STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

AFDELING TOEGEPASTE WISKUNDE

TW 109

On the acceleration of Richardson's method IV
A non-symmetrical case

by

T.M.T. Coolen

and

P.J. van der Houwen

July 1968.

The Mathematical Centre at Amsterdam, founded the 11th of February, 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications, and is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.) and the Central Organization for Applied Scientific Research in the Netherlands (T.N.O.), by the Municipality of Amsterdam and by several industries.

Contents

1. Introduction

Richardson's method, and its acceleration, the reduction-elimination method, employed to solve iteratively matrix equations of the type

$$Lu = f,$$

where L is a symmetric matrix, have been discussed extensively in the previous three reports [4], [5] and [6] in this series. The essence of the reduction-elimination method is, as we have seen in [4], the reduction of the late eigenfunctions of L, followed by the elimination of the remaining eigenfunctions. It has been shown that the average rate of convergence rises condiderably, compared with Richardson's method.

However, both methods have only been applied to symmetrical matrix equations. A model problem, the Dirichhlet problem for Laplace's equation in a square of side π, has been worked out in [6].
In this same report another way of solving this model problem was suggested: a non-symmetrical matrix equation was constructed, of which the solution also satisfied Laplace's equation. It was shown that the P-condition number [1] of the five-point (+) and (×) schemes one can associate with this matrix equation, are one fourth of the P-condition numbers of the corresponding Jacobi (+) and (×) schemes respectively, from which it follows that the asymptotic rates of convergence are twice the asymptotic rates of convergence of the Jacobi schemes.

In this report Richardson's method and the reduction-elimination method are both applied to the non-symmetrical matrix equation. However, when these methods were tested on the Electrologica X8 computer, some complications turned up. A detailed analysis is given here for the model problem, in which is shown, that these were entirely due to the fact that the schemes used are non-symmetric and that using such non-symmetrical schemes leads to some unexpected phenomena. A large number of numerical results are given, and one will also find a comparison with the numerical results of other non-stationary and stationary iterative methods for solving the model problem.

In the last section, in which stationary processes are discussed, also an extension of the method of successive overrelaxation is suggested.

Finally, through the results of the investigations carried out in this report we were led to the question unto which measure the accuracy of the solution of the boundary value problem is influenced by the choice one makes from the various discrete analogues of the Laplace operator, which only differ in the orientation of the molecule with respect to the coordinate axes. It would be interesting to examine whether this influence also depends on the orientation of the boundaries of the boundary value problem.

2. Preliminaries

In the previous reports of this series, [4], [5] and [6], the matrix equations.

(2.1) $\quad Lu = f$

in which L is a symmetric operator, was discussed extensively. In this section we shall recall some of the main results of the theory developed there. Furthermore, a few additional remarks will be made about non-symmetric matrices L.

An approximation of the solution of the matrix equation (2.1) is given by a sequence of functions u_k, the terms of which are calculated by means of the iterative process

(2.2) $$u_{k+1} = (1 - \omega_k L)u_k + \omega_k f, \quad k = 0, 1, 2, \ldots,$$

where u_0 is the initial approximation. Writing

$$u_k = u + v_k$$

so that v_k may be considered as the error of the approximation u_k, it follows from (2.2) that

$$u_k = u + P_k(L)v_o,$$

in connection with which one defines the average rate of convergence for K iterations as the quantity

(2.3) $$R(K) = - \frac{\ln ||P_K(L)||}{K}.$$

If the matrix L is symmetrix, then (2.3) is equivalent to

(2.4) $$R(K) = - \frac{\ln \sigma(P_K(L))}{K},$$

where $\sigma(P_K(L))$ is the spectral radius of the matrix $P_K(L)$.

Let us denote the number of iterations by K, and let us assume that the eigenvalues λ_i of L satisfy

$$a = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_M = \sigma(L) = b.$$

Under the condition $a > 0$ Richardson's method may be used to find an approximate solution of (2.1), which describes an iterative process

having an average rate of convergence

$$R(K) \approx 2\sqrt{\frac{a}{b}} - \frac{\ln 2}{K} \tag{2.5}$$

for sufficiently large K. The reduction-elimination method, worked out in [4] and [6], the essence of which is the reduction of the eigenfunctions belonging to eigenvalues inside the interval $[a,b]$, where now $a > \lambda_1$, followed by the elimination of the remaining eigenfunctions, leads to an average rate of convergence

$$R(K+K^*) \approx 2\sqrt{\frac{a}{b}} - \frac{2K^*\frac{a}{b} + \ln \sigma(E_{K^*}) + \ln 2}{K + K^*},$$

where K is the number of iterations performed during the reduction phase, K^* the number performed during the elimination phase, and where $\sigma(E_{K^*})$ represents the spectral radius of the elimination operator $E_{K^*} = C_{K^*}(a^*,b,L)$, see [4].

Average rate of convergence in the non-symmetric case.

Until now we have assumed the matrix L to be symmetric. When L is non-symmetric, it is possible to find a lower bound for the average rate of convergence, if the system of eigenfunctions of L is complete. We may then write the difference v_0 between the initial approximation and the real solution as a linear combination of the eigenfunctions e_i of L

$$v_0 = \sum_{i=1}^{M} \alpha_i e_i.$$

From

$$||P_K(L)v_0|| = ||P_K(L) \sum_{i=1}^{M} \alpha_i e_i|| \leq \max|P_K(\lambda_i)| \sum |\alpha_i|$$

it follows that

$$||P_K(L)|| = \sup_{||v_0||=1} ||P_K(L)v_0|| \leq \max_i |P_K(\lambda_i)| \sup_{||v_0||=1} \sum|\alpha_i|.$$

This leads to

$$R(K) \geq -\frac{\ln \sigma(P_K(L)) + \ln \left(\sup_{||v_0||=1} \sum |\alpha_i| \right)}{K}.$$

Therefore, when the elimination method is used, the average rate of convergence will be

(2.7)
$$R(K) \geq 2\sqrt{\frac{a}{b}} - \frac{K^{*}\sqrt{\frac{a}{b}} + \ln \sigma(E_{K^{*}}) + \ln 2 + \ln \alpha}{K + K^{*}},$$

where

$$\alpha = \sup_{||v_0||=1} \sum |\alpha_i|.$$

The Laplace difference operator

In [6] the difference operator

(2.8)
$$D = \frac{Y_+ + \alpha + Y_-}{\alpha + 2} \cdot \frac{X_+ - 2 + X_-}{\xi^2} + \frac{X_+ + \beta + X_-}{\beta + 2} \cdot \frac{Y_+ - 2 + Y_-}{\eta^2}$$

was introduced, where $X_{\pm}$ and $Y_{\pm}$ represent translations $\pm\xi$ and $\pm\eta$ along the x-axis and y-axis respectively; α and β are weight parameters. Obviously D is a difference analogue of the Laplace differential operator Δ. We may write (2.8) in the form

(2.9)
$$D = L_1(X_+ + X_-)(Y_+ + Y_-) + L_2(X_+ + X_-) + L_3(Y_+ + Y_-) + L_4,$$

where

(2.10)
$$\begin{cases} L_1 = \frac{1}{2}\xi^{-2}(1 + \rho^2 - \gamma), \\ L_2 = \xi^{-2}(\gamma - \rho), \\ L_3 = \xi^{-2}(\gamma - 1), \\ L_4 = -2\gamma\xi^{-2}, \end{cases} \qquad \gamma = \frac{\alpha}{\alpha + 2} + \rho^2 \frac{\beta}{\beta + 2}, \quad \rho = \frac{\xi}{\eta}.$$

Restricting ourselves to the case of a square grid, i.e. $\rho = 1$, $L_2 = L_3$ we find a five-point formula (+) when $\gamma = 2$ ($L_1 = 0$), and a five-point formula ($\times$) when $\gamma = 1$ ($L_2 = L_3 = 0$).

The difference operator D applied in all interior lattice points of a certain region can be represented by a symmetric matrix, which will also be denoted by D. The aim is to solve the matrix equation

$$-Du = f \tag{2.11}$$

by means of an iterative method. In the same report [6] this equation has been treated numerically for a model problem, namely the Dirichlet problem for a square of side π. In section 5 of [6] a second approach, which we shall recall here in the remaining part of this section, to this problem was suggested. Details will be worked out in the next section.

Besides the operator D we introduce the operator

$$D_1 = pL_2X_- + qL_1X_-Y_- + qL_3Y_- + qL_1X_+Y_- , \tag{2.12}$$

where p and q are real parameters. Multiplying (2.11) by $(1 - D_1)^{-1}$ we find

$$-(1 - D_1)^{-1}Du = g, \tag{2.13}$$

in which $g = (1 - D_1)^{-1}f$. Clearly equation (2.13) is equivalent with (2.11). To (2.11) the iterative process

$$u_{k+1} = (1 + \omega_k D)u_k + \omega_k f \qquad k = 0, 1, 2, \ldots, \tag{2.14}$$

may be associated. The scheme corresponding to (2.13) becomes

$$(1 - D_1)u_{k+1} = (1 - D_1 + \omega_k D)u_k + \omega_k f, \quad k = 0, 1, 2, \ldots, \tag{2.15}$$

which resembles Gauss-Seidel's method for elliptic difference equations.

In [6] it is proved that the eigenvalues λ of the non-symmetric operator $L = -(1 - D_1)^{-1}D$ are defined by the equation

$$R = 2P^{\frac{1}{2}}Q^{\frac{1}{2}} \cos n\xi, \qquad n = 1, \ldots, \frac{\pi}{\xi} - 1, \tag{2.16}$$

where

$$\left.\begin{aligned} P &= L_2 + 2L_1(1 - q\lambda)^{\frac{1}{2}}\cos m\eta, \\ Q &= L_2(1 - p\lambda) + 2L_1(1 - q\lambda)^{\frac{1}{2}}\cos m\eta, \\ R &= -L_4 - \lambda - 2L_3(1 - q\lambda)^{\frac{1}{2}}\cos m\eta \end{aligned}\right\} m = 1, \ldots, \frac{\pi}{\eta} - 1,$$

and that the eigenfunctions are of the form

$$P^{-\frac{1}{2}j}Q^{\frac{1}{2}j}(1 - q\lambda)^{\frac{1}{2}l} \sin nj\xi \sin ml\eta, \quad m = 1, \ldots, \frac{\pi}{\eta} - 1, \quad n = 1, \ldots, \frac{\pi}{\xi} - 1. \tag{2.17}$$

In those cases that equation (2.16) reduces to an equation of the form

$$(1 - q\lambda) - A(1 - q\lambda)^{\frac{1}{2}} + B = 0,$$

where A and B are certain functions of p, q, n and m, it is easily verified that

$$0 \leq \lambda \leq \frac{1}{q}$$

whenever A and B satisfy the condition

$$0 \leq B \leq \frac{1}{4}A^2 \tag{2.18}$$

for all n and m.

Here again, confining ourselves to square grids, we find a five-point formula (+) when $L_1 = 0$ and a five-point formula (×) when $L_2 = L_3 = 0$, which will be discussed into details in the next section. We should bear in mind, however, that both formulas are non-symmetric.

3. Eigenvalues and eigenfunctions of the non-symmetric five-point formulas (+) and (×)

The two non-symmetric formulas (+) and (×), mentioned in the preceding section, will be employed to solve the model problem II in [6], i.e. Dirichlet's problem for the potential equation in a square of side π. We shall come across some unexpected peculiarities of these formulas.

Taking $\rho = 1$, and writing $\xi = \eta = h$ and $N = \pi/h$, so that there are $(N - 1)^2$ interior lattice points, we find that equation (2.16) takes the form

$$(3.1)\quad (1 - q\lambda_+) - 2qL_2(\cos nh + \cos mh)(1 - q\lambda_+)^{\frac{1}{2}} - L_4 q - 1 = 0,$$

if the (+) formula is considered, and

$$(3.2)\quad (1 - q\lambda_\times) - 4qL_1 \cos nh \cos mh\,(1 - q\lambda)^{\frac{1}{2}} - L_4 q - 1 = 0,$$

in the case of the (×) formula; here $n, m = 1, \ldots, N - 1$. In order to satisfy the condition (2.18) we choose $q = -1/L_4$, a consequence of which is that the ranges of the eigenvalues λ_+ and $\lambda_\times$ are restricted to

$$0 \leq \lambda_+ \leq \frac{1}{q} = \frac{4}{h^2}$$

and

$$0 \leq \lambda_\times \leq \frac{1}{q} = \frac{2}{h^2}$$

respectively. Equations (3.1) and (3.2) then reduce to

$$(1 - q\lambda_+(n, m))^{\frac{1}{2}} = 2qL_2(\cos nh + \cos mh)$$

and

$$(1 - q\lambda_\times(n, m))^{\frac{1}{2}} = 4qL_1(\cos nh \cos mh),$$

where $n, m = 1, \ldots, N - 1$, from which easily follows

$$(3.3)\quad \lambda_+(n, m) = \frac{4}{h^2}(1 - \tfrac{1}{2}(\cos nh + \cos mh)^2)$$

and

$$(3.4)\quad \lambda_\times(n, m) = \frac{2}{h^2}(1 - \cos^2 nh \cos^2 mh).$$

Formula (2.17) supplies us the corresponding eigenfunctions

$$(3.5)\qquad e_+(n, m) = (1 - q\lambda_+(n, m))^{\frac{j+1}{2}} \sin njh \sin mlh =$$
$$= (\tfrac{1}{2}(\cos nh + \cos mh))^{j+1} \sin njh \sin mlh,$$

$$(3.6)\qquad e_\times(n, m) = (1 - q\lambda_\times(n, m))^{\frac{1}{2}} \sin njh \sin mlh =$$
$$= (\cos nh \cos mh)^{1} \sin njh \sin mlh,$$

where $n, m = 1, \ldots, N - 1$.

Unfortunately, amongst the functions $e_+(n, m)$ given by (3.5), there are always some that are equal to zero, and therefore no eigenfunctions. When N is even, the same holds for the set of functions $e_\times(n, m)$. To be more precise, there are $2N - 3$ pairs (n, m) for which in those cases $e_+(n, m)$ or $e_\times(n, m)$ equal zero. Only if N is odd, the set of functions $e_\times(n, m)$ forms a complete set of eigenfunctions. We shall therefore start by making some remarks about the eigenvalues and eigenfunctions for this case.

Eigenvalues of the Gauss-Seidel (×) formula.

In vieuw of (3.4) we may conclude that

$$\lambda(n, m) = \lambda(n, N - m) = \lambda(N - n, m) = \lambda(N - n, N - m) =$$
$$\lambda(m, n) = \lambda(N - m, n) = \lambda(m, N - n) = \lambda(N - m, N - n).$$

From this it is seen that the eigenvalues $\lambda(k, k')$ for which $k = k'$ appear 4 times, and that the remaining eigenvalues are 8- fold. In order to take along all the eigenvalues it is obviously sufficient to consider only those pairs (n, m) that satisfy $1 \leq n \leq (N - 1)/2$ and $1 \leq m \leq n$. In fig. 3.1 the corresponding area is shaded.

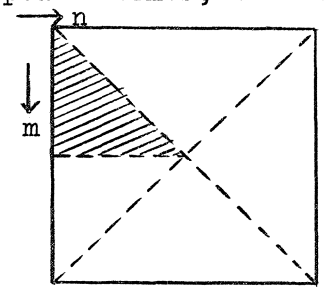

Fig. 3.1

We wish to get an impression how the eigenvalues are distributed in the interval $[a,b]$, where a is the smallest eigenvalue, and $b = 1/q$. To this end we examine the function

$$\phi(x, y) = 1 - \cos^2 x \cos^2 y$$

for $0 \leq x \leq \pi/2$, $0 \leq y \leq \pi/2$. The variable x corresponds with nh, y with mh. Figure 3.2, in which the function $\phi(x, y)$ is sketched, gives us the rough impression that considerably more eigenvalues are situated in the end of the interval $[a,b]$ than in the beginning, that there is an accumulation of eigenvalues in the last part of $[a,b]$. Indeed, without too many difficulties one can calculate that about 60% of the eigenvalues lie in the subinterval $[\frac{4}{5}b, b]$.

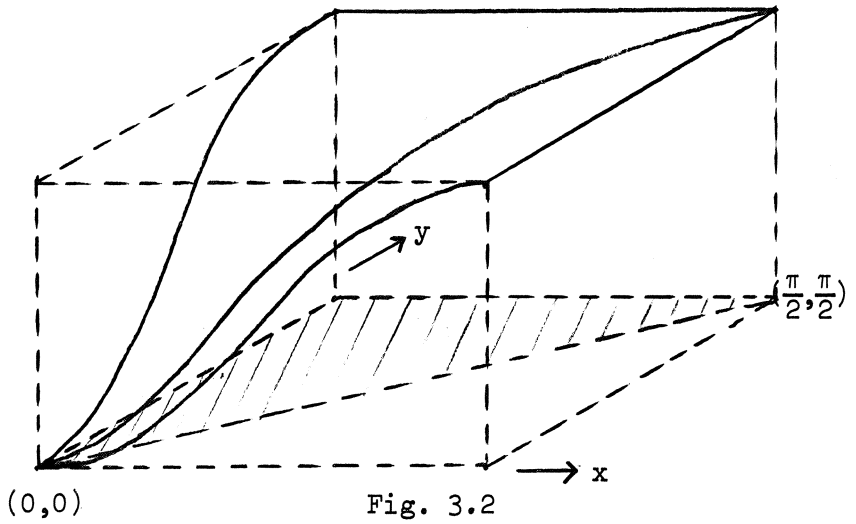

Fig. 3.2

Finally, we note that from (3.3) it follows that the smallest eigenvalue $\lambda(1,1) = \lambda_1$ is approximately 4. The largest eigenvalue $\sigma(L)$ is about $\frac{2}{h^2}$ $(= \frac{1}{q})$, so that the P- condition number becomes

$$\frac{\sigma(L)}{\lambda_1} = \frac{1}{2h^2}.$$

The asymptotic rate of convergence is

$$R(\infty) = 2\sqrt{2}.h$$

Eigenfunctions

We start the discussion of the eigenfunctions by proving the following theorem.

Theorem The eigenfunctions e(n, m), n, m = 1, ..., N - 1, which are given by (3.6), form a complete set, if N is odd.

Proof Eigenfunctions belonging to different eigenvalues are independent. None of the functions e(n, m) of system (3.6) vanish everywhere, because N is odd. If we succeed in showing that the eigenspaces belonging to the 8-fold eigenvalues are 8-dimensional, and those belonging to 4-fold eigenvalues 4-dimensional, then we may consider this theorem to be proved.

We first observe that the Eucledian inner-product

$$(3.7)\quad \langle e(n, m), e(n', m') \rangle = \sum_{j,l=1}^{N-1} (\cos nh \cos mh)^l \sin njh \sin mlh.$$

$$.(\cos n'h \cos m'h)^l \sin n'jh \sin m'lh =$$

$$\sum_{l=1}^{N-1} (\cos nh \cos n'h \cos mh \cos m'h)^l \sin mlh \sin m'lh \sum_{j=1}^{N-1} \sin njh \sin n'jh = 0$$

whenever $n \neq n'$. Here we have made use of the relation

$$\sum_{j=1}^{N-1} \sin nj\frac{\pi}{N} \sin mj\frac{\pi}{N} = 0,$$

if n and m are unequal integers.

Let us consider, taking $m \neq n$, the eight functions

$$e(n, m),\ e(N - n, m),\ e(N - m, n),\ e(m, n) \text{ and}$$

$$e(N - n, N - m),\ e(n, N - m),\ e(m, N - n),\ e(N - m, N - n).$$

Clearly, because of (3.7) the first four functions are mutually perpendicular, and the remaining four functions mutually as well. The question we now ask ourselves is if it is possible to write e(n, m) as a linear combination of the last four eigenfunctions. Since according to (3.7) e(n, m) is perpendicular to all except e(n, N - m), in that case e(n, m) would be a scalar multiple of e(n, N - m). But, as

$$e(n, m) = (\cos nh \cos mh)^l \sin njh \sin mlh$$

and

$$e(n, N - m) = (-1)^{l} (\cos nh \cos mh)^{l} \sin njh \sin (N - m)lh,$$

there follows a contradiction.

Proceeding in this way we can see that the eight mentioned functions are mutually independent. In a same manner one can show that the eigenspaces belonging to 4-fold eigenvalues are 4-dimensional.

The following remark has importance. Although the set (3.6) of eigenfunctions is independent and complete, this does not tell us anything about its conditioning. In fact, the set (3.6) is very ill-conditioned.

We shall give an illustration of this fact, choosing N = 19, the same example for which the majority of the numerical experiments is performed (see section 4). For a few values of n, m, n' and m' the quantity

$$(3.8) \qquad H(n, m; n', m') = \frac{\langle e(n, m), e(n', m') \rangle}{||e(n, m)||.||e(n', m')||}$$

has been calculated, the results of which are listed in table 1. The symbol $||\ \ ||$ denotes the norm corresponding to the Eucledian inner-product. We know that $H(n, m; n', m') = 0$ when $n \neq n'$, so that it is sufficient to examine $H(n, m; n, m')$. The lesser $H(n, m; n, m')$ differs from 1, the smaller the "angle" between $e(n, m)$ and $e(n, m')$ will be.

$n = n' = 1$

m \ m'	1	2	3	4	5	6	7	8	9
1									
2	.3053								
3	.1362	.5912							
4	.1110	.3632	.8007						
5	.0955	.2737	.5976	.9007					
6	.0855	.2330	.4774	.7547	.9441				
7	.0775	.2029	.4019	.6390	.8440	.9657			
8	.0711	.1821	.3533	.5596	.7558	.9025	.9806		
9	.0674	.1707	.3279	.5178	.7048	.8573	.9538	.9937	

$n = n' = 7$

m \ m'	7	8	9
7			
8	.9968		
9	.9923	.9991	

Table 3.1

$n = m' = 9$

m \ m'	8	9
8		
9	.9997	

We have already noticed that the pairs of numbers (n, m) for which n or m are close to $(N - 1)/2$, are exactly those that correspond with large eigenvalues. Looking at table 3.1 with this knowledge, we can easily see that the eigenfunctions belonging to large eigenvalues are those that make the system of eigenfunctions ill-conditioned. We shall come back to this subject when the results obtained by this non-symmetric five-point difference scheme applied to the Dirichlet problem for the square of side π will be discussed (section 4).

Rectangular regions.

This section will be terminated by making some remarks about the Dirichlet problem for a rectangle. Let the sides of the rectangle be π and $\kappa\pi$ $(0 < \kappa \leq 1)$, and assume that a grid consisting of $N \times M$ squares of side $h = \pi/N$ fits exactly into the rectangle. Clearly, $\frac{\pi\kappa}{M} = h$. As in the case of the square, it is easily shown that one must also choose q to be equal to $-1/L_4$. The eigenvalues and eigenfunctions are given by

$$\lambda(n, m) = \frac{2}{h^2}\left(1 - \cos^2 nh \cos^2 \frac{mh}{\kappa}\right)$$

and

$$e(n, m) = \left(\cos nh \cos \frac{mh}{\kappa}\right)^1 \sin njh \sin \frac{mlh}{\kappa}$$

respectively. The smallest eigenvalue

$$\lambda(1, 1) = \frac{2}{h^2}\left(1 - \cos^2 h \cos^2 \frac{h}{\kappa}\right)$$

is larger than the one in the case of the square of side π. Very roughly we may make the estimate

$$\lambda(1, 1) \sim 2\left(1 + \frac{1}{\kappa^2}\right).$$

As one should have expected, the smallest eigenvalue is strongly dependent on the shape of the region. As upperbound for the eigenvalues one may again take $2/h^2$.

Remark.

The iteration scheme with $\gamma = 2$ may be interpreted as the iteration scheme with $\gamma = 1$ applied to a square which is rotated over 45° (see figures 3.3 and 3.4).

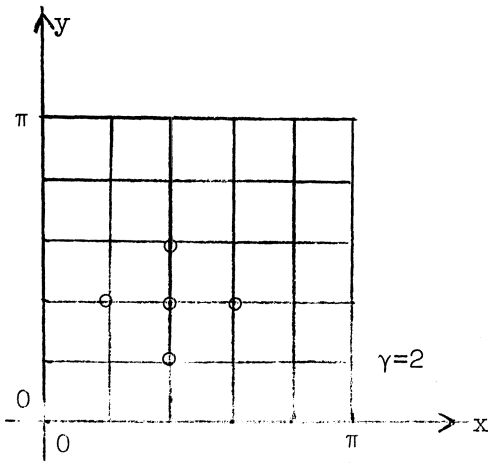

Fig. 3.3

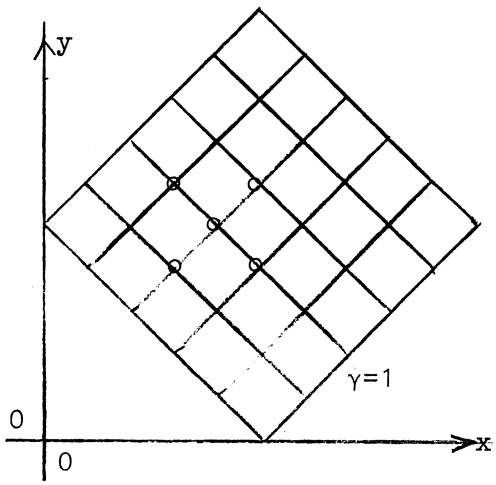

Fig. 3.4

In order to compare results of the schemes with $\gamma = 1$ and $\gamma = 2$, one has to choose $N(\gamma)$ such that $N(2) = N(1)/\sqrt{2}$. Then the difference in the results of the two schemes is only caused by the difference in orientation of the square with respect to the x and y axis.

Furthermore, it is possible to calculate the net function u_k in half the net points when using the (×) formula, for example the points (n, m) of which n and m are both odd or both even. Then the solutions on the two sets of net points are independent.

4. Numerical Results

Numerical calculations for the Dirichlet problem for Laplace's equation in a square of side π, the model problems I and II in [6], were performed on the Electrologica X8 computer. A square grid was chosen, the mesh size of which was $h = \pi/N$. Obviously there are $(N - 1) \times (N - 1)$ interior nodes.

The number

$$R^{*}(K) = -\frac{1}{K} \ln \frac{||Du_K + f||}{||Du_0 + f||}, \tag{4.1}$$

which was already introduced in [5], will serve as an estimate for the rate of convergence after K iterations of the iterative process, in order to enable us to compare the results of the symmetric and non-symmetric schemes. We have

$$R^{*}(K) = -\frac{1}{K} \ln \frac{||Du + Dv_K + f||}{||Du + Dv_0 + f||} = \tag{4.2}$$

$$= -\frac{1}{K} \ln \frac{||P_K(L)\, Dv_0||}{||Dv_0||} \geq R(K).$$

As in [4] $P_K(L)$ is the Chebyshev-polynomial $C_K(a, b, L)$, or a Chebyshev-polynomial operator multiplied by one or more elimination operators $E_{K^{*}}(L)$. In most of the calculations the inhomogenious term f was chosen to be equal to 0.

In [4] and [7] the choice of the degree of the operator $E_{K^{*}_{n,m}}$, eliminating the eigenvalue $\lambda(n, m)$, was discussed. The first way to choose $K^{*}_{n,m}$ is

$$K^{*}_{n,m} = \text{entier}\left(\frac{1}{4}\pi\sqrt{\frac{b}{\lambda(n,m)}}\right) + 1, \tag{4.3}$$

in which case $E_{K^{*}_{n,m}}$ is stable.

One obtains an optimal elimination operator, if one chooses $K^*_{n,m}$ satisfying the equations

$$(4.4) \qquad \begin{cases} 2\sqrt{\frac{a}{b}} + \frac{d}{dx} \ln \sigma\left(E_x(L)\right) = 0, \\ K^*_{n,m} = \text{entier } (x + \tfrac{1}{2}), \end{cases}$$

which are equations (3.19) on page 102 of [7]. The tables given in [7] are of great use.

Special numerical features of the Gauss-Seidel five-point formula (×)

We have already noted that the asymptotic rate of convergence of the Gauss-Seidel five-point formula (×) is $2\sqrt{2}h$, which is twice as much as the corresponding Jacobi scheme. We shall now discuss some special numerical features of this formula.

Richardson's method applied to the five-point formula (×) gave in the case that N = 11 and the degree of the used Chebyshev-polynomial is 27, the following results

initial approximation	$R^*(K)$	$R(K) \geq 2\sqrt{\frac{a}{b}} - \frac{\ln 2}{K} - \frac{\ln\alpha}{K}$
$(x-2)(y-2)\sin x \sin y$	.07	
$\sum_{n,m} e(n,m)$	.73	
$(x-2)(y-2)\sin x \sin y$ (with preconditioning phase of degree 5)	.51	$.78 - \frac{.693}{K} - \frac{\ln\alpha}{K} =$ $= .75 - \frac{\ln\alpha}{K}$

Table 4.1

We see that when a sum of eigenfunctions is chosen as the initial approximation, the expected rate of convergence is actually reached, whereas starting with a rather arbitrary function as $(x-2)(y-2)\sin x \sin y$ leads to a rate of convergence considerably below the expected value. It is at this point that the ill-conditioning of the set (3.6) of eigenfunctions re-enters into our considerations.

A consequence of the ill-conditioning is that the coefficients in the expansion for arbitrary v_0,

$$v_0 = \sum_i \alpha_i e_i \ , \tag{4.5}$$

can be large, so that the quantity $\ln\alpha$ appearing in (2.7) can also be large. During the reduction by means of a Chebyshev-polynomial operator all eigenfunctions are equally damped. Therefore, an extra damping of those eigenfunctions for which α_i is large, is desirable, in order to surpress the influence of the bad conditioning of the set of eigenfunctions. We already know (section 3) that the eigenfunctions belonging to large eigenvalues are those that make small angles with each other, so that we may expect the α_i corresponding with these eigenfunctions to be large. Furthermore, we also know that in the interval $[\frac{4}{5}\,b,\ b]$ 60% of all the eigenfunctions are situated. An extra damping of the eigenfunctions dominating in (4.5) may be effectuated by an extra Chebyshev reduction on the interval $[\frac{4}{5}\,b,\ b]$. The third result in table 4.1 shows the effect of this procedure. Of course the choice $\frac{4}{5}\,b$ is more or less arbitrary, but it turned out to work well. The best way to look at this extra reduction is to see it as a means to prepare the initial approximation in such a way that over-all reduction can be performed successfully. We shall call this <u>pre-conditioning</u>. When this is combined with the elimination of the first two eigenfunctions, the polynomial operator $P_K(L)$ takes the form

$$P_K(L) = E_{K_1*}(L)\ E_{K_2*}(L)\ C_{K_r}(a, b, L) C_{K_c}(\tfrac{4}{5}\,b, b, L),$$

where K_r denotes the degree of the Chebyshev-polynomial used for the over-all reduction, K_c that of the polynomial used for the preconditioning.

There is another phenomenon that points to the strong domination of the late eigenfunctions. After having applied Richardson's method by means of a Chebyshev-polynomial of degree 32 in the case N = 19, the function u_{32} showed globally the tendency pictured in figure 4.1

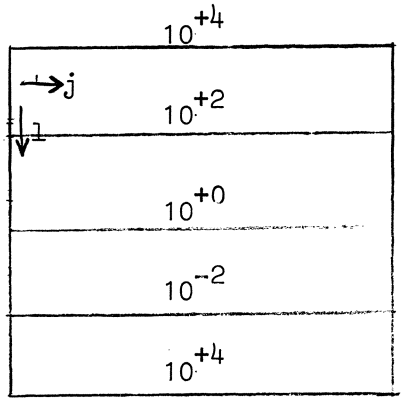

Fig. 4.1

Along horizontal lines the function remains of the same order, but along vertical lines it changes considerably. The eigenfunctions are of the form

(3.6) $e(n, m) = (1 - q\lambda)^{1/2} \sin njh \sin mlh.$

Remembering that $\lambda < \frac{1}{q} = b$ for all λ, it is clear that especially the eigenfunctions belonging to large eigenvalues show strongly the sloping effect indicated in figure 4.1. And, as we have seen, it are exactly these eigenfunctions that remain strongly present after an over-all reduction. This phenomenon gives us an experimental way to determine of which degree the Chebyshev-polynomial for the pre-conditioning can be chosen, namely such that this sloping effect has vanished entirely. From a practical point of view this suggests the performance of the over-all reduction first, and the pre-conditioning afterwards.

It is obvious that as a consequence of the bad conditioning of the eigenfunctions the average rate of convergence after K iterations reduces below its asymptotic value by a considerable amount. From (2.7) it follows that

$$R(K_r, K_1^*, K_2^*) \geq 2\sqrt{\frac{a}{b}} - \frac{c(K_1^*, K_2^*)}{K_r + K_1^* + K_2^*},$$

where K_1^* and K_2^* are the degrees of the operators eliminating the first and second eigenfunctions respectively.
Introducing the pre-conditioning phase, the constant c will also depend on the degree K_c of the operator $C_{K_c}(\frac{4}{5}b, b, L)$. If we set $K = K_c + K_r + K_1^* + K_2^*$, then we may write

$$R(K_c, K_r, K_1^*, K_2^*) \geq 2\sqrt{\frac{a}{b}} - \frac{c(K_c, K_1^*, K_2^*)}{K}.$$

K_1^* and K_2^* are determined from (4.3) or (4.4), the results of which are listed in table (4.2).

$\lambda(n,m)$ / a	4	9.5	$2\sqrt{\frac{a}{b}} - c/K$
4			.47 - c/K
9.5	4		.72 - c/K
14.5	3	4	1.00 - c/K
	4	3	

Table 4.2

Comparison with other non-stationary difference schemes

The Gauss-Seidel (×) formula has been compared with several other schemes. Richardson's method, and the elimination method, supplemented by an extra pre-conditioning phase, for this scheme has been compared with Richardson's method and the elimination method for the Jacobi (×) formula, and with Richardson's method for the Jacobi (+) formula. Comparison with stationary methods will be given in section 5. All calculations were done for the case N = 19, that is 18 × 18 interior nodes.

In table 4.5 the results are listed. The symbols in that table have the following meaning:

stv - startingvector, initial approximation, see table 4.3;

K - total number of iterations performed;

K_r - degree of over-all reduction polynomial operator;

K_c - degree of the pre-conditioning operator;

K_1^*, K_2^* - degrees of the operators eliminating the first and second eigenfunctions respectively;

$[a, b]$ denotes the interval over which over-all reduction is performed;

γ - defined in formula (2.10);

$R^*(K)$ - defined by (4.1);

(i) denotes that the eigenvalue of which the corresponding eigenfunction is to be eliminated is found by the process

$$(4.6)\qquad \begin{cases} q_{k+1} = \dfrac{||u_{k+1} - u_k||}{||u_k - u_{k-1}||} \\ \lambda = \dfrac{1}{\omega_{k-1}} - \dfrac{q_{k+1}}{\omega_k} \end{cases}$$

where ω_k are the relaxation parameters (see [4]);

(ii) denotes that the analytical expression for the eigenvalue, which is known for theis test-case, was used;

R(K) - the theoretical average rate of convergence;

$R(\infty)$ - the asymptotic rate of convergence ($K_r \to \infty$).

For the Jacobi (×) scheme we give in table 4.4 the values for K_1^* and K_2^*, which are obtained in an analogous manner as was done for table 4.2. See also [6].

stv	initial approximation
4	(x - 2)(y - 2) sin x sin y
5	(x - 1)(y - 1)(x - 2)(y - 2) sin x sin y
8	0 on the boundary, 1 in the interior of the square

Table 4.3

a \ λ(n,m)	2	5	$2\sqrt{\frac{a}{b}} - \frac{2K^*\sqrt{\frac{a}{b}} + \ln \sigma(E_k^*) + \ln 2}{K_r + K^*}$
2			.23 - 6.93/K
5	8		.37 - 3.24/K
8	5	7	.47 - 5.68/K
	8	5	.47 - 5.97/K

Table 4.4

In table 4.6 some results for a rectangular region is given. The sides of the rectangle are π and (9/39)π respectively, and is covered by 39 × 9 squares.

Method	Stv	K	K_r	K_c	K_1^*	K_2^*	a	b	γ	$R^*(K)$	R(K)	R(∞)
	4	51	32	19			3.9	73.2	1	.33		.47
	5	51	32	19			3.9	73.2	1	.37		.47
(i)	4	51	29	18	4		9.5	73.2	1	.49		.72
(i)	5	51	29	18	4		9.5	73.2	1	.52		.72
(ii)	5	51	29	18	4		9.5	73.2	1	.53		.72
(i)	8	51	29	18	4		9.5	73.2	1	.49		.72
Gauss-Seidel (×) (ii)	8	51	29	18	4		9.5	73.2	1	.49		.72
(i)	4	51	29	18	4		7.0	73.2	1	.44		.62
(ii)	4	51	29	18	4		7.0	73.2	1	.44		.62
	4	51	27	17	3	4	14.5	73.2	1	.53		1.00
	4	51	27	17	4	3	14.5	73.2	1	.52		1.00
Jacobi (×)	4	51	51				2	146.31	1	.23	.22	.23
(i)	4	53	45		8		5	146.31	1	.29	.31	.37
(ii)	4	53	45		8		5	146.31	1	.29	.31	.37
(ii)	4	51	39		5	7	8	146.31	1	.33	.35	.47
Jacobi (+)	4	51	51				2	292.62	2	.16	.15	.16

Table 4.5: results for the square 19 19 × 19

Gauss-Seidel(×)	4	51	32	19			37.8	308	1	.53		.70
Jacobi(×)	4	51	51				18.9	616	1	.35	.34	.35
Jacobi(+)	4	51	51				18.9	1232	2	.25	.24	.25

Table 4.6: results for the rectangle 39 × 9

Some remarks for the case of even N

In section 3 we restricted ourselves to the Gauss-Seidel (×) formula for odd N. We shall now make some remarks and give some results for the model problem when N is even. Let $n_o = N/2$. Of the set (3.6) those $e_\times(n, m)$ with $n = n_o$ or $m = n_o$ then equal zero, and are therefore no eigenfunctions. It can easily be proved that adding the (2N - 1)-1 functions

$$E(n, m) = \sin njh \sin mlh, \quad n = n_o \text{ or } m = n_o$$

to the set of non-zero functions of (3.6) provides us a base for the space of net functions. For the case N = 20 we have compared experimentally the effect of applying once the operator $1 - \omega L$, where $\omega = [a - (a + b)/2]^{-1}$, to some functions E(n, m) and to some eigenfunctions, the results of which are given in table 4.7.

function	$R^*(1)$
E(1,10)	.091
E(10,1)	.095
E(10,10)	.095
E(10,19)	.095
E(19,10)	.091
$e_*(1,9)$	.146
$e_*(9,9)$	.096
$e_*(19,9)$	.146

Table 4.7

N	K	K_r	K_c	a	b	γ	$R^*(K)$	$R(\infty)$
18	51	32	19	3.9	65.7	1	.36	.49
19	51	32	19	3.9	73.2	1	.33	.47
20	51	32	19	3.9	81.0	1	.30	.44

Table 4.8

We see that the additional functions E(n, m) are not damped much less than the eigenfunctions $e_*(n, m)$. In table 4.8 the results of Richardson's method, supplemented by a pre-conditioning phase, for the Gauss-Seidel (×) formula are compared for the cases N = 18, 19 and 20; stv = 4. Both tables suggest that it is possible to use the Gauss-Seidel (×) scheme for even N as fruitfully as for odd N.

5. A comparison with stationary processes

This section reviews and extends three basic iterative methods of stationary type for solving the matrix equation $Lu = f$ (see Varga [8], chapter 3).

The method of Jacobi

Following Varga [8], p. 57 we express the matrix L as the matrix sum

$$L = C - E - F, \tag{5.1}$$

where C is a diagonal matrix whose entries are the diagonal elements of L, and E and F are respectively strictly lower and upper triangular matrices, whose entries are the negatives of the entries of L respectively below and above the main diagonal of L.

The method of Jacobi (1845) or the method of simultaneous displacements is defined by the formula

$$u_{k+1} = (1 - C^{-1}L)u_k + C^{-1}f, \quad k = 0, 1, 2, \ldots . \tag{5.2}$$

We apply this method to the matrix problem defined in section 3 of reference [6] (the model problem I) . We have $L = -D$, so that

$$\begin{aligned} C &= -L_4, \\ E &= L_2X_- + L_3Y_- + L_1(X_+ + X_-)Y_-, \\ F &= L_2X_+ + L_3Y_+ + L_1(X_+ + X_-)Y_+. \end{aligned} \tag{5.3}$$

From (5.3) we may deduce that (5.2) is equivalent to the iterative process defined by (2.2) with

$$\omega_k = -L_4^{-1} = \frac{\xi^2}{2\gamma} \tag{5.4}$$

for all k. The rate of convergence of this process is given by

$$(5.5) \qquad R = -\frac{\ln \|(1 - C^{-1}L)^k\|}{K} = -\ln \sigma(1 + \frac{\xi^2}{2\gamma} D)$$

$$= -\ln[\max(1 - \frac{\xi^2\lambda_1}{2\gamma}, \frac{\xi^2\sigma(-D)}{2\gamma} - 1)].$$

From [6], p. 8 we may deduce that

$$1 - \frac{\xi^2\lambda_1}{2\gamma} \geq \frac{\xi^2\sigma(-D)}{2\gamma} - 1,$$

hence

$$(5.5') \qquad R = R(\gamma) = -\ln(1 - \frac{\lambda_1}{2\gamma}\xi^2) \sim \frac{\lambda_1}{2\gamma}\xi^2 \sim \frac{1}{\gamma}\xi^2.$$

As one may again expect the five-point formula ($\gamma = 1$) is to be preferred over the five-point (+) formula ($\gamma = 2$).
At the end of this section some results of numerical experiments are given.

The iteration process (5.2) - (5.4) can be accelerated by the so-called gradient method (see Forsythe and Wasow [1], p. 225). Applying the gradient method to the Jacobi process described above we obtain an iterative method of type (2.2) with

$$(5.6) \qquad \omega_k = \frac{2}{\sigma(-D) + \lambda_1} \sim \frac{2}{\sigma(-D)}$$

for all k. In fact, we have a repeated Richardson process with K = 1. The rate of convergence is given by

$$(5.7) \qquad R'(\gamma) \sim -\ln(1 - \frac{2\lambda_1}{\sigma(-D)}) \sim 2\frac{\lambda_1}{\sigma(-D)} \sim \begin{cases} \xi^2 & \text{for } 1 \leq \gamma \leq \frac{3}{2} \\ \frac{1}{2(\gamma - 1)}\xi^2 & \text{for } \frac{3}{2} \leq \gamma \leq 2. \end{cases}$$

In the following figure we have illustrated the behaviour of the rate of convergence as a function of γ.

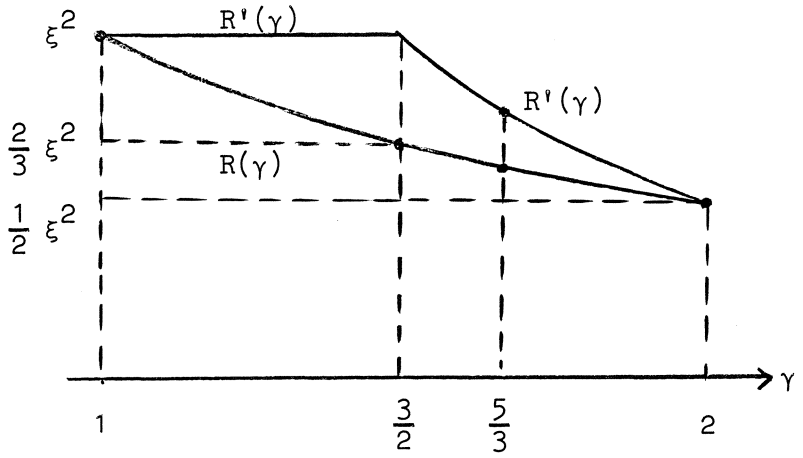

Fig. 5.1

This figure indicates that the gradient method accelerates Jacobi's method if $1 < \gamma < 2$. Thus the nine-point formula which arises for $\gamma = 5/3$ can be accelerated by the gradient method. In fact, we have

$$R(\tfrac{5}{3}) \sim \tfrac{3}{5}\xi^2 \quad , \quad R'(\tfrac{5}{3}) \sim \tfrac{3}{4}\xi^2. \tag{5.8}$$

This behaviour is in agreement with numerical results given in table 5.1.

The method of Gauss-Seidel

The method of Gauss-Seidel or the method of successive displacements is defined by the formula

$$(1 - C^{-1}E)u_{k+1} = (1 - C^{-1}E - C^{-1}L)u_k + C^{-1}f, \; k = 0, 1, 2, \ldots . \tag{5.9}$$

This method was mentioned by Seidel (1874) and used by Gauss. Applying the method to model problem I (see [6], section 3) we obtain the formula

$$(1 + L_4^{-1}E)u_{k+1} = (1 + L_4^{-1}E - L_4^{-1}D)u_k - L_4^{-1}f \tag{5.9'}$$

where E is defined by formula (5.3).

This special case is called Liebmann's method. Formula (5.9') is identical to formula (2.15) with

$$p = q = -L_4^{-1} = \frac{\xi^2}{2\gamma}, \quad \omega_k = -L_4^{-1} = \frac{\xi^2}{2\gamma} \tag{5.10}$$

for all k.

In order to find the rate of convergence of the Liebmann method we have to know the behaviour of

$$\| \left[1 + \omega(1 - D_1)^{-1} D \right]^K \|, \quad \omega = \frac{\xi^2}{2\gamma}$$

as a function of K. From van der Houwen [7], p. 30 we take the formula

$$\| A^K \| \sim \nu K^{p-1} \left[\sigma(A) \right]^{K-p+1} \text{ as } K \to \infty, \tag{5.11}$$

where p is the largest order of all diagonal submatrices J_r of the Jordan normal form J of A with $\sigma(J_r) = \sigma(A)$ and where ν is a constant related to the conditioning of the matrix A and the conditioning of the eigenfunctions of A. Applying (5.11) to the matrix $A = 1 + \omega(1 - D_1)^{-1} D$ we obtain for large values of K

$$R(K) \sim -\ln \sigma(1 + \omega(1 - D_1)^{-1} D) - \frac{c}{K} \tag{5.12}$$

where

$$c = \ln \nu + (p - 1) (\ln K - \ln \sigma(1 + \omega(1 - D_1)^{-1} D)). \tag{5.13}$$

Let us consider the five-point formula (+) and (×). It was shown in section 3 that the eigenvalues λ of $-(1 - D_1)^{-1} D$ approximately vary between 4 and $4\,\xi^{-2}$ and between 4 and $2\,\xi^{-2}$ respectively. From this it follows that the rates of convergence averaged over K iterations are given by respectively

$$R_+(K) \sim \xi^2 - \frac{c_+}{K}, \quad R_\times(K) \sim 2\,\xi^2 - \frac{c_\times}{K}. \tag{5.14}$$

Comparing these results with figure 5.1 we see that asymtotically Liebmann's method converges twice as fast as Jacobi's method. This conclusion was affirmed by numerical experiments (see the table at the end of this section).

The constants c_+ and $c_\times$ corresponding to the initial approximation used were determined experimentally by considering the rate of convergence as a function of $1/K$. For large values of K this function behaves as a linear function whose slope is equal to c_+ or $c_\times$.

In the same manner as the Jacobi method (5.2) - (5.4) was accelerated by choosing more appropriate values for ω_k, we can accelerate, at least asymptotically, Liebmann's method by choosing

$$\omega_k = \frac{2}{\lambda_1 + \sigma(-(1 - D_1)^{-1}D)} \sim \frac{2}{\sigma(-(1 - D_1)^{-1}D)} \tag{5.15}$$

for all k. The average rate of convergence follows from (5.12) where (5.15) is substituted for ω. We find

$$R'_+(K) \sim 2\xi^2 - \frac{c'_+}{K}, \quad R'_\times(K) \sim 4\xi^2 - \frac{c'_\times}{K}, \tag{5.16}$$

which is for large K twice the value of the rate of convergence of Liebmann's method. However, the constants c'_+ or $c'_\times$ will be large since the matrix $A = 1 + \omega(1 - D_1)^{-1}D$ and its eigenfunctions are very ill-conditioned (see the discussion in section 3). A number of experiments confirm this theoretical prediction. Some kind of preconditioning of the initial approximation as was proposed in the preceding section will improve the average rate of convergence of the method considerably (see table 5.1).

Method of successive overrelaxation.

The method of successive overrelaxation (SOR method) is defined by the formula

$$(1 - \Omega C^{-1}E)u_{k+1} = (1 - \Omega C^{-1}E - \Omega C^{-1}L)u_k + \Omega C^{-1}f, \tag{5.17}$$

$$k = 0, 1, 2, \ldots .$$

The parameter Ω is called the relaxation factor. The problem is to find its optimal value, i.e. the value of Ω for which (5.17) converges as fast as possible. For the model problem I, this optimalization problem was solved by Frankel [2] in 1950.

In 1953 Young [9] solved the problem for a wider class of problems in which the matrix L possesses the so-called property A. However, neither Frankel nor Young, and, as far as the authors know, no one else has applied the theory to the five-point approximation (×) of the Laplace operator. In this section we shall give a discussion of the SOR method for the five-point formula (×).

Applying the SOR method (5.17) to model problem I we obtain the formula

$$(5.17') \qquad (1 + \Omega L_4^{-1} E)u_{k+1} = (1 + \Omega L_4^{-1} E - \Omega L_4^{-1} D)U_k - \Omega L_4^{-1} f,$$

where E is defined by (5.3). This formula arises from (2.15) for

$$(5.18) \qquad p = q = -\Omega L_4^{-1} = \Omega \frac{\xi^2}{2\gamma}, \quad \omega_k = -\Omega L_4^{-1} = \Omega \frac{\xi^2}{2\gamma},$$

$$k = 0, 1, 2, \ldots .$$

It is easily verified that the five-point formula (+) and (×) have property A and that the order in which the equations are solved by using (5.17') is consistent with the tridiagonal representations corresponding to these five-point formula (see Forsythe and Wasow [1], p. 244). It then follows from the theory of Young (see [1], p. 253) that the optimal value of Ω is given by

$$(5.19) \qquad \Omega_{opt} = \frac{2}{1 + \sqrt{1 - \sigma^2(1 + \frac{\xi^2}{2\gamma}D)}} \sim 2(1 - \sqrt{\frac{2}{\gamma}}\,\xi), \quad \gamma = 1, 2$$

and the rate of convergence is given by

$$(5.20) \qquad R(K) \sim -\ln(\Omega_{opt} - 1) - \frac{c}{K},$$

where c is a constant which is determined by a formula analogous to (5.13). From (5.19) and (5.20) we find

$$(5.21) \qquad R_+(K) \sim 2\xi - \frac{c_+}{K}, \quad R_\times(K) \sim 2\sqrt{2}\,\xi - \frac{c_\times}{K}.$$

As we already have mentioned, the values of the constants c_+ and $c_\times$ depend strongly on the conditioning of the eigenfunctions of the iterative operator A. In the cases considered here we have $A = 1 + \omega_k(1 - D_1)^{-1}D$ where p, q, and ω_k satisfy (5.18) and (5.19). We shall make plausible that the eigenfunctions of this operator are better conditioned than the eigenfunctions in Richardson's method. We recall that the eigenfunctions of A are given by

$$(3.5) \qquad e_+(n, m) = (1 - q\lambda_+(n,m))^{\frac{j+1}{2}} \sin nj\xi \sin ml\eta,$$

$$(3.6) \qquad e_\times(n, m) = (1 - q\lambda_\times(n,m))^{\frac{1}{2}} \sin nj\xi \sin ml\eta.$$

In the case of Richardson's method $1 - q\lambda$ assumes values in the interval $[c, d]$ where $c \sim 0$ and $d \sim 1$. In figure 5.1a some of these eigenfunctions are illustrated for a constant value of j.

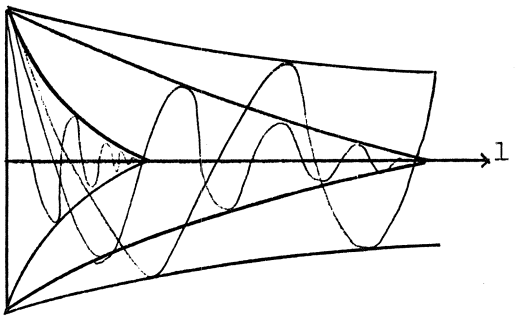

Fig. 5.1a

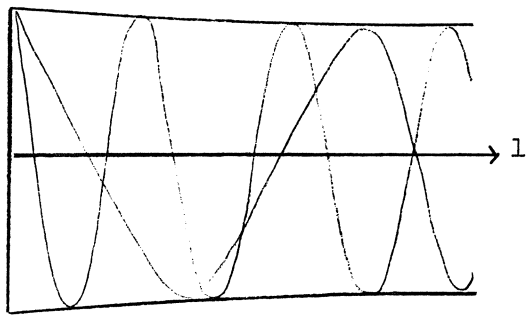

Fig. 5.1b

In case of successive overrelexation, however, we may derive from [6], (5.8) that for all λ

$$(5.23) \qquad |1 - q\lambda| = |1 - \Omega_{opt}| \sim 1 - 2\sqrt{\frac{2}{\gamma}}\,\xi \sim 1, \quad \gamma = 1, 2,$$

i.e. the eigenfunctions e(n, m) are almost orthogonal (see figure 5.1b). From those considerations we may conclude that these constants c_+ and $c_\times$ in (5.21') will be considerably smaller than the constants in the improved Liebmann method. (see table 5.1).

One may ask if it is possible to accelerate the method of successive overrelaxation by the gradient method as was done for methods of Jacobi and Gauss-Seidel. In these two cases the eigenvalues of the iterative operator A are real, so that the optimal value of ω could easily be found. In the case of successive overrelaxation, however, the eigenvalues are complex and it is not easy to see whether there is a better value for ω. We shall prove the following theorem.

Theorem I

Let $u_{k+1} = Au_k + f$ be a stationary process in which the eigenvalues α of A with $|\alpha| = \sigma(A)$ satisfy the inequality $|\alpha - \frac{1}{2}| < \frac{1}{2}$. Then there exists a number $\omega > 1$ such that the process

$$u_{k+1} = (1 - \omega)u_k + \omega(Au_k + f) \tag{5.24}$$

has asymptotically ($K \to \infty$) a larger rate of convergence than the process $u_{k+1} = Au_k + f$.

Proof

We define the function

$$f(\omega,\alpha) = |1 - \omega + \omega\alpha|^2 = \omega^2[1 + |\alpha|^2 - 2\mathrm{Re}\,\alpha] + 2\omega[\mathrm{Re}\,\alpha - 1] + 1. \tag{5.25}$$

For a fixed value of α the value of $f(\omega, \alpha)$ represents the eigenvalue of $1 - \omega + \omega A$ corresponding to α. Let $\omega = \omega_m$ be the value for which $f(\omega, \alpha)$ has a minimum and suppose that $\omega_m > 1$ if $|\alpha| = \sigma(A)$. The function $f(1, \alpha)$ is maximal for $|\alpha| = \sigma(A)$ and as $\omega_m > 1$ for $|\alpha| = \sigma(A)$, there exists a right-hand neighbourhood of $\omega = 1$ where all functions $f(\omega, \alpha)$ with $|\alpha| < \sigma(A)$ are less than any function $f(\omega, \alpha)$ with $|\alpha| = \sigma(A)$ (see figure 5.2).

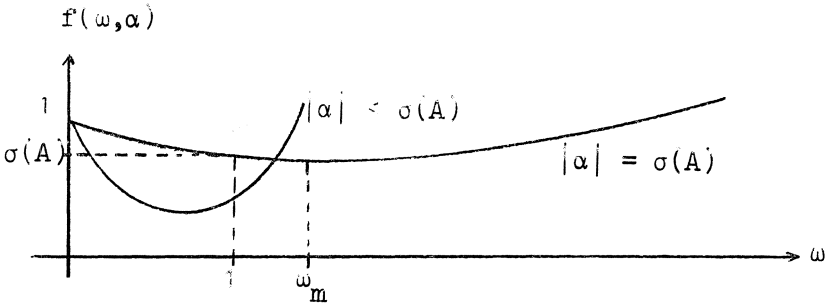

Fig. 5.2

Thus there exists a number $\omega > 1$ for which (5.24) has a smaller spectral radius than $\sigma(A)$ provided that $\omega_m > 1$ when $|\alpha| = \sigma(A)$. Since $\omega_m = (1 - \mathrm{Re}\,\alpha)/(1 + |\alpha|^2 - 2\mathrm{Re}\,\alpha)$ this last condition reduces to $|\alpha - \frac{1}{2}| < \frac{1}{2}$ when $|\alpha| = \sigma(A)$ which proves the theorem.

It is now easy to answer the question if there exists a better value of ω than the value defined by formulae (5.18), (5.19): From (5.23) it follows that the eigenvalues $\alpha = 1 - \omega\lambda = 1 - q\lambda$ of A lie all on a circle with radius $|1 - \omega_{opt}| \sim 1 - 2\sqrt{2/\gamma}\,\xi$. Further we may derive from [6], (5.8) that the real part of α assumes positive as well as negative values. Thus the condition of theorem I is violated so that there exists no better value of ω than the one already considered.

Improvement of the SOR method

In 1956 Garabedian [3] proposed a new approach to investigate convergence properties of iterative processes. His method essentially consists in associating a partial differential equation to the iterative method. He applied his method to the SOR method with respect to the five-point formula (+) and the nine-point formula ($\gamma = 5/3$).
We shall now apply the method of Garabedian to the more general iterative process (2.15), where we drop the condition that the region considered is a square.

We interpret the iterate u_k as a grid function at time $t = t_k = k\tau$. It is easily verified that for $\tau \to 0$, $\xi \to 0$, $\eta \to 0$ formula (2.15) transforms to the partial differential equation

$$WU_t + AU_{xt} + BU_{yt} = \Delta U, \tag{5.26}$$

where

$$\begin{cases} W = \frac{\tau}{\omega}\,(1 - pL_2 - q(2L_1 + L_3)), \\ A = \frac{\tau}{\omega}\,pL_2\,\xi, \\ B = \frac{\tau}{\omega}\,q(2L_1 - L_3)\eta. \end{cases} \tag{5.27}$$

By introducing the variable $z = t + Ax/2 + By/2$ this equation reduces to

$$(5.26') \qquad WU_z + \tfrac{1}{4}(A^2 + B^2)U_{zz} = \Delta U \ .$$

A particular solution of this equation is given by

$$(5.28) \qquad \exp(C_i z)\, E_i(x,y) = \exp(C_i t)\, \exp(\tfrac{1}{2} C_i(Ax + By))\, E_i(x, y),$$

where

$$(5.29) \qquad C_i = -2\, \frac{W \pm \sqrt{W^2 + \delta_i(A^2 + B^2)}}{A^2 + B^2}$$

and where $E_i(x, y)$ is an eigenfunction of the operator Δ satisfying the boundary conditions and corresponding to the eigenvalue δ_i. Since the eigenfunctions $E_i(x, y)$ are complete, we may write the solution U of (5.26) as a linear combination of the particular solutions (5.28). It is now assumed that $U(j\xi, l\eta, k\tau)$ is an approximation of the iterate u_k. Then the factor $\exp(C_i\tau)$ corresponds to $1 - \omega\lambda_i$ where λ_i is an eigenvalue of the operator $L = -(1 - D_1)^{-1}D$ and $\exp(\tfrac{1}{2} C_i(Ax + By))\, E_i(x, y)$ corresponds to the eigenfunction e_i of L. Further, we have for lagre values of t

$$(5.30) \qquad U \sim U_\infty(x, y) + a_1 \exp(C_1 t)\, \exp(\tfrac{1}{2}C_1(Ax + By))\, E_1(x, y),$$

where $U_\infty(x, y)$ is the steady state solution of (5.26), a_1 is a constant and where for C_1 is chosen the minus sign. The most rapidly convergence to the steady state solution is obtained if

$$(5.31) \qquad W^2 + \delta_1(A^2 + B^2) = 0,$$

and

$$(5.32) \qquad \frac{W}{A^2 + B^2} \quad \text{as large as possible.}$$

We shall analyse these conditions for the case $\rho = 1$, i.e. $\xi = \eta = h$. By substituting (5.27) we obtain the conditions

$$(5.31') \qquad (1 + \delta_1 h^2)\, (L_2^2 p^2 + (2L_1 + L_3)^2 q^2) +$$

$$+ 2L_2(2L_1 + L_2)pq - 2L_2 p - 2(2L_1 + L_2)q + 1 = 0,$$

and

(5.32') $\quad L_2^2 p^2 + (2L_1 + L_2)^2 q^2$ as small as possible.

In the case of the SOR method we have the extra requirement $p = q = \omega$.

One can easily verify that

$$(5.33) \qquad p = q = \omega = \frac{h^2}{\gamma + \sqrt{(-\delta_1)(\gamma^2 - 2\gamma + 2)}\, h}$$

satisfies the conditions (5.31') and (5.32'). The value of Ω becomes

$$(5.34) \qquad \Omega = \frac{2\gamma}{\gamma + (-\delta_1)(\gamma^2 - 2\gamma + 2)\, h} \sim 2(1 - \gamma^{-1}(-\delta_1)(\gamma^2 - 2\gamma + 2)\, h).$$

For $\gamma = 2$ and $\gamma = 5/3$ this expression yields the values already given by Garabedian [3]. For $\gamma = 1$, $\gamma = 2$ and $\delta_1 \sim -2$ (The model problem) the asymptotic behaviour for $k \to 0$ of Ω is identical to the optimal values of Ω given by formula (5.19). It may be remarked that the damping effect of the analytical model (5.26), (5.27) does not depend on ω. Therefore, the condition $\omega = p = q$ is not necessary optimal. It was shown above that in the case of the five-point formula (+) and (×) we actually have $\omega = p = q$, but for $\gamma \neq 1, 2$ one cannot easily verify that this value is optimal. In table 5.1 at the end of this section we have listed the values of $R^*(K)$ which are experimentally found for the improtant case $\gamma = 5/3$ for a number of values of ω in the neighbourhood of the SOR value.

We now drop the condition $p = q$. Condition (5.31') may be represented by an ellips in the $(L_2 p, (2L_1 + L_2)q)$ - plane (see figure 5.3).

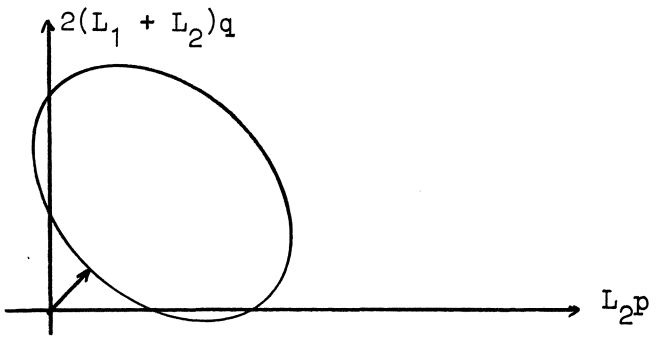

Fig. 5.3

Condition (5.32') is simply the requirement that the pair (p, q) should be such that the corresponding point $(L_2 p, (2L_1 + L_2)q$ is that point on the ellips which is nearest to the origin. From the symmetry of the problem we conclude that

$$L_2 p = (2L_1 + L_2)q \tag{5.35}$$

provided that $L_2 \neq 0$. In the case $L_2 = 0$ one can easily verify that the optimal value of q equals the value given by (5.33) for $\gamma = 1$. By substituting (5.35) into (5.31') we find

$$p = \frac{h^2}{(\gamma - 1)(2 + \sqrt{-2\delta_1}\, h}, \quad q = \frac{h^2}{2 + \sqrt{-2\delta_1}\, h}, \quad 1 < \gamma \leq 2. \tag{5.36}$$

For $1 < \gamma < 2$ these values differ from the values given by (5.33). Thus for $\gamma = 5/3$, one may expect an improvement of the rate of convergence. The value of ω has to be determined experimentally. In table 5.1 some numerical results are listed for the case $\gamma = 5/3$.
We remark that experimental results show that the method with $p = q = \omega$ can be improved by choosing ω different from p and q (see again table 5.1).

Numerical results

We have tested the methods described above for the homogeneous case of the model problem I with the initial approximation

$$u_o = (jh - 2)\,(lh - 2)\sin jh \sin lh,$$

where $h = \pi / N$. In order to compare the numerical rates of convergence with the values obtained in the preceding section we have chosen N = 19. As a measure for the numerical rate of convergence we again use the value of $R^*(K)$ defined by (4.1), p, q and ω are the parameters in (2.12) and (2.15). The number $c^*(K)$ is defined by

$$R^*(K) = R(\infty) - \frac{c^*(K)}{K},$$

which resembles (5.14) and (5.21).

Method	K	γ	ωh^{-2}	ph^{-2}	qh^{-2}	$R^*(K)$	$R(\infty)$	$c^*(K)$
Jacobi (+)	51	2	1/4	0	0	.040	.014	-1.3
Jacobi (×)	51	1	1/2	0	0	.067	.027	-2.0
Jacobi	51	3/2	1/3	0	0	.050	.018	-1.6
Improved Jacobi	51	3/2	1/2	0	0	.067	.027	-2.0
Jacobi	51	5/3	3/10	0	0	.046	.016	-1.5
Improved Jacobi	51	5/3	3/8	0	0	.054	.021	-1.7
Liebmann (+)	51	2	1/4	1/4	1/4	.071	.027	-2.2
Liebmann (+)	150	2	1/4	1/4	1/4	.045	.027	-2.6
Improved Liebmann (+)	51	2	1/2	1/4	1/4	-.35	.053	+20
Improved Liebmann (+)	150	2	1/2	1/4	1/4	-.28	.053	+50
ditto, with preconditioning	51	2	1/2	1/4	1/4	.094	.053	-2.1
Liebmann (×)	51	1	1/2	1/2	1/2	.11	.055	-2.5
Liebmann (×)	150	1	1/2	1/2	1/2	.072	.055	-2.6
Improved Liebmann (×)	51	1	1	1/2	1/2	-.28	.11	+20
Improved Liebmann (×)	150	1	1	1/2	1/2	-.10	.11	+31
ditto, with preconditioning	51	1	1	1/2	1/2	.14	.11	-1.6
SOR, form. (5.18), (5.19)	51	2	.43	.43	.43	.27	.33	2.9
SOR, form. (5.18), (5.19)	51	1	.81	.81	.81	.40	.47	3.5
SOR, form. (5.33)	51	5/3	.51	.51	.51	.30		
SOR	51	5/3	.54	.51	.51	.14		
SOR	51	5/3	.53	.51	.51	.19		
SOR	51	5/3	.52	.51	.51	.24		
SOR	51	5/3	.50	.51	.51	.32		
SOR	51	5/3	.49	.51	.51	.32		
SOR	51	5/3	.48	.51	.51	.31		
Improved SOR	51	5/3	.43	.64	.43	.28		
Improved SOR	51	5/3	.47	.64	.43	.31		
Improved SOR	51	5/3	.51	.64	.43	.24		
Improved SOR	51	5/3	.56	.64	.43	.052		

Table 5.1: results of stationary methods for the square 19 × 19.

6. The ALGOL 60 program

The ALGOL 60 program, which was used to find numerical solutions of the Dirichlet problem for Laplace's equation in a square of side π or a rectangle with sides π and $\kappa\pi$, is now reproduced. It was made to be used on the Electrologica X8 computer of the Mathematisch Centrum at Amsterdam. For a description of the MC standard procedures READ, SPACE, TAB, NLCR, CARRIAGE, NEW PAGE, ABSFIXT, FIXT, FLOT, PRINT and PRINTTEXT, which are all procedures that take care of the lay out given by the line printer, one is referred to [10].

A description of the procedures declared in the program is given now. See also the comments in the procedures themselves.

F(i)	calculates the initial approximation and the inhomogeneous term for the iteration process.
INITIALIZE	introduces the initial approximation and the inhomogeneous term.
START	the parameters for a new phase of the iterative process receive their values from the input tape; the relaxation parameters are calculated.
EIGENVALUE	calculates the dominating eigenvalue according to (4.6).
SEIDEL	performs one iteration.
FIX	a certain iterate $u^*_{k_o}$, where k_o = K fix + 1, is stored, after which the process can go on.
UNFIX	the iterate $u^*_{k_o}$ is picked up again.
OUTPUT	prints the net function u^*_k.

Next the parameters, in the order in which their values should appear on the input tape, are specified.

gtbn	is the number of the input tape.
M, N	specify the mesh length $\xi = \eta = h = \pi / N$, so that one has a rectangle covered by M × N squares.
KK	is the total number of iterations.
case	see the comment at the beginning of the program.

...	after case an arbitrary sequence of ALGOL symbols may follow, which will be reproduced through the line printer. The sequence has to end with a " ; ".
i	selects the initial approximation by means of F(i).
i	selects the inhomogeneous term by means of F(i).
K fix	in order to store $u^*_{k_o}$ one must choose K fix = k_o - 1.

In the procedure START the following parameters occur:

K	is the number of iterations of the phase, initiated by START.
A	is the value of a.
B	us the value of b.
c	is the value of γ
pp	p:= pp × h × h.
qq	q:= qq × h × h, p and q appearing in (2.12).
stat	if the Boolean stat becomes true, then the process is stationary, otherwise non-stationary.
...	(only if stat $\wedge$ case $\neq$ 2) w:= A + (B - A) × READ, and $\omega_k = 1/w$.
W	(only if $\neg$ stat) specifies which zero of $C_k(a, b, \lambda)$ is used in the calculation of the relaxation parameter ω_k.
j	determines the way in which the eigenvalue is calculated: if j = 0 no eigenvalue is calculated. if j = 1 then the procedure EIGENVALUE is activated.
...	(only if j = 1) if READ > 0 then the eucledian norm is used, otherwise the maximum norm. if j = 2 then the eigenvalue
E	(only if j = 2) is read from the input tape. if j = 3 then the analytic expressions for λ(n1, n2), known for this special case, are used. The numbers
n1, n2	(only if j = 3) must then be prescribed.

... if READ > 0 then OUTPUT.

choice if choice = 1 then the iterative process is continued. On the input tape the numbers given above beginning with K have to be prescribed.
if choice = 2 then the procedure UNFIX is activated; again, the numbers beginning with K have to be given.
if choice = 3 then the calculations are restarted with a new inhomogeneous term; the numbers above beginning with the second i have to appear on the input tape.
if choice = 4 the whole program is started again; all the numbers, except gtbn, have to be prescribed again.

Finally, the complete ALGOL 60 program follows.

```
begin comment R1582 TMTC 311067/08, 8157, CALCULATIONS FOR TW
      REPORT 109.
      Iterative solutions of the Dirichlet problem for Laplace's
      equation in a rectangle or square.
      case    A        B          c     pp = qq    stat    scheme
      1       2     8×(N/pi)↑2    2       0         1     Jacobi + stationary
      1       2     4×(N/pi)↑2    1       0         1     Jacobi × stationary
      1       4     4×(N/pi)↑2    2      .25        1     Seidel + stationary
      1       4     2×(N/pi)↑2    1      .5         1     Seidel × stationary
      3       2     8×(N/pi)↑2    2       0        -1     Jacobi + Richardson
      3       2     4×(N/pi)↑2    1       0        -1     Jacobi × Richardson
      2    arbitr.   arbitr.      2    arbitr.     1     SOR +
      2    arbitr.   arbitr.      1    arbitr.     1     SOR ×
      3       4     4×(N/pi)↑2    2      .25        -1    Seidel + Richardson
      3       4     2×(N/pi)↑2    1      .5         -1    Seidel × Richardson;
      integer choice,gtbn, M, N, KK, case;
      gtbn:= READ;
AGAIN: M:= READ;  N:= READ; KK:= READ; case:= READ;
```

```
begin   integer i, j, k, k0, l, K, I, Kfix;
        real L1, L2, L3, L4, S1, S2, S3, S4,
             w, A, B, c, p, q, h, eucl0, max0,
             Qeucl, Qmax, Deucl, Dmax, DDeucl,
             DDmax, E, om, omm, pi, DDeucl fix,
             DDmax fix, omm fix, Eeucl, Emax, Roneindig;
        integer array W[0:KK];
        real array u, v, u0, f[0:M,0:N];
        Boolean stat;

        real procedure F(i); value i; integer i;
        if i = 1 then F:= READ else
        if i = 2 then F:=0 else
        if i = 3 then F:= sin(j × h) × sin(l × h) else
        if i = 4 then F:= sin(j × h) × sin(l × h) × (j × h - 2) ×
                          (l × h - 2) else
        if i = 5 then F:= sin(j × h) × sin(l × h) × (j × h - 2) ×
                          (l × h - 2) × (j × h - 1) × (l × h - 2) else
        if i = 6 then F:= if l = N then read else 0 else
        if i = 7 ∨ i = 9 then
        begin   comment When case = 3 and i = 7 a sum of  eigen-
                functions is used as input function, when i = 9 a sum
                of functions sin(n × j × h) × sin(m × l × h);
                integer n, m; real func;
                func:= 0;
                for k := 1 step 1 until I do
                begin   n:= W[2 × k - 1]; m:= W[2 × k];
                        func:= func + (if i = 7 then (cos(m × h) ×
                        cos(n × h))↑l else 1) × sin(n × j × h) ×
                        sin(m × l × h)
                end;
                F:= func
        end else
        if i = 8 then F:= 1 else
```

```
if i= 10 then
begin integer n,m; F:= SUM(n,1,N,SUM(m,1,M,(cos(n × h) ×
                          cos(m × h)) /\ 1 × sin(n × j × h) ×
                          sin(m × 1 × h)))
end;

procedure INITIALIZE;
begin   NLCR; NLCR;
        PRINTTEXT(‡number of net points boundary included =†);
        ABSFIXT(2,0,M + 1); PRINTTEXT(‡×†); ABSFIXT(2,0,N + 1);
        for 1:= 0 step 1 until N do for j:= 0 step 1 until M
        do f[j, 1]:= F(i);
        for 1:= 0 step 1 until N do
        begin u0[0, 1]:= f[0, 1]; u0[M, 1]:= f[M, 1] end;
        for j:= 0 step 1 until M do
        begin u0[j, 0]:= f[j, 0]; u0[j, N]:= f[j, N] end;
        for 1:= 0 step 1 until N do for j:= 0 step 1 until M
        do v[j, 1]:= u[j, 1]:= u0[j, 1];
        om:= omm:= DDeucl:= DDmax:= Deucl:= Dmax:= 1;
        K:= -1; k0:= 0
end INITIALIZE;

procedure START;
begin   real pp, qq;
        procedure OP(fix, s, x); real x; Boolean fix;
        string s;
        begin   NLCR;
                if fix then begin FIXT(4,3,x); TAB end
                            else PRINT(x);
                TAB; PRINTTEXT(s)
        end;
        k:= K;
        K:= READ; A:= READ; B:= READ; c:= READ;
        pp:= READ; qq:= READ; stat:= READ > 0;
```

```
if  case = 1 then
begin w:= A + (B - A) × READ; S4:= 1/w end else
if case= 2 then
begin   comment The optimal relaxation parameter
        OMEGA is now calculated;
        real pq, labda A, labda BC, labda D, OMEGA;
        pq:= 2 × (1 - cos(h))/(h × h);
        labda A :=  2 -.5 × (2 - c) × pq × pq × h × h;
        labda BC:=  4/(h × h) - pq × (4 - 2 × c);
        labda D :=  8 × (c - 1)/(h × h) + (10 - 6 × c) ×
                    pq -.5 × (2 - c) × pq × pq × h × h;
        labda A :=  1 - h × h/(2 × c) × labda A;
        labda BC:=  1 - h × h/(2 × c) × labda BC;
        labda D :=  1 - h × h/(2 × c) × labda D;
        OMEGA   :=  if labda A < labda BC then
                    (if labda BC < labda D then
                    labda D else labda BC) else
                    labda A;
        OMEGA:=  2/(1 + sqrt(1 - OMEGA × OMEGA));
        S4:= OMEGA × h × h /(2 × c);
        w:=  1/S4
end
else if case = 3 then
begin   l:= K + k;
        for j:= k + 1 step 1 until l do W[j]:= READ
end;
p:= pp × h × h; q:= qq × h × h;
L1:= (1 - c/2)/(h × h);
L2:= L3:= (c - 1)/(h × h);
L4:= - 2 × c/(h × h);
if case = 2 then p:= q:= S4;
S1:= q × L1; S2:= q × L3; S3:= p × L2;
NLCR;
```

```
        OP(true,‹A›,A); OP(true,‹B›,B); OP(true,‹c›,c);
        if case ≠ 2 then
        begin OP(true,‹p/h\2›,pp); OP(true,‹q/h\2›,qq) end
        else OP(false,‹p = q›,p);
        OP(false,‹L1›,L1); OP(false,‹L2 = L3›, L3);
        OP(false,‹L4›,L4); OP(false,‹S1›,S1);
        OP(false,‹S2›,S2); OP(false,‹S3›,S3);
        if (case = 1 ∧ stat) ∨ case = 2 then OP(false,‹S4›,S4);
        NLCR;
        j:= READ;
        comment Here the eigenvalue of which the corresponding
        eigenfunction is to be eliminated, is introduced;
        if j = 0 then goto OUT;
        if j = 1 then E:= if READ > 0 then Eeucl else Emax;
        if j = 2 then E:= READ;
        if j = 3 then
        begin   integer n1,n2; real n1h, n2h;
                n1:= READ; n2:= READ;
                n1h:= n1 × h; n2h:= n2 × h;
                E:= if case = 3 then
                    (if pp < 10-6 ∧ qq < 10-6 then
                    -L4 - 2 × L2 × cos(n1h) - 2 × L3 × cos(n2h)
                    - 4 × L1 × cos(n1h) × cos(n2h)
                    else
                    -L4 + (4 × L2 × L2) × (cos(n1h) + cos(n2h))\2/
                    L4 + 16 × L1 × L1 × (cos(n1h) × cos(n2h))\2/L4)
                else 1
        end;
        A:= (2 × E + B × (cos(pi/(2 × K)) - 1))/(cos(pi/(2 × K)) + 1);
        OP(false,‹E›,E); OP(false,‹A›,A);
        NLCR;
OUT:    I:= K; K:= K + k; k0:= k + 1;
        if k0 = 0 then
```

```
        begin   NLCR;
                Roneindig:= if case = 1 then ln(w/(w - A)) else
                            if case = 2 then 2 × h × sqrt(2/c)
                            else
                            if case = 3 then 2 × sqrt (A/B)
                            else 0;
                PRINTTEXT(‹Roneindig = ›);
                FIXT(2,7,Roneindig); NLCR; NLCR;
                PRINTTEXT(‹     k  eucl          max›);
                SPACE(10); PRINTTEXT(‹conv. eucl›);
                SPACE(3);  PRINTTEXT(‹conv. max ›);
                SPACE(5);  PRINTTEXT(‹eigenv. eucl›);
                SPACE(3);  PRINTTEXT(‹eigenv. max›);
                SPACE(5);  PRINTTEXT(‹w›);
                SPACE(12); PRINTTEXT(‹S4›); SPACE(12);
                PRINTTEXT(‹c eucl       c max›); NLCR
        end
end START;

real procedure EIGENVALUE(eucl norm); Boolean eucl norm;
        EIGENVALUE:= 1/omm - (if eucl norm then Qeucl
        else Qmax)/om;

procedure SEIDEL;
begin   comment the k th residual, average rate of convergence
        and eigenvalue is calculated, and also the (k + 1)st
        iterand and q (see section 7 of TW 104[4]);
        real U, V, eucl, max, R, Reucl, Rmax;
        eucl:= max:= Deucl:= Dmax:= 0;
        if ¬ stat then
        begin   w:= .5 × (A + B + (A - B) × cos((2 × W[k] + 1)
                    × pi/(2 × I)));
                S4:= 1/w
        end;
```

```
for l:= 1 step 1 until N do for j:= 1 step 1 until M do
begin   V:= v[j,l];
        R:= L1 × (v[j + 1,l + 1] + v[j - 1,l + 1] +
                  v[j + 1,l - 1] + v[j - 1,l - 1]) +
            L2 × (v[j + 1,l] + v[j - 1,l]) +
            L3 × (v[j,l + 1] + v[j,l - 1]) +
            L4 × V + f[j,l];
        eucl:= eucl + R × R;
        if max < abs(R) then max:= abs(R);
        U:= u[j,l]:= V + S1 × (u[j - 1,l - 1] - v[j - 1,l - 1]
                               + u[j + 1,l - 1] - v[j + 1,l - 1])
                       + S2 × (u[j,l - 1] - v[j,l - 1])
                       + S3 × (u[j - 1,l] - v[j - 1,l])
                       + S4 × R;
        U:= abs(U - V);
        Deucl:= Deucl + U × U;
        if Dmax < U then Dmax:=  U
end;
NLCR; ABSFIXT(3,0,k);
Qeucl:= sqrt(Deucl/DDeucl); Qmax:= Dmax/DDmax;
eucl:= sqrt(eucl);
FLOT(7,2,eucl/((N + 1) × (N + 1))); FLOT(7,2,max);
if k = 0 then begin eucl0:= eucl; max0:= max ;
                    SPACE(52)
              end
else
begin   Reucl:= ln(eucl0/eucl)/k; Rmax:= ln(max0/max)/k;
        FIXT(2,7,Reucl); FIXT(2,7,Rmax);
        Eeucl:= EIGENVALUE(true);
        Emax:= EIGENVALUE(false);
        FIXT(4,7, Eeucl); FIXT(4,7, Emax)
end;
FIXT(4,7,w); FIXT(4,7,S4);
```

```
        if k ≠ 0 then
        begin   FIXT(4,7, k × (Roneindig - Reucl));
                FIXT(4,7,k × (Roneindig - Rmax))
        end;
        comment now the next iteration is prepared for;
        for l:= 1 step 1 until N do for j:= 1 step 1 until M
        do v[j,l]:= u[j,l];
        DDeucl:= Deucl; DDmax:= Dmax; omm:= om
end SEIDEL;

procedure FIX;
begin   for l:= 1 step 1 until N do for j:= 1 step 1 until M
        do u0[j,l]:= u[j,l];
        DDeucl fix:= Deucl; DDmax fix:= Dmax; omm fix:= om
end FIX;

procedure UNFIX;
begin   K:= Kfix;
        for l:= 1 step 1 until N do for j:= 1 step 1 until M
        do v[j,l]:= u0[j,l];
        DDeucl:= DDeucl fix; DDmax:= DDmax fix; omm:= omm fix;
        PRINTTEXT(⌞We return to K = ⌟);
        ABSFIXT(3,0,K + 1); NLCR
end UNFIX;

procedure OUTPUT;
begin   integer s;
        s:= if M < 19 ∨ N < 19 then 1 else 2;
        if LINE NUMBER > 53 - (if M < N then N else M)
        then NEW PAGE else CARRIAGE(3);
        PRINTTEXT(⌞Solution after ⌟); ABSFIXT(3,0,K + 1);
        PRINTTEXT(⌞iterations⌟); NLCR; NLCR;
        if N ≤ M then PRINTTEXT(⌞j horizontal⌟)
                 else PRINTTEXT(⌞l horizontal⌟);
```

```
        if N ≤ M then
        begin   for l:= 1 step s until N do
                for j:= 1 step s until M do
                begin   if j=1 then NLCR;
                        FLOT(1,2,u[j,l])
                end
        end
        else
        for j:= 1 step s until M do for l:= 1 step s until N do
        begin   if l = 1 then NLCR;
                FLOT(1, 2, u[j, l])
        end;
        CARRIAGE(4)
end OUTPUT;
comment OUTPUT only gives a reasonable lay-out when N < 39;

comment program really begins now;
PRINTTEXT(ǂResults R 1582 TMTC 311067/08, 8157, ǂ);
PRINTTEXT(ǂinputtape numberǂ); ABSFIXT(3,0,gtbn);
for i:= RESYM while i ǂ91 do PRSYM(i);
NLCR; i:= READ;
if i = 7 ∨ i = 9 then
begin   I:= READ; for k:= 1 step 1 until I do
        begin W[2 × k - 1]:= READ; W[2 × k]:= READ;
                PRSYM(98); ABSFIXT(2,0,W[2 × k - 1]);
                ABSFIXT(2,0,W[2 × k]); PRSYM(99)
        end; NLCR
end;
PRINTTEXT(ǂBeginapproximationǂ); ABSFIXT(2,0,i);
pi:= 3.141592653589793; h:= pi/(if M < N then N else M);
M:= M - 1; N:= N - 1;
for l:= 1 step 1 until N do for j:= 1 step 1 until M do
u0[j,l]:= F(i);
```

```
        N:= N + 1; M:= M + 1;
NEXT INHOM TERM:
        i:= READ; NLCR; PRINTTEXT(‡Inhomogeneous term†);
        ABSFIXT(2,0,i); INITIALIZE; Kfix:= READ;
NEW METHOD:
        START; N:= N - 1; M:= M - 1;
        for k:= k0 step 1 until K do SEIDEL;
        if READ > 0 then OUTPUT; choice:= READ;
        if K = Kfix then  FIX;
        if choice = 1 then begin N:= N + 1; M:= M + 1;
                                 goto NEW METHOD
                           end;
        comment now an empty iteration follows in order to calculate the
        last norm;
        k:= K + 1; W[k]:= W[K];  SEIDEL; NLCR; NLCR; NEW PAGE;
        if choice = 2 then begin UNFIX; N:= N + 1; M:= M + 1;
                                 goto NEW METHOD
                           end;
        M:= M + 1; N:= N + 1;
        if choice = 3  then goto NEXT INHOM TERM
end;
if choice = 4  then begin PRINTTEXT(‡Continuation of†); goto AGAIN end
end
```

References

[1] Forsythe, G.E. and Wasow, W.R. [1960] : Finite difference methods for partial differential equations. John Wiley and Sons, Inc., New York, London.

[2] Frankel, S.P. [1950] : Convergence rates of iterative treatments of partial differential equations. Math. Tables Aids Comput. 4, 65-75.

[3] Garabedian, P.R. [1956]: Estimation of the relaxation factor for small mesh size. Math. Tables and Other Aids to Comput. 10, 183-185.

[4] Houwen, P.J. van der [1967]: On the acceleration of Richardson's method I. Theoretical part. Report TW 104, Mathematisch Centrum, Amsterdam.

[5] Houwen, P.J. van der [1967]: On the acceleration of Richardson's method II. Numerical aspects. Report TW 107, Mathematisch Centrum Amsterdam.

[6] Houwen, P.J. van der [1967]: On the acceleration of Richardson's method III. Applications. Report TW 108, Mathematisch Centrum, Amsterdam.

[7] Houwen, P.J. van der [1968]: Finite difference methods for solving partial differential equations. Mathematical Centre Tract 20, Amsterdam.

[8] Varga, R. [1962]: Matrix iterative analysis. Prentice-Hall, Inc. Englewood Cliffs, New Jersey.

[9] Young, D.M [1954]: Iterative methods for solving partial difference equations of elliptic type. Trans. Amer. Math. Soc. 76, 92-111.

[10] Kruseman Aretz, F.E.J. [1966]: Sect. MC-ALGOL 60-systeem voor de X8, MR 81 Mathematisch Centrum, Amsterdam.