JOURNAL OF COMPUTATIONAL PHYSICS 23, 124–134 (1977)

Mesh Refinement and Local Inversion of Elliptic Partial Differential Equations*

JAMES M. HYMAN

Theoretical Division, University of California, Los Alamos Scientific Laboratory, Los Alamos, New Mexico 87545

Received September 15, 1975; revised July 7, 1976

A fast numerical method is developed to find an approximate solution to a general class of mildly nonlinear elliptic partial differential equations with Dirichlet boundary conditions in one, two, and three dimensions. The method is based on a local mesh refinement technique which provides an initial guess for iterative algorithms, and can be used to refine the mesh in multigrid methods.

1. INTRODUCTION

The local inversion method, LIM, will first be described for the one-dimensional boundary value problem

$$\begin{aligned} u'' + cu &= f(x), \\ u(0) = u_0, &\qquad u(1) = u_1. \end{aligned} \tag{1}$$

The basic idea is to expand a solution known on a coarse mesh to a fine mesh while maintaining high accuracy. The generalization of the method to higher dimensions and other equations will be discussed later.

Suppose the solution to Eq. (1) is known at the points x_{2i} but not x_{2i+1}, $i = 0, 1, 2,..., N$ on the evenly spaced grid shown in Fig. 1. A finite difference approximation of Eq. (1) by Taylor series expansion of the solution about the point x_{2i+1} is

$$\frac{1}{h^2}(u_{2i} - 2u_{2i+1} + u_{2i+2}) + cu_{2i+1} = f_{2i+1} + \frac{h^2}{12}u'''',$$

or

$$u_{2i+1} = \frac{1}{2 - h^2c}(u_{2i} + u_{2i+2} - h^2f_{2i+1}) - \frac{h^4u''''}{12(2 - h^2c)}, \tag{2a}$$

* Work performed under the auspices of the U.S. Energy Research Development Administration.
The United States Government's right to retain a nonexclusive royalty-free license in and to copyright covering this paper is acknowledged.

ISSN 0021-9991

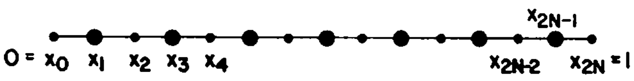


FIG. 1. The unit interval is divided into $2N$ equally spaced mesh points. The solution is known at x_{2i} but not at x_{2i+1}, $i = 0, 1, 2, \ldots N$.

where $u_i = u(x_i)$, $f_i = f(x_i)$, and $h = x_{i+1} - x_i$. Equation (2a) gives us an approximate solution to Eq. (1) at x_{2i+1}. This approximation can be improved by estimating the leading error term and eliminating it. u'''' is easily approximated by differentiating Eq. (1) twice, solving for u'''' and using a centered difference scheme to estimate f''_{2i+1}. The resulting equation is

$$u_{2i+1} = \frac{u_{2i} + u_{2i+2} - (h^2/12)[f_{2i} + f_{2i+2} + (10 - ch^2) f_{2i+1}]}{2 - ch^2 + (c^2h^4/12)} + O(h^6)\ldots.. \quad (2b)$$

This refinement process can now be repeated to approximate the solution at the midpoints of the x_i until the desired resolution is obtained. The method is self-starting since the original coarse grid may be the boundary conditions. When used as a self-starting method, the error remains approximately

$$O(h^6) = (h^6/1440)(5f'''' - 2u'''''') = 0.00001(5f'''' - 2u''''''). \quad (3)$$

This error can be estimated by finite differences at selected points on the final mesh. If the solution is relatively smooth, the error may be quite acceptable. Otherwise, another method must be used to start the LIM on a finer original mesh. (Remark: By using more function values, or deferred corrections [6], the error can be reduced even further.)

The refinement does not depend heavily on the constancy of c or the linearity of the original equation. The difference approximation to the equation needs only to be solved for u_{2i+1} as a function of its neighbors, where the solution is known. For complicated nonlinear problems the LIM can be used to give a good initial guess of the solution. This approximation can be improved using a standard iterative method such as nonlinear SOR. In two- and three-dimensional problems, the savings in using a combination of methods is substantial.

2. Two-Dimensional Problems

The LIM in two dimensions will be described for the equation

$$\begin{aligned} \Delta u + cu &= f(x, y) \quad \text{in } D^2 \\ u &= g(x, y) \quad \text{on } \partial D^2. \end{aligned} \quad (4)$$

D^2 is the unit square and ∂D^2 is its boundary. Suppose that the solution is known at the points (x_{2i}, y_{2j}) $i, j = 0, 1, 2, \ldots, N$ on the evenly spaced $2N \times 2N$ grid in Fig. 2a.

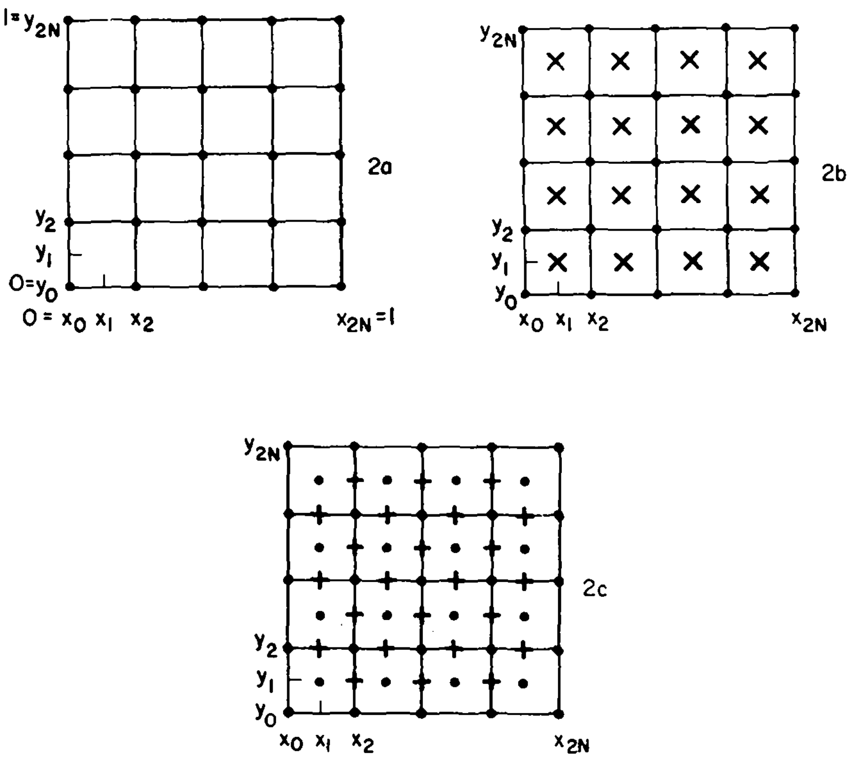


FIG. 2. (a) The solution to Eq. (4) is known at the $\cdot$ points, (x_{2i}, x_{2j}) $i, j = 0, 1, 2, \ldots N$. (b) Equation (5a) provides an approximate solution at the $\times$ points, (x_{2i+1}, x_{2j+1}) $i, j = 0, 1, 2, \ldots N - 1$. (c) Equation (5b) uses the $\times$ point to approximate the solution at the $+$ points, (x_{2i+1}, x_{2j}) or (x_{2i}, x_{2j+1}) $i, j = 0, 1, 2, \ldots N$.

At the point (x_{2i+1}, y_{2j+1}) by rotating the standard five-point difference approximation to Eq. (4) by 90° we have

$$(1/2h^2)(u_{2i,2j} + u_{2i+2,2j} + u_{2i,2j+2} + u_{2i+2,2j+2} - 4u_{2i+1,2j+1}) + cu_{2i+1,2j+1}$$
$$= f_{2i+1,2j+1} + O(h^2),$$

or

$$u_{2i+1,2j+1} = \frac{u_{2i,2j} + u_{2i+2,2j} + u_{2i,2j+2} + u_{2i+2,2j+2} - 2h^2 f_{2i+1,2j+1}}{4 - 2h^2c} + O(h^4). \quad (5a)$$

These points are called $\times$ points since they use the four corners of the surrounding grid points to approximate the solution in the center. Figure 2b shows their location graphically.

The remaining points (x_{2i+1}, y_{2j}) and (x_{2i}, y_{2j+1}) use the standard five point $+$ difference scheme to approximate the Laplacian. At these points the finite difference analog of Eq. (4) is

$$u_{k,l} = \frac{u_{k-1,l-1} + u_{k+1,l-1} + u_{k-1,l+1} + u_{k+1,l+1} - h^2 f_{k,l}}{4 - h^2 c} + O(h^4), \tag{5b}$$

where

$$k = 2i+1, \; l = 2j \quad \text{or} \quad k = 2i, \; l = 2j+1, \qquad i,j = 1, 2, \ldots, N-1,$$

Equation (4) is now solved on a $4N \times 4N$ grid. The process can be repeated refining the solution to an $8N \times 8N$ grid.

This method exactly reproduces two-dimensional cubic polynomial solutions and works well for equations with smooth solutions. A higher order LIM which reproduces two-dimensional quintics when $c \neq 0$ uses the two equations

$$u_{2i+1,2j+1} = \frac{1}{4-2bc}\left[\bar{u}_\times - \frac{1}{3c}(8 + 6bc - 10h^2ac) f_{2i+1,2j+1} + \frac{2}{3c}(2 - h^2ac)\bar{f}_+ - \frac{1}{6c}(4 + h^2ac)\bar{f}_\times\right], \tag{6a}$$

where

$$\begin{aligned}
\bar{u}_\times &= u_{2i,2j} + u_{2i+2,2j} + u_{2i,2j+2} + u_{2i+2,2j+2},\\
\bar{f}_\times &= f_{2i,2j} + f_{2i+2,2j} + f_{2i,2j+2} + f_{2i+2,2j+2},\\
\bar{f}_+ &= f_{2i+1,2j+2} + f_{2i+1,2j} + f_{2i+2,2j+1} + f_{2i,2j+1},\\
a &= 2/(12 + ch^2),\\
b &= h^2 - (1/12)\, ch^4 + (1/24)\, h^6 c^2 a,
\end{aligned}$$

and

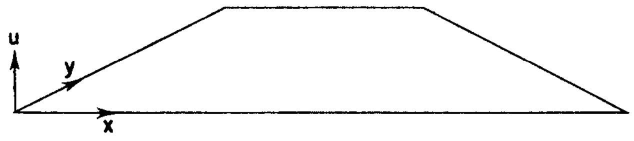
$$u_{k,l} = \frac{1}{4-bc}\left[\bar{u}_+ + \frac{1}{3c}(2 - 3bc + 5h^2ac) f_{k,l} + \frac{1}{12c}(2 - h^2ac)\bar{f}_+ - \frac{1}{3c}(1 + h^2ac)\bar{f}_\times\right], \tag{6b}$$

with

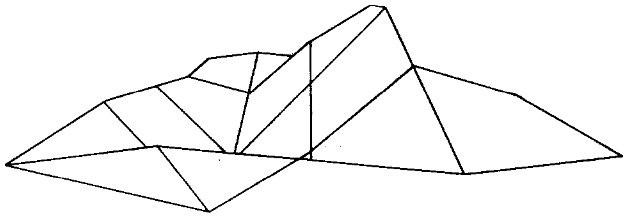
$$\begin{aligned}
\bar{u}_+ &= u_{k,l+1} + u_{k,l-1} + u_{k+1,l} + u_{k-1,l},\\
\bar{f}_+ &= f_{k,l+1} + f_{k,l-1} + f_{k+1,l} + f_{k-1,l},\\
\bar{f}_\times &= f_{k-1,l-1} + f_{k+1,l-1} + f_{k-1,l-1} + f_{k+1,l+1},\\
a &= 2/(12 + ch^2),\\
b &= h^2 - (1/12)\, ch^4 + (1/24)\, h^6 c^2 a,
\end{aligned}$$

where

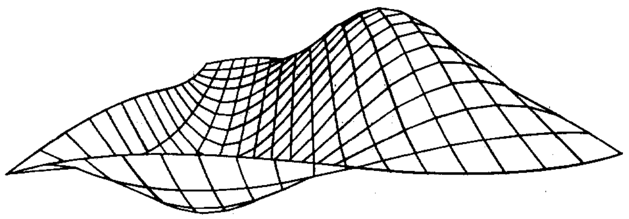
$$k = 2i+1, \; l = 2j \quad \text{or} \quad k = 2i, \; l = 2j+1, \qquad i,j = 1, 2, \ldots, N-1.$$



3a



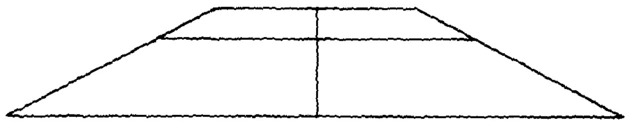
3c



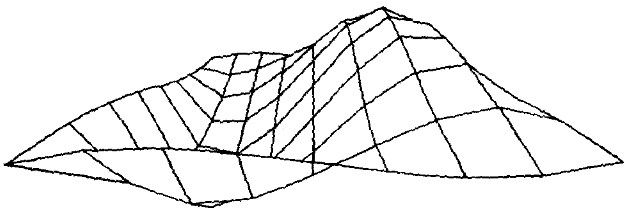
3e

FIG. 3. (a) The original solution is known only on the boundary of the unit square. (b) Using the boundary data and $f(\frac{1}{2}, \frac{1}{2})$ an approximate solution is found at $(\frac{1}{2}, \frac{1}{2})$. (c) Equations (6a, b) refine the mesh to 4 × 4. (d) The mesh is refined to 8 × 8. (e) The mesh is refined to 16 × 16. (f) The mesh is refined to 32 × 32.

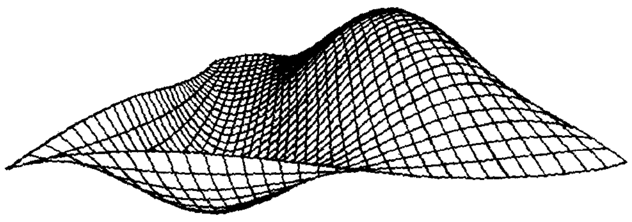
As in one dimension, the LIM is self-starting. However, for solutions with sharp gradients more accuracy is obtained when it is combined with other standard methods such as SOR, Hockney's or Buneman's method [2, 4, 5]. It has also been used successfully in the mesh refinement stage of Brandt's multigrid method [1]. An initial approximation of the solution on a coarse mesh may also be found using a high-order LIM developed by using many boundary points and more internal function values. It was found rarely necessary to start with more than 9 internal points when using Eqs. (6) to refine the mesh.



3b



3d



3f

FIG. 3. *Continued*

Figure 3 displays how the self-starting LIM using Eqs. (6) refines the solution to Eq. (4) on a 32×32 grid with $c = -1$ and the solution

$$u = -(2x - 1)(2y - 1)(x + y)(x + y - 1)(x + y - 2).$$

The solution is a two-dimensional quintic and therefore LIM is exact.

In Section 5 a nontrivial example is analyzed comparing Eqs. (5) and (6).

3. Three-Dimensional Problems

We again describe the LIM for Poisson's Equation with Dirichlet boundary conditions,

$$\begin{aligned} \Delta u + cu &= f(x, y, z) \qquad \text{in } D^3, \\ u &= g(x, y, z) \qquad \text{on } \partial D^3. \end{aligned} \tag{7}$$

Refining the mesh in three dimensions is similar to the algorithms described in one and two dimensions. In one dimension a single finite difference formulation of the equation was necessary to solve for the unknown points. In two dimensions two different finite difference schemes were necessary to solve for the unknown points. In three dimensions four finite difference formulations of Eq. (7) are needed to refine the mesh.

Assume that the solution to Eq. (7) is known on a $2N \times 2N \times 2N$ evenly spaced grid at the points (x_{2i}, y_{2j}, z_{2k}) $i, j, k = 0, 2, \ldots, N$. We first approximate the solution at the midpoint of each cube of points where the solution is known i.e., the points $(x_{2i+1}, y_{2j+1}, z_{2k+1})$. At these points

$$\begin{aligned} u_{2i+1,2j+1,2k+1} = \frac{1}{8 - 4h^2c} (&u_{2i,2j,2k} + u_{2i,2j+2,2k} + u_{2i,2j,2k+2} \\ &+ u_{2i,2j+2,2k+2} + u_{2i+2,2j,2k} + u_{2i+2,2j+2,2k} \\ &+ u_{2i+2,2j,2k+2} + u_{2i+2,2j+2,2k+2} - 4h^2 f_{2i+1,2j+1,2k+1}) + O(h^4). \end{aligned} \tag{8a}$$

The solution at the remaining points are at the center of an octahedra of known points and can be approximated by one of the following difference schemes.

$$\begin{aligned} u_{l,m,n} = \frac{1}{8 - 2h^2c} (&2u_{l-1,m,n} + 2u_{l+1,m,n} + u_{l,m,n+1} + u_{l,m-1,n-1} + u_{l,m+1,n-1} \\ &+ u_{l,m+1,n+1} - 2h^2 f_{l,m,n}) + O(h^4), \end{aligned}$$

$$\begin{array}{lclc} l = 2i & & l = 2i + 1 & \\ m = 2j + 1 & \text{or} & m = 2j & i, j, k = 1, 2, \ldots, N - 1, \\ n = 2k + 1 & & n = 2k & \end{array} \tag{8b}$$

$$\begin{aligned} u_{l,m,n} = \frac{1}{8 - 2h^2c} (&2u_{l,m-1,n} + 2u_{l,m+1,n} + u_{l-1,m,n-1} + u_{l+1,m,n-1} \\ &+ u_{l-1,m,n+1} + u_{l+1,m,n+1} - 2h^2 f_{l,m,n}) + O(h^4), \end{aligned}$$

$$\begin{array}{lclc} l = 2i + 1 & & l = 2i & \\ m = 2j & \text{or} & m = 2j + 1 & \\ n = 2k + 1 & & n = 2k & i, j, k = 1, 2, \ldots, N - 1, \end{array} \tag{8c}$$

or

$$\begin{aligned} u_{l,m,n} = \frac{1}{8 - 2h^2c} (&2u_{l,m,n-1} + 2u_{l,m,n+1} + u_{l-1,m-1,n} + u_{l+1,m-1,n} \\ &+ u_{l-1,m-1,n} + u_{l+1,m+1,n} - 2h^2 f_{l,m,n}) + O(h^4), \end{aligned}$$

$$\begin{array}{lclc} l = 2i + 1 & & l = 2i & \\ m = 2j + 1 & \text{or} & m = 2j & \\ n = 2k & & n = 2k + 1 & i, j, k = 1, 2, \ldots, N - 1. \end{array} \tag{8d}$$

4. Generalizations

A square mesh is not a necessity but a convenience when using the LIM. The difference equations are easily adopted for rectangular meshes. The refined mesh rectangles are similar to the original rectangular region, and the difference formulation of the equation is invariant as the mesh is refined. Nonrectangular regions such as the L-shaped region in Fig. 4 should be subdivided into rectangles and refined simultaneously.

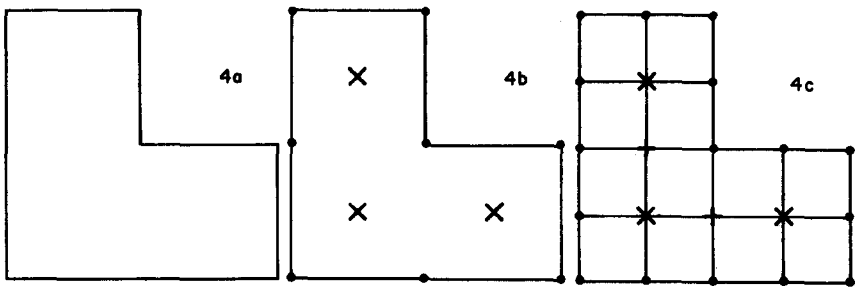


Fig. 4. (a) Dirichlet data is given on an L-shaped region. (b) An approximate solution is constructed using Eq. (5a). (c) The solution is refined using Eq. (5b). The mesh can be refined further by repeating this process.

Local refinement as illustrated in Fig. 5 presents no problems for the method. This is particularly useful in two and three dimensions where, after the equation has been solved, finer resolution is needed in a small portion of the solution. The LIM provides high accuracy and less work than spline interpolation methods and will display local behavior which traditional interpolatory methods may miss.

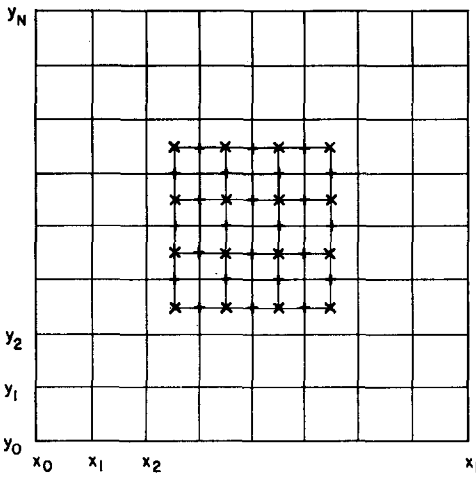


Fig. 5. The mesh is refined only in a neighborhood of the center. The refined solution will exhibit local behavior which standard interpolation methods may miss.

For Neumann and periodic boundary value problems the method can be successfully applied after an initial approximation is made using another method. In two and three dimensions, as the mesh is refined, boundary points must also be calculated. The boundary points are found by constructing fictitious points outside the region. In two dimensions the boundary points are $\times$ points, and in three dimensions they are centers of octrahedra.

5. Implementation and Numerical Results

A Fortran subroutine was written to expand a solution of

$$\begin{aligned} \Delta u + cu &= f && \text{in } D^M \\ u &= g && \text{on } \partial D^M \qquad M = 1, 2, 3. \end{aligned} \tag{9}$$

The programs were compiled and run using the CHAT compiler on a CDC 7600 at Los Alamos Scientific Laboratory. A major advantage of the LIM is the simplicity of the code. The subroutine to expand a solution to a finer mesh was only 28 Fortran statements in two dimensions and 73 statements for three-dimensional problems.

Figure 6 reflects the results of a series of two-dimensional tests on the unit square with $c = -1$ and the solution

$$u = \log[(x + 1/4)^2 + (y + 1/4)^2].$$

The equation was approximated with Buzbee's excellent Buneman elliptic *PDE* solver, the LIM and a combination of the two. The self-starting LIM approximated the initial center point by inverting the finite difference formula for the Laplacian from Collatz [3, p. 542],

$$u_{i,j} = \frac{((1/4) + (h^2c/52))(4\bar{u}_+ + \bar{u}_\times) - (h^2/52)(4\bar{f}_+ + \bar{f}_\times) - h^2 f_{i,j}}{5 - h^2c} + O(h^6) \tag{10}$$

$\bar{u}_+$, $\bar{u}_\times$, $\bar{f}_+$, and $\bar{f}_\times$ are defined as in Eqs. (6). Either Eqs. (5) or Eqs. (6) could have been used to approximate the center point, however, Eq. (10) was used since it has a significantly lower truncation error.

Once the solution is approximated on a 5×5 grid, the center point can be recomputed by its nearest neighbors using Eq. (10). The mesh refinement is then restarted on a 3×3 grid using the new value for the center point. As Fig. 6c shows this procedure can greatly improve the accuracy. A second iteration using the newly computed solution on a 5×5 grid to again approximate the center point reduced the error in Fig. 6c by one-half. Further itterations did not significantly improve the accurary in this example.

Figure 6 confirms that, when refining a very coarse mesh, accuracy is gained by using a higher-order LIM. Once the local truncation error of the LIM interpolation is less than the accuracy of the initial interior starting points there is no advantage in

	Mesh	9×9	17×17	65×65	129×129
			LIM		
a.	Average error	4.6×10^{-3}	4.0×10^{-3}	3.6×10^{-3}	3.5×10^{-3}
	Maximum error	3.2×10^{-2}	3.2×10^{-2}	3.2×10^{-2}	3.2×10^{-2}
	CPU time	0.0003	0.001	0.016	0.06
b.	Average error	7.5×10^{-4}	6.4×10^{-4}	5.8×10^{-4}	5.7×10^{-4}
	Maximum error	2.6×10^{-3}	2.6×10^{-3}	2.6×10^{-3}	2.6×10^{-3}
	CPU time	0.0007	0.003	0.08	0.18
c.	Average error	3.0×10^{-4}	2.6×10^{-4}	2.3×10^{-4}	2.3×10^{-4}
	Maximum error	9.4×10^{-4}	9.4×10^{-4}	9.4×10^{-4}	9.4×10^{-4}
	CPU time	0.0015	0.004	0.08	0.18
d.	Average error			2.4×10^{-4}	2.4×10^{-4}
	Maximum error			8.1×10^{-4}	8.1×10^{-4}
	CPU time			0.04	0.17
e.	Average error			2.4×10^{-4}	2.4×10^{-4}
	Maximum error			8.1×10^{-4}	8.1×10^{-4}
	CPU time			0.015	0.06
f.	Average error		2.3×10^{-9}	2.0×10^{-9}	1.8×10^{-9}
	Maximum error		1.6×10^{-7}	1.6×10^{-7}	1.6×10^{-7}
	CPU time		0.002	0.08	0.18
			Buneman		
g.	Average error	1.2×10^{-3}	2.7×10^{-4}	1.5×10^{-5}	3.8×10^{-6}
	Maximum error	3.1×10^{-3}	8.1×10^{-4}	5.1×10^{-5}	1.3×10^{-5}
	CPU time	0.002	0.01	0.20	0.89

FIG. 6. 7600 CPU times in seconds and relative errors in approximating the solution to Eq. (9). (a) Self-starting LIM using Eqs. (5) and Eq. (10). (b) Self-starting LIM using Eqs. (6) and Eq. (10). (c) The center point is recomputed with Eq. (10) from the approximation in (b) on a 5×5 grid. (d) Equations (6) refine the 17×17 Buneman solution. (e) Equations (5) refine the 17×17 Buneman solution. (f) Equations (6) refine the exact solution at 9 internal points. (g) Buneman solution using Buzbee's package.

using the higher order formulas. In Fig. 6f we see that if a few internal points are known accurately then the solution can be refined effectively.

Equation (9) was also solved for simple three-dimensional test problems using Eqs. (8). The 7600 CPU time to refine the solution to a $32 \times 32 \times 32$ mesh was 0.15 sec. No comparisons in accuracy have been made with other methods.

To approximate the solution to Eq. (9) at N points requires $O(N)$ operations and storage allocations. Therefore, these execution times scale linearly for larger problems.

ACKNOWLEDGMENT

I wish to express my appreciation for the encouragement and helpful comments of Dr. B. Buzbee, Prof. G. Golub, and Dr. B. Swartz.

References

1. A. Brandt, "MLAT I. The Multi-Grid Method," preprint IBM Thomas J. Watson Research Center, Yorktown Heights, New York.
2. B. L. Buzbee, G. H. Golub and C. W. Nielson, On direct methods for solving Poisson's equa*tions*, *SIAM J. Numer. Anal.* 7 (1970), 627–656.
3. L. Collatz, "The Numerical Treatment of Differential Equations," Appendix, Springer–Verlag, New York, 1966.
4. R. W. Hockney, A Fast Direct Solution of Poisson's Equation Using Fourier Analysis. *J. Assoc. Comput. Mach.* (1965), 95–113.
5. J. M. Ortega and N. C. Rheinbolt, "Iterative Solution of Nonlinear Equations in Several Variables," Academic Press, New York, 1970.
6. V. Pereyra, "Variable Order Variable Step Finite Difference Methods for Nonlinear Boundary Value Problems," Lecture Notes in Mathematics, No. 363, Springer–Verlag, New York, 1974.