

AD-A079 534

SOUTHEASTERN MASSACHUSETTS UNIV NORTH DARTMOUTH

F/6 20/13

A COMPUTATIONAL METHOD FOR SOLVING PARABOLIC PARTIAL DIFFERENTI--ETC(U)

NOV 79 T B SHEN

AFOSR-78-3538

UNCLASSIFIED

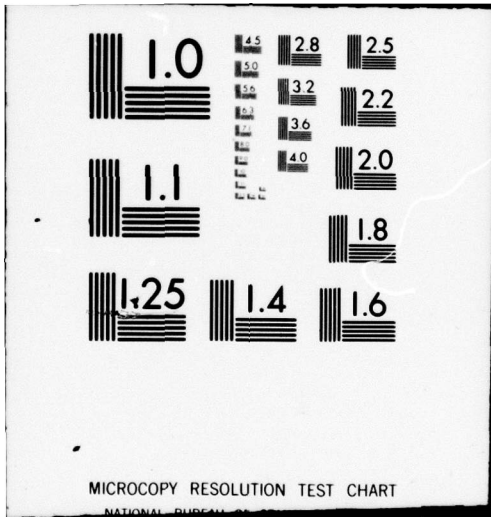
AFOSR-TR-80-0017

NL

| OF |
AD A
079534



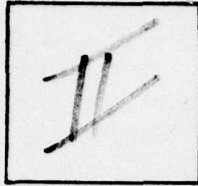
END
DATE
FILMED
3-80
DDC



PHOTOGRAPH THIS SHEET

ADA 079534

DTIC ACCESSION NUMBER



LEVEL



INVENTORY

AFOSR-TR-80-0017

DOCUMENT IDENTIFICATION

Grant AFOSR-78-3538

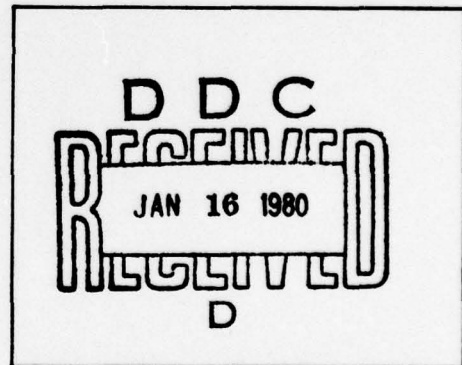
DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

| | |
|--------------------|---|
| ACCESSION FOR | |
| NTIS | GRA&I <input checked="" type="checkbox"/> |
| DTIC | TAB <input type="checkbox"/> |
| UNANNOUNCED | <input type="checkbox"/> |
| JUSTIFICATION | |
| | |
| | |
| BY | |
| DISTRIBUTION / | |
| AVAILABILITY CODES | |
| DIST | AVAIL AND/OR SPECIAL |
| A | |

DISTRIBUTION STAMP



DATE ACCESSIONED

80 1 14 040

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

AFOSR-TR- 80 - 0017

ADA079534

**A COMPUTATIONAL METHOD FOR SOLVING PARABOLIC
PARTIAL DIFFERENTIAL EQUATIONS**

by

**Thomas B. C. Shen
Associate Professor of Mechanical Engineering
Southeastern Massachusetts University**

November 1979

Final Report

prepared for

**Air Force Office of Scientific Research
under
Grant AFOSR - 78 - 3538**

**Approved for public release;
distribution unlimited.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|--|---|
| 1. REPORT NUMBER AFOSR-TR- 80 - 0017 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) A COMPUTATIONAL METHOD FOR SOLVING PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS | 5. TYPE OF REPORT & PERIOD COVERED <u>FINAL</u> 01 JAN 78 - 31 JUL 79 | |
| | 6. PERFORMING ORG. REPORT NUMBER | |
| 7. AUTHOR(s) THOMAS B. C. SHEN | 8. CONTRACT OR GRANT NUMBER(s) AFOSR-78-3538 | |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS SOUTHEASTERN MASSACHUSETTS UNIVERSITY NORTH DARTMOUTH, MA 02747 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2308D9 61102F | |
| 11. CONTROLLING OFFICE NAME AND ADDRESS AIR FORCE OFFICE OF SCIENTIFIC RESEARCH/NA BLDG 410 BOLLING AIR FORCE BASE DC 20332 | 12. REPORT DATE NOV 79 | |
| | 13. NUMBER OF PAGES 34 | |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | 15. SECURITY CLASS. (of this report) UNCLASSIFIED | |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) NEW COMPUTATIONAL TECHNIQUES; RECURRENCE TECHNIQUE; PARABOLIC DIFFERENTIAL EQUATION SOLUTIONS; ORDINARY LINEAR AND NONLINEAR EQUATION SOLUTION; TURBULAR FLOW HEAT TRANSFER ANALYSIS; COAXIAL LAMINAR TURBULAR FLOWS; RADIATIVE FIN ANALYSIS; HOT SURFACE FUEL SPRAY IGNITION AND COMBUSTION | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper presents a new numerical method, the error method, for solving para- bolic type partial differential equations, linear or nonlinear. In particular, by comparing with the regular successive iterative method, more beneficial results in the application to nonlinear problems. Three nonlinear examples were studied by using this method. All resulted in large reduction of number of iteration loops and CPU time required in comparing to the corresponding regular successive method used. Generalization and modification of this method appears appropriate to extend its application to elliptical type partial differential | | |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

equation so that problems with "isolated" events (such as those with ignition spots) may be handled with this method.

Unclassified

TABLE OF CONTENTS

Abstract

Preface

I. Introduction

II. The Successive Iterative Method

III. The Error Method

IV. The Application of Error Method to Non-linear Problems

(1) A Laminar Tubular Flow Heat Transfer Problem

(2) A Coaxial Laminar Tubular Flow Problem

(3) A Radiative Fin Problem

V. Discussion and Conclusion

Appendix A--H

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is
approved for public release IAW AFR 199-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

ABSTRACT

This paper presents a new numerical method, the error method, for solving parabolic type partial differential equations, linear or non-linear. In particular, by comparing with the regular successive iterative method, more beneficial results in the application to non-linear problems. Three non-linear examples were studied by using this method. All resulted in large reduction of number of iteration loops and CPU time required in comparing to the corresponding regular successive method used.

Generalization and modification of this method appears appropriate to extend its application to elliptical type partial differential equation so that problems with "isolated" events (such as those with ignition spots) may be handled with this method.

PREFACE

The financial support of Air Force Office of Scientific Research through Grant AFOSR-78-3538 is highly appreciated. Many thanks to the great patience of Dr. Bernard T. Wolfson who managed this program throughout the entire period.

I am grateful to the computer services provided by computer centers of both Harvard University and Southeastern Massachusetts University.

I. Introduction

This paper presents a new numerical method, the error method, for solving parabolic type partial differential equations, linear or non-linear. Observing from the cases studied, it appears that the application of the error method to non-linear problems results in more beneficial than to linear problems in comparing with the conventional workable successive iterative numerical scheme. This new method features wide applicability. Its computational scheme converges fast. The following sections will describe the fundamental derivation of this method, its applications to non-linear problems and the future potential developments.

II. The Successive Iterative Method

To construct the basic numerical scheme of the error method, let's start by reviewing the conventional successive iterative method. Consider the problem of an one-dimensional, unsteady heat conduction in a solid plane wall with finite thickness. Its governing equation is,

$$T_t = T_{xx} \quad , \quad (1)$$

where T , the unknown temperature field and x, t , the effective spacious and time coordinates. The set of initial and boundary conditions may be,

$$\left. \begin{array}{l} t = 0, \quad T = T_0(x) \\ x = 0, \quad T = C_1 \\ x = 1, \quad T = C_2 \end{array} \right\} \quad (2)$$

(2)

where l , the effective wall thickness and C_1, C_2 , two fixed temperature values on the two wall surfaces. $T_0(x)$ represents the initial temperature profile across the wall thickness.

To such a problem, the finite difference successive iterative numerical method can be used. The field, along x direction, is first

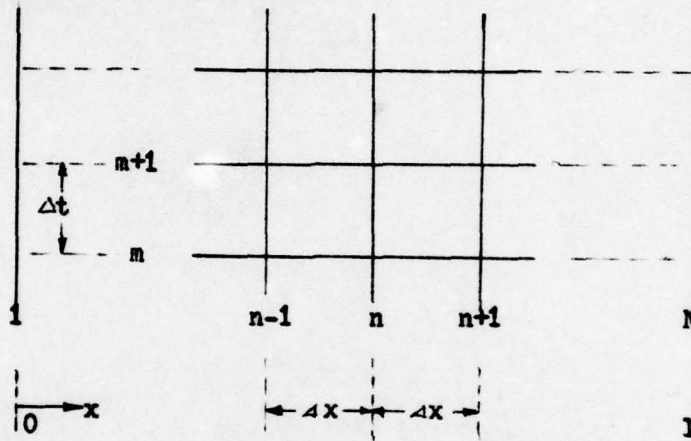


Figure 1. Grid System for the Numerical Scheme

divided into $(N-1)$ grids. Each grid sizes Δx . An implicit finite difference equation is constructed from equation (1), i.e.

$$T_{m+1,n} = \frac{T_{m,n+1} + T_{m,n-1}}{2 \left(1 + \frac{\Delta x^2}{\Delta t}\right)} + F(T_{m,*}) \quad (3)$$

$$\text{where } F(T_{m,*}) \equiv \frac{T_{m,n+1} - 2T_{m,n} + T_{m,n-1}}{2\Delta x^2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta t}\right)} + \frac{T_{m,n}}{\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta t}\right)}$$

and n varying from 2 to $N-1$. The solution of equation (3), together with the two boundary conditions specified, determine the various $T_{m+1,n}$ as $T_{m,n}$ are

(3)

given. With successive iterative method, solution of equation (3) starts by assuming a set of $T_{m+1,x}$ values, together with the specified information for $F(T_{m,x})$ from time instant, m , substituting into the right side of equation (3), then calculating a new set of $T_{m+1,x}$. This new set of $T_{m+1,x}$ is then replacing the old set of $T_{m+1,x}$ to plug into the right side of equation (3) again and calculating the next new set of $T_{m+1,x}$. Such procedure repeats itself until the calculated set of $T_{m+1,x}$ revealing no significant difference from the previous set of $T_{m+1,x}$. The last set of calculated $T_{m+1,x}$ values are considered the satisfied solution. The selection of the starting $T_{m+1,x}^{(0)}$ profile is not unique. Ordinarily, to use the profile, $T_{m,x}$, as the first trial profile, $T_{m+1,x}^{(0)}$ appear not a bad choice. Since, intuitively, as we may expect that the correct unknown profile, $T_{m+1,x}$, should not be different from the known profile of the previous time instant, $T_{m,x}$, too much if Δt is small enough and no "isolated" event occurs. To assure the stable convergence of the calculation steps and lead to reasonable solution, it appears that the criterion,

$$\frac{\Delta x^2}{\Delta t} > 2 \quad (4)$$

ought to be observed in selecting the relative sizes of Δx and Δt .

According to above description, a Fortran program (SPD1), attached as Appendix A, was constructed to allow digital computer to perform the steps in the iterative computation procedure for this one dimensional, unsteady heat conduction problem. A case with the initial condition, $T_0(x)$, possessing

one surface temperature at elevated level was studied. Some calculated temperature profiles at various time instants are shown in Table 1. In this table, the number of iteration loop required for calculating each temperature profile is also indicated; so is the total CPU time used. Also, cases with highly non-linear initial conditions were also investigated with this program, the results indicate that the successive method appears very effective to handle such linear problems.

Table 1 Some Temperature Profiles for a One-Dimensional, Unsteady Heat Conduction Problem (Successive Method)

Initial Temperature Profile: $T=200.0$ at $x=0.0$ and $t=0.0$

$T=40.0$ at $x=0.0$ and $t=0.0$

Boundary Conditions: $T=200.0$ at $x=0.0$ and $t=0.0$

$T=40.0$ at $x=1.0$ and $t=0.0$

The Label: Orderly number of Temp Profile, No. of Iteration Loop, Time Inst. Values of Temperature Profile

| | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|
| | 1 | 0 | 0.00000 | | |
| 200.00000 | 40.00000 | 40.00000 | 40.00000 | 40.00000 | 40.00000 |
| 40.00000 | 40.00000 | 40.00000 | 40.00000 | 40.00000 | 40.00000 |
| 40.00000 | 40.00000 | 40.00000 | 40.00000 | 40.00000 | 40.00000 |
| 40.00000 | 40.00000 | 40.00000 | | | |
| | 20 | 8 | 0.01900 | | |
| 200.00000 | 167.54154 | 137.17704 | 110.59675 | 88.81256 | 72.08412 |
| 60.03503 | 51.88440 | 46.69865 | 43.59019 | 41.83147 | 40.89040 |
| 40.41314 | 40.18326 | 40.07784 | 40.03172 | 40.01242 | 40.00468 |
| 40.00168 | 40.00053 | 40.00000 | | | |
| | 40 | 9 | 0.03900 | | |
| 200.00000 | 177.23933 | 155.20179 | 134.54165 | 115.78639 | 99.29813 |
| 85.25890 | 73.67917 | 64.42534 | 57.25884 | 51.87905 | 47.96330 |
| 45.19894 | 43.30546 | 42.04644 | 41.23314 | 40.72176 | 40.40711 |
| 40.21456 | 40.09160 | 40.00000 | | | |
| | 60 | 9 | 0.05900 | | |
| 200.00000 | 181.46837 | 163.32715 | 145.94209 | 129.63219 | 114.65246 |
| 101.18298 | 89.32492 | 79.10333 | 70.47551 | 63.34362 | 57.56936 |
| 52.98915 | 49.42818 | 46.71205 | 44.67555 | 43.16817 | 42.05653 |
| 41.22417 | 40.56913 | 40.00000 | | | |

III The Error Method

In applying conventional successive numerical scheme for solving parabolic partial differential equations, we may "feel" one feature, i.e., the "passiveness" of the computation scheme. Except selecting the sizes of the independent variables and specifying the initial condition, there is no more handle which can be used to alter or control the development of the computation. Furthermore, the convergent rates of the successive method are often small, especially, to non-linear problems. Frequently, hundreds, or even, thousands iteration loops are required to achieve a single step solution. On account of all these unsatisfactory characteristics with the conventional successive method, the error method is developed. Essentially, the major difference between the successive and the error method is that the later imposes mechanism for estimating and correcting "errors" of trial values of the unknown quantities in a regular successive routing. So the overall computation procedure is greatly shortened. To illustrate the basic scheme of the error method, let us re-consider the unsteady, one-dimensional heat conduction problem of last section. Let $T_{m+1,n}^{(1)}$ represents the first trial temperature profile at $m+1$ time instant, $T_{m+1,n}^*$ represents the "true solution" (still unknown yet). It is clear,

$$T_{m+1,n}^* = T_{m+1,n}^{(1)} + \delta_n \quad (5)$$

where δ_n , the deviation of trial profile, $T_{m+1,n}^{(1)}$, from the "true solution", $T_{m+1,n}^*$. To start the iteration process by substituting $T_{m+1,n}^{(1)}$ into the right side of Equation (3), just as same as done in successive method, and decomposing

(6)

$T_{m+1,n}^{(1)}$ according to equation (5), we obtain

$$T_{m+1,n}^{(2)} = \frac{T_{m+1,n+1}^* + T_{m+1,n-1}^*}{2(1 + \frac{\Delta x^2}{\Delta t})} + F(T_{m,n}^*) - \frac{\delta_{n+1} + \delta_{n-1}}{2(1 + \frac{\Delta x^2}{\Delta t})}, \quad (6)$$

where $T_{m+1,n}^{(2)}$, the calculated temperature profile at $m+1$ time instant based on $T_{m+1,n}^{(1)}$. Subtracting both sides of equation (6) by $T_{m+1,n}^{(1)}$, we then obtain,

$$R_n = T_{m+1,n}^{(2)} - T_{m+1,n}^{(1)} = \delta_n - \frac{\delta_{n+1} + \delta_{n-1}}{2(1 + \frac{\Delta x^2}{\Delta t})}. \quad (7)$$

Equation (7), the error equation, provides $N-2$ required equations for solving set of N unknowns, δ s. The other two equations required are formed from the two boundary conditions. For this particular case, it is obvious that $\delta_1 = \delta_N = 0$. By determining the set of δ values from equation (7), the trial values, $T_{m+1,n}^{(1)}$, can then be modified and iteration processes repeat, if needed, until the satisfactory "true solution" is achieved. To solve equation (7) and the corresponding boundary conditions for δ , a set of recurrency formular are derived. Let

$$\delta_n = A_n \delta_{n-1} + B_n, \quad (8)$$

where A_n and B_n are undetermined correlation parameters between neighboring δ s. Re-arrange equation (7), we obtain,

$$\delta_{n+1} = 2(1 + \frac{\Delta x^2}{\Delta t})(\delta_n - R_n) - \delta_{n-1}. \quad (9)$$

(7)

Eliminating δ_{n-1} from equation (9) and equation (8), we have the recurrency formular,

$$A_{n+1} = 2\left(1 + \frac{\Delta x^2}{\Delta t}\right) - \frac{1}{A_n} \quad (10)$$

$$B_{n+1} = \frac{B_n}{A_n} - 2\left(1 + \frac{\Delta x^2}{\Delta t}\right) R_n$$

To start the operation on the recurrency formular, we know, from equation (9) and boundary condition, $\delta_1 = 0$, that

$$A_3 = 2\left(1 + \frac{\Delta x^2}{\Delta t}\right) \quad (11)$$

$$B_3 = -2\left(1 + \frac{\Delta x^2}{\Delta t}\right) R_2$$

After all the A_n and B_n , for $3 \leq n \leq N$, are determined, since $\delta_N = 0$, therefore,

$$\delta_{N-1} = -\frac{B_N}{A_N} \quad (12)$$

The rest δ s are calculated from equation (8) in reverse order. The determined δ values are used to modify the trial values of unknown variables, $T_{m+1, *}$, and the iteration cycles repeat, if needed, until satisfactory "true solution" is obtained. It is clear, to this problem, a linear case, equation (7) is a linear equation of δ . However, for non-linear problem, equation (7) would be a non-linear equation of δ . To facilitate the calculation, linearisation procedure is used to convert equation (7) into linear form of δ . The other

steps of derivation are identical to those described above.

For clarity, the above description about the derivation of the basic error method will be summarized as follows:

(1) Corresponding to the governing differential equations as well as the boundary conditions, form the implicit finite difference equations, ready to start the regular iteration procedure,

(2) By using the first trial values of the unknown variables, say $T_{m+1,x}^{(1)}$, calculate the unknown variables, say $T_{m+1,x}^{(2)}$,

(3) By recognizing $T_{m+1,x}^* = T_{m+1,x}^{(1)} + \delta_n$, where $T_{m+1,x}^*$, the "true solution", deduce the linear error equations, $R_n = f(\delta_{m+1}, \delta_n, \delta_{n-1})$, where $R_n = T_{m+1,x}^{(2)} - T_{m+1,x}^{(1)}$, from the finite difference numerical scheme,

(4) Let $\delta_{m+1} = A_{n+1} \delta_n + B_{n+1}$, where A_{n+1} , B_{n+1} , the undetermined correlation parameters. Derive the recurrency formular for A_{n+1} , B_{n+1} , i.e.,

$$A_{n+1} = G(A_n)$$

$$B_{n+1} = H(A_n, B_n),$$

from the error equations. Determine the starting A , B values from the related boundary condition,

(5) Calculate A_n , B_n for n up to N . Determine δ_n , first, with the help of one boundary condition and then with the correlation formular,

$$\delta_{m+1} = A_{n+1} \delta_n + B_{n+1},$$

(6) Modify $T_{m+1,x}^{(1)}$ and repeat steps (2) to (5), if needed, until the satisfactory "true solution" is obtained.

With above procedure, a Fortran program, SPD2, was constructed and attached as Appendix B. One case, same as studied by SPD1, was investigated by this program. The results are identical to those from SPD1. Table 2 shows

the comparison of the number of iteration loops and CPU time required by SPD2 and SPD1 for that particular case studied. From Table 2, it appears

Table 2. A comparison of the Number of Iteration Loop and CPU Time Required by Successive and Error Methods for the Problem Studied in Table 1

| Program | No. of iteration loop per time step | Total CPU time required for 60 time steps |
|---------|-------------------------------------|---|
| SPD1 | 6--9 (most, 9) | 5.30 sec |
| SPD2 | 1 (occasionally 2, once 3) | 4.53 sec |

that the CPU time saved by the error method in reference to successive method is only about 20 %, a value not very impressive. However, this observation may also be interpreted as that the successive method is already fairly effective in applying to linear problem.

IV The Application of Error Method to Non-linear Problems

Three non-linear examples are presented to illustrate the application and effectiveness of the error method. To facilitate the presentation, brief qualitative description of these three problems are given first as follows:

(1) A steady laminar tubular flow heat transfer problem with fluid thermal conductivity varying linearly with local temperature. This problem features with non-linear heat conduction term in the transverse direction to the tube flow.

(2) A steady coaxial laminar tubular flow problem with all the fluid properties varying with local temperature. The problem is sketched in Figure 2. As shown in this diagram, it is clear that radius direction slope of the

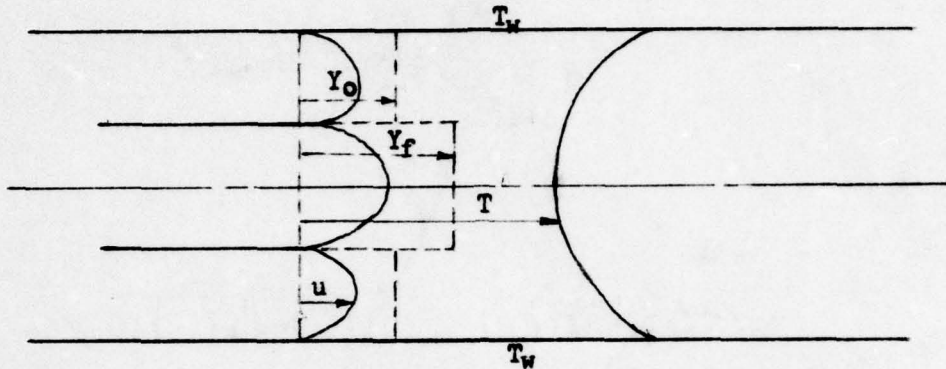
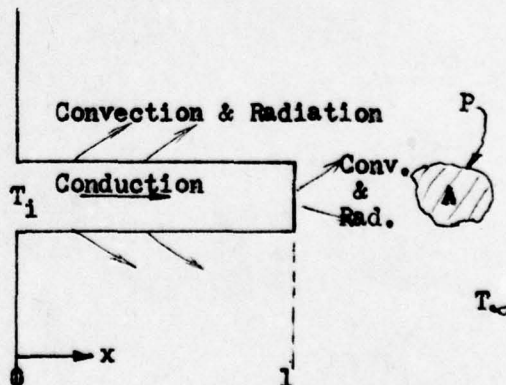


Figure 2. The Coaxial Tubular Flow Sketch

initial velocity profile is discontinuous at inner tube wall location. The concentration profiles of both the inner and outer tube gases are discontinuous at the inner tube wall location. All the diffusion terms in the transverse direction are non-linear.

(3) An unsteady, one-dimensional fin problem with radiation. The

Figure 3. The Sketch of a radiative fin



schematic diagram is shown in Figure 3. To this problem, the highly non-linear radiation term presents in both differential equation as well as the boundary condition.

The formulation of all three problems described above are shown

The formulation of all three problems described are shown in

Table 3.

Table 3. Formulation of the Three Non-linear Problems

| Items | Tube H.T. Problem | Coaxile Tube Problem | Radiative fin Problem |
|---------------------------------|---|--|--|
| Governing Equations | $\frac{\partial}{\partial x}(Pu) = 0$ $Pu \frac{\partial T}{\partial x} = \frac{1}{Re Pr} + \frac{\partial}{\partial r} (Tr \frac{\partial T}{\partial r})$ $PT = 1$ | $\frac{\partial}{\partial x}(Pu) = 0$ $Pu \frac{\partial T}{\partial x} = \frac{1}{Re Pr} + \frac{\partial}{\partial r} (Tr \frac{\partial T}{\partial r})$ $Pu \frac{\partial Y_0}{\partial x} = \frac{1}{Re Sc} + \frac{\partial}{\partial r} (Tr \frac{\partial Y_0}{\partial r})$ $Pu \frac{\partial Y_f}{\partial x} = \frac{1}{Re Sc} + \frac{\partial}{\partial r} (Tr \frac{\partial Y_f}{\partial r})$ $PT = 1$ | $\frac{\partial^2 \theta}{\partial x^2} = \frac{\partial \theta}{\partial t} + D\theta$ $\theta \equiv \frac{T - T_\infty}{T_i - T_\infty}$ $D \equiv 1 + C_1(\theta^3 + 4T_r\theta + 6T_r^2\theta + 4T_r^3)$ $C_1 \equiv \epsilon \sigma (T_i - T_\infty)^3 / h$ $T_r = T_\infty / (T_i - T_\infty)$ |
| Initial and Boundary Conditions | $x=0, T = T_0(r)$ $r=0, \frac{\partial T}{\partial r} = 0$ $r=0.5, T = T_w$ | $x=0, T = T_0(r), Y_0 = Y_{00}(r), Y_f = Y_{f0}(r)$ $r=0, \frac{\partial T}{\partial r} = \frac{\partial Y_0}{\partial r} = \frac{\partial Y_f}{\partial r} = 0$ $r=0.5, T = T_w, \frac{\partial Y_0}{\partial r} = \frac{\partial Y_f}{\partial r} = 0$ | $t=0, \theta = 1$ $x=0, \theta = 1$ $x = ml, \frac{\partial \theta}{\partial x} = -c_c D\theta$ $ml \equiv l \sqrt{\frac{hP}{kA}}$, dimensionless fin length $c_c \equiv \frac{h}{km}$, Biot number |
| Finite Difference equations | $T_{2,j} = \left\{ \left(\frac{C_p C_j}{4} \right)^2 + C_p C_j T_{1,j} + \frac{J_{12}}{2} (T_{1,j+1} + T_{2,j+1}) + \frac{J_{32}}{2} (T_{1,j-1} + T_{2,j-1}) \right\}^{\frac{1}{2}} - T_{1,j} - \frac{C_p C_j}{4}$ $C_j \equiv (Pu)_j$ $C_p = \frac{8\Delta r^2 Re Pr}{\Delta x}$ $J_{12} = (J - \frac{1}{2}) / (J - 1)$ $J_{32} = (J - \frac{3}{2}) / (J - 1)$ $T_{2,N} = T_w$ | Temperature expressions are identical to those in left column, $Y_{2,j} = \frac{J_{12} T_{H1}}{C_{sj}} (Y_{1,j+1} + Y_{2,j+1}) + \frac{J_{32} T_{L1}}{C_{sj}} (Y_{1,j-1} + Y_{2,j-1}) + \frac{C_{sj} - J_{12} T_{H1} - J_{32} T_{L1}}{C_{sj}} Y_{1,j}$ $C_{sj} = C_s C_j + J_{12} T_{H1} + J_{32} T_{L1}$ $C_s = \frac{8\Delta r^2 Re Sc}{\Delta x}$ $T_{H1} = T_{1,j+1} + T_{2,j+1} + T_{1,j-1} + T_{2,j-1}$ $T_{L1} = T_{1,j} + T_{2,j} + T_{1,j-1} + T_{2,j-1}$ $J_{12} = (J - \frac{1}{2}) / (J - 1)$ $J_{32} = (J - \frac{3}{2}) / (J - 1)$ Y can be Y_0 or Y_f | $\theta_{2,j} = \frac{\theta_{x1}}{2\Delta x^2 C_d} - \frac{C_u}{C_d} \theta_{1,j}$ $\theta_{x1} = \theta_{2,j+1} + \theta_{2,j-1} + \theta_{1,j+1} + \theta_{1,j-1}$ $C_d = \frac{1}{2}\Delta x^2 + \frac{1}{2}\Delta t + 0.5 D_{1/2,j}$ $C_u = \frac{1}{2}\Delta x^2 - \frac{1}{2}\Delta t + 0.5 D_{1/2,j}$ $\theta_{2,N} = \frac{1}{c_e} \left\{ \frac{2\theta_{1,N+1} - \theta_{1,N}}{2\Delta x} - \frac{\Delta x}{2\Delta t} \left(\frac{\theta_{1,N+1} - \theta_{1,N}}{\Delta x} - \theta_{1,N'} \right) - c_c \left[\frac{D_N \theta_{1,N}}{2} + D_{N'} \frac{\Delta x}{b} \left(\frac{\theta_{1/2,N+1}}{4} + \frac{3\theta_{1,N}}{8} \right) \right] \right\}$ $C_e = \frac{1}{2\Delta x} + \frac{3\Delta x}{8\Delta t} + C_c \left(\frac{D_N}{2} + \frac{3D_{N'}}{8} \cdot \frac{\Delta x}{b} \right)$ N' , the location $\frac{1}{2}\Delta x$ from N point |

(Continue at next page)

Table 3. (continued)

| Items | Tube H.T. Problem | Coaxial Tube Problem | Radiative Fin Problem |
|---|---|---|--|
| Linearized Error Equations | $\delta_{j+1} = C_a(\delta_j - R_j) - C_b \delta_{j-1}$ $C_a = \frac{2 C_{au}}{J_{12}(T_{1,j+1} + T_{2,j+1})}$ $C_{au} = \left\{ \left(\frac{T_{1,j}}{T_1} \right)^2 + C_p C_j T_{1,j} + \frac{J_{12}}{2} (T_{1,j+1} + T_{2,j+1})^2 + \frac{J_{22}}{2} (T_{1,j-1} + T_{2,j-1})^2 \right\}^{\frac{1}{2}}$ $C_b = \frac{J_{12}(T_{1,j-1} + T_{2,j-1})}{J_{12}(T_{1,j+1} + T_{2,j+1})}$ $R_j = T_{2,j}^{(2)} - T_{2,j}^{(1)}$ | Temperature error eqs. are identical to those in left column, $\delta_{j+1} = C_a(\delta_j - R_j) - C_b \delta_{j-1}$ $C_a = \frac{C_{sj}}{J_{12} \cdot T_{11}}$ $C_b = \frac{J_{32} \cdot T_{-1}}{J_{12} \cdot T_{11}}$ $R_j = Y_{2,j}^{(2)} - Y_{2,j}^{(1)}$ δ can be δ_f or δ_o R can be R_f or R_o . | $\delta_{j+1} = C_a \delta_j - \delta_{j-1} - C_b R_j$ $C_a = 24X^2 C_d + \frac{4t \theta_{21} + 4X^2}{24t C_d} C_1 \left[\frac{3}{2} \theta_{12,j}^2 + 4T_r \theta_{12,j} + 3T_r^2 \right]$ $C_b = 24X^2 C_d$ $R_j = \theta_{2,j}^{(2)} - \theta_{2,j}^{(1)}$ $\delta_N = S_g \delta_{N+1} + R_N C_e / S_g d$ $S_g = S_{gu} / S_g d$ $S_{gu} = \frac{1}{24X} - \Delta X / 8at - 0.25 C_c \frac{\Delta X}{b} \cdot \left[\frac{3 \theta_{1,N} C_{D(N-1)}}{8} + \frac{D_N}{2} + \frac{\theta_{1,N-1} C_{D(N-1)}}{4} + \frac{3 \theta_{2,N} C_{D(N-1)}}{8} \right]$ $S_g d = \frac{1}{24X} + \frac{34X}{84E} + C_c \left\{ \frac{D_N}{2} + \frac{34X D_N}{8b} + \left[\left(\frac{1}{2} + \frac{9}{32} \frac{\Delta X}{b} \right) \theta_{1,N} + \frac{\theta_{1,N}}{2} + \frac{9}{32} \frac{\Delta X}{b} \theta_{1,N} + \frac{34X}{16b} \theta_{1,N-1} \right] C_{DN} \right\}$ $C_{Dj} = C_1 \left(\frac{3}{2} \theta_{12,j}^2 + 4T_r \theta_{12,j} + 3T_r^2 \right)$ |
| Recurrency formular | $A_{j+1} = C_a - \frac{C_b}{A_j}$ $B_{j+1} = \frac{C_b B_j}{A_j} - C_a R_j$ | $A_{j+1} = C_a - \frac{C_b}{A_j}$ $B_{j+1} = \frac{C_b B_j}{A_j} - C_a R_j$ A, B, R, C_a, C_b are for all T, Y_f and Y_o fields. | $A_{j+1} = C_a - \frac{1}{A_j}$ $B_{j+1} = \frac{B_j}{A_j} - C_a R_j$ |
| Information derived from boundary conditions | $A_2 = 1.0, B_2 = 0.0$ $C_{N-1} = -\frac{B_N}{A_N}$ | For temperature : $A_2 = 1.0, B_2 = 0.0$ $\delta_{N-1} = -\frac{B_N}{A_N}$ For Y_f and Y_o : $A_2 = 1.0, B_2 = 0.0$ $\delta_N = \delta_{N-1} = \frac{B_N}{1 - A_N}$ all A, B, δ can be applied to Y_o and Y_o fields. | $A_2 = 1.0, B_2 = 0.0$ $\delta_{N-1} = \frac{B_N - \frac{R_N C_e}{S_g d}}{S_g - A_N}$ |

(continued on next page)

Table 3. (continued)

| Items | Tube H.T. Problem | Coaxial Tube Problem | Radiative Fin Problem |
|---|--|---|--|
| Modified Trial Profile for next iteration, if needed | $T_{2j}^{(i')} = T_{2j}^{(i)} + \delta_j$ | $T_{2j}^{(i')} = T_{2j}^{(i)} + \delta_j T$ $Y_{f2j}^{(i')} = Y_{f2j}^{(i)} + \delta_j Y_f$ $Y_{o2j}^{(i')} = Y_{o2j}^{(i)} + \delta_j Y_o$ | $\Theta_{2j}^{(i')} = \Theta_{2j}^{(i)} + \delta_j$ |
| Nomenclature unspecified by above and left expressions, | ρ , fluid density u , fluid velocity T , fluid temperature x , tube axial coord. r , tube radial coord. Re , Reynolds number Pr , Prandtl number $T_o(r)$, entrance temperature Profile T_w , tube wall temperature j (or J), j th node point along r -direction N , total number of nodal points along r -direction δ , deviation of trial value from "true solution." | Y_o , outer tube gas conc. Y_f , inner tube gas conc. $Y_{oo}(r)$, entrance Y_o profile $Y_{fo}(r)$, entrance Y_f Profile Sc , Schmidt number | T_1 , initial and fin root temperature T_∞ , environment temp. ϵ , emissivity σ , Stef-Boltzmann Cons. h , conv. h. t. Coef. l , fin length P , fin cross section perimeter length k , fin thermal cond. A , fin cross section area |

Based upon formular in Table 3, computer programs with error method were designed for the above three problems as SPD6, SPD4 and SPD8. They are attached as Appendix F, D and H. Three corresponding programs (SPD5, SPD3 and SPD7), based on successive method, for these three problems are also constructed and attached as Appendix E, C and G. Computational effectiveness of these programs were investigated on several sample cases. Table 4 shows several computational effectiveness of various programs run.

Table 4 Comparison of CPU time and Number of Iteration Loop required by Successive and Error Method for the three non-linear problems

| Items | | Tube Flow | Coaxile Flow | Radiative Fin |
|----------------------------|-------|---|--|--|
| Given Conditions | | Re=500.00 Pr=1.0 T _w =3.0 N=41 M (total steps calculated)= 80 | Re=500.00 Pr=1.0 Sc=1.0 T _w =3.0 N=41 M=40 | m1=1.0 C =2.0 C =0.2 T _r =0.4 B(fin th.)=0.05 N=41 M=80 |
| Program used | Succ | SPD5 | SPD3 | SPD7 |
| | Error | SPD6 | SPD4 | SPD8 |
| No. of Loops used per step | Succ | 49-73 | 85-127 | 7-9 |
| | Error | 3 | 4 | 4 |
| Total CPU Time | Succ | 53.85 sec | 67.70 sec | 29.05 sec |
| | Error | 5.55 sec* | 30.18 sec | 4.71 sec* |

*print only 1/10th as many lines as did by the comparable programs.

Table 4 indicates that, by using error method, the CPU time used can be reduced to about half or less in reference to those corresponding calculation done by successive method.

V Discussion and Conclusion

(1) Investigation in previous sections has indicated that the error method developed up to this stage does provide us an alternative way for solving parabolic partial differential equations. The utilization of this method, in particular to non-linear problems, results in significant reducing CPU time and number of iteration loops required. This feature, together with its wide applicability may turn this method into an attractive tool.

(2) The recurrency technique suggested here can be considered as an effective method for solving the three line matrix.

(3) The applicability of the error method to ordinary differential equations, linear or non-linear, is obvious.

(4) To linear parabolic type equation, the above recurrency technique can be applied directly to the unknown parameters instead of the errors or the trial values of the unknown parameters. It thus provides another alternative method. To non-linear equation, however, the direct application of the recurrency technique to the unknown parameters may not be able to lead to numerical scheme which can convergence as fast as the current error method.

(5) Along the line of designing new computational method, a systematic exploitation of the applicability of the error method to problems with elliptical type partial differential equation may not be inappropriate.

APPENDIX A

DAY

Tuesday, November 13, 1979 16:31:02

@TY SPD1.FOR

```

00100 C      SPD1--SOLUTION FOR LINEAR HEAT TRANSFER PROBLEM,
00200 C      ONE DIMENSIONAL UNSTEADY HEAT CONDUCTION
00300 C      SUCCESSIVE METHOD                      SEPTEMBER 17,1979
00400      DIMENSION U(101),UN(101),RS(101),UNO(101)
00450      OPEN(UNIT=1,ACCESS='SEQIN',DATA=LOGUE)
00500      READ(1,10) N,MAX
00600      10 FORMAT(4I10)
00700      READ(1,20) U(1),UI
00800      20 FORMAT(8F10.5)
00850      DY=1.0/FLOAT(N-1)
00900      DT=DY*DY/2.5
01000      DO 30 I=2,N
01100      U(I)=UI
01200      30 CONTINUE
01300      M=1
01350      TIME=0.0
01400      WRITE(5,150) N,NL,TIME,(U(I),I=1,N)
01500      DO 40 I=1,N
01600      UN(I)=U(I)
01700      UNO(I)=UR(I)
01800      40 CONTINUE
01900      N1=N-1
02000      CF1=2.0*(1.0+DY*DY/DT)
02100      CF2=1.0+DT/(DY*DY)
02200      50 NL=0
02300      DO 60 I=2,N1
02400      IP=I+1
02500      I1=I-1
02600      RS(I)=(U(IP)+U(I1)-2.0*U(I))/CF1+U(I)/CF2
02700      60 CONTINUE
02800      70 NT=0
02900      DO 90 I=2,N1
03000      IF=I+1
03100      I1=I-1
03200      UN(I)=(UR(IP)+UR(I1))/CF1+RS(I)
03300      DNO=ABS(UR(I)-UNO(I))
03400      IF(DNO-0.000001) 85,85,80
03500      80 NT=1
03600      85 UNO(I)=UR(I)
03700      90 CONTINUE
03800      IF(NT-1) 120,100,100
03900      100 NL=NL+1
04000      GO TO 70
04100      120 M=M+1
04150      TIME=TIME+DT
04200      DO 130 I=1,N
04300      U(I)=UN(I)
04400      UNO(I)=UR(I)
04500      130 CONTINUE
04600      WRITE(5,150) M,NL,TIME,(UN(I),I=1,N)
04700      150 FORMAT(10X,2I10,F10.5,/20(5X,6(F9.5,1X)/))
04800      IF(M-MAX) 50,200,200
04900      200 STOP
05000      END

```

APPENDIX B

```

DAY
Wednesday, November 14, 1979 11:37:07
@TY SPD2.FOR
00100 C      SPD2--NEW METHOD FOR LINEAR HEAT TRANSFERPROBLEM,
00200 C      ONE DIMENSIONAL UNSTEADY HEAT CONDUCTION
00300 C      ERROR METHOD
00400      DIMENSION U(101),UN(101),RS(101),A(101),B(101),
00500      IR(101),DEL(101)
00600      OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
00700      READ(1,10) N,MAX
00800      10 FORMAT(4I10)
00900      READ (1,20) U(1),UI
01000      20 FORMAT(8F10.5)
01100      DY=1.0/FLOAT(N-1)
01200      DT=DY*DY/2.5
01300      DO 30 I=2,N
01400      U(I)=UI
01500      30 CONTINUE
01600      M=1
01700      TIME=0.0
01800      WRITE(5,150) M,NL,TIME,(U(I),I=1,N)
01900      DO 40 I=1,N
02000      UN(I)=U(I)
02100      40 CONTINUE
02200      N1=N-1
02300      CF1=2.0*(1.0+DY*DY/DT)
02400      CF2=1.0+DT/(DY*DY)
02500      50 NL=0
02600      DO 60 I=2,N1
02700      IP=I+1
02800      I1=I-1
02900      RS(I)=(U(IP)+U(I1))-2.0*U(I))/CF1+U(I)/CF2
03000      60 CONTINUE
03100      70 NT=0
03200      A(2)=-10.0**25
03300      B(2)=0.0
03400      DEL(N)=0.0
03500      DO 130 I=2,N1
03600      IP=I+1
03700      I1=I-1
03800      UNI=(UN(IP)+UN(I1))/CF1+RS(I)
03900      DNO=ABS(UNI-UN(I))
04000      R(I)=UNI-UN(I)
04100      A(I+1)=CF1-1.0/A(I)
04200      90 B(I+1)=B(I)/A(I)-CF1*R(I)
04300      93 IF(DNO.LE.0.00001) GO TO 130
04400      NT=1
04500      130 CONTINUE
04600      IF(NT-1)200,140,140
04700      140 DO 180 I=2,N1
04800      IR1=N-I+1
04900      IR2=IR1+1
05000      DEL(IR1)=DEL(IR2)/A(IR2)-B(IR2)/A(IR2)
05100      UN(IR1)=UN(IR1)+DEL(IR1)
05200      180 CONTINUE
05300      NL=NL+1
05400      GO TO 70
05500      200 M=M+1
05600      TIME=TIME+DT
05700      DO 230 I=1,N
05800      U(I)=UN(I)
05900      230 CONTINUE
06000      WRITE(5,150) M,NL,TIME,(UN(I),I=1,N)

```

```
06100 150 FORMAT(10X,2I10,F10.5,/20(5X,6(F9.5,1X)/))
06200 IF(M-MAX) 50,300,300
06300 300 STOP
06400 END
@
```

APPENDIX C

```

DAY
Tuesday, November 13, 1979 19:09:36
@TY SPD3.FOR
00100 C   SPD3--DIFFUSION IN DOUBLE TUBE FLOW PROBLEM, JULY/31/79
00200 C   SUCCESSIVE METHOD
00300     DIMENSION T(2,101), YF(2,101), YD(2,101), U(2,101), RD(2,101),
00400     1C(101), DLT(101), DLTN(101), RD1(101), TN(101), YFN(101),
00500     2YDN(101), ROYF(101), ROYD(101), DLYF(101), DYFN(101),
00600     3R(101), DLYD(101), DYDN(101)
00700     OPEN(UNIT=1, ACCESS='SEQIN', DIALOGUE)
00800     READ(1,10) M,N,RE,PR,SC,TW,WP,CJ
00900     10 FORMAT(2I10,10F10.5)
01300     DR=0.5/FLOAT(N-1)
01350     DX=DR*DR*RE/2.5
01360     WRITE(5,15) M,N,DX,RE,PR,SC,TW,WP,CJ
01370     15 FORMAT(2I5,8(F10.5,1X)/)
01380 C   BOUNDARY CONDITIONS
01400     R(1)=0.0
01500     N1=N-1
01600     DO 16 J=1,N1
01700     J1=J+1
01800     R(J1)=R(J)+DR
01900     16 CONTINUE
02000     LNI=0.5*WP/DR+1.0
02100     COP=8.0*DR**2*RE*PR/DX
02200     COS=COP*SC/PR
02300     DO 30 J=1,N
02400     AJ=J-1
02500     DO 25 I=1,2
02600     T(I,J)=1.0+(1W-1.0)*(A.J*DR*2.0)**2
02700     RD(I,J)=1.0/T(I,J)
02800     IF(J-LNI) 17,17,19
02900     17 YF(I,J)=1.0
03000     YD(I,J)=0.0
03100     U(I,J)=CJ*(1.0-(2.0*R(J)/WP)**2)
03200     GO TO 25
03300     19 YF(I,J)=0.0
03400     YD(I,J)=0.5
03500     U(I,J)=1.0-(2.0*R(J))**2+(1.0-WP**2)*ALOG(1.0/(2.0*R(J)
03600     1))/ALOG(WP)
03700     25 CONTINUE
03800     C(J)=U(1,J)*RD(1,J)
03900     30 CONTINUE
04000     C(N)=0.0
04100     C(2)=C(1)
04200     C(LNI)=0.25*(C(LNI+1)+C(LNI-1))
04300     U(1,N)=0.0
04400     U(2,N)=0.0
04500     YD(1,LNI)=0.25
04600     YD(2,LNI)=0.25
04700     YF(1,LNI)=0.5
04800     YF(2,LNI)=0.5
04900     IK=1
05000     SMTUR=0.0
05100     SMUR=0.0
05200     DO 33 J=2,N1
05300     AJ1=J-1
05400     SMTUR=SMTUR+T(1,J)*U(1,J)*AJ1
05500     SMUR=SMUR+U(1,J)*AJ1
05600     33 CONTINUE
05700     TBBB=SMTUR/SMUR
05800     QW=260.0*(TW-T(1,N1))/DR
05900     ANULT=QW/(260.0*(TW-TBBB))

```

```

06000      WRITE(5,535) IK,NIT,(T(1,J),J=1,N)
06100      WRITE(5,570) ANUL,T,RW
06200      WRITE(5,950) (U(1,J),J=1,N)
06300      WRITE(5,565) X,(RO(1,J),J=1,N)
06400      WRITE(5,566) (YF(1,J),J=1,N)
06500      WRITE(5,568) (YO(1,J),J=1,N)
06600      X=0.0
06700      43 IK=IK+1
06800      IF(IK-M)46,46,1703
06900      46 X=X+DX
07000      C      UNKNOWN S NEAR THE CENTRAL LINE(CENTRAL R.C.)
07100      NIT=1
07200      48 DO 57 I=1,2
07300      T(I,2)=T(I,1)
07400      YF(I,2)=YF(I,1)
07500      YO(I,2)=YO(I,1)
07600      57 CONTINUE
07700      C      UNKNOWN S IN THE FIELD(2,N1)
07800      61 NT=0
07900      DO 1300 J=2,N1
08000      J1=J-1
08100      AJ1=J1
08200      AJ12=(AJ1+0.5)/AJ1
08300      AJ32=(AJ1-0.5)/AJ1
08400      CDP=COP*C(J)+4.0*T(1,J)
08500      SQT2=SQRT((0.25*CDP*C(J))**2+CDP*C(J)*T(1,J)+0.5*AJ12*
08600      1 (T(1,J+1)+T(2,J+1))**2+0.5*AJ32*(T(1,J-1)+T(2,J-1))**2)
08700      TN(J)=-0.25*CDP+SQT2
08800      TUS=T(1,J+1)+T(2,J+1)+T(1,J)+T(2,J)
08900      TLS=T(1,J-1)+T(2,J-1)+T(1,J)+T(2,J)
09000      CD2J=COS*C(J)+AJ12*TUS+AJ32*TLS
09100      YFN(J)=(AJ12*TUS*(YF(1,J+1)+YF(2,J+1))+AJ32*TLS*
09200      1 (YF(1,J-1)+YF(2,J-1))+(COS*C(J)-AJ12*TUS-AJ32*TLS)*
09300      2 YF(1,J))/CD2J
09400      YON(J)=(AJ12*TUS*(YO(1,J+1)+YO(2,J+1))+AJ32*TLS*
09500      1 (YO(1,J-1)+YO(2,J-1))+(COS*C(J)-AJ12*TUS-AJ32*TLS)*
09600      2 YO(1,J))/CD2J
09700      ROT(J)=TN(J)-T(2,J)
09800      ROYF(J)=YFN(J)-YF(2,J)
09900      ROYD(J)=YON(J)-YO(2,J)
10000      AROT=ABS(ROT(J))
10100      AROYF=ABS(ROYF(J))
10200      AROYD=ABS(ROYD(J))
10300      IF(AROT-0.000001) 572,572,1150
10400      572 IF(AROYF-0.000001)590,590,1150
10500      590 IF(AROYD-0.000001)1300,1300,1150
10600      1150 NT=1
10700      1300 CONTINUE
10800      IF(NT) 1567,1567,1330
10900      1330 NIT=NIT+1
11000      DO 1538 J=2,N1
11100      T(2,J)=TN(J)
11200      YF(2,J)=YFN(J)
11300      YO(2,J)=YON(J)
11400      1538 CONTINUE
11500      T(2,N)=TW
11600      YF(2,N)=YF(2,N1)
11700      YO(2,N)=YO(2,N1)
11800      T(2,1)=T(2,2)
11900      YF(2,1)=YF(2,2)
12000      YO(2,1)=YO(2,2)
12100      GO TO 61
12200      1567 QW=260.0*(TW-T(2,N1))/DR
12300      DO 919 J=1,N
12400      RO(2,J)=1.0/T(2,J)
12500      U(2,J)=C(J)/RO(2,J)

```

```

12600      919  CONTINUE
12700      SMTUR=0.0
12800      SMUR=0.0
12900      DO 908 J=2,N1
13000      J1=J-1
13100      SMTUR=SMTUR+T(2,J)*U(2,J)*R(J)*(2.0*DR)
13200      SMUR=SMUR+U(2,J)*R(J)*(2.0*DR)
13300      908  CONTINUE
13400      TBBB=SMTUR/SMUR
13500      ANULT=QW/(260.0*(TW-TBBB))
13600      WRITE(5,535) IK,NIT,(T(2,J),J=1,N)
13700      WRITE(5,570) ANULT,QW
13800      WRITE(5,950) (U(2,J),J=1,N)
13900      WRITE(5,565) X,(RO(2,J),J=1,N)
14000      WRITE(5,566) (YF(2,J),J=1,N)
14100      WRITE(5,568) (YO(2,J),J=1,N)
14200      535  FORMAT(2X,' 1,QW',2I5,5(F9.5,1X))/25
14300      1   (17X,5(F9.5,1X)/)
14400      570  FORMAT(17X,F9.5,1X,F9.3)
14500      950  FORMAT(2X,'    U',10X,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
14600      565  FORMAT(2X,'  X,RO',F10.6,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
14700      566  FORMAl(2X,'    YF',10X,5(F9.5,1X)/25(17X,5(F9.5,1X)/))
14800      568  FORMAl(2X,'    YO',10X,5(F9.5,1X)/25(17X,5(F9.5,1X)/))
14900      DO 1679 J=1,N
15000      T(1,J)=T(2,J)
15100      YF(1,J)=YF(2,J)
15200      YO(1,J)=YO(2,J)
15300      RO(1,J)=1.0/T(1,J)
15400      U(1,J)=C(J)/RO(1,J)
15500      1679 CONTINUE
15600      1697 GO TO 43
15700      1703 STOP
15800      END

```

@

APPENDIX D

DAY

Tuesday, November 13, 1979 19:08:00

@TY SPD4.FOR

```

00100 C   SPD4--DIFFUSION IN DOUBLE TUBE FLOW PROBLEM, JULY/31/79
00200 C   ERROR METHOD
00300     DIMENSION T(2,101), YF(2,101), YO(2,101), U(2,101), RO(2,101),
00400     1C(101), 1N(101), YFR(101),
00500     2YOR(101), AT(101), BT(101), AY(101), BYF(101),
00600     3R(101), BYO(101), DEL(101), DEF(101), DEO(101)
00700     OPEN(UNIT=1, ACCESS='SEQIN', DIALOGUE)
00800     READ(1,10) M,N,RF,PR,SC,TW,WP,CJ
00900     10 FORMAT(2I10,10F10.5)
01300     DR=0.5/FLOAT(N-1)
01350     DX=DR*DR*RF/2.5
01360     WRITE(5,15) M,N,DX,RE,PR,SC,TW,WP,CJ
01370     15 FORMAT(2I5,8(F10.5,1X)/)
01380 C   BOUNDARY CONDITIONS
01400     R(1)=0.0
01500     N1=N-1
01600     DO 16 J=1,N1
01700     J1=J+1
01800     R(J1)=R(J)+DR
01900     16 CONTINUE
02000     LNI=0.5*WP/DR+1.0
02100     COP=8.0*DR**2*RF*PR/DX
02200     COS=COP*SC/PR
02300     DO 30 J=1,N
02400     AJ=J-1
02500     DO 25 I=1,2
02600     T(I,J)=1.0+(TW-1.0)*(A.J)*DR**2.0)**2
02700     RO(I,J)=1.0/T(I,J)
02800     IF(J-LNI) 17,17,19
02900     17 YF(I,J)=1.0
03000     YO(I,J)=0.0
03100     U(I,J)=CJ*(1.0-(2.0*R(J)/WP)**2)
03200     GO TO 25
03300     19 YF(I,J)=0.0
03400     YO(I,J)=0.5
03500     U(I,J)=1.0-(2.0*R(J))**2+(1.0-WP**2)*ALOG(1.0/(2.0*R(J)
03600     1))/ALOG(WP)
03700     25 CONTINUE
03800     C(J)=U(1,J)*RO(1,J)
03900     30 CONTINUE
04000     C(N)=0.0
04100     C(2)=C(1)
04200     C(LNI)=0.25*(C(LNI+1)+C(LNI-1))
04300     U(1,N)=0.0
04400     U(2,N)=0.0
04500     YO(1,LNI)=0.25
04600     YO(2,LNI)=0.25
04700     YF(1,LNI)=0.5
04800     YF(2,LNI)=0.5
04900     IK=1
05000     SMTUR=0.0
05100     SMUR=0.0
05200     DO 33 J=2,N1
05300     AJ1=J-1
05400     SMTUR=SMTUR+T(1,J)*U(1,J)*AJ1
05500     SMUR=SMUR+U(1,J)*AJ1
05600     33 CONTINUE
05700     TBBB=SMTUR/SMUR
05800     QW=260.0*(TW-T(1,N1))/DR
05900     ANULT=QW/(260.0*(TW-TBBB))

```

```

06000 WRITE(5,535) IK,NJT,(T(1,J),J=1,N)
06100 WRITE(5,570) ANULT,QW
06200 WRITE(5,950) (U(1,J),J=1,N)
06300 WRITE(5,565) X,(RO(1,J),J=1,N)
06400 WRITE(5,566) (YF(1,J),J=1,N)
06500 WRITE(5,568) (YO(1,J),J=1,N)
06600 X=0.0
06700 43 IK=IK+1
06800 IF(IK-M)46,46,1703
06900 46 X=X+DX
07000 C UNKNOWN S NEAR THE CENTRAL LINE(CENTRAL B.C.)
07100 NIT=1
07200 48 DO 57 I=1,2
07300 T(I,2)=T(I,1)
07400 YF(I,2)=YF(I,1)
07500 YO(I,2)=YO(I,1)
07600 57 CONTINUE
07700 C UNKNOWN S IN THE FIELD(2,N1)
07800 61 NT=0
07900 AT(2)=1.0
08000 BT(2)=0.0
08100 DO 1300 J=2,N1
08200 J1=J-1
08300 AJ1=J1
08400 AJ12=(AJ1+0.5)/AJ1
08500 AJ32=(AJ1-0.5)/AJ1
08600 CCOF=COF*C(J)+4.0*T(1,J)
08700 SQT2=SQR T((0.25*COF*C(J))**2+COF*C(J)*T(1,J)+0.5*AJ12*
08800 1(T(1,J+1)+T(2,J+1))**2+0.5*AJ32*(T(1,J-1)+T(2,J-1))**2)
08900 TN(J)=-0.25*CCOF+SQT2
09000 ROT=TN(J)-T(2,J)
09100 AROT=ABS(ROT)
09200 CAT=2.0*SQT2/(AJ12*(T(1,J+1)+T(2,J+1)))
09300 CBT=AJ32*(T(1,J-1)+T(2,J-1))/(AJ12*(T(1,J+1)+T(2,J+1)))
09400 AT(J+1)=CAT-CBT/AT(J)
09500 BT(J+1)=CBT*BT(J)/AT(J)-CAT*ROT
09600 345 IF(AROT-0.000001) 1300,1300,1150
09700 1150 NT=1
09800 1300 CONTINUE
09900 IF(NT.LE.0) GO TO 1567
10000 DEL(N1)=-BT(N)/AT(N)
10100 T(2,N1)=T(2,N1)+DEL(N1)
10200 DO 1313 J=2,N1
10300 NJ=N-J
10400 NJ1=N+1-J
10500 DEL(NJ)=(DEL(NJ1)-BT(NJ1))/AT(NJ1)
10600 T(2,NJ)=T(2,NJ)+DEL(NJ)
10700 1313 CONTINUE
10800 T(2,1)=T(2,2)
10900 T(2,N)=TW
11000 NIT=NJT+1
11100 GO TO 61
11200 1567 NY=0
11300 AY(2)=1.0
11400 BYF(2)=0.0
11500 BYO(2)=0.0
11600 DO 1400 J=2,N1
11700 J1=J-1
11800 AJ1=J1
11900 AJ12=(AJ1+0.5)/AJ1
12000 AJ32=(AJ1-0.5)/AJ1
12100 TUS=T(1,J+1)+T(2,J+1)+T(1,J)+T(2,J)
12200 TLS=T(1,J)+T(2,J)+T(1,J-1)+T(2,J-1)
12300 CO2J=COF*C(J)+AJ12*TUS+AJ32*TLS
12400 YFN(J)=(AJ12*TUS*(YF(1,J+1)+YF(2,J+1))+AJ32*TLS*
12500 1 (YF(1,J-1)+YF(2,J-1))+(COF*C(J)-AJ12*TUS-AJ32*TLS)*

```

```

12600      2  YF(1,J))/C02J
12700      YON(J)=(AJ12*TUS*(YO(1,J+1)+YO(2,J+1))+AJ32*TLS*
12800      1  (YO(1,J-1)+YO(2,J-1)))+(COS*C(J)-AJ12*TUS-AJ32*TLS)*
12900      2  YO(1,J))/C02J
13000      ROYF=YFN(J)-YF(2,J)
13100      ROYO=YON(J)-YO(2,J)
13200      ARDYF=ABS(ROYF)
13300      ARYD=ABS(ROYD)
13400      CAY=C02J/(AJ12*TUS)
13500      CBY=AJ32*TLS/(AJ12*TUS)
13600      AY(J+1)=(CAY-CBY/AY(J)
13700      BYF(J+1)=CBY*BYF(J)/AY(J)-CAY*ROYF
13800      BYD(J+1)=CBY*BYD(J)/AY(J)-CAY*ROYD
13900      IF(ARDYF.LE.0.000001) GO TO 1395
14000      NY=1
14100      1395 IF(ARYD.LE.0.000001) GO TO 1400
14200      NY=1
14300      1400 CONTINUE
14400      IF(NY.LE.0) GO TO 1576
14500      DEF(N1)=BYF(N)/(1.0-AY(N))
14600      DEO(N1)=BYD(N)/(1.0-AY(N))
14700      YF(2,N1)=YF(2,N1)+DEF(N1)
14800      YO(2,N1)=YO(2,N1)+DEO(N1)
14900      DO 1414 J=2,N1
15000      NJ=N-J
15100      NJ1=N+1-J
15200      DEF(NJ)=(DEF(NJ1)-BYF(NJ1))/AY(NJ1)
15300      DEO(NJ)=(DEO(NJ1)-BYD(NJ1))/AY(NJ1)
15400      YF(2,NJ)=YF(2,NJ)+DEF(NJ)
15500      YO(2,NJ)=YO(2,NJ)+DEO(NJ)
15600      1414 CONTINUE
15700      YF(2,1)=YF(2,2)
15800      YF(2,N)=YF(2,N1)
15900      YO(2,1)=YO(2,2)
16000      YO(2,N)=YO(2,N1)
16100      NIT=NIT+1
16200      GO TO 1567
16300      1576 RW=260.0*(TW-T(2,N1))/DR
16400      DO 919 J=1,N
16500      RO(2,J)=1.0/T(2,J)
16600      U(2,J)=C(J)/RO(2,J)
16700      919 CONTINUE
16800      SMTUR=0.0
16900      SMUR=0.0
17000      DO 908 J=2,N1
17100      J1=J-1
17200      SMTUR=SMTUR+T(2,J)*U(2,J)*R(J)*(2.0*DR)
17300      SMUR=SMUR+U(2,J)*R(J)*(2.0*DR)
17400      908 CONTINUE
17500      TBBB=SMTUR/SMUR
17600      ANULT=RW/(260.0*(TW-TBBB))
17700      WRITE(5,535) IK,NIT,(T(2,J),J=1,N)
17800      WRITE(5,570) ANULT,RW
17900      WRITE(5,950) (U(2,J),J=1,N)
18000      WRITE(5,565) X,(RO(2,J),J=1,N)
18100      WRITE(5,566) (YF(2,J),J=1,N)
18200      WRITE(5,568) (YO(2,J),J=1,N)
18300      535 FORMAT(2X,' T,RW',2I5,5(F9.5,1X))/25
18400      1  (17X,5(F9.5,1X)/)
18500      570 FORMAT(17X,F9.5,1X,F9.3)
18600      950 FORMAT(2X,' U',10X,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
18700      565 FORMAT(2X,' X,RO',F10.6,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
18800      566 FORMAT(2X,' YF',10X,5(F9.5,1X)/25(17X,5(F9.5,1X)/))
18900      568 FORMAT(2X,' YO',10X,5(F9.5,1X)/25(17X,5(F9.5,1X)/))
19000      DO 1679 J=1,N
19100      T(1,J)=T(2,J)

```

```
19200      YF(1,J)=YF(2,J)
19300      YD(1,J)=YD(2,J)
19400      RO(1,J)=1.0/T(1,J)
19500      U(1,J)=C(J)/RO(1,J)
19600      1679 CONTINUE
19700      1697 GO TO 43
19800      1703 STOP
19900      END
```

@

APPENDIX E

DAY

Tuesday, November 13, 1979 20:03:13

@TY SPDS.FOR

```

00100 C      SPDS--NON LINEAR TUBE FLOW HEAT TRANSFER PROBLEM,
00150 C      SUCCESSIVE METHOD          NOVEMBER 1,1979
00200      DIMENSION T(2,101),U(2,101),RO(2,101),
00300      1C(101),TN(101),
00500      3R(101)
00600      OPEN(UNIT=1,ACCESS='SERIAL',DIALOGUE)
00700      READ(1,10) M,N,RE,PR,TW
00800      10 FORMAT(2I10,10F10.5)
01200      DR=0.5/FLOAT(N-1)
01250      DX=DR*DR*RE/2.5
01260      WRITE(5,15) M,N,DX,RE,PR,TW
01270      15 FORMAT(2I5,8(F10.5,1X)/)
01280 C      BOUNDARY CONDITIONS
01300      R(1)=0.0
01400      N1=N-1
01500      DO 16 J=1,N1
01600      J1=J+1
01700      R(J1)=R(J)+DR
01800      16 CONTINUE
02000      COP=8.0*DR**2*RE*PR/DX
02200      DO 30 J=1,N
02300      AJ=J-1
02400      DO 25 I=1,2
02500      T(I,J)=1.0+(TW-1.0)*(AJ*DR*2.0)**2
02600      RO(I,J)=1.0/T(I,J)
03000      U(I,J)=1.0-(2.0*R(J))**2
03600      25 CONTINUE
03700      C(J)=U(1,J)*RO(1,J)
03800      30 CONTINUE
03900      C(N)=0.0
04000      U(1,N)=0.0
04100      U(2,N)=0.0
05200      IK=1
05300      SMTUR=0.0
05400      SMUR=0.0
05500      DO 33 J=2,N1
05600      AJ1=J-1
05700      SMTUR=SMTUR+T(1,J)*U(1,J)*AJ1
05800      SMUR=SMUR+U(1,J)*AJ1
05900      33 CONTINUE
06000      TBBB=SMTUR/SMUR
06100      QW=260.0*(TW-T(1,N1))/DR
06200      ANULT=QW/(260.0*(TW-TBBB))
06300      WRITE(5,535) IK,NIT,(T(1,J),J=1,N)
06400      WRITE(5,570) ANULT,QW
06500      WRITE(5,950) (U(1,J),J=1,N)
06600      WRITE(5,565) X,(RO(1,J),J=1,N)
06900      X=0.0
07000      43 IK=IK+1
07100      IF(IK-M)46,46,1703
07200      46 X=X+DX
07300 C      UNKNOWNS NEAR THE CENTRAL LINE(CENTRAL B.C.)
07400      NIT=1
07500      48 DO 57 I=1,2
07600      T(I,2)=T(I,1)
07700      U(I,2)=U(I,1)
07800      RO(I,2)=RO(I,1)
08100      57 CONTINUE
08200 C      UNKNOWNS IN THE FIELD(2,N1)
08300      61 NT=0

```

```

08400      DO 1300 J=2,N1
08500      J1=J-1
08600      AJ1=J1
08700      AJ12=(A,J1+0.5)/AJ1
08800      AJ32=(A,J1-0.5)/AJ1
08900      CCOF=COP*C(J)+4.0*T(1,J)
09000      SQT2=SQRT((0.25*CCOF*C(J))**2+COP*C(J)*T(1,J)+0.5*AJ12*
09100      1(T(1,J+1)+1(2,J+1))**2+0.5*AJ32*(T(1,J-1)+T(2,J-1))**2)
09200      TN(J)=-0.25*CCOF+SQT2
10100      ROT=TN(J)-T(2,J)
10400      AROT=ABS(ROT)
10700      IF(AROT-0.000001) 1300,1300,1150
11000      1150 NT=1
11100      1300 CONTINUE
11200      IF(NT.LE.0) GO TO 1567
11300      DO 1313 J=2,N1
11400      T(2,J)=TN(J)
11700      1313 CONTINUE
11800      T(2,1)=T(2,2)
12250      NIT=NIT+1
12300      GO TO 61
12400      1567 QW=260.0*(TW-T(2,N1))/DR
12420      DO 919 J=1,N
12440      RO(2,J)=1.0/T(2,J)
12460      U(2,J)=C(J)/RO(2,J)
12480      919 CONTINUE
12500      SMTUR=0.0
12600      SMUR=0.0
12700      DO 908 J=2,N1
12800      J1=J-1
12900      SMTUR=SMTUR+T(2,J)*U(2,J)*R(J)*(2.0*DR)
13000      SMUR=SMUR+U(2,J)*R(J)*(2.0*DR)
13100      908 CONTINUE
13200      TBBB=SMTUR/SMUR
13300      ANULT=QW/(260.0*(TW-TBBB))
13400      WRITE(5,535) IK,NIT,(T(2,J),J=1,N)
13500      WRITE(5,570) ANULT,QW
13600      WRITE(5,950) (U(2,J),J=1,N)
13700      WRITE(5,565) X,(RO(2,J),J=1,N)
14000      535 FORMAT(2X,' T,QW',2I5,5(F9.5,1X)/25
14100      1 (17X,5(F9.5,1X)/))
14200      570 FORMAT(17X,F9.5,1X,F9.3)
14300      950 FORMAT(2X,' U',10X,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
14400      565 FORMAT(2X,' X,RO',F10.6,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
14700      DO 1679 J=1,N
14800      T(1,J)=T(2,J)
15100      RO(1,J)=1.0/T(1,J)
15200      U(1,J)=C(J)/RO(1,J)
15300      1679 CONTINUE
15400      1697 GO TO 43
15500      1703 STOP
15600      END

```

@

APPENDIX F

```

DAY
Tuesday, November 13, 1979 20:28:37
@TY SPD6.FOR
00100 C      SPD6--NONLINEAR TUBE FLOW HEAT TRANSFER PROBLEM,
00150 C      ERROR METHOD,          NOVEMBER 4, 1979
00200      DIMENSION T(2,101),U(2,101),RO(2,101),
00300      1C(101),A(101),B(101),TW(101),
00500      3R(101),DEL(101),TD(2,101)
00600      OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
00700      READ(1,10) M,N,RE,PR,TW
00800      10 FORMAT(2I10,10F10.5)
01200      DR=0.5/FLOAT(N-1)
01250      DX=DR*DR*RE/2.5
01270      IP=1
01275      WRITE(5,15) M,N,DX,RE,PR,TW
01280      15 FORMAT(2I5,8(F10.5,1X)/)
01285 C      BOUNDARY CONDITIONS
01300      R(1)=0.0
01400      N1=N-1
01500      DO 16 J=1,N1
01600      J1=J+1
01700      R(J1)=R(J)+DR
01800      16 CONTINUE
02000      COP=8.0*DR**2*RE*PR/DX
02200      DO 30 J=1,N
02300      AJ=J-1
02400      DO 25 I=1,2
02500      T(I,J)=1.0+(TW-1.0)*(AJ*DR**2.0)**2
02600      RO(I,J)=1.0/T(I,J)
03000      U(I,J)=1.0-(2.0*R(J))**2
03600      25 CONTINUE
03700      C(J)=U(1,J)*RO(1,J)
03800      30 CONTINUE
03900      C(N)=0.0
04000      U(1,N)=0.0
04100      U(2,N)=0.0
05200      IK=1
05300      SMTUR=0.0
05400      SMUR=0.0
05500      DO 33 J=2,N1
05600      AJ1=J-1
05700      SMTUR=SMTUR+T(1,J)*U(1,J)*A,1
05800      SMUR=SMUR+U(1,J)*AJ1
05900      33 CONTINUE
06000      TBBB=SMTUR/SMUR
06100      QW=260.0*(TW-T(1,N1))/DR
06200      ANULT=QW/(260.0*(TW-TBBB))
06300      WRITE(5,535) IK,NIT,(T(1,J),J=1,N)
06400      WRITE(5,570) ANULT,QW
06500      WRITE(5,950) (U(1,J),J=1,N)
06600      WRITE(5,565) X,(RO(1,J),J=1,N)
06900      X=0.0
07000      43 IK=IK+1
07050      IP=IP+1
07100      IF(IK-M)46,46,1703
07200      46 X=X+DX
07300 C      UNKNOWN NEAR THE CENTRAL LINE(CENTRAL B.C.)
07400      NIT=1
07500      DO 57 I=1,2
07600      T(I,2)=T(I,1)
07700      U(I,2)=U(I,1)
07800      RO(I,2)=RO(I,1)
08100      57 CONTINUE

```

```

08200 C      UNKNOWNNS IN THE FIELD(2,N1)
08300      61 NT=0
08340      A(2)=1.0
08380      B(2)=0.0
08400      DO 1300 J=2,N1
08500      J1=J-1
08600      AJ1=J1
08700      AJ12=(AJ1+0.5)/AJ1
08800      AJ32=(AJ1-0.5)/AJ1
08900      CCOF=COP*C(J)+4.0*T(1,J)
09000      SQT2=SQRT((0.25*CCOF*C(J))*2+COP*C(J)*T(1,J)+0.5*AJ12*
09100      1(T(1,J+1)+1(2,J+1))*2+0.5*AJ32*(T(1,J-1)+T(2,J-1))*2)
09200      TN(J)=-0.25*CCOF+SQT2
10100      ROT=TN(J)-T(2,J)
10400      AROT=ABS(ROT)
10420      CA=2.0*SQT2/(AJ12*(T(1,J+1)+T(2,J+1)))
10440      CB=AJ32*(T(1,J-1)+T(2,J-1))/(AJ12*(T(1,J+1)+T(2,J+1)))
10460      A(J+1)=CA-CB/A(J)
10480      B(J+1)=CB*B(J)/A(J)-CA*ROT
10700      IF(AROT-0.000001) 1300,1300,1150
11000      1150 NT=1
11100      1300 CONTINUE
11200      IF(NT.LE.0) GO TO 1567
11220      DEL(N1)=-B(N)/A(N)
11240      T(2,N1)=T(2,N1)+DEL(N1)
11300      DO 1313 J=2,N1
11350      NJ=N-J
11400      NJ1=N+1-J
11450      DEL(NJ)=(DEL(NJ1)-B(NJ1))/A(NJ1)
11500      T(2,NJ)=T(2,NJ)+DEL(NJ)
11700      1313 CONTINUE
11800      T(2,1)=T(2,2)
11820      T(2,N)=TW
12250      NIT=NIT+1
12300      GO TO 61
12400      1567 QW=260.0*(TW-T(2,N1))/DR
12410      DO 919 J=1,N
12420      RO(2,J)=1.0/T(2,J)
12430      U(2,J)=C(J)/RO(2,J)
12440      919 CONTINUE
12500      SMTUR=0.0
12600      SMUR=0.0
12700      DO 908 J=2,N1
12800      J1=J-1
12900      SMTUR=SMTUR+T(2,J)*U(2,J)*R(J)*(2.0*DR)
13000      SMUR=SMUR+U(2,J)*R(J)*(2.0*DR)
13100      908 CONTINUE
13200      TBBB=SMTUR/SMUR
13300      ANULT=QW/(260.0*(TW-TBBB))
13320      IF(IP.NE.11) GO TO 731
13340      IP=1
13400      WRITE(5,535) JK,NIT,(T(2,J),J=1,N)
13500      WRITE(5,570) ANULT,RW
13600      WRITE(5,950) (U(2,J),J=1,N)
13700      WRITE(5,565) X,(RO(2,J),J=1,N)
14000      535 FORMAT(2X,' T,QW',2I5,5(F9.5,1X)/25
14100      1 (17X,5(F9.5,1X)/))
14200      570 FORMAT(17X,F9.5,1X,F9.3)
14300      950 FORMAT(2X,' U',10X,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
14400      565 FORMAT(2X,' X,RO',F10.6,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
14700      731 DO 1679 J=1,N
14800      T(1,J)=T(2,J)
15100      RO(1,J)=1.0/T(1,J)
15200      U(1,J)=C(J)/RO(1,J)
15300      1679 CONTINUE
15400      1697 GO TO 43

```

15500
15600

1703 STOP
END

@

APPENDIX G

DAY

Tuesday, November 13, 1979 21:14:26

@TY SPD7.F

@TY SPD7.FOR

```

00100 C      SPD7--HEAT TRANSFER WITH RADIATION,AN UNSTEADY,
00200 C      ONE DIMENSIONAL FIN PROBLEM,NOVEMBER 13,1979
00300 C      SUCCESSIVE METHOD
00400      DIMENSION T(2,101),D(101),TN(101)
00500      OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
00600      READ(1,10) GC,CC,RC,TR,B
00700      READ(1,20) M,N
00800      10  FORMAT(8F10.5)
00900      20  FORMAT(4I10)
01000      N1=N-1
01100      DX=GC/FL0AT(N1)
01200      DT=DX*DX/2.5
01300      WRITE(5,30) GC,CC,RC,TR,B,DX,DT,M,N
01400      30  FORMAT(10X,7F10.5,2I10/)
01500      DO 50 I=1,2
01600      DO 40 J=1,N
01700      T(I,J)=1.0
01800      40  CONTINUE
01900      50  CONTINUE
02000      IK=1
02100      NIT=1
02150      TIME=0.0
02200      WRITE(5,53) IK,NIT,TIME,(T(1,J),J=1,N)
02300      53  FORMAT(2X,2I5,F8.5,1X,5(F9.5,1X)/,20(21X,5(F9.5,1X)/)/)
02400      56  NIT=1
02500      58  NT=0
02600      TN(1)=T(1,1)
02700      DO 80 J=2,N
02800      TM=0.5*(T(1,J)+T(2,J))
02900      D(J)=1.0+RC*(TK**3+4.0*TR*TM**2+6.0*TR**2*TM+4.0*TR**3)
03000      IF(J.EQ.N) GO TO 87
03100      UCF=1.0/DX**2-1.0/DT+0.5*D(J)
03200      ACF=1.0/DX**2+1.0/DT+0.5*D(J)
03300      CTG=T(2,J+1)+T(2,J-1)+T(1,J+1)+T(1,J-1)
03400      TN(J)=0.5*CTG/(DX**2*ACF)-UCF*T(1,J)/ACF
03500      ROT=TN(J)-T(2,J)
03600      AROT=ABS(ROT)
03700      IF(AROT.LE.0.000001) GO TO 77
03800      NT=1
03820      77  IF(J.NE.N1) GO TO 80
03840      DNP=0.25*D(J)
03860      T1P=0.25*T(1,J)
03900      80  CONTINUE
03920      87  DNP=DNP+0.75*D(N)
03940      T1P=T1P+0.75*T(1,N)
03960      TM1=0.5*(T(1,N1)+T(2,N1))
03980      TUP=(TM1-0.5*T(1,N))/DX-0.5*DX*(0.25*T(2,N1)-T1P)/DT
04000      1  -CC*(0.5*D(N)*T(1,N)+DX*DNP*(0.25*TK1+0.375*T(1,N))/B)
04020      TLF=0.5/DX+0.375*DX/DT+CC*(0.5*D(N)+0.375*DX*DNP/B)
04040      TN(N)=TUP/TLF
04200      DO 110 J=1,N
04300      T(2,J)=TN(J)
04400      110 CONTINUE
04450      IF(NT.FR.0) GO TO 145
04500      NIT=NIT+1
04600      GO TO 58
04700      145 IK=IK+1
04750      TIME=TIME+DT
04800      WRITE(5,53) IK,NIT,TIME,(T(2,J),J=1,N)

```

04900 DO 153 J=1,N
05000 T(1,J)=T(2,J)
05100 153 CONTINUE
05200 IF(IK.L1.K) GO TO 56
05300 STOP
05400 END
@

APPENDIX H

DAY

Tuesday, November 13, 1979 21:32:51

@TY SPDB.FOR

```

00100 C SPDB--HEAT TRANSFER WITH RADIATION, AN UNSTEADY,
00200 C ONE DIMENSIONAL FIN PROBLEM, NOVEMBER 13, 1979
00300 C ERROR METHOD
00400 DIMENSION T(2,101), D(101), TN(101), AT(101), BT(101), DEL(10
1)
00500 OPEN(UNIT=1, ACCESS='SEQUENTIAL', DIA=10)
00600 READ(1,10) GC,CC,RC,TR,B
00700 READ(1,20) M,N
00800 10 FORMAT(8F10.5)
00900 20 FORMAT(4I10)
01000 N1=N-1
01100 DX=GC/FLOAT(N1)
01200 DT=DX*DX/2.5
01300 WRITE(5,30) GC,CC,RC,TR,B,DX,DT,M,N
01400 30 FORMAT(10X,7F10.5,2I10/)
01500 DO 50 I=1,2
01600 DO 40 J=1,N
01700 T(I,J)=1.0
01800 40 CONTINUE
01900 50 CONTINUE
02000 IK=1
02050 IP=1
02100 NIT=1
02150 TIME=0.0
02200 WRITE(5,53) IK,NIT,TIME,(T(1,J),J=1,N)
02300 53 FORMAT(2X,2I5,F8.5,1X,5(F9.5,1X)/,20(21X,5(F9.5,1X)/)/)
02400 56 NIT=1
02500 58 NT=0
02600 AT(2)=1.0
02700 BT(2)=0.0
02800 DO 80 J=2,N
02900 TM=0.5*(T(1,J)+T(2,J))
03000 D(J)=1.0+RC*(TM**3+4.0*TR*TM**2+6.0*TR**2*TM+4.0*TR**3)
03100 CDJ=RC*(1.5*TM**2+4.0*TR*TM+3.0*TR**2)
03200 IF(J.EQ.N) GO TO 87
03300 UCF=1.0/DX**2-1.0/DT+0.5*D(J)
03400 ACF=1.0/DX**2+1.0/DT+0.5*D(J)
03500 CTG=T(2,J+1)+T(2,J-1)+T(1,J+1)+T(1,J-1)
03600 TN(J)=0.5*CTG/(DX**2*ACF)-UCF*T(1,J)/ACF
03700 ROT=TN(J)-T(2,J)
03800 AROT=ABS(ROT)
03900 CA=2.0*DX**2*ACF+0.5*(DT*CTG+4.0*DX**2)*CDJ
04000 1 / (DT*ACF)
04100 CB=2.0*DX**2*ACF
04200 AT(J+1)=CA-1.0/AT(J)
04300 BT(J+1)=BT(J)/AT(J)-CB*ROT
04400 IF(AROT.LE.0.000001) GO TO 77
04500 NT=1
04600 77 IF(J.NE.N1) GO TO 80
04700 DNP=0.25*D(J)
04800 CDN1=CDJ
04900 T1NP=0.25*T(1,J)
05000 80 CONTINUE
05100 87 IF(NT.EQ.0) GO TO 145
05200 DNP=DNP+0.75*D(N)
05220 T1NP=T1NP+0.75*T(1,N)
05240 TM1=0.5*(T(1,N1)+T(2,N1))
05260 TUN=(TM1-0.5*T(1,N))/DX-0.5*DX*(0.25*T(2,N1)-T1NP)/DT
05280 1 -CC*(0.5*D(N)*T(1,N)+DX*DNP*(0.25*TM1+0.375*T(1,N))/B)

```

```

05290      TLN=0.5/DX+0.375*DX/DT+CC*(0.5*D(N)+0.375*DNP*DX/B)
05295      ROT=TUR/TLN-T(2,N)
05300      SGRU=0.5/DX-0.125*DX/DT-0.25*CC*DX*(0.375*T(1,N)*CDN1
05400      1  +0.5*DNP+0.125*(T(2,N1)+T(1,N1))*CDN1+0.375*T(2,N)*CDN1)
/B
05500      SGRD=0.5/DX+0.375*DX/DT+CC*(0.5*D(N)+0.375*DX*DNP/B
05600      1  +(0.5+0.28125*DX/B)*T(2,N)*CDJ+(0.5*T(1,N)+0.28125*
05700      2  DX*T(1,N)/B+0.09375*DX*(T(2,N1)+T(1,N1))/B)*CDJ)
05800      SGR=SGRU/SGRD
05840      PRL=0.5/DX+0.375*DX/DT+CC*(0.5*D(N)+0.375*DNP*DX/B)
05900      DEL(N1)=(-BT(N)+ROT*PRL/SGRD)/(-SGR+AT(N))
06000      DEL(N)=AT(N)*DEL(N1)+BT(N)
06100      N2=N1-1
06200      DO 93 J=2,N2
06300      JR=N-J
06400      JR1=N+1-J
06500      DEL(JR)=(DEL(JR1)-BT(JR1))/AT(JR1)
06600      93  CONTINUE
06700      DO 110 J=2,N
06800      T(2,J)=T(2,J)+DEL(J)
06900      110  CONTINUE
07000      NIT=NIT+1
07100      GO TO 58
07200      145  IK=IK+1
07220      TIME=TIME+DT
07240      IP=IP+1
07260      IF(IP.NE.11) GO TO 149
07280      IP=1
07300      WRITE(5,53) IK,NIT,TIME,(T(2,J),J=1,N)
07400      149  DO 153 J=1,N
07500      T(1,J)=T(2,J)
07600      153  CONTINUE
07700      IF(IK.LT.M) GO TO 56
07800      STOP
07900      END
@

```

LOGO

Killed Job 38, User F.1SHEN, Account , TTY 36,
at 13-Nov-79 21:34:06; Used 0:12:23 in 6:11:50