NOAA TM ERL WMPO-1

NOAA Technical Memorandum ERL WMPO-1

U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
Environmental Research Laboratories

A Numerical Experiment With Spiral Scan Successive Over-Relaxation

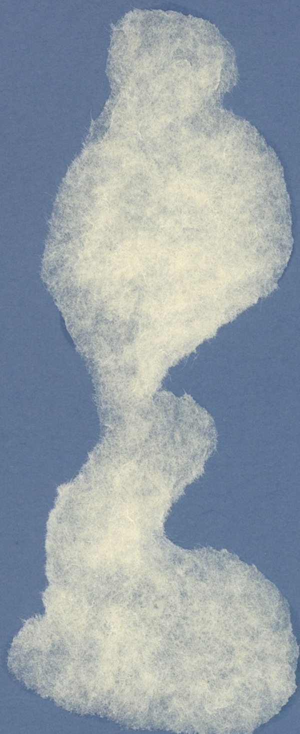
WALTER JAMES KOSS

Weather
Modification
Program Office
BOULDER,
COLORADO
April 1973

NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

NMENTAL RESEARCH LABORATORIES

ATHER MODIFICATION PROGRAM OFFICE



DEPARTMENT OF COMMERCE
UNITED STATES OF AMERICA

IMPORTANT NOTICE

Technical Memoranda are used to insure prompt dissemination of special studies which, though of interest to the scientific community, may not be ready for formal publication. Since these papers may later be published in a modified form to include more recent information or research results, abstracting, citing, or reproducing this paper in the open literature is not encouraged. Contact the author for additional information on the subject matter discussed in this Memorandum.

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
Environmental Research Laboratories

NOAA Technical Memorandum ERL WMPO-1

A NUMERICAL EXPERIMENT WITH SPIRAL SCAN SUCCESSIVE OVER-RELAXATION

Walter James Koss
National Hurricane Research Laboratory

Weather Modification Program Office
Boulder, Colorado
April 1973

TABLE OF CONTENTS

A NUMERICAL EXPERIMENT WITH SPIRAL SCAN SUCCESSIVE OVER-RELAXATION

Walter James Koss

A variant of successive over-relaxation (SOR), in which the iterant spirals in toward the center of the mesh during the numerical solution of the model problem, is compared with standard SOR for a square mesh; Haltiner (1971) states that the use of this variant in a meteorological forecast model yielded an increase in convergence rates. Results here indicate that the Spiral Scan method has a smaller convergence rate than SOR; also, there is an increase in execution real time, which is necessitated by additional machine coding when using the Spiral Scan iteration. Hence, it is concluded that the use of a Spiral Scan iterate is not economically feasible, regardless of the scale of the computer used when numerically solving the model problem.

1. INTRODUCTION

Various methods for obtaining the numerical solution of the Dirichlet problem on a rectangle are now available which, under special circumstances, offer rapid convergence rates and/or relatively economical computer execution times. Of these methods, the iterative SOR (successive over-relaxation) and ADI (alternating direction iterations) methods offer rapid convergence, with the ADI methods being somewhat "faster" than the SOR method (O'Brien, 1972). The ADI methods, though, offer two distinct disadvantages in their application: (1) one must determine iteration parameters which yield rapid convergence (Isaacson and Keller, 1966; Young, 1971), and (2) the difficulty of programming the method for a digital computer. The optimal acceleration parameters for SOR methods are well known (Young, 1971) as well as the convergence theory for these techniques.

Of the direct methods, the one given by Lindzen and Kuo (1969) requires extraordinary computer capability for relatively large matrix systems, and does not offer a substantial increase in operational speed (O'Brien, 1972), whereas the method of Hockney (1965) has proven to be extremely effective when used in conjunction with the Fast Fourier Transform (FFT). Hockney's method requires that the solution be periodic in at least one coordinate direction, a condition not often satisfied in closed region problems; hence, Hockney's method has limited applicability in this respect.

The standard point-by-point iteration SOR method is extremely simple to program for a digital computer when using a programming language such as FORTRAN. Variants of this method, wherein the mesh points are renumbered in the iteration procedure, are in a sense equivalent to the standard SOR method, and their convergence rates can be estimated by knowing the eigenvalues of the SOR coefficient matrix (Young, 1971). Programming a variant which might offer an accelerated convergence over SOR, and the concomitant computer execution times, could possibly make the use of such a variant economically unfeasible.

In this memorandum, we shall examine a variant of SOR in which the point-to-point iteration resembles a spiral over the mesh (fig. 1). At this time the question as to whether the coefficient matrix for the Spiral Scan iteration is consistently ordered (and therefore has Property A) as defined in Young (1971) will not be answered. The convergence rates and computer execution times for the Spiral Scan relaxation will be compared with standard SOR computations in order to determine the

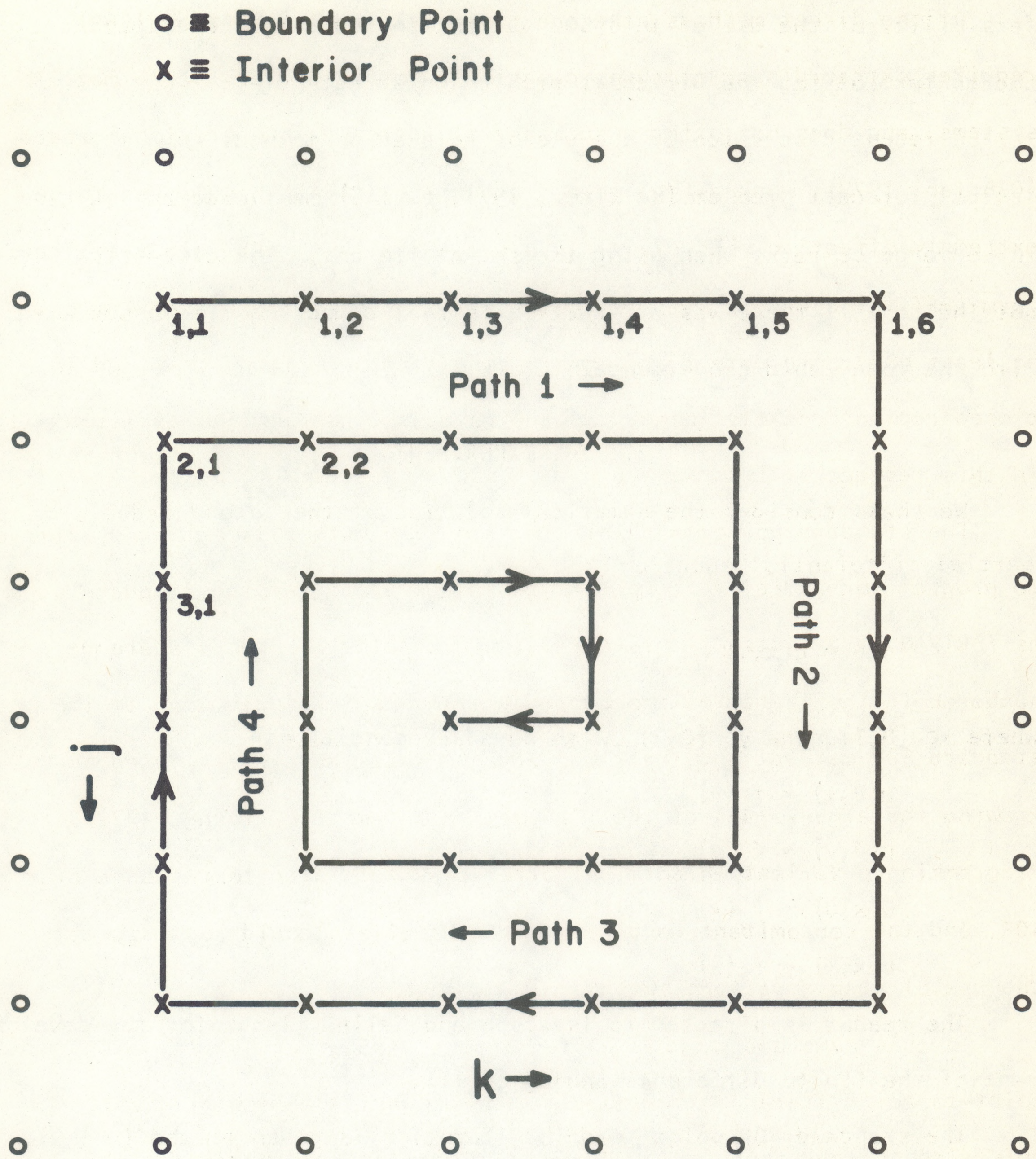


Figure 1. The Spiral Scan for a 6x6 interior mesh. The scan path is indicated by the line connecting the interior mesh points. j and k are the usual index parameters.

feasibility of the use of this SOR variant in reducing the computing time needed for solving the Dirichlet problem on a rectangle.

A recent discussion of the use of relaxation methods in a meteorological forecast problem (Haltiner, 1971, p. 115) mentioned an increase in convergence rates when using the spiral iterant. The discussion did not indicate if there was a reduction in real computing time associated with the more rapid convergence.

2. THE EXPERIMENT

We shall consider the numerical solution of the second order partial differential equation

$$\nabla^2 u = g(x,y) \qquad (1)$$

where $x \varepsilon [0,1]$, and $y \varepsilon [0,1]$, with boundary conditions

$$u(0,y) = f_1(y),$$
$$u(1,y) = f_2(y),$$
$$u(x,0) = f_3(x), \text{ and}$$
$$u(x,1) = f_4(x).$$

The reader is directed to Isaacson and Keller (1966) for the development of the finite difference analog of (1).

The standard SOR point-to-point iteration is given at the $(\nu + 1)$ iterate by

$$U_{j,k}^{(\nu+1)} = U_{j,k}^{(\nu)} + \omega(V_{j,k}^{(\nu+1)} - U_{j,k}^{(\nu)}) \qquad (2)$$

where

$$V_{j,k}^{(\nu+1)} = \theta_x(U_{j-1,k}^{(\nu+1)} + U_{j+1,k}^{(\nu)}) + \theta_y(U_{j,k-1}^{(\nu+1)} + U_{j,k+1}^{(\nu)}) + \delta^2 g_{j,k}, \tag{3}$$

$$\theta_x = \frac{\Delta y^2}{2(\Delta x^2+\Delta y^2)}, \quad \theta_y = \frac{\Delta x^2}{2(\Delta x^2+\Delta y^2)}, \text{ and } \delta^2 = \frac{\Delta x^2 \Delta y^2}{2(\Delta x^2+\Delta y^2)};$$

$1 \leq j \leq J$, $1 \leq k \leq K$; j, J, k, and K are positive integers; $\nu = 0,1,2,3,\ldots$, $\Delta x = 1/J$; and $\Delta y = 1/K$. The acceleration parameter is ω, and $S_{j,k}$ is the value of S at the interior mesh point $(j,k) = (x_j, y_k)$, where $x_j = j\Delta x$, and $y_k = k\Delta y$.

The $(\nu+1)$ iteration proceeds by first computing $U_{1,1}^{(\nu+1)}$ and then the other elements on the coordinate line $j = 1$ for $k = 2(1)K$. The same procedure is then done for $j = 2(1)J$.

For the Spiral Scan iteration, the expression for $V_{j,k}^{(\nu+1)}$ has four distinct variants depending on the path of the iteration. The paths, given schematically in figure 1, are categorized below:

path 1: j fixed, k increasing,

path 2: k fixed, j increasing,

path 3: j fixed, k decreasing, and

path 4: k fixed, j decreasing.

The variants of $V_{j,k}^{(\nu+1)}$ are given below:

path 1: $$V_{j,k}^{(\nu+1)} = \theta_x(U_{j-1,k}^{(\nu+1)} + U_{j+1,k}^{(\nu)}) + \theta_y(U_{j,k-1}^{(\nu+1)} + U_{j,k+1}^{(\nu)}) + \delta^2 g_{j,k}$$

$$\text{path 2:}\quad V_{j,k}^{(\nu+1)} = \theta_x\left(U_{j-1,k}^{(\nu+1)} + U_{j+1,k}^{(\nu)}\right) + \theta_y\left(U_{j,k-1}^{(\nu)} + U_{j,k+1}^{(\nu+1)}\right) + \delta^2 g_{j,k}$$

$$\text{path 3:}\quad V_{j,k}^{(\nu+1)} = \theta_x\left(U_{j-1,k}^{(\nu)} + U_{j+1,k}^{(\nu+1)}\right) + \theta_y\left(U_{j,k-1}^{(\nu)} + U_{j,k+1}^{(\nu+1)}\right) + \delta^2 g_{j,k}$$

$$\text{path 4:}\quad V_{j,k}^{(\nu+1)} = \theta_x\left(U_{j-1,k}^{(\nu)} + U_{j+1,k}^{(\nu+1)}\right) + \theta_y\left(U_{j,k-1}^{(\nu+1)} + U_{j,k+1}^{(\nu)}\right) + \delta^2 g_{j,k}.$$

The $(\nu+1)$ iterate proceeds point-to-point, as indicated in figure 1. Here $J = K = 20$ for all the numerical integrations. Two distinct problems are considered in the experiment.

Problem 1: $g(x_j, y_k) = x_j(x_j - 1) + y_k(y_k - 1)$ for $1 \leq j \leq J$, $1 \leq k \leq K$, and $f_1 = f_2 = f_3 = f_4 = 0$. The analytic solution is $u(x, y) = \frac{1}{2} xy(x - 1)(y - 1)$; this function has a maximum value at the center of the region of the integration. The initial (guess) field for each relaxation is $u(x, y) = 0$.

Problem 2: $g = f_1 = f_2 = f_3 = f_4 = 0$. The analytic solution is $u(x, y) = 0$. The initial (guess) field for each relaxation is a randomly generated net function given by $u(x_j, y_k) = 2Q - 1$, where Q is from a uniform distribution of random numbers such that $0 \leq Q \leq 1$. The guess field is the same for each relaxation.

. The acceleration parameter ω was stepped through the interval [1.61, 1.79] in steps of 0.01 to determine the optimum value (ω_{opt}) of the parameter on the basis of iteration count. The real-time estimate for each relaxation was obtained for each case in order to measure qualitatively the effect of the additional machine instructions needed to carry out the Spiral Scan iteration. Finally the computer programs were compiled under two different compilers[1] for Problem 1. One compiler, (FTN, Opt = 2) yields a machine program which is optimized compared with the other (RUN 23), and hence yields shorter execution times.

3. RESULTS

Figure 2 shows the results of the integrations for Problem 1. On the basis of iteration count, the Spiral Scan (a) gives a slight

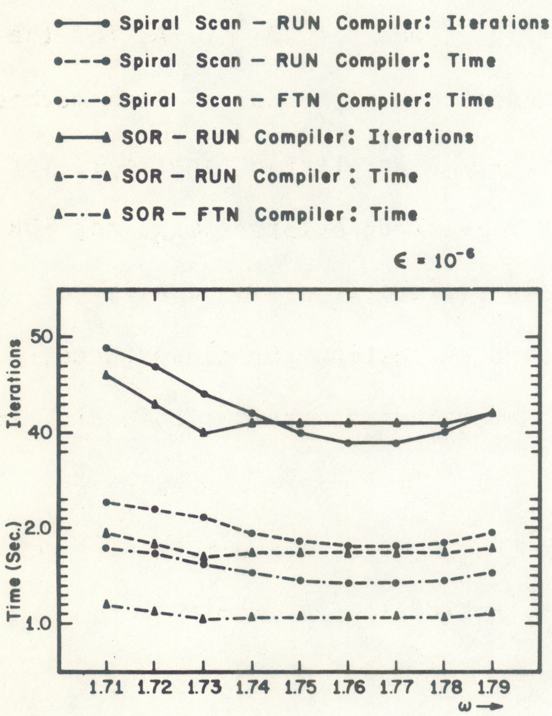


Figure 2. Iteration count and real execution time versus the acceleration parameter ω for Problem 1.

---

[1]All computations were performed on a CDC 6400 computer system.

reduction in the amount of computing needed, but, (b) on a real time basis, SOR performs the integrations in less time for each compiler mode. From (a) we could infer that the convergence rate is larger for the Spiral Scan, but ω_{opt} is larger than ω_{opt} for the SOR method, which may not be the case on a theoretical basis. For the SOR computations, $\omega_{opt} = 1.73$ agrees with the theoretical estimate (Young, 1971).

Figure 3 gives the results for Problem 2. All computations were made with computer coding generated by the optimizing compiler FTN. Figure 3 (top) is for a cut-off tolerance $\varepsilon = 10^{-6}$.[2] Clearly, in this case, SOR is superior in both having a larger convergence rate, and in taking less real time per iteration. For the Spiral Scan, $\omega_{opt} = 1.72$; whereas, ω_{opt} for SOR is 1.66. The latter value is suspect since the theoretical estimate for this case is 1.729. Figure 3 (bottom) shows the results from computations with $\varepsilon = 10^{-9}$, and $\varepsilon = 10^{-12}$. As for the cases with $\varepsilon = 10^{-6}$ the SOR converges faster than the Spiral Scan method in both iteration count and real time. It is interesting that ω_{opt} for the Spiral Scan remains near 1.73 as ε is varied, but that ω_{opt} for SOR approaches the theoretical value as ε is decreased. This behavior of ω_{opt} for SOR may be related to the method of testing for convergence; a percent change in $u_{j,k}$ criteria may have given a more reliable distribution of ω versus iteration count.

---

[2]The cut-off tolerance ε is defined by $|\text{Residual}| \leq \varepsilon$, where

$$u_{j,k}^{(\nu+1)} = u_{j,k}^{(\nu)} + \omega * \text{Residual}.$$

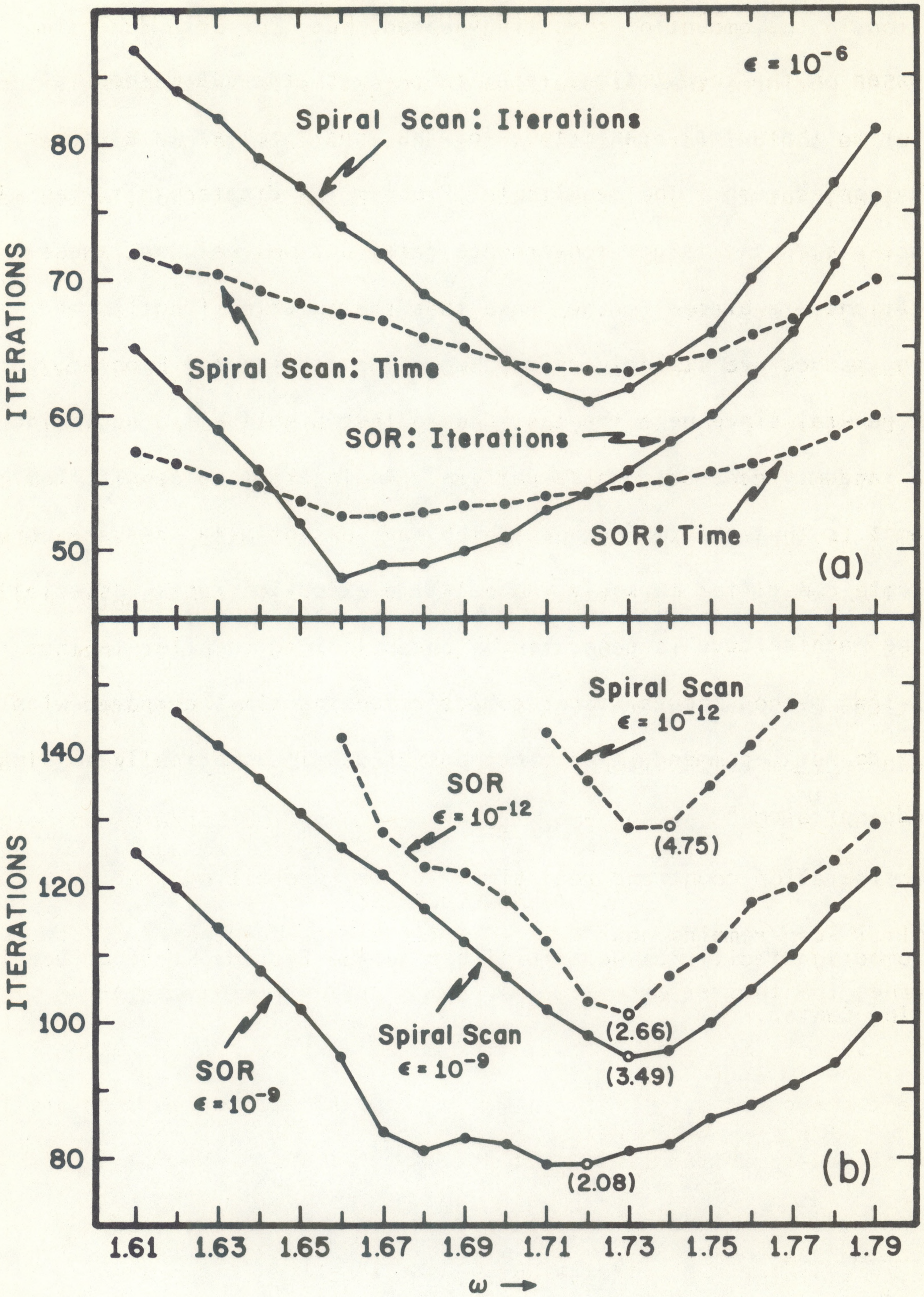


Figure 3. (a) Iteration count and real execution time versus the acceleration parameter ω for Problem 2. The relaxation cut-off tolerance (ε) is 10^{-6} in these computations. (b) Same as for (a) except that ε has the values 10^{-9} and 10^{-12}, and the real execution time (in brackets) is given for the minimum iteration count only.

4. CONCLUSION

Based on the computations reported on here, the SOR method is superior to the Spiral Scan method for the model problem in a square equally spaced mesh. The results for Problem 1 indicated that the Spiral Scan had a slightly larger convergence rate, but we feel that these computations are biased in the sense that the solution function and the scanning method are strongly correlated. The results for Problem 2 are rather general since here the test was against a null field beginning with a randomly generated noise pattern. An interesting result from Problem 1 is that the two methods which have essentially the same convergence rate can differ markedly in real-time execution rates, especially when the machine code is generated by an optimizing compiler. Since the Spiral Scan method incurs longer computer running times compared with SOR, we do <u>not</u> recommend it as a method for use in numerically solving the model problem.

5. ACKNOWLEDGEMENT

Computing facilities were furnished by the Florida State University Computing Center.

6. REFERENCES

Haltiner, G. J. (1971): Numerical Weather Prediction. New York, John Wiley, 316 pp.

Hockney, R. W. (1965): A fast direct solution of Poisson's equation using Fourier analysis. Journal of the Association for Computing Machinery, 12, No. 1, 95-113.

Isaacson, E. and H. B. Keller (1966): Analysis of Numerical Methods. New York, John Wiley, 541 pp.

Lindzen, R. and H. L. Kuo (1969): A reliable method for the numerical integration of a large class of ordinary and partial differential equations. Monthly Weather Review, 97, No. 10, 732-736.

O'Brien, J. J. (1972): Unpublished lecture notes. Department of Meteorology, Florida State University, Tallahassee, Florida, p. 36.

Young, D. (1971): Iterative Solution of Large Linear Systems. New York, Academic Press, 570 pp.

LIST OF FIGURES