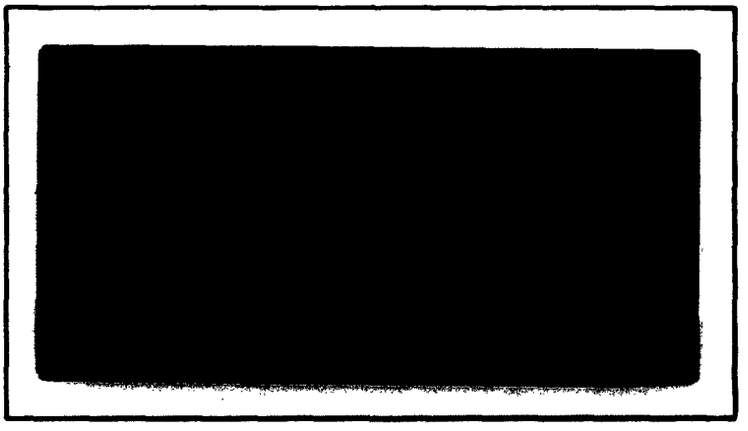
AIR FORCE INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY
UNITED STATES AIR FORCE



SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

AF-WP-O-OCT 63 3,500

An Investigation of Convergence Techniques for Implicit Numerical Solution of the Diffusion Equation for Transient Heat Transfer

THESIS

GA-63

Robert Theodore Poppe
Capt USAF

AF-WP-O-SEP 63 25

An Investigation of Convergence Techniques for Implicit Numerical Solution of the Diffusion Equation for Transient Heat Transfer

THESIS

Presented to the Faculty of the School of Engineering of
the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Robert Theodore Poppe, B.S.
Capt USAF

Graduate Astronautical Engineering

August 1963

Preface

The road to the end was a bit tortuous. I set out to compare the rate of convergence to solution of five or six iterative methods of solving the implicit finite difference representation of the equation of transient heat transfer, to do an analysis on the effect on the rate of convergence of a spiral grid scan, and a detailed error analysis. Toward the last third of the allotted time, the relationship between the adapted Wegstein technique and successive over-relaxation appeared and the decision was made to follow this line of investigation. To this end, this relationship between the two methods is now rather obvious, though, in my earlier work on this thesis it was not at all obvious to me. Time did not permit a thorough investigation designed to use this relationship for possible improvement of the adapted Wegstein technique.

I wish to express my appreciation to Dr. Bernard Kaplan of the Physics Department of the Air Force Institute of Technology for his assistance and guidance. Most of all, my gratitude (and more) goes to my wife who still insists that having me constantly off in my litter-strewn corner of the house is better than having me away on temporary duty somewhere.

Robert T. Poppe

Contents

Contents

Contents

List of Figures and Tables

Figure

List of Figures and Tables

Abstract

This study investigates the practical application of two convergence techniques designed to increase the rate of convergence of the method of successive displacements (Gauss-Siedel) for the implicit numerical solution of the diffusion equation of transient heat transfer. A sample problem of determining the temperature distribution in a cube with a constant internal heat source and fixed boundary temperatures is solved to provide the necessary data.

The results provide a theoretical basis for the adapted Wegstein technique that was not previously available. This theoretical basis brings to light the fact that successive overrelaxation and the adapted Wegstein technique are based on the same theoretical background.

A procedure based on estimating the maximum eigenvalue of the method of successive displacements is used to make an approximation of the relaxation factor for successive overrelaxation. This procedure is shown to be a practical method of finding the proper relaxation factor to estimate the difficult-to-determine optimum factor. The savings using this procedure was about 50% of the iterations required to obtain the same solution by successive displacements.

A comparison of the two accelerating techniques is made. Items of comparison are: the number of iterations

Abstract

required to obtain a solution of the sample problem, the progress of the solution with increasing iterations, the characteristics of the accelerating factor determined while computing in the normal successive displacements mode, and the error associated with the solution of the sample problem.

The principal results of this comparison are:

1. A 65% reduction in iterations over successive displacements when the optimum relaxation factor is used and about 50% and 30% reductions for successiver overrelaxation with an estimated relaxation factor and the adapted Wegstein technique, respectively.

2. Successive overrelaxation produces a smooth convergence to solution, whereas, the adapted Wegstein technique is ragged.

3. Both the single maximum eigenvalue used to compute the relaxation factor of the method of successive overrelaxation and the Wegstein slopes used to compute an accelerating factor for each individual node converge to the same value, but at different rates.

4. The error associated with the accelerated solutions is less than that encountered in the method of successive displacements.

An additional investigation is made of the effects on

Abstract

the rate of convergence of successive displacements of scanning the finite difference grid in a spiral mode from the fixed boundary conditions. The rate of convergence is compared to that obtained using a conventional serial scanning procedure. The results obtained show a small decrease in the number of iterations required for the spiral mode over the serial scan, but the savings are not significant. A tentative conclusion is reached that no change in scanning procedure from serial scan will produce a significant decrease in the iterations required to solve the transient heat transfer problem by successive displacements.

I. Introduction

Background

The analytical solution to partial differential equations of engineering and applied physics can not in general be obtained except in very special cases. Thus, approximate methods of solution have been developed, and the most popular of these procedures is the method of finite differences. Prior to the development of large scale digital computers, practical numerical solution of the finite difference equations was almost entirely performed by pencil and paper relaxation methods. When computers became available, it was found that the original relaxation methods were not as successful as some systematic computation of points in a convienent cyclic order (Ref 4:242). As a result, the development of new and the adaptation of old solution procedures for use with digital computers has been the subject of intensive study in the past ten tc fifteen years.

The state-of-the-art at the present time provides a number of digital computer routines for solution of the finite difference representation of the common types of partial differential equations encountered in engineering. For instance, the finite difference representation of Poisson's equation, $\nabla^2 u = f(u)$ for $u = \delta(x,y)$ in the rectangular region R with U described on the boundary S-- the equation encountered in such diverse engineering routines

as elasticity, heat transfer, electromagnetism and fluid dynamics--has at least ten possible representations for use in the solution. (Ref 4:282). The theoretical basis for these diverse methods are well developed for the simple, two dimensional, rectangular geometry involved. The theoretical basis for extending these routines into cases of irregular geometry and three dimensions is still rather fragmentary. The state of development does, however, permit an experimental approach to the investigation of possible extensions to the existing theory.

The specific case of the solution of the equation of non-steady heat transfer represents an important computer routine in a given engineering facility. In three dimensions, this equation is of the form

$$\frac{1}{K} \nabla \cdot K \nabla u(x,y,z,t) = \frac{1}{\alpha} \frac{\partial}{\partial t} u(x,y,z,t) + f(x,y,z,t)$$

The presence of the time dependence permits formation of two popular forms of the finite difference representation, depending on whether the approximations of the space derivatives are set to provide the forward difference (or explicit) equation or the backward difference (or implicit) equation. The implicit form leads to the requirement to solve a large system of simultaneous linear equations (realistically, 500 or more). The explicit form, lacking the mathematical complexities of the implicit form, can be solved by a step-by-step solution of

explicit relationships, starting with given initial values of u in the region R for a given time t.

The explicit form has one basic disadvantage, however. In moving forward in time from the time of given initial conditions, stability considerations require that a relationship between grid size and time increments be maintained such that decreasing the size of the grid requires a corresponding decrease in time step. In computations over long periods of time where data requirements for specific times indicate fairly large increments could be used, the size of the time step is restricted by the useable grid size. This restriction becomes a frustration that creates a vast and often prohibitive amount of computation to obtain a solution. The speed and capacity of modern computers has permitted wider application of the explicit method to problems involving a relatively large number of computational steps, but there is still a practical limit to the number of time steps to be computed. Even so, the solution of the explicit equation remains a prime computer method of obtaining approximate solutions to parabolic partial differential equations of interest.

The solution of systems of linear equations has been a subject of intensive study by mathematicians for many years. As a result, certain systematic iteration procedures suitable for hand calculation have been available to solve the linear systems of the implicit method. Yet, the additional complexities of these procedures and the need for iteration to

solution made the trade-off point between implicit and explicit methods heavily in favor of the explicit form for the class of problems that could be attacked by hand numerical calculation. After large scale digital computers offered potential for rapidly solving large numbers of simultaneous equations, interest in the implicit form was renewed. Crank and Nicholson seem to have been the first to use implicit methods. In 1947, these men demonstrated that the stability restrictions of the explicit method do not apply to the implicit form (Ref 4:102). This promise of freedom of choice of space and time increments, subject only to truncation error and convergence considerations, offers attractive advantages that offset the disadvantage of complexity of the implicit form.

As previously mentioned, a number of routines for solution of implicit equations by digital computers are in current use. Since the system of equations is generally large, the more direct method of matrix inversion is not practical; thus most routines use an iterative procedure. Of these useful routines, the ones that are in common use are based on the method of successive displacements--often referred to as the Gauss-Siedel method. Successive displacements is probably the simpliest iterative procedure, is easily adapted to computer programs, and is therefore, widely used (Ref 10:374). Routines have been developed to speed the rate of convergence of the method of successive displacements.

In his doctoral dissertation in 1950, D. Young, working in two dimensions, extended earlier work by Frankel and provided a theoretical basis for a process that increases the rate of convergence by systematically overrelaxing the Gauss-Siedel solution of boundary value problems where u is specified on the boundary. Though the theoretical basis is somewhat restrictive, the practical success of overrelaxation has led to the application of the method to other problems with a significant increase in the rate of convergence being realized (Ref 4:242). Specifically, overrelaxation has been applied to the problem of transient heat transfer. In 1958, J. Wegstein suggested a means of accelerating the convergence of iterative solutions of problems of the form (Ref 9). This procedure was empirically adapted to the solution of the implicit representation of the transient heat transfer equation by B. Kaplan and N. Clark in 1958 (Ref 5).

In summary, analytical solutions to partial differential equations are generally not available, so finite difference methods are used to obtain approximate solutions. Two popular forms of the finite difference representation are available--the explicit and the implicit form. The explicit form may be solved by a simple step-by-step procedure, but this form suffers from restrictive stability considerations. In fact, stability restrictions make the explicit form unsuitable for the class of problems requiring solution over large time increments (e.g. the determination of temperatures in a

nuclear reactor after shut down). Thus, one must resort to the implicit form for a solution. Despite relative freedom of choice of grid size and time steps, if the number of iterations required to obtain a solution for each time step is too large, the implicit form may be equally useless. Then the only hope of obtaining a solution on a given type of computer is to speed the rate of convergence of the iterative solution, thereby reducing the number of iterations required. Aside from making some solutions possible, a successful method of speeding convergence of a slower method of solving the system of simultaneous equations also results in attractive economy of computer operation. With these potential gains as impetus, a major effort within the general field of research in numerical analysis is being devoted to finding rapidly converging solution procedures.

Purpose and Scope

This study is an attempt to add to the practical knowledge of the use of techniques for accelerating the iterative solution of the implicit form of the finite difference representation of the transient heat transfer problem. Specifically, the study provides a comparison of two techniques for accelerating the method of successive displacements; successive overrelaxation and an adapted Wegstein's accelerating technique. The primary points of comparison are rate of convergence to solution, associated error, and simpli-

city and applicability of the method.

The comparison of the two methods is provided by the solution of a sample heat transfer problem of conduction in a cubic space which has a constant, distributed source of heat energy in the interior region and boundaries that are maintained at a constant temperature. The exact problem chosen has an analytical solution which provides the standard for measure of error magnitudes.

Two additional points in the general area of accelerating the rate of convergence of the method of successive displacements are investigated.

1. Using an experimental approach, an attempt is made to determine the effect on the rate of convergence of scanning the finite difference grid in a spiral mode from the fixed boundary conditions. The rate of convergence is then compared to the rate obtained using a conventional serial scanning sequence.

2. The theoretical basis for the accelerating effect of successive overrelaxation and the adapted Wegstein technique is reviewed for similiarities and differences of the two methods. This information is then used as a basis for experimental attempts to alter the basic adapted Wegstein technique to improve the speed of convergence of this method.

II. Theory

The Partial Differential Equation of Transient Heat Transfer

From the theory of heat transfer, the partial differential equation of heat conduction with an internal heat source or sink may be expressed as the parabolic partial differential equation (Ref 1:45)

$$\frac{1}{a}\frac{\partial u}{\partial t} = \frac{1}{K}\nabla \cdot K \nabla u + \frac{1}{K}\dot{q}(x,y,z,t) \qquad (1)$$

where $u(x,y,z,t)$ is the temperature (°C)

K is the conductivity of the medium (cal/cm-sec-°C)

$\dot{q}$ is the strength of the source or sink (positive for a source--negative for a sink) in units of energy generated or absorbed per unit volume time (cal/cm^3-sec)

a is the diffusivity (cm^2/sec) which may be determined from $a = K/\rho C_p$ with ρ as density (gr/cm^3) and C_p as the heat capacity (cal/gr-°C)

∇ is the del operator of vector analysis which is expressed as

$$\nabla = \left(\vec{i}\frac{\partial}{\partial x} + \vec{j}\frac{\partial}{\partial y} + \vec{k}\frac{\partial}{\partial z}\right) \qquad (2)$$

with $\vec{i}$, $\vec{j}$, and $\vec{k}$ the usual cartesian unit vectors

With the simplifying assumption that the conductivity is a constant, equation 1 reduces to

$$\frac{1}{a}\frac{\partial u}{\partial t} = \nabla^2 u + \frac{1}{K}\dot{q}(x,y,z,t) \qquad (3)$$

The Finite Difference Representation

The finite difference representation of equation 3 that is used in this study is obtained by replacing the first partial derivative with respect to time by the approximation

$$\frac{\partial u}{\partial t} = \frac{U(\iota, \jmath, k, t+\Delta t) - U(\iota, \jmath, k, t)}{\Delta t} \qquad (4)$$

and the second partial with respect to a given space variable by an approximation of the form

$$\frac{\partial^2 u}{\partial x^2} = \frac{U(\iota+1, \jmath, k, t+\Delta t) - 2U(\iota, \jmath, k, t+\Delta t) + U(\iota-1, \jmath, k, t+\Delta t)}{h_x^2} \qquad (5)$$

where $$h_x = \Delta x \qquad (6)$$

to obtain the implicit form, or by

$$\frac{\partial^2 u}{\partial x^2} = \frac{U(\iota+1, \jmath, k, t) - 2U(\iota, \jmath, k, t) + U(\iota-1, \jmath, k, t)}{h_x^2} \qquad (7)$$

to obtain the explicit form. How these approximations are formed using truncated Taylor series expansions for a sample elliptic and parabolic partial differential equation and the resulting matrix forms for a sample serial scan are shown in detail in Appendix A.

Substituting the approximations given in equations 4 and 5 and similiar expressions for the partial derivatives with respect to the y and z space variables into equation 3, one obtains the implicit form of the finite difference equation

$$\frac{U(i,j,k,t+\Delta t)-U(i,j,k,t)}{a\Delta t} = \frac{U(i+1,j,k,t+\Delta t)+U(i-1,j,k,t+\Delta t)}{h_x^2} + \frac{U(i,j+1,k,t+\Delta t)+U(i,j-1,k,t+\Delta t)}{h_y^2} + \frac{U(i,j,k+1,t+\Delta t)+U(i,j,k-1,t+\Delta t)}{h_z^2} - \left[\frac{2}{h_x^2}+\frac{2}{h_y^2}+\frac{2}{h_z^2}\right]U(i,j,k,t+\Delta t)+\frac{\dot{q}}{K} \quad (8)$$

From this equation, one can see that if the temperatures over the region are known at time t and the boundary values and source strength $\dot{q}$ are fixed for any time, then the temperatures at the interior nodes at time $t+\Delta t$ form a system of N simultaneous equations in N unknowns; one unknown for each interior node point. In matrix notation, this system may be expressed as

$$\bar{\bar{A}}\bar{U} = \bar{C} \quad (9)$$

Alternately, one may use equation 7 to approximate the second partial to obtain the explicit form

$$\frac{U(i,j,k,t+\Delta t)-U(i,j,k,t)}{a\Delta t} = \frac{U(i+1,j,k,t)+U(i-1,j,k,t)}{h_x^2} + \frac{U(i,j+1,k,t)+U(i,j-1,k,t)}{h_y^2} + \frac{U(i,j,k+1,t)+U(i,j,k-1,t)}{h_z^2} - \left[\frac{2}{h_x^2}+\frac{2}{h_y^2}+\frac{2}{h_z^2}\right]U(i,j,k,t)+\frac{1}{K}\dot{q} \quad (10)$$

Here, each interior node temperature at $t+\Delta t$ is expressed explicitly in terms of known temperatures at time t so that

there is no need to solve a system of simultaneous equations.

Solution Procedures for Systems of Linear Equations

It has been shown that the implicit form of the parabolic equation * leads to a system of simultaneous finite difference equations. The following forms of the solution of the system will be considered:

1. The method of successive displacements (often referred to as the Gauss-Siedel method)
2. Successive overrelaxation
3. Adapted Wegstein convergence technique

Successive Displacements

Solution Procedure

For simplicity in the expression of relationships, the following notation is adopted

$$U^{(n)}(i,j,k,t) \tag{11}$$

where n = the number of the iteration

Consider the two dimensional parabolic equation

$$\nabla^2 u = \frac{1}{a}\frac{\partial u}{\partial t} \tag{12}$$

in the implicit difference form with the gird size equal in the x and y direction (i.e. $h_x = h_y = h$) or

---

* A system of simultaneous equations also results from the finite equations of elliptic boundary value problems -- See Appendix A.

$$U(x,y,t+\Delta t) = \frac{r}{1+4r}\left[U(x+1,y,t+\Delta t)+U(x-1,y,t+\Delta t)\right.$$
$$\left.+U(x,y+1,t+\Delta t)+U(x,y-1,t+\Delta t)\right] + \frac{1}{1+4r}\left[U(x,y,t)\right] \quad (13)$$

where
$$r = \frac{a\Delta t}{h^2} \quad (14)$$

Using the method of successive displacements to solve the system of equations resulting from the above sample equation involves the following steps*

1. Estimating a starting value of U for the interior grid points (i.e. those points in the region considered, excluding the known boundary points)

2. Improving these initial estimates according to an arbitrary but fixed ordering of points.

3. Using improved values as soon as available

---

* There is a similiar procedure called simultaneous displacements which is generally not as rapidily converging as successive displacements, though it may converge in cases where successive displacements diverges (Ref 3:133). The essential difference is that in simultaneous displacements no improved value of U is used until all values are improved. Since convergence is generally slower, simultaneous displacements is not a popular method (Ref 4:226).

4. Continuing the iterative cycle until the absolute value of U of the present iteration minus the value of U of the previous iteration is equal to or less than an established criteria put into the computer as input data.

Thus, if the iteration notation is added to equation 13 the result is

$$U^{(n+1)}(x,y,t+\Delta t) = \frac{r}{1+4r}\left[U^{(n)}_{(x+1,y,t+\Delta t)} + U^{(n+1)}_{(x-1,y,t+\Delta t)} + U^{(n)}_{(x,y+1,t+\Delta t)} + U^{(n+1)}_{(x,y-1,t+\Delta t)}\right] + \frac{1}{1+4r}\left[U(x,y,t)\right] \qquad (15)$$

Convergence

Probably the simpliest statement of the necessary conditions for convergence of the method of successive displacements is given by Forsythe and Wasow. They state that, "If (the matrix) A (of equation 9) has diagonal dominance * and is not reducible, then the method of successive displacements converges" (Ref 4:236). In practice, decreasing the grid size has the effect of decreasing the diagonal dominance by increasing the sum of the off-diagonal terms. Because the diagonal is weakened, the convergence is slower.

This simple statement of convergence is, however, more restrictive than necessary. A less restrictive but more

---

* See "Definitions", Appendix C

complex criteria may be developed as follows (Ref 4:209). Partition the matrix A into three matrices such that

$$\bar{\bar{A}} = (\bar{\bar{E}} + \bar{\bar{D}} + \bar{\bar{F}}) \tag{16}$$

where E is the lower triangular matrix of a_{ij} where $i > j$ and there are zeros elsewhere

F is the upper triangular matrix of a_{ij} where $i < j$ and there are zeros elsewhere

D has the diagonal elements a_{ij} and zeros elsewhere

It is then possible to write the method of successive displacements in the following matrix form (Ref 4:236)

$$\bar{U}^{(m+1)} = -(\bar{\bar{D}}+\bar{\bar{E}})^{-1}\bar{\bar{F}}\,\bar{U}^{(m)} + (\bar{\bar{D}}+\bar{\bar{E}})^{-1}\bar{C} \tag{17}$$

Then the necessary and sufficient condition for convergence is that all the eigenvalues of the matrix $-(\bar{\bar{D}}+\bar{\bar{E}})^{-1}\bar{\bar{F}}$ are less than one in modulus. Further, these eigenvalues (η_i) are the zeros of the determinantal equation

$$\det(\eta\bar{\bar{E}} + \eta\bar{\bar{D}} + \bar{\bar{F}}) = 0 \tag{18}$$

Examples of the forms of these matrices are given in Appendix B.

Since the eigenvalues of the large matrices encountered in practical problems are not easily determined, this criterion is not very useful for estimating whether a given system will converge for the method of successive displacements. The size of the maximum eigenvalue does, however,

play an important role in accelerating the convergence of the method of successive displacements. This role is discussed later.

One more important point must be made. The method of successive displacements depends critically on the order in which the various unknowns are computed, since the size of the eigenvalues depend on this order and the smaller the maximum eigenvalue, the faster the process converges (Ref 4:218,257).

Successive Overrelaxation

Solution Procedure

Successive overrelaxation is the first method considered of the two methods of accelerating the convergence of successive displacements that are compared in this study. In this method acceleration is achieved by a simple modification of the equation for solution by successive displacements (Ref 9:388). The modified form for the transient heat equation in three dimensions is

$$\begin{aligned} U^{(m+1)}_{(i,j,k,t+\Delta t)} = \omega \Big\{ \frac{r}{1+4r} \Big[& U^{(m+1)}_{(i-1,j,k,t+\Delta t)} + U^{(m)}_{(i+1,j,k,t+\Delta t)} \\ & + U^{(m+1)}_{(i,j-1,k,t+\Delta t)} + U^{(m)}_{(i,j+1,k,t+\Delta t)} \\ & + U^{(m+1)}_{(i,j,k-1,t+\Delta t)} + U^{(m)}_{(i,j,k+1,t+\Delta t)} \\ & + \frac{1}{1+4r} \Big[U(i,j,k,t) + \frac{q \Delta t}{\rho} \dot{q} \Big] \Big\} \\ & + [1-\omega] U^{(m)}_{(i,j,k,t+\Delta t)} \end{aligned} \tag{19}$$

where ω = relaxation factor

The procedure for solution is the same as for the method of

successive displacements. Note that when $\omega = 1$, the form reduces to the method of successive displacements.

Applicability of the Method

Young in his doctoral dissertation showed that successive overrelaxation improves the convergence rate of the method of successive displacements for a class of matrices with a property he calls Property (A) * (Ref 4:243). No attempt will be made here to describe in detail Young's proof of the theoretical basis for the method of successive overrelaxation. Rather, interest in the theory behind the method will be centered on the determination of the optimum relaxation factor. A detailed description of the theoretical background may be found in Forsythe and Wasow beginning on page 242 (Ref 4). Specifically, the method is applicable to the approximations of the parabolic partial differential equation of heat transfer used in this study (Ref 4:105).

It is of interest to note that overrelaxation has been so successful that the method has been profitably applied to problems with matrices that do not have Property (A). This success has led to work toward the extension of Young's proof to a more general class of problems and has provided a basis for the speculation that the method may be useful for any problem (Ref 4:260,261).

---

* See "Definitions" Appendix C

Determining the Optimum Relaxation Factor

When the optimum relaxation factor is known, significant savings in iterations can be accrued by using overrelaxation in the solution of partial differential equations. There is unfortunately, no simple direct way of accurately determining the optimum factor prior to the start of computations except for simple problems involving simple geometry. Compounding the problem is the rate of degradation of savings as the chosen relaxation factor moves away from the optimum value. A characteristic plot of relaxation factor versus number of iterations to solution is shown in figure 1. The result is that determining the optimum factor is the most important problem in using the method of successive overrelaxation (Ref 4:257). A plot such as shown in figure 1 is a way of determining the optimum factor, though, of course, a very impractical way unless a large number of similiar problems are to be considered.

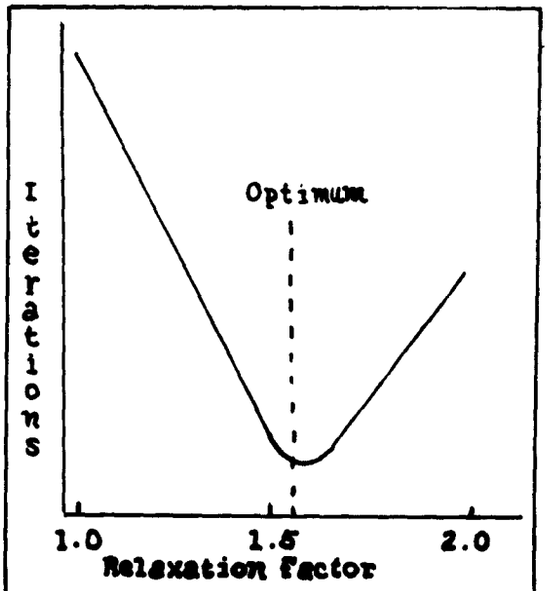


Fig. 1
Sample Plot of Relaxation Factor Versus Iterations to Solution

Figure 1 demonstrates additional important properties of the relaxation factor. First, the limits on the optimum factor are $1<\omega<2$ where $\omega=1$ is equivalent to the method of successive displacements. Factors less than one are termed underrelaxation and are known to be less profitable than $\omega>1$ for acceleration (Ref 4:308). The upper limit is based on the maximum eigenvalue of the method of successive displacements. This relationship will be described in more detail later. The second property is related to the shape of the curve in the vicinity of the optimum factor. The smaller rate of change of slope on the high side of the optimum demonstrates that it is better to overestimate the relaxation factor than to underestimate it. A theoretical basis for this property has been developed (Ref 4:257).

Various means of estimating the optimum factor have been proposed. As a starting point in reviewing the more prominent method's, consider the means for determining the factor for Laplace's equation over a regular space. For the case of equal grid size (h) and the finite difference representation given by equation 74, the optimum relaxation factor may be found from (Ref 9:365)

$$\omega_b = 1 + \frac{\lambda}{(1+\sqrt{1-\lambda})^2} \qquad (19a)$$

where

$$\lambda = \cos^2 \frac{\pi}{M} \approx 1 - \left(\frac{\pi}{M}\right)^2 \tag{19b}$$

$$M = \frac{1}{h} \tag{19c}$$

The variable λ is referred to as "the spectral radius * of the linear transformation defined by the Gauss Siedel (i.e. successive displacements) method" (Ref 9:387). The relationship of λ to the progress of the solution by successive displacements may be demonstrated as follows. Define

$$d^{(m+1)} = U^{(m+1)} - U^{(m)} \tag{19d}$$

Then λ may be described as the limiting ratio of $d^{(m+1)}$ and $d^{(m)}$ or (Ref 9:387)

$$\lambda = \lim_{m \to \infty} \frac{d^{(m+1)}}{d^{(m)}} \tag{19e}$$

Young goes on to give the following formula for computing λ for Laplace's equation for a rectangle with sides $a = Rh$ and $b = Sh$ where R and S are integers (Ref 9:389)

$$\lambda = \left[\frac{1}{2}\left(\cos \frac{\pi}{R} + \cos \frac{\pi}{S}\right)\right]^2 \tag{19f}$$

He then advises that for regions other than a rectangle, may be estimated for a rectangle containing the region under consideration.

For the parabolic equation of the form

---

* See "Definitions", Appendix

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \tag{19g}$$

and the finite difference approximation of

$$U(i,t+\Delta t) = \frac{r}{2}\Big[U(i+1,t) + U(i-1,t) + U(i+1,t+\Delta t) + U(i-1,t+\Delta t) - 2U(i,t+\Delta t)\Big] + \Big[1-r\Big]U(i,t) \tag{19h}$$

Young provides the following upper bound for λ (Ref 9:403)

$$\lambda \leq \left(\frac{r}{1+r}\right)^2 \tag{19i}$$

where

$$r = \frac{\Delta t}{h^2} \tag{19j}$$

Then for the two-dimensional parabolic equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{19k}$$

the upper bound for λ becomes (Ref 9:405)

$$\lambda \leq \left(\frac{2r}{1+2r}\right)^2 \tag{19l}$$

Of course, none of the above formulae for λ provide a precise expression for determining an optimum relaxation factor for the parabolic equation of transient heat transfer. They do, however, provide an insight into the nature of the dependence of ω on the parameters of the problem.

Since there is no direct method of computing the optimum relaxation factor prior to the start of the iterative process, a number of possibilities for estimating the factor have been suggested. Principal interest in this study is in

the process of estimating the factor by running the computer program with $\omega=1$ (i.e. by successive displacements) (Ref 4: 368,369). How this leads to the optimum factor is described as follows. One may define the optimum factor by

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\eta_1^{(1)}}} \tag{20}$$

where $\eta_1^{(1)}$ is the dominant eigenvalue of the matrix $-(\bar{\bar{D}}+\bar{\bar{E}})^{-1}\bar{\bar{F}}$ of the method of successive displacements

Under the assumption that the dominant eigenvalue $\eta_1^{(1)}$ is real, a condition that will exist for example when the A matrix is symmetric with all $a_{ii}>0$ -- a common occurrence with finite difference equations (Ref 4:252)--the value of $\eta_1^{(1)}$ may be determined from the limiting ratio of the norms of the residuals* of two successive iterations, or (Ref 4:369)

$$\eta_1^{(1)} = \lim_{m\to\infty} \frac{\|Y_{m+1}\|}{\|Y_m\|} \tag{21}$$

where

$$\|Y_m\| = \left[\sum_{i=1}^{N} \left(U_i^{(m+1)} - U_i^{(m)}\right)^2\right]^{1/2} \tag{22}$$

N = the number of unknowns in the system

This process is subsequently referred to as the norm of the residuals. Alternately, one may use the first power norm

$$Q_m = \frac{\|Y_{m+1}\|_{(1)}}{\|Y_m\|_{(1)}} \tag{23}$$

where

*See "Definitions", Appendix C

$$\|Y_m\|_{(1)} = \sum_{i=1}^{N} \left| U_i^{(m+1)} - U_i^{(m)} \right| \tag{24}$$

$$\lim_{m \to \infty} Q_m \approx \eta_1^{(1)} \tag{25}$$

The apparent disadvantage of running for a number of iterations with $\omega=1$ is that one loses the effect of acceleration while ω is one. Unless there is a rapid convergence to the vicinity of $\eta_1^{(1)}$, this loss of acceleration may degrade the method to a point where very little acceleration is achieved.

Is it possible then to determine the optimum factor while running with an estimate? To begin this discussion, the effect of the successive overrelaxation modification of the successive displacements equation on the pertinent eigenvalues is of interest. The eigenvalues of the overrelaxation equation are given by the expression (Ref 4:247)

$$\det \left[(\omega^{-1}\bar{\bar{D}} + \bar{\bar{E}})\eta + (1-\omega^{-1})\bar{\bar{D}} + \bar{\bar{F}} \right] = 0 \tag{26}$$

where D, E, and F are as given on page 14
This compares to equation 18, the equation for finding the eigenvalues of the method of successive displacements. If one defines

$$\bar{\bar{H}}(\omega) = -(\omega^{-1}\bar{\bar{D}} + \bar{\bar{E}})^{-1} \left[\bar{\bar{F}} + (1-\omega^{-1})\bar{\bar{D}} \right] \tag{27}$$

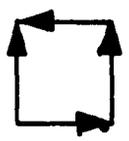
then two important properties of $\bar{\bar{H}}(\omega)$ can be cited. "The eigenvalues $\eta_i^{(\omega)}$ of $\bar{\bar{H}}(\omega)$ remain the same for any consistent order σ" (Ref 4:251). Order here implies the order in which the points are taken. A consistent order is described later.

Thus, "when the matrix A has Property (A), the value of is independent of the order σ for a considerable class of orders (called consistent)" (Ref 4:261).

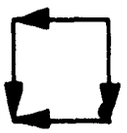
The problem encountered in running with a trial relaxation factor to determine the optimum is the "sequence of values of $\|Y_m\|$ in any norm is a rather irregular sequence for ω near ω_{opt}" (Ref 4:370), and there is difficulty in determining which trial value of ω is the best. Experiments by Ortega have indicated that the sequence of values of the ratios of the first power norm Q_m demonstrate differences between values in the sequence that are low in magnitude for $\omega < \omega_{opt}$ and relatively high in magnitude for trial near or slightly larger than ω_{opt} (Ref 4:371). The results thus far have been inconclusive, however.

The remaining point to be covered in this subparagraph is the definition of consistent order. According to Young, an order is consistent "if and only if each elementary square mesh of the net is bounded by four arrows with zero circulation" (Ref 4:245,246). Figure 2 demonstrates consistent and nonconsistent orders.

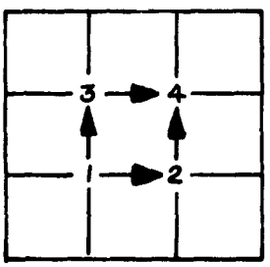
A pertinent question is what happens when one uses an nonconsistent order for successive overrelaxation? Varga has proven that for LaPlace's equation, no nonconsistent order has a rate of convergence as great as that common to all consistent orders (Ref 4:259). Powers investigated this point



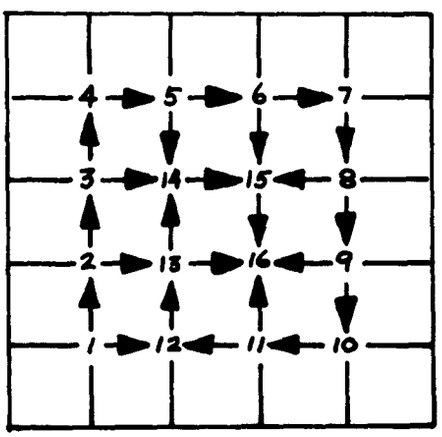
1. Nonconsistent



2. Consistent



3. Consistent Serial Scan



4. Nonconsistent Spiral Scan

Fig. 2

Examples of Consistent and Nonconsistent Ordering

for the Dirichlet difference equation * for a rectangle and some of her results have been displayed in Table 1. (Ref 7) These results are generally considered inconclusive since she chose to use a measure of convergence criteria of

$$\sum_{i,j} \left[U_{(i,j)}^{(m+1)} - U_{(i,j)}^{(m)} \right] \leq 2^{-33}$$

which proved to be relatively insensitive (Ref 4:259).

Convergence

The question of convergence is simply covered by the following quotation. "The power of Young's method lies in its acceleration of the rate of convergence of an already convergent process, not in any ability to create a convergent algorithm...." (Ref 4:254).

The Adapted Wegstein Convergence Technique

The Wegstein Technique

The true Wegstein technique, which is the basis for this second method of accelerating the method of successive displacements, was originally developed for accelerating the iterative solution of a single equation-not a system of linear equations. If the problem is to determine the solution of

$$F(x) = 0 \tag{28}$$

one may often express the equation in the form (Ref 8:9)

$$x = S(x) \tag{29}$$

*See "Definitions," Appendix C

Table 1

Sample Results of Solution of the Dirichlet Difference Equation by Consistent and Nonconsistent Orders

Order[a]	Number of Iterations	
	Using Optimum ω	With $\omega = 1$
Consistent		
Serial[b]	34	146
Even-Odd[c]	30	146
Nonconsistent		
Ordered by rows, direction reversed for each row	43	148
Spiral[d]	41	147
Random selection	33	147
Spiral border, random interior	36	147

This data is extracted from reference 6

[a] The problem was 14 nodes along the y axis and 6 nodes along x

[b] This scan sequence was: U(1,1) to U(1,14), then U(2,1) to U(2,14), and so on to U(6,1) to U(6,14)

[c] Starting with U(1,1), the points for which i+j is even were computed first in sequence. Then returning to U(2,1) the remaining points were computed in sequence.

[d] U(1,j) was taken for j = 1,...,14; then U(i,14) i = 2,3,4,5 then U(6,j) for j = 14, 13,..., 2, 1; then U(i,1) for i = 5,4,3,2. This pattern was then repeated starting with U(2,2) to U(2,13) etc.

Then an iterative solution can assume the following algorithm

$$x^{(m+1)} = f\left[x^{(m)}\right] \tag{30}$$

Alternately, if $F(x)=0$ cannot be written in the form given by equation 29, the algorithm

$$x^{(m+1)} = x^{(m)} + \Gamma F(xm) \tag{31}$$

where $\Gamma \neq 0$ is some appropriately chosen constant can be used.

Assuming $F(x)=0$ has a solution, the normal iterative process is to assume a value of $x=x^{(0)}$, use this value in equation 30 (or 31) and solve for $x^{(1)}$. Then $x^{(1)}$ is used to find $x^{(2)}$ and so on until the absolute value of the residual (i.e. $|x^{(m+1)} - x^{(m)}|$) is less than some specified criterion. The sequence of values of $x^{(m)}$ will show any one of the following characteristics (Ref 8:9)

1. Oscillate and converge
2. Oscillate and diverge
3. Converge monotonically
4. Diverge monotonically

To obtain the accelerating technique, the basic iterative equation is modified to give (Ref 8:9)

$$\tilde{x}^{(m+1)} = q\tilde{x}^{(m)} + (1-q)x^{(m+1)} \tag{32}$$

where q is the accelerating factor

the tilde ($\sim$) indicates a value computed by the accelerating technique

Wegstein states that with the appropriate selection of "q", one may cause convergent cases to converge more rapidly and divergent cases to become convergent.

The procedure used is best described by a diagram of the process (Ref 8:11). This diagram is provided in figure 3. Starting with the estimate $\tilde{x}^{(m)} = x^{(0)}$, compute an improved value $x^{(m+1)}$ using $x = f[\tilde{x}^{(m)}]$. On this first iteration, the modified form (equation 32) is not used and the values for the next iteration are set up thru the center path on the diagram. On subsequent iterations, the process loops thru the outside loop until the residuals are as small as desired.

The derivation of "q" may be described in the following manner (Ref 8:9-11). "Geometrically, the solution of $x = f(x)$ amounts to the problem of finding the point of intersection P (see figure 4) of the curves $y = x$ and $y = f(x)$. The iteration $x^{(m+1)} = f[x^{(m)}]$ can be represented graphically as follows. Pass a vertical line through a point $[x^{(m)}, x^{(m)}]$ on $y = x$ so that it intersects the curve $y = f(x)$ at some point A with the coordinates $[x^{(m+1)}, x^{(m+1)}]$."

"The ideal location for $\tilde{x}^{(m+1)}$ of $\tilde{x}^{(m+1)} = q\tilde{x}^{(m)} + (1-q)x^{(m+1)}$ on AB would, of course, be the intersection point C with the normal to AB drawn through P. Thus, 'q' should be chosen such that

$$\frac{q}{1+q} = \frac{\overline{BC}}{\overline{AC}} \qquad (33)$$

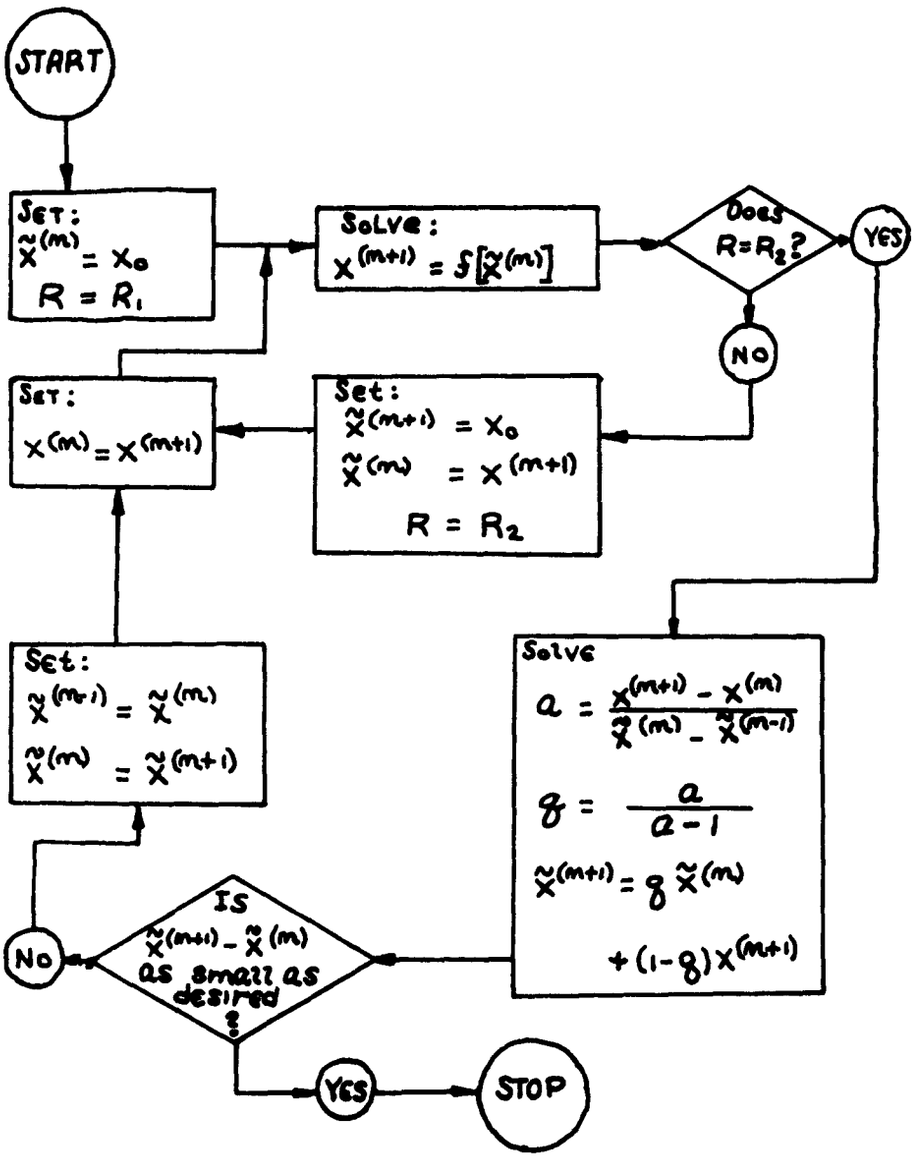


Fig. 3

Schematic Diagram of the Wegstein Technique

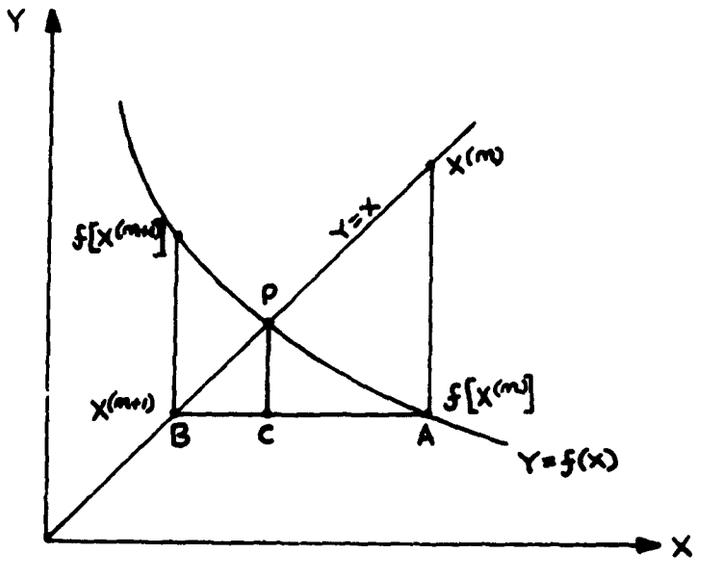


Fig. 4

A Graphical Representation of the Wegstein Accelerating Factor

To determine 'q' approximately, observe that $\overline{PC} \cong \overline{BC}$ and

$$-a = \overline{PC} / \overline{AC} \tag{34}$$

where 'a' is a value of f'(x) between A and P. Thus,

$$-a = \frac{q}{1-q} \tag{35}$$

or

$$q = \frac{a}{a-1} \tag{36}$$

Since a more convenient expression is lacking, 'a' can be approximated by a suitable difference quotient"

$$a \approx \frac{f[x^{(m)}] - f[x^{(m-1)}]}{x^{(m)} - x^{(m-1)}} = \frac{x^{(m+1)} - x^{(m)}}{x^{(m)} - x^{(m-1)}} \tag{37}$$

A variation of the above is to compute a new value of "q" after each iteration using

$$a \approx \frac{x^{(m+1)} - x^{(m)}}{\tilde{x}^{(m)} - \tilde{x}^{(m-1)}} \tag{38}$$

Adaptation to Simultaneous Equations

Kaplan and Clark used the basic Wegstein technique and through a series of experiments empirically adapted it to the solution of the system of equations resulting from the implicit finite difference representation of the equation of transient heat transfer. The forms of the equations used are identical to equations 32, 36, and 38

$$\tilde{U}^{(m+1)} = q\,\tilde{U}^{(m)} + (1-q)\,U^{(m+1)} \tag{39}$$

where

$$q = \frac{a}{a-1} \tag{40}$$

$$a = \frac{U^{(m+1)} - U^{(m)}}{\tilde{U}^{(m)} - \hat{U}^{(m-1)}} \qquad (41)$$

The basic solution procedure used is the method of successive displacements (Ref 6:80).

The steps of the adapted process is presented in simplified diagrammatic form in figure 5 (Ref 5:10). On the first iteration, the initial values are set up but no acceleration is attempted. This process of setting up but not accelerating the solution is continued until the solution proceeds to a specific number of iterations. This number is predetermined prior to running the program and is input data to the computer. When the appropriate iteration for the first application of the technique has been reached, a slope "a" is computed for each node, tested to insure that it is less than one and greater than zero, and if the criteria is met, an accelerating factor "q" is computed (Ref 6:80). This value of "q" that has been computed for each node is further tested to insure that it does not exceed a given maximum value and if the value exceeds the maximum, "q" is set equal to the maximum value. Should "q" not exceed the maximum, the value computed from the slope is used. Then the value of "q" resulting from this test is used to accelerate the solution using equation 39. The accelerated value of temperature then replaces the value computed by the method of successive displacements in the computer memory and the process proceeds to the next node. Should the slope test fail, the accelerating technique is not

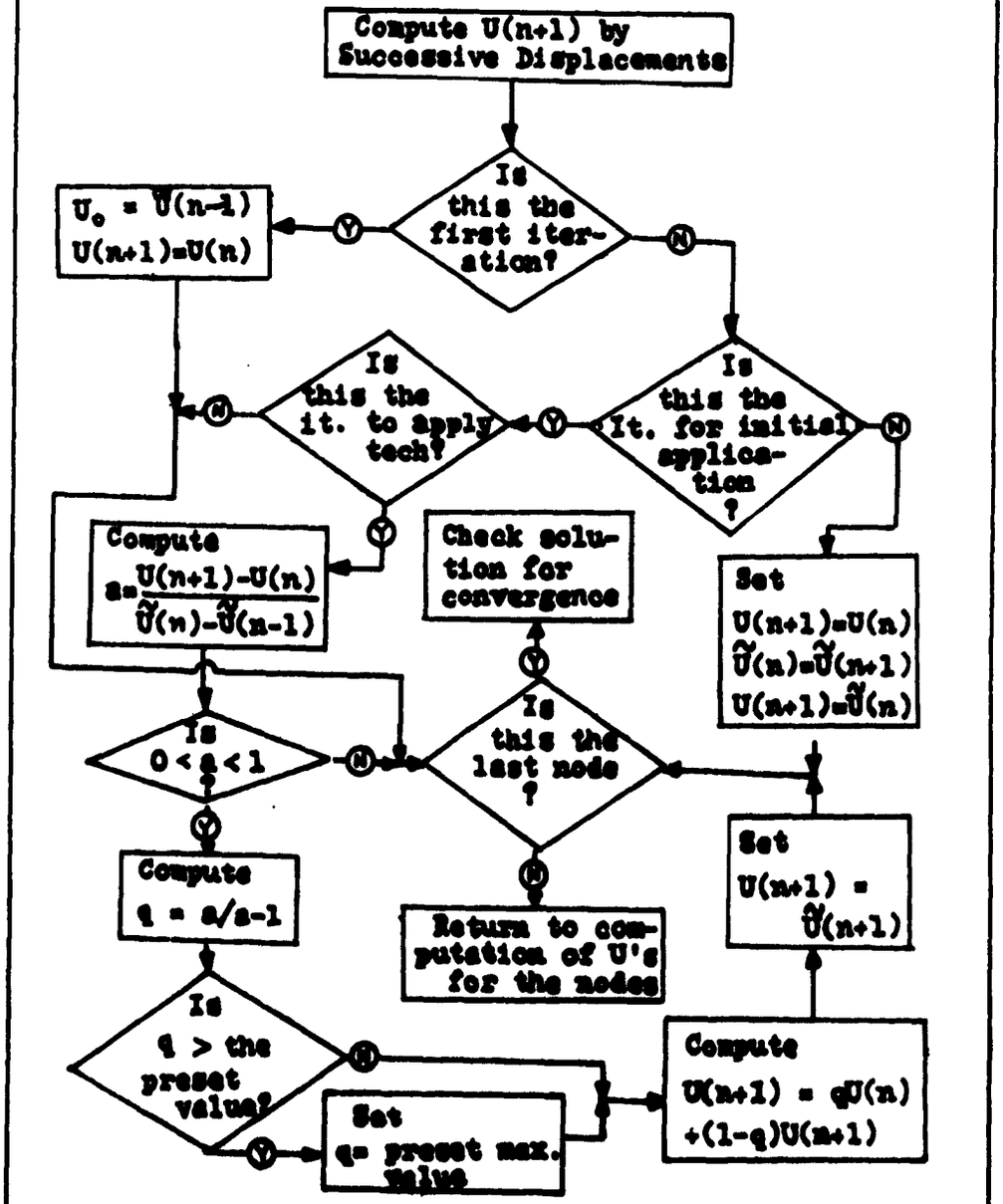


Fig. 5

Block Diagram of the Adapted Wegstein Technique

applied and the value of temperature computed by successive displacements is used in subsequent computations. Following this initial application of the technique to selected nodes, the pertinent tests and applications of the accelerating factor are only made at a repetitious interval of iterations specified by an input to the computer (e.g. once every five iterations). For those iterations when the technique is not applied, the value of temperature computed by the method of successive displacements is used as before the first application.

In the process of refining this adaptation of the Wegstein technique, Kaplan and Clark made the following observations on the size of pertinent parameters.

1. The correct iteration to apply the technique is when the value of the slope "a" is greater than zero or less than one.

2. If the technique is applied too soon, the solution may actually be slowed down. Kaplan and Clark provide the following criterion for the first application of the technique. The initial application should be made on the iteration which is equal in number to the depth of the deepest node from the boundary (Ref 5:6). For example, a one dimensional problem of 30 nodes has the deepest node 15 from the boundary, so the initial application of the accelerating technique is made on iteration number 15.

3. The solution may also be slowed down if the technique is applied too often. This result is attributed to the fact that "the slopes of all the functions do not settle down immediately" (Ref 5:2). The criterion suggested for frequency of application of the technique is given as one half the distance to the deepest node from the boundary (Ref 5:6). Thus, for the example in paragraph 2 above, the technique would be applied every seventh iteration.

A Comparison of Successive Overrelaxation and the Adapted Wegstein Technique

An apparent similiarity between the method of successive displacements and the adapted Wegstein technique can be demonstrated. This relationship may be shown as follows. For convenience let

$$U_{SD}^{(m+1)} = U_{(x,y,z,t+\Delta t)}^{(m+1)}$$

as computed by the method of successive displacements. Then the equation used for overrelaxation may be expressed as

$$U_{(x,y,z,t+\Delta t)}^{(m+1)} = \omega\left[U_{SD}^{(m+1)}\right] + (1-\omega)U_{(x,y,z,t+\Delta t)}^{(m)} \quad (43)$$

The equation for the adapted Wegstein technique may be rewritten in the form

$$\tilde{U}_{(x,y,z,t+\Delta t)}^{(m+1)} = (1-q)\left[U_{SD}^{(m+1)}\right] + q\tilde{U}_{(x,y,z,t+\Delta t)}^{(m)} \quad (44)$$

Assuming a correspondence between $U_{(x,y,z,t+\Delta t)}^{(m)}$ and

$\tilde{U}^{(m)}_{(x,y,z,t+\Delta t)}$, it can be seen that

$$\omega = 1 - q \tag{45}$$

Since, furthermore, it is known that ω is restricted to values between 1 and 2, it would follow that for the relationship between "q" and ω to hold, the range of values of "q" would be

$$-1 < q < 0 \tag{46}$$

The fact that q is negative is consistent with the findings of Kaplan and Clark (Ref 5:5). Kaplan was not as restrictive on the maximum negative value of "q", however. He presents one series of runs where "q" was restricted to values between -100 and 0.

Further evidence of a potential tie-in between the two methods of acceleration is found in the discussion of estimating the relaxation factor by running with $\omega = 1$ to obtain an estimate of the maximum eigenvalue $\eta_1^{(1)}$. As previously shown, the optimum relaxation factor may be found from the equation

$$\omega_{opt} = \frac{2}{1 + (1 - \eta_1^{(1)})^{1/2}} \tag{20}$$

In addition, $\eta_1^{(1)}$ may be expressed in the expanded form

$$\eta_1^{(1)} = \lim_{m \to \infty} \frac{\left[\sum_{L=1}^{M} \left(U_L^{(m+2)} - U_L^{(m+1)}\right)^2\right]^{1/2}}{\left[\sum_{L=1}^{M} \left(U_L^{(m+1)} - U_L^{(m)}\right)^2\right]^{1/2}} \tag{47}$$

Now, since $1 < \omega_{opt} < 2$ from equation 20 above, the limits on $\eta_1^{(1)}$ can be shown to be

$$0 \leq (1-\eta_i^{(i)})^{1/2} \leq 1$$

or since $(1-\eta_i^{(i)})^{1/2}$ is positive,

$$\begin{aligned} 0 &\leq (1-\eta) \leq 1 \\ 1 &\leq -\eta \leq 0 \\ 1 &\geq \eta \geq 0 \end{aligned} \tag{48}$$

By comparison, the equation for the slope test of the adapted Wegstein technique is given by

$$a = \frac{U^{(m+1)} - U^{(m)}}{U^{(m)} - U^{(m-1)}} \tag{41}$$

The similiarity between this slope equation and the equation for $\eta_i^{(i)}$ is obvious. An additional similiarity is found in the fact that Kaplan experimentally determined that the slope should be restricted to values between 0 and 1, which is compatible with the restriction on η.

Though there is a definite similiarity in the way $\eta_i^{(i)}$ and "a" are computed, a major difference lies in the method of application of the factor computed from "a" and $\eta_i^{(i)}$. In the adapted Wegstein technique an accelerating factor "q" is computed separately for each grid point, while for overrelaxation the same relaxation factor is used for all grid points.

Further evidence that the two methods are closely related can be derived using an expression for accelerating the convergence of the method of successive displacements given by V. N. Faddeeva (Ref 3:241). Her expression is

$$\bar{x} \approx \bar{x}^{(m)} + \frac{\bar{x}^{(m+1)} - \bar{x}^{(m)}}{1-\mu_i} \tag{49}$$

where $\bar{X}$ is a vector

μ_i is determined from the ratio of the components of the vectors $\bar{X}^{(m+1)} - \bar{X}^{(m)}$ and $\bar{X}^{(m)} - \bar{X}^{(m-1)}$

Since the notation is in vectors, consider an n component vector for $\bar{X}$.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \approx \begin{bmatrix} x_1^{(m)} \\ x_2^{(m)} \\ x_3^{(m)} \\ \vdots \\ x_N^{(m)} \end{bmatrix} + \frac{1}{1-\mu_i} \left\{ \begin{bmatrix} x_1^{(m+1)} \\ x_2^{(m+1)} \\ x_3^{(m+1)} \\ \vdots \\ x_N^{(m+1)} \end{bmatrix} - \begin{bmatrix} x_1^{(m)} \\ x_2^{(m)} \\ x_3^{(m)} \\ \vdots \\ x^{(m)} \end{bmatrix} \right\} \tag{50}$$

Equating the first component

$$x_1 \approx x_1^{(m)} + \frac{1}{1-\mu_i} \left[x_1^{(m+1)} - x_1^{(m)} \right] \tag{51}$$

rationalizing and rearranging

$$x_1 \approx \frac{(1-\mu_i) x_1^{(m)} + x_1^{(m+1)} - x_1^{(m)}}{1-\mu_i}$$

$$x_1 = \left[\frac{1}{1-\mu_i} \right] x_1^{(m+1)} - \left[\frac{\mu_i}{1-\mu_i} \right] x_1^{(m)} \tag{52}$$

Now let

$$1 - \omega = \frac{\mu_i}{\mu_i - 1} \tag{53}$$

then $-\omega = \frac{\mu_i}{\mu_i - 1} - 1 = \frac{\mu_i - \mu_i + 1}{\mu_i - 1} = \frac{1}{\mu_i - 1}$

$$\omega = \frac{1}{1-\mu_i} \tag{54}$$

Substituting these expressions into equation 51, one obtains

$$x_i \approx \omega x_i^{(m+1)} + (1-\omega) x_i^{(m)} \tag{55}$$

which is the form for overrelaxation.

Alternately, let

$$q = \frac{\mu_i}{\mu_{i-1}} \tag{56}$$

then

$$1-q = 1 - \frac{\mu_i}{\mu_{i-1}} = \frac{\mu_{i-1}-\mu_i}{\mu_{i-1}} = \frac{1}{1-\mu_i} \tag{57}$$

so that one may write equation 51 in the form

$$x_i \approx (1-q) x_i^{(m+1)} + q x_i^{(m)} \tag{58}$$

which is, of course, the form of the adapted Wegstein technique without the tildes ($\sim$).

As in successive overrelaxation, μ_i is described as the largest proper number (eigenvalue) of the matrix A_i, where

$$\bar{\bar{A}}_i = (\bar{\bar{I}} - \bar{\bar{B}})^{-1} \bar{\bar{C}} \tag{59}$$

$$\bar{x} = \bar{\bar{A}}_i \bar{x} + \bar{\bar{F}} \tag{60}$$

and B and C are the triangular constituent matrices of the matrix A of the system

$$\bar{x} = \bar{\bar{A}} \bar{x} + \bar{\bar{F}} \tag{61}$$

Though the system under consideration is of the form $\bar{\bar{A}}\bar{U} = \bar{C}$ it can be simply demonstrated that this form can be rewritten in the form of equation 60. (see Appendix B)

Return to the assumed correspondence between $U^{(m)}(x,y,z,t+\Delta t)$ and $\tilde{U}^{(m)}(x,y,z,t+\Delta t)$. As previously described, the tilde ($\sim$) indicates a value computed by the Wegstein technique as differentiated from a value computed by the method of successive displacements. In a sense, the $U^{(m)}(x,y,z,t+\Delta t)$ value of overrelaxation should also have a tilde since it represents a value computed using the accelerating factor (ω). This analogy is not as straight forward as it seems, however. The Wegstein technique as adapted by Kaplan and Clark is not applied at every iteration, whereas the overrelaxation factor is. Since for those iterations where the Wegstein technique is not applied, the $\tilde{U}^{(m)}$ is set equal to the $U^{(m)}$ value, if the application of the Wegstein technique is restricted to every other iteration, the tilde loses its significance. Further, if application of the technique is restricted to every fourth iteration, the tilde loses its significance in the equation for computing the slope "a". Since the frequency of application is usually equal to the number of nodes to the deepest node from the boundary, and a problem only four nodes deep is quite small, the frequency of application is generally greater than every fourth iteration.

One more important point must be made. The Wegstein technique for a single equation can force convergence on a divergent iteration process. When, however, the method was

adapted to solution of a system of linear equations, no determination was made as to whether convergence could be forced on a diverging set of linear equations. On the other hand, the method of successive overrelaxation cannot force convergence on a diverging set of linear equations.

III. Procedures

Computer Programming

The basic computer used for this study was the IBM 1620 with associated IBM 1623 additional memory unit and IBM 1622 card reader/punch. The 1620 proved to be a relatively slow system and some of the computer runs would have taken an excessive amount of time on this system. To circumvent this situation, the IBM 7090 computer of the Aeronautical Systems Division was employed for the longer runs. As it turned out, the IBM 7090 using the relatively new FORTRAN IV programming system proved to be well over 100 times faster than the IBM 1620 for the programs run. Lack of funds prevented exclusive use of the IBM 7090, however.

The FORTRAN language system and its variations were used for programming; FORTRAN II for the three dimensional programs run on the IBM 1620 and FORTRAN IV for the IBM 7090. Pertinent programs are included in Appendix D. It should be emphasized that these programs were written in the simpliest possible manner consistent with the requirement to provide an accurate specific set of data. No attempt was made to generalize a given program to fit a number of situations. The individual programs were not specifically reviewed for possible revisions to reduce running time or required memory spaces. The single exception is the program

for the analytical solution of the sample problem. Running time was particularly critical on this program, so it was thoroughly reviewed to insure that the program steps provided the fastest solution procedure consistent with FORTRAN language.

Standard Used for Comparison of Speed of Convergence

From the discussion above, one can visualize the difficulty in using time as a measure of speed of convergence. The problem is further complicated by the fact that different computers may do a given operation in different ways and in different times. Thus one must, so to speak, nondimensionalize the rate of convergence and machine time to solution to have a meaningful standard. This is generally accomplished by using the number of iterations to solution as the standard between iterative procedures rather than time. An iteration is defined as the computation required to make a single improvement in the values of all unknowns. This study uses the number of iterations to compare the various methods in rate of convergence.

Determining the Relaxation Factor to Use for Successive Overrelaxation

Since there is no general method for computing the optimum relaxation factor (ω) prior to the start of the iterative

process, there is a problem of what factor to use for overrelaxation in the comparison of methods. The emphasis in this study is on the practical application of a given method to the solution of the transient heat transfer problem. Thus, to be practical the method of obtaining the relaxation factor (ω) should be automated to the maximum extent possible and should require a minimum amount of prior knowledge of the successive overrelaxation process by the using engineer. Of all the methods suggested for estimating ω, the estimation of the maximum eigenvalue by running with $\omega=1$ fits the above criteria better than other methods discovered in the literature. This method was used to provide the number of iterations to solution for the method of successive overrelaxation.

The above selection does not preclude an interest in what successive overrelaxation can do under optimum conditions, however. To determine the optimum factor needed for optimum overrelaxation, a plot of iterations versus various relaxation factors was used (see page 16).

Error Measurement

Since there usually is no exact procedure for determining the error associated with a given finite difference approximation, a common procedure is to select a problem with an analytical solution to use as the standard for comparison with the finite difference solution. This procedure was adopted in this study.

The Sample Problem Employed

Again using the criterion of practicality, a three-dimensional problem was chosen. An analytical solution for a three-dimensional problem for heat transfer over a rectangle with a constant internal heat source was found in literature. Specifically, if heat is produced for $t > 0$ at the constant rate $\dot{q}$ per unit time unit volume in a rectangle of dimensions $x = b$, $y = c$, $z = d$ and the surface is kept at zero degrees, then the temperature over the region $0 < x < b$, $0 < y < c$, $0 < z < d$ is given by (Ref 1:363)

$$u = \frac{64\, a\, \dot{q}}{K \pi^3} \sum_{l=0}^{\infty} \sum_{m=0}^{\infty} \sum_{m=0}^{\infty} \frac{\sin\left\{(2l+1)\frac{\pi x}{b}\right\} \sin\left\{(2m+1)\frac{\pi y}{c}\right\} \sin\left\{(2m+1)\frac{\pi z}{d}\right\} \left(1 - \epsilon^{-\alpha t}\right)}{(2l+1)(2m+1)(2m+1)\,\alpha}$$

where

$$\alpha = \pi^2 a \left[\frac{(2l+1)^2}{b^2} + \frac{(2m+1)^2}{c^2} + \frac{(2m+1)^2}{d^2}\right]$$

Choosing to work in three dimensions did limit the size of the problem to be solved due to the limited storage capacity of the IBM 1620 system. When the above problem was solved, a cube one centimeter on a side was used for the region of interest. It was then necessary to use symmetry to permit selection of grid sizes as small as 0.1 cm.

Diffusivity (a) and conductivity (K) were chosen as 1.0 cm^2/sec and 1.0 cal/cm-sec-°C respectively. These values approximate the parameters for copper whose true values are 1.14 cm^2/sec for diffusivity and 0.93 cal/cm-sec-°C for conductivity (Ref 1:497). The source strength ($\dot{q}$) was limited to 2000 cal/cm^3-sec to permit final temperatures within the region to be of a physically reasonable size. Unless otherwise stated, the above values of a, K, and $\dot{q}$ were used in all the solutions obtained.

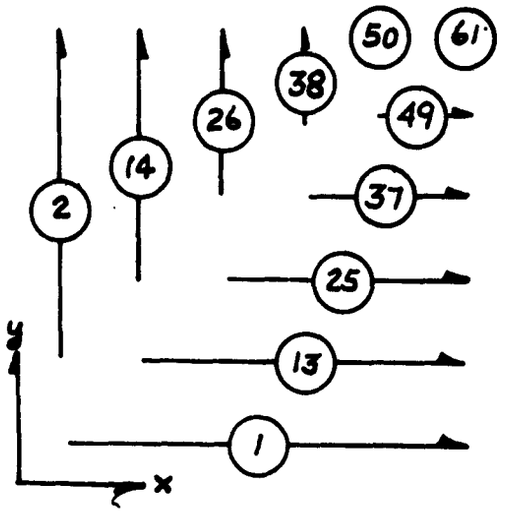
Comparing Spiral Versus Serial Scan

If the size of the residuals are observed after each iteration, one immediately notices that the residuals for points close to the fixed boundary conditions meet the convergence criterion much sooner than those furtherest from the boundary. Intuitively, one could see where the repeated use of at least one "correct" value for the adjacent nodes to compute the value of the point being considered could conceivably hasten the convergence of this point to the desired solution. Thus, it seems logical to assume that any scanning procedure which would use all the boundary values as soon as possible in the solution sequence would converge faster than a serial scan. The optimum scan for achieving this condition is to spiral inward from the boundary.

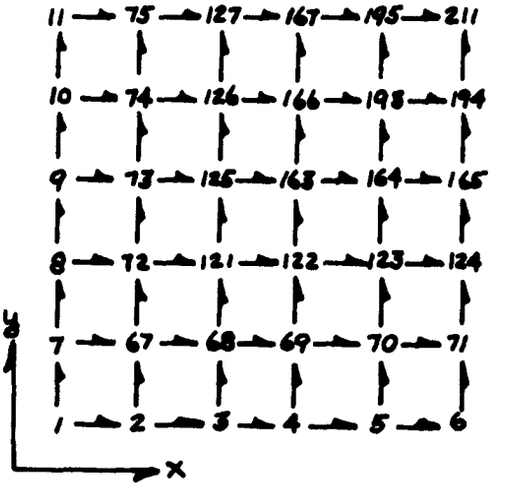
The particular spiral scans used are shown in figures 6 and 7. As one can see from the figures, the three-dimensional case where symmetry is used can be classed as a consistent scan (see page 22), while the full spiral used in the two-dimensional case is nonconsistent. The full spiral case was restricted to two dimensions by the lack of storage capacity of the IBM 1620 system used in the computations. The particular full spiral used was chosen to permit maximum use of the "DO loops" of the FORTRAN language system, thereby achieving a degree of simplicity in the program.

While testing various types of spirals for consistency prior to the choice of the one to use, a conclusion was reached that all full spiral scans are nonconsistent. This conclusion is based on the following fact. As one proceeds normally inward from the boundary following the direction of the higher numbered points, the elementary squares of the mesh show consistency in the region close to the boundary. When the center of the grid is approached, nonconsistencies develop due to the changing direction of the higher numbers. On the other hand, consistent orders exhibit the property of a single general direction for the progression of higher numbered grid points.

The Method of Terminating the Iterative Process



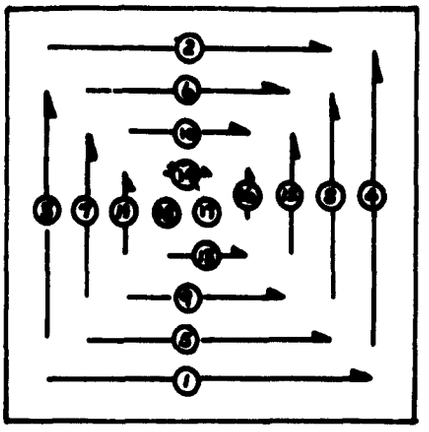
1. By sweep for Z = 0.1 Plane (Scan through sequence 1 and 2 then move back through Z in 1-2 sequence, etc)



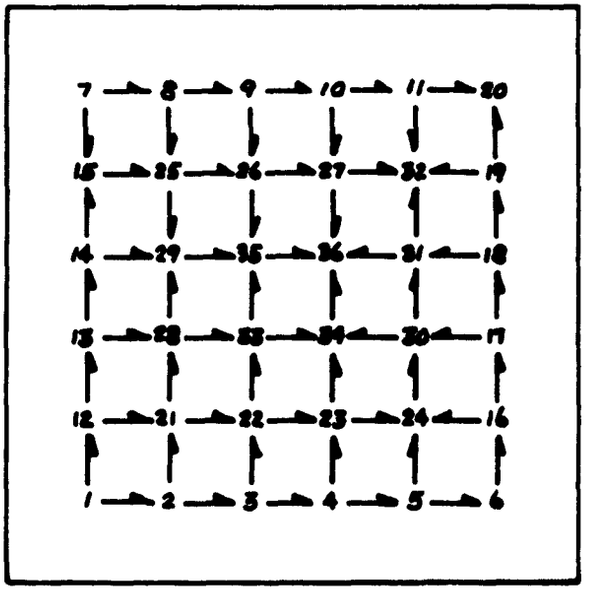
2. By individual grid point for Z = 0.1 Plane

Fig. 6

Diagram of the Three-Dimensional Semi-Spiral Scan Used



1. By sweep



2. By individual grid point

Fig. 7

Diagrams of the Two-Dimensional Full Spiral Scan Used in the Comparison of Spiral and Serial Scans

The iterative process is generally concluded by measuring the size of the residuals and terminating the program when the size of the residuals meet some predetermined criterion. Some possible criteria that may be used are:

1. Requiring the average of all the residuals to be less than some specified number.

2. Requiring the sum of the residuals to be less than some specified number (see page 22).

3. Requiring the individual residuals to be less than a specified number--if one fails the test, all fail.

The fact that all residuals do not converge at the same rate has been mentioned on page 44. In methods 1 and 2 above, any major inbalance in convergence rate could result in a relatively large inaccuracy in the points which converge at the slowest rate. In practical applications, method 3 is used to avoid this potential inaccuracy. Further, method 3 is the most sensitive measure of convergence.

In view of the above, the convergence criterion used in this study required all points to be less than a given number or a new iteration was begun.

Selection of the Initial Estimate of Temperature

A good initial estimate of temperatures throughout the region of interest could reduce the number of iterations required to solve the problem by the method of successive

displacements and the two accelerating techniques used in this study. All runs were made with an initial estimate of zero degrees for the temperature of all internal grid points, however. Aside from reasons of standardization among solution procedures, starting at zero removes any screening of the convergence of the running computation of relaxation factor (using $\omega = 1$) by differences in accuracy among the estimates of temperatures at individual points.

Procedure for the Attempt to Force Convergence Using the Adapted Wegstein Technique

A quick check was set up to investigate whether the adapted Wegstein technique could force convergence. The test was to use the adapted Wegstein technique on the following set of equations which were known to give a divergent solution when the method of successive displacements was used in an attempt to solve the system.

$$\begin{aligned} x_1 + 2x_2 - 2x_3 &= 1 \\ x_1 + x_2 + x_3 &= 3 \\ 2x_1 + 2x_2 + x_3 &= 5 \end{aligned} \tag{63}$$

This set is, incidentally, a case where convergence can be obtained using the method of simultaneous displacements (see footnote page 12).

IV. Results

Serial versus Spiral Scan

The results of the comparison of the rate of convergence of a spiral and serial scan are displayed in tables 2 and 3.

There was no observed advantage in the use of the consistent semi-spiral scan using symmetry as applied to the three dimensional problem, since the number of iterations remained the same for both serial and spiral scan even when the convergence criterion was tightened until the computer cut-off rounding procedure caused all residuals to become zero.

When a two-dimensional problem was solved using a nonconsistent full spiral, some small savings in iterations did occur for the spiral scan. Furthermore, as the grid size (h) was decreased to cause the solution to converge more slowly, the savings in iterations showed a slight increase.

Determining the Optimum Relaxation Factor

Plots of number iterations versus relaxation factor are shown in figures 9 and 10. Figure 8, demonstrating the progress of the temperatures at the center of the cube with increasing time, is also pertinent to this particular dis-

Table 2

Iterations to Solution for a Serial Scan and a Three-Dimensional Semi-Spiral Scan

Scan Type	Total Number of Iterations		
	Residual Size		
	0.005	0.00005	less than 10^{-8}
Serial[a]	91	145	195
Spiral[a]	91	145	195
	Residual Size		
	1.0	0.05	0.005
Serial[b]	31	51	67
Spiral[b]	31	51	67

[a] $t = 2.0$ sec; $h_x = h_y = h_z = 0.1$ cm

[b] $t = 0.2$ sec; $h_x = h_y = h_z = 0.125$ cm

Solutions by the method of successive displacements

Table 3

Iterations to Solution for a Serial Scan and a Two-Dimensional Full Spiral Scan

Scan Type	Total Number of Iterations		
	Residual Size		
	0.5	0.005	0.00005
Serial[a]	29	66	102
Spiral[a]	28	65	101
Serial[b]	69	218	
Spiral[b]	67	215	
Serial[c]	80	688	
Spiral[c]	78	684	

[a] $t = 0.2$ sec; $h_x = h_y = 0.1$ cm
[b] $t = 0.2$ sec; $h_x = h_y = 0.05$ cm
[c] $t = 0.2$ sec; $h_x = h_y = 0.025$ cm
Solutions by the method of successive displacements

cussion. This figure shows that the temperature has essentially reached a steady state value after only 0.2 seconds for the parameters a, K, and $\dot{q}$ used in this study.

Figure 9 clearly demonstrates a dependence of the relaxation factor on the time variable by the shift in the optimum factor for the times of 0.01 and 0.5 seconds. The point that the optimum factor is the same for 0.5 and 2.0 seconds is attributed to the fact that for these times the temperatures within the cube are at their steady state values. This is further confirmed by figure 10, where there is only a small shift in the optimum between 0.2 (nearing steady state) and 2.0 seconds (well into steady state conditions).

A space dependence of the relaxation factor is demonstrated by a combination of the two figures. For a time of 2.0 seconds there is a shift in the optimum factor between the case of unequal grid sizes for the three coordinate directions.

Comparing the Accelerating Factors of the Adapted Wegstein Technique and Successive Overrelaxation

Figures 11 and 12 display the results of making a running estimate of the relaxation factor while iterating with the method of successive displacements. The relaxation factor (ω) was computed for each iteration using the current estimate of the maximum eigenvalue $\eta_i^{(j)}$. This eigenvalue was,

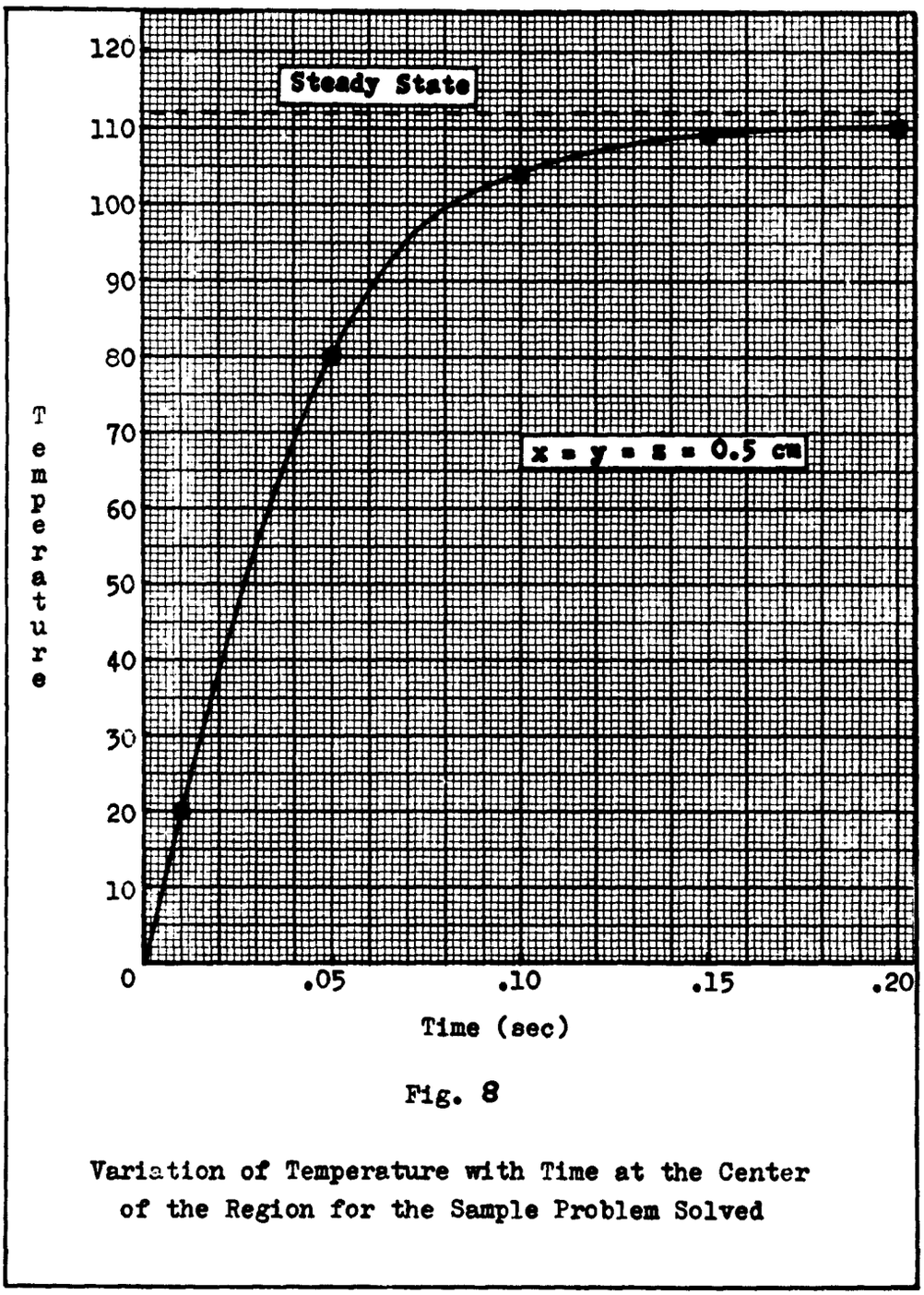


Fig. 8

Variation of Temperature with Time at the Center of the Region for the Sample Problem Solved

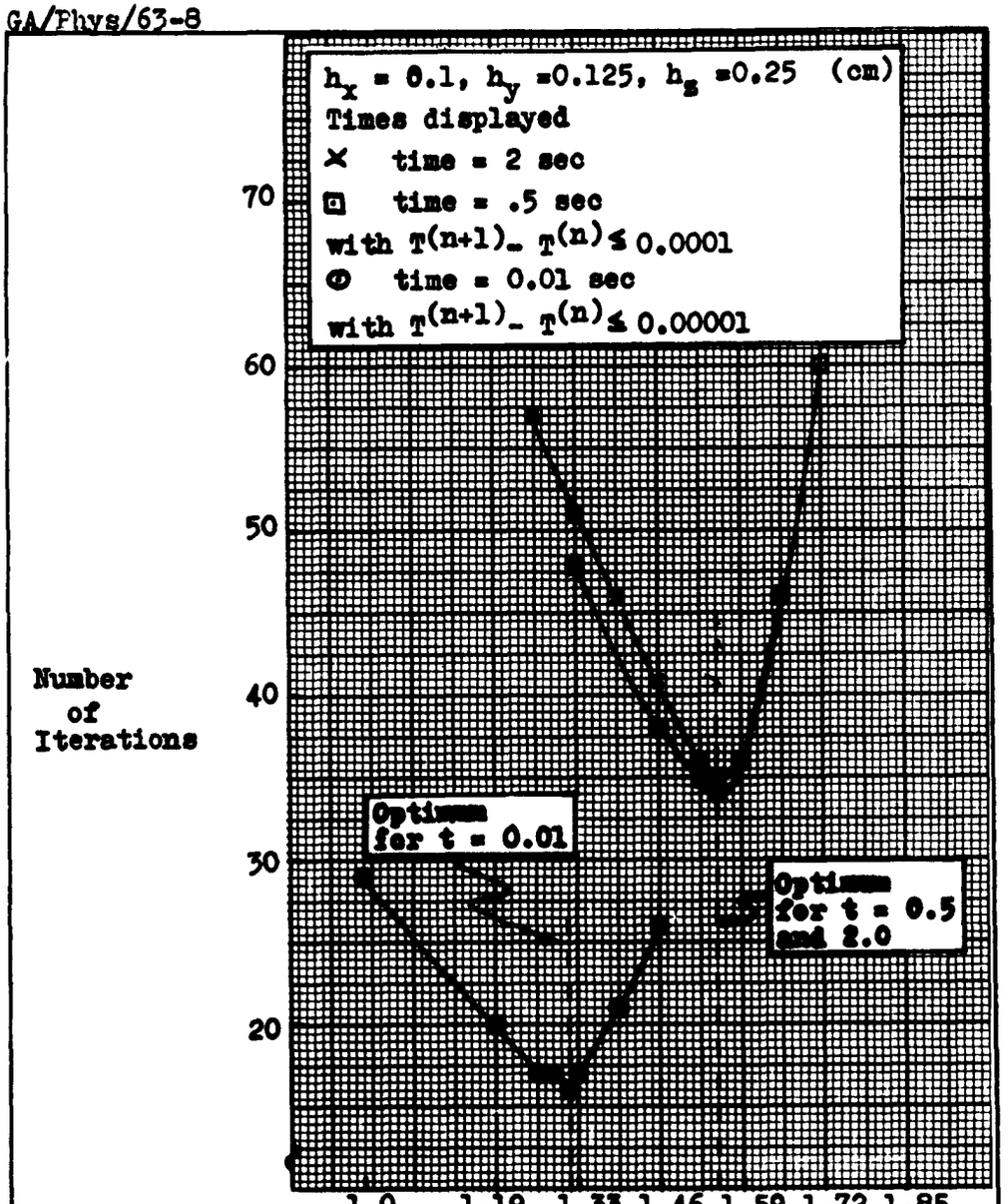


Fig. 9

Variation of Number of Iterations
for Varying Magnitude of Relaxation Factor

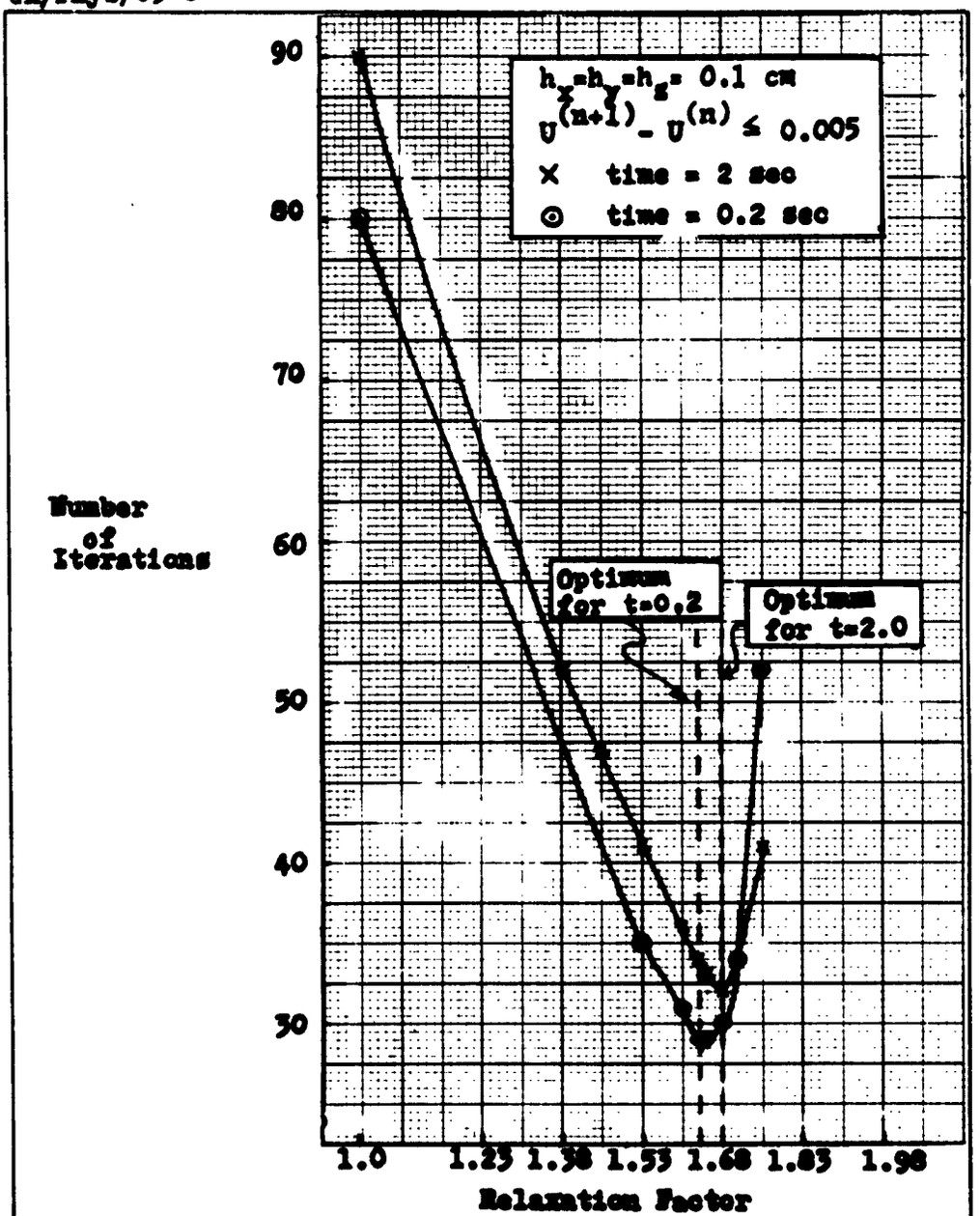


Fig. 10

Variation of Number of Iterations
for Varying Magnitude of Relaxation Factor

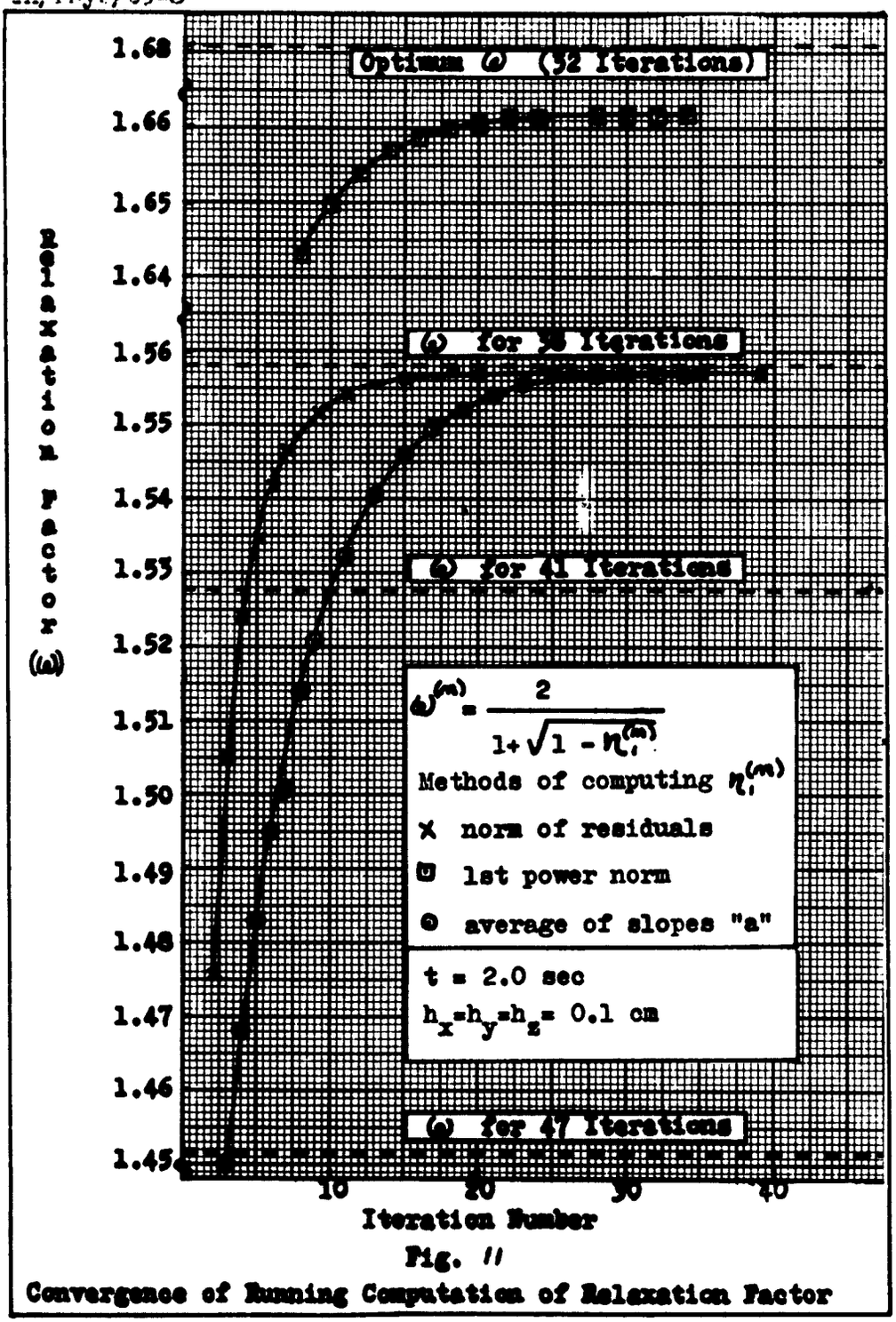


Fig. 11

Convergence of Running Computation of Relaxation Factor

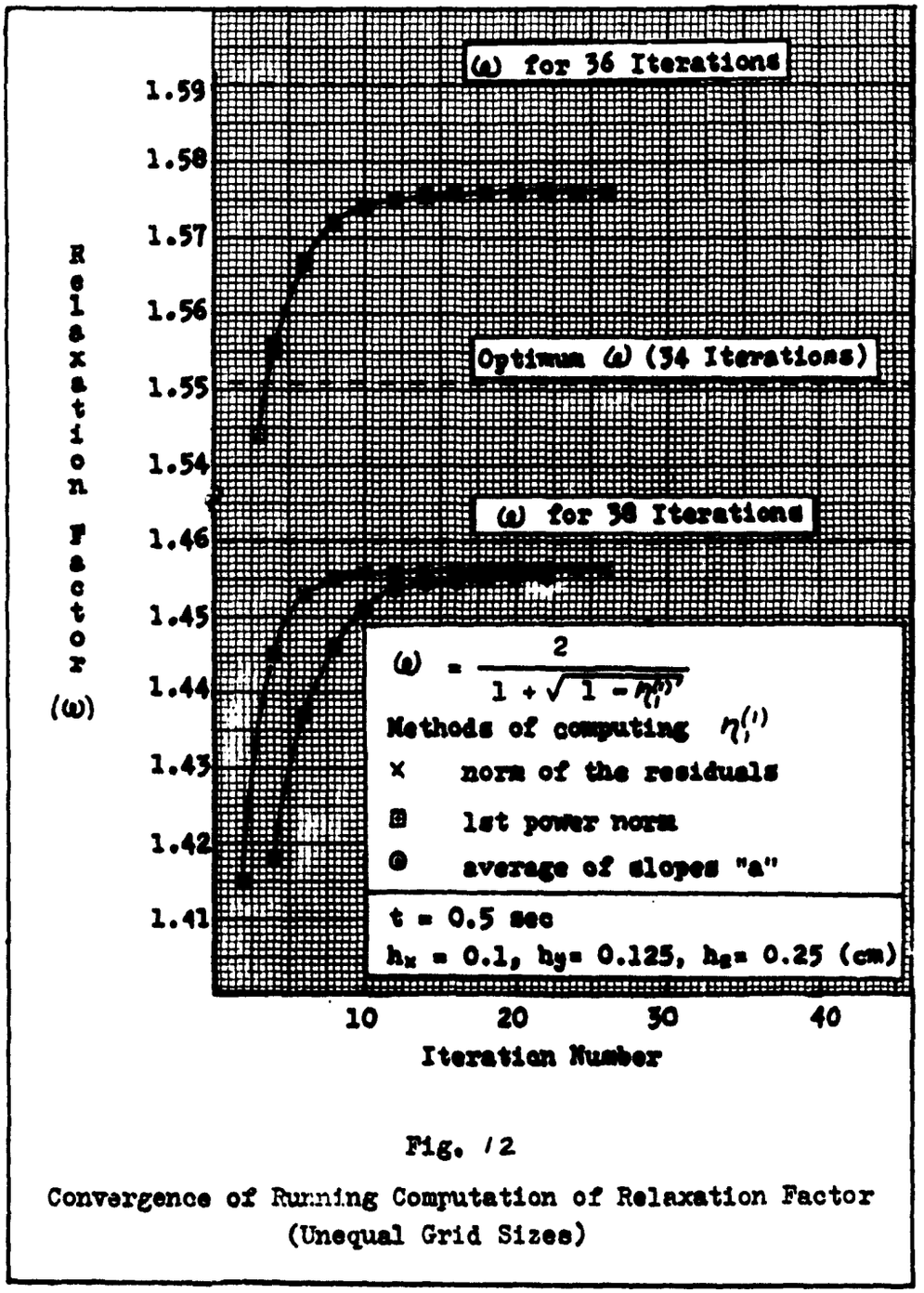


Fig. 12

Convergence of Running Computation of Relaxation Factor
(Unequal Grid Sizes)

in turn, computed in three different ways: as the norm of the residuals (equation 21), as the first power norm of the residuals (equation 23), and as the average of the Aegstein slopes computed from $\sum_{i=1}^{N} a_i$/number of interior nodes. Figure 11 is for equal grid sizes, while figure 12 is for the more general case of unequal grid sizes.

The important features displayed by figures 11 and 12 are:

1. The use of the norm of the residuals to find the optimum factor converged rapidily and smoothly to a value that was less than the known optimum.

2. The procedure using the first power norm converged to a value closer to the optimum factor (once to a value greater than ω_{opt} and once to a value less than ω_{opt}) than the method using the norm, but at a slower rate. In the case of equal grid sizes, for example, if one were to accept a value of ω as the relaxation factor when $\omega^{(m)} - \omega^{(m-1)} < .001$ then, using the norm would provide a value in 12 iterations while the first power norm does not meet this criterion until after 16 iterations.

3. The average value of the slopes does provide an $\eta_i^{(i)}$, and therefore, an ω which converges to the same value as the method using the norm. The convergence is slower, however. To meet the same criteria as used in item 2 above, 21 iterations would be required.

At the same time the above data was computed, the individual slopes of the interior grid points were observed every fifth iteration. The primary observations of interest are:

1. All the individual slopes converged to the same value and this value was identical to the maximum eigenvalue obtained by use of the norm of the residuals.

2. The individual values of the slopes converged in different ways and at different rates. Characteristically, the points nearest the boundary which normally were the fastest to arrive at a solution (i.e. meet the convergence criteria for the residuals) started at values of slope considerably below the final value and converged rather slowly. On the other hand, the points furtherest from the boundary which are normally the last to reach a solution, started with values of slope greater than the final value and converged faster than the points close to the boundary. This fact is displayed in figure 13 for two extreme grid points.

Observations on the Characteristics of the Convergence of the Solution for the Three Procedures Used

Figure 14 is a plot of the progress of the solution with increasing iterations for the node in the center of the cube. The higher rate of convergence for the two accelerating procedures is to be expected. The significant point displayed

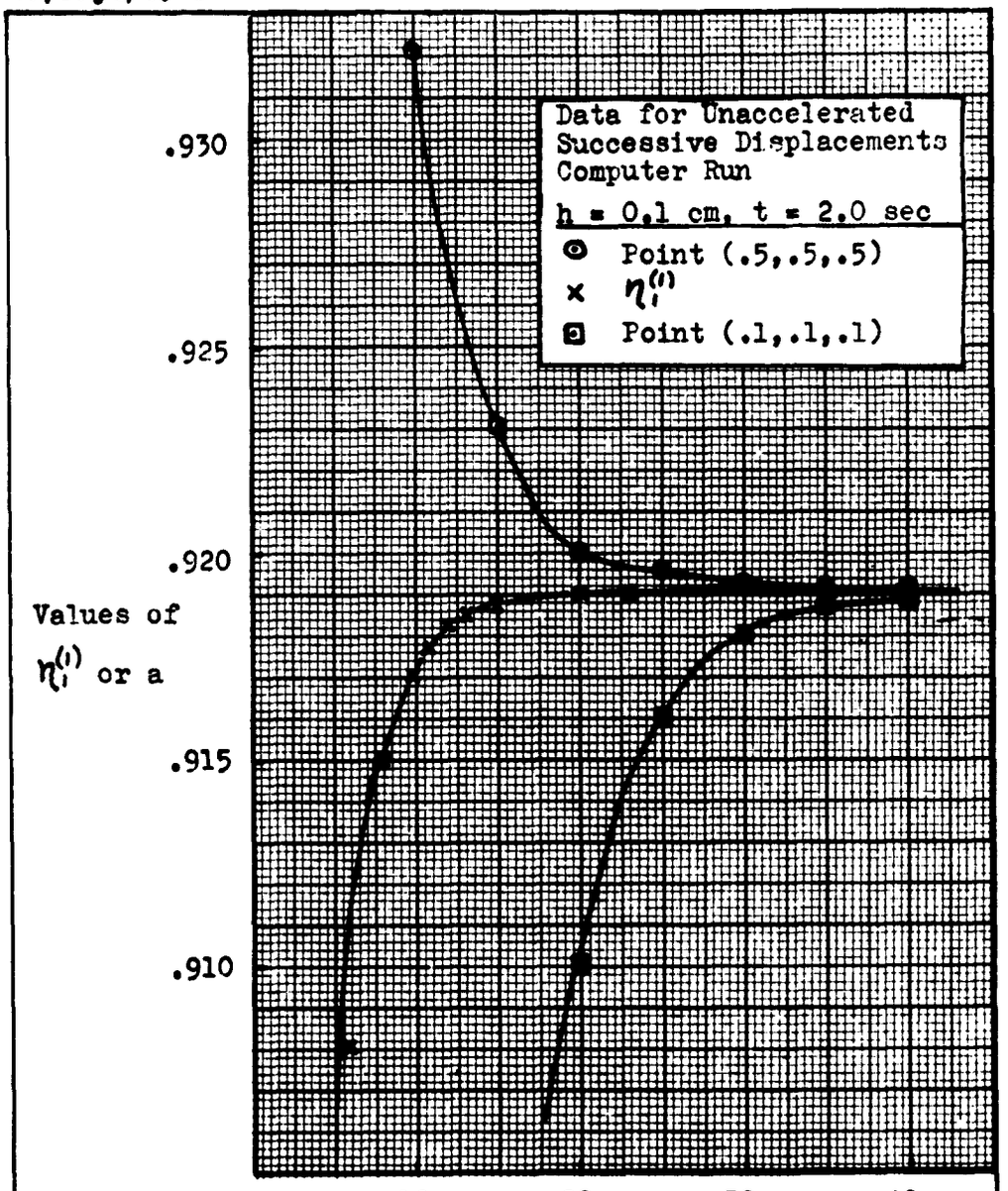


Fig. 13

Convergence of Slopes of Representative Grid Points

by this figure is the contrast between the modes of convergence of the two accelerating procedures. As in the case of successive displacements, the solution by successive overrelaxation converges through a smooth progression of values of temperatures. In contrast to this smooth progression, the convergence of the adapted Wegstein technique is ragged with spurts of acceleration followed by a smoothing process. This raggedness is due, of course, to the intermittent application of the Wegstein technique. Note that as the solution approaches the final value each application of the Wegstein accelerating factor tends to cause the solution to overshoot the final value. This is then followed by a period of use of the method of successive displacements which pulls the value back toward the final temperature. As a matter of fact, for every solution obtained by use of the adapted Wegstein technique the process met the convergence criteria and the program terminated on an iteration that was computed by the method of successive displacements and was about two-thirds or more of the way along toward the next application of the technique. For instance, in the solution displayed in figure 14 the process stopped on iteration number 58 where the last prior application of the technique was on iteration number 54 with the next application due on iteration 60.

There is some danger of misunderstanding being created by showing only one grid point in figure 14. One must recognize that the over-acceleration does not have to occur on the individ-

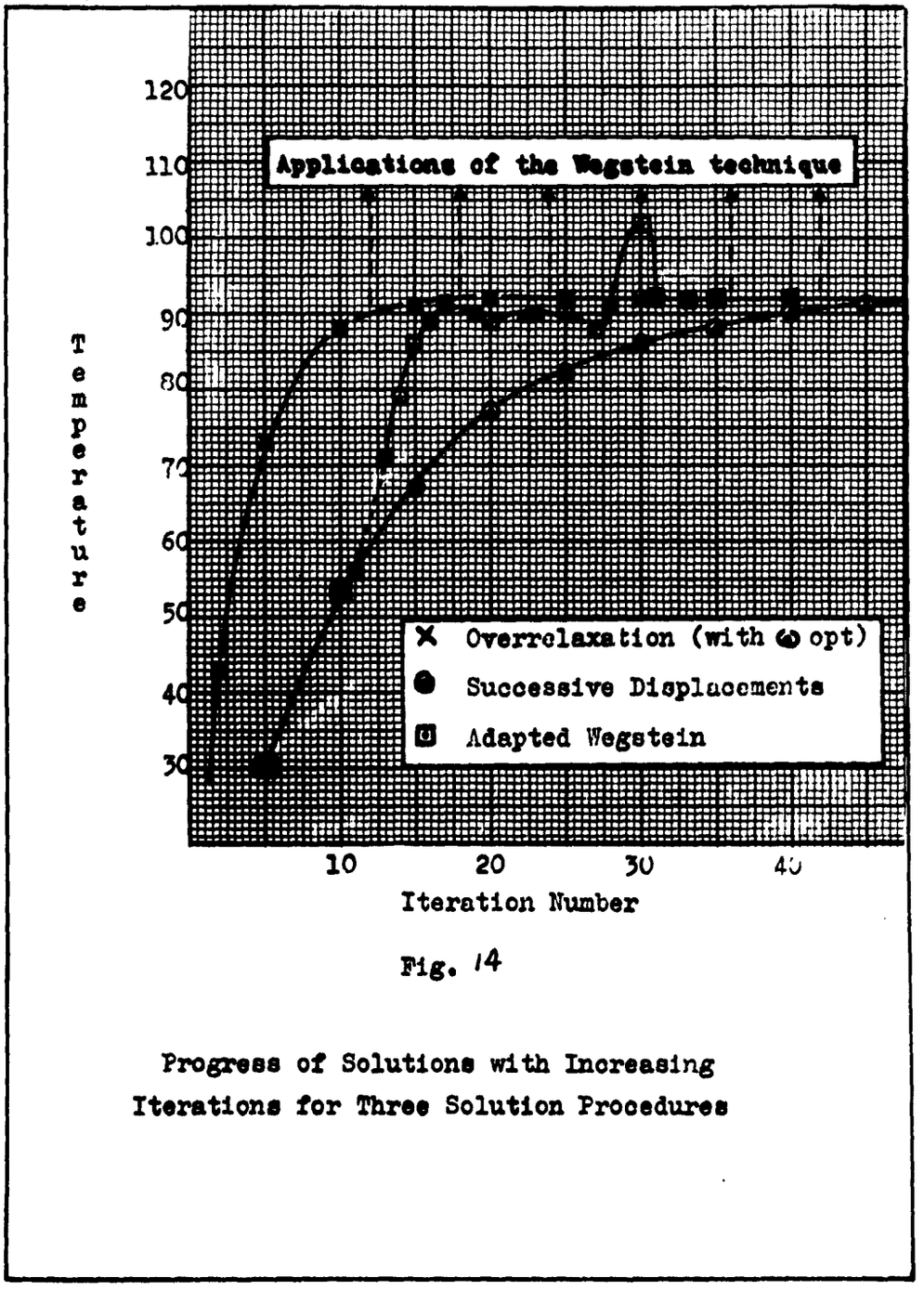


Fig. 14

Progress of Solutions with Increasing Iterations for Three Solution Procedures

ual grid point to cause an overshoot. The computational procedure uses adjacent grid points to calculate the temperature at the point of interest, so any over-acceleration of a nearby point will produce a significant effect on adjacent nodes and a diminishing effect as one moves away from the node that was over-accelerated. Thus, there is an interrelated cause and effect relationship that is not apparent from figure 14.

One added point must be cited. With the parameters selected for solution of the sample problem, the rate of convergence to solution turned out to be higher than desired to demonstrate the effects of acceleration. To circumvent this situation, two basic procedures were used to increase the number of iterations. Since the available memory capacity of the IBM 1620 prevented decrease of grid size below 0.1 cm, tighter convergence criteria than one might expect to choose in a practical problem was used. Second, the span between initial estimate and final temperature was maximized by using times that brought the temperature near or well into steady state conditions. Increasing the source strength and therefore, the final temperature had a negligible effect on increasing iterations due to the increased contribution of the $\dot{q}$ term of equation 10 to the size of the temperature step for each iteration.

Effectiveness of the Accelerating Techniques

Table 4 shows the results of solving the sample problem

Table 4

Results of Accelerating the Method of Successive Displacements by Using Successive Overrelaxation or the Adapted Wegstein Technique

Item / Method	Iterations		Convergence Criteria		Savings	
	Run 1[a]	Run 2[b]	Run 1	Run 2	Run 1	Run 2
Successive Displacements	80	91 145	.005 --	.005 .00005	---	---
Overrelaxation with known $\omega_{optimum}$[c]	29	32 48	.005	.005 .00005	64%	65% 67%
Overrelaxation by Estimating ω While Running with $\omega=1$[d]	42	47 69	.005	.005 .00005	48%	48% 52%
Normal Adapted Wegstein Technique[e]	58	58 89	.005	.005 .00005	28%	36% 39%

[a] For these runs, t = 0.2 sec; Δt = 0.2 sec; all grid sizes were equal at 0.1 cm

[b] For these runs, t = 2.0 sec; Δt = 2.0 sec; all grid sizes were equal at 0.1 cm

[c] The optimum values of the relaxation factor for these runs were taken from figure .

[d] When $\omega^{(m+1)} - \omega^{(m)} < .001$ the running computation of ω was terminated and the current value of $\omega^{(m)}$ was then used to continue the solution by regular overrelaxation. ω was computed using the norm of the residuals to find $\eta_i^{(1)}$.

[e] The deepest node was six from the boundary, so the technique was first applied on the 12th iteration and on every sixth iteration thereafter. q was restricted to a maximum of -100.

for two different times using the same grid size for both problems. The primary items of interest in this table are:

1. Successive overrelaxation using the optimum relaxation factor showed the fastest rate of convergence with savings in iterations of the order of 65% over the method of successive displacements.

2. Significant savings did result from the more practical procedure of making a running estimate of relaxation factor from the residuals of the method of successive displacements.

3. The adapted Wegstein technique was only about 57% as effective in accelerating successive displacements as overrelaxation with the running computation of relaxation factor.

4. Except for the case of the adapted Wegstein technique, savings remained essentially constant between runs using the same convergence criterion. In the case of the adapted Wegstein a significant increase in savings (8%) occurred for the run with the slower rate of convergence. This is due to the ragged nature of the convergence of the adapted Wegstein technique.

5. All methods showed a slight increase in savings for a decrease (tightening) in convergence criterion.

Since the two methods of computing a running estimate of the optimum relaxation factor (first power norm and norm

of the residuals of the method of successive displacements) had approximately equal rates of convergence, a comparison of the two methods was made. Table 5 displays these results and shows that the first power norm further increased the savings in iterations over the procedure using the norm. It is obvious that though the first power norm was slightly slower converging to a final value of relaxation factor, the fact that this method produced a better estimate of the optimum factor than the norm overcame the slower start.

Forcing Convergence with the Adapted Wegstein Technique

Figure 15 displays the results of attempting to force convergence on the set of three simultaneous equations given in equation 63. Only one of the three unknowns is plotted since the iterative solutions exhibited the same general divergent characteristics.

Initially, the normal procedure for using the adapted Wegstein technique was attempted with the initial application on the fourth iteration and every other iteration thereafter. This procedure completely failed because the divergence was such that the slope test failed in every attempt to apply the technique. The process was terminated after 10 iterations.

A second attempt was made using the same frequency of application as in the first attempt with the slope test deleted.

Table 5

A Comparison of Methods of Making a Running Computation of Relaxation Factor for Use with Successive Overrelaxation

Method of Making the Running Computation of ω	Iterations		[e] Savings (%)	
	Run 1[a]	Run 2[b]	Run 1	Run 2
First Power Norm[c]	40	43	50	53
Norm of the Residuals[d]	42	47	48	48

[a] For these runs, $t = \Delta t = 0.2$ sec; all grid sizes equal at 0.1 cm

[b] For these runs, $t = \Delta t = 2.0$ sec; all grid sizes equal at 0.1 cm

[c] See equation 23

[d] See equation 21

[e] All savings compared to the method of successive displacements

All convergence criteria were 0.005

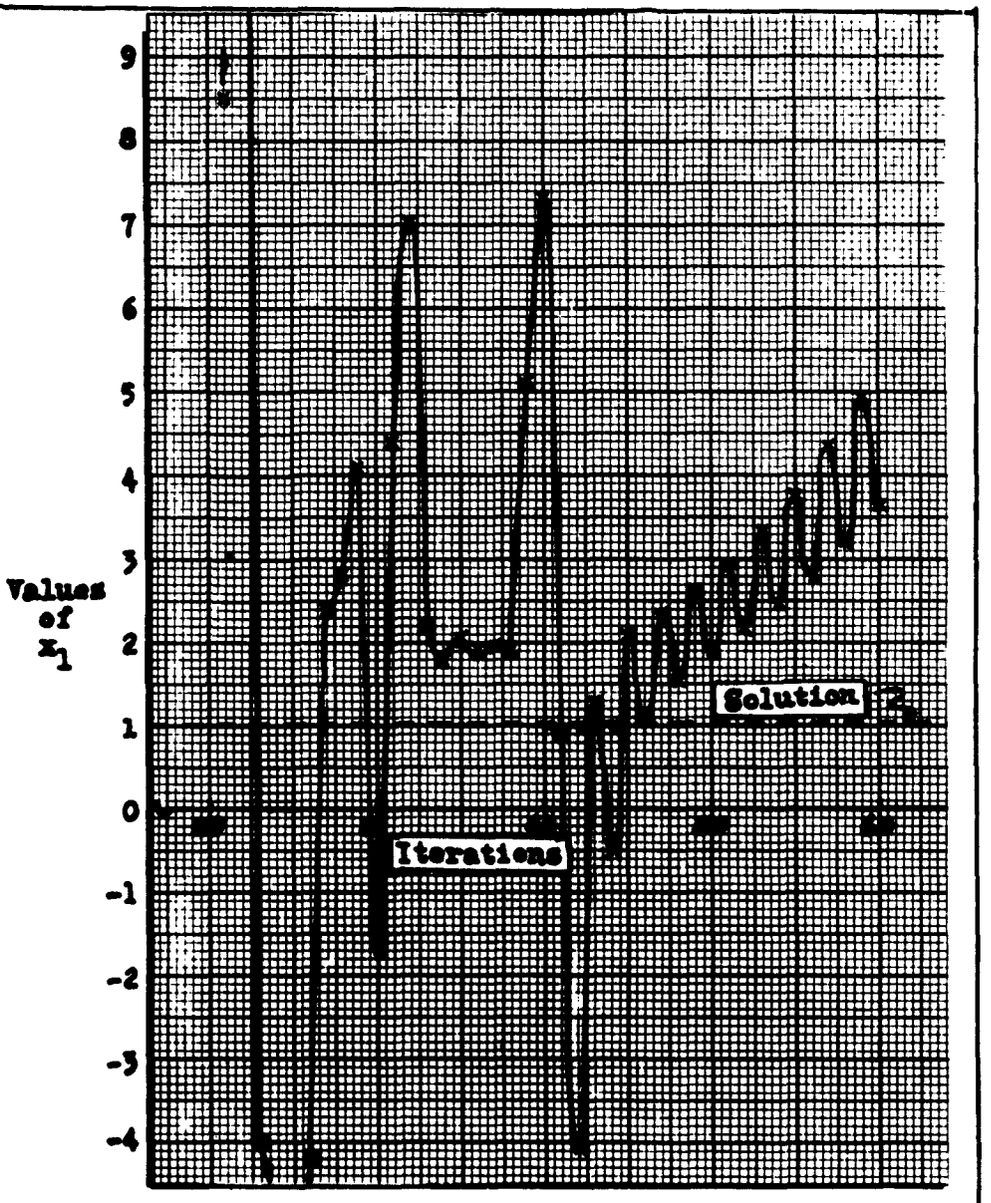


Fig. 15

Results of Attempting to Force Convergence with the Adapted Wegstein Technique

The results obtained are the ones displayed in figure 14.

In a third attempt, the technique was applied at the second iteration and on every other iteration thereafter. The slope test was again deleted. These results displayed the same oscillating divergence as the second case.

Finally, an attempt was tried using the same frequency of application as the third case, but this time the slope test was used again. The slope test still prevented the application of the accelerating technique and the process was terminated after 10 iterations.

Though the solution continued to diverge, there was a change in the mode of divergence when the adapted Wegstein technique was applied. When an attempt was made to solve the system using only successive displacements, the solutions diverged monotonically. With the use of the adapted Wegstein technique, these solutions became oscillating and divergent.

Error

Table 6 compares the error at selected grid points resulting from the three solution techniques used in this study. The important points are demonstrated by this table:

1. The solutions for the successive overrelaxation and adapted Wegstein methods are more accurate than the method of successive displacements. This is a result of the technique

Table 6

Tabulation of Percent Error Resulting from a Given Solution Procedure

Grid Point (cm)			Solution Procedure						
			Analytic	Suc. Displacements		Suc. Overrelaxation		Adapted Wegstein	
x	y	z	Temperature	Temperature	Error (%)	Temperature	Error(%)	Temperature	Error(%)
			t = 0.2 sec	Δt = 0.2 sec		$h_x = h_y = h_z$ = 0.1 cm			
0.1	0.5	0.5	47.01	40.16	14.57	40.17	14.57	40.18	14.56
0.2	0.5	0.5	78.42	65.88	15.99	65.90	15.97	65.92	15.96
0.3	0.5	0.5	98.05	81.40	16.98	81.43	16.95	81.45	16.93
0.4	0.5	0.5	108.7	89.62	17.55	89.66	17.52	89.66	17.52
0.5	0.5	0.5	112.1	92.18	17.77	92.22	17.73	92.25	17.71
			t = 2.0 sec	Δt = 2.0 sec		$h_x = h_y = h_z$ = 0.1 cm			
0.1	0.5	0.5	47.12	45.73	2.95	45.75	2.91	45.75	2.91
0.2	0.5	0.5	78.64	76.12	3.20	76.15	3.71	76.16	3.16
0.3	0.5	0.5	98.35	95.04	3.37	95.08	3.32	95.11	3.28
0.4	0.5	0.5	109.05	105.3	3.44	105.3	3.44	105.4	3.43
0.5	0.5	0.5	112.42	108.5	3.56	108.6	3.56	108.6	3.56

All runs were made with a convergence criteria of 0.005

used for terminating the iterations. Using the residuals as the measure of convergence results in more accurate answers for a rapidly converging solution than for a slowly converging one. Figure 16 demonstrates this change of convergence error due to a change in convergence rate. In the results shown in table 6, this effect is somewhat moderated by the rather tight convergence criteria of 0.005 used to obtain the solutions.

2. The solutions for 2.0 seconds show about one-fifth of the error encountered for 0.2 seconds. Since the time step for 2.0 seconds is ten times greater than 0.2 seconds and truncation error * is a function of grid and time step size, one would expect the reverse results. As previously cited, at 2.0 seconds the solution is well into steady state conditions, while 0.2 seconds is just approaching steady state. These results demonstrate the fact that when one desires to use a transient heat transfer program to obtain a steady state solution, the best results are obtained with a time step that puts one well into steady state conditions. Taking the larger time step permits more complete convergence of the iterative solution.

Testing a Revision to the Adapted Wegstein Technique

The apparent relationship between overrelaxation and the adapted Wegstein technique suggested certain potential

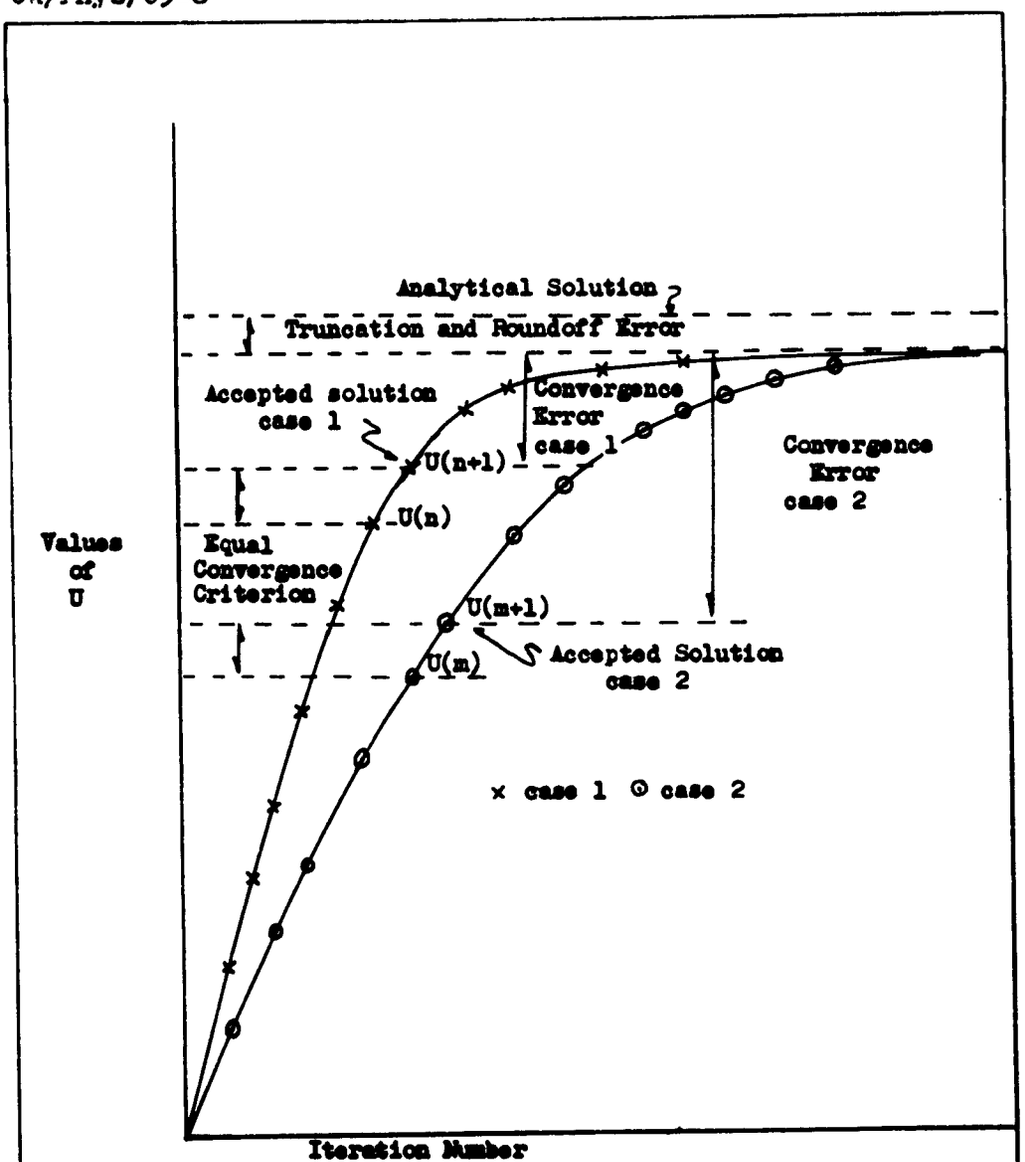


Fig. 16

An Example of Change of Error When the Same Convergence Criteria is Used for Two Problems with Different Rates of Convergence

revisions to the adapted Wegstein technique. A limited amount of time was available to look into possible revisions, so the investigation was restricted to a relatively obvious revision that one might make to smooth the ragged convergence of the regular technique.

The revision attempted was to replace ω by its equivalent in terms of $\eta_i^{(0)}$ (see equation 20) in equation 45 to obtain a new expression for computing "q" of

$$q = 1 - \frac{2}{1 + \sqrt{1 - a}} \qquad (40a)$$

This equation, in conjunction with the slope test restricting values of "a" to $0 < a < 1$, would keep the acceptable values of "q" within the limits specified by equation 46. It was hoped that this revision would also take advantage of potential offered by the individuality of the modes of convergence of the slopes for individual grid points. As previously suggested, the nodes with the slowest rate of convergence to solution have slopes that converge through a sequence of values that, when used in equation 40a, would give better q's as the values are greater than optimum. Since the convergence was assumed to be smoother than the regular adapted Wegstein technique, the acceleration was applied on every iteration beginning with iteration six for the first run with this revision. A second run was then made with the technique applied every iteration beginning with iteration two-- the earli-

iest possible iteration that values of "a" can be computed.

The results of this revision are shown in table 7. The complete failure of this revision was apparently due to a combination of the small size of the q's used and the number of rejected slopes that resulted in unaccelerated nodes.

Computer Running Times

During the initial attempts to obtain an analytical solution using the IBM 1620 computer, a wholly unexpected difficulty arose. The time required to obtain an analytical solution using the IBM 1620 was impossibly long. For instance, it took 8 minutes and 20 seconds per grid point to do the triple summation using equal maximum summation indices of only 5. A graphical estimate of the time to do a summation with maximum indices at 20 showed an approximate time of two hours per grid point. Further, another plot of solutions versus maximum summation indices indicated that the summation indices should be at least 20 to obtain the three-place accuracy desired.

This situation forced the use of the faster IBM 7090 system. The same problem solved on the 7090 took only 13.5 seconds per grid point for maximum indices of 20, and 36 seconds per point for indices of 30.

This incident prompted interest in other comparisons. For instance, it took the 7090 only 49 seconds to solve a two-dimensional problem of 400 nodes that took 218 iterations

Table 7

Tabulation of the Effects of a Revision to the Adapted Wegstein Procedure

Method	Iterations	
	Number	Savings[a] (%)
Regular adapted Wegstein	58	36
Revised:[b] Apply technique on every iteration beginning with iteration 6	72	21
Revised:[b] Apply technique on every iteration beginning with iteration 2	70	27

[a] Savings over the method of successive displacements

[b] Using equation 40a in lieu of equation 40 to compute q

whereas, the IBM 1620 used roughly one hour and 45 minutes to solve a three-dimensional problem of 125 nodes that took only 91 iterations.

V. Conclusions and Recommendations

Conclusions

It is apparent that the adapted Wegstein technique and successive overrelaxation have the same theoretical basis. The prime differences are in the method of application of the accelerating factor and the used/permitted size of the accelerating factor.

The results obtained in this study lead one to consider the adapted Wegstein technique as a gross overrelaxation procedure. This technique apparently obtains its acceleration power from a series of gross overrelaxations of a few nodes and the ability of the method of successive displacements to distribute this large acceleration to other adjacent nodes while pulling the value of the overrelaxed node back in line. The problem with this procedure is the lack of control of the spurts of over acceleration in the late stages of the iterative process. How one might control this to better advantage is an unanswered question. The results obtained when an attempt was made to smooth the acceleration tentatively indicate that no simple revision of the adapted Wegstein technique is going to provide a significant increase in the ability of the procedure to reduce iterations. There is no intent to suggest that the method will never approach the method of overrelaxation in total savings of iterations.

The results obtained cannot support such a conclusion. The best chance for improvement seems to be in the use of the individuality of the convergence of the slope "a" of the nodes. To use this characteristic effectively, however, one must find a way to isolate the disturbing effect on the slopes of having the adjacent nodes accelerated. In this way the slope test will not reduce the number of nodes that receive acceleration.

A possible procedure for isolating the slopes would be to generate a matrix of q's on the nth iteration without applying the technique. Then the q's generated on the nth iteration would be applied on the (n+1)st iteration. This would be followed by a period of iterations by successive displacements to smooth the slopes. Then the same procedure of delayed application could be used again, and so on. Using the normal method of computing q (equation 40) would permit values that tend to overaccelerate, so that this aspect of the adapted Wegstein technique would be retained. A disadvantage of this method would be the requirement to store another matrix equal in size to the number of unknowns.

The results using the optimum factor are obviously the maximum savings that can be achieved by the process of estimating ω for successive overrelaxation. It is also likely that this optimum overrelaxation represents the maximum attainable savings for the adapted Wegstein technique.

Yet, a chance series of events could cause a gross over-acceleration early in the process. When after a few iterations of successive displacements, the temperatures could be brought back to a value near their final value before the smoother successive overrelaxation process. One cannot, of course, build a procedure on the basis of chance, but the results obtained permit the speculation that it might be possible to increase the probability of occurrence of this chance.

The adapted Wegstein technique suffers from two other significant disadvantages when compared with successive overrelaxation. The technique requires a more complex program and the storage of one additional matrix of values equal in size to the number of unknowns. With computers the size of the 7090 with a capacity of 40,000 words available, the storage space required may not be a critical requirement. This does, however, restrict the use of the method to relatively large computer facilities. In view of these mechanical disadvantages of programming and storage, the adapted Wegstein technique would have to be able to produce greater savings than the procedure of making a running estimate of the relaxation factor before the technique is widely accepted.

For successive overrelaxation, the results indicate that the practical procedure of making a running estimate of relaxation factor does result in significant savings of

iterations. These results also indicate that the best procedure to use is the first power norm when estimating the factor. This is fortunate since the first power norm requires less computation than the norm because the residuals are not squared prior to summing. More conclusive proof is required, however. The simplicity of the sample problem could have otimized the convergence of the first power norm. Despite the results, the more general case could be that the use of the first power norm degrades the procedure compared to the norm since the two were relatively close in the results obtained. Supporting this assertion are the facts that in the two cases of comparison tabulated, the norm did reach a final value sooner and was consistently lower than optimum while the first power norm produced a factor that for one run was greater than optimum and for the other less.

Despite the simplicity of the test to see if the adapted Wegstein technique can force convergence, one must conclude that it cannot. This follows logically from the fact that the adapted Wegstein technique has the same theoretical basis as successive overrelaxation and successive overrelaxation has been shown to be theoretically incapable of forcing convergence. Furthermore, the technique uses and depends heavily upon the method of successive displacements which is a significant difference from the Wegstein technique applied to a single equation. Thus, the adapted Wegstein technique cannot

force convergence if the method of successive displacements diverges.

Since there was no real break away of one method in the results obtained for the comparison of serial and spiral scans, the results seem inconclusive. On the other hand, the results can support a conclusion that no change in scan procedure will produce a significant acceleration of the solution by successive displacements.

Despite their limitations, Power's results suggest the second possible conclusion. Her results show only a small increment of change in the number of iterations by successive displacements for the various scan procedures used. This is consistent with the results obtained in this investigation. The absence of a source of acceleration from a relaxation factor or equivalent to provide the large temperature increments per iteration in the initial iterations supports the second conclusion. Furthermore, since the full spiral scan is nonconsistent, it is unlikely that using successive over-relaxation or the adapted Wegstein technique with the full spiral for further acceleration will prove to be as good as acceleration with a serial scan. Thus, the facts available from this study seem to indicate that no significant advantage can be achieved by use of other than a consistent serial scan.

Recommendations

This investigation provides a theoretical basis for the adapted Wegstein technique which was not previously defined, but fails to use this knowledge in any broad experimentation designed to improve the technique. There is, therefore, a justification for further study and comparison to improve the technique and draw a firm conclusion as to the better method of accelerating the method of successive displacements.

It is recommended that any further expansion of this work be done with sample heat transfer problems of the type solved in a practical situation in order that performance may be better related to real situations.

Bibliography

1. Carslaw, H. S., and J. C. Jaeger. Conduction of Heat in Solids (Second Edition). London: Oxford University Press, 1959.

2. Crandall, S. H. Engineering Analysis. New York: McGraw Hill Book Co., Inc., 1956.

3. Faddeeva, V. N. Computational Methods of Linear Algebra. New York: Dover Publications Inc., 1959.

4. Forsythe, G. E. and W. R. Wasow. Finite Difference Methods for Partial Differential Equations. New York: John Wiley and Sons Inc., 1960.

5. Kaplan S. and N. Clark. "Accuracy and Convergence Techniques for Implicit Numerical Solution of the Diffusion Equation for Transient Heat Transfer". Technical Paper. Cincinnati: General Electric Flight Propulsion Division, 1961.

6. -----. "Accuracy and Convergence Techniques for Implicit Numerical Solution of the Diffusion Equation for Transient Heat Transfer". Transactions of the American Nuclear Society, 4:80,81 (June 1961).

7. Powers, M.R. "Solution of Dirichlet Difference Equation by Successive Overrelaxation Process". Master's Paper. Los Angeles: University of California, August 1955.

8. Wegstein, J. H. "Accelerating Convergence of Iterative Processes." Communications of the Association for Computing Machinery, 1:9-13 (June 1958).

9. Young, D. "The Numerical Solution of Elliptic and Parabolic Partial Differential Equations." Modern Mathematics for the Engineer (Second Series) edited by E. F. Beckenbach. New York: McGraw Hill Book Co., Inc., 1961.

10. -----. "Iterative Methods for Solving Partial Difference Equations of the Elliptic Type." Transactions of the American Mathematical Society, 76:92-111 (1954).

Appendix A

Constructing the Finite Difference Equation

Two particular types of partial differential equations are to be considered in this appendix, the elliptic equation and the parabolic equation.

A typical elliptic partial differential equation is the Laplacian,

$$\nabla^2 u = 0 \tag{64}$$

where

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)$$

$$u = f(x, y, z)$$

The boundary conditions specify u on the boundary of the region considered.

The parabolic partial differential equation considered has as its form

$$\nabla^2 u = \frac{1}{\alpha} \frac{\partial u}{\partial t} \tag{65}$$

with the following boundary conditions

$$u(x, y, z, t_0) = f(x, y, z) \qquad \text{in the region}$$
$$u(x, y, z, t) = f(x, y, z, t) \qquad \text{on the boundary}$$

The approximations to the partial derivatives are formed from truncated Taylor series expansions in the following manner. Expanding about point (x_0, y_0, z_0, t_0) one may obtain

$$u(x_o+\Delta x, y_o, z_o, t_o) = u(x_o, y_o, z_o, t_o) + \frac{\Delta x}{1!}\frac{\partial u}{\partial x}\Big]_{x_o,y_o,z_o,t_o} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 u}{\partial x^2}\Big]_{x_o,y_o,z_o,t_o} + \cdots \quad (66)$$

or

$$u(x_o-\Delta x, y_o, z_o, t_o) = u(x_o, y_o, z_o, t_o) - \frac{\Delta x}{1!}\frac{\partial u}{\partial x}\Big]_{x_o,y_o,z_o,t_o} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 u}{\partial x^2}\Big]_{x_o,y_o,z_o,t_o} + \cdots \quad (67)$$

Truncating equations 66 and 67 at the second partial derivative term and adding the two expressions, one obtains the approximation for the second partial derivative term of

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{U(x_o+\Delta x, y_o, z_o, t_o) - 2U(x_o, y_o, z_o, t_o) + U(x_o-\Delta x, y_o, z_o, t_o)}{(\Delta x)^2} \quad (68)$$

The expression for the first partial derivative is obtained by subtracting the two equations to obtain

$$\frac{\partial u}{\partial t} \approx \frac{U(x_o, y_o, z_o, t_o+\Delta t) - U(x_o, y_o, z_o, t_o)}{\Delta t} \quad (69)$$

Note that as the intervals Δx, Δy, Δz or Δt become smaller, fewer and fewer terms are required to obtain a good approximation since the interval is raised to successively higher powers with each new term making such terms as

$$\frac{(\Delta x)^3}{3!}\frac{\partial^3 u}{\partial x^3}, \quad \frac{(\Delta x)^4}{4!}\frac{\partial^4 u}{\partial x^4}, \quad \frac{(\Delta x)^5}{5!}\frac{\partial^5 u}{\partial x^5}, \text{ etc}$$

less significant. Thus, if one reduces the time and/or space interval, a more accurate solution should result.

This fact does constitute a simple test of the validity of one's solution. For the test, one halves the grid size used for the solution at hand and determines a new solution with this smaller grid size. If the two solutions give answers that are in reasonable agreement with each other, then one has some assurance that the original solution is correct.

Starting first with the elliptic equation (number 64), when one substitutes the expressions given in equation 68 for the partial derivatives, the finite difference approximation

$$\left[\frac{2}{(\Delta x)^2}+\frac{2}{(\Delta y)^2}+\frac{2}{(\Delta z)^2}\right]U(x_0,y_0,z_0) = \frac{U(x_0+\Delta x,y_0,z_0)+U(x_0-\Delta x,y_0,z_0)}{(\Delta x)^2} + \frac{U(x_0,y_0+\Delta y,z_0)+U(x_0,y_0-\Delta y,z_0)}{(\Delta y)^2} + \frac{U(x_0,y_0,z_0+\Delta z)+U(x_0,y_0,z_0-\Delta z)}{(\Delta z)^2} \tag{70}$$

is obtained. One may generalize this equation using the following notation

$$\begin{array}{lll} \Delta x = h_x & x_l = x_0 + l\Delta x = i & l = 1,2,3,\ldots l \\ \Delta y = h_y & y_j = y_0 + j\Delta y = j & j = 1,2,3,\ldots m \\ \Delta z = h_z & z_k = z_0 + k\Delta z = k & k = 1,2,3,\ldots n \end{array} \tag{71}$$

Then equation 70 may be written in the general form

$$\left[\frac{2}{h_x^2}+\frac{2}{h_y^2}+\frac{2}{h_z^2}\right]U(l,j,k) = \frac{U(l+1,j,k)+U(l-1,j,k)}{h_x^2} + \frac{U(l,j+1,k)+U(l,j-1,k)}{h_y^2} + \frac{U(l,j,k+1)+U(l,j,k-1)}{h_z^2} \tag{72}$$

If $h_x = h_y = h_z = h$ the form simplifies to

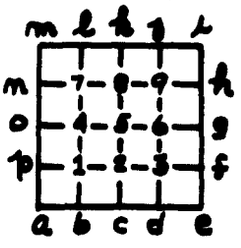
$$U(i,j,k) = \frac{1}{6}\left[U(i+1,j,k) + U(i-1,j,k) + U(i,j+1,k) + U(i,j-1,k) + U(i,j,k+1) + U(i,j,k-1)\right] \quad (73)$$

For two dimensions, the form becomes

$$U(i,j,k) = \frac{1}{4}\left[U(i+1,j) + U(i-1,j) + U(i,j+1) + U(i,j-1)\right] \quad (74)$$

Thus, U(i,j,k) is expressed in terms of values of U at nodes one point away in the direction of the coordinate axes. Since the given boundary values only specify U's on the boundary, each U(i,j,k) is expressed in terms of other unknown values at the interior node points or a combination of unknown interior points and known boundary points. The result is a system of simultaneous linear equations equal in number to the number of interior nodal points.

As an example of how the system of equations is constructed, consider the two dimensional grid with mesh points numbered as shown and boundary values lettered.



With boundary values prescribed, Ua thru Up are known and may be considered constant. The unknowns can be described by the following nine finite difference equations (for the

nine interior nodes).

$$
\begin{array}{rrrrrrrrrr}
-4U_1 & +U_2 & & +U_4 & +U_p & +U_b & & & & =0 \\
U_1 & -4U_2 & +U_3 & & +U_5 & +U_c & & & & =0 \\
 & U_2 & -4U_3 & & & +U_6 & +U_d & +U_f & & =0 \\
U_1 & & & -4U_4 & +U_5 & & +U_7 & +U_e & & =0 \\
 & U_2 & & +U_4 & -4U_5 & +U_6 & & +U_8 & & =0 \\
U_9 & & +U_3 & & +U_5 & -4U_6 & & & +U_9 & =0 \\
U_2 & +U_m & & +U_4 & & & -4U_7 & +U_8 & & =0 \\
U_8 & & & & +U_5 & & +U_7 & -4U_8 & +U_9 & =0 \\
U_3 & +U_n & & & & +U_6 & & +U_8 & -4U_9 & =0
\end{array}
\tag{75}
$$

Rearranging and placing the constant boundary values on the right, the system can be written in the matrix form

$$
\begin{bmatrix}
-4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4
\end{bmatrix}
\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \\ U_8 \\ U_9 \end{bmatrix}
=
\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \end{bmatrix}
\tag{76}
$$

or in matrix notation

$$
\bar{\bar{A}}\bar{U} = \bar{C} \tag{77}
$$

It is interesting to note that the matrix $\bar{\bar{A}}$ is equal to its transpose (or $\bar{\bar{A}} = \bar{\bar{A}}^T$) which indicates that $\bar{\bar{A}}$ is a

symmetric matrix.

As in the case of the elliptic equation, equation 68 is used in the parabolic equation to approximate the second partial derivative. Then equation 69 is used to approximate the partial with respect to time. This time, however, there is a choice of time level to use in the approximation of the second partial derivative. Which level of time, t or $t + \Delta t$, is set into the approximation of $\nabla^2 u$ is optional, but the two levels lead to two different forms of the parabolic finite difference representation.

To obtain the explicit form, one choses the level t to get

$$\begin{aligned} U(i,j,k,t+\Delta t) = {} & \frac{a\Delta t}{h_x^2}\left[U(i+1,j,k,t) + U(i-1,j,k,t)\right] \\ & + \frac{a\Delta t}{h_y^2}\left[U(i,j+1,k,t) + U(i,j+1,k,t)\right] \\ & + \frac{a\Delta t}{h_z^2}\left[U(i,j,k+1,t) + U(i,j,k-1,t)\right] \\ & + \left[1 - 2a\Delta t\left(\frac{1}{h_x^2} + \frac{1}{h_y^2} + \frac{1}{h_z^2}\right)\right]U(i,j,k,t) \end{aligned} \qquad (78)$$

This equation is known as the explicit form since U(x,y,z,t+ t) is explicitly expressed in terms of known values at the level t.

Since u at any x, y, z and some starting time t_i are given, the solution can be started at $t = t_i$, where the U's on the right side of the above equation are knowns, and solve for the unknown U(i,j,k,t+Δt) for each mesh point in the region.

Once U is known for all mesh points at the level $t_i + \Delta t$ the process is repeated to obtains U's at the level of $t_i + 2\Delta t$, and so on to the time desired.

Though this explicit form does not lead to the requirement to solve a system of equations, it does have the disadvantage of being restricted by stability considerations. Unless the following criteria are met, an instability exists that causes a growth of errors from time step to time step which ultimately reaches the point where error dominates the solution (Ref 4:92). For this particular finite difference representation, stability will exist if

$$R < \frac{1}{6} \tag{80}$$

for the three-dimensional problem, or

$$R < \frac{1}{4} \tag{81}$$

for the two-dimensional problem, where for $h_x = h_y = h_z = h$

$$R = \frac{a\Delta t}{h^2} \tag{82}$$

One can see that any change in space (or time) increment must be accompanied by a proportional change in time (or space) increment.

Alternately, $\nabla^2 u$ can be set at the level $t + \Delta t$ to get the implicit form (i.e. "a group of components of U at level $t + \Delta t$ are defined simultaneously in such an interrelated manner that it is necessary to solve a linear subsystem for the whole subset of components at once before a single one

can be determined" (Ref :267)

$$\left[1 + 2a\Delta t\left(\frac{1}{h_x^2} + \frac{1}{h_y^2} + \frac{1}{h_z^2}\right)\right] U(\lambda, j, k, t+\Delta t)$$
$$- \frac{a\Delta t}{h_x^2}\left[U(\lambda+1, j, k, t+\Delta t) + U(\lambda, j-1, k, t+\Delta t)\right]$$
$$- \frac{a\Delta t}{h_y^2}\left[U(\lambda, j+1, k, t+\Delta t) + U(\lambda, j-1, k, t+\Delta t)\right] \qquad (83)$$
$$- \frac{a\Delta t}{h_z^2}\left[U(\lambda, j, k+1, t+\Delta t) + U(\lambda, j, k-1, t+\Delta t)\right] = U(\lambda, j, k, t)$$

For the case of equal grid size in the three coordinate directions, one may write

$$[1+6r]\, U(\lambda, j, k, t+\Delta t)$$
$$- r\left[U(\lambda+1, j, k, t+\Delta t) + U(\lambda-1, j, k, t+\Delta t)\right.$$
$$+ U(\lambda, j-1, k, t+\Delta t) + U(\lambda, j+1, k, t+\Delta t)$$
$$\left. + U(\lambda, j, k+1, t+\Delta t) + U(\lambda, j, k-1, t+\Delta t)\right] = U(\lambda, j, k, t) \qquad (84)$$

where

$$r = \frac{a\Delta t}{h^2}$$

For two dimensions and equal grid sizes, one may write

$$[1+4r]\, U(\lambda, j, t+\Delta t)$$
$$- r\left[U(\lambda+1, j, t+\Delta t) + U(\lambda-1, j, t+\Delta t)\right.$$
$$\left. + U(\lambda, j+1, t+\Delta t) + U(\lambda, j-1, t+\Delta t)\right] = U(\lambda, j, t) \qquad (85)$$

Since U at any x, y, z, and some starting time t_i is specified by given initial conditions, the solution can be

commenced at $t_i + \Delta t$ and U(i,j,k,t) would be known and the remaining U's form a system of simultaneous equations at the level $t_i + \Delta t$. This need to solve a system of equations resulted from the finite difference representation of the elliptic equation, and a comparison of the matrix forms of the elliptic and parabolic equations is of interest.

Taking the same sample grid system that was used for the elliptic equation, the system of equations for the implicit form are

$$
\begin{aligned}
(1+4\lambda)U_1 - \lambda U_2 - \lambda U_4 &= -(U_P+U_b) + U_{1,t} \\
-\lambda U_1 + (1+4\lambda)U_2 - \lambda U_3 - \lambda U_5 &= -U_c + U_{2,t} \\
-\lambda U_2 + (1+4\lambda)U_3 - \lambda U_6 &= -(U_d+U_F) + U_{3,t} \\
-\lambda U_1 + (1+4\lambda)U_4 - \lambda U_5 - \lambda U_7 &= -U_e + U_{4,t} \\
-\lambda U_2 - \lambda U_4 + (1+4\lambda)U_5 - \lambda U_6 - \lambda U_8 &= U_{5,t} \\
-\lambda U_3 - \lambda U_5 + (1+4\lambda)U_6 - \lambda U_9 &= U_g - U_{6,t} \\
-\lambda U_4 + (1+4\lambda)U_7 - \lambda U_8 &= -(U_l + U_m) + U_{7,t} \\
-\lambda U_5 - \lambda U_7 + (1+4\lambda)U_8 - \lambda U_9 &= -U_h + U_{8,t} \\
-\lambda U_6 - \lambda U_8 + (1+4\lambda)U_9 &= -(U_j + U_k) + U_{9,t}
\end{aligned}
\qquad (86)
$$

where $U_{m,t} = U(i,j,k,t)$--known from initial conditions or previous computations

$U_m = U(i,j,k,t+\Delta t)$

In matrix form the system becomes

In matrix form the system becomes

$$\begin{bmatrix} (1+4r) & -r & 0 & -r & 0 & 0 & 0 & 0 & 0 \\ -r & (1+4r) & -r & 0 & -r & 0 & 0 & 0 & 0 \\ 0 & -r & (1+4r) & 0 & 0 & -r & 0 & 0 & 0 \\ -r & 0 & 0 & (1+4r) & -r & 0 & -r & 0 & 0 \\ 0 & -r & 0 & -r & (1+4r) & -r & 0 & -r & 0 \\ 0 & 0 & -r & 0 & -r & (1+4r) & 0 & 0 & -r \\ 0 & 0 & 0 & -r & 0 & 0 & (1+4r) & -r & 0 \\ 0 & 0 & 0 & 0 & -r & 0 & -r & (1+4r) & -r \\ 0 & 0 & 0 & 0 & 0 & -r & 0 & -r & (1+4r) \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \\ U_8 \\ U_9 \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \end{bmatrix} \tag{87}$$

which in matrix notation is

$$\bar{\bar{A}}'\bar{U} = \bar{C}' \tag{88}$$

These matrices have the same general form as those for the elliptic equation and, thus, the implicit form may be thought of as equivalent to solving an elliptic equation at each time step.

Appendix B

Sample Matrix Forms

The system under consideration is of the form AU = C where sample expanded forms are given in equations 76 and 86 in Appendix A. The example parabolic equation used in Appendix A will be used as an example in this Appendix to develop the forms of the matrices that comprise the expression for determining the pertinent eigenvalues of the solution procedures considered in this study.

To begin this development, divide each equation of the system given in equation 86 by the common diagonal element of the A matrix to obtain (Ref 3:128)

$$\begin{aligned} U_1 - \frac{r}{(1+4r)}U_2 \qquad\qquad - \frac{r}{(1+4r)}U_4 \qquad\qquad &= C_1 \\ -\frac{r}{(1+4r)}U_1 + \quad U_2 \quad - \frac{r}{(1+4r)}U_3 \qquad - \frac{r}{(1+4r)}U_5 &= C_2 \\ \text{etc} \end{aligned} \tag{89}$$

After moving the off-diagonal terms to the right, the system can be written in matrix notation as

$$\bar{U} = \bar{\bar{S}}\bar{U} + \bar{C}' \tag{90}$$

where

$$\bar{C}' = \begin{bmatrix} \frac{C_1}{1+4r} \\ \frac{C_2}{1+4r} \\ \vdots \\ \vdots \\ \frac{C_m}{1+4r} \end{bmatrix}$$

$$\bar{\bar{S}} = \begin{bmatrix} 0 & \frac{r}{1+4r} & 0 & \frac{r}{1+4r} & \cdots & 0 & 0 \\ \frac{r}{1+4r} & 0 & \frac{r}{1+4r} & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \frac{r}{1+4r} & \cdots & \frac{r}{1+4r} & 0 \end{bmatrix} \tag{91}$$

The matrix S may be subdivided into two triangular constituent matrices (Ref 7:134)

$$\bar{\bar{S}} = \bar{\bar{B}} + \bar{\bar{C}} \tag{92}$$

where

$$\bar{\bar{B}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{r}{1+4r} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{r}{1+4r} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{r}{1+4r} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & \frac{r}{1+4r} & 0 & \frac{r}{1+4r} & 0 \end{bmatrix} \tag{93}$$

$$\bar{\bar{C}} = \begin{bmatrix} 0 & \frac{r}{1+4r} & 0 & \frac{r}{1+4r} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{r}{1+4r} & 0 & \frac{r}{1+4r} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{r}{1+4r} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{r}{1+4r} & 0 & \frac{r}{1+4r} & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{94}$$

Now, note that the B matrix contains the elements associated with the (n + 1)st iteration and the C matrix with the nth iteration of the method of successive displacements for the consistent serial scan shown in Appendix A.

One may then write

$$\bar{U}^{(m+1)} = \bar{\bar{B}}\bar{U}^{(m+1)} + \bar{\bar{C}}\bar{U}^{(m)} + \bar{C}' \tag{95}$$

which may be rearranged to

$$\bar{U}^{(m+1)} - \bar{\bar{B}}\bar{U}^{(m+1)} = \bar{\bar{C}}\bar{U}^{(m)} + C' \tag{96}$$

$$(\bar{\bar{I}} - \bar{\bar{B}})\bar{U}^{(m+1)} = \bar{\bar{C}}\bar{U}^{(m)} + \bar{C}' \tag{97}$$

$$\bar{U}^{(m+1)} = (\bar{\bar{I}} - \bar{\bar{B}})^{-1}\bar{\bar{C}}\bar{U}^{(m)} + (\bar{\bar{I}} - \bar{\bar{B}})^{-1}\bar{C}' \tag{98}$$

which is equivalent to the form

$$\bar{X} = \bar{\bar{A}}_1\bar{X} + \bar{\bar{F}} \tag{61}$$

From the section on theory one can find two descriptions of the partitioning of the A matrix to find the eigenvalues of the method of successive displacements. Equation 16 partitions the A matrix into

$$\bar{\bar{A}} = (\bar{\bar{E}} + \bar{\bar{D}} + \bar{\bar{F}}) \tag{16}$$

to arrive at the determinantal equation for the eigenvalues of

$$\det(\eta\bar{\bar{E}} - \eta\bar{\bar{D}} + \bar{\bar{F}}) = 0 \tag{18}$$

In this case the forms for the submatrices are

$$\bar{\bar{D}} = \begin{bmatrix} (1+4k) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (1+4k) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (1+4k) & 0 & 0 & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (1+4k) \end{bmatrix} \tag{99}$$

$$\bar{\bar{E}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\mu & 0 & -\mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\mu & 0 & -\mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\mu & 0 & -\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mu & 0 & -\mu & 0 \end{bmatrix} \tag{100}$$

$$\bar{\bar{F}} = \begin{bmatrix} 0 & -\mu & 0 & -\mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\mu & 0 & -\mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\mu & 0 & -\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mu & 0 & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{101}$$

Alternately, equation 59 gives the form of partitioning of the A matrix as

$$\bar{\bar{A}} = (\bar{\bar{I}} - \bar{\bar{B}})^{-1} \bar{\bar{C}} \tag{59}$$

For this form, Mrs Fadeeva advises that the eigenvalues may be found from the determinantal equation (Ref :135)

$$\det\left[\bar{\bar{C}} - (\bar{\bar{I}} - \bar{\bar{B}})\eta\right] = 0 \tag{102}$$

The forms of the B and C matrix are given in equations 93 and 94. The I matrix is an identity matrix of the usual form of

$$\overline{\overline{I}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{103}$$

Note that the relationships between the D, E, and F and I, C, and B matrices may be expressed as

$$\overline{\overline{I}} = \frac{-1}{1+4\pi}\overline{\overline{D}} \tag{104}$$

$$\overline{\overline{B}} = \frac{1}{1+4\pi}\overline{\overline{E}} \tag{105}$$

$$\overline{C} = \frac{-1}{1+4\pi}\overline{\overline{F}} \tag{106}$$

The equivalence of equations 18 and 102 can be demonstrated in the following manner. First, rearrange 18 to

$$\det\left[\overline{\overline{F}} + (\overline{\overline{D}} + \overline{\overline{E}})\eta\right] = 0 \tag{107}$$

Now if one multiplies this expression by the reciprocal of the diagonal element of the D matrix, solutions of the determinantal equation are uneffected. This action results in

$$\det\left\{\left(\frac{1}{1+4\pi}\right)\overline{\overline{F}} + \left[\left(\frac{1}{1+4\pi}\right)\overline{\overline{D}} + \left(\frac{1}{1+4\pi}\right)\overline{\overline{E}}\right]\right\} = 0 \tag{108}$$

which by the use of equations 104 to 106 can be rewritten in the exact form of equation 102.

Appendix C

Definitions

Diagonal Dominance:

For a matrix to possess the characteristic of diagonal dominance, the following relationship must exist for the elements of the matrix:

$$|a_{ii}| > \sum_{\substack{j=1 \\ i \neq j}}^{N} |a_{ij}| \qquad i = 1, 2, 3 \ldots M \tag{109}$$

Property (A): (Ref 4:243)

A square matrix A of order N is said to have property (A) if there exists a permutation matrix such that is diagonally block tridiagonal.

Spectral Radius: (Ref 10:94)

The spectral radius is defined as the maximum of the moduli of the eigenvalues of a given matrix.

Dirichlet Difference Equation:

The finite difference equation that represents the partial differential equation $\nabla^2 u = 0$ in the region with $u = f$ on the boundary.

Residual:

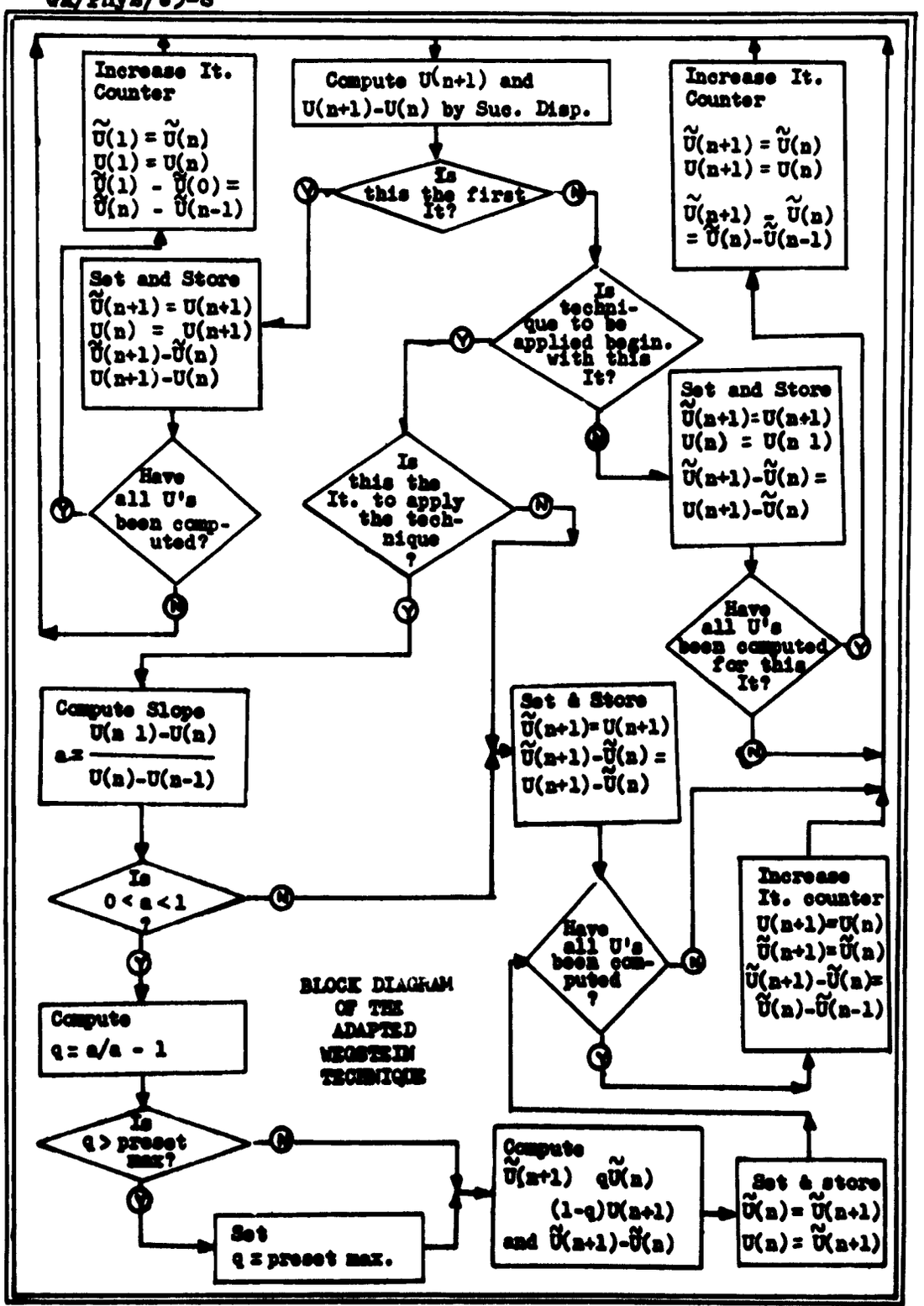
The difference between two successive improved values at a given node, i.e.

$$R_{i,j,k} = U^{(m+1)}_{(i,j,k)} - U^{(m)}_{(i,j,k)}$$

```
C      SOLUTION OF THREE SIMULTANEOUS EQUATIONS BY THE GAUSS
C      SIEDEL METHOD WITH OPTION FOR WEGSTEIN ACCELERATION
C
C      SPECIFY DIMENSIONED VARIABLES
C
       DIMENSION A(3,3),B(3),XNEW(3),X(3),DIFF(3)
       DIMENSION WX(3),WXNEW(3),WDIFF(3)
C
C      READ INPUT DATA
       READ,A(1,1),A(1,2),A(1,3),A(2,1),A(2,2),A(2,3)
       READ,A(3,1),A(3,2),A(3,3)
       READ,B(1),B(2),B(3)
       READ,X(1),X(2),X(3),ERR
       READ,IAPLY,APPLY;WMAX,WSET
C
C      SET INITIAL VALUES AND ESTIMATES
       DO 60 I=1,3
       DIFF(I)=0.0
 60    WDIFF(I)=0.0
C
C      COUNT IS AN EXIT CONTROL THAT COUNTS THE NUMBER
C      OF PASSES THRU THE COMPUTATIONAL SEQUENCE
       COUNT=0.0
C
C      SET ITERATION COUNTER
       ITNUM=0
 35    ITNUM=ITNUM+1
       COUNT=COUNT+1.0
       PRINT 104
 104   FORMAT (/)
       PRINT 6,ITNUM
 6     FORMAT(9HITERATION,I4/)
C
C      WHEN A GIVEN RESIDUAL FAILS TO PASS THE CONVERGENCE
C      TEST, ICHEK IS SET = 1 AND NO MORE RESIDUALS ARE CHECKED
C      UNTIL THE NEXT ITERATION
       ICHEK=0
C
C      COMPUTATIONAL SEQUENCE
       N=0
       PRINT 30
 30    FORMAT(13H VALUES OF X ,3X,14HX(N+1) - X(N) /)
 46    N=N+1
       IF(N-1)1,1,40
 40    IF(N-2)2,2,41
 41    IF(N-3)3,3,42
 1     XNEW(1)=1./A(1,1)*(B(1)-A(1,2)*X(2)-A(1,3)*X(3))
       DIFF(1)=XNEW(1)-X(1)
       GO TO 43
 2     XNEW(2)=1./A(2,2)*(B(2)-A(2,1)*X(1)-A(2,3)*X(3))
```

```
      DIFF(2)=XNEW(2)-X(2)
      GO TO 43
  3   XNEW(3)=1./A(3,3)*(B(3)-A(3,1)*X(1)-A(3,2)*X(2))
      DIFF(3)=XNEW(3)-X(3)
C
C     PRINT OUT RESULTS EVERY ITERATION
 43   PRINT 62,XNEW(N),DIFF(N)
 62   FORMAT (2E14.8/)
      X(N)=XNEW(N)
C
C     BYPASS THE ACCELERATING TECHNIQUE BY SETTING SWITCH ONE OFF
      IF(SENSE SWITCH 1)71,46
C
C     SET UP WEGSTEIN VALUES FOR THE FIRST ITERATION
 71   IF(ITNUM-1)44,44,45
 44   WDIFF(N)=DIFF(N)
      WX(N)=XNEW(N)
      X(N)=XNEW(N)
      GO TO 46
C
C     INPUT IAPPLY DETERMINES THE ITERATION FOR THE FIRST
C     APPLICATION OF THE TECHNIQUE
 45   IF(ITNUM-IAPLY)47,53,53
C
C     THE INPUT APPLY DETERMINES HOW OFTEN THE TECHNIQUE IS APPLIED
 53   IF(COUNT-APPLY)47,48,48
 47   WDIFF(N )=XNEW(N)-WX(N)
      WX(N)=XNEW(N)
      X(N)=XNEW(N)
      GO TO 46
 48   IF(N-3) 63,54,54
 54   COUNT=0.
C
C     THE ADAPTED WEGSTEIN TECHNIQUE
 63   WTEST=DIFF(N)/WDIFF(N)
      WFACT=WTEST/(WTEST-1.)
      PRINT 107,WTEST,WFACT
 107  FORMAT (/8HSLOPE = ,E14.8,2X,9HFACTOR = ,E14.8)
C
C     BYPASS THE SLOPE TEST BY SETTING SWITCH 2 OFF
      IF(SENSE SWITCH 2) 105,50
 105  IF(WTEST)47,47,49
 49   IF(WTEST-1.)50,47,47
C
C     TEST ACCELERATED SOLUTIONS AGAINST ESTABLISHED
C     CONVERGENCE CRITERIA WHEN TECHNIQUE IS USED
 50   IF(ABS(WFACT)-WMAX)52,52,51
 51   WFACT=-WSET
```

```
 52   WXNEW(N)=WFACT*WX(N)+(1.-WFACT)*XNEW(N)
      PRINT 61,WXNEW(N)
 61   FORMAT(/11HWXNEW(N) = ,E14.8/)
      WDIFF(N)=WXNEW(N)-WX(N)
C
C     WHEN THE WEGSTEIN TECHNIQUE IS BYPASSED
C     TEST GAUSS SIEDEL SOLUTIONS FOR CONVERGENCE
      IF(ABS(WDIFF(N))-ERR) 80,80,81
 80   ICHEK=1
 81   WX(N)=WXNEW(N)
      X(N)=WXNEW(N)
      GO TO 46
C
C     TEST TO SEE IF A NEW ITERATION IS REQUIRED
 42   IF(ICHEK) 72,72,35
 72   DO 33 J=1,3
      IF(ABS(DIFF(J))-ERR) 33,33,35
 33   CONTINUE
C
C     PRINT OUT INPUT DATA FOR REFERENCE PURPOSES
      DO 100 J=1,3
      DO 100 I=1,3
100   PRINT 101,A(I,J),I,J
101   FORMAT(/9HA(I,J) =,E14.8,6H  I = ,I3,6H  J = ,I3)
      DO 102 I=1,3
102   PRINT 103,B(I)
103   FORMAT(/9H  B(I) = ,E14.8)
C
 13   STOP
      END
```


Increase It. Counter
$\tilde{U}(1) = \tilde{U}(n)$
$U(1) = U(n)$
$\tilde{U}(1) - \tilde{U}(0) = \tilde{U}(n) - \tilde{U}(n-1)$
Compute U(n+1) and U(n+1)-U(n) by Suc. Disp.
Is this the first It?
Increase It. Counter
$\tilde{U}(n+1) = \tilde{U}(n)$
$U(n+1) = U(n)$
$\tilde{U}(n+1) - \tilde{U}(n) = \tilde{U}(n) - \tilde{U}(n-1)$
Set and Store
$\tilde{U}(n+1) = U(n+1)$
$U(n) = U(n+1)$
$\tilde{U}(n+1) - \tilde{U}(n)$
$U(n+1) - U(n)$
Is technique to be applied begin. with this It?
Set and Store
$\tilde{U}(n+1) = U(n+1)$
$U(n) = U(n 1)$
$\tilde{U}(n+1) - \tilde{U}(n) = U(n+1) - \tilde{U}(n)$
Have all U's been computed?
Is this the It. to apply the technique ?
Have all U's been computed for this It?
Compute Slope
$a = \frac{U(n 1) - U(n)}{U(n) - U(n-1)}$
Set & Store
$\tilde{U}(n+1) = U(n+1)$
$\tilde{U}(n+1) - \tilde{U}(n) = U(n+1) - \tilde{U}(n)$
Is $0 < a < 1$?
Have all U's been computed ?
Increase It. counter
$U(n+1) = U(n)$
$\tilde{U}(n+1) = \tilde{U}(n)$
$\tilde{U}(n+1) - \tilde{U}(n) = \tilde{U}(n) - \tilde{U}(n-1)$
BLOCK DIAGRAM OF THE ADAPTED WEGSTEIN TECHNIQUE
Compute
$q = a/a - 1$
Is $q >$ preset max?
Compute
$\tilde{U}(n+1)$ $q\tilde{U}(n)$
$(1-q)U(n+1)$
and $\tilde{U}(n+1) - \tilde{U}(n)$
Set & store
$\tilde{U}(n) = \tilde{U}(n+1)$
$U(n) = \tilde{U}(n+1)$
Set
q = preset max.

GA/PHYS/63-8

```
C       HEAT TRANSFER IN CUBE-NONSTEADY, CONSTANT SOURCE BY GAUSS SIEDEL
C       USING WEGSTEIN CONVERGENCE TECHNIQUE
C
        DIMENSION U(7,7,7),UOLD(7,7,7),WDIFF(7,7,7),WU(7,7,7)
C
C       READ INPUT DATA
        READ 61, LASTX,LASTY,LASTZ
 61     FORMAT (3I3)
        READ53,DIFUS,COND,DELT,SORC
 53     FORMAT(4F10.0)
        READ 54,ERR,XGRID,YGRID,ZGRID,TMAX
 54     FORMAT(5F10.0)
        READ 56,IRITE,JRITE,KRITE,JITER,KITER
 56     FORMAT (5I4)
        READ 460,IAPPLY,APPLY,WMAX
 460    FORMAT(I4,2F10.0)
C
C       SET BOUNDARY VALUES
        DO 3 K=1,LASTZ
        DO 3 J=1,LASTY
        DO 3 I=1,LASTX
 601    UOLD(I,J,K)=0.0
        U(I,J,K)=0.0
 3      WU(I,J,K)=U(I,J,K)
C
C       COMPUTE CONSTANTS
        A=(2./(XGRID**2))+(2./(YGRID**2))+(2./(ZGRID**2))
        B=DIFUS*DELT
        C=B/(XGRID**2)
        D=B/(YGRID**2)
        E=B/(ZGRID**2)
        G=1./(1.+B*A)
        F=G*B*SORC/COND
C       SET TIME STEPS AND ITERATION COUNTER
        TIME=0.
 14     TIME=TIME+DELT
        ITNUM=0
        COUNT=0.0
C
C       CHOOSE EITHER PUNCHED OR PRINTED OUTPUT BY SETTING SWITCH ONE
C       OFF FOR PUNCHED OUTPUT OR ON FOR PRINTED OUTPUT.
        IF(SENSE SWITCH 1)81,80
 80     PUNCH 50,TIME
        GO TO 1
 81     PRINT 50,TIME
 50     FORMAT(/18HNEW TIME INCREMENT,F10.3)
 1      ITNUM=ITNUM+1
        COUNT=COUNT+1.0
        IF(SENSE SWITCH 1)83,82
 82     PUNCH 51,ITNUM
```

```
      GO TO 84
 83   PRINT 51,ITNUM
 51   FORMAT (/15X,16HITERATION NUMBER,I4/)
C
C     COMPUTATIONAL STEPS
 84   ILAST=LASTX-1
      JLAST=LASTY-1
      KLAST=LASTZ-1
      KTEST=KLAST+ILAST+JLAST
      ICHECK=0
      DO 2 K=2,KLAST
      DO 2 J=2,JLAST
      DO 2 I=2,ILAST
      KK=I+J+K
      P=G*C*(U(I+1,J,K)+U(I-1,J,K))
      Q=G*D*(U(I,J+1,K)+U(I,J-1,K))
      R=G*E*(U(I,J,K+1)+U(I,J,K-1))
      S=G*UOLD(I,J,K)
      UNEW=F+P+Q+R+S
      DIFF=UNEW-U(I,J,K)
C
C     ON THE FIRST ITERATION THE INITIAL VALUESS OF THE WEGSTEIN
C     VARIABLES ARE SET UP BUT NO WEGSTEIN COMPUTATION IS DONE
      IF(ITNUM-1)125,125,470
 125  WDIFF(I,J,K)=DIFF
      WU(I,J,K)=UNEW
      GO TO 102
C
C     INPUT IAPPLY TELLS WHAT ITERATION WILL BE THE FIRST  TO APPLY THE
C     WEGSTEIN TECHNIQUE ON
 470  IF(ITNUM-IAPPLY) 452,450,450
C
C     THE INPUT APPLY DETERMINES HOW OFTEN THE TECHNIQUE IS APPLIED
 450  IF(COUNT-APPLY) 452,109,109
C
 452  WDIFF(I,J,K)=UNEW-WU(I,J,K)
      WU(I,J,K)=UNEW
      GO TO 102
C
C     THE WEGSTEIN TECHNIQUE
 109  IF(WDIFF(I,J,K)) 709,289,709
 709  WTEST=DIFF/WDIFF(I,J,K)
      IF(WTEST) 289,289,100
 100  IF(WTEST-1.) 101,289,289
 101  WEGFAC=WTEST/(WTEST-1.)
C
C     REJECT VALUES OF THE WEGSTEIN ACCELERATING FACTOR THAT ARE ABOVE
C     A PRESET VALUE AND SET THE FACTOR EQUAL THE PRE SET VALUE
      IF(ABSF(WEGFAC)-WMAX) 453,453,454
 454  WEGFAC=-WMAX
```

```
 453  WUNEW=WEGFAC*WU(I,J,K)+(1.-WEGFAC)*UNEW
      WDIFF(I,J,K)=WUNEW-WU(I,J,K)
      WU(I,J,K)=WUNEW
      U(I,J,K)=WUNEW
C
C     TEST SOLUTIONS AGAINST ESTABLISHED CRITERIA
      IF(ICHECK)103,103,116
 103  IF(ABSF(WDIFF(I,J,K))-ERR)116,116,105
 105  ICHECK=1
 116  IF(JITER-J) 110,110,290
 110  IF(KITER-K) 112,112,290
 112  IF(SENSE SWITCH 1)114,113
 113  PUNCH 115,I,J,K,WUNEW,WDIFF(I,J,K),WTEST
      GO TO 290
 114  PRINT 115,I,J,K,WUNEW,WDIFF(I,J,K),WTEST
 115  FORMAT(5X,3I4,3E14.8)
      GO TO 290
 289  U(I,J,K)=UNEW
      WDIFF(I,J,K)=UNEW-WU(I,J,K)
      WU(I,J,K)=UNEW
 290  IF(KTEST-KK) 481,481,480
 481  COUNT=0.0
 480  IF(WTEST-1.0) 2,107,107
 102  U(I,J,K)=UNEW
      IF(ICHECK)106,106,107
 106  IF(ABSF(DIFF)-ERR)107,107,118
 118  ICHECK=1
 107  IF(JITER-J)119,119,2
 119  IF(KITER-K) 120,120,2
 120  IF(SENSE SWITCH 1) 122,121
 121  PUNCH 52,I,J,K,UNEW,DIFF
 52   FORMAT (5X,3I4,2E14.8)
      GO TO 2
 122  PRINT 52,I,J,K,UNEW,DIFF
 2    CONTINUE
C
C     SET VALUES ONE GRID POINT BEYOND PLANES OF SYMMETRY
      L=LASTX
      DO 4 N=2,LASTZ
      DO 4 M=2,LASTY
 4    U(L,M,N)=U(L-2,M,N)
      M=LASTY
      DO 5 L=2,LASTX
      DO 5 N=2,LASTZ
 5    U(L,M,N)=U(L,M-2,N)
      N=LASTZ
      DO 6 L=2,LASTX
      DO 6 M=2,LASTZ
```

```
 6     U(L,M,N)=U(L,M,N-2)
C
C      PRINT RESULTS OF ITERATION
       IF(ITNUM-1)1,1,150
 150   IF(ICHECK)10,10,1
 10    IF(SENSE SWITCH 1)88,87
 87    PUNCH 60
       GO TO 89
 88    PRINT 60
 60    FORMAT (//10X,26HFINAL RESULTS OF ITERATION//)
 89    DO 7 K=2,KLAST,KRITE
       DO 7 J=2,JLAST,JRITE
       DO 7 I=2,ILAST,IRITE
       IF(SENSE SWITCH 1)91,90
 90    PUNCH 130,I,J,K,U(I,J,K)
       GO TO 7
 91    PRINT 130,I,J,K,U(I,J,K)
 130   FORMAT (10X,3I4,E14.8)
 7     CONTINUE
C
C      CHECK TO SEE IF LAST TIME INCREMENT HAS BEEN REACHED
       IF(TMAX-TIME)17,17,12
C
C      PREPARE FOR A NEW TIME INCREMENT
 12    DO 15 I=2,LASTX
       DO 15 J=2,LASTY
       DO 15 K=2,LASTZ
 15    UOLD(I,J,K)=U(I,J,K)
       GO TO 14
C
C      PRINT OUT INPUT DATA
 17    IF(SENSE SWITCH 1 )94,93
 93    PUNCH 55,DIFUS,COND,DELT,SORC
       PUNCH 57,XGRID,YGRID,ZGRID,TMAX,ERR
       PUNCH 58,IRITE,JRITE,KRITE,KITER,JITER
       PUNCH 59,IAPPLY,APPLY,MAX
       GO TO 13
 94    PRINT 55,DIFUS,COND,DELT,SORC
  55   FORMAT(11HDIFFUSIVITY,F10.4,6X,12HCONDUCTIVITY,F10.4/
     310HDELTA TIME,F10.4,6X,15HSOURCE STRENGTH,F10.4)
       PRINT 57,XGRID,YGRID,ZGRID,TMAX,ERR
 57    FORMAT(/6HXGRID=,F10.4,6X,6HYGRID=,F10.4,6X,6HZGRID=,F10.4/
     48HMAX TIME,F10.4,18HSTOP WHEN ABSDIFF-,F10.5,15HIS ZERO OR LESS)
       PRINT 58,IRITE,JRITE,KRITE,KITER,JITER
 58    FORMAT(/25HOUTPUT CONTRCLS ARE IRITE,I3,2X,5HJRITE,I3/
     55HKRITE,I3,7H  KITER,I3,7H  JITER,I3)
       PRINT 59,IAPPLY,APPLY,MAX
 59    FORMAT (/7HIAPPLY=,I4,4X,6HAPPLY=,F10.5,4X,4HMAX=,F10.5)
 13    STOP
       END
```

GA/PHYS/63-8

```
C       HEAT TRANSFER IN CUBE-NONSTEADY, CONSTANT SOURCE BY GAUSS SIEDEL
C       WITH PROVISION FOR COMPUTING RELAXATION FACTOR AND WEGSTEIN DATA
C
C       SPECIFY DIMENSIONED VARIABLES
        DIMENSION U(7,7,7),UOLD(7,7,7),DIFF(7,7,7)
C
        READ 61, LASTX,LASTY,LASTZ
 61     FORMAT (3I3)
        READ53,DIFUS,COND,DELT,SORC
 53     FORMAT(4F10.0)
        READ 54,ERR,XGRID,YGRID,ZGRID,TMAX
 54     FORMAT(5F10.0)
        READ 56,IRITE,JRITE,KRITE,JITER,KITER,CHOSE,BOUND,CARD,COMP
 56     FORMAT(5I4,4F10.0)
        READ 124,ITSET,CONVRG
 124    FORMAT (I3,F10.0)
C
C       SET BOUNDARY VALUES
        DO 3 I=1,LASTX
        DO 3 J=1,LASTY
        DO 3 K=1,LASTZ
C
C       CHOOSE METHOD OF SETTING INITIAL VALUES BY LETTING BOUND=0.0
C       TO READ FROM CARD INPUT OR BOUND=1.0 TO SET ALL POINTS EQUAL ZERO
        IF(BOUND) 100,100,101
 100    READ 102,U(I,J,K)
 102    FORMAT (E14.8)
        UOLD(I,J,K)=U(I,J,K)
        DIFF(I,J,K)=0.
        GO TO 3
 101    UOLD(I,J,K)=0.0
        DIFF(I,J,K)=0.
        U(I,J,K)=0.0
 3      CONTINUE
        GRELAX=0.
        AVEA=0.0
        WRELAX=0.0
        POINTS=(LASTX-2)*(LASTY-2)*(LASTZ-2)
        ILAST=LASTX-1
        JLAST=LASTY-1
        KLAST=LASTZ-1
C
C       WHEN APPLICABLE CAN TAKE INITIAL VALUES FROM PUNCHED OUTPUT BY
C       USING CARD =0.0 OR-1.0 AND SETTING BOUND = 1.0
        IF(CARD) 200,200,201
 200    DO 202 K=2,KLAST
```

```
      DO 202 J=2,JLAST
      DO 202 I=2,ILAST
 202  READ 203,U(I,J,K)
 203  FORMAT (22X,E14.8)
      L=LASTX
      DO 210 N=2,LASTZ
      DO 210 M=2,LASTY
 210  U(L,M,N)=U(L-2,M,N)
      M=LASTY
      DO 211 L=2,LASTX
      DO 211 N=2,LASTZ
 211  U(L,M,N)=U(L,M-2,N)
      N=LASTZ
      DO 212 L=2,LASTX
      DO 212 M=2,LASTZ
 212  U(L,M,N)=U(L,M,N-2)
      GSETA=0.0
C
C     COMPUTE CONSTANTS
 201  A=(2./(XGRID**2))+(2./(YGRID**2))+(2./(ZGRID**2))
      B=DIFUS*DELT
      C=B/(XGRID**2)
      D=B/(YGRID**2)
      E=B/(ZGRID**2)
      G=1./(1.+B*A)
      F=G*B*SORC/COND
C
C     SET TIME STEPS AND ITERATION COUNTER
      TIME=0.
 14   TIME=TIME+DELT
      ITNUM=0
      ICOUNT=0
C
C     CHOOSE EITHER PUNCHED OR PRINTED OUTPUT BY SETTING CHOSE=0.0
C     FOR PUNCHED OUTPUT OR CHOSE=1.0 FOR PRINTED OUTPUT
      IF(CHOSE)80,80,81
 80   PUNCH 50,TIME
      GO TO 1
 81   PRINT 50,TIME
 50   FORMAT(/18HNEW TIME INCREMENT,F10.3)
 1    ITNUM=ITNUM+1
      ICOUNT=ICOUNT+1
      TOTALA=0.0
      TOTGS=0.0
      IF(CHOSE)82,82,83
 82   PUNCH 51,ITNUM
      GO TO 84
 83   PRINT 51,ITNUM
 51   FORMAT (/15X,16HITERATION NUMBER,I4/)
C
```

```
C       COMPUTATIONAL STEPS
 84     DO 2 K=2,KLAST
        DO 2 J=2,JLAST
        DO 2 I=2,ILAST
        P=G*C*(U(I+1,J,K)+U(I-1,J,K))
        Q=G*D*(U(I,J+1,K)+U(I,J-1,K))
        R=G*E*(U(I,J,K+1)+U(I,J,K-1))
        S=G*UOLD(I,J,K)
        UNEW=F+P+Q+R+S
        IF(SENSE SWITCH 1) 190,191
 190    RUNEW=GRELAX*UNEW+(1.-GRELAX)*U(I,J,K)
        UNEW=RUNEW
 191    DIFOLD=DIFF(I,J,K)
        DIFF(I,J,K)=UNEW-U(I,J,K)
        IF(SENSE SWITCH 1) 116,98
 98     IF(COMP) 240,240,241
 240    TOTGS=TOTGS+ABSF(DIFF(I,J,K))
        GO TO 242
 241    TOTGS=TOTGS+DIFF(I,J,K)**2
 242    IF(ITNUM-1) 116,116,196
 196    IF(ITSET-ICOUNT) 20,20,116
 20     WTEST=DIFF(I,J,K)/DIFOLD
        TOTALA=TOTALA+WTEST
        WEGFAC=WTEST/(WTEST-1.)
        IF(SENSE SWITCH 2) 192,116
 192    IF(CHOSE) 750,750,751
 750    PUNCH 951,I,J,K,WTEST,WEGFAC
        GO TO 116
 751    PRINT 951,I,J,K,WTEST,WEGFAC
 951    FORMAT (3I3,2E14.8)
 116    IF(JITER-J) 110,110,2
 110    IF(KITER-K)8,8,2
 8      IF(CHOSE)85,85,86
 85     PUNCH 52,I,J,K,UNEW,DIFF(I,J,K)
        GO TO 2
 86     PRINT 52,I,J,K,UNEW,DIFF(I,J,K)
 2      U(I,J,K)=UNEW
        IF(ITSET-ICOUNT) 975,975,976
 975    ICOUNT=0
C
C       SET VALUES ONE GRID POINT BEYOND PLANES OF SYMMETRY
 976    L=LASTX
        DO 4 N=2,LASTZ
        DO 4 M=2,LASTY
 4      U(L,M,N)=U(L-2,M,N)
        M=LASTY
        DO 5 L=2,LASTX
        DO 5 N=2,LASTZ
 5      U(L,M,N)=U(L,M-2,N)
        N=LASTZ
```

```
      DO 6 L=2,LASTX
      DO 6 M=2,LASTZ
 6    U(L,M,N)=U(L,M,N-2)
      GSDIF=SQRTF(TOTGS)
      IF(SENSE SWITCH 1) 198,99
 99   IF(ITNUM-1) 111,111,123
 123  GSETA=GSDIF/GDENOM
      ROLD=GRELAX
      GRELAX=2.0/(1.+SQRTF(1.-GSETA))
      AVEA=TOTALA/POINTS
      WRELAX=2.0/(1.+SQRTF(1.-AVEA))
 111  GDENOM=GSDIF
      IF(CHOSE) 243,243,244
 243  PUNCH 181,GSETA,GSDIF,GRELAX,WRELAX,AVEA
      GO TO 245
 244  PRINT 181,GSETA,GSDIF,GRELAX,WRELAX,AVEA
 181  FORMAT (8HGSETA = ,E14.8,2X,9HGSDIFF = ,E14.8,11H  GRELAX = ,E14.8
     2,/,11H  WRELAX = ,E14.8,4X,7HAVEA = ,E14.8)
 245  IF(SENSE SWITCH 3) 743,198
 743  IF(ITNUM-1) 198,198,197
 197  IF(ABSF(GRELAX-ROLD)-CONVRG) 199,199,198
 199  PRINT 180
 180  FORMAT (50HOVERRELAXATION FACTOR HAS MET CONVERGENCE CRITERIA/
     232HSW1 ON TO OVERRELAX, PRESS START)
      PAUSE
C
C     TEST SOLUTIONS AGAINST ESTABLISHED CRITERIA
 198  DO 10 K=2,KLAST
      DO 10 J=2,JLAST
      DO 10 I=2,ILAST
      IF(ABSF(DIFF(I,J,K))-ERR) 10,10,1
 10   CONTINUE
C
C     PRINT RESULTS OF ITERATION
      IF(CHOSE)87,87,88
 87   PUNCH 60
      GO TO 89
 88   PRINT 60
 60   FORMAT (//10X,26HFINAL RESULTS OF ITERATION//)
 89   DO 7 K=2,KLAST,KRITE
      DO 7 K=2,KLAST,KRITE
      DO 7 J=2,JLAST,JRITE
      DO 7 I=2,ILAST,IRITE
      IF(CHOSE)90,90,91
```

```
 90    PUNCH 52,I,J,K,U(I,J,K),DIFF(I,J,K)
       GO TO 7
 91    PRINT 52,I,J,K,U(I,J,K),DIFF(I,J,K)
 52    FORMAT(10X,3I4,2E14.8)
 7     CONTINUE
C
C      CHECK TO SEE IF LAST TIME INCREMENT HAS BEEN REACHED
       IF(TMAX-TIME)17,17,12
C
C      PREPARE FOR A NEW TIME INCREMENT
 12    DO 15 I=2,LASTX
       DO 15 J=2,LASTY
       DO 15 K=2,LASTZ
 15    UOLD(I,J,K)=U(I,J,K)
       GO TO 14
C
C      PRINT OUT INPUT DATA
 17    IF(CHOSE)93,93,94
 93    PUNCH 55,DIFUS,COND,DELT,SORC
       PUNCH 57,XGRID,YGRID,ZGRID,TMAX,ERR
       PUNCH 58,IRITE,JRITE,KRITE,KITER,JITER,CHOSE
       PUNCH 766,CONVRG
       GO TO 13
 94    PRINT 55,DIFUS,COND,DELT,SORC
  55   FORMAT(11HDIFFUSIVITY,F10.4,6X,12HCONDUCTIVITY,F10.4/
     310HDELTA TIME,F10.4,6X,15HSOURCE STRENGTH,F10.4)
       PRINT 57,XGRID,YGRID,ZGRID,TMAX,ERR
 57    FORMAT(/6HXGRID=,F10.4,6X,6HYGRID=,F10.4,6X,6HZGRID=,F10.4/
     48HMAX TIME,F10.4,18HSTOP WHEN ABSDIFF-,E14.8,16H IS ZERO OR LESS)
       PRINT 58,IRITE,JRITE,KRITE,KITER,JITER,CHOSE
 58    FORMAT(/25HOUTPUT CONTRCLS ARE IRITE,I3,2X,5HJRITE,I3/
     55HKRITE,I3,2X,5HKITER,I3,2X,5HJITER,I3,2X,5HCHOSE,F6.3)
       PRINT 766,CONVRG
 766   FORMAT (38HWHEN OMEGA(N+1) -OMEGA(N) IS LESS THAN,E14.8/
     253HSTOP COMPUTATION AND START OVERRELAXATION, IF DESIRED)
 13    STOP
       END
```

```
GA/PHYS/63-8
C
C      ANALYTICAL SOLUTION-3 DIMENSIONS, CONSTANT SOURCE
C
C      SPECIFY DIMENSIONED VARIABLES
       DIMENSION A(30),B(30),C(30)
C
C      READ INPUT DATA
       READ INPUT TAPE 2,60,LMAX,MMAX,NMAX
 60    FORMAT(3I3)
       READ INPUT TAPE 2,61,DIFUS,SORC,COND,XMAX,YMAX,ZMAX
 61    FORMAT(6F10.0)
       READ INPUT TAPE 2,62,DELX,DELY,DELZ,DELT
 62    FORMAT(4F10.0)
       READ INPUT TAPE 2,63,TMAX,XGRID,YGRID,ZGRID
 63    FORMAT(4F10.0)
C
C      INITIALIZE AND SET DESIRED TIME
       TIME=0.
 5     TIME=TIME+DELT
       WRITE OUTPUT TAPE 3,51,TIME
 51    FORMAT(/5X,18HNEW TIME INCREMENT,F10.5)
C
C      PRINT COLUMN HEADER FOR RESULTS
       WRITE OUTPUT TAPE 3,53
 53    FORMAT(//8X,1HL,3X,1HM,3X,1HN,4X,11HTEMPERATURE,9X,4HTINC)
C
C
C      COMPUTE CONSTANTS
       PI=3.1415926
       D=64.*DIFUS*SORC/(COND*PI*PI*PI)
       E=DIFUS*PI*PI
       DO 32 I=1,LMAX
 32    A(I)=2*(I-1)+1
       DO 33 J=1,MMAX
 33    B(J)=2*(J-1)+1
       DO 34 K=1,NMAX
 34    C(K)=2*(K-1)+1
C
C      INITIALIZE AND SET DESIRED X,Y,Z
       Z=0.
 35    Z=Z+DELZ
       Y=0.
 30    Y=Y+DELY
       X=0.
 8     X=X+DELX
       TEMP=0.
C
C      INITIALIZE SUMMATION VARIABLE FOR TEMPERATURE INCREMENTS
       TINC=0.
C
C      COMPUTE TEMPERATURE INCREMENTS
```

```
      DO 10 N=1,NMAX
      DO 10 M=1,MMAX
      DO 10 L=1,LMAX
      F=A(L)**2/XMAX**2+B(M)**2/YMAX**2+C(N)**2/ZMAX**2
      ALPHA=E*F
      DENOM=A(L)*B(M)*C(N)*ALPHA
      P=A(L)*PI*X/XMAX
      Q=B(M)*PI*Y/YMAX
      R=C(N)*PI*Z/ZMAX
      S=SINF(P)*SINF(Q)*SINF(R)
      T=ALPHA*TIME
C
C     RESTRICT EXPONENT TO PREVENT SUBROUTINE OVERFLOW
      IF(T-30.) 3,3,2
 2    V=1.0
      GO TO 6
 3    V=1.-(1./EXPF(T))
 6    TINC=D*S*V/DENOM
C
C     SUM TEMPERATURE INCREMENTS
      TEMP=TEMP+TINC
C
C     PRINT RESULTS WHEN DO LOOP INDICES ARE EQUAL FOR MONITOR ONLY
      IF(L-M)10,7,10
 7    IF(M-N)10,1,10
 1    I=L-1
      J=M-1
      K=N-1
      WRITE OUTPUT TAPE 3,52,I,J,K,TEMP,TINC
 52   FORMAT(/5X,3I4,2X,E14.8,2X,E14.8/)
 10   CONTINUE
C
C     PRINT FINAL RESULTS OF COMPUTATION
      WRITE OUTPUT TAPE 3,64
 64   FORMAT(/5X,13HFINAL RESULTS/)
      WRITE OUTPUT TAPE 3,50,X,Y,Z,TEMP
 50   FORMAT(5X,2HX=,F10.4,4H  Y=,F10.4,4H  Z=,F10.4,3X,5HTEMP=,E14.8/)
C
C     CHECK TO SEE IF LAST TIME AND SPACE INCREMENTS
C     HAVE BEEN REACHED
      IF(XGRID-X)15,15,8
 15   IF(YGRID-Y)20,20,30
 20   IF(ZGRID-Z)25,25,35
 25   IF(TMAX-TIME)31,31,5
C
C     PRINT OUT INPUT DATA FOR REFERENCE PURPOSES
 31   LL=LMAX-1
      MM=MMAX-1
      NN=NMAX-1
      WRITE OUTPUT TAPE 3,54,LL,MM,NN
```

```
   54    FORMAT(/5X,25HMAXIMUM SUMMATION INDEXES,3I5)
         WRITE OUTPUT TAPE 3,55,DIFUS,SORC
   55    FORMAT(/5X,11HDIFFUSIVITY,F10.4,3X,15HSOURCE STRENGTH,F10.4)
         WRITE OUTPUT TAPE 3,56,COND,XMAX,YMAX,ZMAX
   56    FORMAT(/5X,12HCONDUCTIVITY,F9.4,3X,13HMAXIMUM X,Y,Z,3F10.5)
         CALL EXIT
         END
```

```
C       HEAT TRANSFER IN CUBE-NONSTEADY, CONSTANT SOURCE BY
C       SUCCESSIVE OVERRELAXATION
C
        DIMENSION U(7,7,7),UOLD(7,7,7),DIFF(7,7,7)
C
C       READ INPUT DATA
        READ 61, LASTX,LASTY,LASTZ
 61     FORMAT (3I3)
        READ53,DIFUS,COND,DELT,SORC
 53     FORMAT(4F10.0)
        READ 54,ERR,XGRID,YGRID,ZGRID,TMAX
 54     FORMAT(5F10.0)
        READ 56,IRITE,JRITE,KRITE,JITER,KITER,CHOSE,BOUND
 56     FORMAT(5I4,2F10.0)
        READ 59,RFACT
 59     FORMAT (E14.8)
C
C       SET BOUNDARY VALUES
        DO 3 I=1,LASTX
        DO 3 J=1,LASTY
        DO 3 K=1,LASTZ
C
C       THE INPUT BOUND DETERMINES METHOD OF SETTING INITIAL
C       ESTIMATES OF TEMPERATURE( BOUND= 0.0 OR LESS AND READ
C       ESTIMATE FROM CARDS, IF BOUND IS GREATER THAN ZERO ALL
C       POINTS ARE SET TO ZERO)
        IF(BOUND) 100,100,101
 100    READ 102,U(I,J,K)
 102    FORMAT (E14.8)
        UOLD(I,J,K)=U(I,J,K)
        GO TO 3
 101    UOLD(I,J,K)=0.0
        U(I,J,K)=0.0
 3      CONTINUE
C
C       COMPUTE CONSTANTS
        A=(2./(XGRID**2))+(2./(YGRID**2))+(2./(ZGRID**2))
        B=DIFUS*DELT
        C=B/(XGRID**2)
        D=B/(YGRID**2)
        E=B/(ZGRID**2)
        G=1./(1.+B*A)
        F=G*B*SORC/COND
C
C       SET TIME STEPS AND ITERATION COUNTER
        TIME=0.
 14     TIME=TIME+DELT
        ITNUM=0
C
```

```
C       THE INPUT CHOSE DETERMINES WHETHER THE OUPUT IS PUNCHED
C       OR PRINTED (SET CHOSE = 0.0 OR LESS FOR PUNCHED OUTPUT
C       OR GREATER THAN ZERO FOR PRINTED OUTPUT)
        IF(CHOSE)80,80,81
 80     PUNCH 50,TIME
        GO TO 1
 81     PRINT 50,TIME
 50     FORMAT(/18HNEW TIME INCREMENT,F10.3)
 1      ITNUM=ITNUM+1
        IF(CHOSE)82,82,83
 82     PUNCH 51,ITNUM
        GO TO 84
 83     PRINT 51,ITNUM
 51     FORMAT (/15X,16HITERATION NUMBER,I4/)
C
C       COMPUTATIONAL STEPS
 84     ILAST=LASTX-1
        JLAST=LASTY-1
        KLAST=LASTZ-1
        DO 2 K=2,KLAST
        DO 2 J=2,JLAST
        DO 2 I=2,ILAST
        P=G*C*(U(I+1,J,K)+U(I-1,J,K))
        Q=G*D*(U(I,J+1,K)+U(I,J-1,K))
        R=G*E*(U(I,J,K+1)+U(I,J,K-1))
        S=G*UOLD(I,J,K)
        UNEW=((F+P+Q+R+S)*RFACT)+((1.0-RFACT)*U(I,J,K))
        DIFF(I,J,K)=UNEW-U(I,J,K)
C
C       COMPLETE OR PARTIAL OUTPUT OF TEMPERATURES CAN BE OBTAINED
C       FOR EACH ITERATION BY PROPER CHOICE OF THE INPUTS JITER AND KITER
 116    IF(JITER-J)110,110,2
 110    IF(KITER-K)8,8,2
 8      IF(CHOSE)85,85,86
 85     PUNCH 52,I,J,K,UNEW,DIFF(I,J,K)
        GO TO 2
 86     PRINT 52,I,J,K,UNEW,DIFF(I,J,K)
 52     FORMAT(10X,3I4,2E14.8)
 2      U(I,J,K)=UNEW
C
C       SET VALUES ONE GRID POINT BEYOND PLANES OF SYMMETRY
        L=LASTX
        DO 4 N=2,LASTZ
        DO 4 M=2,LASTY
 4      U(L,M,N)=U(L-2,M,N)
        M=LASTY
        DO 5 L=2,LASTX
        DO 5 N=2,LASTZ
 5      U(L,M,N)=U(L,M-2,N)
        N=LASTZ
```

```
      DO 6 L=2,LASTX
      DO 6 M=2,LASTZ
 6    U(L,M,N)=U(L,M,N-2)
C
C     TEST SOLUTIONS AGAINST ESTABLISHED CRITERIA
      DO 10 K=2,KLAST
      DO 10 J=2,JLAST
      DO 10 I=2,ILAST
      IF(ABSF(DIFF(I,J,K))-ERR) 10,10,1
 10   CONTINUE
C
C     PRINT RESULTS OF ITERATION
      IF(CHOSE)87,87,88
 87   PUNCH 60
      GO TO 89
 88   PRINT 60
 60   FORMAT (//10X,26HFINAL RESULTS OF ITERATION//)
 89   DO 7 K=2,KLAST,KRITE
      DO 7 J=2,JLAST,JRITE
      DO 7 I=2,ILAST,IRITE
      IF(CHOSE)90,90,91
 90   PUNCH 52,I,J,K,U(I,J,K),DIFF(I,J,K)
      GO TO 7
 91   PRINT 52,I,J,K,U(I,J,K),DIFF(I,J,K)
 7    CONTINUE
C
C     CHECK TO SEE IF LAST TIME INCREMENT HAS BEEN REACHED
      IF(TMAX-TIME)17,17,12
C
C     PREPARE FOR A NEW TIME INCREMENT
 12   DO 15 I=2,LASTX
      DO 15 J=2,LASTY
      DO 15 K=2,LASTZ
 15   UOLD(I,J,K)=U(I,J,K)
      GO TO 14
C
C     PRINT OUT INPUT DATA
 17   IF(CHOSE)93,93,94
 93   PUNCH 55,DIFUS,COND,DELT,SORC
      PUNCH 57,XGRID,YGRID,ZGRID,TMAX,ERR
      PUNCH 58,IRITE,JRITE,KRITE,KITER,JITER,CHOSE
      PUNCH 62, REACT
      GO TO 13
   94   PRINT 55,DIFUS,COND,DELT,SORC
  55  FORMAT(11HDIFFUSIVITY,F10.4,6X,12HCONDUCTIVITY,F10.4/
     310HDELTA TIME,F10.4,6X,15HSOURCE STRENGTH,F10.4)
      PRINT 57,XGRID,YGRID,ZGRID,TMAX,ERR
 57   FORMAT(/6HXGRID=,F10.4,6X,6HYGRID=,F10.4,6X,6HZGRID=,F10.4/
     48HMAX TIME,F10.4,18HSTOP WHEN ABSDIFF-,E14.8,16H IS ZERO OR LESS)
```

```
      PRINT 58,IRITE,JRITE,KRITE,KITER,JITER,CHOSE
58    FORMAT(/25HOUTPUT CONTRCLS ARE IRITE,I3,2X,5HJRITE,I3/
     55HKRITE,I3,2X,5HKITER,I3,2X,5HJITER,I3,2X,5HCHOSE,F6.3)
      PRINT 62, RFACT
62    FORMAT(/22HRELAXATION FACTOR USED,3X,E14.8)
13    STOP
      END
```

GA/PHYS/63-8

```
C       HEAT TRANSFER,CONSTANT INTERNAL SOURCE, TWO DIMENSIONS
C       SOLUTION BY GAUSS SIEDEL WITH SERIAL SCAN
C
        DIMENSION U(41,41),UOLD(41,41),DIFR(41,41)
        READ INPUT TAPE 2,700,LASTX,LASTY,IRITE,JRITE,JITER
  700   FORMAT (5I4)
        READ INPUT TAPE 2,701,DIFUS,COND,SORC,XGRID,YGRID
  701   FORMAT (5F12.0)
        READ INPUT TAPE 2,702,TMAX,DELT,ERR
  702   FORMAT (3F12.0)
        READ INPUT TAPE 2,703,BOUND,PITER
  703   FORMAT (2F10.0)
C
C       PRINT OUT INPUT DATA FOR REFERENCE PURPOSES
        WRITE OUTPUT TAPE 3,53,DELT,SORC
  53    FORMAT (11HDELTA TIME=,F10.4,6X,16HSOURCE STRENGTH=,F10.4/)
        WRITE OUTPUT TAPE 3,55,DIFUS,COND
  55    FORMAT(/12HDIFFUSIVITY=,F10.4,6X,13HCONDUCTIVITY=,F10.4/)
        WRITE OUTPUT TAPE 3,56
  56    FORMAT(/27HTHIS RUN IS 2D GAUSS-SIEDEL/)
        WRITE OUTPUT TAPE 3,57,XGRID,YGRID,TMAX
  57    FORMAT (6HXGRID=,F10.4,3X,6HYGRID=,F10.4,3X,9HMAX TIME=,F10.4/)
        WRITE OUTPUT TAPE 3,58,IRITE,JRITE,JITER
  58    FORMAT (/6HIRITE=,I4,8H  JRITE=,I4,8H  JITER=,I4/)
C
C       IF BOUND = 0.0 OR -1.0,READ INITIAL VALUES AS INPUT DATA, OR IF
C       BOUND = 1.0,SET ALL POINTS EQUAL ZERO.
        DO 3 J=1,LASTY
        DO 3 I=1,LASTX
        IF (BOUND) 100,100,101
  100   READ INPUT TAPE 2,704,U(I,J)
  704   FORMAT (F12.0)
        DIFR(I,J)=0.0
        UOLD(I,J)=U(I,J)
        GO TO 3
  101   UOLD(I,J)=0.0
        DIFR(I,J)=0.0
        U(I,J)=0.0
    3   CONTINUE
C
C       COMPUTE CONSTANTS
        ILAST=LASTX-1
        JLAST=LASTY-1
  201   A=(2./(XGRID**2))+(2./(YGRID**2))
        B=DIFUS*DELT
        C=B/(XGRID**2)
        D=B/(YGRID**2)
        G=1./(1.+B*A)
        F=G*B*SORC/COND
C
C       SET TIME STEPS AND ITERATION COUNTER
        TIME=0.
   14   TIME=TIME+DELT
        ITNUM=0
        WRITE OUTPUT TAPE 3,50,TIME
```

```
 50   FORMAT(/18HNEW TIME INCREMENT,F10.3)
 1    ITNUM=ITNUM+1
C
C     WHEN PRINTING OUT DATA EACH ITERATION SET PITER =0.0,OR -1.0
C     OTHERWISE SET PITER = 1.0.
      IF (PITER) 306,306,84
 306  WRITE OUTPUT TAPE 3,51,ITNUM
 51   FORMAT (/15X,16HITERATION NUMBER,I4/)
C
C     COMPUTATIONAL STEPS
 84   DO 2 J=2,JLAST
      DO 2 I=2,ILAST
      P=G*C*(U(I+1,J)+U(I-1,J))
      Q=G*D*(U(I,J+1)+U(I,J-1))
      S=G*UOLD(I,J)
      UNEW=F+P+Q+S
      DIFR(I,J)=UNEW-U(I,J)
 116  IF(JITER-J) 110,110,2
 110  WRITE OUTPUT TAPE 3,52,I,J,UNEW,DIFR(I,J)
 2    U(I,J)=UNEW
C
C     CHECK CONVERGENCE OF SOLUTION AGAINST ESTABLISHED CRITERIA
 29   DO 10 J=2,JLAST
      DO 10 I=2,ILAST
      IF(ABSF(DIFR(I,J))-ERR) 10,10,1
 10   CONTINUE
C
C     PRINT OUT FINAL RESULTS OF ITERATION
      WRITE OUTPUT TAPE 3,60
      WRITE OUTPUT TAPE 3,51,ITNUM
      WRITE OUTPUT TAPE 3,54,ERR
 54   FORMAT(/34HSTOP ITERATION IF ABS(T(N+1)-T(N)-,E14.8,10H=0 OR LESS)
 60   FORMAT (//31H     FINAL RESULTS OF ITERATION//)
      WRITE OUTPUT TAPE 3,602
 602  FORMAT (46H  I    J        TEMPERATURE      DIFF,T(N+1)-T(N))
 89   DO 7 J=1,LASTY,JRITE
      DO 7 I=1,LASTX,IRITE
 7    WRITE OUTPUT TAPE 3,52,I,J,U(I,J),DIFR(I,J)
 52   FORMAT (2I4,4X,E15.8,4X,E15.8)
C
C     IF DESIRED,READ IN NEW CONVERGENCE CRITERIA AND START WHERE LEFT OFF
      READ INPUT TAPE 2,705,ERR
 705  FORMAT (F10.0)
      IF(100.-ERR) 302,300,300
 300  WRITE OUTPUT TAPE 3,305,ERR
 305  FORMAT(//27HNEW CONVERGENCE CRITERIA = ,E14.8//)
      GO TO 1
C
C     CHECK TO SEE IF LAST TIME INCREMENT HAS BEEN REACHED
 302  IF(TMAX-TIME) 17,17,12
 12   DO 15 I=2,LASTX
      DO 15 J=2,LASTY
 15   UOLD(I,J)=U(I,J)
      GO TO 14
C
 17   CALL EXIT
```

```
C      HEAT TRANSFER,CONSTANT INTERNAL SOURCE, TWO DIMENSIONS
C      SOLUTION BY GAUSS SIEDEL WITH SPIRAL SCAN
       DIMENSION U(41,41),UOLD(41,41),DIFR(41,41)
       READ INPUT TAPE 2,700,LASTX,LASTY,IRITE,JRITE,JITER
  700  FORMAT (5I4)
       READ INPUT TAPE 2,701,DIFUS,COND,SORC,XGRID,YGRID
  701  FORMAT (5F12.0)
       READ INPUT TAPE 2,702,TMAX,DELT,ERR
  702  FORMAT (3F12.0)
       READ INPUT TAPE 2,703,BOUND,PITER
  703  FORMAT (2F10.0)
C
C      PRINT OUT INPUT DATA FOR REFERENCE PURPOSES
       WRITE OUTPUT TAPE 3,53,DELT,SORC
  53   FORMAT (11HDELTA TIME=,F10.4,6X,16HSOURCE STRENGTH=,F10.4/)
       WRITE OUTPUT TAPE 3,55,DIFUS,COND
  55   FORMAT(/12HDIFFUSIVITY=,F10.4,6X,13HCONDUCTIVITY=,F10.4/)
       WRITE OUTPUT TAPE 3,56
  56   FORMAT(/44HTHIS RUN IS 2D GAUSS-SIEDEL WITH FULL SPIRAL/)
       WRITE OUTPUT TAPE 3,57,XGRID,YGRID,TMAX
  57   FORMAT (6HXGRID=,F10.4,3X,6HYGRID=,F10.4,3X,9HMAX TIME=,F10.4/)
       WRITE OUTPUT TAPE 3,58,IRITE,JRITE,JITER
  58   FORMAT (/6HIRITE=,I4,8H  JRITE=,I4,8H  JITER=,I4/)
C
C      IF BOUND = 0.0 OR -1.0,READ INITIAL VALUES AS INPUT DATA, OR IF
C      BOUND = 1.0,SET ALL POINTS EQUAL ZERO.
       DO 3 J=1,LASTY
       DO 3 I=1,LASTX
       IF (BOUND) 100,100,101
  100  READ INPUT TAPE 2,704,U(I,J)
  704  FORMAT (F12.0)
       DIFR(I,J)=0.0
       UOLD(I,J)=U(I,J)
       GO TO 3
  101  UOLD(I,J)=0.0
       DIFR(I,J)=0.0
       U(I,J)=0.0
  3    CONTINUE
C
C      COMPUTE CONSTANTS
       ILAST=LASTX-1
       JLAST=LASTY-1
  201  A=(2./(XGRID**2))+(2./(YGRID**2))
       B=DIFUS*DELT
       C=B/(XGRID**2)
       D=B/(YGRID**2)
       G=1./(1.+B*A)
       F=G*B*SORC/COND
C
C      SET TIME STEPS AND ITERATION COUNTER
       TIME=0.
  14   TIME=TIME+DELT
       ITNUM=0
C
       WRITE OUTPUT TAPE 3,50,TIME
```

```
 50   FORMAT(/18HNEW TIME INCREMENT,F10,3)
 1    ITNUM=ITNUM+1
C
C     WHEN PRINTING OUT DATA EACH ITERATION SET PITER =0.0,OR -1.0
C     OTHERWISE SET PITER = 1.0.
      IF (PITER) 306,306,84
 306  WRITE OUTPUT TAPE 3,51,ITNUM
 51   FORMAT (/15X,16HITERATION NUMBER,I4/)
C
C     COMPUTATIONAL STEPS
 84   II=1
      JJ=2
      III=ILAST
      JJJ=JLAST
      COUNT=0.0
 30   COUNT=COUNT+1.0
      IF(COUNT-1.0) 31,31,32
 31   J=JJ
      II=II+1
      GO TO 34
 32   IF(COUNT-2.0) 33,33,36
 33   III=III-1
 45   J=JJJ
 34   DO 35 I=II,III
      P=G*C*(U(I+1,J)+U(I-1,J))
      Q=G*D*(U(I,J+1)+U(I,J-1))
      S=G*UOLD(I,J)
      UNEW=F+P+Q+S
      DIFR(I,J)=UNEW-U(I,J)
 116  IF(JITER-J) 110,110,35
 110  WRITE OUTPUT TAPE 3,52,I,J,UNEW,DIFR(I,J)
 35   U(I,J)=UNEW
      IF(II-III) 46,29,46
 46   GO TO 30
 36   IF(COUNT-3.0) 37,37,38
 37   I=II
      JJ=JJ+1
      JJJ=JJJ-1
      JEND=JJJ
      GO TO 39
 38   I=III+1
      JEND=JJJ+1
 39   DO 44 J=JJ,JEND
      P=G*C*(U(I+1,J)+U(I-1,J))
      Q=G*D*(U(I,J+1)+U(I,J-1))
      S=G*UOLD(I,J)
      UNEW=F+P+Q+S
      DIFR(I,J)=UNEW-U(I,J)
 40   IF(JITER-J) 41,41,44
 41   WRITE OUTPUT TAPE 3,52,I,J,UNEW,DIFR(I,J)
 44   U(I,J)=UNEW
      IF(COUNT-3.0) 30,30,47
 47   COUNT=0.0
      GO TO 30
C
C     CHECK CONVERGENCE OF SOLUTION AGAINST ESTABLISHED CRITERIA
```

```
 29   DO 10 J=2,JLAST
      DO 10 I=2,ILAST
      IF(ABSF(DIFR(I,J))-ERR) 10,10,1
 10   CONTINUE
C
C     PRINT OUT FINAL RESULTS OF ITERATION
      WRITE OUTPUT TAPE 3,60
      WRITE OUTPUT TAPE 3,51,ITNUM
      WRITE OUTPUT TAPE 3,54,ERR
 54   FORMAT(//34HSTOP ITERATION IF ABS(T(N+1)-T(N)-,E14.8,10H=C OR LESS)
 60   FORMAT (///31H      FINAL RESULTS OF ITERATION//)
      WRITE OUTPUT TAPE 3,602
 602  FORMAT (46H  I    J        TEMPERATURE      DIFF,T(N+1)-T(N))
 89   DO 7 J=1,LASTY,JRITE
      DO 7 I=1,LASTX,IRITE
 7    WRITE OUTPUT TAPE 3,52,I,J,U(I,J),DIFR(I,J)
 52   FORMAT (2I4,4X,E15.8,4X,E15.8)
C
C     IF DESIRED,READ IN NEW CONVERGENCE CRITERIA AND START WHERE LEFT OFF
      READ INPUT TAPE 2,705,ERR
 705  FORMAT (F10.0)
      IF(100.-ERR) 302,300,300
 300  WRITE OUTPUT TAPE 3,305,ERR
 305  FORMAT(//27HNEW CONVERGENCE CRITERIA = ,E14.8//)
      GO TO 1
C
C     CHECK TO SEE IF LAST TIME INCREMENT HAS BEEN REACHED
 302  IF(TMAX-TIME) 17,17,12
 12   DO 15 I=2,LASTX
      DO 15 J=2,LASTY
 15   UOLD(I,J)=U(I,J)
      GO TO 14
 17   CALL EXIT
      END(1,0,0,0,0,0,1,0,0,1,0,0,0,0,0)
```

GA/PHYS/63-8

OVERRELAXATION DATA RUN TIME = 0.2 SEC,H = 0.1 CM,ERR = 0.005

NEW TIME INCREMENT .200

ITERATION NUMBER 1

node			temperature	residual
2	6	6	.11835672E+02	.11835672E+02
3	6	6	.18639951E+02	.18639951E+02
4	6	6	.22405365E+02	.22405365E+02
5	6	6	.24404534E+02	.24404534E+02
6	6	6	.25423319E+02	.25423319E+02

ITERATION NUMBER 2

2	6	6	.19837496E+02	.80018240E+01
3	6	6	.31652511E+02	.13012560E+02
4	6	6	.38395091E+02	.15989726E+02
5	6	6	.42103991E+02	.17699457E+02
6	6	6	.43346459E+02	.17923140E+02

ITERATION NUMBER 3

2	6	6	.25360193E+02	.55226970E+01
3	6	6	.40854991E+02	.92024800E+01
4	6	6	.50007493E+02	.11612402E+02
5	6	6	.54920496E+02	.12816505E+02
6	6	6	.56663761E+02	.13317302E+02

ITERATION NUMBER 4

2	6	6	.29337214E+02	.39770210E+01
3	6	6	.47617934E+02	.67629430E+01
4	6	6	.58493375E+02	.84858820E+01
5	6	6	.64348760E+02	.94282640E+01
6	6	6	.66207560E+02	.95437990E+01

ITERATION NUMBER 5

2	6	6	.32258438E+02	.29212240E+01
3	6	6	.52541071E+02	.49231370E+01
4	6	6	.64693040E+02	.61996650E+01
5	6	6	.71143600E+02	.67948400E+01
6	6	6	.73277410E+02	.70698500E+01

ITERATION NUMBER 6

2	6	6	.34390562E+02	.21321240E+01
3	6	6	.56157207E+02	.36161360E+01
4	6	6	.69202630E+02	.45095900E+01

```
5   6   6 .76175C30E+02 .50314300E+01
6   6   6 .78357930E+02 .50805200E+01

  ITERATION NUMBER   7

2   6   6 .35958967E+02 .15684050E+01
3   6   6 .58782838E+02 .26256310E+01
4   6   6 .72517520E+02 .33148900E+01
5   6   6 .79789910E+02 .36148800E+01
6   6   6 .82129770E+02 .37718400E+01

  ITERATION NUMBER   8

2   6   6 .37093114E+02 .11341470E+01
3   6   6 .60712752E+02 .19299140E+01
4   6   6 .74915920E+02 .23984000E+01
5   6   6 .82479230E+02 .26893200E+01
6   6   6 .84836000E+02 .27062300E+01

  ITERATION NUMBER   9

2   6   6 .37930510E+02 .83739600E+00
3   6   6 .62109970E+02 .13972180E+01
4   6   6 .76687140E+02 .17712200E+01
5   6   6 .84398650E+02 .19194200E+01
6   6   6 .86847700E+02 .20117000E+01

  ITERATION NUMBER  10

2   6   6 .38532838E+02 .60232800E+00
3   6   6 .63139090E+02 .10291200E+01
4   6   6 .77958190E+02 .12710500E+01
5   6   6 .85836100E+02 .14374500E+01
6   6   6 .88286590E+02 .14388900E+01

  ITERATION NUMBER  11

2   6   6 .38980282E+02 .44744400E+00
3   6   6 .63880880E+02 .74179000E+00
4   6   6 .78908140E+02 .94995000E+00
5   6   6 .86855170E+02 .10190700E+01
6   6   6 .89361170E+02 .10745800E+01

  ITERATION NUMBER  12

2   6   6 .39299367E+02 .31908500E+00
3   6   6 .64432660E+02 .55178000E+00
4   6   6 .79580200E+02 .67206000E+00
5   6   6 .87623640E+02 .76847000E+00
6   6   6 .90125210E+02 .76404000E+00
```

```
  ITERATION NUMBER  13

2    6    6 .39540271E+02 .24090400E+00
3    6    6 .64824630E+02 .39197000E+00
4    6    6 .80090450E+02 .51025000E+00
5    6    6 .88164880E+02 .54124000E+00
6    6    6 .90700150E+02 .57494000E+00

  ITERATION NUMBER  14

2    6    6 .39707595E+02 .16732400E+00
3    6    6 .65121470E+02 .29684000E+00
4    6    6 .80445510E+02 .35506000E+00
5    6    6 .88575520E+02 .41064000E+00
6    6    6 .91105250E+02 .40510000E+00

  ITERATION NUMBER  15

2    6    6 .39838593E+02 .13099800E+00
3    6    6 .65328000E+02 .20653000E+00
4    6    6 .80719400E+02 .27389000E+00
5    6    6 .88862960E+02 .28744000E+00
6    6    6 .91413090E+02 .30784000E+00

  ITERATION NUMBER  16

2    6    6 .39925239E+02 .86646000E-01
3    6    6 .65487780E+02 .15978000E+00
4    6    6 .80907200E+02 .18780000E+00
5    6    6 .89082480E+02 .21952000E+00
6    6    6 .91627760E+02 .21467000E+00

  ITERATION NUMBER  17

2    6    6 .39997249E+02 .72010000E-01
3    6    6 .65596810E+02 .10903000E+00
4    6    6 .81054130E+02 .14693000E+00
5    6    6 .89235080E+02 .15260000E+00
6    6    6 .91792690E+02 .16493000E+00

  ITERATION NUMBER  18

2    6    6 .40041580E+02 .44331000E-01
3    6    6 .65682530E+02 .85720000E-01
4    6    6 .81153440E+02 .99310000E-01
5    6    6 .89352440E+02 .11736000E+00
6    6    6 .91906360E+02 .11367000E+00
```

ITERATION NUMBER 19

2	6	6	.40081412E+02	.39832000E-01
3	6	6	.65740310E+02	.57780000E-01
4	6	6	.81232300E+02	.78860000E-01
5	6	6	.89433450E+02	.81010000E-01
6	6	6	.91994770E+02	.88410000E-01

ITERATION NUMBER 20

2	6	6	.40103881E+02	.22469000E-01
3	6	6	.65786110E+02	.45800000E-01
4	6	6	.81284740E+02	.52440000E-01
5	6	6	.89496200E+02	.62750000E-01
6	6	6	.92054930E+02	.60160000E-01

ITERATION NUMBER 21

2	6	6	.40125995E+02	.22114000E-01
3	6	6	.65816870E+02	.30760000E-01
4	6	6	.81327140E+02	.42400000E-01
5	6	6	.89539200E+02	.43000000E-01
6	6	6	.92102350E+02	.47420000E-01

ITERATION NUMBER 22

2	6	6	.40137283E+02	.11288000E-01
3	6	6	.65841240E+02	.24370000E-01
4	6	6	.81354770E+02	.27630000E-01
5	6	6	.89572760E+02	.33560000E-01
6	6	6	.92134160E+02	.31810000E-01

ITERATION NUMBER 23

2	6	6	.40149585E+02	.12302000E-01
3	6	6	.65857690E+02	.16450000E-01
4	6	6	.81377620E+02	.22850000E-01
5	6	6	.89595570E+02	.22810000E-01
6	6	6	.92159600E+02	.25440000E-01

ITERATION NUMBER 24

2	6	6	.40155211E+02	.56260000E-02
3	6	6	.65870600E+02	.12910000E-01
4	6	6	.81392120E+02	.14500000E-01
5	6	6	.89613520E+02	.17950000E-01
6	6	6	.92176410E+02	.16810000E-01

ITERATION NUMBER 25

2	6	6	.40162044E+02	.68330000E-02
3	6	6	.65879430E+02	.88300000E-02
4	6	6	.81404490E+02	.12370000E-01
5	6	6	.89625630E+02	.12110000E-01
6	6	6	.92190C80E+02	.13670000E-01

ITERATION NUMBER 26

2	6	6	.40164849E+02	.28050000E-02
3	6	6	.65886260E+02	.68300000E-02
4	6	6	.81412050E+02	.75600000E-02
5	6	6	.89635220E+02	.95900000E-02
6	6	6	.92198960E+02	.88800000E-02

ITERATION NUMBER 27

2	6	6	.40168617E+02	.37680000E-02
3	6	6	.65891C00E+02	.47400000E-02
4	6	6	.81418780E+02	.67300000E-02
5	6	6	.89641660E+02	.64400000E-02
6	6	6	.92206290E+02	.73300000E-02

ITERATION NUMBER 28

2	6	6	.40170C28E+02	.14110000E-02
3	6	6	.65894620E+02	.36200000E-02
4	6	6	.81422690E+02	.39100000E-02
5	6	6	.89646760E+02	.51000000E-02
6	6	6	.92210980E+02	.46900000E-02

ITERATION NUMBER 29

2	6	6	.40172096E+02	.20680000E-02
3	6	6	.65897150E+02	.25300000E-02
4	6	6	.81426350E+02	.36600000E-02
5	6	6	.89650210E+02	.34500000E-02
6	6	6	.92214920E+02	.39400000E-02

FINAL RESULTS OF ITERATION

node			temperature	residual
2	2	2	.11337487E+02	.22420000E-02
3	2	2	.16195678E+02	.19200000E-03
4	2	2	.18495888E+02	.20890000E-02

5	2	2	.19558723E+02	.61000000E-04
6	2	2	.19871819E+02	.16750000E-02
2	3	2	.16195676E+02	.18700000E-03
3	3	2	.24076847E+02	.79000000E-03
4	3	2	.28071184E+02	.77900000E-03
5	3	2	.29983233E+02	.11780000E-02
6	3	2	.30552516E+02	.94500000E-03
2	4	2	.18495887E+02	.20860000E-02
3	4	2	.28071186E+02	.77900000E-03
4	4	2	.33137749E+02	.23800000E-02
5	4	2	.35623870E+02	.94200000E-03
6	4	2	.36373500E+02	.21360000E-02
2	5	2	.19558725E+02	.61000000E-04
3	5	2	.24983231E+02	.11720000E-02
4	5	2	.35623872E+02	.94600000E-03
5	5	2	.38433894E+02	.17440000E-02
6	5	2	.39286017E+02	.12150000E-02
2	6	2	.19871821E+02	.16760000E-02
3	6	2	.30552516E+02	.94400000E-03
4	6	2	.36373503E+02	.21390000E-02
5	6	2	.39286017E+02	.12130000E-02
6	6	2	.40172094E+02	.20640000E-02
2	2	3	.16195675E+02	.18800000E-03
3	2	3	.24076848E+02	.79200000E-03
4	2	3	.28071187E+02	.78200000E-03
5	2	3	.29983231E+02	.11730000E-02
6	2	3	.30552520E+02	.95200000E-03
2	3	3	.24076848E+02	.79300000E-03
3	3	3	.37132153E+02	.12610000E-02
4	3	3	.44140288E+02	.16250000E-02
5	3	3	.47593100E+02	.17370000E-02
6	3	3	.48633861E+02	.18450000E-02
2	4	3	.28071187E+02	.78000000E-03
3	4	3	.44140283E+02	.16160000E-02
4	4	3	.53091383E+02	.18840000E-02
5	4	3	.57599982E+02	.22970000E-02
6	4	3	.58972680E+02	.21910000E-02
2	5	3	.29983232E+02	.11730000E-02
3	5	3	.47593105E+02	.17460000E-02
4	5	3	.57599981E+02	.22940000E-02
5	5	3	.62705980E+02	.24100000E-02
6	5	3	.64270530E+02	.25800000E-02
2	6	3	.30552517E+02	.94400000E-03
3	6	3	.48633858E+02	.18410000E-02
4	6	3	.58972676E+02	.21850000E-02
5	6	3	.64270530E+02	.25800000E-02
6	6	3	.65897140E+02	.25100000E-02
2	2	4	.18495887E+02	.20900000E-02

3	2	4	.28071185E+02	.77900000E-03
4	2	4	.33137748E+02	.23770000E-02
5	2	4	.35623870E+02	.94500000E-03
6	2	4	.36373500E+02	.21320000E-02
2	3	4	.28071185E+02	.77900000E-03
3	3	4	.44140289E+02	.16230000E-02
4	3	4	.53091380E+02	.18720000E-02
5	3	4	.57599985E+02	.22990000E-02
6	3	4	.58972683E+02	.21930000E-02
2	4	4	.33137744E+02	.23710000E-02
3	4	4	.53091387E+02	.18890000E-02
4	4	4	.64588170E+02	.34500000E-02
5	4	4	.70494550E+02	.25200000E-02
6	4	4	.72310C30E+02	.34800000E-02
2	5	4	.35623874E+02	.95100000E-03
3	5	4	.57599976E+02	.22810000E-02
4	5	4	.70494550E+02	.25200000E-02
5	5	4	.77198940E+02	.32400000E-02
6	5	4	.79270700E+02	.29800000E-02
2	6	4	.36373498E+02	.21300000E-02
3	6	4	.58972678E+02	.21890000E-02
4	6	4	.72310C20E+02	.34700000E-02
5	6	4	.79270690E+02	.29700000E-02
6	6	4	.81426340E+02	.36400000E-02
2	2	5	.19558725E+02	.63000000E-04
3	2	5	.29983231E+02	.11730000E-02
4	2	5	.35623870E+02	.94400000E-03
5	2	5	.38433898E+02	.17520000E-02
6	2	5	.39286C16E+02	.12130000E-02
2	3	5	.29983233E+02	.11770000E-02
3	3	5	.47593105E+02	.17430000E-02
4	3	5	.57599985E+02	.22990000E-02
5	3	5	.62705970E+02	.23900000E-02
6	3	5	.64270530E+02	.25800000E-02
2	4	5	.35623874E+02	.95100000E-03
3	4	5	.57599977E+02	.22840000E-02
4	4	5	.70494550E+02	.25200000E-02
5	4	5	.77198940E+02	.32400000E-02
6	4	5	.79270700E+02	.29800000E-02
2	5	5	.38433898E+02	.17490000E-02
3	5	5	.62705980E+02	.24100000E-02
4	5	5	.77198940E+02	.32500000E-02
5	5	5	.84818670E+02	.33100000E-02
6	5	5	.87186380E+02	.36300000E-02
2	6	5	.39286018E+02	.12150000E-02
3	6	5	.64270530E+02	.25800000E-02
4	6	5	.79270680E+02	.29500000E-02
5	6	5	.87186370E+02	.36200000E-02

```
     6    6    5 .89650210E+02 .34500000E-02
     2    2    6 .19871821E+02 .16770000E-02
     3    2    6 .30552519E+02 .95000000E-03
     4    2    6 .36373501E+02 .21340000E-02
     5    2    6 .39286017E+02 .12160000E-02
     6    2    6 .40172C98E+02 .20700000E-02
     2    3    6 .30552517E+02 .94400000E-03
     3    3    6 .48633865E+02 .18510000E-02
     4    3    6 .58972679E+02 .21840000E-02
     5    3    6 .64270530E+02 .25800000E-02
     6    3    6 .65897130E+02 .24900000E-02
     2    4    6 .36373502E+02 .21370000E-02
     3    4    6 .58972683E+02 .21930000E-02
     4    4    6 .72310C30E+02 .34800000E-02
     5    4    6 .79270700E+02 .29800000E-02
     6    4    6 .81426340E+02 .36400000E-02
     2    5    6 .39286018E+02 .12150000E-02
     3    5    6 .64270530E+02 .25800000E-02
     4    5    6 .7927068OE+02 .29500000E-02
     5    5    6 .87186370E+02 .36200000E-02
     6    5    6 .89650210E+02 .34400000E-02
     2    6    6 .40172C96E+02 .20680000E-02
     3    6    6 .65897150E+02 .25300000E-02
     4    6    6 .81426350E+02 .36600000E-02
     5    6    6 .89650210E+02 .34500000E-02
     6    6    6 .92214920E+02 .39400000E-02
DIFFUSIVITY    1.0000      CONDUCTIVITY    1.0000
DELTA TIME      .2000      SOURCE STRENGTH 2000.0000

XGRID=     .1000      YGRID=     .1000      ZGRID=     .1000
MAX TIME    .2000STOP WHEN ABSDIFF- .50000000E-02 IS ZERO OR LESS

OUTPUT CONTROLS ARE IRITE  1  JRITE  1
KRITE  1  KITER  6  JITER  6  CHOSE 0.000
```

DATA RUN GAUSS SIEDEL T=0.2 SEC ERR=0.005 H = 0.1

NEW TIME INCREMENT .200

ITERATION NUMBER 1

	node		temperature	residual
2	6	6	.49352900E+01	.49352900E+01
3	6	6	.61509471E+01	.61509471E+01
4	6	6	.64493826E+01	.64493826E+01
5	6	6	.65222818E+01	.65222818E+01
6	6	6	.65399740E+01	.65399740E+01

ITERATION NUMBER 2

2	6	6	.88801862E+01	.39448962E+01
3	6	6	.11733575E+02	.55826280E+01
4	6	6	.12595459E+02	.61460770E+01
5	6	6	.12844798E+02	.63225170E+01
6	6	6	.12909294E+02	.63693200E+01

ITERATION NUMBER 3

2	6	6	.12185228E+02	.33050420E+01
3	6	6	.16783346E+02	.50497710E+01
4	6	6	.18385662E+02	.57902030E+01
5	6	6	.18910285E+02	.60654870E+01
6	6	6	.19052105E+02	.61428110E+01

ITERATION NUMBER 4

2	6	6	.15032849E+02	.28476210E+01
3	6	6	.21351303E+02	.45679570E+01
4	6	6	.23794226E+02	.54085640E+01
5	6	6	.24671974E+02	.57616890E+01
6	6	6	.24916819E+02	.58647140E+01

ITERATION NUMBER 5

2	6	6	.17527906E+02	.24950570E+01
3	6	6	.25486291E+02	.41349880E+01
4	6	6	.28814165E+02	.50199390E+01
5	6	6	.30098025E+02	.54260510E+01
6	6	6	.30464459E+02	.55476400E+01

ITERATION NUMBER 6

2	6	6	.19736819E+02	.22089130E+01
3	6	6	.29231800E+02	.37455090E+01
4	6	6	.33450971E+02	.46368060E+01
5	6	6	.35171236E+02	.50732110E+01
6	6	6	.35670706E+02	.52062470E+01

ITERATION NUMBER 7

2	6	6	.21705484E+02	.19686650E+01
3	6	6	.32626222E+02	.33944220E+01
4	6	6	.37718101E+02	.42671300E+01
5	6	6	.39886695E+02	.47154590E+01
6	6	6	.40524592E+02	.48538860E+01

ITERATION NUMBER 8

2	6	6	.23467933E+02	.17624490E+01
3	6	6	.35703599E+02	.30773770E+01
4	6	6	.41633842E+02	.39157410E+01
5	6	6	.44248880E+02	.43621850E+01
6	6	6	.45025903E+02	.45013110E+01

ITERATION NUMBER 9

2	6	6	.25050775E+02	.15828420E+01
3	6	6	.38494323E+02	.27907240E+01
4	6	6	.45219152E+02	.35853100E+01
5	6	6	.48268970E+02	.40200900E+01
6	6	6	.49182403E+02	.41565000E+01

ITERATION NUMBER 10

2	6	6	.26475626E+02	.14248510E+01
3	6	6	.41025641E+02	.25313180E+01
4	6	6	.48496204E+02	.32770520E+01
5	6	6	.51962594E+02	.36936240E+01
6	6	6	.53007382E+02	.38249790E+01

ITERATION NUMBER 11

2	6	6	.27760516E+02	.12848900E+01
3	6	6	.43322061E+02	.22964200E+01
4	6	6	.51487394E+02	.29911900E+01
5	6	6	.55348087E+02	.33854930E+01
6	6	6	.56517688E+02	.35103060E+01

ITERATION NUMBER 12

2	6	6	.28920777E+02	.11602610E+01
3	6	6	.45405671E+02	.20836100E+01
4	6	6	.54214692E+02	.27272980E+01
5	6	6	.58445213E+02	.30971260E+01
6	6	6	.59732212E+02	.32145240E+01

ITERATION NUMBER 13

2	6	6	.29969620E+02	.10488430E+01
3	6	6	.47296406E+02	.18907350E+01
4	6	6	.56699244E+02	.24845520E+01
5	6	6	.61274204E+02	.28289910E+01
6	6	6	.62670778E+02	.29385660E+01

ITERATION NUMBER 14

2	6	6	.30918556E+02	.94893600E+00
3	6	6	.49012283E+02	.17158770E+01
4	6	6	.58961106E+02	.22618620E+01
5	6	6	.63855123E+02	.25809190E+01
6	6	6	.65353378E+02	.26826000E+01

ITERATION NUMBER 15

2	6	6	.31777684E+02	.85912800E+00
3	6	6	.50569593E+02	.15573100E+01
4	6	6	.61019115E+02	.20580090E+01
5	6	6	.66207443E+02	.23523200E+01
6	6	6	.67799611E+02	.24462330E+01

ITERATION NUMBER 16

2	6	6	.32555933E+02	.77824900E+00
3	6	6	.51983078E+02	.14134850E+01
4	6	6	.62890829E+02	.18717140E+01
5	6	6	.68349745E+02	.21423020E+01
6	6	6	.70028350E+02	.22287390E+01

ITERATION NUMBER 17

2	6	6	.33261224E+02	.70529100E+00
3	6	6	.53266087E+02	.12830090E+01
4	6	6	.64592516E+02	.17016870E+01
5	6	6	.70299569E+02	.19498240E+01

```
6   6   6 .72057505E+02 .20291550E+01

  ITERATION NUMBER  18

2   6   6 .33900624E+02 .63940000E+00
3   6   6 .54430714E+02 .11646270E+01
4   6   6 .66139197E+02 .15466810E+01
5   6   6 .72073331E+02 .17737620E+01
6   6   6 .73903912E+02 .18464070E+01

  ITERATION NUMBER  19

2   6   6 .34480452E+02 .57982800E+00
3   6   6 .55487917E+02 .10572030E+01
4   6   6 .67544674E+02 .14054770E+01
5   6   6 .73686290E+02 .16129590E+01
6   6   6 .75583279E+02 .16793670E+01

  ITERATION NUMBER  20

2   6   6 .35006383E+02 .52593100E+00
3   6   6 .56447635E+02 .95971800E+00
4   6   6 .68821617E+02 .12769430E+01
5   6   6 .75152556E+02 .14662660E+01
6   6   6 .77110165E+02 .15268860E+01

  ITERATION NUMBER  21

2   6   6 .35483510E+02 .47712700E+00
3   6   6 .57318872E+02 .87123700E+00
4   6   6 .69981606E+02 .11599890E+01
5   6   6 .76485135E+02 .13325790E+01
6   6   6 .78498025E+02 .13878600E+01

  ITERATION NUMBER  22

2   6   6 .35916428E+02 .43291800E+00
3   6   6 .58109802E+02 .79093000E+00
4   6   6 .71035240E+02 .10536340E+01
5   6   6 .77695965E+02 .12108300E+01
6   6   6 .79759219E+02 .12611940E+01

  ITERATION NUMBER  23

2   6   6 .36309280E+02 .39285200E+00
3   6   6 .58827842E+02 .71804000E+00
```

```
4   6   6 .71992185E+02 .95694500E+00
5   6   6 .78795996E+02 .11000310E+01
6   6   6 .80905099E+02 .11458800E+01

  ITERATION NUMBER  24

2   6   6 .36665808E+02 .35652800E+00
3   6   6 .59479710E+02 .65186800E+00
4   6   6 .72861239E+02 .86905400E+00
5   6   6 .79795230E+02 .99923400E+00
6   6   6 .81946054E+02 .10409550E+01

  ITERATION NUMBER  25

2   6   6 .36989397E+02 .32358900E+00
3   6   6 .60071513E+02 .59180300E+00
4   6   6 .73650439E+02 .78920000E+00
5   6   6 .80702814E+02 .90758400E+00
6   6   6 .82891585E+02 .94553100E+00

  ITERATION NUMBER  26

2   6   6 .37283106E+02 .29370900E+00
3   6   6 .60608790E+02 .53727700E+00
4   6   6 .74367081E+02 .71664200E+00
5   6   6 .81527080E+02 .82426600E+00
6   6   6 .83750353E+02 .85876800E+00

  ITERATION NUMBER  27

2   6   6 .37549710E+02 .26660400E+00
3   6   6 .61096567E+02 .48777700E+00
4   6   6 .75017817E+02 .65073600E+00
5   6   6 .82275630E+02 .74855000E+00
6   6   6 .84530266E+02 .77991300E+00

  ITERATION NUMBER  28

2   6   6 .37791716E+02 .24200600E+00
3   6   6 .61539409E+02 .44284200E+00
4   6   6 .75608685E+02 .59086800E+00
5   6   6 .82955381E+02 .67975100E+00
6   6   6 .85238512E+02 .70824600E+00

  ITERATION NUMBER  29
```

```
2    6    6 .38011403E+02 .21968700E+00
3    6    6 .61941450E+02 .40204100E+00
4    6    6 .76145185E+02 .53650000E+00
5    6    6 .83572632E+02 .61725100E+00
6    6    6 .85881652E+02 .64314000E+00

  ITERATION NUMBER  30

2    6    6 .38210836E+02 .19943300E+00
3    6    6 .62306459E+02 .36500900E+00
4    6    6 .76632308E+02 .48712300E+00
5    6    6 .84133107E+02 .56047500E+00
6    6    6 .86465647E+02 .58399500E+00

  ITERATION NUMBER  31

2    6    6 .38391882E+02 .18104600E+00
3    6    6 .62637847E+02 .33138800E+00
4    6    6 .77074589E+02 .44228100E+00
5    6    6 .84642014E+02 .50890700E+00
6    6    6 .86995924E+02 .53027700E+00

  ITERATION NUMBER  32

2    6    6 .38556245E+02 .16436300E+00
3    6    6 .62938702E+02 .30085500E+00
4    6    6 .77476153E+02 .40156400E+00
5    6    6 .85104090E+02 .46207600E+00
6    6    6 .87477400E+02 .48147600E+00

  ITERATION NUMBER  33

2    6    6 .38705457E+02 .14921200E+00
3    6    6 .63211850E+02 .27314800E+00
4    6    6 .77840747E+02 .36459400E+00
5    6    6 .85523636E+02 .41954600E+00
6    6    6 .87914566E+02 .43716600E+00

  ITERATION NUMBER  34

2    6    6 .38840922E+02 .13546500E+00
3    6    6 .63459835E+02 .24798500E+00
4    6    6 .78171769E+02 .33102200E+00
5    6    6 .85904559E+02 .38092300E+00
6    6    6 .88311493E+02 .39692700E+00
```

ITERATION NUMBER 35

2	6	6	.38963905E+02	.12298300E+00
3	6	6	.63684976E+02	.22514100E+00
4	6	6	.78472308E+02	.30053900E+00
5	6	6	.86250416E+02	.34585700E+00
6	6	6	.88671880E+02	.36038700E+00

ITERATION NUMBER 36

2	6	6	.39075556E+02	.11165100E+00
3	6	6	.63889382E+02	.20440600E+00
4	6	6	.78745170E+02	.27286200E+00
5	6	6	.86564423E+02	.31400700E+00
6	6	6	.88999084E+02	.32720400E+00

ITERATION NUMBER 37

2	6	6	.39176923E+02	.10136700E+00
3	6	6	.64074960E+02	.18557800E+00
4	6	6	.78992909E+02	.24773900E+00
5	6	6	.86849521E+02	.28509800E+00
6	6	6	.89296165E+02	.29708100E+00

ITERATION NUMBER 38

2	6	6	.39268951E+02	.92028000E-01
3	6	6	.64243445E+02	.16848500E+00
4	6	6	.79217827E+02	.22491800E+00
5	6	6	.87108364E+02	.25884300E+00
6	6	6	.89565886E+02	.26972100E+00

ITERATION NUMBER 39

2	6	6	.39352499E+02	.83548000E-01
3	6	6	.64396410E+02	.15296500E+00
4	6	6	.79422030E+02	.20420300E+00
5	6	6	.87343369E+02	.23500500E+00
6	6	6	.89810770E+02	.24488400E+00

ITERATION NUMBER 40

2	6	6	.39428353E+02	.75854000E-01
3	6	6	.64535285E+02	.13887500E+00
4	6	6	.79607429E+02	.18539900E+00
5	6	6	.87556730E+02	.21336100E+00

```
6   6   6 .90033097E+02 .22232700E+00

  ITERATION NUMBER  41

2   6   6 .39497219E+02 .68866000E-01
3   6   6 .64661367E+02 .12608200E+00
4   6   6 .79775748E+02 .16831900E+00
5   6   6 .87750442E+02 .19371200E+00
6   6   6 .90234955E+02 .20185800E+00

  ITERATION NUMBER  42

2   6   6 .39559741E+02 .62522000E-01
3   6   6 .64775836E+02 .11446900E+00
4   6   6 .79928569E+02 .15282100E+00
5   6   6 .87926315E+02 .17587300E+00
6   6   6 .90418219E+02 .18326400E+00

  ITERATION NUMBER  43

2   6   6 .39616504E+02 .56763000E-01
3   6   6 .64879765E+02 .10392900E+00
4   6   6 .80067315E+02 .13874600E+00
5   6   6 .88085990E+02 .15967500E+00
6   6   6 .90584608E+02 .16638900E+00

  ITERATION NUMBER  44

2   6   6 .39668038E+02 .51534000E-01
3   6   6 .64974121E+02 .94356000E-01
4   6   6 .80193279E+02 .12596400E+00
5   6   6 .88230956E+02 .14496600E+00
6   6   6 .90735667E+02 .15105900E+00

  ITERATION NUMBER  45

2   6   6 .39714828E+02 .46790000E-01
3   6   6 .65059780E+02 .85659000E-01
4   6   6 .80307644E+02 .11436500E+00
5   6   6 .88362575E+02 .13161900E+00
6   6   6 .90872821E+02 .13715400E+00

  ITERATION NUMBER  46

2   6   6 .39757305E+02 .42477000E-01
3   6   6 .65137556E+02 .77776000E-01
```

```
4   6   6 .80411472E+02 .10382800E+00
5   6   6 .88482070E+02 .11949500E+00
6   6   6 .90997336E+02 .12451500E+00

 ITERATION NUMBER  47

2   6   6 .39795870E+02 .38565000E-01
3   6   6 .65208166E+02 .70610000E-01
4   6   6 .80505739E+02 .94267000E-01
5   6   6 .88590558E+02 .10848800E+00
6   6   6 .91110385E+02 .11304900E+00

 ITERATION NUMBER  48

2   6   6 .39830885E+02 .35015000E-01
3   6   6 .65272270E+02 .64104000E-01
4   6   6 .80591321E+02 .85582000E-01
5   6   6 .88689054E+02 .98496000E-01
6   6   6 .91213024E+02 .10263900E+00

 ITERATION NUMBER  49

2   6   6 .39862671E+02 .31786000E-01
3   6   6 .65330470E+02 .58200000E-01
4   6   6 .80669023E+02 .77702000E-01
5   6   6 .88778474E+02 .89420000E-01
6   6   6 .91306204E+02 .93180000E-01

 ITERATION NUMBER  50

2   6   6 .39891531E+02 .28860000E-01
3   6   6 .65383313E+02 .52843000E-01
4   6   6 .80739565E+02 .70542000E-01
5   6   6 .88859661E+02 .81187000E-01
6   6   6 .91390804E+02 .84600000E-01

 ITERATION NUMBER  51

2   6   6 .39917735E+02 .26204000E-01
3   6   6 .65431285E+02 .47972000E-01
4   6   6 .80803612E+02 .64047000E-01
5   6   6 .88933371E+02 .73710000E-01
6   6   6 .91467613E+02 .76809000E-01

 ITERATION NUMBER  52
```

```
2   6   6 .39941523E+02 .23788000E-01
3   6   6 .65474840E+02 .43555000E-01
4   6   6 .80861762E+02 .58150000E-01
5   6   6 .89000295E+02 .66924000E-01
6   6   6 .91537348E+02 .69735000E-01

  ITERATION NUMBER  53

2   6   6 .39963120E+02 .21597000E-01
3   6   6 .65514382E+02 .39542000E-01
4   6   6 .80914552E+02 .52790000E-01
5   6   6 .89061050E+02 .60755000E-01
6   6   6 .91600660E+02 .63312000E-01

  ITERATION NUMBER  54

2   6   6 .39982730E+02 .19610000E-01
3   6   6 .65550283E+02 .35901000E-01
4   6   6 .80962483E+02 .47931000E-01
5   6   6 .89116210E+02 .55160000E-01
6   6   6 .91658140E+02 .57480000E-01

  ITERATION NUMBER  55

2   6   6 .40000530E+02 .17800000E-01
3   6   6 .65582878E+02 .32595000E-01
4   6   6 .81005999E+02 .43516000E-01
5   6   6 .89166291E+02 .50081000E-01
6   6   6 .91710325E+02 .52185000E-01

  ITERATION NUMBER  56

2   6   6 .40016691E+02 .16161000E-01
3   6   6 .65612470E+02 .29592000E-01
4   6   6 .81045506E+02 .39507000E-01
5   6   6 .89211758E+02 .45467000E-01
6   6   6 .91757701E+02 .47376000E-01

  ITERATION NUMBER  57

2   6   6 .40031366E+02 .14675000E-01
3   6   6 .65639336E+02 .26866000E-01
4   6   6 .81081373E+02 .35867000E-01
5   6   6 .89253035E+02 .41277000E-01
6   6   6 .91800718E+02 .43017000E-01
```

ITERATION NUMBER 58

2	6	6	.40044689E+02	.13323000E-01
3	6	6	.65663728E+02	.24392000E-01
4	6	6	.81113936E+02	.32563000E-01
5	6	6	.89290513E+02	.37478000E-01
6	6	6	.91839766E+02	.39048000E-01

ITERATION NUMBER 59

2	6	6	.40056783E+02	.12094000E-01
3	6	6	.65685872E+02	.22144000E-01
4	6	6	.81143500E+02	.29564000E-01
5	6	6	.89324538E+02	.34025000E-01
6	6	6	.91875226E+02	.35460000E-01

ITERATION NUMBER 60

2	6	6	.40067765E+02	.10982000E-01
3	6	6	.65705979E+02	.20107000E-01
4	6	6	.81170341E+02	.26841000E-01
5	6	6	.89355427E+02	.30889000E-01
6	6	6	.91907413E+02	.32187000E-01

ITERATION NUMBER 61

2	6	6	.40077734E+02	.99690000E-02
3	6	6	.65724233E+02	.18254000E-01
4	6	6	.81194713E+02	.24372000E-01
5	6	6	.89383472E+02	.28045000E-01
6	6	6	.91936636E+02	.29223000E-01

ITERATION NUMBER 62

2	6	6	.40086786E+02	.90520000E-02
3	6	6	.65740805E+02	.16572000E-01
4	6	6	.81216833E+02	.22120000E-01
5	6	6	.89408934E+02	.25462000E-01
6	6	6	.91963168E+02	.26532000E-01

ITERATION NUMBER 63

2	6	6	.40095003E+02	.82170000E-02
3	6	6	.65755850E+02	.15045000E-01
4	6	6	.81236921E+02	.20088000E-01
5	6	6	.89432052E+02	.23118000E-01

```
6   6   6 .91987258E+02 .24090000E-01

  ITERATION NUMBER  64

2   6   6 .40102464E+02 .74610000E-02
3   6   6 .65769510E+02 .13660000E-01
4   6   6 .81255156E+02 .18235000E-01
5   6   6 .89453039E+02 .20987000E-01
6   6   6 .92009131E+02 .21873000E-01

  ITERATION NUMBER  65

2   6   6 .40109237E+02 .67730000E-02
3   6   6 .65781911E+02 .12401000E-01
4   6   6 .81271717E+02 .16561000E-01
5   6   6 .89472095E+02 .19056000E-01
6   6   6 .92028985E+02 .19854000E-01

  ITERATION NUMBER  66

2   6   6 .40115387E+02 .61500000E-02
3   6   6 .65793169E+02 .11258000E-01
4   6   6 .81286745E+02 .15028000E-01
5   6   6 .89489396E+02 .17301000E-01
6   6   6 .92047015E+02 .18030000E-01

  ITERATION NUMBER  67

2   6   6 .40120970E+02 .55830000E-02
3   6   6 .65803392E+02 .10223000E-01
4   6   6 .81300392E+02 .13647000E-01
5   6   6 .89505100E+02 .15704000E-01
6   6   6 .92063380E+02 .16365000E-01

  ITERATION NUMBER  68

2   6   6 .40126038E+02 .50680000E-02
3   6   6 .65812672E+02 .92800000E-02
4   6   6 .81312784E+02 .12392000E-01
5   6   6 .89519360E+02 .14260000E-01
6   6   6 .92078239E+02 .14859000E-01

  ITERATION NUMBER  69

2   6   6 .40130640E+02 .46020000E-02
3   6   6 .65821096E+02 .84240000E-02
```

```
4   6   6 .81324033E+02 .11249000E-01
5   6   6 .89532308E+02 .12948000E-01
6   6   6 .92091727E+02 .13488000E-01

  ITERATION NUMBER  70

2   6   6 .40134818E+02 .41780000E-02
3   6   6 .65828749E+02 .76530000E-02
4   6   6 .81334246E+02 .10213000E-01
5   6   6 .89544059E+02 .11751000E-01
6   6   6 .92103976E+02 .12249000E-01

  ITERATION NUMBER  71

2   6   6 .40138611E+02 .37930000E-02
3   6   6 .65835693E+02 .69440000E-02
4   6   6 .81343517E+02 .92710000E-02
5   6   6 .89554730E+02 .10671000E-01
6   6   6 .92115094E+02 .11118000E-01

  ITERATION NUMBER  72

2   6   6 .40142055E+02 .34440000E-02
3   6   6 .65841997E+02 .63040000E-02
4   6   6 .81351936E+02 .84190000E-02
5   6   6 .89564420E+02 .96900000E-02
6   6   6 .92125192E+02 .10098000E-01

  ITERATION NUMBER  73

2   6   6 .40145183E+02 .31280000E-02
3   6   6 .65847723E+02 .57260000E-02
4   6   6 .81359579E+02 .76430000E-02
5   6   6 .89573216E+02 .87960000E-02
6   6   6 .92134357E+02 .91650000E-02

  ITERATION NUMBER  74

2   6   6 .40148021E+02 .28380000E-02
3   6   6 .65852923E+02 .52000000E-02
4   6   6 .81366517E+02 .69380000E-02
5   6   6 .89581199E+02 .79830000E-02
6   6   6 .92142679E+02 .83220000E-02

  ITERATION NUMBER  75
```

```
2   6   6 .40150599E+02 .25780000E-02
3   6   6 .65857640E+02 .47170000E-02
4   6   6 .81372816E+02 .62990000E-02
5   6   6 .89588450E+02 .72510000E-02
6   6   6 .92150230E+02 .75510000E-02

  ITERATION NUMBER  76

2   6   6 .40152938E+02 .23390000E-02
3   6   6 .65861925E+02 .42850000E-02
4   6   6 .81378536E+02 .57200000E-02
5   6   6 .89595029E+02 .65790000E-02
6   6   6 .92157088E+02 .68580000E-02

  ITERATION NUMBER  77

2   6   6 .40155063E+02 .21250000E-02
3   6   6 .65865812E+02 .38870000E-02
4   6   6 .81383728E+02 .51920000E-02
5   6   6 .89601005E+02 .59760000E-02
6   6   6 .92163316E+02 .62280000E-02

  ITERATION NUMBER  78

2   6   6 .40156991E+02 .19280000E-02
3   6   6 .65869343E+02 .35310000E-02
4   6   6 .81388442E+02 .47140000E-02
5   6   6 .89606429E+02 .54240000E-02
6   6   6 .92168968E+02 .56520000E-02

  ITERATION NUMBER  79

2   6   6 .40158743E+02 .17520000E-02
3   6   6 .65872551E+02 .32080000E-02
4   6   6 .81392722E+02 .42800000E-02
5   6   6 .89611359E+02 .49290000E-02
6   6   6 .92174101E+02 .51330000E-02

  ITERATION NUMBER  80

2   6   6 .40160331E+02 .15880000E-02
3   6   6 .65875463E+02 .29120000E-02
4   6   6 .81396610E+02 .38880000E-02
5   6   6 .89615831E+02 .44730000E-02
6   6   6 .92178763E+02 .46620000E-02
```

FINAL RESULTS OF ITERATION

	node		temperature	residual
2	2	2	.11335533E+02	.18500000E-03
3	2	2	.16193666E+02	.33900000E-03
4	2	2	.18492290E+02	.45300000E-03
5	2	2	.19555513E+02	.52000000E-03
6	2	2	.19867755E+02	.54300000E-03
2	3	2	.16193666E+02	.33900000E-03
3	3	2	.24072778E+02	.61800000E-03
4	3	2	.28065678E+02	.82800000E-03
5	3	2	.29976634E+02	.95200000E-03
6	3	2	.30545704E+02	.99300000E-03
2	4	2	.18492290E+02	.45300000E-03
3	4	2	.28065678E+02	.82800000E-03
4	4	2	.33129558E+02	.11070000E-02
5	4	2	.35615056E+02	.12730000E-02
6	4	2	.36363747E+02	.13260000E-02
2	5	2	.19555513E+02	.52000000E-03
3	5	2	.29976634E+02	.95200000E-03
4	5	2	.35615056E+02	.12730000E-02
5	5	2	.38423278E+02	.14650000E-02
6	5	2	.39275131E+02	.15240000E-02
2	6	2	.19867755E+02	.54300000E-03
3	6	2	.30545704E+02	.99300000E-03
4	6	2	.36363747E+02	.13260000E-02
5	6	2	.39275131E+02	.15240000E-02
6	6	2	.40160331E+02	.15880000E-02
2	2	3	.16193666E+02	.33900000E-03
3	2	3	.24072778E+02	.61800000E-03
4	2	3	.28065678E+02	.82800000E-03
5	2	3	.29976634E+02	.95200000E-03
6	2	3	.30545704E+02	.99300000E-03
2	3	3	.24072779E+02	.61900000E-03
3	3	3	.37124431E+02	.11340000E-02
4	3	3	.44129680E+02	.15150000E-02
5	3	3	.47580711E+02	.17430000E-02
6	3	3	.48620848E+02	.18170000E-02
2	4	3	.28065678E+02	.82800000E-03
3	4	3	.44129680E+02	.15150000E-02
4	4	3	.53076940E+02	.20240000E-02
5	4	3	.57583003E+02	.23290000E-02
6	4	3	.58954961E+02	.24250000E-02
2	5	3	.29976634E+02	.95200000E-03
3	5	3	.47580711E+02	.17430000E-02

4	5	3	.57583003E+02	.23290000E-02
5	5	3	.62686205E+02	.26780000E-02
6	5	3	.64249778E+02	.27950000E-02
2	6	3	.30545704E+02	.99300000E-03
3	6	3	.48620848E+02	.18170000E-02
4	6	3	.58954961E+02	.24250000E-02
5	6	3	.64249778E+02	.27950000E-02
6	6	3	.65875463E+02	.29120000E-02
2	2	4	.18492289E+02	.45200000E-03
3	2	4	.28065678E+02	.82800000E-03
4	2	4	.33129558E+02	.11070000E-02
5	2	4	.35615056E+02	.12730000E-02
6	2	4	.36363747E+02	.13260000E-02
2	3	4	.28065678E+02	.82800000E-03
3	3	4	.44129680E+02	.15150000E-02
4	3	4	.53076940E+02	.20240000E-02
5	3	4	.57583003E+02	.23290000E-02
6	3	4	.58954961E+02	.24250000E-02
2	4	4	.33129558E+02	.11070000E-02
3	4	4	.53076940E+02	.20240000E-02
4	4	4	.64567954E+02	.27030000E-02
5	4	4	.70471552E+02	.31110000E-02
6	4	4	.72285538E+02	.32390000E-02
2	5	4	.35615056E+02	.12730000E-02
3	5	4	.57583003E+02	.23290000E-02
4	5	4	.70471552E+02	.31110000E-02
5	5	4	.77171912E+02	.35810000E-02
6	5	4	.79242543E+02	.37310000E-02
2	6	4	.36363747E+02	.13260000E-02
3	6	4	.58954961E+02	.24250000E-02
4	6	4	.72285538E+02	.32390000E-02
5	6	4	.79242543E+02	.37310000E-02
6	6	4	.81396610E+02	.38880000E-02
2	2	5	.19555513E+02	.52100000E-03
3	2	5	.29976634E+02	.95300000E-03
4	2	5	.35615056E+02	.12730000E-02
5	2	5	.38423278E+02	.14650000E-02
6	2	5	.39275131E+02	.15240000E-02
2	3	5	.29976634E+02	.95300000E-03
3	3	5	.47580711E+02	.17430000E-02
4	3	5	.57583003E+02	.23290000E-02
5	3	5	.62686205E+02	.26780000E-02
6	3	5	.64249778E+02	.27950000E-02
2	4	5	.35615056E+02	.12730000E-02
3	4	5	.57583003E+02	.23290000E-02
4	4	5	.70471552E+02	.31110000E-02

```
 5   4   5 .77171912E+02 .35810000E-02
 6   4   5 .79242543E+02 .37310000E-02
 2   5   5 .38423278E+02 .14650000E-02
 3   5   5 .62686205E+02 .26780000E-02
 4   5   5 .77171912E+02 .35810000E-02
 5   5   5 .84787288E+02 .41220000E-02
 6   5   5 .87153425E+02 .42930000E-02
 2   6   5 .39275131E+02 .15240000E-02
 3   6   5 .64249778E+02 .27950000E-02
 4   6   5 .79242543E+02 .37310000E-02
 5   6   5 .87153425E+02 .42930000E-02
 6   6   5 .89615831E+02 .44730000E-02
 2   2   6 .19867755E+02 .54300000E-03
 3   2   6 .30545704E+02 .99300000E-03
 4   2   6 .36363747E+02 .13260000E-02
 5   2   6 .39275131E+02 .15240000E-02
 6   2   6 .40160331E+02 .15880000E-02
 2   3   6 .30545704E+02 .99300000E-03
 3   3   6 .48620848E+02 .18170000E-02
 4   3   6 .58954961E+02 .24250000E-02
 5   3   6 .64249778E+02 .27950000E-02
 6   3   6 .65875463E+02 .29120000E-02
 2   4   6 .36363747E+02 .13260000E-02
 3   4   6 .58954961E+02 .24250000E-02
 4   4   6 .72285538E+02 .32390000E-02
 5   4   6 .79242543E+02 .37310000E-02
 6   4   6 .81396610E+02 .38880000E-02
 2   5   6 .39275131E+02 .15240000E-02
 3   5   6 .64249778E+02 .27950000E-02
 4   5   6 .79242543E+02 .37310000E-02
 5   5   6 .87153425E+02 .42930000E-02
 6   5   6 .89615831E+02 .44730000E-02
 2   6   6 .40160331E+02 .15880000E-02
 3   6   6 .65875463E+02 .29120000E-02
 4   6   6 .81396610E+02 .38880000E-02
 5   6   6 .89615831E+02 .44730000E-02
 6   6   6 .92178763E+02 .46620000E-02
DIFFUSIVITY    1.0000      CONDUCTIVITY    1.0000
DELTA TIME      .2000      SOURCE STRENGTH 2000.0000

XGRID=     .1000      YGRID=     .1000      ZGRID=     .1000
MAX TIME    .2000STOP WHEN ABSDIFF- .50000000E-02 IS ZERO OR LESS

OUTPUT CONTROLS ARE IRITE  1  JRITE  1
KRITE  1  KITER  6  JITER  6  CHOSE 0.000
```

GA/Phys/63-8

DATA R UN REGULAR WEGSTEIN -- T = 0.2 SEC, GRIDS EQUAL AT 0.1 CM

NEW TI ME INCREMENT .200

ITERATIØN NUMBER 1

	NØDE		TEMPERATURE	RESIDUAL	SLØPE
2	6	6	.49352900E+01	.49352900E+01	
3	6	6	.61509471E+01	.61509471E+01	
4	6	6	.64493826E+01	.64493826E+01	
5	6	6	.65222818E+01	.65222818E+01	
6	6	6	.65399740E+01	.65399740E+01	

ITERATIØN NUMBER 2

2	6	6	.88801862E+01	.39448962E+01	
3	6	6	.11733575E+02	.55826280E+01	
4	6	6	.12595459E+02	.61460770E+01	
5	6	6	.12844798E+02	.63225170E+01	
6	6	6	.12909294E+02	.63693200E+01	

ITERATIØN NUMBER 3

2	6	6	.12185228E+02	.33050420E+01	
3	6	6	.16783346E+02	.50497710E+01	
4	6	6	.18385662E+02	.57902030E+01	
5	6	6	.18910285E+02	.60654870E+01	
6	6	6	.19052105E+02	.61428110E+01	

ITERATIØN NUMBER 4

2	6	6	.15032849E+02	.28476210E+01	
3	6	6	.21351303E+02	.45679570E+01	
4	6	6	.23794226E+02	.54085640E+01	
5	6	6	.24671974E+02	.57616890E+01	
6	6	6	.24916819E+02	.58647140E+01	

ITERATIØN NUMBER 5

2	6	6	.17527906E+02	.24950570E+01	
3	6	6	.25486291E+02	.41349880E+01	
4	6	6	.28814165E+02	.50199390E+01	
5	6	6	.30098025E+02	.54260510E+01	
6	6	6	.30464459E+02	.55476400E+01	

ITERATIØN NUMBER 6

2	6	6	.19736819E+02	.22089130E+01	
3	6	6	.29231800E+02	.37455090E+01	
4	6	6	.33450971E+02	.46368060E+01	
5	6	6	.35171236E+02	.507321 0E+01	
6	6	6	.35670706E+02	.52062470E+01	

ITERATION NUMBER 7

2	6	6	.21705484E+02	.19686650E+01
3	6	6	.32626222E+02	.33944220E+01
4	6	6	.37718101E+02	.42671300E+01
5	6	6	.39886695E+02	.47154590E+01
6	6	6	.40524592E+02	.48538860E+01

ITERATION NUMBER 8

2	6	6	.23467933E+02	.17624490E+01
3	6	6	.35703599E+02	.30773770E+01
4	6	6	.41633842E+02	.39157410E+01
5	6	6	.44248880E+02	.43621850E+01
6	6	6	.45025903E+02	.450131 0E+01

ITERATION NUMBER 9

2	6	6	.25050775E+02	.15828420E+01
3	6	6	.38494323E+02	.27907240E+01
4	6	6	.45219152E+02	.35853100E+01
5	6	6	.48268970E+02	.40200900E+01
6	6	6	.49182403E+02	.41565000E+01

ITERATION NUMBER 10

2	6	6	.26475626E+02	.14248510E+01
3	6	6	.41025641E+02	.25313180E+01
4	6	6	.48496204E+02	.32770520E+01
5	6	6	.51962594E+02	.36936240E+01
6	6	6	.53007382E+02	.38249790E+01

ITERATION NUMBER 11

2	6	6	.27760516E+02	.12848900E+01
3	6	6	.43322061E+02	.22964200E+01
4	6	6	.51487394E+02	.2991 900E+01
5	6	6	.55348087E+02	.33854930E+01
6	6	6	.56517688E+02	.35103060E+01

ITERATION NUMBER 12

ITERATION NUMBER 13

2	6	6	.44852152E+02	.101 0451E+02
3	6	6	.86393980E+02	.23417471E+02
4	6	6	.78428443E+02	.15310434E+02
5	6	6	.73398767E+02	.10540817E+02
6	6	6	.70871016E+02	.89507020E+01

ITERATION NUMBER 14

2	6	6	.51368571E+02	.65164190E+01
3	6	6	.85345189E+02	-.10487910E+01
4	6	6	.83703602E+02	.52751590E+01
5	6	6	.80649055E+02	.72502880E+01
6	6	6	.79693043E+02	.88220270E+01

ITERATION NUMBER 15

2	6	6	.51339767E+02	-.28804000E-01
3	6	6	.77532625E+02	-.78125640E+01
4	6	6	.84251769E+02	.54816700E+00
5	6	6	.85209332E+02	.45602770E+01
6	6	6	.85549536E+02	.58564930E+01

ITERATION NUMBER 16

2	6	6	.47631880E+02	-.37078870E+01
3	6	6	.74328679E+02	-.32039460E+01
4	6	6	.84750529E+02	.49876000E+00
5	6	6	.88309144E+02	.30998120E+01
6	6	6	.89347939E+02	.37984030E+01

ITERATION NUMBER 17

2	6	6	.45810482E+02	-.18213980E+01
3	6	6	.72268586E+02	-.20600930E+01
4	6	6	.84915766E+02	.16523700E+00
5	6	6	.90204296E+02	.18951520E+01
6	6	6	.91824796E+02	.24768570E+01

ITERATION NUMBER 18

3	6	6	-.1341 200E+03	-.20638058E+03 .99188337E+00

ITERATION NUMBER 19

2	6	6	.85797480E+01	-.35130178E+02
3	6	6	.46369107E+02	.180481 0E+03
4	6	6	.75833670E+02	.25041713E+02
5	6	6	.88148656E+02	.24035820E+01
6	6	6	.89534468E+02	-.10191980E+01

ITERATION NUMBER 20

2	6	6	.34557193E+02	.25977445E+02
3	6	6	.51224439E+02	.48553320E+01
4	6	6	.74969384E+02	-.86428600E+00
5	6	6	.86551262E+02	-.15973940E+01
6	6	6	.89934059E+02	.39959100E+00

ITERATION NUMBER 21

2	6	6	.33674101E+02	-.88309200E+00
3	6	6	.59306274E+02	.80818350E+01
4	6	6	.77646077E+02	.26766930E+01
5	6	6	.87125141E+02	.57387900E+00
6	6	6	.89426491E+02	-.50756800E+00

ITERATION NUMBER 22

2	6	6	.37124091E+02	.34499900E+01
3	6	6	.61684080E+02	.23778060E+01
4	6	6	.78234057E+02	.58798000E+00
5	6	6	.86822557E+02	-.30258400E+00
6	6	6	.89560956E+02	.13446500E+00

ITERATION NUMBER 23

2	6	6	.37850149E+02	.72605800E+00
3	6	6	.62959550E+02	.12754700E+01
4	6	6	.78700857E+02	.46680000E+00
5	6	6	.87051002E+02	.22844500E+00
6	6	6	.89524159E+02	-.36797000E-01

ITERATION NUMBER 24

3	6	6	.67402930E+02	.44433800E+01	.77697084E+00

ITERATION NUMBER 25

2	6	6	.39712428E+02	.10285740E+01
3	6	6	.64827160E+02	-.25757700E+01
4	6	6	.79679853E+02	-.87063000E-01
5	6	6	.87572706E+02	.24242500E+00
6	6	6	.90034450E+02	.25863100E+00

ITERATION NUMBER 26

2	6	6	.39474969E+02	-.23745900E+00
3	6	6	.64796037E+02	-.31 23000E-01
4	6	6	.79854291E+02	.17443800E+00
5	6	6	.87786705E+02	.21399900E+00
6	6	6	.90260794E+02	.22634400E+00

ITERATION NUMBER 27

2	6	6	.39567337E+02	.92368000E-01
3	6	6	.64803561E+02	.75240000E-02
4	6	6	.79942841E+02	.88550000E-01
5	6	6	.87940229E+02	.15352400E+00

```
 6   6   6 .90443049E+02 .18225500E+00

          ITERATION NUMBER  28

 2   6   6 .39596002E+02 .28665000E-01
 3   6   6 .64865914E+02 .62353000E-01
 4   6   6 .80058366E+02 .1 552500E+00
 5   6   6 .88090688E+02 .15045900E+00
 6   6   6 .90593803E+02 .15075400E+00

          ITERATION NUMBER  29

 2   6   6 .39644080E+02 .48078000E-01
 3   6   6 .64943841E+02 .77927000E-01
 4   6   6 .80170791E+02 .1 242500E+00
 5   6   6 .88221417E+02 .13072900E+00
 6   6   6 .90733244E+02 .13944100E+00

          ITERATION NUMBER  30

 6   6   6 .10174910E+03 .11015860E+02 .98750008E+00

          ITERATION NUMBER  31

 2   6   6 .40534931E+02 .49760100E+00
 3   6   6 .65530987E+02 .33360400E+00
 4   6   6 .80543187E+02 .20694300E+00
 5   6   6 .90331534E+02 .19631770E+01
 6   6   6 .91917293E+02-.98318100E+01

          ITERATION NUMBER  32

 2   6   6 .40449083E+02-.85848000E-01
 3   6   6 .65608256E+02 .77269000E-01
 4   6   6 .80981496E+02 .43830900E+00
 5   6   6 .89076081E+02-.12554530E+01
 6   6   6 .92268235E+02 .35094200E+00

          ITERATION NUMBER  33

 2   6   6 .40187156E+02-.26192700E+00
 3   6   6 .65619279E+02 .11023000E-01
 4   6   6 .80895394E+02-.86102000E-01
 5   6   6 .89281976E+02 .20589500E+00
 6   6   6 .91747795E+02-.52044000E+00

          ITERATION NUMBER  34

 2   6   6 .40113039E+02-.74117000E-01
 3   6   6 .65607579E+02-.1 700000E-01
 4   6   6 .81012387E+02 .1 699300E+00
 5   6   6 .89179240E+02-.10273600E+00
 6   6   6 .91798949E+02 .51 54000E-01
```

ITERATION NUMBER 35

2	6	6	.40069229E+02	-.43810000E-01
3	6	6	.65632031E+02	.24452000E-01
4	6	6	.81031 34E+02	.18747000E-01
5	6	6	.89235540E+02	.56300000E-01
6	6	6	.91775925E+02	-.23024000E-01

ITERATION NUMBER 36

2	6	6	.40047945E+02	-.21284000E-01	.32695731E+00
3	6	6	.65659270E+02	.27239000E-01	.52699165E+00
5	6	6	.89292120E+02	.56580000E-01	.50126110E+00

ITERATION NUMBER . 37

2	6	6	.40055292E+02	.73470000E-02
3	6	6	.65676297E+02	.17027000E-01
4	6	6	.81134991E+02	.41078000E-01
5	6	6	.89314418E+02	.22298000E-01
6	6	6	.91870999E+02	.39107000E-01

ITERATION NUMBER 38

2	6	6	.40065759E+02	.10467000E-01
3	6	6	.65700144E+02	.23847000E-01
4	6	6	.81160295E+02	.25304000E-01
5	6	6	.89347487E+02	.33069000E-01
6	6	6	.91898458E+02	.27459000E-01

ITERATION NUMBER 39

2	6	6	.40075656E+02	.98970000E-02
3	6	6	.65718252E+02	.18108000E-01
4	6	6	.81186919E+02	.26624000E-01
5	6	6	.89374843E+02	.27356000E-01
6	6	6	.91928420E+02	.29962000E-01

ITERATION NUMBER 40

2	6	6	.40084553E+02	.88970000E-02
3	6	6	.65735925E+02	.17673000E-01
4	6	6	.81209764E+02	.22845000E-01
5	6	6	.89401222E+02	.26379000E-01
6	6	6	.91955065E+02	.26645000E-01

ITERATION NUMBER 41

2	6	6	.40092966E+02	.84130000E-02
3	6	6	.65751456E+02	.15531000E-01
4	6	6	.81230797E+02	.21033000E-01
5	6	6	.89424951E+02	.23729000E-01
6	6	6	.91979914E+02	.24849000E-01

```
                ITERATION NUMBER  42

 2   6   6 .40664700E+02 .57173400E+00 .98549863E+00

                ITERATION NUMBER  43

 2   6   6 .40155975E+02-.50872500E+00
 3   6   6 .65941433E+02 .81778000E-01
 4   6   6 .81918920E+02 .23045200E+00
 5   6   6 .90107529E+02 .32909000E+00
 6   6   6 .92504389E+02 .33788200E+00

                ITERATION NUMBER  44

 2   6   6 .40192710E+02 .36735000E-01
 3   6   6 .66048172E+02 .10673900E+00
 4   6   6 .81743366E+02-.17555400E+00
 5   6   6 .90085776E+02-.21753000E-01
 6   6   6 .92657030E+02 .15264100E+00

                ITERATION NUMBER  45

 2   6   6 .40202726E+02 .10016000E-01
 3   6   6 .66018333E+02-.29839000E-01
 4   6   6 .81726789E+02-.16577000E-01
 5   6   6 .90038601E+02-.47175000E-01
 6   6   6 .92623301E+02-.33729000E-01

                ITERATION NUMBER  46

 2   6   6 .40206900E+02 .41740000E-02
 3   6   6 .66031618E+02 .13285000E-01
 4   6   6 .81677748E+02-.49041000E-01
 5   6   6 .89980809E+02-.57792000E-01
 6   6   6 .92571394E+02-.51907000E-01

                ITERATION NUMBER  47

 2   6   6 .40216700E+02 .98000000E-02
 3   6   6 .66020183E+02-.11435000E-01
 4   6   6 .81643890E+02-.33858000E-01
 5   6   6 .89934384E+02-.46425000E-01
 6   6   6 .92519805E+02-.51589000E-01

                ITERATION NUMBER  48

 2   6   6 .40218714E+02 .20140000E-02 .17061224E+00
 3   6   6 .65997780E+02-.22403000E-01 .66200262E+00
```

ITERATION NUMBER 49

2	6	6	.40217137E+02-.15770000E-02
3	6	6	.65992337E+02-.54430000E-02
4	6	6	.81535184E+02-.58734000E-01
5	6	6	.89807301E+02-.66910000E-01
6	6	6	.92404153E+02-.62883000E-01

ITERATION NUMBER 50

2	6	6	.40214339E+02-.27980000E-02
3	6	6	.65964920E+02-.27417000E-01
4	6	6	.81505052E+02-.30132000E-01
5	6	6	.89769072E+02-.38229000E-01
6	6	6	.92352080E+02-.52073000E-01

ITERATION NUMBER 51

2	6	6	.40205378E+02-.89610000E-02
3	6	6	.65949978E+02-.14942000E-01
4	6	6	.81500347E+02-.47050000E-02
5	6	6	.89751201E+02-.17871000E-01
6	6	6	.92324220E+02-.27860000E-01

ITERATION NUMBER 52

2	6	6	.40199461E+02-.59170000E-02
3	6	6	.65945630E+02-.43480000E-02
4	6	6	.81494320E+02-.60270000E-02
5	6	6	.89736533E+02-.14668000E-01
6	6	6	.92308033E+02-.16187000E-01

ITERATION NUMBER 53

2	6	6	.40196567E+02-.28940000E-02
3	6	6	.65941553E+02-.40770000E-02
4	6	6	.81488275E+02-.60450000E-02
5	6	6	.89726712E+02-.98210000E-02
6	6	6	.92295861E+02-.12172000E-01

ITERATION NUMBER 54

4	6	6	.81432630E+02-.55645000E-01 .90190239E+00

ITERATION NUMBER 55

2	6	6	.40189914E+02-.23360000E-02
3	6	6	.65925560E+02-.1 8410000E-01
4	6	6	.81468953E+02 .36323000E-01
5	6	6	.89694835E+02-.56250000E-02
6	6	6	.92262128E+02-.15831000E-01

```
                ITERATIØN NUMBER  56

     2    6    6 .40188478E+02-.14360000E-02
     3    6    6 .65928388E+02 .28280000E-02
     4    6    6 .81463657E+02-.52960000E-02
     5    6    6 .89693365E+02-.14700000E-02
     6    6    6 .92258601E+02-.35270000E-02

                ITERATIØN NUMBER  57

     2    6    6 .40187951E+02-.52700000E-03
     3    6    6 .65925081E+02-.33070000E-02
     4    6    6 .81462624E+02-.10330000E-02
     5    6    6 .89690499E+02-.28660000E-02
     6    6    6 .92256453E+02-.21480000E-02

                ITERATIØN NUMBER  58

     2    6    6 .40186625E+02-.13260000E-02
     3    6    6 .65923581E+02-.15000000E-02
     4    6    6 .81459925E+02-.26990000E-02
     5    6    6 .89688134E+02-.23650000E-02
     6    6    6 .92253860E+02-.25930000E-02

          FINAL RESULTS ØF ITERATIØN

          2    2    2 .1 338654E+02
          4    2    2 .18499369E+02
          6    2    2 .19876607E+02
          2    6    2 .19876652E+02
          4    6    2 .36385764E+02
          6    6    2 .40186622E+02
          2    2    6 .19876605E+02
          4    2    6 .36385683E+02
          6    2    6 .40186572E+02
          2    6    6 .40186625E+02
          4    6    6 .81459925E+02
          6    6    6 .92253860E+02
DIFFUS IVITY     1.0000      CØNDUCTIVITY     1.0000
DELTA  TIME       .2000       SØURCE STRENGTH 2000.0000

XGRID       .100X       YGRID=       .1000       ZGRID=       .1000
MAX TI ME      .2000STØP WHEN ABSDIFF-     .005001S ZERØ ØR LESS

ØUTPUT  CØNTRØLS ARE IRITE  2  JRITE  4
KRITE   4  KITER  6  JITER  6
```

Vita

Robert Theodore Poppe was born on 17 February 1930 in Covington, Kentucky, the son of Ruth A. Poppe and the late Raymond H. Poppe. After completing his work in 1948 at Newport High School, Newport, Kentucky, he did undergraduate work at the University of Cincinnati, Cincinnati, Ohio and Morehead State College, Morehead, Kentucky until January 1951 when he enlisted in the U.S. Air Force. He was discharged in June of that same year to enter the U. S. Naval Academy at Annapolis, Maryland. Capt. Poppe graduated in June 1955 with a degree of Bachelor of Science and was commissioned a Second Lieutenant in the U. S. Air Force. After graduation, he attended the Ground Electronics School at Keesler Air Force Base, Mississippi. During the period from 1956 till 1961, he was assigned to maintenance activities in ground radar and classified work in the field of electronics. In August of 1961 he entered the two year graduate program in Astronautics at the Air Force Institute of Technology Resident School at Wright-Patterson A.F.B., Ohio.