KONINKLIJK NEDERLANDS METEOROLOGISCH INSTITUUT

WETENSCHAPPELIJK RAPPORT
SCIENTIFIC REPORT

W.R. 79-1

S.J. Bijlsma and R.J. Hoogendoorn

Experiments with iterative methods for solving the discrete Poisson equation

De Bilt, 1979

Publikatienummer: K. N. M. I. W. R. 79-1 (MBW)

U. D. C.: 518. 5

Summary.

After the development of the well-known iterative methods for solving systems of linear equations resulting from the finite difference approximations to elliptic partial differential equations the last decade some new iterative methods have been proposed. In this report we apply the fastests of these methods to the relatively simple problem of solving the Poisson equation with Dirichlet boundary conditions on an octagon. Moreover we pay some attention to stop criteria for iterative procedures. Experiments are given comparing the number of iterations for different stop criteria and also the iteration time ratio with respect to the point SOR method.

Royal Netherlands Meteorological Institute,
P.O.Box 201,
3730 AE De Bilt/Holland.

1. Introduction.

The systematic development of the well-known iterative methods for the solution of systems of linear equations resulting from the finite difference approximations to partial differential equations has taken place in the fifties and early sixties. In a foregoing report (Bijlsma, 1977) we applied some of these iterative methods to the numerical solution of the Poisson equation with Dirichlet boundary conditions on an octagon. It appeared that of the methods considered the alternating-direction implicit (ADI) method and the two-line successive overrelaxation (2LSOR) method applied to the cyclically reduced equations along diagonal mesh lines converged most rapidly. In order to reduce the computing time of the 2LSOR method we introduced "extra" boundary points so that the method could be applied to the remaining points in a very simple way. It is true the resulting matrix problem didn't satisfy the theory entirely, so that we couldn't expect the minimal number of iterations (belonging to this method) for a certain error reduction factor, but we supposed that this might be compensated by the reduced computing time. In the present report we shall make further inquiries. For the sake of completeness we also consider the two-line cyclic Chebyshev semi-iterative (2LCC) method along the diagonal mesh lines.

During the last decade some new iterative methods have been proposed. We mention the method of Stone (1968) which is called the strongly implicit (SIP) method, and applications of the conjugate gradient method of Hestenes and Stiefel (1952) such as the method of Reid (1972) and the incomplete Cholesky conjugate gradient (ICCG) method of Meijerink and Van der Vorst (1977). In fact the SIP and ICCG methods are based on an approximate factorization of the coefficient matrix. Although the SIP method works quite well in rather difficult cases where the well-known iterative methods hardly converge, it is slower than, for instance, the ADI method when solving simple elliptic equations on regular regions (see Stone, 1968, p. 551). For applications of the ICCG method to complicated problems the reader is referred to Kershaw (1978).

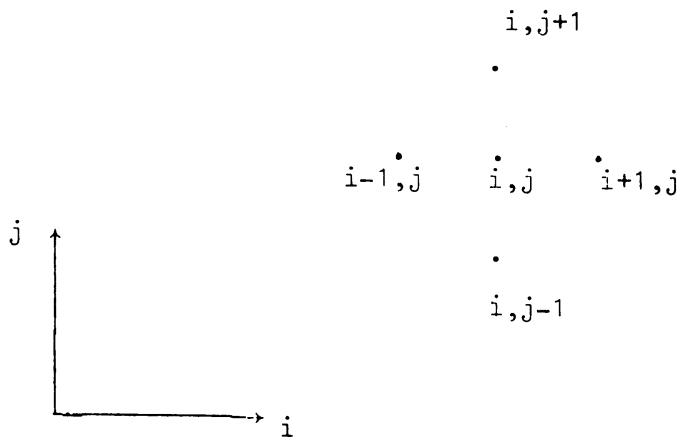
Results of this method when solving the Laplace equations on a square with mixed Dirichlet and Neumann boundary conditions are found in Meijerink and Van der Vorst (1977, p. 158). For such a problem the method of Reid (1972) appears to be two times faster than the point SOR method.

In this report we consider the ICCG method, the ADI method, two versions of the 2LSOR (2LCC) method and the point SOR method. These methods will be compared with respect to the minimal number of iterations for different stop criteria and computing time per iteration when solving the Poisson equation

$$-\nabla^2 U = F(x,y) \qquad (1.1)$$

defined on a region in the (x,y) plane bordered by a regular octagon (see Fig. 2, section 2). On the boundary we have U = 0. We now impose a uniform square mesh (x_i, y_j) with mesh side d on this region.

Using the notation $u(i,j) = U(x_i,y_j)$, $f(i,j) = F(x_i,y_j)$ application of the usual 5-point difference approximation to (1.1) at an interior point (i,j)



gives the difference equation

$$-u(i-1,j) - u(i+1,j) - u(i,j-1) - u(i,j+1) + 4u(i,j) = d^2 f(i,j). \qquad (1.2)$$

Labeling the n interior mesh points according to some ordering the system of difference equations (1.2) is equivalent to the system of linear equations

$$Au = k, \tag{1.3}$$

where the n components of the vector u are the values of the grid function u(i,j) at the ordered interior mesh points. The matrix A is a symmetric nxn matrix with at most 5 non-zero elements per row (see (1.2)). The vector k contains the inhomogeneous terms from (1.2) and moreover the boundary grid function values (which are zero in our case). We may write

$$A = D - E - F, \tag{1.4}$$

where D is a diagonal matrix, whose diagonal elements are equal to 4, and E and F are strictly lower and upper triangular matrices with non-zero elements equal to 1.

2. Survey of the methods considered.

In this section we shall give a short description of the methods which are applied to solve system (1.3).

(1) point SOR method. (Varga (1962), chapter 3 and 4.)

If the mesh points are ordered, for instance, by the natural ordering (i.e. a point (i,j) occurs before (i',j') if j<j' or if j=j' and i<i') then the point SOR method applied to (1.2) can be written as

$$\tilde{u}^{(m+1)}(i,j) = \frac{1}{4}\left(u^{(m+1)}(i-1,j) + u^{(m+1)}(i,j-1) + u^{(m)}(i,j+1) + u^{(m)}(i+1,j) + d^2 f(i,j)\right),$$

$$u^{(m+1)}(i,j) = u^{(m)}(i,j) + \omega\left(\tilde{u}^{(m+1)}(i,j) - u^{(m)}(i,j)\right), \quad m \geq 0. \tag{2.1}$$

Using (1.3) and (1.4), (2.1) is equivalent to

$$(D-\omega E)\, u^{(m+1)} = (\omega F + (1-\omega)D)\, u^{(m)} + \omega k$$

or

$$u^{(m+1)} = (I-\omega L)^{-1}\,(\omega U + (1-\omega)\, I)\, u^{(m)} + \omega D^{-1} k,$$

where $L = D^{-1}E$ and $U = D^{-1}F$. The value of ω which minimizes the spectral radius of the successive overrelaxation matrix $H(\omega) = (I-\omega L)^{-1}\,(\omega U + (1-\omega)I)$, $\rho(H(\omega))$ (i.e. the magnitude of the eigen values of $H(\omega)$ of largest magnitude) is given by

$$\omega = \omega_b = \frac{2}{1+\sqrt{1-\rho^2(B)}},$$

where $\rho(B)$ is the spectral radius of the point Jacobi matrix $B = D^{-1}(E+F) = L + U$. Moreover $\rho(H(\omega_b)) = \omega_b - 1$. Labeling the mesh points by σ_1 (or red-black) ordering (i.e. all points (i,j) with i+j even occur before those with i+j odd) the average rate of convergence (see section 3) is improved if we use $\omega=1$ for the first complete iteration (Sheldon, 1959). For numerical results, see Bijlsma (1977).

(2) 2LSOR (2LCC) method applied to the cycically reduced matrix equations along diagonal mesh lines.

Using (1.4) matrix equation (1.3) is equivalent to

$$u = D^{-1}(E+F)u + D^{-1}k = Bu + D^{-1}k. \tag{2.2}$$

If the mesh points are ordered by σ_1 ordering the point Jacobi matrix B has the form

$$B = \begin{bmatrix} 0 & F \\ F^T & 0 \end{bmatrix}, \tag{2.3}$$

where the null diagonal submatrices of B are square and F^T is the transpose of F. By a partition of the vectors u and $D^{-1}k = g$ relative to the partitioning (2.3), equation (2.2) gives the pair of equations

$$\begin{aligned} u_1 &= Fu_2 + g_1, \\ u_2 &= F^T u_1 + g_2, \end{aligned} \tag{2.4}$$

so that

$$\begin{aligned} u_1 &= FF^T u_1 + Fg_2 + g_1, \\ u_2 &= F^T F u_2 + F^T g_1 + g_2. \end{aligned} \tag{2.5}$$

Equations (2.4) are called the cyclic reduction of the matrix equation (2.2) (see Varga, 1962, section 5.4) It was Hageman's (1962) idea to apply block iterative methods to these cyclically reduced matrix equations.

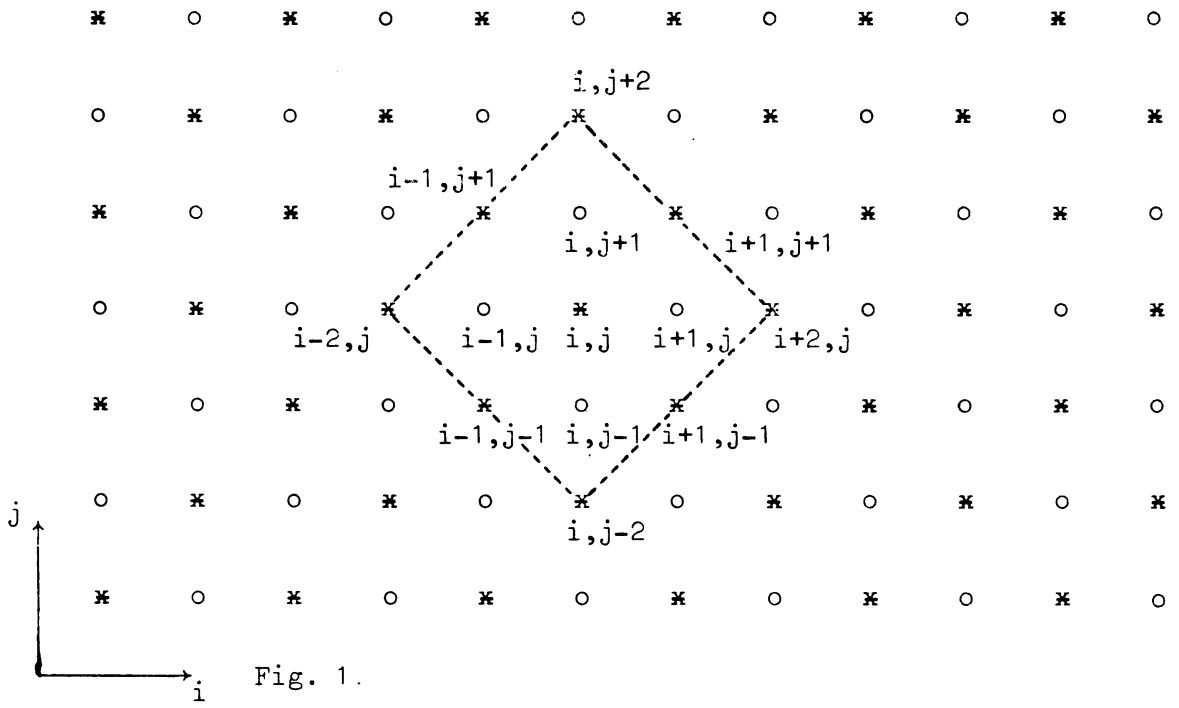


Fig. 1.

If u_1 is defined on the (*) mesh points from Fig. 1 equation (2.4) is equivalent to

$$
\begin{aligned}
u(i,j) &= \tfrac{1}{4}\,(u(i,j-1) + u(i-1,j) + u(i+1,j) + u(i,j+1) + d^2 f(i,j)),\\
u(i,j-1) &= \tfrac{1}{4}\,(u(i,j) + u(i,j-2) + u(i-1,j-1) + u(i+1,j-1) + d^2 f(i,j-1)),\\
u(i-1,j) &= \tfrac{1}{4}\,(u(i,j) + u(i-2,j) + u(i-1,j-1) + u(i-1,j+1) + d^2 f(i-1,j)),\\
u(i+1,j) &= \tfrac{1}{4}\,(u(i,j) + u(i+2,j) + u(i+1,j+1) + u(i+1,j-1) + d^2 f(i+1,j)),\\
u(i,j+1) &= \tfrac{1}{4}\,(u(i,j) + u(i,j+2) + u(i+1,j+1) + u(i-1,j+1) + d^2 f(i,j+1)),
\end{aligned}
\qquad (2.4)'
$$

and the first equation of (2.5) becomes

$$
\begin{aligned}
u(i,j) = \tfrac{1}{4}\,u(i,j) + \tfrac{1}{8}\,(&u(i+1,j+1) + u(i-1,j+1) + u(i-1,j-1) + u(i+1,j-1)) +\\
&\tfrac{1}{16}\,(u(i,j+2) + u(i-2,j) + u(i,j-2) + u(i+2,j)) + d^2 \hat{f}(i,j),
\end{aligned}
\qquad (2.5)'
$$

where

$$
\hat{f}(i,j) = \tfrac{1}{4}\,f(i,j) + \tfrac{1}{16}\,(f(i,j-1) + f(i-1,j) + f(i+1,j) + f(i,j+1)).
$$

Scheme (2.5)' corresponds to a 9-point approximation (on the (*) mesh points) at the point (i,j), which is surrounded by 4 interior (o) mesh points. Is one of the (o) points, say (i-1,j), a boundary point then the contribution of u(i-1,j) in (2.4)' will be cancelled and scheme (2.5)' will be changed. Let the points of the diagonal (*) mesh lines be labeled by diagonals (i.e. a point (i,j) occurs before (i',j') if $i+j < i'+j'$). We suppose that the number of mesh lines is even. Let $u_{1,i}$ be the vector defined on the i-th block of two successive diagonal mesh lines then $u_1 = (u_{1,1}, \ldots, u_{1,s})^T$ if there are s blocks. If the number of mesh lines is odd, we can introduce a block consisting of one mesh line. The matrix $C = I - FF^T$ from (2.5) corresponding with this partition can be written as

$$C = \begin{bmatrix} C_{1,1} & C_{1,2} & & & O \\ C_{2,1} & C_{2,2} & C_{2,3} & & \\ & \ddots & \ddots & \ddots & \\ & & & & C_{s-1,s} \\ O & & & C_{s,s-1} & C_{s,s} \end{bmatrix} .$$

Matrix C is a tridiagonal block matrix because the i-th block has only coupling coefficients with blocks (i-1) and (i+1). Let n_i be the number of mesh points of the i-th block then the $n_i \times n_i$ diagonal submatrices $C_{i,i}$ (i = 1, ... , s) contain the mutual coupling coefficients of the mesh points of this block. The matrix equation for u_1 from (2.5) becomes

$$\begin{bmatrix} C_{1,1} & C_{1,2} & & & O \\ C_{2,1} & C_{2,2} & C_{2,3} & & \\ & \ddots & \ddots & \ddots & \\ & & & & C_{s-1,s} \\ O & & & C_{s,s-1} & C_{s,s} \end{bmatrix} \begin{bmatrix} u_{1,1} \\ \vdots \\ u_{1,s} \end{bmatrix} = \begin{bmatrix} \hat{f}_{1,1} \\ \vdots \\ \hat{f}_{1,s} \end{bmatrix} , \qquad (2.6)$$

where $\hat{f}_1 = g_1 + Fg_2$. Because matrix C satisfies property A^{π} and is consistently ordered (Varga, 1962, chapter 4) the block successive overrelaxation method can be applied, yielding

$$C_{i,i}\,\tilde{u}_{1,i}^{(m+1)} = C_{i,i-1}\,u_{1,i-1}^{(m+1)} - C_{i,i+1}\,u_{1,i+1}^{(m)} + \hat{f}_{1,i}\,,$$

$$u_{1,i}^{(m+1)} = u_{1,i}^{(m)} + \omega\,(\tilde{u}_{1,i}^{(m+1)} - u_{1,i}^{(m)}),\ m \geq 0. \tag{2.7}$$

As before, the value of the relaxation factor ω which minimizes the spectral radius of the block successive overrelaxation matrix is given by

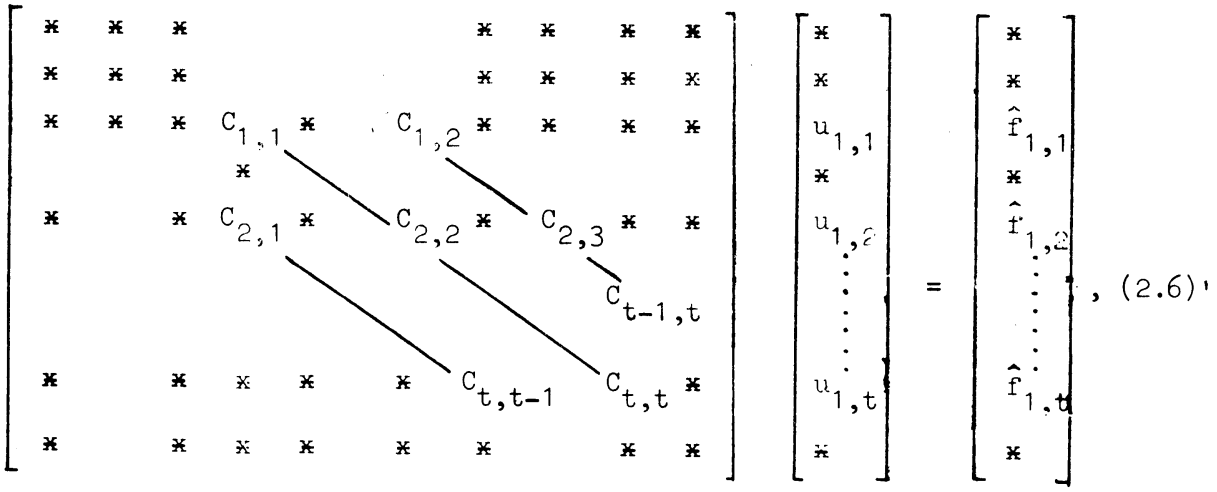
$$\omega = \omega_b = \frac{2}{1 + \sqrt{1-\rho^2(B)}}\,,$$

where $\rho(B)$ is the spectral radius of the block Jacobi matrix of C i.e. $B = I - D^{-1}C$, where D is a block diagonal matrix containing the diagonal submatrices of C. To simplify the matrix inversion from (2.7) the mesh points of the i-th block of two successive diagonal mesh lines are labeled as follows

	* 3	* 5	* 7	* 9	
* 1	* 2	* 4	* 6	* 8	etc.,

so that the matrix $C_{i,i}$ takes the form of a symmetric, seven-diagonal matrix, and the first equation of (2.7) can be directly solved, for instance, by Cholesky decomposition (see p. 16). We shall call this method the 2LSOR (1) method. We shall discuss now a modification of the method which needs less computational work. Therefore we consider first of all those points, satisfying scheme $(2.5)'$, which can be ordered by blocks consisting of two diagonal mesh lines of equal length. Analogous to the foregoing we apply the block SOR method to these points.

The components of u_1 at the remaining ("extra") points are properly handled. Matrix equation (2.6) can be written in this case, using the same notation



, (2.6)'

if there are t pairs of lines of equal length. The asterisks stand for the matrices and vectors resulting from the application of the point or line Gauss-Seidel method at the "extra" points.
If we number the $2m_i$ mesh points of the i-th block as follows

$$\begin{matrix} * & * & * & * \\ 1 & 2 & 3 & 4 \\ * & * & * & * \\ m_i+1 & m_i+2 & m_i+3 & m_i+4 \end{matrix} \qquad \text{etc.,}$$

so that

$$u_{1,i} = \begin{bmatrix} u_{1,i}^{(1)} \\ u_{1,i}^{(2)} \end{bmatrix} ,$$

and if we write analogous to the first equation of (2.7)

$$C_{i,i}\, \tilde{u}_{1,i} = h_i , \qquad (2.7)'$$

then equation (2.7)' can be written as

$$\begin{bmatrix} E_1 & E_2 \\ E_2 & E_1 \end{bmatrix} \begin{bmatrix} \tilde{u}_{1,i}^{(1)} \\ \tilde{u}_{1,i}^{(2)} \end{bmatrix} = \begin{bmatrix} h_i^{(1)} \\ h_i^{(2)} \end{bmatrix} , \qquad (2.7)''$$

where in view of (2.5)' the $m_i x m_i$ matrices E_1 and E_2 are equal to

$$E_1 = \begin{bmatrix} 3/4 & 1/8 & & & \\ 1/8 & 3/4 & 1/8 & & O \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1/8 \\ O & & & 1/8 & 3/4 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 1/8 & 1/16 & & & \\ 1/16 & 1/8 & 1/16 & & O \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1/16 \\ O & & & 1/16 & 1/8 \end{bmatrix} \quad \text{and} \quad h_i = \begin{bmatrix} h_i^{(1)} \\ h_i^{(2)} \end{bmatrix}.$$

Let (Parter, 1959)

$$P = \begin{bmatrix} I & I \\ -I & I \end{bmatrix}, \qquad P^{-1} = \tfrac{1}{2}\begin{bmatrix} I & -I \\ I & I \end{bmatrix},$$

where I is the $m_i x m_i$ identity matrix, then

$$P^{-1} C_{i,i} P = \begin{bmatrix} E_1+E_2 & 0 \\ 0 & E_1-E_2 \end{bmatrix} = L,$$

so that with

$$\tilde{u}_{1,i} = P\tilde{x}_i, \tag{2.8}$$

equation (2.7)'' gives

$$L\tilde{x}_i = P^{-1} h_i.$$

After inversion of the tridiagonal matrix L the vector $\tilde{u}_{1,i}$ is found from (2.8). We shall call this method the 2LSOR (2) method. We can number the blocks of mesh points such that the block Jacobi matrix of C from (2.6) has a block form analogous to (2.3) (σ_1 ordering). In this case (2.7) is applied first of all to the even blocks and then to the odd. If the iteration parameter ω, during each of these successive iterations, is changed according to

$$\omega_1 = 1, \quad \omega_2 = \frac{2}{2-\rho^2(B)}, \quad \omega_{i+1} = \frac{1}{1-\omega_i \rho^2(B)/4}, \quad i \geq 2,$$

where $\rho(B)$ is again the spectral radius of the block Jacobi matrix of C, then we are applying the two-line cyclic Chebyshev semi-iterative (2LCC (1)) method. In a slightly different way we can apply the foregoing to the 2LSOR (2) method. Although this variant of the 2LSOR (1) (or 2LCC (1)) method doesn't follow the theory exactly it appears to be very fast in practical cases.

(3) ADI method.

We express the matrix A from (1.3) as the matrix sum

$$A = H+V,$$

where H and V result from difference approximations of $-\frac{\delta^2}{\delta x^2}$ and $-\frac{\delta^2}{\delta y^2}$. Suppose we have a grid with s horizontal mesh lines. If we label the mesh points by natural ordering and if we call the vector u(i,j) defined on these ordered mesh points $u_h = (u_{h,1}, \ldots, u_{h,s})^T$, then the matrix H corresponding with this vector has the form

$$H = \begin{bmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_s \end{bmatrix}, \quad H_j = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ 0 & & -1 & 2 \end{bmatrix}_{n_j x n_j}, \quad \sum_{j=1}^{s} n_j = n,$$

where n_j is the number of mesh points of the j-th horizontal mesh line. Analogous considerations hold with respect to the matrix V if we number the mesh points along vertical mesh lines.
We rewrite equation (1.3) in the form

$$(H + \omega I)\, u = (\omega I - V)\, u + k,$$

or

$$(V + \omega I)\, u = (\omega I - V)\, u + k.$$

The Peaceman-Rachford (1955) alternating-direction implicit (ADI) iterative method is defined by

$$\begin{aligned}(H+\omega_{m+1}I)u^{(m+\frac{1}{2})} &= (\omega_{m+1}I - V)u^{(m)} + k,\\ (V+\omega_{m+1}I)u^{(m+1)} &= (\omega_{m+1}I - H)u^{(m+\frac{1}{2})} + k,\ m \geq 0,\end{aligned} \tag{2.9}$$

where the acceleration parameters ω_m are to be chosen so as to make the convergence of the method rapid. Combination of the pair of equations of (2.9) gives

$$u^{(m+1)} = T_{\omega_{m+1}} u^{(m)} + l_{\omega_{m+1}},\ m \geq 0,$$

where

$$T_\omega = (V+\omega I)^{-1}(\omega I - H)(H+\omega I)^{-1}(\omega I - V),$$

$$l_\omega = (V+\omega I)^{-1}\{(\omega I - H)(H+\omega I)^{-1} + I\}k.$$

The matrix T_ω is called the Peaceman-Rachford matrix. In this report we apply the ADI method in the form given by Wachspress and Habetler (1960) (obtained by rewriting iteration process (2.9) in a suitable way)

$$w^{(0)} = k - (V - \omega_1 I)u^{(0)},$$

$$v^{(m+1)} = \{I - 2\omega_{m+1}(H+\omega_{m+1}I)^{-1}\}w^{(m)}, \tag{a}$$

$$u^{(m+1)} = (V+\omega_{m+1}I)^{-1}(k - v^{(m+1)}), \tag{b}$$

$$w^{(m+1)} = v^{(m+1)} + (\omega_{m+2} + \omega_{m+1})u^{(m+1)},\ m \geq 0 \tag{c}$$

with 4 and 8 Wachspress (1962) iteration parameters. After the calculation of $w^{(o)}$, $v^{(1)}$ and $u^{(1)}$ the cycle (c, a, b) is applied as long as further iteration is required.

(4) ICCG method.

We start again from the system of linear equations (1.3)

$$Au = k,$$

where the matrix A is symmetric and positive definite. First we shall give an outline of the conjugate gradient method of Hestenes and Stiefel (1952). Let $u^{(0)}$ be an initial guess then we can try to approximate the solution u of (1.3) by putting

$$u^{(m)} = u^{(0)} + \sum_{i=1}^{m} \alpha_i A^{i-1} (Au^{(0)} - k), \qquad (2.10)$$

where the α_i are to be chosen so as to minimize

$$|| u^{(m)} - u ||_A,$$

where $|| x ||_A^2 = (x, Ax)$, i.e. the inproduct of x and Ax. We can write (2.10) equivalently as

$$u^{(m)} - u = u^{(0)} - u + \sum_{i=1}^{m} \alpha_i A^i (u^{(0)} - u)$$

or

$$u^{(m)} - u = P_m (A) (u^{(0)} - u),$$

where $P_m(A)$ is a polynomial in the matrix A of degree m.

Compare also the method of Richardson (Varga, 1962, p. 141). Parameters α_i which minimize $||u^{(m)} - u||_A$ can be found very simply by orthogonalization of the set of vectors

$$A^i (u^{(0)} - u) \qquad i = 1, \ldots, m$$

in the $||\ ||_A$ norm. For an application of the Gram-Schmidt orthogonalization procedure, see Beckman (1960, p. 63). If the orthogonal set is written as

$$u_i^* = \sum_{j=1}^{i} C_{i,j} A^j (u^{(0)} - u) \qquad i = 1, \ldots, m$$

then $(u_i^*, Au_j^*) = 0$ for $i \neq j$ and $(u_i^*, Au_i^*) > 0$ since the matrix A is positive definite and (2.10) becomes

$$u^{(m)} - u = u^{(0)} - u + \sum_{i=1}^{m} \alpha_i^* u_i^*. \tag{2.11}$$

Parameters α_i^* which minimize $H(\alpha_i^*) = ||u^{(m)} - u||_A$ follow from

$$\frac{dH(\alpha_i^*)}{d\alpha_i^*} = 0$$

so that

$$\alpha_i^* = - \frac{(u_i^*, A(u^{(0)} - u))}{(u_i^*, Au_i^*)}, \quad i = 1, \ldots, m, \tag{2.11}'$$

i.e. $-\sum_{i=1}^{m} \alpha_i^* u_i^*$ is the projection of $u^{(0)} - u$ in the space spanned by $u_1^*, \ldots, u_m^*$. A recursive application of the foregoing is given by the algorithm of Hestenes and Stiefel: Let $r_0 = k - Au^{(0)}$ and $p_0 = r_0$, then

$$\begin{aligned}
&a_i = (r_i, r_i) / (p_i, Ap_i), \\
&u^{(i+1)} = u^{(i)} + a_i p_i, \\
&r_{i+1} = r_i - a_i Ap_i, \\
&b_i = (r_{i+1}, r_{i+1}) / (r_i, r_i), \\
&p_{i+1} = r_{i+1} + b_i p_i, \qquad i = 0, 1, \ldots
\end{aligned} \tag{2.12}$$

The vectors p_i are the u_i^* from (2.11). From (2.11) and (2.11)' it is clear that $u^{(0)} - u = -\sum_{i=1}^{n} \alpha_i^* u_i^*$ so that $u^{(n)} = u$. If the matrix A has r district eigen values then the $A^j (u^{(0)} - u)$, $j = 0,1\ldots$ lie in a r-dimensional subspace, so that $u^{(r)} = u$. It appears to be advantageous to apply the conjugate gradient method to the solution of systems of linear equations of which the coefficient matrix has many nearly-degenerate eigen values. This is of course the case if A does not differ very much from the identity matrix. Meijerink and Van der Vorst (1977) did accomplish this by giving an incomplete Cholesky decomposition of A, so that $A \approx LDL^T$ and by applying the conjugate gradient method with matrix $(LDL^T)^{-1}A$. If a complete Cholesky decomposition is given by

$$A = LDL^T,$$

where $L = (l_{i,j})$ is a lower triangular matrix and $D = (d_{i,j})$ a diagonal matrix with

$$l_{i,j} = a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}\, l_{j,k}\, d_{k,k}, \qquad i = j, (j+1), \ldots, n,$$

$$d_{i,i} = l_{i,i}^{-1},$$

then one of the choices for an approximated decomposition is to take $l_{i,j} = 0$ if $a_{i,j} = 0$ leading to the ICCG (0) method. For further details, see Meijerink and Van der Vorst (1977) and Kershaw (1978). Iteration schema (2.12) is modified as follows: Let $r_0 = k - Au^{(0)}$ and $p_0 = (LDL^T)^{-1} r_0$, then

$$a_i = (r_i, (LDL^T)^{-1} r_i) / (p_i, Ap_i),$$

$$u^{(i+1)} = u^{(i)} + a_i p_i,$$

$$r_{i+1} = r_i - a_i Ap_i,$$

$$b_i = (r_{i+1}, (LDL^T)^{-1} r_{i+1}) / (r_i, (LDL^T)^{-1} r_i),$$

$$p_{i+1} = (LDL^T)^{-1} r_{i+1} + b_i p_i, \qquad i = 0,1, \ldots .$$

The ICCG(0) method, treated in section 4 will be referred to as the ICCG method.

3. Stop criteria for iterative methods.

In the foregoing section we discussed some iterative methods (although the conjugate gradient method is often regarded as a direct one) for the solution of a system of linear equations

$$Au = k. \tag{3.1}$$

The error after m iterations, $\varepsilon^{(m)} = u^{(m)} - u$ can be written in a general way as

$$\varepsilon^{(m)} = K_m \varepsilon^{(0)} , \tag{3.2}$$

where for instance $K_m = I + \sum_{i=1}^{m} \alpha_i A^i$ for the CG method,

$K_m = \{(I-\omega L)^{-1} (\omega U + (1-\omega)I)\}^m$ for the point SOR method with analogous expressions for the block methods and $K_m = \prod_{j=1}^{m} T_{\omega_j}$,

$T_\omega = (V + \omega I)^{-1} (\omega I - H) (H + \omega I)^{-1} (\omega I - V)$ for the ADI method. In the following we shall consider some stop criteria that could be handled to terminate the iteration proces. We shall first define the vector (and corresponding matrix) norms that we are using in the experiments. We regard vectors and matrices with real elements.

Definition 3.1.

Let x be a vector with components $x_1, \ldots, x_n$. Then

$$\| x \|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{\frac{1}{2}} \text{ is the } l_2 \text{ (or Euclidean) norm of } x,$$

$$\| x \|_\infty = \max_{1 \le i \le n} |x_i| \text{ is the } l_\infty \text{ norm of } x.$$

Definition 3.2.

Let $A = (a_{i,j})$ be a n x n matrix with eigen values λ_i, $1 \leq i \leq n$.

Then

$$||A||_2 = \max \frac{||Ax||_2}{||x||_2}$$ is the l_2 (or spectral) norm of the

matrix A,

$$||A||_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{i,j}|$$ is the l_∞ norm of A and for the sake

of completeness,

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i|$$ is the spectral radius of A.

In the following $||\ ||$ stands both for $||\ ||_2$ and $||\ ||_\infty$.

Definition 3.3.

Let A be a convergent n x n matrix, i.e. $\rho(A)<1$. If, for some positive integer m, $||A^m||_2<1$, then

$$R(A^m) \equiv \ln \{(||A^m||_2)^{1/m}\} = \frac{\ln ||A^m||_2}{m}$$

is the average rate of convergence for m iterations of the matrix A and

$$R_\infty(A) \equiv -\ln \rho(A)$$

is the asymptotic rate of convergence for the matrix A.

Using vector and matrix norms in equation (3.2) it follows that

$$||\varepsilon^{(m)}|| \leq ||K_m|| \cdot ||\varepsilon^{(0)}||,$$

so that we are led to the problem of minimizing $|| K_m ||$, $m > 0$, (this doesn't hold for the ICCG method) in order to obtain a largest possible error reduction factor. Let us suppose that we have solved this minimization problem, so that the iteration parameters that minimize $|| K_m ||$ for the different iterative methods are known. Then experiments are made in the following two cases. In the first case we suppose that $k = 0$ in (3.1) so that equation (3.1) has the solution $u = 0$. Starting from an initial approximation $u^{(0)}$ we determine the minimum number of iterations, so that

$$\frac{|| u^{(m)} ||}{|| u^{(0)} ||} < 10^{-q}, \qquad q \text{ positive integer.} \tag{3.3}$$

In the second case these experiments are repeated using a vector k resulting from a forcing function taken from the practice of numerical weather prediction.

Now we consider the minimum number of iterations so that

$$|| Au^{(m)} - k || < 10^{-q}, \qquad q \text{ positive integer,} \tag{3.4}$$

starting with $u^{(0)} = 0$. We proceed with two theorems.

Theorem 3.1. (Varga, 1962, p. 10, 11, 15).

Let A be a n x n matrix. Then

$$|| A ||_2 = (\rho(A^T A))^{\frac{1}{2}},$$

where A^T is the transpose of A, and moreover

$$\rho(A) \leq || A ||.$$

Theorem 3.2. (Varga, 1962, p. 65).

Let A be an arbitrary n x n matrix such that $\rho(A) > 0$. Then,

$$|| A^m || \sim \nu \binom{m}{p-1} \{\rho(A)\}^{m-(p-1)}, \quad m \to \infty, \tag{3.5}$$

where p is the largest order of all diagonal submatrices J_γ of the Jordan normal form of A (Varga, p. 13) with $\rho(J_\gamma) = \rho(A)$, and ν is a positive constant.

For simplicity we confine ourselves to the point SOR method in the following. The matrix $H(\omega)$ from section 2 satisfies (3.5) of theorem 3.2 with $p = 2$ so that by choosing

$$\omega = \omega_b = \frac{2}{1 + (1-\lambda)^{\frac{1}{2}}} , \tag{3.6}$$

where $\lambda = \rho^2(B)$, $||\{H(\omega)\}^m||$ is minimized for large values of m. In this case λ can be found as follows. Solution of (3.1) by means of the Gauss-Seidel method (i.e. $\omega = 1$ in (2.1)) gives

$$u^{(m+1)} = (I-L)^{-1} U u^{(m)} + (I-L)^{-1} k.$$

Defining

$$\sigma_m = || u^{(m)} - u^{(m-1)} ||_\infty ,$$

then

$$\frac{\sigma_{m+1}}{\sigma_m} \to \lambda, \quad m \to \infty , \tag{3.7}$$

and the best value of ω follows from (3.6). Another way of determining ω_b experimentally is the use of criterion (3.3). As an example we take the problem of the next section where the Poisson equation is solved on a regular octagon, as sketched in Figure 2. Using (3.6) and (3.7) we find in that case $\omega_b = 1.8628$ after 489 iterations. In the following Table we give the number of iterations needed to satisfy criterion (3.3) with $q = 3$ for different values of ω, using the $||\ ||_\infty$ norm.

ω	iterations
1.80	141
81	132
82	123
83	113
84	104
85	93
86	82
87 *	76 *
88	83
89	83
90	87

Table 1.

Another stop criterion that one might use, is to terminate the iteration process, if

$$||\, u^{(m+1)} - u^{(m)} \,|| < 10^{-q}, \quad q \text{ positive integer.} \tag{3.8}$$

It is easy so see that this criterion doesn't treat different SOR methods equivalently. Therefore we write

$$u^{(m+1)} - u^{(m)} = \varepsilon^{(m+1)} - \varepsilon^{(m)} =$$

$$(\{H(\omega)\}^{m+1} - \{H(\omega)\}^{m})\ \varepsilon^{(0)} = (H(\omega) - I)\ \{H(\omega)\}^{m} \varepsilon^{(0)} \quad ,$$

so that

$$||u^{(m+1)} - u^{(m)}|| \leq ||H(\omega) - I|| \cdot ||\{H(\omega)\}^m|| \cdot ||\varepsilon^{(0)}||. \quad (3.9)$$

Further we have

$$H(\omega) - I = (I - \omega L)^{-1}(\omega U + (1-\omega)I - (I - \omega L)) =$$

$$-\omega (I - \omega L)^{-1} (I-B),$$

so that

$$||H(\omega)-I|| \leq \omega ||I - B|| \cdot ||(I-\omega L)^{-1}||. \quad (3.10)$$

Since L is a strictly lower triangular n xn matrix, we have that

$$(I - \omega L)^{-1} = I + \omega L + \ldots + \omega^{n-1} L^{n-1}$$

is a lower triangular matrix with

$$||(I - \omega L)^{-1}|| = N(\omega) = ||I + \omega L + \ldots + \omega^{n-1} L^{n-1}||, \quad (3.11)$$

so that, since L is nonnegative, $N(\omega)$ is an increasing functior of ω.

Combination of (3.9), (3.10) and (3.11) gives

$$||u^{(m+1)} - u^{(m)}|| \leq \omega N(\omega) \cdot ||\{H(\omega)\}^m|| \cdot ||I - B|| \cdot ||\varepsilon^{(0)}|| \quad (3.12)$$

If criterion (3.8) is used to compare different SOR methods with optimal relaxation parameters, for instance point and block methods, it is clear from (3.12) that the more rapid methods will be favoured. It may be still worse if criterion (3.8) is used to determine the "best" value of ω.

In this case one doesn't find this value of ω by minimizing $||\{H(\omega)\}^m||$, for $m \to \infty$ leading to $\omega = \omega_b$, but the "best" value of ω is found from minimization of

$$\omega N(\omega) \cdot ||\{H(\omega)\}^m||$$

after m iterations, causing a decrease in ω with respect to ω_b, as long as the decrease of $\omega N(\omega)$ will be larger than the increase of $||\{H(\omega)\}^m||$. We shall illustrate this using the same example as in Table 1. In Table 2 we give the number of iterations needed to satisfy criterion (3.8) with q=3 for different values of ω, using the $||\ ||_\infty$ norm.

ω	iterations
1.80	90
81	86
82	82
83	78
84	74
85 *	70 *
86	72
87	74
88	79
89	87
90	89

Table 2.

If the point SOR method is applied with equations and unknowns ordered so that the matrix B has the form (2.3) it is possible to obtain (Sheldon, 1959)

$$||\{H(\omega_b)\}^m||_2 = \left(\frac{2m}{\rho(B)} + \sqrt{\frac{4m^2}{\rho^2(B)} + 1}\right) \{\rho(H(\omega_b))\}^m$$

4. Results.

In this section we shall describe the experiments which were carried out on the regular octagon as sketched in Fig. 2, where the number of interior mesh points is indicated for horizontal and vertical mesh lines. The total number of interior mesh points amounts to 1624.

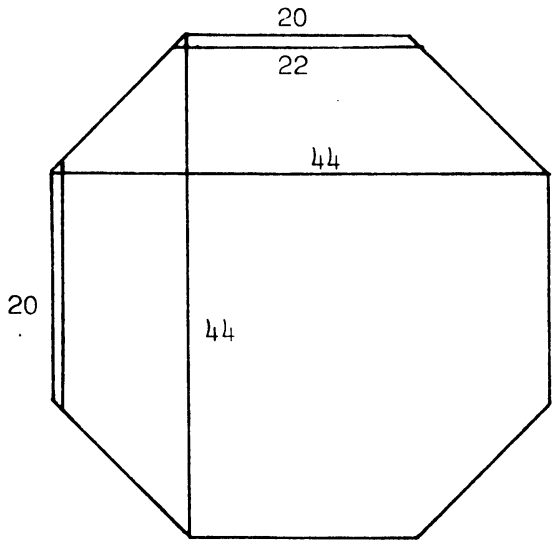


Fig. 2.

The basis of comparison used, described in the foregoing sections is briefly mentioned. We consider two cases. In the first case we suppose that $k = 0$ in (3.1) so that $u = 0$. Starting with $u_i^{(0)} = 1$, $i = 1, \ldots, 1624$, where $u_i^{(0)}$ is the i-th component of $u^{(0)}$, we shall determine the minimum number of iterations

so that

$$\frac{||u^{(m)}||}{||u^{(0)}||} < 10^{-q} \quad , \quad q = 1, \ldots, 10. \tag{4.1}$$

For the block iterative methods applied to the cyclically reduced matrix equation, the m-th iterate is extended over the whole grid, if necessary, by means of one of the equations (2.4). The computed optimal relaxation parameters (or corresponding spectral radius of the Jacobi matrix) for the SOR (and CC) methods are given below.

point SOR	$\omega_b = 1.8628$
2LSOR (1)	$\omega_b = 1.6586$
2LCC (1)	$\rho(B) = 0.9786$
2LSOR (2)	$\omega_b = 1.6592$
2LCC (2)	$\rho(B) = 0.9787$

We remind that the 2LSOR (2) and 2LCC (2) methods do not satisfy the theory exactly. These computed values were verified experimentally by applying (4.1) for different values of ω. Some results are given in Fig. 3 through 5. It is clear that the real optimal parameters are slightly larger than the computed values, as was to be expected. Results of comparative tests of the iterative methods using criterion (4.1) are given in Table 3, 4.

	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
point SOR	43	59	76	88	108	128	138	152	176	193
2LSOR (1) α_1 ord.	11	16	21	24	33	40	46	49	54	63
2LSOR (2) α_1 ord.	12	17	22	26	33	39	45	49	53	60
2LCC (1)	11	15	23	27	33	38	44	51	57	61
2LCC (2)	11	15	21	24	32	38	44	48	53	59
ICCG	12	14	19	25	29	32	36	39	41	44
ADI (4 par)	6	7	10	14	16	18	22	26	27	30
ADI (8 par)	6	10	11	14	18	20	24	26	30	34
2LSOR (1) nat. ord.	14	18	25	28	35	43	48	53	57	64
2LSOR (2) nat. ord.	15	19	25	29	33	42	47	53	57	61

Table 3, $||\ ||_\infty$

	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
point SOR	32	52	66	78	100	115	132	147	165	183
2LSOR (1) σ_1 ord.	9	15	19	23	31	38	43	47	51	60
2LSOR (2) σ_1 ord.	10	16	20	24	27	37	43	48	52	55
2LCC (1)	7	12	19	25	31	35	41	48	54	59
2LCC (2)	9	14	18	22	29	36	42	46	51	55
ICCG	10	12	18	23	27	30	34	38	40	42
ADI (4 par)	4	6	10	12	14	18	20	23	26	30
ADI (8 par)	4	8	10	13	16	18	22	26	28	32
2LSOR (1) nat. ord.	11	17	22	27	33	39	45	49	55	63
2LSOR (2) nat. ord.	11	18	23	28	31	39	46	50	55	58

Table 4, $||\ ||_2$

Next the 2LSOR (2) method (σ_1 ordering)(with the 2LCC (2) method the most rapid SOR (CC) method) is compared with the point SOR method, the ADI method and the ICCG method. The total number of iterations was normalized by the relative amount of arithmetic per iteration required by each method. The arithmetic requirement for the different methods were weighted as follows (obtained by comparing actual machine time)

point SOR	1
2LSOR (1), 2LCC (1)	1.80
2LSOR (2), 2LCC (2)	1.05
ADI	2.67
ICCG	2.28

The numbers of observed iterations were multiplied by these values and called normalized iterations. Results are given in Fig. 6 and 7. In the second case the vector k is taken from practice. We now start with $u_i^{(0)} = 0$, $i = 1, \ldots, 1624$ and determine the number of iterations so that

$$||\ Au^{(m)} - k\ || = ||\ A\varepsilon^{(m)}\ || < 10^{-q},\ q = 1, \ldots, 8. \qquad (4.2)$$

As before optimal relaxation parameters of the SOR and CC methods are verified experimentally. Results are found in Fig. 8 through 15. Due to the typical effect of the matrix A, criterion (4.2) is extremely suited to determine the optimal relaxation parameter experimentally. Comparative test, analogous to Table 3 and 4, using criterion (4.2) are found in Table 5 and 6.

	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
point SOR	76	91	104	119	138	152	170	184
2LSOR (1) σ_1 ord.	22	27	33	39	44	49	55	60
2LSOR (2) σ_1 ord.	22	28	33	39	45	50	56	62
2LCC (1)	23	28	34	39	45	50	55	61
2LCC (2)	23	29	34	40	46	51	57	62
ICCG	22	26	31	38	41	46	50	53
ADI (4 par)	9	13	15	17	21	24	27	29
ADI (8 par)	9	13	15	17	21	23	27	29
2LSOR (1) nat. ord.	23	30	35	40	46	51	57	63
2LSOR (2) nat. ord.	23	29	34	41	47	52	58	64

Table 5, $||\ ||_\infty$

	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
point SOR	86	100	115	133	148	164	180	198
2LSOR (1) σ_1 ord.	27	33	38	44	49	55	60	66
2LSOR (2) σ_1 ord.	27	32	38	44	49	55	61	66
2LCC (1)	28	33	39	44	50	55	61	66
2LCC (2)	28	33	39	45	50	56	61	67
ICCG	26	31	38	41	46	50	53	57
ADI (4 par)	12	15	17	21	24	26	29	33
ADI (8 par)	13	15	17	20	23	26	29	33
2LSOR (1) nat. ord.	28	34	39	45	50	56	62	67
2LSOR (2) nat. ord.	28	34	40	45	51	57	63	68

Table 6, $||\ ||_2$

Finally in Fig. 16 and 17 the 2LSOR(2) method is compared with the ADI method, the ICCG method and the point SOR method with respect to the number of normalized iterations. For the sake of completeness we also give the number of iterations for the point SOR method so that

$$\frac{\| u^{(m)} - u \|_2}{\| u \|_2} < 10^{-q} , \qquad q = 1,2,3,4,5 ,$$

where u satisfies (3.1), for different values of ω, see Fig. 18. These results can be compared with those of Fig. 3.

5. References.

Beckman, F.S. (1960) — The solution of linear equations by the conjugate gradient method, in: Mathematical methods for digital computers, A. Ralston, H.S. Wilf eds, John Wiley, New York, pp. 62-72.

Bijlsma, S.J. (1977) — Iteratieve methoden voor het oplossen van elliptische partiële differentiaal-vergelijkingen, Wetenschappelijk Rapport Kon. Ned. Meteor. Inst., De Bilt, WR 77-8.

Hageman, L.A. (1962) — Block iterative methods for two-cyclic matrix equations with special application to the numerical solution of second-order self-adjoint elliptic partial differential equations in two dimensions, WAPD-TM-327, Bettis Atomic Power Laboratory, Pittsburgh.

Hestenes, M. and Stiefel, E. (1952) — Method of conjugate gradients for solving linear systems, report 1659, Nat. Bur. Standards.

Kershaw, D.S. (1978) — The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations, J. Comp. Phys., 26, pp. 43-65.

Meijerink, J.A. and van der Vorst, H.A. (1977) — An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. Comp., 31, pp. 148-162.

Parter, S.V. (1959) — On "two-line" iterative methods for the Laplace and biharmonic difference equations, Num. Math., 1, pp. 240-255.

Reid, J.K. (1972) — The use of conjugate gradients for systems of linear equations possessing "property A", SIAM J. Numer. Anal., 9, pp. 325-332.

Sheldon, J.W. (1959) — On the spectral norms of several iterative processes, J. Assoc. Comput. Mach., 6, pp. 494-505.

Stone, L.S. (1968) — Iterative solution of implicit approximations of multidimensional partial differential equations, SIAM J. Numer. Anal., 5, pp. 550-558.

Varga, R.S. (1962) — Matrix iterative analysis, Prentice-Hall, Inc., Englewood Cliffs, N.J.

Wachspress, E.L. and Habetler, H.J. (1960) — An alternating-direction-implicit iteration technique, J. Soc. Indust. Appl. Math., 8, pp. 403-424.

Wachspress, E.L. (1962) — Optimum alternating-direction-implicit iteration parameters for a model problem, J. Soc. Indust. Appl. Math, 10, pp. 339-350.

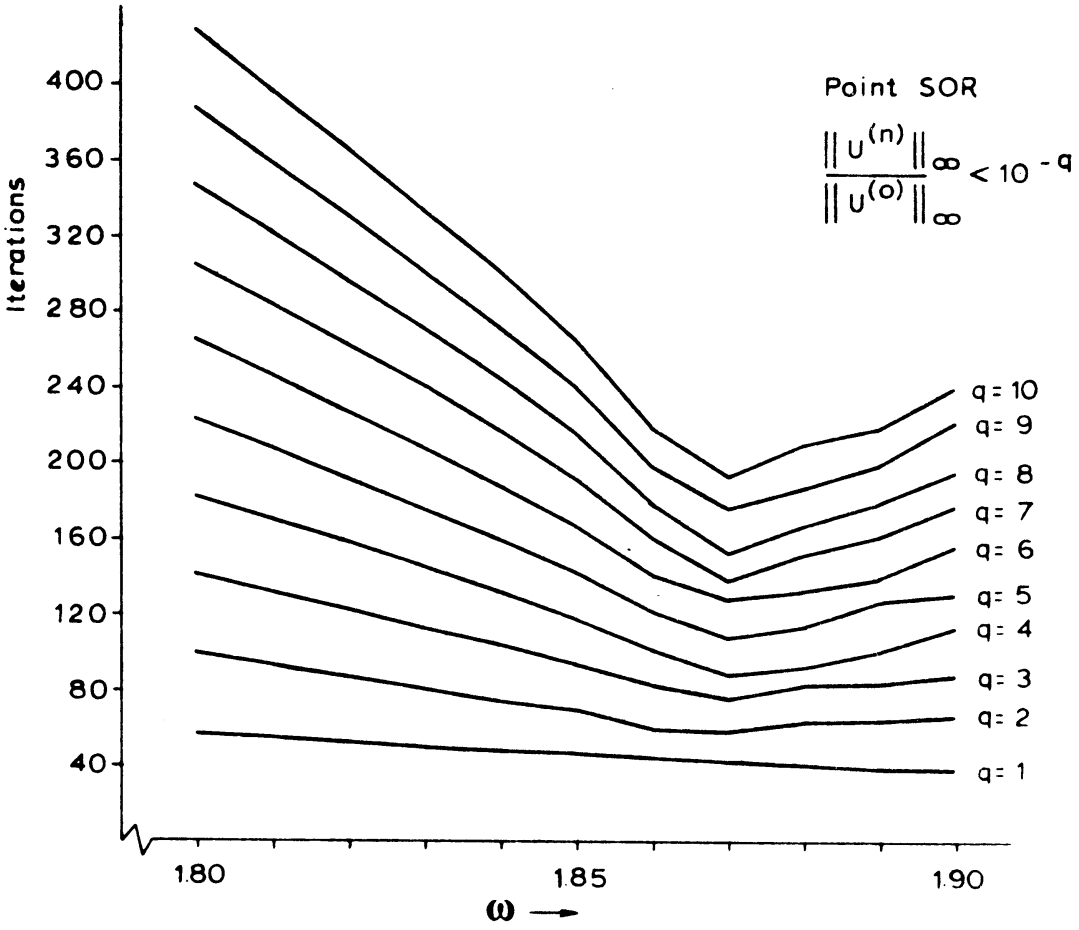

Point SOR
$\frac{\|U^{(n)}\|_\infty}{\|U^{(o)}\|_\infty} < 10^{-q}$
Iterations
400
360
320
280
240
200
160
120
80
40
q = 10
q = 9
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
1.80
1.85
1.90
ω →

Fig. 3

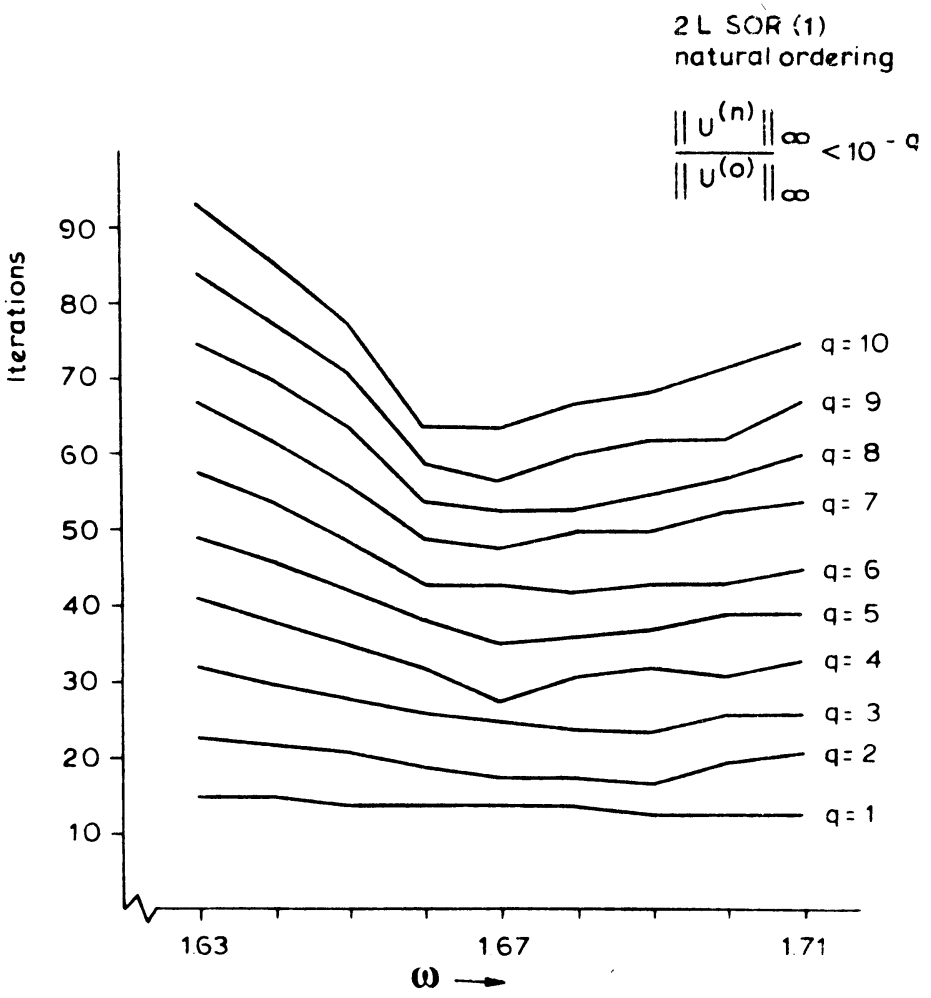

2 L SOR (1)
natural ordering
$\frac{\|U^{(n)}\|_\infty}{\|U^{(0)}\|_\infty} < 10^{-q}$
Iterations
90
80
70
60
50
40
30
20
10
q = 10
q = 9
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
1.63
1.67
1.71
ω

Fig. 4

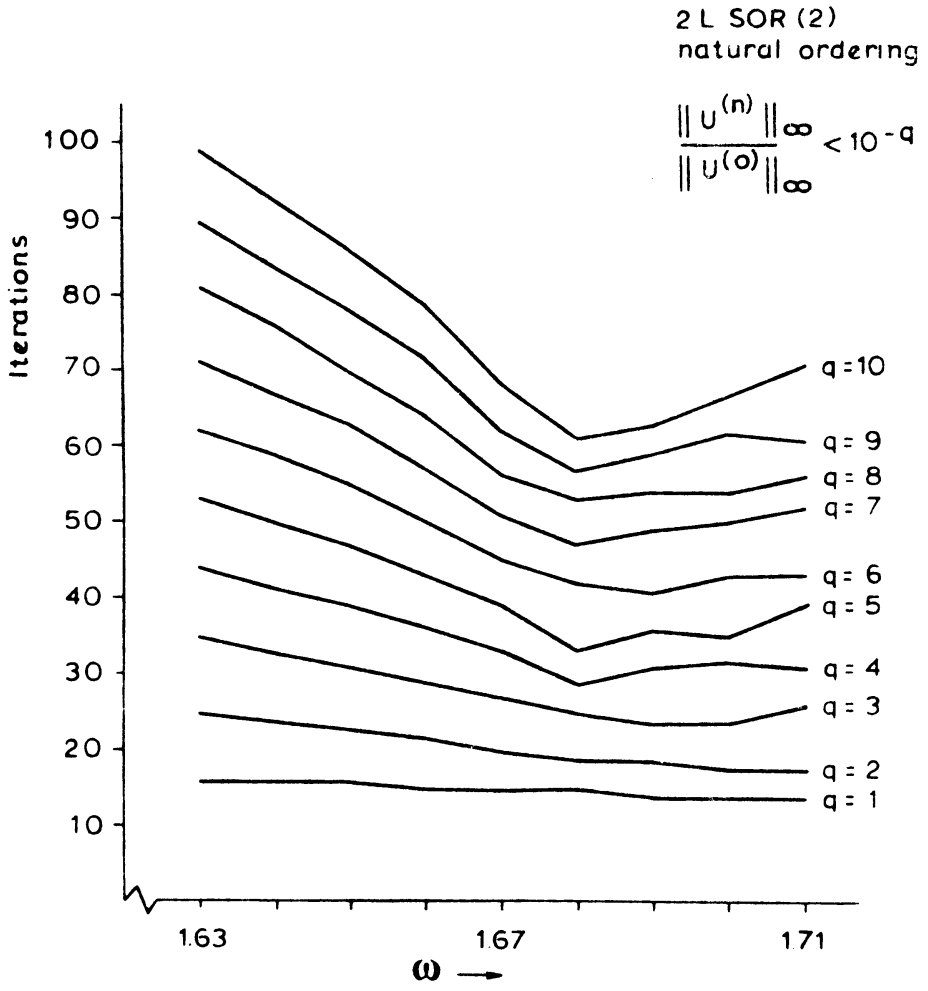

2 L SOR (2)
natural ordering
$\frac{\|U^{(n)}\|_\infty}{\|U^{(o)}\|_\infty} < 10^{-q}$
Iterations
100
90
80
70
60
50
40
30
20
10
q=10
q= 9
q= 8
q= 7
q= 6
q= 5
q= 4
q= 3
q= 2
q= 1
1.63
1.67
1.71
ω →

Fig. 5

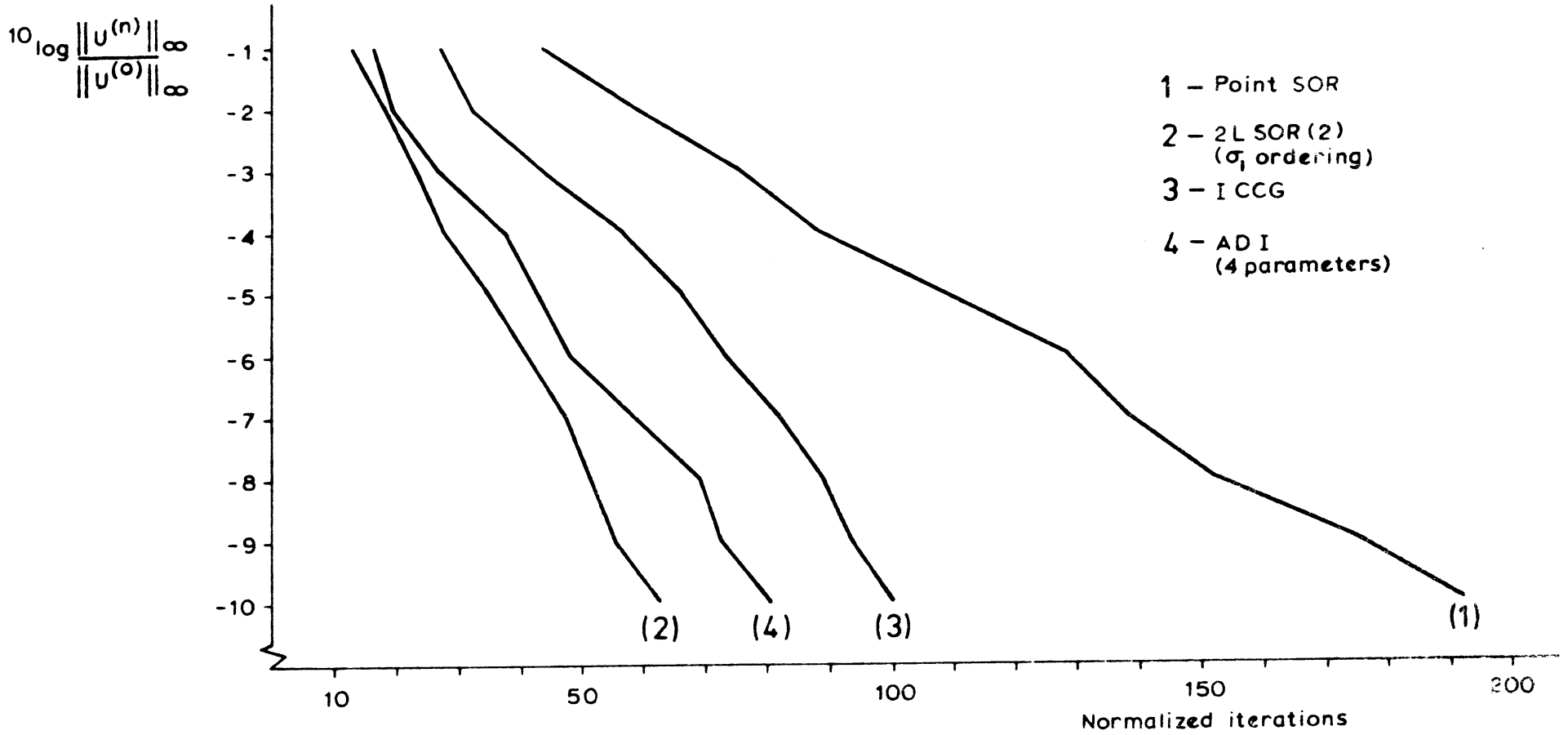

$^{10}\log \frac{\|u^{(n)}\|_\infty}{\|u^{(o)}\|_\infty}$
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
1 – Point SOR
2 – 2L SOR (2)
(σ_1 ordering)
3 – I CCG
4 – AD I
(4 parameters)
(2)
(4)
(3)
(1)
10
50
100
150
200
Normalized iterations

Fig. 6

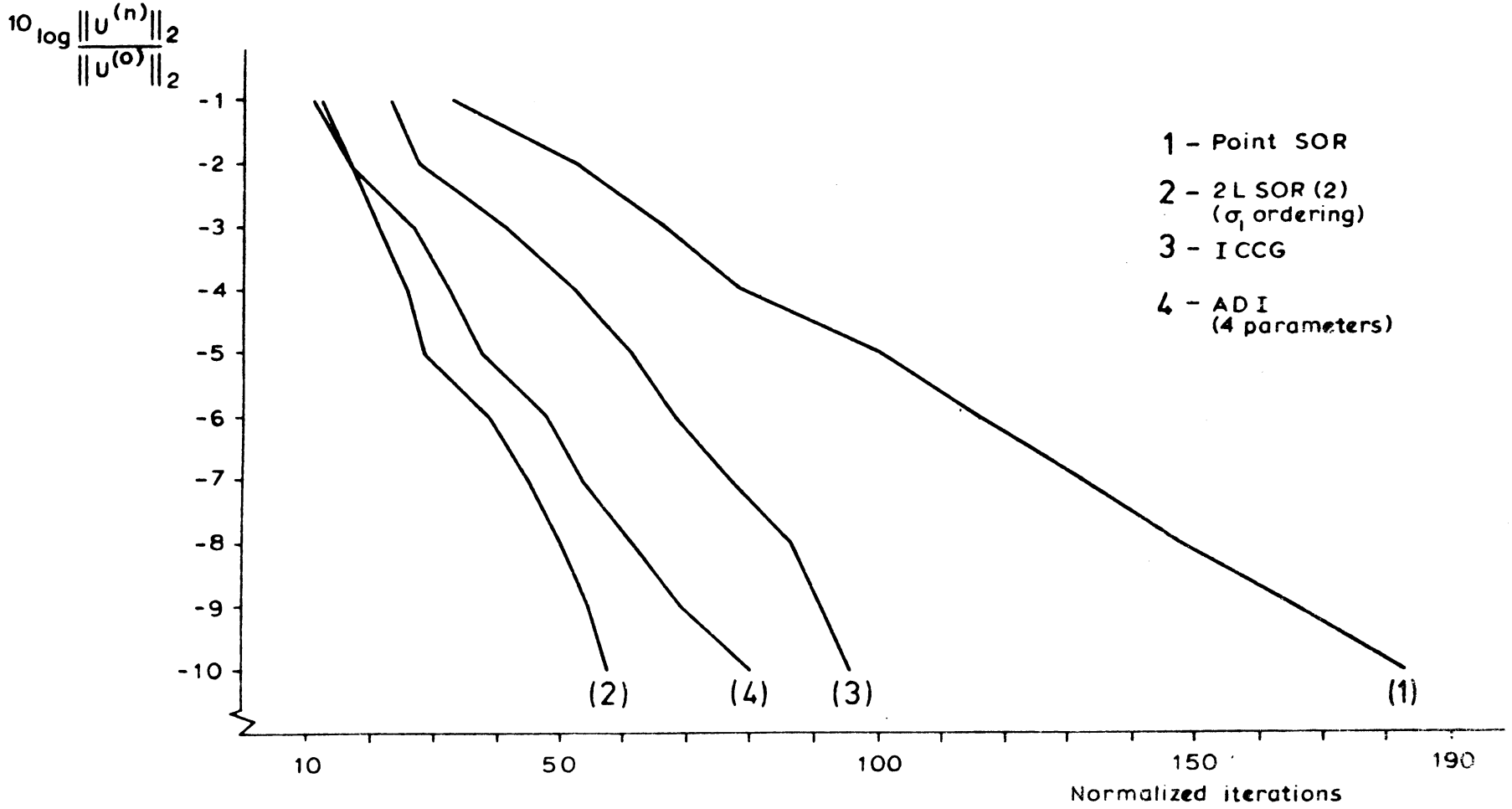

$^{10}\log \frac{\|u^{(n)}\|_2}{\|u^{(o)}\|_2}$
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
1 - Point SOR
2 - 2 L SOR (2) (σ_1 ordering)
3 - I CCG
4 - ADI (4 parameters)
(2)
(4)
(3)
(1)
10
50
100
150
190
Normalized iterations

Fig. 7

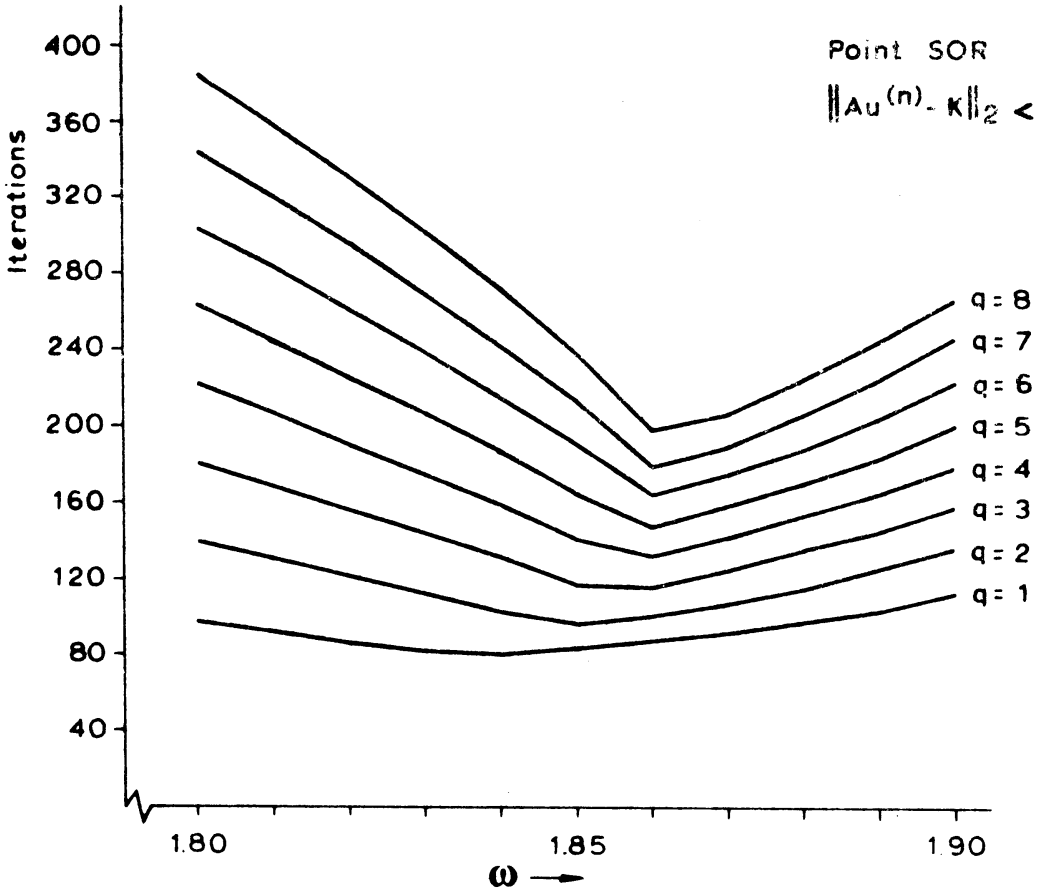

Point SOR
$\|Au^{(n)} - K\|_2 < 10^{-q}$
Iterations
400
360
320
280
240
200
160
120
80
40
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
1.80
1.85
1.90
ω →

Fig. 8

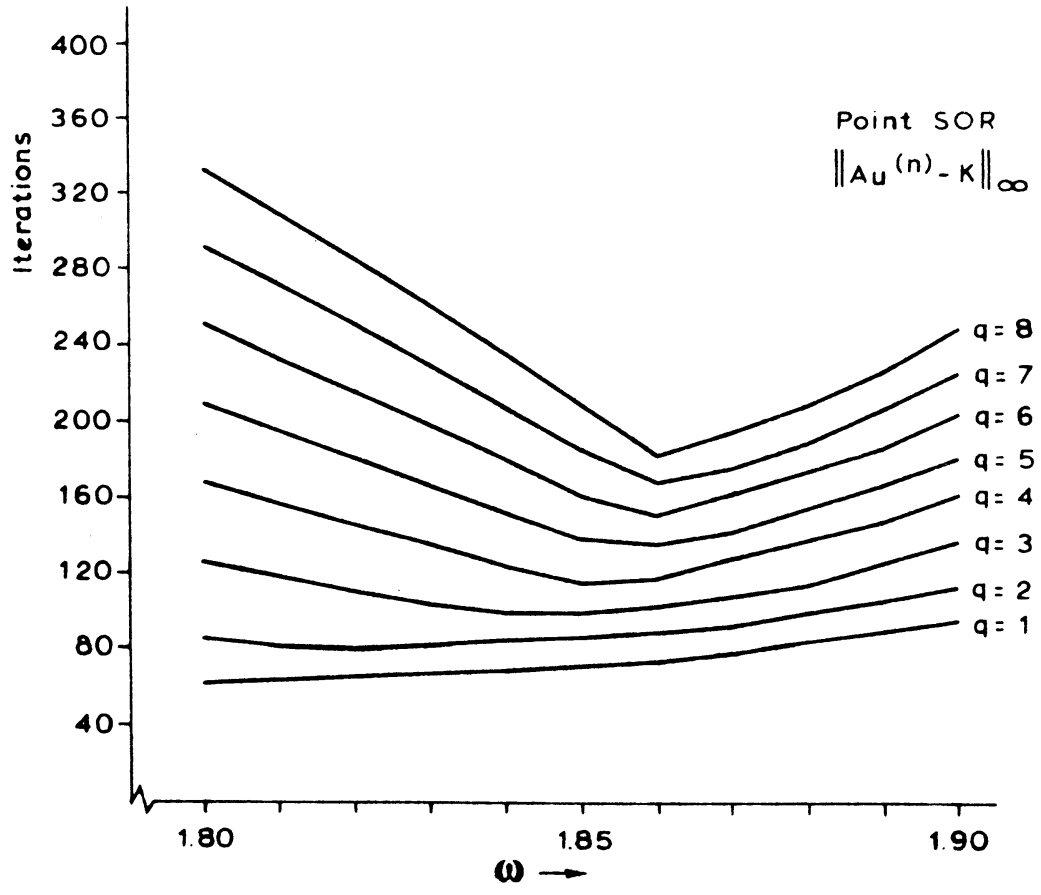

Point SOR
$\|Au^{(n)} - K\|_\infty < 10^{-q}$
Iterations
400
360
320
280
240
200
160
120
80
40
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
1.80
1.85
1.90
ω →

Fig. 9

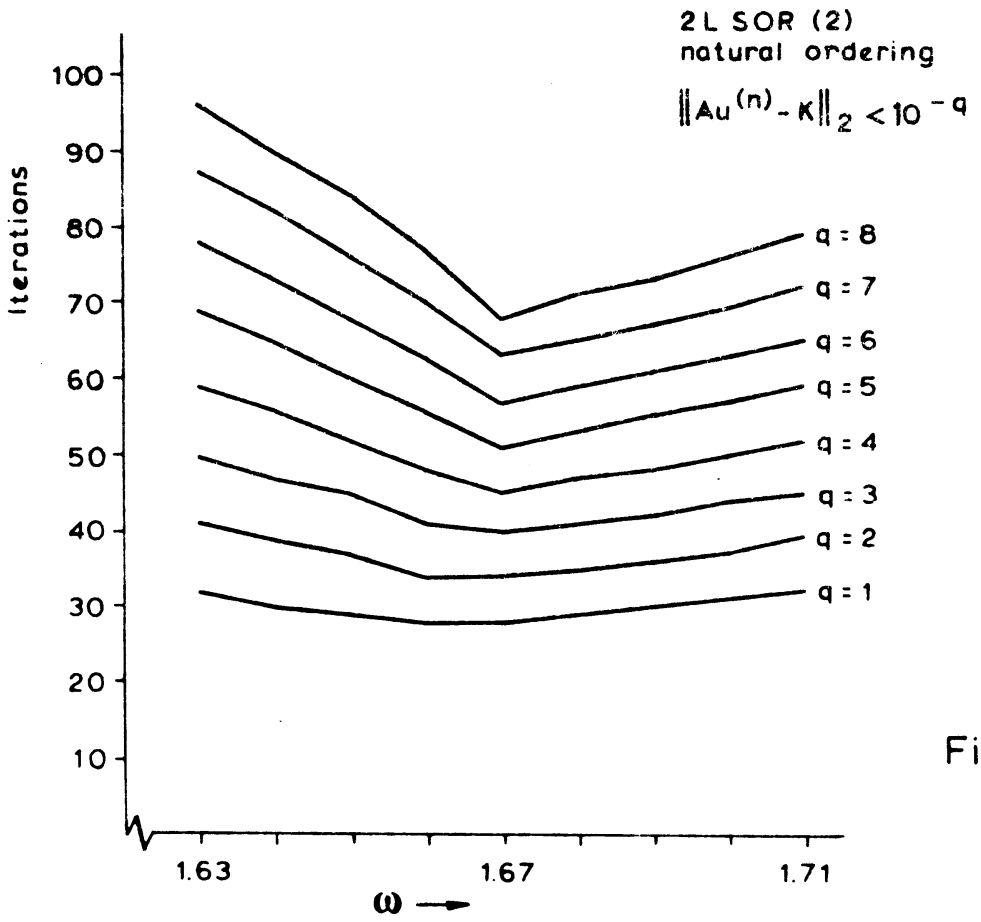

2 L SOR (2)
natural ordering
$\|Au^{(n)} - K\|_2 < 10^{-q}$
Iterations
100
90
80
70
60
50
40
30
20
10
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
1.63
1.67
1.71
ω

Fig. 10

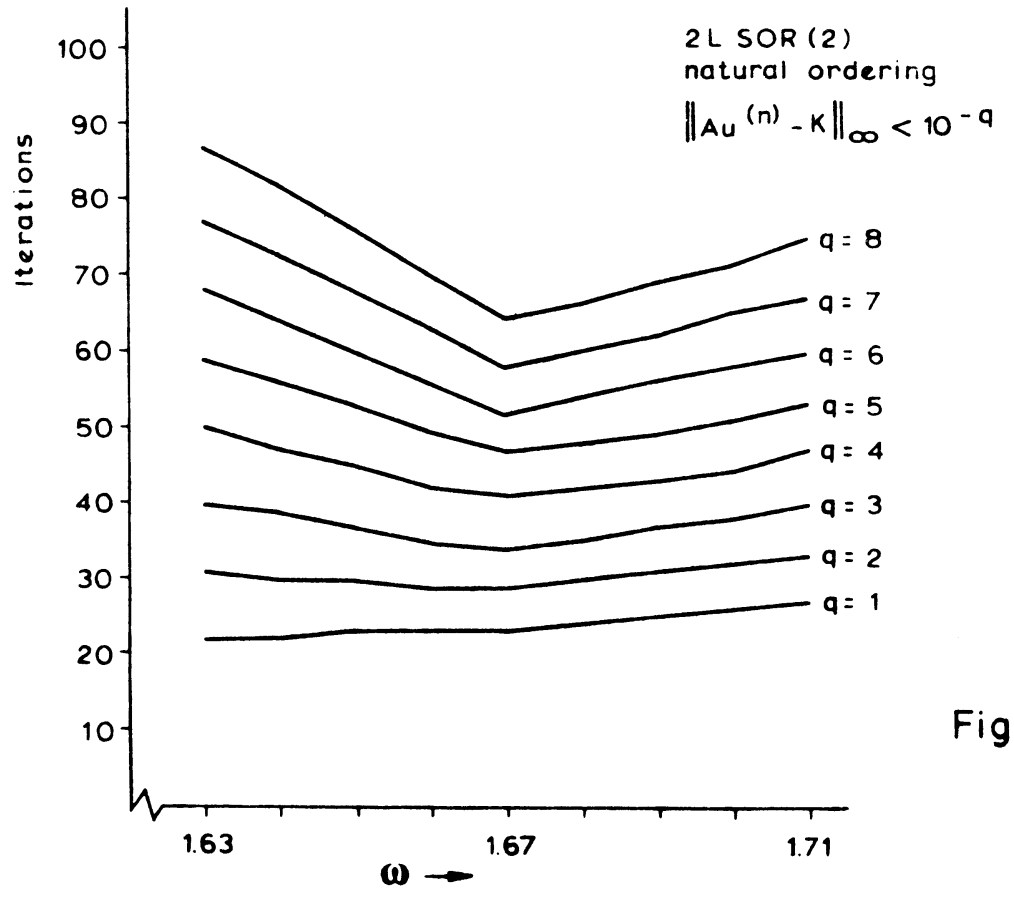

2 L SOR (2)
natural ordering
$\|Au^{(n)} - K\|_\infty < 10^{-q}$
Iterations
100
90
80
70
60
50
40
30
20
10
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
1.63
1.67
1.71
ω

Fig. 11

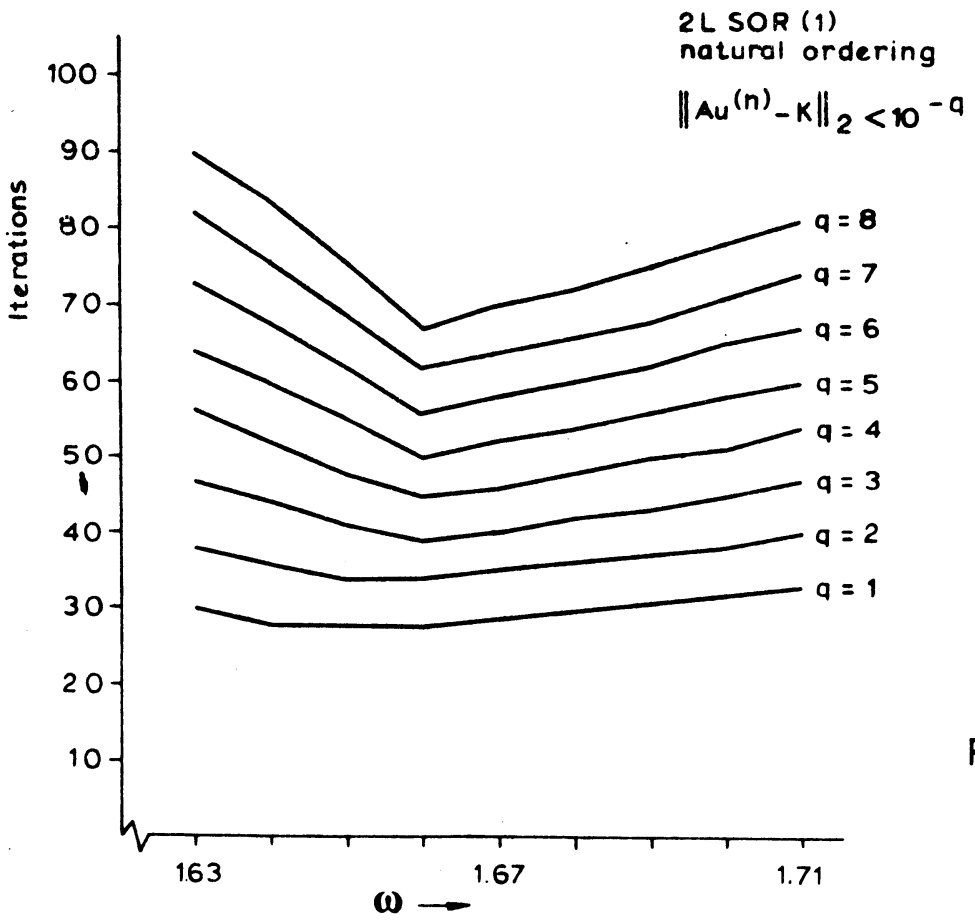

2L SOR (1)
natural ordering
$\|Au^{(n)} - K\|_2 < 10^{-q}$
Iterations
100
90
80
70
60
50
40
30
20
10
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
1.63
1.67
1.71
ω

Fig. 12

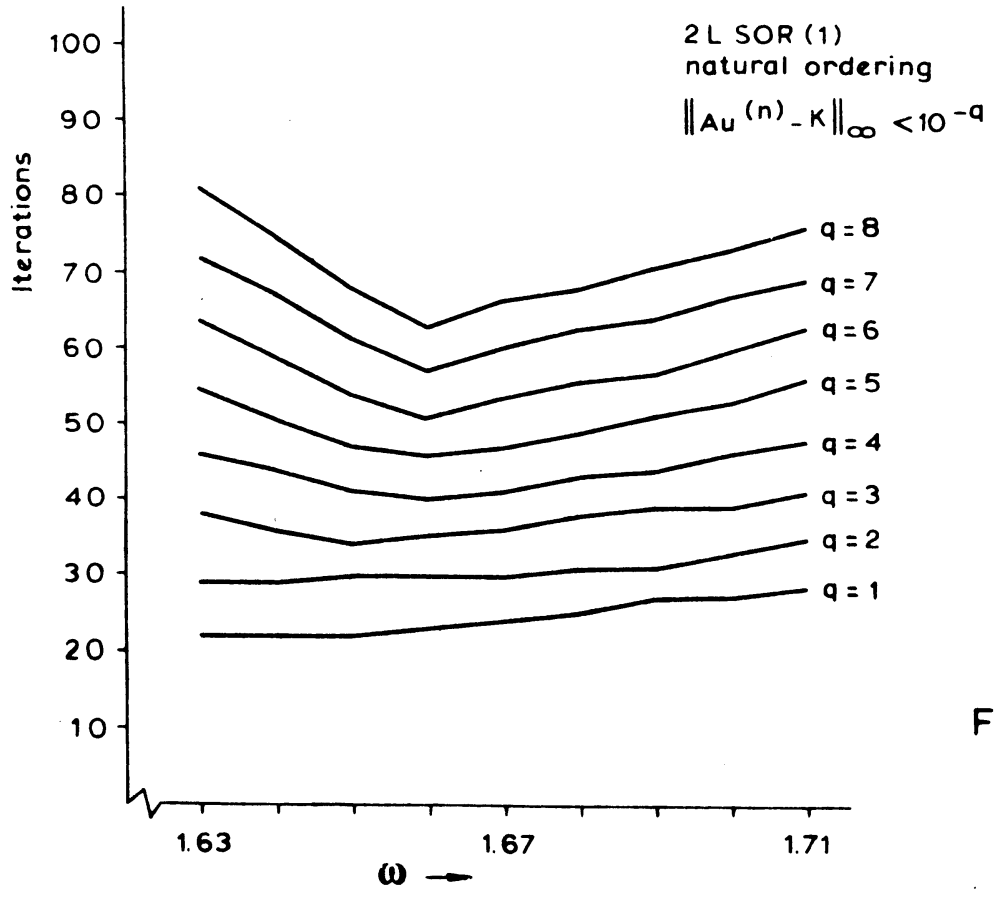

2L SOR (1)
natural ordering
$\|Au^{(n)} - K\|_\infty < 10^{-q}$
Iterations
100
90
80
70
60
50
40
30
20
10
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
1.63
1.67
1.71
ω

Fig. 13

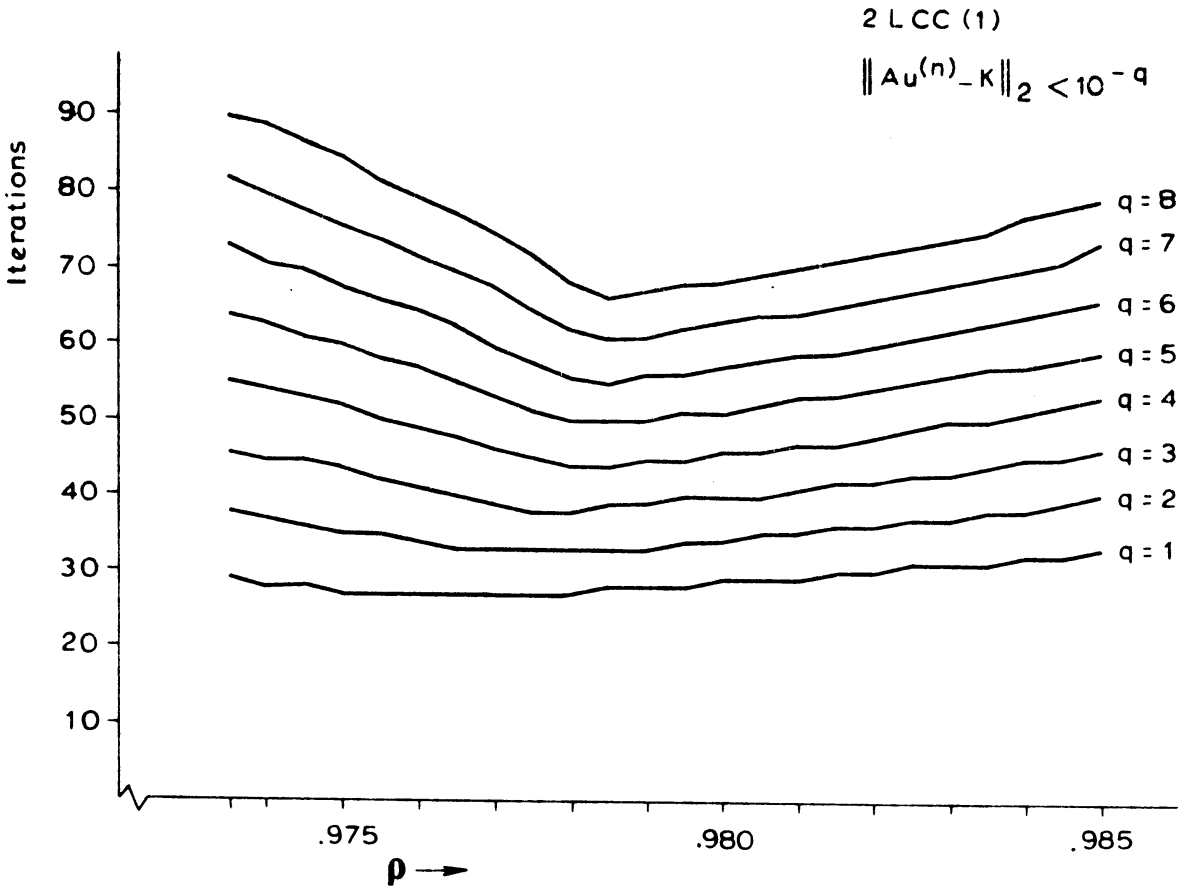

2 L CC (1)
$\|Au^{(n)} - K\|_2 < 10^{-q}$
Iterations
90
80
70
60
50
40
30
20
10
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
.975
.980
.985
ρ →

Fig. 14

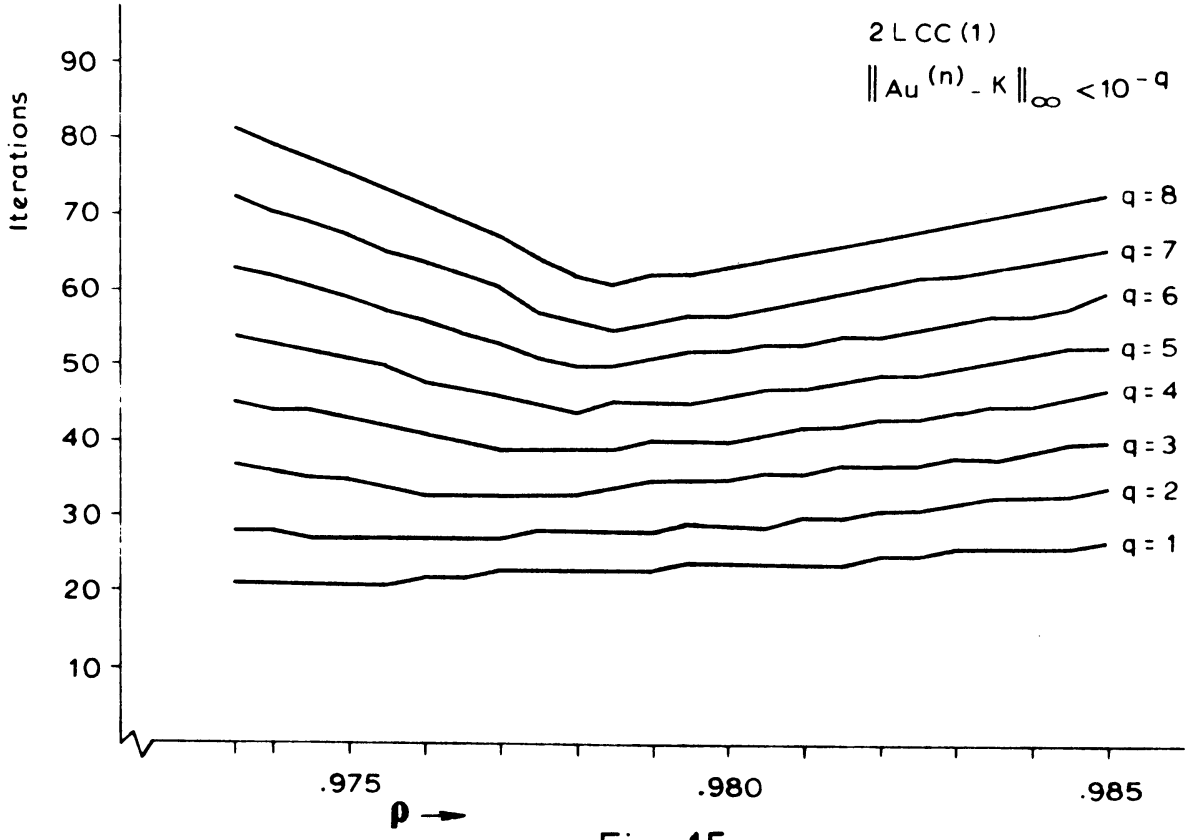

2 L CC (1)
$\|Au^{(n)} - K\|_\infty < 10^{-q}$
Iterations
90
80
70
60
50
40
30
20
10
q = 8
q = 7
q = 6
q = 5
q = 4
q = 3
q = 2
q = 1
.975
.980
.985
ρ →

Fig. 15

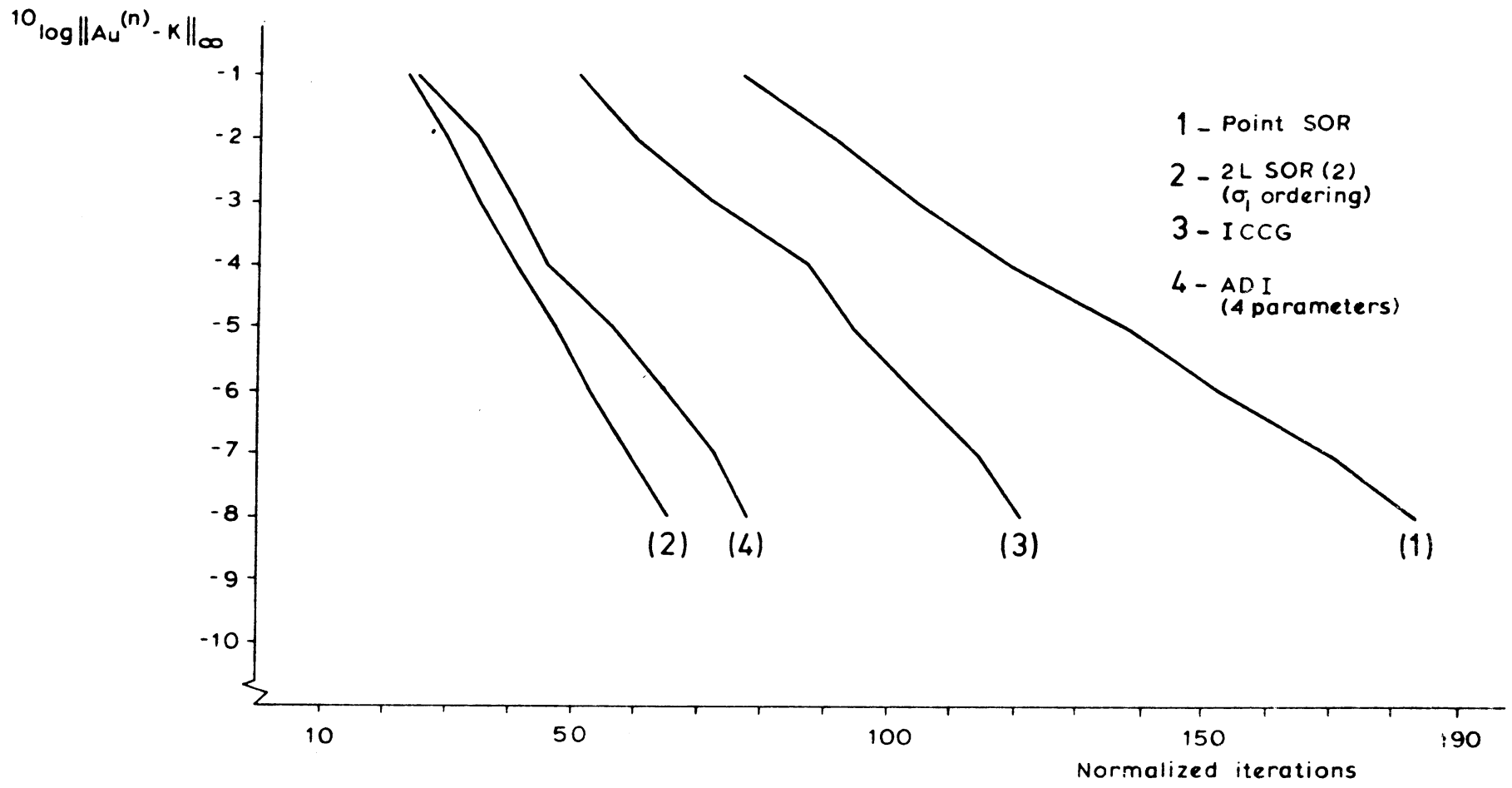

$^{10}\log \|Au^{(n)} - K\|_\infty$
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
10
50
100
150
190
Normalized iterations
1 - Point SOR
2 - 2L SOR (2) (σ_1 ordering)
3 - ICCG
4 - ADI (4 parameters)
(2)
(4)
(3)
(1)

Fig. 16

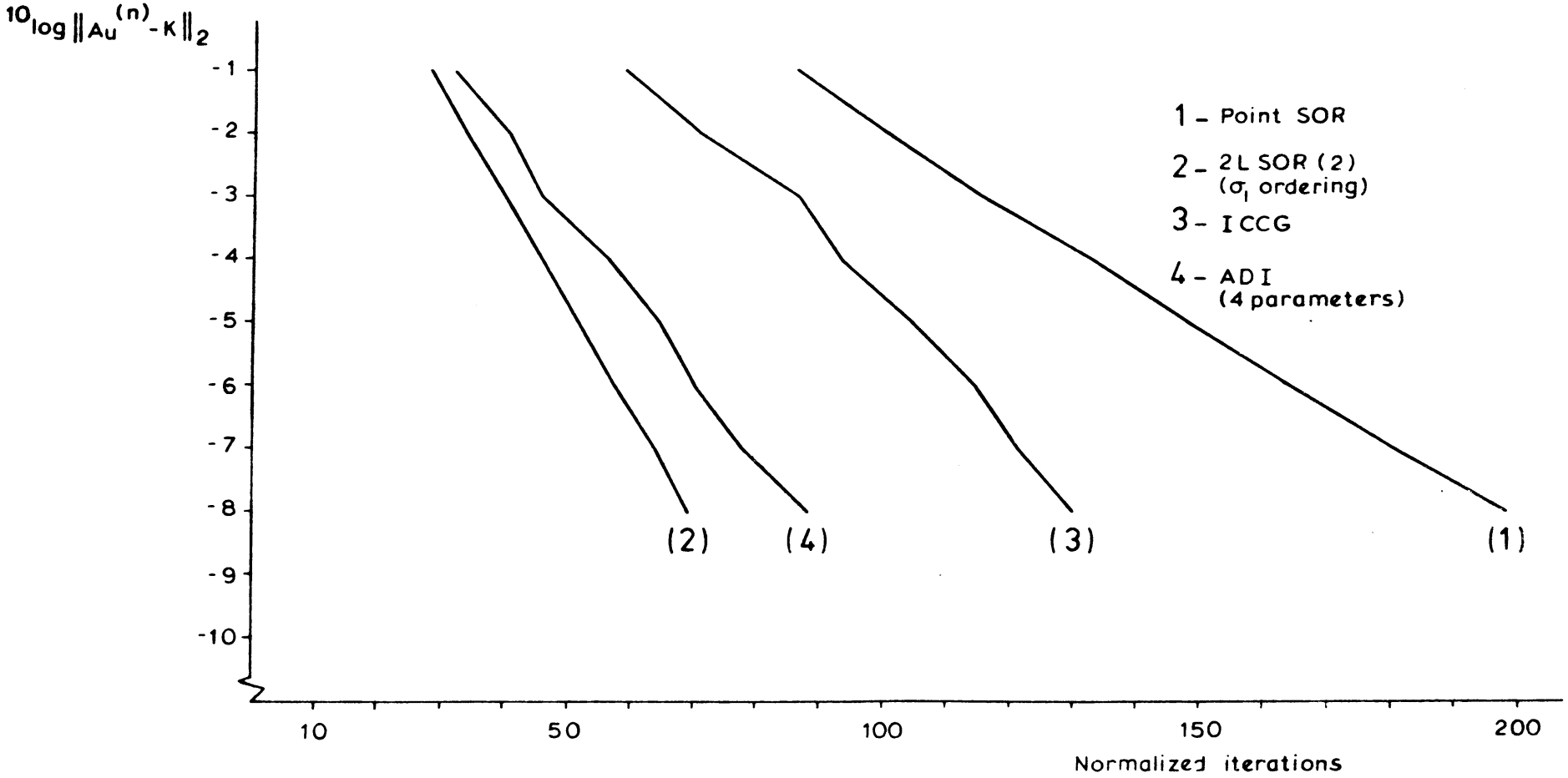

$^{10}\log \|Au^{(n)} - K\|_2$
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
10
50
100
150
200
Normalized iterations
1 - Point SOR
2 - 2L SOR (2) (σ_1 ordering)
3 - ICCG
4 - ADI (4 parameters)
(2)
(4)
(3)
(1)

Fig. 17

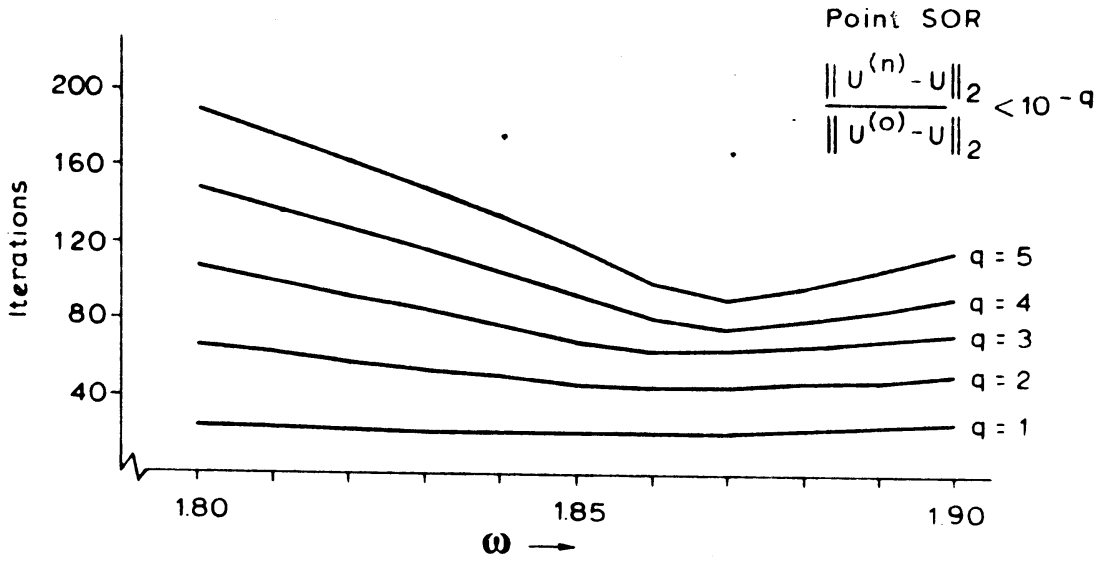

Point SOR
$\frac{\|U^{(n)}-U\|_2}{\|U^{(o)}-U\|_2} < 10^{-q}$
Iterations
200
160
120
80
40
q = 5
q = 4
q = 3
q = 2
q = 1
1.80
1.85
1.90
ω →

Fig. 18