STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

AFDELING TOEGEPASTE WISKUNDE

Report TW 108

On the acceleration of Richardson's method III
Applications

by

P.J. van der Houwen

October 1967

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.

Introduction

The solution of the Dirichlet problem for Laplace's equation leads to matrix equations of positive type, which may be solved iteratively by Richardson's method. As was shown in the first report on the present investigation of Richardson's method (reference [3]), it is important to use a matrix equation with a small condition number. In this report we shall construct matrix equations, for which the condition number is smaller than the condition number of the schemes usually employed in practice. Further, some results are given when we apply the elimination method [3] to these equations, in which it appears that there was a considerable gaim in the rate of convergence.

2. The Laplace difference operator

In this section we discuss the discretization of the Laplace operator Δ. We shall investigate a more general difference operator than is usually considered.

On a grid with rectangular meshes of sides ξ and η, we define the difference operator D by the formula

$$D = \frac{X_+ + \alpha + X_-}{\alpha + 2} \cdot \frac{X_+ - 2 + X_-}{\xi^2} + \frac{Y_+ + \beta + Y_-}{\beta + 2} \cdot \frac{Y_+ - 2 + Y_-}{\eta^2}. \tag{2.1}$$

$X_{\pm}$ and $Y_{\pm}$ represent translations $\pm \xi$ and $\pm \eta$ along the x-axis and the y-axis respectively. The parameters α and β are weight parameters. It is obvious from the structure of D, that (2.1) represents a difference analogue of the Laplace operator. If $\alpha = \beta = \infty$ we have the usual five-point Laplace difference operator [1].

The approximation error

The approximation error is defined as the difference between the analytic operator and the difference operator. We shall compare the operators Δ and D.

We write the operator D in the form

$$D = L_1(X_+ + X_-)(Y_+ + Y_-) + L_2(X_+ + X_-) + L_3(Y_+ + Y_-) + L_4, \tag{2.2}$$

where the coefficients L_i are defined by

$$\begin{cases} L_1 = \frac{1}{2}\,\xi^{-2}(1 + \rho^2 - \gamma), \quad L_2 = \xi^{-2}(\gamma - \rho^2), \quad L_3 = \xi^{-2}(\gamma - 1), \\ \qquad\qquad\qquad\qquad\qquad\qquad L_4 = -2\,\xi^{-2}\gamma \\ \gamma = \dfrac{\alpha}{\alpha + 2} + \rho^2 \dfrac{\beta}{\beta + 2}, \quad \rho = \dfrac{\xi}{\eta} \end{cases} \tag{2.3}$$

Using Taylor's formula we may write $X_+ + X_-$ and $Y_+ + Y_-$ as differential operators of infinite order:

$$X_+ + X_- = 2 + \xi^2 \frac{\partial^2}{\partial x^2} + \frac{1}{12}\,\xi^4 \frac{\partial^4}{\partial x^4} + \frac{1}{360}\,\xi^6 \frac{\partial^6}{\partial x^6} + \frac{1}{20160}\,\xi^8 \frac{\partial}{\partial x^8} \cdots$$

A similar expression holds for $Y_+ + Y_-$.
Substituting these expressions into (2.2) and observing that

$$(2.4)\qquad 4L_1 + 2L_2 + 2L_3 + L_4 = 0,\quad 2L_1 + L_2 = \xi^{-2},\quad 2L_1 + L_3 = \eta^{-2},$$

yields

$$(2.5)\qquad D = \Delta + \frac{1}{12}\,\xi^2\left(\frac{\partial^4}{\partial x^4} + 12L_1\,\xi^2\rho^{-2}\,\frac{\partial^4}{\partial x^2\partial y^2} + \rho^{-2}\,\frac{\partial^4}{\partial y^4}\right)$$

$$+ \frac{1}{360}\,\xi^4\left(\frac{\partial^6}{\partial x^6} + 30L_1\,\xi^2\left(\rho^{-2}\,\frac{\partial^6}{\partial x^4\partial y^2} + \rho^{-4}\,\frac{\partial^6}{\partial x^2\partial y^4}\right) + \rho^{-4}\,\frac{\partial^6}{\partial y^6}\right)$$

$$+ \frac{1}{20160}\,\xi^6\left(\frac{\partial^8}{\partial x^8} + 14L_1\,\xi^2\left(4\rho^{-2}\,\frac{\partial^8}{\partial x^6\partial y^2} + 10\rho^{-4}\,\frac{\partial^8}{\partial x^4\partial y^4} + \right.\right.$$

$$\left.\left. + 4\rho^{-6}\,\frac{\partial^8}{\partial x^2\partial y^8}\right) + \rho^{-6}\,\frac{\partial^8}{\partial y^8}\right) + \ldots\ .$$

In general we have a <u>second order approximation</u> i.e.

$$(2.6)\qquad D = \Delta + O(\xi^2),\quad \xi \to 0.$$

However, when we restrict the considerations to harmonic functions $U(\Delta U = 0)$, we are able to obtain higher approximations. Putting $\frac{\partial^2}{\partial y^2} = -\frac{\partial^2}{\partial x^2}$ we find

$$(2.7)\qquad D = \Delta + \frac{1}{12}\,\xi^2\rho^{-2}(\rho^2 - 12L_1\xi^2 + 1)\,\frac{\partial^4}{\partial x^4}$$

$$+ \frac{1}{360}\,\xi^4(\rho^2 - 1)(\rho^2 + 1 - 30L_1\xi^2)\,\frac{\partial^6}{\partial x^6}$$

$$+ \frac{1}{20160}\,\xi^6\rho^{-6}(\rho^6 + 1 - 28L_1\xi^2\rho^4(2\rho^4 - 5\rho^2 + 2))\,\frac{\partial^8}{\partial x^8} + \ldots\ .$$

If we choose γ in such a way that

$$(2.8) \qquad L_1 = \frac{\rho^2 + 1}{12\xi^2} ,$$

we obtain a fourth order approximation:

$$(2.9) \qquad D = \Delta + \frac{1-\rho^4}{240} \xi^4 \frac{\partial^6}{\partial x^6} + \ldots .$$

In terms of γ and ρ condition (2.8) assumes the form

$$(2.10) \qquad \gamma = \frac{5}{6}(1 + \rho^2).$$

If in addition,we choose, $\rho = 1$ $(\xi = \eta)$, so that $\gamma = \frac{5}{3}$, a sixth order approximation is obtained:

$$(2.11) \qquad D = \Delta + \frac{1}{3024} \xi^6 \frac{\partial^8}{\partial x^8} + \ldots .$$

We find from (2.3) and (2.10) that this special case is known as the nine-point Laplace difference formula [1].

The considerations above hold for netpoints sufficiently far removed from the boundary. For instance, in figure 1 the nine-point formula is only applicable in the netpoints denoted by dots. In practice one may approximate the boundary Γ by the curve Γ^*, sothat the nine-point formula is also applicable in the starred netpoints. If the boundary conditions are of the first kind we have for Laplace's equation a very accurate difference scheme valid for all netpoints

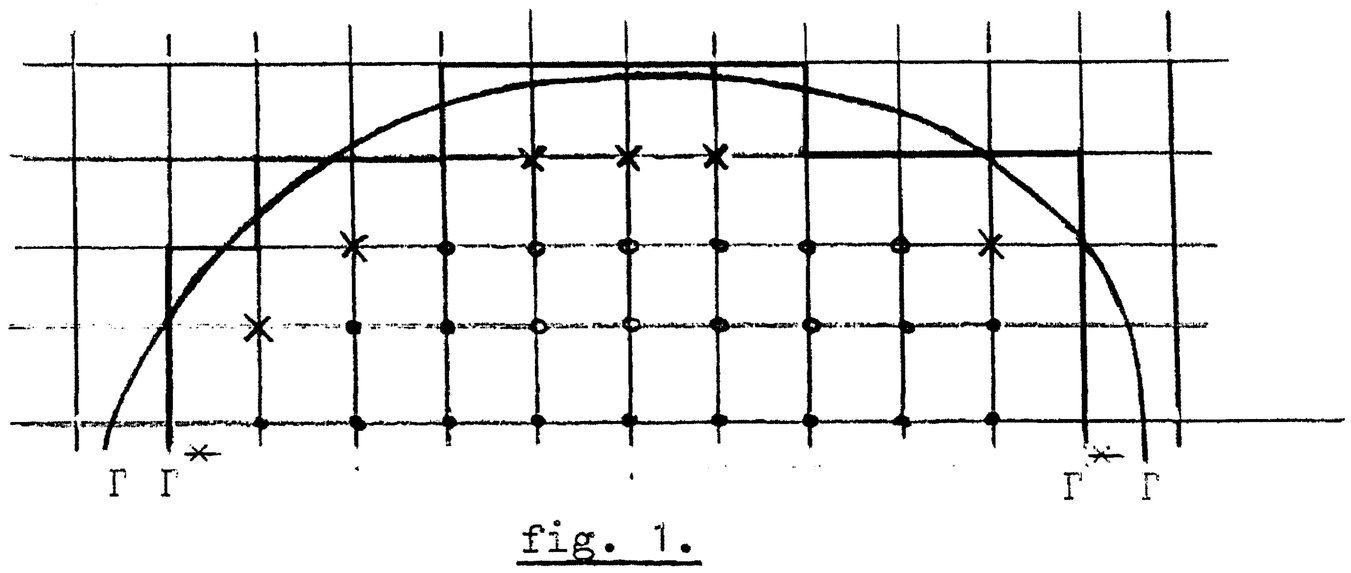

fig. 1.

The eigenvalues of the Laplace difference operator

We consider a discrete Dirichlet problem with a boundary of type Γ^* (see figure 1). The difference operator D which is applied at all

internal netpoints can be represented as a symmetric matrix. We shall also denote this matrix by D. The rows of this matrix representation corresponding to netpoints surrounded by internal netpoints have the form

$$(2.12) \qquad (\ldots, L_1, L_3, L_1, \ldots, L_2, L_4, L_2, \ldots, L_1, L_3, L_1, \ldots),$$

where L_4 is the diagonal entry. The other rows have the same structure, but where the difference formula uses boundary values, the coefficients L_i are absent. One may easily verify that the matrix D is symmetric.

Theorem I.

If γ and ρ satisfy the condition

$$(2.13) \qquad \max(1, \rho^2) \leq \gamma \leq 1 + \rho^2,$$

then the eigenvalues δ_i of D satisfy the inequalities

$$(2.14) \qquad -4\xi^{-2}\gamma < \delta_i < 0.$$

Further, the largest eigenvalue δ_1 has multiplicity 1.

Proof.

We write the matrix D in the form

$$D = L_4 + (D - L_4),$$

and we make the matrix $D - L_4$ non-negative by requiring that the matrix entries L_1, L_2 and L_3 are non-negative [5]. From (2.3) we obtain for γ and ρ the condition (2.12).

There exists an extensive theory concerning non-negative matrices [5]. One of the main results, due to Perron and Frobenius, states that an irreducible non-negative matrix has a simple eigenvalue which is equal to the spectral norm of the matrix and is smaller than the maximum norm of the matrix. Applying this to the matrix $D - L_4$ we find that the eigenvalue $\sigma(D - L_4)$ has multiplicity 1 and satisfies the inequality

$$\sigma(D - L_4) < 4L_1 + 2L_2 + 2L_3 = -L_4.$$

Hence

$$+2L_4 < \delta_i < 0,$$

where the largest eigenvalue δ_1 of D corresponds to the largest eigenvalue of $D - L_4$. Therefore δ_1 has multiplicity 1. Substituting $-2\xi^{-2}\gamma$ for L_4 (expression 2.3) we have (2.14).

In the following we assume that (2.13) is satisfied (for the operators characterized by $\gamma = \frac{5}{6}(1 + \rho^2)$, this means that $\frac{1}{\sqrt{5}} \leq \rho \leq \sqrt{5}$). We use Richardson's method for solving the matrix equation

(2.15) $\quad Du + f = 0.$

Putting $L = -D$ we get a matrix L with positive eigenvalues $\lambda = -\delta$. We have seen in [3] that the condition number $\sigma(L)/\lambda_1$ has to be as small as possible. From (2.14) we have

(2.16) $\quad \frac{\sigma(L)}{\lambda_1} < 4\,\frac{\gamma}{\lambda_1 \xi^2}.$

Because λ_1 has multiplicity 1 it is likely that for every γ and ρ

$$\lambda_1 = \Lambda_1 + O(1)$$

for $\xi \to 0$, Λ_1 being the lowest eigenvalue of $-\Delta$ (cf. [1]). Therefore the upperbound (2.16) depends essentially upon γ. Theorem I suggests for γ the value

(2.17) $\quad \gamma = \mathrm{Max}(1, \rho^2) = \xi^2\,\mathrm{Max}(\frac{1}{\xi^2}, \frac{1}{\eta^2})$

which leads to the least upperbound for the condition number, i.e.

(2.18) $\quad \frac{\sigma(L)}{\lambda_1} < \frac{4}{\lambda_1}\,\mathrm{Max}(\frac{1}{\xi^2}, \frac{1}{\eta^2})$

It may be remarked that this value of γ leads to a <u>seven-point formula</u>, and if $\rho = 1$ to an other <u>five-point formula</u> (we shall add the symbol (X) to distinguish it from the ordinary five-point formula). From the numerical point of view the values $\gamma = \rho = 1$ are very advantageous. The five-point formula (×) arises for $\gamma = 1 + \rho^2$. Its condition number satisfies the inequality

(2.19) $$\frac{\sigma(L)}{\lambda_1} < \frac{4}{\lambda_1}\left(\frac{1}{\xi^2} + \frac{1}{\eta^2}\right)$$

and the very accurate nine-point formula obtained for $\gamma = \frac{5}{6}(1 + \rho^2)$ satisfies the inequality

(2.20) $$\frac{\sigma(L)}{\lambda_1} = \frac{10}{3\lambda_1}\left(\frac{1}{\xi^2} + \frac{1}{\eta^2}\right)$$

Comparing (2.19) and (2.20) one might think that the nine-point formula is preferred over the five-point formula. This is not so for the nine-point formula is nearly twice as laborious. However in the case of the homogeneous Laplace equation the nine-point formula may be superior in view of the high accuracy of the approximation.

3. The model problem I.

We investigate the Dirichlet problem for the square of side π. This problem will be called the model problem I.

It is well-known that the eigenfunctions of the operator $L = -D$ are given by

(3.1) $$e(n.m) = \sin nj\xi \ \sin ml\eta \ ,$$

where $n = 1, 2, \ldots, \frac{\pi}{\xi} - 1$ and $m = 1, 2, \ldots, \frac{\pi}{\eta} - 1$.
Substituting (3.1) into the equation $Le = \lambda e$ gives the eigenvalues

(3.2) $$\lambda(n.m) = -L_4 - 2L_2 \cos n\xi - 2L_3 \cos m\eta - 4L_1 \cos n\xi \ \cos m\eta.$$

We are interested in the extreme values of the function $\lambda(n.m)$. Since λ is a harmonic function of the variables $\cos n\xi$ and $\cos m\eta$, where

(3.3) $$-\cos\xi \leq \cos n\xi \leq \cos\xi, \ -\cos\eta \leq \cos m\eta \leq \cos\eta,$$

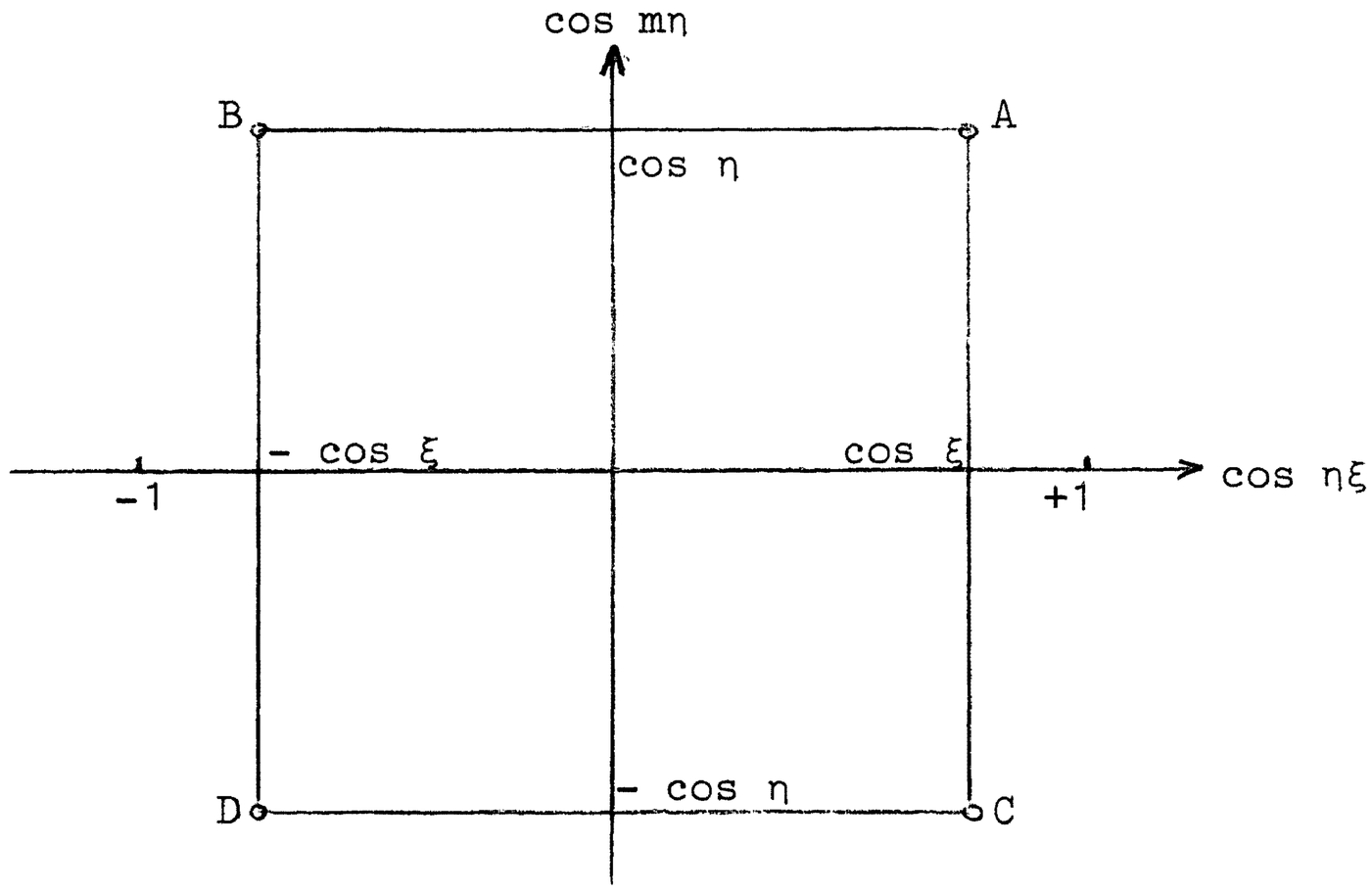

fig. 2.

The eigenvalue λ assumes its extrema at the boundary of the region (3.3); and because λ is a linear function of cos $n\xi$ and cos $m\eta$ along the boundary, the extrema are assumed in the corner points A, B, D and C (see figure 2).

From (3.2) and (2.3) we find

$$\begin{cases} \lambda_A = 2 - \dfrac{1+\rho^2-\gamma}{2\rho^2}\, pq\xi^2 \\ \lambda_B = 4\xi^{-2} - \left[p + \dfrac{2+\rho^2-2\gamma}{\rho^2}\, q\right] \\ \lambda_C = 4\rho^2\xi^{-2} - \left[p(1+2\rho^2-2\gamma) + q\right] \\ \lambda_D = 4(2\gamma-1-\rho^2)\xi^{-2} + \left[(2+3\rho^2-3\gamma)p + \dfrac{3+2\rho^2-3\gamma}{\rho^2}\, q\right] \\ \qquad\qquad - \left[\dfrac{1+\rho^2-\gamma}{2\rho^2}\, pq\xi^2\right], \end{cases}$$

where

$$(3.5) \qquad p = 2\,\frac{1-\cos\xi}{\xi^2}, \quad q = 2\,\frac{1-\cos\eta}{\eta^2}\ .$$

Clearly the condition number of the problem is given by

$$(3.6) \qquad \frac{\sigma(L)}{\lambda_1} = \frac{\mathrm{Max}(\lambda_B,\ \lambda_C,\ \lambda_D)}{\mathrm{Min}(\lambda_A,\ \lambda_D)}\ .$$

A first approximation to $\frac{\sigma(L)}{\lambda_1}$ is

$$(3.7) \qquad \frac{\sigma(L)}{\lambda_1} \sim 2\,\frac{\mathrm{Max}(1,\ \rho^2,\ 2\gamma-1-\rho^2)}{\xi^2}$$

where γ has to satisfy the inequality $\lambda_D \geq 2$. Hence we have

$$(3.8) \qquad 2\gamma - 1 - \rho^2 \geq \tfrac{1}{2}\xi^2\left(2\gamma - 3\rho^2 - 2 + \frac{3\gamma-2\rho^2-3}{\rho^2}\right)$$

as a first approximation.

From (3.7) and (3.8) it follows that the optimal condition number is assumed for

$$(3.9) \qquad \tfrac{1}{2}(1+\rho^2) \leq \gamma \leq \tfrac{1}{2}(1+\rho^2) + \tfrac{1}{2}\,\mathrm{Max}(1,\ \rho^2).$$

In figure 3 the behaviour of the condition number as a function of γ is illustrated

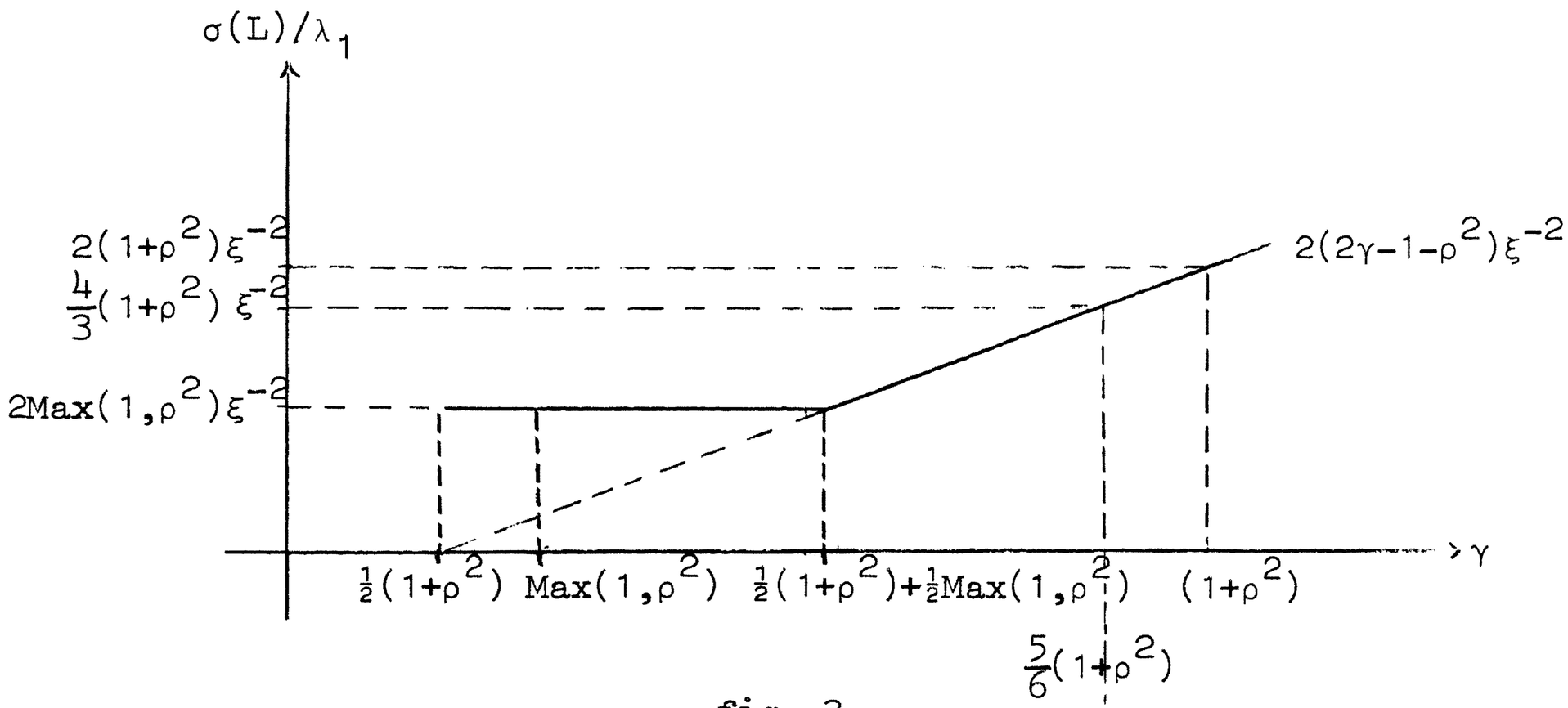

fig. 3.

In the following table the results for the model problem I are compared with the results for general Dirichlet problems derived in the preceding section.

TABLE I Condition numbers for the model problem I

Type of formula	γ	approximated condition number for model problem I	General upperbound
5 point formula (+)	$1+\rho^2$	$2(1+\rho^2)\xi^{-2}$	$2(1+\rho^2)\ \xi^{-2}\ (2\lambda_1^{-1})$
7 point formula	$\mathrm{Max}(1,\rho^2)$	$2\ \mathrm{Max}(1,\rho^2)\xi^{-2}$	$2\ \mathrm{Max}(1,\rho^2)\xi^{-2}\ (2\lambda_1^{-1})$
9 point formula	$\frac{5}{6}\ (1+\rho^2)$	$\frac{4}{3}\ (1+\rho^2)\xi^{-2}$	$\frac{5}{3}\ (1+\rho^2)\xi^{-2}.(2\lambda_1^{-1})$

We see that in the case of the model problem I ($\lambda_1 \sim 2$) the general upperbound is very close to the true value of the condition number.

4. Numerical results for model problem I

In this section numerical results will be given for model problem I with

(4.1) $\rho = 1;\quad \gamma = 1,\ 1.5;\quad \xi = \pi/20.$

They will be compared with those values, expected in the theory.

Theoretical rates of convergence for the elimination method

From [3] formula (4.1) we have the following expression for the rate of convergence for the elimination method

$$(4.2)\qquad R(K+K^*) = \ln(y_0+\sqrt{y_0^2-1}) - \frac{K^*\ln(y_0+\sqrt{y_0^2-1}) + \ln\sigma(E_{K^*}) + \ln 2}{K+K^*}$$

where $y_0 = (b+a)/(b-a)$, K and K^* are the numbers of iterates necessary to perform the reduction phase and the elimination phase respectively, and where $\sigma(E_{K^*})$ is the spectral norm of the elimination operator. The value of a is chosen as an approximation of one of the first eigenvalues of -D, and b is chosen as an approximation of the spectral norm of -D.

From (3.2) we find

(4.3) $\qquad a = n^2 + m^2, \qquad b = 162,$

where (n, m) indicates the eigenvalue $\lambda(n, m)$ of $L = -D$ (we remark that the relation $\lambda(n, m) \sim n^2 + m^2$ holds for small values of the positive integers n,m only). The number K^* is the sum of the numbers $K^*_{n,m}$ where $K^*_{n,m}$ is the number of iterates needed to eliminate the eigenfunction e(n, m) (see [3] section 4). In [3] we derived two formulae for $K^*_{n,m}$. The first (formula (5.2)) gives $K^*_{n,m}$ explicitly in terms of b and the eigenvalue $\lambda(n, m)$, i.e.

$$(4.4) \qquad K^*_{n,m} = \text{entier}\left(\tfrac{1}{4}\,\pi\sqrt{\frac{b}{\lambda(n, m)}}\right) + 1,$$

the second formula (formula (6.6)) gives $K^*_{n,m}$ implicitly, i.e.

$$(4.5) \qquad \ln(y_0 + \sqrt{y_0^2 - 1}) = \ln(T_{K^*_{n,m}+1}(y_0^*)) - \ln(T_{K^*_{n,m}}(y_0^*)),$$

where $y_0^* = (b \cos(\pi/2\, K^*_{n,m}) + \lambda(n, m)/(b - \lambda(n, m))$. In this case $K^*_{n,m}$ depends on a, b and $\lambda(n, m)$.

The value of $\sigma(E_{K^*})$ is defined by (cf. [3] section 4)

$$\sigma(E_{K^*}) = \sigma(\Pi C_{K^*_{n,m}}(a^*_{n,m}, b, L)).$$

We shall estimate $\sigma(E_{K^*})$ by the formula

$$(4.6) \qquad \sigma(E_{K^*}) \sim \Pi T^{-1}_{K^*_{n,m}}(y_0^*).$$

We bear in mind that $\sigma(E_{K^*})$ is smaller in theory.

By means of the formulae above and table IV given in [4], we obtain the following table:

TABLE II Lower bounds for the average rate of convergence of the elimination problem applied to model problem I satisfying (4.1) and (4.3)

a \ λ(n,m)	2	5	8	10	σ(E_{K^*})		Lower bound for $R(K+K^*)$	
2					1	1	0.222 - 0.693/K	
5	8				0.70	0.70	0.351 - 3.144/(K+8)	
8	5	7			0.55	0.62	0.444- 5.867/(K+15)	0.444- 5.543/(K+12)
10	4	6	8		0.40	0.13	0.497- 8.226/(K+17)	0.497- 7.599/(k+18)
13	4	4	5	7	0.22	0.15	0.566-11.065/(K+21)	0.566-10.142/(K+20)
$K^*_{n,m}$	8	5	4	4				

In the (a, λ(n, m))-diagram the values of $K^*_{n,m}$ satisfying (4.5) are given. The values defined by (4.4) are listed in the last row of the table. In the columns headed by σ(E_{K^*}) and $R(K+K^*)$ the first values correspond to values of $K^*_{n,m}$ calculated from (4.4), the second values to values of $K^*_{n,m}$ calculated from (4.5).

Numerical values for the rate of convergence of the elimination method

In our calculation we have considered the homogeneous case of modelproblem I. The startingvector u_0 was selected from the following table of functions

stv	starting vector
1	u_0 := read
2	u_0 := 0
3	u_0 := sin x sin y
4	u_0 := (x-2)(y-2) sin x sin y
5	u_0 := (x-1)(y-1)(x-2)(y-2) sin x sin y
6	u_0 := if y = π then 1 else 0

This table corresponds to the procedure F(i) of the ALGOL 60 program given at the end of this paper.
In the following we denote the staringvector, which is used in the calculation by its number stv.

It may be remarked that startingvector 6 yields the boundary value problem considered by Young and Warlick [7].

In table III the results of a sequence of experiments on the X8 Electrologica computer of the Mathematical Centre are reproduced. We have calculated the values of the quantity

$$R^{*}(K+K^{*}) = -\frac{1}{(K+K^{*})} \ln \frac{||Lu^{*}_{K+K^{*}}||}{||Lu_0||} \tag{4.7}$$

where $u^{*}_{K+K^{*}}$ denotes the solution which is actually found. As was shown in [4] section 3, $R^{*}(K+K^{*})$ is bounded below by $R(K+K^{*})$, whose values may be derived from table II.

TABLE III Numerical rates of convergence for model problem I

$K+K^{*}$	K_1^{*}	K_2^{*}	a	b	γ	stv	$R^{*}(K+K^{*})$	$R(K+K^{*})$
27	0	0	2	162	1.5	4	0.21	0.20
27	0	0	2	162	1.5	5	0.23	0.20
27	8	0	5	162	1.5	4	0.25	0.24
27	8	5	8	162	1,5	4	0.26	0.23
81	0	0	2	162	1.5	4	0.21	0.21
52	8	5	8	162	1.5	4	0.32	0.33
52	8	5	8	162	1	4	0.31	0.33
52	8	5	8	162	1.5	5	0.35	0.33
52	8	5	8	162	1	5	0.36	0.33
52	8	5	8	162	1.5	6	0.35	0.33
52	8	5	8	162	1	6	0.35	0.33
51	5	7	8	162	1.5	4	0.31	0.34
51	5	7	8	162	1.5	5	0.36	0.34
51	5	7	8	162	1.5	6	0.34	0.34
51	5	7	8	162	1	6	0.34	0.34
60	8	0	5	162	1.5	4	0.29	0.30
60	8	5	8	162	1.5	4	0.30	0.35
60	8	5	8	162	1.5	5	0.35	0.35
60	8	5	8	162	1.5	6	0.30	0.35

In the first series of experiments we have chosen a relatively low value for $K+K^*$. A rough knowledge of the eigenvalue of $e(n, m)$ was sufficient in order to eliminate this eigenfunction. We used the approximation $\lambda(n, m) \sim n^2 + m^2$.
Table II shows a modest improvement of the numerical rate of convergence $R^*(K+K^*)$. This agrees with the theory.

In a second series of experiments we used larger values for $K+K^*$, which gave a stronger reduction of the negative term in the expression for $R(K+K^*)$ (see table II) than the reduction in the first experiments, so that a considerable improvement is expected.
We tested the elimination method applied to the eigenfunction $e(1, 1)$ and $e(1, 2)$. The corresponding eigenvalues were calculated from the exact formula (3.2) as the approximations $\lambda(1, 1) \sim 2$ and $\lambda(1, 2) \sim 5$ were too rough for this case.
The results of our experiments justify the expectation above. Even in the most unfavourable testcase i.e. stv = 4, we obtained a 50% higher rate of convergence than with Richardson's method. Moreover, these high rates of convergence were reached after less iterations (about 50) than required by Richardson's method (about 81). In fact the number of iterates were so chosen that theoretically both Richardson's method and the elimination method gave the same accuracy for the approximation u_{K+K^*}. This could be verified by comparing the values of the quantity $(K+K^*)\ R(K+K^*)$, which is a measure for the accuracy reached after $K+K^*$ iterations. Thus the elimination method gave after approximately 50 iterations about the same result as Richardson method after 81 iterations.

Finally we did experiments without using the fact that we knew the eigenvalues $\lambda(n, m)$. After the first phase of our method, in which the late eigenfunctions were reduced, the eigenvalue of the dominating eigenfunction was calculated by one of the formulae given in [3] section 7.
The first eigenvalue was known with reasonable accuracy after about 47 iterates. The elimination of the first eigenfunction immediately revealed an estimate of the second eigenvalue, so that we could also eliminate the second eigenfunction.

5. The model problem II

Model problem II is the Dirichlet problem for the square of side π, where the operator $L = -D$ is replaced by the operator.

$$L = -(1 - D_1)^{-1}D. \tag{5.1}$$

D_1 is defined by

$$D_1 = pL_2X_- + qL_1X_-Y_- + rL_3Y_- + sL_1X_+Y_-, \tag{5.2}$$

where p, q, r and s are real parameters.

The matrix problem $-Du = f$ becomes

$$-(1 - D_1)^{-1}Du = g, \tag{5.3}$$

where $g = (1 - D_1)^{-1}f$, and the iteration scheme becomes

$$(1 - D_1)u_{k+1} = (1 - D_1 + \omega D)u_k + \omega f, \tag{5.4}$$

which is an explicit scheme resembling Gauss-Seidel's method [1] for elliptic differential equations.

The eigenvalues of the operator L

The eigenvalues λ and eigenfunctions e of L satisfy the relation

$$(\lambda(1 - D_1) + D)e = 0. \tag{5.5}$$

From (2.2) and (5.2) we obtain the equation

$$\begin{aligned}\{L_4 + \lambda + L_2[(1-p\lambda)X_- + X_+] + L_3[(1-r\lambda)Y_- + Y_+] \\ + L_1[(1-q\lambda)Y_- + Y_+]X_- \\ + L_1[(1-s\lambda)Y_- + Y_+]X_+\}e = 0.\end{aligned} \tag{5.5'}$$

From this equation the eigenvalues λ may be derived. We shall employ the following theorem.

Theorem II

The eigenfunction $a_n(j\xi)$ of the operator $PX_+ + QX_-$, which satisfy the boundary conditions $a(0) = a(\pi) = 0$, are of the form

$$a_n(j\xi) = P^{-\frac{1}{2}j}\, Q^{\frac{1}{2}j} \sin nj\xi, \quad n = 1, \ldots, \pi/\xi - 1, \tag{5.6}$$

and the eigenvalues μ are given by

(5.7) $$\mu_n = 2\sqrt{PQ}\cos n\xi, \quad n = 1, \ldots, \pi/\xi - 1.$$

Proof.
See Frankel [2] or van de Vooren and Vliegenthart [6].

Corollary
If r = s = q then the eigenvalues $\lambda(n, m)$ of L are defined by equation

(5.8) $$R = 2\sqrt{PQ}\cos n\xi, \quad n = 1, 2, \ldots, \pi/\xi - 1,$$

where

$$P = L_2 + 2L_1\sqrt{1-q\lambda}\cos m\eta$$

$$Q = L_2(1-p\lambda) + 2L_1\sqrt{1-q\lambda}\cos m\eta$$

$$R = -L_4 - \lambda - 2L_3\sqrt{1-q\lambda}\cos m\eta$$

and m = 1, 2, ..., $\pi/\eta - 1$.

Proof
Let us write $e = e(n, m) = a_n(j\xi)\, b_m(l\eta)$. From theorem II it follows that the functions

(5.9) $$b_m(l\eta) = (1-q\lambda)^{\frac{1}{2}l}\sin ml\eta, \quad m = 1, \ldots, \pi/\eta - 1,$$

are eigenfunctions of the operator $(1-q\lambda)Y_- + Y_+$ with eigenvalues

(5.10) $$\nu_m = 2\sqrt{1-q\lambda}\cos m\eta.$$

Substituting $e = a_n(j\xi)\, b_m(l\eta)$ into (5.5') and using (5.9) and (5.10) we have the equation

(5.11) $$(PX_+ + QX_-)\, a_n(j\xi) = Ra_n(j\xi),$$

where P, Q and R are defined above.
Formula (5.11) implies that R is an eigenvalue of the operator $PX_+ + QX_-$; on the other hand we have for the eigenvalues of $PX_+ + QX_-$ the expression (5.7). This proves equality (5.8).

In general, equation (5.8) leads to a quartic equation in $\sqrt{1-q\lambda}$. In this paper we consider some particular cases in which (5.8) reduces to a quadratic equation of the form

(5.12) $$(1-q\lambda) - A\sqrt{1-q\lambda} + B = 0,$$

where A and B are certain functions of p, q, n and m.
It is easily verified that λ satisfies the inequality

$$0 < \lambda < \frac{1}{q}, \tag{5.13}$$

when A and B satisfy the following inequalities for all n and m

$$0 \leq B \leq \tfrac{1}{4}A^2, \quad A - B < 1. \tag{5.14}$$

(a) The five-point formula (+)

This formula arises for $\gamma = 1+\rho^2$ or $L_1 = 0$. We obtain

$$P = L_2,\ Q = L_2(1-p\lambda),\ R = -L_4 - \lambda - 2L_3\sqrt{1-q\lambda}\cos m\eta.$$

Putting p = q the eigenvalue equation becomes of type (5.12):

$$(1-q\lambda) - 2q(L_2\cos n\xi + L_3\cos m\eta)\sqrt{1-q\lambda} - L_4 q - 1 = 0.$$

The inequalities (5.14) lead to

$$q = \frac{-1}{L_4}. \tag{5.15}$$

Further we have $\lambda_1 = \lambda(1, 1) \sim 4$, so that

$$\frac{\sigma(L)}{\lambda_1} \sim -\frac{1}{4}\ L_4 = \tfrac{1}{2}(1+\rho^2)\xi^{-2}. \tag{5.16}$$

The condition number of the corresponding difference scheme in Jacobi's method is four times larger as (5.16).

b) The five-point formula (×)

This second five-point formula is obtained for $\gamma = \rho^2 = 1$ or $L_2 = L_3 = 0$. In the same manner as above we find that

$$q = \frac{-1}{L_4}, \tag{5.17}$$

and $\lambda_1 = \lambda(1, 1) \sim 4$, hence

$$\frac{\sigma(L)}{\lambda_1} \sim -\tfrac{1}{4}\ L_4 = \tfrac{1}{2}\xi^{-2}. \tag{5.18}$$

Again this condition number is $\frac{1}{4}$ the value obtained for the corresponding scheme in Jacobi's method.

6. The ALGOL program

In this section we give an ALGOL program for the reduction-elimination method applied to model problem I and II where $\xi = \eta$. The program is intended to be used on the Electrologica X8 computer of the Mathematical Centre Amsterdam. Hence it uses the MC procedures abs, sqrt, sin, cos, ln, exp, read, SPACE, NLCR, NEW PAGE, ABSFIXT, FIXT, FLOT and PRINTTEXT. For a description of these procedures we refer to reference [8].
For the sake of convenience we represent the program in a more laborious form than necessary for practical purposes.

First we describe the procedures declared in the program:

F(i)	calculates the begin approximation and inhomogeneous term of the iteration process.
COMPUTE	represents the complete reduction-elimination method.
INPUT	introduces the begin approximation and inhomogeneous term into the procedure COMPUTE.
INFORMATION	introduces and calculates parameters which are necessary in the reduction and elimination phases.
JACOBI	represents the first order scheme given in [3] by formula (1.2) with $\alpha_k = 1$.
FRANKEL	represents the second order scheme given in [3] by formula (1.2) with $\alpha_k \neq 1$.
SEIDEL	calculates the fundamental eigenvalues of the operator L by means of formulae given in [3], section 7.
RESIDUAL	calculates the function $Lu_k^* - f$.
DIFF SCHEME	calculates the coefficients of the matrix equations and performs one iteration.
OUTPUT	prints the function u_k^*.

Next we specify the parameters which form the input for the program

getbnr	is the number of the input tape.
N	specifies the mesh length: $\xi = \eta = \pi/N$.
K	is the total number of iterations.
i	selects the begin approximation by means of F(i).

i selects the inhomogeneous term by means of F(i).

The following numbers have to be repeated for each reduction phase and for each eigenfunction of which the elimination is desired.

K is the number of iterations of the reduction phase or elimination process.

n selects the procedure by which the reduction or elimination is achieved:
if n = 1 then JACOBI is used,
if n = 2 then FRANKEL is used,
if n = 3 then SEIDEL is used.

A is the value of a.

B is the value of b.

c is the value of γ.

j determines the eigenvalue calculation:
if j = 0 then no eigenvalue is calculated.
if j = 1 then procedure EIGENVALUE is activated. Hence one has to prescribe the numbers method and norm, which indicate the type of formula and norm respectively.
if j = 2 then the eigenvalue must be given as an additional number on the input tape.
if j = 3 then the eigenvalue $\lambda(n_1, n_2)$ is calculated from formula (3.2). In this case the numbers n_1 and n_2 have to be prescribed.

p, q are arbitrary when n = 1 or n = 2.
if n = 2 then $p\xi^2$ and $q\xi^2$ equal the numbers p and q introduced in the preceding section.

W specifies in the cases n = 1 and n = 3 which zero of $C_K(a, b, \lambda)$ is used in the calculation of the relaxation parameter.

The remaining numbers are most easily defined by the statements:

if read > 0 then iteration process is continued;
if read > 0 then OUTPUT;
if read > 0 then the whole program is started again.

Finally, we give the complete ALGOL program.

```
begin   comment Opdrachtnr R1205, codenr vdH 111166 / II.
                Iteratieve oplossing van de LAPLACE vergelijking m.b.v.
                de Richardson methode toegepast op de schemas van
                Jacobi, Frankel en Seidel;
BEGIN: begin    integer getbnr; getbnr:= read;
         begin integer i, j, l, k0, N, K;
               real    h;
               array   U0[0:M,0:M];

               real procedure F(i); integer i;
               begin   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 - 1) else
                       if i = 6 then F:= if l = N then read else 0
end;

procedure COMPUTE;
begin   integer k, n, n1, n2;
        real    L1, L2, L3, L4, J, F1, F2, F3, S1, S2,
                S3, S4, U, R, R1, R2, E, W, I, A, B,
                a, b, c, p, q, h, om1, om2, V1, V2;
        array   f, r, u, v[0:N,0:N], arrayn, arrayW, arrayI,
                arrayA, arrayB[1:K], arrayc[0:K];
```

```
procedure INPUT;
begin   NLCR; NLCR; PRINTTEXT(↓ Aantal netpunten (N+1)×(N+1)= ↓);
        ABSFIXT(4,0,(N+1)×(N+1)); h:= 3.141592653589793 / N;
        for j:= 0 step 1 until N do for l:= 0 step 1 until N do
        begin   v[j,l]:= u[j,l]:= U0[j,l]; f[j,l]:= F(i) end;
        k:= k0; N:= N - 1; NLCR; NLCR;
        SPACE(43); V1:= V2:= om1:= om2:=1
end;

procedure INFORMATION;
begin   K:= read; n:= read; A:= read; B:= read; c:= read; j:= read;
        if k = k0 then arrayc[k0]:= c;
        if j = 0  then goto AA;
        if j = 1  then E:= EIGENVALUE(arrayn[k],read,read);
        if j = 2  then E:= read;
        if j = 3  then
        begin   n1:= read; n2:= read;
                E:= -L4 - 2×L2 × (cos(n1 × h) + cos(n2 × h))
                        -4 × L1 × cos(n1 × h) × cos(n2 × h)
        end;
        A:=(2 × E + B × (cos((N+1)×h/(2×K)) - 1)) / (cos((N+1)×h/(2×K)) + 1);
        p:= read; q:= read;
AA:     for j:= k + 1 step 1 until k + K do
        begin   arrayn[j]:= n; arrayA[j]:= A; arrayB[j]:= B; arrayc[j]:= c;
                if n = 1 ∨ n = 3 then
                begin arrayW[j]:= read; arrayI[j]:= K end;
                if n = 2 then begin arrayW[j]:= j - k; arrayI[j]:= 0 end;
        end;
        K:= K + k
end;
```

```
procedure JACOBI;
for j:= 1 step 1 until N do for l:= 1 step 1 until N do
begin    U:= v[j,l]:= u[j,l];
         u[j,l]:=U + J × r[j,l]
end;

procedure FRANKEL;
for j:= 1 step 1 until N do for l:= 1 step 1 until N do
begin    U:= u[j,l];
         u[j,l]:= F1 × U + F2 × r[j,l] + F3 × v[j,l];
         v[j,l]:= U
end;

procedure SEIDEL;
for l:= 1 step 1 until N do for j:= 1 step 1 until N do
begin    U:= v[j,l]:= u[j,l];
         u[j,l]:= U + S2 × (u[j-1,l] - v[j-1,l])
                    + S4 × (u[j,l-1] - v[j,l-1])
                    + S3 ×((u[j-1,l-1] - v[j-1,l-1]) + (u[j+1,l-1] - v[j+1,l-1]))
                    + S1 × r[j,l]
end;

procedure OUTPUT;
begin    integer s;
         NLCR; NLCR; PRINTTEXT(⊄ Oplossing na ⊅);
         ABSFIXT(3,0,k); PRINTTEXT(⊄ iteraties ⊅); NLCR;
         N:= N + 1;
         if N < 12 then s:= 1 else if N < 24 then s:= 2 else
         if N ≤ 36 then s:= 3 else if N ≤ 48 then s:= 4 else goto END;
         for l:= N step -s until 0 do
         begin NLCR; for j:= 0 step s until N do FLOT(5,2,u[j,l]) end;
         NEWPAGE;
END:     N:= N - 1
end;
```

```
real procedure EIGENVALUE(n, method,norm); value method, norm;
                integer n, method, norm;
begin    if n = 1 ∨ n = 3 ∨ n = 4 then
         begin    if method = 1 then
                  begin    if norm = 1 then EIGENVALUE:= 1/om2 - S1/om1;
                           if norm = 2 then EIGENVALUE:= 1/om2 - S2/om1
                  end;
                  if method = 2 then
                  begin    if norm = 1 then EIGENVALUE:= 1 / om1 - F1;
                           if norm = 2 then EIGENVALUE:= 1 / om1 - F2
                  end
end;

if n = 2 then
begin    A:= arrayA[k]; B:= arrayB[k];
         I:= sqrt(A) + sqrt(B); I:= I × I;
         W:= sqrt(A) - sqrt(B); W:= W × W;
         if method = 1 then
         begin    if norm = 1 then
                  EIGENVALUE:= 4 ×F3×(sqrt(A×B) - F3)/(I - 4×F3);
                  if norm = 2 then
                  EIGENVALUE:= 4×S3×(sqrt(A×B) - S3)/(I - 4×S3);
         end;
         if method = 2 then
         begin    if norm = 1 then
                  EIGENVALUE:= (-I×S1×S1+2×(A+B)×S1-W)/(4×S1);
                  if norm = 2 then
                  EIGENVALUE:= (-I×S2×S2+2×(A+B)×S2-W) /(4×S2);
         end;
         if method = 3 then
         begin    if norm = 1 then
                  EIGENVALUE:= 4×F1×(sqrt(A×B) - F1)/(W + 4×F1);
                  if norm = 2 then
                  EIGENVALUE:= 4×F2×(sqrt(A×B) - F2)/(W + 4×F2);
         end;
end
end;
```

```
procedure RESIDUAL;
begin   F3:= R1; S3:= R2; R1:= R2:= U:= W:= I:= F1:= F2:= 0;
        c:= arrayc[k];
        L1:= (1 - c/2) / (h × h); L2:= L3:= (c - 1) / (h × h);
        L4:= - 2 × c / (h × h);
        for j:= 1 step 1 until N do for l:= 1 step 1 until N do
        begin   U:= u[j,l];
                R:= r[j,l]:= L1 × (u[j+1,l+1] + u[j-1,l+1]
                                 +  u[j+1,l-1] + u[j-1,l-1])
                          + L2 × (u[j+1,l] + u[j-1,l])
                          + L3 × (u[j,l+1] + u[j,l-1])
                          + (L4 + T) × U - f[j,l];
                R1:= R1 + R × R; R2:= if R2 < abs(R) then abs(R) else R2;
                I := I  + U × U; W := if W  < abs(U) then abs(U) else W;
                U:= abs(U - v[j,l]);
                F1:= F1 + U × U; F2:= if F2 < abs(U) then abs(U) else F2;
end;

j:= (N + 2) × (N + 2); SPACE(2);
FLOT(2,2,sqrt(R1)/j); FLOT(2,2,R2); SPACE(2);
if k = k0 then begin a:= R1; b:= R2 end;
if k > k0 then
begin   FLOT(2,2,sqrt(F1/I)); FLOT(2,2,F2/W);
        S1:= sqrt(F1 / V1); S2:= F2 / V2; V1:= F1; V2:= F2;
        F3:= sqrt(F3 / F1); S3:= S3 / F2;
        F1:= sqrt(R1 / F1); F2:= R2 / F2;
        FIXT(5,16,EIGENVALUE(n,1,1)); FIXT(5,16,EIGENVALUE(n,1,2));
        NLCR; SPACE(45); FIXT(2,7,ln(a/R1) / (2 × k)); SPACE(8);
        FIXT(2,7,ln(b/R2) / k); SPACE(6);
        FIXT(5,16,EIGENVALUE(n,2,1));FIXT(5,16,EIGENVALUE(n,2,2));
        if n = 2 then
        begin NLCR; SPACE(83); FIXT(5,16,EIGENVALUE(n,3,1));
              FIXT(5,16,EIGENVALUE(n,3,2))
        end
    end
end;
```

```
procedure DIFF SCHEME;
begin   k:= k + 1; n:= arrayn[k]; W:= arrayW[k]; I:= arrayI[k];
        A:= arrayA[k]; B:= arrayB[k]; NLCR; ABSFIXT(3,0,k);

        if n = 1 then
        begin   PRINTTEXT(ᚲ   ja ᚦ);
                W:= .5 × (A + B + (A - B) × cos(1.57079 × (2 × W + 1)/I));
                I:= A := B:= 0; om2:= om1; om1:= J:= 1 / W; JACOBI
        end;

        if n = 2 then
        begin   PRINTTEXT(ᚲ   fj ᚦ);
                if A > -B then
                begin   I:= (A + B) / (A - B);
                        F1:= I + sqrt( I × I -1); F2:= I - sqrt(I × I - 1);
                        W:= -.5 × (A + B) + .5 × (A - B) ×
                        (F1 ↑(W + 1) + F2 ↑(W + 1)) / (F1 ↑ W + F2 ↑ W);
                        I:= .25 × (A + B); A:= B := 0
                end;
                F1:= 4 × I / (2 × I + W);
                F2:= 2 / (2 × I + W);
                F3:= (W - 2 × I) / (2 × I + W); FRANKEL
        end;

        if n = 3 then
        begin   PRINTTEXT(ᚲ   seᚦ);
                W:= .5×(A + B + (A - B)×cos(1.57079 × (2 × W + 1)/I));
                om2:= om1; S1:= om1:= 1 / W; S2:= p × h × h × L2;
                S3:= q × h × h × L1; S4:= q × h × h × L3;
                W:= (1 - S2 - 2 × S3 - S4) / S1;
                A:= S2 × h / S1; B:= (2 × S3 + S4) × h / S1;
                SEIDEL
        end;
        ABSFIXT(3,2, W); ABSFIXT(3,2,I); FIXT(2,2,A); FIXT(2,2,B);
        PRINTTEXT(ᚲ)ᚦ);
end;
```

```
CALC:  INPUT; INFORMATION; RESIDUAL;
ITER:  DIFF SCHEME; RESIDUAL;
       if k < K then goto ITER;
       if read > 0 then begin INFORMATION; goto ITER;end;
       if read > 0 then OUTPUT;
       N:= N + 1; k0:= k;
       for j:= 0 step 1 until N do for 1:= 0 step 1 until N do
          U0[j,1]:= u[j,1];
end;

BEGIN OF THE PROGRAM:

NLCR; PRINTTEXT(‹ resultaten vdh / 111166 / II / R1205.›);
NLCR; NLCR; PRINTTEXT(‹ getalbandnummer›); ABSFIXT(3,2,getbnr);
N:= read; K:= read; h:= 3.141592653589793 / N;
i:= read; NLCR; PRINTTEXT(‹ beginapproximatie nr›); ABSFIXT(2,0,i);
   for j:= 0 step 1 until N do for 1:= 0 step 1 until N do U0[j,1]:= F(i);
   i:= read; NLCR; PRINTTEXT(‹ inhomogene term    nr›); ABSFIXT(2,0,i);
   COMPUTE;
   if read > 0 then begin NEW PAGE; goto BEGIN end
  end
 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 444 pp.

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

[3] Houwen, P.J. van der [1967], "On the acceleration of Richardson's method I" (Theoretical part). Report TW 104, Math. Centrum, Amsterdam.

[4] Houwen, P.J. van der [1967], "On the acceleration of Richardson's method II" (Numerical aspects). Report TW 107, Math. Centrum, Amsterdam.

[5] Varga, Richard [1962], "Matrix iterative analysis". Prentice-Hall, Inc. Englewood-Cliffs, New Yersey.

[6] Vooren, A.I. van de and Vliegenthart A.C. [1967], "On the 9-point difference formula for Laplace's equation". Report TW 42, Math. Inst., University Groningen.

[7] Young, D.M. and Warlick C.H. [1953], "On the use of Richardson's method for the numerical solution of Laplace's equation on Ordvac". B.R.L. Memo Report No. 707, Aberdeen Proving Ground, Md.

[8] Kruseman Aretz, F.E.J. [1966], "Het MC-ALGOL 60-systeem voor de X8". Rapport MR 81, Math. Centrum, Amsterdam.