JOURNAL OF COMPUTATIONAL PHYSICS 100, 284–293 (1992)

Iterative Schemes and Algorithms for Adaptive Grid Generation in One Dimension

PRABIR DARIPA

Department of Mathematics, Texas A & M University, College Station, Texas 77843

Received June 8, 1990; revised July 1, 1991

Some iterative adaptive grid generators, developed by the author, are numerically explored in detail to assess their relative merits against conventional grid generators, based on a direct method of integration and interpolation. We find that some of these iterative adaptive grid generators are preferable to a direct method of integration and interpolation. In contrast with a direct method, appropriate use of these iterative adaptive grid generators produces adaptive grids with a smooth variation of the grid spacing ratio and resolution. All adaptive grid generators are a subclass of a more general iterative map. General features of this iterated map which are related to the function to be resolved are briefly discussed. The results obtained here supplements recent investigations on these adaptive grid generators by the author.

1. INTRODUCTION

Adaptive grids play an important role in constructing numerical solutions of a class of partial differential equations (pdes) at an optimal computational cost [1–6, 10–24]. The basic idea is to distribute a fixed number of grid points in a way that will capture the essential features of a function which usually has rapidly varying properties. Therefore some suitable strategy is required to construct such adaptive grids. Some of the properties which are desirable in an adaptive grid are [7–9]: (i) smooth variation of resolution and grid spacing ratio; (ii) the highest possible resolution in the regions of interest with a fixed number of total grid points; and (iii) a modest level of computational and algorithmic complexity. In order to generate such adaptive grids of "superior quality," a new procedure based on approximate continuous descriptions of the grid spacing ratio was invoked in [7] (see also [8, 9]).

There were four main components in [7]: (i) derivation of some new adaptive grid generators based on a continuous description of the grid spacing ratio; (ii) analysis of these grid generators; (iii) construction of adaptive grids using these grid generators; and (iv) application of these adaptive grids. There, the purpose of constructing adaptive grids was to be able to assess the merits and to study the various properties of these grids. Ideally it is necessary to find an exact solution of the iterative map (Eq. (3.7)) which uses one of these grid generators. Since it is not possible to do so for an arbitrary adaptive function $w(x)$ (see Eq. (2.1)), a numerical method was devised that will produce an almost exact solution. This numerical method was to use the iterated map recursively to generate the adaptive grid with a large number of grid points. (It is important to realize that the method requires a very good estimate of one of the unknown grid locations of the adaptive grid (see (3.7)) and this is possible only with a large number of grid points and it will not work for an arbitrary number of grid points. See the explanation following Eq. (6.1) in Section 6 of [7].) In [7], Figs. 1 through 5 were generated using this method. However, for the purpose of application it is important to be able to generate adaptive grids with an arbitrary number of grid points. In order to do so, an alternative method was devised which requires solving a set of coupled equations and it was briefly mentioned there (see Section 7 and Fig. 6 in [7] or the statement following Eq. (3.7) of this paper).

In [7], the following numerical issues have not been dealt with, which need to be addressed: (i) what are the difficulties with the direct methods (addressed in Section 2 and Fig. 3 of this paper)?; (ii) what are the algorithms to generate the adaptive grids using the new grid generators (Section 4)?; (iii) how to solve the appropriate set of equations and what are the effects of various iterative methods such as Gauss–Seidel, Jacobi, SOR, etc. (Section 5)?; (iv) what are the effects of various free parameters on the properties of the adaptive grids (Figs. 4 through 9)?. The purpose of this paper is to provide answers to these questions through numerical experiments which have not been addressed earlier. Thus, the emphasis in this paper is to describe some practical algorithms, to discuss performance of various numerical methods, and to study the effect of various parameters on the grid properties. Such a detailed numerical study is carried out here with a view toward possible applications in solving time-dependent partial differential

0021-9991/92 $5.00

equations which exhibit singular or near-singular behavior. Some applications are given in [7].

In addition, this paper discusses resolution-based grid generators (Section 3.2) and general iterated maps as grid generators (Section 3.4 and Fig. 1). Thus this paper supplements as well as complements the author's earlier work [7].

In Section 2, we discuss the main reasons behind developing this new procedure. In Section 3, we briefly review some of the essential results from [7] which are required for further development of these results in this paper. In Section 4, we discuss various possible modifications to be invoked in the theory in order to devise practical algorithms. In Section 5, we discuss the numerical results and finally we conclude in Section 6.

2. MOTIVATION

Consider one-dimensional continuous invertible mapping, given by

$$x_\zeta = c(k)\, w(x; k) \qquad \text{on} \quad x_l \leqslant x \leqslant x_u, \tag{2.1}$$

which maps the interval Ω_ζ in ζ space to the interval Ω_x in the x space. Here and below $x_\zeta = dx/d\zeta$ and $x_{\zeta\zeta} = d^2x/d\zeta^2$. The uniformly spaced ζ-grid points (grid points on the ζ axis will be referred as ζ-grid points) are mapped onto non-uniformly spaced x-grid points by this mapping which is referred to as an adaptive grid. In (2.1) k is a user-specified parameter, $c(k)$ is a scaling constant, and $w(x; k)$ is suitably chosen so that the adaptive grid captures the essential features of the function $f(w)$ ($w(x; k)$ depends on $f(x)$). We denote by h_ζ the spacing between two consecutive ζ-grid points and $x_i = x(\zeta_i)$. Without any loss of generality we can take $h_\zeta = 1$ such that the grid indices form the ζ-coordinate system. Two important properties of an adaptive grid, resolution $s(x)$ and the grid spacing ratio $r(x_i)$ are defined as

$$s(x) = d\zeta/dx \tag{2.2}$$

and

$$r_i = \frac{x_{i+1} - x_i}{x_i - x_{i-1}}. \tag{2.3}$$

Note that resolution is the Jacobian of the mapping (2.1). A simple approach of generating an adaptive grid is by direct integration of (2.1), i.e.,

$$\zeta(x) = \frac{1}{c} \int_{x_l}^{x} \frac{dx}{w(x; k)}, \qquad \text{on} \quad 0 \leqslant \zeta \leqslant N, \tag{2.4}$$

followed by interpolation. We shall refer to this method as the direct method. The direct method, conceivably the simplest of the adaptive grid generation methods, usually has several disadvantages:

(i) It usually produces grids with oscillatory grid spacing ratio (see Fig. 3). Size of the oscillations depends on the number of adaptive grid points. This may unfavorably affect the spectral radius of the iteration matrix that arises in the numerical solution of time-dependent pdes and may cause convergence problems. Most of the methods in the literature [20–23] generate adaptive grids with scant regard to this grid smoothness property. This was the main reason behind generating the new methods [7] which we explore further in this paper.

(ii) Since the function $f(x)$ has fast varying properties, either the integration procedure has to be adaptive or a large number of points will be required for this purpose.

(iii) Usually maximum resolution in this method is a slowly varying function of the number of grid points and the parameter k. If maximum resolution were a fast varying function of some suitable parameter, it would provide a way to obtain very high resolution in the areas of interest without changing the total number of grid points in the entire domain of computation. This flexibility may produce considerable savings in computational time in solving pdes without any loss in the maximum resolution of the function.

The most undesirable factor of the direct method is the oscillations in the grid spacing ratio. Some of the procedures to eliminate these oscillations are the following:

(i) Postprocessing of the grid spacing ratio to make it a smooth function and then to reconstruct the grids from this modified grid spacing ratio by some suitable procedure (see Section 4); the complete procedure involves generation of the adaptive grid by direct method, calculation of the grid spacing ratio, and subsequent smoothing to eliminate oscillations, and, finally, reconstruction of the adaptive grid from this smoothed grid spacing ratio;

(ii) developing some approximate continuous description of the discrete grid spacing ratio (2.3) and then using this to construct an adaptive grid.

This second method (ii) bypasses the grid generation procedure by the direct method and subsequent smoothing of the grid spacing ratio; it does not suffer from any of the shortcomings mentioned in the previous paragraph. In addition, this method provides increased flexibility in adjusting the maximum values of resolutions and grid spacing ratios for a fixed k and a fixed number of adaptive grid points (see Section 5 and also [7]).

It will be worthwhile to compare these new procedures and assess the relative advantages and disadvantages. In order to do so, we need the continuous description of the discrete grid spacing ratio (2.3). This has been derived in [7] which is reviewed briefly below.

3. BASIC THEORY

3.1. *Grid Spacing Ratio Based Grid Generators*

In [7], we proposed new procedures of adaptive grid generation based on the approximate continuous descriptions of the discrete grid spacing ratio (2.3). The basic idea is to use Taylor series expansions of $x(\zeta)$ in (2.3) with appropriate truncations depending on the order of accuracy desired. Some nontrivial manipulations may be required to get higher order accurate continuous descriptions of (2.3). Here we mention two second-order accurate continuous analogs of (2.3) which are given by (see [7])

$$R_1(x) = 1 + \alpha(x) \tag{3.1}$$

and

$$R_2(x) = \frac{2 + \alpha(x)}{2 - \alpha(x)}, \tag{3.2}$$

where $\alpha(x)$ is given by

$$\alpha(x) = h_\zeta \frac{x_{\zeta\zeta}}{x_\zeta}, \tag{3.3}$$

and R_1 and R_2 are continuous representations of the discrete grid spacing ratio $r(x)$. With mapping (2.1), Eq. (3.3) can be written as

$$\alpha(x; k) = c(k)\, h_\zeta w_x(x; k). \tag{3.4}$$

We take $x_0 = x_l$ and $x_N = x_u$, where N is the total number of intervals. Using Eqs. (3.1) to (3.4), we rewrite $R_1(x)$ and $R_2(x)$:

$$R_1(x) = 1 + c(k)\, w_x(x; k) \tag{3.5}$$

and

$$R_2(x) = \frac{2 + c(k)\, w_x(x; k)}{2 - c(k)\, w_x(x; k)}. \tag{3.6}$$

In fact, a hierarchy of higher order methods can be devised, if so desired, by following similar procedures. In [7], we have derived a similar formula which is accurate to first-order approximation.

Below $R(x)$ refers, in general, to any appropriate continuous approximation to (2.3) including $R_1(x)$ and $R_2(x)$. Having introduced this notation, the new grid generation equation, obtained by equating (2.3) and $R(x)$, can be written as

$$x_{i+1} = x_i + (x_i - x_{i-1})\, R(x_i), \qquad i = 1, \ldots, N-1. \tag{3.7}$$

Thus knowing any two consecutive grid point values, (3.7) can be used to calculate all preceding and successive grid point values. In practice, however, only the boundary points x_0 and x_N of the domain Ω_x are known. Therefore, the construction of adaptive grids by the finite difference equation (3.7) would require solving a system of $(N-1)$-coupled difference equations.

In this procedure, the grid spacing ratio is a smooth function by construction and the order of accuracy is implicit in the formulation. Also note that the adaptive integration and/or interpolation procedures of direct methods are not necessary. We should note that, in (2.1) and (2.4), $c(k)$ is a scaling constant which is given by

$$c(k) = \frac{1}{N} \int_{x_l}^{x_u} \frac{dx}{w(x; k)}. \tag{3.8}$$

It should be emphasized that the adaptive grid generated by (3.7) with $R = R_1$ or R_2 will be second-order accurate to the adaptive grid generated by the exact solution of (3.1), provided the value of the constant "c," obtained from (3.8), is used in calculating R_1 and R_2. However, the constant c in (3.5) and (3.6) can be allowed to be arbitrary at the loss of second-order accuracy when Eq. (3.7) is used for grid generation. As we will see, this extra degree of freedom provides some useful control over the grid attributes. In passing we should mention that with the number of grid points approaching infinity, the adaptive grid obtained from (3.7) will still converge to the exact solution given by (3.1), provided the constant c in (3.7) converges to the scaling constant (3.8). These remarks will have bearing when we devise the adaptive algorithms in Section 4.

Equation (3.7) is a very simple looking grid generation equation for any user-specific choice of $R(x)$, as long as it remains positive. Of course, $R(x)$ must be properly chosen to obtain the desired adaptivity. Two such candidates as mentioned earlier are R_1 or R_2. Choosing $R(x)$ to be either R_1 and R_2 ensures the appropriate adaptivity, the appropriate order of accuracy, and the consistency with Eq. (2.1). In view of the grid generation equation (3.7), we may refer to $R(x)$ as a grid generator. Though our study below will be restricted to these two grid generators, it is, however, not necessary to do so and one may use any suitable form of $R(x)$ in (3.7). We generalize this concept in Section 3.4.

3.2. *Resolution Based Grid Generators*

It should be noted that discrete analogs of (2.2), which is more relevant here for computing discrete resolution, are given by [7]

$$n_i = \frac{1}{x_{i+1} - x_i}, \tag{3.9}$$

to first-order approximation, and

$$n_i = \frac{2}{x_{i+1} - x_{i-1}}, \tag{3.10}$$

to second-order approximation, in the grid index space. Note that the resolution is a measure of concentration of the grid points.

It should be noted that one can also devise methods based on resolution. In fact, by setting $n_i = s(x_i)$ and using Eq. (2.1), (2.2), (3.9), and (3.10), one obtains the grid generation equations based on the resolution,

$$x_{i+1} = x_i + cw(x_i; k), \qquad i = 0, ..., N-1, \tag{3.11}$$

to first-order approximation, and

$$x_{i+1} = x_{i-1} + 2cw(x_i; k), \qquad i = 1, ..., N-1, \tag{3.12}$$

to second-order approximation. Other equivalent approximations can also be devised and may be worth exploring. However, in our numerical study we will be interested in using (3.5) and (3.6) only as grid generators in (3.7).

3.3. *Selection of Parameters*

In general, it is a good idea to invoke some free parameters in the grid generator $R(x)$ in (3.7). Then the properties of the adaptive grids generated by (3.7) can be easily controlled by adjusting these parameters. However, the parameters might have to be chosen carefully so that the mapping remains monotonic. To explain these in some detail, consider using $R_2(x)$, given by (3.6), as the grid generator in (3.7). We note that $R_2(x)$ depends on k which is an adjustable parameter. Thus the parameter k can be adjusted to control various properties of the adaptive grids [7]. Moreover, since $R_2(x)$ must be positive for monotonic mapping, we obtain from (3.6) that this is equivalent to

$$|cw_x(x; k)| < 2. \tag{3.13}$$

Similar constraints can be found using (3.5) when R_1 is used. This limits the choices for k. Of course, the admissible domain of k depends on the form of $w(x; k)$ and the constant c. As mentioned earlier, the constant c in $R_2(x)$ can be chosen arbitrarily. One appropriate choice could be the scaling constant (3.8). Note that the values that can be assigned to the constant c must also satisfty (3.13).

This increased flexibility in adjusting the constant c allows one to generate grids with different resolutions and grid spacing ratios keeping the number of grid points N and the parameter k fixed. As we will see in the next section, with a small number of grid points and with $c(k)$ given by (3.8), the adaptive grid obtained has poor resolution. This situation improves greatly upon adjusting the constant c with N and k fixed. It must be noted that the parameter k also affects these properties of the adaptive grid. In connection with the variation of grid properties with "c" we should note the following useful results.

PROPOSITION 3.1. *The grid spacing ratio $R(x; k, c)$ is an increasing (decreasing) function of c if $R \geqslant 1$ ($R \leqslant 1$).*

Proof. It follows upon differentiation of (3.5) and (3.6) with respect to c. ∎

From this proposition, it also follows that $R_{\max}$ ($R_{\min}$) is an increasing (decreasing) function of c if $R_{\max} \geqslant 1$ ($R_{\min} \leqslant 1$).

We would like to note the variation of extremal values of resolution observed numerically, as a following proposition (see Section 4).

PROPOSITION 3.2. *The maximum (minimum) resolution is an increasing (decreasing) function of c.*

Many other such useful results may be found in [7].

3.4. *Iterated Map as a Grid Generator*

In general all grid generators, including the ones mentioned in this section, can be embodied within a general iterated map,

$$x_{i+1} = h(x_i, x_{i-1}), \tag{3.14}$$

where $h(x_i, x_{i-1})$ is a suitable mapping function. This mapping function depends on two grid points, since this mapping for grid generation must conform to the two specified values of x_0 and x_N, where N, the number of grid intervals, is user-specified. Without any loss of generality we take, as before, $x_0 = 0$ and $x_N = 1$. In (3.14) $h(\,,\,)$ depends on two consecutive grid points. However, this can be further generalized by letting it depend on any two arbitrary grid points.

Constructing an appropriate $h(\,,\,)$ in (3.14) and then using it to generate the set of points $\{x_i\}$ are the main problems in adaptive grid generation. Intuitively, $h(\,,\,)$ should generate a large concentration of grid points where the function $f(x)$ needs to be resolved most. In other words, the major dynamics of the map (3.14) should take place in such critical regions. This should be the basic guideline behind choosing the map $h(\,,\,)$ even in an ad hoc manner.

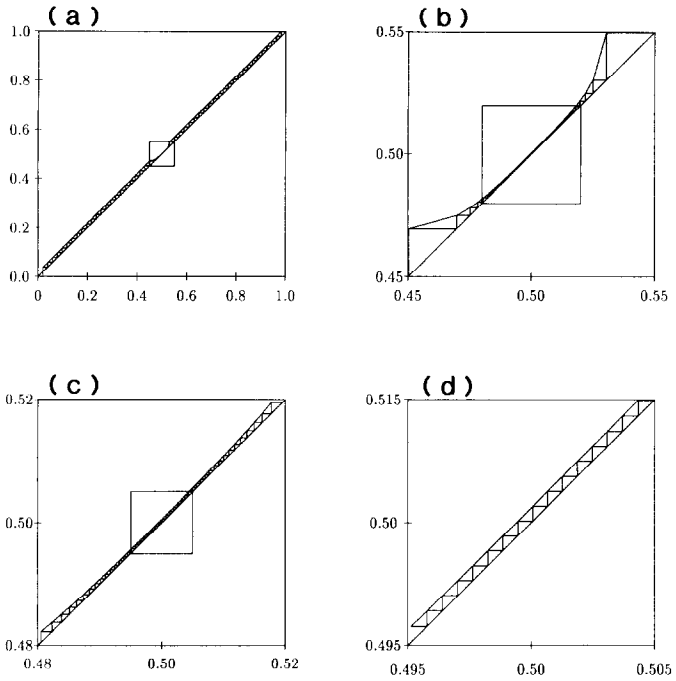

FIG. 1. Iterative map of a one-dimensional adaptive grid generator. The iterative map shown here corresponds to alg-II with $R = R_2$, $c = 0.4$, $k = 1$, and $N = 100$.

To gain further insight into this, we next consider the example of the previous section from the viewpoint of this iterated map. This viewpoint may help in generating adaptive grids without any recourse to pde formulation as in here.

Note that the grid generation equation (3.7) can be written as (3.14) with $h(\ ,\ )$ given by

$$h(x_i, x_{i-1}) = x_i(1 + R(x_i)) - R(x_i)\, x_{i-1}. \tag{3.15}$$

An adaptive grid can be generated by first knowing the two initial grid points: x_0 and x_1. We take $x_0 = 0$. Use of any arbitrary guess for x_1 in (3.15) will in general not produce $x_N = 1$. This can be made very precise as follows. From the grid generation equation (3.7) we have

$$\begin{aligned} x_{i+1} &= x_i + R(x_i)(x_i - x_{i-1}) \\ &= x_i + R(x_i)\, R(x_{i-1})(x_{i-1} - x_{i-2}) \\ &= x_i + x_1 \prod_{n=1}^{i} R(x_n). \end{aligned} \tag{3.16}$$

By induction, from (3.16) we obtain

$$x_{i+1} = x_1 \sum_{j=0}^{i} \prod_{n=1}^{j} R(x_n), \tag{3.17}$$

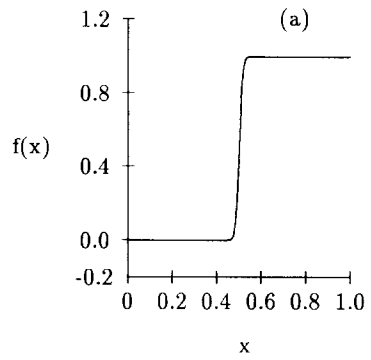

FIG. 2. Function (5.2) in physical space.

which in turn implies

$$x_N = x_1 \sum_{j=0}^{N-1} \prod_{n=1}^{j} R(x_n). \tag{3.18}$$

With $x_N = 1$, the x_1 must be correctly selected so that (3.18) is satisfied. Since it is difficult to make an exact initial guess of x_1, the grids have to be generated in an iterative manner (see next section).

In Fig. 1 we show the iterated map (3.15) corresponding to the function $f(x)$ shown in Fig. 2. This is generated by the method mentioned in the next section. Note that the major dynamics of the iterated map takes place in the region with major features in the function $f(x)$. This region is shown within the square box in Fig. 1a. Hardly much can be inferred from this. Subsequent blowups in Figs. 1b–d convey the levels of refinement in the grid size that is achievable under the simple dynamics of (3.15). Due to the severe nonlinearity in $h(\ ,\ )$, nonuniqueness in the adaptive grid cannot be ruled out.

This study should suggest that an ad hoc procedure for generating adaptive grids can easily be set up for any complicated function by selecting the function $h(\ ,\ )$ with the following properties: $h(\ ,\ )$ should be essentially parallel to the diagonal in Fig. 1 except for sudden dips in regions where the function has essential features. However, $h(\ ,\ )$ should not intersect the diagonal in Fig. 1, so that the mapping remains one-to-one.

4. NUMERICAL METHOD

In this section we describe practical algorithms to generate adaptive grids using our grid generators. We use grid generation equation (3.7) and we take $x_0 = 0$, $x_N = L$ without any loss of generality. The $(N-1)$ coupled difference equations (3.7) are denoted by

$$X = AX + F, \tag{4.1}$$

where

$$X=(x_1, x_2, x_3, ..., x_{N-1})^T,$$
$$F=(0, 0, 0, ..., 0, L)^T$$

and A is a tridiagonal matrix with

$$A_{ii}=-r(x_i), \qquad A_{i,i-1}=r(x_i), \qquad A_{i,i+1}=1.$$

The matrix equation (4.1) can be rewritten as

$$QX=F, \tag{4.2}$$

where $Q=I-A$ is a diagonally dominant matrix. Equations (3.7) and (4.2) are nonlinear due to the dependency of the matrices A and Q on the unknown solution vector X. Equation (4.2) can be solved by various iterative methods to be discussed shortly.

The algorithm for generating adaptive grids can be made very simple. We briefly describe two algorithms based on how the constant c (see (3.5), (3.6)) is set during the iterations. As mentioned earlier, the user has flexibility in adjusting the value of c, k, and N.

ALGORITHM I. Initially the grid point coordinates are guessed and then these values are updated by solving (4.2) in an iterative loop until some convergence criterion is met. In each iteration, the value of c is updated by using the following formula:

$$c(k)=\int_{x_0}^{x_1}\frac{dx}{w(x;k)}. \tag{4.3}$$

This is obtained by integrating (2.1) up to the first adaptive grid point. Note that this is equivalent to (3.8). Henceforth we shall refer to the method using this algorithm with $R=R_1$ or $R=R_2$ as alg-I(R_1) and alg-I(R_2), respectively.

ALGORITHM II. This is same as Algorithm I except that the value of c is kept constant at a user-specified value during the iterations. The values which the constant "c" can assume have inherent limitations as discussed earlier. Therefore the constant c cannot be set arbitrarily, otherwise the iteration will not converge. A practical guide to select c is given later in Section 5. Henceforth we shall refer to these algorithms using $R=R_1$ or $R=R_2$ as alg-II(R_1) and alg-II(R_2), respectively.

It must be stressed at this point that some care should be exercised in selecting the criterion for convergence; otherwise the error in the grid location can be more than the smallest size of the adaptive grid which is not known a priori. One can dynamically set it, say at 1 % of the smallest grid spacing of the current iteration level.

5. NUMERICAL STUDY

The numerical study is aimed at comparing the various methods (direct, alg-I(R_1), alg-I(R_2), alg-II(R_1), and alg-II(R_2). Though our calculations have been extensive, we present only a few results for the sake of brevity.

In this section we present numerical results for a typical model problem. In the numerical experiments below, we use the first derivative adaptive function:

$$w(x;k)=(k+f_x^2)^{-1/2}. \tag{5.1}$$

In (5.1), $f(x)$ is the function to be resolved and the parameter k is a user-specified parameter. The value of k

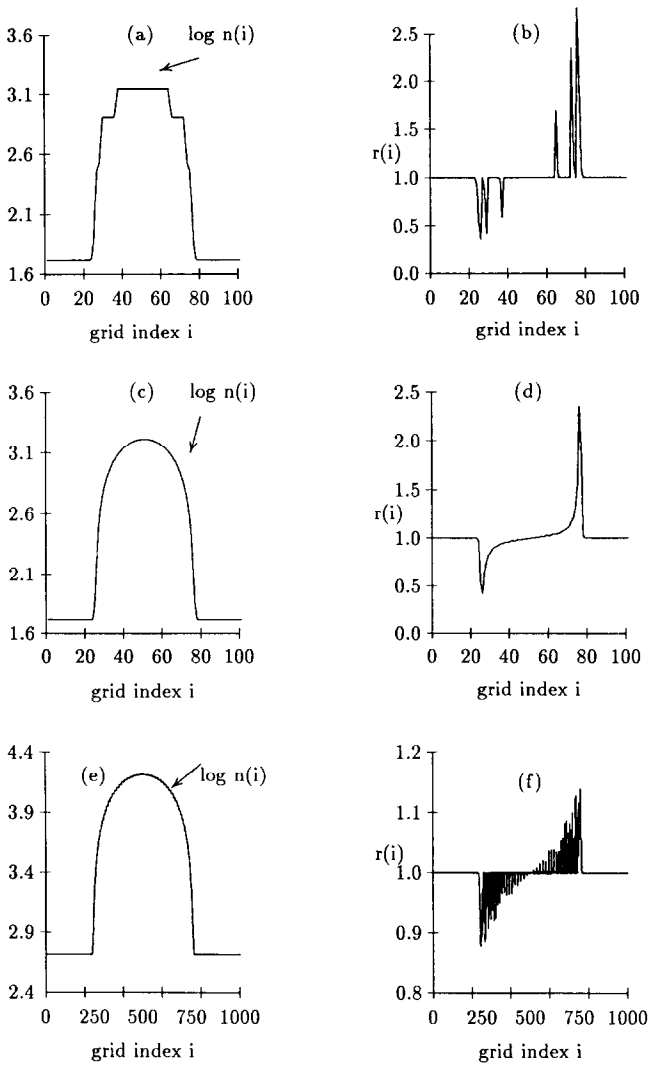

FIG. 3. Effect of adaptive and nonadaptive integration on the resolution (n) and grid spacing ratio (R) in the direct method: Figures (a) and (b) show the effect of nonadaptive integration ($N=100$); (c) and (d) show the effect of adaptive integration ($N=100$); (e) and (f) show the effect of adaptive integration with a large number of adaptive grid points ($N=1000$).

should be chosen carefully (see also [7]). From (5.1) we find that $w(x;k) \approx 1/\sqrt{k}$ in regions where $f(x)$ is nearly constant and $w(x;k) \approx 1/|f_x|$ in regions of rapid variation in $f(x)$. This and the mapping $x_\zeta = cw(x;k)$ imply that grid concentration will decrease with decreasing k in regions where the function is essentially constant. Thus smaller values of k will provide better resolution for a fixed number of grid points. However, k cannot be chosen small enough to violate (3.8). Usually, k is chosen to be one in most practical instances. More elaborate analysis in this regard can be found in [7].

As a test case, we consider the function

$$f(x) = [1 + \mathrm{sgn}(x - x_c)\{1 - \exp(-aX^2 + \tfrac{1}{2})\}]/2, \qquad (5.2)$$

where $X = 1/\sqrt{(2a)} + |x - x_c|$, $a = 2000$, $x_c = \frac{1}{2}$, $\mathrm{sgn}(x - x_c) = 1$ if $x \geqslant x_c$ and $\mathrm{sgn}(x - x_c) = -1$ if $x < x_c$. This function is shown in Fig. 2. The Gaussian term contributes smooth but rapid variation for the function values in the vicinity of the center of the domain. The higher the value of a in (5.2), the steeper the function is.

To explore the "worthiness" of the new grid generators based on the grid spacing ratio, we generate the adaptive grids by various methods. Figure 3 shows the adaptive grid properties generated by the direct method. Nonadaptive integration of (2.4) produces oscillations (Fig. 3b) which are partially eliminated by adaptive integration (Fig. 3d). (In Figs. 3a through 3d, the number of adaptive grid points is 100. The number of grid points used in the integration is 100 for Figs. 3a through 3d and 1000 for Figs. 3e and 3f.) However, with a large number of adaptive grid points, oscillations still persist, even after the integration is carried out with a large number of grid points (see Fig. 3f).

In Fig. 4 we show the adaptive grids obtained with alg-I($R = R_2$) and $N = 100$. Resolution and grid spacing ratios are seen to be smooth functions. (We should recall that this non-oscillatory behavior is implicit by construction.) Note that resolution and grid spacing ratios are relatively insensitive to the value of k used. Comparing with the direct method, we see that by switching from direct to iterative alg-I($R = R_2$), there is a loss of maximum resolution at the gain of no-oscillation in R.

Neither of these methods offer the ideal combination: very high resolution and non-oscillatory grid spacing ratio. A remedy to this situation is to allow "c" in alg-I to be user-specified, i.e., to use alg-II. In Fig. 5, we show that the effect of varying the constant "c" on the grid properties for fixed $k = 1$. Note that a drastic gain in the maximum resolution can be obtained by this simple strategy. Curves I in Fig. 5 are essentially the ones close to the results obtained using alg-I (see Fig. 4). Increasing the value of "c" increases the resolution and grid spacing ratio. In alg-II, we can also

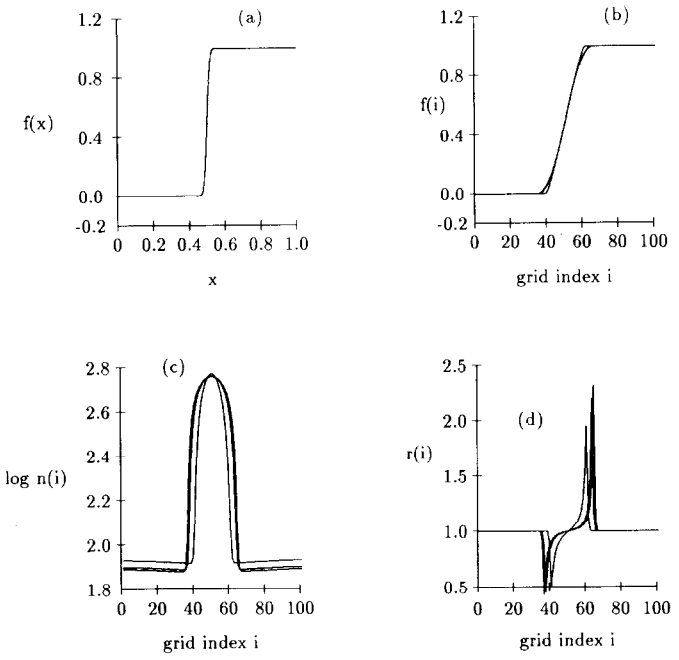

FIG. 4. Effect of k on grid properties in alg-I($R = R_2$): properties of the adaptive grid as a function of the constant k with fixed $N = 100$. (a) function (5.2) in the physical space; (b) function (5.2) in the index space; (c) discrete resolution; (d) grid spacing ratio. Figures (b), (c), and (d) are in the reference frame. The values of the constant k for three different families I, II, III of curves in each of Fig. 1b, c, and d are 0.5, 1, 10, respectively. Curves I and II are almost identical in each of (b), (c), and (d).

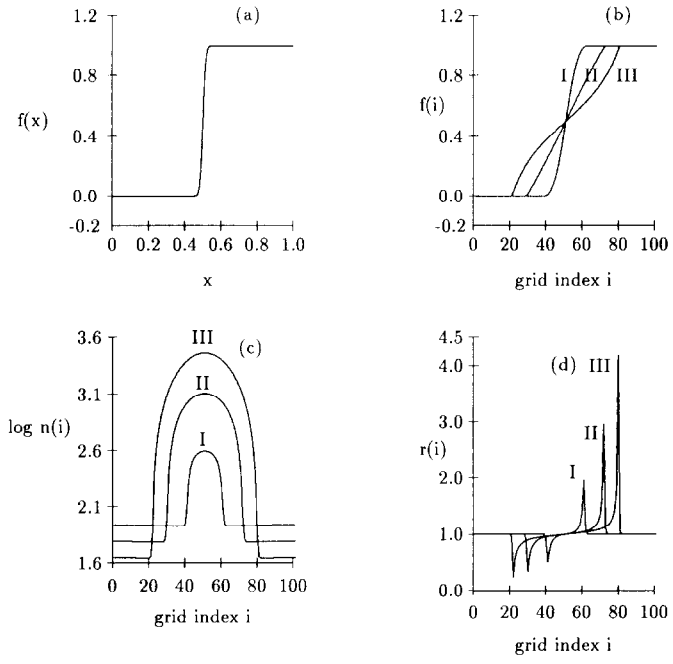

FIG. 5. Effect of the constant c in alg-II($R = R_2$): adaptive grid as a function of the constant c with fixed $N = 100$ and fixed $k = 1$. (a) function (5.2) in the physical space; (b) function (5.2) in the index space; (c) the discrete resolution; (d) grid spacing ratio. Figures (b), (c), and (d) are in the reference frame. The values of the constant c for three different families I, II, III of curves in each of Fig. 1b, c, and d are 0.01, 0.02, and 0.04, respectively.

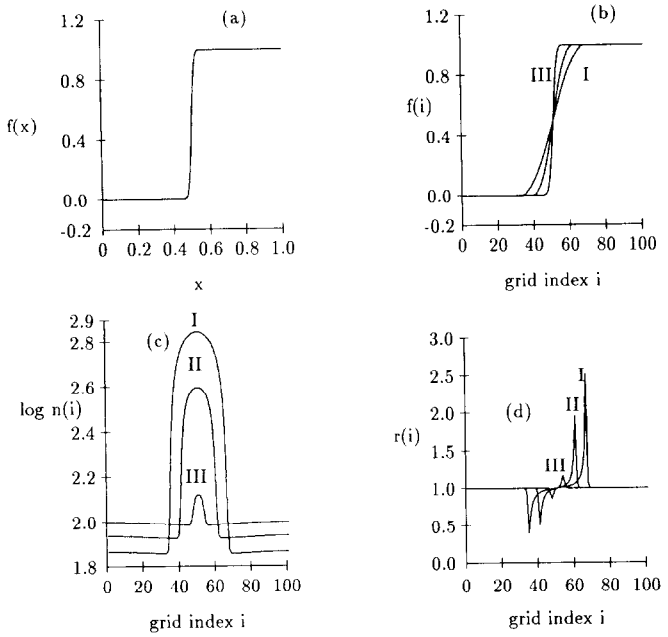

FIG. 6. Effect of k on grid properties in alg-II($R = R_2$): properties of the adaptive grid as a function of k with fixed $N = 100$ and fixed $c = 0.01$. (a) function (5.2) in the physical space; (b) function (5.2) in the index space; (c) discrete resolution; (d) grid spacing ratio. Figures (b), (c), and (d) are in the reference frame. The values of the constant k for three different families I, II, III of curves in each of Fig. 1b, c, and d are 0.5, 1, 10, respectively.

allow k to vary. The results of such a study are shown in Fig. 6. We find that better resolution is obtained with smaller values of k.

From Fig. 5 and 6, we note that an increase in "c" has the same effect as a decrease in k on the properties. Therefore, the effect of an increase in k can be canceled by decreasing c by an appropriate value, thereby producing marginal changes in the grid properties. This is what happens in alg-I (see Fig. 4). There the constant c, computed by (4.3), is an increasing function of k due to Eq. (5.1) (see Fig. 7).

Figure 8 compares the effect of k in these three methods. For alg-II($R = R_2$), we have used $c = 0.04$. Note that the

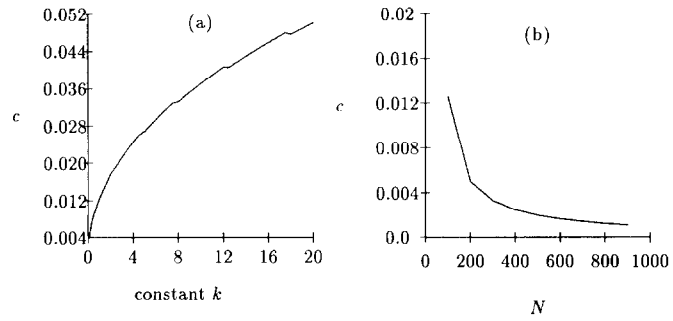

FIG. 7. Effect of the constant k and N on c in alg-I($R = R_2$).

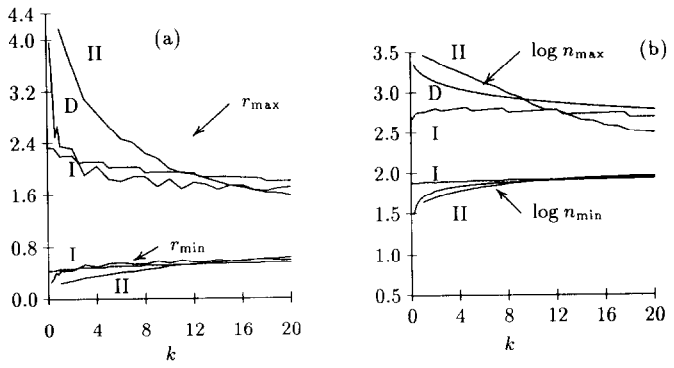

FIG. 8. Effect of k: (a) extremal values of grid spacing ratio r; (b) extremal values of resolution n: Curves D, I, II refer to the methods Direct, alg-I($R = R_2$), and alg-II($R = R_2$) with $c = 0.04$, respectively. The in-between curve in the lower branch is by the Direct method (not lettered above).

minimal values of resolution and grid spacing ratio are insensitive to the value of k and also to these methods. In contrast, the maximal values of these properties are sensitive to both the methods and the value of k (at least for $k < 0$.) The parameter k is commonly chosen to be one in the direct method. Note that with the same value of k, alg-II with $c = 0.04$ gives almost a twofold increase in resolution (Fig. 8a). This implies that for the same resolution, the number of adaptive grid points required will be less in alg-II than in the direct method. This will entail considerable savings in numerical solution of time-dependent problems, where the adaptive grids will be required at each time level.

In Fig. 9 we show the effect of varying "c" in alg-II on extremal values of the resolution and grid spacing ratio. There is usually an upper bound on "c," beyond which $R < 0$ (see Fig. 8b and (3.13)) and, hence, alg-II will not work. In spite of this limitation, alg-II can be preferable to the direct method for reasons mentioned above.

Although, the above conclusions hold in general for all N, the relative degree of advantages and disadvantages are

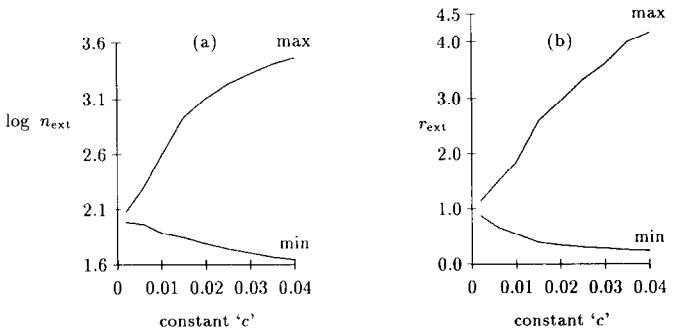

FIG. 9. Effect of the constant c in alg-II($R = R_2$): Maximum and minimum values of grid spacing ratio "r" and resolution "n" as functions of the scaling constant "c" with $N = 100$ and $k = 1$. Upper branch (lower branch) corresponds to maximum (minimum).

N-dependent because the constant c in alg-I depends on N (see Fig. 7b). To assess this quantitatively, we define the relative resolution in the following way:

$$\tilde{n} = (n_i/n_u). \tag{5.3}$$

Here n_u is the resolution with uniform grid spacing and n_i is the resolution with an adaptive grid at location "i" in the index space. Note that extremal values of $\tilde{n}$ depend on N and, hence, they are a good basis for comparing the usefulness of adaptive grids for various N. In Fig. 10 we show this within a moderate domain of N, normally used for adaptive grid generation.

Some comments on the iterative schemes that we have used in solving (4.2) should be made. The convergence rates of the iterative schemes may depend on the number of grid points, the type of the iterative methods used, the initial guess of the adaptive grid, and the values of the flexible parameters, c and k. From our extensive numerical experiments we draw some general conclusions in these matters.

We find that the convergence rate for iterative schemes using Gauss–Seidel and red–black Gauss–Seidel methods were, in general, better than SOR and Jacobi methods. In fact, we find that the l_1 and l_∞ error norm curves for Gauss–Seidel and red-black Gauss–Seidel methods are identical, for all practical purposes. The SOR method does not do as well as these methods. A multigrid method can be used to accelerate the convergence rates. In our numerical study, the initial grid in the iterative schemes was either a uniformly spaced grid or the adaptive grid obtained from a direct method using nonadaptive integration. The number of iterations required in the second case were less than in the first case.

We find that the effect of variations of k on the convergence rate is very mild. In contrast, the effect of variations of c on the convergence rate is dramatic. In general, the number of iterations is large for smaller values of c and is small for larger values of c. Fortunately, this scenario is tolerable, since the maximal value of resolution is usually a monotonic function of c (see Fig. 9).

So far in this section we have presented results that support the use of alg-II($R = R_2$) for adaptive grid generation. However, for successful adaptive grid generation by this method, it should be clear that one needs to select the values k and c carefully. We recommend using small values for k, roughly in the range of 0.2 to 0.5. However, for c, the largest possible value should be selected so that (3.2) is satisfied. (In other words, as long as grid spacing ratio remains positive.) This can, of course, be calculated using (3.2). However, one can safely avoid this and use the following procedure: use alg-I initially for a few iterations and, once the convergence rate slows down, switch to alg-II with the value of c equal to a constant.

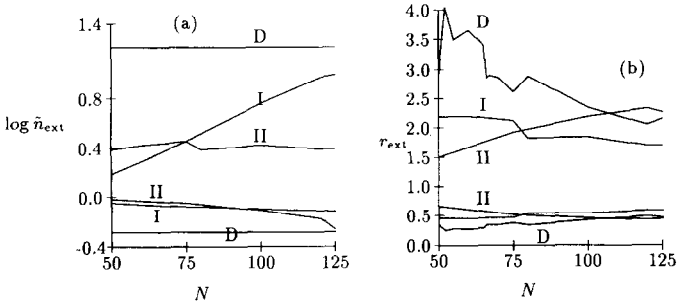

FIG. 10. Effect of N on extremal values of grid spacing ratio "r" and relative resolution "$\tilde{n}$" in the direct method (D), the alg-I($R = R_2$), and alg-II($R = R_2$) with $k = 1$. Upper branch (lower branch) corresponds to maximum (minimum) in each of the methods.

6. CONCLUSION

A new set of iterative adaptive grid generators have been introduced. We have rigorously carried out numerical experiments with some of these, while other generators can also be tested similarly to assess their "worthiness." Some applications of these adaptive grid generators in solving time-dependent pdes are given in [7].

We should note that there are many other types of adaptive grid generators which I have not mentioned here. There is a growing interest in generating adaptive grids which will suit specific needs. Here we have devised some spacing-ratio-based generators and some iterative approaches to constructing the grids using these. Some essential numerical studies have been carried out to illustrate the strengths and weaknesses of these approaches.

ACKNOWLEDGMENTS

It is a pleasure to thank one of the referees for valuable suggestions. This research in part has been supported by the NSF Grant DMS-8803669 to Texas A & M University.

REFERENCES

1. S. Adjerid and J. E. Flaherty, *SIAM J. Numer. Anal.* **23**, 778 (1986).
2. I. Babuska, J. Chandra, and J. E. Flaherty, *Adaptive Computational Methods for Partial Differential Equations* (SIAM, Philadelphia, 1983).
3. M. J. Berger, NYU Report No. DOE/ER/03077-727, 1987 (unpublished).
4. M. Bieterman and I. Babuska, *J. Comput. Phys.* **63**, 33 (1986).
5. J. G. Blom, J. M. Sanz-Serna, and J. G. Verwer, *J. Comput. Phys.* **74**, 191 (1988).
6. J. U. Brackbill and J. S. Saltzman, *J. Comput. Phys.* **46**, 342 (1982).
7. P. Daripa, *SIAM. J. Numer. Anal.* **29**, 1635 (1991).
8. P. Daripa, *Appl. Math. Lett.* **4**, 91 (1991).

9. P. Daripa, IMA Report No. 612, 1990 (unpublished).
10. S. F. Davis and J. E. Flaherty, *SIAM. J. Sci. Stat. Comput.* **3**, 6 (1982).
11. E. A. Dorfi and L. Drury, *J. Comput. Phys.* **69**, 175 (1987).
12. H. A. Dwyer, AIAA Paper 83-0449, 1983 (unpublished).
13. H. A. Dwyer, R. J. Key, and B. R. Sanders, *AIAA J.* **18**, 1205 (1980).
14. H. A. Dwyer, *AIAA J.* **22**, 1705 (1984).
15. R. J. Gelinas and S. K. Doss, *J. Comput. Phys.* **40**, 202 (1981).
16. G. W. Hedstrom and G. H. Rodrigue, "Adaptive-Grid Methods for Time-Dependent Partial Differential Equations," *Multigrid Methods*, edited by W. Hackbusch and U. Trottenberg (Springer-Verlag, New York, 1982), p. 474.
17. B. Larrouturou, *SIAM. J. Sci. Stat. Comput.* **10**, 742 (1989).
18. K. Matsuno and H. A. Dwyer, *J. Comput. Phys.* **77**, 40 (1988).
19. K. Miller, *SIAM. J. Numer. Anal.* **18**, 1033 (1981).
20. M. M. Pervaiz and J. R. Baron, *Commun. Appl. Numer. Methods* **4**, 97 (1988).
21. J. D. Ramshaw, *J. Comput. Phys.* **59**, 193 (1985).
22. J. D. Ramshaw, *J. Comput. Phys.* **67**, 214 (1986).
23. J. G. Verwer, J. G. Blom, and J. M. Sanz-Serna, *J. Comput. Phys.* **82**, 454 (1989).
24. K. H. Winkler, thesis, University of Gottingen, 1976 (unpublished).