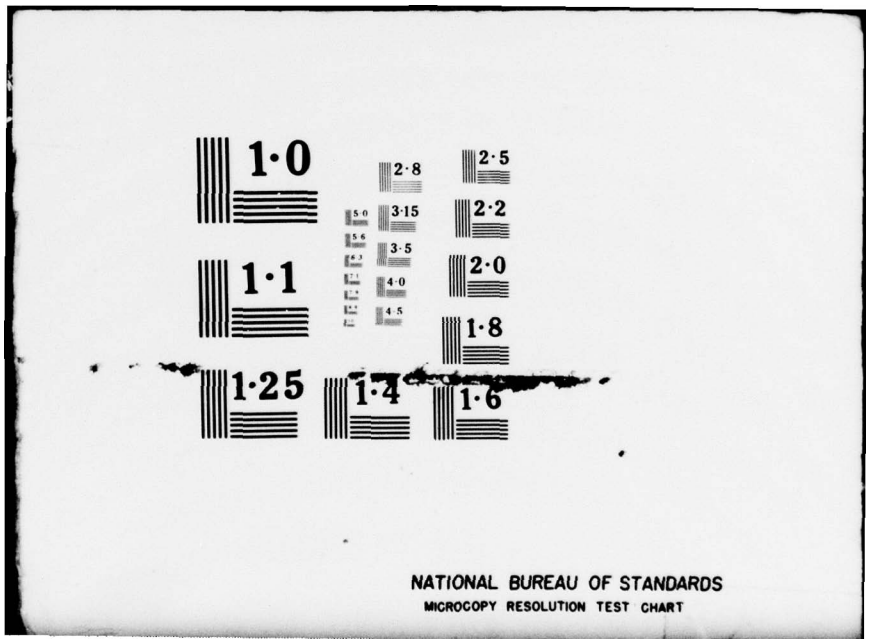
AD-A056 360 NAVAL RESEARCH LAB WASHINGTON D C F/G 12/1
AN EFFICIENT DIRECT SOLVER FOR SEPARABLE AND NON-SEPARABLE ELLI--ETC(U)
MAR 78 R V MADALA
UNCLASSIFIED NRL-MR-3668 SBIE-AD-E000 175 NL

1 OF 1
ADA
056360

END
DATE FILMED
8-78
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

NRL Memorandum Report 3668

An Efficient Direct Solver for Separable and Non-Separable Elliptic Equations

RANGARAO V. MADALA

Plasma Physics Division

LEVEL II

March 1978

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NRL Memorandum Report 3668
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER: NRL-MR-3668

4. TITLE (and Subtitle): AN EFFICIENT DIRECT SOLVER FOR SEPARABLE AND NON-SEPARABLE ELLIPTIC EQUATIONS
5. TYPE OF REPORT & PERIOD COVERED: Interim report on a Continuing NRL Problem
6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): Rangarao V. Madala
8. CONTRACT OR GRANT NUMBER(s):

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, D.C. — RR033-03
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NRL Problem No. 403-32; ONR Subtask Code RR0330341

11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Arlington, Virginia
12. REPORT DATE: March 1978
13. NUMBER OF PAGES: 18 (19 p.)

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

SBIE — AD-E000 175

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Elliptic Equations, Direct Solver, Marching Method

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The basic error vector propagation method (EVP) for the solution of elliptic equations is described. It's advantages and limitations are compared to other direct solvers and iterative methods. A new technique called "stabilized error vector propagation" (SEVP) is presented. This method has most of the advantages of the EVP algorithm and, in addition, it is stable for all grid sizes. By solving Poisson's equation with Dirichlet boundary conditions, SEVP is found to be 3 to 10 times faster than competative direct methods on a vector computer and requires an order of magnitude smaller computer memory. SEVP is at least 10 times faster → next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

cont.

than SOR. The efficiency of the SEVP method is found to increase for grids stretched in the marching direction while other methods tend to deteriorate under similar conditions.

CONTENTS

AN EFFICIENT DIRECT SOLVER FOR SEPARABLE AND NON-SEPARABLE ELLIPTIC EQUATIONS

1. Introduction

The numerical solution of elliptic partial differential equations is frequently required for atmospheric problems. These equations may be separable or non-separable and their solutions are subject to have a variety of boundary conditions. If we write the elliptic finite difference form of the differential equation as $A\,\mathbf{x} = \mathbf{F}$, where A is the coefficient matrix, $\mathbf{F}$, the forcing function and $\mathbf{x}$ the required solution, then A can be diagonally dominant, weakly dominant or even non-diagonally dominant. Even though a number of iterative and direct solvers are available in the literature for solving elliptic equations, no one method is efficient for all types of equations. The iterative methods such as SOR (successive over-relaxation), ADI (alternate direct implicit) and hybrid methods such as optimized block implicit relaxation (Dietrich, et al., 1975) are very efficient for diagonally dominant equations. But they converge very slowly or may even diverge for the weakly or non-diagonally dominant equations.

The direct solvers developed by Hockney (1965), Buneman (1969) and Rosmond and Faulkner (1976) are very powerful, but restricted to separable equations. For non-separable equations the direct solvers of Lindzen-Kuo (1969) and Crout (1941) are available but require large amounts of computer memory and thus are limited to small arrays.

The error vector propagation method (EVP) (Roche; 1971, 1977, Hirota et al.; 1970) is applicable to any type of elliptic equation and requires an order of magnitude less memory than Lindzen-Kuo. However, as shown by Meavaney and Leslie (1972), EVP is unstable for a large

*Manuscript submitted November 23, 1977.

number of grid points. In this paper we develop a stabilized error vector propagation method (SEVP) which is stable for any number of grid points and retains most advantages of EVP.

In Section 2 a review of EVP upon which SEVP is based is given. The SEVP is described in Section 3. A comparison of the computer time and memory requirements for several representative direct and iterative solvers is given in Section 4. A summary is given in Section 5.

2. Error Vector Propagation Method (EVP)

A general two dimensional elliptic equation in finite difference form can be written as

$$AX(i,j)\,x(i-1,j) + AY(i,j)\,x(i,j-1) + BB(i,j)\,x(i,j) + CX(i,j)\,x(i+1,j) + CY(i,j)\,x(i,j+1) = F(i,j) \tag{1}$$

where the coefficients AX, AY, BB, CX and CY may be functions of both i and j, and F is the forcing function. We now review the EVP procedure described by Roche (1971, 1977) for solving equation (1) over a rectangular region shown in Figure 1 using Diriclet boundary conditions. By rearranging the terms, equation (1) can be written as

$$x(i,j+1) = (F(i,j) - AX(i,j)\,x(i-1,j) - AY(i,j)\,x(i,j-1) - BB(i,j)\,x(i,j) - CX(i,j)\,x(i+1,j))\,(CY(i,j))^{-1} \tag{2}$$

It is clear from equation (2) that if we know the solution on any two consecutive rows then we can march in j to obtain the required solution. Since the solution is known only on the boundaries B_1, B_2, B_3 and B_4, the EVP method requires an initial guess of the solution on the row interior to B_1 to march forward in j. Based on this inital guess a particular solution, $x^P(i,j)$ is obtained which satisfies equation (2) and all the boundary conditions except along B_2. If we assume that the particular solution deviates from the real solution, $x(i,j)$, by $x^H(i,j)$, then

$$x(i,j) = x^P(i,j) + x^H(i,j) \tag{3}$$

substituting equation (3) into equation (2) and noting that the particular solution satisifies equation (2) and boundary conditions at B_1, B_3, B_4, we obtain the following homogeneous equation

$$x^H(i,j+1) = -(AX(i,j)\,x^H(i-1,j) + AY(i,j)\,x^H(i,j-1) + BB(i,j)\,x^H(i,j) + CX(i,j)\,x^H(i+1,j))\,(CY(i,j))^{-1} \tag{4}$$

with zero boundary conditions along B_1, B_3, B_4. Along B_2 we find from equation (3)

$$x^H(i,N) = x(i,N) - x^P(i,N) \tag{5}$$

We can relate the homogeneous solution on any two rows by using $(M-2)$ independent vectors, where M represents total number of grid points in i including boundary points. Since the solution along B_2 (given by equation (5)) is known, we will relate it to the solution on the row just interior to the boundary B_1. In other words, if the vectors ϵ_1 and ϵ_2 (Figure 1) represent the homogeneous solution on the second and last rows respectively, then we can find an influence matrix $R_{1,1}$ such that

$$\epsilon_2 = \epsilon_1 R_{1,1} \tag{6}$$

In order to obtain $R_{1,1}$, we start with the kth unit vector along the second row and march equation (4) in j, with zero boundary conditions along B_1, B_3. The kth row elements of the $R_{1,1}$ matrix are the elements of Eq. (4) along B_2 boundary. By varying the values of k from 2 to $(M-1)$ we complete $R_{1,1}$.

From equation (6) and the error vector (given by equation (5)) we compute the homogeneous solution on the second row. Using these values on the second row and zero boundary conditions on B_1, B_3 and B_4 we march equation (4) in j to obtain the homogeneous solution throughout the region. Then by superposition of the particular and homogeneous solutions we

obtain the complete solution. Due to round off errors in computing the inverse of the matrix $R_{1,1}$, this solution does not satisfy the boundary condition along B_2 exactly. The accuracy of the solution is improved by recomputing the homogeneous part of the solution a few times. Normally, about six iterations are required for 20 grid points in marching direction to obtain an error of 10^{-14} on a computer with 56 bit word precision. The number of iterations required increases as the number of grid points in the marching direction is increased up to about 25; beyond 25 the method becomes unstable.

The influence matrix, $R_{1,1}$, depends only upon the coefficients of the elliptic equation and therefore can be computed without knowledge of the forcing function and the boundary conditions. This step is called the preprocessor. If we want to solve equation (1) repeatedly with the same coefficients but with different forcing functions and boundary conditions, the influence matrix needs to be computed only once and stored in the memory for further use.

Although the EVP method is very efficient and requires a very small amount of computer memory, it is unstable for large number of grid points in the marching direction (Meavancey and Leslie, 1972). For computers with 24 bit precision, the limit is about 12 grid points, which may be increased by going to higher precision or by using the two directonal marching method described by Hirota et. al. (1970). Even with 56 bit precision and two way marching the limit is extended to only about 40 grid points.

3. Stabilized Error Vector Propagation Method (SEVP)

In the SEVP method, the integration region is divided into blocks, each of which is stable for the EVP method. Further, any two consecutive blocks have two rows in common. The division of the integration region into three blocks is shown in Figure 2. As with the EVP method, SEVP is divided into two steps: the preprocessor step and the solution step. In the preprocessor step, (2×NBLK-1) influence matrices are computed where NBLK is the total

number of blocks. NBLK of the influence matrices relate the values of the homogeneous solution on the second and last rows of each block, while the remaining (NBLK-1) matrices relate the homogeneous solution on the second rows of consecutive blocks.

a. Preprocessor

The first $(M-2)$ unit vectors on the second row of the first block are used to march in j direction, using equation (4) and zero boundary conditions on B_1, B_3 and B_4, to obtain influence matrices for the last two rows of the block. If ϵ_1, ϵ_2 and ϵ_3 (shown in Figure 2) represent the homogeneous solution on the second and last two rows of the first block, then

$$\epsilon_2 = \epsilon_1 R_{1,1} \tag{7}$$

$$\epsilon_3 = \epsilon_1 R_{1,2} \tag{8}$$

where $R_{1,1}$ and $R_{1,2}$ are the influence matrices for the last two rows of the block.

Combining (7) and (8) to eliminate ϵ_1 we obtain

$$\epsilon_3 = \epsilon_2 R_{1,1}^{-1} R_{1,2} = \epsilon_2 S_1 \tag{9}$$

The matrices $R_{1,1}$ and S_1 relate homogeneous solutions on the second and last rows of the block and on the last two rows of the block, respectively.

To obtain the influence matrices for the second block, we start with the unit vectors on the second row of the second block, and compute the corresponding values on the first row of the block from equation (9) to get the equivalent of a boundary condition for this block. The homogeneous solution on the first row corresponding to the kth unit vector on the second is the kth row of the matrix, S_1. Using $(M-2)$ unit vectors on the second row with corresponding vectors on the first row and zero boundary values on B_3 and B_4, we march equation (4) in j to obtain influence matrices for the last two rows of the second block. If the vectors ϵ_3, ϵ_4 and ϵ_5 represents a homogeneous solution on the second and last two rows of the block,

respectively, then

$$\epsilon_4 = \epsilon_3 R_{2,1} \tag{10}$$

and

$$\epsilon_5 = \epsilon_3 R_{2,2} \tag{11}$$

where $R_{2,1}$ and $R_{2,2}$ are the influence matrices for the last two rows of the block. Eliminating ϵ_3 from (10) and (11) we obtain

$$\epsilon_4 = \epsilon_5 S_2 \quad S_2 = R_{2,2}^{-1} R_{2,1} \tag{12}$$

Repeating the procedure described above for the third block (last block in Figure 2) we will obtain a matrix, $R_{3,2}$, relating homogeneous solution on the second and last rows of the block. If ϵ_5 and ϵ_6 represents homogeneous solution on these rows, then

$$\epsilon_6 = \epsilon_5 R_{3,2} \tag{13}$$

As in the EVP method, all the influence matrices depend only on the coefficients of equation (1).

b. Solution

We obtain the required solution by one forward and one backward sweep of the blocks. During the forward sweep, an approximate solution is obtained which satisfies the equation and the boundary conditions everywhere except on Boundary B_2. In the backward sweep, this solution is corrected to obtain the exact solution.

The forward sweep is started with the second row on the first block with an initial guess for the elements of that row and the given boundary values along the first row. This procedure is exactly the same as EVP. When the solution has been marched to the end of the first block, arbitrary values are assigned along the block boundary to obtain the correction vec-

tor, ϵ_3. The second sweep forward generates a homogeneous solution for the first block. Iterations are applied, if necessary, to reduce round off error.

For the second block we take the arbitrary solution assigned along the first block boundary (second row second block) and the solution along the first row of the second block and sweep forward again. An arbitrary solution is again imposed along the last row of the second block to obtain the correction vector ϵ_5 . On the second (or homogeneous) sweep of the second block we use the correction vectors ϵ_2 and ϵ_3 to form the homogeneous solution. ϵ_3 is computed from ϵ_5 by equation (11) and ϵ_2 is obtained from ϵ_3 using equation (9). The procedure continues until all blocks are swept out. For the last block, we use the boundary B_2 instead of a guess value to find ϵ_6.

The particular solution we have obtained on the forward sweep satisfies equation (1) and the given boundary conditions everywhere except on the block boundaries. More importantly, the last block contains the exact solution since the computation of the correction vector ϵ_6 is based upon the known boundary conditions along B_2.

The backward sweep now corrects the errors introduced in the particular solution by guessing the solutions along the block boundaries. Using equations (12) and (13) we calculate ϵ_4 and ϵ_5 for the homogeneous sweep of the last block and obtain the total solution for the last block which has two rows in common with the second block. To obtain the homogeneous solution for the second block we need to find the difference, ϵ_5', between the particular solution on the last row of the block and the exact solution. One method of obtaining ϵ_5' is to store the particular solution from the forward sweep and subtract it from the exact solution. However, it is much easier to recompute the particular solution by backing up three rows into the second block and repeating a small segment of the forward sweep. After ϵ_5' is obtained we sweep the second block to obtain the homogeneous solution which is then added to the partic-

ular solution. This procedure is repeated until we finish the blocks. Error reduction iterations are applied at this step also.

The stabilization of EVP by this method is achieved by the introduction of the artificial boundaries. Propagation of error is very severe in EVP, and when the error exceeds the accuracy of the computer it cannot be corrected by addition of a homogeneous solution. The introduction of artificial boundaries before the error becomes too large limits the error in each block. Since the influence matrices for small blocks have small error levels, the artificial solution generated by the introduction of these boundaries can be easily corrected to obtain an extremely accurate final solution.

4. Computer Time and Memory Requirements.

As a test problem for SEVP, Poisson's equation is solved over a square region with zero homogeneous boundary conditions. The test problem is also solved with the Lindzen-Kuo (Lindzen and Kuo, 1969), Crout (1943) and SOR methods. The former two are direct solvers. The relaxation by SOR is stopped when the normalized error (normalized with the forcing function) reaches 10^{-4}. A comparison of the computing time taken by these methods and SEVP is given in Figure 3. When the grid dimension exceeds 60 for Lindzen-Kuo (L-K) and 50 for Crout methods the computer memory is exhausted. It is clear from the figure that the SEVP method is more efficient than the other three methods. For $M = N = 40$, the SEVP method is about two times faster than L-K and Crout methods and 20 times faster than SOR. The computation times given in Figure 3 were obtained with double precision (56 bit precision) operations on a vector computer.

Figure 4 shows the computer time required by the preprocessor step and the solution step of SEVP on a vector computer. The ratio of computing time required by the preprocessor step to that required by the solution step is about 3 for $M = N = 10$ and increases with increase in

grid dimension. The L-K method can also be separated into a preprocessor step (computing α matrices) and a solution step. For the test problem single precision on 32 bit word computer is adequate for the L-K method.

Figure 5 gives a comparison of the computer time taken by the single precision Lindzen-Kuo and double precision SEVP methods to solve the test problem. The top two curves are the total computer time (preprocessor and solution steps) required while the lower two curves are for the solution steps only. It is clear from the figure that SEVP method is faster than Lindzen-Kuo method for both the preprocessor and solution steps.

Table 1 gives the auxiliary memory requirements for the three direct methods for a 32 bit word computer using double precision. It is clear from the table that the SEVP method's auxiliary memory requirement is an order of magnitude smaller than that required by the other two direct methods.

5. Selection of the Marching Direction and the Block Size

If $NB(k)$ represents the maximum number of points in the marching direction of the kth block, then

$$NB(k) \leqslant PA$$

where P is a constant which depends only upon the computer precision and A represents the minimum value of $CY(i,j)\ CX^{-1}(i,j)$ if we march in j and of $CX(i,j)\ CY^{-1}(i,j)$ if we march in i. It is clear from this equation that we can reduce the auxillary memory required by the SEVP method by choosing the marching direction in such a way that $A \geqslant 1$. It can also be shown that the method requires less computing time to solve the elliptic equation in the case when $A \geqslant 1$ than in cases $A < 1$. Therefore, the computing speed of the method increases with $|A|$. On the other hand, the Lindzen-Kuo and Crout methods deteriorate with the

increasing A and became unstable for $A \geqslant 100$.

6. Summary and Conclusions

A new method called SEVP was developed to solve elliptic equations which has most of the advantages of the EVP method and is stable for any number of grid points. In comparison with Lindzen-Kuo, Crout and SOR methods this method is faster and requires about an order of magnitude smaller computer memory than the Lindzen-Kuo and Crout methods.

The procedure described in Section 3 to solve elliptic equations with Dirichelet boundary conditions can be easily be extended to other boundary conditions such as Neumann, periodic or mixed. It can be also be used to solve elliptic equations over a region with irregular boundaries.

7. Acknowledgments

The author would like to thank Dr. Mark Schoeberl for reading the manuscript and making many useful suggestions. The author would especially like to thank Dr. David Dietrich for stimulating his interest in the stabilization of the EVP method. The author also would like to acknowledge the receipt of the computer programs for the Lindzen-Kuo and Crout methods from Dr. Mark Schoeberl and Dr. Niels Winsor, respectively.

This research was supported by the Office of the Naval Research and the Naval Environmental Prediction Research Facility Block Fund.

8. REFERENCES

Buneman, 0., 1969: A compact non-iterative Poisson solver. SVIPR Report No. 294, Stanford University, Stanford, California.

Crout, P.D., 1941: A Short Method for Evaluating Determinants and Solving Systems of Linear

Equations with Real or Complex Coefficients. Trans. AIEE, 60, 1235.

Dietrich, D., B. E.McDonald, and A. Warn-Varnas, 1975: Optimized block-implicit relaxation. J. Comp. Phy., 18, 421-439.

Hirota, I., T. Tokioka, and M. Nishiguchi, 1970: A direct solution of Poisson's equation by generalized sweep-out method. J. of the Meteo. Soc. of Japan, 48, 161-167.

Hockney, R.W., 1965: A fast direct solution of Poisson's equation using Fourier analysis. J. Assoc. of Comp. Machinery, 12, 95-113.

Lindzen, R.S., and H.-L. Kuo, 1969: A reliable method for the numerical integration of a large class of ordinary and partial differential equations. Mon. Weat. Rev., 97, 732-734.

McAvaney, B.J., and L. M. Leslie, 1972: Comments on "A Direct Solution of Poisson's Equation by Generalized Sweep-out Method". J. of the Meteorological Society of Japan, 50, 136-137.

Roache, P. J., 1971: A New Direct Methods for the Discretized Poisson Equation. Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, University of California, Berkeley, September 15-19, 1970, Springer-Verlag, New York, Berlin, Germany, 1971, 462pp.

Roche, P. J. 1977: Marching methods for elliptic problems: Part I. Submitted to Numerical Heat Transfer. pp. 48-53.

Rosmond, T. E., and F. D. Faulkner, 1976: Direct solution of elliptic equations by block cyclic reduction and Factorization. Mon. Weat. Rev., 104, 641-649.

Table 1 – A Comparison of the Auxiliary Memory Requirements (in thousands of words) for SEVP, Lindzen-Kuo and Crout Methods on 32 Bit Word Computer Using Double Precision.

GRID SIZE	CROUT	SEVP	LINZEN-KUO
10	4.4	0.2	2.8
20	33.2	2.4	17.6
30	111.6	5.4	57.6
40	262.4	16.0	134.4
50	510.0**	25.0	260.0
60	878.4*	50.4	446.4
70	1391.6*	88.2	705.6*
80	2073.6	115.1	1049.6*
90	2978.4*	178.2	1490.4*
100	4040.0*	220.0	2040.0*

*More than the total central memory available on computer at the Naval Research Laboratory.

**Normalized error is more than 10^{-4}.

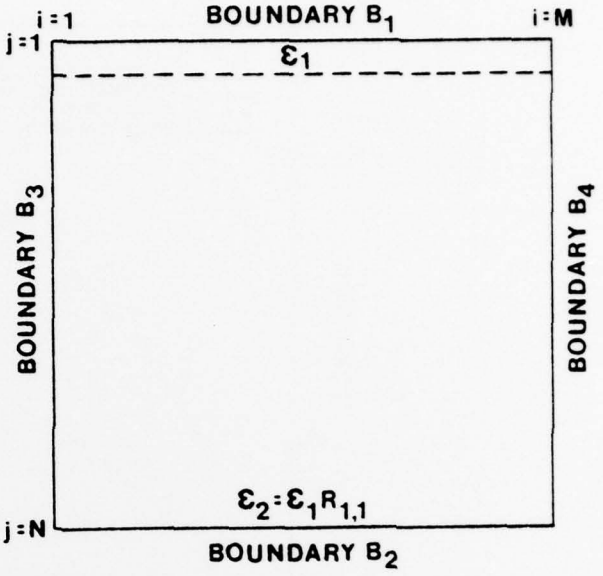


Fig. 1 – The region over which the elliptic equation is solved by EVP method B_1, B_2, B_3, and B_4 are the boundaries.

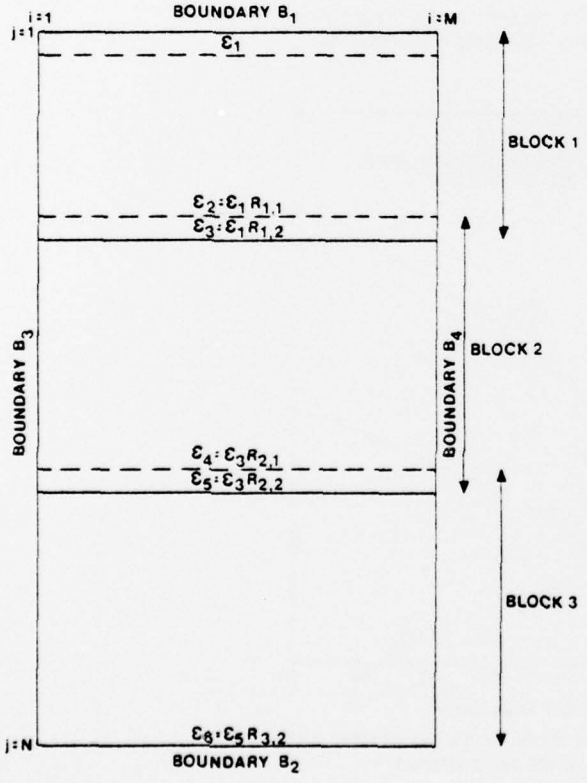


Fig. 2 – A three block division of the region for the SEVP method.

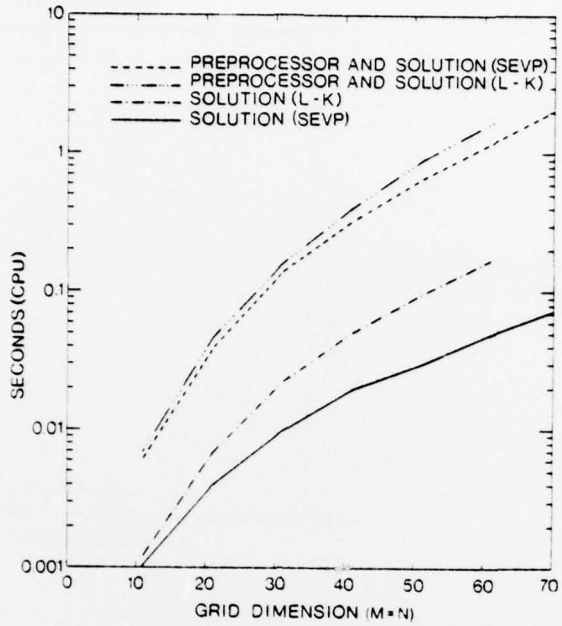


Fig. 3 – A comparison of the computing time requirements for the SEVP, Lindzen-Kuo, Crout and SOR methods.

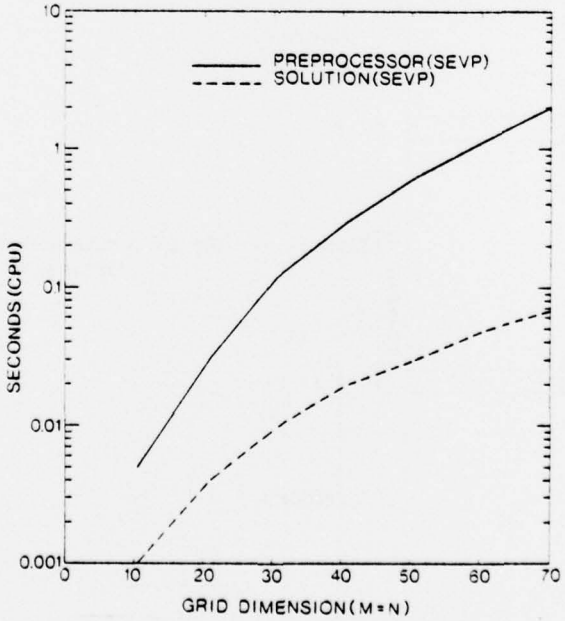


Fig. 4 – Computer time taken by the preprocessor and solution steps of the SEVP method.

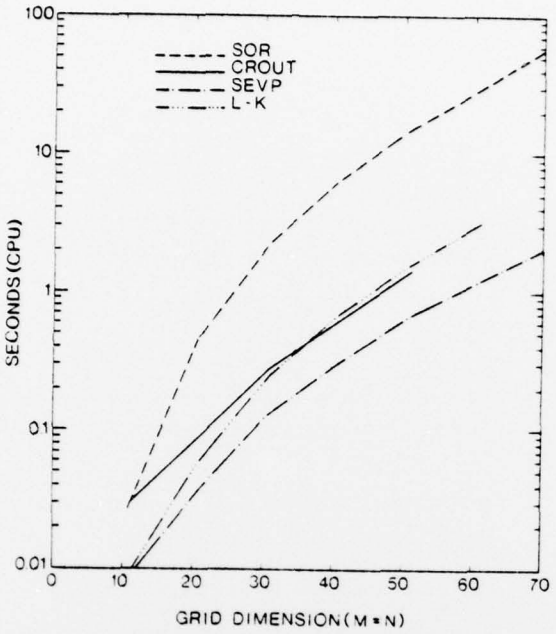


Fig. 5 – A comparison of the computer time requirements of single precision Lindzen-Kuo and double precision SEVP methods using a vector computer with 32 bit word length. The top curves give the total time taken by both steps: the preprocesser and the solution steps. The lower two curves give the time taken by the solution steps.