数理解析研究所講究録
91巻 1970年 29-81

Acceleration of the Relaxation Method by an Integro-Differential Process

by Eiichi Goto

Department of Physics, University of Tokyo,
Bunkyo-ku, Tokyo, JAPAN
and
The Institute of Physical and Chemical Research,
Yamatomachi, Saitama, JAPAN

Abstract: A new process for accelerating the convergence of the relaxation method by making use of numerical integrations together with numerical differentiations is proposed. While the conventional S.O.R.(successive over relaxation) requires about $0.36 \cdot N \cdot D$ sweeps to solve Poisson's equation in a square to the accuracy of D decimals by dividing it into N×N meshes, the new process requires only $1.07(D+0.6)(\log_{10}N+0.1)$ sweeps. Although the computation needed per sweep in the new process becomes several times more than that in S.O.R., the total amount of computation decreases considerably for large N.

§1 Introduction and Heuristic Remarks

The convergence rate of the relaxation method, which is widely used for the numerical solution of elliptic partial differential equations such as Poisson's equation, has been considerably improved by the development of S.O.R.(successive over relaxation) schemes.[1)]

---

1) D.J. Evans: "Estimation of the line over-relaxation factor and convergence rates of an alternating direction line over-relaxation technique", The Computer Journal, Vol.7, pp.318-321 (1964)

Nevertheless, there still exist a great number of computational problems in which the solution of Poisson's equations or the similar form bottle necks in the computational procedure. Further acceleration of the relaxation procedure is highly desirable, therefore, for many problems. In the following treatment, for the sake of simplicity and clarity, we shall focus our attention to the relaxation solution of the Poisson's equation

$\Delta \phi(x,y)=\rho(x,y)$ in a unit square, $0\leq x\leq 1$, $0\leq y\leq 1$ and subjected to the condition $\phi=0$ at the boundary.

In the relaxation method the square is divided into N×N meshes of size $h=1/N$, and the Poisson's equation is approximated by the finite difference equation

$$\Delta^{*}\phi=(\phi_{i+1,j}+\phi_{i-1,j}+\phi_{i,j+1}+\phi_{i,j-1}-4\phi_{i,j})/h^{2}=\rho_{i,j}. \quad ...(1)$$

The relaxation process starts from an initial function $\phi^{(0}_{i,j}$ and the result $\phi^{(s}_{i,j}$ of the s'th sweep is computed according to

$$\phi^{(s}_{i,j} = \phi^{(s-1}_{i,j} + A_s(\Delta^{*}\phi^{(s,s-1} - \rho_{i,j}), \quad ...(2)$$

where i and j are made to run from 1 to N-1 during a single sweep, the superscript (s,s-1 means to use the new value $\phi^{(s}_{i,j}$ for evaluating (1) if it has already been computed and A_1, A_2, A_3 ... are series of numbers called the successive relaxation factors.

In the classical relaxation method, the same constant $h^2/4$ is used for the relaxation factors throughout the entire relaxation process. In this classical case, the error can be

expected to decrease only by a factor of $\exp(-E_{1,1}^2/N^2)$ per sweep in the average, where $E_{1,1}^2$ is the minimum eigenvalue of the eigenvalue problem

$$\Delta\phi + E_{1,1}^2\phi = 0 \quad \ldots(3)$$

and $\phi = 0$ on the boundary square, and specially $E_{1,1}^2=2\pi^2$ in the present case. Hence, in case the error in the initial function $\phi^{(0}$ is required to be suppressed by a factor of 10^{-D} or by a factor of D decimal figures, the total number S of sweeps needed is given by

$$S = \log_e 10 \cdot D \cdot N^2/E_{1,1}^2 = 0.117 \cdot D \cdot N^2. \quad \ldots(4)$$

In the S.O.R., the S.O.R. factors A_s's are made to change in such a way as to increase the rate of convergence. A suitable choice of the S.O.R. factors is known to give a convergence factor of $\exp(-\sqrt{2}E_{1,1}/N)$ per sweep. Hence, the number S of sweeps needed to suppress the error by a factor of D decimals is given by

$$S = \log_e 10 \cdot D \cdot N/\sqrt{2}E_{1,1} = 0.3665 \cdot D \cdot N, \quad \ldots(5)$$

which is much less than that in the classical method. The amount of computation needed to solve Poisson's equation as well as similar problems has thus been reduced considerably by the development of the S.O.R. method.

The nature of the convergence rate of the relaxation processes can be understood intuitively by tracing the diffusion of errors. Let us suppose that the initial function $\phi^{(0}_{i,j}$ has a delta function like error at (i',j'), namely,

$$\Delta^*\phi^{(0}_{i,j} - \rho_{i,j} = h^{-2}\delta^{i'}_{i}\delta^{j'}_{j}. \quad ...(6)$$

By tracing, numerically or analytically, the behaviour of the errors in the successive iterants, which are the values of $\Delta^*\phi^{(s}_{i,j} - \rho_{i,j}$, the relaxation process will be found to be nothing but a diffusion process of the errors towards the boundary where they are absorbed and disappeared by virtue of the boundary condition. In the classical case, the errors are made to diffuse by one mesh unit $h=1/N$ per sweep. Since the mean distance of diffusion increases only as the square

root of the time in diffusion processes, about N^2 sweeps will be needed to make a considerable portion of the error to diffuse through N meshes so that they may be absorbed at the boundary. This argument clearly indicates the reason why N^2 apears in the convergence rate formula (4) in the classical case.

In the optimum S.O.R., the errors are made to diffuse by $\sqrt{N}$ mesh units per sweep in the average, which results in the apearance of N instead of N^2 in the rate formula (5). The situation is thus greatly improved in the S.O.R.. Nevertheless, about N sweeps are still needed to make a considerable portion of the errors diffuse to the boundary. If, therefore, a process which enforces the errors to be diffused to the boundary in fewer sweeps could be deviced, further acceleration of the relaxation method would be possible.

The slowness of the convergence of the conventional

relaxation schemes may be ascribed to the local nature of the difference operator Δ^* of (1) used in the successive correction scheme (2). Since the operator Δ^* is the finite difference analogue of the Laplacian differential operator Δ, it deals only with nearest neighbouring sites. The corrections of errors in the relaxation process (2), therefore, can only be local in its nature, while the diffusion of errors, which is the heart of the relaxation schemes, are essentially global in its nature. This situation may be improved by making use of global operators such as the finite difference analogue of integration operators. For example, consider the integration of the following differential equation

$$df/dx + af = \delta(x-x'), \quad \ldots(7)$$

starting from x=0 and df/dx=0.

The result is given by

$$f(x)=0 \text{ for } x<x' \text{ and } f(x)=\exp(-a(x-x')) \text{ for } x>x' \quad \ldots(8)$$

Equation (8) indicates that the effect of a delta function sited at the locality x' propagates to the +x direction to all x greater than x'. Similarly, by starting the integration of (7) from x=1 and proceeding to the -x direction, the effect can be made to propagate in the -x direction. The situation will be essentially the same even if the integration of (7) is replaced by its finite difference analogue.

Guided by heuristics as described above, a new relaxation process of the following form was deviced:

$$\phi^{(s}_{i,j} = \phi^{(s-1}_{i,j} + \overset{*}{L}(A_s)(\Delta^*\phi^{(s-1}_{i,j} - \rho_{i,j}), \quad \ldots(9)$$

in which $L(A_s)$ is no longer a constant as in the conventional relaxation scheme (2) but consists of the combination of the finite difference analogues of integration operations of the type (7). $\overset{*}{L}(A_s)$ also includes a series of parameters A_s which are closely related to the integration parameter a of (7). The new relaxation process of the type (9)

will be called S.I.D.R.(Successive Integro-Differential Relaxation) hereinafter. The number of sweeps needed in the S.I.D.R. will be shown to be proportional to $\log N$ instead of N in the S.O.R., Thus, a considerable reduction in the amount of computation will be achieved especially in case of large N.

§2 The S.I.D.R. Algorithm

Before describing the finite difference algorithm for the S.I.D.R., analytical formulas corresponding to the case of infinitesimal h and infinite N will be given, because they are simpler and more tractable than the finite difference formulas.

We now consider an iteration scheme of the following form

$$\phi^{(s}(x,y)=\phi^{(s-1}(x,y)+L(A_s)(\Delta\phi^{(s-1}(x,y)-\rho(x,y)), \quad ...(10)$$

in which $L(A_s)$'s are linear operators including positive parameters A_1, A_2, A_3 ... to be called S.I.D.R. factors. The linear operator L(A) with a positive parameter A is defined to be the result of the following operations which map $W_0(x,y)$ into $W_6(x,y)$:

Let a be defined by

$$a=\sqrt{A}; \quad ... (11)$$

Regarding y as a parameter independent of x, integrate

$$(\frac{d}{dx}+a)W_1(x,y)=W_0(x,y) \quad ...(12)$$

from x=0 and $W_1(0,y)=0$ in +x direction until x=1 to obtain

$W_1(x,y)$;

Integrate $\left(-\frac{d}{dx}+a\right)W_2(x,y)=W_0(x,y)$ ....(13)

from x=1 and $W_2(1,y)=-W_1(1,y)$ in -x direction until x=0;

$$W_3(x,y)=W_1(x,y)+W_2(x,y)-W_2(0,y)\sinh(a(1-x))/\sinh(a); \quad ...(14)$$

Integrate $\left(\frac{d}{dy}+a\right)W_4(x,y)=W_3(x,y)$ ...(15)

from y=0 and $W_4(x,0)=0$ in +y direction until y=1;

Integrate $\left(-\frac{d}{dy}+a\right)W_5(x,y)=W_3(x,y)$ ...(16)

from y=0 and $W_5(x,1)=-W_4(x,1)$ in -y direction until y=0;

$$W_6(x,y)=L(A)W_0(x,y)$$

$$=\frac{1}{2}\left(W_4(x,y)+W_5(x,y)-W_5(x,0)\sinh(a(1-y))/\sinh(a)\right). \quad ...(17)$$

Let $\psi^{(s}(x,y)$ denote the deviation of the s'th approximate function $\phi^{(s}(x,y)$ from the true solution $\phi(x,y)$. Namely,

$$\psi^{(s}(x,y)=\phi^{(s}(x,y)-\phi(x,y). \quad ...(18)$$

In terms of deviation $\psi(x,y)$, (18) can be rewritten as

$$\psi^{(s}(x,y)=\psi^{(s-1}(x,y)+L(A_s)\Delta\psi^{(s-1}(x,y). \quad ...(19)$$

We shall make use of the following eigenfunction expansion

$$\psi^{(s}(x,y)=\Sigma_{m,n}c^{(s}_{m,n}\Psi_{m,n}(x,y), \quad ...(20)$$

where $\Psi_{m,n}(x,y)$ are the eigenfunctions of the eigenvalue problem

$(\Delta+E^2_{m,n})\Psi_{m,n}(x,y)=0$ and $\Psi_{m,n}(x,y)=0$ at the boundary. (21)

Specially for our present case of a unit square, the eigenvectors and the eigenvalues $E^2_{m,n}$'s are given by:

$\Psi_{m,n}(x,y)=\sin(m\pi x)\sin(n\pi y)$, ...(22)

$E^2_{m,n}=E^2_m+E^2_n=m^2\pi^2+n^2\pi^2$ and $E_m=m\pi$, (23)

where m and n are positive integers.

From the construction of L(A), the eigenfunctions $\Psi_{m,n}(x,y)$ of the Lapalacian are simultaneously the eigenfunctions of L(A) and the eigenvalue $Q_{m,n}$ of L(A) is given by

$Q_{m,n}=2A/((E^2_m+A)(E^2_n+A))$. ...(24)

Hence, the eigenfunction expansion coefficients $c^{(s}_{m,n}$ of the s'th deviation $\psi^{(s}$ are readily obtained in terms of the eigenfunction expansion coefficients $c^{(0}_{m,n}$ of the initial deviation $\psi^{(0}$ as

$$c^{(s}_{m,n}=c^{(0}_{m,n}\prod_{t=1}^{s}\frac{(E^2_m-A_t)(E^2_n-A_t)}{(E^2_m+A_t)(E^2_n+A_t)}. \quad ...(25)$$

Now proceeding to the finite difference case, $L^*(A)$, which is the finite difference analogue of the operator L(A) used in (10) and which is to be used in the iteration scheme of the form (9), is defined to be the result of the following operations mapping $W_{0;i,j}$ into $W_{6;i,j}$.

Let a^* be defined by

$$a^*=\sqrt{A+A^2h^2/4}\ -Ah/2; \qquad ...(26)$$

Corresponding to (12) integrate numerically the following equation with a scale factor U

$$(W_{1;i,j}-W_{1;i-1,j})/h+a^*W_{1;i-1,j}=UW_{0;i,j}$$

or $$W_{1;i,j}=(1-a^*h)W_{1;i-1,j}+hUW_{0;i,j} \qquad ...(27)$$

starting from $W_{1;0,j}=0$ successively for i=1,2,3... until i=N;

Integrate $$W_{2;i,j}=(1-a^*h)W_{2;i+1,j}+hUW_{0;i,j} \qquad ...(28)$$

starting from $W_{2;N,j}=-W_{1;N,j}$ successively for i=N-1,N-2... until i=0;

$$W_{3;i,j}=W_{1;i,j}+W_{2;i,j}-hUW_{0;i,j}-W_{2;0,j}\cdot f(a^*,1), \qquad ...(29)$$

where $f(a^*,i)=\dfrac{(1-a^*h)^{N-i}-(1-a^*h)^{i-N}}{(1-a^*h)^{N}-(1-a^*h)^{-N}}$; ...(30)

After integrating (27), (28) and (29) for j=1,2,3...N-1, integrate

$W_{4;i,j}=(1-a^*h)W_{4;i,j}+hUW_{3;i,j}$ ...(31)

starting from $W_{4;i,0}=0$ successively for j=1,2,3... until j=N;

Integrate $W_{5;i,j}=(1-a^*h)W_{5;i,j+1}+hUW_{3;i,j}$ ...(32)

starting from $W_{5;i,N}=-W_{4;i,N}$ successively

for j=N-1,N-2... until j=0;

$W_{6;i,j}=L^*(A)W_{0;i,j}$

$=\dfrac{V}{2}(W_{4;i,j}+W_{5;i,j}-hUW_{3;i,j}-W_{5;i,0}f(a^*,j))$, ...(33)

where V is a correction factor given by

$V=\dfrac{1}{(1+Ah^2/4)\cdot U^2}$ . ...(34)

The following formulas corresponding to formulas (18) to (23)

will be selfexplanatory.

Deviation: $\psi^{(s}_{i,j}=\phi^{(s}_{i,j}-\phi_{i,j}$. ...(35)

The equations satisfied by the deviations:

$\psi^{(s}_{i,j}=\psi^{(s-1}_{i,j}+L^*(A_s)\Delta^*\psi^{(s-1}_{i,j}$. ...(36)

Eigenvector expansion: $\psi^{(s}_{i,j}=\Sigma_{m,n}c^{*(s}_{m,n}\Psi_{m,n;i,j}$ ...(37)

Eigenvalue problem: $(\Delta^*+E^2_{m,n})\Psi_{m,n;i,j}=0$ ...(38)

and $\Psi_{m,n;i,j}=0$ at the boundary.

Eigenvectors: $\Psi_{m,n;i,j}=\sin(m\pi i/N)\sin(n\pi j/N)$. ...(39)

Eigenvalues: $E^{*2}_{m,n}=E^{*2}_m+E^{*2}_n=4N^2(\sin^2(m\pi/2N)+\sin^2(n\pi/2N))$

and $E^{*2}_m=4N^2\sin^2(m\pi/2N)$, ...(40)

where m and n are positive integers less than N.

The eigenvectors $\Psi_{m,n;i,j}$ of the finite difference Laplacian Δ^* are simultaneously the eigenvectors of $L^*(A)$ and the eigenvalues $Q^*_{m,n}$ of $L^*(A)$ are given by

$Q^*_{m,n}=2A/((E^{*2}_n+A)(E^{*2}_n+A))$, ...(41)

which has exactly the same form as (24).

In exactly the same way as in (25), the eigenvector expansion coefficients $c^{*(s}_{m,n}$ of the s'th deviation $\psi^{(s}_{i,j}$ is readily obtained in terms of the eigenvector coefficients $c^{*(0}_{m,n}$ of the initial deviation $\psi^{(0}$ as

$$c^{*(s}_{m,n}=c^{*(0}_{m,n}\prod_{t=1}^{s}\frac{(E^{*2}_m-A_t)(E^{*2}_n-A_t)}{(E^{*2}_m+A_t)(E^{*2}_n+A_t)}. \quad ...(42)$$

It will be immediately noticed from (42) that the S.I.D.R.

converges in case all S.I.D.R. factors A_s's are fixed to a single positive constant A, since the absolute magnitude of the function $(z-A)/(z+A)$ is less than unity in case z and A are both positive. Better schemes for faster convergence will be treated in the section 3.

We shall now estimate the amount of computation in each sweep of the S.I.D.R. algorithm and compare it with that of the S.O.R.. Since the amount of computation greatly depends upon the hardware and software of the computer to be used, only a very rough estimate will be meaningful as a machine independent measure. In view of the single step operation of the S.O.R. of (9), the part of computation time in a single sweep which is proportional to the meshes N^2 in the S.O.R. will be something like

$$T=N^2(6t_a+2t_m+t_s+6t_1), \quad \ldots(43)$$

where t_a is the time to set a number from the memory in the multiplicand register or to add it into or substract it from

the accumulator, t_m is the multiplication time, t_s is the time to store a number into the memory and t_i is the time to set the address of a new array element in the index register. In S.I.D.R., the time needed for the evaluation of Laplacian and substraction of ρ will be almost the same as (43) except the number of multiplications, one per mesh point instead of two. In view of formulas (27) to (34), the best choice of the scale factor U is U=N=1/h, which reduces about $6N^2$ multications per sweep. The stepwise numerical integrations (27), (28), (31) and (32) will require $4N^2(t_a+t_m+t_s+2t_i)$ of time per sweep. (29) will require $N^2(3t_a+t_m+t_s+2t_i)$ of time provided that the function f(a,i) of (30) is tabulated as an array. (33) combimed with the addition in (9) together will require $N^2(5t_a+2t_m+t_s+2t_i)$ of time. Therefore, the total time per sweep will be given by

$$T'=N^2(18t_a+8t_m+7t_s+18t_i). \quad ...(44)$$

Let R_M denote the ratio of the two times T'/T evaluated

by taking the multiplication time only into account and R denote the ratio evaluated by using a somewhat more realistic assumption:

$t_a=t_s=t_1$ and $t_m=4t_a$. ...(45)

From (43) and (44), these ratios become $R_M=8/2=4$ and $R=75/21=3.57$. If it had not been for the optimum choice of the scale factor U, the ratios would have been $R_M=(8+8)/2=8$ and $R(75+32)/21=5.09$.

§3 Optimization of the Convergence of the S.I.D.R.

By convergence of S.I.D.R. we shall require that after S iterations all of the eigenvector expansion coefficients $c^{*(S}_{m,n}$ of the S'th deviation be suppressed by a factor of D decimals or by a factor $\bar{k}$ in comparison with the initial coefficients $c^{*(0}_{m,n}$. Namely,

$$\left|c^{*(S}_{m,n}\right| \leq \left|c^{*(0}_{m,n}\right| \bar{k} \quad \ldots(46)$$

for $\bar{k}=10^{-D}$ and for all m and n.

We define a function $F_S(z)$ by

$$F_S(z)=\prod_{s=1}^{s=S}\left(\frac{z-A_s}{z+A_s}\right), \quad \ldots(47)$$

where A_s's are the S.I.D.R. factors. In view of (42), the requirement (46) can be rewritten as

$$(F_S(z))^2=F_S^2(z)\leq\bar{k} \quad \ldots(48)$$

within the interval $0\leq kB_S=B_0\leq z\leq B_S$, $\ldots(49)$

in which B_0 and B_S are lower and upper bounds of E^{*2}_m's of (40) and k is defined to be the ratio of the two bounds: $k=B_0/B_S\leq 1$. In the present case of a unit square,

the best choices for the bounds and k are

$$B_0=E_1^{*2},\quad B_S=E_{N-1}^{*2}\quad\text{and}\quad k=E_{N-1}^{*2}/E_1^{*2}=\tan^2(\pi/2N)\geq\pi^2/4N^2. \qquad ...(50)$$

The optimization of convergence of the S.I.D.R., therefore, reduces to the problem of finding a set of S.I.D.R. factors, for given k and $\bar{k}$, which minimizes the total number of sweeps under the requirements (48) and (49).

Not the optimum but a reasonably good convergence can be obtained by the following very simple method which may be called the octave method. Starting from $A_1=B_0$, we place one A per octave until the b'th one A_b exceeds B_S. Namely,

$$A_1=B_0,\quad A_2=2B_0,\quad \ldots\quad A_b=2^bB_0\geq B_S=B_0/k. \qquad ...(51)$$

Defining $f_b(z)$ by

$$f_b(z)=\prod((z-A_1)/(z+A_1))\ldots((z-A_b)/(z+A_b)), \qquad ...(52)$$

and using the fact that $(z-A)/(z+A)\leq 1/3$ for $A/2\leq z<2A$, i.e., for z within an octave from A, we readily obtain $|f_b(z)|\leq(1/3)^2$ within the interval $B_0\leq z\leq B_S$, since there are two A's within an octave from any z in the interval (49).

Using the set of A's of (51) repeatedly for p times, we obtain

$$F_S^2(z)=(f(z))^{2p}\leq(1/3)^{4p}. \quad ...(53)$$

Using $b\leq\log_2(1/k)+1$, $k\geq\pi^2/4N^2$, $10^{-D}=\bar{k}(1/3)^{4p}$ and $S=pb$, we obtain

$$S\leq(2/((\log_{10}2)\cdot(\log_{10}81)))\cdot D(\log_{10}N-\log_{10}(\pi/2))$$

$$=3.48\cdot D(\log_{10}N-0.196), \quad ...(54)$$

which clearly indicates a logarithmic dependence of S on N.

The function $F_S(z)$ of the optimum convergence is similar to Tchebycheff's polynomials in many aspects. If the problem were to find a polynomial f(z) of degree S behaving like z^S for sufficiently large z and having the smallest absolute magnitude in the interval $-1\leq z\leq 1$, f(z) would be given, in terms of the well known Tchebycheff's polynomial

$T_S(x)=\cos(S\cdot\arccos(x))$, by

$$f(z)=2^{-S}T_S(z). \quad ...(55)$$

Using two parameters u and v, (55) can be rewritten in the following parametric form:

$z=\cos(u)$, ...(56)

$v=Su$, ...(57)

and $f(z)=2^{-S}\cos(v)$ ...(58)

In the present optimization problem, however, the class of functions to be considered is not the class of polynomials but is the class of rational functions having the form of (47). A problem, essentially the same in its nature as the present one, arose and has been solved in a closed form by electrical engineers in connection with the design of the best wave filters.[2)]

---

2) See for example, W. Cauer: "Theorie der Linearer Wechselstromschaltungen", Becker u. Erler, Leipzig (1941) and its English translation, "Synthesis of Linear Communication Networks", Mcgrow Hill, New York (1958).

In the design of filters, a certain class of rational functions of the frequency is required to be minimized and/or maximized in the given frequency interval or intervals and the optimum solution is given in terms of elliptic functions. The solution to the present problem can also be given in a closed form in terms of elliptic functions.

We shall tentatively rephrase the problem as the problem of the realization of the best rejection rate $\bar{k}$ for the given k and integer S and prove the following result:

The Main Result

The best S.I.D.R. factors for the rephrased problem are given by

$$A_s=B_0 sn(K+iK'(2s-1)/2S,k) \quad \text{for} \quad s=1,2,\ldots S, \qquad \ldots(59)$$

where i denotes $\sqrt{-1}$ (not the integer used in the previous section), sn is the Jacobian elliptic function of module k, and K, K' are the values of the following complete elliptic integrals of the first kind

$$K=K(k)=\int_0^{\pi/2}(1-k^2\sin^2\theta)^{-(1/2)}d\theta \qquad \ldots(60)$$

and $$K'=K(\sqrt{1-k^2}). \qquad \ldots(61)$$

$F_S(z)$, defined to be of the form (47) with A's of (59), behaves like Fig.1, which shows a special case of S=3. In the interval $B_0 \leq z \leq B_S$, $F_3(z)$ changes its s at the three zero points $z=A_1, A_2, A_3$ and the absolute magnitude

takes on a maximum value of $\sqrt{k}$ for four times at $z=B_0$, B_1, B_2 and B_3.

In the general case, $F_S(z)$ changes its sign at the S zero points $z=A_1, A_2 \ldots A_S$ and takes on a maximum value of $\sqrt{k}$ for S+1 times within the interval $B_0 \leq z \leq B_S$ at

$$z=B_s=B_0 sn(K+iK's/S,k) \quad \text{for} \quad s=0,1,\ldots S. \qquad \ldots(62)$$

The behaviour of $F_S(z)$ for positive z is similar to that of the Tchebycheff's polynomial (55). For negative values of z, on the other hand, $F_S(z)$ behaves quite differently from any polynomial. In the interval $-B_S \leq z \leq -B_0$, $F_S(z)$ has poles at $z=-A_s$, and its absolute magnitude takes on minimum values of $\sqrt{1/k}$ at $z=-B_s$. The properties for negative z are readily obtained from the following identity satisfied by any function of the form (47):

$$F_S(z)F_S(-z)=1. \qquad \ldots(63)$$

Proceeding to the proof of the main result and the "mini-max" property of $F_S(z)$ just stated, we introduce a parameter u, v

and a function $F^*_S(z)$ by

$$z=B_0 sn(u,k), \quad ...(64)$$

$$v=-\frac{i\bar{K}'}{2K}u+i\frac{\bar{K}'}{2}-(2S-1)\bar{K}, \quad ...(65)$$

$$F^*_S(z)=G(u)= \quad H(v)=\sqrt{\bar{k}}\, sn(v,\bar{k}), \quad ...(66)$$

where $\bar{K}$, $\bar{K}'$ are the values of the following complete elliptic integrals $\bar{K}=K(\bar{k})$, $\bar{K}'=K(\sqrt{1-\bar{k}^2})$ ...(67)

satisfying $$\frac{K'\bar{K}'}{K\bar{K}}=4S. \quad ...(68)$$

$\bar{k}$ is to be computed from (68) and the best computational procedure is to use the parameter $q=q(k)$ of the theta functions. Namely, using

$$\exp(-\pi K'/K)=q=q(k) \quad ...(69)$$

and $$\exp(-\pi\bar{K}'/\bar{K})=\bar{q}=q(\bar{k}), \quad ...(70)$$

(68) can be rewritten in the following form

$$\ln(q)\cdot\ln(\bar{q})=4\pi^2 S. \quad ...(71)$$

Formulas (64), (65) and (66) are similar in their forms to (56), (57) and (58). The main difference consists in the apearance of elliptic functions instead of circular

functions. (64) means that the function H(v) is to be regarded as a function G(u) of u by virtue of (65) and also as a function $F_S^*(z)$ of z by virtue of (64). In respect to (64), z is made to vary from $-\infty$ to $+\infty$ along the real axis by varying u along the edge of a retangle in the complex u plane of which the four vertices are at $-K+iK'$, $-K$, $+K$ and $+K+iK'$. At the same time v varies along the edge of another rectangle in the v plane of which the four vertices are at $\overline{K}+i\overline{K}'$, $-(2S-1)\overline{K}+i\overline{K}'$, $-(2S-1)\overline{K}$ and $\overline{K}$. Table 1 shows the correspondence of the values of z, u, v and H(v) for S=3. It will be immediately noticed that the values of $F_3^*(z)=H(v)$ at the special points $z=\infty$, 0, $\pm B_s$'s and $\pm A_s$'s agree with the values of $F_3(z)$ shown in Fig.1. Now, let $F_S(z)$ of (47) with (59) be regarded as a function f(u) of u. Namely,

$$f(u)=F_S(z(u)). \quad \ldots(72)$$

f(u) is obviously a doubly periodic function with 4K and

$2iK'$ being the two periods. Since $sn(u,k)$ is an elliptic function of order two, it has two zeros and two poles within a period-rectangle of which the four vertices are at

$u=2K+iK'$, $-2K+iK'$, $2K-iK'$ and $-2K+iK'$. ...(73)

In regard to the period rectangle (73), the order of $f(u)$ is at most $2S$ because both the numerator and the denominator of (47) are polynomials of degree S. Conversly, since $f(u)$ has poles at $u=-K-i(2s-1)K'/2S$ and zeros at $u=K+i(2s-1)K'/2S$ within the period-rectangle (73), the order of $f(u)$ is at least $2S$. Hence, $f(u)$ is necessarily of order $2S$ and there are no extra poles nor extra zeros besides those just mentioned. Next, consider $F_S^*(z)=G(u)$ of (66) as a function of u. $4K$ and $2iK'$ are also the two periods of $G(u)$ by virtue of (65) and (68). In regard to the period-rectangle of (73), $G(u)$ is an elliptic function of order $2S$ and all of the zeros and poles are positioned exactly in the same place as those of $f(u)$ by its construction. Hence, the ratio $f(u)/G(u)$ is an elliptic

function regular throughout its period-rectangle which implies that the ratio of the two is a constant. The value of the constant is readily seen to be unity by evaluating special values of the both functions, say for $u=iK'$ or $z=\infty$. Hence, the two functions $F_S(z)$ of (47) with constants of (59) and $F_S^*(z)$ of (66) are identical. Namely,

$$F_S^*(z)=F_S(z) \quad \text{for all } z. \qquad \ldots(74)$$

The "mini-max" property of $F_S(z)$ stated in connection with (62) is now an immediate consequence of the following well known property of the elliptic function used in (66). Namely, for real v $sn(v,\bar{k})$ takes on a maximum value of 1 at $v=(4n+1)\bar{K}$ and a minimum value of -1 at $v=(4n-1)\bar{K}$, where n denotes integers.

We now prove that the choice of S.I.D.R. factors given by (59) is the best. Suppose there exists another set of positive numbers $\bar{A}_1, \bar{A}_2, \ldots\bar{A}_S$ giving a function $\bar{F}_S(z)$ of the form (47) and resulting into a better suppression of the eigenvector expansion coefficients, which means

$$\bar{F}_S(z) < F_S(z) \quad \ldots(75)$$

for all z in the interval $B_0 \leq z \leq B_S$. $\quad \ldots(76)$

The difference between the two functions can be written in the following form:

$$F_S(z)-\bar{F}_S(z)=N(z)/D(z), \quad \ldots(77)$$

in which the denominator D(z) is a polynomial of degree 2S and the numerator N(z), a polynomial of degree 2S-1 or less since the difference (77) should tend to zero when z increases to infinity. Since the difference (77) is continuous in the interval (76) and changes its sign at least once in each of the intervals (B_s, B_{s+1}) because of (75) and (62), the numerator of (77) has at least S distinct zeros in the interval (76). For negative values of z we make use of the identity (63) and rewrite (77) as

$$F_S(-z)-\bar{F}_S(-z)=N(-z)/D(-z)=(\bar{F}_S(z)-F_S(z))/(F_S(z)\bar{F}_S(z)), \quad \ldots(78)$$

which indicates that N(z) has also S distinct zeros in the interval $-B_S \leq z \leq -B_0$. Having 2S or more distinct zeros and being a polynomial of degree 2S-1 or less, N(z) must be identically zero but this is obviously a contradiction. The existence of $\bar{F}_S(z)$ is thus denied and the main result is proved.

The exact relation among k, $\bar{k}$ and S is given by (71).

Actually, k and $\bar{k}$ will be very small numbers in most cases of practical interest. In such cases the following approximation to the function q(k) can be used:

$$q(k)=k^2/16. \quad ...(79)$$

Using (79), (71) can be rewritten as

$$\ln(k/4)\cdot\ln(\bar{k}/4)=\pi^2 S. \quad ...(80)$$

In most cases the values of k and $\bar{k}$ will be specified at the beginning as

$$k=\pi^2/4N^2 \quad ...(81)$$

and $$\bar{k}=10^{-D}. \quad ...(82)$$

In such cases, we first compute S by using (81) and (82) in (80) or in (71) if necessary. The S thus computed is generally not an integer. We, therefore, take the smallest integer not less than the S just computed and redefine it as the integer S. Using this integer S in (59), we compute the numbers A_s's to start the S.I.D.R. process. The convergence after S sweeps will be slightly better than the value specified at the beginning. Using the values of (81) and (82),

(80) can be rewritten in the following form

$$S=(2(\log_e 10)^2/\pi^2)(D+\log_{10}4)(\log_{10}N+\log_{10}(\pi/4))$$

$$=1.074(D+0.602)(\log_{10}N+0.105). \quad ...(83)$$

Comparing (83) with (54) of the octave method, the number of sweeps is observed to be reduced by a factor of 3 by the optimization of the convergence.

(83) indicates that the number of sweeps in the S.I.D.R. depends logarithmically on N, while the number of sweeps in the S.O.R., given by formula (5), depends linearly on N. Therefore, for sufficiently large N, the computation time needed in the S.I.D.R. will be much less than that needed in the S.O.R.. In making a comparison of the computation times, the ratio of the times per sweep R_M or R, treated at the end of section 2, must be taken into account. Using D=10 and R_M=4, we find from (83) and (5) that the overall computation time in the S.I.D.R. becomes less than that in the S.O.R. when N is greater than 17. If R=3.37 is used instead of R_M=4, the same holds when N is greater than 13. In case of D=10 and N=1000, the respective numbers of sweeps needed in the S.I.D.R. and S.O.R. are 36 and 3665. Thus, the time needed in the S.O.R. becomes 25.5 or 30.0 times more than that in the S.I.D.R., depending upon the assumptions R_M=4 or R=3.57.

We now consider the effects of round off errors.

Let $e^{(s}_{m,n}$ be the eigenvector expansion coefficients (c.f. (39)) of the round off errors which take place in the s'th iteration, e_M be the upper bound of the absolute magnitudes of $e^{(s}_{m,n}$'s and the function $F_{s,S}(z)$ be defined as

$$F_{s,S}(z)=\left(\frac{z-A_{s+1}}{z+A_{s+1}}\right)\cdots\left(\frac{z-A_S}{z+A_S}\right). \quad \cdots(84)$$

The eigenvector expansion coefficients $f_{m,n}$ of the effect of the round off errors in the final result are given by

$$f_{m,n}=\Sigma_s e^{(s}_{m,n} F_{s,S}(E^{*2}_m)F_{s,S}(E^{*2}_n). \quad \cdots(85)$$

A crude upper bound to $f_{m,n}$ can be obtained by making use of the fact that the absolute magnitude of $F_{s,S}(z)$ never exceeds unity for positive z. Namely,

$$f_{m,n} \leq \Sigma_s e^{(s}_{m,n} \leq Se_M. \quad \cdots(86)$$

While the convergence of the S.I.D.R. is independent of the sequential order of using the S.I.D.R. factors, the effect of round off errors does strongly depend upon it. For example, consider the case in which the S.I.D.R. factors are used in the ascending order and let the expansion coefficients of the round off errors for the lowest eigenvector

be given rather systematically by $e_{1,1}^{(s} = e_M$. In such a case (186) gives a reasonably good estimate of $f_{1,1}$, since the absolute magnitudes of most of the $F_{s,S}(E_1^{*2})$'s are actually very close to unity. Now consider the case of D=10 and N=1000 of which the number of sweeps has already been shown to be S=36. (86) indicates that the maximum worst case error of $36e_M$ for this case. The situation can be improved, however, by suitably reshuffling the sequential order of the S.I.D.R. factors. From the main result (59) giving the rules to compute the S.I.D.R. factors, it is readily seen that $F_{36}(z)$ includes $F_{12}(z)$ as a factor. Namely,

$$F_{36}(z)=F_{12}(z)\, Q_{24}(z), \quad \ldots(87)$$

where $Q_{24}(z)$ is the quotient including 24 of A_s's.

Similarly, $F_{12}(z)$ includes $F_4(z)$ as a factor. Namely,

$$F_{12}(z)=F_4(z)Q_8(z). \quad \ldots(88)$$

Using N=1000 and S=12 in (83) we obtain D=3.0 which implies

$$(F_{12}(z))^2 \leq \bar{k}=10^{-D}=10^{-3} \text{ for } z \text{ in the interval (76).} \quad \ldots(89)$$

Similarly, for $F_4(z)$ we obtain

$(F_4(z))^2 \leq 10^{-0.60} = 0.25.$ ...(90)

We now use the 24 A_s's in $Q_{24}(z)$ first. At the end of the 24th iteration, the effect of round off errors can accumulate up to $24e_M$ in terms of eigenvector expansion coefficients in the worst case. This effect, however, is suppressed by a factor of 10^{-3} because of (89) during the last 12 iterations. Hence, the effect of the round off errors in the first 24 iterations is negligibly small in the final result. Next, we use the 8 A_s's in $Q_8(z)$ of (88) from the 25th to the 32nd iterations. The maximum errors can accumulate up to $8e_M$ during these iterations but they are suppressed by a factor of 0.25 as indicated by (90) during the last 4 iterations. Hence, the effect of these 8 iterations in the final result is atmost $0.25 \times 8e_M = 2e_M$. The effect of round off errors in the last 4 iterations can accumulate up to $4e_M$ in the final result. The worst effect of round off errors to be expected in the final result, therefore, does not exceed $6e_M$ in its eigenvector expansion coefficients, which is only one sixth

of that to be expected in the cases of using the A_s's in a purely ascending or descending order. From the considerations made above, we can safely conclude that round off errors do not cause any serious troubles in the S.I.D.R..

§4 Concluding Remarks

1. The S.I.D.R. algorithm, with slight modifications of some constants, can be used with the 9 point formula instead of the 5 point formula used in (1) to approximate the Laplacian operator more closely. In case of the 9 point formula, an extra computation time of $N^2(4t_a+t_m+4t_1)$ per sweep will have to be added to both (43) and (44).
Thence, the time ratios, S.I.D.R. vs. S.O.R., will become $R_M=(8+1)/(2+1)=3$ and $R=(75+12)/(21+12)=2.64$.

2. The S.I.D.R. algorithm can be generalized in a straightforward way for the solution of Poisson's equations in rectangles subjected to inhomogenious boundary conditions.

3. An intuitive explanation to the apearance of the logarithmic term in the convergence rate formula (83) for the S.I.D.R. may be given considering the relaxation process as a diffusion process in the following way.

The homogenious boundary condition (i.e., the function should be zero)can be made to be satisfied by dividing thee entire x,y plane into unit squares and by placing negative

image processes in every neighbouring squares. Thence, the term with $f(a^*,1)$ in (29) and (33) can be regarded as a faithful representation of the influences of all of the images throughout the x, y plane. The treatment of the boundary condition in the S.I.D.R. being thus quite satisfactory, it can hardly be the cause of the apearance of the logarithmic term in (73).

Consider now an analytical analogue of S.I.D.R. (cf. (10)) with $\rho=\delta(x-x_0)\delta(y-y_0)$ and starting from an initial function of $\phi^{(0}=0$. The solution to this problem is the Green's function $G(x,x_0,y,y_0)$ satisfing $\Delta G=\delta(x-x_0)\delta(y-y_0)$ and the homogenious boundary condition. The result of the first iteration (cf. (10)) becomes

$$\phi^{(1}=-\frac{1}{2}\exp(-\sqrt{A_1}(x-x_0+y-y_0))+B^{(1}(x,y), \quad \ldots(91)$$

where $B^{(1}(x,y)$ is a function arising from the hypabolic sine terms in (14), (17), and is smooth in the unit square. The first term in (91) is continuous but not smooth at the locality of $(x,y)=(x_0,y_0)$. By tracing the behaviour of the succeeding

iterants $\phi^{(2}$, $\phi^{(3}$... it will be seen that they all consist of exponential terms similar to that in (91). Since the Green's function G has a logarithmic singularity at (x_0,y_0), an infinite number of continuous terms will be necessary to represent the discontinuous Green's function. Even when the logarithmic term is truncated at h=1/N in the finite difference case, an increasing number of terms will still be needed as N increases in order to make a reasonably good approximation. Formula (83) may thus be interpreted as that about $\log_{10}N$ iterations are needed to approximate the Green's function to a single decimal figure. In terms of the diffusion analogy as used in the introduction, about $\log_{10}N$ sweeps are needed to make a considerable portion (90%) of the errors to be diffused to and get eliminated at the boundary. The possibility of further speeding up the diffusion and elimination of the errors is an interesting open question.

4. While newly computed values are used whenever possible in the S.O.R. as the superscript (s,s-1 in (2) implies, older

values are used throughout the correction process (9) in the S.I.D.R.. Calling the former progressive and the latter conservative, a conservative S.I.D.R. scheme was developed in this paper in order to facilitate the development of the convergence theory given in section 3. There exist two motivations to develop progressive S.I.D.R. schemes, namely, saving in storage spaces and speeding up of computations.

Consider the following algorithn to be called quadrant S.I.D.R.. Let the unit square be divided into four equal subsquares I, II, III and IV. Let subsquare I be the one at $0\leq x,y\leq \frac{1}{2}$. Let a function $X^{(s}(x)$ be assigned to the boundary line $y=\frac{1}{2}$, $0\leq x\leq 1$ between the subsquares and $Y^{(s}(y)$, to the line $x=\frac{1}{2}$, $0\leq y\leq 1$ and let the initial values $X^{(0}(x)$ and $Y^{(0}(y)$ be zero. In respect to subsquare I, the finite difference analogues of the following computations are to be performed in the s'th iteration:

$W_1(x,y)=\Delta\phi^{(s}(x,y)-f(x,y);$

$a_s=\sqrt{A_s};$

Integrate $(-\frac{d}{dx}+a_s)W_2(x,y)=W_1(x,y)$ from $x=\frac{1}{2}$,

$W_2(\frac{1}{2},y)=Y^{(s-1}(y)$ until $x=0$;

Integrate $(\frac{d}{dx}+a_s)W_3(x,y)=W_1(x,y)$ from $x=0$,

$W_3(0,y)=-W_2(0,y)$ until $x=\frac{1}{2}$;

$W_4(x,y)=W_2(x,y)+W_3(x,y)$ and $Y^{(s}(y)=W_3(\frac{1}{2},y)$ for $0\leq y\leq\frac{1}{2}$;

Integrate $(-\frac{d}{dy}+a_s)W_5(x,y)=W_4(x,y)$ from $y=\frac{1}{2}$,

$W_5(x,\frac{1}{2})=X^{(s-1}(x)$ until $y=0$;

Integrate $(\frac{d}{dy}+a_s)W_6(x,y)=W_4(x,y)$ from $y=0$,

$W_6(x,0)=-W_5(x,0)$ until $y=\frac{1}{2}$;

$X^{(s}(x)=W_6(x,\frac{1}{2})$ for $0\leq x\leq\frac{1}{2}$;

$\phi^{(s}(x,y)=\phi^{(s-1}(x,y)+(W_5(x,y)+W_6(x,y))/2$ for $0\leq x,y\leq\frac{1}{2}$.

Similar operations are to be performed on the remaining subsquares. The functions $X^{(s}(x)$ and $Y^{(s}(y)$ serve to carry the influences of errors from one subsquare to the others. Comparing the quadrant S.I.D.R. with the conservative S.I.D.R. disclosed in section 2, the subtraction of exponential terms in (14), (17), (29) and (33) are eliminated by suitably choosing the initial value in each integration, which results into the

saving of $2N^2$ multiplications per sweep. The quadrant algorithm may be said to be conservative within each subsquare and progressive in respect to the subsquares. While N^2 extra storage locations are needed to store the values in the conservative S.I.D.R., $N^2/4$ locations will be sufficient in the quadrant algorithm.

There are many heuristic and intuitive reasons to believe that the quadrant S.I.D.R. would converge equally well as the conservative S.I.D.R.. Unfortunately, however, the clear-cut convergence theory of section 3 is not applicable to progressive algorithms like the quadrant S.I.D.R. because the eigenfunctions of the iterative operations are no longer independent of the values of the S.I.D.R. factors A_s's. The convergence property of the quadrant algorithm thus gives rise to another open question.

5. It will be the most interesting theme to apply the S.I.D.R. to problems other than Poisson's equations in squares and rectangles, for example, Poisson's equations in

arbitrarily shaped domains, axially symmetric and three dimentional Poissson's equations, Helmholtz's equations and eigenvalue problems. Because the speeding up of diffusion of errors is essential feature of the S.I.D.R., it should be applicable, intuitively, to problems mentioned above, The convergence theory developed in section 3, however, can not be applied for the same reason as in the case of remark 4.

6. Whenever there exists an exact and finite algorithm for the solution of a problem, iterative methods will generally yield to the exact algorithm in respect to the speed of computation when the required accuracy exceeds a certain yielding points. Since the finite difference solution of Poisson's equations is nothing but the solution of simultaneous linear equations, there exists exact algorithms. The fastest exact algorithm for the solution of Poisson's equations in squares and rectangles, within the scope of the author's knowledge, is the Fourier transform method as used by Hockney[3] combined with the FFT(Fast Fourier Transform)[4] developed by Cooley and Tucker.[5]

---

3) R.W. Hockney, "A Fast Direst Solution of Poisson's Equation Using Fourier Analysis", JACM, Vol. 12, pp.95-113 (1965).

4) The author is indebted to Professor H. Takahasi, Director of Computer Centre of the University of Tokyo, for drawing his attention to the FFT method.

5) J. W. Cooley and J. W. Tuckey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation, Vol. 19, pp. 297-301 (1965).

The Fourier transform method for the solution of Poisson's equation, $\Delta\phi=\rho$ in a unit square and $\phi=0$ at the boundary, consists of the finite difference analogues of the following three processes.

P1. Take the Fourier (sine) transform of $\rho(x,y)$ in either dimention x or y, say in y, and obtain the harmonic (i.e. $\sin(n\pi y)$) components $\bar{\rho}_n$ of ρ for n=1,2...N-1.

P2. Solve the ordinary differential equations

$$\frac{d^2\bar{\phi}_n}{dx^2} - n^2\pi^2\bar{\phi}_n=\bar{\rho}_n,$$

satisfied by the harmonic components $\bar{\phi}_n$ of ϕ. Using the "marching method"[3], which is equivalent to the mapping of $W_0(x,y)$ into $W_3(x,y)$ of (12) and (14) with $a=n\pi$, this process can be performed with about $4N^2$ multiplications.

P3. Perform an inverse Fourier transform to obtain ϕ from $\bar{\phi}_n$'s.

FFT is to be used in processes P1 and P3. Assuming that N is a power of two , $2N\log_2 N$ multiplications of complex numbers are needed to perform the FFT on a one dimentional

array of size N. A single multiplication of complex numbers would generally consist of four multiplications of real numbers. For the FFT of a real valued array, however, two real multiplications per complex multiplication has been shown to be sufficient.[6),7)]

---

6) H. Takahasi, "D6/TC/FFTR", A Library Program of the Computer Centre, University of Tokyo. (1966)

7) C. D. Bergland, "A Fast Fourier Transform Algorithm for Real Valued Series", CACM. Vol. 11, pp. 703-710 (1968)

Hence, $2\times N(2N\log_2 N)+4N^2=(26.4\log_{10}N+4)N^2$ multiplications are needed in the FFT solution of the Poisson's equation. Comparing this number with the $8N^2$ multiplications per sweep (cf. (44)) and the number of sweeps (83) of the S.I.D.R., we see that the S.I.D.R. yields to the FFT beyond the yielding point at about three decimal figures of accuracy. The FFT method, therefore, will be superior to the S.I.D.R. when more than three decimals of accuracy is needed. The S.I.D.R. will become superior when less than three decimals of accuracy is sufficient or when a series of similar problems is to be solved so that the result

of the preceeding problem can be used as a close approximation to the solution of the succeeding problem.

The FFT and the S.I.D.R. are thus complementary with each other in the solution of Poisson's equation in squares as well as in rectangles. The most interesting question consists in their applicability or adaptability to more general types of problems. While a great difficulty is anticipated in using the FFT method in other types of problems such as the solution of Laplacian or Poisson's equations in arbitrarily shaped domains, the S.I.D.R. seems to be applicable but the truth remains open as discussed in the previous remarks.

7. The present stage of the development of the S.I.D.R. is incomplete in many respects, especially in that the range and limitations of its applications are not well known. In order to clarify these points, more sophisticated convergence theories will have to be developed and empirical facts should be compiled from computer experimentations. Nevertheless, the present results, disclosed in section 2, 3 and on pure

mathematical reasoning, would be sufficient as an existence proof of a new kind of relaxation processes.

References and Footnotes

1) D.J.Evans, "Estimation of the line over-rel xation factor and convergence rates of an alternating direction line over-relaxation technique", The Computer Journal, Vol. 7, pp. 318-321 (1964)

2) See for Example, W. Cauer, " Theorie der Linearer Wechselstrom-schaltungen", Becker u. Erler, Leipzig (1941) and its English translation, " Synthesis of Linear Communication Networks", Mc Grow Hill, New york (1958).

3) R.W. Hockney, " A Fast Dirct Solution of Poisson's Equation Using Fourier Analysis", JACM, Vol. 12, pp. 95-113 (1965).

4) The author is indebted to Proffesor H. Takahasi, Director of the Computer Centre of the University of Tokyo, for drawing his attention to the FFT method.

5) J.W. Cooley and J.W. Tuckey, " An Algorithm for the machine calculation of complex Fourier series", Mathematics of Computation, Vol. 19, pp. 297-301 (1965)

6) H. Takahasi, "D6/TC/FFTR", a library program of the Computer Centre, University of Tokyo (1967).

7) G.D. Bergland, "A fast Fourier Transform Algorithm for Real Valued Series", CACM, Vol. 11, pp. 703-710 (1968).

Figure Caption

Fig. 1

Behaviour of the function $F_3(z)$

Table Caption

Table 1
Correspondence of the values of u, v, z and H(v) for S=3.
(ε means an infinitesimal positive number)

Table 1

Correspondence of the Values of z, u, v and H(v) for S=3
(ε means an infinitesimal positive number.)

z	u	v	$H(v)=F^*_3(z)$
$-\infty$	$-\varepsilon+iK'$	$\overline{K}+i\overline{K}/2+i\varepsilon$	1
$-B_3$	$-K+iK'$	$\overline{K}+i\overline{K}'$	$1/\sqrt{\overline{k}}$
$-A_3$	$-K+i5K'/6$	$0+i\overline{K}'$	∞
$-B_2$	$-K+i4K'/6$	$-\overline{K}+i\overline{K}'$	$-1/\sqrt{\overline{k}}$
$-A_2$	$-K+i3K'/6$	$-2\overline{K}+i\overline{K}'$	∞
$-B_1$	$-K+i2K'/6$	$-3\overline{K}+i\overline{K}'$	$1/\sqrt{\overline{k}}$
$-A_1$	$-K+iK'/6$	$-4\overline{K}+i\overline{K}'$	∞
$-B_0$	$-K$	$-5\overline{K}+i\overline{K}'$	$-1/\sqrt{\overline{k}}$
0	0	$-5\overline{K}+i\overline{K}'/2$	-1
B_0	K	$-5\overline{K}$	$-\sqrt{\overline{k}}$
A_1	$K+iK'/6$	$-4\overline{K}$	0
B_1	$K+i2K'/6$	$-3\overline{K}$	$\sqrt{\overline{k}}$
A_2	$K+i3K'/6$	$-2\overline{K}$	0
B_2	$K+i4K'/6$	$-\overline{K}$	$-\sqrt{\overline{k}}$
A_3	$K+i5K'/6$	0	0
B_3	$K+iK'$	$\overline{K}$	$\sqrt{\overline{k}}$
∞	$\varepsilon+iK'$	$\overline{K}+K'/2-i\varepsilon$	1

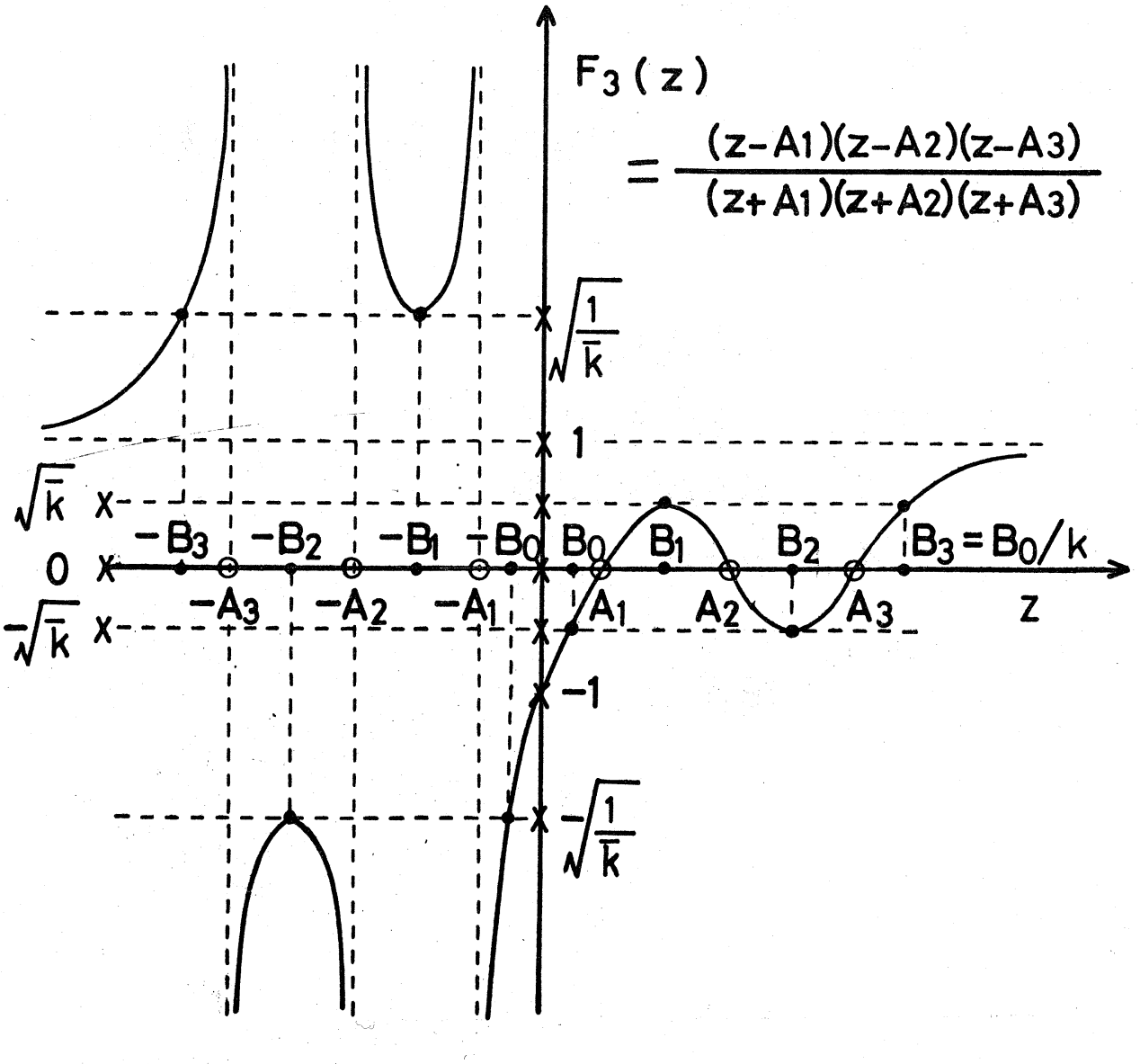
F3 (z)
= (z-A1)(z-A2)(z-A3) / (z+A1)(z+A2)(z+A3)
√1/k
1
√k
0
-√k
-B3
-B2
-B1
-B0
B0
B1
B2
B3 = B0/k
-A3
-A2
-A1
A1
A2
A3
z
-1
-√1/k

Fig. 1