NUMERICAL ANALYSIS PROJECT
MANUSCRIPT NA-80-05

SEPTEMBER 1980

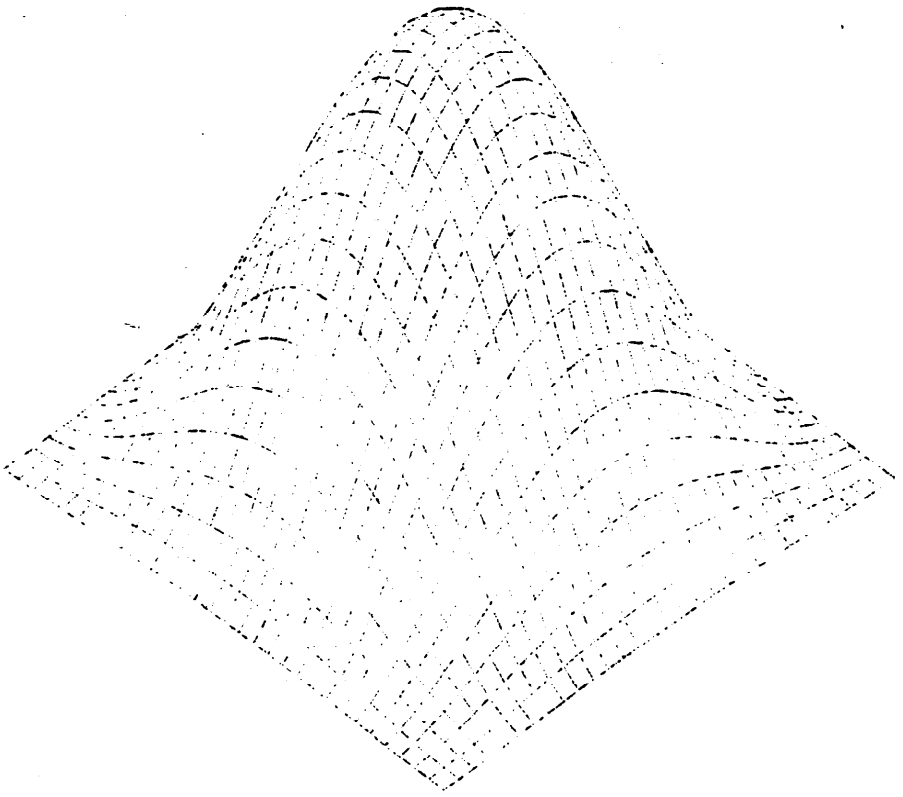
EFFICIENT SOLUTION OF THE BIHARMONIC EQUATION

BY

PETTER BJØRSTAD

NUMERICAL ANALYSIS PROJECT
COMPUTER SCIENCE DEPARTMENT
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305

EFFICIENT SOLUTION OF THE BIHARMONIC EQUATION



Petter Bjørstad[1]

Computer Science Department
Stanford University
Stanford, California

A new method for the numerical solution of the first biharmonic problem in a rectangular region is outlined. The theoretical complexity of the method is $N^2 + O(N)$ storage and $O(N^2)$ arithmetic operations. (In order to achieve a prescribed accuracy on an N by N grid.) Numerical results from a computer code that requires

[1]Supported in part by the Department of Energy Contract DE-AC03-76SF00326-PA #30.

$aN^2 + bN^2 \log N + O(N)$ operations with $b << a$, are presented using both a scalar and a vector computer. Extensions and some applications of the method for solving eigenvalue problems and certain nonlinear problems are mentioned.

I. INTRODUCTION

Consider the (first) Dirichlet problem for the biharmonic operator in a square S :

$$\begin{aligned} \Delta^2 u(x,y) &= f(x,y) && (x,y) \in S \\ u(x,y) &= g(x,y) && (x,y) \in \partial S \\ u_n(x,y) &= h(x,y) && (x,y) \in \partial S . \end{aligned} \tag{1}$$

Here u_n denotes the normal derivative of u with respect to the exterior normal. Let S be covered by a regular grid with N^2 interior gridpoints. The biharmonic operator is approximated using the 13-point stencil

$$\begin{array}{ccccc} & & 1 & & \\ & 2 & -8 & 2 & \\ 1 & -8 & 20 & -8 & 1 \\ & 2 & -8 & 2 & \\ & & 1 & & \end{array}$$

combined with quadratic extrapolation near the boundary. This scheme has truncation error $\|u - u_h\|$ of order h^2 where $h = 1/(N+1)$. The discrete problem can be written as

$$\frac{1}{h^4} A u_h = b$$

where A is defined by the discretization defined above and b is a function of the data f, g and h.

Many methods for the numerical solution of the above linear system have been proposed. They can roughly be classified as follows:

i) Iterative methods working on the matrix A.

ii) Direct methods working on the matrix A.

iii) Iterative methods based on reducing the biharmonic problem to a coupled system of two second order equations involving the Laplace operator.

iv) Direct methods taking advantage of the fact that A can be split into $L^2 + V$, where L is the discrete Laplace operator and V has low rank.

The first approach i) can be found in many early papers on the subject (6), (13). Its main disadvantage is related to the fact that A has condition number proportional to N^4 resulting in slow convergence of the iterative techniques.

Approach ii) is discussed in (2), (12) and (15). The theoretical complexity of a direct method is $O(N^3)$ arithmetic operations and $O(N^2 \log N)$ storage locations using for example nested dissection. This and other sparse matrix methods for the problem were studied and compared in (15). This study indicates that the constants in the above estimates are quite large and that a regular band solver is competitive even when the number of unknowns approach one thousand. A block elimination scheme is discussed in (2), while a band solver is being used in (12). Both these methods require a prohibitive amount of storage if N is large and they have a typical running time proportional to N^4, unacceptable for fine grid calculations.

The third and fourth approach is essentially two different ways of looking at the same underlying problem. A method based on iii) above was introduced by Smith in (16). It had a running time of $O(N^3)$. This was later improved to $O(N^{5/2})$ by Ehrlich (9) and Smith (17), (18). A drawback is the need to estimate iteration parameters.

The last approach iv) was pioneered by Golub (11) and a refined implementation is given by Buzbee and Dorr (5). This implementation, which is a direct method, requires $O(N^3)$ arithmetic operations. Despite being an $O(N^3)$ method it proved very competitive with the $O(N^{5/2})$ methods on realistic problems because those methods have an actual cost of $cN^{5/2}$ with c rather large.

Based on the above results it was concluded in (14) that the solution of the first biharmonic problem was an order of magnitude more difficult than the solution of Poisson's equation, even on parallel computers. The results of this paper show that the problems have the same complexity when a prescribed accuracy is desired.

II. OUTLINE OF THEORY

A brief description of the theory behind the new method will now be given. For a more detailed analysis with proofs and computer algorithms see (3) and (4).

Define the elementary matrices

$$R = \text{Tridiagonal } [-1,2,-1]_{N\times N}$$

and

$$T = \text{Diagonal } [1,0,\ldots 0,1]_{N\times N} .$$

The matrix A can be written

$$A = [(I \otimes R) + (R \otimes I)]^2 + 2(T \otimes I) + 2(I \otimes T)$$

where I is an N by N identity matrix and standard tensor product notation is used. Notice that the matrix

$$L = (I \otimes R) + (R \otimes I)$$

is the standard 5-point difference approximation to the Laplace operator.

Let

$$B = L^2 + 2(T \otimes I) .$$

Solving linear systems involving this matrix is no more difficult than solving Poisson's equations. Several $O(N^2)$ methods for this problem are known (1), (8) and such a method must be used in order to obtain an $O(N^2)$ method for the present problem. (Alternatively, an $O(N^2 \log N)$ method can be used, resulting in an overall operation count of $O(N^2 \log N)$.

The matrix $2(I \otimes T)$ can be written

$$2(I \otimes T) = 2\, U\, U^T$$

where U is an N^2 by $2N$ matrix.

Using this and the generalization of the Sherman-Morrison formula (7) gives

$$A^{-1} = B^{-1}(I - 2\, U\, C^{-1}\, U^T\, B^{-1})$$

where

$$C_{2N \times 2N} = I + 2\, U^T\, B^{-1}\, U .$$

It is clear that the original problem can be solved efficiently if there is a way to solve the linear system of equations involving the matrix C in no more than (N^2) operations.

Define a real normalized sine transform of a vector $x \in R^n$ by the relation

$$y = Q_N x ,$$

and note that

$$Q_N = Q_N^{-1} = Q_N^T .$$

Let S_k be block number k in the block diagonal matrix

$$R(I \otimes Q_N) B (I \otimes Q_N) R^T$$

where R is the permutation matrix defined by

$$R(D \otimes E)R^T = E \otimes D .$$

Similarly, define $\tilde{S}_k$ to be the corresponding block when B is replaced by L^2. Let $Px \equiv \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and $Py \equiv \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ where x_1 contains the odd numbered components of x, while x_2 contains the even numbered components.

The following theorems have important implications for the construction of efficient numerical methods.

Theorem 1.

Solving the linear system $Cx = y$ is equivalent to solving the two linear systems

$$T_1(x_1 + x_2) = y_1 + y_2$$

$$T_2(x_1 - x_2) = y_1 - y_2$$

where

$$T_1 = I + \frac{8}{N+1} \sum_{k=1,3\ldots}^{N} \sin^2 \frac{k\pi}{N+1} S_k^{-1}$$

$$T_2 = I + \frac{8}{N+1} \sum_{k=2,4\ldots}^{N} \sin^2 \frac{k\pi}{N+1} S_k^{-1}$$

Theorem 2.

Let T_i denote T_1 or T_2 and let $\tilde{T}_i$ be the corresponding matrix defined using $\tilde{S}_k$ instead of S_k. Then

$$P(Q\tilde{T}_i Q)^{-\frac{1}{2}} Q T_i Q (Q\tilde{T}_i Q)^{-\frac{1}{2}} P^T = I - \begin{bmatrix} (F^{i1})^T F^{i1} & 0 \\ 0 & (F^{i2})^T F^{i2} \end{bmatrix}$$

where all the singular values of F^{i1} and F^{i2} are in the interval

$$0 \leq \sigma < 0.8 .$$

Theorem 1 and 2 are proved in (3).

Since linear systems involving $\tilde{T}_i$ can be solved easily and since a matrix-vector product $T_i x$ can be formed at a cost of $O(N^2)$ arithmetic operations, it follows that the conjugate gradient method can be used to solve linear systems having coefficient matrix T_i. (Steepest descent would also work.) Moreover, the number of iterations required to achieve a prescribed accuracy is constant, independent of the grid size N.

III. COMPUTER IMPLEMENTATIONS

A large number of different computer implementations of the above ideas are possible. Which one is best will depend on the computer used and also on the underlying application. For a discussion of different alternatives see (3).

	N = 255		N = 511	
	SOLV	FFT	SOLV	FFT
IBM 370/168	7109	4324	-	-
CRAY-1 OFF=v	1804	882	-	-
CRAY-1 ON=v	251	548	878	2148

FIGURE 1. Time in milliseconds to solve the biharmonic equation on an NxN grid.

It is possible to design implementations that are well suited for parallel and vector computers. In order to illustrate this, a particular computer implementation was used on an IBM 370/168 and also on the CRAY-1 computer. Figure 1 displays some preliminary timing results.

Remarks:

i) The total solution time is the sum of the time spent in a fast Fourier transform routine (FFT) and in the remainder of the code (SOLV).

ii) The FORTRAN H(OPT=3) compiler was used on the 168 while the CFT compiler on the CRAY-1 was used with and without the vectorization option. (ON=v and OFF=v)

iii) The same FORTRAN code was used in all three cases with the single exception that a special vector innerproduct routine (written by Oscar Buneman) was used in the vectorized run.

iv) The iterative part of the algorithm was terminated when the 2-norm of the residual fell below 10^{-10}. (This typically takes less than 10 iterations.)

v) No attempt was made to optimize the Fourier transform part of the code. (The time spent in this part of the code can be substantially reduced, by doing the Fourier transforms in parallel.) The table should illustrate the practical behavior of the new algorithm; it does not represent a fully optimized implementation.

vi) It is estimated that the N=255 case would take at least 50 times longer using any of the earlier methods mentioned (aside from the difficulties of running previous algorithms in core for this grid size).

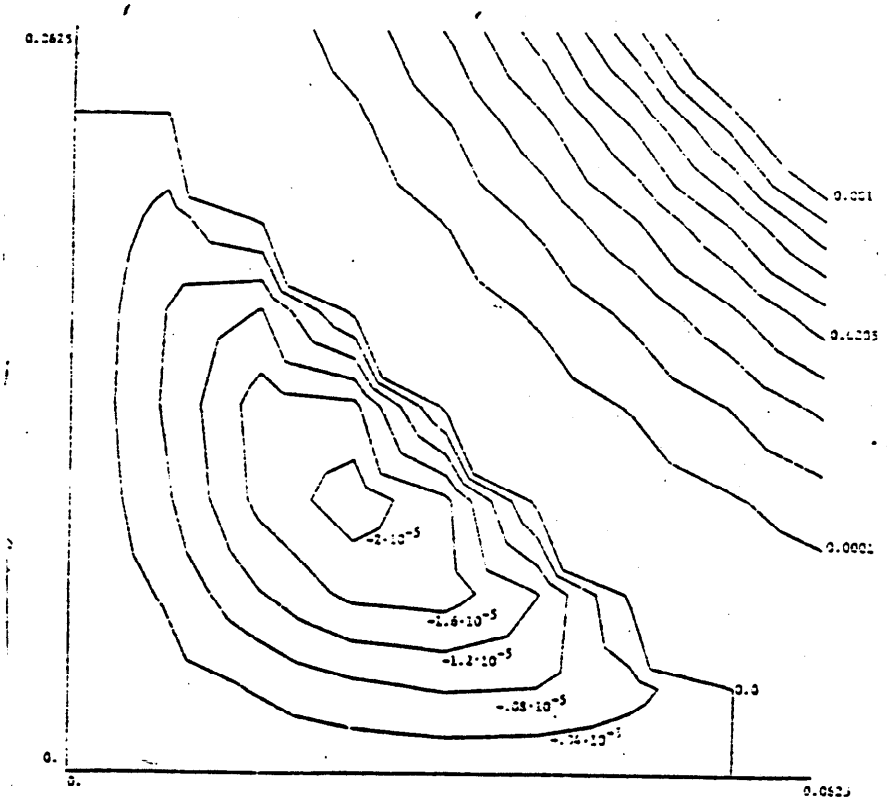
IV. EXTENSIONS AND APPLICATIONS

The handling of $N \times M$ grids $(N \neq M)$ is straightforward. It is also possible to solve more general fourth order problems having separable lower order terms in addition to the biharmonic operator. In particular, it is useful to be able to solve the following generalization:

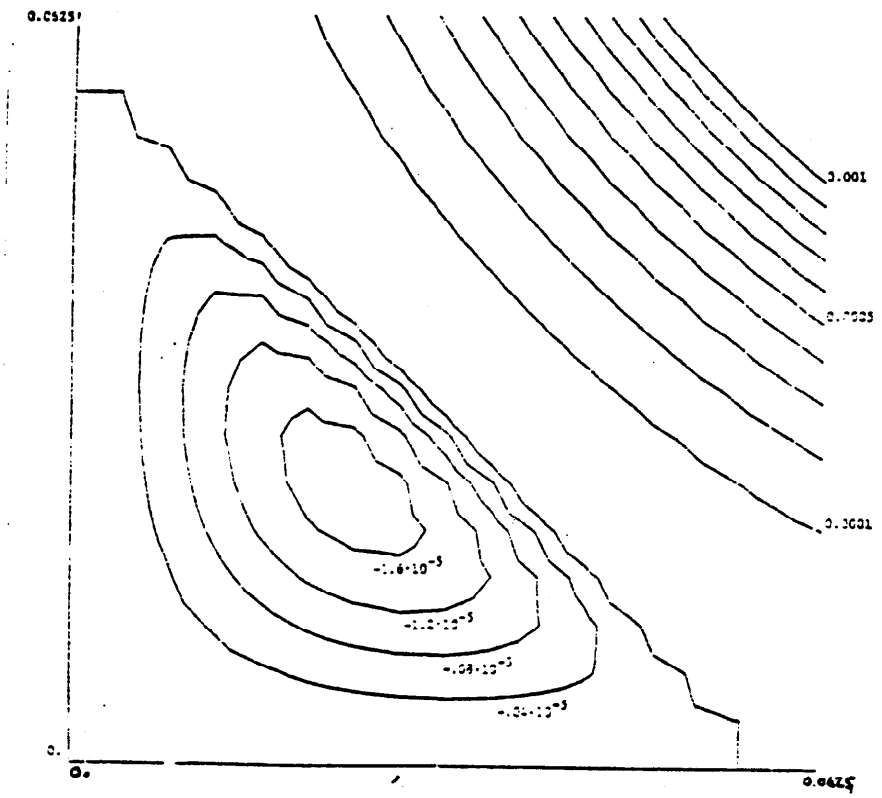
$$\begin{aligned} \Delta^2 u + \alpha \Delta u + \beta u &= f \qquad \text{in } R \\ u &= g \qquad \text{on } \partial R \\ u_n &= h \qquad \text{on } \partial R \end{aligned} \tag{2}$$

where α and β are two scalar parameters, and R is a rectangle. Formulation (2) was used (with $\alpha = 0$) in a Rayleigh iteration to compute a few of the eigenvalues of the biharmonic operator. Due to the cubic rate of convergence, this method is more satisfactory than the method reported in (2). Figure 2 shows the lowest eigenfunction near a corner. The entire plot is inside the 4 cells closest to the corner in the picture shown at the beginning of this paper. This reveals the existence of nodal lines near the corner and also gives a good description of the eigenfunction in this area. The presence of this phenomenon has been suspected, but due to the gridsize limitations of earlier algorithms, it has not been clearly demonstrated before. The theory outlined in Part II can be used to analyze this behavior also in the continuous case.

The existence of an efficient biharmonic solver makes it a useful computational tool in the construction of numerical methods for more complicated fourth order problems much in the same way as fast Poisson solvers have been used in the past ten years. As an illustration consider the driven cavity



N = 127



N = 255

FIGURE 2. Contour plots of the first biharmonic eigenfunction near a corner.

model problem for the nonlinear, time dependent Navier Stokes equation (19). Introducing a stream function Ψ in the usual way, the equation was solved using the following scheme:

$$\Delta^2 \Psi_{k+1} - \frac{R}{\Delta_t} \Delta \Psi_{k+1} = R(\Psi_y \Delta \Psi_x - \Psi_x \Delta \Psi_y)_k - \frac{R}{\Delta_t} \Delta \Psi_k$$

where k denotes the current time level. Notice that this again is a special case of (2) (this time with $\beta = 0$). This problem was solved in a square region with Reynold's number $R = 200$ and boundary conditions $\Psi = 0$ and $\Psi_n = 0$ except at the side $y = 1$ where

$$\Psi_n = \begin{cases} -\sin t & 0 < t < \pi/2 \\ -1 & t \geq \pi/2 \ . \end{cases}$$

This corresponds to an acceleration of the moving wall up to the standard velocity used in the stationary case. A 31×31 grid was used and 500 timesteps each of length 0.01 were used. (This is smaller than required for stability with this Reynold's number.) The run required approximately one minute on an IBM 370/168 and the velocity fields are shown in Figure 3a at two different time levels. It should be mentioned that the above scheme is unsatisfactory for large Reynold's number, due to the fact that the nonlinear term is fully explicit. The flow is not stationary at time 5, but changes very slowly into a final state after time equal 20 with a main vortex $\Psi = .105$ at coordinates (.41, .66) in agreement with stationary calculations.

ACKNOWLEDGMENTS

The author wishes to express his appreciation to Gene Golub for suggesting this problem and for contributing in many interesting discussions. Thanks are also due to Professor

Oscar Buneman of Stanford University for making the experiments on the CRAY-1 possible, and the Stanford Linear Accelerator for providing good computing facilities.

REFERENCES

1. Bank, R. E., and Rose, D. J. (1977). Marching algorithms for elliptic boundary value problems. I: The constant coefficient case. SIAM J. Numer. Anal. 14, pp. 792-829.
2. Bauer, L., and Reiss, E. L. (1972). Block five diagonal matrices and the fast numerical solution of the biharmonic equation. Math. Comp. 26, pp. 311-326.
3. Bjørstad, P. E. (1980). Numerical solution of the biharmonic equation, Ph.D. Thesis, Stanford University.
4. Bjørstad, P. E. Fast numerical solution of the biharmonic equation. To appear.
5. Buzbee, B. L., and Dorr, F. W. (1974). The discrete solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions. SIAM J. Numer. Anal. 11, pp. 753-763.
6. Conte, S. D., and Dames, R. T. (1960). On an alternating direction method for solving the plate problem with mixed boundary conditions. J. Assoc. Comput. Mach. 7, pp. 264-273.
7. Dahlquist, G., and Björck, Å. (1974). Numerical methods. Prentice-Hall, Inc., New Jersey.
8. Detyna, E. (1979). Point cyclic reductions for elliptic boundary value problems. I: The constant coefficient case. J. Comp. Phys. 33, pp. 204-216.

9. Ehrlich, L. W. (1971). Solving the biharmonic equation as coupled finite difference equations. SIAM J. Numer. Anal. 8, pp. 278-287.

10. Ehrlich, L. W. (1973). Solving the biharmonic equation in a square: A direct versus a semidirect method. Comm. ACM, 16, pp. 711-714.

11. Golub, G. H. An algorithm for the discrete biharmonic equation. Unpublished. See Appendix of (10).

12. Gupta, M. M., and Manohar, R. P. (1979). Direct solution of the biharmonic equation using non-coupled approach. J. Comp. Phys. 33, pp. 236-248.

13. Parter, S. V. (1959). On two line iterative methods for the Laplace and biharmonic difference equations. Num. Mat. 1, pp. 240-252.

14. Sameh, A. H., Chen, S. C., and Kuck, O. J. (1976). Parallel Poisson and biharmonic solvers. Computing 17, pp. 219-230.

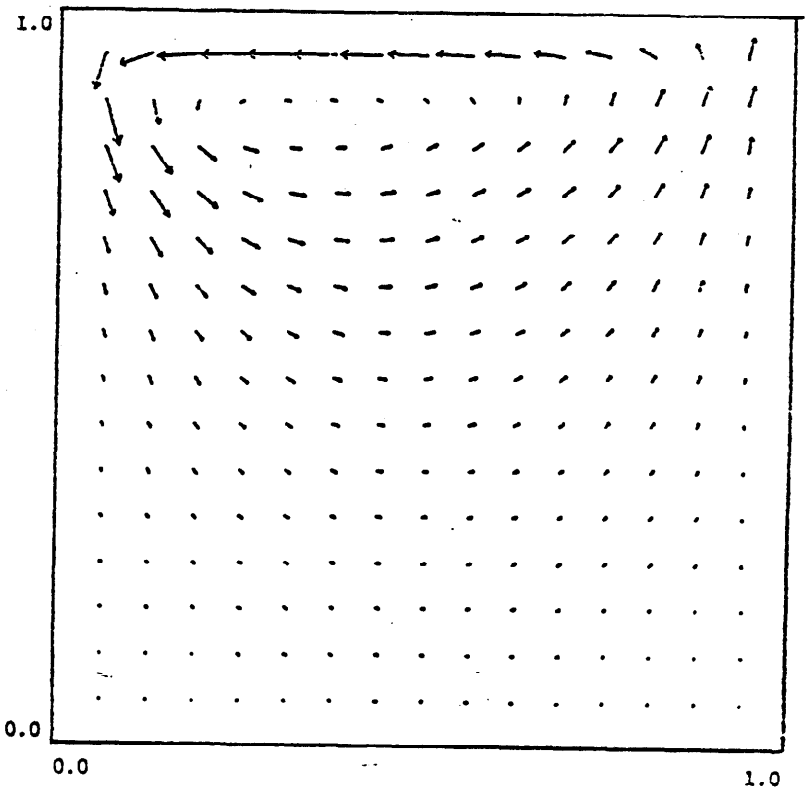
15. Sherman, A. H. (1975). On the efficient solution of sparse systems of linear and nonlinear equations. Ph.D. Thesis. Dept. of Comp. Sci., Yale Univ.

16. Smith, J. (1968). The coupled equation approach to the numerical solution of the biharmonic equation by finite differences, I. SIAM J. Numer. Anal. 5, pp. 323-339.

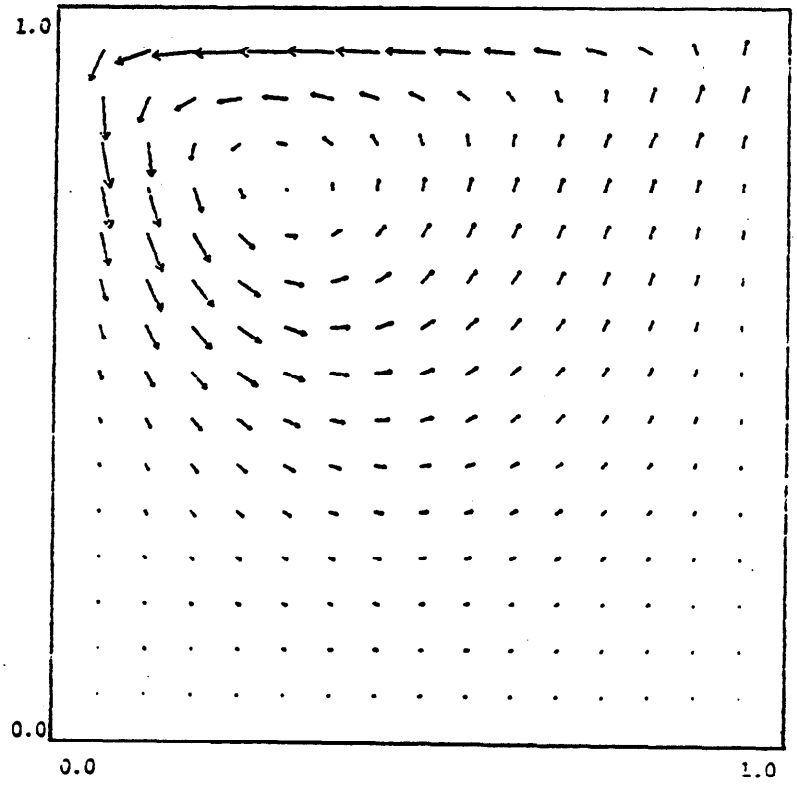
17. Smith, J. (1970). The coupled equation approach to the numerical solution of the biharmonic equation by finite differences, II. SIAM J. Numer. Anal. 7, pp. 104-111.

18. Smith, J. (1973). On the approximate solution of the first boundary value problem for $\nabla^4 u = f$. SIAM J. Numer. Anal. 10, pp. 967-982.

19. Teman, R. (1977). Navier Stokes Equations, North Holland, Amsterdam.

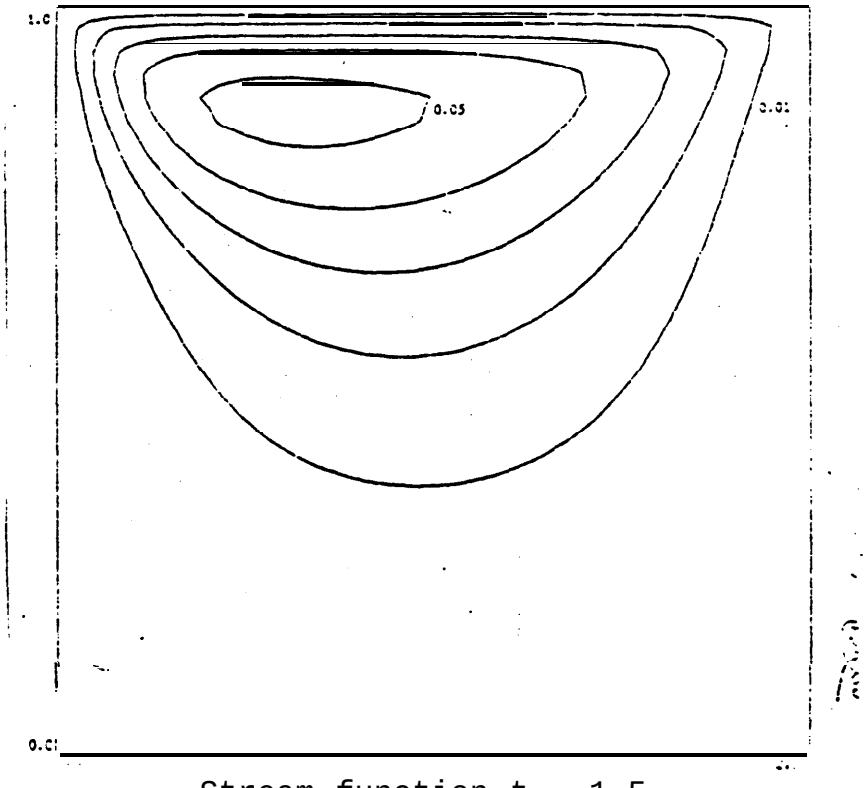


Velocity field t = 1.5

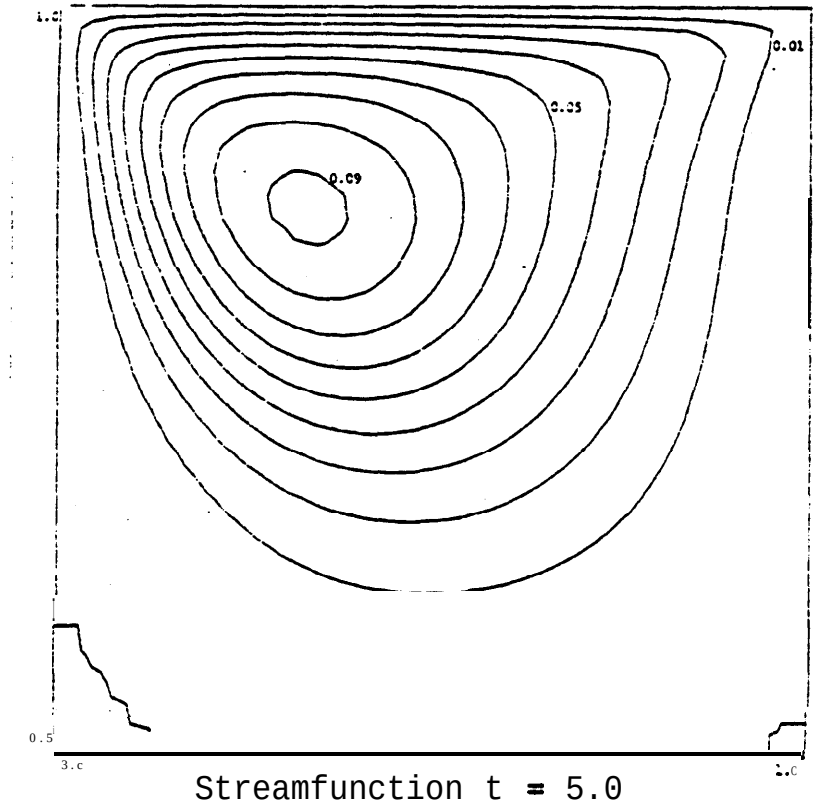


Velocity field t = 5.0

FIGURE 3a. Discrete solution of the time dependent Navier-Stokes equation at Reynold's number 200.



Stream function t = 1.5



Streamfunction t = 5.0

FIGURE 3b. Discrete solution of the time dependent Navier-Stokes equation at **Reynold's** number 200.