INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 9, 599–607 (1989)

NUMERICAL PERTURBATION METHOD FOR APPROXIMATE SOLUTION OF POISSON'S EQUATION ON A MODERATELY DEFORMING GRID

JACK STRIGBERGER

Hal B. Wallis Research Facility of the Eisenhower Medical Center, 39000 Bob Hope Drive, Rancho Mirage, CA 92270, U.S.A.

SUMMARY

In problems such as the computation of incompressible flows with moving boundaries, it may be necessary to solve Poisson's equation on a large sequence of related grids. In this paper the LU decomposition of the matrix $\mathbf{A}_0$ representing Poisson's equation discretized on one grid is used to efficiently obtain an approximate solution on a perturbation of that grid. Instead of doing an LU decomposition of the new matrix **A**, the RHS is perturbed by a Taylor expansion of $\mathbf{A}^{-1}$ about $\mathbf{A}_0$. Each term in the resulting series requires one 'backsolve' using the original **LU**.

Tests using Laplace's equation on a square/rectangle deformation look promising; three and seven correction terms for deformations of 20% and 40% respectively yielded better than 1% accuracy.

As another test, Poisson's equation was solved in an ellipse (fully developed flow in a duct) of aspect ratio 2/3 by perturbing about a circle; one correction term yielded better than 1% accuracy.

Envisioned applications other than the computation of unsteady incompressible flow include: three-dimensional parabolic problems in tubes of varying cross-section, use of 'elimination' techniques other than LU decomposition, and the solution of PDEs other than Poisson's equation.

KEY WORDS Numerical solution of Poisson's equation Unsteady incompressible flow Moving boundaries Numerical solution of partial differential equations Numerical perturbation method Deforming grids

1. INTRODUCTION

An important component of most methods for the computation of unsteady incompressible flow is the efficient 'Poisson solver'. One such method which is good for problems with fixed grids is that of Dwyer.[1] This method uses an LU decomposition; the **L** and **U** matrices are computed before time marching begins and are used over and over again in the 'backsolve' at each time step. The purpose of the present paper is to introduce a simple and efficient systematic procedure capable of extending a method such as Dwyer's to problems with moderately deforming (e.g. time-varying) grids.

In the sections which follow, it will be assumed that we are using an LU decomposition to solve the system of simultaneous linear algebraic equations generated by finite differencing Poisson's or Laplace's equation on a large sequence of 'related' simply connected regions. Potential generalizations to other problems will be discussed in the final section.

0271–2091/89/050599–09$05.00
© 1989 by John Wiley & Sons, Ltd.

Received 24 May 1988
Revised 9 September 1988

2. DERIVATION OF THE METHOD

Suppose that we have computed and saved the matrices $\mathbf{L}_0$ and $\mathbf{U}_0$, the LU decomposition of the matrix $\mathbf{A}_0$ representing the finite differencing of Poisson's equation on some 'base' grid. We now wish to solve Poisson's equation on a large sequence of 'related' grids. By 'related' we mean:

1. All the grids have the same number and topology of grid points.
2. All the grids are generated from 'moderate' deformations of the base grid; what is meant by 'moderate' will become clearer from the examples in the next section.

The problem at hand can be written as

$$\mathbf{A}\mathbf{u} = \mathbf{f}. \tag{1}$$

It will now be shown that this problem is equivalent to

$$\mathbf{A}_0\mathbf{u} = \mathbf{f} + \sum_{i=1}^{\infty} \mathbf{f}_i', \tag{2}$$

where the evaluation of each $\mathbf{f}_i'$ requires one backsolve using the LU decomposition of $\mathbf{A}_0$.

First we rewrite (1) as

$$\mathbf{u} = \mathbf{A}^{-1}\mathbf{u}. \tag{3}$$

Next we perturb A about $\mathbf{A}_0$, i.e. we introduce

$$\mathbf{A}' = \mathbf{A} - \mathbf{A}_0, \tag{4}$$

where it is assumed that because the grid deformation is moderate, $\| \mathbf{A}' \mathbf{A}_0^{-1} \| \ll 1$. Therefore (3) becomes

$$\mathbf{u} = (\mathbf{A}_0 + \mathbf{A}')^{-1}\mathbf{f} = ((\mathbf{I} + \mathbf{A}'\mathbf{A}_0^{-1})\mathbf{A}_0)^{-1}\mathbf{f} = \mathbf{A}_0^{-1}(\mathbf{I} + \mathbf{A}'\mathbf{A}_0^{-1})^{-1}\mathbf{f}. \tag{5}$$

We next carry out a Taylor expansion of the term in parentheses, assuming that $\mathbf{A}'\mathbf{A}_0^{-1}$ is within the radius of convergence of the expansion; this yields

$$\mathbf{u} = \mathbf{A}_0^{-1}(\mathbf{I} - \mathbf{A}'\mathbf{A}_0^{-1} + (\mathbf{A}'\mathbf{A}_0^{-1})^2 - \dots)\mathbf{f}. \tag{6}$$

Finally, multiplying through by $\mathbf{A}_0$ and defining some new terms, we obtain

$$\mathbf{A}_0\mathbf{u} = \mathbf{f} + \mathbf{A}'(-\mathbf{v}_1 + \mathbf{v}_2 - \dots) = \mathbf{f} + \sum_{i=1}^{\infty} \mathbf{f}_i', \tag{7}$$

where

$$\mathbf{A}_0\mathbf{v}_1 = \mathbf{f} \tag{8a}$$

$$\vdots$$

$$\mathbf{A}_0\mathbf{v}_i = \mathbf{A}'\mathbf{v}_{i-1}, \quad i = 2, 3, \dots, \tag{8i}$$

and $\mathbf{f}_i' = (-1)^i \mathbf{A}'\mathbf{v}_i$.

It is thereby seen that one may use the LU decomposition of $\mathbf{A}_0$ in place of that of A, provided that one modifies the right-hand side of the equation by adding on a series of 'correction' terms, each of which requires for its evaluation one backsolve using the original LU decomposition.

Use of a truncated version of (7) to accurately and efficiently approximate (1) relies on the following conditions to hold:

1. The number of $\mathbf{v}_i$ required should be small, i.e. $\| \mathbf{A}'\mathbf{v}_i \| \ll 1$ at small i, to allow for early truncation of the infinite series in (7).

2. The work required to solve (8a) and all of the required (8*i*) is significantly less than that required to calculate the LU decomposition of **A**.

For the first condition to hold, the norm of $\mathbf{A}'\mathbf{A}_0^{-1}$ must be small. For the examples presented in the following section, it is assumed that this norm is roughly $O(\varepsilon)$, where ε is the parameter characterizing the grid deformation.

As for the second condition, the operation count for the LU factorization can be at least an order of magnitude higher than that for the backsolve; specific ratios for different grid sizes are presented below.

In the following section, two sample problems are used to test the validity of the method.

3. EXAMPLES

Solution of Laplace's equation in a square deformed into a rectangle

Consider the situation depicted in Figure 1. There, a unit square has been 'squashed' into a rectangle by lengthening the vertical sides to $1+\varepsilon$ and shortening the horizontal ones to $1-\varepsilon$.

The finite difference solution of Laplace's equation on the rectangle, subject to the Dirichlet boundary conditions $u = \sin \pi x \cosh \pi y$, is carried out as follows:

1. The square is overlaid with a uniform grid; the standard five-point finite difference stencil is applied, and each resulting equation is normalized so as to make the diagonal coefficient equal to 1, yielding the matrix $\mathbf{A}_0$.
2. The LU decomposition of $\mathbf{A}_0$ is computed and stored.
3. The rectangle is overlaid with a uniform grid for each direction; the standard five-point finite difference stencil is applied, and each resulting equation is normalized so as to make the diagonal coefficient equal to 1, yielding the matrix **A**.
4. The matrix $\mathbf{A}' = \mathbf{A} - \mathbf{A}_0$ is formed and stored.
5. (8) and (7) are solved to yield the approximate solution for the rectangle.

We note that steps 1 and 2 are independent of the degree of deformation, ε, and need be done only once; steps 3–5 must be repeated for each new rectangular grid (i.e. for each new ε).

Given the above 'algorithm' and the structure of (7) and (8), it is clear that in order to save storage in a computer code, **A** may be overwritten by **A**′, since **A** is used only to get **A**′. On the other hand, in general, it is necessary to store $\mathbf{A}_0$ and not overwrite it with $\mathbf{L}_0\mathbf{U}_0$, since $\mathbf{A}_0$ is needed for the computation of $\mathbf{A}' = \mathbf{A} - \mathbf{A}_0$ on each new grid. Of course, since $\mathbf{A}_0$ is very sparse (as is **A**′), the additional storage required is less than that already required by $\mathbf{L}_0\mathbf{U}_0$.

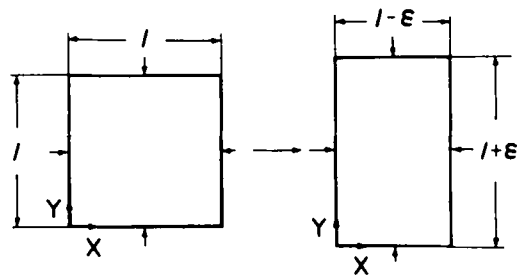

Figure 1. Square deformed into rectangle: geometry

The above procedure was tested for $\varepsilon = 0{\cdot}2$ with 5^2, 11^2, 21^2 and 41^2 grid points and for $\varepsilon = 0{\cdot}40$ with 5^2 and 41^2 grid points.

The computations were carried out on a CRAY-XMP/48, using the NAG Mark 11 sparse matrix routines[2] F01BRF or F02BSF and F04AXF for the LU decomposition and backsolve respectively.

Figure 2 shows the behaviour of the maximum percentage error, i.e. max{|[solution of (1) − solution of (7)]/solution of (1)|}, as a function of the number of terms retained in the series on the right-hand side of (7). It is seen that better than 1% accuracy is obtained after only three and seven 'corrections' for $\varepsilon = 0{\cdot}20$ and $0{\cdot}40$ respectively. Going to four and eleven corrections improves the accuracy to better than 0·10%.

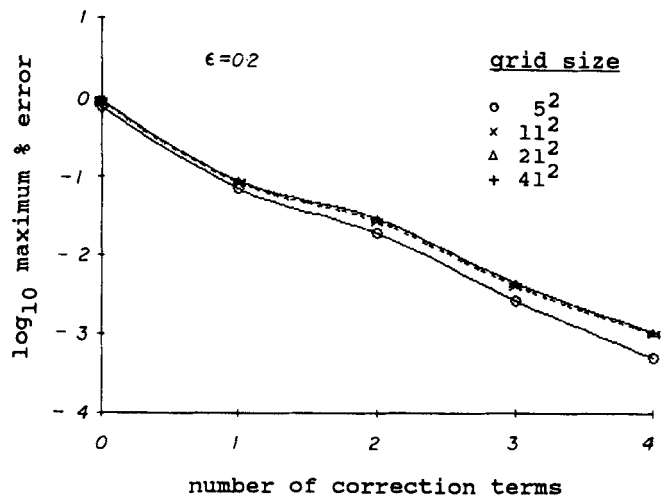

Figure 2(a). Convergence of rectangle/square problem with 20% deformation for different size grids

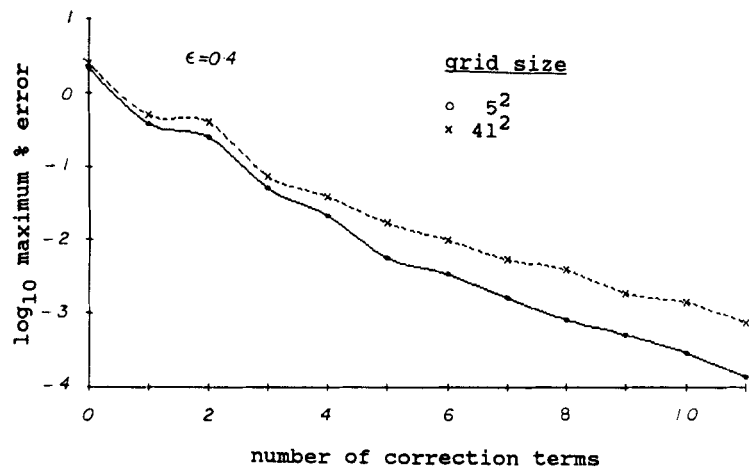

Figure 2(b). Convergence of rectangle/square problem with 40% deformation for different size grids

As an example of how the solution improves with an increased number of correction terms on right-hand side of (7), Figure 3 shows a plot of $u(x, y)$ at $x = 0{\cdot}06$ (where the error with no iterative correction was near maximum) from the 41^2 grid with $\varepsilon = 0{\cdot}40$ for selected numbers of correction terms.

What has been gained by using the perturbation scheme instead of the LU factorization of A? Figure 4 attempts to answer this question. On very coarse grids the answer is: very little.

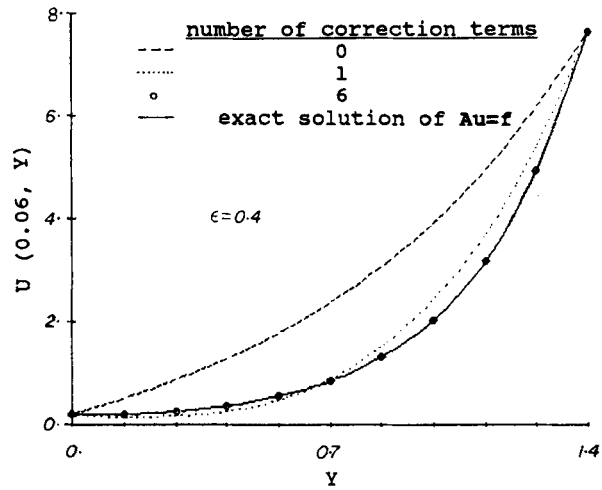

Figure 3. Rectangle/square problem with 40% deformation and 41^2 grid: solution as a function of y at fixed x (0·06) for different numbers of iterative corrections

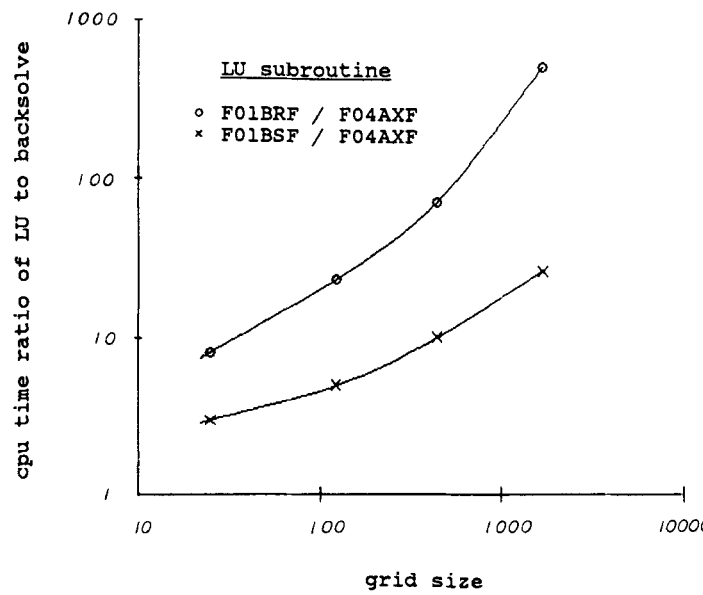

Figure 4. Ratio of CPU time for LU decomposition/backsolve as a function of grid size

However, on, say, the 41^2 grid it is seen that the LU decomposition requires between 26 and 500 times (using subroutines F01BSF and F01BRF respectively) as much CPU time as does each backsolve. It should be noted at this point that F01BRF does the LU decomposition completely from scratch, while F01BSF utilizes the pivoting 'strategy' already developed and saved by F01BRF. Of course, in order to use F01BSF, one must first use F01BRF. Furthermore, it turns out that the use of F01BSF is limited by the size of ε. For example, it was found that on the 41^2 grid, one could not use F01BSF for $\varepsilon > 0{\cdot}125$. So, if for example we consider the case of $\varepsilon = 0{\cdot}40$ on the 41^2 grid with 1% accuracy (which is often reasonable, as evident from Figure 3), we find that the perturbation scheme requires less than 2% of the CPU time required by the full LU decomposition scheme (F01BRF). For ε small enough to use F01BSF, if we use the perturbation scheme instead, two iterations suffice for 1% accuracy; this requires less than 15% of the computation time of F01BSF.

From a first glance at Figures 2(a) and 2(b), one may be concerned over the decreased convergence rate as the grid is made finer. However, from Figure 2(a) one also notes that the 11^2, 21^2 and 41^2 curves almost coincide, implying that an asymptote has been reached and indicating that the curve for, say, the 41^2 grid gives a good approximation for the convergence rate for arbitrarily fine grids (for ε fixed at 0·20; spot checks for $\varepsilon = 0{\cdot}40$ on the 11^2 and 21^2 grids compared with the results on the 41^2 grid indicate a very similar trend). It is also apparent from Figure 2 that the convergence rate is roughly linear in ε.

To obtain a somewhat different perspective on the convergence rate as a function of grid size and ε, the spectral radius ρ of $\mathbf{A}'\mathbf{A}_0^{-1}$ was examined for several coarse grids and values of ε. Since $\mathbf{A}'\mathbf{A}_0^{-1}$ is the effective numerical perturbation parameter, it is assumed that $\|\mathbf{A}'\mathbf{A}_0^{-1}\|$, and hence $\rho(\mathbf{A}'\mathbf{A}_0^{-1})$, should give a reasonable measure of the convergence rate. The results of this set of numerical experiments are shown in Figures 5 and 6. From Figure 5, where ρ is plotted against the order of $\mathbf{A}$ resulting from the interior grid points (the boundary points are ignored here because of the Dirichlet boundary conditions), we speculate that ρ does indeed reach a 'reasonable' limit as the number of grid points is increased. (To keep down the computational cost of this study, the finest grid considered was the 11^2 one; fortunately, the curves for ρ seems to level off by this range of grid fineness.)

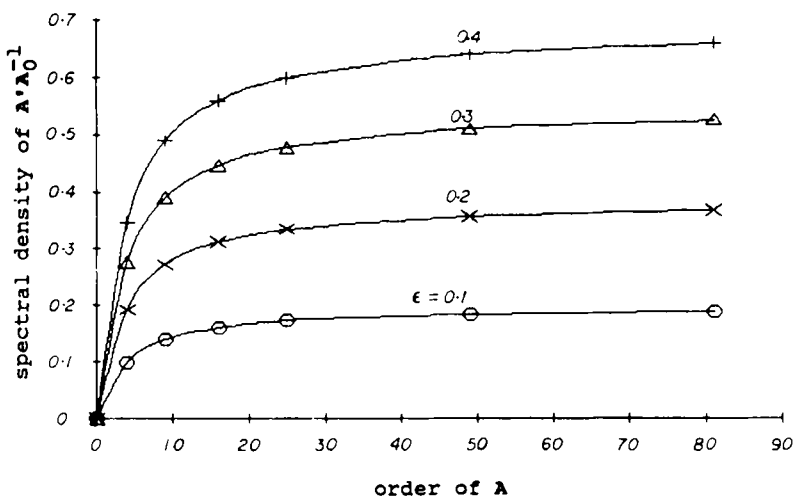

Figure 5. Spectral density of $\mathbf{A}'\mathbf{A}_0^{-1}$ as a function of the order of $\mathbf{A}$

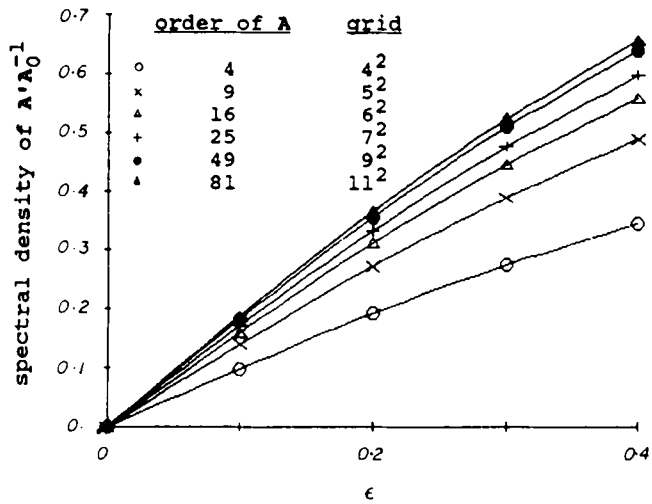

Figure 6. Spectral density of $A'A_0^{-1}$ as a function of grid deformation

Figure 6 implies that for a given grid, the spectral radius of $A'A_0^{-1}$ increases less than linearly as the grid deformation ε is increased. The slight discrepancy between this result and a comparison of Figures 2(a) and 2(b) for the 5^2 and 41^2 grids may be due to the incomplete correlation between spectral radius and convergence rate, especially as represented by the percentage error 'norm' used here.

Velocity distribution for flow in a circular duct deformed into an elliptical one

Fully developed laminar flow of a Newtonian fluid in a duct of hydraulic radius h is governed by

$$\nabla^2 u = -1, \tag{9}$$

where u is the axial velocity non-dimensionalized by $h^2(-dp/dx)/\mu$.[3]

We have considered the solution of (9) for an elliptical duct by perturbing about a unit circular one. The solution procedure is analogous to that of the rectangle/square case. The 21^2 radially stretched grids are shown in Figure 7; the semi-minor and semi-major axes of the ellipse are $1-\varepsilon$ and $1+\varepsilon$ respectively, with $\varepsilon = 0{\cdot}20$ in this case. The elliptical grid is non-orthogonal, requiring a nine-point stencil. For convenience, the computational domains were restricted to one quarter of the ellipse/circle, and Dirichlet boundary conditions were furnished by applying the exact solution at the computational boundaries (including the inner boundary, set at about $r = 0{\cdot}10$ for computational convenience). The percentage error reduction versus number of correction terms is given in Figure 8. In this case it is seen that 1% and 0·10% accuracy are achieved with only one and three corrections respectively.

It was found possible to raise ε to as high as 0·60 at least, though for this much deformation, ten correction terms were required for 1% accuracy.

The ellipse/circle problem with 20% deformation was also tested with Laplace's equation and boundary conditions $u = r\cos\theta$. Two and three corrections were required for 1% and 0·1% accuracy respectively, indicating a mild dependence of convergence on the exact form of the right-hand side of (1).

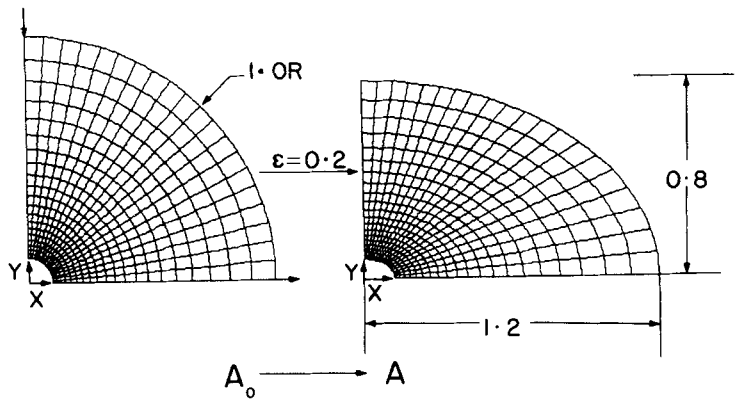

Figure 7. Geometry and grids for ellipse/circle problem with 20% deformation

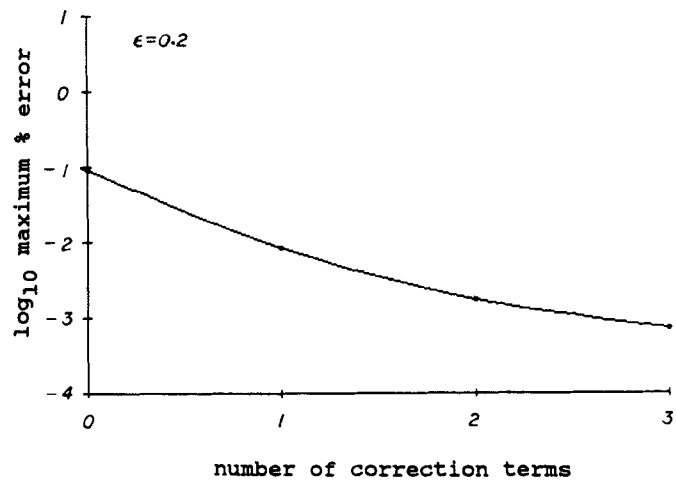

Figure 8. Convergence of ellipse/circle problem with 20% deformation

4. DISCUSSION, POTENTIAL APPLICATIONS AND CONCLUSIONS

It has been seen from both examples attempted that it is possible to efficiently compute the solution of Poisson's equation on one grid by using the LU decomposition of the matrix from a 'related' grid and by perturbing the right-hand side appropriately.

We now point out some potential applications of the method.

1. In the computation of unsteady incompressible flow with moving boundaries, the coefficients of $\mathbf{A}$ are functions of time; if $\| (\mathbf{A}(t) - \bar{\mathbf{A}})\bar{\mathbf{A}}^{-1} \|_{\max}$ is not too large (where $\bar{\mathbf{A}}$ is the matrix due to the 'mean' grid), then the perturbation method with a small number of correction terms may prove to be an efficient way of extending existing methods for problems with fixed grids.

2. Dwyer[1] mentions the use of the reapplication of a single LU decomposition to three-dimensional flows in tubes of constant cross-section; the perturbation approach may extend this method to problems where the cross-section varies.
3. While we have considered only LU decomposition, it may be possible to effectively use the perturbation approach with other Poisson solvers. For example, if one has to solve Poisson's equation on a grid which is moderately different from one on which a fast FFT solver could be used, then the problem could be broken down into a short perturbation series of such easy-to-solve FFT problems.
4. The method may be applicable to algorithms for partial differential equations other than Poisson's, e.g.Merkle's[4] and Guerra's[5] low-Mach-number Euler equation solvers. In fact, any problem requiring the repeated solution of a linear system of equations with 'moderate' variation in the elements of the system matrix may be a candidate for the present method. The present method may therefore prove useful in the solution of problems such as 'panel' methods in aerodynamics when the panels move a bit (e.g. flap deflection), and in iterative methods for mildly non-linear problems, where at each iteration the equations are linearized, yielding a matrix moderately different from iteration to iteration.

ACKNOWLEDGEMENTS

This work was supported by the Corwin D. Denney Foundation, with computer resources provided by the National Science Foundation and the San Diego Supercomputer Center; this support is hereby gratefully acknowledged. The author also thanks Deborah Sloss for preparation of this manuscript.

REFERENCES

1. H. Dwyer and S. Ibrani, 'Time accurate solutions of the incompressible and three-dimensional Navier–Stokes equations', *AIAA 26th Aerospace Sciences Meeting*, Reno, Nevada, 11–14 January 1988, *AIAA Paper 88-0418*.
2. *NAG Fortran Library Manual*, Numerical Algorithms Group, Oxford, U.K., 1984.
3. F. White, *Viscous Fluid Flow*, McGraw-Hill, 1974, p. 121.
4. C. L. Merkle and M. Athavale, 'Time-accurate unsteady incompressible flow algorithms based on artificial compressibility', *AIAA 7th CFD Conf.*, Honolulu, Hawaii, July 1987, *AIAA Paper 87-1137*.
5. J. Guerra and B. Gustafsson, 'A numerical method for incompressible and compressible flow problems with smooth solutions', *J. Comput. Phys.*, **63**, 377–397 (1986).