INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **48**:541–564
Published online 4 March 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.948

A model study for the steady state error of numerical approximations of inviscid flow equations on Cartesian grid

Zi-Niu Wu*,†

Department of Engineering Mechanics, Tsinghua University, Beijing 100084, People's Republic of China

SUMMARY

The widely used locally adaptive Cartesian grid methods involve a series of abruptly refined interfaces. In this paper we consider the influence of the refined interfaces on the steady state errors for second-order three-point difference approximations of flow equations. Since the various characteristic components of the Euler equations should behave similarly on such grids with regard to refinement-induced errors, it is sufficient enough to conduct the analysis on a scalar model problem. The error we consider is a global error, different to local truncation error, and reflects the interaction between multiple interfaces. The steady state error will be compared to the errors on smooth refinement grids and on uniform grids. The conclusion seems to support the numerical findings of Yamaleev and Carpenter (*J. Comput. Phys.* 2002; **181**:280–316) that refinement does not necessarily reduce the numerical error. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: Cartesian grid; multiple interfaces; steady state error; three-point difference approximation

1. INTRODUCTION

The locally adaptive Cartesian grid method is now common in Computational Fluid Dynamics [1–12]. In this method, the entire grid is composed of divided zones (which will be called subgrids for convenience) each having a uniform mesh size and with abrupt mesh refinement at the interfaces. In a conventional Cartesian grid method, a cell is equally divided in both directions so that a cell is split into four subcells once a refinement is needed. This is called anisotropic Cartesian grid. It can also be divided into two based on refinement only in one

*Correspondence to: Zi-Niu Wu, Department of Engineering Mechanics, Tsinghua University, Beijing 100084, People's Republic of China.
†E-mail: ziniuwu@tsinghua.edu.cn

Contract/grant sponsor: NSF; contract/grant numbers: 10472056, 10376016, 10025210
Contract/grant sponsor: NKBRSF; contract/grant number: 2001CB409600

Received 11 December 2003
Revised 11 August 2004
Accepted 1 October 2004

direction (anisotropic refinement, see References [12, 13]). More recently, a nonet Cartesian grid method [14] is proposed for which a cell can be divided into nine subcells or six subcells so that a very strong refinement ratio exists.

The particular feature of the Cartesian grid method is the existence of multiple refinement interfaces which are separated by subgrids of uniform mesh size, see Figure 1 for a typical adaptive Cartesian grid. With the well-established stability and convergence theory, the solution behaviour can be easily predicted for the case of a single grid interface. For instance, with a second-order accurate difference scheme in each subgrid, and a first-order accurate (locally second-order accurate) interface treatment, the global accuracy is still second order according to the convergence theorem of Gustafsson [15]. The stability of a single mesh refinement interface, independent of the refinement degree, has been studied long before, see for instance References [2, 16–18]. Little attention has been paid to the theoretical questions (accuracy, stability, convergence, uniqueness) of a problem with a series of interface. Besides, a standard approach for global error analysis of such a problem is not readily available. One generally believes that an adaptive Cartesian grid method has some advantages: it is fast for grid generation, and accurate due to the local grid regularity. It is not yet known how the existence of multiple interfaces changes the accuracy. In Reference [19], we have studied the problem of mesh refinement induced dissipation which is closely but not exactly related to stability. For a general semi-discrete three-point difference approximation with uneven mesh spacing, if the wave moves in the fine-to-coarse direction then the dissipation is positive (stabilizing), and if the wave moves in the coarse-to-fine direction then the dissipation is negative (de-stabilizing). Moreover, the amount of dissipation is insensitive to the subgrid width if the total refinement degree is fixed. For the fully discrete Lax–Wendroff scheme, the lower bound of the stability region is increased by mesh refinement, while the upper bound is reduced.

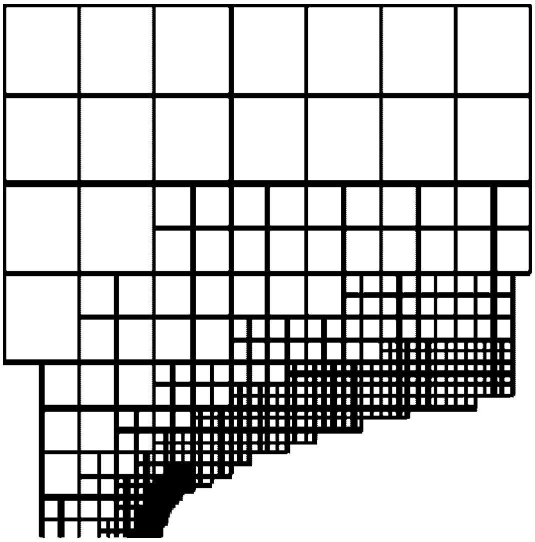

Figure 1. Adaptive Cartesian grid near an airfoil.

A theoretical analysis for the error caused by the existence of a series of abrupt interfaces still lacks. But numerical study of errors on such types of grid has been studied previously, see Reference [20] and the references cited there. Yamaleev and Carpenter have discussed the accuracy of adaptive grid methods, the smooth refinement method and the abrupt refinement method, for shock wave computation using various schemes. The grid refinement study shows that for a second-order scheme, neither grid adaptation strategy improves the numerical solution accuracy compared to that calculated on a uniform grid with the same number of grid points. For a fourth-order scheme, the dominant first-order error component is reduced by the grid adaptation, while the design-order error component drastically increases because of the grid non-uniformity. As a result, both grid adaptation techniques improve the numerical solution accuracy only on the coarsest mesh or on very fine grids that are seldom found in practical applications because of the computational cost involved.

In this paper we will consider the difference approximation of a flow equations on a Cartesian grid and address the question of steady state errors caused by multiple interfaces. In order to drop out the unnecessary complexity, we will perform the analysis just by considering a representative model equation on a grid with a series of interfaces. The model is just a scalar hyperbolic equation. The conclusions hold for the case of system of equations as resulted from gas dynamics, since the problem can be diagonalized and the various scalar components should behave similarly on the same grid. In this paper we do not consider the dicretization errors of the viscous dissipation terms on Cartesian grids. The derivation of the steady state errors is rather straightforward and does not involve complex algebra as commonly seen in modern numerical analysis. Hence the paper is more suitable for the community of CFD than those in the field of numerical analysis.

The model problem suitable for analysis is presented in Section 2. In Section 3, we study the numerical error for a second-order non-dissipative scheme, in which the influence of the subgrid width is analysed. In Section 4, we consider the case of the Lax–Wendroff scheme.

Section 5 is devoted to a general three-point scheme with standard treatment of the source term of the exact equation. The case of constant numerical viscosity or variable numerical viscosity, which corresponds to local time-stepping or uniform time-stepping for the Lax–Wendroff scheme, is investigated. In Section 5, we also consider the difference between conservative treatment and non-conservative treatment and the numerical error for mesh refinement problem is compared with the error on a uniform grid.

In Section 6, we will demonstrate some numerical results in order to check some complicated analytical results and then make some conclusions.

2. STATEMENT AND DEFINITION OF THE PROBLEM

A common feature of the adaptive Cartesian grid method is that it uses an isotropic refinement strategy. The grid generation starts with a background Cartesian grid with square or rectangular cells. The final grid is generated by the recursive subdivision of a single cell (parent) into four equal cells (children) when and where it is necessary. The hierarchical relation between the children and parents is stored and used for cell neighbour searching. The resulting grid thus involves cells of various sizes. A subgrid of level l with $l=0,1,\dots,L$ refers to all the cells having the same mesh size h_l. The mesh size satisfies the relation $h_l=r^l h_0$ where r, typically with $r=\frac{1}{2}$, is the refinement ratio and h_0 is the mesh size on the coarse grid. Since a cell is refined equally in both directions independently of the geometry and the flow gradient, the resulting Cartesian grid is isotropic. There are two types of Cartesian grid method: one is the conventional isotropic Cartesian grid method [8, 11] and the other is the anisotropic Cartesian grid method proposed by the present author [12, 13]. Figure 2 is the anisotropic counter part of the grid for Figure 1.

There are also situations for which one would require the grid refinement to be more rapid than the quadratic method as presented above. Figure 3 is an example. This is a transonic

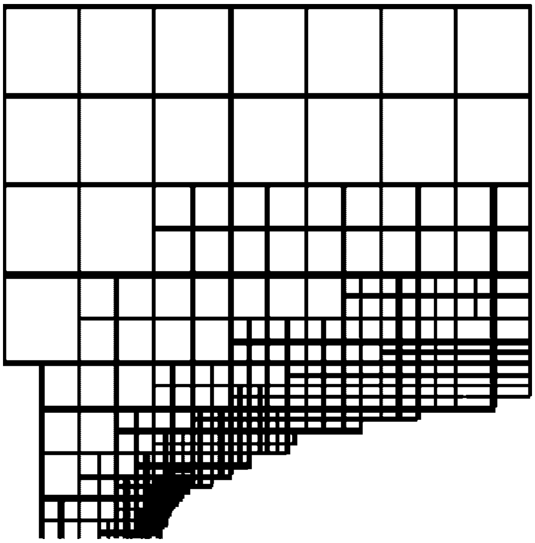

Figure 2. Anisotropic Cartesian grid for the same geometry as for Figure 1.

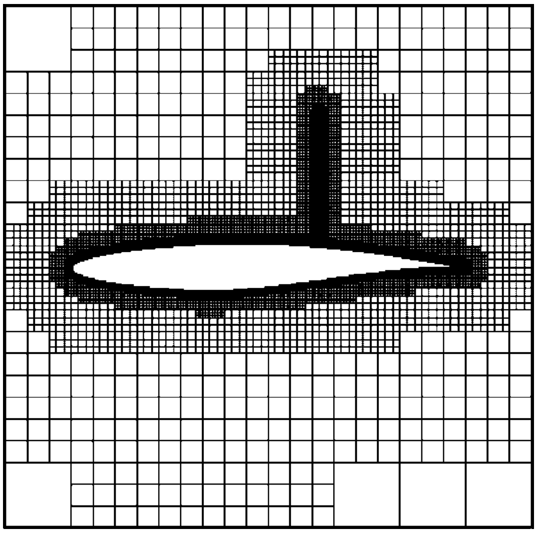

Figure 3. Nonet grid for a NACA2822 airfoil.

flow around a RAE2822 airfoil, the Mach number is 0.729 and the angle of attack is 2.31°. In this case there is a straight shock above the airfoil. In order to resolve the shock without adding too many unnecessary grid points elsewhere, an alternative approach is to use nonet grid, for which a cell would be divided into nine subcells in the istropic approach or six cells in the anisotropic approach (see Figure 4 for an example).

The use of nonet grid would lead to very fine shock structure (see Figure 5) while keeping the number of grid points to a reasonable value.

An immediate question is that, while a strong refinement can efficiently resolve discontinuities, how the cell interfaces, with a refinement ratio equal to $\frac{1}{2}$ for the quadratic approach and $\frac{1}{3}$ for the nonet approach, affects the accuracy in the smooth flow regions? For a smooth transition of the grid, the refinement has to cross the smooth flow regions.

It is impossible to perform a rigorous analysis in the complex 2D or even 3D case. Instead, a one dimension study for a model equation would be more instructive. This is why we want to build a one-dimensional equivalent.

Now we describe the one-dimensional equivalent of the Cartesian grid method suitable for error analysis. The grid system is displayed in Figure 6, the entire grid is composed of a certain number of subgrids G_l of different levels separated by interfaces. Let h_l be the mesh size of level l with $0 \leqslant l \leqslant L$. The interface separating the two adjacent levels l and $l+1$ will be called interface $I_{l+1/2}$. Usually in a Cartesian grid method the number of grid points normal to the interface in each level is almost a constant and that the refinement ratio at subgrid interfaces is a constant. We will assume $h_l = h_0 r^l$ where r with $r<1$ is a constant independent of l. Additionally, the number of grid points in each subgrid is constant and equal to p. The integer p will also be called the subgrid width.

For convenience, let us define the total refinement degree by $r_T = h_L/h_0$. The local refinement degree $r = h_{l+1}/h_l$ is related to r_T by $r = r_T^{1/L}$. To be more general, $1/r$ is not necessarily

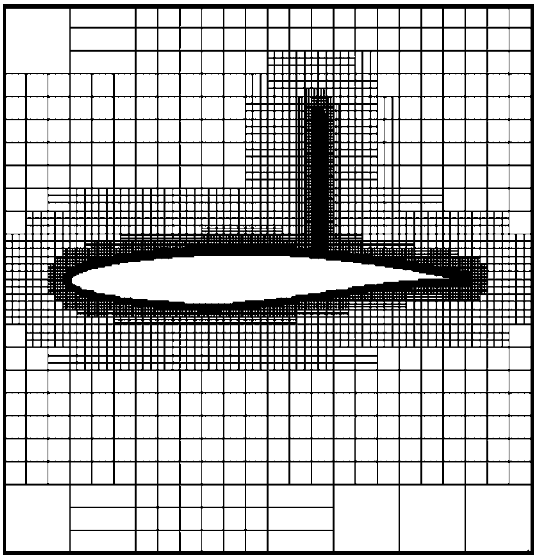

Figure 4. Anisotropic nonet grid for a NACA2822 airfoil.

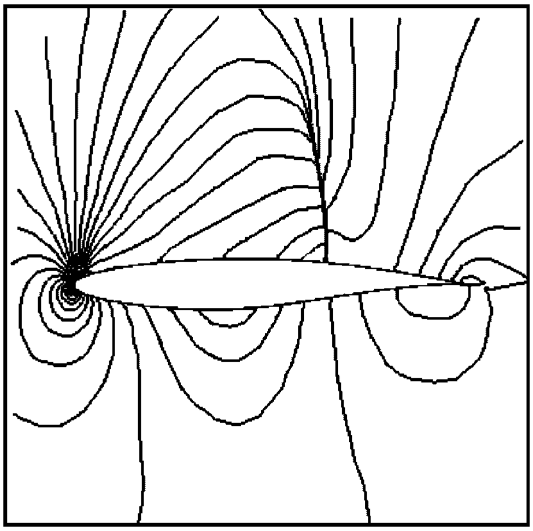

Figure 5. Mach contours around the RAE2822 airfoil. The use of nonet grid leads to very fine shock structure while keeping the number of grid points to a reasonable value.

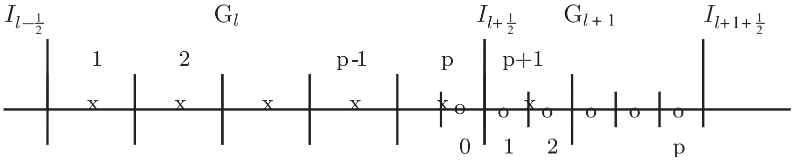

Figure 6. One-dimensional Cartesian grid with a series of interfaces and with each subgrid having the same width.

an integer in the subsequent analysis, though in the adaptive Cartesian grid method, one takes $r=\frac{1}{2}$ without exception. Precisely, we will let r take all values in the interval (0,1). For instance, if $p=1$ and $r\rightarrow 1$, then we obtain a smooth refinement method. This allows us to compare the accuracy of the adaptive Cartesian grid method with that of a smooth refinement method. If we take $r<\frac{1}{2}$, one can see what would happen if the grid is over-refined. For instance, the case with $r=\frac{1}{3}$ corresponds to a Cartesian grid in which a cell is divided into three in each direction. According to the knowledge of the author, no one has yet used such a Cartesian grid.

In varying the parameter p and r, we must keep the total refinement degree r_T to be fixed. This is a practically useful constraint. In computing flow problems, the largest mesh size is given, which may depend on the far flow field. The smallest mesh size depends on the finest local flow structure, as in the boundary layers. Our question is: keeping the total refinement degree fixed, which choice of r gives the best accuracy, smooth variation ($r\rightarrow 1$) or abrupt refinement ($r=\frac{1}{2},\frac{1}{3},\ldots$)?

For convenience, we define an averaged refinement degree by $r_0 = r^{1/p} = r_T^{1/pL}$. It means that, if the local refinement degree for each mesh point is r_0, then the total refinement degree is r_T $(= r_0^{pL})$.

In order to perform a detailed analysis of the influence of the multiple interfaces on the numerical error, we will consider the following scalar equation:

$$u_t + u_x = s(x), \quad 0 < x < 1 \tag{1}$$

where $s(x)$ is a given function.

The reason of choosing the simplified model is that: we are just interested in whether a second-order accurate difference approximation has a global second-order accuracy under the influence of a series interfaces. Moreover, we want to know whether the Cartesian grid method remains as accurate as a conventional grid refinement method such as the smooth refinement method. More generally, for numerical analysis (of accuracy, stability, TVD behaviour, entropy condition, etc.) in CFD problems, it is very common to perform the analysis on a scalar equation. Then the problem becomes tractable, and the results remain meaningful since the original system of equations is almost diagonalizable (it is indeed diagonalizable for the Euler equations in gas dynamics). Another important reason is that, in the present problem a series of interfaces exist so that it is very difficult to consider the full system of Navier–Stokes equations for detailed theoretical analysis.

The steady state numerical solution is often achieved through the iteration of a time-dependent difference approximation. When the numerical solution converges in the sense that it no longer varies in time (in some numerical sense), then we consider the numerical solution to be the steady state one.

On the Cartesian grid including the particular case of smooth refinement with $p = 1$ and $r \to 1$, let the numerical solution at grid j of subgrid l be $u_{l,j}^n$. A three-point difference approximation can be generally written in the following viscous form:

$$\begin{aligned} u_{l,j}^{n+1} = u_{l,j}^n &- \frac{1}{2}\lambda_l(u_{l,j+1}^n - u_{l,j-1}^n) + \frac{1}{2}\lambda_l Q_l^{(\text{num})}(u_{l,j+1}^n - 2u_{l,j}^n + u_{l,j-1}^n) \\ &+ k_l s_{l,j} - \frac{q}{2}k_l^2 (s_x)_{l,j} \end{aligned} \tag{2}$$

where $l \in \mathscr{L}$ and $j \in \mathscr{J}$, with $\mathscr{L} = \{0, 1, \dots, L\}$ and $\mathscr{J} = \{1, 2, \dots, p\}$. Here λ_l denotes the ratio between the time step k_l and the mesh size h_l, $Q_l^{(\text{num})}$ is the numerical viscosity coefficient, and q is some constant (see Sections 4 and 5). A multipoint difference scheme can also be put into form (2) with the influence of the grid points outside of those of a three-point scheme factored into the viscosity coefficient, see for instance Reference [21] for the case of a scalar equation and Reference [22] for the case of a system of equations.

At steady state with $u_{l,j}^n|_{n\to\infty} = u_{l,j}$, the above difference equation reduces to

$$-\frac{\lambda_l}{2}(u_{l,j+1} - u_{l,j-1}) + \frac{\lambda_l}{2}Q_l^{(\text{num})}(u_{l,j+1} - 2u_{l,j} + u_{l,j-1}) + k_l f_{l,j} - \frac{q}{2}k_l^2(s_x)_{l,j} = 0 \tag{3}$$

The difference equation (2) includes three typical cases: second-order non-dissipative scheme (Section 3), Lax–Wendroff scheme (Section 4), and dissipative schemes with first-order treatment of the source term (Section 5).

For the interface $I_{l-1/2}$ separating the subgrids $l-1$ and l, we have two interface unknowns $u_{l-1,\,p+1}$ and $u_{l,0}$ which are used in but not provided by the difference equation in each block. In higher dimensions it is often enough to consider a linear interpolation. This, in the one-dimensional case, reduces to the well-known Browning–Kreiss–Oliger [16] interface condition:

$$\frac{u_{l-1,\,p} + u_{l-1,\,p+1}}{2} = \frac{u_{l,0} + u_{l,1}}{2}, \quad \frac{u_{l-1,\,p+1} - u_{l-1,\,p}}{h_{l-1}} = \frac{u_{l,1} - u_{l,0}}{h_l} \tag{4}$$

The above interface condition is globally first-order accurate or locally second-order accurate.

For the starting level $l=0$, we need a boundary condition at the left boundary point $j=0$. This, for the present problem, should be defined by a Dirichlet condition:

$$u_{0,0} = 0 \tag{5}$$

For the ending level $l=L$, we need a boundary condition at the right boundary point $j=p+1$. This, for the present problem, should be defined by some extrapolation. For the present study, it is sufficient enough to consider the following locally second-order accurate (globally first-order accurate) linear extrapolation condition:

$$u_{L,\,p+1} = 2u_{L,\,p} - u_{L,\,p-1} \tag{6}$$

For a Cauchy problem with only one interface and for a second-order accurate interior difference equation, a locally second-order accurate boundary treatment is sufficient to maintain the global accuracy to be second order [15]. But the present problem involves a series of interfaces.

Since we are considering at most second-order accurate difference approximations, the dominant error will be essentially due to the second-order derivative of the solution. It is thus sufficient to consider the case $u_{xx} = \text{Const}$. By (1), we have $u_{xx} = s_x$ in the case of a steady state. Thus we will assume $s = x$ so that the steady state solution is given by $u = \frac{1}{2}x^2$ with $u_{xx} = 1$. There is no difficulty to extend the present results to the case with $u = (1/k!)x^k$ and the results remain essentially the same. Since any smooth solution can be represented by a linear combination of $(1/k!)x^k$ through Taylor expansion, the present conclusion remains true for any solutions.

The numerical error at the point l and j, which we denote $e_{l,j}$, is defined as the difference between the numerical solution $u_{l,j}$ and the exact solution $u(x_{l,j})$:

$$e_{l,j} = u_{l,j} - u(x_{l,j}), \quad l \in \mathscr{L}, \quad j \in \mathscr{J}$$

Making use of (4) and $u_{xxx} = 0$, it is straightforward to show that the interface condition for the numerical error is given by

$$\frac{e_{l-1,\,p} + e_{l-1,\,p+1}}{2} = \frac{e_{l,0} + e_{l,1}}{2} + \frac{1}{8}\,h_0^2(r^{2l} - r^{2(l-1)})$$

$$\frac{e_{l-1,\,p+1} - e_{l-1,\,p}}{h_{l-1}} = \frac{e_{l,1} - e_{l,0}}{h_l}$$

which can be solved to give

$$e_{l,0}=\frac{2r}{1+r}\,e_{l-1,p}+\frac{1-r}{1+r}\,e_{l,1}+\frac{r^{2l-1}}{4}(1-r)h_0^2 \tag{7}$$

$$e_{l-1,p+1}=\frac{r-1}{1+r}\,e_{l-1,p}+\frac{2}{1+r}\,e_{l,1}-\frac{r^{2(l-1)}}{4}(1-r)h_0^2 \tag{8}$$

From the boundary conditions (5) and (6), we obtain two additional relations for the errors

$$e_{0,0}=0 \tag{9}$$

$$e_{L,p+1}=2e_{L,p}-e_{L,p-1}-r^{2L}h_0^2 \tag{10}$$

3. SECOND-ORDER NON-DISSIPATIVE SCHEME

In this case both the numerical viscosity $Q_l^{(\mathrm{num})}$ and the constant q vanish. The corresponding steady state difference equation reduces to

$$-\frac{1}{2}\frac{1}{h_l}(u_{l,j+1}-u_{l,j-1})+s_{l,j}=0 \tag{11}$$

This is a non-dissipative scheme with a second-order accuracy on a uniform grid.

Lemma 3.1
Let the interface condition and the boundary conditions be defined by (4) and (5)–(6). Then for the difference equation (11), the numerical error is given by

$$e_{l,j}=\left[\tfrac{1}{4}(-1)^{p(L-1)}r_0^{Lp}(-1+(-1)^{pl+j}r_0^{pl})+\tfrac{1}{8}(1-r_0^{2pl})\right]h_0^2 \tag{12}$$

Proof
From (11), it is straightforward to show that the error $e_{l,j}$ satisfies the following equation which has no source term:

$$e_{l,j+1}-e_{l,j-1}=0$$

whose general solution is given by

$$e_{l,j}=a_l+b_l(-1)^j \tag{13}$$

Here a_l and b_l are constants independent of j. Introducing (13) into the interface relations (7)–(8) leads to the following recursive relations:

$$a_l=a_{l-1}+\tfrac{1}{8}(1-r_0^{2p})h_{l-1}^2$$

$$b_l=(-1)^p r_0^p b_{l-1}$$

which can be solved to give the following explicit expressions for a_l and b_l:

$$a_l = a_0 + \tfrac{1}{8}(1 - r_0^{2pl})h_0^2, \quad b_l = (-1)^{lp} r_0^{lp} b_0, \quad 1 \leqslant l \leqslant L \tag{14}$$

Introducing (13) into the boundary relations (9)–(10) gives

$$a_0 + b_0 = 0, \quad b_L = \tfrac{1}{4}(-1)^p h_L^2 \tag{15}$$

Combining the second equation of (14) for $l = L$ and the second relation of (15) leads to

$$b_0 = \tfrac{1}{4}(-1)^{p(1-L)} r_0^{Lp} h_0^2$$

By the above expression for b_0 and the first equation of (15), we finally have

$$a_0 = -\tfrac{1}{4}(-1)^{p(1-L)} r_0^{Lp} h_0^2$$

Introducing the above expressions for a_0 and b_0 into (14) leads to explicit expressions for a_l and b_l which can then be used in (13) to yield (12). □

Though the grid system involves multiple interfaces and that each interface treatment is only globally first order accurate, Lemma 3.1 implies second-order convergence rate, as can be more clearly stated in the following proposition.

Proposition 3.2
The problem defined by (4), (5)–(6), and (11) has a second-order convergence rate with respect to the coarsest mesh size h_0.

Now we consider how the subgrid width p influences the global accuracy. For that purpose, we keep the total refinement degree r_T and the total number of mesh points $p(L+1)$ fixed.

Proposition 3.3
Let pL and r_0, or equivalently r_T with $r_T \ll 1$, be fixed. Then near the fine side of the grid, i.e. when $l \rightarrow L$, the Cartesian grid method is as accurate as the smooth refinement method, that is, the numerical error is independent of the subgrid width p.

Proof
By $r_T = r_0^{Lp} \ll 1$, for $l \rightarrow L$ Equation (12) in Lemma 3.1 can be simplified as

$$e_{l,j} \rightarrow \tfrac{1}{8} h_0^2$$

so that $e_{l,j}$ is independent of p. This proves the second part. □

Remark 3.4
One would readily imagine that the Cartesian grid method, which involves closely related abrupt refinement interfaces, should be less accurate than the smooth refinement method. However, Proposition 3.3 clearly shows that in the fine side of the grid, which is the most important part for mesh refinement purpose, the numerical error is almost independent of the subgrid width if the total refinement degree is kept fixed.

4. SECOND-ORDER DISSIPATIVE SCHEME: LAX–WENDROFF SCHEME

The numerical viscosity $Q_l^{(\mathrm{num})}$ and the constant q are defined by

$$Q_l^{(\mathrm{num})} = \lambda_l, \quad q = 1$$

The corresponding steady state difference equation reduces to

$$-\frac{1}{2h_l}(u_{l,j+1} - u_{l,j-1}) + \frac{1}{2h_l}\lambda_l(u_{l,j+1} - 2u_{l,j} + u_{l,j-1}) + s_{l,j} - \frac{1}{2}k_l(s_x)_{l,j} = 0 \qquad (16)$$

This scheme has a second-order accuracy on a uniform grid and comes from the derivation of the classical Lax–Wendroff scheme [23]. To see that, let us just repeat the derivation of the Lax–Wendroff scheme for an equation with a source term, that is Equation (1). First write

$$u_{l,j}^{n+1} - u_{l,j}^{n} = k_l(u_t)_{l,j}^n + \tfrac{1}{2}k_l^2(u_{tt})_{l,j}^n + o[k_l^2] \qquad (17)$$

Dropping the high-order terms in (17), replacing u_t and u_{tt} by

$$\begin{aligned} u_t &= -u_x + s \\ u_{tt} &= (-u_x + s)_t = -(u_t)_x = -(-u_x + s)_x \\ &= u_{xx} - s_x \end{aligned}$$

where we have used the exact equation (1), we arrive at the following approximation:

$$u_{l,j}^{n+1} - u_{l,j}^{n} = -k_l(-u_x)_{l,j}^n + \tfrac{1}{2}k_l^2(u_{xx})_{l,j}^n + k_l s_{l,j} - \tfrac{1}{2}k_l^2(s_x)_{l,j}$$

which, in the steady state case, can be written as

$$-k_l(u_x)_{l,j} + \tfrac{1}{2}k_l^2(u_{xx})_{l,j} + k_l s_{l,j} - \tfrac{1}{2}k_l^2(s_x)_{l,j} = 0$$

Replacing the space derivatives by second-order central differences in the above equation leads to (16).

For steady state computation, it is natural to use local time-stepping so that we take a constant λ_l, that is $\lambda_l = \lambda$. With $u_{xx} = 1$ and $s = x$, the difference approximation (16) can be used to yield the following error equation:

$$(\lambda - 1)e_{l,j+1} - 2\lambda e_{l,j} + (\lambda + 1)e_{l,j-1} = 0 \qquad (18)$$

We remark that the above equation has no source term. Note that, by considering a uniform time stepping method so that λ_l depends on l, we can still solve the problem. The general solution of the above equation is found to be

$$e_{l,j} = a_l + b_l \kappa^j, \quad \kappa = \frac{\lambda + 1}{\lambda - 1} \tag{19}$$

Introducing this general solution into the interface conditions (7)–(8) leads to

$$a_l = a_0 + \frac{\kappa^p(\kappa + 1)}{2}(1 - r_0^p)\frac{1 - (r_0\kappa)^{pl}}{1 - (r_0\kappa)^p}\, b_0 + \frac{1}{8}(1 - r_0^{2pl})h_0^2 \tag{20}$$

$$b_l = (r_0\kappa)^{pl} b_0 \tag{21}$$

Introducing this general solution into the boundary conditions (9)–(10) leads to

$$a_0 = \frac{(r_0\kappa)^{-pL} r_0^{2pL}}{\kappa^{p-1}(\kappa - 1)^2}\, h_0^2$$

$$b_0 = -\frac{(r_0\kappa)^{-pL} r_0^{2pL}}{\kappa^{p-1}(\kappa - 1)^2}\, h_0^2$$

Under the assumption that $r_T \ll 1$, Equations (19)–(21) can be combined to yield the following lemma.

Lemma 4.1
Let $r_T \ll 1$. Then the numerical error of the problem defined by (4)–(6), and (16) is given by

$$e_{l,j} = \tfrac{1}{8}(1 - r_0^{2pl})h_0^2 \tag{22}$$

The following proposition directly follows from Lemma 4.1.

Proposition 4.2
The problem defined by (4)–(6), and (16) has a second-order convergence rate with respect to the coarsest mesh size h_0.

Similar as for Proposition 3.3, the following proposition holds.

Proposition 4.3
Let r_T with $r_T \ll 1$ and pL be fixed. Then

(a) near the coarse side of the grid, i.e. when $l \ll L$, the Cartesian grid (with large p) method is less accurate than the smooth refinement (with $p = 1$);
(b) near the fine side of the grid, i.e. when $l \to L$, the Cartesian grid method is as accurate as the smooth refinement method, that is, the numerical error is independent of the subgrid width p.

5. DISSIPATIVE SCHEME WITH STANDARD TREATMENT OF THE SOURCE TERM

The numerical viscosity $Q_l^{(\text{num})}$ is generally non-zero but the constant q vanishes. The corresponding steady state difference equation is defined by

$$-\frac{1}{2h_l}(u_{l,j+1} - u_{l,j-1}) + \frac{1}{2h_l} Q_l^{(\text{num})}(u_{l,j+1} - 2u_{l,j} + u_{l,j-1}) + s_{l,j} = 0 \tag{23}$$

Setting $Q_l^{(\text{num})} = 0$ recovers scheme (11). Setting $Q_l^{(\text{num})} = 1$ leads to the well-known first-order upwind scheme. Setting $Q_l^{(\text{num})} = \lambda_l$ yields the Lax–Wendroff scheme with standard treatment of the source term. Such a treatment for the source term is quite standard. See for instance Reference [24].

Since setting $Q_l^{(\text{num})} = \lambda_l$, with λ_l denoting the ratio between the time step and the local mesh size, still covers all the possible cases through the use of constant and local time-steppings, we will simply let $Q_l^{(\text{num})} = \lambda_l$. For example, if we let $\lambda_l = 1$, then we recover the first-order accurate upwind scheme, if we let $\lambda_l = \text{Const}$ which means local time-stepping, then we recover a central difference scheme with constant artificial dissipation, etc.

From (23) the following error equation can be derived in a straightforward way:

$$(\lambda_1 - 1)e_{l,j+1} - 2\lambda_l e_{l,j} + (\lambda_l + 1)e_{l,j-1} = d_{l,j} \tag{24}$$

where $d_{l,j} = -\lambda_l h_l^2$ is a source term. The error due to $d_{l,j}$ will be coupled with the errors due to interface treatment.

The general solution of (24) is found to be

$$e_{l,j} = a_l + b_l \kappa_l^j + \frac{j}{2}\lambda_l h_l^2, \quad \kappa_l = \frac{\lambda_l + 1}{\lambda_l - 1} \tag{25}$$

where a_l and b_l are constants to be determined by the interface and boundary conditions.

5.1. Reference error: error on a uniform grid ($r_0 = 1$)

In order to see how (abrupt) mesh refinement improves the accuracy, it is useful to compare the error with the error e_{ref} on a uniform grid ($h_l = h_0$). To avoid confusion, here we assume that $p = 1$. This is not a restriction since the grid is uniform in the entire domain. For convenience, e_{ref} will be called reference error. Comparing with the reference error, we will be able to answer the question: does mesh refinement necessarily reduce the numerical error with respect to a method on a coarse but uniform grid?

When $r_0 = 1$, we can write $a_l = a$, $b_l = b$, $\lambda_l = \lambda$, $\kappa_l = \kappa$. There is no need to use interface conditions since they degenerate to trivial identities. Introducing (25) with $a_l = a$ and $b_l = b$ into the boundary conditions (9)–(10) leads to

$$a = \tfrac{1}{4}\kappa^{-L}(\lambda^2 - 1)h_0^2, \quad b = -\tfrac{1}{4}\kappa^{-L}(\lambda^2 - 1)h_0^2$$

With the help of (25), the error at point l, now simply denoted as $e_{\text{ref}}(l)$, is found to be

$$e_{\text{ref}}(l) = \frac{1}{4}\kappa^{-L}(\lambda^2 - 1)(1 - \kappa^{-l})h_0^2 + \frac{l}{2}\lambda h_0^2 \approx \frac{l}{2}\lambda h_0^2 \tag{26}$$

For purpose of direct comparison with the result for $r_0 < 1$, it is convenient to relate the reference error to the abscise x. Since for $r_0 = 1$, $l = x/h_0$ (ignoring the discussion of whether the point $l = 0$ locates at $x = h_0$ or at $x = \frac{1}{2}h_0$), the reference error (26) can be more conveniently written as

$$e_{\text{ref}}(x) \approx \tfrac{1}{2}\lambda h_0 x \tag{27}$$

We will compare (27) with the numerical error $e_{l,j}$ for the general case with $r_0 < 1$ and $p > 1$.

First remark that

$$x_{l,j} = \sum_{l^0=0}^{l^0=l-1} p h_{l^0} + j h_l = \left(\sum_{l^0=0}^{l^0=l-1} p r^{l^0} + j r^l \right) h_0 = \left(\frac{p(1-r^l)}{1-r} + j r^l \right) h_0$$

Thus, at the point $x_{l,j}$ or l, j of any abrupt grid system with $r_0 < 1$ and $p > 1$, the reference error (27), which is the error for a uniform grid, can be written as

$$\begin{aligned} e_{\text{ref}}(l,j) = e_{\text{ref}}(x_{l,j}) &\approx \frac{1}{2}\lambda \left(\frac{p(1-r^l)}{1-r} + j r^l \right) h_0^2 \\ &= \frac{1}{2}\lambda \left(\frac{p(1-r_0^{pl})}{1-r_0^p} + j r_0^{pl} \right) h_0^2 \end{aligned} \tag{28}$$

5.2. *Error for constant time-stepping*

For constant time-stepping, $k_l = k$ so that $\lambda_l = k/h_l = \lambda_L r_0^{p(L-l)}$. Let $\eta = \lambda_L r_0^{pL}$. Then (25) can be rewritten as

$$e_{l,j} = a_l + b_l R(l)^j + \frac{j}{2}\eta r_0^{pl} h_0^2, \quad R(l) = \frac{\eta + r_0^{pl}}{\eta - r_0^{pl}} \tag{29}$$

Introducing (29) into the interface conditions (7)–(8) leads to the following expressions for a_l and b_l:

$$a_l = a_0 + Q(l) b_0 + \frac{1}{4}\eta(2p + 1 - r_0^p)\frac{1 - r_0^{pl}}{1 - r_0^p} h_0^2 + \frac{1}{8}(1 - r_0^{2pl}) h_0^2 \tag{30}$$

$$b_l = b_0 S(l) \tag{31}$$

where

$$Q(l) = \frac{1}{2}\sum_{i=1}^{l} S(i) \left(\frac{[R(i-1)+1][R(i)-1]}{R(i-1)-1} - R(i) - 1 \right)$$

$$S(l) = r_0^p \prod_{i=0}^{i=l-1} R(i)^p \frac{\prod_{i=0}^{i=l-1}(R(i)-1)}{\prod_{i=0}^{i=l-1}(R(i)-1)}$$

Introducing (29) into the boundary conditions (9)–(10) leads to the following expressions for a_0 and b_0:

$$a_0 = \frac{1}{4}\frac{r_0^{2pl}}{S(L)}(\lambda_L^2 - 1)h_0^2, \quad b_0 = -\frac{1}{4}\frac{r_0^{2pl}}{S(L)}(\lambda_L^2 - 1)h_0^2 \tag{32}$$

The error expression (29) involves complicated formulas for the constants a_l and b_l. It is thus very difficult to draw conclusions in an explicit way. Here we just consider the particular case of $\lambda_L \to 1$. This is the upper bound of the stability condition of usual explicit schemes. In this case $a_0 = 0$ and $b_0 = 0$ according to (32). Then with the help of (30) and (31), the error expression (29) can be simplified as

$$e_{l,j} = \frac{1}{4}\left[r_T\left(\frac{2p}{1-r_0^p}+1\right)(1-r_0^{pl}) + \frac{1}{2}(1-r_0^{2pl}) + 2r_T r_0^{pl} j\right]h_0^2 \tag{33}$$

Thus we have proved.

Proposition 5.1
Let r_T be fixed and let $\lambda_L \to 1$. The problem defined by (4)–(6), and (23), with constant time-stepping, has a second-order convergence rate with respect to the coarsest mesh size h_0.

Similar as for Propositions 3.3 and 4.3, the following proposition holds.

Proposition 5.2
Let r_T with $r_T \ll 1$ and pL be fixed and let $\lambda_L \to 1$. Then for the problem defined by (4)–(6), and (23), with constant time-stepping,

(a) near the coarse side of the grid, i.e. when $l \ll L$, the Cartesian grid (with large p) method is less accurate than the smooth refinement (with $p = 1$);
(b) near the fine side of the grid, i.e. when $l \to L$, the Cartesian grid method is as accurate as the smooth refinement method, that is, the numerical error is independent of subgrid width p.

Proof
We just consider the second part. In fact, for $l \to L$ and $r_T \ll 1$, (33) reduces to

$$e_{l,j} = \tfrac{1}{2} h_0^2$$

which is independent of p. □

Proposition 5.3
When constant time-stepping is used, refining the mesh from the left to the right, in a way such that the coarsest mesh size is equal to the mesh size on the corresponding uniform grid, always improves the accuracy.

Proof
Compare the numerical error (33) with the reference error (28) and write

$$\frac{e_{l,j}}{e_{\text{ref}}(l,j)} = \frac{\frac{1}{4}[r_T((2p/(1-r_0^p))+1)(1-r_0^{pl}) + \frac{1}{2}(1-r_0^{2pl}) + 2r_T r_0^{pl} j]h_0^2}{\frac{1}{2}\lambda((p(1-r_0^{pl})/(1-r_0^p)) + j r_0^{pl})h_0^2}$$

$$= \frac{r_T((2p/(1-r_0^p))+1)(1-r_0^{pl}) + \frac{1}{2}(1-r_0^{2pl}) + 2r_T r_0^{pl} j}{2\lambda(p(1-r_0^{pl})/(1-r_0^p) + j r_0^{pl})}$$

$$= \frac{((2pr_T/(1-r_0^p)) + r_T + \frac{1}{2}(1+r_0^{pl}))(1-r_0^{pl}) + 2r_T r_0^{pl} j}{2\lambda(p(1-r_0^{pl})/(1-r_0^p) + j r_0^{pl})}$$

A straightforward calculation using the above equation shows that

$$\left|\frac{e_{l,j}}{e_{\text{ref}}(l,j)}\right| < 1 \quad \forall l, j$$

for all $r_0 < 1$. □

5.3. *Error for local time-stepping*

For the well-known local time-stepping method useful for steady state computation, the time step k_l is proportional to the mesh size h_l so that λ_l is a constant independent of l. Let us denote $\lambda_l = \lambda$. Introducing (25) into the interface conditions (7)–(8) leads to the following recursive relations:

$$a_l = a_{l-1} + \alpha b_{l-1} + \beta h_{l-1}^2 \tag{34}$$

$$b_l = \gamma b_{l-1} + \theta h_{l-1}^2 \tag{35}$$

where

$$\alpha = \kappa^p \frac{\lambda}{\lambda - 1}(1 - r_0^p), \quad \kappa = \frac{\lambda + 1}{\lambda - 1}$$

$$\beta = \tfrac{1}{4}[\lambda(2p+1) + \tfrac{1}{2} - \left(\lambda + \tfrac{1}{2}\right) r_0^{2p} - \lambda^2 r_0^p (1 - r_0^p)]$$

$$\gamma = r_0^p \kappa^p$$

$$\theta = \tfrac{1}{4}\,\lambda(\lambda - 1) r_0^p (1 - r_0^p)$$

After a large number of algebraic calculations, we obtain from the recursive relations (34)–(35) the following expressions for a_l and b_l:

$$\begin{aligned} a_l &= a_0 + \alpha \frac{1-\gamma^l}{1-\gamma} b_0 + \sum_{i=0}^{i=l-1} (\alpha\theta q(i) + \beta r_0^{2p}) r_0^{2p(i-1)} h_0^2 \\ &= a_0 + \left(\beta + \frac{\alpha\theta}{r_0^{2p} - \gamma}\right) \frac{1 - r_0^{2pl}}{1 - r_0^{2p}} h_0^2 - \left(\frac{\alpha\theta}{r_0^{2p} - \gamma} h_0^2 - \alpha b_0\right) \frac{1-\gamma^l}{1-\gamma} \end{aligned} \tag{36}$$

$$b_l = \gamma^l b_0 + \theta q(l) r_0^{2p(l-1)} h_0^2 \tag{37}$$

where

$$q(l) = \frac{1-(\gamma/r_0^{2p})^l}{1-(\gamma/r_0^{2p})}$$

Introducing (25) into the boundary conditions (9)–(10), and using (36)–(37), we obtain the following expressions for a_0 and b_0:

$$a_0 = [\gamma^{-L}\theta q(L) r_0^{2p(L-1)} + \tfrac{1}{4}\gamma^{-L}\kappa^{-p}(\lambda^2-1) r_0^{2pL}]h_0^2$$

$$b_0 = -[\gamma^{-L}\theta q(L) r_0^{2p(L-1)} + \tfrac{1}{4}\gamma^{-L}\kappa^{-p}(\lambda^2-1) r_0^{2pL}]h_0^2$$

The above expressions are too complicated to be used in deriving useful results. Thus we only consider the case of $l \to L$ and $r_T \ll 1$. Introducing these expressions into (36)–(37) leads to the final expressions for a_l and b_l which, after dropping high-order terms, can be written as

$$a_1 \approx \left[\left(\beta + \frac{\alpha\theta}{r_0^{2p}-\gamma}\right)\frac{1-r_0^{2pl}}{1-r_0^{2p}} - \frac{\theta}{r_0^{2p}-\gamma}\right]h_0^2, \quad b_l \approx \theta\,\frac{r_0^{2pl}}{r_0^{2p}-\gamma}\,h_0^2$$

Introducing the above expressions for a_l and b_l into (25) leads to the final expression for the error

$$e_{l,j} \approx \left[\left(\beta + \frac{\alpha\theta}{r_0^{2p}-\gamma}\right)\frac{1-r_0^{2pl}}{1-r_0^{2p}} - \frac{\theta}{r_0^{2p}-\gamma} + \theta\,\frac{r_0^{2pl}}{r_0^{2p}-\gamma}\,\kappa^j + \frac{1}{2}\,\lambda r_0^{2pl} j\right]h_0^2 \tag{38}$$

Thus we have proved.

Proposition 5.4
Let r_T with $r_T \ll 1$ be fixed and let $\lambda_L \to 1$ and $l \to L$. The problem defined by (4)–(6), and (23), with local time-stepping, has a second-order convergence rate with respect to the coarsest mesh size h_0.

In fact the above proposition also holds for any $0 \leqslant l < L$.

From (38) there is no difficulty to prove the following.

Proposition 5.5
Let r_T with $r_T \ll 1$ and pL be fixed. Furthermore let $l \to L$. Then for the problem defined by (4)–(6), and (23), with local time-stepping, the numerical error at a fixed abscise is an increasing function of the subgrid width p.

The above proposition also holds for any l with $0 \leqslant l \leqslant L$.

Comparing the different terms in (38), it can be shown that the combination of several terms is relatively very small. Notably, for the parameters considered, it holds that

$$\beta\,\frac{1}{1-r_0^{2p}} \gg \left|\frac{\alpha\theta}{r_0^{2p}-\gamma}\,\frac{1}{1-r_0^{2p}} - \frac{\theta}{r_0^{2p}-\gamma}\right|, \quad r_0^{2pl} j \approx 0$$

Thus (38) can be further simplified as

$$e_{l,j} \approx \frac{\beta}{1-r_0^{2p}} h_0^2 \tag{39}$$

Proposition 5.6
Let $l \to L, r_T \ll 1$ and $p=1$. Then

(a) for $1 \geqslant r_0 \geqslant 2\sqrt{3}-3$, the error $e_{l,1}$ is smaller than $e_{\text{ref}}(l,1)$ for $0 \leqslant \lambda \leqslant 1$, that is, the considered mesh refinement method is more accurate than the uniform mesh method with a mesh size equal to h_0 (coarsest mesh size);
(b) for $r_0 \leqslant 2\sqrt{3}-3$, then there exists a $\delta = \delta(r_0) > 0$ such that the error $e_{l,1}$ is larger than $e_{\text{ref}}(l,1)$ for $1-\delta \leqslant \lambda \leqslant 1$, that is, the mesh refinement method is less accurate than the uniform mesh method with a mesh size equal to h_0 (coarsest mesh size).

Proof
Compare the numerical error (39) with the reference error (28) and write for $p=1$

$$\frac{e_{l,1}}{e_{\text{ref}}(l,1)} \approx \frac{(\beta(p=1)/(1-r_0^2))h_0^2}{\frac{1}{2}\lambda((1-r_0^l)/(1-r_0)+r_0^l)h_0^2} \approx \frac{2\lambda\beta}{1+r_0}$$

The results then follow without any difficulty. □

The above proposition shows that one should not refine the grid too abruptly, otherwise the accuracy is even worse than that of a coarse grid method. Similar result holds for $p>1$.

5.4. Error for conservative local time-stepping

The above local time-stepping method that one would naturally adopt, believing that it results in a conservative treatment, makes the discretization non-conservative so that it cannot be extended to compute solutions with discontinuity. The question of conservation is hotly debated for non-linear problems with shock waves, see for instance Reference [22].

To see that, let us consider the interface $I_{l+1/2}$ which separates the subgrids l and $l+1$, and write the difference equation (2) in conservative form

$$u_{l,j}^{n+1} - u_{l,j}^n = -\lambda_l(f_{l,j+1/2}^n - f_{l,j-1/2}^n) + k_l s_{l,j} - \frac{q}{2} k_l^2 (s_x)_{l,j}$$

where $f_{l,j+1/2} = \frac{1}{2}(u_{l,j}+u_{l,j+1}) - \frac{1}{2}\lambda_l(u_{l,j+1}-u_{l,j})$ is the numerical flux. Then the right-hand numerical flux for the point p of subgrid l is defined by

$$f_{l,p+1/2} = \frac{1}{2}(u_{l,p}+u_{l,p+1}) - \frac{1}{2}\lambda_l(u_{l,p+1}-u_{l,p})$$

and the left-hand numerical flux for the point 1 of subgrid $l+1$ is defined by

$$f_{l+1,1/2} = \frac{1}{2}(u_{l+1,0}+u_{l+1,0}) - \frac{1}{2}\lambda_{l+1}(u_{l+1,1}-u_{l+1,0})$$

Here we have assumed $Q_l^{(\text{num})} = \lambda_l$.

For the interface condition (4), it is very clear that conservation, i.e. $f_{l+1,1/2} = f_{l,p+1/2}$, is ensured only if $k_l = k_{l+1}$, which means that $\lambda_{l+1} \neq \lambda_l$. In order to make the interface treatment conservative within the framework of local time-stepping, we modify the local time-stepping at the interface by taking an averaged time-step

$$k_{l+1/2} = \tfrac{1}{2}(k_l + k_{l+1})$$

for the interface fluxes $f_{l,\,p+1/2}$ and $f_{l+1,1/2}$, so that

$$f_{l,\,p+1/2} = \frac{1}{2}(u_{l,\,p} + u_{l,\,p+1}) - \frac{1}{2}\frac{k_{l+1/2}}{h_l}(u_{l,\,p+1} - u_{l,\,p})$$

$$f_{l+1,1/2} = \frac{1}{2}(u_{l+1,0} + u_{l+1,0}) - \frac{1}{2}\frac{k_{l+1/2}}{h_{l+1}}(u_{l+1,1} - u_{l+1,0})$$

Though the modification is minor, the analysis is heavily complicated. Here we will just consider the case of $p=1$. The main conclusion holds true for $p>1$.

For simplicity, let us denote $u_{l,1} = v_l$ and $e_{l,1} = e_l$. Quite straightforwardly, one can derive the following error equation from the interior difference equation and the interface condition

$$\left(\lambda - \frac{2}{1+r}\right) e_{l+1} - 2\left(\lambda + \frac{r-1}{1+r}\right) e_l + \left(\lambda + \frac{2r}{1+r}\right) e_{l-1} = d_l \tag{40}$$

where

$$d_l = -\frac{\lambda}{2}\left(r - \frac{1}{r}\right) h_l u_x - \frac{1}{8}\left[(1+r)^2\left(1 + \frac{1}{r^2}\right)\lambda + 2(1+r)\left(\frac{1}{r} - 1\right)\right] h_1^2 u_{xx} \tag{41}$$

with $u_{xx} = 1$ and $u_x = x$.

Thus the source term for the error equation depends on the first-order derivative and $d_l = O[h_l]$. This seems to indicate that the conservative local time-stepping method would be less accurate than the previous non-conservative one. But our subsequent analysis, supplemented by numerical test, will show the contrast.

Using the interface conditions (7)–(8) at the interfaces $I_{L-1/2}$ and $I_{l+1/2}$, one can show that the corresponding boundary conditions (6) in terms of the new variable v_l becomes

$$\frac{1+r}{r+2+r^{-1}}\, v_{L-1} + \frac{1+r^{-1}}{r+2+r^{-1}}\, v_{L+1} = v_L$$

and the error constraints due to boundary treatment are given by

$$e_0 = 0 \tag{42}$$

$$\frac{1+r}{r+2+r^{-1}}\, e_{L-1} + \frac{1+r^{-1}}{r+2+r^{-1}}\, e_{L+1} - e_L = -\frac{(1+r)^2}{8r}\, r^{2L} h_0^2 \tag{43}$$

Now we want to replace $u_x = x$ by l in (41). Introducing

$$x = \tfrac{1}{2} h_0 + h_1 + h_2 + \cdots + h_{l-1} + \tfrac{1}{2} h_l$$
$$= \frac{1+r}{2(1-r)}(1 - r^l)h_0$$

into (41) yields

$$d_l = D_1 r^l + D_2 r^{2l}$$
$$D_1 = \frac{1}{4r}\lambda(1+r)^2 h_0^2$$
$$D_2 = -\frac{1}{8r^2}\lambda(1+r)^4 h_0^2 - \frac{1}{4}(r^{-1} - r)h_0^2$$

Introducing the above expressions into (40), one can derive the following general formula for the numerical error:

$$e_l = c_1 + c_2\kappa^l + ar^l + br^{2l} \tag{44}$$

where

$$\kappa = \frac{\lambda + (2r/(1+r))}{\lambda - (2/(1+r))}$$

$$a = \frac{D_1}{\lambda r^{-1}(1-r)^2 + 4(1-r)(1+r)^{-1}} \approx \frac{(1+r)^3\lambda}{16(1-r)r} h_0^2 \tag{45}$$

$$b = \frac{D_2}{(1-r)^2[2r + \lambda(1-r^2)]} \approx -\frac{(1+r)^3\lambda}{16(1-r)r} h_0^2 \tag{46}$$

Introducing (44) into the boundary conditions (42)–(43) we obtain

$$c_1 = -a - b - c_2 \tag{47}$$

$$c_2 = -\frac{\kappa^{-L}}{(r\kappa^{-1} + \kappa)(1+r)^{-1} - 1}\left[\frac{(1+r)^2}{8r} h_0^2 + \frac{b}{r}(1-r)^2\right] r^{2L} \tag{48}$$

Proposition 5.7
Let $r < 1$ and $r_T \to 0$. Then we have

$$e_l = -(a+b) + ar^l + br^{2l} \tag{49}$$

Furthermore, the error e_l takes its maximum value at $l = l_m$ with

$$l_m \approx -\frac{\ln(1+r)}{\ln r} \tag{50}$$

Proof
By $r<1$, we have $|\kappa|>1$. By assumption $r^L=r_T\to 0$ and by (48) we necessarily have $c_2\to 0$ and $c_2\kappa^l\approx 0$. Thus by (47) we have $c_1=-(a+b)$. In consequence, (44) reduces to (49).

Let e_l take its maximum value at l_m, then we have $e_{l_m+1}-e_{l_m}=0$ which, by (49), yields

$$l_m=\frac{\ln -(a/(1+r)b)}{\ln r}\approx -\frac{\ln(1+r)}{\ln r} \qquad \square$$

It is surprising that l_m only depends on r but not on λ and L.

Proposition 5.8
Let $r<1$. Then for large values of l, the conservative local time-stepping method is more accurate than the non-conservative local time-stepping method.

Proof
The error for the non-conservative method is given by (39). When the explicit expression for β is introduced, expression (39), for $j=1$ since $p=1$, can be written as

$$e_{l,1}=\frac{\frac{1}{4}[3\lambda+\frac{1}{2}-(\lambda+\frac{1}{2})r_0^2-\lambda^2 r_0(1-r_0)]}{1-r_0^2}h_0^2$$

which is bounded from zero for large l.

For the conservative treatment, (49) and (45)–(46) clearly show that for large l we have $e_l\approx 0$. $\square$

6. NUMERICAL TEST AND CONCLUDING REMARKS

6.1. Numerical test

The numerical errors derived above are all based on analysis. Here we check some results which involve complicated formulas. Notably, we want to check (38), which is the error based on non-conservative local time-stepping, and (44), which is based on conservative local time-stepping. Also, we want to check (50), which is the location of maximum error. It is fairly interesting that the maximum error locates at the middle of the coarsest mesh and the finest mesh.

Precisely, we will solve the following time-dependent difference equation

$$u_{l,j}^{n+1}=u_{l,j}^n-\tfrac{1}{2}\lambda_l(u_{l,j+1}^n-u_{l,j-1}^n)+\tfrac{1}{2}\lambda_l^2(u_{l,j+1}^n-2u_{l,j}^n+u_{l,j-1}^n)+k_l s_{l,j}$$

with the interface condition (4) and the boundary conditions (5)–(6). Both non-conservative local time-stepping and conservative local time-stepping are considered.

The grid system is defined as follows:

$$L=20, \quad p=1, \quad r=0.8858, \quad h_0=0.1$$

The time step is defined by setting $\lambda_l=\frac{1}{2}$. The numerical steady state solution is obtained by solving iteratively the above difference equation. Once the numerical solution no longer varies in time (in the sense of zero-machine), then we consider the solution to be the steady state one. Double precision is used in order to minimize the effect of rounding errors.

Table I. Comparison between the theory and numerical experiment for non-conservative local time-stepping.

Location x and l	e_{comp}	e_{the} by Equation (38)
0.100143 ($l = 0$)	5.161713e − 03	2.507173e − 03
0.278210 ($l = 2$)	1.381920e − 02	1.218490e − 02
0.417929 ($l = 4$)	1.914930e − 02	1.814312e − 02
0.527558 ($l = 6$)	2.243084e − 02	2.181137e − 02
0.613577 ($l = 8$)	2.445116e − 02	2.406978e − 02
0.681072 ($l = 10$)	2.569499e − 02	2.546019e − 02
0.734037 ($l = 12$)	2.646074e − 02	2.631622e − 02
0.775585 ($l = 14$)	2.693196e − 02	2.684324e − 02
0.808190 ($l = 16$)	2.722050e − 02	2.716771e − 02
0.833773 ($l = 18$)	2.738708e − 02	2.736747e − 02

Table II. Comparison between the theory and numerical experiment for conservative local time-stepping.

Location x and l	e_{comp}	e_{the} by Equation (44)
0.100143 ($l = 0$)	4.9016136e − 03	4.2035016e − 03
0.278210 ($l = 2$)	1.1128670e − 02	8.8077644e − 03
0.417929 ($l = 4$)	1.3317105e − 02	1.0302944e − 02
0.527558 ($l = 6$)	1.3373448e − 02	1.0172450e − 02
0.613577 ($l = 8$)	1.2395163e − 02	9.2674345e − 03
0.681072 ($l = 10$)	1.0998019e − 02	8.0631766e − 03
0.734037 ($l = 12$)	9.5139807e − 03	6.8140388e − 03
0.775585 ($l = 14$)	8.1095053e − 03	5.6466125e − 03
0.808190 ($l = 16$)	6.8511702e − 03	4.6152868e − 03
0.833773 ($l = 18$)	5.7112593e − 03	3.7350710e − 03

Table III. Comparison between the theory and numerical experiment for the location of the maximum error.

Refinement degree	$r = 0.85$	$r = 0.8858$	$r = 0.9$	$r = 0.95$
l_m given by Equation (50)	3.78	5.22	6.09	13.02
l_m given by numerical experiments	4	5	6	13

In Table I we display the comparison between the computed error (which we denote e_{comp}) and the theoretical error (which we denote e_{the}) given by (38) for the non-conservative local time-stepping method. The agreement between the theory and the numerical results is very good.

In Table II we display the comparison between the computed error (which we denote e_{comp}) and the theoretical error (which we denote e_{the}) given by (44) for the conservative local time-stepping method. In comparison with the non-conservative results displayed in Table I, we see that the conservative treatment indeed improves the accuracy (by almost one order of magnitude in the finest part of the grid). There is however a small difference between the theory and the numerical results. This is because that we have used approximated estimates for the constants a_l and b_l.

In Table III we display the comparison between the numerical result and the theoretical equation (50) for the location of maximum error. The agreement is remarkable.

6.2. Concluding remarks

The present study is concerned with the steady state errors for three-point difference equations on a Cartesian grid with multiple and abruptly refined interfaces. For a solution with constant second-order derivative and when the total refinement degree is maintained fixed, it leads to the following conclusions:

(A) the steady state error is a weakly increasing function of the subgrid width;
(B) the steady state error in the finest part of the grid is independent of the subgrid width;
(C) for the standard treatment of the source term, a conservative treatment for local time-stepping improves the accuracy;
(D) the standard treatment of the source term, with non-conservative local time-stepping and with large refinement ratio, is even less accurate than the corresponding scheme on a uniform coarse grid.

Since the accuracy is independent of the subgrid width in the finest part of the grid which is the most important in refinement problems, neither the smooth refinement nor the Cartesian grid method is superior to the other in accuracy. Of course the Cartesian grid method with large subgrid width seems to be more suitable for flows where the fine flow structure is contained in a small zone to be covered by a subgrid with uniform mesh size.

ACKNOWLEDGEMENTS

This work was supported by Chinese NSF (Contract No. 10472056, 10376016, 10025210) and by the China NKBRSF project (Contract No. 2001CB409600).

REFERENCES

1. Arnoy DC, Flaherty JE. An adaptive local mesh refinement method for time-dependent partial differential equations. *Applied Numerical Mathematics* 1989; **5**:257–274.
2. Berger MJ. Stability of interfaces with mesh refinement. *Mathematics of Computation* 1985; **45**:301–318.
3. Bell J, Berger MJ, Saltzman JS, Welcome M. Three dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM Journal on Scientific Computing* 1994; **15**:127–138.
4. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* 1989; **82**:64–84.
5. Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* 1984; **53**:484–512.
6. Coirier WJ, Powell KG. An accuracy assessment of Cartesian-mesh approaches for the Euler equations. *Journal of Computational Physics* 1995; **117**:121–131.
7. Coirier WJ, Powell KG. Solution adaptive Cartesian cell approach for viscous and inviscid flows. *AIAA Journal* 1996; **34**:938–945.
8. De Zeeuw D, Powell KG. An adaptively refined Cartesian mesh solver for the Euler equations. *Journal of Computational Physics* 1993; **104**:56–68.
9. Gropp WD. Local uniform mesh refinement with moving grids. *SIAM Journal on Scientific and Statistical Computing* 1987; **8**:292–304.
10. LeVeque RJ. Cartesian grid methods for flow in irregular regions. In *Numerical Methods in Fluid Dynamics III*, Morton KW, Baines MJ (eds). Clarendon Press: Oxford, 1988; 375–382.

11. Quirk J. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers and Fluids* 1994; **23**:125–142.
12. Wu ZN. Anisotropic Cartesian grid method for viscous flow computations. *Computational Fluid Dynamics Review*, vol. 1. World Scientific Publishing: Singapore, 1998; 93–113.
13. Wu ZN, Li K. Anisotropic Cartesian grid method for steady inviscid shocked flow computation. *International Journal for Numerical Methods in Fluids* 2003; **41**:1053–1084.
14. Li K, Wu ZN. Nonet-Cartesian grid method for shocked flow computations. *Journal on Scientific Computing*, in press.
15. Gustafsson B. The convergence rate for difference approximations to general initial boundary value problems. *SIAM Journal on Numerical Analysis* 1981; **18**:179–190.
16. Browning G, Kreiss H-O, Oliger J. Mesh refinement. *Mathematics of Computation* 1973; **27**:29–39.
17. Ciment M. Stable difference schemes with uneven mesh spacings. *Mathematics of Computation* 1971; **114**: 219–226.
18. Lerat A, Wu ZN. Stable conservative multidomain treatments for implicit Euler solvers. *Journal of Computational Physics* 1996; **123**:45–64.
19. Wu ZN. Numerical dissipation for three-point difference approximations to hyperbolic equations with uneven meshes. *Journal of Computational Mathematics* 2003; **21**:519–534.
20. Yamaleev NK, Carpenter MH. On accuracy of adaptive grid methods for captured shocks. *Journal of Computational Physics* 2002; **181**:280–316.
21. Tadmor E. Numerical viscosity and the entropy condition for conservative difference schemes. *Mathematics of Computation* 1984; **43**:369–381.
22. Wu ZN. Transmission of slowly moving shock waves across a nonconservative interface. *Journal of Computational Physics* 2001; **171**:579–615.
23. Gustafsson B, Kreiss H-O, Sundström A. Stability theory of difference approximations for initial boundary value problems II. *Mathematics of Computation* 1972; **26**:649–686.
24. Lax P, Wendroff B. System of conservation laws. *Communications on Pure and Applied Mathematics* 1960; **XIII**:217–237.