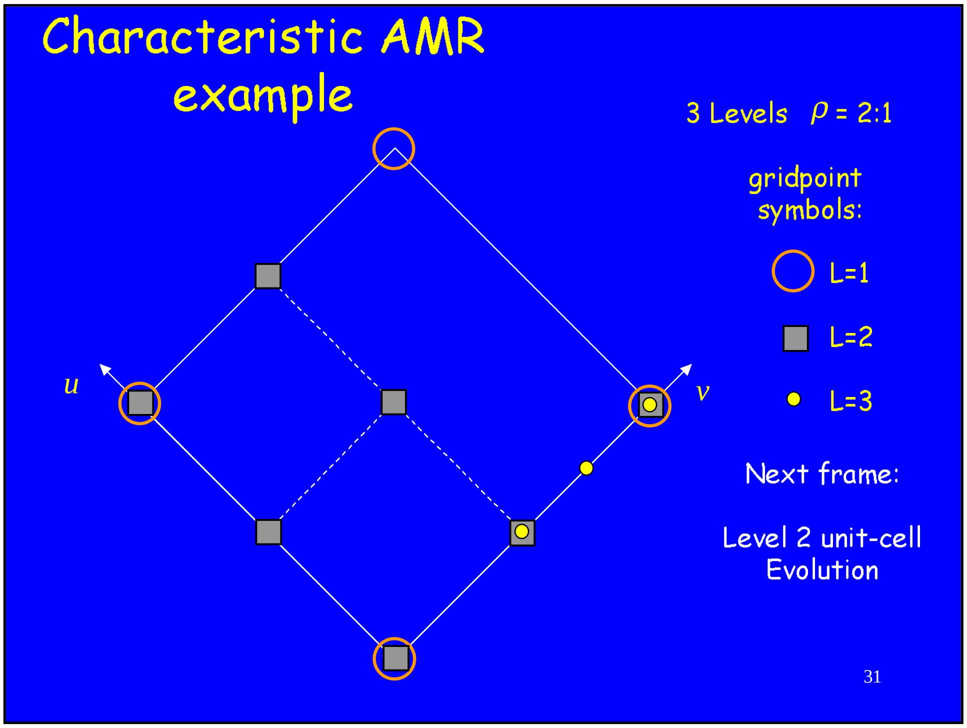
Next frame:

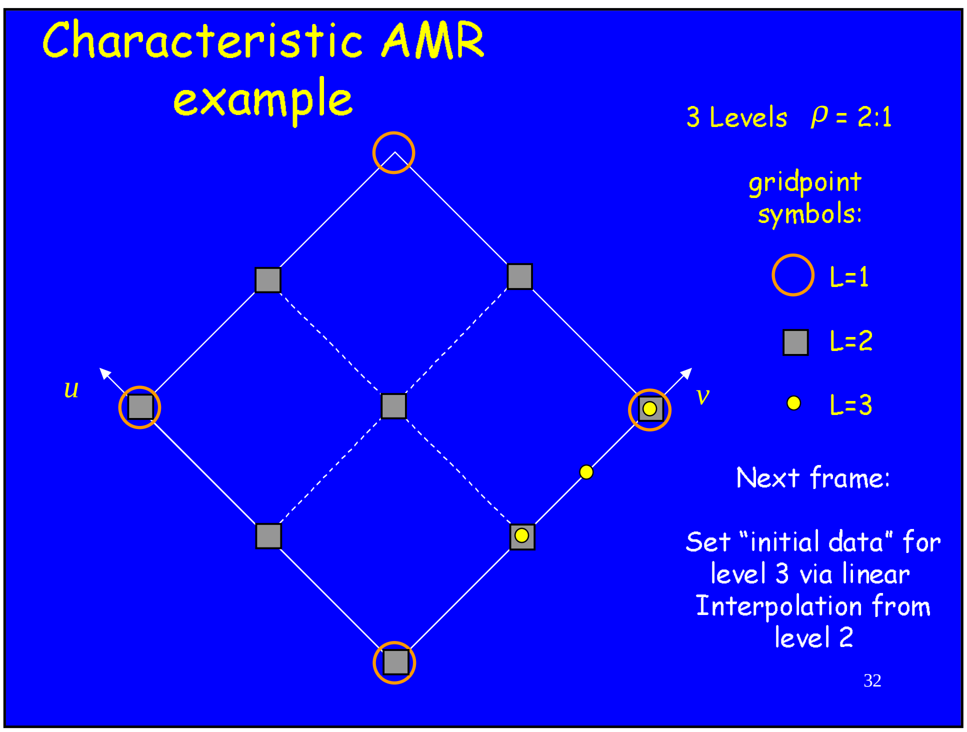
Level 1 unit-cell Evolution

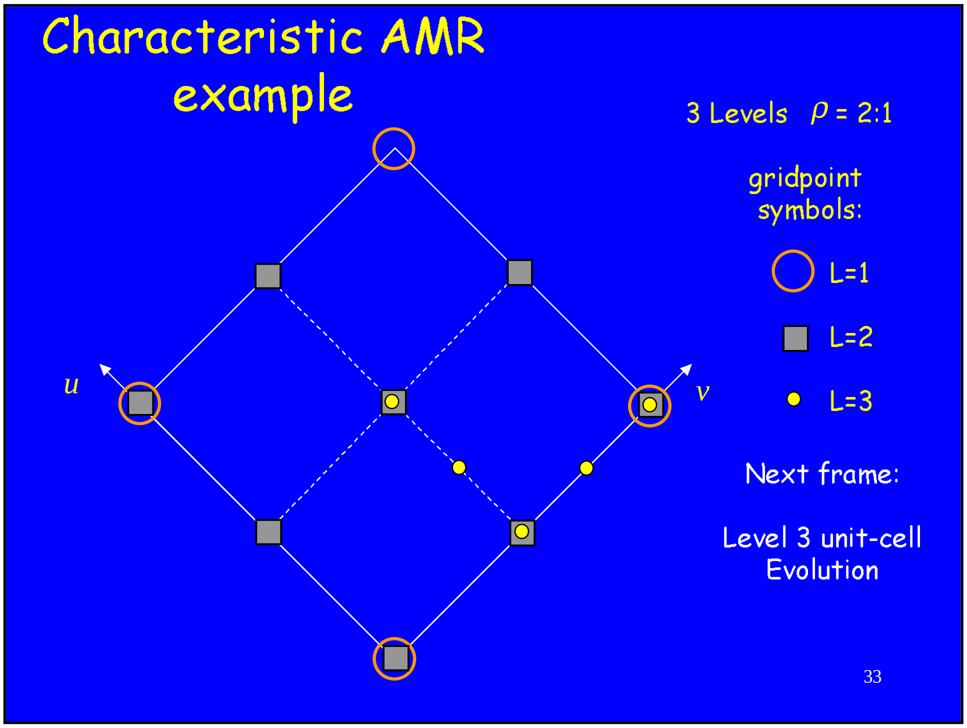
28

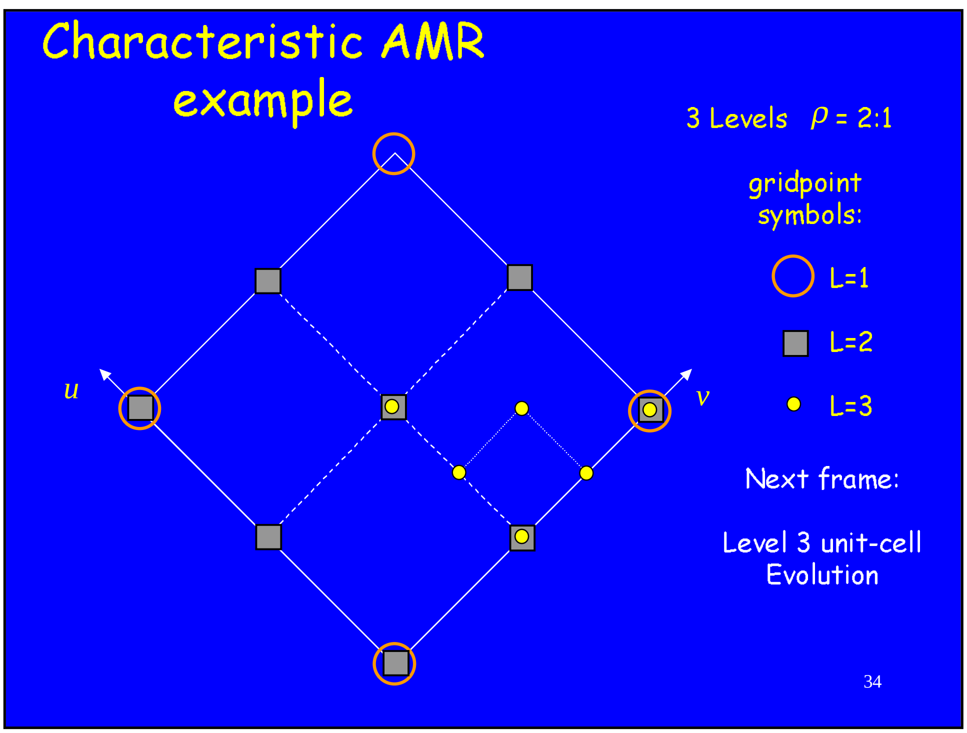

Characteristic AMR example
3 Levels ρ = 2:1
gridpoint symbols:
L=1
L=2
L=3
Next frame:
Level 2 unit-cell Evolution
u
v
29

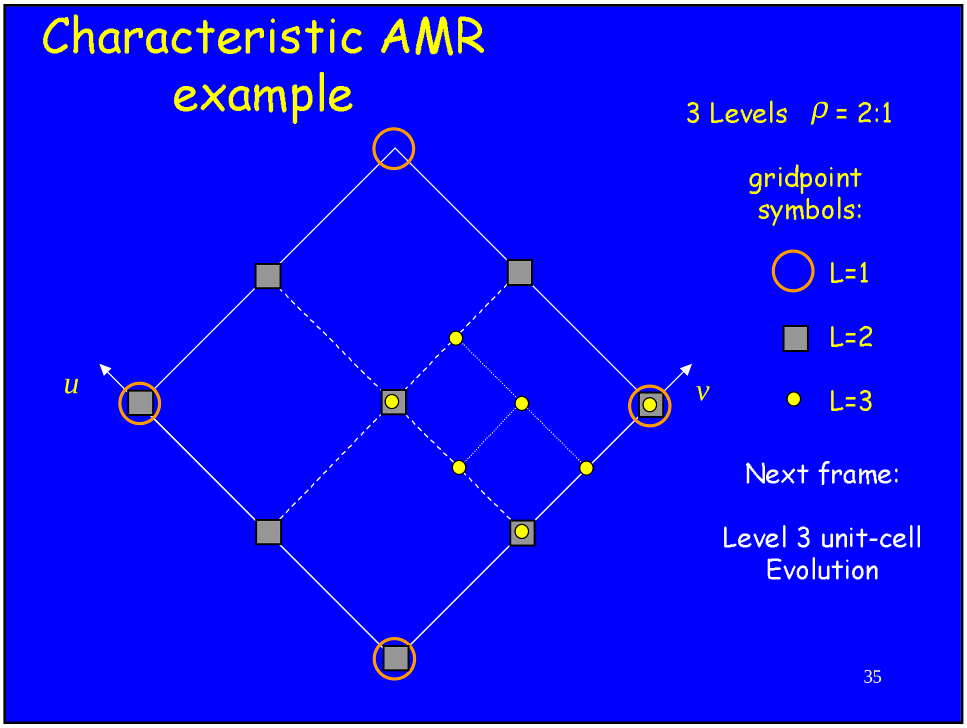

Characteristic AMR example
3 Levels ρ = 2:1
gridpoint symbols:
L=1
L=2
L=3
Next frame:
Level 2 unit-cell Evolution
u
v
30

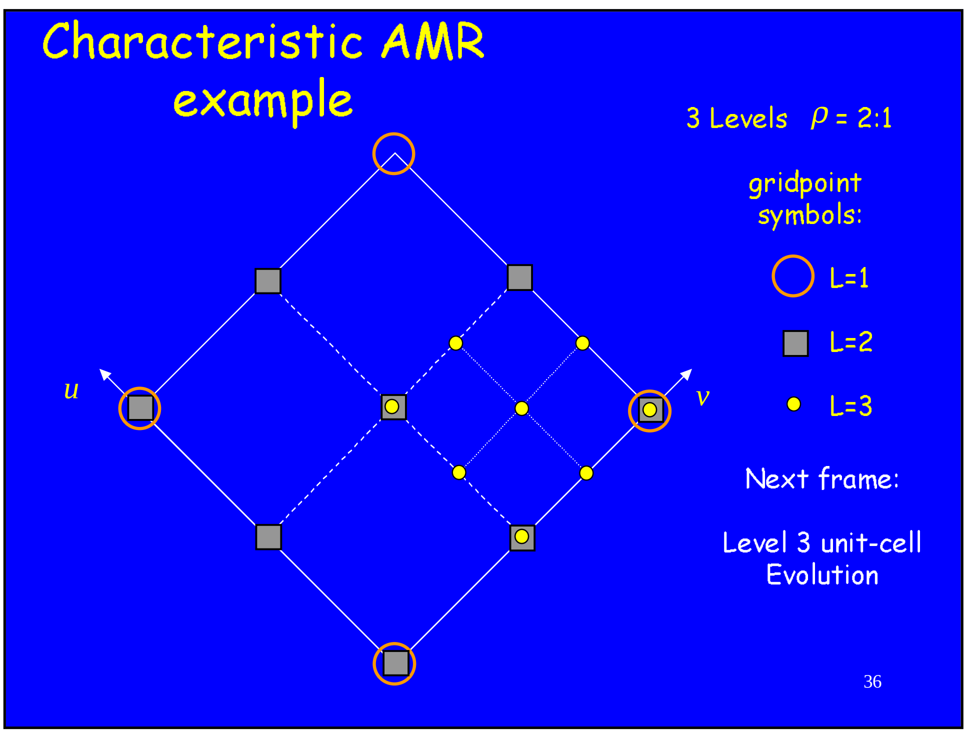

Characteristic AMR example
3 Levels ρ = 2:1
gridpoint symbols:
L=1
L=2
L=3
u
v
Next frame:
Level 2 unit-cell Evolution
31

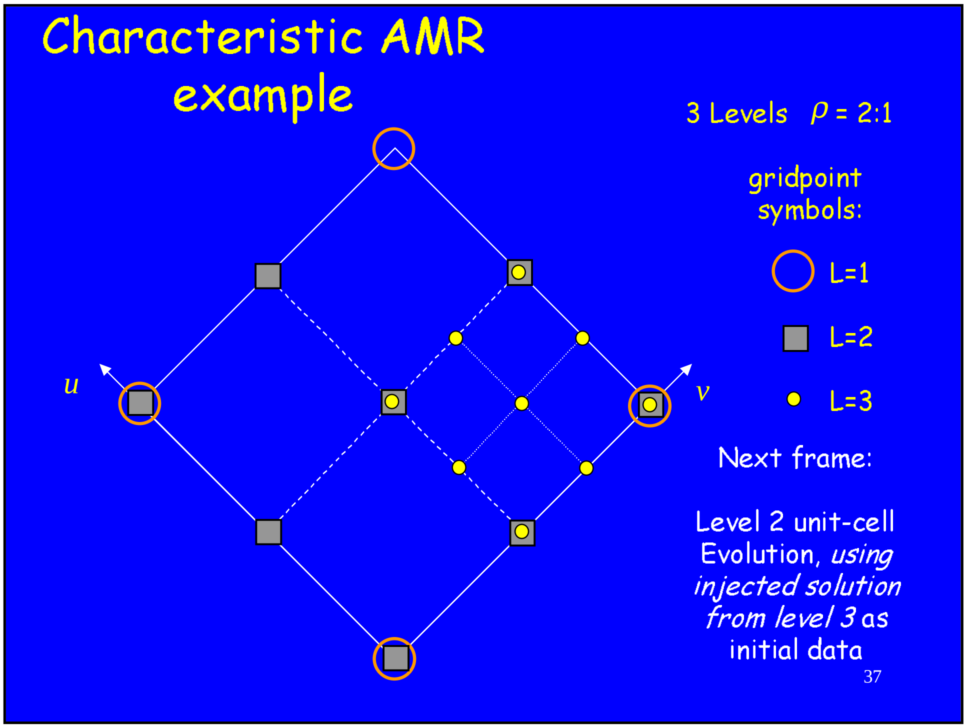

Characteristic AMR example
3 Levels ρ = 2:1
gridpoint symbols:
L=1
L=2
L=3
u
v
Next frame:
Set "initial data" for level 3 via linear Interpolation from level 2
32

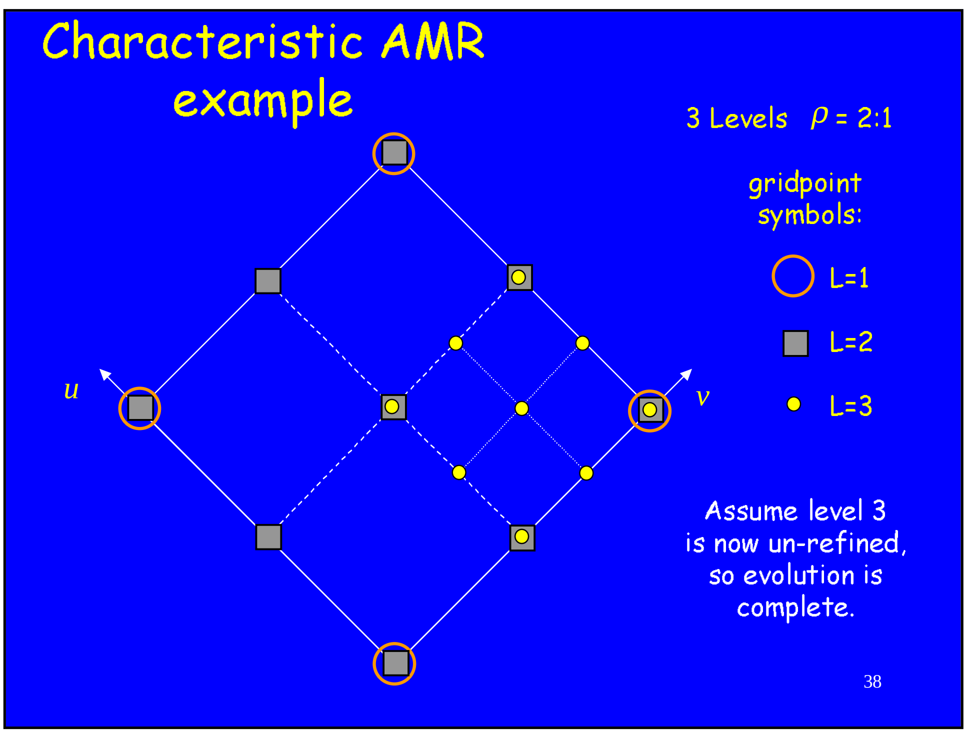

Characteristic AMR example
3 Levels ρ = 2:1
gridpoint symbols:
L=1
L=2
L=3
u
v
Next frame:
Level 3 unit-cell Evolution
33


Characteristic AMR example
3 Levels ρ = 2:1
gridpoint symbols:
L=1
L=2
L=3
u
v
Next frame:
Level 3 unit-cell Evolution
34


Characteristic AMR example
3 Levels ρ = 2:1
gridpoint symbols:
L=1
L=2
L=3
Next frame:
Level 3 unit-cell Evolution
u
v
35


Characteristic AMR example
3 Levels ρ = 2:1
gridpoint symbols:
L=1
L=2
L=3
Next frame:
Level 3 unit-cell Evolution
u
v
36


Characteristic AMR
example
3 Levels ρ = 2:1
gridpoint
symbols:
L=1
L=2
L=3
u
v
Next frame:
Level 2 unit-cell
Evolution, *using*
injected solution
from level 3 as
initial data
37


Characteristic AMR
example
3 Levels ρ = 2:1
gridpoint
symbols:
L=1
L=2
L=3
u
v
Assume level 3
is now un-refined,
so evolution is
complete.
38

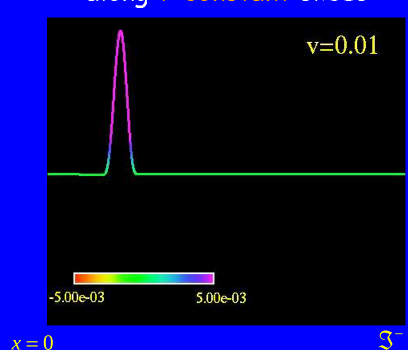
Sample 1D evolution

- From 1D massive scalar field collapse code
  - uses a single ingoing null coordinate v
  - uses a compactified spatial coordinate x : x=0 is the origin of spherical symmetry, and x=1 is $\mathfrak{I}^-$
  - black hole (if present) is treated via excision

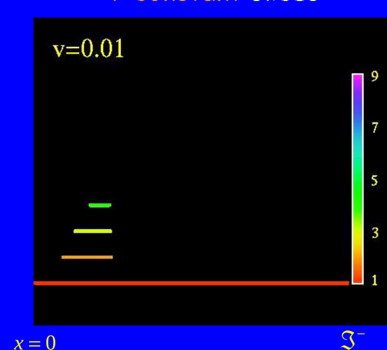
39

Outgoing, massive scalar field pulse in a black hole spacetime

Scalar field times r along v=constant slices



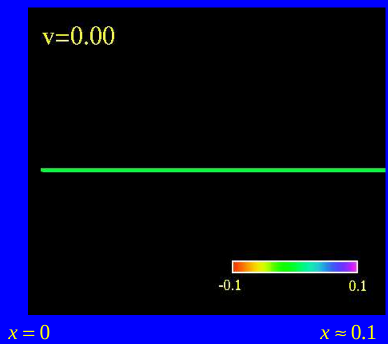
Depth of hierarchy along v=constant slices



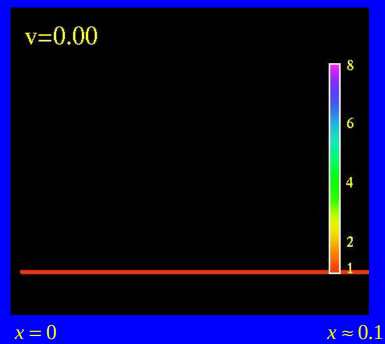
40

Ingoing, massless scalar field, "near" critical collapse

Scalar field along v=constant slices



Depth of hierarchy along v=constant slices



41

Conclusions

- AMR is and will continue to be an essential technique for finite-difference based solutions of many problems of interest in general relativity
- AMR is a "solved problem" compared to some of the other issues (stable evolution schemes, initial conditions, coordinate systems, boundary conditions, etc.) that must be dealt with if numerical relativity is to become a significant tool for gravitational wave astronomy.
  - certainly many "details" must be worked out to make AMR work in GR, but no fundamental problems
- Ratio of reward (speed-up) to implementation effort of AMR is potentially quite large, and faster run-times cannot but help in resolving other issues

42

Hierarchy Construction

- Arguably, the most sensible manner to construct a grid hierarchy is via truncation error (TE) estimates
- Traditionally, in Berger & Oliger AMR, TE estimates are calculated periodically via the following procedure:
  1. 2 copies of the relevant levels of the hierarchy are made: **I**, an identical copy, and *C*, a 2:1 coarsened copy
  2. copy *C* is evolved 1 time step, copy **I** is evolved **n** (typically 2) time steps until it is in sync with copy *C*
  3. the solution on *C* is subtracted from the solution on **I**, which by the usual Richardson expansion gives a function proportional to the solution error

43

Hierarchy Construction

- A shadow hierarchy (Choptuik) economizes the process of TE estimation by evolving in synchrony with the main hierarchy a 2:1 copy of it called the shadow hierarchy
  - cost in speed is small (~1/16 extra time needed in 3D)
  - offers a net savings in memory by not having to replicate the main hierarchy at regridding time
  - evolution procedure for the shadow is identical to that of the main hierarchy, except periodically the shadow grid functions are updated to match those in the main hierarchy

44

Hierarchy Construction

- Can further economize TE calculation by noting that information needed to compute the TE is automatically available prior to the fine-to-coarse grid injection phase within the B&O algorithm
  - i.e. the hierarchy "acts as it's own shadow" --- call this a self-shadow hierarchy (SSH) technique
- Can show that the TE computed with a SSH will differ from that computed via a shadow hierarchy (where the shadow is updated as frequently as possible) by an amount proportional to O(h^4)
- The only overhead for a SSH is that at least two levels are needed in the hierarchy for this to work
  - hence, require that the base level always be fully refined (or equivalently, define the desired "base level" to be level 2)
  - essentially no speed or memory penalty in problems with deep hierarchies
  - almost no additional computational infrastructure needed

45