LA-UR-05-7985

Approved for public release;
distribution is unlimited.

Title:	Two-dimensional convergence study for problems with exact solution: uniform and adaptive grids (LA-UR-05-7985, 2005)
Author(s):	Shengtai Li (T-7) William J. Rider (X-7) Mikhail J. Shashkov (T-7)
Date:	September 30, 2005

Two-Dimensional Convergence Study for Problems with Exact Solution: Uniform and Adaptive Grids

Shengtai Li, William J. Rider, and Mikhail J. Shashkov

Abstract

In this report, We have tested three examples in the convergence analysis of adaptive mesh refinement (AMR) calculation for two different AMR implementations. Test results shows that AMR may not achieve the convergence of equivalent finest uniform grid. In some cases, numerical results with AMR are even worse than those without AMR. AMR can even trigger instability for some problems.

The three examples have exact solutions so that we can calculate the numerical error exactly. The two AMR implementations include a patch-based (AMR-MHD) and a cell-based AMR (RAGE). *Characteristic length scales* (CLS) determines the convergence behavior. A smaller CLS means better convergence. Below is a comparison in CLS between two AMRs (Δ_{avg}=$\sqrt{\text{Total Area}/N}$, $\Delta x_{\min}$=minimal spacing)

	AMR-MHD			RAGE		
Problem	wave	vortex	Noh	wave	vortex	Noh
refinement	22%	22.5%	6%	11%	20%	96%
L_1-CLS	$\Delta x_{\min}$	$\Delta x_{\min} <$ CLS $\ll \Delta x_{avg}$	$\Delta x_{\min}$	$> \Delta x_{avg}$	$\gg \Delta x_{avg}$	$\gg \Delta x_{avg}$
L_2-CLS	$\Delta x_{\min}$	$\Delta x_{\min} <$ CLS $\ll \Delta x_{avg}$	$\Delta x_{\min}$	$> \Delta x_{avg}$	$\gg \Delta x_{avg}$	$\Delta x_{\min}$
L_∞-CLS	$\Delta x_{\min}$	$\Delta x_{\min} <$ CLS $\ll \Delta x_{avg}$	N/A	$> \Delta x_{avg}$	$\gg \Delta x_{avg}$	N/A

The comparison, combined with other test results, reflects the *ineffectiveness* of RAGE AMR for selected problems: *(a) it has large initialization error in the first step; (b) the refinement criteria do not work well; (c) results with AMR are worse than without AMR for high-resolution grid; (d) AMR with more than 1-level refinement does not work better than with only 1-level refinement.*

Two issues occur in both AMRs: (1) AMR with locked time step has larger dispersion error than corresponding fine uniform grid. (2) AMR can trigger *carbuncle* instability at shock front for cylindrical (r, z) coordinate near z-axes.

The overall error-cost performance of AMR calculations in RAGE is as follows

Problem	Error	$100^2(1)$	$100^2(2)$	$100^2(3)$	$200^2(1)$	$200^2(2)$	$400^2(1)$
wave	L_1 t=1	×	×	×	×	×	×
	L_2 t=1	×	×	×	×	×	×
	L_∞ t=1	√	√	√	√	×	×
wave	L_1 t=10	×	×	×	×	×	×
	L_2 t=10	√	√	√	×	×	×
	L_∞ t=10	√	√	√	√	×	×
vortex	L_1 t=1	√	×	×	×	×	×
	L_2 t=1	√	×	×	×	×	×
	L_∞ t=1	√	×	×	×	×	×
vortex	L_1 t=10	√	×	×	×	×	×
	L_2 t=10	√	×	×	×	×	×
	L_∞ t=10	√	√	×	√	×	×
Noh	L_1	×	×	×	×	×	×
	L_2	×	×	×	×	×	×
	L_∞	×	×	×	×	×	×

(√ means better, × means worse, than the uniform grid; $n^2(m)$ denotes $n \times n$ base grid with m AMR levels.)

1 Introduction

In this report, we describe a process of asymptotic convergence analysis for Eulerian, compressible hydrodynamics simulations that use locally adaptive mesh refinement (AMR) grid. This analysis compares the numerical errors of uniform grid and corresponding AMR grid to find possible defect of AMR implementations, and provides numerical estimates of the asymptotic convergence rate.

Convergence analysis of numerical solutions to partial differential equations (PDEs) is typically performed on a uniform, fixed mesh ([10, 11, 12, 13]). Such regular meshes greatly simplify both the method and implementation of the asymptotic convergence analysis. Previous report [15] extends the methodology to Lagrangian staggered grid hydrodynamics algorithms that involve unstructured dynamically evolving meshes.

AMR calculation has been popular in the past twenty years for its capabilities to handle problems with multiple length and time scales. AMR is straightforward in theory, which locally adds cells to where they are needed. In this report, we focus ourselves on structured AMR that has been used for Eulerian compressible hydrodynamics simulations. The structure AMR consists grids of different refinement levels. The refinement ratio between two adjacent levels is constant. Since Berger [1] proposed her first AMR algorithm, many AMR implementations and packages are available for research, such as CHOMBO, FLASH, DAGH, OVERTURE, RAGE. However, the verification analysis for AMR is still in its infancy. The reason is partially due to the complexity of AMR data structure and its detachment from the numerical algorithm. It is assumed in AMR community that *AMR should achieve the same accuracy in refinement region as the corresponding fine uniform grid*. However, we will see this assumption may not be true in many cases.

The structure AMR has at least two distinguish implementations: *patch*-based and *cell*-based. The patch-based AMR differs from the cell-based in its data structure and management. Clustering, which groups the refined cells into logically rectangular grids (called patch), can transform the cell-based AMR into patch-based. Most of AMR implementations are patch-based, including our in-house version AMR-MHD [7]. The RAGE code, however, is cell-based. It is outside of the scope of this report to provide a detailed explanation on the difference of the two AMR implementations; see more references for details.

Our test cases consists of three model problems. All of them have exact solutions. The first two problems are smooth and have periodic boundary conditions so that they can be run for a long time. The first problem is a simple linear wave propagation problem. The second one has the same initial conditions and exact solutions in density field but the pressure and velocities are not constant. The third problem has a discontinuity developed and propagated.

In all of the three examples we solve the compressible Euler equations of gas dynamics in 2-D:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x + \mathbf{g}(\mathbf{q})_y = 0 \tag{1}$$

where

$$\begin{aligned} \mathbf{q} &= (\rho, \rho u, \rho v, E)^T, \\ \mathbf{f}(\mathbf{q}) &= (\rho u, \rho u^2 + p, \rho uv, u(E+p))^T, \\ \mathbf{g}(\mathbf{q}) &= (\rho v, \rho uv, \rho v^2 + p, v(E+p))^T. \end{aligned}$$

Here ρ is density, (u, v) is the velocity, E is the total energy, p is the pressure, and

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2) \tag{2}$$

where γ is a constant related to the ideal gas.

RAGE uses a different hydro solver (see [8]) from AMR-MHD. As an independent comparison and check for the code implementation, we have also implemented the hydro solver of AMR-MHD into RAGE as an alternative solver. This hydro solver is a general flux-based hydro-solver and is often called MUSCL-Hancock scheme. It first predicts the space-time centered values at the cell-interface by Hancock predictor, and then uses a Riemann solver to evaluate the flux at the interface.

All of our computation for RAGE code uses version rage20041126.032. Gisler reported convergence studies using version rage20040220.003 in [9]. We are informed that there is no big change in hydro-algorithm and AMR structure between these two versions.

The outline of this report is as follows. We first describe a general background for convergence analysis and AMR in Section 2. Then we test the three examples with uniform and AMR grid.

2 Background

2.1 General Error Ansatz

The general global error ansatz for combined space and time error analysis proposed in [14] is

$$\xi^* = \xi + A(\Delta\tilde{x})^p + B(\Delta\tilde{t})^q + C(\Delta\tilde{x})^r(\Delta\tilde{t})^s + H.O.T,$$

where A, B, C are constant convergence coefficients, p, q, r, s are the convergence rate. In this report, we restrict our attention only to the case of spatial convergence, i.e., we solve the standard discretization error model

$$||\xi - \xi^*|| = A(\Delta\tilde{x})^p, \tag{3}$$

where ξ and ξ^* are the computed and exact solution vectors. The norm $||\cdot||$ can be L_1, L_2, and L_∞, defined as

$$\begin{aligned} ||f||_1 &= \frac{1}{V}\int_\Omega |f(\mathbf{x})| d\mathbf{x} \\ ||f||_2 &= \sqrt{\frac{1}{V}\int_\Omega |f(\mathbf{x})|^2 d\mathbf{x}} \\ ||f||_\infty &= \max_{\mathbf{x}\in\mathbf{\Omega}} |f(\mathbf{x})| \end{aligned}$$

where $V = \int_\Omega d\mathbf{x}$ is the total volume of the cells. In discrete space, these quantities can be approximated as

$$V = \sum_k \Delta V_k$$

$$
\begin{aligned}
||f||_1 &= \frac{1}{V}\sum_k |f(\mathbf{x}_k)|\Delta V_k \\
||f||_2 &= \sqrt{\frac{1}{V}\sum_k |f(\mathbf{x}_k)|^2\Delta V_k} \\
|f||_\infty &= \max_k |f(\mathbf{x}_k)|
\end{aligned}
$$

The characteristic length scale, $\Delta\tilde{x}$, is a constant for uniform grid. For non-uniform grid such as AMR, we attempt to find a $\Delta\tilde{x}$ that has similar convergence behavior as the uniform grid. Note that the smaller the $\Delta\tilde{x}$, the better convergence we achieve.

2.2 AMR algorithm

In this subsection, we describe AMR data structure and management in brief.

2.2.1 AMR-MHD

We used the Berger-Colella [2] AMR strategy in AMR-MHD to handle those regions that need fine grid resolution. The detail of our implementation for general time-dependent PDEs was described in [3, 4].

Given a grid, we first flag those cells that require refinement and then cluster them into several rectangular grids, called *patch*. For each newly-generated patch, we interpolate the solutions from the coarse grid to the new grid. If the old mesh already has some refinement that overlaps with the new mesh, the solution should be copied from the old fine mesh to the new one. The whole regridding procedure from the old mesh solution to the new mesh solution is called *prolongation*. If the initial conditions are defined analytically, the initial conditions on the new grid are also defined analytically. Each patch can be treated as a single grid. Further refinement can be done recursively until no refinement is needed or the finest refinement level is reached.

During time integration, the coarse grid advances first, and then the fine grid. Boundary conditions of the fine grid are defined by either external boundary conditions, or adjacent sibling patches, or interpolations from the coarse parent grid. After advancing the fine grid, we transfer the more accurate fine grid solutions to the parent coarse grid. This procedure is called *restriction*. Restriction could cause a loss of conservation at interface between a fine and a coarse grid. Berger and Colella [2] proposed a flux correction step to secure that the consistent fluxes are used at the fine-coarse interface.

How to flag the cells has crucial impact on numerical errors. Several approaches are available in the literatures. We have implemented some of them in AMR-MHD. For the test problems in this report, we used a simple but effective approach that has been described in [3, 4]. Take a 2-D example, we use an error monitor function defined as

$$
\begin{aligned}
\mathrm{SPCMON}(i,j) := \max_k \mathrm{SPCTOL}(k)(&\quad |\Delta x^2 u^k_{xx}(i,j)| \\
&+ \quad |\Delta y^2 u^k_{yy}(i,j)|)
\end{aligned} \tag{4}
$$

where

$$
\mathrm{SPCTOL}(k) := \frac{\mathrm{SPCWGT(k)}}{\mathrm{UMAX}(k)\cdot \mathrm{TOLS}}.
$$

is defined for each grid cell for the kth solution component $u^k(i,j)$. The weighting factor $0 < \text{SPCWGT} \leq 1$ is a user specified parameter to indicate the relative importance of a PDE component. The parameters UMAX denotes the approximate maximum absolute value for each component, and TOLS is the spatial error tolerance. The second order derivative is approximated by a three point finite-difference method. For Euler equations, we use only the density profile in (4), i.e., SPCWGT(1)=1 and SPCWDT(2:4)=0. We initial a level refinement if there is a point where $\text{SPCMON}(i,j) > 1.0$, and then all grid points with $\text{SPCMON}(i,j) > 0.5$ are flagged.

For a dimensional-split algorithm, the time stepsize is usually determined by the maximum CFL number in each direction,

$$\text{CFL} = \delta t \cdot \max\left\{ \frac{|v_x| + c_s}{\delta x}, \frac{|v_y| + c_s}{\delta y}, \frac{|v_z| + c_s}{\delta z} \right\}, \tag{5}$$

For most of schemes, CFL should be no greater than 1. Note that the time stepsize is usually determined by the finest grid where the local spacing is smaller. Berger and Colella used a local time step method to improve the efficiency. This local time step method allows the fine grid takes smaller time step than the coarse grid before they reach the same time level.

Two concerns lead to the local time step for AMR: efficiency and accuracy. First, the local time step approach can save the computational effort associated with advancing the solution on the coarser levels with unnecessarily fine time steps. Second, a larger number of fine time steps to advance the solution on a coarser level will introduce more numerical diffusion and dispersion by accumulated effects. The accuracy concern is not issue in many applications. This is because the most interesting regions are covered by the fine levels and those that do not require refinement contain not very much structure. Another reason is that the accuracy may increase as the time step gets smaller. The latter reason is true only for some specific discretization and smooth problems.

Numerical simulations shows that the local time step can greatly improve the computational efficiency for one-dimensional applications. However, we obtain only slightly improvement in efficiency for two and three dimensional problems. We have also found that the local time stepping had difficulties in implicit time integration or high order method of lines (MOL) approach, because the local time stepping requires the boundary values (at the coarse-fine interface) that must be interpolated in both time and space from the coarse grid. Furthermore, the local time step also require extra memory allocated to store the temporary copies of the solution for time interpolation. The local time step approach is also difficult to achieve the load balance for large-scale parallel computation, because the time integration must be done sequentially for different levels. That is, the coarse level must be advanced first to prepare the boundary conditions for the fine levels. Due to this reason, most of available parallel AMR packages (e.g., ASC RAGE and FLASH code) use *locked time step* to advance the fine and coarse levels simultaneously.

The locked time step approach advances all of the grid points with the same time step, and hence the stepsize will be determined by the finest grid cells. The coarse grid has to use this small time step even though it can take a large time step locally. In this report, we will study how the locked time step impacts the convergence analysis by using the AMR-MHD package.

A natural consequence of the "patch-based" AMR, such as AMR-MHD, is the use of "ghost" cells to link a block to its neighboring blocks of both higher and lower resolution. The advantage of the patch-based AMR is that each block is a regular Eulerian grid and legacy code for a single grid can be reused easily. The disadvantages is that any curved contour of the mesh is resolved with extraneous cells, which may increase the computational work.

2.2.2 RAGE AMR

The RAGE hydro-code uses a cell-based AMR package, called continuous AMR (CAMR). It has been documented in detail in RAGE manuals. CMAR refines or enlarges each cell of the AMR mesh at each time step, depending on the local activity in that cell. Advantages of using the CAMR code include more efficient use of the computational cells. The challenge of CAMR is to create a robust data structure and import the existing code that developed for a single grid. It has only the locked time step method. The time stepsize in RAGE is controlled by two parameters: TSTAB and CSTAB. TSTAB is defined as an upper bound for

$$\text{TSTAB} \geq \delta t \cdot \left\{ \frac{|v_x|}{\delta x} + \frac{|v_y|}{\delta y} + \frac{|v_z|}{\delta z} \right\}, \tag{6}$$

and CSTAB is defined as an upper bound for

$$\text{CSTAB} \geq \delta t \cdot \max \left\{ \frac{|v_x| + c_s}{\delta x}, \frac{|v_y| + c_s}{\delta y}, \frac{|v_z| + c_s}{\delta z} \right\}, \tag{7}$$

where c_s denotes the sound speed, v_x, v_y and v_z are velocities in each direction. CSTAB is actually an upper bound for the CFL number. TSTAB is an auxiliary time step stability factor. It limits the time step according to velocity constraints. The manual describes the time step is limited by

$$\max[v_x/\delta x, v_y/\delta y, v_z/\delta z] \cdot \delta t \leq \text{TSTAB}. \tag{8}$$

However, the code implementation (release version 20050331.000) uses (6) instead of (8). These two are quite different for multi-dimensional problem. The default values of CSTAB and TSTAB are 0.9 and 0.2. Note that a TSTAB of 0.2 may correspond to a CFL number of 0.1 in many cases.

3 Linear Wave Propagation Problem

We first test a simple problem where velocity and pressure remains constant during the time evolution. The Euler equations can be reduced to a linear wave equation with respect to density ρ. However, we still solve the full Euler equations for the solutions. As pointed out in [14], one caveat to this problem is that it tests only the linear fields in the governing Euler equations.

The problem is subject to the following initial conditions:

$$\begin{aligned} \rho_{t=0} &= \left(1 - \frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}\right)^{1/(\gamma-1)} \\ (u,v,p)_{t=0} &= (1,1,1), \end{aligned}$$

where $r^2 = x^2 + y^2$ and $\epsilon = 5$. The density is chosen so that only a small region needs refinement if AMR is used. The computational domain is defined as $[-5,5] \times [-5,5]$, where periodic boundary conditions are adopted in both directions. The exact solutions for Euler equations with the above initial and boundary conditions is just a linear advection of the density profile with the constant velocity (1,1).

3.1 Results of AMR-MHD

We first use the package AMR-MHD and solve the problem with four different uniform grids: 40×40, 80×80, 160×160, and 320×320. The CFL number is kept around constant during the time integration. To study the impact of CFL number on accuracy of numerical solutions, we have tested two CFL numbers in this example:CFL=0.9 or CFL=0.45. Fig. 1 shows the errors in L_1, L_2, and L_∞ norm for different uniform grids. The L_1 and L_2 error indicates that the solutions reach the asymptotic range after 80×80 resolution.

Fig. 2 shows the impact of CFL number on numerical errors. Note that the solutions with a smaller CFL number are more accurate for the low resolutions, and less accurate for the high resolutions. The CFL number has more impact on the L_∞ error than on the L_1 and L_2 error.

To test the accuracy of AMR methods, we used three base uniform grids: 40×40, 80×80, and 160×160. The error tolerance is TOLS=0.002, by which we catches as much as possible the large error region for 80×80 base grid. We had used TOLS=0.001, which generates the same refined region as TOLS=0.002 for 80×80 base grid in the first step. The refinement ratio between different levels in AMR is two. The overall CFL number is kept around 0.9. Fig. 3 shows the numerical errors for the AMR results as well as the uniform grid result with different CFL numbers. It is easy to see that for the base grids 40×40 and 80×80, the AMR results with either local time step method or locked time step are almost identical to those obtained with the finest resolutions. In the case of base grid 160×160, the AMR achieves the same accuracy as the finest grid for the L_∞ error and less accuracy for the L_1 and L_2 errors. This is because the refinement region for the high resolution base grid is much smaller than for the low resolution base grid due to a fixed TOLS. Note that the AMR with the locked time step method achieves almost the same accuracy as with the local time step method.

Fig. 4 shows the convergence results of the AMR method at different times with respect to (w.r.t.) different length scales:

- $\Delta x_{\min}$, defined as the spacing of the finest level grid;
- $\Delta x_{\max}$, defined as the spacing of the base grid (the coarsest level);
- $\Delta x_{\text{avg}} = \sqrt{\text{Total area}/Nc}$, defined as the average spacing of the whole AMR grid, where Nc is the total number of cells;

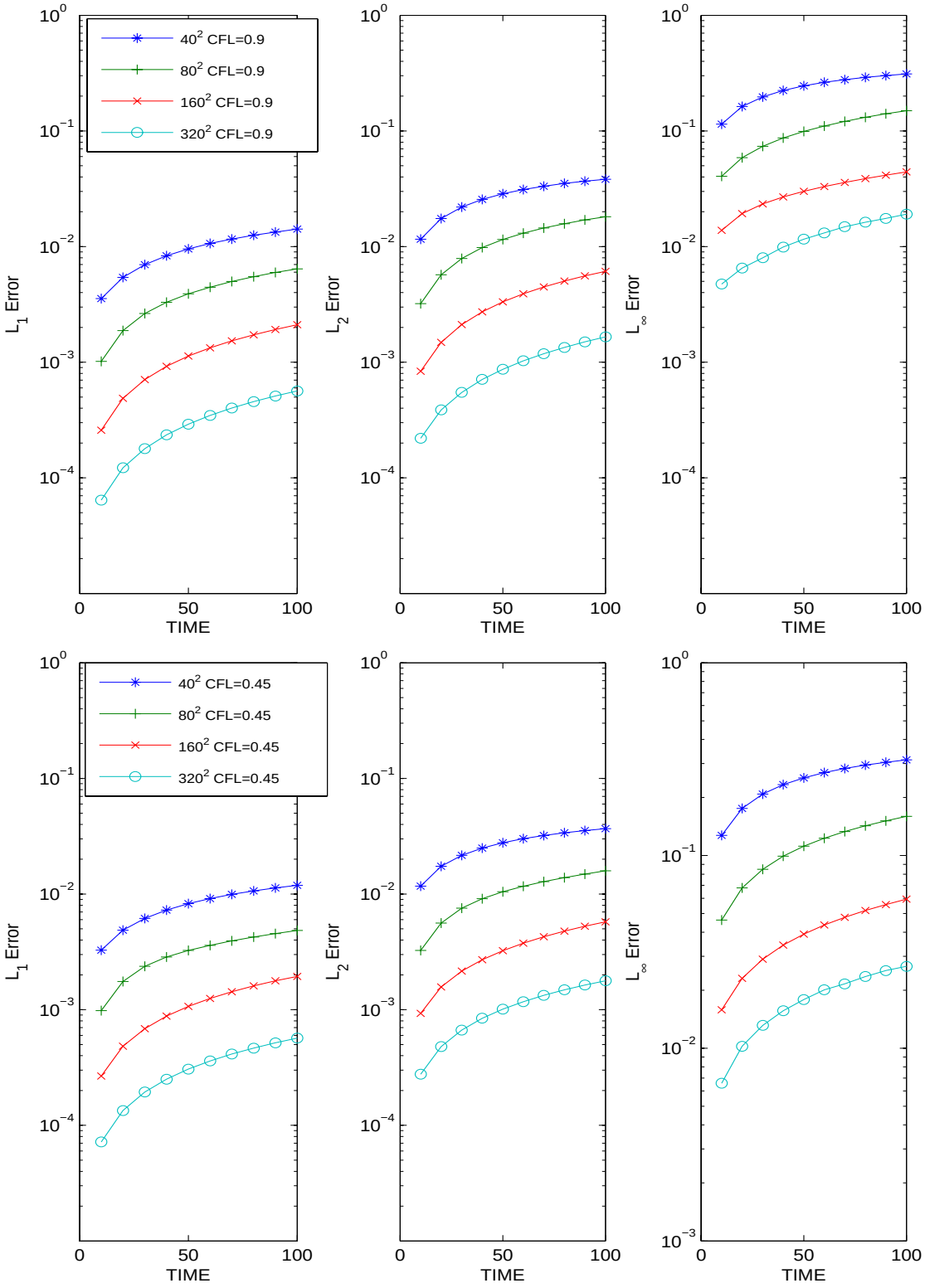

Figure 1: Numerical errors (L_1, L_2, and L_∞) for the linear wave propagation example at different times for different uniform grids. AMR-MHD is used.

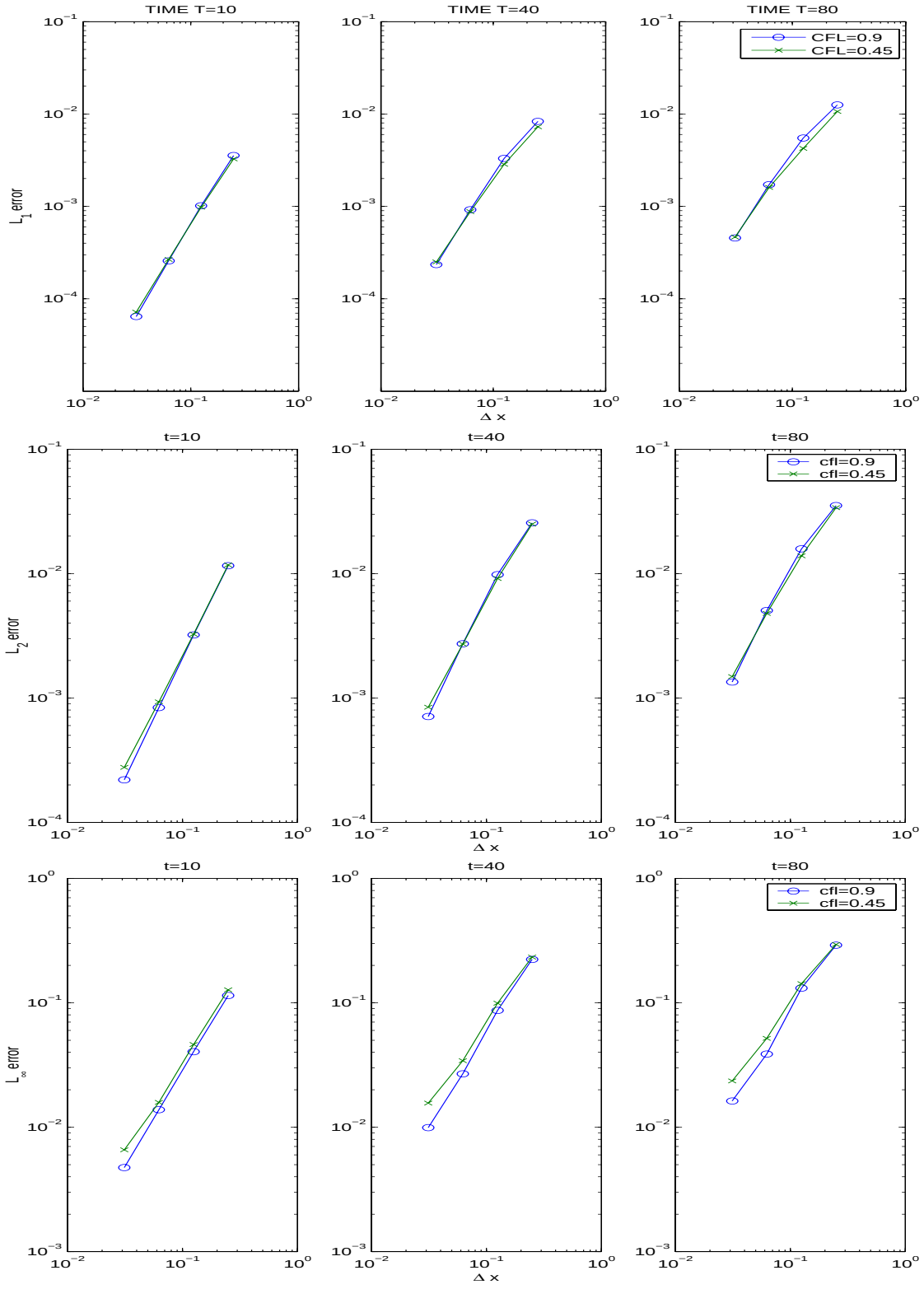

Figure 2: Numerical errors (L_1, L_2, and L_∞) for the linear wave propagation example at different times for different CFL numbers. AMR-MHD is used.

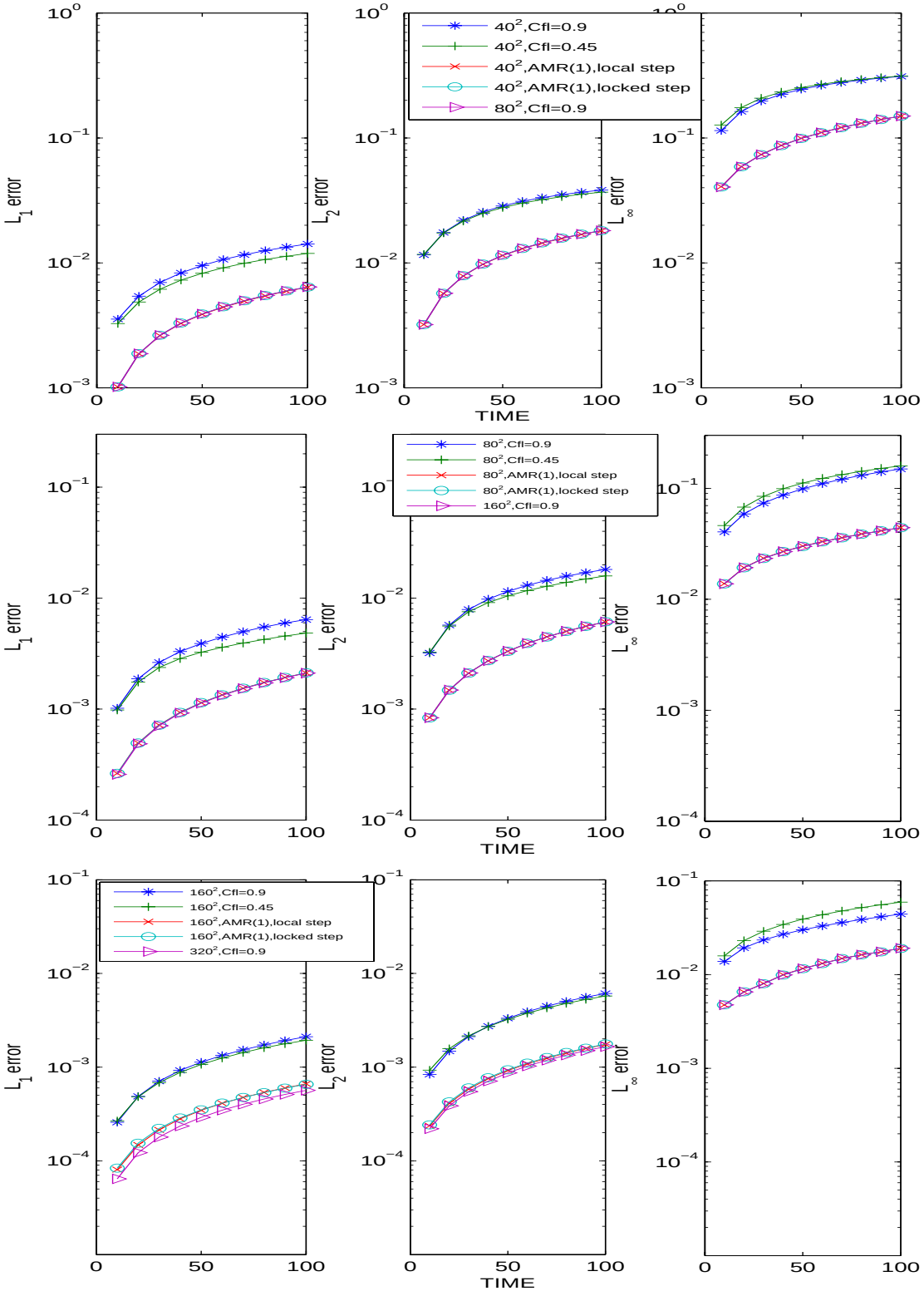

Figure 3: Numerical errors (L_1, L_2, and L_∞) for the linear wave propagation example at different times for different CFL numbers. AMR-MHD is used

- $\Delta x_{\text{mean}} = \sum \Delta x_i / Nc$, defined as the mean spacing of the whole AMR grid.

All of the four characteristic length scales (CLS) have been used in the convergence analysis of Lagrangian dynamic moving grid in [15]. The results in Fig. 4 indicates that the convergence length scale ($\Delta\tilde{x}$) in Eq. 3 for the AMR method is the minimum spacing, $\Delta x_{\min}$, i.e., *the finest resolution that has been used*. This is more obvious for the results of the low resolutions and for error norm L_∞. Therefore, we can say that the AMR works well for this problem just as expected.

For reference and comparison, Fig. 5 shows the density plots at different times.

3.2 Results of RAGE

For RAGE package we solved the problem with four different uniform grids: 100×100, 200×200, 400×400, and 800×800. The default values of CSTAB (0.9) and TSTAB (0.2) are used. Since we use periodic boundary conditions, the solution returns to the initial one after every 10 seconds. Fig. 6 shows the results at t=10, 50 and 100 for 800×800 grid, where the density profile is preserved very well after such a long integration. We expect AMR in RAGE works as expected as for AMR-MHD package. However, the plots in Figure 7, which show the results of 100×100 base grid with different refinement levels, indicate the AMR results get corrupted even at time $t = 10$. To see it clearly, we plot the grid refinement and error distribution in Fig. 8 at different times.

The L_1, L_2 and L_∞ errors and convergence orders are calculated for all grids. Figures 9 and 10 show the results for uniform grid. We also calculated the maximum CFL number used for each grid, which is roughly 0.267. In term of L_1 norm, the RAGE algorithm achieves the 2nd order accuracy for uniform grid.

To study the effect of CFL number on the accuracy of the numerical solutions, we also tested the case of TSTAB=0.5, which corresponds to CFL=0.63. The results are shown in Figure 11. For some unknown reasons, RAGE does not allow a TSTAB larger than 0.5, which can cause the instability. For comparison, we also showed the results of the MUSCL-Hancock scheme in RAGE. Note that for MUSCL-Hancock, the CFL number can be as large as one. The results of AMR-MHD (see Figure 3) shows that for this problem, the smaller the CFL number, the more accurate the solutions are. However, Figure 11 shows a different story: the smaller the CFL number, the less accurate the solutions are. This is true even with the same hydro solver MUSCL-Hancock. The reason is not clear to us by now.

To test the accuracy and convergence of the AMR method, we used three base uniform grids: 100×100, 200×200, and 400×400. The refinement ratio between different levels in AMR is taken as two. The default values of CSTAB and TSTAB are used in time step control. Figure 12 shows the results for 100×100 base grid with different refinement levels compared with the same resolution uniform grids. It is clear that AMR cannot achieve the same accuracy as the uniform grid with the same resolution. This is in contrast to the AMR-MHD results in previous subsection. We also noticed that the results for the three-level refinement are even worse than those with just one-level or two-level refinement. For L_∞ norm, we noticed in Figure 12 that the one-level refinement achieves almost the same accuracy as the uniform grid with the same resolution. However, more refinement levels generate even worse results at later times.

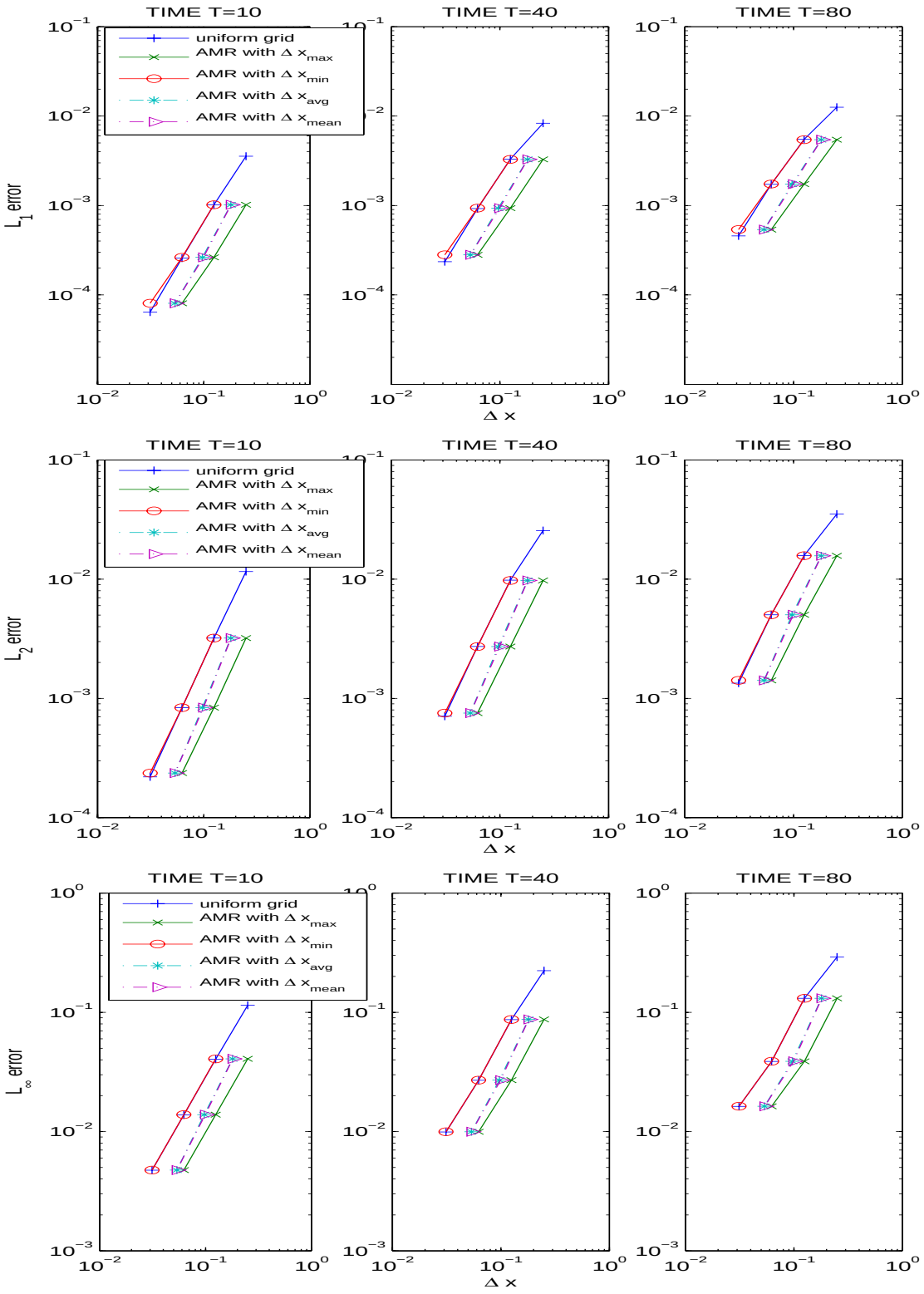

Figure 4: Numerical errors (L_1, L_2, and L_∞) for the linear wave propagation example at different times. AMR-MHD is used

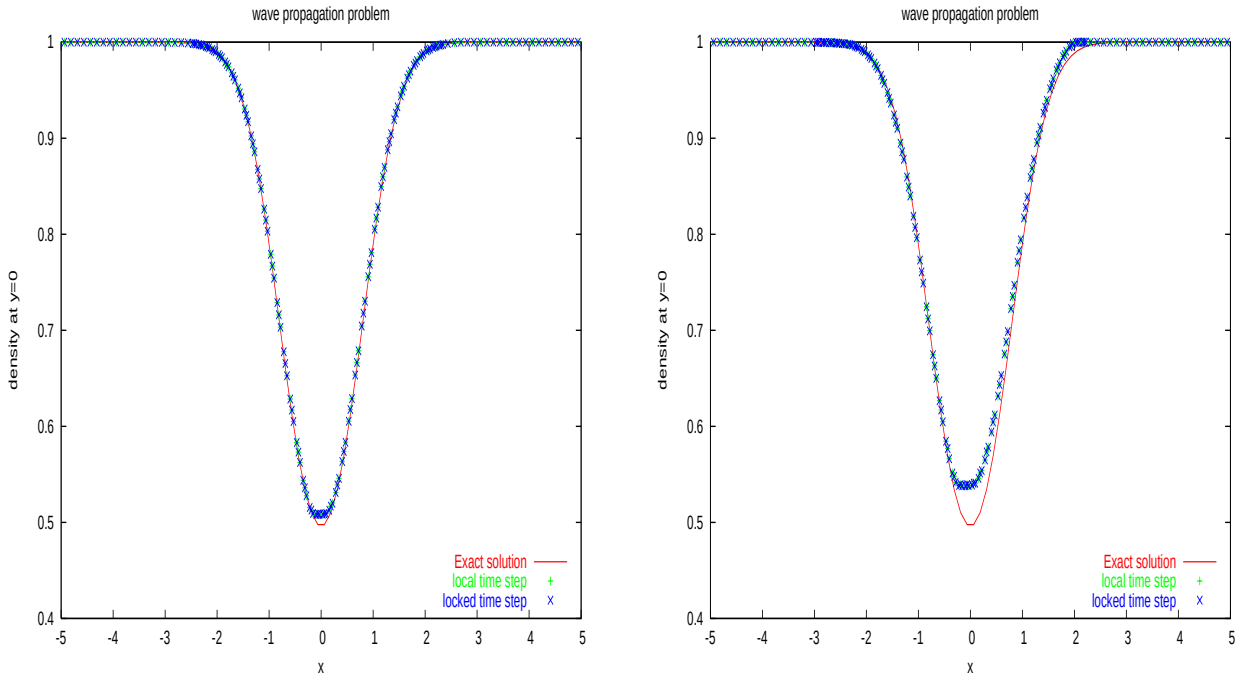

Figure 5: The density plot at $y = 0$ for wave propagation problem at $t = 10$ (left) and $t = 100$(right). AMR-MHD is used.

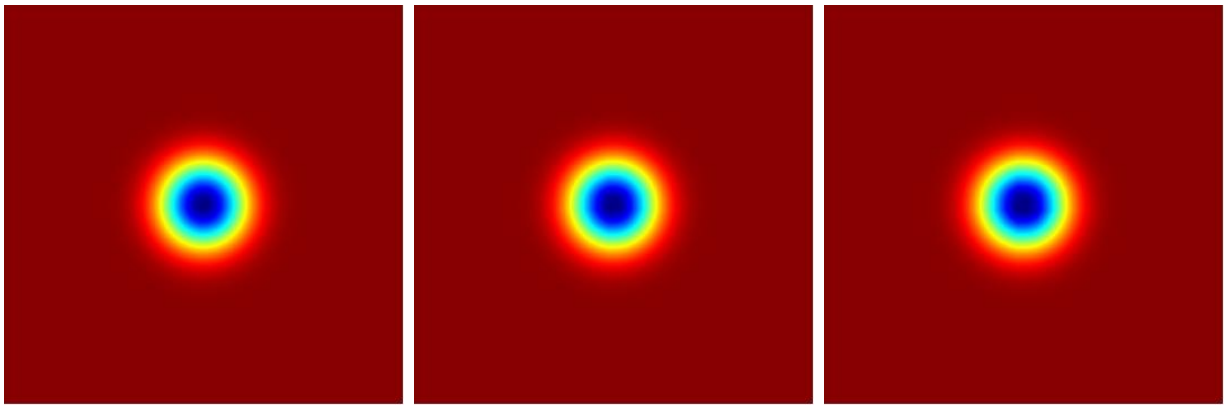

Figure 6: Density plot at t =10(left),50(middle), and 100(right) for 800×800 uniform grid. Linear wave problem was solved with RAGE package.

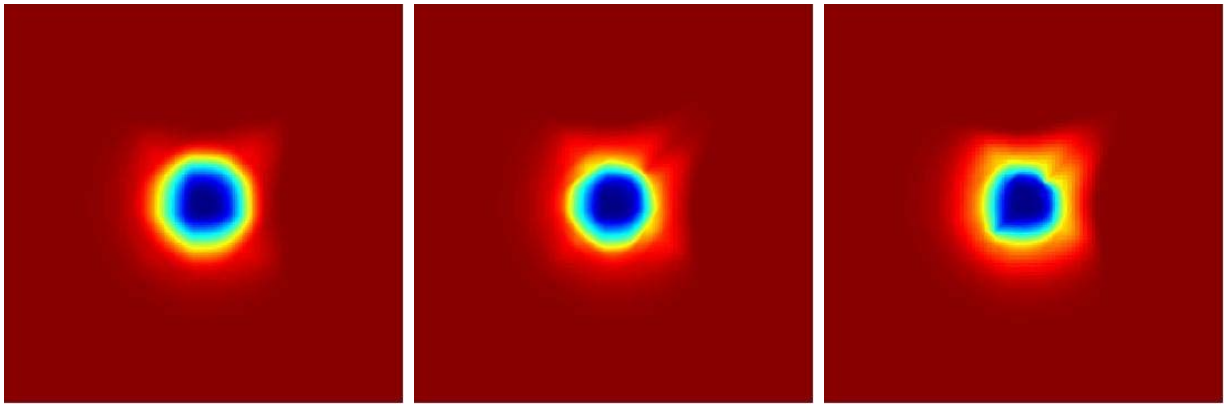

Figure 7: Density plot of 100×100 with AMR of different refinement levels at $t = 100$ for linear wave problem. Left: 1-level refinement, middle: 2-level refinement, right: 3-level refinement. RAGE is used.

Figure 13 shows the results of different base grids with only one refinement level, compared with the uniform grid with the same resolution. In term of L_∞ error, only the 100×100 base grid with AMR achieves almost the same accuracy as the same resolution uniform grid. In terms of L_1 and L_2 errors, the AMR results are even worse than the uniform results without refinement at early times. Figure 14 shows the errors at early times.

To understand length scale of the AMR grid, we plot the average spacing for different refinements and different base grid in Figure 15. It shows that the 100×100 grid with three-level refinement sometimes has larger average spacing than with two-level refinement. This should not have happened if the AMR performs properly, since we assume in AMR the more refinement, the more points we should have. This partially explains why the three-level refinement is worse than the results of two-level refinement in Figures 12 and 13.

To study the error-cost performance of AMR, we plot the error convergence behavior at different times using the average spacing. Note that the computational cost is proportional to the number of the cells, which determines the average spacing. Figure 16 shows the performance of AMR calculation versus that of the uniform grid calculation. The point above the uniform grid convergence line means that AMR has more cells but larger error than the uniform grid. Considering that AMR activities and data management add at least 15% overhead to the total computation, we summarize on the performance of the AMR calculation in Table 1.

4 Vortex evolution

The next example has almost the same initial and boundary conditions as described in previous section. The mean flow is $\rho = 1$, $p = 1$, and $(u, v) = (1, 1)$. We add, to the mean flow, an isentropic vortex (perturbations in (u, v) and the temperature $T = p/\rho$, no perturbation in the entropy $S = p/\rho^\gamma$):

$$(\delta u, \delta v) = \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(-y, x),$$

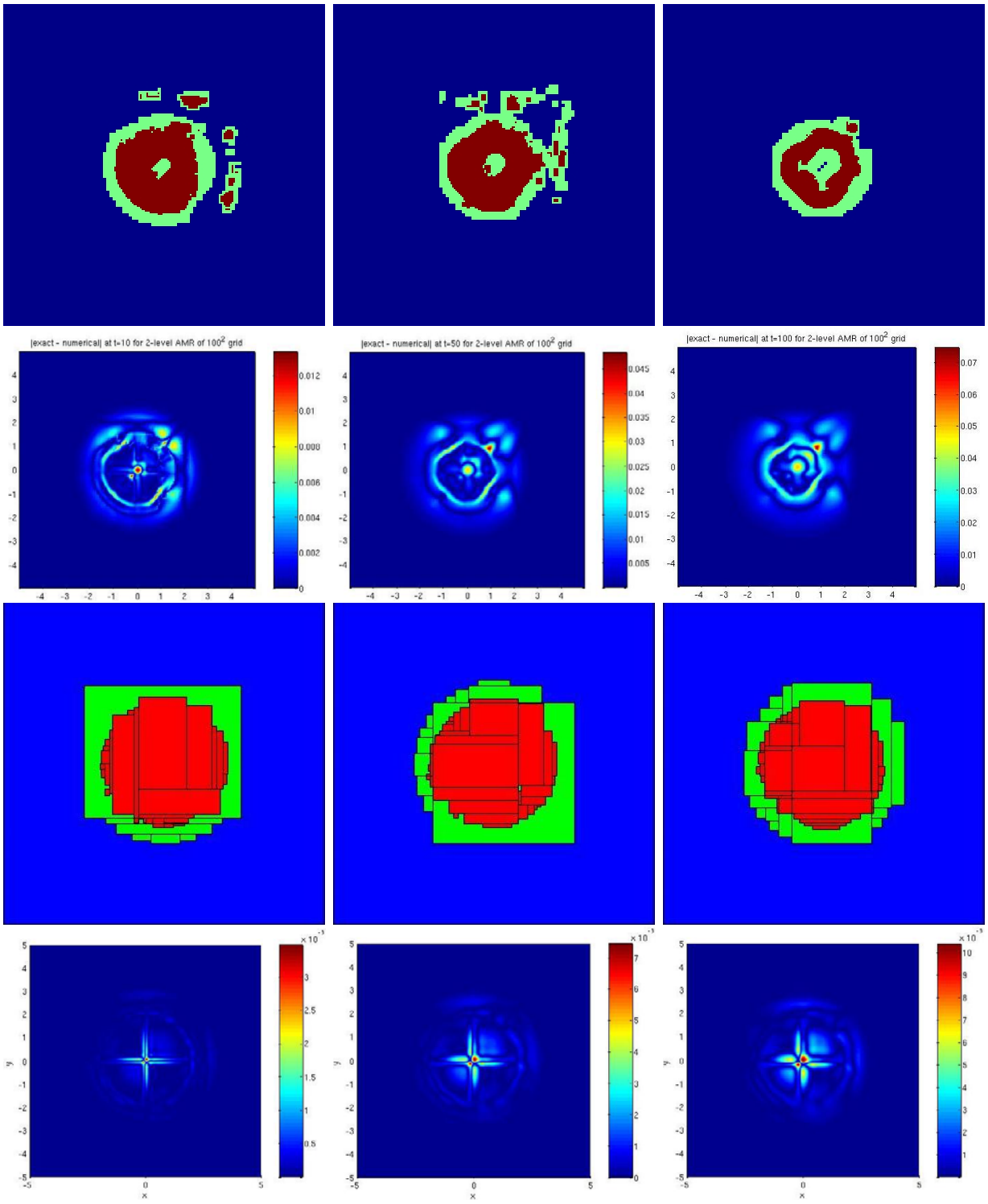

Figure 8: Refinement and error distribution with 2-level refinement for linear wave problem. Left: t=10; middle: t=50,, right: t=100. RAGE is used in the top two lines with base grid 100x100. AMR-MHD is used in the bottom two lines with base grid 100x100.

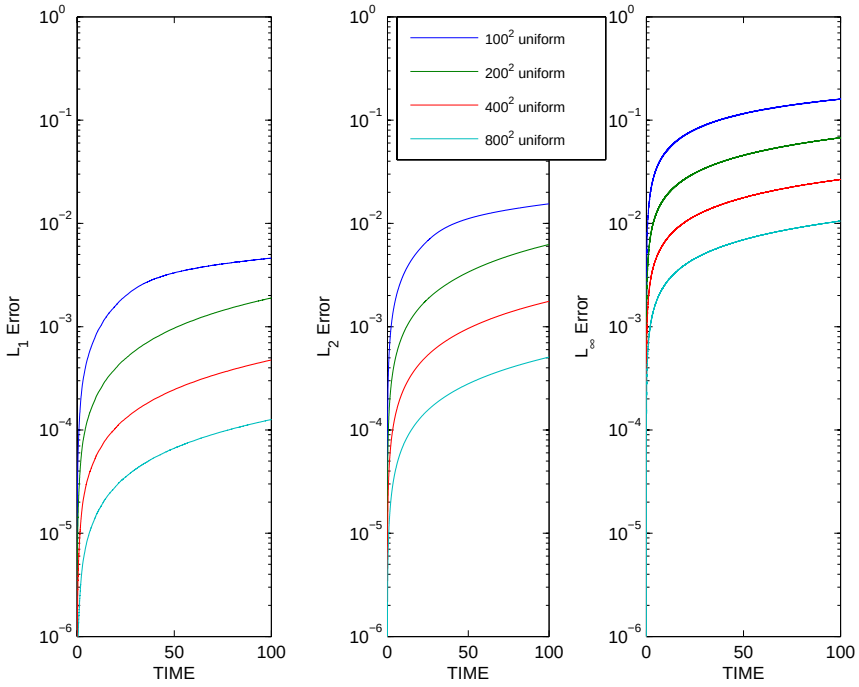

Figure 9: Numerical errors of different grids for linear wave problem. RAGE is used.

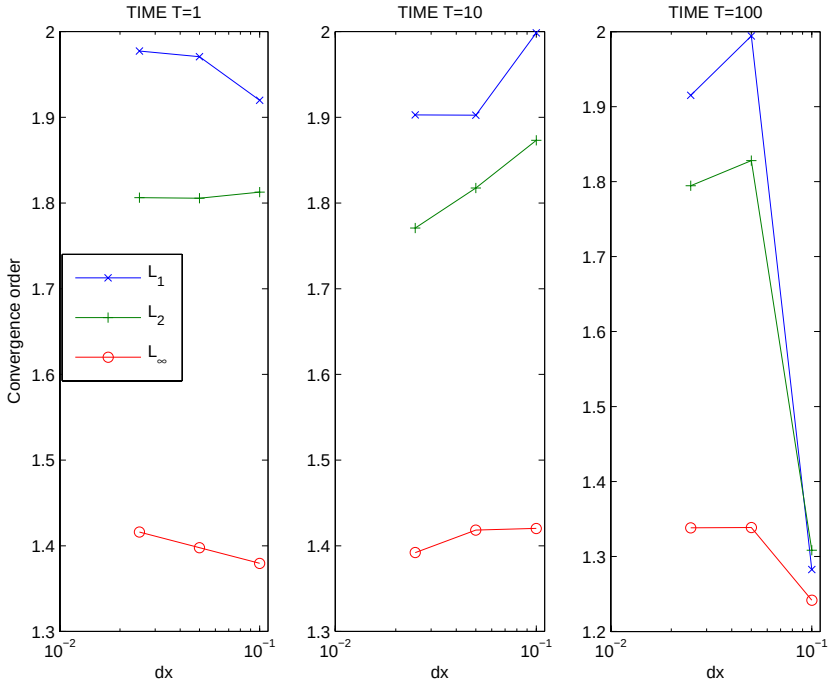

Figure 10: Convergence order at different times for linear wave problem. RAGE is used.

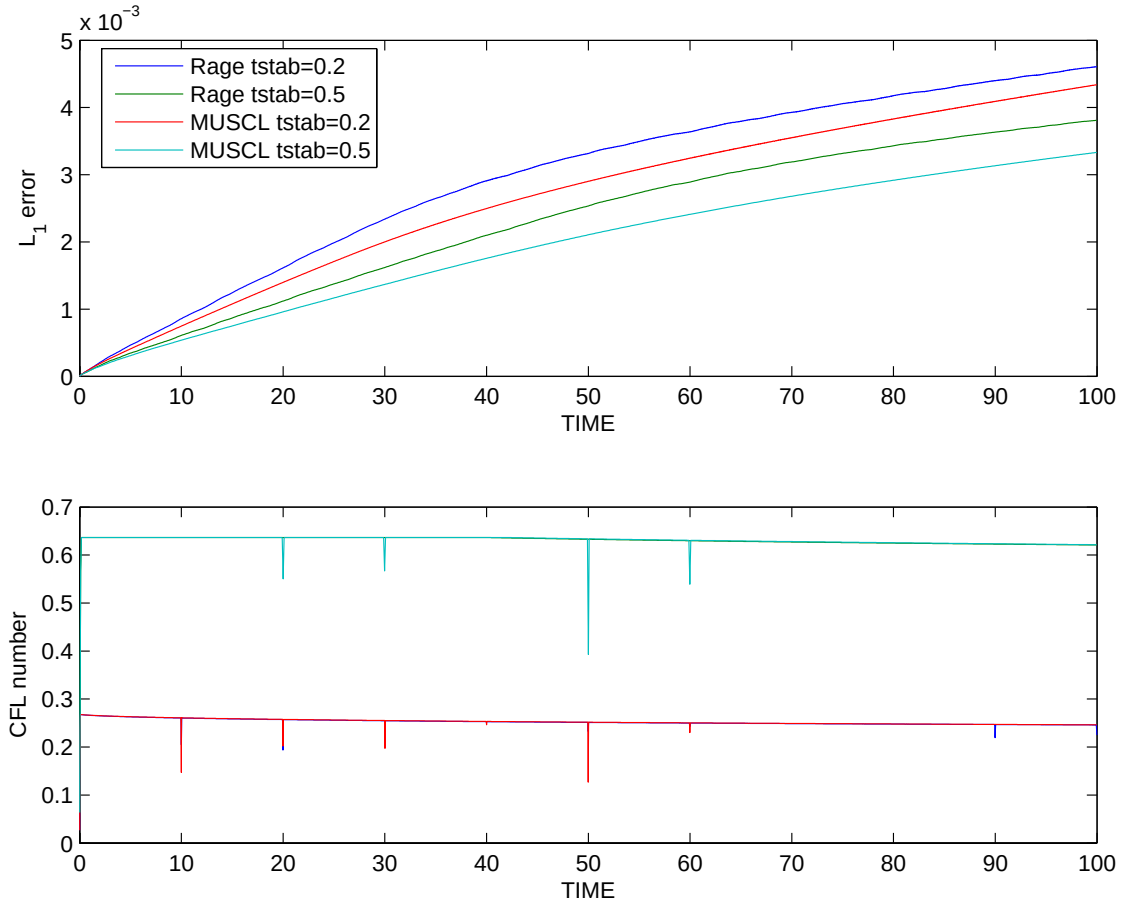

Figure 11: Numerical errors for different CFL numbers and different algorithms for linear wave problem. 100×100 grid is used. The bottom plot has the same legend as the top one. RAGE is used.

Table 1: Error-cost performance of AMR computation for linear wave problem, where $\surd$ means it is better than uniform grid and $\times$ means it is worse than uniform grid. RAGE is used.

	100^2AMR(1)	100^2AMR(2)	100^2AMR(3)	200^2AMR(1)	200^2AMR(2)	400^2AMR(1)
L_1 t=1	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_2 t=1	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_∞ t=1	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
L_1 t=10	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_2 t=10	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$
L_∞ t=10	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
L_1 t=100	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$
L_2 t=100	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\times$
L_∞ t=100	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\times$

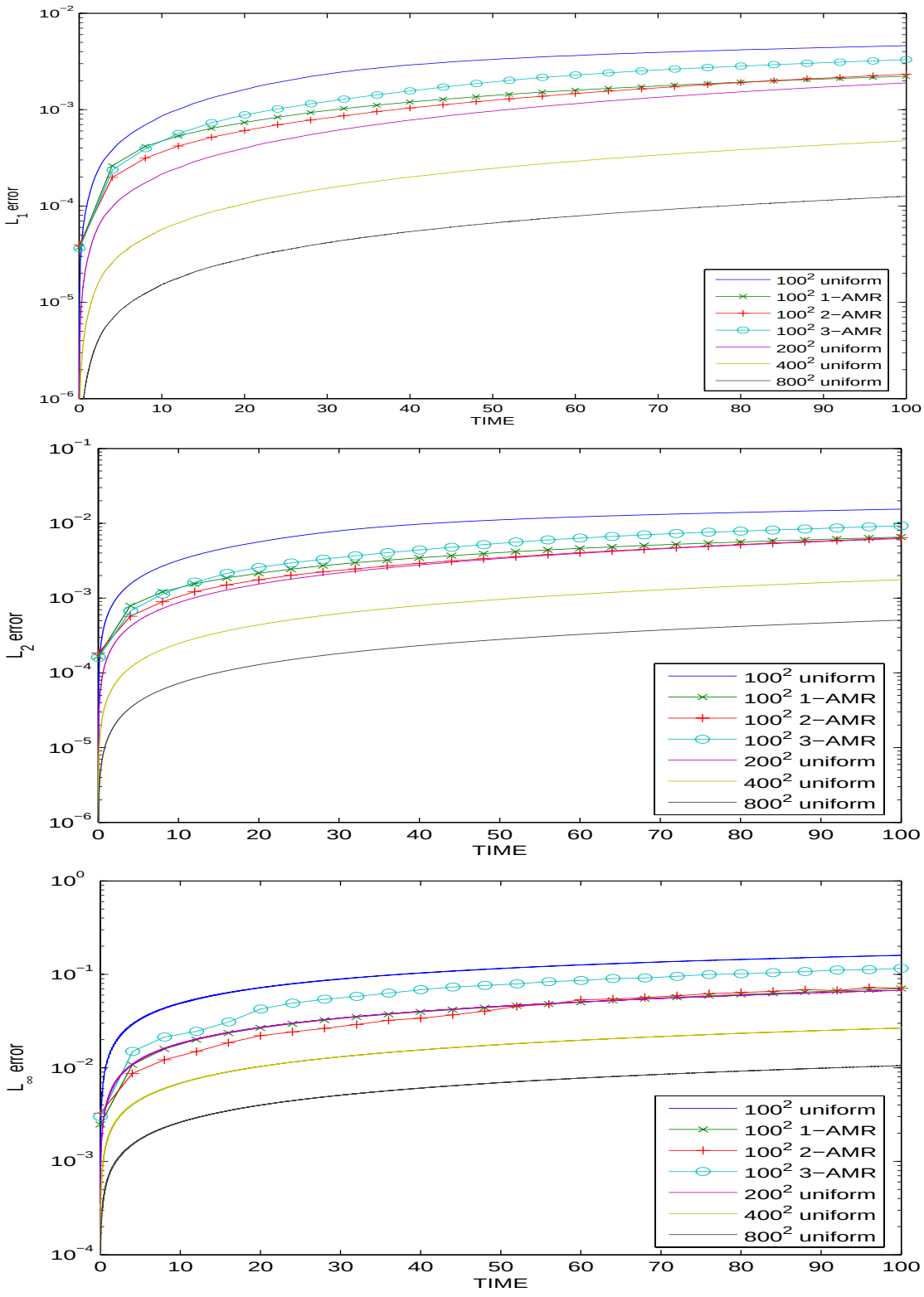

Figure 12: Numerical errors of 100×100 grid with different refinement levels for linear wave problem. RAGE is used.

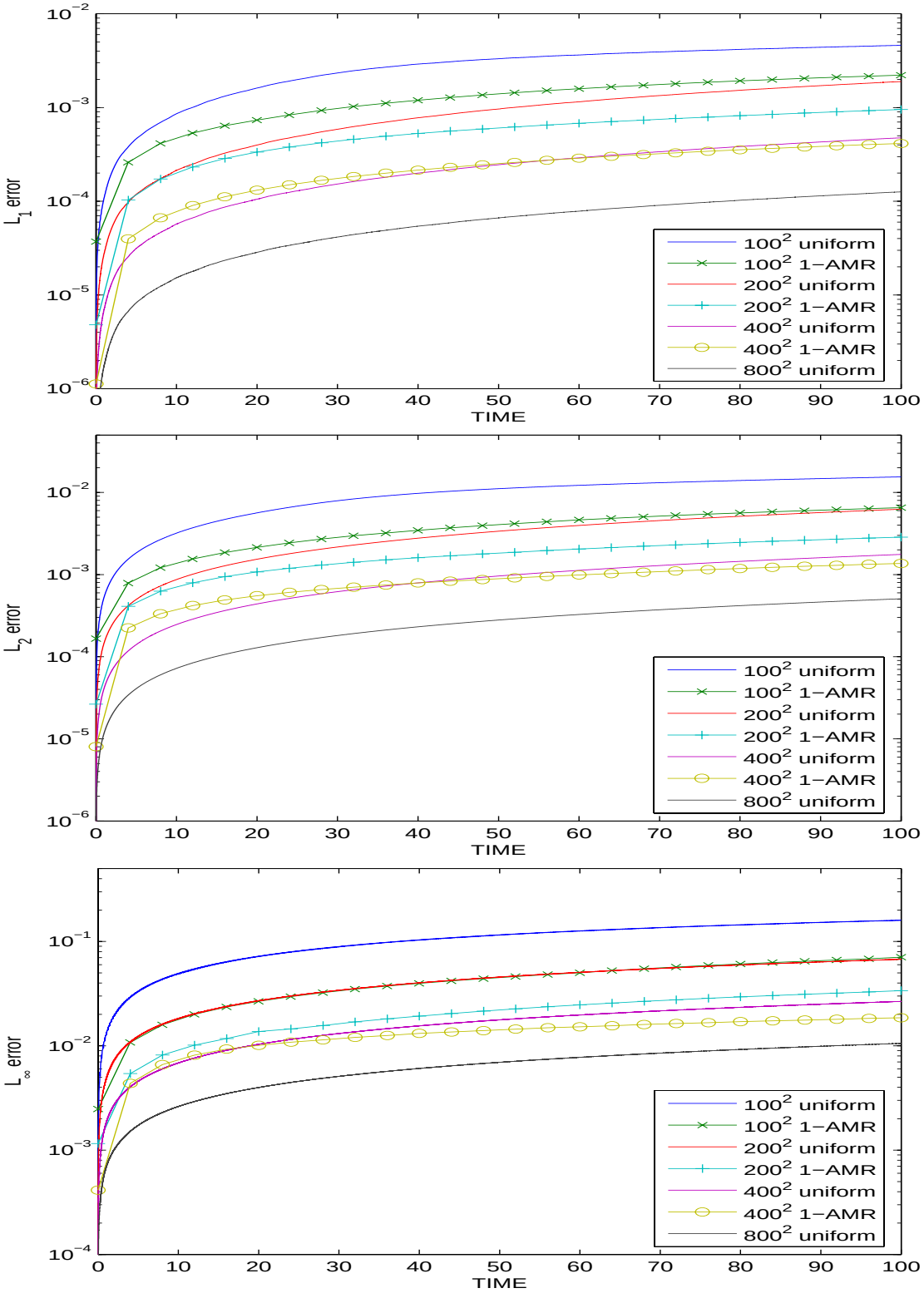

Figure 13: Numerical errors of different base grids with only one refinement level for linear wave problem.

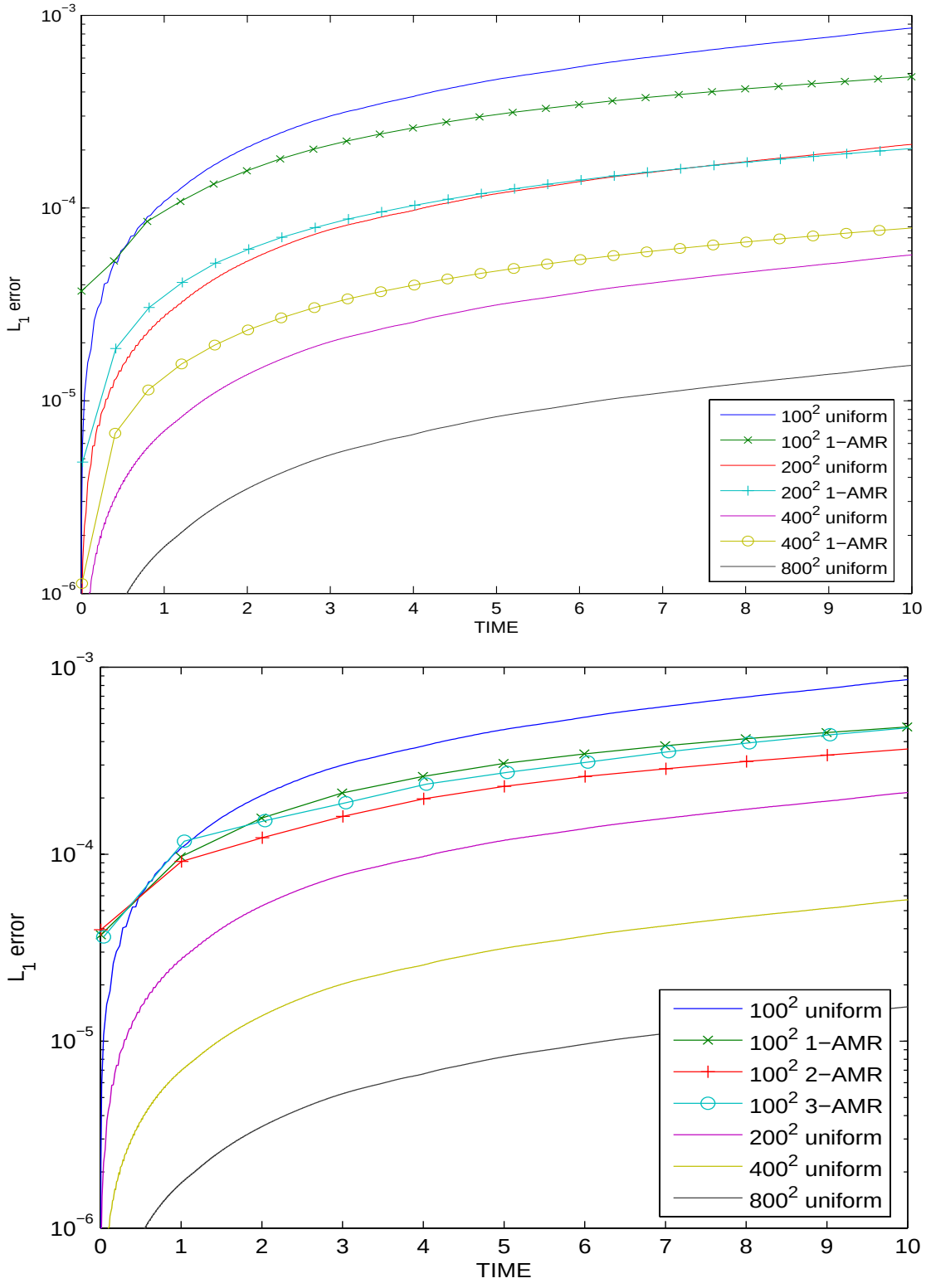

Figure 14: Numerical L_1 error of AMR grids at early time, compared with the same resolution uniform grid, for Linear wave problem. RAGE is used.

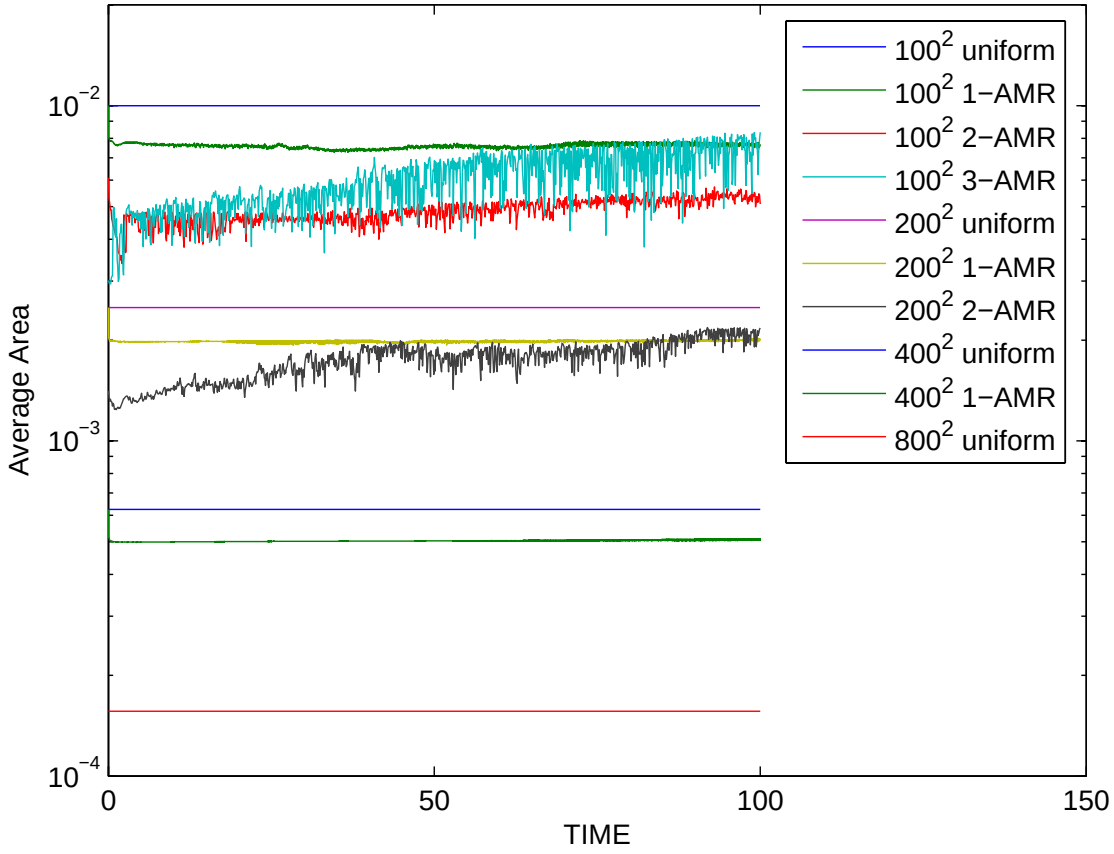

Figure 15: Average cell-volume of different grids with refinement for linear wave problem. RAGE is used.

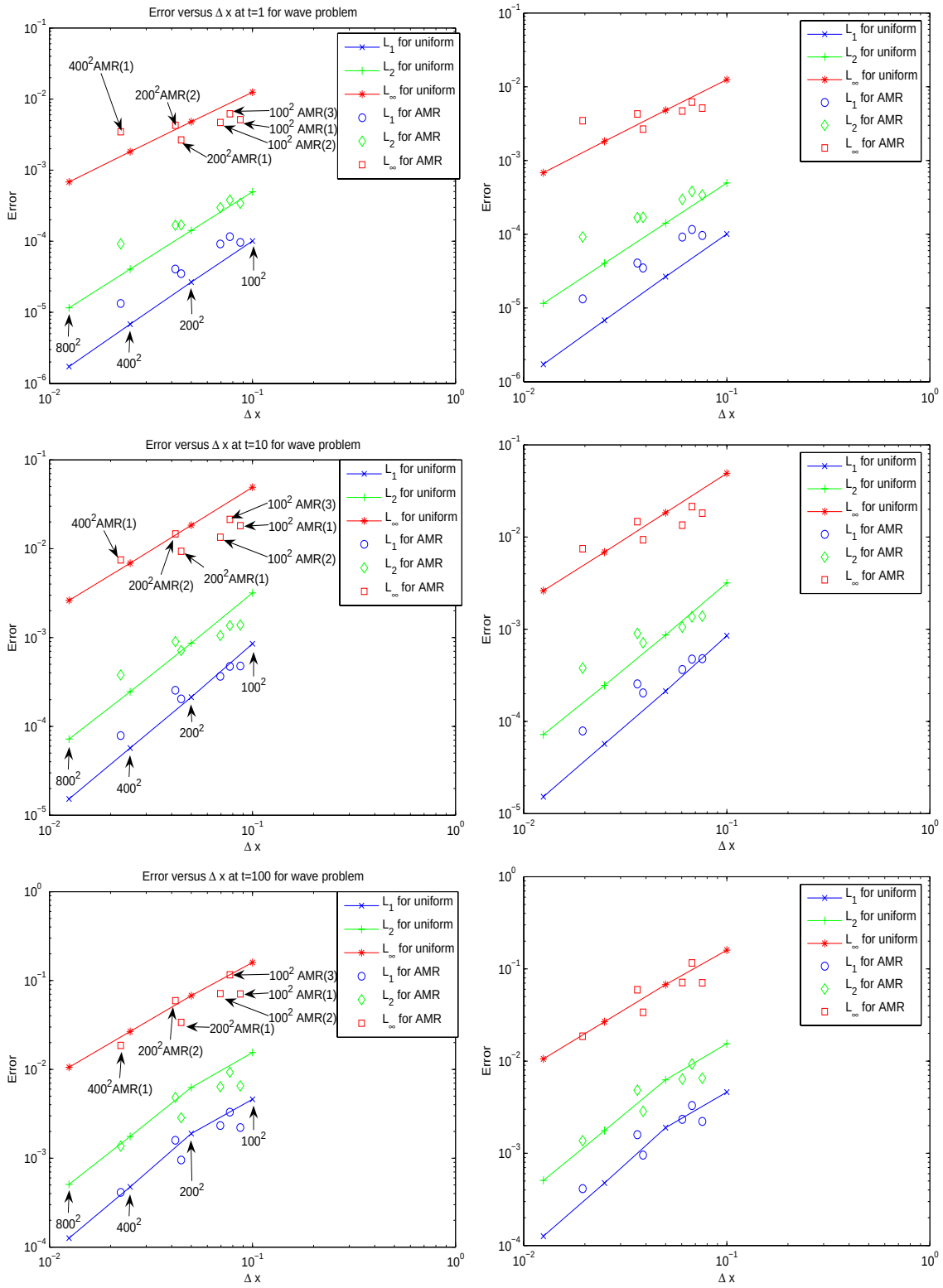

Figure 16: The error convergence behavior of AMR grid, compared with those of uniform grid. The right plot has considered 15% overhead of AMR while the left has not. The point above the uniform grid convergence line indicates that AMR has more cells but larger error than the uniform grid. RAGE is used.

$$\begin{aligned} \delta T &= -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2 e^{1-r^2}}, \\ \delta S &= 0, \end{aligned}$$

where $r^2 = x^2 + y^2$, and the vortex strength $\epsilon = 5$. The computational domain is taken as $[-5, 5] \times [-5, 5]$, extended periodically in both directions.

It can be verified easily that the initial density profile is the same the first example, and the exact solution of the Euler equations with the above initial and boundary conditions is just the passive convection of the vortex with the mean velocity (1,1). Therefore, we expect the density behaves exactly as the first examples.

The solution is very smooth. No shocks are considered at this stage. We use the same four different uniform grids as the previous example to solve the problem. The problem set-up and tests, except the initial conditions, are the same as the previous example.

4.1 Results of AMR-MHD

Fig. 17 shows the convergence order at different times for different error indicators and different CFL numbers. It is clear that the asymptotic range is achieved from the resolution 80x80 grid. Fig. 18 shows the numerical errors for the four uniform grids. Fig. 19 shows the asymptotic convergence for at specific times. Three times ($t = 10, 40, 80$) are chosen. The numerical results show that the CFL number has dramatical effect on the accuracy of the solutions. The smaller the CFL number, the less accurate the solutions are.

Figs.20 to 22 shows the numerical errors of the AMR grid with local time step and with locked time step, compared with the results of the uniform grid. From Fig. 20, we can see that the AMR with locked time step may yield solutions that are worse than those of no refinement at all. These figures also show that the error accumulating rate with time seems determined by the CFL number in the coarse-level grid after a short period of time. This may be a bad news for many AMR users that the solution with local mesh refinement may be worse than that of without AMR after a period of time.

Fig. 23 and 24 show the numerical convergence behavior for the AMR results compared with those obtained by the uniform grid. We also study the characteristic length scale for the AMR grid. At time $t = 10$, Fig. 23 shows that the results for the AMR with minimum spacing is closer to the corresponding uniform grid results. The deviation at fine resolution (at 320×320) is because the refinement region is become smaller and smaller as the grid becomes finer and finer. However, at time $t = 40$ and $t = 80$, the results for the AMR with average (or mean) spacing is closer to the corresponding uniform grid results. Fig. 24 shows the results for the locked time step method. There is no improvement changing from the $40x40$ to $80x80$ grid.

We also attempted to calculate the convergence order based on two AMR grid with the same refinement level but different base grid. Fig. 26 shows the results.

In Fig. 27, we study the characteristic length-scale for the AMR method at $t = 10$. We assume the error model for the AMR grid is the same as that of the uniform grid. If we know the coefficient A and rate p from the uniform convergence analysis, we can define the best characteristic length scale for an AMR grid by inverse interpolation:

$$\Delta\tilde{x} = \exp((\log ||\xi - \xi^*|| - A)/p).$$

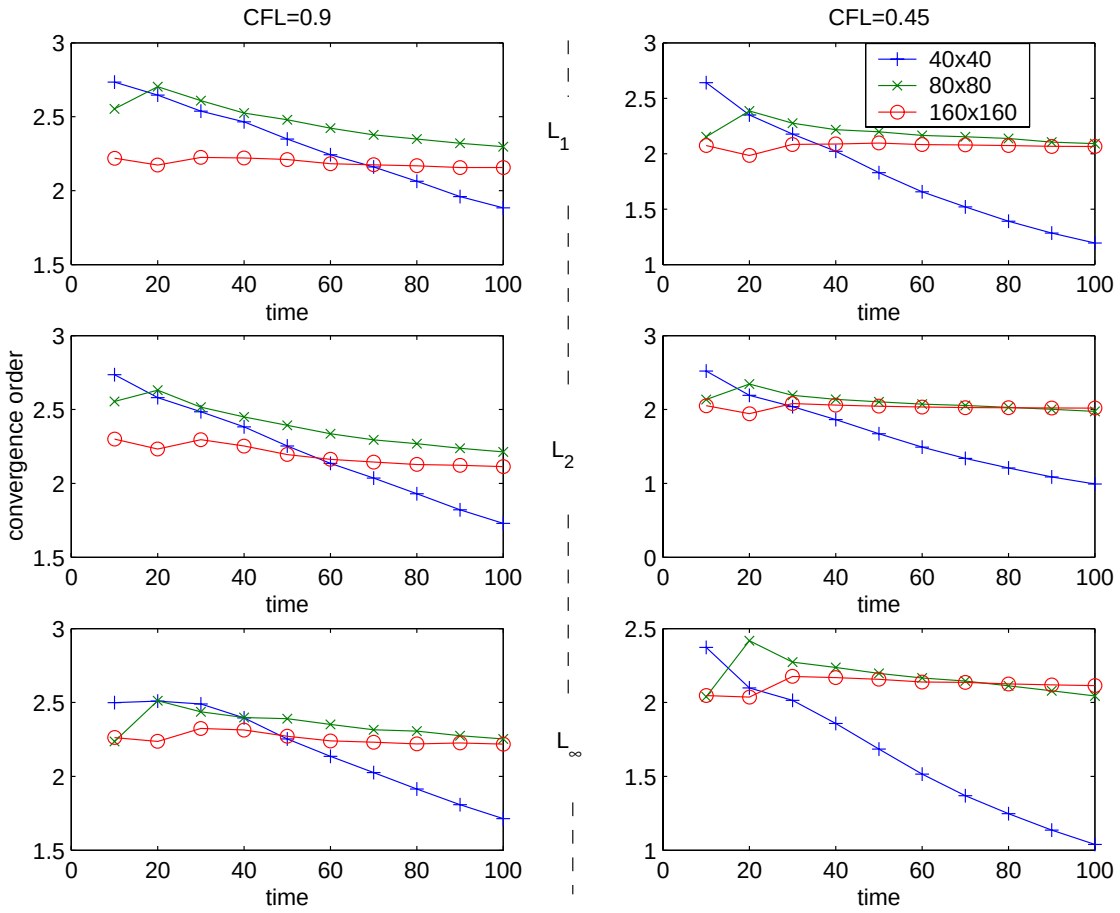

Figure 17: Convergence order at different times for different error indicators and different CFL numbers

In Fig. 27, we derived that the corresponding length-scale for the AMR grid at $t = 10$ is 0.069 if the L_1 error is considered, whereas the average space is 0.0972.

Compared the results of the first example (linear wave propagation) and the second one, we can see how the nonlinearity in Euler equations impacts the numerical error and convergence order. For these two examples, the density profile is exactly the same and is advected with the same velocity. However, the CFL number has more impact on the numerical errors on the second one than on the first one. The reason why we are so focus on the CFL number is that the AMR with locked time step method has different CFL numbers in different refinement levels. We had thought that the small CFL number in the coarse level would not have much impact on the accuracy overall because the large error region has been covered with the fine level that has a large CFL number. This is no longer true for the second examples.

Compared the numerical results in Figs. 5 and 25, we can see that the later one has large dispersion error at $t = 100$, even if the amplitude error is smaller than the first example. The large phase error may be due to the small CFL number used in the coarse level.

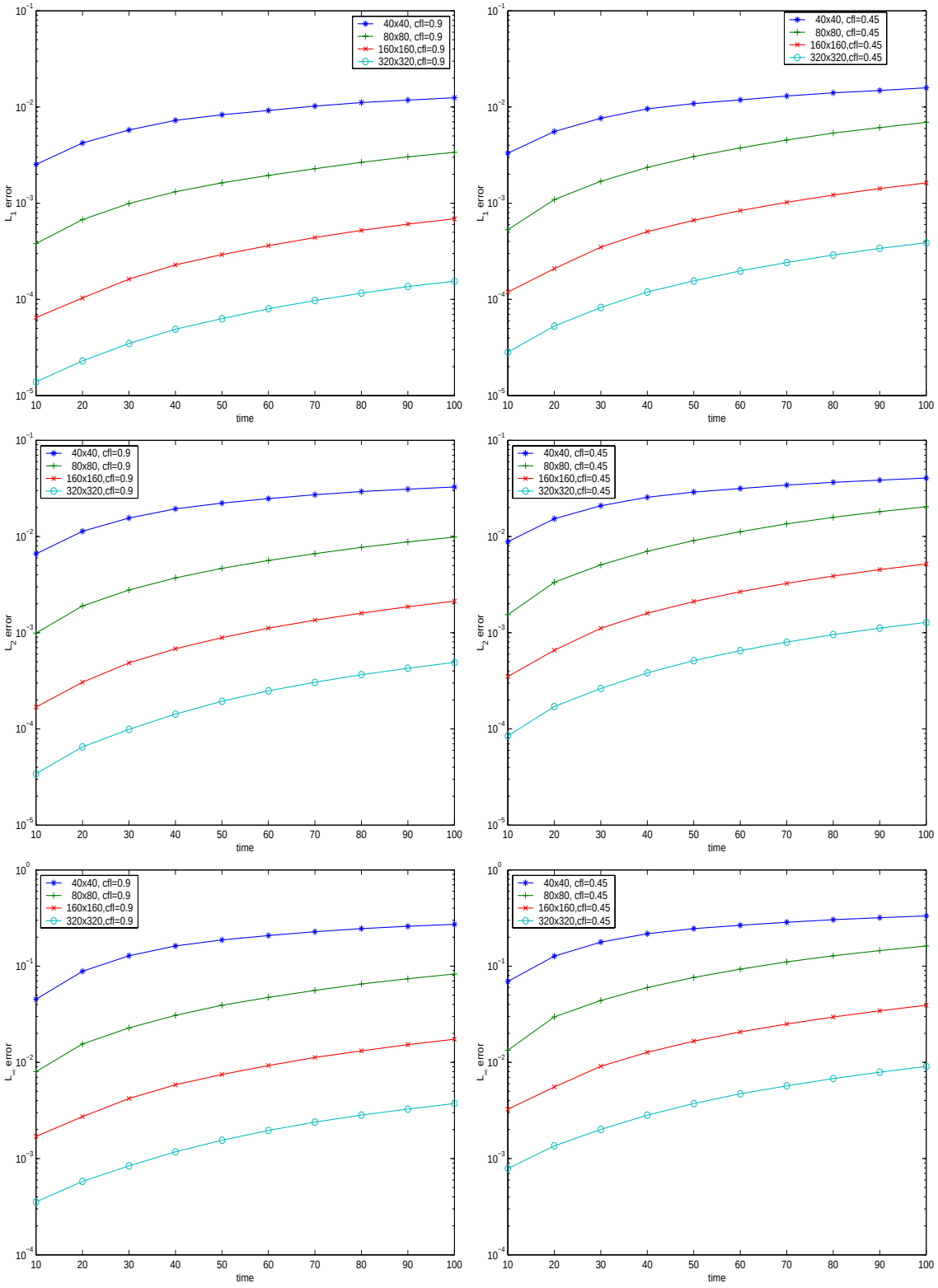

Figure 18: Numerical errors (L_1, L_2, and L_∞) for the vortex evolution example at different times for different uniform grids. AMR-MHD is used.

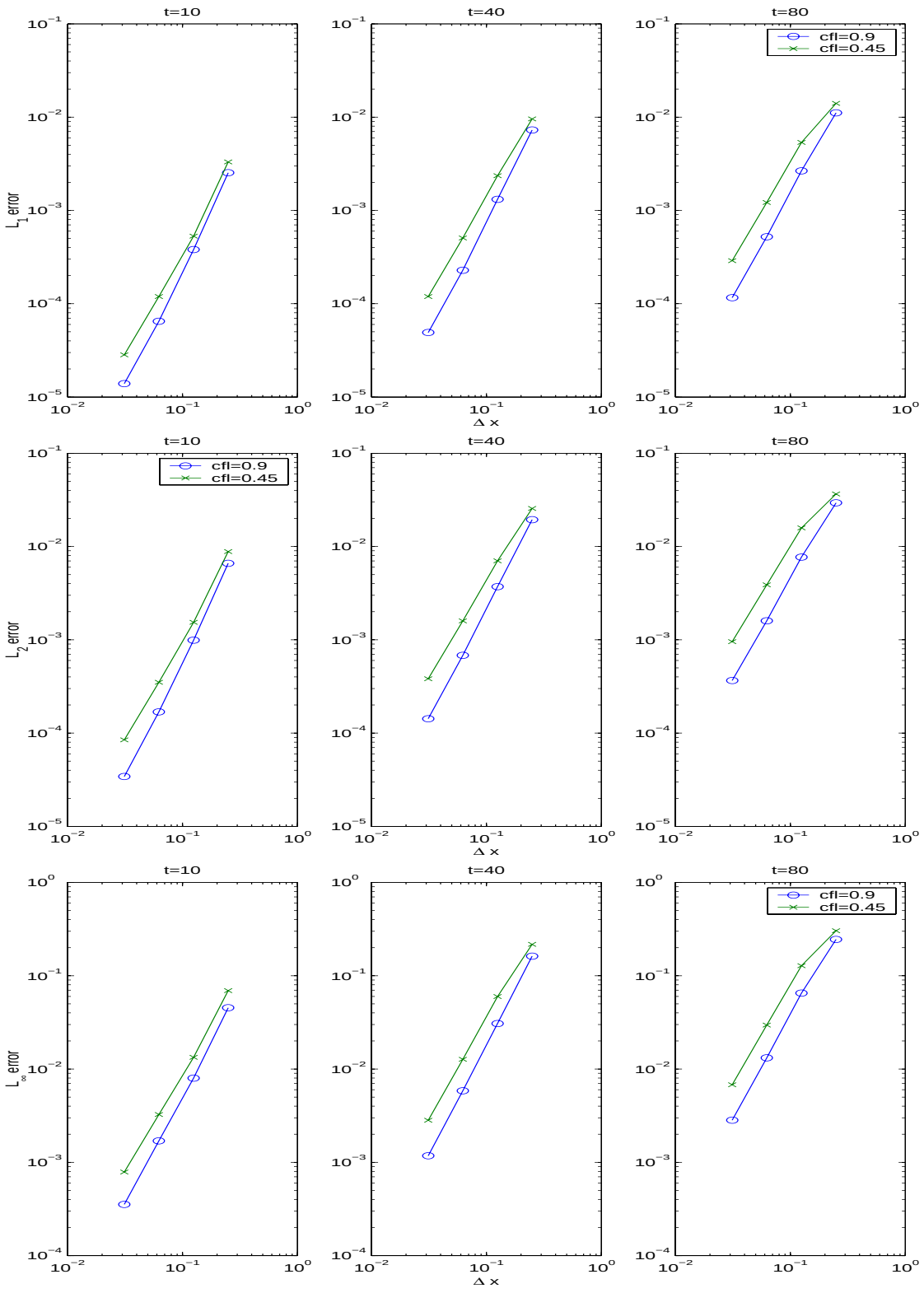

Figure 19: Numerical errors (L_1, L_2, and L_∞) for the vortex evolution example at different times for different CFL numbers. AMR-MHD is used.

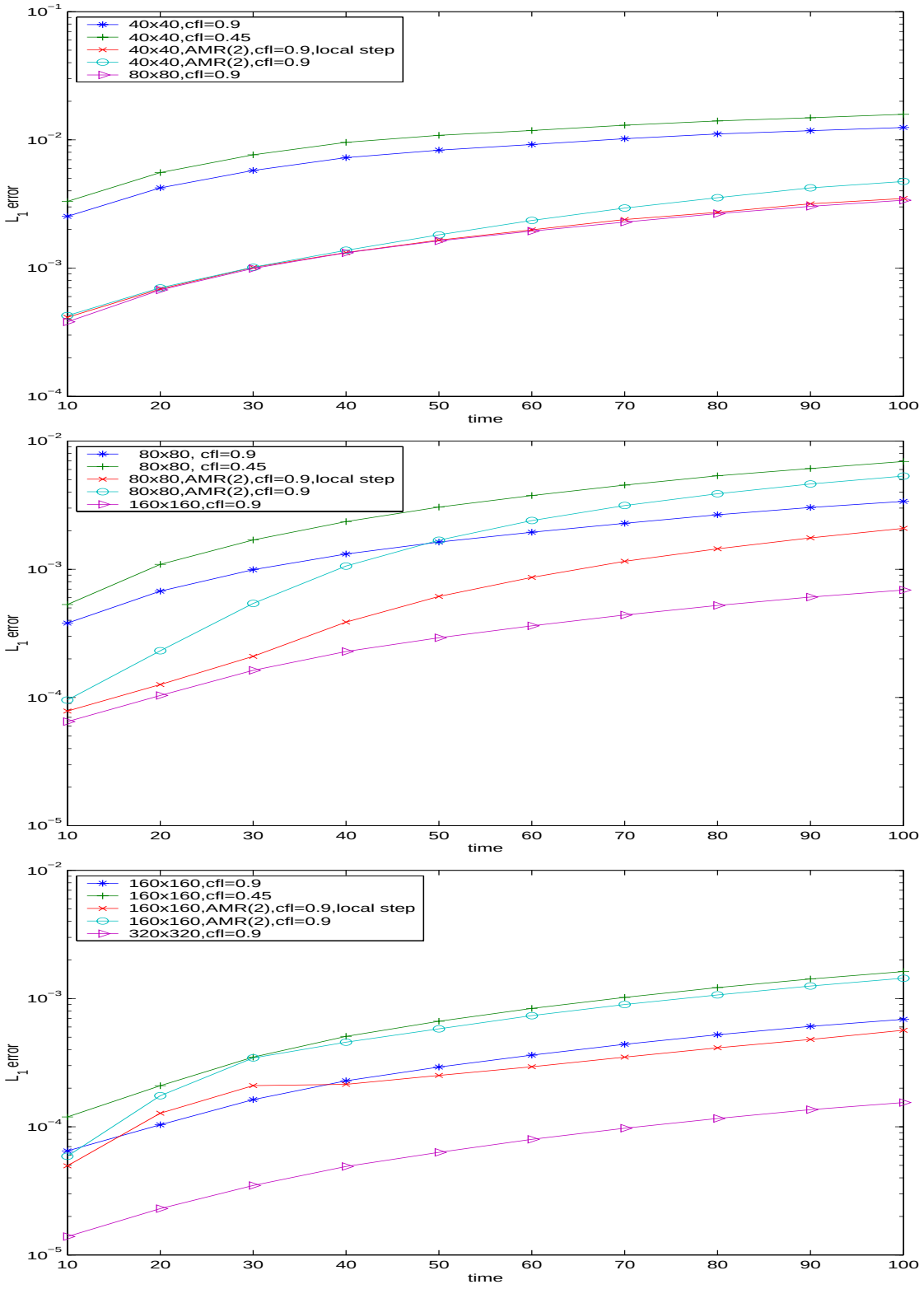

Figure 20: Numerical errors (L_1) for the vortex evolution example at different times for different CFL numbers. AMR-MHD is used.

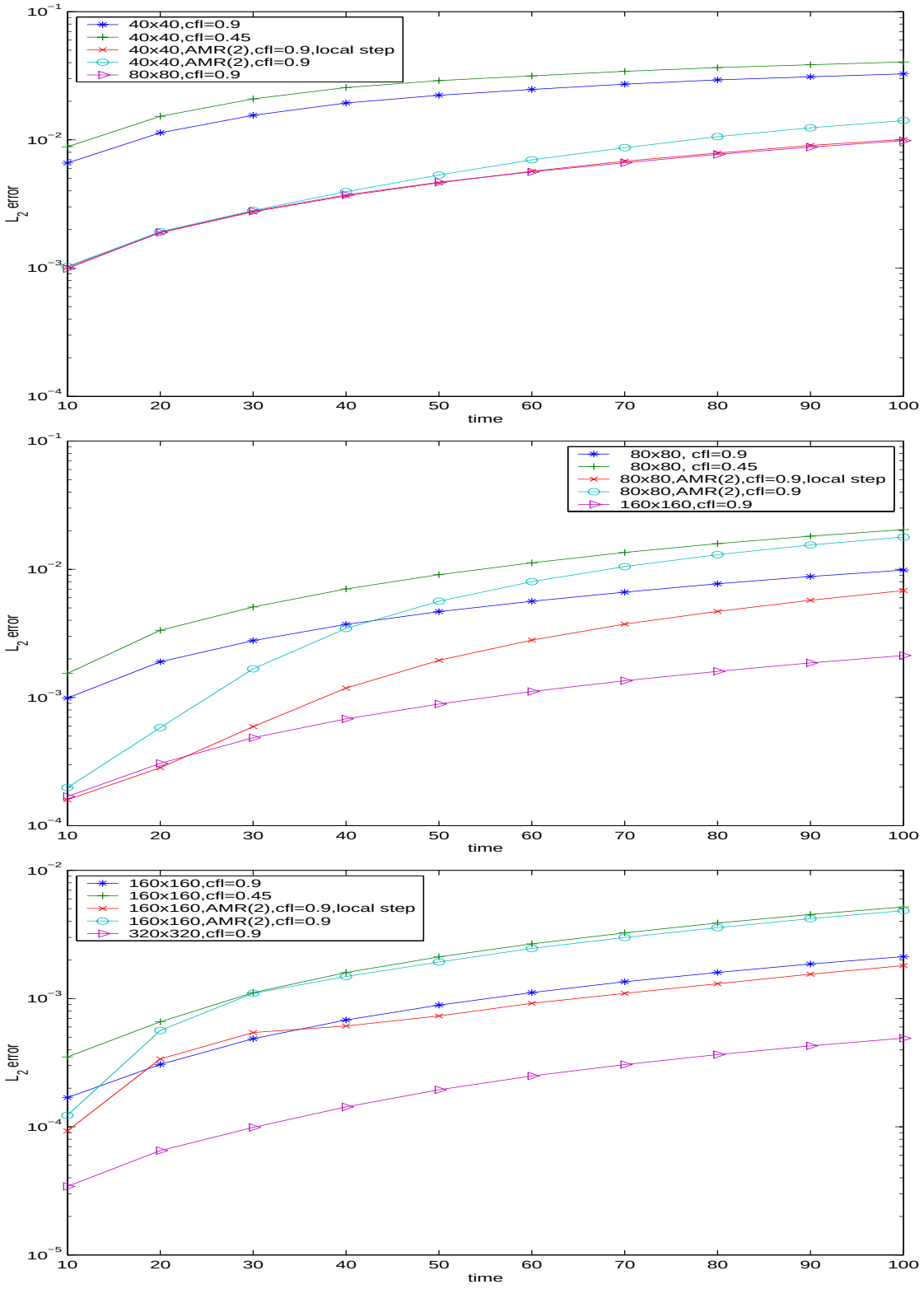

Figure 21: Numerical errors (L_2) for the vortex evolution example at different times for different CFL numbers. AMR-MHD is used.

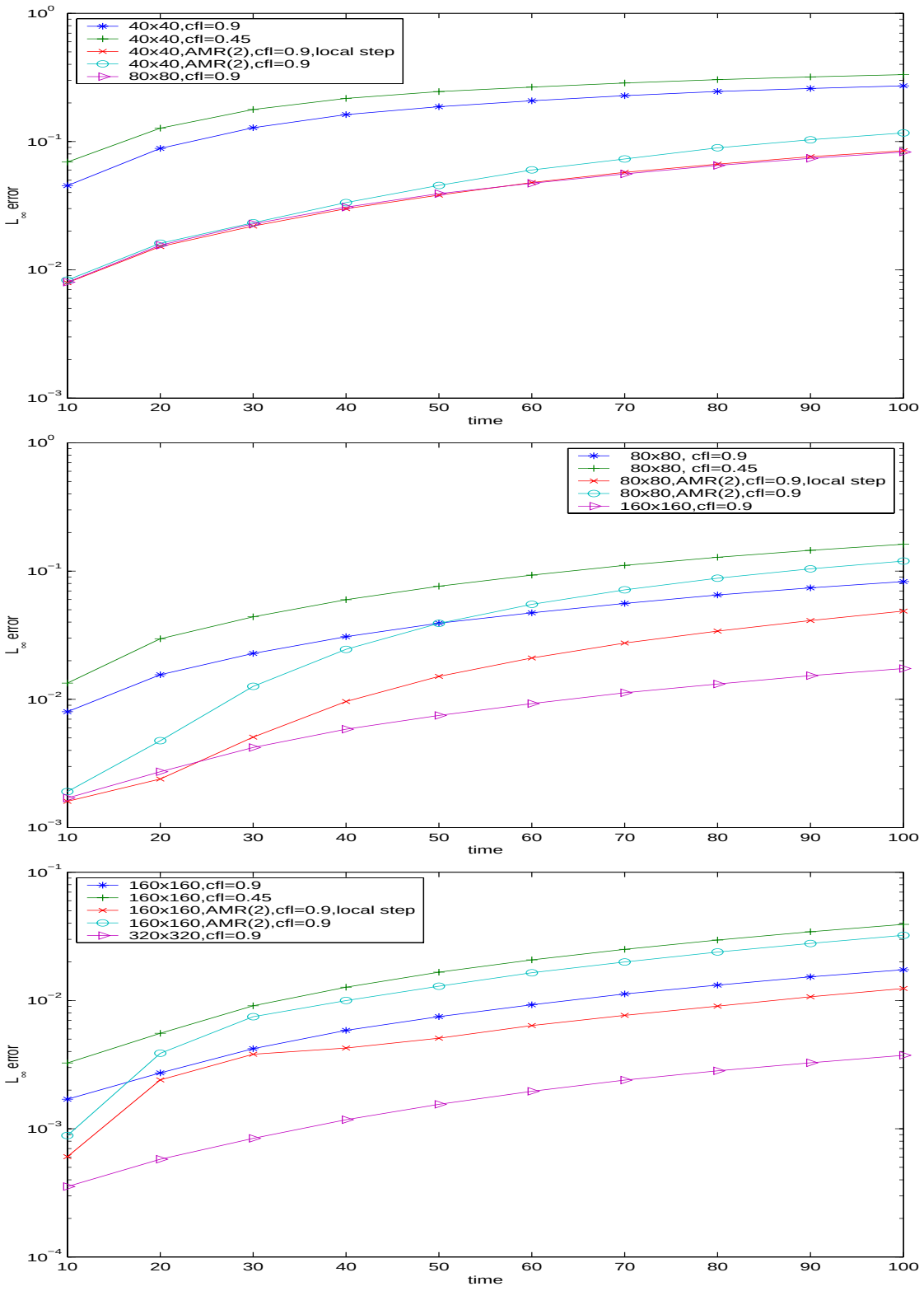

Figure 22: Numerical errors (L_∞) for the vortex evolution example at different times for different CFL numbers. AMR-MHD is used.

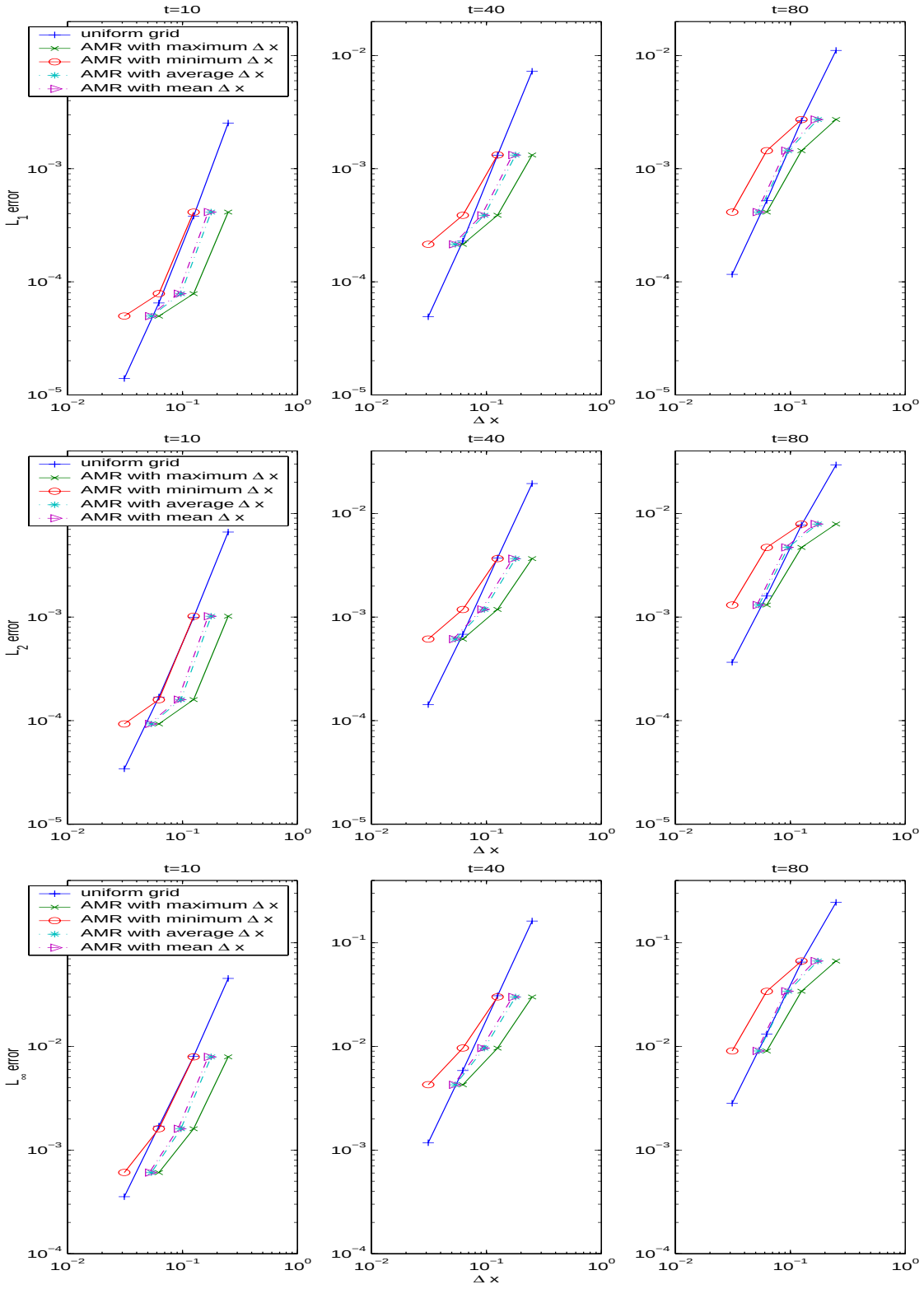

Figure 23: Numerical convergence behaviors for the vortex evolution problem for different error indicators (L_1, L_2, and L_∞) at different times. AMR with local time step method is used. AMR-MHD is used.

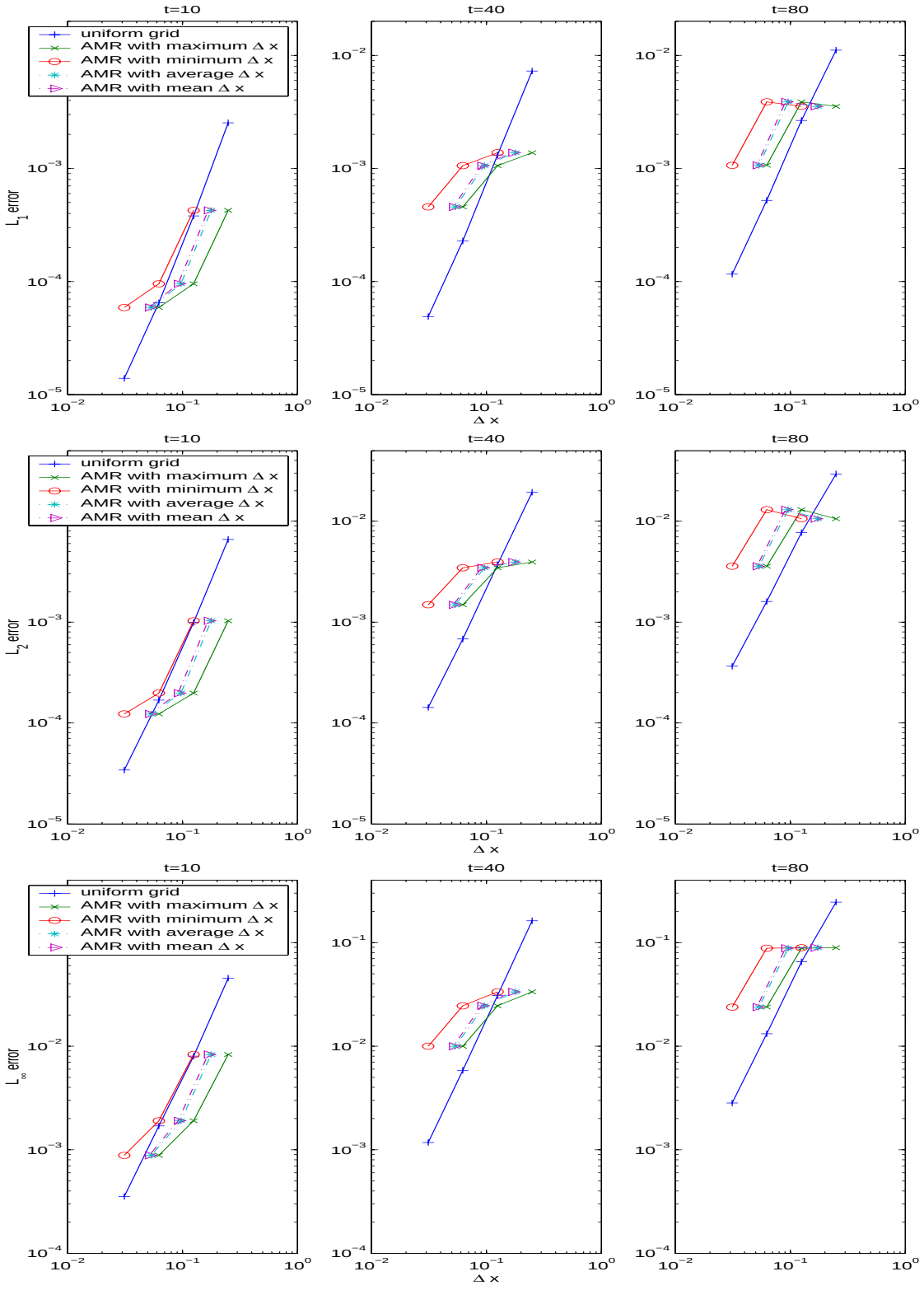

Figure 24: Numerical errors (L_1, L_2, and L_∞) for the vortex evolution problem at different times. AMR with locked time step method is used. AMR-MHD is used

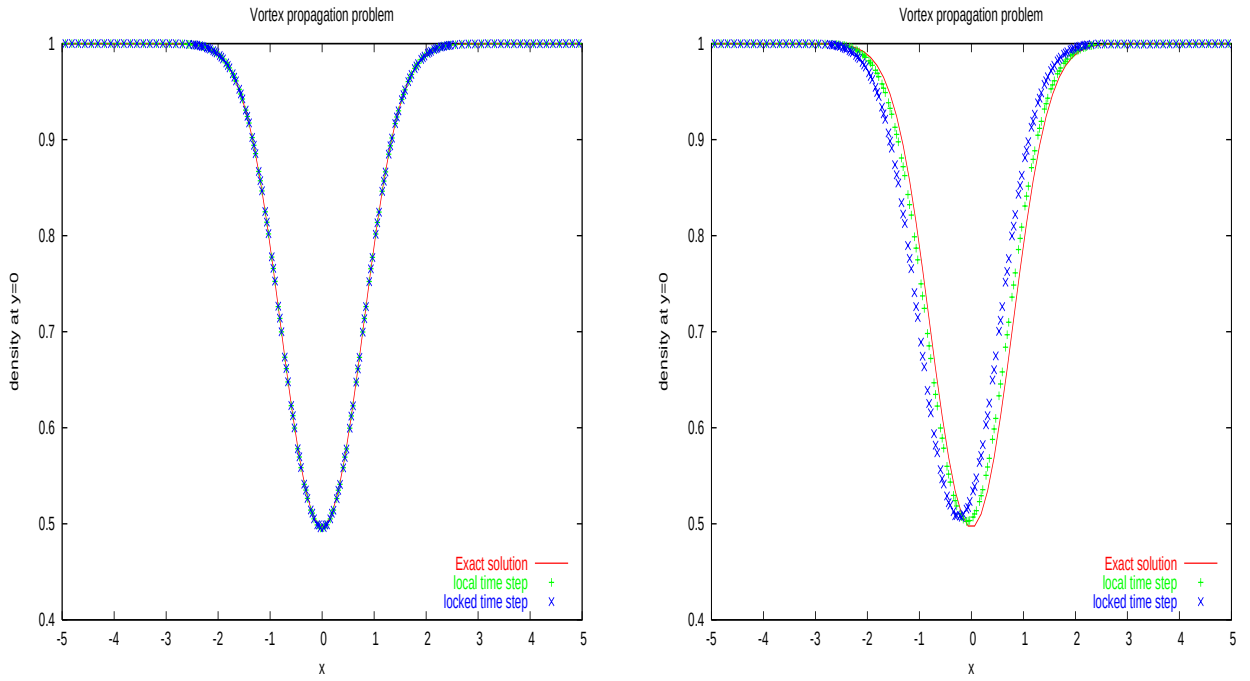

Figure 25: The density plot at $y = 0$ for vortex evolution problem at $t = 10$ (left) and $t = 100$(right). AMR-MHD is used.

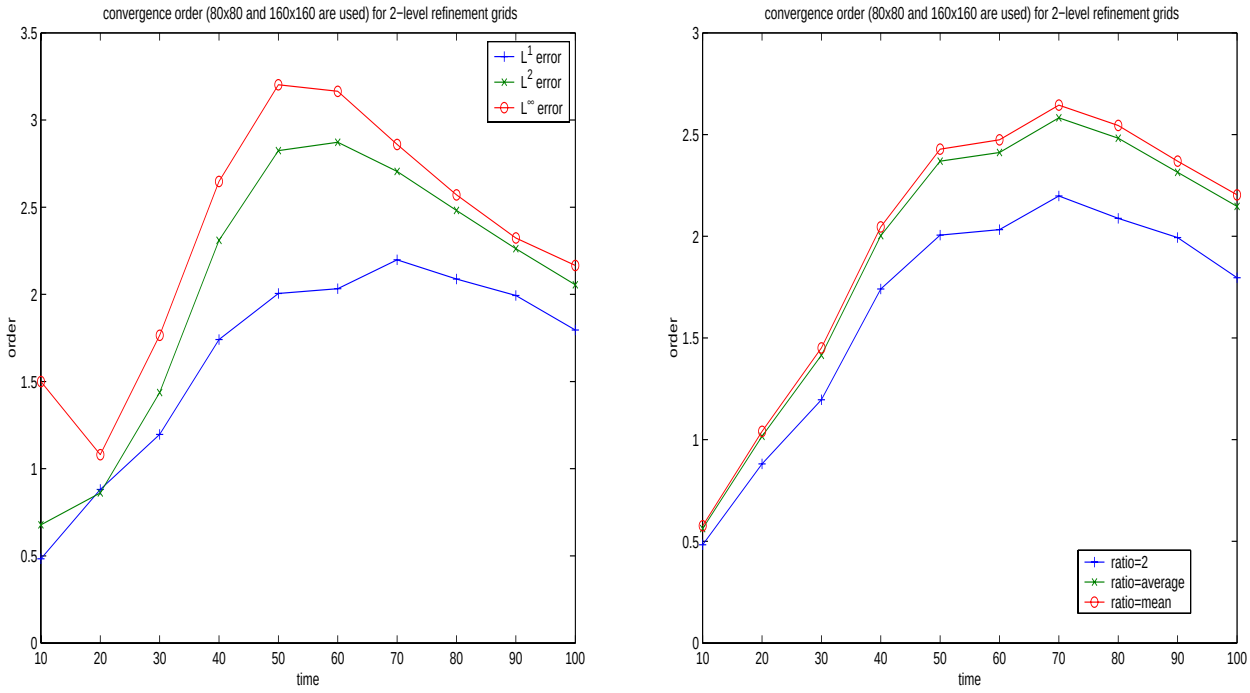

Figure 26: Convergence order for AMR grid with different resolutions at different times. AMR-MHD is used.

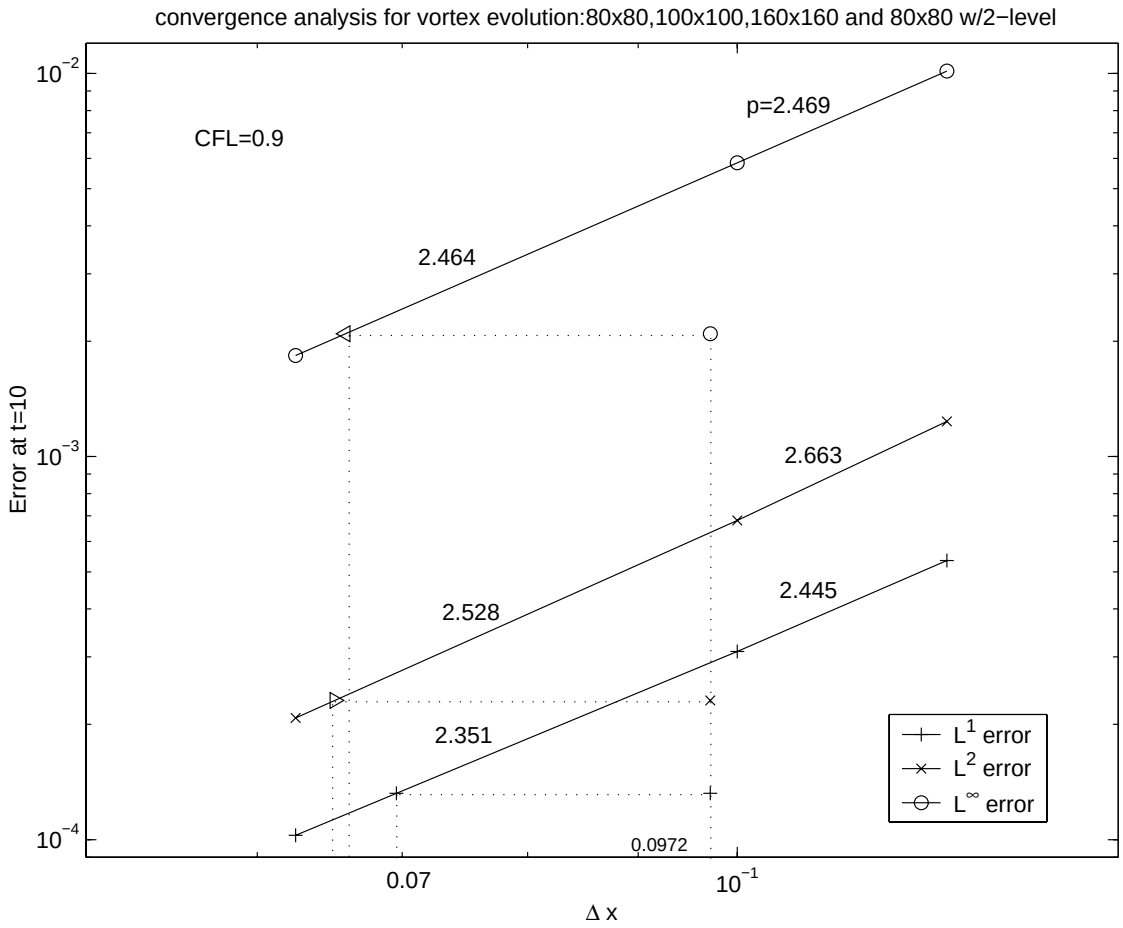

Figure 27: Length scale analysis for the AMR grid with 2-level refinement at $t = 10$. AMR-MHD is used.

4.2 Results of RAGE

Figure 28 shows the errors in different norms and different times. We can see the oscillations occur at later times for the fine grid. We think it is because the error in the vortex region has been propagated to other flat regions. We actually observed that the whole profile depicted in Figure 5 goes up and down at later times in the results of the fine grid. One interesting thing is that we did not observe such kind of feature in AMR-MHD results by the time $t = 100$.

We also calculated the convergence order at different times for different error norms. The results are shown in Figure 29. We observed some negative orders in the plot. The results at $t = 100$ was corrupted by the oscillations which may cause the negative order. The L_∞ error between 400×400 and 800×800 looks bizarre, which is shown in Figure 30. The error distribution for both grids at $t = 1$ is shown in Figure 31. Overall the convergence rate is not very good for high resolution uniform grid.

To study the effect of CFL number on the accuracy of the numerical solutions, we also tested the case of TSTAB=0.5. The results are shown in Figure 32. For comparison, we

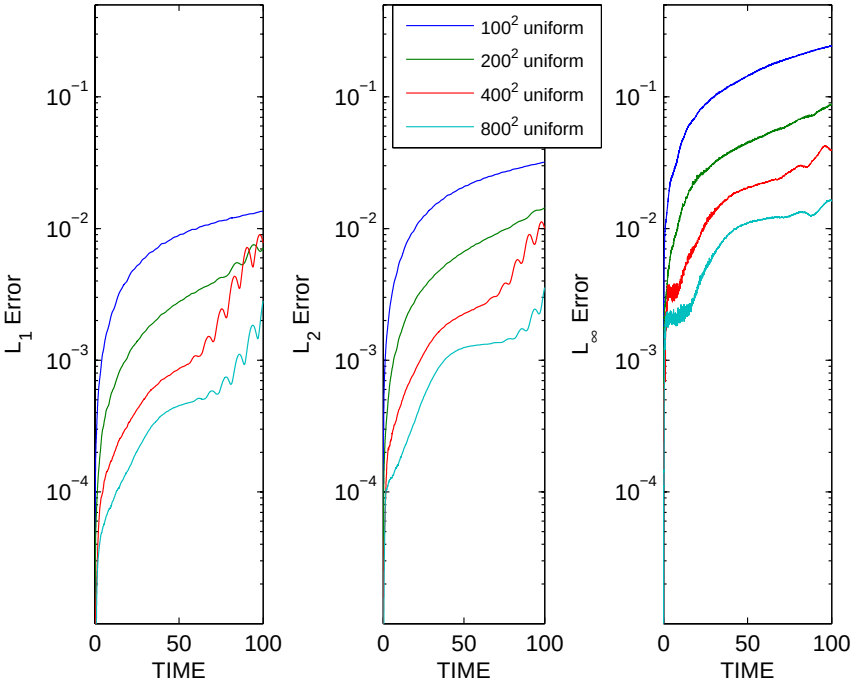

Figure 28: Numerical errors of different grids for vortex propagation problem. RAGE is used.

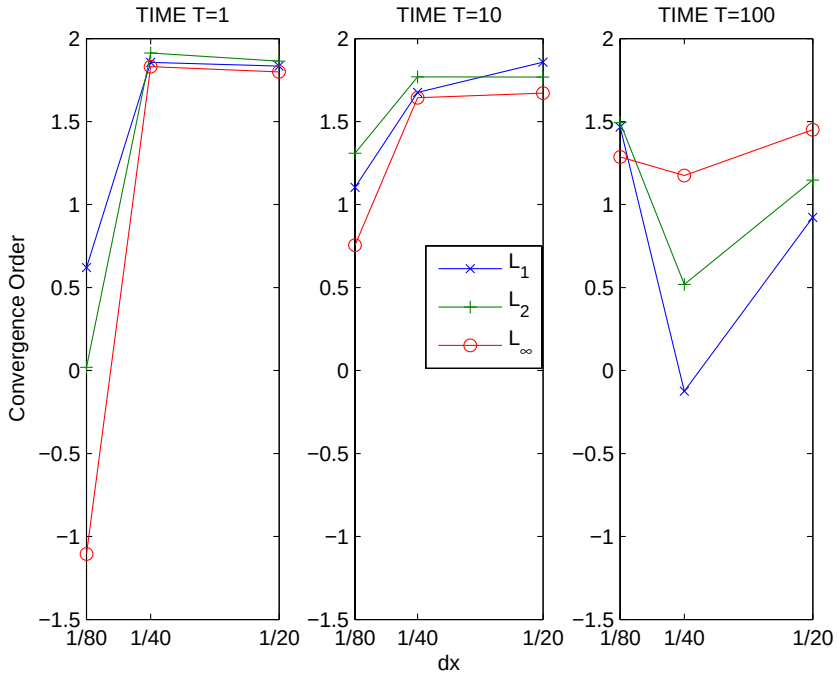

Figure 29: Convergence order at different times for vortex propagation problem. RAGE is used.

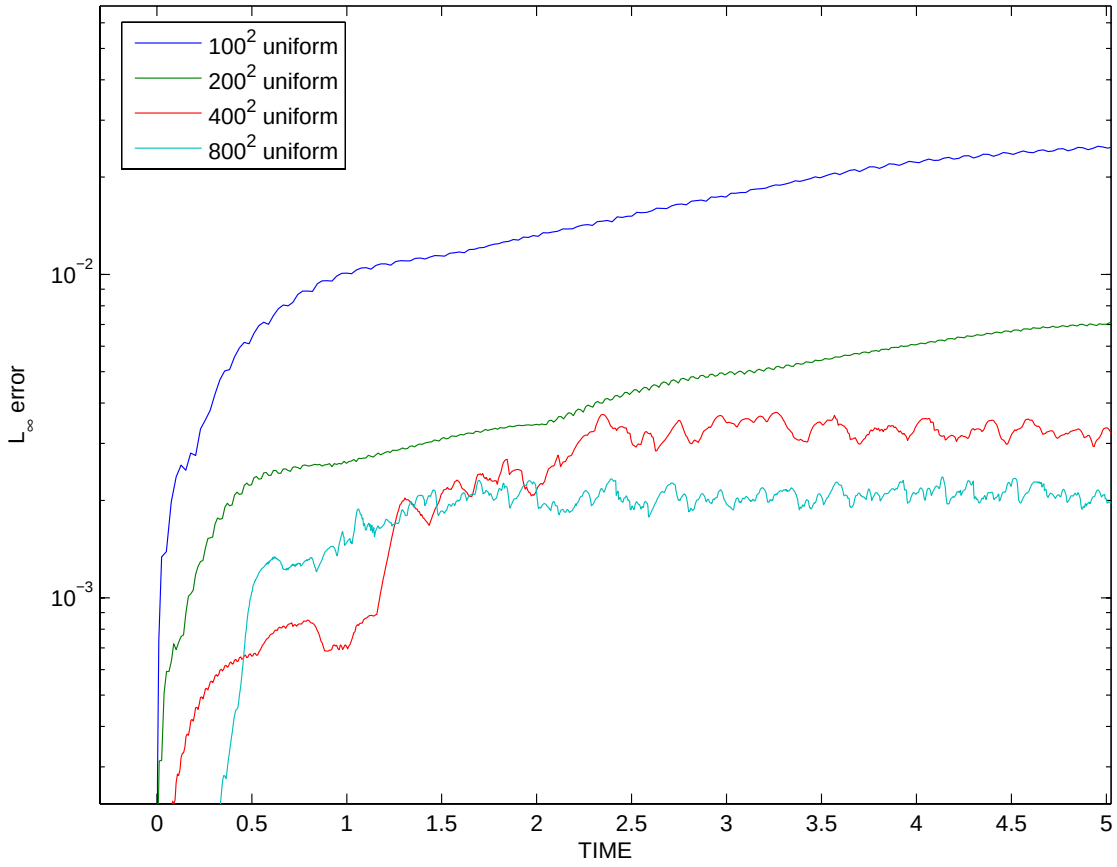

Figure 30: Numerical errors of different grids for vortex propagation problem. RAGE is used.

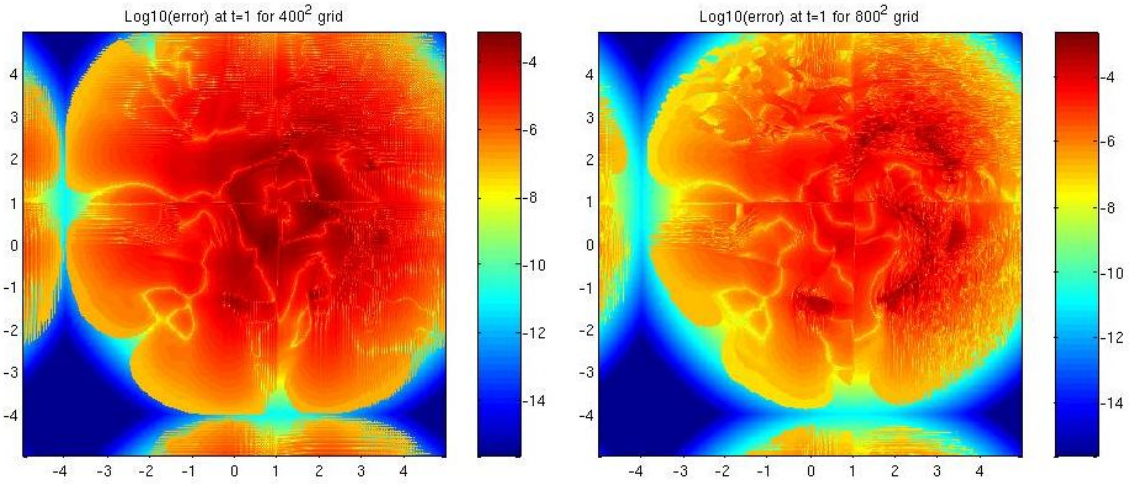

Figure 31: Error distribution of different grids at $t = 1$ for vortex propagation problem. RAGE is used.

included the results of the MUSCL-Hancock scheme with different TSTABs. To show that TSTAB cannot exceed 0.5 for RAGE hydro, we test the case of TSTAB=1.0, the result is shown in Figure 33. Figure 32 tells us that the difference between different CFL numbers is not as large as in AMR-MHD case. This is mainly because both CFL numbers are far smaller than one. There is another parameter in RAGE that may affect the accuracy of the numerical method. It is "hydrobet", which controls the artificial viscosity. The default value in RAGE is hydrobet=0.25. We tested the case of hydrobet=0.0. The result is also shown in Figure 33. Note that the artificial viscosity has less impact than the CFL number.

To test the accuracy and convergence of the AMR method, we used three base uniform grids: 100×100, 200×200, and 400×400. The refinement ratio between different levels in AMR is taken as two. The default values of CSTAB and TSTAB are used in time step control. Figure 34 shows the results for 100×100 base grid with different refinement levels compared with the same resolution uniform grids. The oscillations occur at earlier time than for the uniform grid with the same resolution. If we consider the solution before oscillations, the result of the three-level refinement AMR is worse than that of the two-level refinement AMR. All of the AMR results are worse than that of 200×200 uniform grid in term of L_1 and L_2 errors. In term of L_∞ error, only the two-level refinement is slightly better than the 200×200 uniform grid at early times.

Figure 35 shows the effect of CFL number on AMR process. It shows that the larger the CFL number, the smaller the L_1 error. Since the refinement region varies with the time stepsize, we also plot how the L_1 error divided by the average spacing changed with time and CFL number. The results are also shown in Figure 35. Another interesting thing related to the AMR refinement criteria in RAGE is that the refinement region increases a lot with the increase of the time stepsize. Figure 36 shows the average spacing and CFL number change for different CFL numbers and different schemes.

Figure 37 shows the results of different base grids with only one refinement level, compared with the uniform grids with the same resolution. As observed in the previous example, the 400×400 with 1-level refinement is worse than 400×400 without refinement. The 200×200 with 1-level refinement is only a slightly better than 200×200 without refinement at early times. The errors for early times are shown in Figure 38.

To understand length scale of the AMR grid, we plot the average volume for different refinements and different base grids in Figure 39. The refinement region is much larger than the previous example, although they have the same density profile. Figure 40 shows the refinement region and error distribution for 100×100 grid with 2-level refinement.

To study the error-cost performance of AMR, we plot the error convergence behavior at different times using the average spacing. The results are shown in Figure 41. Table 2 shows the summary on the performance of the AMR results.

Compared with the results of the first example (linear wave propagation), the error accumulation in time and convergence order at later times is quite different. Comparing the average volume in Figure 39 with that in 15 shows that AMR behaviors quite differently from the results of AMR-MHD. In AMR-MHD, the refinement is controlled by the curvature in density profile. Therefore, both examples generate almost the same size of the average volume. However, for RAGE code, the refinement region for vortex propagation problem is much larger than that for linear wave problem. It indicates that the refinement criteria in RAGE are quite different from those in AMR-MHD package.

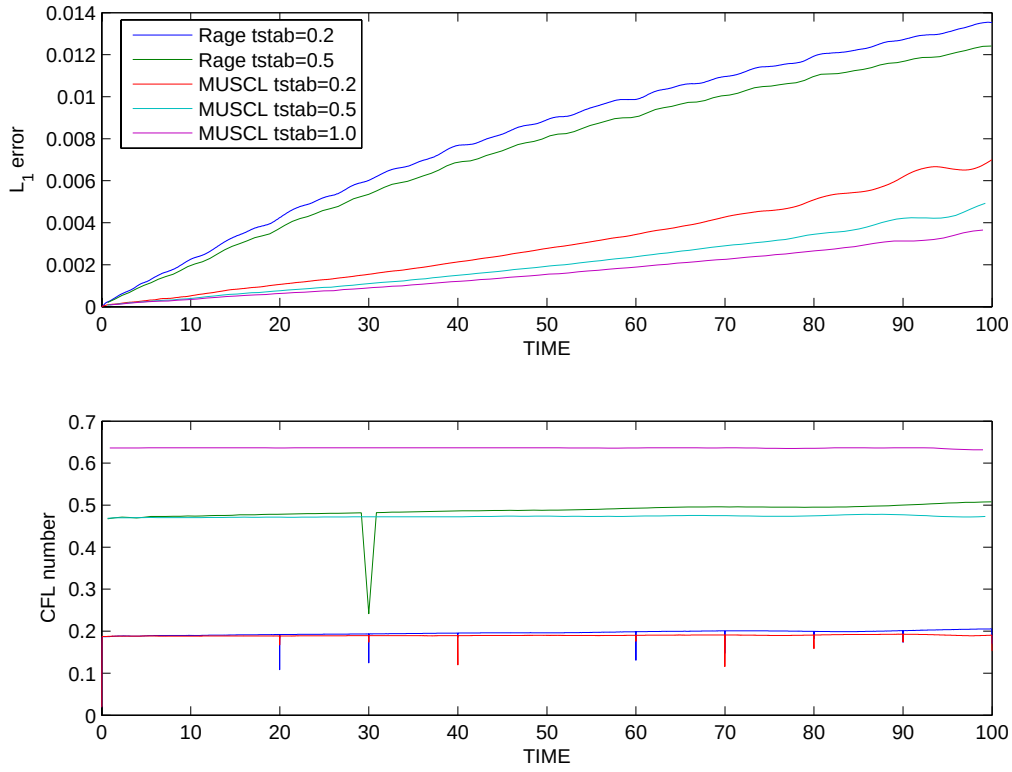

Figure 32: Numerical errors for different CFL numbers and different algorithms, 100×100 grid is used. RAGE is used.

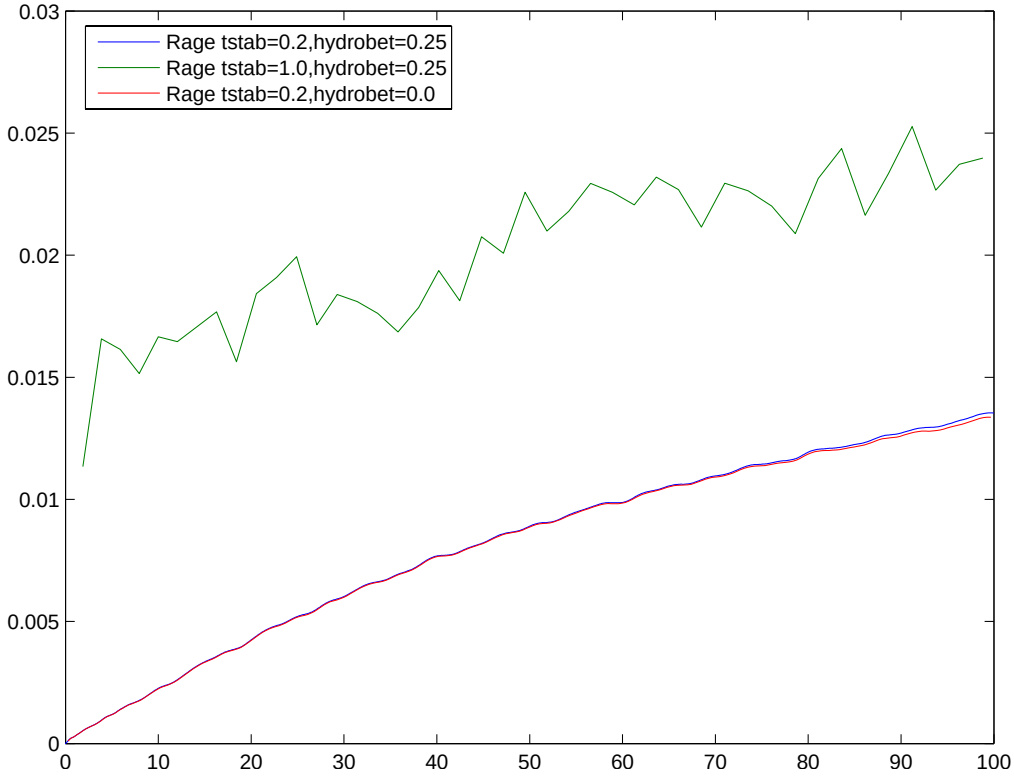

Figure 33: Numerical for a large TSTAB and different artificial viscosity. RAGE is used.

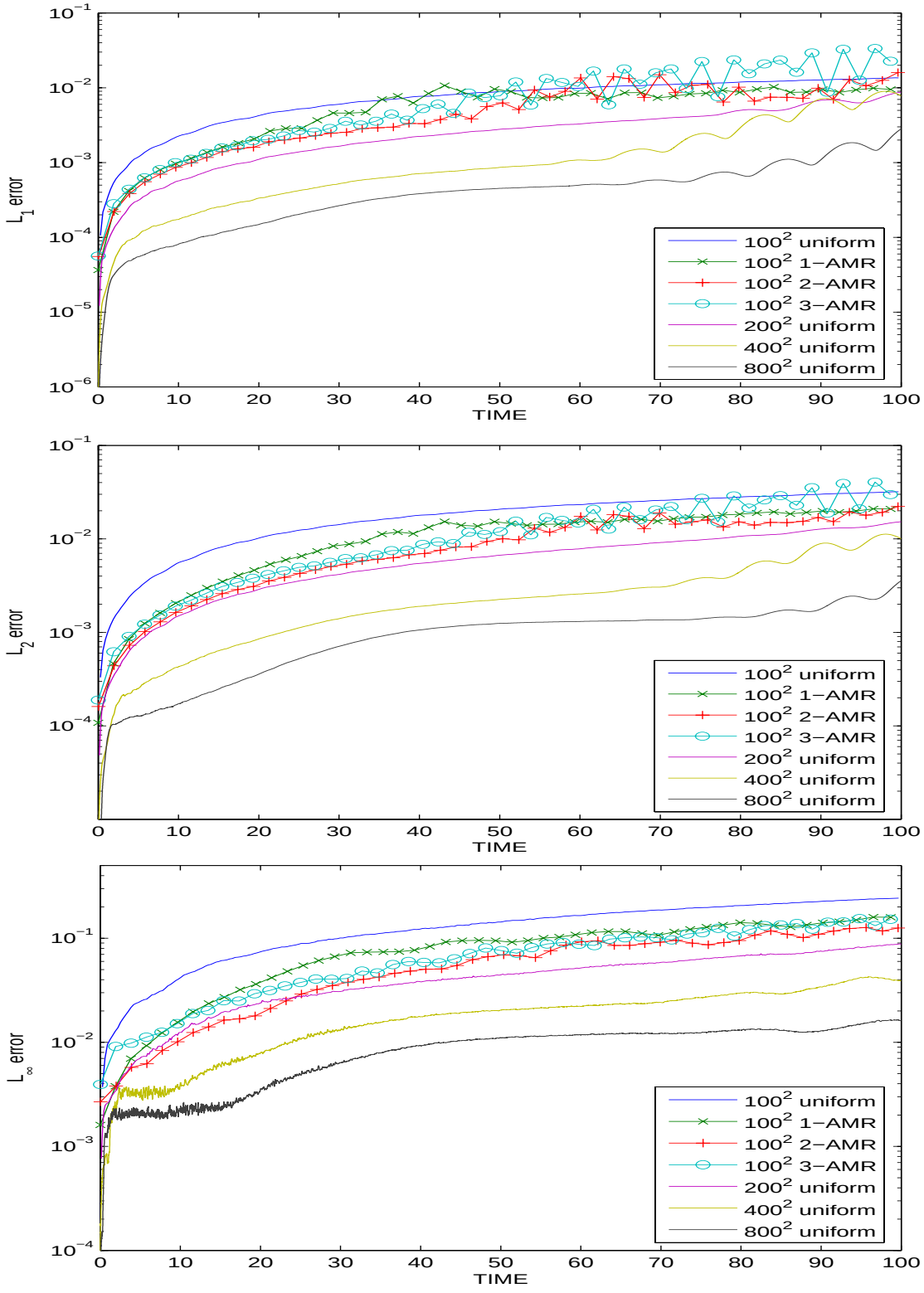

Figure 34: Numerical errors for different refinement levels for vortex propagation problem. RAGE is used.

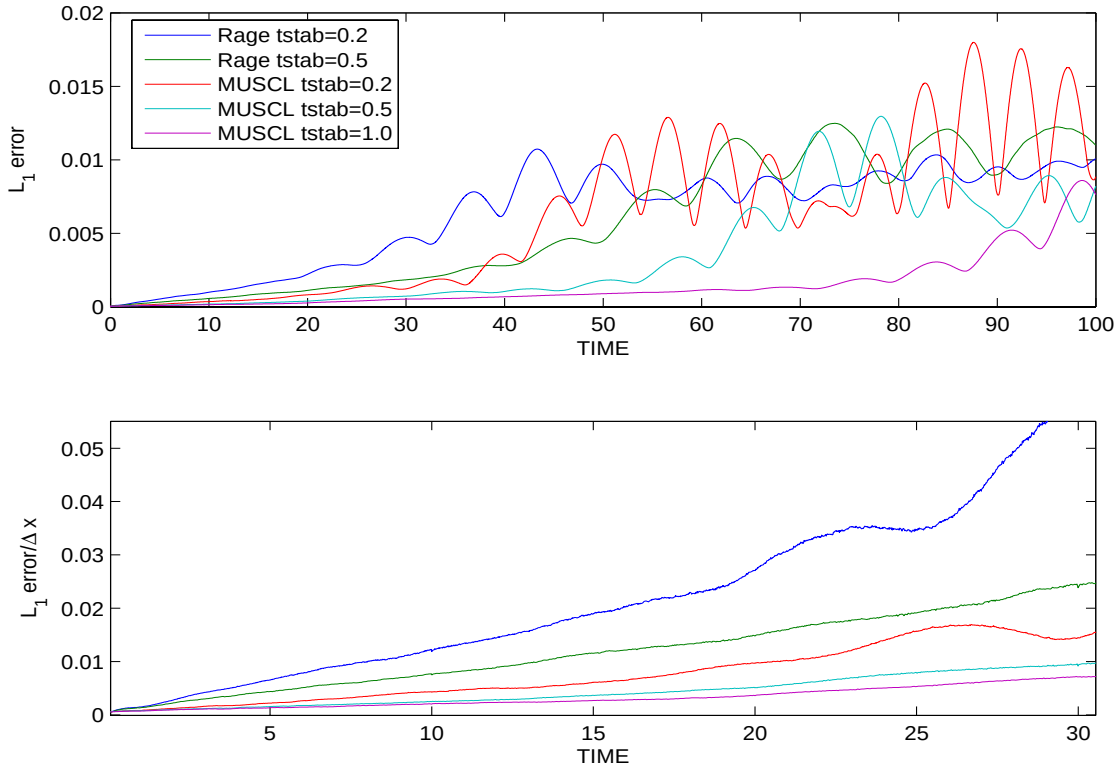

Figure 35: Numerical error for 100×100 with 1-level refinement grid for vortex propagation problem. The bottom figure has the same legend as the top figure. RAGE is used.

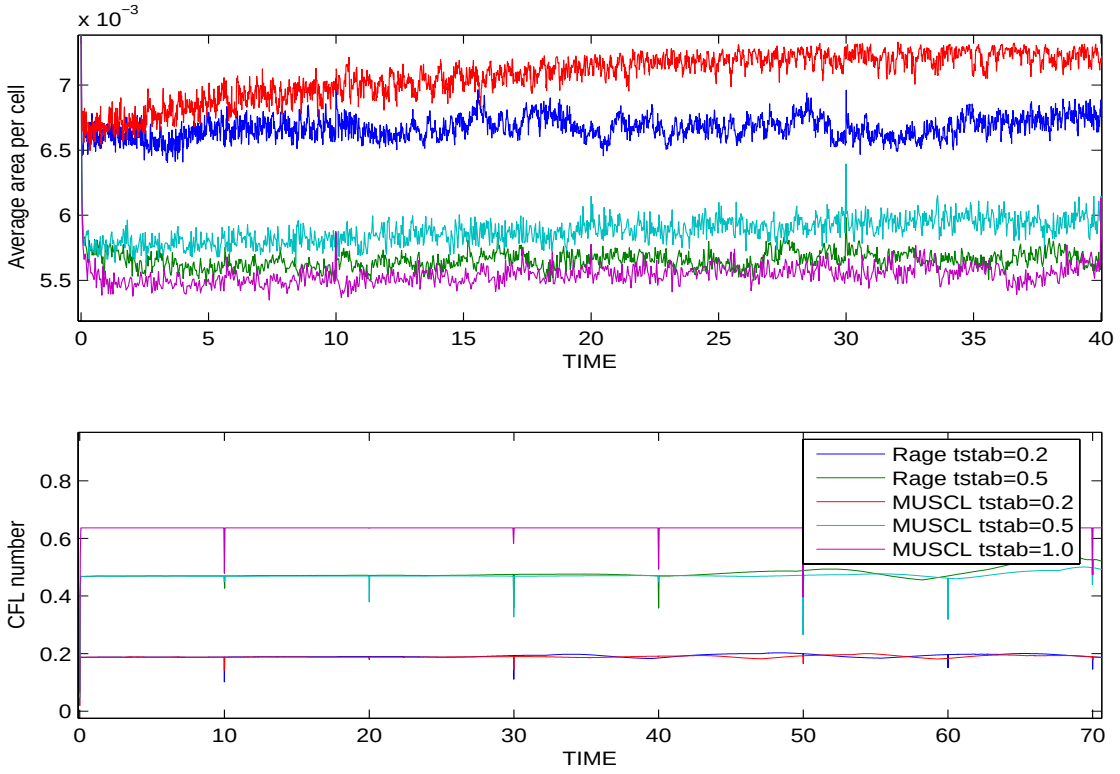

Figure 36: How refinement region changes with CFL number and time for vortex propagation problem. 100×100 with 1-level refinement grid is used. RAGE is used.

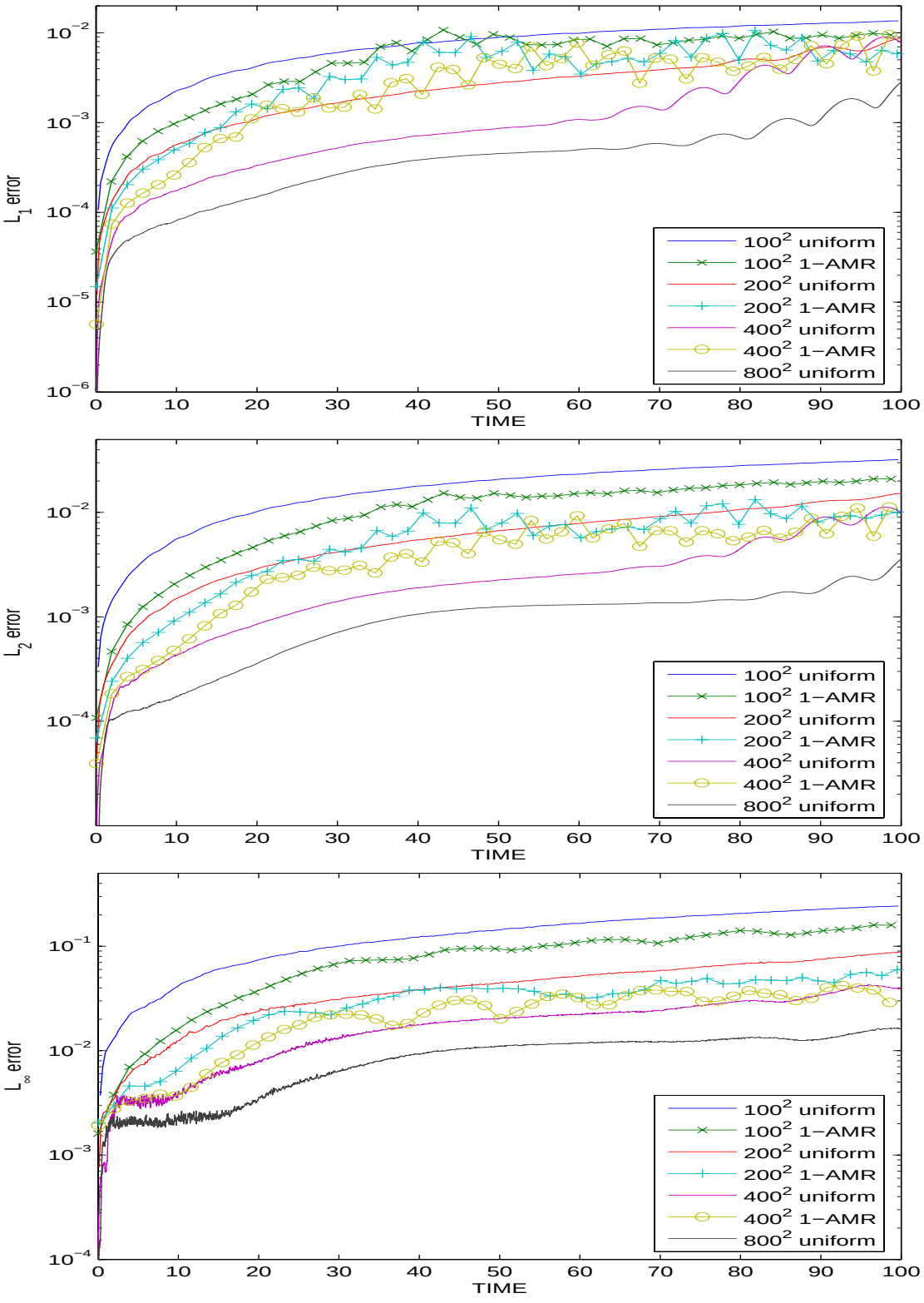

Figure 37: Numerical errors for different base grid with only one refinement level for vortex propagation problem. RAGE is used.

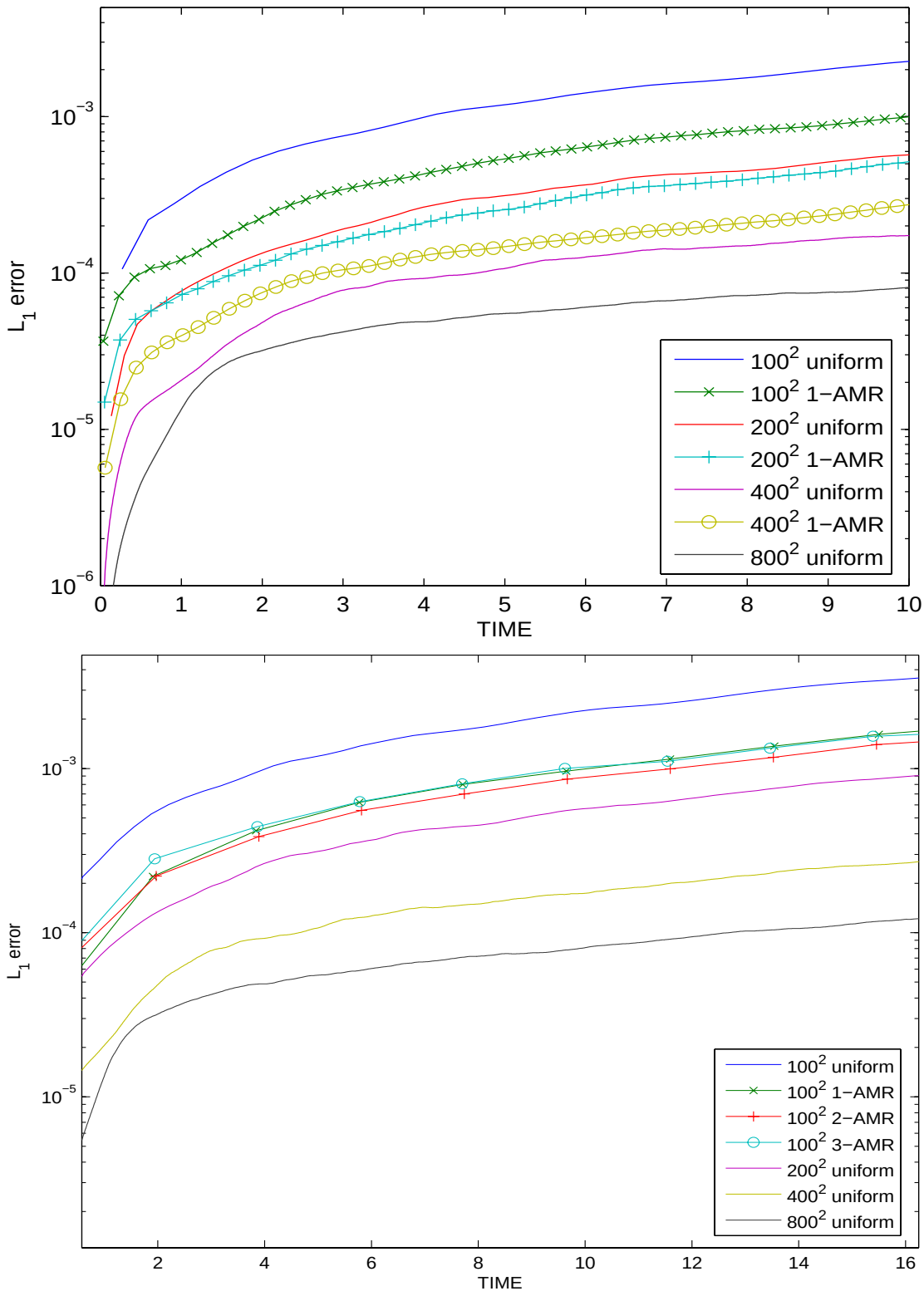

Figure 38: Numerical L_1 error at early times for AMR, compared with the same resolution uniform grid for vortex propagation problem. RAGE is used.

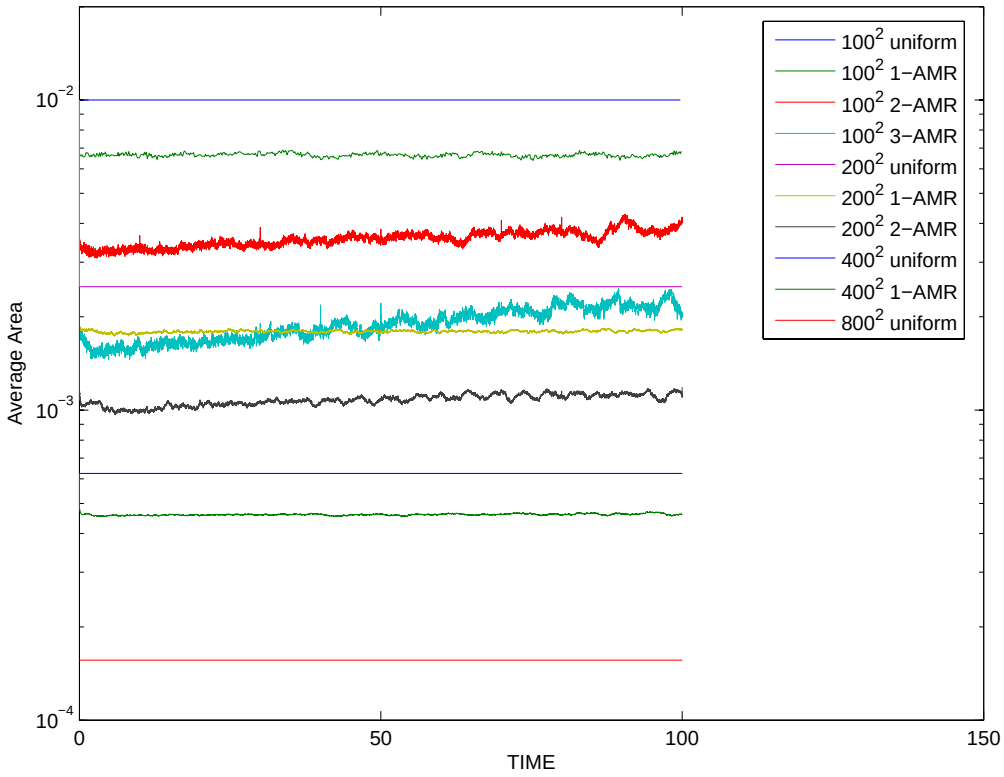

Figure 39: Average spacing of different grids with refinement for vortex propagation problem. RAGE is used.

Table 2: Error-cost performance of AMR computation for vortex propagation problem, where $\surd$ means it is better than uniform grid and $\times$ means it is worse than uniform grid. RAGE is used.

	100^2AMR(1)	100^2AMR(2)	100^2AMR(3)	200^2AMR(1)	200^2AMR(2)	400^2AMR(1)
L_1 t=1	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$
L_2 t=1	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$
L_∞ t=1	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$
L_1 t=10	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$
L_2 t=10	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$
L_∞ t=10	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\times$
L_1 t=100	$\times$	$\times$	$\times$	$\surd$	$\times$	$\times$
L_2 t=100	$\times$	$\times$	$\times$	$\surd$	$\times$	$\times$
L_∞ t=100	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$

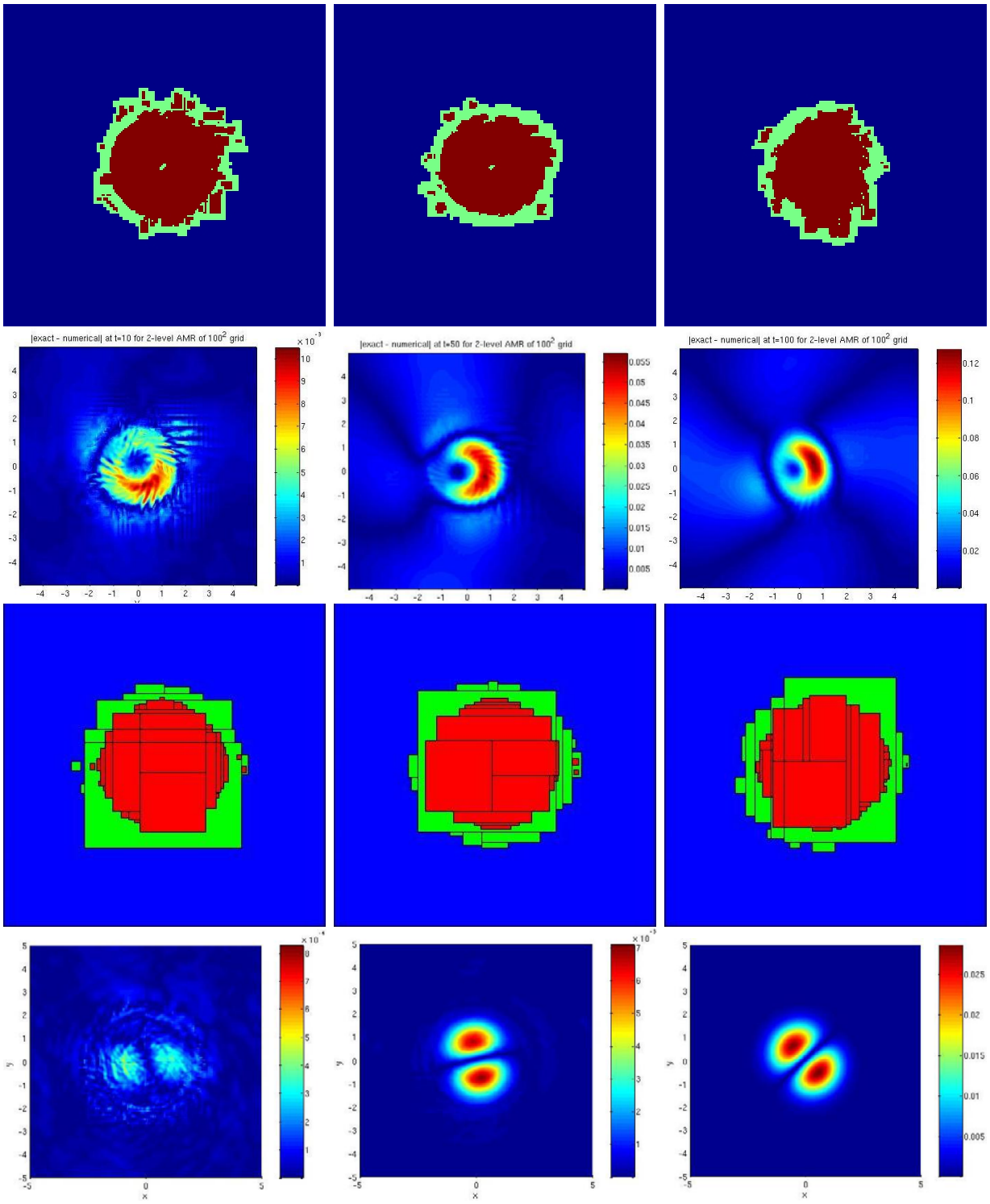

Figure 40: Refinement and error distribution with 2-level refinement for vortex problem. Left: t=10; middle: t=50,, right: t=100. RAGE is used in the top two lines with base grid 100x100. AMR-MHD is used in the bottom two lines with base grid 100x100.

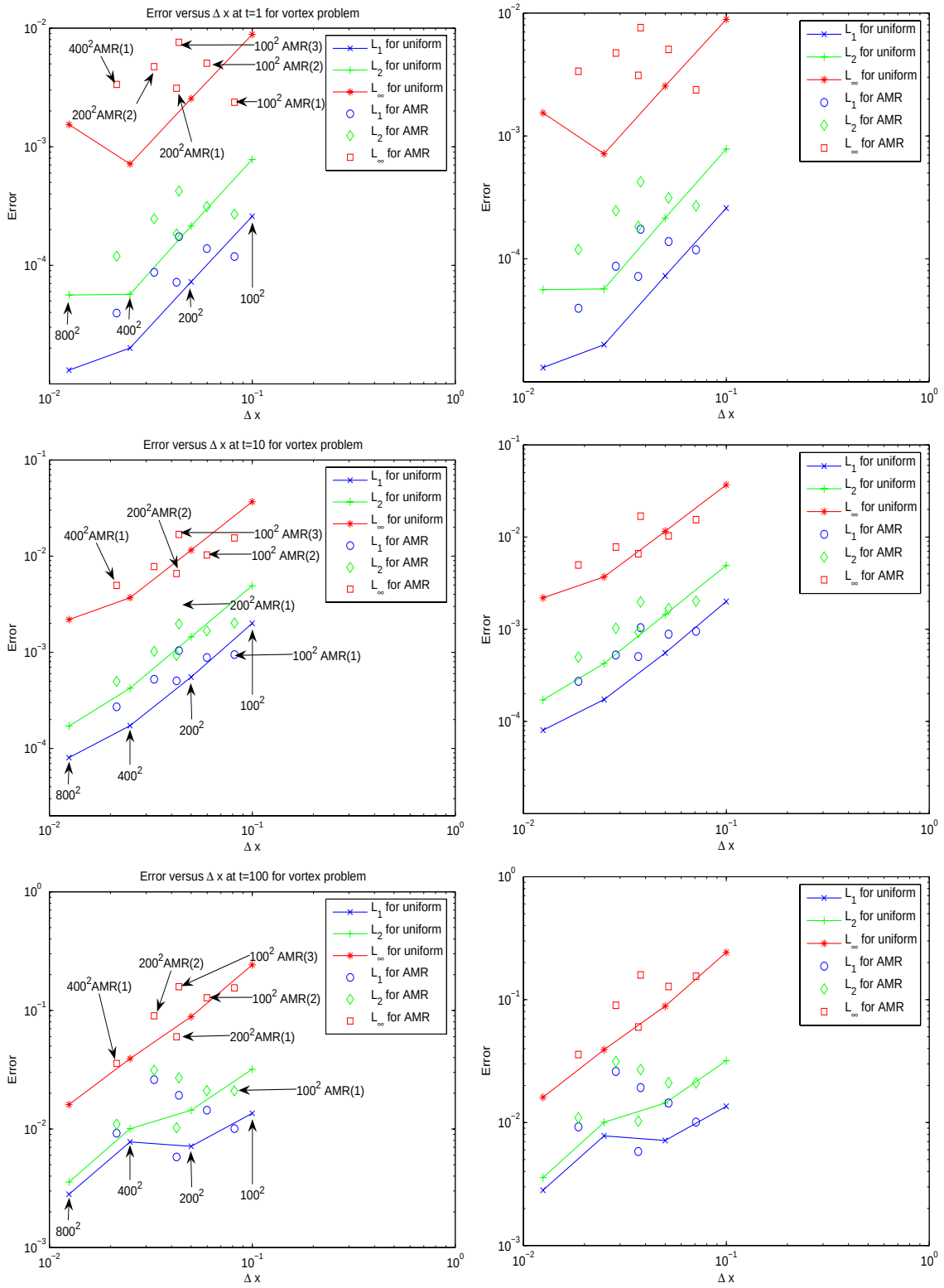

Figure 41: The error convergence behavior of AMR grid, compared with those of uniform grid. The right plot has considered 15% overhead of AMR while the left has not. The point above the uniform grid convergence line indicates that AMR has more cells but larger error than the uniform grid. RAGE is used to solve the vortex problem.

The AMR of RAGE also performs badly in the first step. Figure 42 shows different base grids with 1-level refinement compared with the same resolution uniform grid. The error jumps very high for the AMR calculations in the first step. Initially, we thought it might be because we used a cell-point value at midpoint of the cell in the initial conditions as the cell-average value. We recalculated the initial condition by integral over the whole cell. It does not improve the results. The comparison is shown in Figure 43. We should point out that the initial starting error is not unique to this problem. We got similar results in our linear wave problem.

5 Noh's Problem

In this section, we present the test results for Noh's problem. Gisler [9] reported the convergence studies on Noh's 3D problem solved on a (r, z) 2D cylindrical grid. He reached a conclusion that the convergence is not achieved on the adaptive grids. Even at a single level of refinement. there is noticeable deterioration in the solution when compared with a uniform grid at the same equivalent resolution. At third and fourth level of the refinement of 100×100 grid, a low-density bubble appears. Standard convergence analysis indicates the solution is diverging with increasing resolution.

5.1 Noh's 2D problem solved in Cartesian grid

We first test Noh's 2D problem solved on a 2-D Cartesian grid. It has an exact solution which is an infinite strength circularly symmetric shock reflecting from the origin. Behind the shock (i.e., inside the circle) the density is 16, the velocity is 0 and the pressure is 16/3. The shock speed is 1/3 and ahead of the shock, that is for $\sqrt{x^2 + y^2} > t/3$, the density is $(1 + t/\sqrt{x^2 + y^2})$ while the velocity and pressure remain the same as initially, i.e. velocity is directed toward the origin and pressure is zero.

5.1.1 Results of AMR-MHD

As reported in [5], many numerical schemes failed to solve this problem. We use four different uniform grids: 50×50, 100×100, 200×200, and 400×400. We first test it with a dimensional splitting scheme. For the AMR algorithm, we use a error tolerance TOLS=0.02.

Since this problem has a strong shock, the global convergence is no more than the first order. Fig. 44 shows the result of AMR with one-level refinement. The base grid is 100×100. Due to the dimensional splitting, the solution is not fully symmetric along the diagonal line.

Figs 45 and 46 show that the result of the uniform grid with resolution 200×200 sometimes is worse than that of the uniform grid with resolution 100×100. It is not clear to us why this happens. We think it might be due to the split scheme or the CFL number is too large. We recalculated the solution with unsplit scheme. We also tried a small CFL number for the split scheme. In both cases, the result for 200×200 grid is more accurate than that of 100×100 grid.

Fig. 45 also shows that the L_2 and L_∞ norm is not good for this problems to study the convergence behavior. This is because the error near the shock front dominates the total

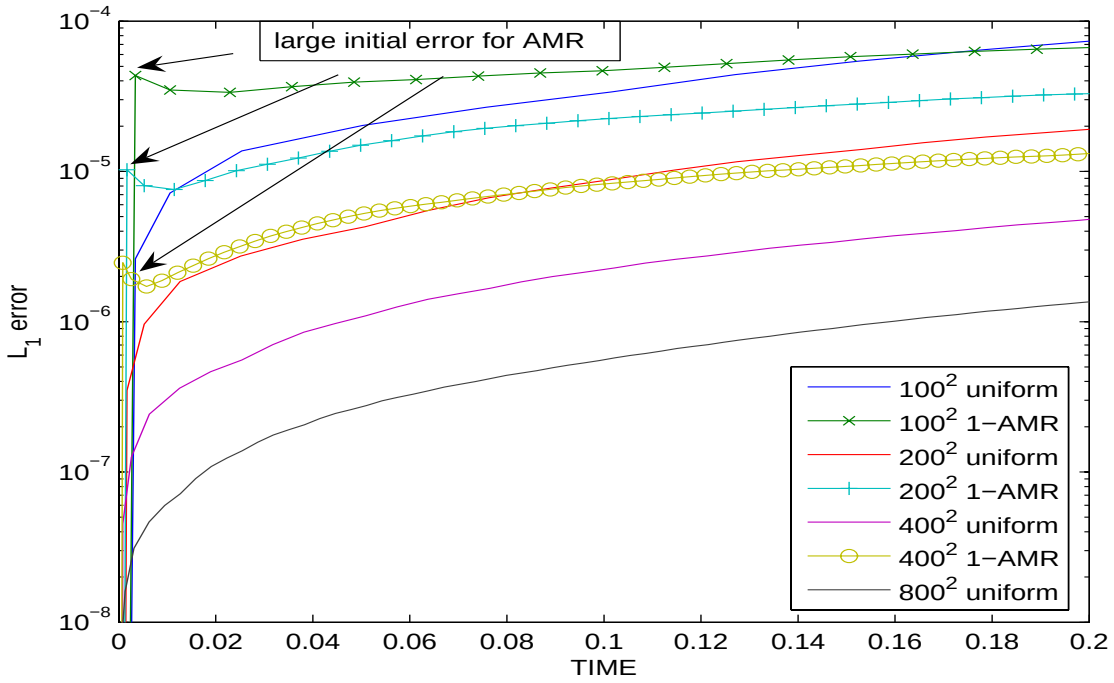

Figure 42: Numerical errors of AMR at the beginning of the time integration for vortex propagation problem. RAGE is used.

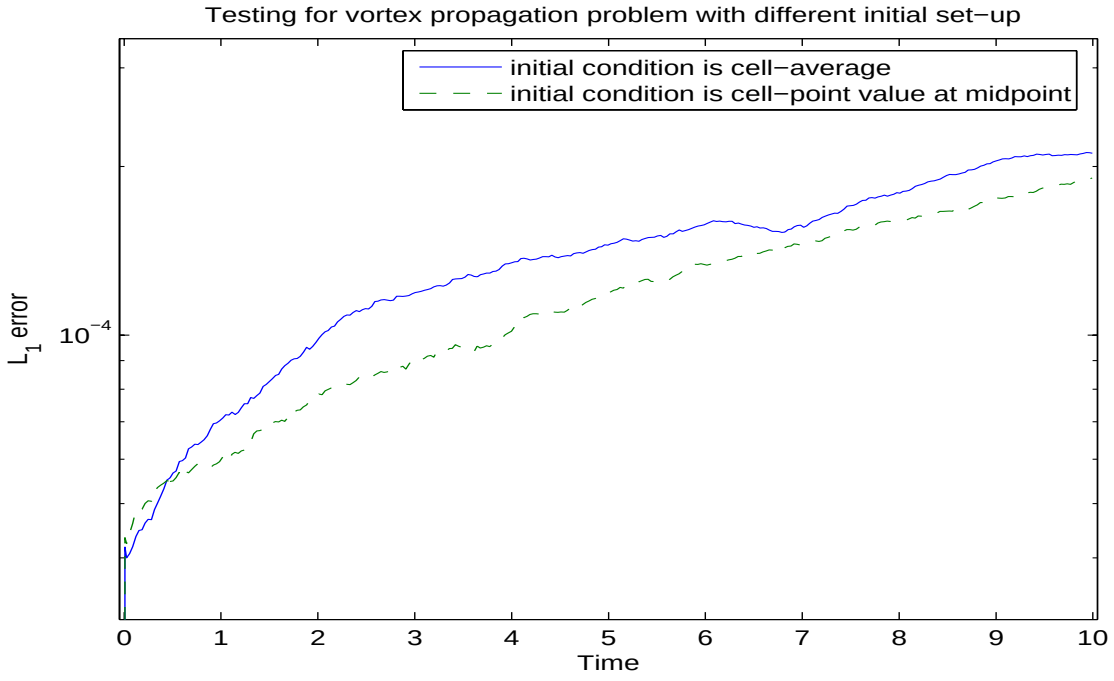

Figure 43: Numerical errors of different initial condition calculations for vortex propagation problem. RAGE is used.

error and we cannot obtain even the first order global convergence. We think it might be true that the L_1 error norm should be used in the convergence analysis for all the problems that contain discontinuities.

Fig. 46 shows the high CFL number produces more accurate results except for the 200×200 case. However, the difference between the two results is small, compared with that for the vortex propagation problem.

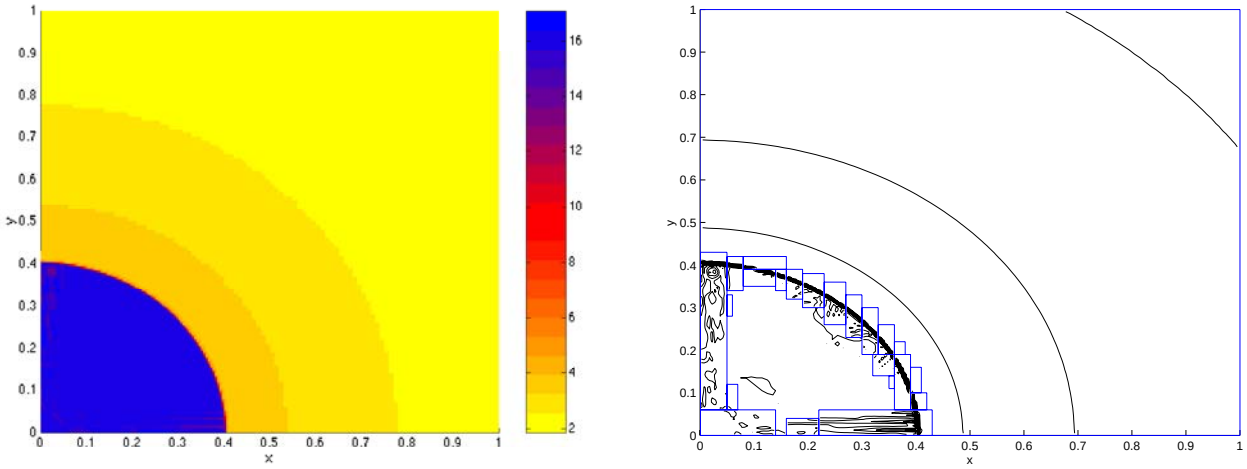

Figure 44: The density color and contour plot with refinement for Noh's 2D problem at $t = 1.2$. 20 contours between 2 and 16 are used. AMR-MHD is used.

Fig. 47 shows the numerical error of the AMR results, compared with the uniform grid results. We can see that for the base grid 200×200 and 50×50, the error of the AMR results is almost the same as that of the finest grid results. We also noticed that the local time step and locked time step give the same level of numerical errors.

Fig. 48 shows the numerical convergence behavior of AMR with one-level refinement. We can see that the results for AMR with minimum spacing as the characteristic length scale is almost identical to that of the uniform grid, except for the special 200×200 case.

5.1.2 Results of RAGE

RAGE does not allow direct implementation of the Dirichlet boundary conditions. Gisler [9] uses a much larger domain than actually used to ensure that the reflection from the boundary will not reach the interested region. We set up the problem domain differently from [9]. We defined the simulation domain as [0,1.2]×[0,1.2], where only [0,1]×[0,1] is the region of interest. The extra regions, [0,1.2]×[1,1.2] and [1,1.2]×[0,1], are defined as freeze regions. Numerical tests show that it does not matter how big the freeze region is. The values in freeze regions are restored to the exact solution after every time step by a routine in module "module_test_hydro". We modified that routine for the purpose of this example. After these modifications, the reflection from the boundary can be neglected.

We use four different uniform grids: 100×100, 200×200, 400×400, and 800×800. The freeze regions are also discretized into the same kind of grid. Therefore, the grids we actually use in simulation are 120×120, 240×240, 480×480, and 960×960. As Gisler [9],

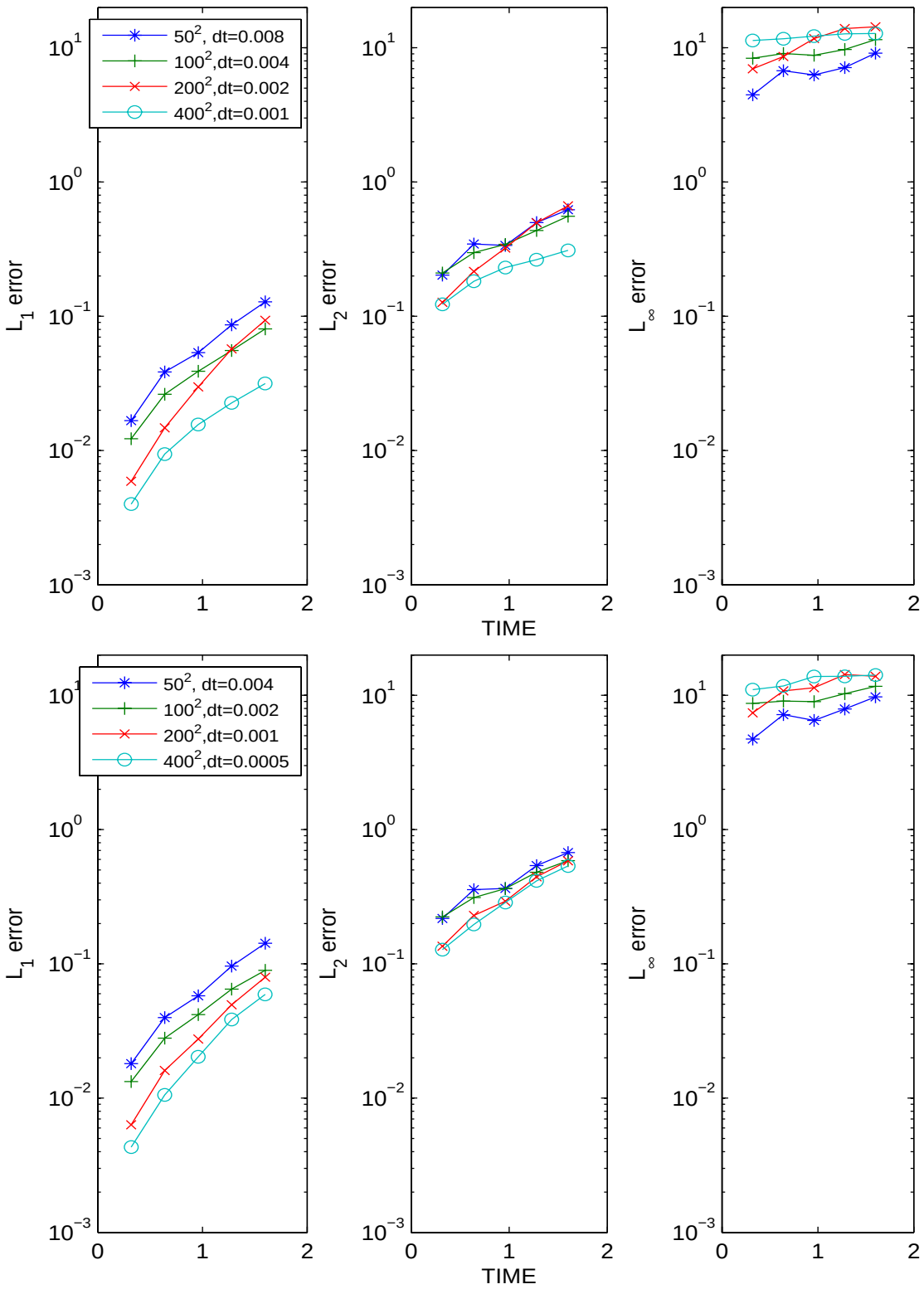

Figure 45: Numerical errors (L_1, L_2, and L_∞) for the Noh's problem at different times for different uniform grids. AMR-MHD is used.

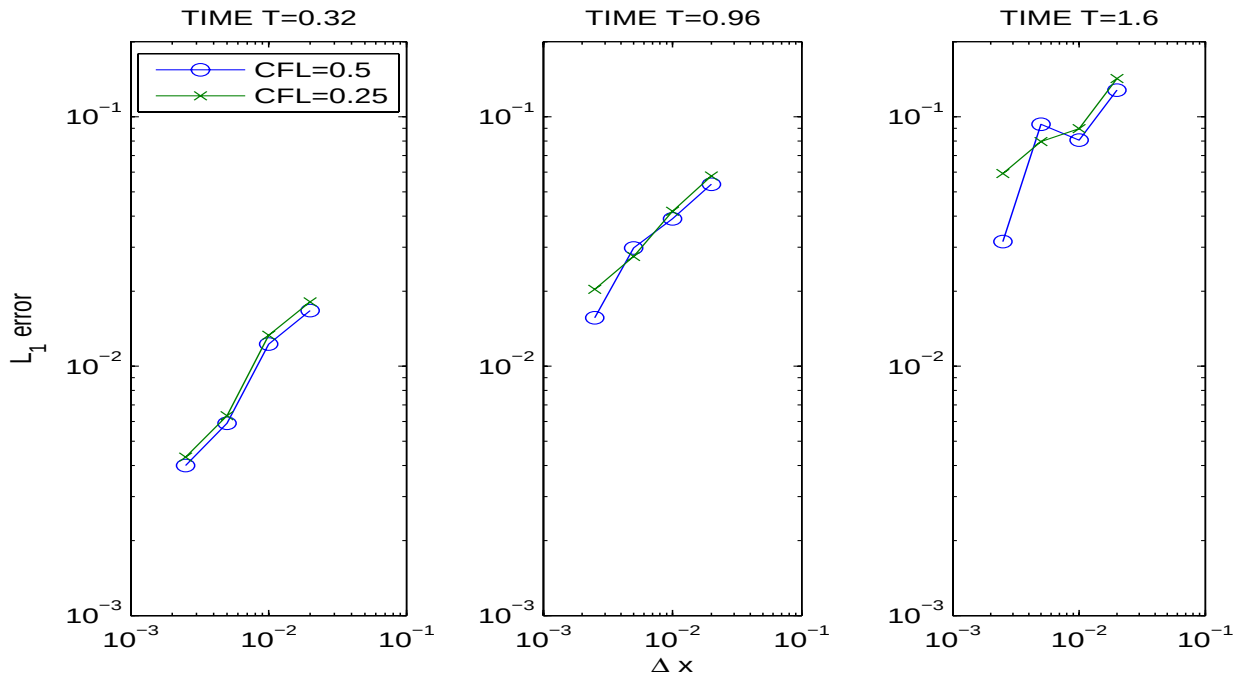

Figure 46: Numerical errors (L_1, L_2, and L_∞) for the Noh's problem at different times for different CFL numbers. AMR-MHD is used.

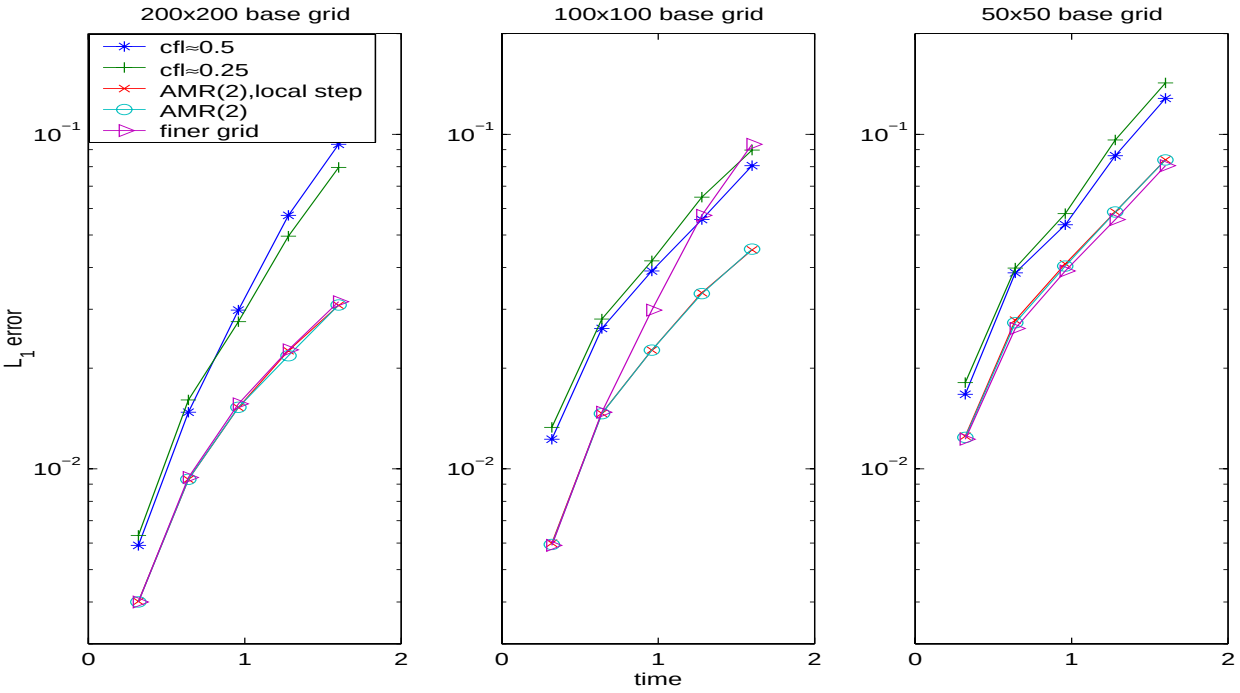

Figure 47: Numerical errors (L_1) of the AMR results compared with the uniform grid results for the Noh's problem at different base grid. AMR-MHD is used.

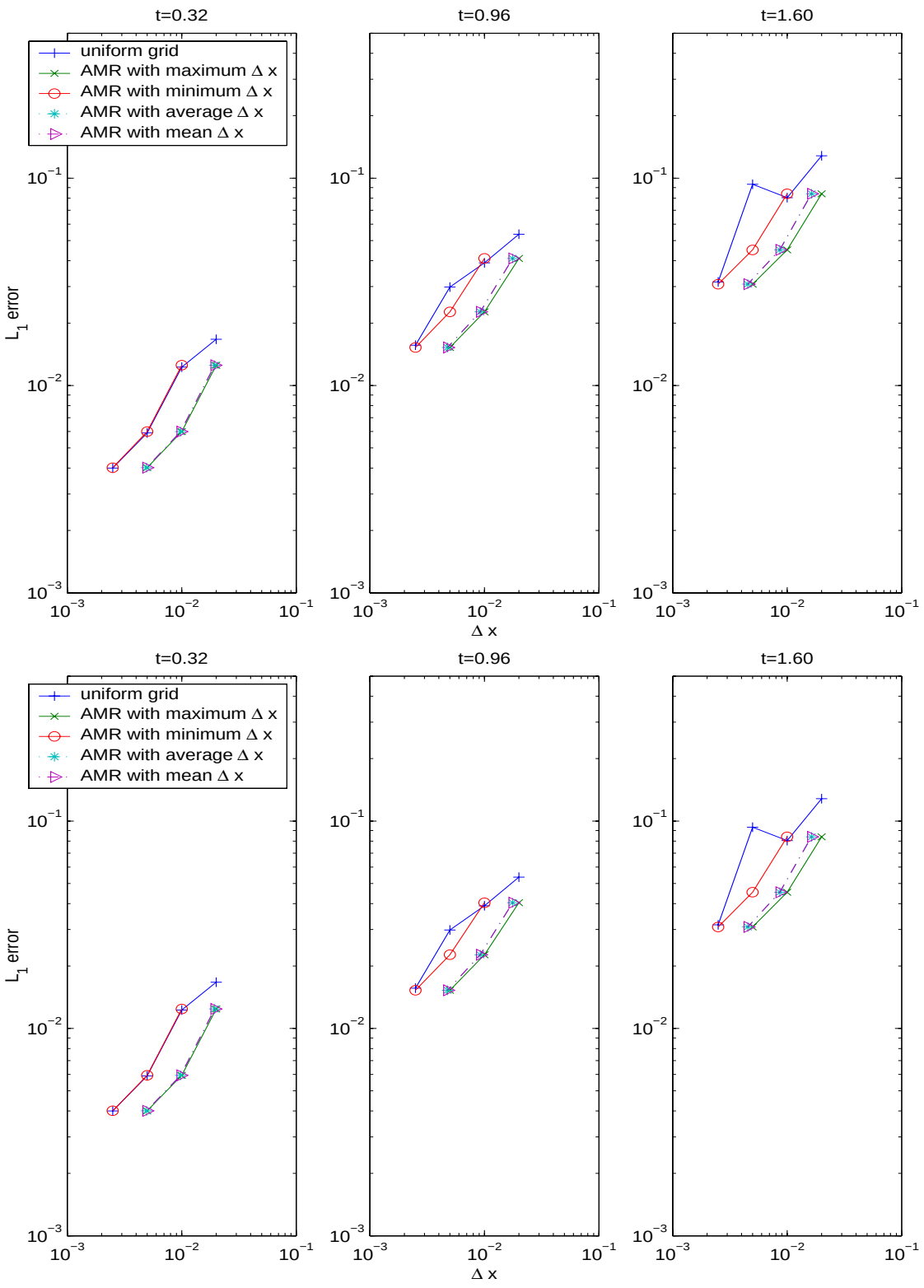

Figure 48: Numerical errors (L_1) for the Noh's problem at different times. The top one is for the local time step and the bottom one is for the locked time step. AMR-MHD is used.

we split the whole domain into several bands. The bands, numbered 1 through 8 from the innermost radial band outward, are chosen in such a way as to bracket the shock (in band 4) and to have roughly similar numbers of the cells. The last band (8) is defined differently from [9]. It includes the rest of interested domain ([0,1.0]×[0.1.0]).

Figure 49 shows the convergence order for uniform grid at different region bands. It is clear that the second order accuracy is achieved in regions outside of shock (pre-shock). Even though the region 8 is larger and close to the boundaries, the second order accuracy is still achieved, which demonstrates that our boundary freeze set-up works. In regions inside of the shock (post-shock), the convergence results are not so good. We even get negative order in some regions.

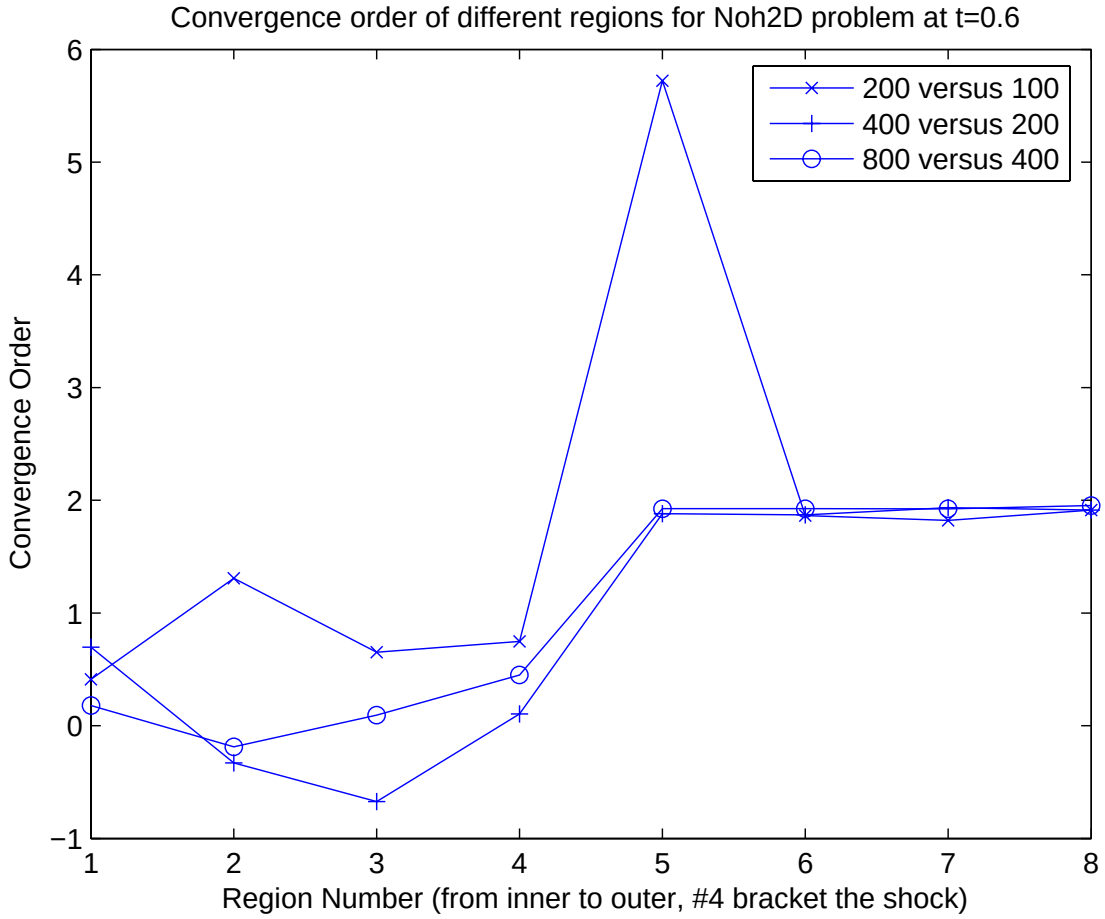

Figure 49: Convergence order of different regions for Noh2d Problem. RAGE is used.

Figure 50 shows the result of L_1 error for different regions and different grids. It shows the RAGE algorithm achieves the 2nd order accuracy in pre-shock bands for uniform grid.

Figure 51 shows the results of L_1 error for 100×100 grid with different refinement levels. For the pre-shock regions, it shows that except the band5 with 1-level refinement, the AMR results are not convergence. It indeed diverges, which is in good agreement with the results of Gisler [9]. For post-shock regions, The AMR is slightly better in band1 and band2 than

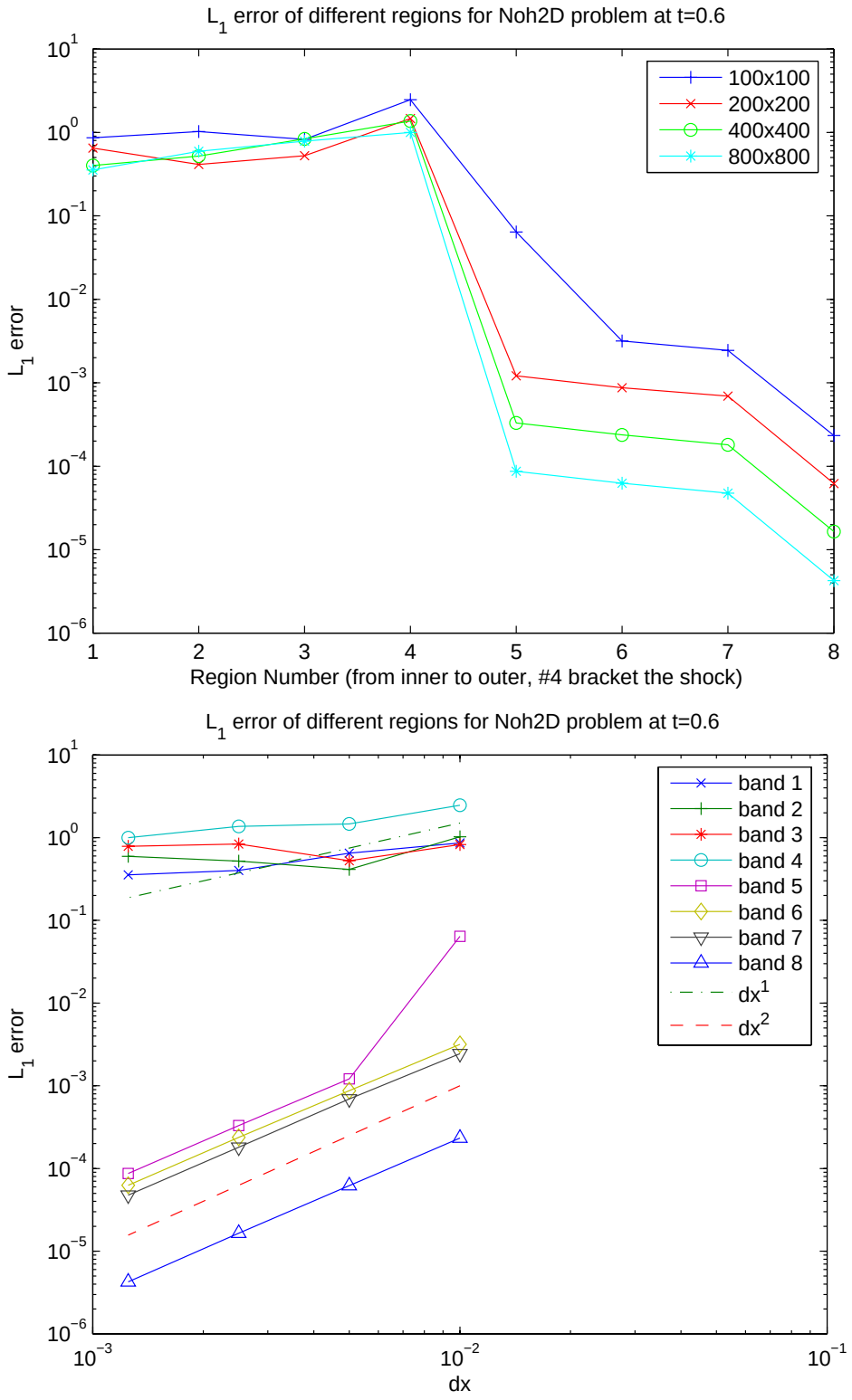

Figure 50: Numerical errors of different grids for Noh2d problem. RAGE is used.

the uniform grid without refinement. Compared with the results of Gisler [9], our three-level refinement results are better. No density bubble appears in the either third or fourth level. The difference may indicate that the 2-D and 3-D Noh problem is quite different in numerical simulation.

Figure 52 shows the results of L_1 error and convergence behavior for different region bands for the same resolution grids with different refinement levels. It consists of two sets of grids: 100×100 with 3-level refinement, 200×200 with 2-level refinement, 400×400 with 1-level refinement, and 800×800 uniform grid; 100×100 with 2-level refinement, 200×200 with 1-level refinement, and 400×400 uniform grid. It shows the more refinement levels, the larger the error.

Figures 53 and 54 show the results of L_1 and L_2 error respectively for the whole time history. The results of L_2 norm shows that at least at early times, the AMR achieves almost the same accuracy as the uniform grid with the finest resolution. This is in good agreement with the results of AMR-MHD. To see that more clearly, we split the whole time history into three stages, which is shown in Figure 55. It shows that at early time, the error in L_2 norm is in the same level for both the AMR and uniform grid, Then in the second stage, the AMR results of 400^2 begin to deviate from the uniform grid, and the other two still maintain the same accuracy level. At the last stage, only the AMR results of the 100^2 grid maintain the same accuracy level, and the other two deviate from the uniform grid results. We remark that the consistency of AMR and uniform grid results may be due to the fact that almost everywhere is refined by RAGE refinement algorithm for this problem.

Figure 56 shows the average cell-volume for different refinements and different base grids. It indicates the AMR refines almost everywhere after a short period of time. This is quite different from the results of AMR-MHD, where only a small region (6%) along the shock front gets refined. Gisler [9] found the same results using the RAGE code. Consider the overhead of AMR processing, the AMR of RAGE is not efficient for this problem.

To study the error-cost performance of AMR, we plot the error convergence behavior at different times using the average spacing. The results are shown in Figure 57. Table 3 shows the summary on the performance of the AMR results for Noh2d problem.

Table 3: Error-cost performance of AMR computation for Noh2d problem, where $\surd$ means it is better than uniform grid and $\times$ means it is worse than uniform grid. RAGE is used.

	100^2AMR(1)	100^2AMR(2)	100^2AMR(3)	200^2AMR(1)	200^2AMR(2)	400^2AMR(1)
L_1 t=0.2	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_2 t=0.2	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_∞ t=0.2	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_1 t=0.4	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_2 t=0.4	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_∞ t=0.4	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_1 t=0.6	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_2 t=0.6	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
L_∞ t=0.6	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$

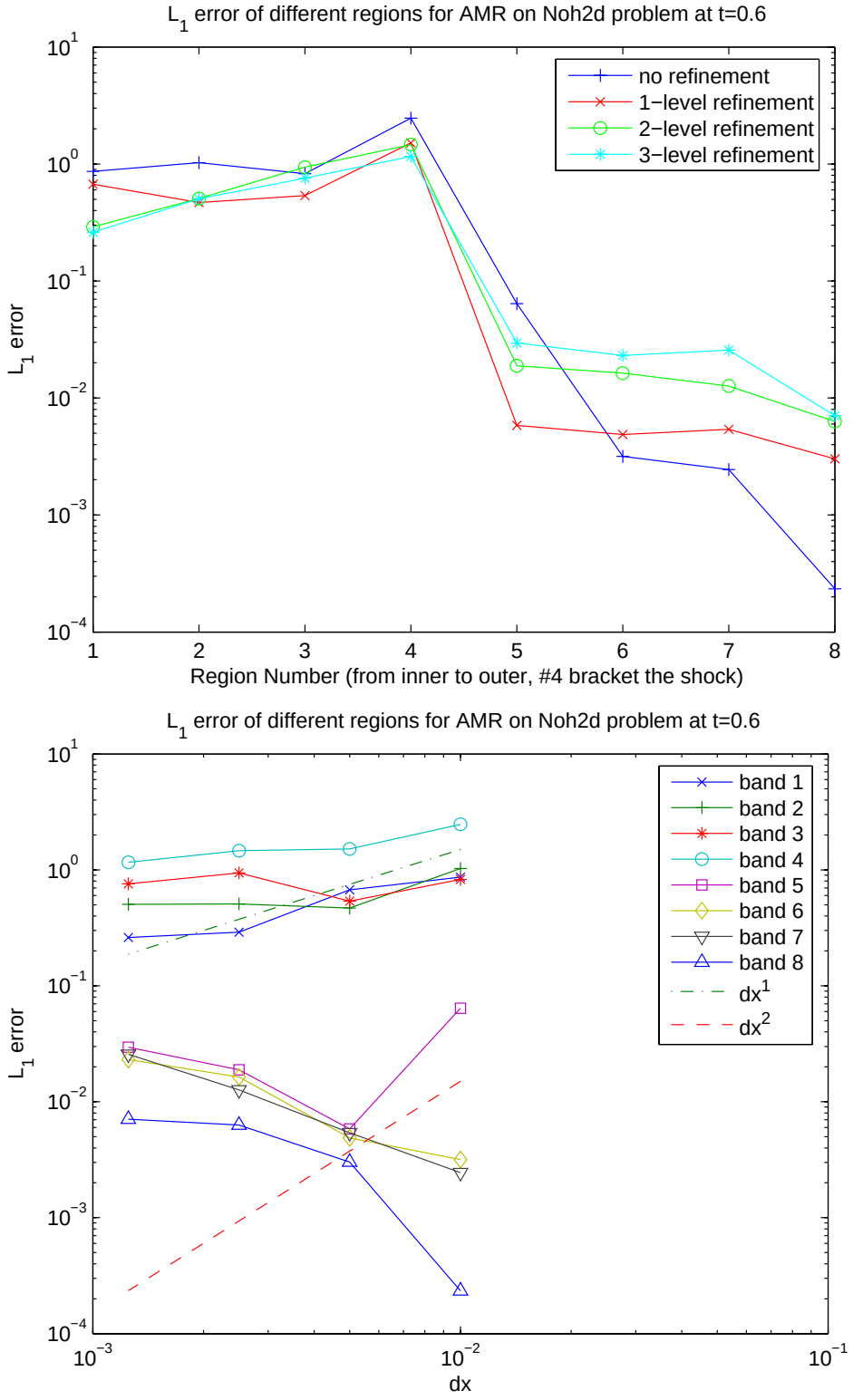

Figure 51: Numerical errors of 100×100 grid with different refinement levels for Noh2d problem. RAGE is used.

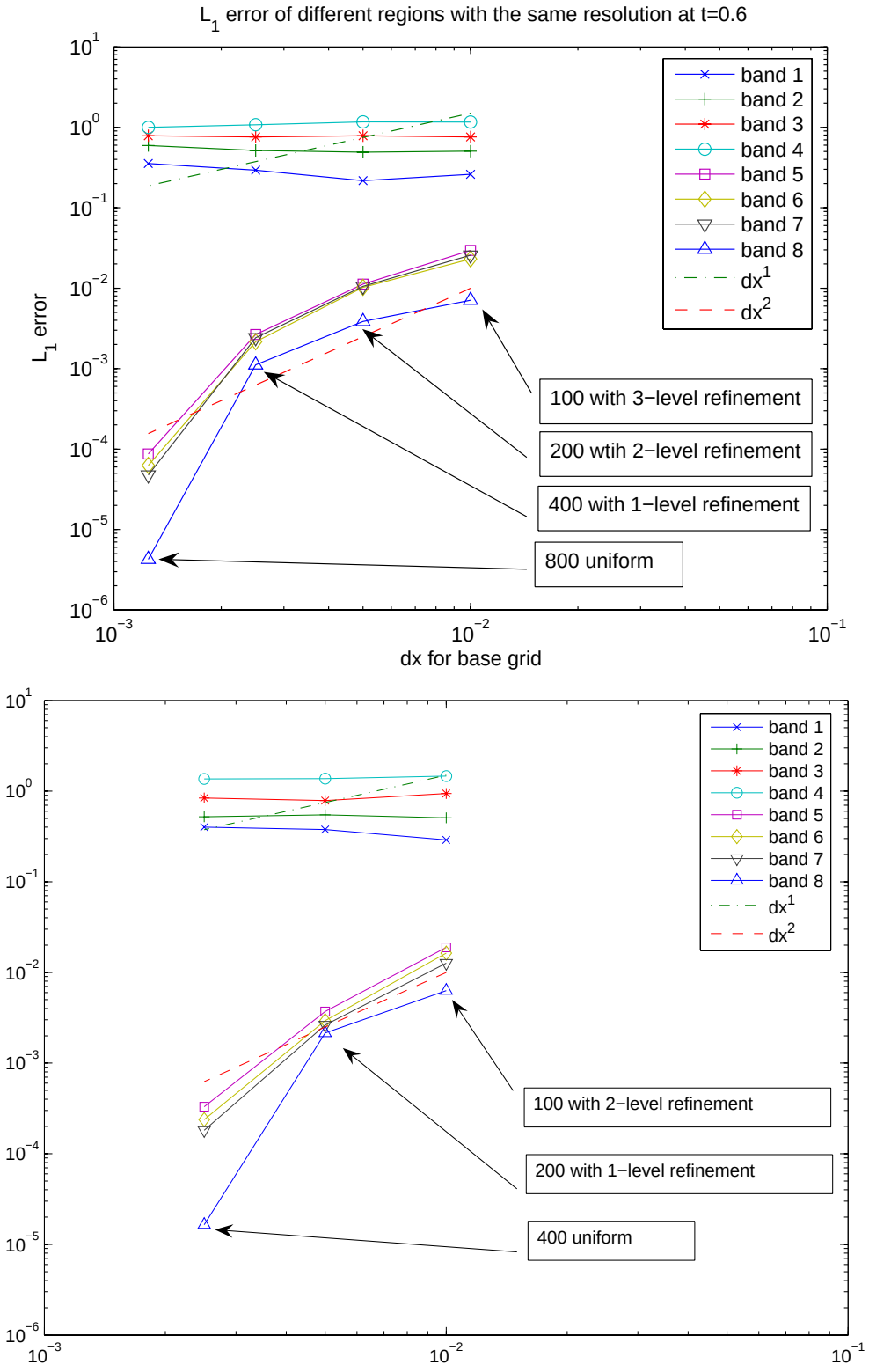

Figure 52: Numerical errors of grids with the same resolution and different refinement levels for Noh2d problem. RAGE is used.

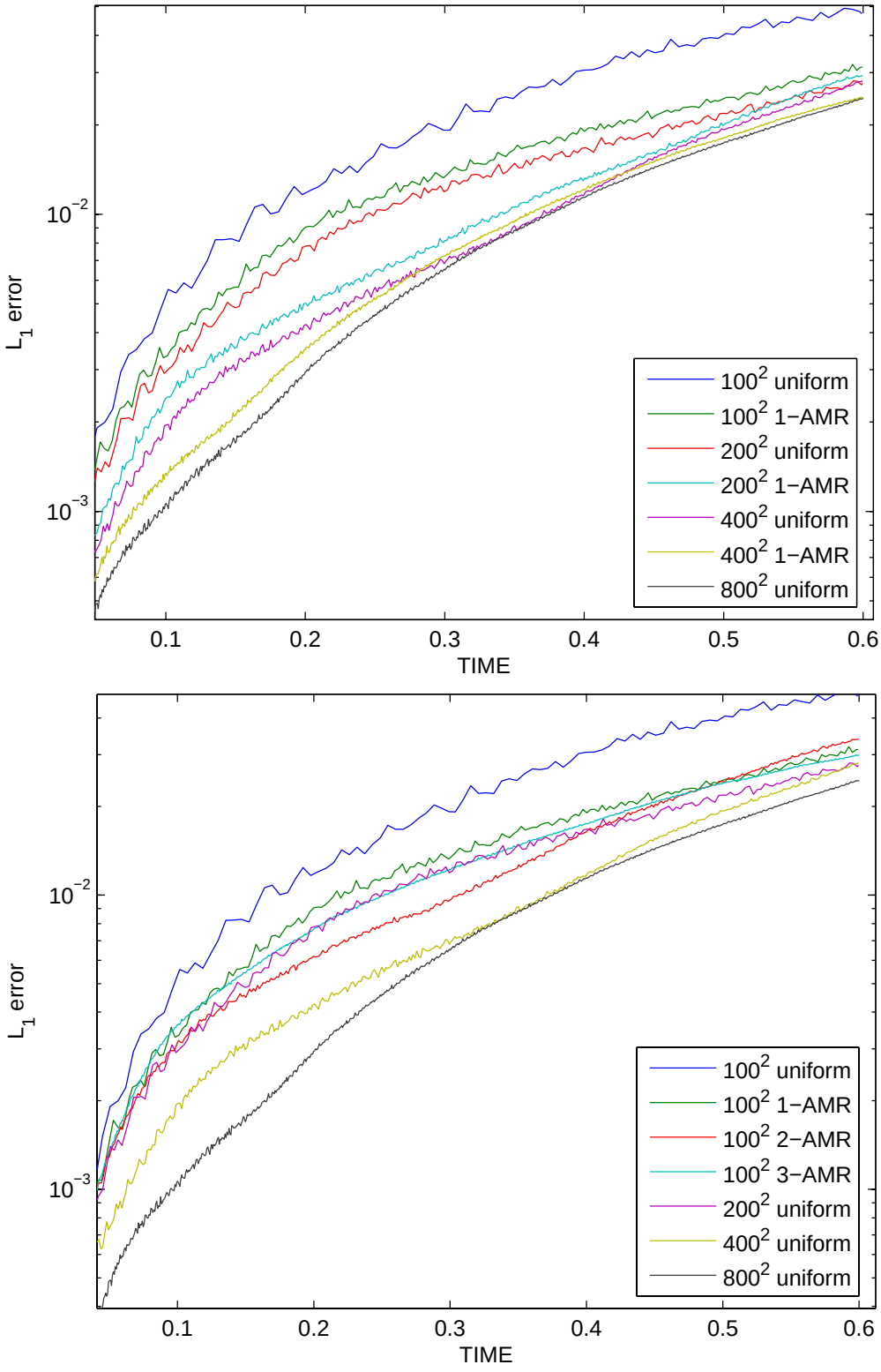

Figure 53: Numerical errors of AMR for Noh2d problem. RAGE is used.

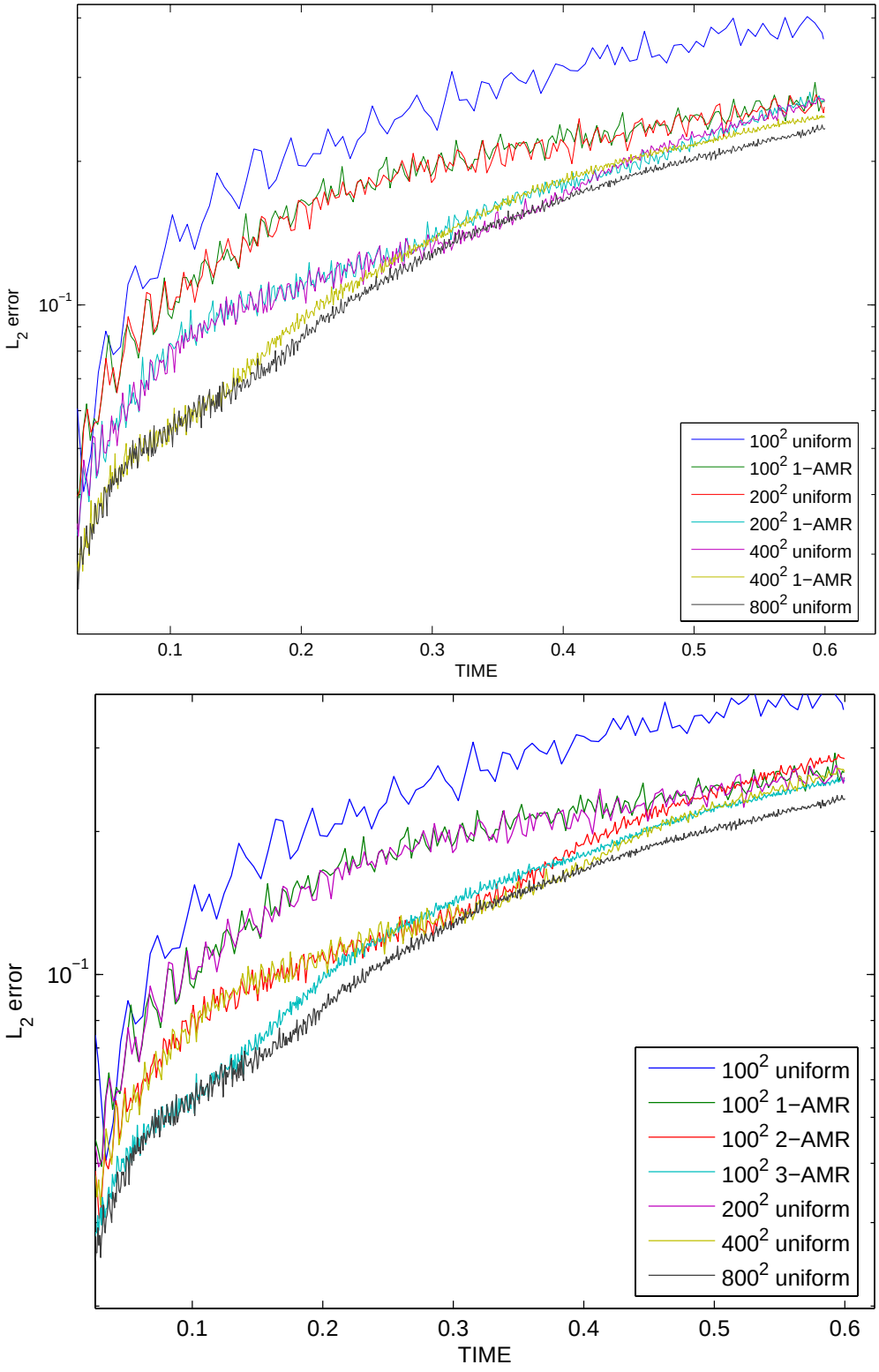

Figure 54: Numerical errors of AMR for Noh2d problem. RAGE is used.

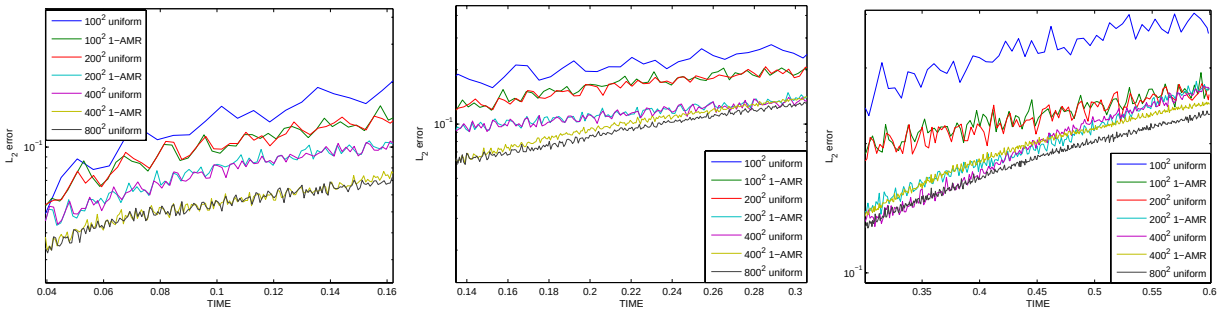

Figure 55: Numerical errors of AMR for Noh2d problem. RAGE is used.

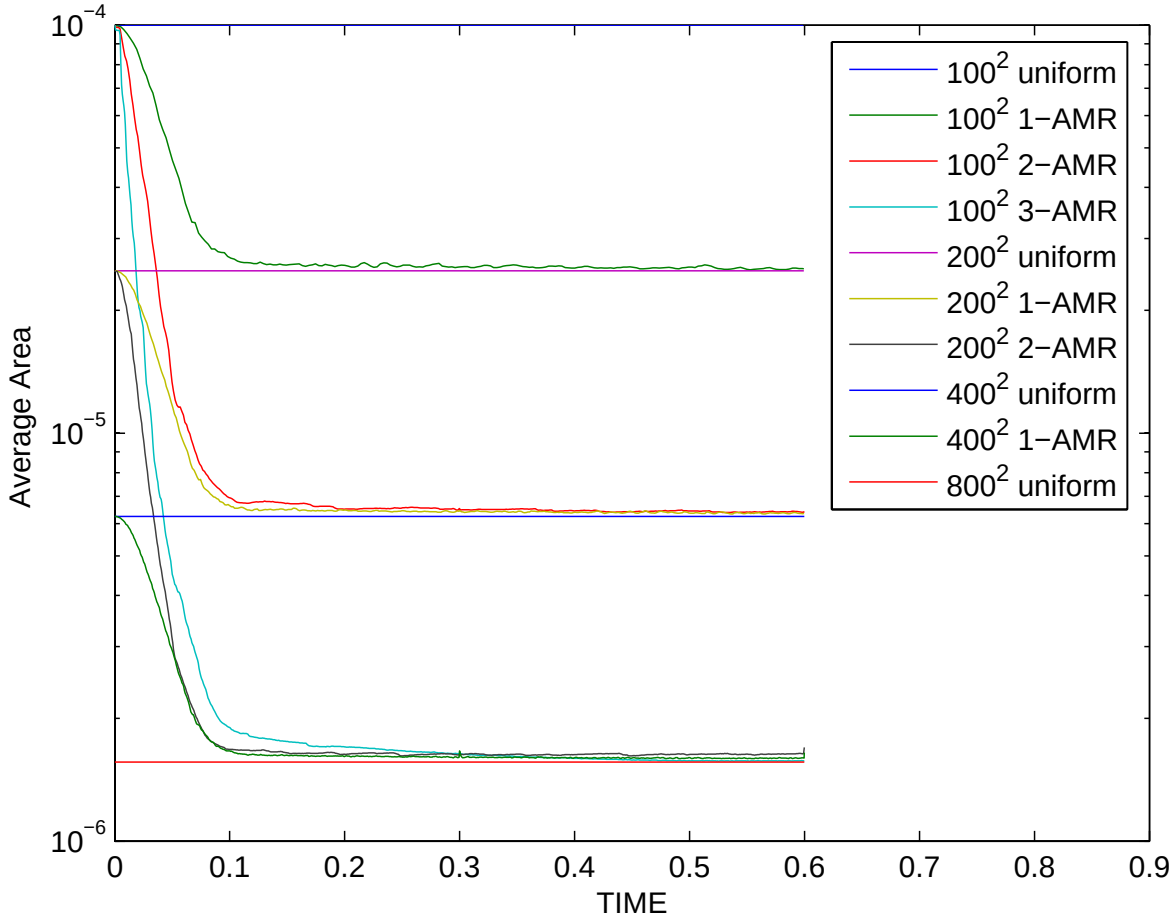

Figure 56: Average cell volume of different grids with refinement for Noh2d problem. RAGE is used.

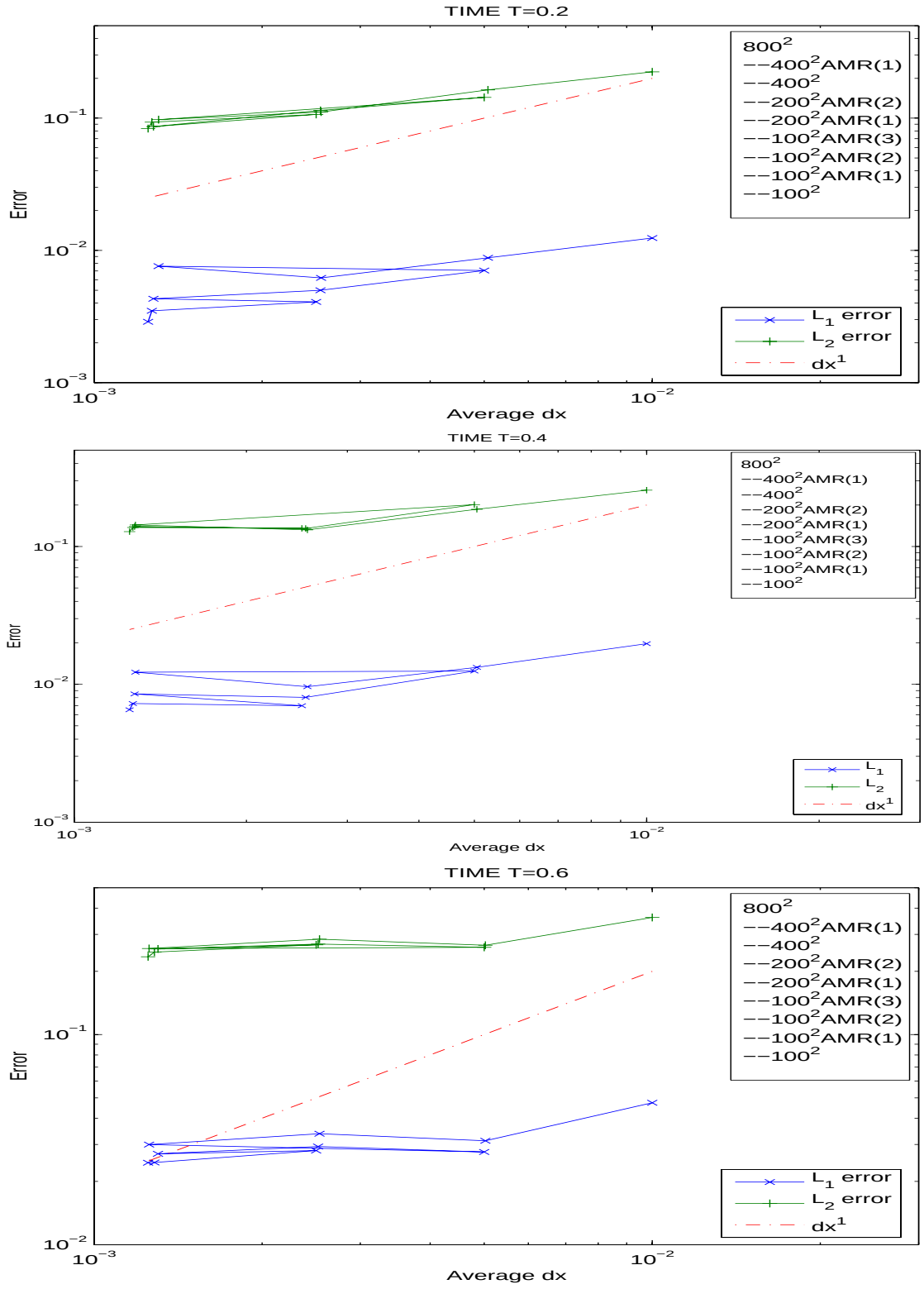

Figure 57: The error convergence behavior for different base grids and different refinement levels for Noh2d problem. RAGE is used.

5.2 Noh 3D problem solved in (r,z) cylindrical grid

For comparison with Gisler's results [9], we also solved the Noh 3D problem in (r,z) cylindrical grid. The grid selection and set-up is the same as Noh 2D example. We used the freeze regions to handle the reflection boundary conditions. The freeze regions not only can reduce the number of cells, it also eliminate the effect of the reflection from the boundaries. Figure 58 shows the difference in L_1 error between our set-up and Gisler's set-up.

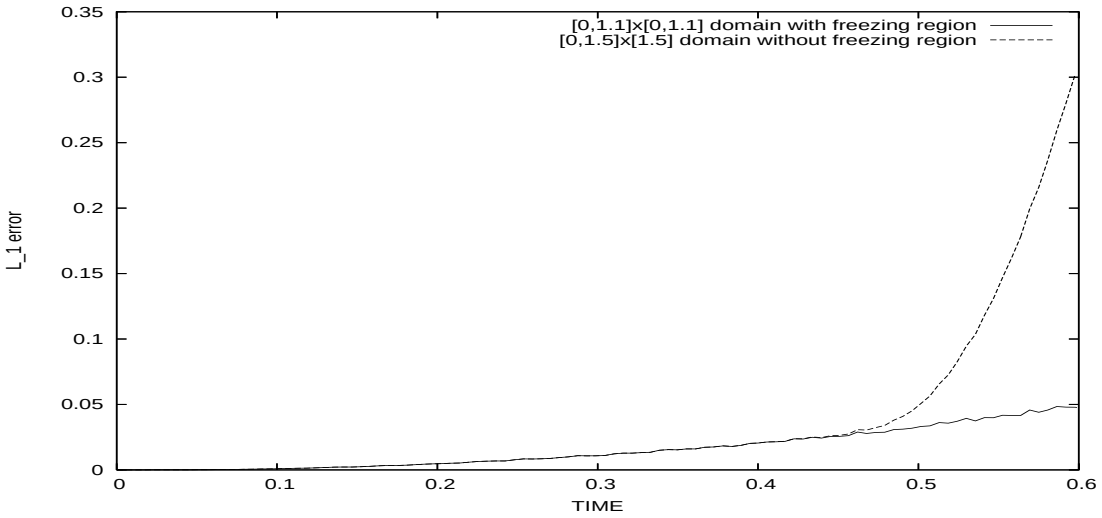

Figure 58: L_1 error in different treatment of the boundary conditions for Noh3d problem. RAGE is used.

Although our initial grid set-up is different from Gisler [9], we obtained almost the same convergence results for uniform and AMR simulations. Figure 59 shows the convergence results of L_1 error for the uniform grid. Figure 60 shows the convergence of AMR with different refinement levels for 100×100 base grid. It shows that AMR is not converged at all. Compared our convergence results with those in Gisler [9], it seems that our convergence is better than the results in [9]. We think the reason is that in [9] all of the errors are calculated based on 400×400 HDF output from the Rage, while we calculated the error for each original cell.

As reported in Gisler [9], the three-level refinement of 100×100 grid has a surprising and catastrophic anomaly on the axis, which turns out to be a low density bubble straddling the shock. It did not happen for the Noh2d examples. It also did not happen for the same resolution refinement with different base grids. The results and error distribution is shown in Figures 61 and 62. It is also worth noting that the anomaly also occurs for the other base grids (e.g. 50×50 and 200×200) with 3-level refinement.

Figure 63 shows the overall L_1 and L_2 error in time history for grid with only 1-level refinement. As in Noh2d example, in term of the L_2 error, the AMR achieves almost the same accuracy as the finest uniform grids at early time. This is probably because the error is dominated in post-shock region.

Figure 64 shows the results of the average cell-volume for different AMR grid. Compared with Figure 56, it takes even shorter time for the AMR to refine almost everywhere. To

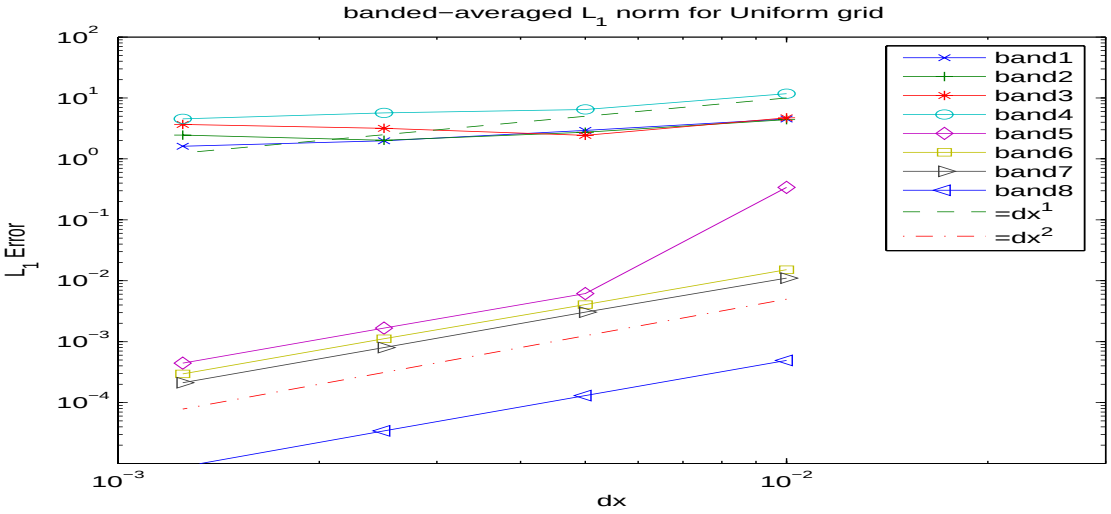

Figure 59: L_1 error convergence for uniform grid for Noh3d problem.

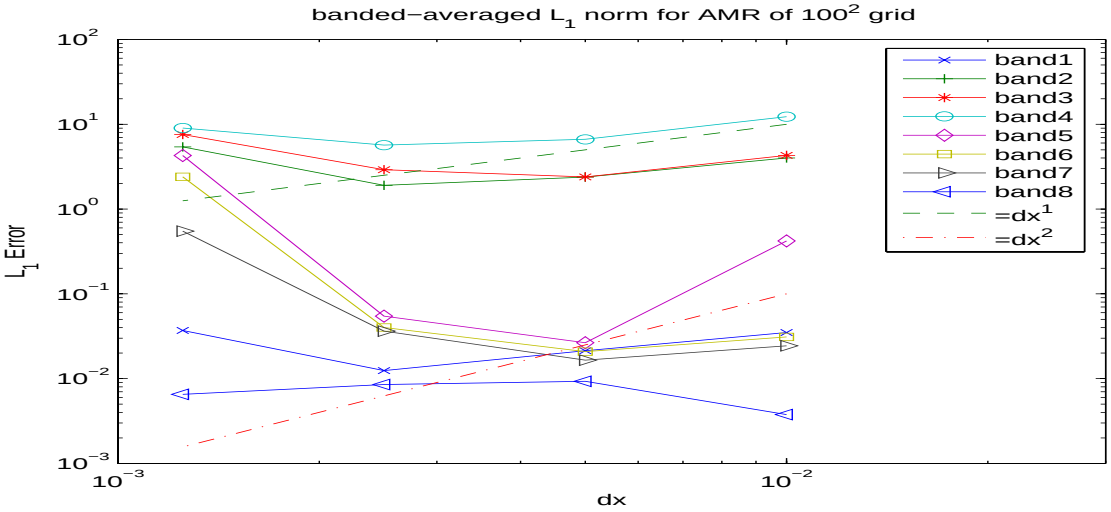

Figure 60: L_1 error convergence for AMR of 100×100 grid for Noh3d problem. RAGE is used.

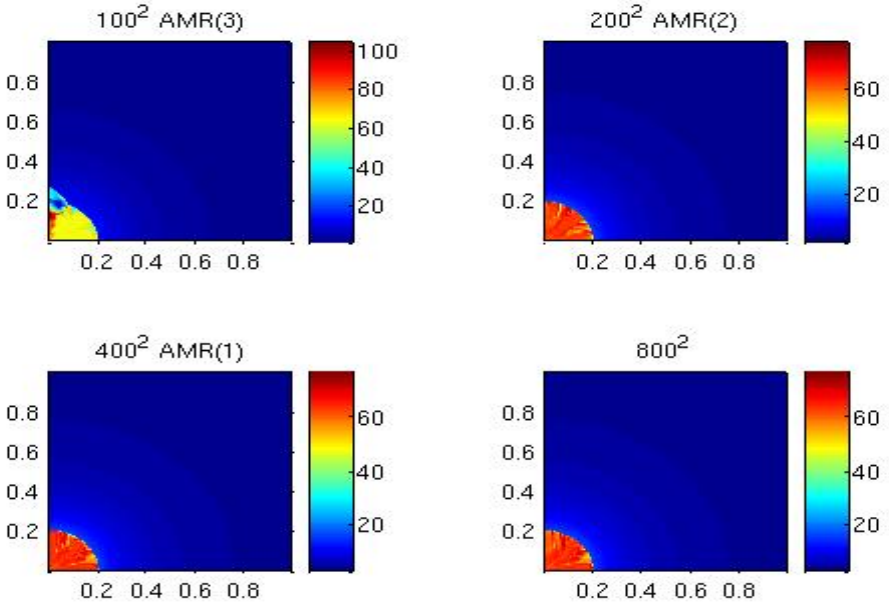

Figure 61: Density plot for different refinement of the same resolutions for Noh3d problem. RAGE is used.

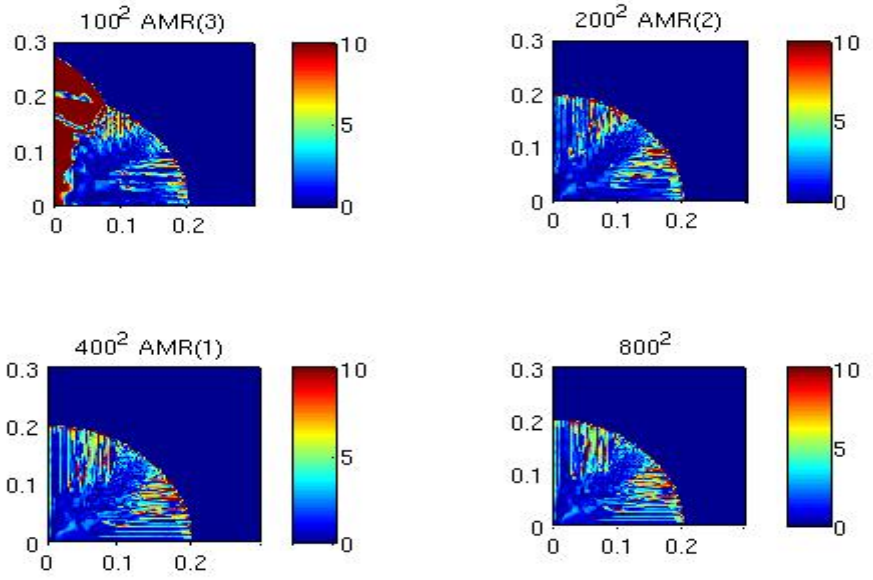

Figure 62: Error distribution for different refinement of the same resolutions for Noh3d problem. RAGE is used.

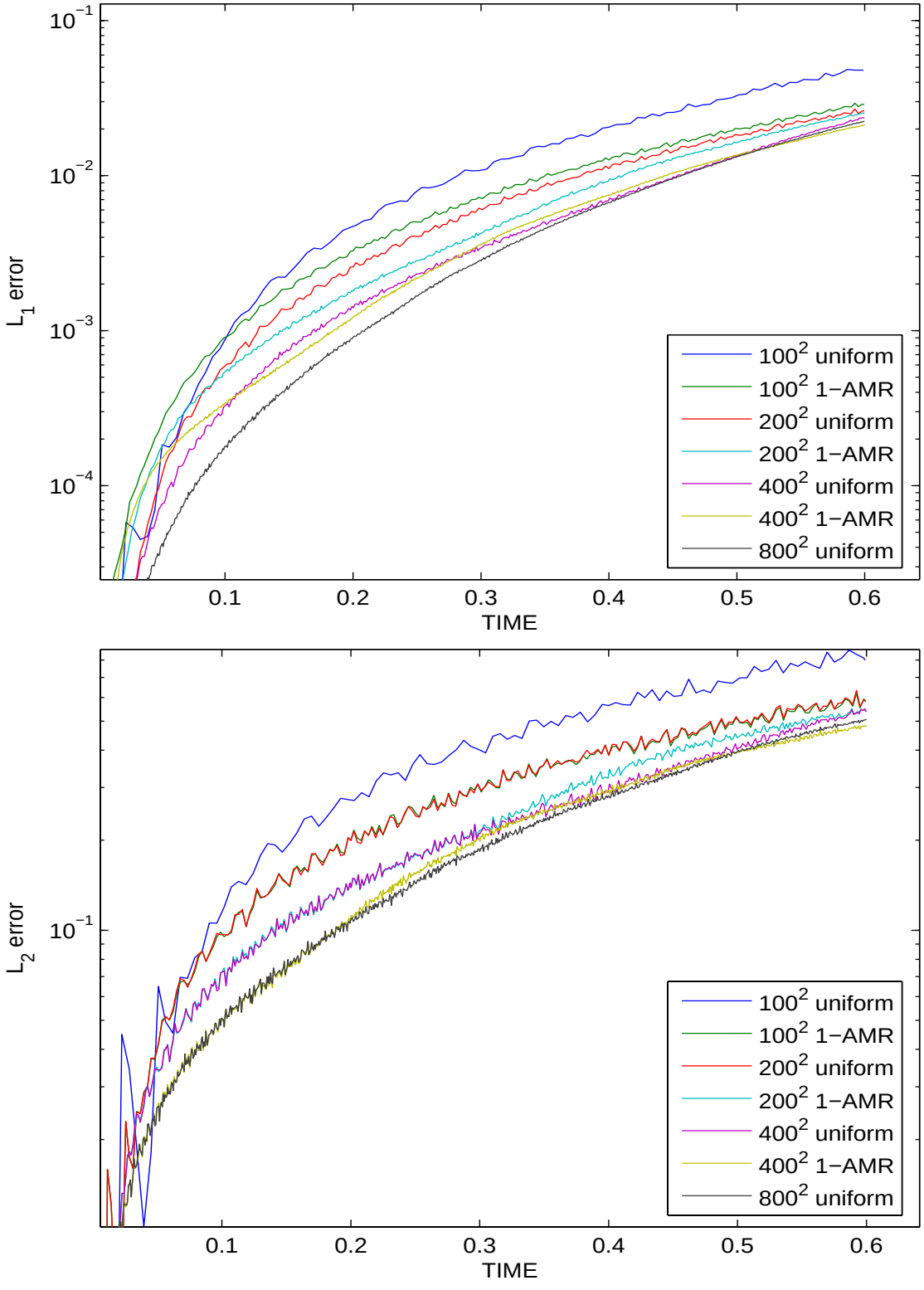

Figure 63: L_1 and L_2 errors in time history for Noh3d problem. RAGE is used.

show where the refinement happens, we plot the refinement at $t = 0.6$ for 100×100 grid with 3-level refinement in Figure 65.

5.3 Study of the carbuncle phenomenon

As suggested in [9], the anomaly behavior for 3-level refinement is probably related to *carbuncle phenomenon*. In this section, we re-investigate this problem with the AMR package (AMR-MHD). We tested it with base grid 50×50. As for RAGE code, a low density bubble (see Figure 66) straddles the shock near the Z-axes. Gisler pointed out the problem must arises during de-refinement. To test this, we fixed the refinement near the Z-axes during the simulation in AMR-MHD. The refinement region [0,0.1]×[0,0.3] is fixed all the time for all refinement levels. There is no refinement or de-refinement in this fixed region during the simulation. Figure 66 shows the carbuncle anomaly still occurs. This indicates that the reason may not be due to the de-refinement during the simulation.

If AMR works normally, the AMR that refines everywhere should be identical to the corresponding uniform grid results. This has been verified with our AMR-MHD package, since we can force AMR to refine everywhere. The next question we try to answer is why the carbuncle anomaly occurs only near the Z-axes. We have forced a refinement band [0,0.1]×[0,1] along Z-axes, and found a surprising result. The carbuncle anomaly disappears (see Figure 67). However, a refraction wave appears in the forced refinement band outside shock. This can be vividly seen from the contour plot in Figure 67. The fluctuation is due to the refraction at the coarse-fine interface, because there is no such thing if we refine everywhere. We also observed that the more refinement level, the stronger the fluctuation. This fluctuation can trigger the carbuncle phenomenon for many Riemann solver (Quirk 1994).

Quirk pointed out the the carbuncle phenomenon can be suppressed by a carbuncle free Riemann solver. We tested it with the one-state HLLE Riemann solver and the results are shown in Figure 68. We also tested the adaptive Riemann solver as suggested in Quirk [16]: in shock region the HLLE Riemann solver is used, otherwise, the Roe's Riemann solver is used. We have used two different shock detection: one is based on pressure and the other is based on the strong compression. For both approaches, we obtained almost the same results as with the one-state HLLE solver. The results are shown in Fig. 69. However we still observed some high overshot near the intersection of shock and $r = 0$.

We also tried to vary the size of the forced refinement region near $r = 0$ regions. We first change the size in Z-direction. We have seen one of the results when $z = 0.3$ in Figure 66. Figure 70 shows the other variations ($z = 0.6$ and $z = 0.8$).

We also vary the size of the forced refinement by changing the length in r-direction. Figure 71 shows the results when $r = 0.05$ and $r = 0.01$. We can see that $r = 0.05$ result is better than that of $r = 0.01$.

One research issue related to this problem is that why the refraction at the coarse-fine interface for this problem is so strong near the Z-axes and can be neglected in other regions or for other problems. Is there any method to detect this? We will address these issues in the future research.

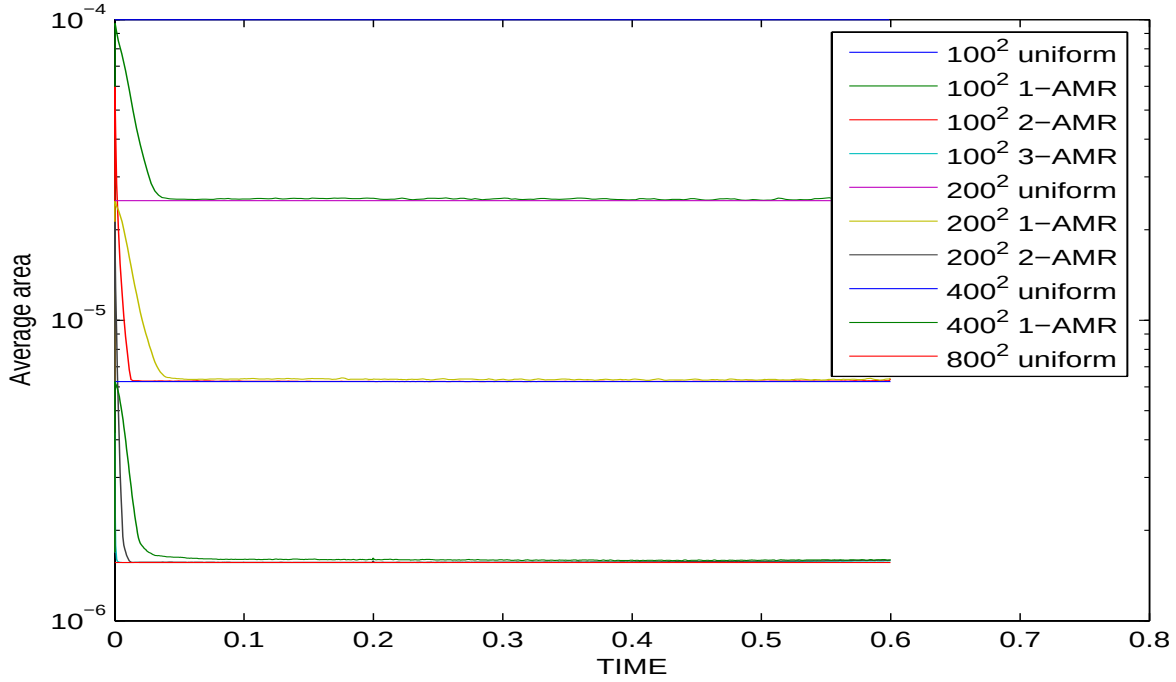

Figure 64: Average cell volume of different grids with refinement for Noh3d problem. RAGE is used.

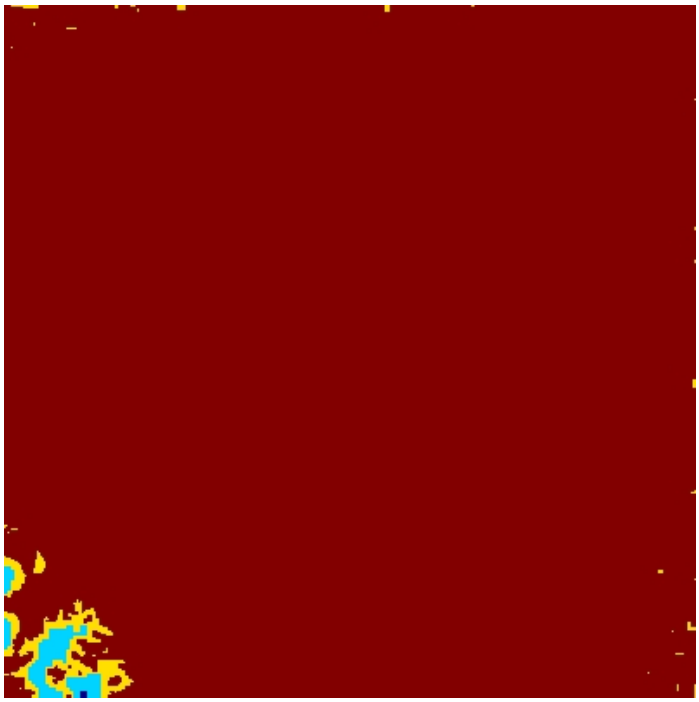

Figure 65: AMR Refinement at $t = 0.6$ for 100×100 grid with 3-level refinement. Refinement levels: 1st level(blue), 2nd level(light blue), 3rd level(yellow), 4th level(red). RAGE is used.

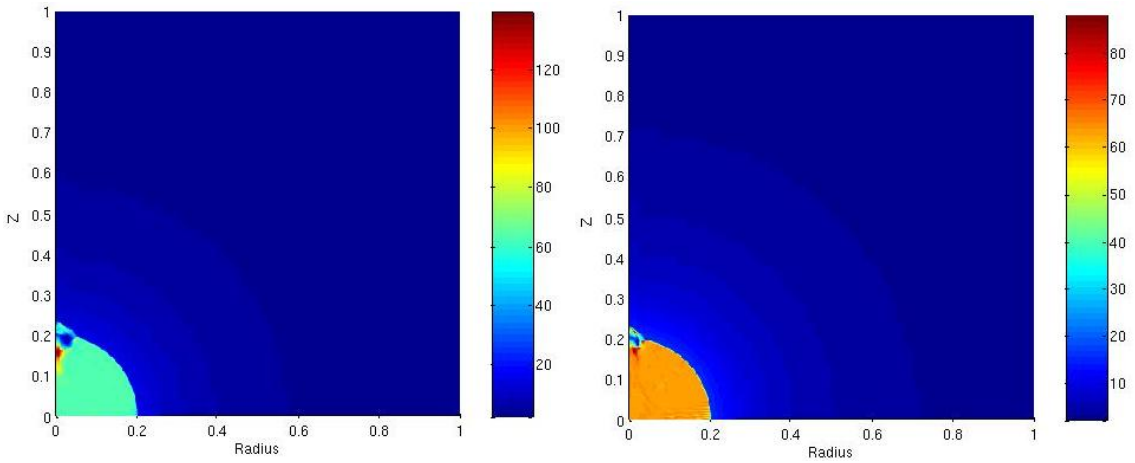

Figure 66: "Carbuncle phenomenon" for 3-level AMR results of Noh3d problem solved by AMR-MHD package. Left: AMR(3); Right: AMR(3) with fixed refinement near $r = 0$. AMR-MHD is used.

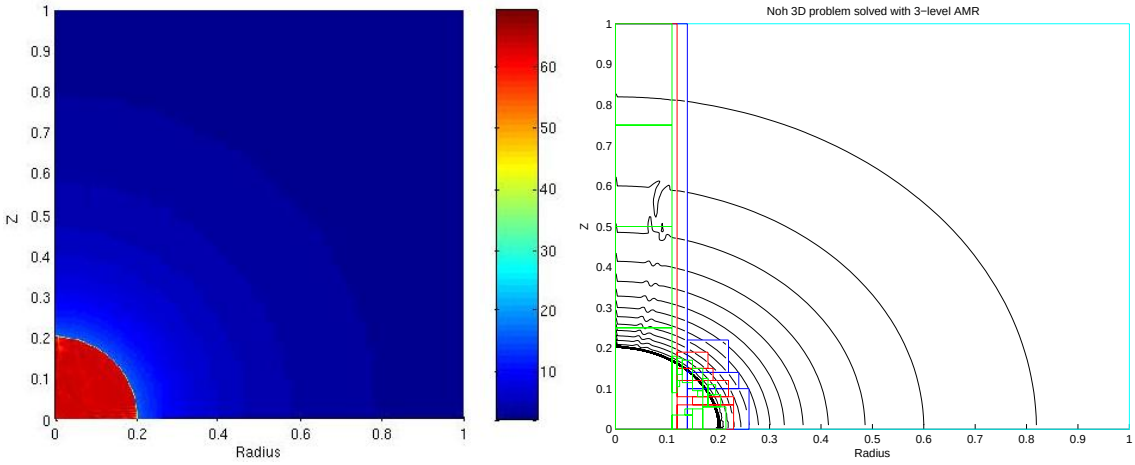

Figure 67: Carbuncle disappears for 3-level AMR results of Noh3d problem solved by AMR-MHD package for fixed refinement along $r = 0$. Instead a refraction wave appears in the forced refinement band. AMR-MHD is used.

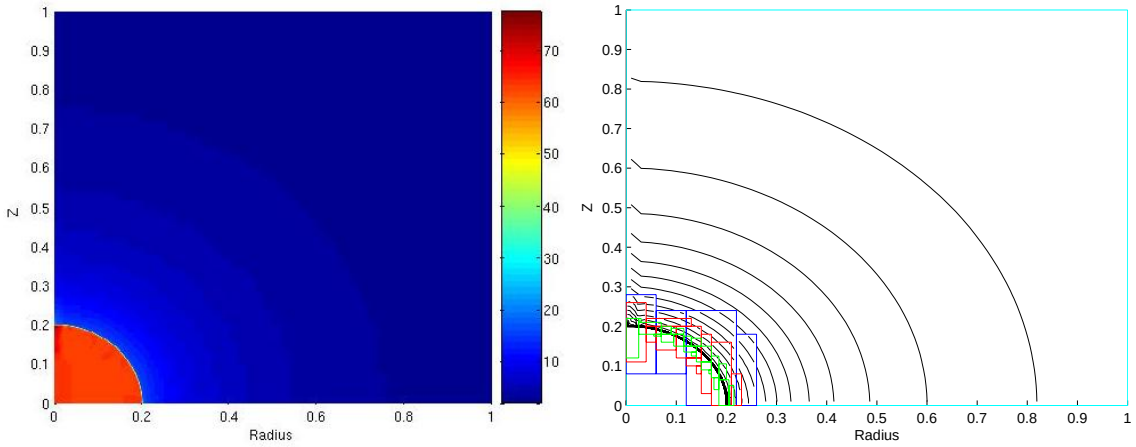

Figure 68: Carbuncle disappears for HLLE Riemann solver. AMR-MHD is used.

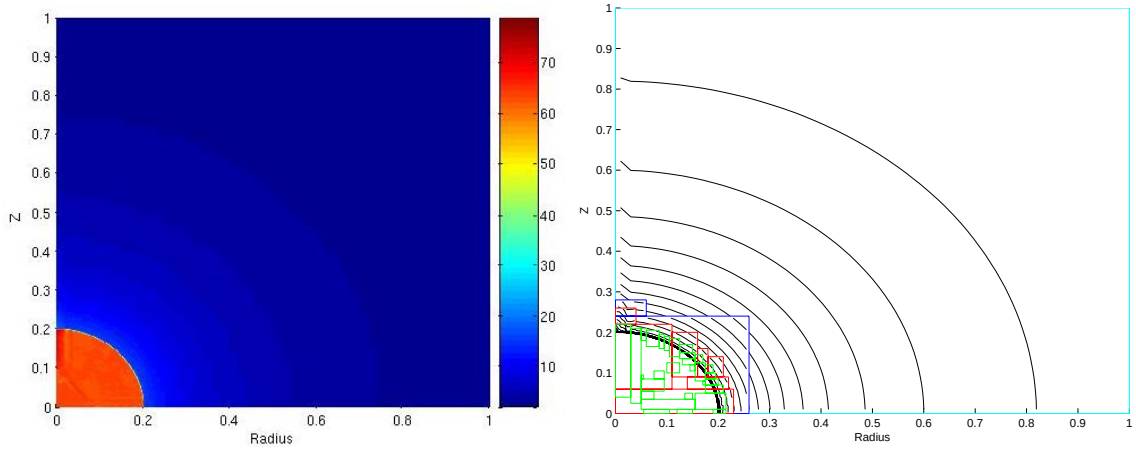

Figure 69: Carbuncle disappears for adaptive Riemann solver. AMR-MHD is used.

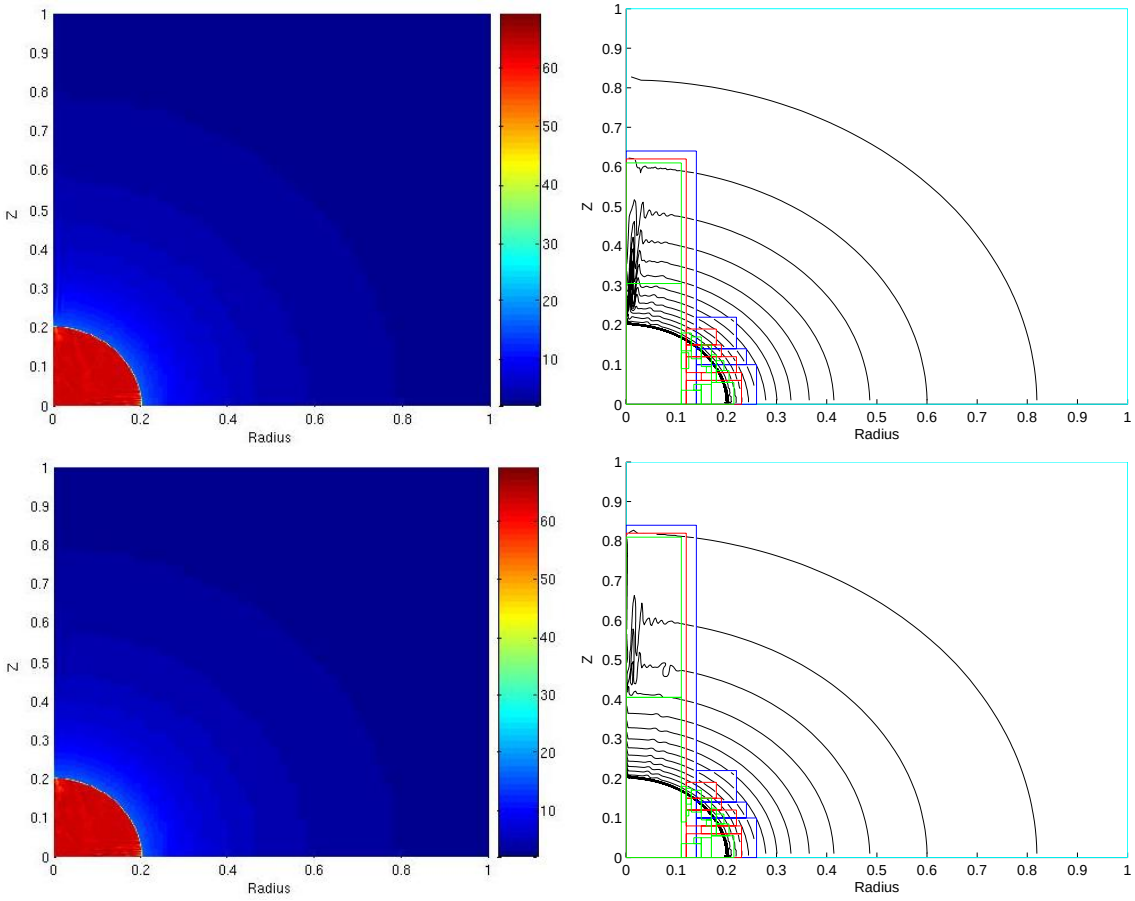

Figure 70: Carbuncle anomaly changes with the forced refinement. AMR-MHD is used.

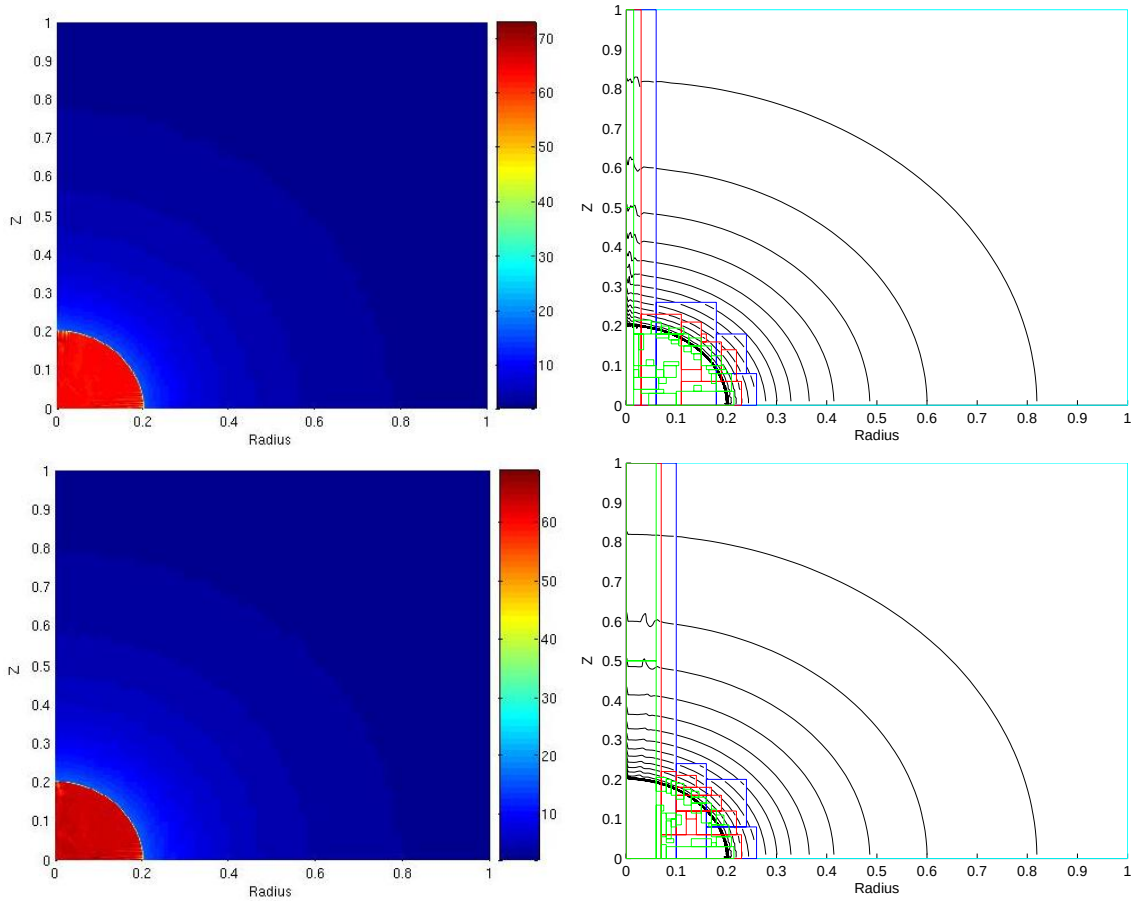

Figure 71: Carbuncle anomaly changes with the forced refinement. AMR-MHD is used.

6 Conclusions

In this report, we have considered three 2-D simulations of smooth and discontinuous problems for convergence analysis of AMR calculations. Two different AMR packages are used and compared: the patch-based AMR-MHD and the cell-based RAGE code. Since they use different hydro algorithms as their basic solver, the comparison cannot tell which AMR package is better. The data shown in this report can be taken only as a reference, *NOT* as evidence to compare these two different AMR packages.

For AMR-MHD, the results shows that AMR works as expected for the linear wave and Noh 2D problem. It achieves the accuracy of finest resolutions. This is especially true for L_∞ error norm. However for the vortex problem, the AMR results with locked time step are worse than those of uniform grid without refinement at later stage of the time integration.

We remark that the error tolerance TOLS does have impact on the convergence behavior of AMR, especially for the locked time step method. On one hand, if we refine almost everywhere as TOLS$\rightarrow 0$, then both the local time step and locked time step achieves the same accuracy as the finest uniform grid results. On the other hand, if we refine only a few cells as TOLS$\rightarrow \infty$, then both the local time step and locked time step achieves only the coarse grid accuracy with different CFL numbers. For the later case, the locked time step may be worse than the uniform grid results because it corresponds to the coarse uniform grid results with a smaller CFL number. If there is no refinement as TOLS$\rightarrow \infty$, both time stepping method will have the same results.

We are interested in the range of TOLS where the local time step is much better than the locked time step, and also the area of large error is covered by the refined grid. A crucial issue needed to be answered is that why the small error in the coarse region could trigger large error in refined region. The phase (dispersion) error may play an important factor. We will continue our research on this topic.

AMR has another problem when applied to (r, z) cylindrical coordinate. It can trigger the carbuncle instability along the z-axes if the solution has a strong shock wave. Numerical results show the fluctuation that generates at the coarse-fine interface may be amplified by more refinement levels and then trigger the carbuncle anomaly at the shock front. One research issue is where the fluctuation comes from and how to suppress and/or eliminate it.

The numerical results of RAGE package reveal some defects with the current AMR implementation. Two outstanding issues are the AMR initialization and AMR refinement criteria. We will address these issues in the future research.

Acknowledgements

This work was performed by Los Alamos National Laboratory, which is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. The authors thank many individuals for their generous time and effort towards this effort include Jerry S. Brock, James R. Kamm, John W. Grove, Edward D. Dendy, Billy J. Archer, and Robert B. Lowrie.

References

[1] M. J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984), 484-512.

[2] M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989), 64-84.

[3] J.M. Hyman, and S. Li, Interactive and dynamic control of adaptive mesh refinement with nested hierarchical grids, Los Alamos National Laboratory Report (1998).

[4] J. M. Hyman and S. Li, Solution Adapted Nested Grid Refinement for 2-D PDEs, Los Alamos National Lab. Report, LA-UR-98-5463 (1998).

[5] R. Liska and B. Wendroff, Comparison of several difference schemes on 1d and 2d test problems for the Euler equations. Technical Report LA-UR-01-6225, Los Alamos National Lab, 2001.

[6] Shengtai Li, Spatial Convergence Analysis for a 2-D Adaptive Mesh Refinement Code. Technical Report LA-UR-05-3078, Los Alamos National Lab, 2005.

[7] S. Li and H. Li, A Modern Code for Solving Magneto-hydrodynamic or Hydrodynamic Equations, Technical Report *LA-UR-03-8925*, Los Alamos National Lab, 2003.

[8] M. Clover, The "new hydro" coding in RAGE, 2002, SAIC report.

[9] Galen Gisler, Two-dimensional convergence study of the Noh and Sedov problems with RAGE: Uniform an Adaptive grids, Technical Report LA-UR-05-xxxx, Los Alamos National Lab, 2005.

[10] J. R. Kamm and W. J. Rider, 2-D convergence analysis of the RAGE hydro-code. Technical Report LA-UR-98-3972, Los Alamos National Lab, 1998.

[11] J. R. Kamm and W. J. Rider, Verification analysis of the direct Eulerian 2-D RAGE hydrodynamics algorithm, Technical Report LA-UR-99-5234, Los Alamos National Lab, 2000.

[12] J. R. Kamm and W. J. Rider, Spatial convergence analysis of 1-D RAGE on the spherically symmetric Noh Problem, Technical Report LA-UR-00-xxxx, Los Alamos National Lab, 2000.

[13] J. R. Kamm and W. J. Rider, Consistent metric for code verification, Technical Report LA-UR-02-3794, Los Alamos National Lab, 2002.

[14] J. R. Kamm, W. J. Rider and J. S. Brock, Combined space and time convergence analysis of a compressible flow algorithm, *AIAA-2003-4241*, 2003.

[15] J. R. Kamm, J. S. Brock, C. L. Rousculp, and W. J. Rider, Verification of an ASCI SHAVANO project hydrodynamics algorithm, Technical Report LA-UR-03-6999, Los Alamos National Lab, 2003.

[16] J. J. Quirk, A contribution to the great Riemann solver debate, *Int. J. Num. Meth. Fluids*, 18 (1994), 555-574.