INTERFACE PROCEDURES FOR OVERLAPPING GRIDS*

C. Wayne Mastin
Department of Mathematics and Statistics
Mississippi State University
Mississippi State, MS 39762

Abstract

Interpolation at grid boundaries is studied for the purpose of solving partial differential equations using either implicit or conservative explicit finite-difference methods on multi-component overlapping grid systems.

(NASA-CR-176460) INTERFACE PROCEDURES FOR OVERLAPPING GRIDS (Mississippi State Univ.) 23 p HC A02/MF A01 CSCL 12A
N86-18061
Unclas
G3/64 05312

JAN 1986
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

*Research supported by NASA Langley Research Center under Grant No. NSG 1577.

Introduction A multi-component grid system, in which several computation grids are used, is required in the numerical solution of many fluid dynamics problems involving flow within or about a complicated geometric configuration. From a grid construction point of view, the simplest procedure is to generate each component grid independently with a sufficient overlap so that information can be transmitted from one grid to the other. The development and analysis of solution procedures on this type of grid system was studied by Starius [8,9]. The practical application of the method to the solution of problems in computational fluid dynamics was demonstrated in the papers by Atta [1], Atta and Vadyak [2], and Thompson [11]. This was followed by further studies on interpolation techniques by Kreiss [5] and Mastin and McConnaughey [6]. Each successful application, such as the recent results of Steger and Buning [10] and Benek, et al [3], serves to reinforce the need for additional work on the implementation of numerical methods on overlapping grids.

Two popular numerical methods in computational fluid dynamics are the alternating direction implicit algorithms and the explicit algorithms derived for the solution of conservation laws. Both methods have been used in the solution of problems on composite grid systems. However, in each case there are properties of the numerical solution which are lost when information is transmitted between the individual grids. With the implicit algorithms, there is no technique for generating advanced solution values at all boundary points of each component grid. Lagging some of the boundary point values can lead to a loss of accuracy in the solution of transient problems. Of course, the temporal step length could be reduced, but that would defeat the purpose of choosing an implicit method. It is also possible that lagging may effect the

stability of the method, although no problems of that kind have been reported.

A conservative finite-difference scheme is often selected when solving partial differential equations in conservation form. Since the classical interpolation formulas were not derived with conservation properties in mind, their use in finite-difference approximations on composite grids would result in the loss of an exact conservation property. An interpolation scheme for conservative finite-difference methods was first proposed by Rai [7]. He considered composite grid systems which did not overlap but joined along common grid lines. Berger [4] indicated how the method of Rai could be generalized and extended to overlapping grids.

This report will describe ways of eliminating the time lag in implicit solutions and will present a general algorithm for constructing conservative interface conditions. The variables x and y are used as the spatial variables in the partial differential equations. Since, in the general case, one would need to solve the equations on a curvilinear grid, all equations would be transformed to curvilinear coordinates before applying the finite-difference algorithm. In each development the partial differential equation is sufficiently generally so that the method can be applied to the original or the transformed equation without change. For simplicity, mixed derivative terms which are normally lagged in the single grid case and source terms are not included in the partial differential equations.

Implicit Methods The fundamental concepts are quite simple and can be demonstrated by considering the one-dimensional equation

$$u_t = Lu \quad , \tag{1}$$

where L is a differential operator of the form

$$Lu = au_x + bu_{xx}.$$

Since implicit methods generally require linearization of the difference equations, it may as well be assumed that L is linear. Suppose that two grids G_1 and G_2 are given on the intervals [a,d] and [c,b], respectively, where

$$a < c < d < b.$$

If M denotes the usual second order difference approximation of L, then the Crank-Nicolson equation can be written as

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{2}(Mu_i^{n+1} + Mu_i^n). \tag{2}$$

Here i is the spatial index, n is the temporal index, and Δt is the step length. Now suppose values on G_1 and G_2 are known at level n and values at level n+1 are to be computed on G_1. While solution values needed in (2) at x=d can be interpolated from G_2 for level n, the corresponding values at level n+1 are unavailable. If these unknown values are replaced by the values at level n, then the local truncation error at the neighboring interior point is increased by a term on the order of $O(\Delta t^2/\Delta x^2)$. The value Δx represents the spatial grid spacing on G_1, or the spacing at x=d in the case of a nonuniform grid. In any event, when Δx is small, this lagging of solution values will seriously degrade the temporal accuracy of the approximation. The error can be reduced by following a particular order in updating the solution values at the interior grid boundary points. The correct sequence of computations is indicated in the following steps.

1. Calculate u^{n+1} on G_1 with level n values at x=d.
2. Calculate u^{n+1} on G_2 with level n+1 values at x=c.
3. Calculate u^{n+2} on G_2 with level n+1 values at x=c.
4. Calculate u^{n+2} on G_1 with level n+2 values at x=d.

Now the error induced by using the previous value at x=d in step 1 is offset by the use of the advanced value in step 4. In fact, the local truncation error at the neighboring interior point is increased by a term of order $O(\Delta t^3/\Delta x^2)$ when the solution is advanced from level n to n+2. The same error reduction would also occur at x=c.

Clearly, this four-step alternating grid scheme is only a partial solution. Unless the solution exhibited a linear growth or decay, there would still be points with a local truncation error of order one whenever $\Delta t \approx \Delta x$. However, this does not necessarily mean that the global error in the numerical solution would be increased to that order. The actual error in the solution would also depend on other factors such as the extent of the overlap. Note that the same updating procedure could be applied to implicit methods other than the Crank-Nicholson method, but the reduction in local truncation error would not be the same.

The alternating grid concept has also been used in the development of another method for implementing implicit algorithms on composite grid systems. This method also alternately employs the forward difference explicit equation

$$u_i^{n+1} = u_i^n + \Delta t \, Mu_i^n \tag{3}$$

and the backward difference implicit equation

$$u_i^{n+1} = u_i^n + \Delta t \; Mu_i^{n+1} \; . \tag{4}$$

The computational sequence is illustrated in the following four-step procedure which advances the solution from level n to level n+2.

1. Calculate u^{n+1} on G_1 using (3).
2. Calculate u^{n+1} on G_2 using (4).
3. Calculate u^{n+2} on G_2 using (3).
4. Calculate u^{n+2} on G_1 using (4).

The method alternates the explicit and implicit calculations in the same manner as the well-known hopscotch algorithm. Thus, the name hopscotch will be associated with this method. The method has several desireable properties. All values needed at the grid boundaries c and d can be computed by interpolation from solution values at the correct time level. The overall method is second-order accurate in time and unconditionally stable. This fact follows by noting that the combined sequence of (3) followed by (4) is equivalent to a Crank-Nicolson step with step length of $2\Delta t$.

The consequences of lagging solution values at grid boundaries can be even more serious for multi-dimensional problems. Suppose for example that the operator L in (1) is defined as

$$Lu = au_x + bu_y + cu_{xx} + du_{yy} \; .$$

Let M_x and M_y denote the difference approximations of the x and y derivative parts of L. If the parabolic equation (1) is solved by an ADI method, such as Peaceman-Rachford, the algorithm becomes

$$u^{n+1/2} = u^n + \frac{\Delta t}{2}(M_x u^{n+1/2} + M_y u^n) , \tag{5a}$$

$$u^{n+1} = u^{n+1/2} + \frac{\Delta t}{2}(M_x u^{n+1/2} + M_y u^{n+1}). \tag{5b}$$

Now the error that occurs in the first step of the algorithm is further magnified in the second step. This argument can be made more precise by noting that the Peaceman-Rachford ADI method is a perturbation of the two-dimensional Crank-Nicolson method with a perturbation term

$$\frac{\Delta t^2}{4} M_x^2 M_y^2 (u^n - u^{n+1}) .$$

A lagged value in this term produces a truncation error term on the order of $O(\Delta t^3/\Delta x^2 \Delta y^2)$. The alternating grid procedure would reduce this to $O(\Delta t^4/\Delta x^2 \Delta y^2)$. The one-dimensional hopscotch algorithm would not be a computationally efficient method for two-dimensional problems. However, the same effect can be realized by inserting additional steps in the ADI algorithm. The procedure is again demonstrated using two grids G_1 and G_2. Note that equation (5a) can be written as

$$v^{n+1/2} = u^n + \frac{\Delta t}{2}(M_x u^n + M_y u^n) \tag{6a}$$

$$u^{n+1/2} = v^{n+1/2} + \frac{\Delta t}{2}(M_x u^{n+1/2} - M_x u^n), \tag{6b}$$

while (5b) can be replaced by

$$v^{n+1} = u^{n+1/2} + \frac{\Delta t}{2} (M_x u^{n+1/2} + Mu^{n+1/2}) \tag{6c}$$

$$u^{n+1} = v^{n+1} + \frac{\Delta t}{2} (M_y u^{n+1} - M_y u^{n+1/2}) \ . \tag{6d}$$

These split forms would require additional computations, and should only be used to generate interpolated values at interior grid boundaries. The following steps illustrate one possible method of computation.

1. Calculate $v^{n+1/2}$ on G_1 using (6a)
2. Calculate $u^{n+1/2}$ on G_2 using (5a)
3. Calculate $u^{n+1/2}$ on G_1 using (6b)
4. Calculate v^{n+1} on G_2 using (6c)
5. Calculate u^{n+1} on G_1 using (5b)
6. Calculate u^{n+1} on G_2 using (6d)

Note that at each step the necessary boundary values for one grid can be interpolated from values at the correct level on the other grid.

The efficacy of the alternating grid and hopscotch methods is exhibited in the solution of a one-dimensional model problem. The parabolic equation

$$u_t + (u-c)u_x = \mu \, u_{xx} \tag{7}$$

has an exact solution

$$u(x,t) = \frac{1}{2}\left(1 - \tanh \frac{2(x+1)+(2c-1)t}{4\mu}\right).$$

This equation is solved on the interval [-2,2] with the exact initial value at t=0 and boundary values at x=-2 and x=2. A second order linearization and the usual central difference approximations are used. An overlapping set of two grids on the intervals [-2,.125] and [-.125,2] is constructed. The solution for values of c=0.4 and μ=0.05 is computed using three different methods. The form of the actual solution indicates that an increase in t would result in a translation of the graph in the positive x direction. When a numerical solution is computed with the Crank-Nicolson equation (2) and the values at x=±.125 are lagged, there is a marked deviation between the numerical and analytic solutions as they pass through the overlap interval. Although the numerical solution lags behind the actual solution, they are qualitatively similar with no indication of instability in the numerical solution. A comparison of the solutions at various times is plotted in Figure 1. The lag in the numerical solution is eliminated when the alternating grid method is used. A careful examination of Figure 2 reveals an anomaly in the graph at the grid points adjacent to the interior boundary points x=±.125. This is more evident on the enlargement in Figure 3. Note that the problem occurs only at the points where the exceptional difference approximation is employed. The most accurate numerical solution for this example is calculated using the hopscotch algorithm. That solution appears in Figure 4. In all of these figures, linear interpolation was used to determine solution values at grid boundaries.

Methods for Conservation Laws. If the conservation equation

$$u_t + [f(u)]_x = 0 \tag{8}$$

is solved on a composite grid system, then there must be some means of transferring the flux f(u) from one grid to the other. There are two feasible alternatives. Either the solution u can be calculated by interpolation and then f(u) evaluated, or f(u) can be interpolated directly from the flux values on the other grid. The conservative difference schemes which will be discussed require interpolation of fluxes. However, before proceeding in that direction, a comparison of the two interpolation techniques will be included.

Suppose a solution value u^* at a boundary point of grid G_1 is computed by linear interpolation from the solution values u_{i-1} and u_i defined on grid G_2. Then an interpolation formula of the form

$$u^* = \alpha u_{i-1} + \beta u_i, \ \alpha + \beta = 1,$$

holds, and the flux can be evaluated as $f(u^*)$. Now if u_0 is the actual value of the solution at the boundary point of G_1, and hence the true flux value is $f(u_0)$, then the interpolation procedure introduces an error as is seen in the following expansion.

$$f(\alpha u_{i-1}+\beta u_i) = f(u_0)+f_u(u_0)(\alpha u_{i-1}+\beta u_i-u_0)$$

$$+ \ \ldots$$

Since an expansion at the boundary point x_0 yields

$$\alpha u_{i-1} + \beta u_i - u_0 = \frac{1}{2} u_{xx0}(\alpha(x_{i-1}-x_0)^2 + \beta(x_i-x_0)^2),$$
$$+ \ldots$$

the leading term in the local truncation error is

$$\frac{1}{2} f_u(u_0)u_{xxo}(\alpha(x_{i-1}-x_0)^2 + \beta(x_i-x_0)^2).$$

Whenever the option of interpolating the fluxes is selected, then the boundary flux f^* is calculated directly as

$$f^* = \alpha\, f(u_{i-1}) + \beta f(u_i).$$

Expanding about the solution u_0, and noting the additional second order term,

$$\alpha\, f(u_{i-1}) + \beta\, f(u_i) = f(u_0) + f_u(u_0)(\alpha u_{i-1}+\beta u_i-u_0)$$
$$+ \frac{1}{2} f_{uu}(u_0)(\alpha(u_{i-1}-u_0)^2 + \beta(u_i-u_0)^2)$$
$$+ \ldots \quad .$$

The leading term in the local truncation error now has the form

$$\frac{1}{2} (f_u(u_0)u_{xxo} + f_{uu}(u_0)u_{xo})(\alpha(x_{i-1}-x_0)^2 + \beta(x_i-x_0)^2)).$$

It is clear that both procedures give an interpolation error which is $O(\Delta x^2)$.

There are many conservative finite-difference algorithms for solving conservation laws of the type (8). Most of the basic algorithms of practical interest can be written as

$$u_i^{n+1} - u_i^n = g(u_{i+1}^n, u_i^n) - g(u_i^n, u_{i-1}^n). \tag{9}$$

For notational convenience, let

$$g_{i+1/2} = g(u_{i+1}, u_i) \ ,$$

with the implication here being that $g_{i+1/2}$ is an approximation at $x_{i+1/2}$. This notation is appropriate for the central difference approximations such as the Lax or Lax-Wendroff schemes. When one-sided or upwind differencing is used, the fractional index i+1/2 would be replaced by i or i+1.

Given a grid with gridpoints x_i, i=0,1,...,I, the discrete conservation property states that

$$\sum_{i=1}^{I-1} u_i^{n+1} = \sum_{i=1}^{I-1} u_i^n + g_{I-1/2}^n - g_{1/2}^n \ .$$

The same result can be obtained from (8) by using numerical integration from $x_{1/2}$ to $x_{I-1/2}$ and the flux approximation determined by g. It is this derivation that will be used in the composite grid approach.

Let G_1 and G_2 be grids defined on the intervals [a,d] and [c,b], a<c<d<b. The grid G_1 has points x_i, i=0,1,...,I and grid spacing Δx, and G_2 has points y_j, j=0,1,...,J and spacing Δy. The difference equations will be written in terms of scaled solution values v and w defined by

$$v = u\Delta x \quad \text{and} \quad w = u\Delta y$$

On G_1, the difference equation has the form

$$v_i^{n+1} - v_i^n = h_{i+1/2}^n - h_{i-1/2}^n$$

and on G_2,

$$w_j^{n+1} - w_j^n = k_{j+1/2}^n - k_{j-1/2}^n \quad .$$

There is good reason for writing the equations in this form. First of all, the grid spacing need not be included in the interpolation formulas, but more importantly, this is the required form of the difference equations when computing on moving grids.

The correct interface conditions can now be derived by extending the grid functions to piecewise linear functions and integrating. Suppose a value $k_{1/2}$ is needed. Then the interval [a,b] is partitioned into two subintervals $[a,y_{1/2}]$ and $[y_{1/2},b]$. If $y_{1/2}$ lies in the interval $[x_{i-1/2},x_{i+1/2}]$, and $h_{i-1/2}$ and $h_{i+1/2}$ are known, then the value for $k_{1/2}$ can be calculated from the integral property

$$\int_{x_{1/2}}^{y_{1/2}} h_x + \int_{y_{1/2}}^{y_{J-1/2}} k_x = k_{J-1/2} - h_{1/2} \quad .$$

Assuming that h and k are piecewise linear, it is easily seen that the needed value is the linear interpolant defined as

$$k_{1/2} = \alpha\, h_{i-1/2} + \beta\, h_{i+1/2} \ ,$$

where

$$\alpha = \frac{x_{i+1/2}-y_{1/2}}{x_{i+1/2}-x_{i-1/2}} \ , \ \beta = \frac{y_{1/2}-x_{i-1/2}}{x_{i+1/2}-x_{i-1/2}} \ .$$

By the same argument, the interval [a,b] can be partitioned into $[a,x_{I-1/2}]$ and $[x_{I-1/2},b]$ and the interpolation formula for the value $h_{I-1/2}$ is

$$h_{I-1/2} = \alpha\, k_{j-1/2} + \beta\, k_{j+1/2}$$

where

$$\alpha = \frac{y_{j+1/2}-x_{I-1/2}}{y_{j+1/2}-y_{j-1/2}}\ ,\ \beta = \frac{x_{I-1/2}-y_{j-1/2}}{y_{j+1/2}-y_{j-1/2}}\ .$$

The same linear interpolation would be used on a nonuniform grid. Of course, the scaling factor would vary from point to point. An interpolation formula could also have been derived using the original equation (9), however the difference in grid spacing on G_1 and G_2 would have resulted in the appearance of a scaling factor in the interpolation formula.

A modification of this approach can be used to develop conservative interface conditions for two and three-dimensional problems. The general two-dimensional conservation law is

$$u_t + f_x + g_y = 0,$$

where x and y are now the spatial variables. A difference approximation has the form

$$v^{n+1}_{i,j} = v^n_{i,j} + h^n_{i+1/2,j} - h^n_{i-1/2,j} + k^n_{i,j+1/2} - k^n_{i,j-1/2}. \quad (10)$$

The grid function v is the product of the solution u and the Jacobian (or cell area), and the values of h and k are, up to a scalar factor, flux values in

the direction of the curvilinear coordinate lines. Let G_1 and G_2 be overlapping grids and suppose values of h are required along the i = 1/2 grid line of G_1. For now, it is assumed that the endpoints of the grid line are on the boundary of the physical region. The points of G_1 along the grid line are labeled using parameter values from any convenient parameterization. Thus, let

$$p_j = (x_{1/2,j},\ y_{1/2,j}),\ j=0,1,\ldots,J,$$

while points of intersection of the i = 1/2 grid line of G_1 with all grid lines of G_2 are ordered and labeled

$$q_\ell = (x_{i,j+\delta},\ y_{i,j+\delta}) \text{ or } (x_{i+\delta,j},\ y_{i+\delta,j}),\ \ell=0,1,\ldots,L$$

where δ denotes a fractional index between 0 and 1, and i,j are the indices of some point in G_2. The first step in the transfer of flux values from G_2 to G_1 is to define flux values at the points q_ℓ. If q_ℓ lies on an i=constant grid line, as in the first case above, then a value h^*_ℓ is computed by interpolating the grid function k. On the other hand, if q_ℓ lies on a j=constant grid line, then h^*_ℓ is computed by interpolating h. Now the i = 1/2 grid line divides the physical region into two parts, one covered by G_1 and the other covered by a subset of G_2. If piecewise linear flux functions are constructed along the grid lines in each subregion and an integration of the flux derivatives over the complete region is performed, then the conservation property requires that

$$\frac{h_0}{2} + \sum_{j=1}^{J-1} h_j + \frac{h_J}{2} = \frac{h^*_0}{2} + \sum_{\ell=1}^{L-1} h^*_\ell + \frac{h^*_L}{2}\ . \tag{11}$$

Here the i and n indices in (10) have been suppressed.

The interpolation formula will be defined using a set of basis functions. Two additional parameter values are introduced by extrapolating from the parametric interval. Let $q_{-1} = 2q_0 - q_1$ and $q_{L+1} = 2q_L - q_{L-1}$. Let ψ_ℓ be the piecewise linear function, with knots q_ℓ, $\ell=-1,0,\ldots,L+1$, defined as

$$\psi_\ell(q_m) = \begin{cases} 1, & m=\ell \\ 0, & m\neq\ell \end{cases}$$

where $\ell=0,1,\ldots,L$, and $m=-1,0,\ldots,L+1$. The following integrals can be easily computed from the parametric values of the points along the grid line.

$$\Delta_\ell = \int_{q_{-1}}^{q_{L+1}} \psi_\ell$$

$$\Delta_{\ell,0} = \int_{p_0}^{p_{1/2}} \psi_\ell$$

$$\Delta_{\ell,j} = \int_{p_{j-1/2}}^{p_{j+1/2}} \psi_\ell \quad , \; j=1,2,\ldots,J-1$$

$$\Delta_{\ell,J} = \int_{p_{J-1/2}}^{p_J} \psi_\ell$$

These integrals are used in calculating the coefficients of the interpolation formulas. The formulas can now be written as

$$h_j = 2 \sum_{\ell=0}^{L} \frac{\Delta_{\ell,j}}{\Delta_\ell} h_\ell^* \text{ for } j=0 \text{ and } j=J,$$

and (12)

$$h_j = \sum_{\ell=0}^{L} \frac{\Delta_{\ell,j}}{\Delta_\ell} h_\ell^* \text{ for } j=1,2,\ldots,J-1 \quad .$$

The fact that property (11) holds is readily verified.

$$\frac{h_0}{2} + \sum_{j=1}^{J-1} h_j + \frac{h_J}{2} = \sum_{j=0}^{J} \sum_{\ell=0}^{L} \frac{\Delta_{\ell,j}}{\Delta_\ell} h_\ell^*$$

$$= \sum_{\ell=0}^{L} h_\ell^* \frac{1}{\Delta_\ell} \sum_{j=0}^{J} \Delta_{\ell,j}$$

$$= \frac{h_0^*}{2} + \sum_{\ell=1}^{L-1} h_\ell^* + \frac{h_L^*}{2}$$

A few remarks are sufficient to indicate how the same interpolation method can be employed in more general composite grid configurations. If interpolation is required on a boundary component consisting of several i=constant and j=constant segments, then each boundary segment could be treated separately with either an h value or a k value calculated from equations (12). However, the extrapolated parameter values would not be used in computing the coefficients, but instead a single parameterization would be defined for the entire boundary component. If the boundary component were a closed contour in the interior of the physical region, then the special boundary interpolation formulas for j=0 and j=J in (12) would be unnecessary.

The selection of a set of piecewise linear basis functions to define the interpolation coefficients may be changed with only slight modification. One could just as easily use piecewise constant functions or use higher degree polynomials such as quadratics or cubics. The degree of interpolation may have differing effects on the numerical solution. The use of a piecewise

constant basis may produce shock-like discontinuities, whereas a linear basis would tend to smear out any discontinuities in the solution.

The conservative finite-difference scheme of MacCormack is used to solve the parabolic equation (7) which can be written in conservation form as

$$u_t + (1/2(u-c)^2 - \mu\, u_x)_x = 0.$$

This equation is solved on two overlapping grids on the intervals [-2,.25] and [-.25,2] with the same values of c=0.4 and μ=0.05 as in the previous section. The numerical dissipation in the MacCormack scheme permits the use of a coarser grid than was used for the implicit methods. Figure 4 contains the solution plotted for various values of t. For this example, there was no noticeable difference between solutions computed with flux values interpolated and those computed with interpolation of solution values.

Conclusions. The accuracy of the transient solution of a hyperbolic or parabolic partial differential equation is dependent upon the procedures used to transfer information between grids in a composite grid system. The error in the numerical solution can be reduced by using the techniques developed here. While the attempt has been to construct algorithms that are easy to implement, the degree of difficulty would ultimately be linked to the complexity of the grid structure.

References

1. E. H. Atta, "Component-Adaptive Grid Interfacing," AIAA Paper No. 81-0382, January 1981.

2. E. H. Atta and J. Vadyak, "A Grid Interfacing Zonal Algorithm for Three Dimensional Transonic Flows About Aircraft Configurations," AIAA Paper No. 82-1017, June 1982.

3. J. A. Beneck, P. G. Buning, and J. L. Steger, "A 3-D Chimera Grid Embedding Technique," AIAA Paper No. 85-1523-CP, July 1985.

4. M. J. Berger, "On Conservation at Grid Interfaces," ICASE Report No. 84-43, September 1984.

5. B. Kreiss, "Construction of a Curvilinear Grid," SIAM J. Sci. Stat. Comput., 4(1983), 270-279.

6. C. W. Mastin and H. V. McConnaughey, "Computational Problems on Composite Grids," AIAA Paper No. 84-1611, June 1984.

7. M. M. Rai, "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations," AIAA Paper No. 84-0164, January 1984.

8. G. Starius, "Composite Mesh Difference Methods for Elliptic Boundary Value Problems," Numer. Math., 28(1977), 243-258.

9. G. Starius, "On Composite Mesh Difference Methods for Hyperbolic Differential Equations," Numer. Math., 35 (1980), 241-255.

10. J. L. Steger and P. G. Buning, "Developments in the Simulation of Compressible Inviscid and Viscous Flow on Supercomputers," NASA TM-86674, February 1985.

11. D. S. Thompson, "A Mesh Embedding Approach for Prediction of Transonic Wing/Body/Store Flow Fields," NASA CP-2201, October 1981, 403-427.

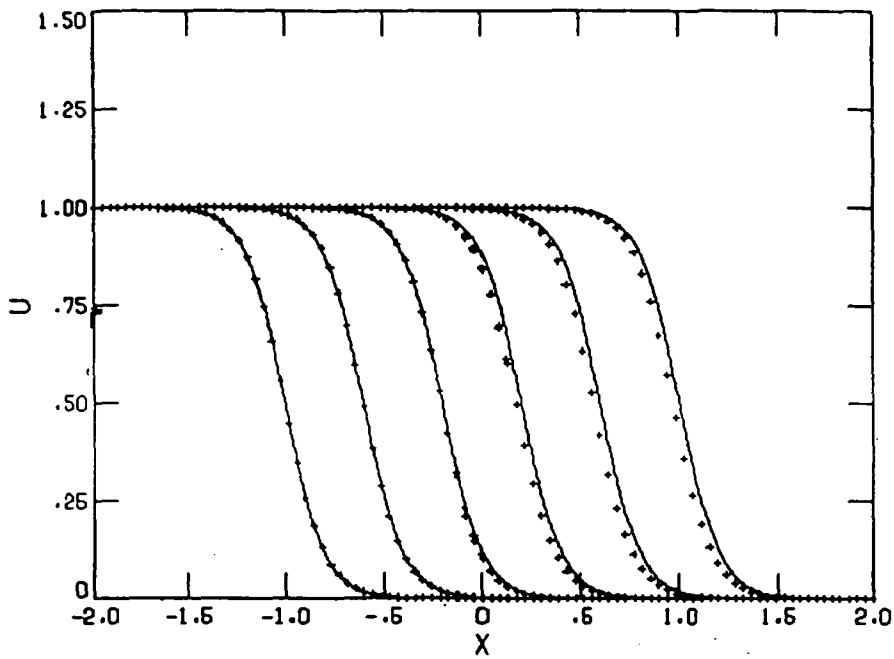

Figure 1. Implicit solution with boundary values lagged

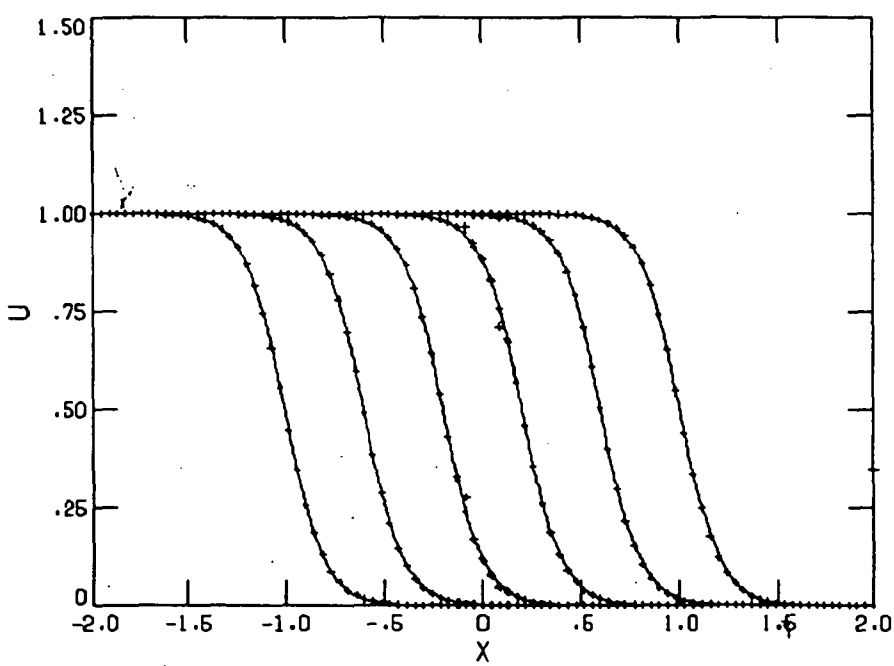

Figure 2. Implicit solution with alternating grid update at boundary

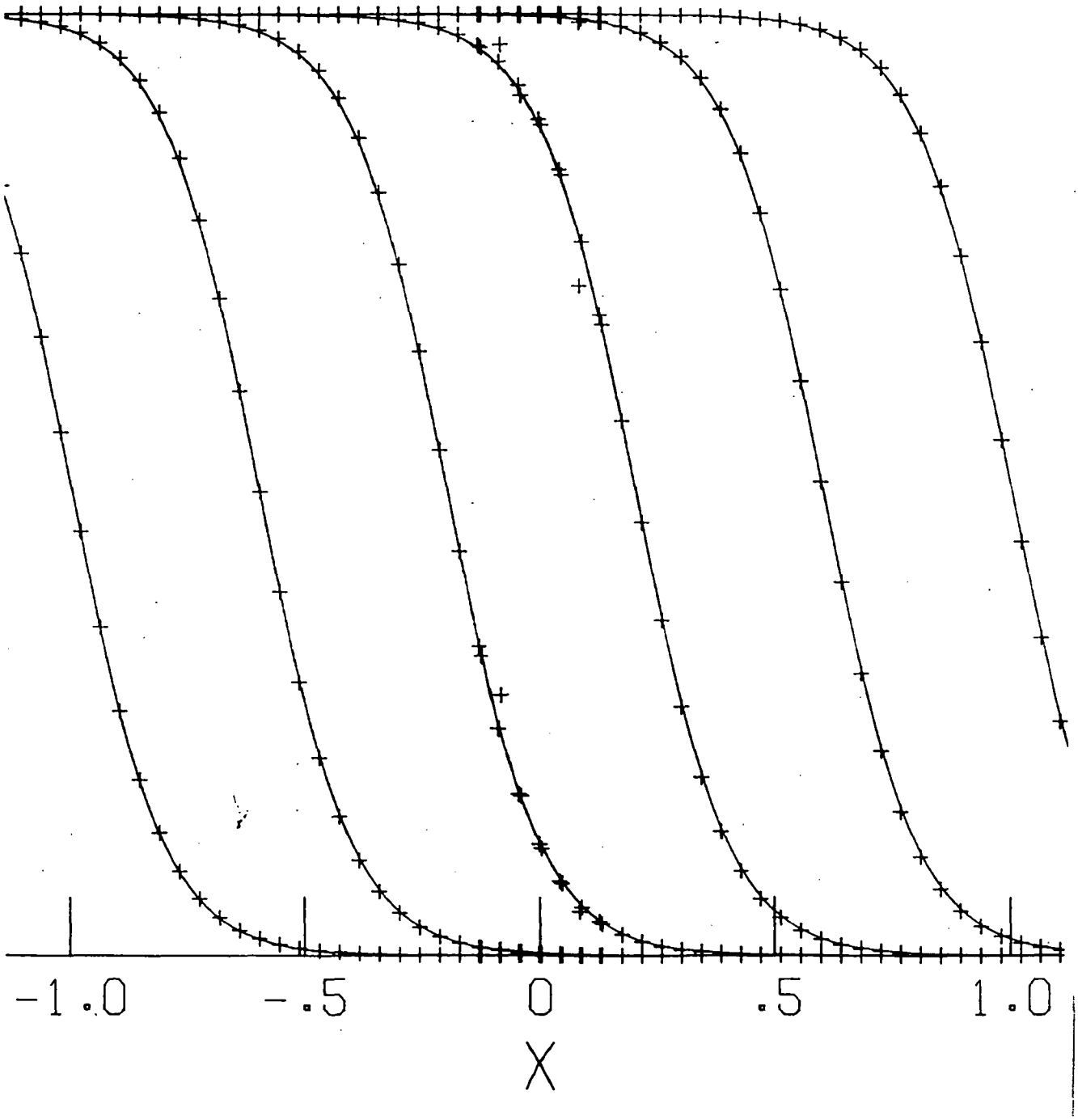

Figure 3. Spurious values resulting from alternating grid update

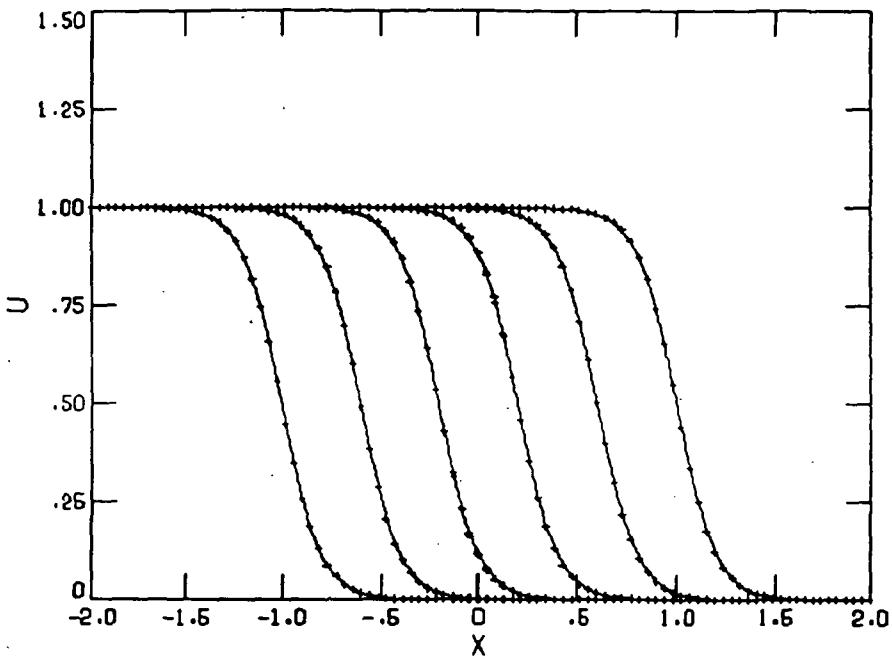

Figure 4. Implicit solution with grid hopscotch

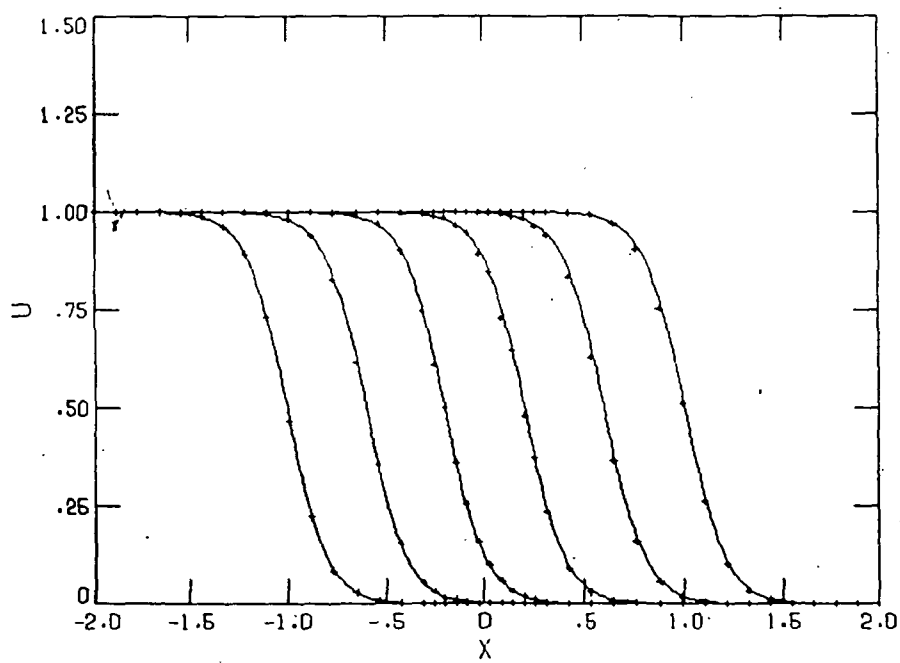

Figure 5. Conservative explicit solution