1976

Algorithmic Progress in Solving Partial Differential Equations

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
76-173

ALGORITHMIC PROGRESS IN SOLVING PARTIAL DIFFERENTIAL EQUATIONS

John R. Rice
Computer Science Department
Purdue University
Lafayette, Ind.
47907

CSD-TR 173

January 1, 1976

ALGORITHMIC PROGRESS IN SOLVING PARTIAL DIFFERENTIAL EQUATIONS

John R. Rice
Computer Science Department
Purdue University

January 1, 1976
CSD-TR 173

PREFACE

This is a working paper to attempt to document the progress that has been made over the past 30 years in computational methods for solving partial differential equations. Hopefully the assumptions made in this study are clear, but they may well be disputed. Comments or suggestions about them are welcome. Some computational efforts are quickly derived here, but others are taken from the literature without an independent check that the same definition of effort is used. Corrections of these estimates are invited. There are probably some computational methods which I have overlooked which should be included. Please let me know of your favorite (or your own) method which should be included (I would appreciate exact references and computational effort estimates). Note that this study shows up some obscure methods as being very attractive, which suggests that there are other lesser known methods which are also attractive.

The end purpose of this study is to present one or two simple graphs demonstrating the progress over the past 30 years. These graphs are to be included in the Numerical Computations section of the COSERS report.

0. BRIEF SUMMARY OF STUDY. We consider two model problems: the very special Poisson problem and a general, variable coefficients elliptic boundary value problem with curved boundaries. Both model problems have well-behaved solutions which are to be computed to .1% accuracy. We trace the various computational algorithms for these problems from 1945 to 1975 and then make some conjectures about future progress. The conclusions are summarized by the following table of gains in computational speed due to algorithm improvement.

	Estimated 1945-1975	Conjectured 1975-1985	1945-1985
Poisson Problem 2-Dimensions	800	3	2500
3-Dimensions	50,000	3	150,000
General Elliptic 2-Dimensions	360,000	500	180,000,000
Curved Boundaries 3-Dimensions	12,000,000,000	25,000	300,000,000,000,000

It is significant to note that for the general three dimensional problem and for both the periods 1945-75 and 1945-85 we have

The gain in speed from algorithm improvement

exceeds the gain in speed from hardware improvement

(i.e. from desk calculators to typical, large 1975 or 1985 computers)

1. METHOD AND ASSUMPTIONS OF THE STUDY.

We choose two model problems and trace the computational effort to solve them by various methods. The dates assigned are when the method was first published or, in some cases, when I considered the method to be feasible and natural to attempt.

Model Problem 1. $U_{xx} + U_{yy} = f(x,y)$ 2-Dimensions

$U_{xx} + U_{yy} + U_{zz} = f(x,y,z)$ 3-Dimensions

Dirichlet Boundary Conditions in the unit rectangle or cube

Accuracy of .001 (= .1%) required.

Model Problem 2. Let L be a general, non-separable, elliptic operator with variable coefficients. Let D be a simple domain with one or more curved boundaries.

$Lu = f(x,y)$ 2-Dimensions

$Lu = f(x,y,z)$ 3-Dimensions

Dirichlet Boundary Conditions on D.

Accuracy of .001 (= .1%) required.

Assumptions

(1) About the problem:

A. The solution u is smooth (several continuous derivatives) and of size 1. It is "simple" in the sense that there are very few oscillations in any cross-section. The coefficients of L behave similarly.

(2) About the methods:

A. The space discretization h determines the error in a consistant manner. Specifically, for a method of order p, the values of h and N = 1/h required for 10^{-3} accuracy are

p	1	2	3	4
h/N	.0125/800	.0333/30	.0833/12	.125/8

Note that if the error were h^p then the values of N would be 1000, 32, 10, 6, respectively so this assumption implies that the coefficient of h^p in the error increases some with p.

B. The computational effort is proportional to the number of multiplications required. The rate for performing multiplications is 100,000 per second. This assumption envisages an "average" 1975 large scale computer which performs 1 million operations per second and for every multiplication there are 9 other operations (adds, fetches, stores, etc.).

2. MODEL PROBLEM ONE - THE POISSON PROBLEM

The methods are introduced in chronological order after the set of three assumed to be initial conditions for the computer era.

2.1 1945 - Cramer's Rule, Gauss Elimination (Band), Gauss-Seidel Iteration

The standard second order finite difference discretization is used. Thus N = 30 in each case.

a. Cramer's Rule. The evaluation of determinants is done by expansion by minors taking into account the zero structure of the matrix. For 2-dimensions the matrix is block tridiagonal of size $N^2 \times N^2$. The expansion of the first N rows leads to $9 * 4^{N-2}$ submatrices of order $N(N-1) \times N(N-1)$ which are about the same nature as the original. Thus we estimate the total work to be $(9*4^{N-2})^N$ or, simplifying, 4^{N^2}. A similar argument suggests that 5^{N^3} is a reasonable estimate in 3-Dimensions.

b. Gauss-Elimination. The work to solve Ax = b for a matrix of order M and band width K is MK^2. For the model problem we have

2-Dimensions:	$M = N^2$, $K = N$	for	N^4 multiplies
3-Dimensions:	$M = N^3$, $K = N^2$	for	N^7 multiplies

c. Gauss-Seidel. See [Forsythe and Wasow (1960) p. 283]. The multiplies per iteration is four or six times the matrix order. The initial error is assumed to be 1 and is reduced by $1-h^2$ each iteration. The number r of iterations required thus satisfies

$$(1-h^2)^r = 10^{-3}$$

for both 2 and 3 dimensions. In terms of N this becomes

$$r = 3N^2 \sim 2N^2 \log N$$

The total number of multiplies is then

2-Dimensions:	$12N^4$	(N=30)
3-Dimensions:	$18N^5$	(N=30)

2.2 1954-SOR See [Young (1954)] and [Forsythe and Wasow (1960)]. The usual second order finite difference discretization is made which requires 5 or 7 multiplies per equation per iterations and gives N=30. The optimum relaxation parameter is used which reduces the error by 1-2h each iteration. If the initial error is 1 then the number r of iterations satisfies

$$(1-2h)^r = 10^{-3}$$

or, in terms of N,

$$r = 3N/2 \sim N\log N$$

The total number of multiplies is then

2-Dimensions	$7.5\ N^3$
3-Dimensions	$10.5\ N^4$

2.3 1955-ADI. See [Peaceman and Rachford (1955)] and [Forsythe and Wasow (1960)]. The usual second order finite difference approximation is used and gives N=30. The number of multiplications and divisions per iteration is $8N^2$ (see [Lynch, Rice and Thomas (1964) p. 194]) and the number r of iterations is $2.2\ \log^2 N$ for the two dimensional case. For three dimensions the work per iteration is $12\ N^3$ and we assume that the same number of iterations are required. We are unaware of an analysis which gives the number of iterations in three dimensions.

The totals are then

2-Dimensions	$18N^2\log^2 N$
3-Dimensions	$27N^3\log^2 N$

2.4 1956 - SOR for the 9-Point Star See [Garabedian (1956)] and [Forsythe and Wasow (1960) p. 266]. This difference approximation to the Laplacian is sixth order but for a low accuracy like 10^{-3} we assume that it behaves as if it were fourth order and thus we take N=8. The number of multiplies per equation per iteration is 8 or 26. The error is reduced by 1-2.04h per iteration which makes r satisfy

$$(1-2.04h)^r = 10^{-3}$$

or, in terms of N,

$$r = 3N/2 \sim N\log N$$

The total number of multiplies is then

2-Dimensions	$12N^3$
3-Dimensions	$39N^4$

2.5 1964 - Tensor Product See [Lynch, Rice and Thomas (1964)]. Both the 5-point and 9-point finite difference approximations are studied. We first consider the 5-point star which gives N=30. The number of multiplies is given for 2-Dimensions and for 3-Dimensions it is found in the same way. The result is

2-Dimensions	$2N^3$	(5-point star,N=30)
3-Dimensions	$2N^4$	(7-point star,N=30)

The number of multiplies for the 9-point star is shown to be the same as for the 5-point star (except for lower order terms). We again assume N=8 as though the 9-point star were fourth order rather than the actual sixth order.

2-Dimensions	$2N^3$	(9-point star, N=8)
3-Dimensions	$2N^4$	(9-point star, N=8)

2.6 1965 - ADI for 9-point star See [Lynch, Rice and Thomas (1965)]
The 9-point star approximation to the two dimensional Laplacian is analyzed as the basis of an ADI method. Again we take N=8 for this approximation. The multiplications and divisions per iteration is given as 16 N^2 and the number of iterations is 10.4 log2 N. The three dimensional case is not analyzed, but we assume the same number of iterations is required and the work per iteration is seen to be $24N^3$. The results are

2-Dimensions	$166\ N^2 \log_2 N$
3-Dimensions	$250\ N^3 \log_2 N$

A second ADI scheme is also analyzed which has a convergence rate somewhat faster than the more straight forward adaption of the 9-point formula to an ADI method. The work per iteration is doubled, but the number of iterations is reduced by a factor of about 4.33 for a net decrease of 2.16 in the multiplications. The results are

2-Dimensions	$77N^2 \log_2 N$
3-Dimensions	$116N^3 \log_2 N$

2.7 1965 - FFT Direct Solution See [Hochney (1965)] and [Dorr (1970)]. This method is based on the observation that the solution can be explicitly expressed as a finite Fourier series and then the Fast Fourier Transform can be used to sum the series. This is for the 5-point star so N=30 is used and the operations count is given for two dimensions. No count is given for three dimensions but it should be proportional to $N^3 \log_2 N$ and we assume the coefficient is 7. This results in

2-Dimensions	$5N^2 \log_2 N$
3-Dimensions	$7N^3 \log_2 N$

Presumably these counts are valid only for $N = 2^k$ and thus we adjust N to be 32 for this method.

Certain numerical instability problems for this method were not solved until about 1969 by Buneman, but they could be controlled for moderate N by using long words. See [Buzbee, Golub and Nielson (1970)].

2.8 1968 - ADI for 9-point Star with Smooth Initial Guess See [Lynch and Rice (1968)]. This is identical to the method of 2.6 except that the initial guess is chosen to make the initial error smooth and 4 corresponding ADI parameters are chosen so that only 3 iterations are required to reduce the error by 10^{-3}. The resulting multiplications are

2-Dimensions	$71\ N^2 \log^2 N$
3-Dimensions	$107\ N^3 \log^2 N$

for the first mentioned method of 2.6 and for the second we have

2-Dimensions	$33\ N^2 \log^2 N$
3-Dimensions	$50\ N^3 \log^2 N$

No analysis has been done for the three dimensional case and we have assumed that the number of iterations is the same as in two dimensions.

2.9 1970→1974 - Cyclic Reduction. See [Buzbee, Golub and Nielson (1970)], [Sweet (1974)] and [Swarztrauber (1974)]. This is a general "divide and conquer" method analogous to the idea behind the Fast Fourier Transform. It is applicable to matrices which are tensor products of a certain type i.e. block tridiagonal with constant matrices on the diagonals. The first paper used $N = 2^k$ for the reduction while the later ones use other prime factors (especially 3 and 5) for N in the manner similar to extensions of the FFT. It is applied to the 5-point star and the operation count in 2-Dimensions is given (without derivation) by [Dorr (1970)] as

2-Dimensions	$9/2\ N^2 \log_2 N$
3-Dimensions	$7\ \ N^3 \log_2 N$

We assume the 3-Dimension count given here is the correct extension, no analysis has been made to justify this. We take N=32 here.

The later extensions allow factor of 3 and 5 so that N=30 may be used. However, the execution time is said to increase by about 30% which results in the following work estimates.

2-Dimensions	$6\ N^2 \log_2 N$
3-Dimensions	$9\ N^3 \log_2 N$

2.10 Tabulated Summary for Model Problem One. The previous sections listed numerous methods and they fall into three categories: (1) widely used and tested methods (2) methods which have been theoretically analyzed but not tested in practice (3) methods where we are conjecturing the multiplication counts required. The entries from class (2) are marked with one asterisk (*) and those from class (3) by two asterisks (**) in the table below. The entries are obtained by merely substituting in the appropriate values for N.

2.11 Remarks on Future Developments. It seems plausible that the fastest methods shown can be put into practice in a stable and efficient manner. Furthermore, an examination of these techniques suggests that a factor of 2 or 3 can yet to be made without discovering essentially new methods. This implies that within 5 to 10 years one will be able to solve this model problem with 300-500 multiplications in 2-Dimensions and 2500-4000 multiplications in three dimensions. At this point one is probably reasonably close to the ultimate barrier implied by the computational complexity of this problem. Note that we are suggesting that this is a simpler problem than normally believed.

Within 10 years it is likely that architectural advances will significantly impact this problem and result is an additional increase in speed of 10 or 50 for these two problems. Indeed special chips may well be feasible for such common problems.

The total gain in speed from algorithm improvement from 1945 to the present is a factor 800 + for the two dimensional case and 53000 for the three dimensional case. We are projecting further gains of 6-10 and 10-15 for algorithm improvements in the future.

Year	Method	N	2-Dimensions Multiplies		2-Dimensions Time	3-Dimensions Multiplies		3-Dimensions Time	3-Dimensions Storage
1945	Cramer's Rule	30	10^{542}	*	10^{530} years	$10^{18,900}$	*	$10^{18,900}$ years	—
	Gauss-Seidel Iteration	30	9,700,000		97 sec	437,000,000		1.2 hours	190,000
	Gauss Elimination (Band)	30	810,000		8 sec	2.2^{+10}		2.5 days	25,000,000
1954	SOR	30	202,500		2 sec	8,500,000		85 sec	190,000
1955	ADI	30	35,000		.35 sec	1,600,00	**	16 sec	190,000
1956	SOR for 9-Point Star	8	6144	*	.06 sec	160,000	*	1.6 sec	15,000
1964	Tensor Product, 5-Point Star	30	54,000		.54 sec	1,600,000		16 sec	80,000
	Tensor Product, 9-Point Star	8	1024	*	.01 sec	8200	*	.08 sec	1,500
1965	ADI for 9-Point Star, #2	8	4019	*	.04 sec	48,000	**	.48 sec	1,000
	FFT Solution, 5-Point Star	32	15,000		.15 sec	628,000		6 sec	32,000
1968	ADI, 9-Point Star, Smooth start	8	1720	*	.017 sec	20,600	**	.2 sec	1,000
1970	Cyclic Reduction	30	23,000		.23 sec	1,150,000	**	11 sec	27,000

Table 1. Multiplication counts and estimated execution times for Model Problem 1 (the Poisson problem) and various computational methods. An estimate of the number of words of working storage is given for the 3-dimensional case.

3. MODEL PROBLEM TWO - GENERAL LINEAR ELLIPTIC PROBLEM

The methods are in chronological order. Some of them are "hypothetical" in the following sense. It seems to me (and clearly this must be a subjective thing) that the state of the art was such that such computations could have been tried and they were probably successfully tried by someone.

To be specific, we assume that the region is such that the band width is 50% larger than if the region were a rectangle and we assume that the same number of points is used as if it were a rectangle.

3.1 1945 - Simple differences and Gauss Elimination. The basic method was to use second order differences and simply modify them near the boundary. The literature did contain some second order methods for Dirichlet boundary conditions, but we assume that it was very unlikely that they would have been used (indeed some of them were probably never used in real computations). Thus we assume that a first order method is used at the boundary. For low accuracy requirements this makes the entire computation first order. Thus we have N=800 and the formulas for Gauss elimination are

2-Dimensions	$N^2 * (1.5N)^2$
3-Dimensions	$N^3 * (1.5N^2)^2$

3.2 1955 - Simple Differences and Iterative Methods. This is a hypothetical case. We still assume simple differences and first order approximations to the problem because of the curved boundaries. We assume that over-relaxation is used with some effectiveness, specifically, the rate of convergence is as good as Gauss-Seidel for Model Problem 1. Thus we have N=800 still and the number of iterations is assumed to satisfy

2-Dimensions	$(1 - h^2)^r = 10^{-3}$
3-Dimensions	$(1 - 4h^2)^r = 10^{-3}$

These are solved for r in terms of N. We further assume that the irregular nature of the system of equations doubles the work per iteration compared to Model Problem 1. The net results are

2-Dimensions	$10N^2 * 3N^2$
3-Dimensions	$14N^3 * 12N^2$

3.3 1960 - Better Approximations to the Boundary Conditions. This is a hypothetical case. It was probably realized in two ways. First, more complicated finite difference expressions were used for curved boundaries to give an overall second order method. Second, finite element methods (based on piecewise linear elements) were used in structural engineering problems which also gave a second order method. Thus we assume that N=30 suffices and that the resulting system of linear equations was solved directly by Gauss elimination. We further assume that the bandwidth is 3N/2 and that the formation of the system of equations requires 75% of the total work. The results are then

2-Dimensions	$4 * N^2 * (1.5N)^2$
3-Dimensions	$4 * N^3 * (1.5N^2)^2$

3.4 1965 - Iterative methods for general matrices. This is a hypothetical case. We assume that an iterative technique is used for the matrices generated with the 1960 method. It is assumed that the convergence obtained is somewhat better than Gauss-Seidel for Model Problem 1, that is the number r of iterations satisfies, for 2 and 3 dimensions, respectively

$$(1 - h^{7/4})^r = 10^{-3} \quad , \quad (1 - 4h^{7/4})^r = 10^{-3}$$

Thus we have N=30 and the formulas are

2-Dimensions: $4 * 5N^2 * 3N^{7/4}$

3-Dimensions: $4 * 7N^3 * 12N^{7/4}$

3.5 1970 - Galerkin with Hermite cubics. This is a hypothetical case. We assume that the problem of curved boundaries here was solved by approximating them by piecewise polynomials and, further, that the inaccuracy in this approximation reduced this potentially fourth order method's accuracy so that N=12 (rather than N=8) is required. Note that this method has four unknowns per element so the orders of the matrices are $16N^2$ and $64N^3$, respectively. We further assume that the work to form the equations constitutes 90% of the total work. Gauss eliminations is used to solve the resulting system of linear equations and the formulas are

2-Dimensions: $10 * 4N^2 * (3N)^2$

3-Dimensions: $10 * 8N^3 * (6N^2)^2$

3.6 1975 - Collocation with Hermite cubics. This is a hypothetical case. The curved boundaries are handled as in 1970, but the linear equations are found from the tensor product of the collocation method with Gauss points. See [deBoor and Swartz, 1973]. The effect of this is to reduce the work of forming the equations from 90% to 65% of the total. This changes the factor of 10 in the 1970 formulas to a 3:

$$\text{2-Dimensions:} \qquad 3 * 16N^2 * (6N)^2$$
$$\text{3-Dimensions:} \qquad 3 * 64N^3 * (24N^2)^2$$

3.7 1976 - The HODIE method. This is a hypothetical case. This (as yet experimental) method allows one to accommodate curved boundaries in a fourth order method with negligible added overhead. See [Lynch and Rice, 1976]. Thus we have N=8. For a rectangular domain the overhead to form the equations is 75% for N=8 and we assume an additional factor of two for the curved boundaries. The order of the linear system are N^2 or N^3, respectively, and we assume they are solved by Gauss elimination. The resulting formulas are

$$\text{2-Dimensions:} \qquad 8 * N^2 * (1.5N)^2$$
$$\text{3-Dimensions:} \qquad 8 * N^3 * (1.5N^2)^2$$

3.8 Tabulated Summary for Model Problem Two. Since this model problem actually refers to a broad class of problems, there is considerable difficulty in tracing its history. Clearly there will be examples in or nearly in this class which make the estimates given here grossly optimistic or pessimistic. We have tried to keep our attention on that very rare creature, the "typical" problem. The entries in Table 2 are obtained by merely substituting in the appropriate values of N.

			2-Dimensions		3-Dimensions		
Year	Method	N	Multiplies	Time	Multiplies	Time	Storage
1945	Differences and Gauss Elimination	800	9^{+11}	80 days	5^{+20}	2^{+8} years	8^{+8}
1955	Differences and iteration	800	1^{+13}	3 years	5^{+16}	15,000 years	5^{+6}
1960	Better boundary approximations	30	7,300,000	73 sec	2^{+11}	22 days	5^{+7}
1965	Iteration for 1960 method	30	20,000,000	200 sec	4^{+10}	5 days	300,000
1970	Galerkin with Hermite cubics	12	7,500,000	75 sec	1^{+11}	11 days	170,000
1975	Collocation with Hermite cubics	12	2,500,000	25 sec	3^{+10}	4 days	170,000
1976	HODIE	8	75,000	1 sec	4^{+7}	6 min	75,000

Table 2. Multiplications counts and estimated execution times for Model Problem 2 and various computational methods. An estimate of the number of words of working memory is given for the 3-dimensional case.

The gain in speed from algorithm improvement from 1945 to 1975 is very impressive: 360,000 in two dimensions and 12 billion in three dimensions. Much, if not most, of this comes from using second order methods, with techniques that were latent in 1945 but not implemented (even hypothetically) until 15 years later. Unfortunately, it is difficult to "prove" that this amount of progress has been made. The actual progress depends on the particular problem at hand and the "typical" problem envisaged in this study is rare indeed. Some recent discussions of the evaluation of methods appears in [Fix and Larsen, 1971], [Birkhoff and Fix, 1974] and [Houstis, Lynch, Papatheodorou and Rice, 1975]. It should be noted that there are currently many practitioners who doubt the advantage of second order methods over first order methods for complex problems (note that our Model Problem 2 does not really qualify as complex). Needless to say, there are even more who are skeptical of the value of fourth order methods. There is a clear need for more concrete quantitative data on the performance of these methods.

It is significant to note for this three dimensional model problem:

<u>The gain in speed from algorithm improvement exceeds the gain in speed from hardware advances (i.e. from desk calculators to a IBM 360/65)</u>

3.9 Remarks on Future Developments. The new HODIE method promises factors of 25 and 750 in the gains for speed and it has yet to be perfected. Further increases by factors of 3 to 5 should be expected merely from perfecting such methods. Furthermore, we should expect iterative methods and adaptive methods to have a large eventual impact. Within 10 years the multiply counts may well be down to 5,000 for two dimensional problems and 1,500,000 for three dimensional problems. This represents total projected gains of 500 and 20,000 for algorithm improvements in the future. Furthermore, these gains in speed will be accompanied by even more significant improvements in reliability.

REFERENCES

1. Birkhoff, Garrett (1971) The numerical solution of elliptic equations, CBMS 1, SIAM.

2. Birkhoff, G. and Fix, G. (1974) Higher order linear finite element methods A report of work supported by the AEC and ONR, 33 pages.

3. Buzbee, B.L., Golub, G.H. and Nielson, C.W. (1971) On direct methods for solving Poisson's equations. SIAM J. Numer. Anal. 7 pp. 627-656.

4. deBoor, C.W. and Swartz, B. (1973) Collocation at Gaussian points, SIAM J. Numer. Anal. 10 pp. 582-606.

5. Dorr, F.W. (1970) The direct solution of the discrete Poisson problem on a rectangle. SIAM Review, 12 pp. 248-263.

6. Fix, G. and Larsen, K. (1971) The convergence of SOR iterations for partial differential equations, SIAM J. Numer. Anal. 8 pp. 536-547.

7. Forsythe, G.E. and Wasow, W.R. (1960) Finite-Difference Methods for Partial Differential Equations, John Wiley.

8. Garabedian, P.R. (1956) Estimation of the relaxation factor for small mesh size, MTAC, 10 pp. 183-185.

9. Hochney, R.W. (1965) A fast direct solution of Poisson's equation using Fourier analysis, J. Assoc. Comp. Mach. 12 pp. 95-113.

10. Houstis, E.N., Lynch, R.E., Papatheodorou, T.S. and Rice, J.R. (1975) Development, evaluation and selection of methods for elliptic partial differential equations, Ann. Assoc. Int. Calcul Anal. 11, pp. 98-103.

11. Lynch, R.E., Rice, J.R. and Thomas, D.H. (1964) Direct solution of partial difference equations by tensor product methods. Numer. Math. 6 pp. 185-199.

12. Lynch, R.E., Rice, J.R. and Thomas, D.H. (1965) Tensor product analysis of alternating direction implicit methods, J. Soc. Indust. Appl. Math., 13 pp. 995-1006.

13. Lynch, R.E. and Rice, J.R. (1976) The HODIE method for elliptic partial differential equations.

14. Peaceman, D.W. and Rachford, H.H. (1955) The numerical solution of parabolic and elliptic differential equations, J. Soc. Indust. Appl. Math., 3 pp. 28-41.

15. Swarztrauber, P.N. (1974) A direct method for the discrete solution of separable elliptic equations, SIAM J. Numer. Anal. 11 pp. 1136-1150.

16. Sweet, R.A. (1974) A generalized cyclic reduction algorithm, SIAM J. Numer. Anal. 11 pp. 506-520.

17. Young, D. (1954) Iterative methods for solving partial difference equations of elliptic type, Trans. Amer. Math. Soc. 76 pp. 92-111.