SIAM J. NUMER. ANAL.
Vol. 28, No. 6, pp. 1635–1660, December 1991

005

A NEW THEORY FOR ONE-DIMENSIONAL ADAPTIVE GRID GENERATION AND ITS APPLICATIONS*

PRABIR DARIPA†

Abstract. The theory of a new approach to adaptive grid generation in one dimension is developed. The approach is based on approximating either the resolution or the grid spacing ratio on discrete lattice points by continuous variables. The order of accuracy of these approximations in a suitable reference frame characterizes the various methods. Approximations that are first- or second-order accurate in a suitable reference coordinate are derived in this paper. The free parameters associated with these methods provide flexibility in generating a large family of adaptive grids with smooth grid spacing ratio and high resolution. A selected group of this family of adaptive grids may prove very useful in adaptive computations of partial differential equations. The adaptive grids are numerically generated using these approximations. Numerical examples are given that exemplify the usefulness of these adaptive grids.

Key words. finite difference, adaptive grid

AMS(MOS) subject classification. 39A10

1. Introduction.

Adaptive grids are used when the solution of a partial differential equation (pde) exhibits singular or near-singular behavior. Examples occur in shock waves, traveling waves, and boundary layers. A correct selection of adaptive grids will produce a highly accurate solution at an optimal cost. However, the use of adaptive grids may increase the level of algorithmic complexity. The overall advantages of using adaptive grids in an appropriate way compensate for the inconvenience involved in generating and using these grids in the numerical solution of a pde.

There are two basic approaches to adaptive grid computations: the embedded grid method and the moving grid method (see [1]–[26]). A suitable account of these methods is given in Berger [3]. The embedded grid methods use a relatively coarse initial grid at each time level, with finer grid patches in the regions requiring refinement. The refined mesh patches generally consist of grid patterns that are similar in shape to the initial level coarse grid pattern but that have a more dense grid pattern. The refined mesh is generally obtained by subdividing the coarse grid element requiring refinement into two or more smaller elements. Berger [3] presented a locally adaptive mesh refinement algorithm which is internally conservative, provides estimates of the local errors, and provides heuristics for the automatic generation of the embedded grid. This technique includes several features for error estimation that provide consistent and efficient estimates of the errors across the entire mesh.

In the moving grid method, the grid points evolve with time in an appropriate manner so as to improve the accuracy, reliability, and robustness of the solution at an optimal cost. Flaherty and coworkers [1], [2] have developed a moving mesh strategy in their finite element code based on equidistribution of discretization error. K. Miller [21] and K. Miller and R. Miller [20] have developed a node movement scheme based on minimization of residuals. Harten and Hyman [16] have developed a moving mesh scheme in one dimension for hyperbolic equations based on averaging of local characteristic velocities. Bieterman and Babuška [4] have applied the classical method of lines to move the nodes. Dwyer [11], [13] and Matsuno and Dwyer [19] have

* Received by the editors August 27, 1990; accepted for publication (in revised form) January 29, 1991. This research in part has been supported by the National Science Foundation grant DMS-8803669 to Texas A&M University.

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

used Poisson's equation to generate adaptive grids. Dwyer, Kee, and Sanders [12] have also used one-dimensional integral to generate adaptive grids. The contributions of Sanz-Serna and coauthors [5], [26] and Ramshaw [24], [25] should also be cited. The reader is referred to Larrouturou [18] and Pervaiz and Baron [22] for some interesting applied problems using moving mesh schemes.

Here we briefly outline the idea behind the moving grid method of adaptive computations. One of the ways to describe the idea of the moving grid method of computation is as follows. A partial differential equation

$$u_t = L(x, u, u_x, \cdots) \tag{1.1}$$

is transformed into

$$u_T - u_x x_T = L(x, u, u_x, \cdots) \tag{1.2}$$

by means of the transformation $\zeta = \zeta(x,t)$, $T = t$. In (1.1) and (1.2), $L(x, u, \cdots)$ is a differential operator which may depend on x, u, u_x, u_{xx}, and higher derivatives. The mapping is carefully constructed so that in the (ζ, T) system the function $u(\zeta, T)$ is slowly varying in T and ζ.

It should be noted that the grid movement is associated with the second term in (1.2). This system is augmented by an appropriate mapping function which closes the system and couples the evolution of grid points with the evolution of the solution function u. There are some variations on this basic idea, such as explicitly decoupling the motion of the grid points and the solution variable u. The idea behind moving the grids through the formulation (1.2) is to provide smoothing of near-singular temporal development of the solution. This smoothing has the effect of allowing a larger time step in numerical solution of (1.2) and appropriate augmented equations. However, as the grids evolve there may arise problems such an grid entanglement, rapid variations in grid spacing, and oscillations in the grid spacing ratio. The grid entanglement problem may be avoided if the continuous movement of the grids is replaced by explicit generation of the adaptive grids at each instant. This amounts to setting $x_T = 0$ in (1.2). Thus, the new coordinate ζ is a function of x only where the grids are constructed to adapt some appropriate properties of the solution u at each instant. Even though this is the best possible simplification of the adaptive grid computation, the problems of rapid variation in grid spacings and oscillations in grid spacing ratio may persist. We briefly elaborate on this now.

In the numerical solution of (1.2), the discrete analogue of the system (1.2), even without the second term involving x_T, reduces to

$$Au^{(n+1)} = B(u^n), \tag{1.3}$$

which advances the solution from time level n to time level $n+1$. (The form of equation (1.3) depends on the numerical method used.) In (1.3), A is a matrix that depends on u^{n+1} for a nonlinear problem. The matrix A and the vector B in (1.3) depend on the adaptive grids, and if the grid spacings change too rapidly or the grid spacing ratio oscillates, obtaining an accurate solution of (1.3) may be difficult. For nonlinear cases, there may be convergence and/or stability problems.

This paper develops a new approach to generating adaptive grids in one dimension. The adaptive grids generated by these methods have the following special properties: (i) very high resolution, yet satisfying the desired bounds on grid spacing ratio; (ii) no oscillations in the grid spacing ratio; and (iii) flexibility in the order of accuracy in an appropriate reference plane to be made precise below. However, in this paper we consider only second-order accurate formulas exclusively for numerical purposes. The purpose of the paper is to explore the feasibility of the new idea expounded here and assess its merits in real applications. We address some theoretical issues of these methods along with some numerical justifications of our analysis. We also address applications of these adaptive grids in adaptive computations of partial differential equations.

In §2 we define the properties of adaptive grids. In §3 we construct approximations to these properties by continuous variables. Using these continuous descriptions and a suitable definition for the mapping function, various useful theorems relating grid spacing ratio, resolution, and the form of adaptivity are established in §4. In that section, various theorems address how many useful properties of the adaptive grids can be inferred directly from the function without actually generating the adaptive grids. In §5 we mention some typical forms of adaptivity most often used in practice. In §6 we briefly discuss some computational results which support our analysis of §4 and provide a meaningful guide to constructing practical algorithms. In §7 we describe such an algorithm and in §8 we illustrate the use of such adaptive grids through some numerical examples. Finally, we conclude in §9.

2. Some definitions.

Let Ω_ζ and Ω_x be intervals in ζ and x space, respectively. Consider a one-dimensional monotonic mapping $x(\zeta) : \Omega_\zeta \rightarrow \Omega_x$, with the property that uniformly spaced ζ-grid points (grid points on ζ axis will be referred to as ζ-grid points) are mapped to nonuniformly spaced x-grid points. The mapping $x(\zeta)$ will be constructed so that rapidly varying functions f in x map into slowly varying functions in ζ. The desired adaptivity is incorporated in the choice of $x(\zeta)$. Let us denote by h_ζ the spacing between two consecutive ζ-grid points and let $x_i = x(\zeta_i)$. The x-grid points constitute an adaptive grid. It is convenient to introduce the following properties.

Resolution (s). The definition of resolution is motivated by the requirement that the grid point concentration should be higher at locations of rapid variation in the function values $f(x)$. Therefore the mapping $x(\zeta)$, which depends on $f(x)$ as will be seen later, should be carefully constructed so that the regions containing rapid variations in $f(x)$ are magnified much higher than the other regions. In this sense, the resolution is synonymous with local magnification here, and hence a convenient definition for resolution seems to be

$$s(x) = \frac{d\zeta}{dx}. \tag{2.1}$$

Note that according to this definition, resolution is a continuous function.

Grid spacing ratio (r). The grid spacing ratio at the grid point x_i is defined as the ratio of the two adjacent grid spacings, i.e.,

$$r_i = \frac{x_{i+1} - x_i}{x_i - x_{i-1}}. \tag{2.2}$$

Note that $r_i > 0$ for a monotonic mapping and that it is a discrete function.

3. Approximations. Either the resolution or the grid spacing ratio can be used as the basic generator of the adaptive grids. However, these need to be approximated numerically to establish the connection between the discrete adaptive grid coordinates and the mapping $x_\zeta : \Omega_\zeta \to \Omega_x$. Below we use the notation n_i and $R(x_i)$ to denote the discrete approximation to the resolution $s(x_i)$ and the continuous approximation to r_i, respectively.

3.1. Approximations to resolution. Approximations of various orders to $s(x)$ can be obtained easily using Taylor series. As seen from (2.1), we are merely approximating the first derivative x_ζ. An appropriate first-order approximation n to s at a grid point x_i is

$$n_i = \frac{h_\zeta}{x_{i+1} - x_i}. \tag{3.1}$$

Similarly, other first-order approximations can be derived. Using Taylor series, we find an appropriate second-order approximation to $s(x)$:

$$n_i = \frac{2h_\zeta}{x_{i+1} - x_{i-1}}. \tag{3.2}$$

Equations (3.1) and (3.2) can be derived by expanding x_{i+1} and x_{i-1} about x_i. These can be used to develop algorithms suitable for the grid generations [11]. Note that according to definition (3.1), resolution measures the concentration of grid points.

3.2. Approximations to grid spacing ratio. In the same spirit as above, definition (2.2) of the grid spacing ratio can be extended as continuous variables within various orders of approximations.

3.2.1. First-order approximation. Using Taylor series, it is easy to rewrite (2.2) in the following form:

$$r_i = \frac{x_\zeta(\zeta_i)}{x_\zeta(\zeta_{i-1})} + O(h_\zeta). \tag{3.3}$$

Therefore, to first-order accuracy, we can define

$$R_3(x_i) = \frac{x_\zeta(\zeta_i)}{x_\zeta(\zeta_{i-1})}, \tag{3.4}$$

which is the approximation to $r(x)$ due to neglect of $O(h_\zeta)$ terms in (3.3).

3.2.2. Second-order approximations. We now derive two second-order accurate continuous analogues of (2.2). It follows from (2.2) that

$$r(x) - 1 = h_\zeta \frac{D_+ D_- x}{D_- x} \quad \text{and} \quad r(x) + 1 = 2\frac{D_0 x}{D_- x}, \tag{3.5}$$

where D_+, D_-, and D_0 are the forward, backward, and central difference operators, respectively. For notational convenience, we have suppressed the subscript i in (3.5).

Using Taylor series, it is easy to see that these difference operators are related to their corresponding differential operators by

$$x_{\zeta\zeta} = D_+D_-x + O(h_\zeta^2) \tag{3.6}$$

and

$$x_\zeta = D_-x + O(h_\zeta) = D_+x + O(h_\zeta) = D_0x + O(h_\zeta^2). \tag{3.7}$$

Using (3.5)–(3.7), we have

$$r(x) - 1 = h_\zeta \frac{x_{\zeta\zeta} + O(h_\zeta^2)}{x_\zeta + O(h_\zeta)} = h_\zeta \frac{x_{\zeta\zeta}}{x_\zeta} + O(h_\zeta^2). \tag{3.8}$$

Alternatively, this may be written as

$$r(x) = 1 + \alpha + O(h_\zeta^2), \tag{3.9}$$

where

$$\alpha(x) = h_\zeta \frac{x_{\zeta\zeta}}{x_\zeta}. \tag{3.10}$$

Therefore, to second-order approximation we may write

$$r(x_i) = R_1(x_i), \tag{3.11}$$

where we have defined

$$R_1(x) = 1 + \alpha(x). \tag{3.12}$$

In (3.11) we use the subscript 1 on R to distinguish it from another second-order approximation R_2, which we derive below. Thus equation (3.11) is an appropriate second-order accurate continuous extension of (2.2), provided that $x_\zeta \neq 0$, for all $\zeta \in \Omega_\zeta$ (see (3.10)). In fact, $x_\zeta \neq 0$ is guaranteed for a one-to-one smooth mapping (see §4). Since addition of higher-order terms in (3.9) does not alter the order of accuracy, it is possible to derive other second-order accurate continuous analogues of (2.2). For example, it follows from (3.5), (3.6), and (3.7) that

$$\frac{r(x)+1}{r(x)-1} = \frac{2}{h_\zeta} \frac{D_0x}{D_+D_-x} = \frac{2}{h_\zeta} \frac{x_\zeta + O(h_\zeta^2)}{x_{\zeta\zeta} + O(h_\zeta^2)}. \tag{3.13}$$

Note that we have again suppressed the subscript i. A simple algebraic manipulation gives

$$r(x) = \frac{2+\alpha(x)}{2-\alpha(x)} + O(h_\zeta^2), \tag{3.14}$$

where α is again given by (3.10). Therefore, to second-order approximation we can write

$$r(x_i) = R_2(x_i), \tag{3.15}$$

where we have defined

$$R_2(x) = \frac{2+\alpha(x)}{2-\alpha(x)}. \tag{3.16}$$

Expanding $R_2(x)$ in α and dropping second- and higher-order terms in α, we recover $R_1(x)$. Since at $\alpha = 0, R_1 = R_2 = 1$, which corresponds to uniform grid spacing, α may be interpreted as a measure of nonuniformity between two consecutive grid spacings.

We will be interested in generating adaptive grids using the second-order approximation methods (3.11) and (3.15). Some constraints arise due to numerical approximations which we mention below.

3.3. Constraints. The natural constraint that arises due to the approximation (3.11) is that

$$\alpha > -1, \tag{3.17}$$

so that mapping is monotonic, i.e., $R_1 = 1 + \alpha > 0$. Similarly, the approximation method (3.15) satisfies the monotonic mapping property $R_2 > 0$, provided

$$-2 < \alpha < 2. \tag{3.18}$$

For reasons mentioned in §1, it may be necessary that the grid spacing ratio satisfies

$$0 < r_l \leq r(x_i) \leq r_u \quad \forall x_i \in \Omega_x, \tag{3.19}$$

where r_l and r_u are given. We approximate this requirement to second-order accuracy by

$$0 < r_l \leq R \leq r_u, \tag{3.20}$$

where R is either R_1 or R_2, depending on the method used. It follows from (3.12) and (3.16) that (3.20) will impose constraints on α, which is summarized in the following proposition.

PROPOSITION 3.1. *Given r_l, and r_u (both must be positive), α must satisfy*

$$a \leq \alpha(x) \leq b \quad \forall x \in \Omega_x, \tag{3.21}$$

where

$$a = \begin{cases} r_l - 1 & \textit{for } R = R_1, \\ 2\dfrac{r_l - 1}{r_l + 1} & \textit{for } R = R_2, \end{cases} \tag{3.22}$$

and

$$b = \begin{cases} r_u - 1 & \textit{for } R = R_1, \\ 2\dfrac{r_u - 1}{r_u + 1} & \textit{for } R = R_2. \end{cases} \tag{3.23}$$

Proof. This follows easily from (3.12), (3.16), and (3.20). □

Due to the leftmost inequality in (3.20), the natural constraints (3.17) and (3.18) are embodied in (3.21).

4. Equations of grid generation. Consider the generation of one-dimensional adaptive grids by direct integration of

$$x_\zeta = c(k)w(x;k) \text{ on } x_l \le x \le x_u, \tag{4.1}$$

which maps $\Omega_\zeta \to \Omega_x$. The constant $c(k)$ in (4.1) is a scaling constant which maps Ω_x exactly onto Ω_ζ. As we will see later (see (4.9)), this scaling constant depends on the choice of the parameter k which we have shown explicitly in (4.1). From (3.4) and (4.1) we see that the function $\alpha(x;k)$ also depends on the parameter k.

The adaptive grids are generated on the x axis. Without loss of generality, the indices of the adaptive grid points on the x axis are taken as the coordinates of the corresponding ζ-grid points, i.e., $\Omega_\zeta : [\zeta_0 = 0, \zeta_N = N] \to \Omega_x : [x_0 = x_l, x_N = x_u]$. Therefore, in the ζ coordinate we have uniform spacing, given by $h_\zeta = 1$. This convention gives a domain Ω_ζ that is dynamically increasing as the number of x-grid points increases. Thus, the two second-order accurate approximations of $r(x)$ in the previous section should be interpreted as $O(\frac{1}{N^2})$ accurate, which is the order of the derivatives hidden in the $O(h_\zeta^2)$ terms in §3.

The function $w(x;k)$ is chosen to have the form

$$w(x;k) = \tilde{w}(x, f(x), f_{xx}; k). \tag{4.2}$$

The parameter k has been introduced so that this can be adjusted to obtain desired bounds on grid spacing ratio. The required adaptivity is dictated by the choice of $w(x;k)$. However, the function $w(x;k)$ must be properly chosen so that the mapping is one to one.

The following theorem guarantees monotonicity of the mapping provided that $w(x;k)$ has the same sign for all $x \in \Omega_x$. Even though this is very obvious, we mention it here for the sake of completeness.

THEOREM 4.1. *If $w(x;k)$ does not change sign in Ω_x and $x_u > x_l$, then $x_\zeta > 0$, (i.e. the mapping is monotonic).*

Proof. Since $0 \leq \zeta \leq N$, it follows that

$$N = \int_{x_l}^{x_u} \zeta_x dx = \frac{1}{c(k)} \int_{x_l}^{x_u} \frac{dx}{w(x;k)}. \tag{4.3}$$

If $w(x;k) > 0$, we have from (4.3)

$$c(k) > 0. \tag{4.4}$$

Similarly, it also follows that $c(k) < 0$ if $w(x;k) < 0$ in Ω_x. Thus $w(x;k)$ and $c(k)$ have the same sign in Ω_x, and hence

$$x_\zeta = c(k)w(x;k) > 0 \quad \forall\, x \in [x_l, x_u]. \qquad \square$$

Usually, the form of $\tilde{w}(.\ ;k)$ is chosen so that it is positive in $[x_l, x_u]$. Henceforth we will assume that $w(x;k)$ is positive for all $x \in \Omega_x$.

The adaptive grids may be generated by direct integration of (4.1). However, in this procedure the grid spacing ratio often turns out be a very oscillatory function. Therefore a postprocessing operation would be required before these grids could be used for practical purposes. The postprocessing would involve smoothing of the grid spacing ratio and regeneration of the adaptive grids using this smoothed grid spacing ratio. An alternative and better procedure would be to initially define a smooth grid spacing ratio function and use this directly for the grid generation. This not only avoids the problem above but is also less expensive. This has been the purpose behind developing the grid spacing ratio based methods in §3. In addition, use of (3.11) or (3.15) would generate higher-order grids as justified earlier. However, as we have seen, these approximations induce additional constraints (3.21). This, in turn, imposes a constraint on $w_x(k)$ which is summarized in the following proposition.

PROPOSITION 4.2. *Given r_l and r_u, the $w(x;k)$ must satisfy*

$$a \leq \frac{1}{N} g(x_u;k) w_x(x;k) \leq b, \tag{4.5}$$

where $g(x;k)$ is defined as

$$g(x;k) = \int_{x_l}^{x} \frac{dy}{w(y;k)}, \tag{4.6}$$

and the constants a and b are given in (3.22) *and* (3.23).

Proof. From (4.1) we have

$$x_{\zeta\zeta} = c(k) w_x x_\zeta. \tag{4.7}$$

With $h_\zeta = 1$, α in (3.10) reduces to

$$\alpha(x;k) = c(k) w_x(x;k). \tag{4.8}$$

Using (4.3) and (4.6), we have

$$c(k) = \frac{1}{N} g(x_u; k). \tag{4.9}$$

It follows from (4.8) and (4.9) that

$$\alpha(x; k) = \frac{1}{N} g(x_u; k) w_x(x; k). \tag{4.10}$$

Relation (4.5) follows from (3.21) and (4.10). □

Therefore $w(x, k)$ must be chosen so that it satisfies (4.5). In grid generation, it is of interest to know the locations of maximum and minimum resolution. This is addressed in the following proposition. For the purpose of simplicity, the endpoint extremà will not be dealt with in the rest of this paper.

PROPOSITION 4.3. *Resolution attains its extremal values at zeros of α or, equivalently, at $R = 1$.*

Proof. From (2.1) and (4.1) we have

$$s(x) = (c(k)w(x; k))^{-1}. \tag{4.11}$$

Differentiating $s(x)$ with respect to x shows that

$$s_x = -(c(k))^{-1} w^{-2} w_x; \tag{4.12}$$

therefore $w_x = 0$ where $s_x = 0$. Substituting this in (4.8) gives $\alpha = 0$. Then from (3.12) and (3.16) we have $R_1 = 1$ and $R_2 = 1$. □

4.1. Parametric control of (4.5). Proposition 4.2 provides the relation (4.5) which must be satisfied by a proper choice of the mapping function $w(x; k)$ and N. However, in practice it may be convenient to vary one or both of the free parameters k and N for a fixed form of $w(x; k)$. Below we give details on how to choose N and k so that (4.5) is satisfied.

A set of specified values of r_l and r_u may be divided into two basic groups: (i) $r_u > 1$, $r_l < 1$ and (ii) $r_u \geq r_l \geq 1$.

Remark. The case ($r_l = 1$, $r_u = 1$,) corresponds to uniformly spaced grids.

In terms of the constants a and b defined in (3.22) and (3.23), these two basic groups correspond to (i) $a < 0$, $b > 0$, and (ii) $b > a \geq 0$, respectively. The functional form of $w(x; k)$ is usually fixed by the desired adaptivity (see §5). We now discuss how to select k or N in case (i). Case (ii) may be handled by a minor modification of case (i).

4.1.1. Fixed k, adjustable N. Since $a < 0$ and $b > 0$, it follows from (4.5) that N must be chosen so that

$$N = \max\left(\frac{g(x_u; k)}{b} \max_x w_x, \frac{g(x_u; k)}{a} \min_x w_x \right). \tag{4.13}$$

Thus N depends on the function $f(x)$. This means that the number of grid points has to be adjusted very often in time-dependent problems, where $f(x)$ evolves in time. This may be inconvenient and undesirable. An alternative procedure would be to keep N fixed and vary the parameter k.

4.1.2. Fixed N, adjustable k. To select proper values of k, so that (4.5) is satisfied, we need some results. Hereafter, we assume that $w(x;k)$ is of the following form:

$$w(x;k) = (k+v(x))^{\frac{-1}{2}}, \tag{4.14}$$

where

$$v(x) = \tilde{v}(x, f_x, f_{xx}, \cdots), \tag{4.15}$$

where $v(x) \geq 0$ and k is strictly positive. The form (4.15) is chosen because most of the adaptions in use belong to this class (see §5).

THEOREM 4.4. *$\alpha(x;k)$ is a monotone function of k. In particular,* $\ln|\alpha(x;k)|$ *is a decreasing function of k and is given by*

$$\frac{\partial \ln|\alpha|}{\partial k} = h(x;k), \tag{4.16}$$

where

$$h(x;k) = -\frac{1}{2g(x;k)}\int_{x_l}^{x_u} \frac{[3w^2(x;k)+w^2(t;k)]}{w(t;k)}\,dt. \tag{4.17}$$

(*Note that $h(x;k)$ is negative.*)

Proof. Using (4.10) and simple calculations, we find that

$$\alpha_k = \frac{1}{N}\int_{x_l}^{x_u} \frac{W(w_x(x;k), w(t;k))}{w^2(t;k)}\,dt, \tag{4.18}$$

where W is the Wronskian of w_x and w, and α_k is the partial derivative of α with respect to k. Using (4.14), expression (4.18) reduces to

$$\alpha_k = -\frac{\alpha}{2g(x;k)}\int_{x_l}^{x_u} \frac{[3w^2(x;k)+w^2(t;k)]}{w(t;k)}\,dt. \tag{4.19}$$

Expression (4.16) follows. Note that α is a monotone function of k since $\alpha_k = 0$ for $\alpha_k = 0$ from (4.19). □

Let $\alpha_{\max} \in \{\alpha(x;k) : \alpha_x = 0,\ \alpha_{xx} < 0\}$ and $\alpha_{\min} \in \{\alpha(x;k) : \alpha_x = 0,\ \alpha_{xx} > 0\}$. Thus $\alpha_{\max}$ and $\alpha_{\min}$ are the values of a local maximum and a local minimum, respectively.

COROLLARY 4.5. *If $\alpha_{\max} \geq 0$ and $\alpha_{\min} \leq 0$, then $\alpha_{\max}$ and $\alpha_{\min}$ are monotone functions of k.*

Proof. It follows from (4.19) that

$$\frac{\partial \alpha_{\max}}{\partial k} \leq 0, \qquad \frac{\partial \alpha_{\min}}{\partial k} \geq 0, \tag{4.20}$$

and

$$\frac{\partial|\alpha_{\max}-\alpha_{\min}|}{\partial k} \le 0. \tag{4.21}$$

Hence $\alpha_{\max}$ and $\alpha_{\min}$ are monotone functions of k. □

Below the notation R denotes both R_1 and R_2.

PROPOSITION 4.6. *The grid spacing ratio $R(x;k)$ is a monotone function of k. In particular, R is a decreasing (increasing) function of k if $R \ge 1$ ($R \le 1$).*

Proof. The first part follows immediately upon differentiating (3.12) and (3.16) and using Theorem 4.4. On noticing in (3.12) and (3.16) that $R \ge 1$ if $\alpha \ge 0$ and $R \le 1$ if $\alpha \le 0$, the second part follows from Theorem 4.4.

Let $R_{\max} \in \{R(x;k) : R_x = 0, R_{xx} < 0\}$ and $R_{\min} \in \{R(x;k) : R_x = 0, R_{xx} > 0\}$. Thus $R_{\max}$ and $R_{\min}$ are the values of a local maximum and a local minimum, respectively. □

COROLLARY 4.7. *If $R_{\max} \ge 1$ and $R_{\min} \le 1$, then $R_{\max}$ and $R_{\min}$ are monotone functions of k.*

Proof. It follows from previous proposition that

$$\frac{\partial R_{\max}}{\partial k} \le 0, \qquad \frac{\partial R_{\min}}{\partial k} \ge 0, \tag{4.22}$$

and

$$\frac{\partial|R_{\max}-R_{\min}|}{\partial k} \le 0. \tag{4.23}$$

Hence the corollary follows. □

Theorem 4.4 through Corollary 4.7 are useful in finding a suitable value of k so that (3.21) is satisfied. It is clear that (3.21) can always be satisfied by choosing a large value of k. However, it will be desirable to find a best possible value of k ($= k_{\text{opt}}$) so that (3.21) is satisfied for all $k > k_{\text{opt}}$. We take up this issue in §5.

Due to the continuous descriptions of the resolution and the grid spacing ratio, further analysis can easily be carried out to establish some interesting properties. Through such analysis we find that the critical points where resolution $s(x;k)$ attains its extremal values do not depend on the parameter k and are given by $x^* : v(x^*) = 0$. In essence, we can calculate these critical points without generating the adaptive grids. Even though this method may seem to be particularly of theoretical interest, nonetheless it can be of interest in studying convergence properties of adaptive grids with increasing number of adaptive grid points.

5. Remarks on the theorems and choices of $w(x)$.

At this point it is worthwhile to emphasize the importance of the theorems of the previous section. The constraints on the form of the adaptive function $w(x;k)$ and the value of the parameter k in $w(x;k)$ are embodied in Proposition 3.1 and Theorem 4.1 through Proposition 4.3. Theorem 4.4 through Corollary 4.7 pertain to a specific form of $w(x;k)$, namely, (4.14). These results, in particular Theorem 4.4 through Corollary (4.7), are useful in the selection of a proper value of the parameter k (see §6).

The specific form (4.14) of the adaptive function $w(x;k)$ is chosen because a large number of adaptive functions in practice is included in this class [5], [13], [19], [26].

Appropriate adaptivity, of course, is dictated by the problem. Usually, however, the function $w(x)$ is chosen so that it adapts the function $f(x)$ or $f_x(x)$ or $f_{xx}(x)$ or a combination of these. We cite some typical forms of $w(x)$ for various types of adaption.

(a) *Function adaption.*

$$w(x) = (k_0 + (f)^{2l_0})^{\frac{-1}{2}}. \tag{5.1}$$

(b) *First derivative adaption.* This form is usually used to resolve sharp gradients.

$$w(x) = (k_1 + (f_x)^{2l_1})^{\frac{-1}{2}}. \tag{5.2}$$

(c) *Second derivative adaption.* This form is usually used to resolve sharp corners.

$$w(x) = (k_2 + (f_{xx})^{2l_2})^{\frac{-1}{2}}. \tag{5.3}$$

(d) *First and second derivative adaption.* This form is usually used to resolve sharp corners and sharp gradients.

$$w(x) = (1 + k_1 \times (f_x)^{2l_1} + k_2 \times (f_{xx})^{2l_2})^{\frac{-1}{2}}. \tag{5.4}$$

(e) *Spatial adaption.* This form is usually used to resolve boundary layers.

$$w(x) = Ce^{-x/\delta}. \tag{5.5}$$

Note that, consistent with Theorem 4.1, $w(x)(> 0)$ is of one sign in all these cases and these forms are in class (4.14), justifying our interest in the analysis of §5. In all of these adaptions, k_0, k_1, k_2 are arbitrary constants and l_0, l_1, l_2 are integer constants. Usually the constants l_0, l_1, l_2 are taken to be 1. k_0, k_1, and k_2 can be tuned to control function, first and second derivative adaptivities, respectively. As seen from (4.1) and the various adaptions mentioned above, the higher the values of l_1, l_2, the more severe is the concentration of grid points in regions of steep gradients and high curvature, respectively.

In passing, we should mention that adaptive functions quite different from these have been used in practice by some authors [12], [13].

6. Validation.

This section is devoted to computations that support our analysis of the previous sections. We should recall that the functions R_1 and R_2 approximate the discrete grid function $r(x)$ to second-order accuracy. Equating the continuous and the discrete definitions of this grid function (i.e., (3.11) or (3.15)), an appropriate algorithm can be set up for grid generation. Since our analysis here is based on the continuous description, we carry out the computations below with a large number of grid points for the purpose of validation. We set up the following simple and direct algorithm which will work when the number of adaptive grid points is large. In applications, however, the adaptive grid points will not be large and the algorithm has to be modified, as seen in §8.

From (3.11) and (3.15), we obtain the following explicit form of generating the adaptive grids:

$$x_{i+1} = x_i + (x_i - x_{i-1})R(x_i), \tag{6.1}$$

where R can be either R_1 or R_2. We apply this equation for i = 1, $N - 1$. This method has the advantage that it is possibly the simplest way to generate these grids. However, the serious disadvantage is that this method requires a correct initial guess for x_1 and the constant c (R depends on α and α depends on the constant c through (4.8)). Otherwise, x_N obtained from (6.1) for i = $N - 1$ will not be exactly what it should be, i.e., the right boundary of Ω_x. However, for large N, c and x_1 are easy to obtain from direct integration of (4.1). x_1 is obtained from interpolation and the constant c is obtained from (4.9). In the computations below, we use the continuous function $R = R_2$ (this has been found to be a more efficient grid generator than others) and the following adaptive function:

$$w(x; k) = (k + {f_x}^2)^{\frac{-1}{2}}. \tag{6.2}$$

In (6.2), the parameter k is user specified. The parameter k should be properly selected because it affects some of the important features of the adaptive grids mentioned in §4. In particular, in the limit of extreme values of this parameter k, we have

$$w(x; 0) = |{f_x}^{-1}|, \qquad w_x(x; 0) = -\text{sgn}(f_x){f_x}^{-2}f_{xx}, \tag{6.3}$$

and

$$w(x; \infty) = 0, \qquad w_x(x; \infty) = 0. \tag{6.4}$$

Then it follows from (6.4) that when k approaches infinity, $\alpha = cw_x$ approaches zero and the grid spacing ratio R (see (3.12) and (3.16)) approaches unity. Thus, in this limit we have uniformly spaced grid points.

It also follows from (6.3) that in the other limit, k approaching zero, $\alpha = cw_x$ becomes singular at zeros of f_x and f_{xx}, which is not allowed by the constraint (3.21). We remark that in most practical instances, the function f_x is likely to have a zero in the domain of interest (this is true in the examples below). Thus it will be necessary to choose k greater than some minimum value k_{min}. In fact (3.21), (4.10), and (4.14) imply that the parameter k must lie within some range and this can be easily calculated if desired.

Example 1. We consider the function

$$f(x) = [1 + \text{sgn}(x - x_c)\{1 - \exp(-aX^2 + 1/2)\}]/2, \tag{6.5}$$

where $X = 1/\sqrt{(2a)} + |x - x_c|$, $a = 2000$, $x_c = \frac{1}{2}$, $\text{sgn}(x - x_c) = 1$ if $x \geq x_c$ and $\text{sgn}(x - x_c) = -1$ if $x < x_c$.

This function is shown in Fig. 6.1(a). The Gaussian term contributes smooth but rapid variation in the function values in the vicinity of the center of the domain. The higher the value of a in (6.5), the steeper the function is. (We should note that a smoother function than this is considered in [19] with no success.) The adaptive grid

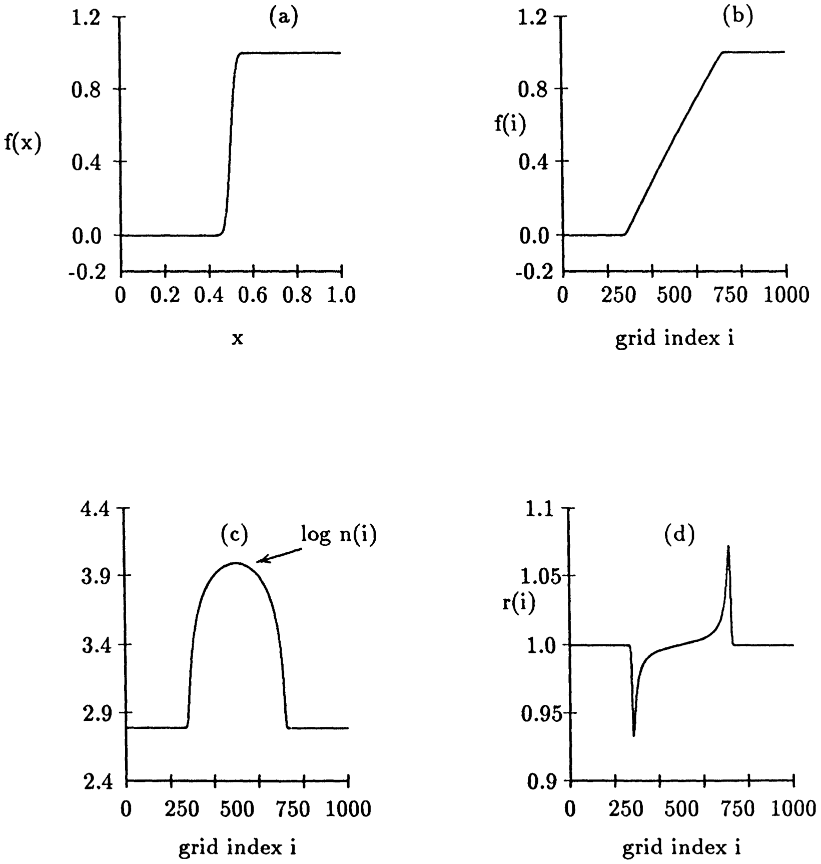

FIG. 6.1. *Second-order adaptive grids with* $N = 1001$ *and* $k = 1$. (a) *function* (6.5) *in physical space*; (b) *function* (6.5); (c) *discrete resolution*; (d) *grid spacing ratio. Figures* (b), (c), *and* (d) *are in the reference frame.*

should be able to resolve this fast variation. We generate the adaptive grids using $k = 1$ in (6.2) and N=1001 grid points. The function in the reference frame (grid index space) is shown in Fig. 6.1(b). We see that the function is considerably spread out in the region of interest. In Fig. 6.1(c) we have plotted $\log(n(i))$, where $n(i)$ are computed from (3.1). We see that the maximum grid concentration is 2×10^4, equivalent to having 2×10^4 uniformly spaced grid points. Near the endpoints of the domain, where the function is constant, the resolution is at a minimum and is about 6×10^2. This amounts to roughly a twofold increase in the grid spacing there.

Fig. 6.1(d) shows the variation of grid spacing ratio in the reference frame. We notice an asymptotic grid spacing ratio of $r = 1$ near the boundaries of the domain where the function is essentially constant. We also notice a rapid variation in grid spacing in the vicinity of the sharp transition in the gradient of the function (see Fig. 6.1(b)). Such rapid variations in grid spacing ratios is not likely to appear with smaller number of grid points.

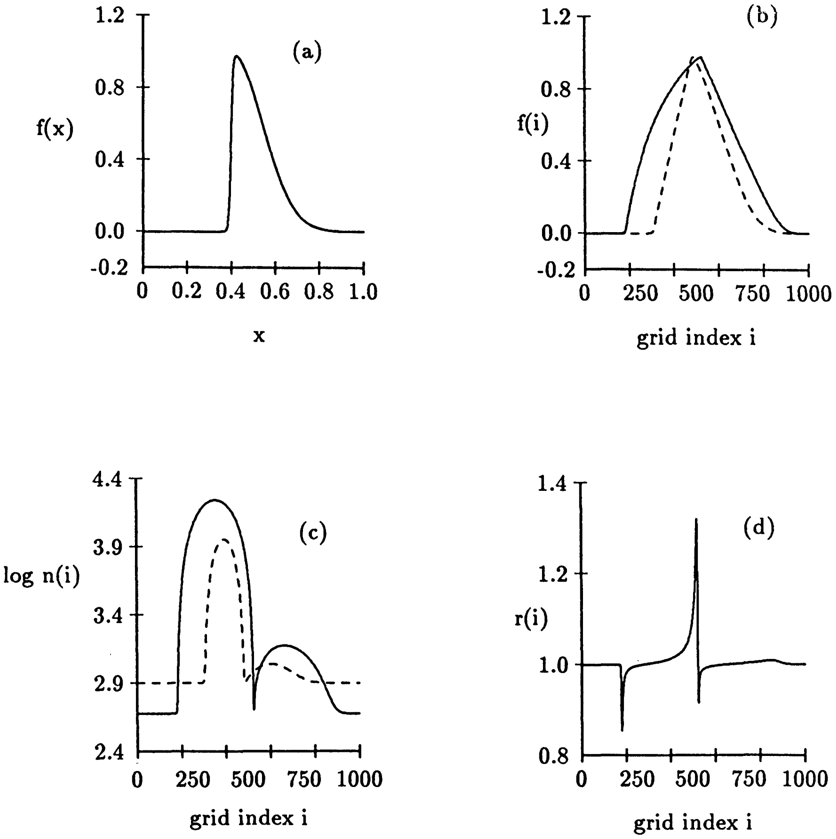

FIG. 6.2. *Effect of k on the adaptive grids $k = 1$ (solid line), $k = 10$ (dashed line):* (a) *function* (6.6) *in physical space,* (b) *function* (6.6); (c) *discrete resolution* (3.1); (d) *grid spacing ratio* (2.2). *Figures* (b), (c), *and* (d) *are in the reference frame.*

It should be seen in (6.5) and Fig. 6.1(a) that the function $(f(x)-1/2)$ is odd about $x = x_c$. In Figs. 6.1(b)–6.1(d), we find that our computation preserves this property, adding to the credibility of the theory and the computation. The computations are done without invoking this symmetry.

Example 2. Here we consider the function [10]

$$(6.6) \quad f(x) = \tfrac{1}{2}[1 + \tanh(10^3(x - 0.4))] \exp[-((x - 0.4)/0.2)^2], \qquad x \in [0, 1].$$

This function, shown in Fig. 6.2(a), is a very good model for the profiles of shock waves smoothed by slow diffusion. Figures 6.2(b)–6.2(d) show the effect of varying the values of the parameter k on the different properties of the function in the reference frame. In these figures, solid and dashed curves correspond to $k = 1$ and $k = 10$,

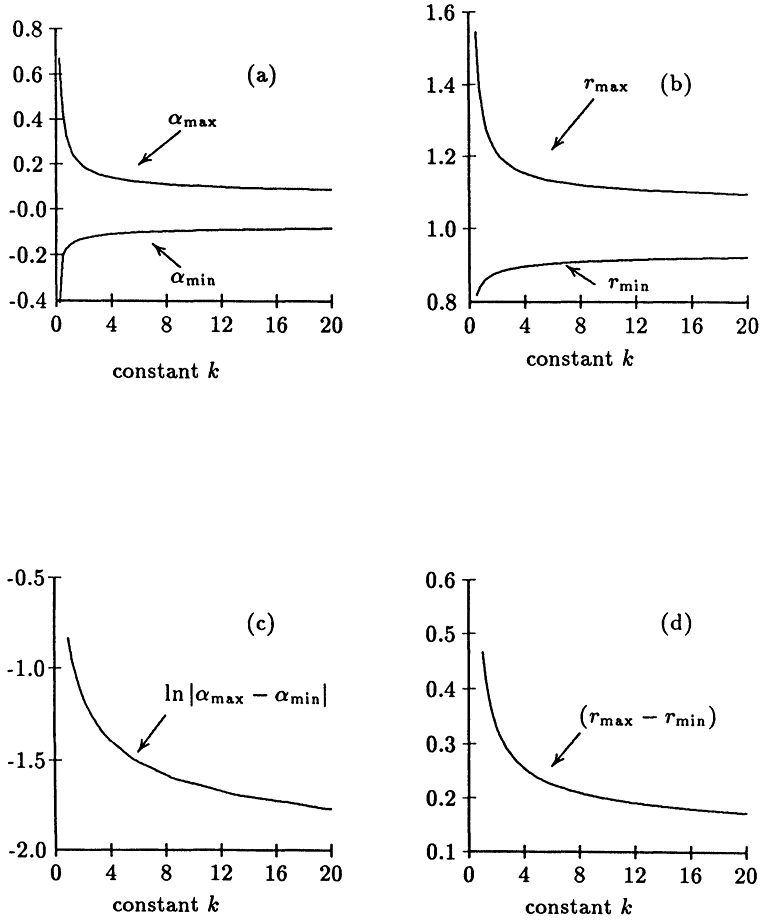

FIG. 6.3. *Grid functions α and r as functions of k for function* (6.6).

respectively. In Fig. 6.2(b) we see that an increase in the value of k decreases the spread of the function in the reference frame. This is expected because $k = \infty$ corresponds to uniform spacing, and hence there is no spreading of the function at all. Figures 6.2(c) and 6.2(d) also support our analysis. We see in these figures that an increase in the value of k decreases the peak resolution and the extremal grid spacing ratio.

Figures 6.3(a)–6.3(d) show the variations of some properties of adaptive grids with k, as obtained from our computations. These computations pertain to the function (6.6) and are done with N=1001. We find that the Figs. 6.3(a)–6.3(d) are consistent with our analysis. In Fig. 6.3(a), our computations show that $\alpha_{\max}$ and $\alpha_{\min}$ are monotone functions of k, consistent with Corollary (4.5). Fig. 6.3(b) shows the monotone property of the extremal grid spacing ratios as functions of k, consistent with Corollary 4.7. These two curves approach 1 asymptotically with $k \to \infty$. In Fig.

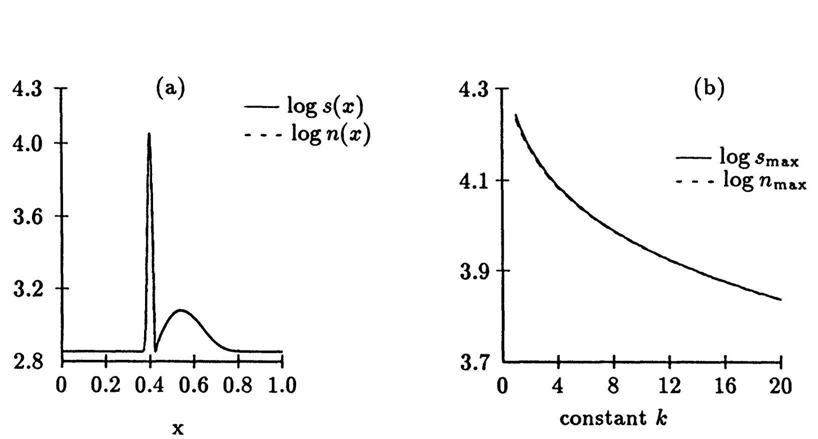

FIG. 6.4. *Comparison of continuous* $s(x)$ *and discrete* $n(x)$ *resolutions with* $N = 1001$ *for the function* (6.4): — *corresponds to* $s(x)$ (2.1); - - - *corresponds to* $n(x)$ (3.1); (a) *discrete and continuous resolutions in physical space with* $k = 5$; (b) *log of maximum values of* $s(x)$ *and* $n(x)$ *as function of the parameter* k.

6.3(c), computations show that $\ln|\alpha_{\max}(k) - \alpha_{\min}|$ is a decreasing function of k, as seen in (4.21). As seen in Fig. 6.3(d), $(r_{\max} - r_{\min})$ is a decreasing function of k and will approach zero with $k \to \infty$, in agreement with our analysis.

In Fig. 6.4(a) we have plotted continuous and discrete resolutions as calculated from (2.1) and (3.1) for the same function (6.6). We see that the comparison is excellent, as these are virtually indistinguishable. In Fig. 6.4(b) we have plotted peak values of both continuous and discrete resolutions, as functions of the parameter k. Here again these two plots seem to agree very well. These calculations justify the use of (3.15) for adaptive grid generation.

Note. It will be useful to note the following which, in part, we have mentioned in §4 and in the above analysis of computational results: (i) peak resolution and peak grid spacing ratios are decreasing functions of k. Thus high values of maximum resolution and low values of grid spacing ratio, both of which are desirable, act against each other. It therefore follows easily that to contain grid spacing ratio within some limit, there is an upper bound on the maximum resolution and hence a lower bound on the value of k (see Fig. 6.3(b)). (Of course, this value of k must be greater than $k_{\min}$, as mentioned earlier.) These observations bear on what follows next.

6.1. Grid generation with r_l and r_u specified. The monotonic profiles of extreme values of $\alpha(k)$ and $R(k)$ in Figs. 6.3(a) and 6.3(b) can be used to advantage in automating the procedure for generating adaptive grids with following properties: (i) highest possible peak resolution and (ii) grid spacing ratio within some prescribed limits r_l and r_u. The importance of the first consideration is obvious, as this is one of the motivating factors behind adaptive grid generation. As mentioned earlier, the second consideration is important for stability of some numerical methods on adaptive grids.

From Fig. 6.5(a) we should notice that for arbitrarily prescribed values of r_l and r_u,

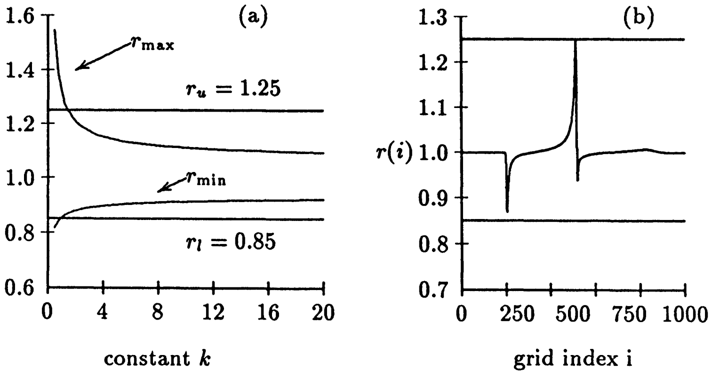

FIG. 6.5. *Optimal choice of k to control grid spacing ratio:* (a) *range of grid spacing ratio and optimal choice of $k = k_{\mathrm{opt}}$;* (b) *grid spacing ratio in the reference frame; horizontal lines in* (a) *and* (b) *are at $r_l = 0.85$ and $r_u = 1.25$.*

$$k(r_l) \neq k(r_u), \tag{6.7}$$

except in very fortunate situations. (In the present formulation of the problem, there does not exist a value k such that $r_{\min}(k) = r_l$ and $r_{\max}(k) = r_u$ for arbitrary r_l and r_u.) From this and Fig. 6.5(a) it follows that k must be chosen so that

$$k \geq \max(k(r_l), k(r_u)). \tag{6.8}$$

This will ensure that the grid spacing ratio is within the prescribed limits. However, in order to have the highest possible resolution, the smallest allowed value of k should be chosen (see Fig. 6.4(b)):

$$k = \max(k(r_l), k(r_u)). \tag{6.9}$$

With this choice of k, adaptive grids will in general have either $r_{\min} = r_l$ or $r_{\max} = r_u$. Regardless of which situation is realized in computation, we will always have $[r_{\max}, r_{\min}] \in [r_u - r_l]$, as seen in Fig. 6.5(a). In Fig. 6.5(b), we show the grid spacing ratio of the adaptive grids that are generated with prespecified $r_l = 0.85$ and $r_u = 1.25$. The graphs $r_{\min}(k)$ and $r_{\max}(k)$ are constructed numerically and the value of k is chosen from these graphs according to (6.9). This is shown in Fig. 6.5(a). With this choice of k, the adaptive grid is generated. The grid spacing ratio of this generated adaptive grid is shown in Fig. 6.5(b). We find $r_{\max} = 1.25$ and $r_{\min} = 0.87015$ (see Fig. 6.5(b)) and hence this adaptive grid is an acceptable grid according to our earlier discussion.

This section has been devoted to justifying the validity of formula (3.1) in constructing a suitable set of grids. However, the practical algorithms for construction of adaptive grids with or without the prescription of r_l and r_u have to be different for the following reasons: (i) if N is small, then the x_N obtained from our algorithm will not in general be the right boundary x_u of Ω_x; (ii) in this algorithm with r_l and r_u specified, the appropriate value of k is obtained from the curves $r_{\min}(k)$ and $r_{\max}(k)$. However, since these curves are monotone, a Newton type algorithm can easily be devised to find the appropriate value of k. This way these curves need not be generated at all.

In §7 we address some practical algorithms for generating adaptive grids. In §8 we apply these to solve some applied problems. To this end, we would like to add the following remarks.

6.2. Remarks. (1) It should be noted that we have used first derivative adaptivity (5.2) in both the examples presented here and that the grid spacing ratio has only one maximum and one minimum (Figs. 6.1(d), 6.2(d)). However, with different adaptivity and different functions, the grid spacing ratio may have quite a few maxima and minima. In this situation, it is likely that $R_{\max}$ has discontinuities in its first derivative. These would arise when the location of the global maximum changes discontinuously as a function of k, which can happen when there are more than one local maximum. The same scenario is likely to happen for the global minimum. Naturally, similar situations also arise for the $\alpha_{\max}(k)$ and $\alpha_{\min}(k)$ curves.

(2) By introducing one free parameter k in the mapping (4.1), we have been able to generate adaptive grids so that grid spacing ratio can be controlled with the qualifications explained previously (see (6.9)). However, there exists a possibility of introducing another parameter so that the generated grid has spacing ratios $r_{\max} = r_u$ and $r_{\min} = r_l$.

7. A practical algorithm. In the previous section we described a simple algorithm which works well with a large number of grid points. The algorithm is well suited for the purpose of demonstrating the worth of the new adaptive grid generators. However, this algorithm needs to be modified so that it works with any arbitrary number of adaptive grid points.

Since the boundary points x_0 and x_N of the domain Ω are known, construction of adaptive grids using (6.1) requires solving a system of $(N-1)$ coupled equations. The $(N-1)$ coupled difference equations (6.1) satisfy the equation

$$X = AX + F, \tag{7.1}$$

where

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

and A is a tridiagonal matrix with

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

The matrix equation (7.1) may be rewritten as

$$QX = F, \tag{7.2}$$

where $Q = I - A$ is a diagonally dominant matrix. Equations (6.1) and (7.2) are nonlinear due to the dependence of the matrices A and Q on the unknown solution vector X. Equation (7.2) can be solved by various iterative methods. Note that R depends on the constant c and may seem to be a free parameter in the present method of solving, subject to appropriate constraints discussed earlier.

Initially, the grid point coordinates are guessed and then these values are updated by solving (7.2) in an iterative loop until some convergence criterion is met. The value of the parameter c may be kept constant at a user-specified value during iterations or may be updated in each iteration using the following formula:

$$c(k) = \int_{x_0}^{x_1} \frac{dx}{w(x;k)}. \tag{7.3}$$

This is obtained by integrating (4.1) up to the first adaptive grid point. Note that this is equivalent to (4.8) when the number of grid points tends to infinity.

Numerical experiments suggest that very high resolution can be obtained by using an appropriate value for the constant c. In fact, we find that increasing c increases the resolution. However, an inappropriate value for the constant c may cause problems: the converged grid values may be nonmonotonic or contain points from outside the domain or both; or the iteration may not converge. Therefore the constant c can not be chosen arbitrarily. The following procedure has been found very successful in our experiments: The constant c should be initially calculated using (7.3) for the first few iterations until the convergence rate slows down, and thereafter c is to be held fixed at a value between two to four times its current value. Also with some trial and error, a suitable value of c can be found which can be kept fixed all through the iterations.

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. For example, this may be dynamically set at 1% of the smallest grid spacing of the current iteration level.

We consider yet another example which will also be of interest in the next section:

$$f(x,t) = 1 - 2\sqrt{\epsilon}\tanh[x/\sqrt{\epsilon}], \tag{7.4}$$

where ϵ is a constant. We chose $\epsilon = 0.001$. The grids were generated for this function using the above algorithm with $k = 1$, $N = 201$ and with various values of the constant c. In Fig. 7.1 we show that the effect of varying the constant c on the grid properties for fixed $k = 1$. Notice that an increase in the value of c increases resolution and grid spacing ratio. In this example, the maximum value of the resolution with c = 0.01, c = 0.015, c = 0.02 and c = 0.03 are, respectively, 2.49, 3.21, 3.59, and 4.13. This is equivalent to having 310, 1621, 3900, and 13500 uniformly spaced points, respectively. This is a remarkable gain obtained only with 201 adaptive grid points. In contrast we obtain only a modest gain in resolution in the traditional method of integration and interpolation. In the method we obtained a resolution of only 2.45, equivalent to having only 280 uniformly spaced points.

Figure 7.1 also suggests that resolution may be nonuniform in the sense that some part of the function may be highly resolved compared to other parts. If the variation in resolution itself is rapid, then the resolved function may develop somewhat steep gradients even in the reference plane. Ideally, slow variation in resolution is

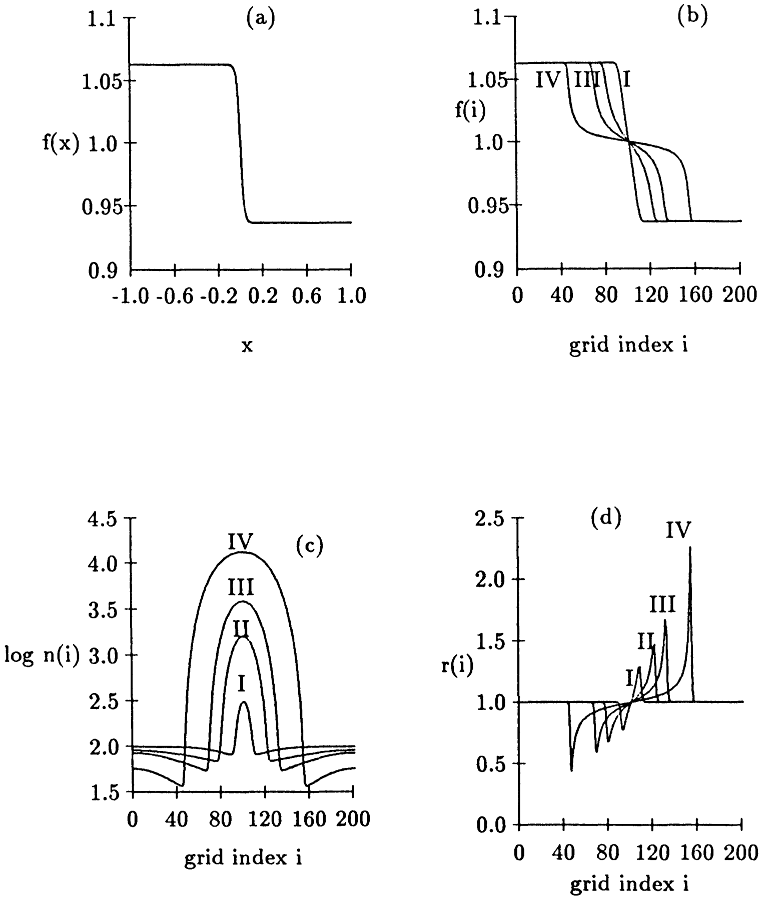

FIG. 7.1. *Adaptive grid properties as a function of the constant c with fixed* $N = 200$ *and fixed* $k = 1$. (a) *function* (7.4) *in the physical space;* (b) *function* (7.4) *in the index space;* (c) *discrete resolution;* (d) *grid spacing ratio. Figures* (b), (c) *and* (d) *are in the reference frame. The values of the constant c for four different families* I, II, III, IV *of the curves in each of the Figs.* 6.1(b), 6.1(c), *and* 6.1(d) *are* 0.01, 0.015, 0.02, *and* 0.03, *respectively. The curve between* I *and* III *in Fig.* 6.1(b) *refers to curve* II.

also desirable. Due to this nonuniformity, an adaptive grid with highest maximum resolution may not always be desirable. In the Fig. 7.1 the adaptive grids obtained with c = 0.02 (curve III in the Fig. 7.1(b)) seems to be more desirable than that (curve IV in Fig. 7.1(b)) with c = 0.03.

8. Numerical examples. In the previous section, adaptive grids were numerically constructed by an algorithm based on our new grid generators. In this section we apply these grids in tracking fronts that arise in convection-diffusion equations and hyperbolic equations. In particular we consider one-dimensional Burgers' equation and Buckley–Leverett equation in porous media flows. Even though these equations are simple examples, nonetheless these are valuable test cases for studying usefulness of these grids. In a sequel, these adaptive grids will be applied to solving nonlinear

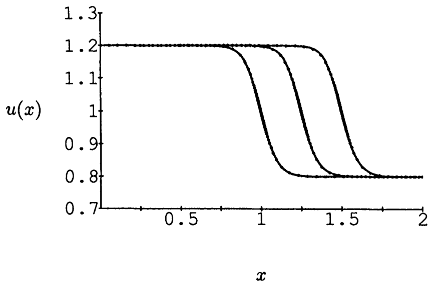

FIG. 8.1. *Comparison of adaptive and exact solution to Burgers' equation at* $t = 0.0$, 0.25, *and* 0.5, *respectively. The wave travels to the right.*

hyperbolic conservation laws by various numerical methods.

8.1. Burgers' equation. Burgers' equation is given by

$$u_t + uu_x = \epsilon u_{xx}, \qquad x \in [0, 1], \tag{8.1}$$

where ϵ is a constant. This constant is to be chosen sufficiently small to have sharp diffusive fronts. In the computations below, we have used $\epsilon = 0.01$. An exact solution to this equation is given by

$$u(x, t) = 1 - 2\sqrt{\epsilon}\tanh[(x - t)/\sqrt{\epsilon}]. \tag{8.2}$$

The initial and boundary data are chosen appropriately using (8.2). Note that the initial data corresponds to the example used in §7, except for the value of ϵ.

There are many methods for adaptive computations of partial differential equations. Since our main focus here is in the use of the grids, we have chosen a simple method, namely, the static regridding method [26]. In static regridding method the adaptive grids are generated at every time interval and computation is advanced one time level on these adaptive grids.

The initial set of adaptive grids is generated using our algorithm of the previous section using $c = 0.02$ and $N = 201$ grid points. The values of the solution to the equation above at the adaptive grid points are contrasted against the exact solution in Fig. 8.1. The speed of the numerical front and other details are seen to be indistiguishable from the correct values within graphical accuracy. This run shows the ability of these new sets of adaptive grid to resolve fronts accurately.

8.2. Buckley–Leverett equation. The one-dimensional Buckley–Leverett equation in porous media is given by

$$s_t + f_x(s) = 0, \qquad x \in [0, 1], \tag{8.3}$$

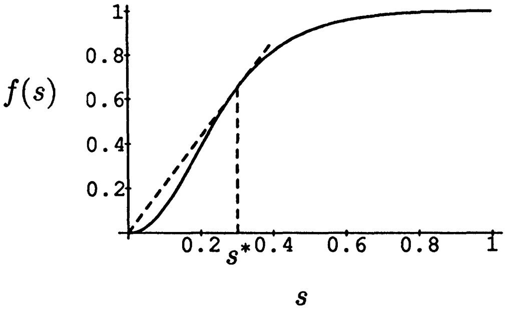

FIG. 8.2. *The nonconvex flux function.*

where

$$f(s) = \frac{s^2}{s^2 + (1-s)^2\mu} \tag{8.4}$$

is usually referred to as the flux function. Appropriate initial and boundary data will be specified later. Physically, (8.3) describes the conservation of mass of water saturation s in porous media flow, e.g., water pushing oil in porous media [14]. The constant μ in (8.4) is the viscosity ratio of the two fluids. Notice that the function $f(s)$ is nonconvex with one inflexion point. The flux function is shown in Fig. 8.2 for $\mu = 0.1$.

Equation (8.1) is solved here by Glimm's random choice method on an adaptive grid generated by our method. The random choice method is a semianalytical technique based on a constructive existence proof for solutions of hyperbolic equations due to Glimm [14]. This was developed into a numerical method by Chorin [7]. We give a very brief description of the method. For details, see [7], [15], [23]. The solution at any time level is approximated by piecewise constant data. To advance the solution in time, the Riemann problem at each location of discontinuity is solved. The solution within each mesh cell at the new time level is the exact solution of the Riemann problem at a point chosen at random within that mesh cell. The timestep is chosen so that the Riemann problem solutions do not interact during that time interval. The method is shown in Fig. 8.3.

In the example below we choose $\mu = 0.1$ and the following initial data:

$$s = s^* \quad \text{for } s \leq 0.5 \quad \text{and} \quad s = 0 \quad \text{for } s > 0.5,$$

where s^* is shown in Fig. 8.2. This allows the initial discontinuity to travel at an exact speed of 2.15831. Figure 8.4 compares the exact solution with the numerical solution at four different time levels at time interval of 0.063. The results compare very well and are within graphical accuracy.

Appendix. Some comments. It is worth noting an alternative formulation which is equivalent to (4.1). On differentiating (4.1), we obtain

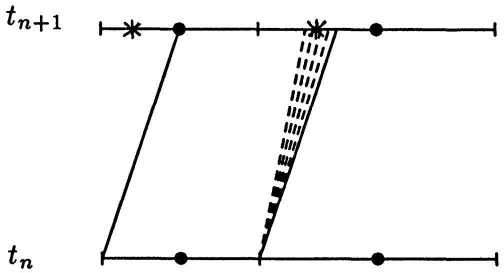

FIG. 8.3. *The adaptive mesh for the random choice method. Each mesh cell is bounded by two neighboring vertical hash marks and is centered about a circle point. The solution within each mesh cell is approximated by the exact solution of the Riemann problems at a randomly chosen point indicated by a star within that mesh cell.*

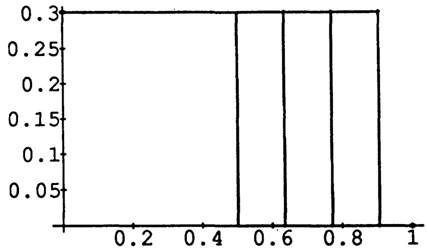

FIG. 8.4. *The comparison of numerical and exact solutions of Buckley–Leverett equations at three different time levels. The solutions are indistinguishable.*

$$x_{\zeta\zeta} = -\left(\frac{w_\zeta^{-1}}{w^{-1}}\right) x_\zeta, \tag{A1}$$

which can be inverted to yield

$$\zeta_{xx} = P(\zeta) \tag{A2}$$

with appropriate $P(\zeta)$. This is the one-dimensional Poisson equation. This equation has been used by Dwyer [13] and Matsuno and Dwyer [16] for adaptive grid generation. In [16], the finite difference approximation is iteratively solved with an initial guess of $x(\zeta)$ until some convergence criterion is met. In their study, Matsuno and Dwyer encountered convergence problems in their iteration scheme for the second test problem mentioned in §6 and suggested developing better methods for obtaining the solution of grid Poisson equation. Thus, in their method there is no guarantee that the iteration will converge [16]. Their method also introduces constraints on the choice of $w(x)$. To elaborate on this, suppose the numerical approximation of $x_{\zeta\zeta}$ is second-order accurate, so that the $x_{\zeta\zeta}$ is obtained from the following truncated Taylor series.

$$x_{i+1} - x_i = x_\zeta + 1/2x_{\zeta\zeta}. \tag{A3}$$

For the mapping to be monotonic, $x_{i+1} - x_i > 0$, and hence from (A3) it follows that

$$x_\zeta + 1/2x_{\zeta\zeta} > 0, \tag{A4}$$

which is equivalent to

$$\frac{x_{\zeta\zeta}}{x\zeta} > -2. \tag{A5}$$

Then, using the mapping (4.1), (A5) may be rewritten as

$$c\frac{w_\zeta}{x_\zeta} > -2, \tag{A6}$$

which is equivalent to

$$w_x > \frac{-2}{c}. \tag{A7}$$

In [16], Matsuno and Dwyer arrive at the same restriction as (A7) but in a somewhat different manner.

Acknowledgments. It is a pleasure to thank Thomas Vogel for proofreading the manuscript. The author thanks the referees for their valuable suggestions which have considerably improved the manuscript.

REFERENCES

[1] S. Adjerid, and J. E. Flaherty, *A moving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations*, SIAM J. Numer. Anal., 23 (1986), pp. 778–796.

[2] I. Babuška, J. Chandra, and J. E. Flaherty, eds., *Adaptive Computational Methods for Partial Differential Equations*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[3] M. J. Berger, *Adaptive finite difference methods in fluid dynamics*, New York University preprint DOE/ER/03077-727, New York, 1987.

[4] M. Bieterman and I. Babuška, *An adaptive method of lines with error control for parabolic equations of the reaction-diffusion type*, J. Comput. Phys., 63 (1986), pp. 33–66.

[5] J. G. Blom, J. M. Sanz-Serna, and J. G. Verwer, *On simple moving grid methods for one-dimensional evolutionary partial differential equations*, J. Comput. Phys., 74 (1988), pp. 191–213.

[6] J. U. Brackbill and J. S. Saltzman, *Adaptive zoning for singular problems in two dimensions*, J. Comput. Phys., 46 (1982), pp. 342–368.

[7] A. Chorin, *Random choice solutions of hyperbolic systems*, J. Comput. Phys., 22 (1976), pp. 517–533.

[8] P. Daripa, *Theory of one dimensional adaptive grid generation*, IMA preprint 612, 1990.

[9] ———, *Using the grid spacing ratio as a continuous variable in one dimensional adaptive grid generation*, App. Math. Lett., 4 (1991), pp. 91–94.

[10] E. A. Dorfi and L. Drury, *Simple adaptive grids for* 1-D *initial value problems*, J. Comput. Phys., 69 (1987), pp. 175–195.

[11] H. A. Dwyer, *Grid adaption for problems with separation, cell Reynolds number, shock boundary layer interactions and accuracy*, AIAA Paper 83-0449.

[12] H. A. Dwyer, R. J. Kee, and B. R. Sanders, *Adaptive grid method for problems in fluid mechanics and heat transfer*, AIAA J., 18 (1980), pp. 1205–1212.

[13] H. A. Dwyer, *Grid adaption for problems in fluid dynamics*, AIAA J., 22 (1984), pp. 1705–1712.

[14] J. Glimm, *Solutions in the large for nonlinear hyperbolic systems of equations*, Comm. Pure Appl. Math., XVIII (1965), pp. 336–354.

[15] J. Glimm and D. Marchesin, *A numerical method for two phase flow with an unstable interface*, J. Comput. Phys., 39 (1981), pp. 179–200.

[16] A. Harten and J. Hyman, *Self adjusting grid methods for one-dimensional hyperbolic conservation laws*, J. Comput. Phys., 50 (1983), pp. 235–269.

[17] G. W. Hedstrom and G. H. Rodrigue, *Adaptive-grid methods for time-dependent partial differential equations*, Proc. Conference on Multigrid Methods, Cologne, November 1981, pp. 474–484.

[18] B. Larrouturou, *A conservative adaptive method for flame propagation*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 742–755.

[19] K. Matsuno and H. A. Dwyer, *Adaptive methods for elliptic grid generation*, J. Comput. Phys., 77 (1988), pp. 40–52.

[20] K. Miller and R. Miller, *Moving finite elements.* I, SIAM J. Numer. Anal., 18 (1981), pp. 1019–1032.

[21] K. Miller, *Moving finite elements.* II, SIAM J. Numer. Anal., 18 (1981), pp. 1033–1057.

[22] M. M. Pervaiz and J. R. Baron, *Temporal and spatial adaptive algorithm for reacting flows*, Comm. Appl. Numer. Methods, 4 (1988), pp. 97–111.

[23] B. J. Plohr, *Modeling of shockless acceleration of thin plates modeled by a tracked random choice method*, AIAA J., 26 (1988), pp. 470–478.

[24] J. D. Ramshaw, *Conservative rezoning algorithm for generalized two dimensional meshes*, J. Comput. Phys., 59 (1985), pp. 193–199.

[25] ———, *Simplified second order rezoning algorithm for generalized two dimensional meshes*, J. Comput. Phys., 67 (1986), pp. 214–222.

[26] J. G. Verwer, J. G. Blom, and J. M. Sanz-Serna, *An adaptive moving grid method for one-dimensional systems of partial differential equations*, J. Comput. Phys., 82 (1989), pp. 454–486.