ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

CSDT Simulation Model for
Prediction of IHX Transient Behavior
Using S/360 CSMP

by

J. F. L. M. Brukx

Reactor Analysis and Safety Division

Presented in Partial Fulfillment of
the Requirements for the Degree of
Master of Mechanical Engineering
in the
Delft University of Technology
The Netherlands

August 1974

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

This document is
PUBLICLY RELEASABLE
Mary R. Hale, ANL
Authorizing Official
Date 8-17-09

RECEIVED
August 21, 1974
Department of Nuclear
Engineering
University of Illinois
Program
Peg

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

TABLE OF CONTENTS

	<u>Page</u>
NOMENCLATURE	8
ABSTRACT	9
I. INTRODUCTION.	9
II. DESCRIPTION OF IHX CONFIGURATION	11
III. SURVEY OF LIQUID-METAL HEAT TRANSFER IN IHX	12
A. Thermophysical Properties	12
B. Flow Conditions	12
C. Cross-sectional Temperature Gradients	13
1. Tube-side Sodium	13
2. Shell-side Sodium	13
D. Axial Heat Diffusion.	14
E. Concept of Heat-transfer Coefficient.	14
F. Influence of Tube Wall	15
IV. MATHEMATICAL MODEL	16
A. Assumptions Basic to IHX Model	16
B. Equations.	16
1. Sodium	16
2. Tube Wall	17
3. Heat Transfer: Sodium to Tube Wall	18
4. Boundary Conditions	19
V. COMPUTATIONAL PROCEDURES.	20
A. Motivation for Use of S/360 CSMP	20
B. Survey of Solution Methods	21
1. CTDS Methods	22
2. CSDT Methods	23
C. Difference Equations	23
1. Sodium	23
2. Tube Wall	24
3. Boundary Conditions	24

TABLE OF CONTENTS

	<u>Page</u>
D. Iterative Method of Solution.	25
1. Parallel Method	25
2. Serial Method.	25
E. Improved Iteration Scheme	26
VI. COMPUTER PROGRAM.	28
A. Basic CSMP	28
B. Function-Store	30
C. CSMP Output Control	31
D. Error Control.	31
VII. SAMPLE RESULTS.	33
A. Prediction of IHX Responses	33
1. Comparison of Steady-state Calculations with Measurements	33
2. Transient Responses of IHX Model.	34
3. Influence of Temperature Dependence of Thermophysical Properties	35
B. Computational Efficiency and Convergence Rate	36
1. Convergence Rate	36
2. Error Resetting	37
3. Computation and Real-time Comparisons	37
VIII. DISCUSSION AND CONCLUSIONS	39
APPENDIXES	
A. Derivation of Equations	42
1. Basic Equations	42
2. Sodium	43
3. Tube Wall	45
4. Heat Transfer from Sodium to Tube Wall	46
B. Approximate Analytical Solution of Linear Fourier Heat- diffusion Equation	47
1. Linear One-dimensional Heat-diffusion Equation	47
2. Solution of Steady-state Equation	48
3. Solution of Variation Equation	48

TABLE OF CONTENTS

	<u>Page</u>
C. Short Description of Main Features of S/360 CSMP	53
1. Introduction	53
2. Application-oriented Concepts	53
3. User-defined Functions	55
4. Run Control	56
5. Machine-produced Documentation	56
D. Thermophysical Properties of Sodium and Type 304 Stainless Steel	57
1. Sodium	57
2. Type 304 Stainless Steel	58
E. Parameters of Sample IHX's	59
1. System Parameters	59
2. Thermohydraulic Conditions	59
3. Heat-transfer Coefficients	59
F. Preparation of Computer Program for a Particular IHX Simulation	61
1. Data Statements	61
2. Execution-control Statements	62
3. Output-control Statements	63
4. Remarks on Computer-program Alterations	63
G. Computer-program Listing	65
1. Basic CSMP Program Segment	65
2. Typical Function-Store Subroutine	71
3. Initialization of Output-control Parameters	76
4. Output-control Subroutine	76
5. Dynamic Error-resetting-control Subroutine	76
6. Revised CSMP MAINEX and SIMOUT Subroutines	77
7. Program for Retrieval of Previously Calculated Results . .	82
ACKNOWLEDGMENTS	83
REFERENCES	84

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	Intermediate Heat Exchanger (IHX) for LMFBR.	11
2.	Representation of IHX Model	16
3.	Typical Spatial Subdivisions of IHX.	23
4.	Schematic Block Diagram of Difference Equations in Parallel Form.	25
5.	Schematic Block Diagram of Difference Equations in Iterative (serial) Form	26
6.	Upper Part of Modified Block Diagram of Difference Equations in Iterative Form.	26
7.	Simplified Flow Diagram of Overall Computer Program.	28
8.	Simplified Flow Diagram of Function-Store Subroutines (FGEN). .	30
9.	ALCO-IHX Shell-side (primary) Temperature Profile at 20% Power	33
10.	ALCO-IHX Shell-side (primary) Temperature Profile at 40% Power	33
11.	ALCO-IHX Shell-side (primary) Temperature Profile during Transient Condition T_1	34
12.	ALCO-IHX Tube-side (secondary) Temperature Profile during Transient Condition T_1	34
13.	ALCO-IHX Shell-side (primary) Temperature Profile during Transient Condition T_2	35
14.	ALCO-IHX Tube-side (secondary) Temperature Profile during Transient Condition T_2	35
15.	ALCO-IHX Model Response to Conditions Imposed as Transient T_1	35
16.	ALCO-IHX Model Response to Conditions Imposed as Transient T_2	35
17.	Influence of β on Convergence Rate	36
18.	Example of Typical Convergence Sequence	37
19.	Effect of Size of Time Increment on Calculated Values of Sodium Outlet Temperatures on Shell Side	38

LIST OF TABLES

<u>No.</u>	<u>Title</u>	<u>Page</u>
I.	Correlations for Heat-transfer Coefficients for Liquid-metal Heat Transfer in Tubes and Tube Bundles under Conditions of Uniform Heat Flux and Fully Established Velocity and Temperature Profiles.	18
II.	Evaluation of Computational System Performance; User's Point of View	20
III.	Effect of Dynamic Error Resetting on Computation Time for the Condition That the Final Value of Relative Error = 10^{-4}	37

NOMENCLATURE

Symbol	Description	Units	Symbol	Description	Units
A	Cross-sectional flow area	ft ²	x	Axial coordinate in direction of tube-side sodium flow	ft
A ₁ , B ₁ , C ₁	Coefficients in Eq. 11	°C	z	Axial coordinate in direction of shell-side sodium flow	ft
A ₂ , B ₂ , C ₂	Coefficients in Eq. 12	°C	α	Thermal diffusivity (k/ρC _p), or parameter in correlation for Nu	ft ² /hr
a	Inner radius of tube	ft	β	Parameter in correlation for Nu, or parameter influencing convergence rate (Fig. 6)	-
b	Outer radius of tube	ft	γ	Parameter in correlation for Nu	-
C _p	Specific heat at constant pressure	Btu/lb-°F	δ	Thickness of boundary layer	ft
D _e	Equivalent diameter of hexagonal single tube element, or hydraulic diameter (4A/P _p)	ft	ε _H	Eddy diffusivity for heat transfer	ft ² /hr
D _i	Inner diameter of tube (ID/12)	ft	ε _M	Eddy diffusivity for momentum transfer	ft ² /hr
D _o	Outer diameter of tube (OD/12)	ft	θ	Angle between axes of tubes and direction of flow	degrees
F _a , F _b , F _{ab}	Coefficients in Eq. B.19	ft ²	μ	Dynamic viscosity	lb/ft-hr
h	Heat-transfer coefficient (film)	Btu/ft ² -hr-°F	ν	Kinematic viscosity (μ/ρ)	ft ² /hr
ID	Inner diameter of tube	in.	ρ	Density	lb/ft ³
I ₀	Modified Bessel function of order zero and first kind	-	τ	Time constant	hr
K ₀	Modified Bessel function of order zero and second kind	-	ø	Rotating coordinate	degrees
k	Thermal conductivity	Btu/ft-hr-°F	ø ₁	Unit velocity potential at rear stagnation point of tube	ft
L _{eff}	Effective heat-transfer length of IHX	ft	ψ̄	Average value of ratio of ε _H and ε _M	-
N	Number of tubes	-	Subscripts		
Nu	Nusselt number	-	i	Inlet	-
OD	Outer diameter of tube	in.	max	Maximum	-
P	Pitch	in.	o	Outlet	-
P _p	Wetted perimeter of flow area A	ft	r	Refers to radial direction	-
p	Transform operator (p ² = s/α _w)	ft ⁻¹	s	Shell side	-
Pe	Peclet number (D _e VρC _p /k)	-	t	Tube side	-
Pr	Prandtl number (C _p μ/k)	-	v,max	Maximum fluid velocity based on minimum flow area between tubes in perpendicular direction	ft/hr
Q	Rate of heat flow to tube wall per unit volume of fluid	Btu/hr-ft ³	w	Tube wall	-
q	Heat flux to tube-wall surface	Btu/hr-ft ²	ws	Tube-wall surface on shell side	-
R	Coefficient in Eqs. 9 and 10	hr-ft ² -°F/Btu	wt	Tube-wall surface on tube side	-
r	Radial coordinate	ft	x	Refers to axial direction	-
R _a , R _b	Coefficients in Eqs. 3 and 4	hr-ft ² -°F/Btu	ø	Refers to tangential direction	-
Re	Reynolds number (D _e ρV/μ)	-	Superscripts		
s	Laplace transform operator	hr ⁻¹	k	Number of iterations	-
T	Temperature	°F	n	To indicate t = nΔt	-
ΔT	Variation of temperature from steady-state value	°F	0	Steady-state condition	-
t	Time	hr	'	To indicate normalized coordinate	-
Δt	Time increment	hr			
V	Average fluid velocity	ft/hr			
W	Sodium mass flow rate	lb/hr			

CSDT Simulation Model for
Prediction of IHX Transient Behavior
Using S/360 CSMP

by

J. F. L. M. Brukx

ABSTRACT

This report describes a digital simulation model to predict the transient behavior of sodium-to-sodium counterflow heat exchangers. The model is programmed for an IBM S/360 Model 75 computer using the S/360 Continuous System Modeling Program (CSMP) as the basic programming tool.

The equations of the mathematical model are solved through use of the serial or Continuous-space/Discrete-time (CSDT) method; the nature of the problem thereby requires an iterative solution search for each instant of time. Axial temperature distributions of sodium and tube-wall surface are obtained at every time step.

It is found that CSMP provides a convenient means for implementing the simulation when certain extensions and modifications to standard usage are used. Results that illustrate model capabilities and matters of computational efficiency are presented for two sample intermediate heat exchangers (IHX's).

I. INTRODUCTION

Design development of liquid-metal-cooled fast breeder reactors (LMFBR's) to be used as large-scale sources of electrical energy is being vigorously pursued in this country and abroad. Among the many areas in which the present technological body of information needs expansion is that of total plant dynamic behavior. A thorough understanding of the dynamic behavior of the plant is of importance to the designer and operator, and is essential to safety evaluations. Since the plant comprises a number of subsystems and components, understanding the aggregate is contingent upon knowledge of each relevant part and its interactions with the integrated system. It is therefore essential to develop means for performing accurate and detailed analyses of major components.

In this study, a detailed simulation model was developed for studying transient responses of sodium-to-sodium intermediate heat exchangers (IHX's)

to induced changes in coolant flow rates and/or temperatures. Two principal considerations or objectives were inherent in the development: (1) the derivation of a mathematical model that adequately characterizes significant physical processes, and (2) the selection of a computational procedure that tends to maximize retention of the characterization provided in the mathematical model, operates efficiently, and is flexible in usage. Furthermore, it was highly desirable that the method developed for the study of IHX's be well suited for similar application to sodium-to-water once-through steam generators.

The IHX model developed here deviates from a more conventional one in several respects. Means for treating the temperature dependence of the thermophysical properties of sodium and for adjusting the thermal conductivity of stainless steel (tube wall material) in the axial direction are included. Provision is also made for a reasonably detailed distributed-parameter characterization of heat conduction in the tube walls that separate the counterflowing coolant regions. At the tube-wall/coolant interface, the effects of different flow regions (axial, cross, or oblique flow) and changes in correlations for heat-transfer coefficients may be easily studied.

Whereas most earlier studies of heat-exchanger dynamics have used Continuous-time/Discrete-space (CTDS) techniques for computation, in this study a serial or Continuous-space/Discrete-time (CSDT) method is used. Although CSDT methods have grown largely with the advent and increasing usage of hybrid computers, it was decided, subsequent to a consideration elaborated later in this report, to implement the calculations purely by digital means. The IBM S/360 Continuous System Modeling Program (CSMP) was chosen as the basic coding language.¹

Certain additions and extensions to standard usage of the basic CSMP were necessary or desirable for the solution of the problem considered here. Chief among these are added function-store routines required to solve the partial differential equations involved. Nonstandard features or extensions to CSMP are described in some detail below.

Primary inputs to the model are time-dependent changes in sodium-inlet temperature and flow rate on either the tube or shell side of the IHX. These changes may be made singly or in any prescribed combinations.

Calculated IHX responses are obtained in the form of sodium and tube-wall surface-temperature distributions at selected time intervals; temperatures corresponding to different steady-state operating conditions can also be calculated directly. Computer output for the results can be obtained in printed form or as CALCOMP plots. Some sample results with respect to model capability and computational efficiency are presented.

II. DESCRIPTION OF IHX CONFIGURATION

The mathematical model and computer program developed here can be used to predict transient temperatures in counterflow, shell, and tube IHX's, for which the heat-transfer interior consists of straight tubes with tube sheets at each end. Such heat-exchanger designs are typical of those for use in LMFBR's. Examples of existing or planned heat exchangers of this general design type are: IHX of EBR-II,² ALCO IHX of SCTI,³ IHX of FFTF,⁴ and several concepts resulting from the 1000-MWe LMFBR Follow-on Studies.⁵⁻⁷ Figure 1 depicts the salient design features of an IHX intended for use in 1000-MWe LMFBR's.

Fig. 1

Intermediate Heat Exchanger (IHX)
for LMFBR. ANL Neg. No. 900-146.

III. SURVEY OF LIQUID-METAL HEAT TRANSFER IN IHX

Typical geometries of the IHX's described earlier in this report cause complex flow patterns and heat-transfer conditions. Analytical descriptions of the basic physical phenomena are facilitated when certain mathematical idealizations and simplifications are made.

Studies of the dynamics of heat exchangers that use nonmetallic fluids have provided a spectrum of simplifying assumptions that have enabled the development of tractable mathematical models for their engineering analysis.

The main content of this chapter is a discussion of heat-transfer-related simplifications and idealizations suitable for liquid metals as compared to those acceptable for nonmetallic fluids.

A. Thermophysical Properties

The general thermophysical properties of liquid metals (for example, sodium and NaK) can be characterized as follows: relatively low melting point, high boiling point, low vapor pressure, low viscosity, high thermal conductivity, and reasonably good specific heat and density values.⁸ The thermophysical properties of nonmetallic liquids are normally taken to be independent of temperature, time, and position. In contrast, the relatively large temperature range over which liquid metals remain liquid indicates the possible necessity of accounting for temperature dependence of their thermophysical properties.

Stein⁹ reports that the need for treatment of temperature dependence of liquid metals appears most likely for "convection with 'small' Reynolds numbers." For the analysis of liquid-metal behavior, under conditions of turbulent forced-convection flow, investigators have generally been content to ignore their temperature dependence.

In an IHX, large temperature transients may occur for large flow unbalances (e.g., full flow on tube side and small flow on shell side); these conditions can result from mechanical failure of, or loss of electrical power to, a pump. Under these conditions, changes in thermophysical properties with temperature might be important.

B. Flow Conditions

Forced-convection flow in LMFBR IHX's generally results in turbulent conditions ($Re > 4000$). This is due to the small values of viscosities of liquid metals and the functional requirements of an IHX for proper heat removal. Thus, the velocity distribution of the fluid in the radial direction at any cross-section location can be assumed essentially uniform. This assumption becomes invalid only for an operating condition far removed from the IHX design point, e.g., natural convection for which the fluid flow may become laminar.

C. Cross-sectional Temperature Gradients

1. Tube-side Sodium

During turbulent flow, the fluid profile divides, in general, into a laminar sublayer near to the wall, a transition layer, and a turbulent core.¹⁰ To describe heat-transfer conditions, a thermal sublayer can be considered with a thickness larger or smaller than the laminar sublayer, depending on the value of the Prandtl number as follows:

$$\text{Pr} = f\left(\frac{\delta_{\text{laminar}}}{\delta_{\text{thermal}}}\right) = \frac{\nu}{\alpha}; \quad \frac{\delta_{\text{laminar}}}{\delta_{\text{thermal}}} \rightarrow \begin{cases} >1 & \text{if } \text{Pr} > 1 \\ <1 & \text{if } \text{Pr} < 1. \end{cases}$$

For nonmetallic fluids ($\text{Pr} > 1$), heat transfer by molecular conduction dominates in the thermal sublayer, whereas diffusion dominates in the turbulent core. This results in the localization of significant temperature gradients immediately adjacent to the duct wall. Therefore, for nonmetallic fluids, it is usually assumed that the temperature distribution in the fluid is uniform over each cross section, and changes only as a function of time and axial location. This permits a one-dimensional treatment of the problem.

For liquid metals, $\text{Pr} \ll 1$ and the thermal conductivity is large; molecular conduction is significant, even in the turbulent core. Thus, significant temperature gradients can exist over the entire cross section. According to Stein,⁹ one important consequence of this difference between liquid metals and nonmetallic liquids in turbulent flow is that heat-transfer conditions for the former are relatively more sensitive to changes in boundary conditions (local changes at the duct wall) and to duct shape (see also Sec. III.E).

2. Shell-side Sodium

The prediction of cross-sectional temperature gradients on the shell side of an IHX is much more difficult due to parallel channel interactions; this is especially true for baffled heat exchangers. Even without baffles, the analyses are complicated due to the existence of several flow regions, e.g., crossflow near the inlet and outlet, and axial flow for a distance in between. Such geometric complications essentially preclude a purely analytical approach, and reliance is placed on experimental results through which expressions for heat-transfer coefficients are developed that correlate with the data. Generally these expressions or "correlations" are based upon a sufficiently varied set of conditions such that certain parameters may be related to different geometries. However, the remaining uncertainties are large. Thus, it is common also to avoid undue complexity in attempting to describe tube-side temperature distributions in the radial direction for engineering-type calculations.

D. Axial Heat Diffusion

The relative influence of axial (longitudinal) heat diffusion compared to axial convection is characterized by the reciprocal of the Peclet number as follows:

$$\frac{1}{Pe} = \frac{\text{axial conduction}}{\text{axial convection}} = \frac{1}{Pr \cdot Re}.$$

This indicates that axial conduction is generally negligible for $Pe \gg 1$.

For turbulent-flow conditions in nonmetallic fluids, Peclet numbers are rarely less than 10^4 ; therefore, axial heat diffusion can be neglected.⁹ With liquid metals, however, the magnitude of the Peclet number is more likely to lie within the range in which axial-conduction effects become important. Schneider¹¹ has found that for the purpose of computing overall heat-transfer rates, axial conduction can be neglected for $Pe \geq 100$, and for the local Nusselt numbers involved, axial conduction had very little effect. Note, however, that these results were obtained for special boundary conditions and that generalization may not be permitted without considering the conditions involved.⁹

E. Concept of Heat-transfer Coefficient

For nonmetallic fluids the "traditional" concept of a heat-transfer coefficient is used, even for transient conditions and nonuniform heat fluxes. The familiar relation is

$$q = h(T_{Na} - T_{surf}).$$

Heat-transfer coefficients are considered relatively insensitive to boundary conditions (temperature or heat-flux distribution), duct shape, and distance from duct inlet. As already mentioned above, liquid-metal heat transfer is more sensitive to these variables.⁹

Most of the experimental correlations for the heat-transfer coefficients of liquid metals have been determined for certain specific conditions, such as steady state, fully developed flow, uniform heat flux or wall temperatures, specific geometry, and flow conditions (parallel, cross, or oblique flows). For most practical design or performance-related calculations, however, there are factors that may significantly affect results, such as thermal entrance region at the inlet and outlet of the heat exchanger; different boundary conditions; different geometry (symmetrical or nonsymmetrical); and transient condition.

Stein⁹ and Brown¹² state that inaccuracies will result if uniform heat-transfer coefficients are used to predict overall heat-transfer rates. For transient conditions and nonuniform heat-flux distributions in an axial direction,

Stein¹³ proposed a new, improved method of heat-transfer prediction for use in engineering design and analysis.

For LMFBR heat exchangers, however, where material selection and wall thickness of the tubes are determined by structural and safety considerations, the thermal resistance of the tube wall is large compared with the thermal resistance of the fluid film ($1/h$). As a result, inaccuracies in h will have little effect on total heat-transfer behavior. Thus, the relative thermal resistance of the tube wall is a significant factor in the heat-transfer process.

F. Influence of Tube Wall

For liquid-metal heat transfer through a stainless steel tube wall, the relative thermal resistance of the wall is large (about one-half of the total). This is due to the large heat-transfer coefficients of liquid metals and the relatively large thickness and low thermal conductivity of the stainless steel tube wall. This results in a large temperature drop over the tube wall compared to the temperature difference between sodium and tube-wall surface. Thus, an accurate description of the influence of the tube wall is more important than for nonmetallic fluids, especially when it is desired to obtain reasonable information about the temperature distribution, both in radial and axial directions of the tube wall itself. For transients with large, unbalanced flow conditions, the temperature dependence of the thermophysical properties of the tube wall may be important. Because of the relatively small temperature drop ($<100^{\circ}\text{F}$) in the radial direction, compared to the total temperature change in the axial direction ($\sim 400^{\circ}\text{F}$), an adjustment of the physical properties only in the axial direction may be sufficient.

IV. MATHEMATICAL MODEL

Based upon the following idealizations, a description of the multielement (tubes and associated sodium) IHX can be reduced to a single element as represented in Fig. 2.

Fig. 2. Representation of IHX Model.
ANL Neg. No. 900-147.

Assumption 1 is justified because of the small ratios of heat-transfer area of the shell or central tube to the total heat-transfer area of the tubes, and their relatively small heat capacity as compared to that of the sodium and tubes. Thus, for most transients of interest, the effects of the central tube and shell have negligible influence on the average temperature distributions within the IHX.

Remaining assumptions used to derive the equations that describe temperature distributions in an average element of the IHX are given in Appendix A.

B. Equations

The basic equations of the model are developed in Appendixes A and B; they are summarized below for each region of the IHX element of Fig. 2.

1. Sodium (shell and tube side)

Combination of mass- and energy-balance equations yields a pair of first-order, nonlinear partial differential equations that describe temperatures in the shell- and tube-side sodium (see Appendix A):

$$\rho_t C_p A_t \left(\frac{\partial T_t}{\partial t} + \frac{1}{\tau_t} \frac{\partial T_t}{\partial x'} \right) + P_t q_t = 0; \quad (1)$$

A. Assumptions Basic to IHX Model

1. Heat exchange between sodium and shell, and, if a central tube is present, between sodium and central tube, is negligible.
2. Use of appropriate correlations for heat-transfer coefficients permits the treatment of baffled (oblique) flow as being purely axial in direction.
3. The transient behavior of an average tube is considered for the purpose of this study.

$$\rho_s C_p A_s \left(\frac{\partial T_s}{\partial t} - \frac{1}{\tau_s} \frac{\partial T_s}{\partial x'} \right) + P_s q_s = 0, \quad (2)$$

where

$$x' = \frac{x}{L_{\text{eff}}};$$

$$\tau_t = \frac{L_{\text{eff}}}{V_t}; \quad \tau_s = \frac{L_{\text{eff}}}{V_s};$$

$$V_t = \frac{W_t}{N \rho_t A_t}; \quad V_s = \frac{W_s}{N \rho_s A_s}.$$

Note: Only the space variable x has been normalized.

2. Tube Wall

To describe transient temperature distributions within the tube wall, the linear Fourier heat-diffusion equation is used. This provides increased accuracy over the conventional lumped-parameter equations. However, to avoid undue complexity in the model, the approximate analytical solution technique developed by DeGreef¹⁴ is used to solve the Fourier equation; this technique is presented in Appendix B. By this method the following two first-order ordinary differential equations that describe the time-dependent inner- and outer-surface temperatures of the tube at each axial location are obtained:

$$\tau_{wt} \frac{dT_{wt}}{dt} + T_{wt} - \tau_w R_a \frac{dq_t}{dt} - R_a q_t - T_{ws} = 0; \quad (3)$$

$$\tau_{ws} \frac{dT_{ws}}{dt} + T_{ws} - \tau_w R_b \frac{dq_s}{dt} - R_b q_s - T_{wt} = 0, \quad (4)$$

where

$$\tau_{wt} = \frac{b^2 - a^2 - 2a^2 \ln \frac{b}{a}}{4\alpha_w}; \quad \tau_{ws} = \frac{a^2 - b^2 + 2b^2 \ln \frac{b}{a}}{4\alpha_w};$$

$$\tau_w = \frac{a^2 + b^2 - \frac{b^2 - a^2}{\ln \frac{b}{a}}}{4\alpha_w};$$

$$R_a = \frac{a}{k_w} \ln \frac{b}{a}; \quad R_b = \frac{b}{k_w} \ln \frac{b}{a}.$$

3. Heat Transfer: Sodium to Tube Wall

Heat transfer between sodium and tube-wall surface is described by the traditional concept of heat-transfer coefficient:

$$q_t = h_t(T_t - T_{wt}); \quad (5)$$

$$q_s = h_s(T_s - T_{ws}). \quad (6)$$

Correlations for h used in these relationships are of the following general form:¹⁵

$$h = \frac{kNu}{D_e}; \quad Nu = \alpha + \beta(\bar{\psi}Pe)^\gamma.$$

Appropriate values of α , β , γ , and $\bar{\psi}$ for both tube and shell side are given in Table I. The heat-transfer coefficients are functions of sodium flow rate and temperature.

TABLE I. Correlations for Heat-transfer Coefficients for Liquid-metal Heat Transfer in Tubes and Tube Bundles under Conditions of Uniform Heat Flux and Fully Established Velocity and Temperature Profiles

Correlation No.	Ref.	Range of Investigation of Application	Correlation for Nusselt Number	Remarks
<u>Tube Side</u>				
1A	15, 16	$Pe < 150$ $0.005 < Pr < 0.03$	$Nu = 7.0$	$\bar{\psi} = 1 - \frac{1.82}{Pr(\epsilon_M/\nu)_{\max}^{1.4}}$
1B	15, 16	$150 < Pe < 10^4$	$Nu = 7.0 + 0.025(\bar{\psi}Pe)^{0.8}$	
2	17	$40 < Pe < 1150$	$Nu = 5.0 + 0.025Pe^{0.8}$	
	18	$0.0057 < Pr < 0.0075$		
<u>Shell Side (Axial-flow Region)</u>				
3A	18, 19	$Pe < 700$ $0.005 < Pr < 0.03$ $1.3 < P/D < 3.0$	$Nu = -2.79 + 3.97(P/D) + 1.025(P/D)^2$ $+ 3.12 \log_{10} Re - 0.265(\log_{10} Re)^2$	$P \equiv \text{pitch}; D = OD$
3B	18, 19	$700 < Pe < 10^4$	$Nu = 6.66 + 3.126(P/D) + 1.184(P/D)^2$ $+ 0.0155(\bar{\psi}Pe)^{0.86}$	
<u>Shell Side (Crossflow Region)</u>				
4	19	Whole Pe range	$Nu = (\phi_1/D)^{0.5} \left(\frac{P - D}{P} \right)^{0.5}$ $\cdot (5.24 + 0.225Pe_{v,\max}^{0.635})$	
<u>Shell Side (Oblique Flow Region)</u>				
5	19	Whole Pe range $30^\circ < \theta \leq 90^\circ$	$Nu = (\phi_1/D)^{0.5} \left(\frac{P - D}{P} \right)^{0.5} \frac{\sin \theta + \sin^2 \theta}{1 + \sin^2 \theta}$ $\cdot (4.60 + 0.193Pe_{v,\max}^{0.614})$	

4. Boundary Conditions

Because of the counterflow design of the heat exchanger, boundary conditions are defined at two different locations as follows:

$$T_t(0,t) = T_{ti}(t) \quad (7)$$

and

$$T_s(1,t) = T_{si}(t). \quad (8)$$

V. COMPUTATIONAL PROCEDURES

Because analytical solutions of the present problem cannot be obtained, numerical approximations to the solutions must be found. The selection of a computer system used to solve the equations of a mathematical model is generally determined by the following factors:

1. Available computer systems.
2. Performance of the computational system from the user's point of view.
3. Nature of problem and study objectives.

A. Motivation for Use of S/360 CSMP

The computer systems available at ANL are (1) an analog computer with ~300 amplifiers; (2) a hybrid computer, consisting of the same analog computer linked with an IBM-1130 digital computer by means of an interface designed and built at Argonne; and (3) a digital computer, S/360 Model 75, that provides the availability of S/360 CSMP.

The main characteristics of each computing system analog, digital, and hybrid, are described in Refs. 20 and 21 and will not be delineated here. Because of the selection of S/360 CSMP, its main features are summarized in Appendix C. More detailed descriptions can be found in Refs. 1 and 22.

Several comparative studies have been done to evaluate the merits of various simulation techniques, such as digital (conventional FORTRAN) analog, hybrid, and CSMP.^{21,23,24} Conclusions obtained from these studies are summarized in Table II. Following Shapiro,²¹ two distinct phases of systems performance are considered: a problem-setup/debug phase, and a problem-execution phase.

TABLE II. Evaluation of Computational System Performance; User's Point of View

	Analog	Digital		Hybrid	Remarks
		FORTRAN	CSMP		
Setup/Debug Phase					
Translation of problem into computer language	Block diagram	Sequential language	Block diagram or equations	Block diagram and sequential language	
Computer-usable input	Patch board	Punched cards	Punched cards	Patch board and cards	Convenient for program storage.
Scaling necessary?	Yes	No	No	Yes	
Number of statements	-	Large	Small	Smaller	
Debugging of test runs	Difficult	Easy	Easy	Very difficult	Diagnostic messages are provided.
Modification of working program	Difficult	Easy	Easy	Difficult	Machine-produced documentation available.
Setup/debug time	Long	Long	Short	Very long	
Execution Phase					
Computation time	Short	Long	Long	Short	
Accuracy of results	Reasonable	High	High	Better than analog possible	Depends also on error analysis.
Error analysis	Difficult	Numerical analysis necessary	Built-in error-specification features	Very difficult	
Input/output features					
Machine-produced documentation	Poor	Good	Good	Good	
Man-machine interface	Good	Depends on peripheral devices; generally poor		Good	

Evaluation of the relative advantages of different simulation techniques, with respect to factors such as total solution time and total cost for solution, depends on the number of production runs. Chubb²⁴ has studied economic tradeoffs between various simulation methods as compared to S/360 CSMP for three different cases. His results indicate that for applications requiring a small number of runs and fast turnaround time, S/360 CSMP is likely to be engineer's best choice. He also found, for every case, that the use of FORTRAN instead of S/360 CSMP was more time-consuming and costly. For applications requiring a large number of runs, the hybrid computer and/or the straight analog computer are less expensive to use than S/360 CSMP.

Based upon the objectives of this study, the following factors influenced the selection of a computational system:

1. As a reference model, the IHX simulation should be flexible in usage (input/output flexibility, convenient program storage).
2. It should be possible to easily make those modifications in the working program that are appropriate for a particular IHX.
3. Debugging and diagnostic features for programming errors are important.
4. Frequent modifications of the program require good, machine-produced documentation.
5. The anticipated small number of runs for each simulation study reduces the importance of execution time as compared to setup/debug time.
6. Results should be accurate and reliable. In particular, restraints due to lack of basic computing tools should essentially be avoided.

Comparing these factors with those relative to CSMP performance, we can conclude that CSMP is probably the best choice for this simulation study.

In addition, the "hybrid-like" programming character of S/360 CSMP is considered to be an advantage in recognition of possible needs for actual implementation of the IHX model on a hybrid computer for purposes such as parameter optimization and total plant studies. For such applications, use of a hybrid computer should be preferable due to better man-machine interaction and better economic efficiency due to the anticipated requirements for a large number of runs. Even for such cases, the CSMP model can serve the useful purpose of providing check calculations to verify the accuracy of hybrid solutions.

B. Survey of Solution Methods

Since CSMP may be viewed conceptually as an analog or hybrid computing system, the alternative methods available may be categorized as

1. Continuous-time/Discrete-space (CTDS) approach;
2. Continuous-space/Discrete-time (CSDT) approach.

1. CTDS Methods

CTDS methods use spatial differencing, and include the traditional analog parallel approach and the hybrid, cell-multiplexing technique. The former has potential for high-speed execution and is the only known method that obviates the need for iterative computation of the IHX equations, whereas the latter is conservative in hardware requirements. Only the parallel approach is of interest if CTDS methods are used in conjunction with CSMP.

Important factors to be considered in the application of the CTDS parallel approach are (1) the influence of the finite-difference representation chosen (forward, central, backward, or other differencing schemes), and (2) the number of finite-difference cells or nodes selected. Comparative studies of the influence of these factors were performed by Wilson²⁵ and very recently by Gunby²⁶ for the IHX simulation of FFTF. Wilson compared the frequency-response characteristics of IHX outlet temperatures to disturbances on the shell-side (hot sodium) inlet temperatures. Gunby compared outlet temperatures responses to: flow transients, inlet temperature ramps, combined flow and inlet temperature changes, and steady-state axial temperature distributions at unbalanced flows.

Results of these studies show that the largest differences between calculated outlet temperatures for the different methods occur on the side having the highest flow rate. Gunby anticipated this result, reasoning that the error in the approximation involved in the $\partial T / \partial x$ term is magnified by high flow rates. Results for steady-state conditions at unbalanced flow (10:1) indicate, that for all the models investigated, the internal axial temperature distributions are not reliable, even if reasonable outlet temperatures are obtained.

Gunby's study indicates that when both reliable axial temperature distributions and outlet-temperature response are required for large disturbances to unbalanced flow conditions, it is difficult to develop a reliable model using the CTDS approach.

With CTDS methods, the required spatial subdivision of IHX in the axial direction generally must be prescribed a priori; selection depends on the temperature gradients in the sodium. However, these gradients vary as functions of space and time during transient conditions; thus, the spatial subdivision should be appropriately adjusted during computation. For the simulation of steam generators, further adjustment is desirable, since the locations of transition boundaries that separate different heat-transfer regimes also change as functions of time during transients.

2. CSDT Methods

CSDT methods "avoid" approximations to the $\partial T / \partial x$ term. Used in conjunction with the CSMP method, they conveniently provide for dynamic adjustment of spatial subdivisions when variable-step-size integration routines are used. This feature is illustrated by results presented in Fig. 3, in which are clearly indicated the differences in appropriate spatial subdivisions that occur for axial sodium-temperature profiles for two different steady-state conditions.

Fig. 3. Typical Spatial Subdivisions of IHX. ANL Neg. No. 900-148.

For any transient by a CSDT method, the proper choice or adjustment of a discrete time interval can be more easily accomplished than can the associated treatment of spatial subdivisions using a CTDS method.

Many methods have been developed in the CSDT category.²⁷⁻³¹ The utility of each is problem-dependent, especially with respect to items such as computational stability, error propagation, and convergence.

A choice between the CTDS and CSDT approaches is strongly influenced by a general principal objective of this work, namely, to maximize retention of the treatment of physical phenomena inherent in the original differential equations. Due to the above considerations, a CSDT approach has been chosen for this study. In particular, the work of Carling³⁰ has served as a guide, and the basic principles of his approach have been adopted here.

C. Difference Equations

With the CSDT approach, the describing equations of the mathematical model given earlier transform to a set of difference equations. By using a two-point backward-differencing scheme with respect to time, and eliminating q_t and q_s in Eqs. 1-4 by substituting Eqs. 5 and 6, Eqs. 1-6 are replaced by Eqs. 9-12 below. In the latter, the superscript n is an integer used to indicate the approximate value of each associated variable for each increment of time; that is, $t = n\Delta t$.

1. Sodium (shell and tube side)

$$\frac{dT_t^n}{dx'} = -R_t h_t (T_t^n - T_{wt}^n) - \frac{\tau_t}{\Delta t} (T_t^n - T_t^{n-1}) \quad (9)$$

and

$$\frac{dT_s^n}{dx'} = R_s h_s (T_s^n - T_{ws}^n) + \frac{\tau_s}{\Delta t} (T_s^n - T_s^{n-1}), \quad (10)$$

where

$$R_t = \frac{P_t NL_{eff}}{C_p W_t} \text{ and } R_s = \frac{P_s NL_{eff}}{C_p W_s}.$$

2. Tube Wall

$$A_1 T_{wt}^n - B_1 T_t^n - T_{ws}^n = C_1 \quad (11)$$

$$A_2 T_{ws}^n - B_2 T_s^n - T_{wt}^n = C_2, \quad (12)$$

where

$$A_1 = \left(1 + \frac{\tau_{wt}}{\Delta t}\right) \left(1 + R_a h_t^n\right);$$

$$A_2 = \left(1 + \frac{\tau_{ws}}{\Delta t}\right) \left(1 + R_b h_s^n\right);$$

$$B_1 = \left(1 + \frac{\tau_w}{\Delta t}\right) R_a h_t^n;$$

$$B_2 = \left(1 + \frac{\tau_w}{\Delta t}\right) R_b h_s^n;$$

$$C_1 = \frac{\tau_{wt}}{\Delta t} + \frac{\tau_w}{\Delta t} R_a h_t^{n-1} (T_{wt}^{n-1} - T_t^{n-1});$$

$$C_2 = \frac{\tau_{ws}}{\Delta t} + \frac{\tau_w}{\Delta t} R_b h_s^{n-1} (T_{ws}^{n-1} - T_s^{n-1}).$$

3. Boundary Conditions

As stated earlier, specified boundary conditions are available only at two different points of the heat exchanger, as follows:

$$T_t(0, t) = T_{ti}(t) \quad (7)$$

and

$$T_s(1, t) = T_{si}(t). \quad (8)$$

Figure 4 is a block-diagram representation of the above equations. Separation of the boundary conditions requires that an iterative procedure be used for the numerical solution of the above equations.

Fig. 4

Schematic Block Diagram of Difference Equations in Parallel Form. ANL Neg. No. 900-149 Rev. 1.

D. Iterative Method of Solution

1. Parallel Method

A possible solution technique for the above set of difference equations is to solve all simultaneously, i.e., in parallel, and systematically search for calculated agreement with the known boundary conditions. In practice, however, this procedure is unsatisfactory due to errors introduced by the inherently unstable character of Eq. 10 that results from its integration in a direction opposite to the fluid flow. To bypass this problem, it is necessary to reverse the direction of integration of Eq. 10, achieved by replacing x' by $x' = 1 - z'$ and performing the integrations required on each side of the IHX in a serial manner. Equation 10 then becomes

$$\frac{dT_s^n}{dz'} = -R_s h_s (T_s^n - T_{ws}^n) - \frac{T_s}{\Delta t} (T_s^n - T_s^{n-1}), \quad (13)$$

which is inherently stable.

2. Serial Method

Serial solutions of the revised set of difference equations requires that certain information, generated from calculations made on a given side of the IHX, be stored and made available for calculations on the other side. Store and retrieve capability is provided by function-store routines, added to CSMP as described below.

The difference equations may now be solved, using a parallel calculation of Eqs. 9, 11, and 12 in the x direction, and similar treatment of Eqs. 11, 12, and 13 in the z direction. The general procedure for solving transient problems is carried out in the following manner.

A steady-state condition is established by making an initial guess of a temperature distribution (on the shell side of the IHX) and then solving the difference equations iteratively, according to the scheme indicated in Fig. 5, until steady-state convergence is achieved. To begin a desired transient, time is incremented by a prescribed amount, input values are changed

appropriately, and the previously computed converged values become the new starting trial values of temperature distributions. A new series of iterations is performed until convergence is again achieved. (Note that converged values for one time step earlier are not shown explicitly in Fig. 5.) The entire process is executed repeatedly for the duration of the problem to be solved. Whenever a converged solution is obtained, the representation of the difference equations as given in Fig. 5 reduces to the exact representation as given in Fig. 4.

Fig. 5

Schematic Block Diagram of Difference Equations in Iterative (serial) Form

E. Improved Iteration Scheme

By comparing the iteration procedure with physical phenomena that occur in the IHX, it was considered that, instead of using sodium temperatures calculated on one side to calculate those on the other, convergence rates could be improved by including the parallel interaction between those temperatures. The manner in which this is accomplished is indicated in Fig. 6. The bar over a variable indicates that the variable is retrieved from storage. The superscript k indicates the k th iteration at a particular point in time, $n\Delta t$. As shown in Fig. 6, an adjustment of retrieved values of $\bar{T}_s^k$ is made by comparing T_t^{k+1} to T_t^{*k} , and multiplying their difference by β . Note that T_t^{*k} is computed from the tube-wall equations using retrieved values of shell-side temperatures. The difference between T_t^{k+1} and T_t^{*k} can be considered as a measure of the error in the retrieved $\bar{T}_s^k$, whose influence on T_t^{k+1} is determined by the value of β . Thus, the value of β influences the rate of convergence.

Fig. 6. Upper Part of Modified Block Diagram of Difference Equations in Iterative Form. ANL Neg. No. 900-152.

This procedure has also been incorporated in the lower part of the iteration scheme shown in Fig. 5. Note that when convergence is reached, $T_t^{*k} = T_t^{k+1}$; therefore the solutions are in agreement with those obtainable from the original scheme.

Proper values of β 's, i.e., those that maximize the convergence rate, cannot be determined a priori; some calculations have been made to search for optimal values. In each case, the same value of β was assigned to the upper and lower parts of the iteration scheme. Results are presented later in this report.

VI. COMPUTER PROGRAM

A Basic CSMP

Calculations required to solve the difference equations in conformance to the algorithm described above are accomplished through use of basic features of CSMP used in conjunction with certain additions and modifications thereto. The latter consists of:

1. Function store and retrieve capabilities.
2. Output control options.
3. Dynamic error resetting of integrator output error specifications.

Figure 7 depicts overall problem flow in a simplified block-diagram representation of the main program. As indicated, all major standard functional segments of CSMP are used, viz, DATA, INIT, DYNAM, and TERMIN. The diagram also illustrates the linkage of CSMP to added non-standard routines at various points. The main CSMP program and the added nonstandard routines are listed in Appendix G.

Prior to the DATA segment are MACRO functions that contain statements defining the temperature dependence of the physical properties of sodium and Type 304 stainless steel.* The MACRO provides for convenient acquisition of that information at various points in the program.

Fig. 7. Simplified Flow Diagram of Overall Computer Program

The DATA segment is used in a straightforward manner to supply appropriate values of parameters and constants needed to define the problem. For appropriate preparation of required DATA input information for a particular IHX, see Appendix F.

*Equations of thermophysical properties of sodium and Type 304 stainless steel as a function of temperatures are given in Appendix D.

The INIT segment is used to initialize the problem properly. The information generated there, e.g., inlet temperature and flow-rate perturbations, is appropriately updated as a function of time whenever time is incremented, i.e., upon satisfactory completion of a series of iterations.

The DYNAM segment is divided into two major branches, conforming to the needs of the split integration scheme (see Fig. 6) required by the CSDT method employed. As indicated in the diagram, an indexing variable (NORUN) is used to direct the flow of computations through the appropriate branch. In the context of the method employed, a "run" includes the integration of the difference equations over the total effective length of the heat exchanger. Similarly, an "iteration" involves completion of a pair of runs, one for each branch in the DYNAM segment.

The basic function of each branch of the DYNAM segment is to solve the difference equations in the manner prescribed by the iteration method described above, thereby obtaining solutions for each side of the heat exchanger. These calculations, performed in a "parallel" fashion, use previously computed data that are retrieved from function-store subroutines (FGEN..) provided as additions to CSMP. Note that portions of the data retrieved in one branch are those generated in the other, and, because of the sign opposition in independent variable coordinates (x' , z'), those data must be acquired in a backward sense. Also note that the form of the difference equations requires the availability of data stored at two levels of time (viz, current and history), one time increment earlier.

The iteration scheme depicted in Fig. 6 shows the existence of "algebraic loops" for the tube-wall equations. To avoid the associated computational difficulties, these algebraic loops were eliminated by appropriate variable substitutions such that explicit expressions for tube-wall surface temperatures were obtained.

For each iteration, calculations are made that are used to measure convergence of solutions and the approach of converged solutions to a new steady-state condition. A similar criterion is used for both measures: The convergence measure is the maximum difference between the tube-side sodium temperatures of two successive iterations; the new steady-state measure is the maximum difference between currently converged shell-side sodium temperatures and those obtained one time step earlier. These measures are then used in the TERMIN segment to set appropriate program flags that control subsequent operations of the overall program.

The TERMIN segment is entered upon completion of each run, and principal tests are made with respect to convergence and total problem completion. Generally, during a transient, convergence of solutions is tested twice for each time increment. The first (ERRCR) constitutes a relatively coarse test that is used to selectively call an added subroutine (ERRSET) that controls

the specified error for integrator outputs. When satisfied, the error specification is reduced and used for final iterations in conjunction with a finer convergence criterion (CONVCR). When solutions satisfy the second convergence test, a convergence flag (CONV) is set; CSMP system variables are set by calling subroutine OUTPUT to invoke standard output if desired, and a return to the INIT segment is made for execution of a single run that completes the current iteration. The function-store routines (FGEN..) are then called from TERMIN for push-down of stored data. Checks are made to determine the problem approach to steady-state, finish conditions and the need or desirability of automatic integrator-error control. If continuance is indicated, time is incremented by a specified amount and appropriate flags are set for resuming the iteration process.

B. Function-Store

Function-store subroutines (FGEN..) have been developed to implement the calculations necessitated by the form of the difference equations given above. A simplified representation of basic features of these subroutines is depicted in Fig. 8. As indicated there, functional aspects of the function-store subroutines may be viewed as consisting of two major divisions: data storage and retrieval. Storage and retrieval capabilities extend over three time levels: current, and history at one and two time steps earlier. However, for this problem, only two time levels are needed.

Fig. 8. Simplified Flow Diagram of Function-Store Subroutines (FGEN). ANL Neg. No. 900-27.

"backward" direction, (4) "push" currently calculated data into next area of storage, and (5) "push" all data down into earlier areas of history. The function-store subroutines respond by storing input data if signaled, and returning to CSMP-UPDATE the outputs called for through calculations performed in the retrieval and interpolation areas. Generally, interpolations are made by the second-order Lagrange method, found to be well matched with the fourth-order Runge-Kutta, variable-step-size integration procedure used. "Push-down" operations are carried out when signaled in an appropriate sequence.

In addition to the basic features of the function-store subroutines discussed above, others are available that add to their utility. If a given difference equation is of first order, thereby requiring only one level of history information, a control signal effectively bypasses the second-level facilities of the routine.

Peripheral devices (IBM 2314 DISK unit) may be used to store and later retrieve selected data. This feature is of value, for example, (1) to store steady-state results for various operating conditions so that subsequent problems can be initiated without first performing steady-state search calculations, or (2) to store IHX outlet temperatures for each time step so that, with a small auxiliary program, plots of the outlet-temperature responses can be obtained from a separate, subsequent computer job. An example for such an auxiliary program is given in Appendix G, Sec. 7.

Finally, upon signal, data may be written to a printer or a card punch. The latter feature is useful should subsequent processing of results be desired.

C. CSMP Output Control

It is desirable, in problems of this type, to have a facility for providing output only for converged results of the calculations. Standard CSMP does not provide this feature of output control. Therefore, a method was developed for suppressing and invoking CSMP, PRINT, and PRTPLT output dynamically upon demand. Control is accomplished by means of an added subroutine (OUTPUT), plus minor alterations made in the CSMP routines SIMOUT and MAINEX. Statement changes in the CSMP routines are indicated in the listings of Appendix G by the appearance of nonstandard sequence identifications. For changes made in SIMOUT, the associated sequence information (card columns 73-80) have the form S***iiii; similarly, for MAINEX, the corresponding form is M***iiii.

Problem preparation for output is made in a standard manner, i.e., by means of the CSMP, PRINT, and PRTPLT cards. Output frequency is normal if no calls are made to subroutine OUTPUT; however, the frequency is controlled when such calls are appropriately made from the INIT and/or TERMIN sections of the CSMP program.

In addition to the utility of this method in controlling the quantity of printed output, it also provides editing capability for CALCOMP output. As used here, families of curves representing values of particular temperature distributions are produced on a single CALCOMP graph for any selected set of time intervals.

D. Error Control

A subroutine (ERRSET) has been written that provides means for dynamically changing the relative and/or absolute error specifications for integrator outputs. It uses CSMP system variables to determine the storage locations of specified error values and then changes those values upon signal as determined by the results of subsidiary calculations.

Use of this feature enhances the efficiency of the iteration process substantially when used in conjunction with variable-step-size integration routines. Early in a series of iterations, a relatively large initial convergence criterion and integrator error specification is used in combination. When the initial convergence criterion is satisfied, both are reduced to final desired values. The error specification and the shape of the temperature distribution determine the number of integration steps per iteration. The advantage of the error resetting results from the following: With an initial large error specification of integrator outputs, the number of integration steps (and consequently, the computation time) is reduced. However, with the final error specification, the desired accuracy is preserved in the final results. This reduces the computation time required to drive solutions to the vicinity of end results.

VII. SAMPLE RESULTS

The simulation model developed here has been used for a substantial number of trial conditions representing a variety of different inputs to several IHX designs. These computations were performed (1) to provide general information on overall capabilities of the model to predict IHX responses and (2) to generate data selected to assess accuracy and efficiency.

A. Prediction of IHX Responses

The model developed here permits the calculation of both steady-state and transient IHX conditions. The responses of IHX to several prescribed perturbations of steady-state input conditions were determined. Confirmation of calculated results through comparison with experimental measurements is essential to verification of the model. However, at present only steady-state temperature distribution data have been published for the ALCO-IHX,³ tested in the Sodium Components Test Installation (SCTI) under responsibility of the Liquid Metal Engineering Center (LMEC).^{32,33} Thus, such comparisons can only be made for steady-state conditions as they exist in that IHX.

1. Comparison of Steady-state Calculations with Measurements

Measured data for the ALCO-IHX are available for steady-state conditions at 20 and 40% power.^{32,33} Shell-side (primary) sodium temperatures were measured with internal thermocouples placed at 16 different locations along the sodium flow path. These measurements are compared to calculations in Figs. 9 and 10. The plots show that there is good agreement in predicted and measured shell-side temperature distributions for both power levels of the heat exchanger.

Fig. 9. ALCO-IHX Shell-side (primary) Temperature Profile at 20% Power

Fig. 10. ALCO-IHX Shell-side (primary) Temperature Profile at 40% Power

System parameters and other input information required to perform these calculations (i.e., thermal-hydraulic operating conditions and heat-transfer coefficients used) are summarized in Appendix E.

2. Transient Responses of IHX Model

To illustrate the capability of the model to calculate transient behavior, two typical transient conditions were chosen:

- a. Transient condition T_1 : a reactor scram followed by removal of secondary (tube-side) pump power 2 sec later.
- b. Transient condition T_2 : failure of primary (shell-side) pumps (except the auxiliary pumps), a reactor scram 0.2 sec later, followed by shut-down of secondary pumps 2 sec after scram.

Figures 11-14 depict time-dependent responses of tube- and shell-side sodium-temperature profiles to the conditions imposed as transients T_1 and T_2 . Figures 15 and 16 show inlet and outlet temperatures as well as flow rates for these transients as a function of time.

Although only sodium temperatures are presented in the figures, the tube-wall surface temperatures are also available as program output.

In addition to the responses of the IHX described above, many others have been determined, e.g., for several prescribed ramp and step-function perturbations of steady-state input conditions.

Fig. 11. ALCO-IHX Shell-side (primary) Temperature Profile during Transient Condition T_1

Fig. 12. ALCO-IHX Tube-side (secondary) Temperature Profile during Transient Condition T_1

Fig. 13. ALCO-IHX Shell-side (primary) Temperature Profile during Transient Condition T_2

Fig. 14. ALCO-IHX Tube-side (secondary) Temperature Profile during Transient Condition T_2

Fig. 15. ALCO-IHX Model Response to Conditions Imposed as Transient T_1

Fig. 16. ALCO-IHX Model Response to Conditions Imposed as Transient T_2

3. Influence of Temperature Dependence of Thermophysical Properties (Sodium; Stainless Steel)

In Sec. III of this report, statements are made that, in the case of transients involving large unbalances in flow rates, the influence of the temperature dependence of thermophysical properties of sodium and stainless steel might be important. Calculations have shown, however, that for the ALCO-IHX, these influences are negligible for both steady-state and transient conditions. Similar conclusions will probably result from study of other IHX designs. It is beneficial, from the standpoint of computer time saving, to treat

thermophysical properties as constants. Thus, whenever considered valid, changes can be made in the main CSMP program (as indicated in Sec. 4 of Appendix F) that are in conformance with that assumption.

B. Computational Efficiency and Convergence Rate

The overall computational efficiency of the simulation model depends upon many factors, including accuracy specifications, choice of Δt , and certain features of the iteration scheme employed. It is also influenced by dynamic-error resetting of integrator output error specifications and whether thermophysical properties are taken as constant or temperature-dependent.

The influence of some of these factors and their interrelationship is discussed below. These results are not obtained from calculations for the ALCO-IHX, but for an IHX as depicted in Fig. 1. System parameters, thermohydraulic conditions, and heat-transfer coefficients used in the calculations are summarized in Appendix E.

Fig. 17. Influence of β on Convergence Rate. ANL Neg. No. 900-156.

optimum with respect to reduction of the number of iterations required for convergence. Comparison of curves for steady and transient states also indicates that the net reduction increases with the total number of iterations required.

In many of the cases studied for this IHX, only three or four iterations were required for each time step; thus, a reduction by one iteration can yield a time saving of 25-30%. For transients involving large unbalanced flow rates, a much larger number of iterations is required. The very steep temperature gradients that occur cause difficulty in satisfying the convergence measure used, i.e., point-by-point agreement, within small tolerance, of temperature values computed in successive runs.

The general character of the convergence of the solutions is illustrated in Fig. 18, in which the time step is $\Delta t = 1$ sec. As can be seen, the

1. Convergence Rate

The alternative iteration schemes discussed in Secs. V.D and V.E are compared in Fig. 17. The calculations were performed without application of the error-resetting feature. Note that the original iteration scheme depicted in Fig. 5 corresponds to that of Fig. 6 with $\beta = 0$. Figure 17 shows that, for the modified iteration scheme (see Fig. 6), selection of a proper value for β (~ 0.5) yields an

Fig. 18. Example of Typical Convergence Sequence. ANL Neg. No. 900-157.

(2) the point, in a series of iterations, at which values are changed; and (3) the total iterations required for convergence without error resetting. Some advantages of the error-resetting feature are illustrated by results of steady-state computations, with and without error resetting, presented in Table III. Although an optimum relation between the several influencing factors depends on the IHX to be studied and the thermohydraulic conditions involved, the results given in Table III indicate that savings in computation time can be substantial.

TABLE III. Effect of Dynamic Error Resetting on Computation Time for the Condition That the Final Value of Relative Error = 10^{-4}

Steady-state Conditions	Computation Time, sec		Improvement in Computation Time with Error Resetting, %
	Without Error Resetting	With Error Resetting	
$T_{ti} = 1000^{\circ}\text{F}$ $T_{si} = 658^{\circ}\text{F}$			
$W_t = 100\%; W_s = 100\%$	3.19	2.88	~10
$W_t = 50\%; W_s = 50\%$	10.60	7.18	~32
$W_t = 20\%; W_s = 20\%$	28.11	17.21	~39

3. Computation and Real-time Comparisons

The ratio of computation time to real time is strongly influenced by the choice of Δt . The value of Δt , however, is also one of the main factors, together with the specified error of the integrator outputs and the convergence criterion, that determines accuracy of results. Effects of different values of Δt on accuracy are indicated by the results presented in Fig. 19. The specified errors for the integrator outputs and the convergence criterion were the same for all cases. The comparisons shown were made using the smallest time step as a reference value. However, this does not mean that use of the smallest time step yields greatest accuracy. A decrease of the time increment increases

unknown end value of the temperature distribution converges rapidly. This is typically true; most of the iterations are required for reshaping the distributions in regions removed from end locations.

2. Error Resetting

The overall effect of using dynamic resetting of error specifications for integrator outputs was found to depend upon several factors, such as (1) the magnitude of the difference in prescribed error specifications;

the total number of convergence tests needed to reach a particular time, thereby increasing the possibility for larger accumulation of residual errors present at each step.

Fig. 19. Effect of Size of Time Increment on Calculated Values of Sodium Outlet Temperatures on Shell Side

A general optimum relation between accuracy, size of Δt , value of β , and error resetting has not been found. However, the present results are sufficient to indicate ranges of the computation-to-real-time ratio. The values obtained range from 2.5 for ramp-function changes in IHX sodium-inlet temperature (of the order of 400°F in 5 sec) to 10.0 for a step-function reduction in flow rate from 100 to 20%.

VIII. DISCUSSION AND CONCLUSIONS

The model developed here provides an effective means for predicting transient responses of an average tube in typically designed sodium heat exchangers (IHX's). Calculated IHX responses are obtained in the form of sodium and tube-wall-surface temperature distributions over the effective heat-transfer length of the IHX at selected time intervals. Computer output of the results can be obtained in printed form or as CALCOMP plots.

With the model, the following items can be studied:

1. IHX responses to input disturbances of time-dependent changes of sodium-inlet temperature and flow rate on either the tube or shell side of the IHX. These changes may be made in any prescribed combinations, including large, unbalanced flow conditions.
2. Influence of treatment of thermophysical properties of sodium as a function of temperature (except specific heat capacity); adjustment of thermal conductivity of tube-wall material (Type 304 stainless steel) in the axial direction.
3. The effect of taking into account different flow regions (axial, cross, or oblique flow).
4. The effect of different correlations for heat-transfer coefficients on IHX responses.

The ability of the model to determine responses of temperature distributions of sodium and tube-wall surface to various operating anomalies of a reactor system, such as scram of the reactor, failure of the flow of primary and secondary sodium, and other unusual temperature excursions, makes it a useful tool in the determination of thermal stresses in the IHX. The model is used in a current study to review the stress analysis of the IHX of EBR-II.³⁴

Figures 9 and 10 show that steady-state calculations for the ALCO-IHX at power levels of 20 and 40% are in good agreement with measured data. However, confirmation of the model through appropriate comparisons with experimentally derived data for transient conditions is essential.

The responses of the IHX model to transient conditions due to a reactor scram followed by a removal of secondary pump power 2 sec later indicate that the outlet temperature of primary sodium (shell side) increases until it approaches the inlet temperature of the primary sodium (see Figs. 11, 12, and 15). In other words, the temperature drop over the reactor core becomes small. Due to the unbalanced flow conditions, however, steep temperature gradients exist in the IHX on the secondary (or tube) side. For transients with balanced flow conditions, the temperature gradients in the IHX do not change appreciably (see Figs. 13, 14, and 16).

The very steep gradients in the temperature profiles at unbalanced flow conditions cause difficulty in satisfying the convergence measure used, i.e., point-by-point agreement, within small tolerances, of temperature values computed in successive runs. Because of this characteristic, it is undesirable to use the error-resetting feature if unbalanced flow conditions are approached. The changes in the temperature profile due to the error resetting require several extra iterations to recover. The error-resetting feature is most effective for balanced flow conditions.

For unbalanced flow conditions, it might be worthwhile to implement another convergence criterion; e.g., instead of looking to a point-by-point agreement of temperature values of successive runs, in vertical direction only, the criterion could be that, within a prescribed circular region, the temperature values of successive runs must agree within small tolerances.

With respect to the application of S/360 CSMP as the basic coding language, the following can be concluded:

1. The CSDT solution method, in conjunction with the variable-step-size integration routines of CSMP, adequately characterizes the temperature distributions by dynamic adjustment of spatial subdivisions. Storage of functions takes place at the integration intervals. For function retrieval, the second-order Lagrange interpolation formula seems to be well-matched with the fourth-order Runge-Kutta integration method, with a variable integration interval and Simpson's Rule used for error estimation.
2. Although an optimum relation has not been found between factors such as size of time interval, value of β , and error specification or resetting that influence both accuracy of results and computational efficiency, the present results are sufficient to lead to a conclusion that the computations are performed satisfactorily.
3. The "hybrid-like" programming character of S/360 CSMP is considered to be an advantage in recognition of possible needs for actual implementation of the IHX model on a hybrid computer for purposes, such as parameter studies and total plant studies. Even for such applications, the CSMP model can serve the useful purpose of providing check calculations to verify the accuracy of hybrid solutions.

As mentioned earlier, the temperature distributions are obtained for the effective heat-transfer length of the IHX only. Thus, the model can be extended to be applicable for other geometries than described in Sec. II of this report, e.g., shell and U-tube heat exchangers. If, for a particular IHX simulation, the inlet and outlet plenums and/or central tube are to be included, it can be done very easily. A plenum can be considered as being an ideal mixing chamber, resulting in an ordinary differential equation that reduces to an algebraic equation after differencing with respect to time; the central tube can be described by a pure delay function.

The computer program developed in this study is not only applicable to simulation of the dynamic behavior of IHX's; the "function-store" subroutines provide a tool for solution of other partial differential equations using the CSDT approach. The solution methods as developed here are of special value to similar analyses of sodium-to-water once-through steam generators. The extensions to the basic S/360 CSMP output control options and error specifications of the integrator outputs are also of general importance for these kinds of problems.

APPENDIX A

Derivation of Equations

1. Basic Equations

In general, the dynamics of heat exchangers are described by the conservation laws of mass, momentum, and energy. For the heat-transfer media of the IHX (liquid sodium and tube-wall material), the following simplifications and/or idealizations are commonly made:

- a. Sodium and tube-wall material are incompressible.
- b. Changes in kinetic and potential energy are negligible.
- c. Dissipation energy is negligible.
- d. There are no internal sources (or sinks) of heat or mass.

Based upon these assumptions, momentum is conserved, and the transient temperature distributions in sodium and the tube wall are then described by only the mass- and heat- (energy-) balance equations. For an infinitesimal volume element, the appropriate general equations in cylindrical coordinates are as follows:³⁵

Mass-balance equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial r}(\rho V_r) + \frac{1}{r} \frac{\partial}{\partial \varphi}(\rho V_\varphi) + \frac{\partial}{\partial x}(\rho V_x) = 0; \quad (\text{A.1})$$

Heat-balance equation:

$$\begin{aligned} & \frac{\partial}{\partial t}(\rho C_p T) + \frac{\partial}{\partial r}(\rho C_p V_r T) + \frac{1}{r} \frac{\partial}{\partial \varphi}(\rho C_p V_\varphi T) + \frac{\partial}{\partial x}(\rho C_p V_x T) \\ & - \frac{1}{r} \frac{\partial}{\partial r} \left(k r \frac{\partial T}{\partial r} \right) - \frac{\partial}{\partial \varphi} \left(k \frac{\partial T}{\partial \varphi} \right) - \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + Q = 0. \end{aligned} \quad (\text{A.2})$$

By means of Eq. A.1, Eq. A.2 can be reduced to

$$\rho \frac{D}{Dt}(C_p T) - \frac{1}{r} \frac{\partial}{\partial r} \left(k r \frac{\partial T}{\partial r} \right) - \frac{1}{r} \frac{\partial}{\partial \varphi} \left(k \frac{\partial T}{\partial \varphi} \right) - \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + Q = 0, \quad (\text{A.3})$$

where

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + V_r \frac{\partial}{\partial r} + V_\varphi \frac{1}{r} \frac{\partial}{\partial \varphi} + V_x \frac{\partial}{\partial x}.$$

Equation A.3 can be further simplified and/or idealized by applying it to the different regions of the IHX element of Fig. 2 and introducing assumptions that are appropriate for each region.

2. Sodium (Shell and Tube Side)

For the sodium regions on the shell and tube sides, the following assumptions are introduced, based on the considerations of Sec. III:

- a. At all cross sections, sodium temperatures and velocities are uniform in both the radial and azimuthal directions.
- b. Axial heat diffusion in the sodium is negligible.
- c. Thermal-entrance effects are negligible.
- d. The specific heat of sodium is constant. Other thermophysical properties of sodium (conductivity, density, and viscosity) are functions of temperature.

By use of these assumptions, Eq. A.3 can be reduced to

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p V_x \frac{\partial T}{\partial x} + Q = 0. \quad (A.4)$$

Then, considering the IHX element of Fig. 2, final equations can be derived for shell- and tube-side sodium.

Instead of considering an infinitesimal volume element $dV = r dr d\varphi dx$, it is only necessary to treat an infinitesimal-length element, dx ; this is accomplished through integration of Eq. A.4 over the cross-sectional areas of the shell- and tube-side sodium, respectively:

Tube-side Sodium

$$\iint_{A_t} \left(\rho C_p \frac{\partial T}{\partial t} + \rho C_p V_x \frac{\partial T}{\partial x} + Q \right) r d\varphi dr = 0,$$

with

$$A_t = \int_0^{2\pi} d\varphi \int_0^{D_i/2} r dr = \frac{\pi}{4} D_i^2 \quad \left(D_i = \frac{ID}{12} \text{ ft} \right);$$

$$V_x = V_t = \frac{W_t}{N \rho_t A_t};$$

$$Q_t = \frac{q_t P_t}{A_t} \frac{dx}{dx} = \frac{q_t P_t}{A_t};$$

$$P_t = \pi D_i.$$

Thus,

$$\rho_t C_{pt} A_t \left(\frac{\partial T_t}{\partial t} + V_t \frac{\partial T_t}{\partial x} \right) + P_t q_t = 0. \quad (\text{A.5a})$$

Shell-side Sodium

$$\iint_{A_s} \left(\rho C_p \frac{\partial T}{\partial t} + \rho C_p V_x \frac{\partial T}{\partial x} + Q \right) r \, d\varphi \, dr = 0,$$

with

$$A_s = \int_0^{2\pi} d\varphi \int_{D_o/2}^{D_e/2} r \, dr = \pi \left(\frac{D_e^2}{4} - \frac{D_o^2}{4} \right) = \frac{\sqrt{3}}{2} P^2 - \frac{\pi}{4} D_o^2$$

$$\left(P = \frac{\text{pitch}}{12} \text{ ft}; \quad D_o = \frac{OD}{12} \text{ ft} \right);$$

$$V_x = -V_s = -\frac{W_s}{N \rho_s A_s};$$

$$Q_s = \frac{q_s P_s}{A_s} \frac{dx}{dx} = \frac{q_s P_s}{A_s};$$

$$P_s = \pi D_o.$$

Thus,

$$\rho_s C_{ps} A_s \left(\frac{\partial T_s}{\partial t} - V_s \frac{\partial T_s}{\partial x} \right) + P_s q_s = 0. \quad (\text{A.6a})$$

The boundary conditions for Eqs. A.5a and A.6a are

$$T_t(0, t) = T_{ti}(t) \quad (\text{A.7a})$$

and

$$T_s(L_{\text{eff}}, t) = T_{si}(t). \quad (\text{A.8a})$$

For use as a reference model, it is convenient to normalize the effective length of the IHX. Substitution of $x = x' L_{eff}$ in Eqs. A.5a and A.6a results in:

Tube-side Sodium

$$\rho_t C_{pt} A_t \left(\frac{\partial T_t}{\partial t} + \frac{1}{\tau_t} \frac{\partial T}{\partial x'} \right) + P_t q_t = 0, \quad (A.5b)$$

where

$$\tau_t = \frac{L_{eff}}{V_t}.$$

Shell-side Sodium

$$\rho_s C_{ps} A_s \left(\frac{\partial T_s}{\partial t} - \frac{1}{\tau_s} \frac{\partial T}{\partial x'} \right) + P_s q_s = 0, \quad (A.6b)$$

where

$$\tau_s = \frac{L_{eff}}{V_s}.$$

The boundary conditions now become

$$T_t(0, t) = T_{ti}(t) \quad (A.7b)$$

and

$$T_s(1, t) = T_{si}(t). \quad (A.8b)$$

3. Tube Wall

For the tube-wall region, the following assumptions can be made:

- a. Temperatures are uniform in the azimuthal direction.
- b. Axial heat conduction is negligible in comparison with radial heat conduction.
- c. All thermophysical properties of the tube wall (stainless steel) are constant in the radial direction. However, they are adjustable in the axial direction.

Based on these assumptions and the fact that tube-wall heat transfer takes place by conduction only, Eq. A.3 reduces to

$$\rho_w C_{pw} \frac{\partial T_w}{\partial t} - \frac{1}{r} \frac{\partial}{\partial r} \left(kr \frac{\partial T_w}{\partial r} \right) = 0 \quad (\text{A.9})$$

or

$$\frac{\partial T_w}{\partial t} - \alpha_w \left(\frac{1}{r} \frac{\partial T_w}{\partial r} + \frac{\partial^2 T_w}{\partial r^2} \right) = 0, \quad (\text{A.10})$$

where

$$\alpha_w = \frac{k_w}{\rho_w C_{pw}}.$$

The boundary conditions for Eq. A.9 are

$$-k_w \frac{\partial T_w}{\partial r} \bigg|_{r=a} = q_t \quad (\text{A.11})$$

and

$$-k_w \frac{\partial T_w}{\partial r} \bigg|_{r=b} = -q_s. \quad (\text{A.12})$$

Appendix B presents an approximate solution of Eq. A.10, in terms of tube-wall surface temperatures.

4. Heat Transfer from Sodium to Tube Wall

The traditional heat-transfer-coefficient concept is used:

$$q_t = h_t (T_t - T_{wt}) \quad (\text{A.13})$$

and

$$q_s = h_s (T_s - T_{ws}). \quad (\text{A.14})$$

APPENDIX B

Approximate Analytical Solution of Linear
Fourier Heat-diffusion Equation

1. Linear One-dimensional Heat-diffusion Equation

In Appendix A, the linear Fourier heat-diffusion equation (Eq. A.9) has been derived. This describes the temperature distribution over an IHX tube wall in the radial direction at each axial location:

$$\alpha_w \left(\frac{\partial^2 T_w}{\partial r^2} + \frac{1}{r} \frac{\partial T_w}{\partial r} \right) - \frac{\partial T_w}{\partial t} = 0, \quad (\text{B.1})$$

with boundary conditions

$$r = a, \quad T_w = T_{wt}, \quad -k_w \frac{\partial T_w}{\partial r} = q_t; \quad (\text{B.2})$$

$$r = b, \quad T_w = T_{ws}, \quad k_w \frac{\partial T_w}{\partial r} = q_s. \quad (\text{B.3})$$

Equation B.1 can be split into a steady-state and a variation equation by substituting

$$T_w = T_w^0 + \Delta T_w,$$

where

$$T_w^0 = \text{initial steady-state function at radius } r$$

and

$$\Delta T_w = \text{variation function in time } t \text{ and radius } r.$$

The steady-state equation is

$$\frac{d^2 T_w^0}{dr^2} + \frac{1}{r} \frac{dT_w^0}{dr} = 0, \quad (\text{B.4})$$

with boundary conditions

$$r = a, \quad T_w^0 = T_{wt}^0, \quad -\frac{dT_w^0}{dr} = q_t^0; \quad (\text{B.5})$$

and

$$r = b, \quad T_w^0 = T_{ws}^0, \quad \frac{dT_w^0}{dr} = q_s^0, \quad (\text{B.6})$$

where

$$q_t^0 = q_s^0 = q^0.$$

The variation equation is

$$\alpha_w \left(\frac{\partial^2 \Delta T_w}{\partial r^2} + \frac{1}{r} \frac{\partial \Delta T_w}{\partial r} \right) - \frac{\partial \Delta T_w}{\partial t} = 0, \quad (\text{B.7})$$

with boundary conditions

$$r = a, \quad \Delta T_w = \Delta T_{wt}, \quad -k_w \frac{\partial \Delta T_w}{\partial r} = \Delta q_t; \quad (\text{B.8})$$

and

$$r = b, \quad \Delta T_w = \Delta T_{ws}, \quad k_w \frac{\partial \Delta T_w}{\partial r} = \Delta q_s. \quad (\text{B.9})$$

2. Solution of Steady-state Equation

The solution of the homogeneous ordinary differential Eq. B.4 can be found by separation of the variables.³⁵ Satisfaction of the boundary conditions B.5 and B.6 then yields

$$T_w^0 - T_{ws}^0 - \frac{r \ln \frac{b}{a}}{k_w} q^0 = 0. \quad (\text{B.10})$$

3. Solution of Variation Equation

For the solution of the linear partial differential Eq. B.7, an approximate analytical-solution technique, as reported by DeGreef,¹⁴ is used. Upon introduction of the Laplace transform of the time dependence, Eq. B.7 becomes

$$\frac{d^2 \Delta T_w}{dr^2} + \frac{1}{r} \frac{d \Delta T_w}{dr} - \frac{s}{\alpha_w} \Delta T_w = 0. \quad (\text{B.11})$$

If we set

$$\frac{s}{\alpha_w} = p^2,$$

Eq. B.11 becomes

$$\frac{d^2 \Delta T_w}{d(pr)^2} + \frac{1}{pr} \frac{d \Delta T_w}{d(pr)} - \Delta T_w = 0. \quad (\text{B.12})$$

This is the modified Bessel equation of order zero, whose general solution is given by

$$\Delta T_w = A I_0(pr) + B K_0(pr), \quad (B.13)$$

where $I_0(pr)$ and $K_0(pr)$ are modified Bessel functions of the zero order and of first and second kind, respectively, and A and B are constants of integration.

Expressions for A and B, determined from the boundary conditions B.8 and B.9 at $r = a$ and $r = b$, are

$$A = \frac{K_0(pb)\Delta T_{wt} - K_0(pa)\Delta T_{ws}}{K_0(pb)I_0(pa) - K_0(pa)I_0(pb)} \quad (B.14)$$

and

$$B = \frac{I_0(pb)\Delta T_{wt} - I_0(pa)\Delta T_{ws}}{K_0(pb)I_0(pa) - K_0(pa)I_0(pb)}. \quad (B.15)$$

Substitution of Eqs. B.14 and B.15 into Eq. B.13 yields

$$\begin{aligned} \Delta T_w = & \frac{K_0(pb)I_0(pr) - I_0(pb)K_0(pr)}{K_0(pb)I_0(pa) - K_0(pa)I_0(pb)} \Delta T_{wt} \\ & + \frac{I_0(pa)K_0(pr) - K_0(pa)I_0(pr)}{K_0(pb)I_0(pa) - K_0(pa)I_0(pb)} \Delta T_{ws}. \end{aligned} \quad (B.16)$$

To avoid undue complexity in the overall IHX model, Eq. B.16 is approximated by truncating the Bessel functions $I_0(pr)$ and $K_0(pr)$ after the second-order terms in pr . Expressions for I_0 and K_0 are as follows:³⁶

$$I_0(pr) = 1 + \frac{(pr)^2}{2^2(1!)^2} + \frac{(pr)^4}{2^4(2!)^2} + \frac{(pr)^6}{2^6(3!)^2} + \dots$$

and

$$K_0(pr) = -\left(\gamma + \ln \frac{pr}{2}\right) I_0(pr) + \sum_{n=1}^{\infty} \frac{1}{(n!)^2} \left(\frac{pr}{2}\right)^{2n} \times \left(\frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n}\right).$$

Truncation after the second-order terms yields

$$I_0(pr) \approx 1 + \frac{(pr)^2}{4} \quad (B.17)$$

and

$$K_0(pr) \approx -\left[\gamma + \ln \frac{pr}{2}\right] \left[1 + \frac{(pr)^2}{4}\right] + \frac{(pr)^2}{4}. \quad (\text{B.18})$$

Substitution of Eqs. B.17 and B.18 in Eq. B.16 yields

$$\Delta T_w = \frac{-\ln \frac{b}{r} + p^2 F_a}{-\ln \frac{b}{a} + p^2 F_{ab}} \Delta T_{wt} + \frac{-\ln \frac{r}{a} + p^2 F_b}{-\ln \frac{b}{a} + p^2 F_{ab}} \Delta T_{ws}, \quad (\text{B.19})$$

where

$$F_a = \frac{-\ln \frac{b}{r}}{4} (r^2 + b^2) + \frac{b^2 - r^2}{4},$$

$$F_b = \frac{-\ln \frac{r}{a}}{4} (a^2 + r^2) + \frac{r^2 - a^2}{4},$$

and

$$F_{ab} = \frac{-\ln \frac{b}{a}}{4} (a^2 + b^2) + \frac{b^2 - a^2}{4}.$$

For the overall model of the IHX, expressions for the tube-wall surface temperatures are needed. They are determined by the boundary conditions for the heat fluxes:

$$-k_w \frac{\partial \Delta T_w}{\partial r} \bigg|_{r=a} = \Delta q_t \quad (\text{B.8})$$

and

$$k_w \frac{\partial \Delta T_w}{\partial r} \bigg|_{r=b} = \Delta q_s. \quad (\text{B.9})$$

Differentiating Eq. B.19 with respect to r and substituting the resulting expression in Eqs. B.8 and B.9 yield

$$\Delta q_t \left(-\ln \frac{b}{a} + p^2 F_{ab} \right) = -k_w \left(\frac{1}{a} + p^2 \frac{\partial F_a}{\partial r} \bigg|_{r=a} \right) \Delta T_{wt} + k_w \left(-\frac{1}{a} + p^2 \frac{\partial F_b}{\partial r} \bigg|_{r=a} \right) \Delta T_{ws}. \quad (\text{B.20})$$

If we set

$$p^2 = \frac{s}{\alpha_w},$$

Eq. B.20 becomes

$$\begin{aligned} \Delta q_t \left(-\ln \frac{b}{a} + \frac{s}{\alpha_w} F_{ab} \right) &= -k_w \left(\frac{1}{a} + \frac{s}{\alpha_w} \frac{\partial F_a}{\partial r} \Big|_{r=a} \right) \Delta T_{wt} \\ &+ -k_w \left(-\frac{1}{a} + \frac{s}{\alpha_w} \frac{\partial F_b}{\partial r} \Big|_{r=a} \right) \Delta T_{ws}. \end{aligned} \quad (B.21)$$

The inverse Laplace transformation of Eq. B.21 to the time domain yields

$$\tau_{wt} \frac{d\Delta T_{wt}}{dt} + \Delta T_{wt} - \tau_w R_a \frac{d\Delta q_t}{dt} - R_a \Delta q_t - \Delta T_{ws} = 0, \quad (B.22)$$

where

$$\tau_{wt} = \frac{a}{\alpha_w} \frac{\partial F_a}{\partial r} \Big|_{r=a} = \frac{b^2 - a^2 - 2a^2 \ln \frac{b}{a}}{4\alpha_w}; \quad (B.23)$$

$$\tau_w = \frac{-F_{ab}}{\alpha_w \ln \frac{b}{a}} = \frac{a^2 + b^2 - \frac{b^2 - a^2}{\ln \frac{b}{a}}}{4\alpha_w}; \quad (B.24)$$

$$R_a = \frac{a}{k_w} \ln \frac{b}{a}. \quad (B.25)$$

Substitution of the expressions for the variational quantities

$$\Delta T_{wt} = T_{wt} - T_{wt}^0,$$

$$\Delta T_{ws} = T_{ws} - T_{ws}^0,$$

and

$$\Delta q_t = q_t - q_t^0$$

into Eq. B.22 yields

$$\tau_{wt} \frac{dT_{wt}}{dt} + T_{wt} - \tau_w R_a \frac{dq_t}{dt} - R_a q_t - T_{ws} = T_{wt}^0 - R_a q_t^0 - T_{ws}^0. \quad (B.26)$$

For $r = a$, Eq. B.9 becomes

$$T_{wt}^0 - T_{ws}^0 - R_a q_t^0 = 0.$$

Thus, Eq. B.26 reduces to

$$\tau_{wt} \frac{dT_{wt}}{dt} + T_{wt} - \tau_w R_a \frac{dq_t}{dt} - R_a q_t - T_{ws} = 0. \quad (B.27)$$

The procedure applied for the boundary condition at $r = b$ results in

$$\tau_{ws} \frac{dT_{ws}}{dt} + T_{ws} - \tau_w R_b \frac{dq_s}{dt} - R_b q_s - T_{wt} = 0, \quad (B.28)$$

where

$$\tau_{ws} = \left. \frac{-b}{\alpha_w} \frac{\partial F_b}{\partial r} \right|_{r=b} = \frac{a^2 - b^2 + 2b^2 \ln \frac{b}{a}}{4\alpha_w}, \quad (B.29)$$

$$\tau_w = \frac{-F_{ab}}{\alpha_w \ln \frac{b}{a}} = \frac{a^2 + b^2 - \frac{b^2 - a^2}{\ln \frac{b}{a}}}{4\alpha_w}, \quad (B.24)$$

and

$$R_b = \frac{b}{k_w} \ln \frac{b}{a}. \quad (B.30)$$

APPENDIX C

Short Description of Main Features of S/360 CSMP

1. Introduction

The S/360 CSMP has been designed by IBM to facilitate the digital simulation of continuous processes on large-scale digital computers. Detailed information is provided in the appropriate IBM publications. The "Application Description"³⁷ and "User's Manual"¹ provide general application guidance, including detailed programming information and a description of input and output facilities. The "System Manual"²² provides information to help the user gain a thorough knowledge of the program logic; although this is not needed to understand the main computer program developed in this study, it is helpful in developing understanding of the implemented additions and modifications to basic features of CSMP.

2. Application-oriented Concepts

S/360 CSMP is an application-oriented input language that enables a user to prepare structure statements describing the functional relationships among variables of a physical system, starting from either an analog block diagram or a representation of that system by ordinary differential equations.

CSMP provides a basic set of 34 functional blocks, which include all conventional analog-computer functions, such as integrators and relays, plus many special-purpose functions commonly required in simulation problems. The basic S/360 CSMP library is augmented by all the functions available in the standard FORTRAN library. S/360 CSMP also accepts FORTRAN statements, allowing the user to handle complex nonlinear and time-variant problems.

A special feature of S/360 CSMP is a sorting algorithm, which sorts the user's input statements and relieves him of the sequencing task.

A NOSORT option is available by identifying a particular section not to be sorted with a NOSORT label. Thus, any procedural statement capability may be included, such as branching on conditions and logical tests.

Specification of data-input information, such as assigning numeric values to parameters, initial conditions, and table entries associated with the problem, is simplified by means of a free format for data-entry statements. The different types of data that can be specified are shown below:

<u>Data Label</u>	<u>Type of Data</u>
PARAM	Parameters
CONST	Constants
INCON	Initial conditions
FUNCTION	Coordinates of an ordinary function
TABLE	Entries in a stored array

Operations associated with the translation, execution, and output phase of the program are specified by control statements. Some relevant control statements are:

<u>Control Label</u>	<u>Operation</u>
NOSORT : :	Specify groups of structure statements not to be sorted
SORT : :	
INITIAL : :	Define a structural segment which is executed only once at the beginning of the run
DYNAMIC : :	Complete description of system dynamics
TERMIN : :	The structure statements within the terminal segment prescribe the computations to be performed upon completion of each run
END or CONTINUE	Run control options (see Sec. 4 below)
TIMER	Specify print interval, print-plot interval, finish time, and integration interval
FINISH	Specify a condition for termination of run
RELERR	Specify relative error for integration routine
ABSERR	Specify absolute error for integration routine
METHOD	Specify integration method
PRINT	Identify variables to be printed
PRTPLT	Identify variables to be print-plotted
TITLE	Print page headings for printed output
LABEL	Print page headings for print-plot

By means of the METHOD statement, a choice may be made among the following integration methods:

ADAMS	Second-order Adams integration with fixed interval
MILNE	Variable-step, fifth-order, predictor-corrector Milne integration method

RECT	Rectangular integration
RKS	Fourth-order Runge-Kutta with variable integration interval; Simpson's Rule used for error estimation
RKSFY	Fourth-order Runge-Kutta with fixed interval
SIMP	Simpson's Rule integration with fixed integration interval
TRAPZ	Trapezoidal integration
CENTRL	A dummy routine that may be replaced by a user-supplied centralized integration subroutine, if desired

Two of the above, viz, RKS and MILNE, allow the integration interval to be adjusted by the system to meet a specified error criterion.

3. User-defined Functions

In addition to the functions contained in the standard S/360 CSMP and FORTRAN libraries, S/360 CSMP provides means to define special functions. Basically, there are three different types; they are identified by the names MACRO, PROCEDURE, and "subprogram."

The MACRO capability is a particularly significant feature of the language. It allows the user to build larger functional blocks from the basic functions available in the library. He thereby can identify a subsection of a simulation block diagram--or the corresponding subset of structure statements--as a parallel functional entity. The statements within a MACRO are automatically sorted. Once defined, the MACRO function can be used any number of times within the simulation.

The PROCEDURE type of user-defined function allows simple application of the logic capabilities of FORTRAN. During sorting, the statements that define the PROCEDURE are treated as a single functional group, and the entire set is moved around as an entity in order to satisfy the input/output sequencing requirements of the sorting algorithm. There is no internal sorting of statements within the PROCEDURE. Generally, a particular PROCEDURE can be used only once within a simulation. However, the PROCEDURE block can be imbedded within a MACRO block and thus used repeatedly.

The FORTRAN subprogram approach permits the user to prepare a separate subprogram to represent functional characteristics. The FORTRAN function is used when the output is a single variable. With multiple outputs, the FORTRAN subroutine is used. This approach permits new blocks to be permanently added to the system library.

4. Run Control

A prescribed run sequence can be established using the CONTINUE, END, and STOP execution control statements, as follows:

A CONTINUE statement permits data and control statements to be changed without resetting the independent variable, and it causes the run to continue. This feature permits such items as integration method, integration interval, and data-output interval to be modified during a simulation study.

An END statement permits data and control statements to be changed, resets the independent variable to zero, and initiates a new run.

A combination of END and STOP statements indicates termination of the sequence. Multiple runs for changes in a parameter can also be performed automatically by specifying the desired values on an appropriate data card. For example, the statement

```
PARAMETER RATE = (5.0, 5.5, 6.0)
```

will yield three runs, one for each value specified for the parameter RATE.

The "nonprescribed" sequence is accomplished by use of the TERMINAL feature, which allows a user to identify a set of statements to be performed only on the termination of a run. This set can define both computations and logical tests for testing run responses, determining the changes to be made between runs, supervising input and output operations, and deciding whether to continue the sequence.

5. Machine-produced Documentation

In the setup/debug phase, problem-checkout facilities include an option for printing current values of variables at specified times and a considerable number of diagnostic messages, which are printed when the program detects an error. Thus, the debugging task is greatly facilitated.

S/360 CSMP automatically prints out the problem-input statements, and provides options for printing output data in standard tabular format, print-plotting in graphic format, and preparation of a data set for user-prepared plotting programs. Page headings can be specified for the tabular and print-plot output. The user can also request that the maximum and minimum values of specified variables be tabulated.

APPENDIX D

Thermophysical Properties of Sodium and Type 304 Stainless Steel1. Sodium

Equations used in the IHX model for calculation of the thermophysical properties of sodium are taken from the compilation of Golden and Tokar.³⁸ These equations and their ranges of application are:

a. Liquid Density

$$\rho \text{ (lb/ft}^3\text{)} = (0.9493 - 0.12485 \times 10^{-3}T)62.43$$

$$\text{for } 411 \leq T \leq 2030^\circ\text{F.} \quad (\text{D.1})$$

b. Liquid Thermal Conductivity

$$k \text{ (Btu/hr-ft-}^\circ\text{F)} = 54.306 - 1.878 \times 10^{-2}T + 2.0914 \times 10^{-6}T^2$$

$$\text{for } 200 \leq T \leq 2500^\circ\text{F.} \quad (\text{D.2})$$

c. Liquid Viscosity

$$\ln \mu \text{ (lb}_m\text{/ft-hr)} = 2.3493 + \frac{914.52}{T + 460} - 1.1340 \log_{10}(T + 460)$$

$$\text{for } 210 \leq T \leq 2500^\circ\text{F.} \quad (\text{D.3})$$

d. Liquid Specific Heat (constant pressure)

$$C_p \text{ (Btu/lb-}^\circ\text{F)} = 0.38952 - 1.10599 \times 10^{-4}(T + 460) + 3.41178 \times 10^{-8}(T + 460)^2$$

$$\text{for } 87 \leq T \leq 1646^\circ\text{F.} \quad (\text{D.4})$$

The specific heat of liquid sodium is taken constant, due to the relatively small change in value (~5%) over the temperature range of interest in this study ($400 \leq T \leq 1200^\circ\text{F}$). The value used is calculated from Eq. D.4 for $T = \frac{1}{2}(T_{ti} + T_{si})$.

Also, the viscosity of liquid sodium is calculated using Eq. D.5 below, which provides a satisfactory fit to the viscosity values calculated from Eq. D.3 over the temperature range of interest. This is done to avoid computation interruptions caused by spurious generation of negative arguments in the FORTRAN ALOG (natural logarithm) subroutine used in solving Eq. D.3.

$$\mu \text{ (lb}_m\text{/ft-hr)} = \frac{367.92}{T} + 0.16994 \text{ for } 400 \leq T \leq 1200^\circ\text{F.} \quad (\text{D.5})$$

2. Type 304 Stainless Steel

Values for the density and specific heat of Type 304 stainless steel are considered to be constant; they are taken from Ref. 39.

Density

$$\rho_w (\text{lb/ft}^3) = 501.12.$$

Specific Heat

$$C_{pw} (\text{Btu/lb-}^\circ\text{F}) = 0.12.$$

Values for thermal conductivity are temperature-dependent based on data from Ref. 40; the data are approximated by the linear expression

Thermal Conductivity

$$k_w (\text{Btu/ft-hr}^\circ\text{F}) = 8.54 + 396.25 \times 10^{-5}T. \quad (\text{D.6})$$

APPENDIX E

Parameters of Sample IHX's

1. System Parameters

In the following, the only system parameters listed are those needed as input to the computer program.

	<u>Parameter</u>	<u>ALCO-IHX</u>	<u>IHX of Fig. 1</u>
Effective length	L_{eff}	22 ft 4 in.	34 ft
Number of tubes	N	461	7410
Outer diameter of tube, in.	OD	0.50	0.75
Inner diameter of tube, in.	ID	0.43	0.666
Pitch (equilateral triangular), in.	P	1.00	1.25
Tube-wall material		Type 304 SS	Type 304 SS

2. Thermohydraulic Conditions

	<u>Parameter</u>	<u>ALCO-IHX</u>	<u>IHX of Fig. 1</u>
Shell-side fluid		Secondary sodium	Secondary sodium
Tube-side fluid		Primary sodium	Primary sodium
Design conditions at 100% power:			
Tube-side inlet temperature, °F	T_{ti}	775	658
Shell-side inlet temperature, °F	T_{si}	1200	1000
Tube-side sodium flow rate, 10^6 lb/hr	W_t	0.855	47.6
Shell-side sodium flow rate, 10^6 lb/hr	W_s	1.140	64.3
Operating conditions at 40% power:			
Tube-side inlet temperature, °F	T_{ti}	755	-
Shell-side inlet temperature, °F	T_{si}	1152	-
Tube-side sodium flow rate, 10^6 lb/hr	W_t	0.3439	-
Shell-side sodium flow rate, 10^6 lb/hr	W_s	0.4266	-
Operating conditions at 20% power:			
Tube-side inlet temperature, °F	T_{ti}	660	-
Shell-side inlet temperature, °F	T_{si}	1070	-
Tube-side sodium flow rate, 10^6 lb/hr	W_t	0.1611	-
Shell-side sodium flow rate, 10^6 lb/hr	W_s	0.20393	-

3. Heat-transfer Coefficients

The correlations for the heat-transfer coefficients used in the simulation studies are taken from Table I.

Tube side: For both IHX's correlation No. 2 is used.

Shell side: For the ALCO-IHX, three regions are considered:

- a. Axial-flow region; 5 ft from inlet on shell side; correlation No. 3B is used.
- b. Crossflow region; 14 ft from end of axial flow region; correlation No. 4 is used.

- c. Cross-axial (oblique) flow region (rest of effective length); correlation No. 5 is used.

The results for the IHX of Fig. 1 are obtained by considering the flow as being purely axial. Thus, correlation No. 3B is used.

Note: Flow bypassing on the shell side of the IHX's is not considered.

APPENDIX F

Preparation of Computer Program for a Particular IHX Simulation

The following material delineates program-data and control statements that must be adjusted for particular simulation studies and IHX designs. Additional information is supplied on program alterations for treatment of thermophysical properties as constants, and use of the disk storage and retrieval of sodium-outlet temperatures, as functions of time.

1. Data Statements

<u>Card No.</u>	<u>Label</u>	<u>S/360 CSMP or FORTRAN Name</u>	<u>Description</u>
CSMP 160	PARAM	WTO	Tube-side mass flow rate of sodium, in lb/hr, for steady-state calculations.
		WSO	Shell-side mass flow rate of sodium, in lb/hr, for steady-state calculations.
CSMP 170	PARAM	TTOI	Sodium-inlet temperature, in °F, on tube side for steady-state calculations.
		TSOI	Sodium-inlet temperature, in °F, on shell side for steady-state calculations.
CSMP 200	FUNCTION	TEMTT	A function for calculating inlet temperature on tube side during transients. The data points on the curve are given in pairs. The first number of a pair indicates a time point; the second indicates the corresponding inlet temperature.
CSMP 210	FUNCTION	TEMTS	Same as for CSMP 200, except here referred to inlet temperature on shell side.
CSMP 220	FUNCTION	FLOWWT	Same as for CSMP 200, except here referred to sodium flow rate on tube side in percentage of WTO.
CSMP 230	FUNCTION	FLOWWS	Same as for CSMP 200, except here referred to sodium flow rate on shell side in percentage of WSO.

Note: The corresponding structure statements for the above FUNCTION's are: CSMP 850, 860, 870, and 890. Nonlinear (NLFGN) or linear (AFGN) function generators can be used.

CSMP 260	PARAM	TIMFIN	End of simulation time (process time), in seconds. TIMFIN = 0.0; for steady-state calculations. TIMFIN > 0.0; for calculation of transients.
CSMP 270	PARAM	DELSEC	Time interval, in seconds, within which converged results are obtained.
CSMP 300	PARAM	SIZE 0.0	Parameter added when CALCOMP plots are desired.
		DELPLT	Time intervals, in seconds (integer), at which output (printed or plotted) is obtained.
CSMP 330	PARAM	ERRCR	First convergence criterion, which, if satisfied, resets the integrator-output error specifications to MINERR (see CSMP 390). It gives the maximum allowable deviation in °F of the tube-side-sodium temperature distribution for two successive runs.

<u>Card No</u>	<u>Label</u>	<u>S/360 CSMP or FORTRAN Name</u>	<u>Description</u>
CSMP 330 (Contd.)		CONVCR	Second (final) convergence criterion. It also gives a maximum allowable deviation in °F of the tube-side-sodium temperature distribution for two successive runs
CSMP 360	PARAM	BETA	A parameter, $0 \leq \text{BETA} \leq 1$, that influences the convergence rate according to the modified iteration scheme of Fig 6.
CSMP 390	PARAM	MINERR	Final relative error in integrator outputs
		MAXERR	Relative error in integrator outputs with which initial iterations are performed
CSMP 420	PARAM	ID	Inner diameter of tube, in inches
		OD	Outer diameter of tube, in inches
		PITCH	Pitch of tubes, in inches
		LEFF	Effective heat-transfer length of IHX, in feet
		N	Number of tubes
CSMP 430	PARAM	XAXIAL	Length of axial flow region in feet, from inlet of shell side
		XCROSS	Length of crossflow region, in feet from end of axial flow region
<u>Note</u> The PROCEDURE function that defines the heat-transfer coefficients on the shell side (see CSMP 2260-2440) has to be appropriately adjusted for a particular IHX design according to data statement CSMP 430			
CSMP 450	PARAM	RHOW	Density of tube wall material in lb/ft ³
		CPW	Specific heat of tube-wall material, in Btu/lb-°F
CSMP 480	PARAM	ALPHAT	Parameters for correlation of heat-transfer coefficient on tube side, corresponding to equation for heat-transfer coefficient
		BETAT	
		PSIT	
		GAMMAT	
CSMP 530	PARAM	WRT	Flag variable for print option of unconverged values of sodium temperature distribution both on tube and shell sides from the function-store subroutines FGEN 1 and FGEN 4 (see CSMP 1270 and CSMP 2110) WRT = 1, print output data WRT = 0, suppress print of output data

2. Execution-control Statements

<u>Card No</u>	<u>Label</u>	<u>S/360 CSMP or FORTRAN Name</u>	<u>Description</u>
CSMP 4000	METHOD	RKS	The particular centralized integration routine to be used for simulation (RKS ADAMS etc, see Appendix C)
CSMP 4010	TIMER	LENGTH	Effective length normalized to a value of 1.0
		OUTDEL	Print increment for print-plot output and preparation of a data tape for CALCOMP plotting

Note For other system variables that can be specified see Ref 1

In Sec. 1 an execution-control statement has been mentioned, viz, CSMP 390 with MINERR and MAXERR parameters, that specify the relative error in integrator outputs. The standard execution-control statements RELERR and ABSERR of S/360 CSMP are overridden when subroutine ERRSET is used.

3. Output-control Statements

<u>Card No.</u>	<u>Label</u>	<u>S/360 CSMP or FORTRAN Name</u>	<u>Description</u>
CSMP 4020	LABEL	(Heading)	Heading for CALCOMP plot of sodium temperature distribution on tube side.
CSMP 4030	PRTPLT	SIZE (X,Y)	Required to get CALCOMP plot. X and Y specify the horizontal and vertical size of the plot, in inches.
		AXORD	Independent variable used in CALCOMP plot.
		TPRIM (MN,MX)	Variable introduced in CSMP 2990 (see listing in Appendix G). MN and MX specify the minimum and maximum values of the variable to scale the plot.
CSMP 4040	LABEL	(Heading)	Same as for CSMP 4020, except here referred to sodium-temperature distribution on shell side.
CSMP 4050	PRTPLT	SIZE (X,Y)	See description of CSMP 4030.
		AXORD	See description of CSMP 4030.
		TSCND (MN,MX)	See description of CSMP 4030; variable defined in CSMP 3000. (See listing in Appendix G.)

Other output-control statements already discussed under "data statements" are: CSMP 300 and 530. If desired, CSMP 4020-4050 can also be replaced by other output-control statements as provided by S/360 CSMP; see Ref. 1.

4. Remarks on Computer-program Alterations

For computations based on assumed constant thermophysical properties of materials, many structure statements included in the DYNAM section of the computer program can be transferred to the INIT section; some can be omitted. This results in a considerable saving in computation time. The corresponding statements to be altered are as follows:

CSMP 1500: KW = SSPROP (TWSKX) becomes KW = SSPROP (TAVG);
 CSMP 1510: RHOT,KT,MUT = NAPROP (TT) becomes
 RHOT,KT,MUT = NAPROP (TAVG);
 CSMP 2500: RHOS,KS,MUS = NAPROP (TS) becomes RHOS,KS,MUS =
 NAPROP (TAVG).

Statements to be transferred to INIT section and inserted between CSMP 1060 and CSMP 1070 are:

CSMP 1340, 1350, 1360, and 1380-1520;
CSMP 1560, 1570, and 1700-1770;
CSMP 2180, 2190, 2200, 2220-2480, 2500, 2510.

Statements to be omitted are:

CSMP 2490, 2550, 2560, and 2660-2730.

If no storage on disk of the sodium-outlet temperature is desired, the following statements are to be omitted:

CSMP 3020-3100, and CSMP 3700-3740.
CSMP 3220 must be replaced by GOTO 3055.

For steady-state calculations, these statements are automatically skipped. If this feature is used, a proper data-set reference number has to be filled out in the argument strings of subroutines FGEN 7 (CSMP 3720) and FGEN 8 (CSMP 3730). For proper additions to the Job Control Language cards, refer to the appropriate IBM OS/360 JCL Manuals.

To read IHX outlet temperatures as a function of time from the disk and to specify the output-control statements for the printed, print-plotted, or CALCOMP plotted output, an auxiliary program has to be used. An example of such a program is given in Appendix G, Sec. 7.

APPENDIX G

Computer-program Listing

1. Basic CSMP Program Segment

* CSMP PROGRAM FOR IHX MODEL	CSMP 10
*	CSMP 20
*	CSMP 30
* DEFINITION OF MACRO FOR THERMAL PROPERTIES OF SODIUM	CSMP 40
MACRO RHO,K,ETA=NAPROP(TEMP)	CSMP 50
K=54.306-(1.878E-2)*TEMP+(2.0E-6)*TEMP**2	CSMP 60
RHO=(.9493-(.12485E-3)*TEMP)*62.43	CSMP 70
ETA=367.92/TEMP+0.1694	CSMP 80
ENDMAC	CSMP 90
* DEFINITION OF MACRO FOR THERMAL CONDUCTIVITY OF SS-304	CSMP 100
MACRO KW=SSPROP(TEMP)	CSMP 110
KW=8.54+396.25E-5*TEMP	CSMP 120
ENDMAC	CSMP 130
*	CSMP 140
* INLET TEMPERATURES AND FLOW RATES FOR STEADY STATE CALCULATIONS	CSMP 150
PARAM WTO=8.55E5,WSO=11.40E5	CSMP 160
PARAM TSOI=1200.,TTOI=775.	CSMP 170
*	CSMP 180
* INLET TEMPERATURES AND FLOW RATES FOR CALCULATIONS OF TRANSIENTS	CSMP 190
FUNCTION TEMT=(0.0,775.),(20.,775.)	CSMP 200
FUNCTION TEMS=(0.0,1200.),(20.,1200.)	CSMP 210
FUNCTION FLOWWT=(0.0,100.),(1.,40.),(20.,40.)	CSMP 220
FUNCTION FLOWWS=(0.0,100.),(20.,100.)	CSMP 230
*	CSMP 240
* FINISH TIME AND TIME INTERVAL	CSMP 250
PARAM TIMFIN=20.	CSMP 260
PARAM DELSEC=1.	CSMP 270
*	CSMP 280
* CALCOMP PARAMETERS	CSMP 290
PARAM SIZE=0.0,DELPLT=5	CSMP 300
*	CSMP 310
* FIRST AND SECOND CONVERGENCE CRITERION	CSMP 320
PARAM ERRCR=5.,CONVCR=1.	CSMP 330
*	CSMP 340
* BETA FOR IMPROVEMENT OF CONVERGENCE RATE(FIG. 6)	CSMP 350
PARAM BETA=.5	CSMP 360
*	CSMP 370
* INTEGRATOR OUTPUT ERROR SPECIFICATIONS	CSMP 380
PARAM MINERR=1.E-4,MAXERR=1.E-3	CSMP 390
*	CSMP 400
* SYSTEM DESIGN PARAMETERS OF IHX	CSMP 410
PARAM ID=.430,OD=.500,PITCH=1.00,LEFF=22.33,N=461.	CSMP 420
PARAM XAXIAL=5.,XCROSS=14.	CSMP 430
* DENSITY AND SPECIFIC OF SS-304	CSMP 440
PARAM RHOW=500.,CPW=.137	CSMP 450
*	CSMP 460
* PARAMETERS FOR CORRELATION OF HEAT TRANSFER COEF. ON TUBE SIDE	CSMP 470
PARAM ALPHAT=5.00,BETAT=.0250,PSIT=1.,GAMMAT=.8	CSMP 480
*	CSMP 490
* INITIALIZATION OF FLAG AND COUNT PARAMETERS	CSMP 500
FIXED NORUN,NOCYCL,FILLUP,TRANSW,NDELT,CONV,WRT,ERFLAG,MAXITR,INT,DELPLTCSMP	510
PARAM NORUN=1,NOCYCL=1,TRANSW=0,NDELT=0,CONV=0,ERFLAG=0,INT=0,ENDSTP=0.0CSMP	520
PARAM WRT=1	CSMP 530
*	CSMP 540
*	CSMP 550
INIT	CSMP 560
*	CSMP 570
AT=PHI*D1*D1/4.	CSMP 580
AS=.867*P*P-PHI*D0*D0/4.	CSMP 590

PT=PHI*D1	CSMP 600
PS=PHI*D0	CSMP 610
D1=10/12.	CSMP 620
D0=00/12.	CSMP 630
DET=D1	CSMP 640
DES=4.*AS/(PHI*D0)	CSMP 650
A=D1/2.	CSMP 660
B=D0/2.	CSMP 670
P=PITCH/12.	CSMP 680
ANB=A*A-B*B	CSMP 690
APB=A*A+B*B	CSMP 700
LBA=ALOG(B/A)	CSMP 710
TAVG=(TTOI+TSOI)/2.	CSMP 720
TR=459.67+TAVG	CSMP 730
TIM=NDEL*DELSEC	CSMP 740
DELHRS=DELSEC/(36.E2)	CSMP 750
PHI=3.14159	CSMP 760
* SPECIFIC HEAT OF SODIUM	CSMP 770
CP=.389352-(1.10599E-4)*TR+(3.41178E-8)*TR**2	CSMP 780
*	CSMP 790
* DISTURBANCE FUNCTIONS OF INLET TEMPERATURES AND FLOW RATES	CSMP 800
*	CSMP 810
NOSORT	CSMP 820
IF(TIM.GT.10.) CONVCR=2.	CSMP 830
IF(TIMFIN.EQ.0.0) GOTO 100	CSMP 840
TTI=NLFGN(TEMTT,TIM)	CSMP 850
TSI=NLFGN(TEMTS,TIM)	CSMP 860
W2=AFGEN(FLOWWT,TIM)	CSMP 870
WT=W2*WTO/100.	CSMP 880
W1=AFGEN(FLOWWS,TIM)	CSMP 890
WS=W1*WSO/100.	CSMP 900
TIMNRM=TIM/TIMFIN	CSMP 910
GOTO 200	CSMP 920
100 CONTINUE	CSMP 930
TTI=TTOI	CSMP 940
TSI=TSOI	CSMP 950
WT=WTO	CSMP 960
WS=WSO	CSMP 970
200 CONTINUE	CSMP 980
*	CSMP 990
* INITIALIZATION OF OUTPUT CONTROL VARIABLES AND INTEGRATOR OUTPUT	CSMP1000
* ERROR SPECIFICATIONS	CSMP1010
CALL OUTPUT(0,0,1)	CSMP1020
IF(NOCYCL.EQ.1.AND.TRANSW+NORUN.EQ.1) RLERR=MAXERR	CSMP1030
IF(NOCYCL.EQ.1.AND.TRANSW+NORUN.EQ.1) CALL ERRSET(RLERR,0,0,1)	CSMP1040
SORT	CSMP1050
*	CSMP1060
*	CSMP1070
DYNAM	CSMP1080
*	CSMP1090
RENAME TIME=AXORD,DELT=DELX,FINTIM=LENGTH	CSMP1100
NOSORT	CSMP1110
IF(NDEL.TGT.0) TRANSW=1	CSMP1120
GOTO(1000,2000),NORUN	CSMP1130
*	CSMP1140
* CALCULATIONS ON TUBE SIDE IN X DIRECTION	CSMP1150
*	CSMP1160
1000 IF(NOCYCL.EQ.1.AND.TRANSW.EQ.0) GOTO 1010	CSMP1170
CALL FGEN4(AXORD,TS,TSKX,TSN1X,TSN2X,2,1,1,0,0,0,16)	CSMP1180
CALL FGEN5(AXORD,TWS,TWSKX,TWSN1X,TWSN2X,2,1,1,0,0,0,16)	CSMP1190
CALL FGEN6(AXORD,HS,HSKX,HSN1X,HSN2X,2,1,1,0,0,0,16)	CSMP1200
FILLUP=1	CSMP1210
GOTO 1020	CSMP1220
1010 FILLUP=0	CSMP1230
DT=(TTOI-TSOI)/(6.*LEFF)	CSMP1240
TWSKX=TT	CSMP1250
1020 CONTINUE	CSMP1260

```

      CALL FGEN1(AXORD,TT,TTKX,TTN1X,TTN2X,0,1,FILLUP,WRT,0,0,16)
SORT
PROCED TTN1X=DUMF3(TRANSW)
      IF(TRANSW.EQ.0) TTN1X=TT
ENDPRO
      TT=INTGRL(TTI,DTTDX)
      DTTDX=-RT*QT-TAUT*(TT-TTN1X)/DELHRS
      RT=PT*N*LEFF/(CP*WT)
      TAUT=LEFF/VT
      VT=WT/(N*RHOT*AT)
      QT=HT*(TT-TWT)
      HT=NUT*KT/DET
*
* THIS PROCED FUNCTION DEFINES THE CORRELATION OF THE HEAT TRANSFER
* COEFFICIENT ON THE TUBE SIDE
PROCED NUT=DUMF6(PET)
      IF(PET.LT.0.0) PET=0.0
      NUT=ALPHAT+BETAT*(PSIT*PET)**GAMMAT
ENDPRO
*
      RET=VT*DET/NHUT
      PRT=MUT*CP/KT
      PET=PRT*RET
      KW=SSPROP(TWSKX)
      RHOT,KT,MUT=NAPROP(TT)
      NHUT=MUT/RHOT
* THIS PROCED FUNCTION CALCULATES THE INNER TUBE WALL SURFACE
* TEMPERATURE
PROCED TWT=DUMF1(TT,HT,KW)
      RA=A*LBA/KW
      RB=B*LBA/KW
      IF(TRANSW-1) 1030,1050,1050
1030 IF(NOCYCL.GT.1) GOTO 1040
      TWT=TT-DT
      GOTO 1060
1040 CONTINUE
      DENOM1=RB*HSKX+RA*HT*(1.+RB*HSKX)
      TWTAST=(1.+RB*HSKX)*TWSKX-RB*HSKX*TSKX
      TTASt=(1.+RA*HT)*TWTAST-TWSKX/(RA*HT)
      TSKXIM=(TSKX+BETA*TT)-BETA*TTAST
      TWT=(RA*HT*(1.+RB*HSKX)*TT+RB*HSKX*TSKXIM)/DENOM1
      GOTO 1060
1050 CONTINUE
      ALPHAW=KW/(RHOW*CPW)
      TAUW=(APB+ANB/LBA)/(4.*ALPHAW)
      TAUWT=-(ANB+2.*A*A*LBA)/(4.*ALPHAW)
      TAUWS=(ANB+2.*B*B*LBA)/(4.*ALPHAW)
      A11=TAUWT/DELHRS
      A12=RA*TAUW/DELHRS
      A21=TAUWS/DELHRS
      A22=RB*TAUW/DELHRS
      CALL FGEN2(AXORD,TWT,TWTKX,TWTN1X,TWTN2X,1,1,1,0,0,0,16)
      CALL FGEN3(AXORD,HT,HTKX,HTN1X,HTN2X,1,1,1,0,0,0,16)
      B1=HT*(A12+RA)
      A1=A11+B1+1.
      C1=A11*TWTN1X+HTN1X*A12*(TWTN1X-TTN1X)
      B2=HSKX*(A22+RB)
      A2=A21+B2+1.
      C2=A21*TWSN1X+HSN1X*A22*(TWSN1X-TSN1X)
      DENOM2=A1*A2-1.
      TWTAST=A2*TWSKX-B2*TSKX-C2
      TTASt=(A1*TWTAST-TWSKX-C1)/B1
      TSKXIM=(TSKX+BETA*TT)-BETA*TTAST
      TWT=(A2*C1+A2*B1*TT+C2+B2*TSKXIM)/DENOM2
1060 CONTINUE
ENDPRO
NOSORT

```

CSMP1270
 CSMP1280
 CSMP1290
 CSMP1300
 CSMP1310
 CSMP1320
 CSMP1330
 CSMP1340
 CSMP1350
 CSMP1360
 CSMP1370
 CSMP1380
 CSMP1390
 CSMP1400
 CSMP1410
 CSMP1420
 CSMP1430
 CSMP1440
 CSMP1450
 CSMP1460
 CSMP1470
 CSMP1480
 CSMP1490
 CSMP1500
 CSMP1510
 CSMP1520
 CSMP1530
 CSMP1540
 CSMP1550
 CSMP1560
 CSMP1570
 CSMP1580
 CSMP1590
 CSMP1600
 CSMP1610
 CSMP1620
 CSMP1630
 CSMP1640
 CSMP1650
 CSMP1660
 CSMP1670
 CSMP1680
 CSMP1690
 CSMP1700
 CSMP1710
 CSMP1720
 CSMP1730
 CSMP1740
 CSMP1750
 CSMP1760
 CSMP1770
 CSMP1780
 CSMP1790
 CSMP1800
 CSMP1810
 CSMP1820
 CSMP1830
 CSMP1840
 CSMP1850
 CSMP1860
 CSMP1870
 CSMP1880
 CSMP1890
 CSMP1900
 CSMP1910
 CSMP1920
 CSMP1930

```

      CALL FGEN2(AXORD,TWT,TWTKX,TWTN1X,TWTN2X,0,1,TRANSW,0,0,0,16)
      CALL FGEN3(AXORD,HT,HTKX,HTN1X,HTN2X,0,1,TRANSW,0,0,0,16)
      IF(NOCYCL.EQ.1.AND.TRANSW.EQ.0) GOTO 2050
      IF(KEEP.EQ.0) GOTO 2050
*   CALCULATION OF CONVERGENCE MEASURE (ERRSTR)
      ERR=ABS(TT-TTKX)
      IF(AXORD.EQ.0.0) ERRSTR=ERR
      ERRST=ERR-ERRSTR
      IF(ERRST.GT.0.0) ERRSTR=ERR
      GOTO 2050
*
*   CALCULATIONS ON SHELL SIDE IN Z DIRECTION
*
2000  CONTINUE
      CALL FGEN1(AXORD,TT,TTKZ,TTN1Z,TTN2Z,2,1,1,0,0,0,16)
      CALL FGEN2(AXORD,TWT,TWTKZ,TWTN1Z,TWTN2Z,2,1,1,0,0,0,16)
      CALL FGEN3(AXORD,HT,HTKZ,HTN1Z,HTN2Z,2,1,1,0,0,0,16)
      CALL FGEN4(AXORD,TS,TSKZ,TSN1Z,TSN2Z,0,1,TRANSW,WRT,0,0,16)
SORT
PROCD TSN1Z=DUMF4(TRANSW)
      IF(TRANSW.EQ.0) TSN1Z=TS
ENDPRO
      TS=INTGRL(TSI,DTSDZ)
      DTSDZ=-RS*QS-TAUS*(TS-TSN1Z)/DELHRS
      RS=PS*N*LEFF/(CP*WS)
      TAUS=LEFF/VS
      VS=WS/(N*RHOS*AS)
      QS=HS*(TS-TWS)
      HS=NUS*KS/DES
*
*   THIS PROCD FUNCTION DEFINES THE CORRELATIONS OF THE HEAT TRANSFER
*   COEFFICIENTS ON THE SHELL SIDE
PROCD NUS=DUMF5(PES,PRS,RES)
      IF(AXORD.GT.XAXIAL/LEFF) GOTO 2001
*   AXIAL FLOW REGION
      EPNUMX=(4.84E-4)*RES+2.16
      X1=1.-1.82/(PRS*(EPNUMX**1.4))
      X2=0.0
      PSIS=FCNSW(X1,X2,X2,X1)
      NUS=17.6+0.0155*(PSIS*PES)**0.86
      GOTO 2003
2001  CONTINUE
      IF(AXORD.GT.(XAXIAL+XCROSS)/LEFF) GOTO 2002
*   CROSS FLOW REGION
      NUS=5.86+0.251*(PES**0.635)
      GOTO 2003
2002  CONTINUE
*   OBLIQUE FLOW REGION
      NUS=0.904*(PES**0.5)
2003  CONTINUE
ENDPRO
*
      RES=VS*DES/NHUS
      PRS=MUS*CP/KS
      PES=PRS*RES
      KW=SSPROP(TWTKZ)
      RHOS,KS,MUS=NAPROP(TS)
      NHUS=MUS/RHOS
*   THIS PROCD FUNCTION CALCULATES THE OUTER TUBE WALL SURFACE
*   TEMPERATURE
PROCD TWS=DUMF2(TS,HS,KW)
      RA=A*LBA/KW
      RB=B*LBA/KW
      IF(TRANSW=1) 2010,2020,2020
2010  CONTINUE
      DENOM1=RB*HS+RA*HTKZ*(1.+RB*HS)

```

CSMP1940
 CSMP1950
 CSMP1960
 CSMP1970
 CSMP1980
 CSMP1990
 CSMP2000
 CSMP2010
 CSMP2020
 CSMP2030
 CSMP2040
 CSMP2050
 CSMP2060
 CSMP2070
 CSMP2080
 CSMP2090
 CSMP2100
 CSMP2110
 CSMP2120
 CSMP2130
 CSMP2140
 CSMP2150
 CSMP2160
 CSMP2170
 CSMP2180
 CSMP2190
 CSMP2200
 CSMP2210
 CSMP2220
 CSMP2230
 CSMP2240
 CSMP2250
 CSMP2260
 CSMP2270
 CSMP2280
 CSMP2290
 CSMP2300
 CSMP2310
 CSMP2320
 CSMP2330
 CSMP2340
 CSMP2350
 CSMP2360
 CSMP2370
 CSMP2380
 CSMP2390
 CSMP2400
 CSMP2410
 CSMP2420
 CSMP2430
 CSMP2440
 CSMP2450
 CSMP2460
 CSMP2470
 CSMP2480
 CSMP2490
 CSMP2500
 CSMP2510
 CSMP2520
 CSMP2530
 CSMP2540
 CSMP2550
 CSMP2560
 CSMP2570
 CSMP2580
 CSMP2590

```

TWSAST=(1.+RA*HTKZ)*TWTKZ-RA*HTKZ*TTKZ
TSAST=((1.+RB*HS)*TWSAST-TWTKZ)/(RB*HS)
TTKZIM=(TTKZ+BETA*TS)-BETA*TSAST
TWS=(RB*HS*(1.+RA*HTKZ)*TS+RA*HTKZ*TTKZIM)/DENOM1
GOTO 2045
2020 CONTINUE
ALPHAW=KW/(RHOW*CPW)
TAUW=(APB+ANB/LBA)/(4.*ALPHAW)
TAUWT=-(ANB+2.*A*A*LBA)/(4.*ALPHAW)
TAUWS=(ANB+2.*B*B*LBA)/(4.*ALPHAW)
A11=TAUWT/DELHRS
A12=RA*TAUW/DELHRS
A21=TAUWS/DELHRS
A22=RB*TAUW/DELHRS
CALL FGEN5(AXORD,TWS,TWSKZ,TWSN1Z,TWSN2Z,1,1,1,0,0,0,16)
CALL FGEN6(AXORD,HS,HSKZ,HSN1Z,HSN2Z,1,1,1,0,0,0,16)
B1=HTKZ*(A12+RA)
A1=A11+B1+1.
C1=A11*TWTN1Z+HTN1Z*A12*(TWTN1Z-TTN1Z)
B2=HS*(A22+RB)
A2=A21+B2+1.
C2=A21*TWSN1Z+HSN1Z*A22*(TWSN1Z-TSN1Z)
DENOM2=A1*A2-1.
TWSAST=A1*TWTKZ-B1*TTKZ-C1
TSAST=(A2*TWSAST-TWTKZ-C2)/B2
TTKZIM=(TTKZ+BETA*TS)-BETA*TSAST
TWS=(A1*C2+A1*B2*TS+C1+B1*TTKZIM)/DENOM2
2045 CONTINUE
ENDPRO
NOSORT
IF(KEEP.EQ.0.OR.CONV.EQ.0) GOTO 2040
* CALCULATION OF NEW STEADY STATE MEASURE(SSMSTR)
SSMEAS=ABS(TS-TSN1Z)
IF(AXORD.EQ.0.0) SSMSTR=SSMEAS
SSMTST=SSMEAS-SSMSTR
IF(SSMTST.GT.0.0) SSMSTR=SSMEAS
2040 CONTINUE
CALL FGEN5(AXORD,TWS,TWSKZ,TWSN1Z,TWSN2Z,0,1,TRANSW,0,0,0,16)
CALL FGEN6(AXORD,HS,HSKZ,HSN1Z,HSN2Z,0,1,TRANSW,0,0,0,16)
TPRIM=TS
TSCND=TTKZ
IF(TIMFIN.EQ.0.0) GOTO 2050
*
* STORAGE OF OUTLET TEMP. AS A FUNCTION OF TIME IN FGEN7 AND FGEN8
* TO BE WRITTEN TO DISK FOR SUBSEQUENT RETRIEVAL AND CALCOMP PLOTTING
IF(CONV.EQ.0.0.OR.AXORD.LT.LENGTH) GO TO 2050
CALL FGEN1(AXORD,TT,TTO,DUM1,DUM2,1,1,1,0,0,0,16)
TTX=TTO
CALL FGEN7(TIMNRM,TS,TTOST,DUM1,DUM2,0,1,0,1,0,0,16)
CALL FGEN8(TIMNRM,TTX,TTOST,DUM1,DUM2,0,1,0,1,0,0,17)
*
2050 CONTINUE
SORT
*
*
TERMIN
*
* TEST FOR APPROACHING SPECIFIED COMPUTER TIME
IF(TLEFT(DUMXYZ).GT.2000.) GO TO 2075
ENDSTP=1.0
WRITE(6,997)
997 FORMAT(' ',1X,'FFFFFFFFFFFF'2X,'TIMELEFT<20SEC. STOP'2X,'FFFFFFFF')
GO TO 3052
2075 CONTINUE
IF(CONV.EQ.1) GOTO 3040
IF(NORUN.EQ.1) GOTO 3005
NORUN=1

```

CSMP2600
CSMP2610
CSMP2620
CSMP2630
CSMP2640
CSMP2650
CSMP2660
CSMP2670
CSMP2680
CSMP2690
CSMP2700
CSMP2710
CSMP2720
CSMP2730
CSMP2740
CSMP2750
CSMP2760
CSMP2770
CSMP2780
CSMP2790
CSMP2800
CSMP2810
CSMP2820
CSMP2830
CSMP2840
CSMP2850
CSMP2860
CSMP2870
CSMP2880
CSMP2890
CSMP2900
CSMP2910
CSMP2920
CSMP2930
CSMP2940
CSMP2950
CSMP2960
CSMP2970
CSMP2980
CSMP2990
CSMP3000
CSMP3010
CSMP3020
CSMP3030
CSMP3040
CSMP3050
CSMP3060
CSMP3070
CSMP3080
CSMP3090
CSMP3100
CSMP3110
CSMP3120
CSMP3130
CSMP3140
CSMP3150
CSMP3160
CSMP3170
CSMP3180
CSMP3190
CSMP3200
CSMP3210
CSMP3220
CSMP3230
CSMP3240
CSMP3250
CSMP3260

NOCYCL=NOCYCL+1	CSMP3270
* TEST FOR ERROR RESETTNG TO REDUCED ERROR (ERRMIN)	CSMP3280
IF(ERFLAG.EQ.0) GOTO 3000	CSMP3290
RLERR=MINERR	CSMP3300
CALL ERRSET(RLERR,0.0,1)	CSMP3310
ERFLAG=0	CSMP3320
3000 CONTINUE	CSMP3330
CALL RERUN	CSMP3340
GOTO 3080	CSMP3350
3005 CONTINUE	CSMP3360
IF(NOCYCL.EQ.1.AND.TRANSW.EQ.0) GOTO 3020	CSMP3370
IF(TRANSW.GT.0) GOTO 3010	CSMP3380
CALL FGEN1(AXORD,TT,TTKX,TTN1X,TTN2X,0,1,1,0,0,0,16)	CSMP3390
CALL FGEN1(AXORD,TT,TTKX,TTN1X,TTN2X,0,1,1,0,0,0,16)	CSMP3400
3010 CONTINUE	CSMP3410
IF(NOCYCL.EQ.1) GOTO 3020	CSMP3420
IF(RLERR.EQ.MINERR) GOTO 3030	CSMP3430
* FIRST CONVERGENCE CRITERION TO SET ERFLAG TO 1 FOR ERROR REDUCTION	CSMP3440
IF(ERRSTR.GT.ERRCR) GOTO 3020	CSMP3450
ERFLAG=1	CSMP3460
3020 CONTINUE	CSMP3470
NORUN=2	CSMP3480
CALL RERUN	CSMP3490
GOTO 3080	CSMP3500
3030 CONTINUE	CSMP3510
* SECOND CONVERGENCE CRITERION TO SET CONV TO 1	CSMP3520
IF(ERRSTR.GT.CONVCR) GOTO 3020	CSMP3530
CONV=1	CSMP3540
WRITE(6,998) TIM	CSMP3550
998 FORMAT(' ',1X,'*****',2X,'TIME=',F10.5,2X,'*****',//)	CSMP3560
* TEST TO INVOKE OUTPUT FOR STEADY STATE AND SELECTED TIME INTERVALS	CSMP3570
IF(NDELT.EQ.0.OR.INT.EQ.DELPLT) CALL OUTPUT(1,1,1)	CSMP3580
GOTO 3020	CSMP3590
3040 CONTINUE	CSMP3600
IF(TRANSW.EQ.0) GOTO 3050	CSMP3610
* CALLS FOR FGEN1-6 FOR PUSH-DOWN TO NEXT TIME LEVEL	CSMP3620
CALL FGEN1(AXORD,TT,TTKX,TTN1X,TTN2X,0,1,1,0,0,0,16)	CSMP3630
CALL FGEN2(AXORD,TWT,TWTKX,TWTN1X,TWTN2X,0,1,1,0,0,0,16)	CSMP3640
CALL FGEN3(AXORD,HT,HTKX,HTN1X,HTN2X,0,1,1,0,0,0,16)	CSMP3650
CALL FGEN4(AXORD,TS,TSKZ,TSN1Z,TSN2Z,0,1,1,0,0,0,16)	CSMP3660
CALL FGEN5(AXORD,TWS,TWSKZ,TWSN1Z,TWSN2Z,0,1,1,0,0,0,16)	CSMP3670
CALL FGEN6(AXORD,HS,HSKZ,HSN1Z,HSN2Z,0,1,1,0,0,0,16)	CSMP3680
3050 CONTINUE	CSMP3690
IF(TIM.LT.TIMFIN.OR.TIMFIN.EQ.0.0) GOTO 3055	CSMP3700
* CALLS FOR FGEN7 AND FGEN8 TO WRITE OUTLET TEMPERATURES TO DISK	CSMP3710
3052 CALL FGEN7(TIMNRM,TT,TTOST,DUM1,DUM2,0,1,0,1,1,0,16)	CSMP3720
CALL FGEN8(TIMNRM,TT,TTOST,DUM1,DUM2,0,1,0,1,1,0,17)	CSMP3730
3055 CONTINUE	CSMP3740
* TEST FOR TIME EQUAL TO FINISH TIME AND SPECIFIED COMPUTER TIME	CSMP3750
* WITHIN SPECIFIED TIME INTERVAL (20 SEC)	CSMP3760
IF(TIM.GE.TIMFIN.OR.ENDSTP.GT.0.0) GO TO 3080	CSMP3770
NORUN=1	CSMP3780
IF(TRANSW.EQ.0) GOTO 3060	CSMP3790
* TEST FOR NEW STEADY STATE CONDITIONS	CSMP3800
IF(SSMSTR.LE.CONVCR) GOTO 3080	CSMP3810
3060 CONTINUE	CSMP3820
* TEST FOR NEED OF ERROR RESETTNG TO LARGER ERROR SPECIFICATION ERRMAX	CSMP3830
IF(SSMSTR.LE.ERRCR) GOTO 3070	CSMP3840
RLERR=MAXERR	CSMP3850
CALL ERRSET(RLERR,0.0,1)	CSMP3860
3070 CONTINUE	CSMP3870
* PROPER INITIALIZATION OF FLAG AND COUNT VARIABLES FOR COMPUTATION	CSMP3880
* OF NEXT TIME INTERVAL	CSMP3890
CONV=0	CSMP3900
NOCYCL=1	CSMP3910
IF(INT.EQ.DELPLT) INT=0	CSMP3920


```

      INT=INT+1
*   INCREMENTATION OF TIME FOR COMPUTATION OF NEXT TIME STEP
      NDEL=NDDEL+1
*   RESUPPRESSION OF OUTPUT
      CALL OUTPUT(0,0,1)
      CALL RERUN
3080  CONTINUE
METHOD RKS
TIMER LENGTH=1.,OUTDEL=.025
LABEL ALCO-IHX PRIMARY SODIUM TEMP. PROFILE WT=20
PRTPLT SIZE(5.,4.),AXORD,TPRIM(600.,1200.)
LABEL ALCO-IHX SECONDARY SODIUM TEMP. PROFILE WT=20
PRTPLT SIZE(5.,4.),AXORD,TSCND(600.,1200.)

```

```

CSMP3930
CSMP3940
CSMP3950
CSMP3960
CSMP3970
CSMP3980
CSMP3990
CSMP4000
CSMP4010
CSMP4020
CSMP4030
CSMP4040
CSMP4050

```

2. Typical Function-Store Subroutine

```

      SUBROUTINE FGEN1(XIN,YIN,YOOUT,YLOUT,Y2OUT,IDIRCT,ILEVEL,
      CNDEL,IPRT,IWRITE,IREAD,IREF)
C
C
C   ARGUMENTS OF SUBROUTINE HAVE FOLLOWING MEANINGS: XIN=INPUT VALUE
C   OF ABSCISSA IN ORDER OF INCREASING VALUES; YIN= CORRESPONDING
C   VALUES OF ORDINATE; YOOUT=OUTPUT VALUE OF ORDINATE FOR IMMEDIATE-
C   LY PRECEDING CALCULATION OF FUNCTION; YLOUT =OUTPUT VALUE OF
C   ORDINATE FOR CALCULATION MADE ONE DELT EARLIER FOR SAME
C   FUNCTION; Y2OUT =OUTPUT VALUE OF ORDINATE FOR CALCULATION
C   MADE TWO DELTS EARLIER FOR SAME FUNCTION; IDIRCT CONTROLS
C   DIRECTION AND RETREIVAL OF FUNCTIONS V.I.Z.: IDIRCT=0, FORWARD
C   STORAGE AND RETREIVAL; IDIRCT=1, FORWARD RETREIVAL ONLY; IDIRCT=2,
C   BACKWARD RETREIVAL ONLY   ILEVEL
C   CONTROLS DEPTH OF HISTORY STORED V.I.Z.: ILEVEL=1 ONE DELT
C   OF PAST VALUES STORED; ILEVEL=2 TWO DELTS OF PAST VALUES STORED
C   NDDEL INDICATES
C   PROBLEM INITIALIZATION; NDDEL=0 INITIAL VALUES ONLY, ALL STORAGE
C   AREAS LOADED; NDDEL=1 DYNAMIC RUNS, PUSH DOWN AS APPROPRIATE
C   IPRT CONTROLS PRINTOUT FOR DEBUGGING: IPRT=0 NO PRINTOUT;
C   IPRT=1 PRINT OUT SPECIFIED DATA
C
C
C   *****
C   USE OF DISK STORAG FACILITY
C
C   IWRITE,IREAD CONTROL WRITING/READING TO/FROM DATA SET WITH
C   REFERENCE NO. GIVEN BY IREF
C   IWRITE/IREAD = 0: NO TRANSMISSION TO/FROM DISK
C   IWRITE=1: WRITE DATA SET ON DISK; SUBROUTINE MUST BE CALLED
C   FROM TERMINAL SECTION OF CSMP.
C   IREAD = 1: READ DATA SET FROM DISK; SUBROUTINE MUST BE CALLED
C   FROM INITIAL SECTION OF CSMP.
C
C   *****
C   *****
C   CARD PUNCHING
C
C   IWRITE CAN ALSO BE USED FOR PUNCHING DATA ON CARDS FROM PUSH-
C   DOWN AREA NO. 2. DATA IS PUNCHED AS A LIST OF 'X' VALUES
C   FOLLOWED BY A LIST OF 'Y' VALUES AS SPECIFIED BY
C   FORMAT NO. 200. IWRITE = 2 ACTIVATES CARD PUNCHING WHEN
C   ROUTINE IS CALLED FROM TERMIN SECTION OF UPDATE.
C
C   *****
C
C   COMMON DUMY1(64),TIME,DELT,DELMIN,FINTIM,PROEL,DUMY2(8414),
C   XKEEP,DUMY3(489),IZOOO,DUMY4(732),IEND

```

```

FGEN 10
FGEN 20
FGEN 30
FGEN 40
FGEN 50
FGEN 60
FGEN 70
FGEN 80
FGEN 90
FGEN 100
FGEN 110
FGEN 120
FGEN 130
FGEN 140
FGEN 150
FGEN 160
FGEN 170
FGEN 180
FGEN 190
FGEN 200
FGEN 210
FGEN 220
FGEN 230
FGEN 240
FGEN 250
FGEN 260
FGEN 270
FGEN 280
FGEN 290
FGEN 300
FGEN 310
FGEN 320
FGEN 330
FGEN 340
FGEN 350
FGEN 360
FGEN 370
FGEN 380
FGEN 390
FGEN 400
FGEN 410
FGEN 420
FGEN 430
FGEN 440
FGEN 450
FGEN 460
FGEN 470
FGEN 480
FGEN 490

```

```

        DIMENSION XCUR(100),XOCUR(100),X1PAST(100),X2PAST(100),
        CYCUR(100),YOCUR(100),Y1PAST(100),Y2PAST(100)
        DATA IDOIN0/0/,IDOIN1/0/,IDOIN2/0/,ICUR/1/,IXFER/1/,NOREAD/0/
1      FORMAT(' ',1X,'PUSH-DOWN 1',2X,8(F10.4,2X)/)
2      FORMAT(' ',1X,'PUSH-DOWN 2',2X,8(F10.4,2X)/)
3      FORMAT(' ',1X,'PUSH-DOWN 3',2X,8(F10.4,2X)/)
4      FORMAT(' ',1X,'FGEN 1',5X,'KEEP=',I4,5X,'ICUR=',I4,5X,'XCUR=',
        CF10.4,5X,'YCUR=',F10.4)
100     FORMAT(' ',1X,'****FGEN1 STORAGE AREA EXCEEDED*****')
200     FORMAT(6(F10.4,2X),3X,'FGEN1')
        GO TO (6000,500,600,4000),IZ0000
C
C
C      IZ0000 IS A CSMP GENERATED VARIABLE USED IN UPDATE TO CONTROL
C      PROGRAM FLOW. IT IS USED HERE AS FOLLOWS:  IZ0000=1; BYPASS
C      ROUTINE;  IZ0000=2, CSMP EXECUTING INITIAL SECTION OF UPDATE, CALL
C      ROUTINE IF STORAGE AREAS ARE TO BE INITIALIZED FROM DISK STORAGE
C      FACILITY;  IZ0000=3, CSMP EXECUTING DYNAMIC SECTION OF UPDATE,
C      USE ROUTINE FOR STORAGE AND/OR RETRIEVAL;  IZ0000=4, CSMP
C      EXECUTING TERMINAL SECTION OF UPDATE, USE ROUTINE FOR PUSH-DOWN
C      OF STORED VALUES AND WRITING INFORMATION TO DISK FACILITY
C      IF DESIRED.
C
C
500     IF(NOREAD.EQ.1) GO TO 6000
        IF(IREAD.EQ.0) GO TO 6000
        NOREAD=1
        READ(IREF) J1MAX,(XOCUR(L),YOCUR(L),L=1,J1MAX)
        DO 550 J=1,J1MAX
            X1PAST(J)=XOCUR(J)
            Y1PAST(J)=YOCUR(J)
            IF(ILEVEL.EQ.1) GO TO 550
            X2PAST(J)=XOCUR(J)
            Y2PAST(J)=YOCUR(J)
550     CONTINUE
        J2MAX=J1MAX
        J3MAX=J1MAX
        GO TO 3015
600     IF(NDELT.EQ.0.AND.KEEP.EQ.0) GO TO 6100
        IF(KEEP.NE.1.OR.IDIRECT.GT.0) GO TO 2020
C
C      CURRENT STORAGE AREA
C
1000    IF(ICUR.LE.100) GO TO 1010
        WRITE(6,100)
        CALL EXIT
        GO TO 6000
1010    CONTINUE
        IF(XIN+.05*DELT.LT.FINTIM) GO TO 1011
        XIN=FINTIM
        ICRSTR=ICUR
        IXFER=2
1011    XCUR(ICUR)=XIN
        YCUR(ICUR)=YIN
        IF(IPRT.EQ.0) GO TO 1015
        XPRINT=XCUR(ICUR)
        YPRINT=YCUR(ICUR)
            WRITE(6,4) KEEP,ICUR,XPRINT,YPRINT
C
C      STEADY-STATE INITIALIZATION OF STORAGE AREAS
C
1015    IF(NDELT.GT.0) GO TO 1020
        J1MAX=ICRSTR
        J2MAX=J1MAX
        XOCUR(ICUR)=XIN
        YOCUR(ICUR)=YIN
        X1PAST(ICUR)=XIN

```

```

Y1PAST(ICUR)=YIN
IF(ILEVEL.EQ.1) GO TO 1020
J3MAX=J1MAX
X2PAST(ICUR)=XIN
Y2PAST(ICUR)=YIN
1020 ICUR=ICUR+1
IF(NDELT.NE.0) GO TO 2020
IF(IXFER.EQ.2) ICUR=1
GO TO 6100

C
C PUSH-DOWN AREA NO. 1
C
2000 J1MAX=ICRSTR
DO 2010 I=1,J1MAX
YOCUR(I)=YCUR(I)
XOCUR(I)=XCUR(I)
2010 CONTINUE
IF(IPRT.EQ.1) WRITE(6,1) (XOCUR(J),YOCUR(J),J=1,J1MAX)
ICUR=1
GO TO 6100

C
C RETRIEVAL AREA NO. 1
C
2020 IF(IDIRCT.LE.1) GO TO 2030
IF(XIN.NE.0.0) GO TO 2025
XPRIME=0.0
Y0OUT=YOCUR(J1MAX)
L1=J1MAX
Y1OUT=Y1PAST(J2MAX)
L2=J2MAX
GO TO (2105,2021),ILEVEL
2021 Y2OUT=Y2PAST(J3MAX)
L3=J3MAX
GO TO 2105
2025 XPRIME=XIN
XIN=FINTIM-XIN
GO TO 2100
2030 IF(XIN.NE.0.0) GO TO 2100
J1MIN=1
J2MIN=1
J3MIN=1
GO TO 2115
2100 DO 2110 J=J1MIN,J1MAX
XINTST=XIN-XOCUR(J)
DELTST=ABS(XINTST)-1.E-3*DELT
IF(DELTST.LE.0.0) GO TO 2101
IF(XINTST.GT.0.0) GO TO 2110
GO TO 2102
2101 IDOINO=1
2102 L1=J
IF(KEEP.EQ.0) GO TO 3100
IF(IDIRCT.LE.1) GO TO 2109
2105 DO 2108 M=1,100
IF(L1.GE.2) K1=L1-M
IF(K1.EQ.1) GO TO 2106
DX0=(XIN-XOCUR(K1))-2.*DELT
IF(DX0.LE.0.0) GO TO 2108
2106 J1MIN=K1
IF(XPRIME.EQ.0.0) GO TO 3105
GO TO 3100
2108 CONTINUE
2109 J1MIN=L1-1
GO TO 3100
2110 CONTINUE
2115 Y0OUT=YOCUR(1)
GO TO 3115

C

```

```

FGEN1170
FGEN1180
FGEN1190
FGEN1200
FGEN1210
FGEN1220
FGEN1230
FGEN1240
FGEN1250
FGEN1260
FGEN1270
FGEN1280
FGEN1290
FGEN1300
FGEN1310
FGEN1320
FGEN1330
FGEN1340
FGEN1350
FGEN1360
FGEN1370
FGEN1380
FGEN1390
FGEN1400
FGEN1410
FGEN1420
FGEN1430
FGEN1440
FGEN1450
FGEN1460
FGEN1470
FGEN1480
FGEN1490
FGEN1500
FGEN1510
FGEN1520
FGEN1530
FGEN1540
FGEN1550
FGEN1560
FGEN1570
FGEN1580
FGEN1590
FGEN1600
FGEN1610
FGEN1620
FGEN1630
FGEN1640
FGEN1650
FGEN1660
FGEN1670
FGEN1680
FGEN1690
FGEN1700
FGEN1710
FGEN1720
FGEN1730
FGEN1740
FGEN1750
FGEN1760
FGEN1770
FGEN1780
FGEN1790
FGEN1800
FGEN1810
FGEN1820
FGEN1830

```

C	PUSH-DOWN AREA 2	FGEN1840
C		FGEN1850
3000	J2MAX=J1MAX	FGEN1860
	DO 3010 J=1,J2MAX	FGEN1870
	X1PAST(J)=XOCUR(J)	FGEN1880
3010	Y1PAST(J)=YOCUR(J)	FGEN1890
	IF(IWRITE.NE.1) GO TO 3015	FGEN1900
	WRITE(IREF) J2MAX,(X1PAST(L),Y1PAST(L),L=1,J2MAX)	FGEN1910
	END FILE IREF	FGEN1920
	REWIND IREF	FGEN1930
3015	IF(IPRT.EQ.1) WRITE(6,2) (X1PAST(J),Y1PAST(J),J=1,J2MAX)	FGEN1940
	IF(IWRITE.EQ.2) WRITE(7,200) (X1PAST(M),M=1,J2MAX)	FGEN1950
	IF(IWRITE.EQ.2) WRITE(7,200) (Y1PAST(M),M=1,J2MAX)	FGEN1960
	GO TO 6000	FGEN1970
C		FGEN1980
C	RETRIEVAL AREA NO. 2	FGEN1990
C		FGEN2000
3100	DO 3110 J=J2MIN,J2MAX	FGEN2010
	XINTST=XIN-X1PAST(J)	FGEN2020
	DELTST=ABS(XINTST)-1.E-3*DELT	FGEN2030
	IF(DELTST.LE.0.0) GO TO 3101	FGEN2040
	IF(XINTST.GT.0.0) GO TO 3110	FGEN2050
	GO TO 3102	FGEN2060
3101	IDOIN1=1	FGEN2070
3102	L2=J	FGEN2080
	IF(KEEP.EQ.0) GO TO (5000,4100),ILEVEL	FGEN2090
	IF(IDIRECT.LE.1) GO TO 3109	FGEN2100
3105	DO 3108 M=1,100	FGEN2110
	IF(L2.GE.2) K2=L2-M	FGEN2120
	IF(K2.EQ.1) GO TO 3106	FGEN2130
	DX1=(XIN-X1PAST(K2))-2.*DELT	FGEN2140
	IF(DX1.LE.0.0) GO TO 3108	FGEN2150
3106	J2MIN=K2	FGEN2160
	IF(XPRIME.EQ.0.0.AND.ILEVEL.EQ.2) GO TO 4105	FGEN2170
	IF(XPRIME.EQ.0.0) GO TO 6100	FGEN2180
	GO TO (5000,4100),ILEVEL	FGEN2190
3108	CONTINUE	FGEN2200
3109	J2MIN=L2-1	FGEN2210
	GO TO (5000,4100),ILEVEL	FGEN2220
3110	CONTINUE	FGEN2230
3115	Y1OUT=Y1PAST(1)	FGEN2240
	GO TO (6000,4115),ILEVEL	FGEN2250
C		FGEN2260
C	PUSH-DOWN AREA NO. 3	FGEN2270
C		FGEN2280
4000	IF(ILEVEL.EQ.1) GO TO 3000	FGEN2290
	J3MAX=J2MAX	FGEN2300
	DO 4010 J=1,J3MAX	FGEN2310
	X2PAST(J)=X1PAST(J)	FGEN2320
4010	Y2PAST(J)=Y1PAST(J)	FGEN2330
	IF(IPRT.EQ.1) WRITE(6,3) (X2PAST(J),Y2PAST(J),J=1,J3MAX)	FGEN2340
	GO TO 3000	FGEN2350
C		FGEN2360
C	RETRIEVAL AREA NO. 3	FGEN2370
C		FGEN2380
4100	DO 4110 J=J3MIN,J3MAX	FGEN2390
	XINTST=XIN-X2PAST(J)	FGEN2400
	DELTST=ABS(XINTST)-1.E-3*DELT	FGEN2410
	IF(DELTST.LE.0.0) GO TO 4101	FGEN2420
	IF(XINTST.GT.0.0) GO TO 4110	FGEN2430
	GO TO 4102	FGEN2440
4101	IDOIN2=1	FGEN2450
4102	L3=J	FGEN2460
	IF(KEEP.EQ.0) GO TO 5000	FGEN2470
	IF(IDIRECT.LE.1) GO TO 4109	FGEN2480
4105	DO 4108 M=1,100	FGEN2490
	IF(L3.GE.2) K3=L3-M	FGEN2500

```

IF(K3.EQ.1) GO TO 4106
DX2=(XIN-X2PAST(K3))-2.*DELT
IF(DX2.LE.0.0) GO TO 4108
4106 J3MIN=K3
IF(XPRIME.EQ.0.0) GO TO 6100
GO TO 5000
4108 CONTINUE
4109 J3MIN=L3-1
GO TO 5000
4110 CONTINUE
4115 Y2OUT=Y2PAST(1)
GO TO 6000

C
C INTERPOLATION (RETRIEVAL) AREA
C
5000 IF(IDOINO.EQ.0) GO TO 5005
Y0OUT=Y0CUR(L1)
GO TO 5020
5005 X1=X0CUR(L1-1)
X2=X0CUR(L1)
Y1=Y0CUR(L1-1)
Y2=Y0CUR(L1)
IF(L1-2) 5006,5006,5007
5006 Y0OUT=FCNINT(XIN,X1,X2,Y1,Y2)
GOTO 5020
5007 X0=X0CUR(L1-2)
Y0=Y0CUR(L1-2)
Y0OUT=FNLINT(XIN,X0,X1,X2,Y0,Y1,Y2)
5020 IF(IDOIN1.EQ.0) GO TO 5025
Y1OUT=Y1PAST(L2)
GO TO 5030
5025 X1=X1PAST(L2-1)
X2=X1PAST(L2)
Y1=Y1PAST(L2-1)
Y2=Y1PAST(L2)
IF(L2-2) 5026,5026,5027
5026 Y1OUT=FCNINT(XIN,X1,X2,Y1,Y2)
GOTO 5030
5027 X0=X1PAST(L2-2)
Y0=Y1PAST(L2-2)
Y1OUT=FNLINT(XIN,X0,X1,X2,Y0,Y1,Y2)
5030 IF(ILEVEL.EQ.1) GO TO 6000
IF(IDOIN2.EQ.0) GO TO 5035
Y2OUT=Y2PAST(L3)
GO TO 6000
5035 X1=X2PAST(L3-1)
X2=X2PAST(L3)
Y1=Y2PAST(L3-1)
Y2=Y2PAST(L3)
IF(L3-2) 5036,5036,5037
5036 Y2OUT=FCNINT(XIN,X1,X2,Y1,Y2)
GOTO 6000
5037 X0=X2PAST(L3-2)
Y0=Y2PAST(L3-2)
Y2OUT=FNLINT(XIN,X0,X1,X2,Y0,Y1,Y2)
6000 GO TO (6100,2000),IXFER
6100 IF(IDIRCT.EQ.2) XIN=TIME
IDOINO=0
IDOIN1=0
IDOIN2=0
IXFER=1
RETURN
END
FUNCTION FCNINT(XIN,X1,X2,Y1,Y2)
DELF=(Y2-Y1)*(XIN-X1)/(X2-X1)
FCNINT=Y1+DELF
RETURN

```

```

FGEN2510
FGEN2520
FGEN2530
FGEN2540
FGEN2550
FGEN2560
FGEN2570
FGEN2580
FGEN2590
FGEN2600
FGEN2610
FGEN2620
FGEN2630
FGEN2640
FGEN2650
FGEN2660
FGEN2670
FGEN2680
FGEN2690
FGEN2700
FGEN2710
FGEN2720
FGEN2730
FGEN2740
FGEN2750
FGEN2760
FGEN2770
FGEN2780
FGEN2790
FGEN2800
FGEN2810
FGEN2820
FGEN2830
FGEN2840
FGEN2850
FGEN2860
FGEN2870
FGEN2880
FGEN2890
FGEN2900
FGEN2910
FGEN2920
FGEN2930
FGEN2940
FGEN2950
FGEN2960
FGEN2970
FGEN2980
FGEN2990
FGEN3000
FGEN3010
FGEN3020
FGEN3030
FGEN3040
FGEN3050
FGEN3060
FGEN3070
FGEN3080
FGEN3090
FGEN3100
FGEN3110
FGEN3120
FGEN3130
FCNI 10
FCNI 20
FCNI 30
FCNI 40

```

END	FCNI	50
FUNCTION FNLINT(XIN,X0,X1,X2,Y0,Y1,Y2)	FNLI	10
A=X1-X0	FNLI	20
B=X2-X1	FNLI	30
C=X2-X0	FNLI	40
D=XIN-X1	FNLI	50
E=XIN-X2	FNLI	60
F=XIN-X0	FNLI	70
FNLINT=Y0*D*E/(A*C)-Y1*F*E/(A*B)+Y2*F*D/(C*B)	FNLI	80
RETURN	FNLI	90
END	FNLI	100

3. Initialization of Output-control Parameters

BLOCK DATA	BLOC	10
COMMON/OUT/ JENTRY,JTEM	BLOC	20
INTEGER JENTRY/0/,JTEM/0/	BLOC	30
END	BLOC	40

4. Output-control Subroutine

SUBROUTINE OUTPUT(IPRINT,IPL0T,IWRITE)	OUTP	10
COMMON DUMY1(8066),KPRINT,DUMY2(906),IZ0000,DUMY3(2),NGRAPH	OUTP	20
COMMON/OUT/ JENTRY,JTEM	OUTP	30
IF(JENTRY.GT.0.AND.IZ0000.EQ.2) GO TO 5000	OUTP	40
IF(JENTRY.GT.0) GO TO 1000	OUTP	50
JENTRY=1	OUTP	60
JPTSTR=KPRINT	OUTP	70
JPLSTR=NGRAPH	OUTP	80
1000 IF(IPRINT.GT.0) GO TO 2000	OUTP	90
KPRINT=0	OUTP	100
GO TO 2500	OUTP	110
2000 KPRINT=JPTSTR	OUTP	120
2500 IF(IPL0T.GT.0) GO TO 3000	OUTP	130
NGRAPH=0	OUTP	140
GO TO 3500	OUTP	150
3000 NGRAPH=JPLSTR	OUTP	160
3500 IF(IWRITE.EQ.0) GO TO 4000	OUTP	170
WRITE(6,100) JPTSTR,JPLSTR,KPRINT,NGRAPH	OUTP	180
100 FORMAT(' ',1X,'SUB. OUTPUT:',2X,'STORED NO. OF PRINT AND PRTPLT VA	OUTP	190
CRS.=' ',2I6,3X,'CONTROLLED NO. OF PRINT AND PRTPLT VARS.=' ',2I6)	OUTP	200
4000 CONTINUE	OUTP	210
JTEM=1	OUTP	220
IF(NGRAPH.GT.0) JTEM=2	OUTP	230
IF(KPRINT.GT.0) JTEM=JTEM+2	OUTP	240
5000 RETURN	OUTP	250
END	OUTP	260

5. Dynamic Error-resetting-control Subroutine

SUBROUTINE ERRSET(RERR,AERR,IPRNT)	ERRS	10
COMMON DUMY1(64),C(8000),DUMY2(11),KREL,KABS,DUMY3(5),	ERRS	20
DUMY4(1716),NOINTG	ERRS	30
LREL)=KREL	ERRS	40
LABS1=KABS	ERRS	50
LSTREL=KREL+NOINTG-1	ERRS	60
LSTABS=KABS+NOINTG-1	ERRS	70
IF(RERR.EQ.0.0.AND.AERR.EQ.0.0) GO TO 2500	ERRS	80
IF(RERR.EQ.0.0) GO TO 1500	ERRS	90
DO 1000 J=LREL1,LSTREL	ERRS	100
1000 C(J)=RERR	ERRS	110

1500	IF(AERR.EQ.0.0) GO TO 2500	ERRS 120
	DO 2000 J=LABS1,LSTABS	ERRS 130
2000	C(J)=AERR	ERRS 140
2500	IF(IPRNT.EQ.0) GO TO 4000	ERRS 150
	WRITE(6,3000) LREL1,LABS1,NOINTG,LSTREL,LSTABS	ERRS 160
3000	FORMAT(' ',1X,'SUB ERRSET:',2X,'LOC. 1ST RELERR=',I4,3X,	ERRS 170
	f'LOC. 1ST ABSERR=',I4,3X,'NO. INTEGRATORS=',I4,3X,	ERRS 180
	f'LOC. LAST RELERR=',I4,3X,'LOC. LAST ABSERR=',I4)	ERRS 190
4000	CONTINUE	ERRS 200
	RETURN	ERRS 210
	END	ERRS 220

6. Revised CSMP MAINEX and SIMOUT Subroutines

	SUBROUTINE MAINEX	MAIN 10
	COMMON DDUM1(64),C(8000),NALAPM,KPOINT,DDUM2(16),RANGE(400),H,KEEP	MAIN 20
	COMMON DDUM3(1214),IFLAG(50),FAM(50),NOINTG,NOSYMB,SYMB(1)	MAIN 30
	COMMON/OUT/ JENTRY,JTEM	MAIN 35
	EQUIVALENCE(IFLAG(12),ISTOP),(DDUM2(9),INTYPE),(DDUM3(493),NGRAPH)	MAIN 40
C		MAIN 50
C	SYSTEM EXECUTION CONTROL STATEMENTS	MAIN 60
C		MAIN 70
C		MAIN 80
C	INITIALIZATION OF RESERVED WORDS AND COUNTERS FOR JOB	MAIN 90
	CALL INITLZ	MAIN 100
C		MAIN 110
C	READ INPUT/OUTPUT CONTROL AND RUN OR CASE SIMULATION CONTROL	MAIN 120
C	READ CSMP DATA STATEMENTS	MAIN 130
9	CALL INTRAN	MAIN 140
C		MAIN 150
C	TEST FOR JOB END OR ERROR EXIT	MAIN 160
	GO TO(10,22),ISTOP	MAIN 170
C		MAIN 180
C	BRANCH DEPENDING ON INTEGRATION TYPE	MAIN 190
C	PERFORM INTEGRATION AND OUTPUT	MAIN 200
CHART	CA-B2	MAIN 210
10	GO TO(11,12,13,14,15,16,17,18),INTYPE	MAIN 220
11	CALL RKS	MAIN 230
	GO TO 19	MAIN 240
12	CALL SIMP	MAIN 250
	GO TO 19	MAIN 260
13	CALL MILNE	MAIN 270
	GO TO 19	MAIN 280
14	CALL RECT	MAIN 290
	GO TO 19	MAIN 300
15	CALL CENTRL	MAIN 310
	GO TO 19	MAIN 320
16	CALL TRAPZ	MAIN 330
	GO TO 19	MAIN 340
17	CALL ADAMS	MAIN 350
	GO TO 19	MAIN 360
18	CALL NOCENT	MAIN 370
C		MAIN 380
C	TEST FOR REQUIRED PRTPLT	MAIN 390
CHART	CA-B4	MAIN 400
19	CONTINUE	MAIN 410
	IF(JENTRY.GT.0) CALL OUTPUT(1,1,0)	MAIN 415
	IF(NGRAPH)21,21,20	MAIN 420
C	GENERATE PRINT PLOT	MAIN 430
CHART	CA-D4	MAIN 440
20	CALL PLOTR	MAIN 450
21	CONTINUE	MAIN 460
C		MAIN 470
C	INSERT CALL TO YOUR OWN PLOTTING ROUTINE HERE.	MAIN 480
C		MAIN 490

```

C      RETURN FOR NEXT CASE                                MAIN 500
C      GO TO 9                                              MAIN 510
C      ALL PROCESSING FROM CSMP DATA STATEMENTS HAS BEEN COMPLETED MAIN 520
22     CONTINUE                                           MAIN 530
C                                                         MAIN 540
C      ADDITIONAL OPERATIONS INVOLVING PLOTTING OF SEVERAL CASES MAIN 550
C      OR MERGING RESULTS WITH OTHER ASSEMBLED DATA      MAIN 560
C      MAY BE PERFORMED USING FORTRAN REFERENCE 15       MAIN 570
C                                                         MAIN 580
C      TRANSFER TO OPERATING SYSTEM FOR THE NEXT JOB.    MAIN 590
C      RETURN                                              MAIN 600
C      END                                                MAIN 610
C      SUBROUTINE SIMOUT                                  SIMO 10
C      DIMENSION IBUF(50),XBUF(50)                       SIMO 20
C      DIMENSION ITITLE(1)                               SIMO 30
C      DIMENSION IBCD(2)                                  SIMO 40
C      DOUBLE PRECISION BEGIN                             SIMO 50
C      DOUBLE PRECISION BUF(50),VAR                      SIMO 60
C      DOUBLE PRECISION PNAME(100),HDNG(50),GLABEL(110),TITLE(55),TYPINT(SIMO 70
$9),WORD,RERUN,ENDFIL                                  SIMO 80
C      COMMON DDUM1(46),TYPINT,C(8000)                   SIMO 90
C      COMMON NALARM,KPOINT,KPRINT,TPRINT,KPLOT,KLAB,TPLOT,KFINIS,KRANGE,SIMO 100
$KLOCK,INTYPE,KREL,KABS,KPT1,TLAST,RLAST,ALAST,KTITLE   SIMO 110
C      COMMON RMIN(100),RMAX(100),TRMIN(100),TRMAX(100) SIMO 120
C      COMMON H,KEEP,TITLE,HDNG,INDEXP(50),IRANGE(100)  SIMO 130
C      COMMON DDUM2(20),WORD,DDUM3(105)                 SIMO 140
C      COMMON TIMEX,NORES,IOP,TFR10,TNEXT,NGRAPH,KGRAPH SIMO 150
C      COMMON DDUM4(150),INDEXPP(50),PNAME,GLABEL,XOUT(50),INDEXG(50) SIMO 160
C...     IFLAG COMMON (50)                               SIMO 170
C      COMMON IDXFAM,NFAM,KFAM,ENDS,RERUN,ENDFIL,IEND,IFIRST,IPLT,IPRINTSIMO 180
$,IFLAG(30),LINE,LINE1,IIFLAG(6)                      SIMO 190
C      COMMON FAM(50),NOINTG,NOSYMB,SYMB(1)             SIMO 200
C      COMMON/OUT/ JENTRY,JTEM                           SIMO 205
C      EQUIVALENCE(C(1),TIME),(C(2),DELT),(C(3),DELMIN),(C(4),FINTIM),(C(SIMO 210
$5),PRDEL),(C(6),OUTDEL)                               SIMO 220
C      EQUIVALENCE(H,DELTA),(TITLE(1),ITITLE(1))        SIMO 230
C      EQUIVALENCE(BUF(1),IBUF(1)),(BUF(26),XBUF(1))    SIMO 240
C      EQUIVALENCE(BEGIN,IBCD(1)),(IIFLAG(1),IDEBUG)    SIMO 250
C...     IFIRST=1 TO INITIALIZE PLOT AND PRINT          SIMO 260
C...     IFIRST=2 FOR NORMAL OUTPUT TIME                SIMO 270
C...     IFIRST=3 FOR TERMINAL CONDITIONS               SIMO 280
C...     IFIRST=4 TO INDICATE END OF RUN                SIMO 290
C      GO TO(1000,1230,1860,2030,1220),IFIRST          SIMO 300
CHART CL-A4                                             SIMO 310
1000  IBLANK=1077952576                                  SIMO 320
      IBCD(1)=-1027225655                                SIMO 330
      IBCD(2)=-717209536                                 SIMO 340
C...     SFT OUTPUT CONTROL VARIABLE, JTEM.            SIMO 350
      JTEM=1                                              SIMO 360
      IF(KGRAPH+NGRAPH)1020,1020,1010                   SIMO 370
1010  JTEM=2                                              SIMO 380
1020  IF(KPRINT)1040,1040,1030                           SIMO 390
1030  JTEM=JTEM+2                                         SIMO 400
1040  CONTINUE                                           SIMO 410
      IF(LINE)1050,1050,1130                             SIMO 420
1050  LINE=60                                             SIMO 430
1060  IF(NFAM-1)1080,1080,1070                           SIMO 440
      END                                                MAIN 610
1070  CALL CSTORE(WORD,IDXFAM,SYMB,IDXFAM,NB,6)         SIMO 450
C...     STORE CSMP/360                                  SIMO 460
1080  IF(KTITLE)1110,1090,1120                           SIMO 470
1090  ITITLE(1)=-1008544553                              SIMO 480
      ITITLE(2)=1077952576                              SIMO 490
C...     STORE BLANK                                    SIMO 500
      DO 1100 I=2,11                                     SIMO 510
      ITITLE(2*I-1)=IBLANK                               SIMO 520
1100  ITITLE(2*I)=IBLANK                                SIMO 530

```


KTITLE=11	SIMO 540
GO TO 1120	SIMO 550
1110 KTITLE=-KTITLE	SIMO 560
1120 LINE1=KTITLE/11+1	SIMO 570
1130 NLINES=(KPRINT+6)/4	SIMO 580
KLIN=60-NLINES	SIMO 590
C... WRITE FIRST LABEL RECORD ON PLOT + PREPAR DATA SETS	SIMO 600
CHART CL-B4	SIMO 610
1140 IF(KGRAPH)1180,1180,1150	SIMO 620
CHART CL-B5	SIMO 630
1150 IF(NFAM-1)1170,1170,1160	SIMO 640
1160 WRITE(13)WORD,C(IDXFAM)	SIMO 650
GO TO 1180	SIMO 660
1170 WRITE(13)IBLANK,IBLANK,IBLANK	SIMO 670
C	SIMO 680
1180 IF(NGRAPH.EQ.0.AND.JENTRY.EQ.0) GO TO 1220	S*** 690
1190 IF(NFAM-1)1200,1200,1210	SIMO 700
1200 WRITE(14)IBLANK,IBLANK,IBLANK	SIMO 710
IF(NGRAPH.EQ.0) WRITE(14) ENDS,(XOUT(I),I=2,KPLOT)	S*** 715
GO TO 1220	SIMO 720
1210 WRITE(14)WORD,C(IDXFAM)	SIMO 730
1220 IFIRST=2	SIMO 740
TPRINT=TLAST	SIMO 750
TPLLOT=TLAST	SIMO 760
CHART CL-C3	SIMO 770
DPT=0.	SIMO 780
DPL=0.	SIMO 790
1230 IPLOT=1	SIMO 800
IPRINT=1	SIMO 810
IF(KGRAPH)1290,1290,1240	SIMO 820
1240 IF(TIMEX-TPLLOT)1250,1260,1260	SIMO 830
1250 IF(IEND)1270,1290,1270	SIMO 840
CHART CL-D3	SIMO 850
1260 IPLOT=2	SIMO 860
DPL=DPL+1	SIMO 870
TIME=TPLOT	SIMO 880
TPLOT=DPL*OUTDEL+TLAST	SIMO 890
1270 DO 1280 I=1,KGRAPH	SIMO 900
K=INDEXG(I)	SIMO 910
1280 XOUT(I)=C(K)	SIMO 920
WRITE(13)(XOUT(I),I=1,KGRAPH)	SIMO 930
1290 IF(KPRINT)1590,1590,1300	SIMO 940
CHART CL-E3	SIMO 950
1300 IF(TIMEX-TPRINT)1310,1320,1320	SIMO 960
1310 IF(IEND)1330,1590,1330	SIMO 970
1320 IPRINT=2	SIMO 980
DPT=DPT+1	SIMO 990
TIME=TPRINT	SIMO1000
TPRINT=DPT*PRDEL+TLAST	SIMO1010
CHART CL-E4	SIMO1020
1330 IF(LINE-KLINE)1520,1340,1340	SIMO1030
CHART CL-E5	SIMO1040
1340 IF(NFAM-1)1400,1400,1350	SIMO1050
1350 IF(INTYPE-10)1360,1380,1380	SIMO1060
1360 WRITE(6,1370)(TITLE(I),I=1,11),TYPINT(INTYPE),WORD,C(IDXFAM)	SIMO1070
1370 FORMAT(1H1,11A6,5X,A8,12H INTEGRATION,5X,A6,1H=,1PE11.4)	SIMO1080
GO TO 1450	SIMO1090
1380 WRITE(6,1390)(TITLE(I),I=1,11),WORD,C(IDXFAM)	SIMO1100
1390 FORMAT(1H1,11A6,17H INTGRL NOT USED,5X,A6,1H=,1PE11.4)	SIMO1110
GO TO 1450	SIMO1120
1400 IF(INTYPE-10)1410,1430,1430	SIMO1130
1410 WRITE(6,1420)(TITLE(I),I=1,11),TYPINT(INTYPE)	SIMO1140
1420 FORMAT(1H1,11A6,5X,A8,12H INTEGRATION)	SIMO1150
GO TO 1450	SIMO1160
1430 WRITE(6,1440)(TITLE(I),I=1,11)	SIMO1170
1440 FORMAT(1H1,11A6,17H INTGRL NOT USED)	SIMO1180
1450 IF(KTITLE-11)1480,1480,1460	SIMO1190

```

1460 WRITE(6,1470)(TITLE(I),I=12,KFTITLE)
1470 FORMAT(1X,11A6)
CHART CL-F4
1480 LINE=LINE1
      IF (IDEBUG)1490,1490,1520
1490 IF (KPRINT-9)1500,1500,1520
1500 WRITE(6,1510)(HDNG(I),I=1,KPRINT)
1510 FORMAT(1H0 2X,A6, 8(6X,A6))
1520 DO 1530 I=1,KPRINT
      K=INDEXP(I)
1530 XOUT(I)=C(K)
      IF (IDEBUG)1540,1540,1570
1540 IF (KPRINT-9)1550,1550,1570
1550 WRITE(6,1560)(XOUT(I),I=1,KPRINT)
1560 FORMAT(1PE12.4,8E12.4)
      LINE=LINE+1
      GO TO 1590
1570 WRITE(6,1580)(HDNG(I),XOUT(I),I=1,KPRINT)
1580 FORMAT(1H0,1X,A6,1H=,1PE12.4,4X, A6,1H=,E12.4,3(6X,A6,1H=,E12.4)/
f(25X,A6,1H=,E12.4,6X,A6,1H=,E12.4,6X,A6,1H=,E12.4,6X,A6,1H=,E12.4)
f)
      LINE=LINE+N LINES
1590 IF (NGRAPH)1660,1660,1600
1600 GO TO(1610,1640),I PLOT
1610 IF (TIMEX-T PLOT)1620,1630,1630
CHART CL-F3
1620 IF (IEND)1640,1670,1640
1630 I PLOT=2
      DPL=DPL+1
      TIME=T PLOT
      T PLOT=DPL*OUTDEL+TLAST
CHART CL-G3
1640 DO 1650 I=1,K PLOT
      K=INDEXPP(I)
1650 XOUT(I)=C(K)
      WRITE(14)(XOUT(I),I=1,K PLOT)
1660 IF (IEND)2210,1670,1730
1670 GO TO(1680,1690,1700,1710),JTEM
1680 TNEXT=FINTIM
      GO TO 1720
1690 TNEXT=T PLOT
      GO TO 1720
1700 TNEXT=T PRINT
      GO TO 1720
1710 TNEXT=AMIN1(T PRINT,T PLOT)
1720 CONTINUE
      GO TO 2210
C... RESET LINE FOR NEXT PRINT OUT
1730 LINE=60
      IF (KRANGE)1820,1820,1740
1740 DO 1750 I=1,KRANGE
      IF (IRANGE(I)-1)1750,1760,1750
1750 CONTINUE
1760 IF (NGRAPH.NE.0) WRITE(6,1770)RMIN(I),RMAX(I)
1770 FORMAT(1H1,10X,16HPROBLEM DURATION, 1PE11.4,4H TO ,E11.4)
      CALL CSTORE(VAR,1,SYMB,1,NB,6)
      IF (NGRAPH.NE.0) WRITE(6,1780)VAR,VAR
1780 FORMAT(1H0/9H VARIABLE,6X,7HMINIMUM,8X,A6,5X,7HMAXIMUM,8X,A6)
      DO 1810 I=1,KRANGE
      I2=IRANGE(I)
      IF (I2-1)1790,1810,1790
1790 CALL CSTORF(VAR,I2,SYMB,I2,NB,6)
      IF (NGRAPH.NE.0) WRITE(6,1800)VAR,RMIN(I),TRMIN(I),RMAX(I),TRMAX(I)
1800 FORMAT(2X,A6,5X,1PE11.4,3(2X,E11.4))
1810 CONTINUE
C... WRITE ENDS ON DATA SETS
1820 IF (KGRAPH)1840,1840,1830

```

```

SIM01200
SIM01210
SIM01220
SIM01230
SIM01240
SIM01250
SIM01260
SIM01270
SIM01280
SIM01290
SIM01300
SIM01310
SIM01320
SIM01330
SIM01340
SIM01350
SIM01360
SIM01370
SIM01380
SIM01390
SIM01400
SIM01410
SIM01420
SIM01430
SIM01440
SIM01450
SIM01460
SIM01470
SIM01480
SIM01490
SIM01500
SIM01510
SIM01520
SIM01530
SIM01540
SIM01550
SIM01560
SIM01570
SIM01580
SIM01590
SIM01600
SIM01610
SIM01620
SIM01630
SIM01640
SIM01650
SIM01660
SIM01670
SIM01680
SIM01690
SIM01700
SIM01710
SIM01720
SIM01730
SIM01740
SIM01750
SIM01760
SIM01770
SIM01780
SIM01790
SIM01800
SIM01810
SIM01820
SIM01830
SIM01840
SIM01850
SIM01860

```

```

1830 WRITE(13)ENDS,(XOUT(I),I=2,KGRAPH)
1840 IF(NGRAPH)2210,2210,1850
1850 WRITE(14)ENDS,(XOUT(I),I=2,KPLOT)
      GO TO 2210
C...  TEST FOR RERUN DUE TO TERMINAL CONDITIONS HERE
1860 IF(KEEP-2)1940,1870,1940
C...  RESET IFIRST FOR NORMAL OUTPUT
1870 IF(KLOCK)1900,1900,1880
1880 WRITE(6,1890)
      1890 FORMAT(42HORERUN FROM TERMIN CANCELED FOR CONTIN RUN)
      KEEP=0
      GO TO 1940
1900 IFIRST=5
      IF(KGRAPH)1920,1920,1910
1910 WRITE(13)RERUN,IBLANK
1920 IF(NGRAPH)2210,2210,1930
1930 WRITE(14)RERUN,IBLANK
      GO TO 2210
C...  TEST FOR PARAMETER FAMILY
CHART CL-J3
1940 IF(NFAM-1)2210,2210,1950
1950 IF(KLOCK)1980,1980,1960
1960 WRITE(6,1970)
      1970 FORMAT(46HONLY LAST VALUE OF FAMILY USED FOR CONTIN RUN)
      GO TO 2210
1980 IF(KFAM-NFAM)1990,2210,2210
CHART CL-J4
1990 KFAM=KFAM+1
      C(IXFAM)=FAM(KFAM)
C...  IEND SET=3 TO INDICATE FAMILY
      ICND=3
C...  RESET IFIRST FOR NORMAL OUTPUT
      IFIRST=5
      IF(KGRAPH)2010,2010,2000
CHART CL-K5
2000 WRITE(13)WORD,C(IXFAM)
2010 IF(NGRAPH)2210,2210,2020
2020 WRITE(14)WORD,C(IXFAM)
      GO TO 2210
C...  END OF RUN. COMPLETE DATA SETS.
2030 IF(KGRAPH)2180,2180,2040
CHART CL-B2
2040 WRITE(13)ENDFIL,IBLANK
      REWIND 13
C...  WRITE IDENTIFICATION RECORDS ON FINAL DATA SET
      WRITE(15)BEGIN,IBLANK
      WRITE(15,2050)TYPINT(INTYPE),KTITLE,NOSYMB,KGRAPH
      2050 FORMAT(A8,3I4)
      WRITE(15,2060)(TITLE(I),I=1,KTITLE)
      2060 FORMAT(55A6)
C...  WRITE NAMES OF VARIABLES BEING PREPARED
      DO 2070 I=1,KGRAPH
      K=INDEXG(I)
CHART CL-C2
2070 CALL CSTORE(BUF(I),K,SYMB,K,NB,6)
      WRITE(15,2060)(BUF(I),I=1,KGRAPH)
C...  WRITE RANGES AND TIMES REACHED FOR VARIABLES
      DO 2100 I=1,KGRAPH
      K=INDEXG(I)
      DO 2080 J=1,KRANGE
      IF(K-IRANGE(J))2080,2090,2080
2080 CONTINUE
2090 IBUF(I)=J
2100 XBUF(I)=RMIN(J)
      WRITE(15)(XBUF(I),I=1,KGRAPH)
      DO 2110 I=1,KGRAPH
      K=IBUF(I)

```

```

SIM01870
SIM01880
SIM01890
SIM01900
SIM01910
SIM01920
SIM01930
SIM01940
SIM01950
SIM01960
SIM01970
SIM01980
SIM01990
SIM02000
SIM02010
SIM02020
SIM02030
SIM02040
SIM02050
SIM02060
SIM02070
SIM02080
SIM02090
SIM02100
SIM02110
SIM02120
SIM02130
SIM02140
SIM02150
SIM02160
SIM02170
SIM02180
SIM02190
SIM02200
SIM02210
SIM02220
SIM02230
SIM02240
SIM02250
SIM02260
SIM02270
SIM02280
SIM02290
SIM02300
SIM02310
SIM02320
SIM02330
SIM02340
SIM02350
SIM02360
SIM02370
SIM02380
SIM02390
SIM02400
SIM02410
SIM02420
SIM02430
SIM02440
SIM02450
SIM02460
SIM02470
SIM02480
SIM02490
SIM02500
SIM02510
SIM02520
SIM02530

```

2110	XBUF(I)=RMAX(K)	SIM02540
	WRITE(15)(XBUF(I),I=1,KGRAPH)	SIM02550
	DO 2120 I=1,KGRAPH	SIM02560
	K=IBUF(I)	SIM02570
2120	XBUF(I)=TRMIN(K)	SIM02580
	WRITE(15)(XBUF(I),I=1,KGRAPH)	SIM02590
	DO 2130 I=1,KGRAPH	SIM02600
	K=IBUF(I)	SIM02610
2130	XBUF(I)=TRMAX(K)	SIM02620
	WRITE(15)(XBUF(I),I=1,KGRAPH)	SIM02630
	J=3*NOSYMB/2+MOD(NOSYMB,2)	SIM02640
	WRITE(15)(SYMB(I),I=1,J)	SIM02650
	WRITE(15)(C(I),I=1,NOSYMB)	SIM02660
C...	COPY OUTPUT ONTO FINAL DATA SET	SIM02670
	READ(13)WORD,BLANK	SIM02680
CHART	CL-D2	SIM02690
2140	WRITE(15)WORD,BLANK	SIM02700
CHART	CL-E2	SIM02710
2150	READ(13)(XOUT(I),I=1,KGRAPH)	SIM02720
	WRITE(15)(XOUT(I),I=1,KGRAPH)	SIM02730
	IF(XOUT(1)-ENDS)2150,2160,2150	SIM02740
2160	READ(13)WORD,BLANK	SIM02750
	IF(WORD-ENDFIL)2140,2170,2140	SIM02760
2170	WRITE(15)WORD,BLANK	SIM02770
	REWIND 13	SIM02780
2180	IF(NGRAPH.EQ.0.AND.JENTRY.EQ.0) GO TO 2200	
2190	WRITE(14)ENDFIL,IBLANK	SIM02800
2200	TLAST=TIME	SIM02810
	KTITLE=-KTITLE	SIM02820
CHART	CL-F2	SIM02830
2210	RETURN	SIM02840
	END	SIM02850

7. Program for Retrieval of Previously Calculated Results

*		
*		
* AUXILIARY PROGRAM FOR READING OUTLET TEMPERATURES FROM DISK		READ 10
PARAM SIZE=0.0		READ 20
INIT		READ 30
NOSORT		READ 40
CALL FGEN7(TIMNRM,TT,TTOST,DUM1,DUM2,0,1,1,1,0,1,16)		READ 50
CALL FGEN8(TIMNRM,TS,TSOST,DUM1,DUM2,0,1,1,1,0,1,17)		READ 60
SORT		READ 70
DYNAM		READ 80
TIMNRM=INTGRL(0.0,1./FINTIM)		READ 90
NOSORT		READ 100
CALL FGEN7(TIMNRM,TT,TTOST,DUM1,DUM2,1,1,1,0,0,0,16)		READ 110
CALL FGEN8(TIMNRM,TS,TSOST,DUM1,DUM2,1,1,1,0,0,0,17)		READ 120
TTO=TTOST		READ 130
TSO=TSOST		READ 140
SORT		READ 150
TIMER FINTIM=20.,DELT=.1,OUTDEL=.25		READ 160
METHOD RKSPX		READ 170
LABEL ALCO-IHX TRANSIENT T1 PRIMARY AND SECUDARY OUTLET TEMP NA-PROP VAR		READ 180
PRTPLT SIZE(5.,5.),TIME,TSO(1100.,1200.),TTO(1100.,1200.)		READ 190

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to Mr. Jack H. Tessier for his continuous advice and active contribution to this study. Thanks also go to Ir. D. G. H. Latzko, Professor of Nuclear Engineering at the Delft University of Technology, The Netherlands, for the initiating efforts that lead to the assignment at the Argonne National Laboratory, and for his guidance and valuable suggestions during this study.

I am indebted to the Dutch Government for supporting my stay at ANL during the academic year 1969/1970, and to Argonne's Center for Educational Affairs under whose auspices the work has been performed.

Acknowledgment and thanks are due Mr. L. W. Fromm, manager of the LMFBR Plant Studies, for his support to this study.

REFERENCES

1. *System/360 Continuous System Modeling Program (360-CX-16X), User's Manual (H20-0367)*, IBM Corporation, Data Processing Division, 112 East Post Road, White Plains, N.Y.
2. L. J. Koch, W. B. Loewenstein, and H. O. Monson, *Addendum to Hazard Summary Report Experimental Breeder Reactor-II (EBR-II)*, ANL-5719 (Addendum) (Jan 1964).
3. *30 Megawatt Heat Exchanger and Steam Generator for Sodium Cooled Reactor System. Vol. I, Thermal and Mechanical Final Design*, APAE-112 (Vol. I) (Jan 1962).
4. *LMFBR Low Capacity IHX Design Studies Supporting FFTF Development, Vol. I, Results, Conclusions, and Recommendations*, WARD-4112-2 (Jan 1970).
5. C. E. Skufelt, ed., *1000 MWe LMFBR, Follow-on Study, Westinghouse Electric Task II and III Report, Conceptual Design*, WARD-2000-97 (Apr 1969).
6. W. R. Gee, *1000 MWe LMFBR, Follow-on Study, General Electric Task II Report, Conceptual Plant Design, System Description, and Costs for a 1000 MWe Sodium-cooled Fast Reactor*, GEAP-5678 (Dec 1968).
7. *1000 MWe LMFBR, Follow-on Study, Task II and III Final Report, Vol. 2, Conceptual System Design Descriptions*, BAW-1328 (Vol. 2) (Mar 1969).
8. C. B. Jackson, *Liquid Metals Handbook (Sodium-NaK Supplement)*, TID-5277, Atomic Energy Commission--Department of the Navy, Washington, D.C. (July 1955).
9. R. P. Stein, *Liquid Metal Heat Transfer*, Advances in Heat Transfer, Vol. 3, Academic Press, New York (1966), pp. 101-174.
10. J. O. Hinze, *Stromingsleer V.C., II*, Delft University of Technology, The Netherlands (1966) (in Dutch).
11. P. J. Schneider, *Effect of Axial Fluid Conduction on Heat Transfer in the Entrance Regions of Parallel Plates and Tubes*, Heat Transfer and Fluid Mechanics Institute, 9th Stanford University, Stanford, Calif., pp. 41-58 (June 1956).
12. R. W. Brown, *Analytical and Experimental Studies of Double-pipe Counter-flow Liquid Metal Heat Exchangers*, ANL-7479 (Aug 1968).
13. *Reactor Development Program Progress Report: July 1967*, ANL-7357, pp. 116-117 (Aug 31, 1967).
14. J. F. DeGreef, "Cylindrical Thermal Equations for the Analogue Computer." *Annales de l'Association Internationale pour le Calcul Analogique* 9, 4 (Oct 1967), pp. 195-197.
15. O. E. Dwyer and R. N. Lyon, "Liquid Metal Heat Transfer," *Proc. Third Int. Conf. on the Peaceful Uses of Atomic Energy*, Geneva 8, pp. 182-191 (1964).
16. R. N. Lyon, *Liquid Metal Heat-Transfer Coefficients*, Chem. Eng. Progr. 47, No. 2, pp. 75-79 (Feb 1951).
17. V. I. Subbotin et al., *A Study of Heat Transfer to Molten Sodium in Tubes (Transl.)*, Soviet J. Atomic Energy 13, 991-994 (1962).
18. O. E. Dwyer, *Heat Transfer to Liquid Metals Flowing In-Line Through Unbaffled Rod Bundles: A Review*, Nucl. Eng. Design 10(1), 3-20 (May 1969); M. W. Manesca and O. E. Dwyer, *Heat Transfer to Mercury Flowing In-line through a Bundle of Circular Rods*, J. Heat Transfer 86, 180-6 (1964).

19. Sheldon Kalish and O. E. Dwyer, *Heat Transfer to NaK Flowing Through Unbaffled Rod Bundles*, *Int. J. Heat Mass Transfer* 10, 1533-1558 (1967).
20. G. A. Bekey and W. J. Karplus, *Hybrid Computation*, John Wiley & Sons, Inc., New York (1968).
21. S. Shapiro, "Analog, Hybrid and Digital Simulation--A Comparative Evaluation," *Fifth International Analogue Computation Meetings*, Lausaemne I, 301-305 (1967), Presses Academiques Europeennes, Bruxelles.
22. *System/360 Continuous System Modeling Program (360A-CX-16X)*, System Manual Y20-0111, IBM Corporation, Data Processing Division, 112 East Post Road, White Plains, N.Y.
23. R. Fatton and A. Tagher, "Ability Comparison and Limits of Analog, Digital, and Hybrid Computation Methods," *Fifth International Analogue Computation Meetings*, Lausaemne I, 335-344 (1967), Presses Academiques Europeennes, Bruxelles.
24. B. A. Chubb, *Economic Evaluation of the CSMP Digital Simulation Language*, *Simulation* 14(3), 101-103 (Mar 1970).
25. B. Wilson, *The Harmonic Response of a Simple Counter-Flow Heat Exchanger with Various Finite-Difference Representations*, TRG Report 424 (Oct 18, 1962).
26. A. L. Gunby, *Intermediate Heat Exchanger Modeling for FFTF Simulation*, BNWL-1367 (May 1970).
27. R. Vichnevetsky, *Analog-Hybrid Solution of Partial Differential Equations in the Nuclear Industry*, *Simulation* 11(6), 269-281 (Dec 1968).
28. R. Vichnevetsky, *A New Stable Computing Method for the Serial Hybrid Computer Integration of Partial Differential Equations*, Spring Joint Computer Conference, AFIPS, Thompson Book Co., Washington, D.C., Vol. 32, 143-150 (1968).
29. L. Dekker, *Solving Two-Point Boundary Problems by a Level-Control Method*, *Annales de l'Association Internationale pour le Calcue Analogique* 11(1), 11-20 (Jan 1969).
30. L. N. Carling, *Hybrid Computer Solution of Heat Exchanger Partial Differential Equations*, *Annales de l'Association Internationale pour le Calcue Analogique* 10(1), 6-18 (Jan 1968).
31. R. Vichnevetsky, "Hybrid Solution of Partial Differential Equations in Nuclear Plants Simulation: A Review," *The Conference on the Effective Use of Computers in the Nuclear Industry*, Knoxville, Tenn., CONF-690401 (Apr 21-23, 1969), pp. 369-390.
32. *Liquid Metal Engineering Center, Monthly Report*, 69LMEC-906 (June 1969).
33. *Liquid Metal Engineering Center, Monthly Report*, 69LMEC-1066 (July 1969).
34. *Reactor Development Program Progress Report: July 1970*, ANL-7726, pp. 58-60 (Aug 27, 1970).
35. J. O. Hinze, *Stroming en Warmte Overdracht I*, Delft University of Technology, The Netherlands (1964) (in Dutch).
36. *British Association for the Advancement of Science, Mathematical Tables, Volume VI, "Bessel Functions, Part I, Functions of Orders Zero and Unity,"* Cambridge University Press (1950).

37. *System/360 Continuous System Modeling Program (360A-CX-16X) Application Description (H20-0240-2)*, IBM Corp., Data Processing Division, 112 East Post Road, White Plains, N.Y.
38. G. H. Golden and J. V. Tokar, *Thermophysical Properties of Sodium*, ANL-7323 (Aug 1967).
39. Taylor Lynman, ed., *Metals Handbook, Vol. I, Properties and Selection of Metals*, 8th Edition, American Society for Metals (1961), p. 422.
40. E. A. Eldridge and H. W. Deem, *Report on Physical Properties of Metals and Alloys from Cryogenic to Elevated Temperatures*, ASTM Special Technical Publication No. 296, Philadelphia, American Society for Testing Materials (1961).