Stability and Error Estimation for Component Adaptive Grid Methods

Joseph Oliger and Xiaolei Zhu

The Research Institute of Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 212, Columbia, MD 21044, (410) 730-2656

This work was begun with support from the Office of Naval Research under contracts N00014-89-J-185-P00006 and N00014-90-J-1344-P00005 and completed with support from the National Science Foundation under grant DMS-9318166 and NASA under contract NAS2-13721.

Abstract

Component adaptive grid (CAG) methods for solving hyperbolic partial differential equations (PDE's) are discussed in this paper. Applying recent stability results for a class of numerical methods on uniform grids, the convergence of these methods for linear problems on component adaptive grids is established here. Furthermore, the computational error can be estimated on CAG's using the stability results. Using these estimates, the error can be controlled on CAG's. Thus, the solution can be computed efficiently on CAG's within a given error tolerance. Computational results for time dependent linear problems in one and two space dimensions are presented.

1 Introduction

Component adaptive grid methods for solving hyperbolic PDE's were introduced in the early 1980's. An overview of the method is given in Section 2. More details can be found in Berger and Oliger [2], and Berger [1]. However, the grid structure used in this paper is different from the one Berger used. As discussed in Section 2, stair step grids like those of Chesshire and Henshaw [4] are used in our CAG methods here, instead of rotated rectangular grids. One major component of the adaptive strategy is to estimate the local truncation error at each grid point, then refine where the estimated errors are larger than a given tolerance δ. The smaller δ is, the smaller the final error ϵ is expected to be in some weighted L_2 norm. However, no quantitative relationship between these two kinds of errors had been established. Recently, new stability results have been developed by Pelle Olsson [6] which allow us to establish such a relationship for large classes of problems and methods. The results can be applied to various classes of problems, e.g., those of hyperbolic, parabolic and hyperbolic-parabolic type, using a large class of numerical methods on uniform grids. As we will see in Section 3, the structures of component adaptive grids allow us to define the solution on piecewise uniform grids. So the stability theories can be applied on CAG's. Convergence for linear problems using these methods on CAG's is proved in Section 3. Also the tolerance δ on local truncation error is estimated in term of the tolerance ϵ on the final error. Furthermore, the results in Section 3 will also help us estimate the final error using simple quadrature, and serve us as guidelines on developing strategies for CAG methods, since we have a very good understanding of the sources and the magnitudes of various computational errors. Finally, some computional results for time dependent problems in one and two space dimensions are given in Section 4.

2 An Overview of Component Adaptive Grids

We first introduce some notation for our discussion. Suppose the problem we wish to solve is written as

$$u_t = Lu + f \quad on \quad \Omega \times [0,T] \tag{1}$$

$$u(0) = u_0 \quad on \quad \Omega \tag{2}$$

$$Bu = b \quad on \quad \partial\Omega \times [0,T] \tag{3}$$

where $\Omega \subset R^d$ is a bounded domain in physical space, L is a spatial partial differetial operator on Ω and $u \in R^n$. We assume this to be a well-posed initial-boundary value problem which is defined in Section 3. Let Ω_h, $\partial\Omega_h$ and $[0,T]_k$ be the discretizations of Ω, $\partial\Omega$ and $[0,T]$, respectively. In Section 3, these discretizations are defined precisely

for our component adaptive grids. For the time being, we can consider them as general grids.

Let v_h be a grid function defined on $\Omega_h \times [0,T]_k$. We will discuss the use of finite difference methods on these grids. Without loss of generality, and avoiding complicated notation, we write our methods in explicit one-step form as

$$\begin{aligned} v_h(t+k) &= L_h\, v_h(t) + k\, f_h(t) \quad on \quad \Omega_h \times [0,T]_k & (4)\\ v_h(0) &= u_{0h} \quad on \quad \Omega_h & (5)\\ B_h\, v_h(t) &= b_h(t) \quad on \quad \partial\Omega_h \times [0,T]_k & (6) \end{aligned}$$

where we use subscripts to denote projections of functions onto the appropriate grids and discretizations of operators on these grids. If u_h is the projection of the exact solution of the above system onto Ω_h, then

$$u_h(t+k) = L_h\, u_h(t) + k\, f_h(t) + k\tau_h \quad on \quad \Omega_h \times [0,T]_k \qquad (7)$$

where τ_h is the local truncation error. This notation will also be used on the piecewise uniform grids which we will discuss next.

2.1 Composite Grids

In real applications, the physical domains often have complicated geometries. In order to use finite difference schemes on these domains, we decompose the physical domain and transform the parts into computational domains. However this topic is not the focus of this paper. Here only a brief introduction is given to make our presentation self contained. Details can be found in Chesshire and Henshaw [4], Venkata, Oliger and Ferziger [8], and Venkata [9].

We begin by forming a base *composite grid*

$$G_0 = \bigcup_j G_{0,j} \qquad (8)$$

which will be characterized by a discretization parameter h_0.

This is well illustrated in Figure 1 where G_0 consists of the *component grids* $G_{0,1}$, $G_{0,2}$, and $G_{0,3}$. $G_{0,2}$ is a stair step grid with grid lines parallel to the coordinate axes. Such grids are called *regular grids.*

Definition 1: A regular grid is a *connected* stair step grid of uniformly spaced points in each coordinate direction, and its grid lines are parallel to the coordinate

axes in either physical or computational space.

No coordinate transformation is needed to solve the equation on regular grids in physical space. The curvilinear grids $G_{0,1}$ and $G_{0,3}$ are defined by specifying their boundaries and cuts. Regular girds in computational space are then mapped onto these grids in physical space using coordinate tranformations. To reduce clutter in Figure 1 , grids $G_{0,1}$ and $G_{0,3}$ are shown only in computational space. The component grids are chosen to obtain a sufficiently accuate representation of $\partial\Omega$ by $\partial\Omega_h$. h_0 is an estimate of the step size required to obtain a sufficiently accuate approximation of the solution over at least some specified fraction of the domain. The difficult problem here is to generate grids on the boundaries. The Bézier family of curves and surfaces are used to generate boundary grids in 2-D and 3-D, respectively (see [8] and [9]).

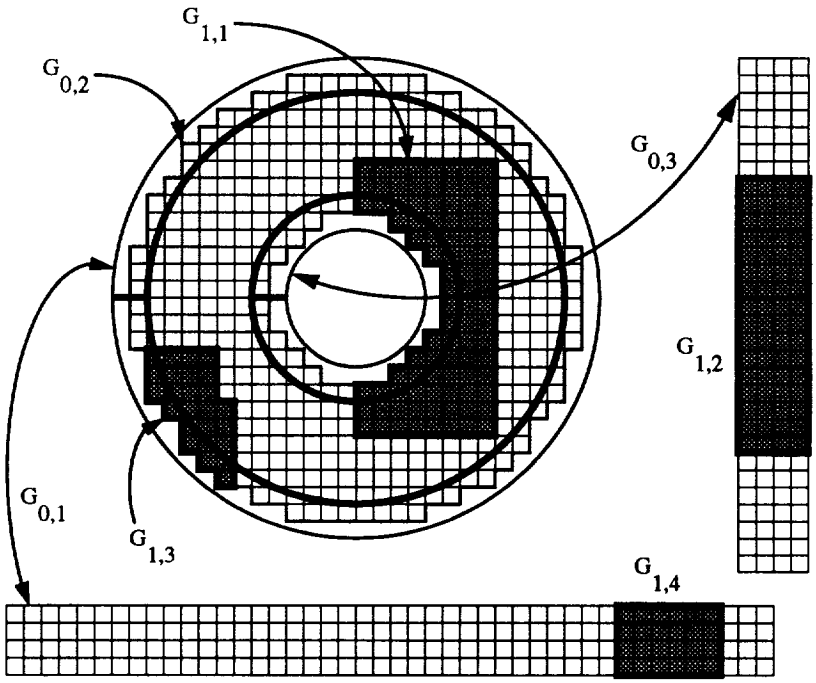

Figure 1: Adaptive Composite Grid Structure

2.2 Component Adaptive Grids

The component grids mentioned in Section 2.1 are specified to describe the domain and its boundaries. Next, we will discuss the use of adaptive grids which are created

and destroyed during the course of the computation in order to maintain sufficient accuracy throughout the entire domain.

During this process, we will create $L-1$ additional refinement levels of composite grids G_l on top of the base grid G_0. So the whole grid system can be written as follows:

$$\bigcup_{l=0}^{L-1} G_l \tag{9}$$

where at each level l, we have

$$G_l = \bigcup_j G_{l,j}. \tag{10}$$

These will have spatial discretization parameters $h_l = h_0/m^l$, where m might range from 2 to 10 depending on the problems being solved. According to Berger and Oliger [2], $m = 4$ is a reasonable choice for many hyperbolic problems. We usually let L be $2, 3$ or 4. As dictated by the nature of the problem and numerical algorithm, we maintain an appropriate relationship between the spatial and the temporal discretization parameters, h_l and k_l. In particular, for problems which are essentially hyperbolic in character, we usually use the same mesh ratio on all levels, i.e.,

$$\lambda = k_l/h_l = constant. \tag{11}$$

Another very important feature for our adaptive grids is that the grids are level nested, i.e., the region G_l is fully contained *within* the region G_{l-1}. See the shaded refined grids $G_{1,1}, G_{1,2}, G_{1,3}$ and $G_{1,4}$ in Figure 1. Since these grids have the same mesh ratio, they have the space-time structure illustrated in Figure 2. We take several time steps on the finer grid for each time step on the coarser grid.

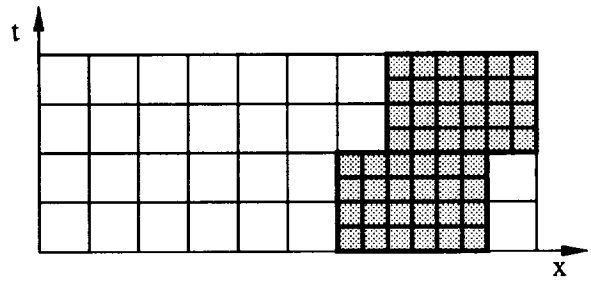

Figure 2: Space-time grid structure

We generate the refinement grids in response to computed estimates of local truncation error. If the solution is sufficiently smooth, a variant of Richardson extrapolation, call *step-doubling*, can be used to estimate the local truncation error τ_h at each

point of the grid. For simplicity, we consider the one-step explicit difference scheme described in equations (4) to (7). We define the operator Q_h as follows:

$$Q_h u_h(x,t) = L_h u_h(x,t) + k f_h(t). \tag{12}$$

If the solution is smooth enough, then the local truncation error can be written as

$$\begin{aligned} k\tau_h(x,t) &= u_h(x,t+k) - Q_h u_h(x,t) \\ &= k(k^{q_1} a(x,t) + h^{q_2} b(x,t)) + kO(k^{q_1+1} + h^{q_2+1}) \\ &= k\tau + kO(k^{q_1+1} + h^{q_2+1}) \end{aligned} \tag{13}$$

where the leading term is denoted by $k\tau$. If u is smooth enough and we take two time steps with the operator Q_h, to leading order the error is $2k\tau$,

$$u_h(x,t+2k) - Q_h^2 u_h(x,t) = 2k\tau + kO(k^{q_1+1} + h^{q_2+1}). \tag{14}$$

Let Q_{2h} be the same difference operator as Q_h but based on mesh widths of $2h$ and $2k$. Also, assume the order of accuracy in time and space are equal, $q_1 = q_2 = q$. Then

$$\begin{aligned} u_h(x,t+2k) - Q_{2h} u_h(x,t) &= 2k((2k)^q a(x,t) + (2h)^q b(x,t)) + O(h^{q+2}) \\ &= 2^{q+1} k\tau + O(h^{q+2}). \end{aligned} \tag{15}$$

Subtract equation (14) from equation (15) and use equation (13) to obtain

$$\tau_h(x,t) = \frac{Q_h^2 u_h(x,t) - Q_{2h} u_h(x,t)}{k(2^{q+1} - 2)} + O(h^{q+1}). \tag{16}$$

The restriction that the accuracy in time and space is the same is not a severe one. For details see Berger [1]. For nonsmooth solutions we no longer have an accurate error estimate. However, the Richardson estimates still provide a good criterion for refinement since the estimates will be large near a singularity.

2.3 Adaptive Grid Generation and Integration

With the basic background mentioned above, we will now explain the adaptive grid generation and integation processes. It is important to organize the data structure for the adaptive composite grid in terms of connected components, i.e., connected sets of component grids at each level. See Figure 3 for the tree representing the grid structure in Figure 1. Details on implementation can be found in [1] and [2].

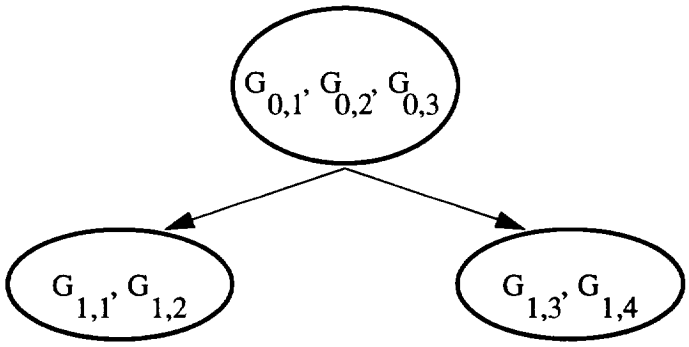

Figure 3: Tree structure of connected components

Evaluation of h_0. Having begun with an estimated initial value of h_0, we perform a trial integration on this grid and estimate the error. If more than a given fraction, say 1/2, of the grid fails to meet the error criterion and needs to be refined, we refine the whole grid with h_0 reduced by a factor of m, repeating this process if necessary. (This overall refinement is also done during the course of the computation if necessary.) The fraction 1/2 is used because of the overhead associated with grid refinement. It has been ascertained experimentally that about 3/4 of a domain can be refined (with the associated overhead) at the same computational cost as needed to compute on the entire refined domain without the adaptive method's overhead. Once h_0 is established, the other h_l are defined in terms of it.

Time integration. The solution is advanced in time as follows. The basic operation is to solve on a connected component. A time step of k_0 is first taken on G_0, and then each grid G_l, $l = 1, 2, \ldots, L-1$ is in turn advanced by k_l, connected component by connected component. Then G_{L-1} is brought up to $t + k_{L-2}$ and so on, until all of the grids are brought up to $t + k_0$. Whenever a refinement is brought up to the time level of the next coarser grid, the point values on the coarser grid under the refinement are replaced by the solution on the refinement. Values of the solution on the interior edges of connected components at each level are obtained by interpolating in time the values already obtained on the next coarser grid. It should become clear at the end of Section 3 that this interpolation needs to have a certain order of accuracy to maintain the optimal rate of convergence.

Grid modification. Local truncation errors are estimated by the process of step-doubling at fixed numbers of time steps on each grid level, usually every 4 to 8 steps depending on the size of buffer zone mentioned below. Points at which

the error criterion is near violation are flagged, clusters of the flagged points are formed, and refinements overlaying the clusters are constructed. Once a new component grid is defined, the solution at previously refined points is retained, and the solution at all other points is obtained by interpolation accurate to a certain order from the parent grid. (At the initial time, in contrast, the solution at each refinement level is obtained from the initial data u_0.) The refined grids are constructed with buffer zones around the flagged points, sufficiently wide so that a phenomenon requiring refinement cannot move out of the refined area before the next regridding. This is necessary to maintain accuracy. The grids are *moved* by constructing a new grid and deleting the old one, see Figure 2. Each refinement level is in turn refined as *necessary* in the same manner, until the finest level has been reconstructed.

3 Stability, Convergence and Error Estimation

3.1 Stability

As mentioned in the beginning of section 2, the original and discretized problems we deal with are formulated in equations (1) - (3) and equations (4) - (6), respectively. Once again, we recall that subscripts h and k are used to represent the discretized domains, operators and variables. Our goal is to use CAG's to compute the solution such that the error is bounded by a given tolerance ϵ, i.e.,

$$||e_h(T)||_{\Omega_h} \leq \epsilon \tag{17}$$

where $||\cdot||_{\Omega_h}$ is some discrete norm and $e_h(t)$ is the discrete error function, both to be defined later.

Before discussing the discrete case, we want to assume that the original initial-boundary problem is *well-posed.* The well-poseness of a large class of problems is discussed in Kreiss and Lorenz [5]. For hyperbolic and parabolic problems, we use the following definition.

Definition 2: The initial-boundary value problem defined by equations (1) - (3) is well-posed if there exists an unique solution which satisfies the following estimate

$$||u(\cdot,t)||_\Omega^2 \leq Ke^{\alpha t}(||u_0(\cdot)||_\Omega^2 + ||f(\cdot,\cdot)||_{\Omega\times[0,t]}^2 + ||b(\cdot,\cdot)||_{\partial\Omega\times[0,t]}^2) \tag{18}$$

where K and α are constants, $t \in [0,T]$ and the norms are defined in the usual way:

$$||u(\cdot,t)||_\Omega^2 = \int_\Omega |u(x,t)|^2 dx \tag{19}$$

$$||b(\cdot,\cdot)||^2_{\partial\Omega\times[0,t]} = \int_0^t \int_{\partial\Omega} |b(x,\tau)|^2 ds\, d\tau \tag{20}$$

$$||f(\cdot,\cdot)||^2_{\Omega\times[0,t]} = \int_0^t \int_{\Omega} |f(x,\tau)|^2 dx\, d\tau \tag{21}$$

and $|\cdot|$ is the standard L_2 norm in R^n, i.e., $|u|^2 = u^*u$.

In order to achieve our goal, some stability results, which are similar to the well-poseness of the original problem, are needed for the finite difference schemes. In his report, Pelle Olsson [6] used energy methods and summation by parts (a discrete version of integration by parts) to establish stability estimates on uniform grids for the semi-discretized case, i.e., when only the spatial domain is discretized. The spatial discretizations used are centered differences with various orders of accuracy. To overcome the difficulty of general boundary conditions, some projection matrices are used so that the solution can be projected into the solution space where the boundary conditions are satisfied. These results hold for a large class of linear problems, e.g., hyperbolic, parabolic and some mixed types. Although only 2-D problems were considered in his report, the results can be easily generalized to higher dimensions. We assume the following stability estimates holds for our finite difference methods on the *uniform* mesh. This is consistent with the results of Olsson [6].

Definition 3: A finite difference method for the initial-boundary value problem (equations (4) to (6)) on a uniform grid is stable if

$$||v_h(t)||^2_{\Omega_h} \leq Ke^{\alpha t}(||u_{0_h}||^2_{\Omega_h} + ||f_h||^2_{\Omega_h\times[0,t]_k} + ||b_h||^2_{\partial\Omega_h\times[0,t]_k}) \tag{22}$$

where K and α are independant of k and h, $t \in [0,T]$ and the discrete norms are weighted L_2 norms in general. Here we can assume they have the following form:

$$||v_h(t)||^2_{\Omega_h} = \sum_{x_j\in\Omega_h} h^d |v_h(x_j,t)|^2 \tag{23}$$

$$||b_h||^2_{\partial\Omega_h\times[0,t]_k} = \sum_{x_j\in\partial\Omega_h, t_i\in[0,t]_k} kh^{d-1} |b_h(x_j,t_i)|^2 \tag{24}$$

$$||f_h||^2_{\Omega_h\times[0,t]_k} = \sum_{x_j\in\Omega_h, t_i\in[0,t]_k} kh^d |f_h(x_j,t_i)|^2. \tag{25}$$

Here we recall d is the dimension of domain Ω. And we assume Ω_h is a uniform rectangular grid in the above definition. In general the underlying discrete innerproduct must be modified, see Olsson [6]. In order to simplify the notation, a single index

j is used for the grid points in d dimensions. For the same reason, we use the same K and α in Definition 2 and 3. In fact, they are not necessarily the same constants. However, in order to approximate the original problem better, it is desirable to have the α in the discrete case to be close to the one in original problem. This is refered to as *strictly stable* in Olsson's report. It can be shown that the two α's are equal up to an error of order $O(h)$ in his estimates.

Before deriving the error estimates for CAG methods, another assumption is necessary. We will assume $K = 1$, where K is the coefficient in equation (22). According to the stability results of Olsson, $K \geq 1$. Let's see what happens if $K > 1$. Since adaptive grids are used, the stability bound in Definition 3 is used whenever a regridding process is carried out. We consider a simple case with homogeneous boundary conditions and zero forcing term. Assume that we regrid at every time step, then we get

$$\begin{aligned} ||v_h(k)||^2_{\Omega_h} &\leq K e^{\alpha k} ||v_h(0)||^2_{\Omega_h} \\ ||v_h(2k)||^2_{\Omega_h} &\leq K^2 e^{2\alpha k} ||v_h(0)||^2_{\Omega_h} \\ &\vdots \end{aligned}$$

So at time T, the growth bound looks like

$$||v_h(T)||^2_{\Omega_h} \leq K^{T/k} e^{\alpha T} ||v_h(0)||^2_{\Omega_h}$$

This bound is useless unless $K \leq 1 + O(k)$. However, we can always absorb K into the term $e^{\alpha t}$ with a new α if $K \leq 1 + O(k)$. Therefore, we take $K = 1$. Fortunately, many problems we are interested in have this nice property since they describe physical phenomena which conserve energy.

3.2 Convergence

Now, we are ready to define Ω_h in the case of CAG's. We use the notation G to represent this discretization of the original domain. Given a composite component grid, which is defined in Section 2 and may contain several levels of subgrids, the domain where the computational solution vector v_h is defined can be constructed in the following way. Suppose there are L levels of subgrids $G_0, \ldots,$ and G_{L-1}, and every component grid $G_{l,j}$ is a regular grid. Then v_h is defined spacially on the set

$$G = G_{L-1} \bigcup \left(\bigcup_{l=1}^{L-2} (G_{l-1} - G_l) \right) \tag{26}$$

where $G_{l-1} - G_l$ is the complement operation. This set is well defined in the 1-D case since we enforce $G_l \subset G_{l-1}$ and there is no overlapping among the subgrids of the same level. But in higher dimensions there may be overlapping areas in G_l. The above definition is still valid if the solution vector is uniquely defined on each G_l. All the components in G_l will have errors of the same magnitude. So the solution vector can be defined using any one of the grids in the overlapping area. Suppose G_{l,j_1}, G_{l,j_2}, ..., G_{l,j_n} have an overlapping area, where $j_1 < j_2 < \ldots < j_n$. Then the solutions of G_{l,j_1} are used in the overlapping area. In other words, by using the smallest-indexed grid in the overlapping area, we can uniquely define the solution vector on every level. Then, equation (26) is well defined in higher dimensions and v_h is defined on a piecewise uniform mesh. We also notice that the set G is a function of time, i.e., $G = G(t)$, because it may change whenever a regridding process is done at any level. By this observation, we can define v_h on the interval $[0, T]$. In fact, v_h is a piecewise constant solution function on $[0, T]$. Its solution values change only when a integration or regridding process is carried out at any level. Because G is a piecewise uniform grid, we can define ∂G to be the union of all the boundaries of these uniform grids. The discrete norms defined in equations (23) to (25) can be easily generalized as follows:

$$||v_h(t)||^2_{G(t)} = \sum_{x_j \in G(t)} h^d_{l(j)} |v_h(x_j, t)|^2 \tag{27}$$

$$||b_h||^2_{\partial G(T)\times[0,T]} = \sum_{t_i \in [0,T]_k} \sum_{x_j \in \partial G(t)} k_{l(i)} h^{d-1}_{l(j)} |b_h(x_j, t_i)|^2 \tag{28}$$

$$||f_h||^2_{G(T)\times[0,T]} = \sum_{t_i \in [0,T]_k} \sum_{x_j \in G(t)} k_{l(i)} h^d_{l(j)} |f_h(x_j, t_i)|^2 \tag{29}$$

where $h_{l(j)}$ is the mesh szie of the grid to which the grid point x_j belongs. For simplicity, we assume the mesh sizes in all directions in a component grid are equal. Now we define the solution error e_h on the same domain of v_h as follow:

$$e_h(t) = v_h(t) - u_h(t) \tag{30}$$

where u_h is the projection of the exact solution onto G.

On each section with uniform grid $G_{uniform}$ and time interval $[t_0, t_1]$, the error e_h satisfies following error equations:

$$e_h(t + k) = L_h\, e_h(t) + k\, \tau_{int} \quad on \quad G_{uniform} \times [t_0, t_1], \tag{31}$$

$$e_h(0) = e_{0h} \quad on \quad G_{uniform}, \tag{32}$$

$$B_h\, e_h(t) = \tau_{bdry} \quad on \quad \partial G_{uniform} \times [t_0, t_1]. \tag{33}$$

Since linear problems are considered here, the error equation is the same as the discrete equation for v_h except the forcing and boundary terms are replaced by the

local truncation errors. Thus, the growth bound for e_h on the piecewise uniform grids can be obtained by applying the stability results in Section 3.1.

Next we are going to prove the error estimation for CAG methods using induction on the number of grid levels. First we consider the case of only one level of grids. Since it is piecewise uniform, we can use the result in equation (22) on each piece of uniform grid. Also we bound the error on grid G by the summation of the errors on the pieces of uniform grid because of the way we define G and the norm $||\cdot||_G$. Assume the local truncation errors of the numerical method used have the accuracy of order p at interior and boundary points. The initial values are also assumed accurate to order p. Then from equation (22), it is easy to see following error bound holds:

$$||e_h(T)||_G^2 \lesssim C_0 e^{\alpha T} h_0^{2p} \tag{34}$$

where $\lesssim$ means "asymptotically less than", since only the leading terms of the truncation errors are estimated here.

Now we assume that equation (34) holds for L levels of grids. Next we will show it for $L+1$ levels. Because of the way we construct the subgrids which are nested in their parent grids, we only need to consider two levels. By induction, we know the error on the finer of the two levels is of order $O(h_1^p)$, so we can always write it as $O(h_0^p)$. Assume the computation starts at $t = t_0$, we have the following error bound until we regrid at $t = t_1$,

$$||e_h(t_1)||_G^2 \leq e^{\alpha(t_1-t_0)}(||e_h(t_1)||_G^2 + ||\tau_{int}||_{G\times[t_1-t_0]}^2 + ||\tau_{bdry}||_{\partial G\times[t_1-t_0]}^2) \tag{35}$$

where τ_{int} is the local truncation error in the interior points, and τ_{bdry} is the truncation error introduced at the boundaries, which include the exterior boundaries ∂G^{ext} and the interior boundaries ∂G^{int}. Suppose a pth order method is used in the interior points. On the exterior boundaries and the interior boundaries, suppose we use qth and rth order methods, respectively. Then the following estimates are obtained:

$$||\tau_{int}||_{G\times[t_1-t_0]}^2 \lesssim C_1 h_0^{2p}\mu(G)(t_1-t_0), \tag{36}$$

$$||\tau_{bdry}||_{\partial G\times[t_1-t_0]}^2 \lesssim C_2 k_0^2 h_0^{2q}\mu(\partial G^{ext})(t_1-t_0) + C_3 k_0^2 h_0^{2r}\mu(\partial G^{int})(t_1-t_0) \tag{37}$$

where $\mu(\cdot)$ is a measure of the grid G and its boundaries which are defined as follows:

$$\mu(G) = \sum_{x_j\in G} h_{l(j)}^d, \tag{38}$$

$$\mu(\partial G) = \sum_{x_j\in \partial G} h_{l(j)}^{d-1}. \tag{39}$$

At $t = t_1$, the regridding process is executed on this regular grid. Another type of error τ_{init}, which is due to the initialization of the subgrids using interpolation, is

introduced. Let's assume this process has error of $O(h_0^s)$. Thus, after the regridding process at $t = t_1$, we will have following error estimate:

$$\begin{aligned} ||e_h(t_{1+})||_G^2 &\leq ||e_h(t_1)||_G^2 + ||\tau_{init}||_G^2 \\ &\lesssim ||e_h(t_1)||_G^2 + C_4 h_0^{2s}. \end{aligned} \tag{40}$$

If equations (35), (36) and (37) are substituted into equation (40), and the fact that $h_l/k_l = constant$ is used throughout computation, the total error after the regridding process at $t = t_1$ is

$$\begin{aligned} ||e_h(t_{1+})||_G^2 \lesssim\ & e^{\alpha(t_1-t_0)}\{||e_h(t_0)||_G^2 + C_1 h_0^{2p}\mu(G)(t_1-t_0) + C_2 h_0^{2(q+1)}\mu(\partial G^{ext})(t_1-t_0) \\ & + C_3 h_0^{2(r+1)}\mu(\partial G^{int})(t_1-t_0)\} + C_4 h_0^{2s}. \end{aligned} \tag{41}$$

where the subscript + on the left hand side of equations (40) and (41) represents the fact that the regridding process is done at time $t = t_1$. All the terms in equation (41) are under control except $\mu(\partial G^{int})$ since we do not know how many subgrids are generated. The following lemma gives a bound on $\mu(\partial G^{int})$.

Lemma 1: Let G_0 be a regular grid in R^d. Assume that there are L levels of grids $G_0, \ldots, G_{L-1}$ with the refinement ratio $m = h_j/h_{j+1}$. Then the following asymptotic inequality holds for small h_0.

$$\mu(\bigcup_{l=1}^{L-1}\bigcup_j \partial G_{l,j}) \lesssim h_0^{-1} C(d,m,L)\mu(G_0) \tag{42}$$

where $C(d,m,L)$ is a constant depending on d, m, and L.

Proof: Since a regular grid is an union of rectangular grids, without loss of generality, we can assume that G_0 is a rectangular grid. First we bound the boundaries of G_1. According to the discussion in section 2, it can be shown that the case with largest interior boundaries as h_0 is very small for $d = 2$ is in Figure 4. So, it is easy to see that the following bound holds:

$$\begin{aligned} \mu(\bigcup_j \partial G_{1,j}) &\lesssim 2d h_0^{d-1}\frac{\mu(G_0)}{2^d h_0^d} \\ &= h_0^{-1}\frac{2d}{2^d}\mu(G_0). \end{aligned}$$

So, for each level l, $l = 1, 2, \ldots, L-1$, we have

$$\begin{aligned} \mu(\bigcup_j \partial G_{l,j}) &\lesssim h_{l-1}^{-1}\frac{2d}{2^d}\mu(G_{l-1}) \\ &\lesssim h_0^{-1}\frac{2d}{2^d}\mu(G_0)(\frac{m}{2^d})^{l-1}. \end{aligned}$$

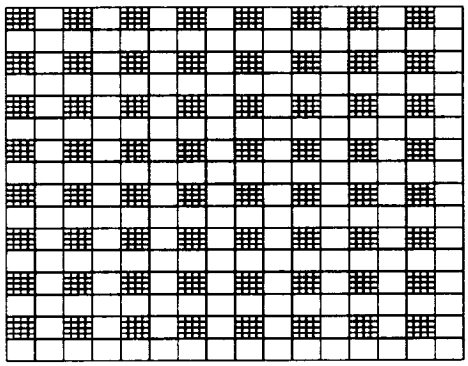

Figure 4: The case with largest interior boundaries in 2-D

Therefore, we have

$$\begin{aligned}\mu(\bigcup_{l=1}^{L-1}\bigcup_{j}\partial G_{l,j}) &\lesssim h_0^{-1}\frac{2d}{2^d}\mu(G_0)(1+(\frac{m}{2^d})+\ldots+(\frac{m}{2^d})^{L-2}) \\ &\lesssim h_0^{-1}C(d,m,L)\mu(G_0)\end{aligned}$$

where

$$C(d,m,L)=\frac{2d(1-(\frac{m}{2^d})^{L-2})}{2^d(1-\frac{m}{2^d})}. \tag{43}$$

Remark: The above lemma tells us is that, under the worst senerio, the error term involving $\mu(\partial G^{int})$ may lose one order of accuracy. However, the bound in the lemma is very pessimistic. In most situations, the number of subgrids is usually quite small. So it is very unlikely the term $\mu(\partial G^{int})$ will actually change the order of accuracy at the interior boundaries. Also we see that it is not a good idea to use too many levels with large mesh ratio, since the constant C in Lemma 1 will be very large when L is large and $\frac{m}{2^d} > 1$.

With this lemma, we can finally bound the error. We notice that for our current proof, we only need the bound for two level case, i.e.,

$$\mu(\bigcup_{j}\partial G_{1,j}) \lesssim h_0^{-1}\frac{2d}{2^d}\mu(G_0). \tag{44}$$

Suppose the worst case is considered here, i.e., the refinement and reinitialization processes are assumed as often as possible. Assuming the time interval is $[0,T]$, then

we have following error bound using equation (41) and (44):

$$\begin{aligned} ||e_h(T)||_G^2 &\lesssim e^{\alpha T}\{||e_h(0)||_G^2 + C_1 h_0^{2p} T\mu(G) + C_2 h_0^{2(q+1)} T\mu(\partial G^{ext}) \\ &\quad + C_3 h_0^{2(r+1)} T h_0^{-1}\mu(G_0) + C_4 h_0^{2s}\}. \end{aligned} \tag{45}$$

If $q = p-1$, $r = p$, $s = p$ and the initial error is order $O(h_0^p)$, then equation (45) can be written as

$$||e_h(T)||_G \lesssim Ce^{\frac{\alpha T}{2}} h_0^p \tag{46}$$

where the constant C may depand on L, the number of levels. Therefore, by the induction argument, we have proved the above error bound for CAD's. Convergence follows as $h_0 \to 0$ with a fixed maximum number of levels. We have just proved the following theorem.

Theorem 1: Suppose a well-posed initial-boundary value problem (equations (1) - (3)) is solved by a numerical method (equations (4) - (6)) using component adaptive grids with a fixed maximum number of levels of grids. Also assume that the stability condition (equation (22)) for the numerical method holds on uniform grids, and the accuracies for the numerical approximations are order p, $p-1$, and p on interior points, exterior boundaries and interior boundaries, respectivelly. Also that the interpolation process used in the CAG method has accuracy of at least order p. Then the numerical solution converges to the exact solution in the norm $||\cdot||_G$ as $h_0 \to 0$, and the order of convergence is p.

3.3 Error Estimation

From the discussion in Section 3.2, the sources and the magnitudes of various computational errors are well understood. We can implement the numerical method so that the local truncation error τ_{int} in the interior points is dominant. Then

$$\begin{aligned} ||e_h(T)||_G &\lesssim \sqrt{e^{\alpha T}\mu(G)\, T}\ \max(\tau_{int}(x,t)) \\ &\lesssim \epsilon. \end{aligned} \tag{47}$$

Therefore, if the bound δ for the local truncational error satisfies following relation

$$\max(\tau_{int}(x,t)) \lesssim \delta = \frac{\epsilon}{\sqrt{e^{\alpha T}\mu(G)\, T}}, \tag{48}$$

then the final error is guaranteed to be bounded by ϵ.

Not only can we bound the error, we can also estimate the final error. Since local truncation errors in all grid points are estimated once every several time steps, simple quadrature can be used to estimate the final error. In many applications the growth factor α is not known in advance but it is very easy to approximate α using the computed solution as the growth factor is asymptotically the same for the solution and the error.

4 Computational Examples

In this section some numerical experiments in one and two space dimensions are presented to illustrate component adaptive grid methods and the error bounds derived in section 3. The programs are written in C because of its capability for dynamic memory allocation and flexible data structures, which are crucial for our CAG methods. A SUN Sparc10 workstation with 96 Mbytes memory is used for both the 1-D and 2-D cases.

Example 1 (1-D wave equation). In this example, we compute the solution to the following wave equation in one space dimension.

$$\begin{aligned} u_t &= -u_x + \alpha u \qquad x \in [0,1], \quad t \in [0,0.4] \\ u(x,0) &= \begin{cases} 0 & \text{if } x \in [0,0.2] \bigcup [0.4,1] \\ f(x) & \text{if } x \in (0.2,0.4) \end{cases} \end{aligned}$$

where

$$f(x) = 50\ (0.01 - (x-0.3)^2 \exp(\frac{0.01}{(x-0.3)^2 - 0.01}).$$

The exact solution is a wave front traveling from left to right with speed 1 and growth rate $e^{\alpha t}$, where α is a constant. Two methods are used to solve this equation: the first order up-wind method and second-order Lax-Wendroff method. Second order Hermite interpolation is used with both methods. In Figure 5, we plot the exact solution and the computational solutions at $t = 0.4$ with $\alpha = 0$ using the Lax-Wendroff method on the coarse grid, 3-level adaptive grids with mesh ratio $m = 4$ and uniform grid with mesh size equal to the smallest of the adaptive grids. Both the solutions on the fine and 3-level adaptive grid are much better than the one on the coarse grid, which has wiggles at the left corner where the local truncation errors are large. However, the adaptive grid uses much less time than the uniform fine grid.

Next we use the results in section 3 to control the local truncation error tolerance δ according to the final error bound ϵ. All the computations here are done with

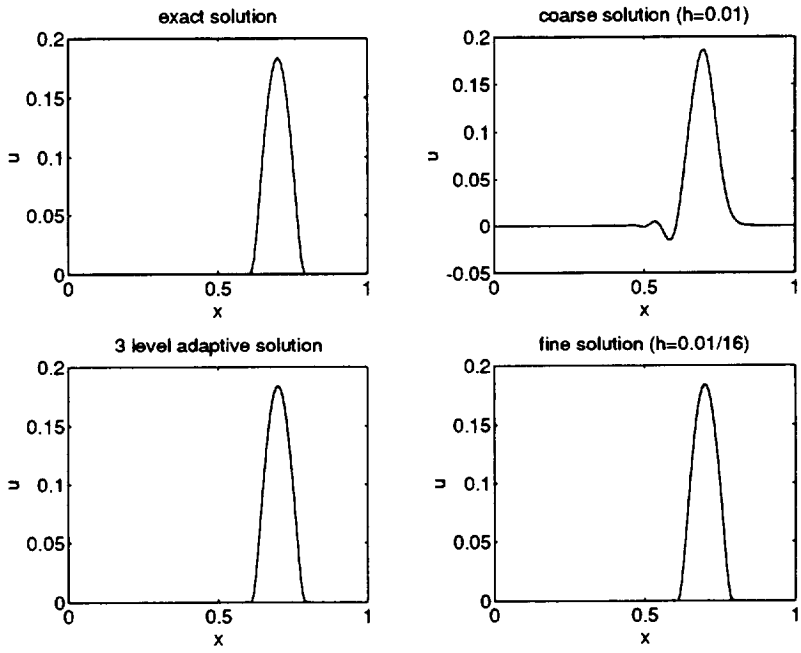

Figure 5: various computional results to 1-D wave equation at t=0.4

following parameters.

mesh ratio	4
buffer zone width	4 points
regrid every	16 steps
coarse mesh	0.01
CFL No. λ	0.9
growth factor α	0
final time T	0.36

We collected the data in Tables 1 and 2. In the first column, the final error bounds are given. The number of levels of adaptive grids used during computation is listed in the second column. The exact error and estimated error using simple quadrature are listed in columns 3 and 4, respectively. In the last two columns, we put the running times for adaptive grids and uniform grids with mesh size equal to the finest mesh

size in the corresponding adaptive grids.

Table 1

Results using the up-wind method for the 1-D wave equation

ϵ	levels	exact $\|\|e_h\|\|_G$	est. $\|\|e_h\|\|_G$	time (sec)	time using fine grid (sec)
5×10^{-3}	1	4.57×10^{-3}	4.54×10^{-3}	0.0	0.0
1×10^{-3}	3	9.37×10^{-4}	9.26×10^{-4}	0.4	4.7
5×10^{-4}	3	3.88×10^{-4}	3.82×10^{-4}	1.0	4.7
1×10^{-4}	4	9.98×10^{-5}	9.90×10^{-5}	12.9	77.0

Table 2

Results using the Lax-Wendroff method for the 1-D wave equation

ϵ	levels	exact $\|\|e_h\|\|_G$	est. $\|\|e_h\|\|_G$	time (sec)	time using fine grid (sec)
1×10^{-3}	2	2.08×10^{-4}	1.88×10^{-4}	0.1	0.4
5×10^{-4}	2	1.86×10^{-4}	1.83×10^{-4}	0.1	0.4
1×10^{-4}	3	3.68×10^{-5}	3.81×10^{-5}	0.8	7.0
5×10^{-5}	3	2.24×10^{-5}	2.15×10^{-5}	1.0	7.0
1×10^{-5}	4	7.75×10^{-6}	7.70×10^{-6}	4.6	115.0
5×10^{-6}	4	2.68×10^{-6}	2.60×10^{-6}	10.0	115.0

Several interesting facts are illustrated in Tables 1 and 2. First of all, we see that our adaptive strategy is very efficient for solving PDE's. It does efficiently generate different subgrids in response to the final error tolerances. For example, when we use Lax-Wendroff with tolerance $\epsilon = 1 \times 10^{-3}$ and $\epsilon = 5 \times 10^{-4}$, two levels of grids are used in both cases. However, in order to satisfy the final error tolerance, the two G_1's are constructed differently. This is shown in Figures 6 and 7. In the case of $\epsilon = 1 \times 10^{-3}$, two small subgrids are generated around the two corners where large local truncation errors appear. When ϵ is reduced to 5×10^{-4}, a large subgrid is created in G_1. The running times in Tables 1 and 2 show that the speedup, i.e., the ratio between the time using a uniform fine mesh and the time using an adaptive grid, increases as the final error tolerance is decreased. In other words, our adaptive strategy is more attractive when high accuracy is needed. This is because only very

small regions (the two corners in this example) need to be refined. The data also illustrate that our simple quadrature error estimation formula gives very satisfactory results. Of cause, one reason we have such accurate estimates is that the equation has constant coefficients. A more realistic variable coefficient problem is considered in the next example. As mentioned in the Section 3, the errors of interpolation and the ones at boundaries are assumed to be relatively small compared to the local truncation errors. Now we compute these two types of error explicitly and list them in Tables 3 and 4. To be consistent with the notation used in Section 3, we use $||e_{init}||_G$ to represent the subgrids' initialization errors caused by interpolation of the coarse grids' data. $||e_{int}||_G$ and $||e_{bdry}||_{\partial G}$ are used to represent the errors caused by local trucation errors on the interior points, and the errors on both exterior boundaries ∂G^{ext} and interior boundaries ∂G^{int}, respectively. Indeed, it is shown that $||e_{init}||_G$ and $||e_{bdry}||_{\partial G}$ are negligible compared to $||e_{int}||_G$. Finally we look at error estimation for different α's, since for some nonlinear problems, the error equations contains non-differentiable terms. The results for different α's using Lax-Wendroff with tolerance $\epsilon = 0.0005$ are shown in Table 5. The estimated growth factors are listed in the table. Our error control and estimation works very well for all three test cases, $\alpha = 0$, 3 and 6.

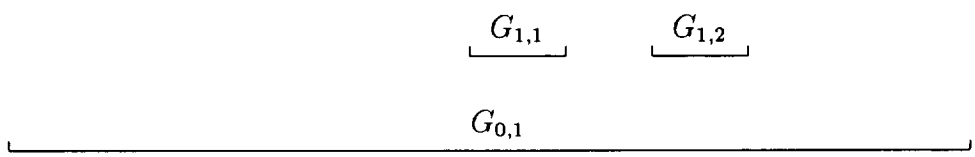

Figure 6: Adaptive grid structure for $\epsilon = 0.001$

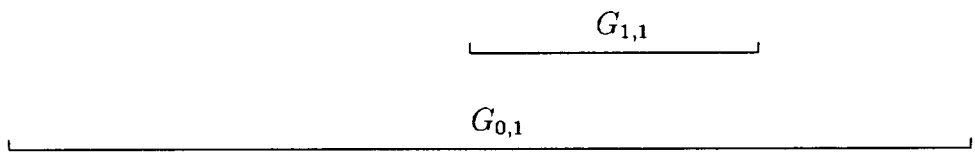

Figure 7: Adaptive grid structure for $\epsilon = 0.0005$

Table 3

Various errors using the up-wind method for the 1-D wave equation

ϵ	levels	exact $\|\|e_{int}\|\|_G$	est. $\|\|e_{int}\|\|_G$	est. $\|\|e_{bdry}\|\|_{\partial G}$	est. $\|\|e_{init}\|\|_G$
5×10^{-3}	1	4.57×10^{-3}	4.54×10^{-3}	5.25×10^{-6}	2.13×10^{-11}
1×10^{-3}	3	9.37×10^{-4}	9.26×10^{-4}	5.55×10^{-7}	3.27×10^{-6}
5×10^{-4}	3	3.88×10^{-4}	3.82×10^{-4}	7.96×10^{-7}	9.16×10^{-7}
1×10^{-4}	4	9.98×10^{-5}	9.90×10^{-5}	6.89×10^{-8}	2.41×10^{-7}

Table 4

Various errors using the Lax-Wendroff method to the 1-D wave equation

ϵ	levels	exact $\|\|e_h\|\|_G$	est. $\|\|e_{int}\|\|_G$	est. $\|\|e_{bdry}\|\|_{\partial G}$	est. $\|\|e_{init}\|\|_G$
1×10^{-3}	2	2.08×10^{-4}	1.88×10^{-4}	1.01×10^{-6}	1.92×10^{-7}
5×10^{-4}	2	1.86×10^{-4}	1.83×10^{-4}	1.04×10^{-9}	7.80×10^{-12}
1×10^{-4}	3	3.68×10^{-5}	3.81×10^{-5}	1.43×10^{-7}	3.17×10^{-7}
5×10^{-5}	3	2.24×10^{-5}	2.15×10^{-5}	7.08×10^{-8}	8.03×10^{-8}
1×10^{-5}	4	7.75×10^{-6}	7.70×10^{-6}	1.31×10^{-8}	1.01×10^{-7}
5×10^{-6}	4	2.68×10^{-6}	2.60×10^{-6}	4.65×10^{-9}	4.06×10^{-8}

Table 5

Estimates using the Lax-Wendroff method
with various α's and $\epsilon = 5 \times 10^{-4}$

α	estimated α	levels	exact $\|\|e_h\|\|_G$	est. $\|\|e_h\|\|_G$
0	-1.43×10^{-4}	2	1.86×10^{-4}	1.83×10^{-4}
3	$2 + 2.56 \times 10^{-4}$	3	2.25×10^{-4}	2.35×10^{-4}
6	$6 + 4.39 \times 10^{-4}$	4	1.05×10^{-4}	1.23×10^{-4}

Example 2 (2-D rotating cone). The rotating cone problem has been used by Berger and Oliger [2] to illustrate the adaptive grid method using rotated rectangular subgrids. The problem is

$$
\begin{aligned}
u_t &= yu_x - xu_y \qquad x \in [-1,1],\ y \in [-1,1] \text{ and } t \in [0, 3.125] \\
u(x,0) &= \begin{cases} 0 & \text{if } (x-0.4)^2 + 1.5y^2 > 0.04 \\ f(x) & \text{if } (x-0.4)^2 + 1.5y^2 < 0.04 \end{cases}
\end{aligned}
$$

where

$$f(x) = 0.25\ (1 - 25((x-0.4)^2 + y^2))^2.$$

The solution is a cone with elliptical base which rotates counterclockwise about the origin. The solution is integrated until the cone is approximately halfway through the first revolution. The Lax-Wendroff method and second order interpolation are used in this example. The boundary conditions are zero inflow and first order extrapolation at outflow. All computations are done with following parameters:

mesh ratio	4
buffer zone width	2 points
regrid every	8 steps
coarse dx	0.05
coarse dy	0.05
CFL No. λ	0.25
final time T	3.125

Snapshot views of the one subgrid at three intermediate time steps are shown in Figure 8. We should mention that this example does not fully explore the power of our stair step subgrids, since the refined regions here are elliptical, which are easily covered by rectangular grids. We expect our stair step subgrids to be more efficient when the geometries of subgrids are more complicated. Various computational results are plotted in Figure 9. It shows that the solution using a coarse grid has lots of oscillations. They are reduced dramatically if 2 levels of adaptive grids are used. If we use 3 levels, the oscillations are invisible. Also, we see that the solutions using adaptive grids are as good as those using uniform grids with the smallest mesh size of the corresponding adaptive grid, which is shown by their errors (see Table 6). Next, our error estimation developed in Section 3 is used to approximate various types of computational errors. The results are listed in Tables 6 and 7.

Since we are solving a variable coefficient problem, the estimated $||e_h||_G$ is slightly larger than the exact one. Also, like the 1-D wave equation example, the local truncation error $||e_{int}||_G$ dominates the other two types of error, $||e_{bdry}||_{\partial G}$ and $||e_{init}||_G$ (see Table 7). Finally, we estimate the efficiency of our stair step subgrids. We use

the two level case as an example. The area of the subgrid is about 16% of the original domain. So, the running time on this subgrid is about $167 \times 16\% = 26.7$ (sec). Also, we know the running time on the coarse grid is 2.7 (sec). Thus, the total time spent on other things besides integrating the grids is about $31.7 - 26.7 - 2.7 = 2.3$ (sec), which is approximately 8% of the total time for this two level adaptive computation. This percentage is less than that reported in [2], which was about 12% when using rotated rectangular subgrids. Like our 1-D example, the 2-D example also shows that the speedup increases as we decrease the final error tolerance. As mentioned earlier, we expect our stair step subgrids to be more efficient when refined regions with more complicated geometries are encountered.

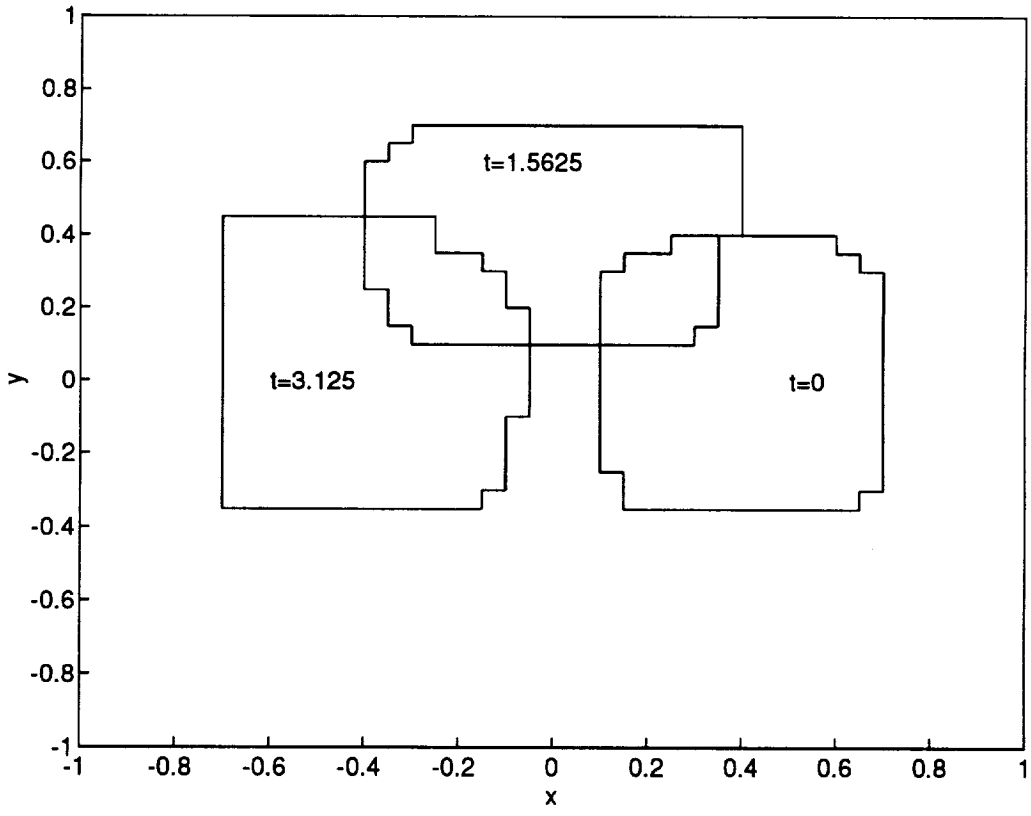

Figure 8: Stair step subgrids for the rotating cone problem

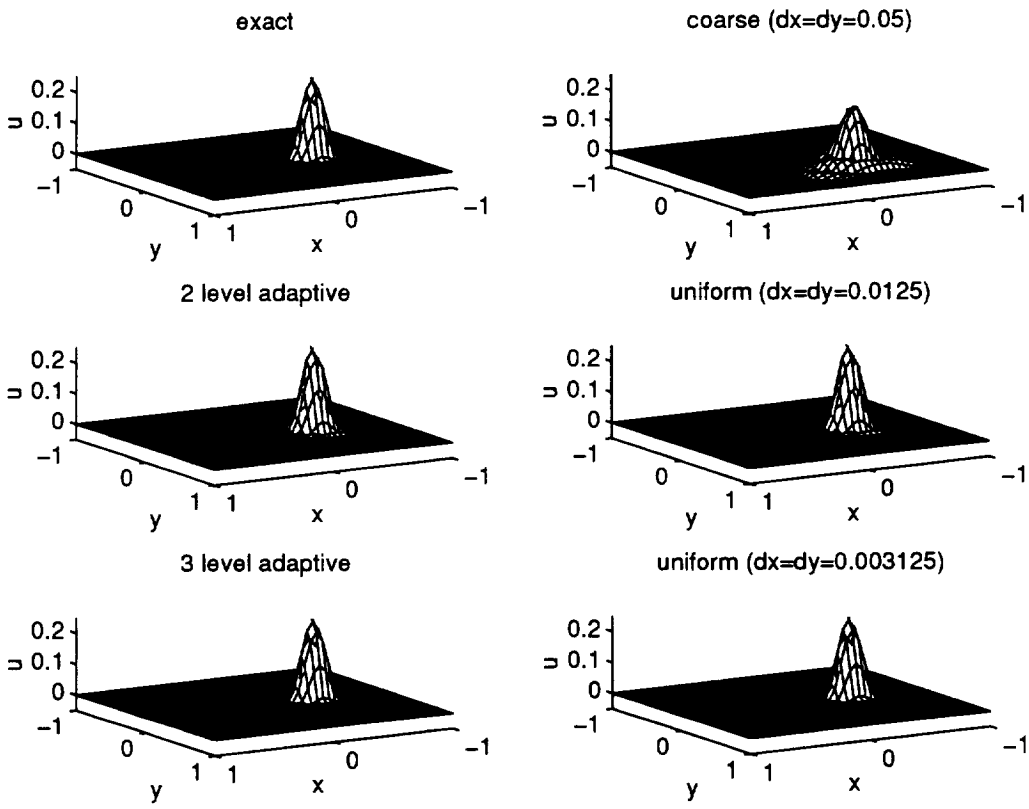

Figure 9: Solutions for the rotating cone problem

Table 6

Results using the Lax-Wendroff method for the 2-D rotating cone problem

ϵ	levels	exact $\|\|e_h\|\|_G$	est. $\|\|e_h\|\|_G$	time (sec)	time using fine grid (sec)	exact $\|\|e_h\|\|_G$ for fine grid
1×10^{-2}	2	5.11×10^{-3}	6.70×10^{-3}	31.7	167.0	5.11×10^{-3}
1×10^{-3}	3	5.32×10^{-4}	7.40×10^{-4}	1024.0	10879.0	5.27×10^{-4}

Table 7

Various errors using the Lax-Wendroff method for the 2-D rotating cone problem

ϵ	levels	exact $\|\|e_h\|\|_G$	est. $\|\|e_{int}\|\|_G$	est. $\|\|e_{bdry}\|\|_{\partial G}$	est. $\|\|e_{init}\|\|_G$
1×10^{-2}	2	5.11×10^{-3}	6.70×10^{-3}	4.77×10^{-6}	1.94×10^{-4}
1×10^{-3}	3	5.32×10^{-4}	7.40×10^{-4}	2.96×10^{-7}	2.06×10^{-5}

5 Conclusions

We have presented an algorithm for solving PDE's and estimating the final error with component adaptive grid methods. Good efficiency of such grids has been illustrated in our 2-D example. Applying recent stability results [6] for a class of numerical methods on uniform grids, we have proven the convergence of these methods for linear problems on CAG's. We obtain satisfactory final error estimates. Since local truncation error estimates are obtained during the process of grid refinement, the amount of extra work to obtain a final error estimate is very small. For linear problems, not only can the algorithm estimate the final error, it can also be used to control the tolerance for the local truncation error in terms of the given final error bound. The next challenge is to estimate errors for nonlinear problems. Although it is very hard to establish such results for general nonlinear PDE's, Olsson and Oliger [7] have devised a technique that makes it possible to obtain energy estimates for initial-boundary value problems for a class of nonlinear conservation laws. The estimates for finite difference methods on such nonlinear problems is currently under development. We think such results will help us estimate and control errors on our CAG's for a large class of nonlinear problems.

Acknowledgment

The authors wish to thank Steve Suhr for some stimulating discussion on the data structure for the 2-D stair step grids and some figures of composite adaptive grids used in this paper.

References

[1] M. BERGER, *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, Ph. D. Thesis, Stanford University, Stanford, CA, 1982 (unpublished).

[2] M. BERGER AND J. OLIGER, *Adaptive Methods for Hyperbolic Partial Differential Equations*, J. Comp. Phys., 53 (1984), pp. 484–512.

[3] B. GUSTAFSSON, H.-O. KREISS AND J. OLIGER, *Time Dependent problems and Difference Methods*, 1995 (to be published by John Wiley & Sons).

[4] G. CHESSHIRE AND W. D. HENSHAW, *Composite Overlapping Meshes for the Solution of Partial Differential Equations*, J. Comp. Phys., 90 (1990), pp. 1–64.

[5] H. O. KREISS AND J. LORENZ, *Initial-Boundary Value Problems and the Navier-Stokes Equations*, volume 136 of *Pure and Applied Mathematics*, Academic Press, INc, San Diego, CA, 1989.

[6] P. OLSSON, *Summation by Parts, Projections and Stability*, RIACS Technical Report 93.04, RIACS, NASA Ames Research Center, Moffett Field, CA, 1993 (unpublished).

[7] P. OLSSON AND J. OLIGER, *Energy and Maximum Norm Estimates for Nonlinear Conservation Laws*, RIACS Technical Report 94.01, RIACS, NASA Ames Research Center, Moffett Field, CA, 1994 (unpublished).

[8] R. G. VENKATA, J. OLIGER, AND J. H. FERZIGER, *Composite Grids for Flow Computations on Complex 3D Domains*, Proc. of the Fifth SIAM Conf. on Domain Decomp. Methods for Partial Differential Equations, (1991), pp. 605-613.

[9] R. G. VENKATA, *Three-dimensional Composite Grid Generation using Bezier Family of Curves and Surfaces*, Second SIAM Conf. on Geom. Design, (1991).