Report 2135

PROGRAMING LANGUAGE FOR THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS USING HYBRID COMPUTERS PHASE I

March 1975

U. S. ARMY MOBILITY EQUIPMENT RESEARCH AND DEVELOPMENT CENTER
FORT BELVOIR, VIRGINIA

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 2135	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PROGRAMING LANGUAGE FOR THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS USING HYBRID COMPUTERS–PHASE I		5. TYPE OF REPORT & PERIOD COVERED Interim; July 1973 through July 1974 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. Thomas Broach Robert M. McKechnie, III		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS USAMERDC, Electrotechnology Department Electrical Equipment Division, ATTN: AMXFB-EA Fort Belvoir, Virginia 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project 1E865803M730
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Mobility Equipment Research and Development Center Fort Belvoir, Virginia 22060		12. REPORT DATE March 1975 13. NUMBER OF PAGES 83
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Analog computers
Computer applications
Digital computers
Hybrid computers
Partial differential equations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The report covers the first phase of work intended to provide a hybrid-computer solution language for partial differential equations. The programing techniques and philosophy are discussed, and a sample problem solution is presented with details. This first attempt has been successful and provides a hybrid-computer scheme which is at least 50 times faster than the comparable purely digital approach. The program makes use of hybrid-computer graphics, with the input applied directly to the Tektronix 4010 Graphics Terminal and the solution curves presented to the graphics terminal on-line.

PREFACE

This work was authorized under CIS Project 1E865803M730, "Improved Data Effectiveness and Availability (IDEA)." This effort is part of the AMC CIS/CAD-E program.

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS (cont'd)

PROGRAMING LANGUAGE FOR THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS USING HYBRID COMPUTERS PHASE I

I. INTRODUCTION

1. **Objective.** The objective of this report is twofold. The first objective is to provide a progress report on hybrid-computer solution techniques for partial differential equations. The second objective is to provide documented details of the solution mechanics and to illustrate the power and speed of the hybrid computer when solving partial differential equations (PDE). A solution-speed comparison between the hybrid and digital techniques shows the hybrid to be the faster of the two.

2. **Background.** The Electrical Equipment Division, U.S. Army Mobility Equipment Research and Development Center (USAMERDC), is involved in the research, development, and engineering of electromagnetic machinery, power conditioners, and power electronics components (SCR's, transistors, and rectifiers). These efforts require the solution of partial differential equations in order to provide flux plots and equipotential plots. When digital-computer techniques are used, these problem solutions are slow and costly. However, by using hybrid-computer techniques, we can reduce these computing costs by a factor of 15 to 25, with a corresponding increase in computing speed by a factor of between 15 and 100. The Electrical Equipment Division has a powerful, interactive hybrid-computer facility (Figure 1), which is part of the CAD-E facility (Figure 2). The hybrid computer is a Digital Equipment Corporation PDP-15/Applied Dynamics AD-4 hybrid computer coupled to a Tektronix 4010 Graphic Terminal. Figure 3 shows the PDP-15/76 digital processor which has a unichannel, 1.2-million-word disk and 16K of core. The AD-4 analog processor (Figure 4) has 96 amplifiers as well as an autopatch capability. The technical paper *Hybrid Computer Solution Techniques for Laplace's Equations,* by the authors of this report, has helped immensely in preparing this report.*

3. **Organization.** This report is divided into five parts: Introduction, Program Philosophy, Computer-Solution Mechanics, Examples, and Conclusions and Future Work. Additional material is given in the three appendixes. The Program Philosophy section describes the philosophy of program development. The section on Computer-Solution Mechanics presents the details of problem setup for the hybrid-computer solution. The Examples section and the appendixes present sample problem solutions and special considerations. This report will provide the basis for comparing the interactive hybrid-computer solution of partial differential equations to the digital-computer approach.

* J. T. Broach and R. M. McKechnie, *Hybrid Computer Solution Techniques for Laplace's Equation,* Proceedings of 1974 Army Numerical Analysis Conference, ARO Report 74-2, pp. 253-271.

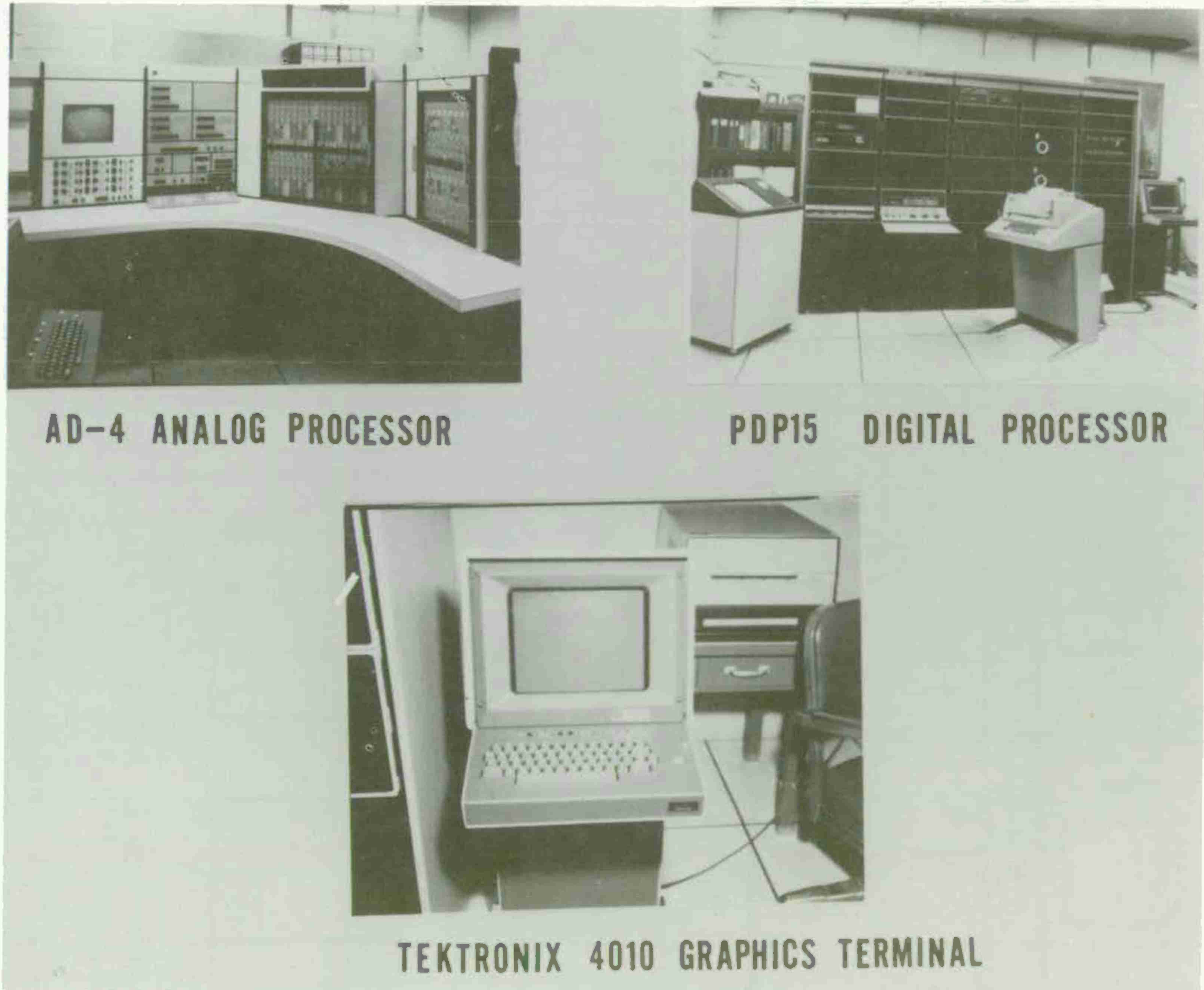


Figure 1. Interactive hybrid-computer facility. 6B-C05781/74

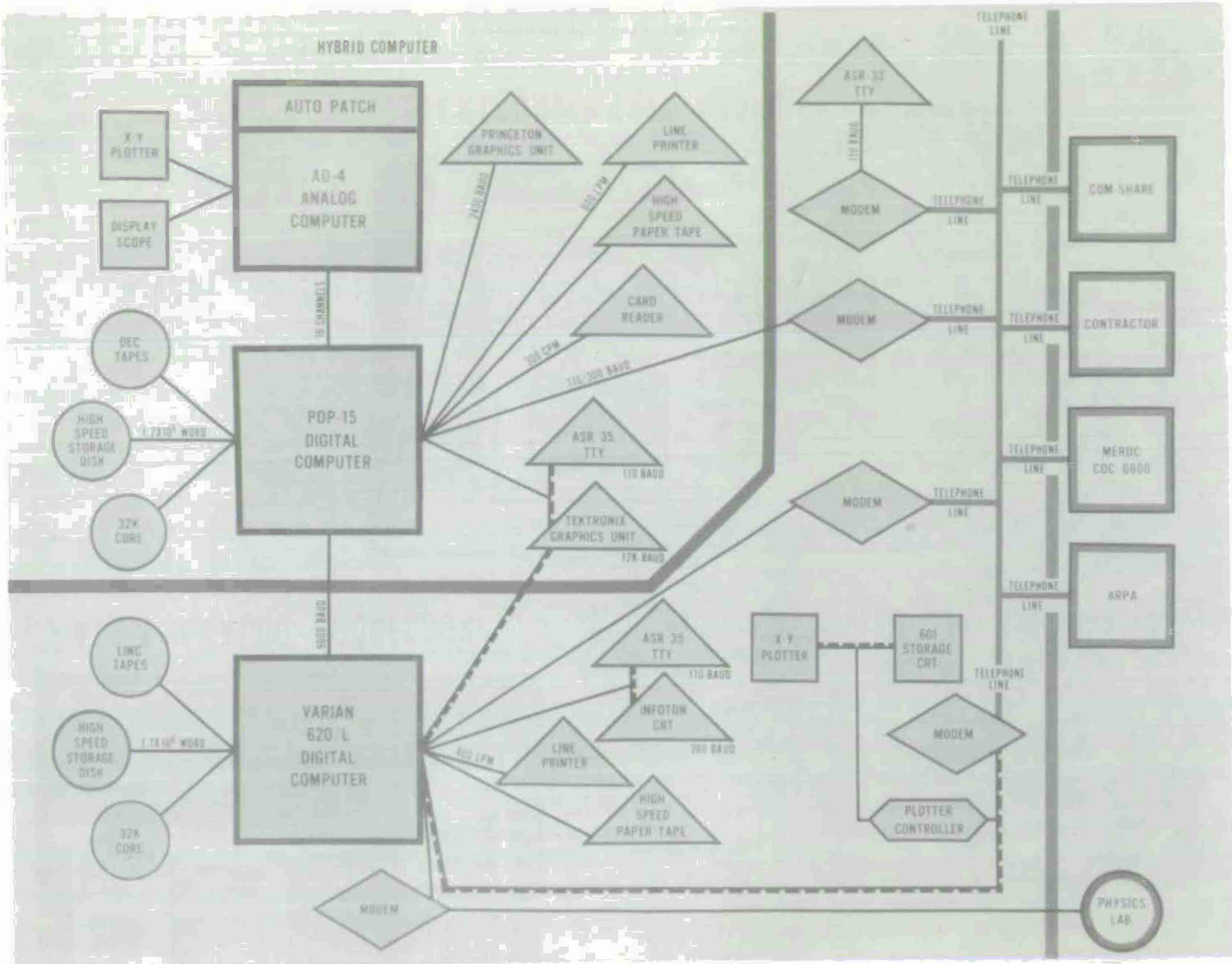


Figure 2. Computer-aided design engineering facility.

1344-74

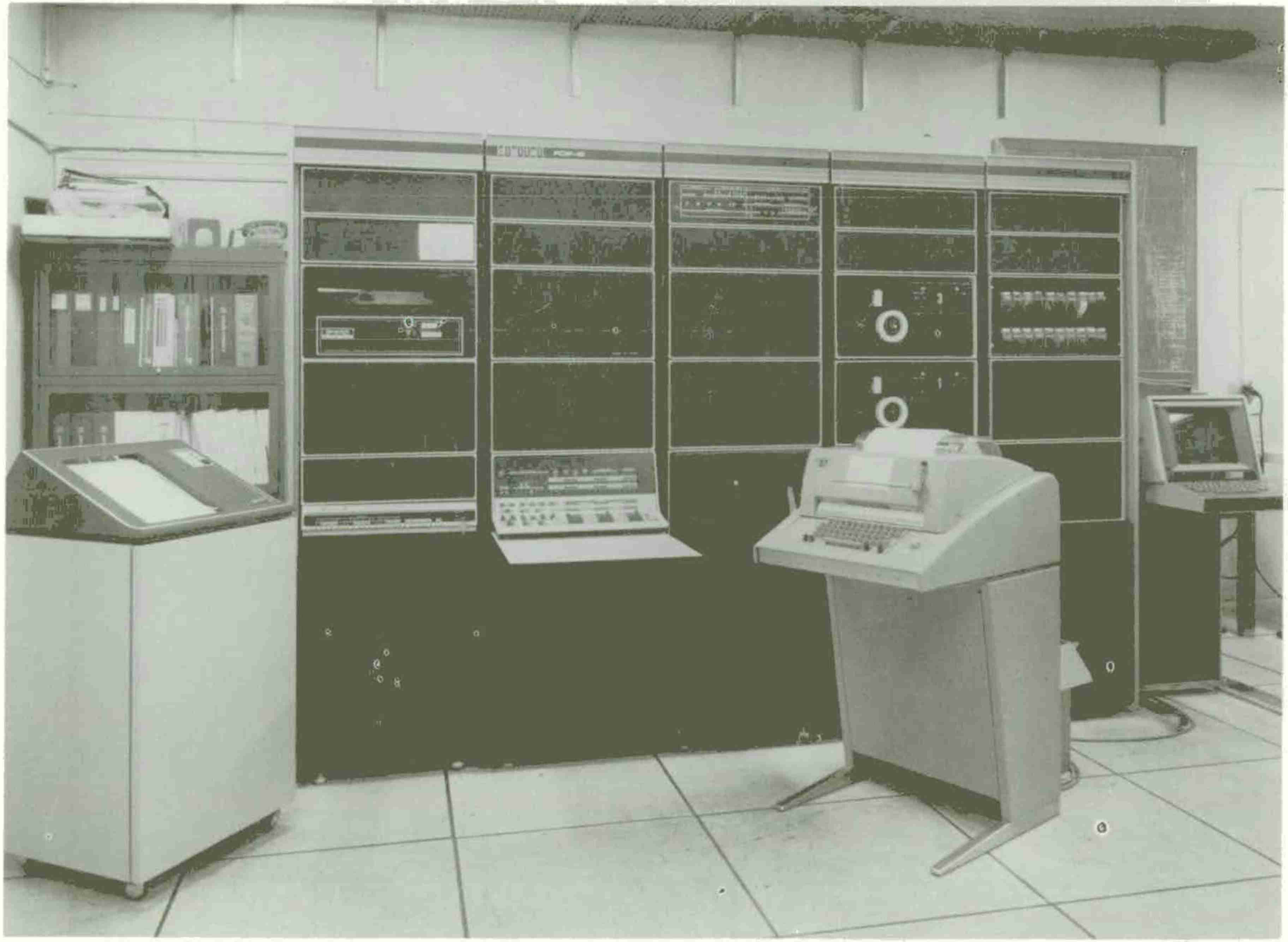


Figure 3. PDP-15/76 digital processor. 13365-74

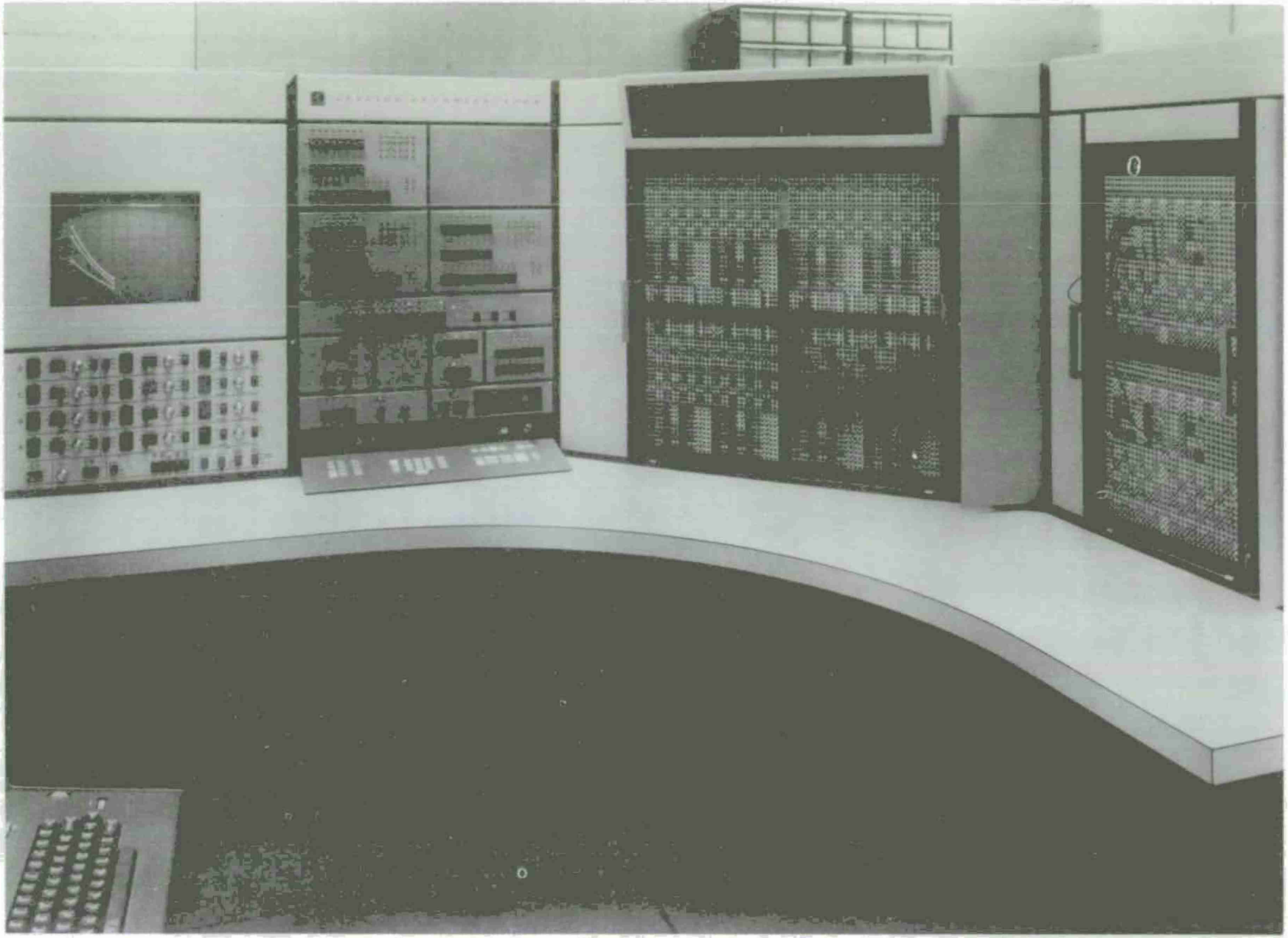


Figure 4. AD-4 analog processor. 13364-74

II. PROGRAM PHILOSOPHY

This report describes a hybrid-computer solution approach to the solution of partial differential equations. However, to understand the reasoning for this method, the pure analog-computer approach to the solution of partial differential equations must be discussed. The technical background for this effort also will be useful to the full understanding of the program philosophy.

4. **Technical Background.** The background of the present work, typical equations, and their method of solution are discussed below.

a. **Status in this Area of Work.** During the early 1960's, much work was accomplished for the solution of partial differential equations on analog computers. With the expected use of hybrid computers, the emphasis was shifted to their utilization. However, the efforts since then have been small, with little to show but theory. In the digital area, work has progressed, mainly because of the easier man/machine interface and because of the efforts of universities and the large computer companies.

b. **Types of Problems.** The Electrical Equipment Division is involved in the solution of partial differential equations for heat transfer and magnetic flux in electric and electronic equipment. As a result, the first problem to be examined and set up will be the diffusion problem and its associated equations. The solution of this type of equation will provide immediate benefits to the Electrical Equipment Division.

c. **Types of Partial Differential Equations.** There are three types of partial differential equations which are representative of a large number of engineering problems encountered:

$$K \frac{\partial \phi}{\partial t} = \nabla^2 \phi + f \qquad \text{(heat equation or diffusion equation),}$$

$$K \frac{\partial^2 \phi}{\partial t^2} = \nabla^2 \phi + f \qquad \text{(wave equation), and}$$

$$K \frac{\partial^2 \phi}{\partial t^2} = \nabla^4 \phi + f \qquad \text{(dynamic structural equation (biharmonic equation)),}$$

where $\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}$ and $\nabla^4 \phi = \frac{\partial^4 \phi}{\partial x^4} + \frac{\partial^4 \phi}{\partial y^4} + \frac{\partial^4 \phi}{\partial z^4} +$

$$+ 2 \left\{ \frac{\partial^4 \phi}{\partial x^2 \, \partial y^2} + \frac{\partial^4 \phi}{\partial x^2 \, \partial z^2} + \frac{\partial^4 \phi}{\partial y^2 \, \partial z^2} \right\} .$$

d. **Usual Methods of Solution.** There are three major techniques of solution: (1) separation of variables, (2) finite difference, and (3) stochastic. Generally, we will use the finite-difference technique because it can handle time-varying boundary conditions and nonlinearities easily. The separation-of-variables technique assumes linearity. For the digital solution, one reduces the partial differential equation to a set of algebraic equations using the finite-difference technique. This means that iterative techniques must be employed to obtain solutions. For the analog solution, one obtains a set of ordinary differential equations using the finite-difference technique.

5. **Analog Approach.** The general approach to be used to solve the two-dimensional Laplace equation, $\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0$, is to use finite differences for one of the space variables and to solve the other variable continuously. On the analog computer, this means we have two choices. We can divide the space such that we solve for a continuous solution as a function of y at each of a series of x-stations, or we can solve for a continuous solution as a function of x at each of a series of y-stations. Basing our calculations on engineering considerations for accuracy, we will try to use only a few stations. This also will reduce the number of analog components. In order to demonstrate solution accuracy and to identify mechanization problems, the first test problem is one that has an exact solution and that is a special case of the more general problem which will be solved as the approach is refined into a programing language.

The interesting general problem for the electrical engineer designing military generators and motors is one which provides the flux or flowline patterns and the equipotential-line patterns of the magnetostatic field in a section of the air gap of the machine. Figure 5 is a diagram of this complicated geometry. Here we need to be able to take care of a complicated geometry with different types of iron and with various boundary conditions. The overall objective is to provide a language which allows the design engineer to draw this picture on the graphic screen, to input the required boundary conditions, to solve the problem on the hybrid computer, and to provide a picture of the desired distributions of flux and potential, displayed on the graphic screen. The first test case is a simplified example, that will allow for an exact solution, which can be used for a comparison of results. Figure 6 is a diagram of a rectangular space used for the first test case.

III. COMPUTER-SOLUTION MECHANICS

6. **Solution Mechanics.** For the test case, we have a rectangular region, and we will investigate the field inside this region when three boundaries are at $\psi=0$ and one is at $\psi=f(x)$. The exact solution for this case is $100\ \psi(x,y) = 100 \sin \frac{\pi x}{a} \cdot \frac{\sinh[\pi(b-y)/a]}{\sinh[\pi b/a]}$,

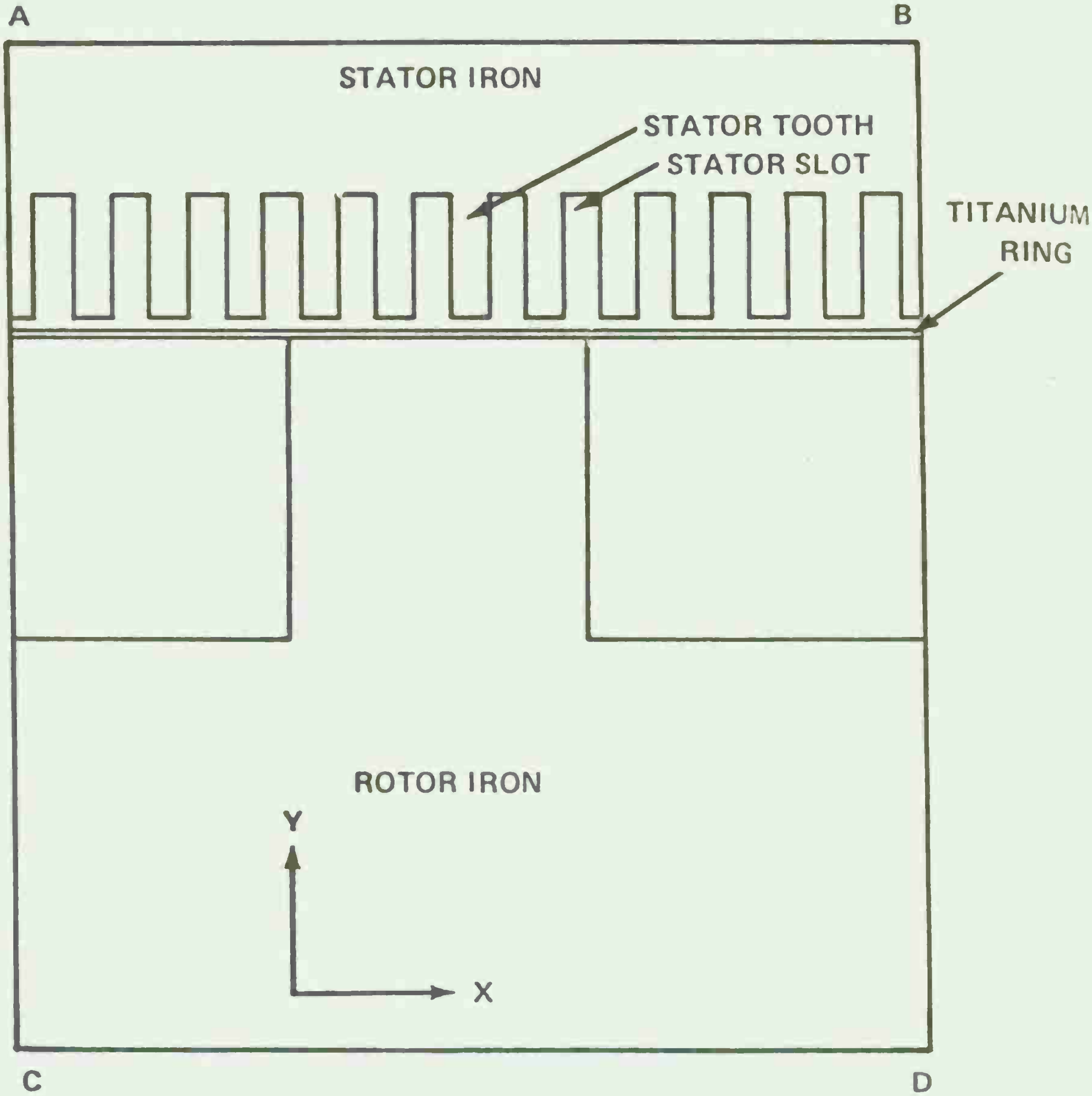


Figure 5. Typical electromagnetic machine geometry.

where a and b are as defined in Figure 6. This solution has been mechanized on the PDP-15 section of the hybrid to provide $\psi(x,y)$ for comparisons. Two analog solutions have been studied (6a and 6b, below).

a. **Continuous x, Discrete y.** In this solution, the analog computer simultaneously solves a set of differential equations at each of a series of y-stations to provide $\psi(x)|_{y_\alpha}$, where α is the station number/location, which will give the value of $\psi(x,y)$ at all points if y is on a station line. Some extrapolation means is assumed: of course, if Δy is small enough, it will not matter. For this solution-method example, we will use six stations in the y-direction. For boundary conditions, we have even derivates (y_o is considered even) specified at the boundaries $y_o = 100 \sin \frac{\pi x}{a}$, and $y_5 = 0$ for all x. Also, y_1, y_2, y_3, and y_4 have a boundary condition of 0 for x=0 and x=a. In Hausner's rules for mechanization (Appendix A), rule 2 states that we should arrange the grid station so that an integer station (y_o, y_5) appears at the boundary since we have even derivatives specified at the boundary. The next Hausner rule (rule 3) says that we should generate high-order derivates with first-order approximations, mechanizing all lower order derivates as summational outputs.

Thus, we let $D_j = \psi_j'' \approx \frac{\psi_{j-1} - 2\psi_j + \psi_{j+1}}{h^2}$ and $\phi_{j-½} = \psi'_{j-½} \approx \frac{-\psi_{j-1} + \psi_j}{h}$, where h is and j is $\phi_{j+½} = \psi'_{j+½} \approx \frac{-\psi_i + \psi_{j+1}}{h}$, so $D_j \approx \frac{-\phi_{j-½} + \phi_{j+½}}{h}$. Thus, we generate five intermediate solutions ($\phi_{1/2}, \phi_{3/2}, \phi_{5/2}, \phi_{7/2}$, and $\phi_{9/2}$) and use eight integrators (Figure 7).

Setting $\frac{\partial^2 \psi_n}{\partial y^2} = \frac{\phi_{n+½} - \phi_{n-½}}{(\Delta y)^2}$, a finite-difference equation for y, in the $\frac{\partial^2 \psi_n}{\partial x^2} = \frac{\partial \psi_n^2}{\partial y^2}$ equation gives us: $\left.\frac{\partial^2 \psi_n}{\partial x^2}\right|_{y_n} = -\left[\frac{\phi_{n+½} - \phi_{n-½}}{(\Delta y)^2}\right]$. Then we can solve for $\psi(x)|_{y_\alpha}$ by using the unscaled equations:

$$\frac{d^2\psi_1}{dx^2} = \frac{\phi_{1/2} - \phi_{3/2}}{(\Delta y)^2}$$

$$\frac{d^2\psi_2}{dx^2} = \frac{\phi_{3/2} - \phi_{5/2}}{(\Delta y)^2}$$

$$\frac{d^2\psi_3}{dx^2} = \frac{\phi_{5/2} - \phi_{7/2}}{(\Delta y)^2}$$

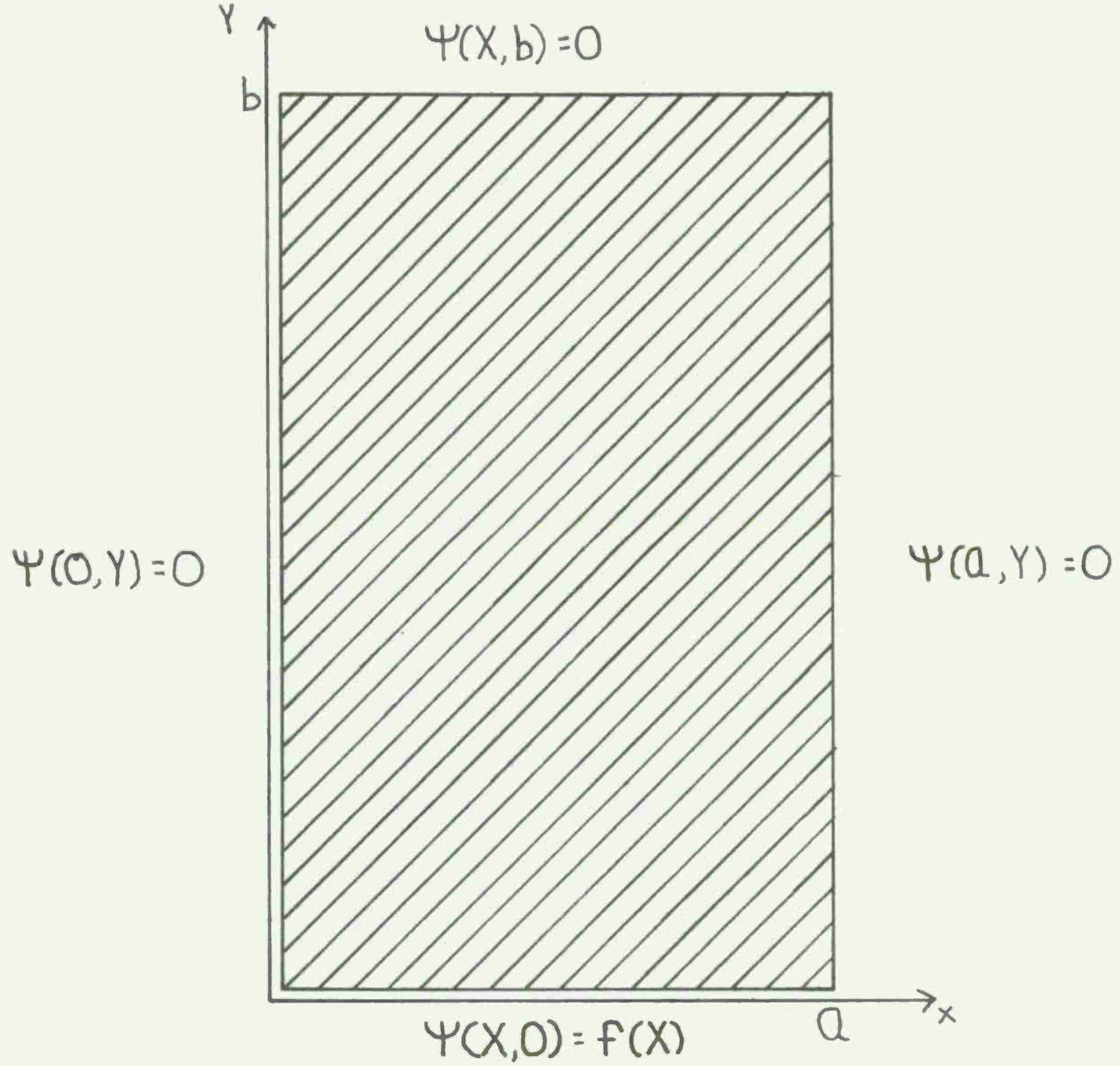


Figure 6. Rectangular space.

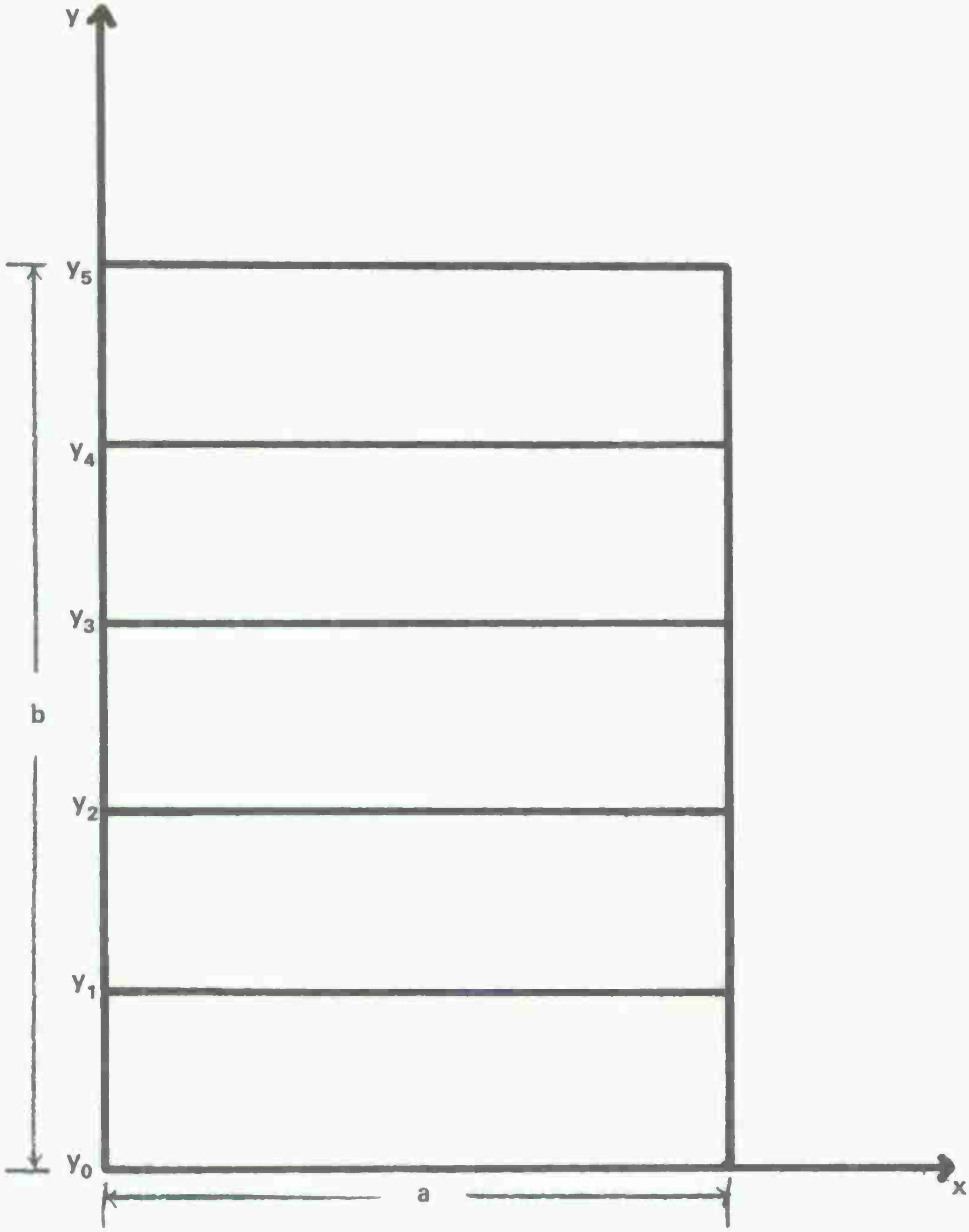


Figure 7. Grid for continuous x, discrete y.

$$\frac{d^2 \psi_4}{dx^2} = \frac{\phi_{7/2} - \phi_{9/2}}{(\Delta y)^2}$$

$$\phi_{1/2} = \psi_1 - 100 \sin \frac{\pi x}{a}$$

$$\phi_{3/2} = \psi_2 - \psi_1$$

$$\phi_{5/2} = \psi_3 - \psi_2$$

$$\phi_{7/2} = \psi_4 - \psi_3$$

$$\phi_{9/2} = 0 - \psi_4$$

For mechanization purposes, we replace t by x in a one-to-one replacement (i.e., 1 second = 1 unit of distance in x).

In the unscaled equation, $\Delta y = \frac{b}{(\text{No. of Stations} - 1)}$, so we have a way to incorporate a and b in the solution. For scaling use the values given in the table are typical.

Variable	Est. Max. Value	Scale Factor	Scaled Computer Variable
ϕ	100 v	$\frac{100}{100}$	$[\phi]$
ψ	100 v	$\frac{100}{100}$	$[\psi]$
ψ'	100 v/s	$\frac{100}{100}$	$[\psi']$

(In our problem as it is set up, the X-generator (Integrator 271) is generating 10 v/s or 0.1 s/v. When we measure 10 volts on X at 10 v/s, we had 1 second, or 1 unit of distance in X, which corresponds to a.) For this problem, we used the initial-condition (IC) pots on the ψ'-integrator to obtain the proper boundary condition for ψ_1 through ψ_4 at x=a. In this problem, we used these pots to make ψ_1, ψ_2, ψ_3, and ψ_4=0 at x=a.

b. **Continuous y, Discrete x.** This method is identical to the continuous x, discrete y method except that the problem space is divided into stations in the x-direction. The problem is solved continuously in the y-direction. This method is discussed in more detail in the examples (section IV).

7. **Special Techniques.** Two special techniques for problem solution may be mentioned.

a. **Dividing Problem Space.** In an effort to minimize equipment and to provide an easy conversion to autopatch, we will divide the problem space into three fixed stations and one variable station. Using symmetry (special case), we get mirror-image solutions in the right half and in the left half of the rectangular space. Therefore, by this consideration, we get 2n–3 solutions for n stations. Using the hybrid-solution control, we will set the variable station at a specified ΔX-spacing from the center station, and a solution will be obtained. Then ΔX will be increased, and the problem will be solved again. This iterative process will be repeated until all specified stations are used. This method allows for linear or nonlinear spacing.

b. **Approaching Boundary Value by Varying IC-Pots.** Another iterative process found to be useful occurs in satisfying the boundary equations. By varying the IC-pots on the ψ-integrators one at a time and in station order from left to right, we can iteratively approach the required boundary value. This method requires that the first pot be varied until the ψ_1-variable equals zero at the prescribed location on the x-axis (x=a) while all other pots are fixed. Then, the second pot is varied until ψ=0 at the same location. This process is repeated sequentially until all variables (ψ_1, ψ_2, ψ_3, and ψ_4) are zero at the same point. This method will be illustrated clearly by the examples, which follow in the next section. Both of the iterative processes described above are performed rapidly by the PDP-15 digital computer.

IV. EXAMPLES

8. **Laplace Equations for Two-Dimensional Solution.** The geometry of this problem dictates use of the continuous y, discrete x solution method. Based on trial solutions, it was determined that six stations are adequate (five fixed and one variable station). Two stations are at the boundaries, x=0 and x=a, where $\psi_o = \psi_5 = 0$. Figure 8 is a diagram of the space, with the variable station shown as a broken line.

For this mechanization, X_o, X_1, X_2, X_3, and X_5 are fixed locations, and X_4 varies. Because of symmetry, X_1 and X_2 will have mirror-image solutions in the right half-space, and X_4 will have mirror-image solutions in the left half-space. Point X_3 is located at $\frac{3a}{6}$, while X_1 is at a/6 and X_2 is at $\frac{2a}{6}$. By symmetry conditions, there will be an identical solution to X_2 at $\frac{4a}{6}$, to X_1 at $\frac{5a}{6}$, and to X_4 at $\frac{(3 \mp K_4)a}{6}$,

with K_4 being specified by the user. For initial conditions along the y=0 boundary,

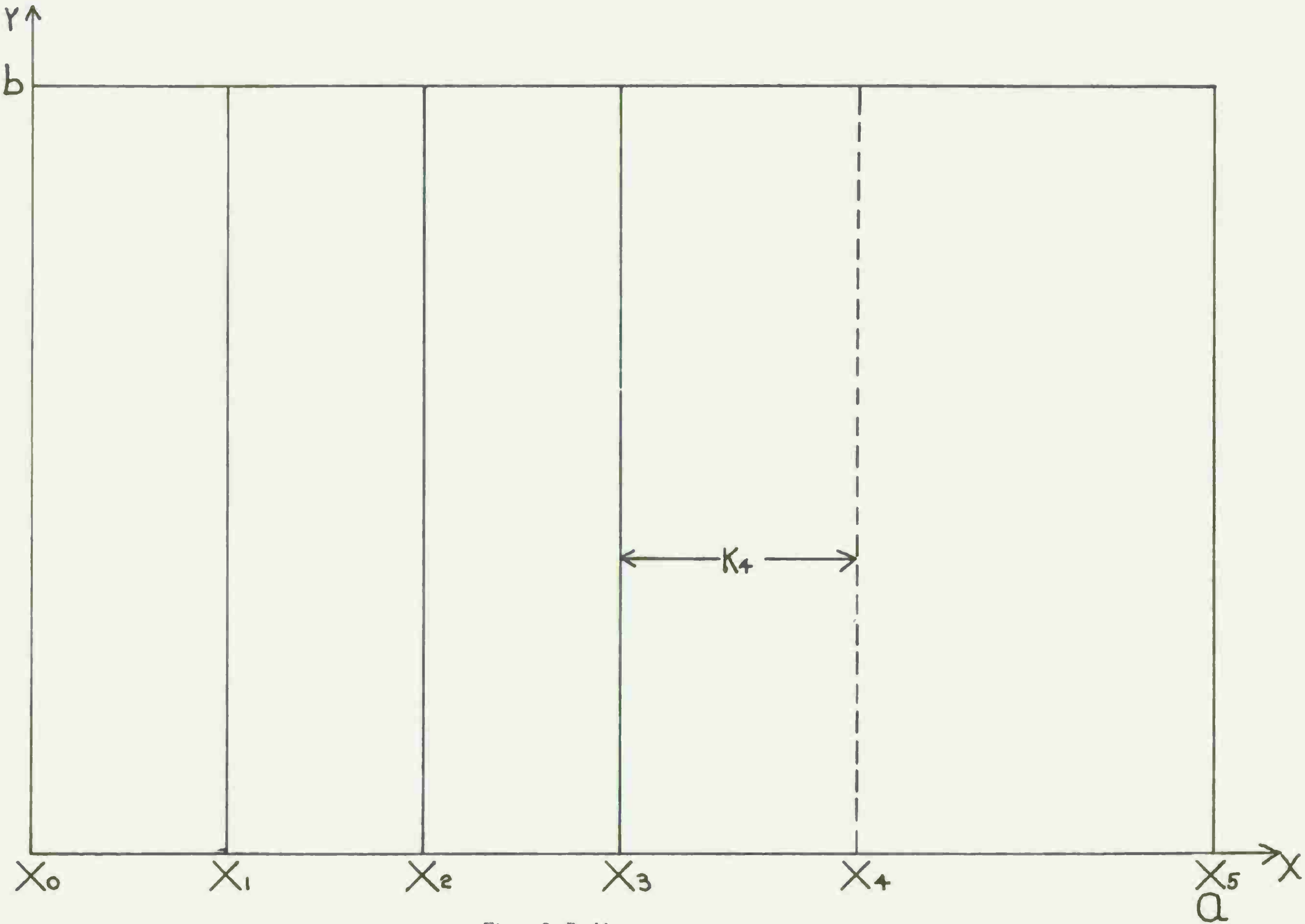


Figure 8. Problem space.

$\psi_o = 100 \sin\left(\frac{\pi}{a}\ (0)\right)$; $\psi_1 = 100 \sin\ \left(\frac{\pi}{a}\ (a/6)\right)$; $\psi_2 = 100 \sin\ \left(\frac{\pi}{a}\ \left(\frac{2a}{6}\right)\right)$; $\psi_3 = 100 \sin\ \left(\frac{\pi}{a}\ \left(\frac{3a}{6}\right)\right)$; $\psi_5 = 100 \sin\ \left(\frac{\pi}{a}\ \left(\frac{6a}{6}\right)\right)$; and $\psi_4 = 100 \sin\left(\frac{\pi}{a}\ \left(\frac{Ka}{6}\right)\right)$; where $K = 3 + K_4$.

When the method described previously was used, it was possible to solve the equations:

Equation			Definition	
ψ_1''	=	$\frac{(\phi_{1/2} - \phi_{3/2})}{\Delta x_{11}}$	$\Delta x_{11} = \frac{a}{6}$	(1)
ψ_2''	=	$\frac{(\phi_{3/2} - \phi_{5/2})}{\Delta x_{21}}$	$\Delta x_{21} = \frac{a}{6}$	(2)
ψ_3''	=	$\frac{(\phi_{5/2} - \phi_{7/2})}{\Delta x_{31}}$	$\Delta x_{31} = ½\left(\frac{a}{6}\right) + ½\,(K_4)$	(3)
ψ_4''	=	$\frac{(\phi_{7/2} - \phi_{9/2})}{\Delta x_{41}}$	$\Delta x_{41} = ½\,(K_4) + ½\,\left(\frac{3a}{6} - K_4\right)$	(4)
$\phi_{1/2}$	=	$\frac{(\psi_1 - \psi_o)}{\Delta x_{12}}$	$\Delta x_{12} = \frac{a}{6}$	(5)
$\phi_{3/2}$	=	$\frac{(\psi_2 - \psi_1)}{\Delta x_{22}}$	$\Delta x_{22} = \frac{a}{6}$	(6)
$\phi_{5/2}$	=	$\frac{(\psi_3 - \psi_2)}{\Delta x_{32}}$	$\Delta x_{32} = \frac{a}{6}$	(7)
$\phi_{7/2}$	=	$\frac{(\psi_4 - \psi_3)}{\Delta x_{42}}$	$\Delta x_{42} = K_4$	(8)
$\phi_{9/2}$	=	$\frac{(\psi_5 - \psi_4)}{\Delta x_{52}}$	$\Delta x_{52} = \frac{3a}{6} - K_4$	(9)

Variable K_4 is defined as follows:

$$K_4 = KR\left(\frac{3a}{6}\right)\ , \tag{10}$$

where KR is the spacing factor.

Changing the equation form, we obtain the following ψ and ϕ values:

$$\ddot{\psi}_1 = \left(\frac{1}{\Delta x_{11}}\right) (\phi_{1/2} - \phi_{3/2}) \tag{11}$$

$$\ddot{\psi}_2 = \left(\frac{1}{\Delta x_{21}}\right) (\phi_{3/2} - \phi_{5/2}) \tag{12}$$

$$\ddot{\psi}_3 = \left(\frac{1}{\Delta x_{31}}\right) (\phi_{5/2} - \phi_{7/2}) \tag{13}$$

$$\ddot{\psi}_4 = \left(\frac{1}{\Delta x_{41}}\right) (\phi_{7/2} - \phi_{9/2}) \tag{14}$$

$$\phi_{1/2} = \left(\frac{1}{\Delta x_{12}}\right) (\psi_1 - \psi_o) \tag{15}$$

$$\phi_{3/2} = \left(\frac{1}{\Delta x_{22}}\right) (\psi_2 - \psi_1) \tag{16}$$

$$\phi_{5/2} = \left(\frac{1}{\Delta x_{32}}\right) (\psi_3 - \psi_2) \tag{17}$$

$$\phi_{7/2} = \left(\frac{1}{\Delta x_{42}}\right) (\psi_4 - \psi_3) \tag{18}$$

$$\phi_{9/2} = \left(\frac{1}{\Delta x_{52}}\right) (\psi_5 - \psi_4) \tag{19}$$

Continuing to change the equation form (since $\psi_o = \psi_5 = 0$), we obtain the following:

$$(\Delta x_{12})\, \phi_{1/2} = \psi_1 \tag{20}$$

$$(\Delta x_{22})\, \phi_{3/2} = \psi_2 - \psi_1 \tag{21}$$

$$(\Delta x_{32})\, \phi_{5/2} = \psi_3 - \psi_2 \tag{22}$$

$$(\Delta x_{42})\, \phi_{7/2} = \psi_4 - \psi_3 \tag{23}$$

$$(\Delta x_{52})\, \phi_{9/2} = (- \psi_4) \tag{24}$$

$$0.01\ \psi_1'' = \left(\frac{0.01}{\Delta x_{11}}\right) \phi_{1/2} - \left(\frac{0.01}{\Delta x_{11}}\right) \phi_{3/2} \qquad (25)$$

$$0.01\ \psi_2'' = \left(\frac{0.01}{\Delta x_{21}}\right) \phi_{3/2} - \left(\frac{0.01}{\Delta x_{21}}\right) \phi_{5/2} \qquad (26)$$

$$0.01\ \psi_3'' = \left(\frac{0.01}{\Delta x_{31}}\right) \phi_{5/2} - \left(\frac{0.01}{\Delta x_{31}}\right) \phi_{7/2} \qquad (27)$$

$$0.01\ \psi_4'' = \left(\frac{0.01}{\Delta x_{41}}\right) \phi_{7/2} - \left(\frac{0.01}{\Delta x_{41}}\right) \phi_{9/2} \qquad (28)$$

$$(\Delta x_{12})\ \phi_{1/2} = (P_{224})\ \psi_1 \qquad (29)$$

$$(\Delta x_{22})\ \phi_{3/2} = (P_{226})\ \psi_2 - (P_{223})\ \psi_1 \qquad (30)$$

$$(\Delta x_{32})\ \phi_{5/2} = (P_{244})\ \psi_3 - (P_{236})\ \psi_2 \qquad (31)$$

$$(K_1 \Delta x_{42})\ \phi_{7/2} = (K_1)(P_{266})\ \psi_4 - (K_1)(P_{247})\ \psi_3 \qquad (32)$$

$$K_1 = \frac{\Delta x_{32}}{\Delta x_{42}} \qquad (33)$$

$$(K_2\ \Delta x_{52})\ \phi_{9/2} = -(K_2\ P_{256})\ \psi_4 \qquad (34)$$

$$K_2 = \frac{\Delta x_{32}}{\Delta x_{52}} \qquad (35)$$

Continuing the rearrangements:

$$P_{232}\ (\Delta x_{12})\ \phi_{1/2} = \left(\frac{0.01}{\Delta x_{11}}\right) \phi_{1/2} \qquad (36)$$

$$P_{245}\ (\Delta x_{22})\ \phi_{3/2} = \left(\frac{0.01}{\Delta x_{11}}\right) \phi_{3/2} \qquad (37)$$

$$P_{227}\ (\Delta x_{22})\ \phi_{3/2} = \left(\frac{0.01}{\Delta x_{21}}\right) \phi_{3/2} \qquad (38)$$

$$P_{233}\ (\Delta x_{32})\ \phi_{5/2} = \left(\frac{0.01}{\Delta x_{21}}\right) \phi_{5/2} \qquad (39)$$

$$P_{243}\ (\Delta x_{32})\ \phi_{5/2} = \left(\frac{0.01}{\Delta x_{31}}\right) \phi_{5/2} \qquad (40)$$

$$K_1 \; P_{253} \; (\Delta x_{42}) \, \phi_{7/2} = \left(\frac{0.01}{\Delta x_{31}} \right) \; \phi_{7/2} \tag{41}$$

$$K_1 \; P_{265} \; (\Delta x_{42}) \, \phi_{7/2} = \left(\frac{0.01}{\Delta x_{41}} \right) \; \phi_{7/2} \tag{42}$$

$$K_2 \; P_{276} \; (\Delta x_{52}) \, \phi_{9/2} = \left(\frac{0.01}{\Delta x_{41}} \right) \; \phi_{9/2} \tag{43}$$

Finally, we obtain the following pot settings:

$$P_{224} = 1 \tag{44}$$

$$P_{226} = 1 \tag{45}$$

$$P_{223} = 1 \tag{46}$$

$$P_{244} = 1 \tag{47}$$

$$P_{236} = 1 \tag{48}$$

$$P_{266} = \frac{\Delta x_{32}}{\Delta x_{42}} \tag{49}$$

$$P_{247} = \frac{\Delta x_{32}}{\Delta x_{42}} \tag{50}$$

$$P_{256} = \frac{\Delta x_{32}}{\Delta x_{52}} \tag{51}$$

$$P_{232} = \frac{0.01}{(\Delta x_{11}) \, (\Delta x_{12})} \tag{52}$$

$$P_{245} = \frac{0.01}{(\Delta x_{11}) \, (\Delta x_{22})} \tag{53}$$

$$P_{227} = \frac{0.01}{(\Delta x_{22}) \, (\Delta x_{21})} \tag{54}$$

$$P_{233} = \frac{0.01}{(\Delta x_{32}) \, (\Delta x_{21})} \tag{55}$$

$$P_{243} = \frac{0.01}{(\Delta x_{32}) \, (\Delta x_{31})} \tag{56}$$

$$P_{253} = \frac{0.01}{(\Delta x_{31})(\Delta x_{42})(K_1)} \quad (57)$$

$$P_{265} = \frac{0.01}{(\Delta x_{41})(\Delta x_{42})(K_1)} \quad (58)$$

$$P_{276} = \frac{0.01}{(\Delta x_{41})(\Delta x_{52})(K_2)} \quad (59)$$

The program will scan the space as previously described, and with four different positions for station X_4 we actually obtain data for 15 equivalent stations as is shown by Figure 9.

In order to obtain the desired plots, it is necessary to perform a core search for a specified ψ-value:

a. Check out the specified X-station and its equivalent image.

b. Use straight-line interpolation between data points.

For example: y value = ITM/10,000, where ITM = b

x-value = x-station location

For a specified X:

a. Start at the maximum ψ-value until ψ in core is less than the specified ψ.

b. Back up one space and check discrete y-values; use linear extrapolation to get specified value x,y data.

9. **Hybrid-Computer Solution.**

a. **General.** The hybrid-computer solution may be illustrated graphically. The problem-space geometry is shown in Figure 8 and the space with the solution grid is shown in Figure 9. The finite-difference equations are shown in Figure 10, and the computer patching diagram is given as Figure 11. A program control flow chart is shown in Figure 12, and the patchboard is shown in Figure 13. Figure 14 shows the logic patchboard.

b. **Computer Program PDR2B.** The computer program is stored in the execute file, PDR2B, in the RMM file on disk. Program listings and subroutines are

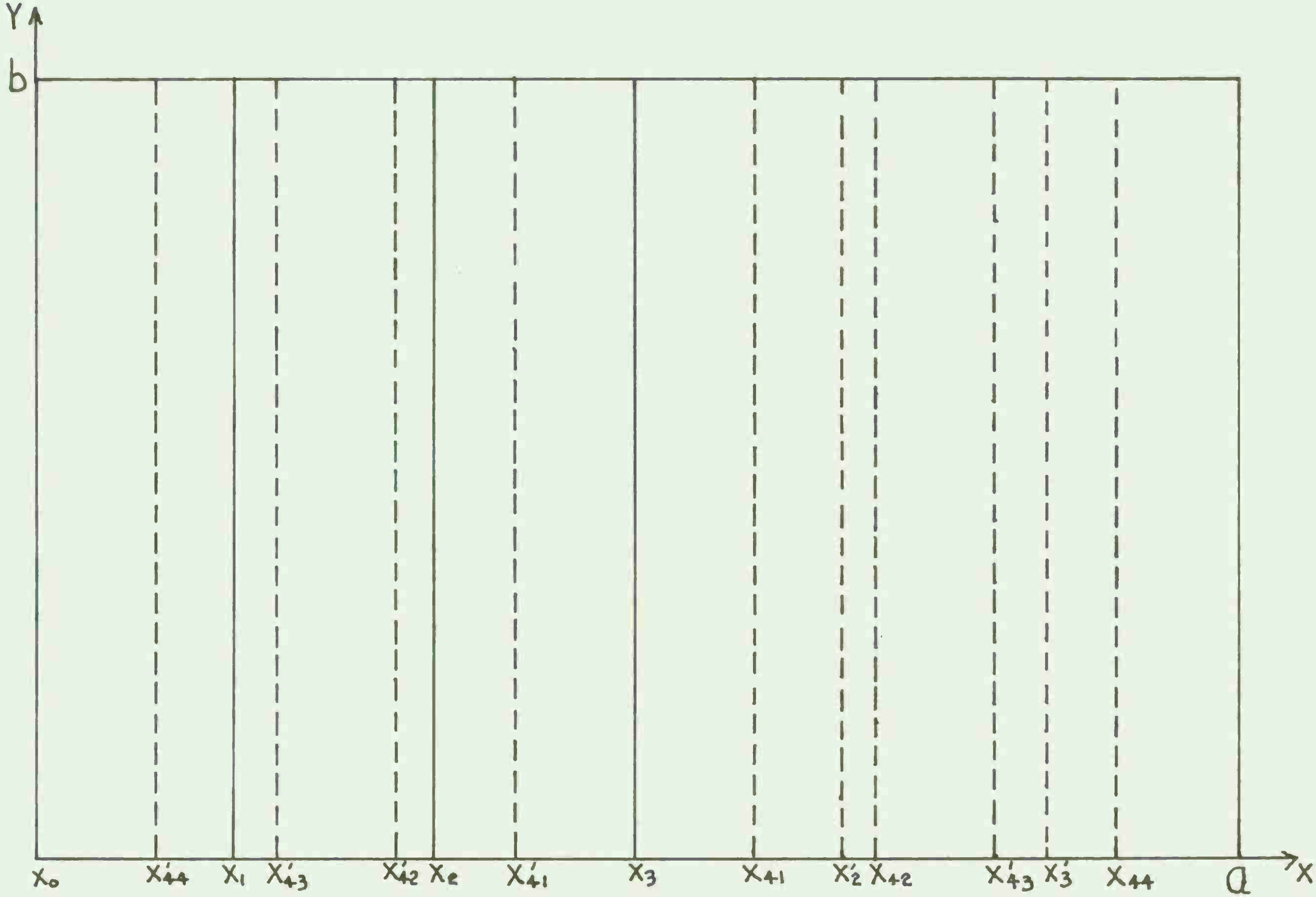


Figure 9. Problem space with grid.

BASIC FINITE DIFFERENCE SCHEME FOR HYBRID COMPUTER

A CHANGE TO AN ORDINARY 2nd ORDER DIFFERENTIAL EQUATION AT EACH X-STATION

$$\ddot{\psi}_1 = \frac{1}{\Delta X_{11}} \left(\phi_{1/2} - \phi_{3/2} \right)$$

$$\ddot{\psi}_2 = \frac{1}{\Delta X_{21}} \left(\phi_{3/2} - \phi_{5/2} \right)$$

$$\ddot{\psi}_3 = \frac{1}{\Delta X_{31}} \left(\phi_{5/2} - \phi_{7/2} \right)$$

$$\ddot{\psi}_4 = \left(\frac{1}{\Delta X_{41}} \right) \left(\phi_{7/2} - \phi_{9/2} \right)$$

WHERE–

$$\phi_{1/2} = \left(\frac{1}{\Delta X_{12}} \right) \left(\psi_1 - \psi_0 \right)$$

$$\phi_{3/2} = \left(\frac{1}{\Delta X_{22}} \right) \left(\psi_2 - \psi_1 \right)$$

$$\phi_{5/2} = \left(\frac{1}{\Delta X_{32}} \right) \left(\psi_3 - \psi_2 \right)$$

$$\phi_{7/2} = \left(\frac{1}{\Delta X_{42}} \right) \left(\psi_4 - \psi_3 \right)$$

$$\phi_{9/2} = \left(\frac{1}{\Delta X_{52}} \right) \left(\psi_5 - \psi_4 \right)$$

Figure 10. Finite-difference equations.

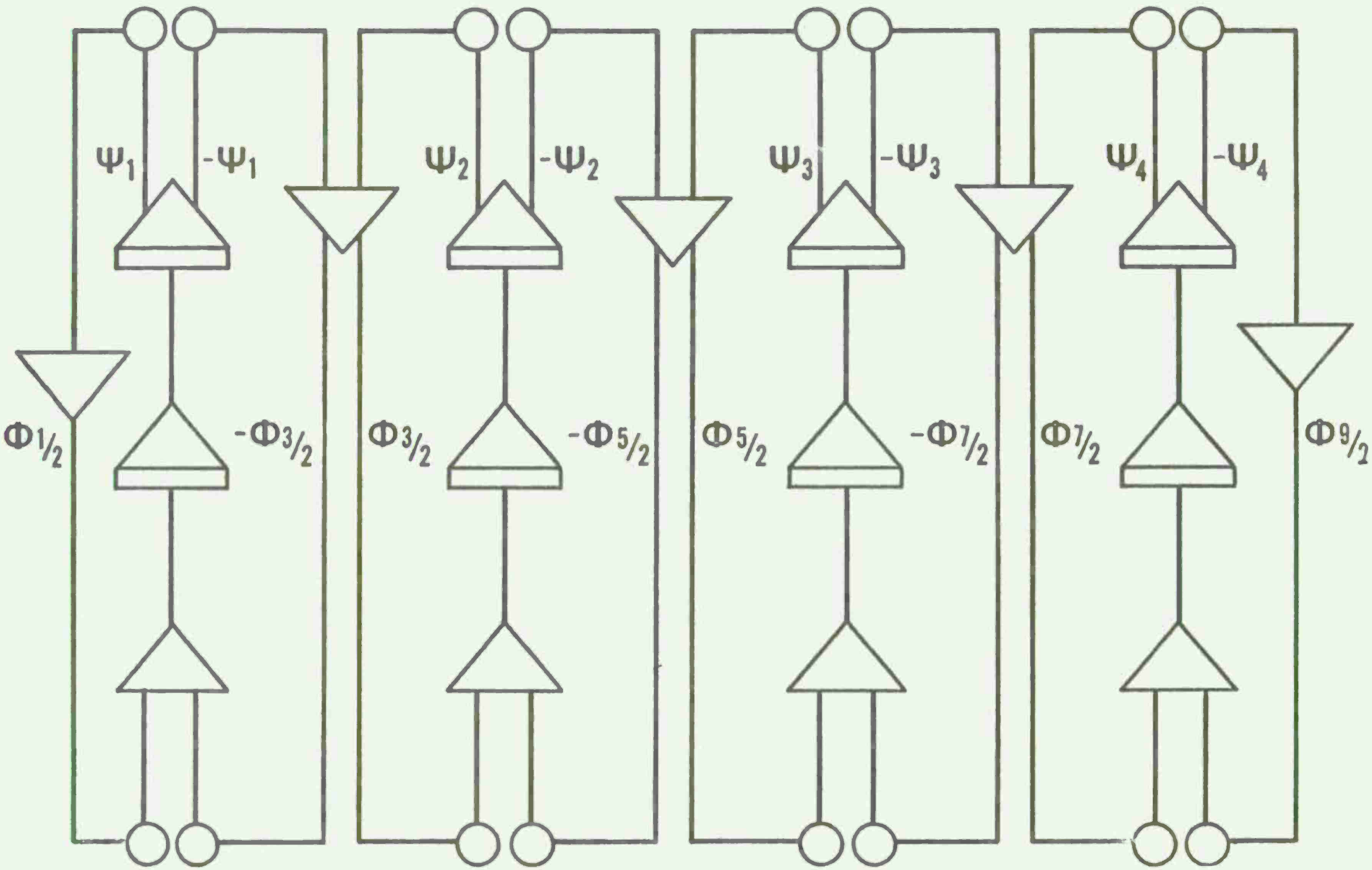


Figure 11. Computer patching diagram.

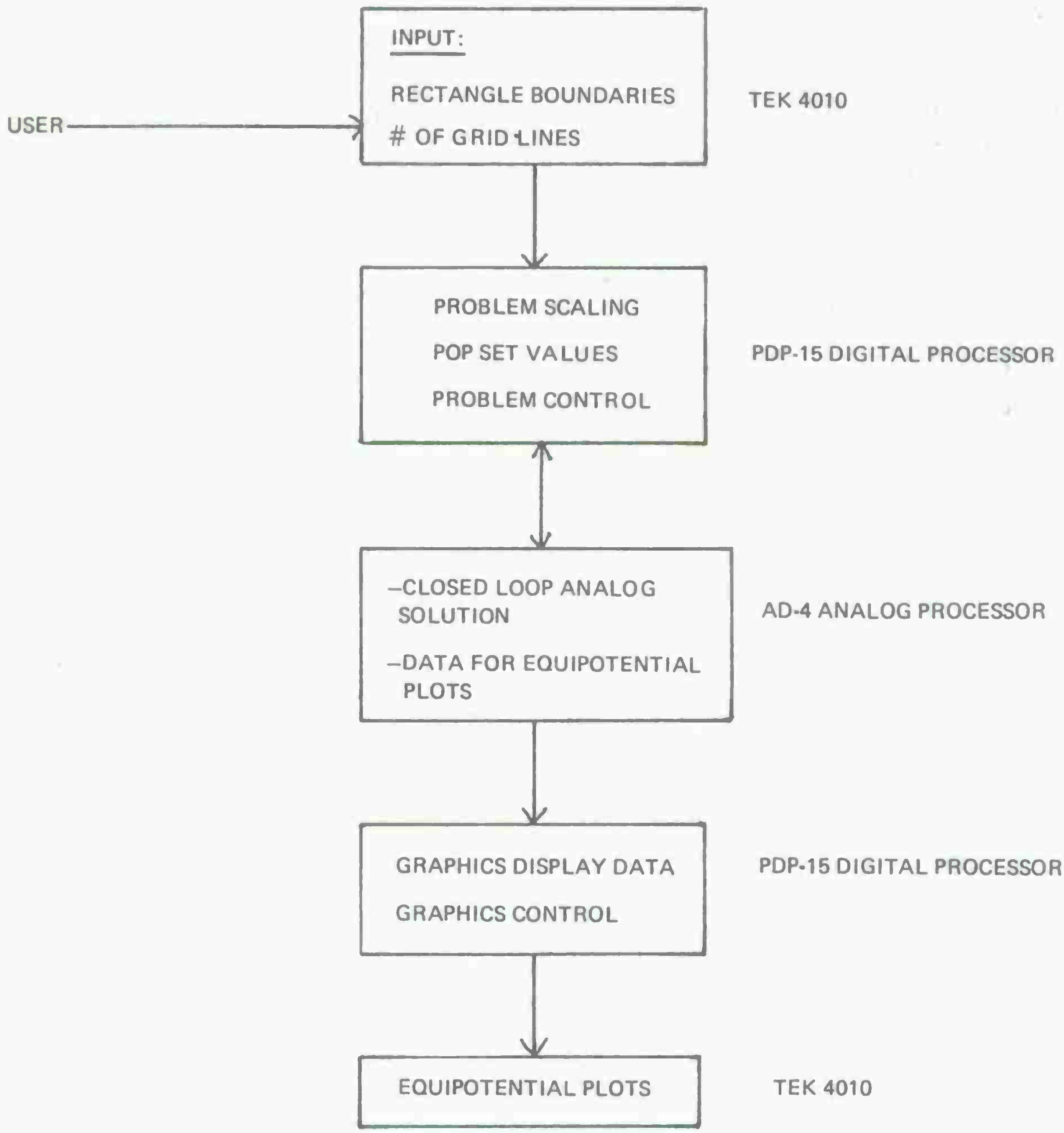


Figure 12. Program flow chart.

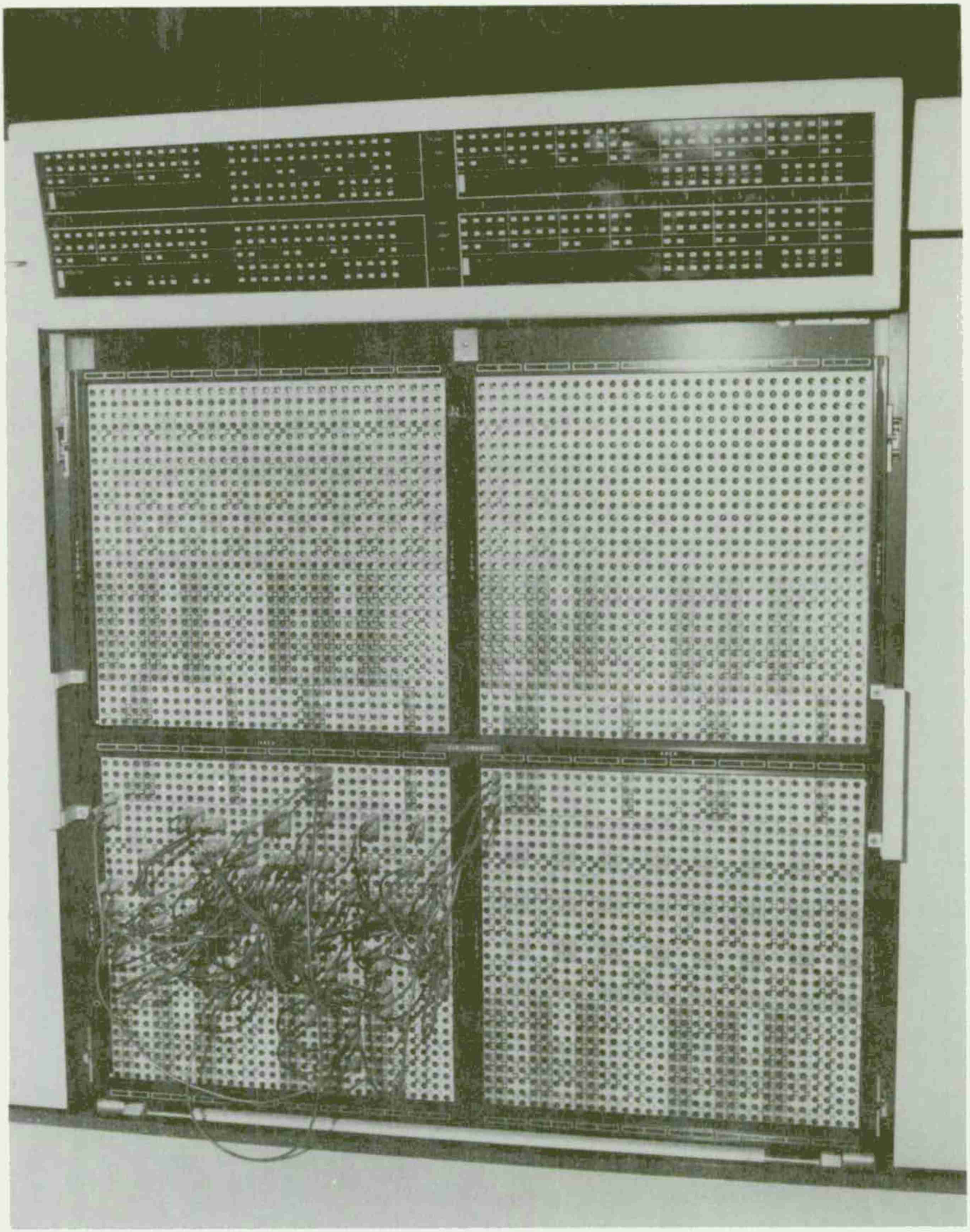


Figure 13. Analog patchboard. 6B-C06697/74

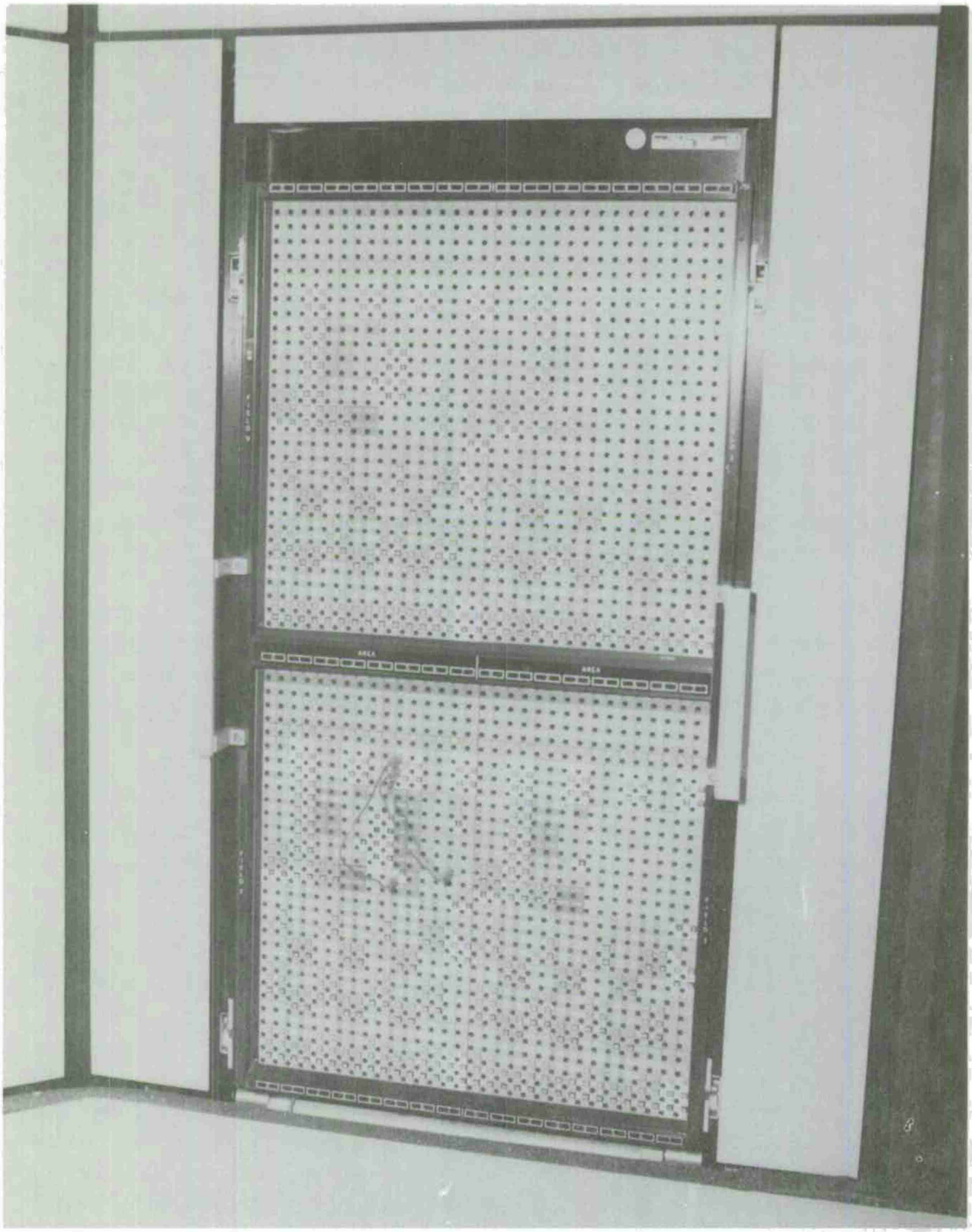
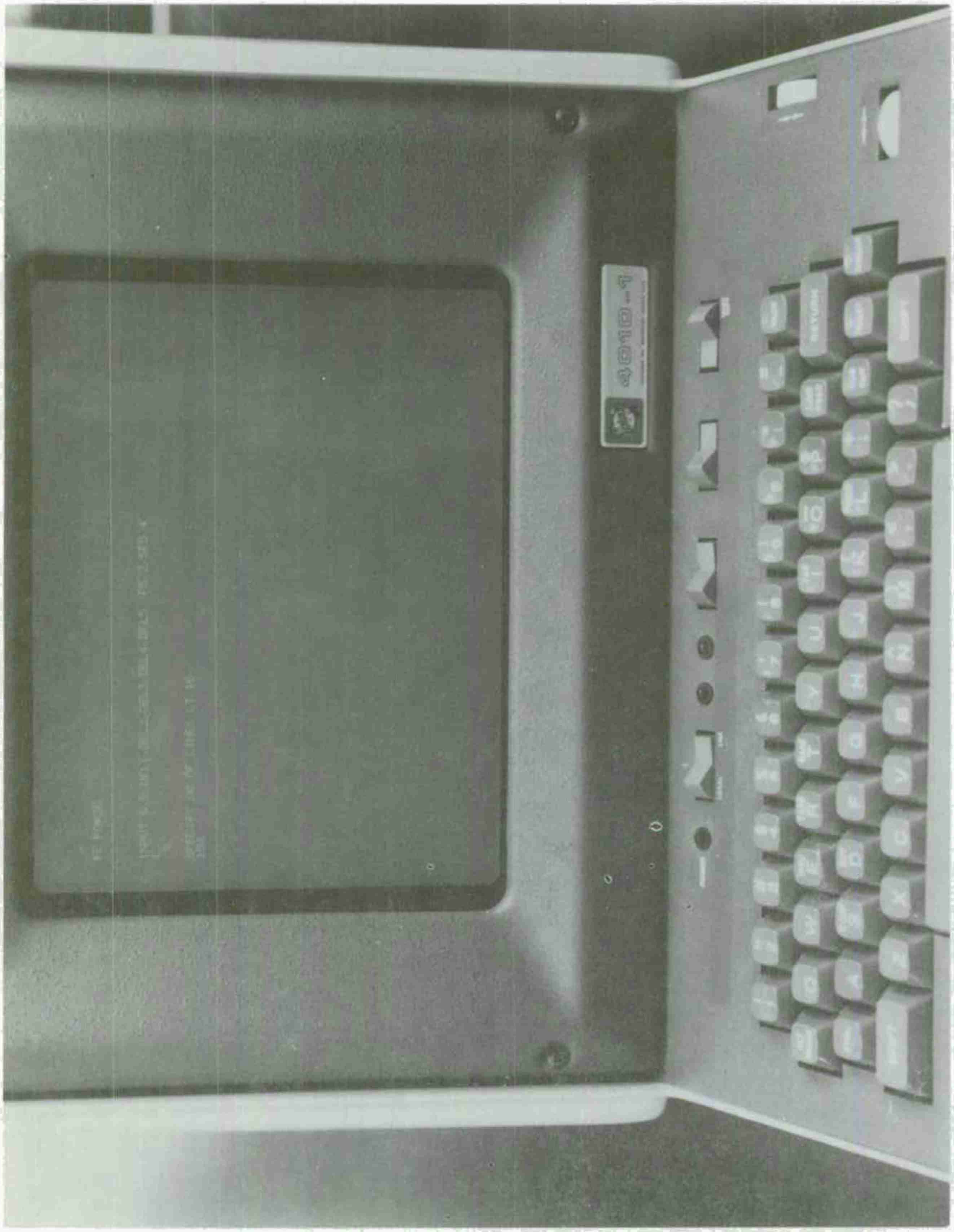


Figure 14. Logic patchboard. 6B-C06696/74

given in Appendix B. The large size of this problem requires "chaining," and the program details are in Appendix C. The following is a description of the use and response of PDR2B. With the PDP-15/AD-4 hybrid up and running, the PDP-15 executive supplies a "$" to indicate user input. To the "$" on the Tektronix 4010, the user types in "E PDR2B." The computer prompting response is a statement for input: "Input A, B, DEL1, DEL2, DEL3, DEL4, DEL5: F5.2, 5F5.4." This allows the user to provide the x and y space dimensions (A and B, respectively). The spacing for the variable grid line, referenced from the center line, is not used. Once this spacing is input, the computer responds with the prompting: "Specify Number of Lines LT16." This allows the user to vary the number of stations for trial solutions. The computer prints the value of DEL (as measured from the center x-station) and the IC-pot values, which are required to satisfy the boundary conditions through the closed-loop, analog iterative process, described in Appendix B. Figure 15 shows the computer prompting. Program solution output is shown by Figures 16 and 17. Figure 18 illustrates the solution with a grid, while Figure 19 depicts the solution without a grid. Normally, for production runs, the problem grid would be well specified; but, for this problem, it was not. Several linear and nonlinear spacings were investigated. It should be noted that the nonlinear grid helps to clarify solution slopes in specific areas of interest. The use of nonlinear grid is optional (i.e., it can be selected as needed). The 16K core of the present PDP-15 digital subsection of the hybrid unit limits us to about 20 grid stations (40 with symmetry), but more would be available if we had written the solution to disk or tape storage and had performed the graphics with another program. Also, the graphics display uses a simplified, point-to-point plotting routine, which could be refined for smoother curves.

The b/a-ratio limits for this method as it is presently programed are between 0.1 and 0.3, mainly because of the assumed scaling. This limitation will be eliminated later, but it is not serious enough to warrant a change for the trial example. Figures 15 through 19, which depict the solution on the Tektronix 4010 Graphic Terminal screen, were used to demonstrate the problem I/O and do not describe accurate solutions. The next set of figures, which is hardcopy output for the Tektronix graphics display, is used to provide the comparison of accuracy between the exact and hybrid solutions for this example. The exact solution uses a mathematical solution subroutine in place of the hybrid subroutine set PDE, MCON, and PDE2 (see Appendix C for more details). All other input and output subroutines stay the same. Using the problem definition parameters (A=1, B=.1) and 10 lines (stations), we can compare results. Note that the computer uses nine lines to divide the right-hand space of the problem into 10 spaces. Figure 20 is the hardcopy output for the hybrid solution, and Figure 21 is the hardcopy output for the exact solution. Appendix C contains $\psi(y)$-data for each X-station generated by the exact and hybrid solutions.



6B-C06689/74

Figure 15. Program computer prompting.

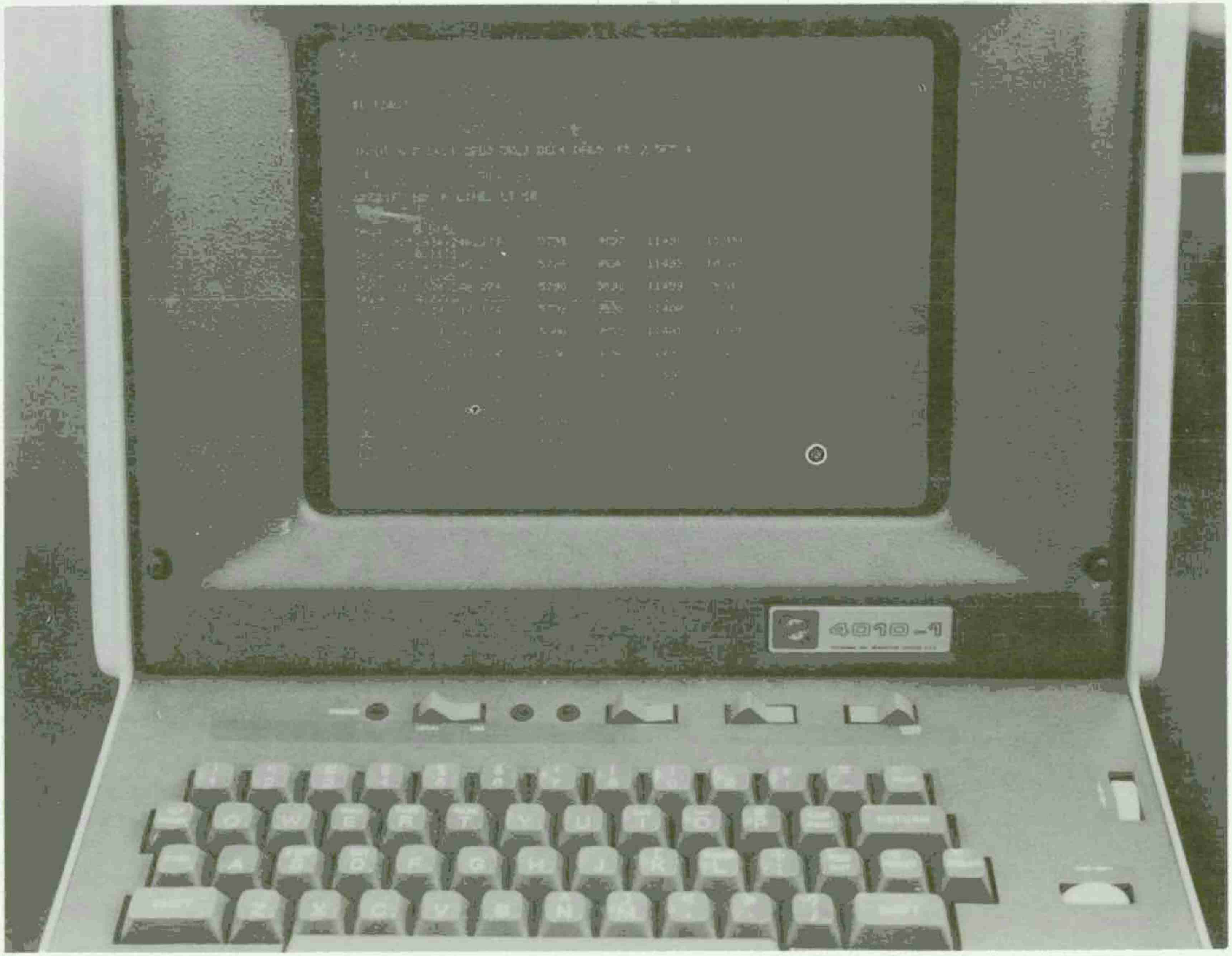


Figure 16. Program solution output (partial).

Figure 17. Program solution output (completed). 6B-C06686/74

Figure 18. Solution with grid.

6B-C06683/74

Figure 19. Solution without grid.

6B-C06679/74

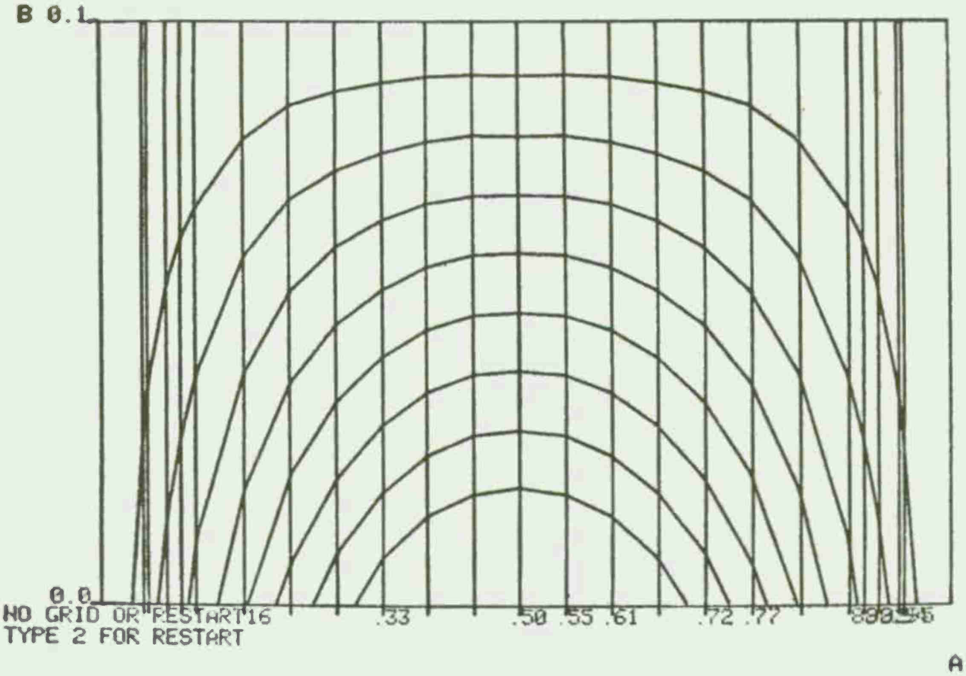


Figure 20. Hardcopy output of hybrid solution.

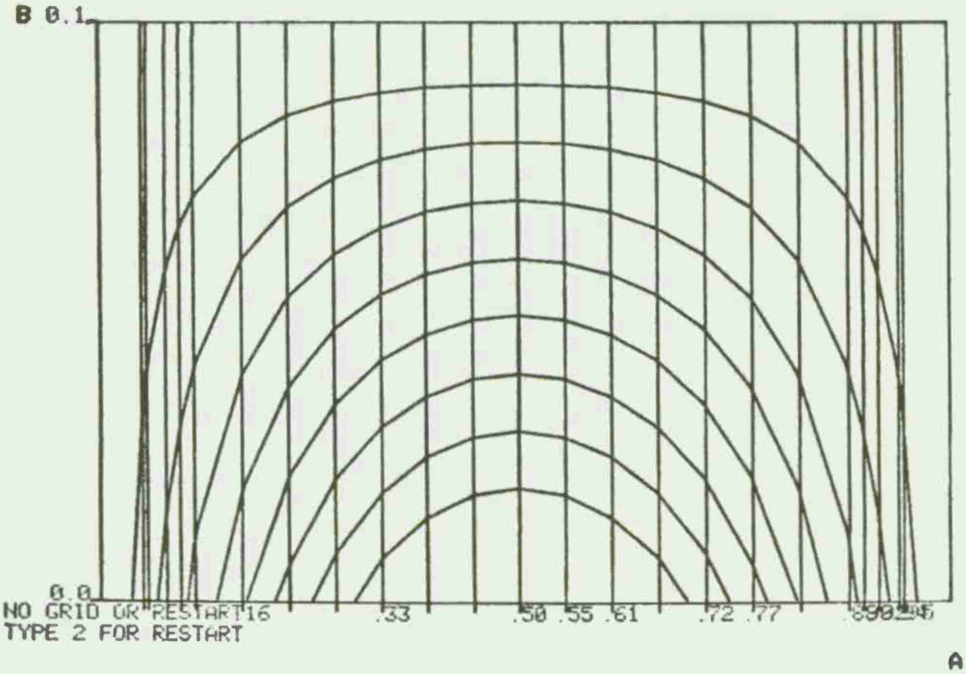


Figure 21. Hardcopy output of exact solution.

In clock time, each hybrid-computer solution set took 30 seconds. (A 33-grid solution, including the symmetry, took about 7 minutes.) The hybrid-computer solution runs 100 times faster than real time and is faster than the exact solution provided by the PDP-15 only. Figures 16 and 17 verify our original assumption: that we could scan the space, while maintaining five stations fixed and one moving, because the first three pot settings (two stations are at the boundary, where ψ=0) always return to the same value at solution; however, the grid station, being moved, changes the pot value.

V. CONCLUSIONS AND FUTURE WORK

10. **Conclusions and Future Work.** So far, we have shown a technique for solving partial differential equations on hybrid computers which is at least 50 times faster than the digital solution. This speed of solution occurs because we solve the problem in a continuous, closed-loop, analog process. Also, we have established an iterative solution technique, which converges rapidly and allows us to maintain overall, simplified digital control over the closed-loop, analog solution process. The comparison of the hybrid solution to the exact analytical solution demonstrates the accuracy of this approach.

The next steps are to generate the problem menus and to solve the field problem for a slot geometry and, then, for other complex geometries. The progress demonstrated to date offers an optimistic outlook for complete success in the future planned work of this project.

APPENDIX A

HAUSNER'S* RULES FOR MECHANIZATION

The following is a list of Hausner's Rules used in this project:

Rule 1 – To obtain a kth-order solution, all approximations must be kth order, including those accounting for boundary conditions.

Rule 2 – If only even derivatives of a dependent variable (such as u, u″, u″″, etc.) are specified at a boundary, arrange the grid stations so that an integer station (say, X_0 or X_1) appears at a boundary. If at least one odd derivative (u′, u‴, etc.) is specified at a boundary, a half-integer station (say, $X_{1/2}$) should be placed at a boundary.

Rule 3 – Generate high-order derivates with first-order-derivative approximations, mechanizing all lower order derivatives as summational outputs.

* A. Hausner, "Analog and Analog/Hybrid Computer Programing," Prentice-Hall 1971, pp 435-436.

APPENDIX B

ANALOG CONTROL ROUTINES

A brief discussion of the analog control routines used to reach solutions is given in this appendix.

B-1. Differentiation with Respect to y. The analog computer actually performs $\frac{d\psi}{dy}$ as $\frac{d\psi}{dt}$, where y is represented as t on a one-to-one basis. The time-base (or y-base) generator, integrator 271, normally is providing 10 v/s; thus, we get 0.1 s/v as the output. Since 1 unit of y is equivalent to 1 second, it takes 0.2 second to provide 0.2 unit of y. This means that the integrator output is 2 volts in 0.2 second (0.1 s/v · 2 volts = 0.2 second). In order to provide the proper output rate for integrator 271, pot 273 is set to 0.01 with 100 volts input. The normal integrator rate is 10 v/s in quadrant two of the analog patchboard.

B-2. Closed-Loop Analog Solution. The fastest possible solution is obtained when the analog computer operates in a closed-loop fashion. The solution control is accomplished as follows: (1) The user provides input parameters to the digital unit; (2) the digital computer uses these parameters to automatically scale the problem, to set the analog comparator pot settings for time (or b) value in order to place the computer in hold, and to set the pots and start the solution; (3) the digital unit waits a sufficient time in order to allow the analog unit to go to "hold," checks end-point values for convergence, resets the computer to run again, and repeats this until convergence occurs; (4) once convergence occurs, the digital unit resets the computer and causes the analog unit to operate for a set number of predetermined increments, at which points the analog comparator places the computer into the "hold" mode and the digital unit samples and stores ψ-, y-, and x-data; (5) this process is repeated until all specified x-stations have been used; (6) once all x-stations have been used, the digital unit asks the user to specify ψ, $\Delta\psi$, and the number of lines to plot; and (7) the digital unit uses these data to search its stored ψ-, y-, and x-data and to provide the plot. The digital unit is programed to provide many variations of the plotting, once the hybrid unit has finished computing, in order to keep from having to recompute each time a new plot variation is needed.

B-3. Analog Comparator Logic. The logic and analog patching needed to accomplish the time (or y-) control is shown by Figure 11. The output of integrator 271 is fed through pot 277 to amplifier 233. The output of amplifier 233 goes to comparator

231 on the analog patchboard. The reference voltage (equivalent to y=b) comes from amplifier 223, which is the other input to comparator 231. When the sum of the inputs goes positive (occurs at the instant y becomes infinitesimally larger than b), a logic 1 is generated by the out-point on the logic patchboard. Since "out" on comparator 231 is connected to SYS Hold, it receives a logic 1, which places the analog unit in the "hold" mode, thus stopping computation. In order to reset properly, the digital unit overrides the patched "hold" mode by a "hold" command, reads the desired ψ-value, places the computer in the IC-mode, and resets the comparator output to logic 0 by setting pot 237 to 0. For the sampling of ψ-, x-, and y-data after convergence tests are met, pot 237 is incremented to the preset values, thus stopping the computer at the desired points of y, reading the data, and continuing to the next point as soon as pot 237 is updated. This process is limited to 12 data points because of the dimension statement, which reflects present core limits. Methods that would allow more points could be used but are not required for the test example.

B-4. Iteration Control. The computer is programed to set all four IC-pots (for the first derivative of ψ) initially and, then, to go through a preset sequence to set the first IC-pot until ψ_1 goes to 0 at y=b. The computer then goes to the second IC-pot and changes it until $\psi_2=0$ at y=b. Each time, all ψ's are sampled to see if they are simultaneously 0 at y=b. This process continues to ψ_3, ψ_4, ψ_1, ψ_2, ψ_3, ψ_4, etc. until $\psi_1=\psi_2=\psi_3=\psi_4=0$ at y=b. This process generally converges in less than 30 seconds (about 10 iterations at most).

APPENDIX C

COMPUTER PROGRAM LISTINGS

C-1. Introduction. This appendix gives a listing of the problem source code, the chaining routine, and the several programs used in this study and depicts the program flowcharts.

The hybrid-computer program consists of a main program (designated subroutine POT) and eight subroutines: PDE, MCON, CON, READSI, PDE2, DISK, DRW, and DRWA. The hybrid-computer program also requires the hybrid routines and the Tektronix routines. The problem requires "chaining" on the 16K core configuration of the PDP-15. The chaining routine produces the XCT and XCU files and allows the program to be run by using E_____PDR2B.

C-2. Hybrid Program Listings. The following listings are the routines used for the hybrid-computer solution.

8/26/74

```
C       SUBROUTINE POT
C       THIS WILL ACT AS THE MAIN PROGRAM
        DIMENSION TST(2)
        COMMON/W/Y(18),IPSI
        COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
        COMMON/POTV/P(20),DEL,A
        COMMON/DIM/B,IBR
        COMMON/P1/NA,JK,KZ
        COMMON/GRD/NPSIG,NTF
        DATA TST(1),TST(2)/3HTST,4H SRC/
        JK=4
        NA=1
        CALL STIND(IE,2237,0)
        WRITE(4,601)
        READ(4,600)A
        READ(4,600)B
        IBR=IFIX(10000.*B)
        WRITE(4,2051)
2051    FORMAT(1X,25HSPECIFY NO OF LINES LT 16)
        READ(4,6004)NLINES
6004    FORMAT(I2)
        DELTX=.5/(FLOAT(NLINES))
        DELTA(1)=DELTX
        NTF=NLINES-1
        DO 2050 NT=2,NTF
        DELTA(NT)=DELTA(NT-1)+DELTX
2050    CONTINUE
        DELTA(1)=1./18.
        DELTA(2)=2./18.
        DELTA(14)=8.6/18.
        DELTA(3)=4./18.
        DELTA(4)=5./18.
        DELTA(15)=8.7/18.
        DELTA(5)=7./18.
        DELTA(6)=7.3/18.
        DELTA(7)=7.6/18.
        DELTA(8)=8./18.
        DELTA(9)=8.1/18.
        DELTA(10)=8.2/18.
        DELTA(11)=8.3/18.
        DELTA(12)=8.4/18.
        DELTA(13)=8.5/18.
601     FORMAT(1X,'INPUT A,B,DEL1,DEL2,DEL3,DEL4,DEL5 :F5.2,5F5.4')
900     CONTINUE
697     FORMAT(1X,23HINPUT  A,DEL: F5.2,F5.4)
        WRITE(4,11)DELTA(NA)
        DEL=DELTA(NA)
11      FORMAT(1X,'DEL=',F10.4)
        CR=2.*DEL
        C4=CR*A+(3./6.)
        DX11=A/6.
        DX21=A/6.
        DX31=(A+(6.*C4))/12.
        DX41=(C4/2.)+((3.*A/6.)-C4)/2.
        DX12=A/6.
        DX22=A/6.
        DX32=A/6.
```

```
      DX42=C4
      DX52=(A/2.)-C4
      C1=DX32/DX42
      C2=DX32/DX52
602   FORMAT(1X,6(1X,F10.4))
      P(1)=1.
      P(2)=1.
      P(3)=.01/(DX11*DX12)
      P(4)=SIN(3.14159/6.)
      P(5)=.01/(DX11*DX22)
      P(6)=1.
      P(7)=SIN(3.14159*2./6.)
      P(8)=1.
      P(9)=.01/(DX22*DX21)
      P(10)=.01/(DX32*DX21)
      P(11)=SIN(3.14159*3./6.)
      P(12)=.01/(DX32*DX31)
      P(13)=1.
      P(14)=DX32/DX42
      P(15)=.01/(DX31*DX42*C1)
      P(16)=DX32/DX52
      P(17)=SIN(3.14159*(1.+CR)/2.)
      P(18)=.01/(DX41*DX42*C1)
      P(19)=DX32/DX42
      P(20)=.01/(DX41*DX52*C2)
      IF(P(14).LT.1.5)GO TO 698
      PTOT1=P(14)*P(15)
      P(14)=1.
      P(15)=PTOT1
      PTOT2=P(19)*P(18)
      P(19)=1.
      P(18)=PTOT2
698   CONTINUE
      IF(P(16).LT.1.5)GO TO 699
      PTOT3=P(16)*P(20)
      P(16)=1.
      P(20)=PTOT3
699   CONTINUE
      DO 700 NP=1,20
C     WRITE(4,6010)P(NP)
700   CONTINUE
600   FORMAT(F5.2,F5.4)
6010  FORMAT(1X,F10.4)
      CALL PDE
      CALL MCON
      CALL PDE2
      NA=NA+1
      JK=JK+1
      IF(NA.GT.NTF)GO TO 3000
      GO TO 900
3000  CONTINUE
      CALL DISK
2021  FORMAT(1X,T4,'ITM',T14,'X1',T22,'X2',T30,'X3',T38,
     1 'X4',T46,'X5',T54,'X6',T62,'X7',T70,'X8')
C     SEARCH FOR SPECIFIED PSI FOR EQUIPOT PRINTOUT
4003  CONTINUE
      WRITE(4,2006)
```

```
2006    FORMAT(1X,11HSPECIFY PSI)
        READ(4,1009)PSI
        READ(4,1009)PSID
        READ(4,1021)NPSI
1021    FORMAT(I2)
        NPSIG=0
4002    CONTINUE
        DO 4000 IZR=1,NPSI
        IPSI=IFIX(PSI*100.)
2022    FORMAT(1X,T9,'PSI',T20,'Y',T31,'X',T42,'XI')
        K=1
        NTF3=NTF+3
        DO 1000 N=1,NTF3
        DO 1001 NA=1,KZ
        IF(ISTA(N,NA).LT.IPSI)GO TO 1002
1001    CONTINUE
1002    NB=NA
        NC=NB-1
        DELSTA=FLOAT(ISTA(N,NC)-ISTA(N,NB))
        DELTM=FLOAT(ITM(NC)-ITM(NB))
        Y(K)=(FLOAT(ITM(NC))-(DELTM*((FLOAT(ISTA(N,NC)-IPSI)/DELSTA))))/
     $ 10000.
        X=XLOC(N)
        XI=A-XLOC(N)
        K=K+1
1000    CONTINUE
1009    FORMAT(2F10.3,I2)
1010    FORMAT(1X,4(1X,F10.3))
        IF(IZR.GT.1)GO TO 4001
        IF(NPSIG.GT.0)GO TO 4001
        CALL DRW
4001    CONTINUE
        CALL DRWA
        PSI=PSI+PSID
4000    CONTINUE
        PSI=PSI-(FLOAT(NPSI)*PSID)
        NPSIG=1
        WRITE(4,1011)
        WRITE(4,1012)
1011    FORMAT(1X,18HNO GRID OR RESTART)
1012    FORMAT(1X,18HTYPE 2 FOR RESTART)
        READ(4,1021)MST
        IF(MST.EQ.2)GO TO 4003
        GO TO 4002
        STOP
        END
```

```
      SUBROUTINE PDE
C     PROGRAM PDE**122573**
      DIMENSION IPT(20),IPTV(20),ADEL(18)
      COMMON/P1/NA,JK,KZ
      COMMON/POTV/P(20),DEL,A
      COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
      DATA IPT/2224,2223,2232,2235,2245,2236,2230,2226,2227,2233,
     1 2242,2243,2244,2247,2253,2256,2264,2265,2266,2276/
      CALL LEX(IE,1)
      CALL TSCAL(IE,0)
      CALL LOAD(IE)
600   FORMAT(F5.2,5F5.4)
405   CONTINUE
      CALL LEX(IE,1)
      DEL=DELTA(NA)
      XLOC(1)=A/6.
      XLOC(2)=2.*A/6.
      XLOC(3)=3.*A/6.
      ADEL(JK)=DEL
      XLOC(JK)=3.*A/6.+DEL
      DO 750 IPV=1,20
      IPTV(IPV)=IFIX(10000.*P(IPV))
750   CONTINUE
      CALL INITA(IE,0)
      CALL CONSO(IE,0)
      CALL LEX(IE,1)
      CALL TSCAL(IE,0)
      CALL LOAD(IE)
5     CONTINUE
      CALL STIND(IE,2277,10000)
      CALL STIND(IE,2275,0)
      DO 10 K=1,20
      CALL STIND(IE,IPT(K),IPTV(K))
10    CONTINUE
C     SET TIME BASE
      CALL STIND(IE,2273,1000)
      CALL READ(IE,0200,IDUM)
      CALL LOAD(IE)
C     WRITE(4,2000)
2000  FORMAT(1X,27HSET IC POTS 260,261,262,263)
      RETURN
      END
```

```
      SUBROUTINE MCON
      INTEGER PSI(100)
      COMMON IJ,IK,IIJ,IDELX
      CALL INITA(IE,0)
      CALL CONSO(IE,0)
      CALL TSCAL(IE,2)
      K=1
200   IJ=2225
      IK=2234
      IL=2246
      IM=2274
      IIJ=0201
      IF(K.GT.1)GO TO 206
      IX=5000
      CALL STIND(IE,IJ,IX)
      CALL STIND(IE,IK,IX)
      CALL STIND(IE,IL,IX)
      CALL STIND(IE,IM,IX)
206   CALL CON(IX,PSI,I,J)
      LX1=IX
      IJ=2234
      IIJ=0221
201   IF(K.EQ.1)GO TO 207
      GO TO 208
207   IX=5000
      GO TO 209
208   IX=LX2
209   CALL CON(IX,PSI,I,J)
      LX2=IX
      IJ=2246
      IIJ=0241
202   IF(K.EQ.1)GO TO 210
      GO TO 211
210   IX=5000
      GO TO 212
211   IX=LX3
212   CALL CON(IX,PSI,I,J)
      LX3=IX
      IJ=2274
      IIJ=0261
203   IF(K.EQ.1)GO TO 213
      GO TO 214
213   IX=5000
      GO TO 215
214   IX=LX4
215   CALL CON(IX,PSI,I,J)
      LX4=IX
      K=K+1
      IJ=2225
      IIJ=0201
      IX=LX1
      CALL READSI(IX,PSI,I,J,IB)
      IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 220
      GO TO 226
220   IJ=2234
      IIJ=0221
      IX=LX2
```

```
      CALL READSI(IX,PSI,I,J,IB)
      IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 225
      GO TO 208
225   IJ=2246
      IIJ=0241
      IX=LX3
      CALL READSI(IX,PSI,I,J,IB)
      IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 235
      GO TO 211
226   GO TO 206
235   CONTINUE
C     PAUSE
      CALL IC(IE)
      CALL STIND(IE,2277,0)
      CALL READ(IE,0200,IVDUM)
      CALL READ(IE,2222,IVDUM)
      CALL WAIT(200)
      RETURN
C     STOP
      END
```

```
      SUBROUTINE CON(IX,PSI,I,J)
      INTEGER PSI(100)
      COMMON IJ,IK,IIJ,IDELX
C     CALL READ(IE,2225,IX325)
C     CALL READ(IE,2225,IX325)
C     CALL WAIT(70)
C     CALL READ(IE,2210,IX310)
C     CALL READ(IE,2210,IX310)
C     CALL WAIT(70)
C     CALL WAIT(70)
      I=1
5     CALL READSI(IX,PSI,I,J,IB)
      IF(I.LE.1)GO TO 50
      IF(PSI(I).EQ.PSI(J))GO TO 900
50    IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 999
      IF(I.GT.1)GO TO 15
      IF(PSI(I).GT.100)GO TO 20
      GO TO 100
      IF(I.EQ.1)GO TO 20
15    IF(PSI(I).LT.0)GO TO 999
      IF(PSI(I).LT.PSI(J))GO TO 20
      GO TO 100
20    IX=IX+IDELX
21    I=I+1
      J=I-1
      CALL READSI(IX,PSI,I,J,IB)
      IF(I.LE.1)GO TO 51
      IF(PSI(I).EQ.PSI(J))GO TO 900
51    IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 999
      IF(PSI(I).LE.-100)GO TO 999
      IF(PSI(I).GT.PSI(J))GO TO 25
      GO TO 20
25    IX=IX-IDELX
      I=1
30    CALL READSI(IX,PSI,I,J,IB)
      IF(I.LE.1)GO TO 52
      IF(PSI(I).EQ.PSI(J))GO TO 900
52    IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 999
31    IX=IX-IDELX
      I=I+1
      J=I-1
      CALL READSI(IX,PSI,I,J,IB)
      IF(I.LE.1)GO TO 53
      IF(PSI(I).EQ.PSI(J))GO TO 900
53    IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 999
      IF(PSI(I).LE.-100)GO TO 999
      IF(PSI(I).GT.PSI(J))GO TO 20
      GO TO 31
100   IX=IX-IDELX
      I=I+1
      J=I-1
      CALL READSI(IX,PSI,I,J,IB)
      IF(I.LE.1)GO TO 54
      IF(PSI(I).EQ.PSI(J))GO TO 900
54    IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 999
      IF(PSI(I).GE.100)GO TO 999
      IF(PSI(I).GT.PSI(J))GO TO 100
```

```
      IX=IX+IDELX
      I=1
110   CALL READSI(IX.PSI,I,J,IB)
      IF(I.LE.1)GO TO 55
      IF(PSI(I).EQ.PSI(J))GO TO 900
55    IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 999
111   IX=IX+IDELX
      I=I+1
      J=I-1
      CALL READSI(IX,PSI,I,J,IB)
      IF(I.LE.1)GO TO 56
      IF(PSI(I).EQ.PSI(J))GO TO 900
56    IF(PSI(I).GE.-100.AND.PSI(I).LE.100)GO TO 999
      IF(PSI(I).GE.100)GO TO 999
      IF(PSI(I).LT.PSI(J))GO TO 100
      GO TO 111
900   WRITE(4,901)
901   FORMAT(1X,2HFU)
999   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE READSI(IX,PSI,I,J,IB)
      INTEGER PSI(100)
      COMMON IJ,IK,IIJ,IDELX
      COMMON/DIM/B,IBR
      IB=9000
      CALL IC(IE)
1006  FORMAT(1X,3I10,2X,I10,2X,I10)
      CALL WAIT(70)
      CALL STIND(IE,IJ,IX)
C
      CALL STIND(IE,2237,IBR)
      CALL WAIT(10)
      CALL READ(IE,0200,IDZ)
      CALL WAIT(70)
C
500   CONTINUE
C
C     ANALOG CONTROL LOOP
C     USES ANALOG COMPARATOR, 331
      CALL OP(IE)
      CALL WAIT(1000)
115   CALL HOLD(IE)
      CALL WAIT(70)
      CALL READ(IE,IIJ,IPSI)
      CALL WAIT(70)
C
      CALL IC(IE)
      CALL WAIT(30)
      CALL STIND(IE,2237,0)
      PSI(I)=IPSI
      CALL WAIT(70)
      CALL READ(IE,IJ,IXP)
      CALL WAIT(70)
C     WRITE(4,1006)IJ,IX,IXP,IPSI,PSI(I)
      CALL WAIT(100)
      IDELX=10
      IF(IABS(PSI(I)).GE.4000)GO TO 10
      IF(IABS(PSI(I)).GE.2000)GO TO 9
      IF(IABS(PSI(I)).GE.1000)GO TO 8
      IF(IABS(PSI(I)).GE.500)GO TO 7
      IF(IABS(PSI(I)).GE.350)GO TO 6
      IDELX=IDELX
      GO TO 125
6     IDELX=2*IDELX
      GO TO 125
7     IDELX=3*IDELX
      GO TO 125
8     IDELX=6*IDELX
      GO TO 125
9     IDELX=9*IDELX
      GO TO 125
10    IDELX=14*IDELX
125   RETURN
      END
```

```
      SUBROUTINE PDE2
C     PROGRAM PDE**122673**
      DIMENSION IPT(20),IPTV(20),ADEL(18)
      COMMON/P1/NA,JK,KZ
      COMMON/POTV/P(20),DEL,A
      COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
      COMMON/DIM/B,IBR
      DATA IPT/2224,2223,2232,2235,2245,2236,2230,2226,2227,2233,
     1 2242,2243,2244,2247,2253,2256,2264,2265,2266,2276/
600   FORMAT(F5.2,5F5.4)
405   CONTINUE
750   CONTINUE
      CALL INITA(IE,0)
      CALL CONSO(IE,0)
      CALL LEX(IE,0)
      CALL IC(IE)
5     CONTINUE
      CALL STIND(IE,2277,10000)
      CALL STIND(IE,2275,0)
C     SET TIME BASE
      CALL STIND(IE,2273,1000)
      ITM(1)=0
      CALL WAIT(70)
C     CALL STIND(IE,2275,10000)
      CALL WAIT(70)
      K=1
      MR=1
300   CONTINUE
      DO 3000 K=1,12
      Y=B*FLOAT(K)/11
      IYAS=IFIX(10000.*Y)
      CALL WAIT(70)
      CALL HOLD(IE)
      CALL STIND(IE,2237,IYAS)
      CALL WAIT(70)
      CALL READ(IE,0200,IVDUM)
      CALL WAIT(100)
      CALL READ(IE,0241,ISTA(3,K))
      CALL WAIT(70)
      CALL READ(IE,0221,ISTA(2,K))
      CALL WAIT(70)
      CALL READ(IE,0201,ISTA(1,K))
      CALL WAIT(70)
      CALL READ(IE,0261,ISTA(JK,K))
      CALL WAIT(70)
      CALL READ(IE,0271,ITM(K))
      CALL WAIT(70)
C     IF(ITM(K).GE IBR)GO TO 102
      J1=ISTA(1,K)
      J2=ISTA(2,K)
      J3=ISTA(3,K)
      J4=ISTA(4,K)
      IX=ITM(K)
      CALL WAIT(100)
400   CONTINUE
      IF(K.EQ.12)GO TO 102
      CALL OP(IE)
```

```
      CALL WAIT(1000)
      CALL HOLD(IE)
C     GO TO 400
C     IF(MR.EQ.1)K=0
      MR=MR+1
C     IF(K.GE.12)GO TO 102
C     K=K+1
C     GO TO 300
3000  CONTINUE
102   CONTINUE
      CALL WAIT(100)
      KZ=K
      CALL IC(IE)
      CALL WAIT(200)
200   FORMAT(1X,5(1X,I7))
      CALL IC(IE)
      CALL WAIT(1000)
      CALL STIND(IE,2275,0)
      CALL WAIT(100)
      I=2225
      DO 2001 NI=1,4
      GO TO (231,227,228,229),NI
229   I=2274
      GO TO 231
228   I=2246
      GO TO 231
227   I=2234
231   CALL WAIT(100)
      CALL READ(IE,I,IV(NI))
      CALL READ(IE,I,IV(NI))
      CALL WAIT(70)
2001  CONTINUE
      CALL WAIT(70)
      WRITE(4,2005)IV(1),IV(2),IV(3),IV(4)
2005  FORMAT(1X,21HPOTS 225,234,246,274:,4(1X,I7))
      CALL WAIT(70)
500   CONTINUE
2020  FORMAT(1X,9(1X,I7))
2021  FORMAT(1X,T4,'ITM',T14,'X1',T22,'X2',T30,'X3',T38,
     1X4',T46,'X5',T54,'X6',T62,'X7',T70,'X8')
      RETURN
      END
```

```
      SUBROUTINE DISK
      DIMENSION TST(2)
      COMMON/W/Y(18),IPSI
      COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
      COMMON/POTV/P(20),DEL,A
      COMMON/DIM/B,IBR
      COMMON/P1/NA,JK,KZ
      COMMON/GRD/NPSIG,NTF
      DATA TST(1),TST(2)/3HTST,4H SRC/
      CALL ENTER(7,TST)
      NTF3=NTF+3
      DO 500 M=1,NTF3
      DO 500 NZ=1,KZ
      WRITE(7,2020)ITM(NZ),ISTA(M,NZ)
500   CONTINUE
2020  FORMAT(1X,9(1X,I7))
      CALL CLOSE(7)
      RETURN
      END
```

```
      SUBROUTINE DRW
      COMMON/W/Y(18),IPSI
      COMMON/DR/Z(38),ZY(38)
      COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
      COMMON/POTV/P(20),DEL,A
      COMMON/DIM/B,IBR
      COMMON/GRD/NPSIG,NTF
      NTF3=NTF+3
      DO 99 N=1,NTF3
C     READ(4,98)XLOC(N),Y(N)
99    CONTINUE
98    FORMAT(2F10.5)
      CALL INITT(0)
      CALL ERASE
      CALL MOVABS(100,100)
      CALL DRWABS(100,700)
      CALL DRWABS(1000,700)
      CALL DRWABS(1000,100)
      CALL DRWABS(100,100)
      NLY=48
      LY=100
      NLYT=IFIX(10.*B)+48
      DO 251 N=1,10
      CALL MOVABS(100,LY)
      CALL DRWABS(90,LY)
      CALL MOVABS(50,LY)
      CALL ANCHO(48)
      CALL ANCHO(46)
      CALL ANCHO(NLY)
      NLY=NLY+1
      IF(NLY.GT.NLYT)GO TO 261
      LY=(600/(NLYT-48))+LY
251   CONTINUE
261   CONTINUE
      DO 200 MT=1,NTF3
      XI=A-XLOC(MT)
      KL=IFIX(XLOC(MT)*(900./A))+100
      KLI=IFIX(XI*(900./A))+100
      CALL MOVABS(KL,90)
      CALL DRWABS(KL,700)
      CALL MOVABS(KLI,700)
      CALL DRWABS(KLI,90)
      CALL MOVABS(KL-10,80)
      XLOCX=XLOC(MT)
      ID1=IFIX(XLOCX*10.)
      ID2=IFIX(XLOCX*100.)-(10*ID1)
      XLOCX=XLOCX+A/8.
      IXC2=48
      IXC1=48
      DO 252 NR=1,9
      IF(ID1.EQ.NR)IXC1=IXC1+NR
      IF(ID2.EQ.NR)IXC2=IXC2+NR
252   CONTINUE
      CALL ANCHO(46)
      CALL ANCHO(IXC1)
      CALL ANCHO(IXC2)
```

```
253     CONTINUE
        CALL MOVABS(20,700)
        CALL ANCHO(66)
        CALL MOVABS(1000,30)
        CALL ANCHO(65)
200     CONTINUE
        CALL MOVABS(50,50)
        CALL HOME
        CALL ANMODE
        RETURN
        END
```

```
      SUBROUTINE DRWA
      COMMON/W/Y(18),IPSI
      COMMON/DR/Z(36),ZY(36)
      COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
      COMMON/POTV/P(20),DEL,A
      COMMON/DIM/B,IBR
      COMMON/GRD/NPSIG,NTF
      IF(NPSIG.NE.1)GO TO 1000
      CALL MOVABS(100,100)
      CALL DRWABS(100,700)
      CALL DRWABS(1000,700)
      CALL DRWABS(1000,100)
      CALL DRWABS(100,100)
1000  CONTINUE
      NTF3=NTF+3
      DO 100 N=1,NTF3
      Z(N)=XLOC(N)
      Z(N+NTF3)=A-XLOC(N)
      ZY(N)=Y(N)
      ZY(N+NTF3)=Y(N)
100   CONTINUE
      ITOT=1
300   CONTINUE
      IF(ITOT.GT.2000)GO TO 400
      NTFR=(2*NTF3)-1
      DO 220 N=1,NTFR
      IF(Z(N+1).LT.Z(N))GO TO 598
      GO TO 220
598   ZV1=Z(N+1)
      ZV2=Z(N)
      ZY1=ZY(N+1)
      ZY2=ZY(N)
      Z(N)=ZV1
      Z(N+1)=ZV2
      ZY(N)=ZY1
      ZY(N+1)=ZY2
      N=1
      ITOT=ITOT+1
      GO TO 300
220   CONTINUE
400   CONTINUE
301   FORMAT(1X,T5,'X',T15,'Y')
302   FORMAT(1X,2F10.5)
      CALL HOME
      PSI=FLOAT(IPSI)/100.
      X=0.
      DO 498 NP=1,1000
      TPSI=100.*SIN((3.14159*X)/A)
      IF(TPSI.GE.PSI)GO TO 497
      X=X+.005
498   CONTINUE
497   IZX=IFIX(X*(900./A))+100
      XEND=A-X
      CALL MOVABS(IZX,100)
      NTFRA=NTFR+1
      DO 411 NQ=1,NTFRA
      IF(NQ.EQ.1)GO TO 473
```

```
      ZDEL=Z(NQ)-Z(NQ-1)
      IF(ZDEL.LT.(.001))GO TO 411
473   CONTINUE
      KLX=IFIX(Z(NQ)*(900./A))+100
      IF(KLX.LT.IZX)GO TO 411
      IZXE=IFIX(XEND*(900./A))+100
      IF(KLX.GT.IZXE)GO TO 411
      KLY=IFIX(ZY(NQ)*(600./B))+100
C     IF(NQ.EQ.9)GO TO 413
465   CONTINUE
      CALL DRWABS(KLX,KLY)
      GO TO 411
413   CONTINUE
      ID1=IPSI/1000
      ID2=(IPSI-(ID1*1000))/100
      ICX2=48
      ICX1=48
      DO 414 N=1,9
      IF(ID1.EQ.N)ICX1=ICX1+N
      IF(ID2.EQ.N)ICX2=ICX2+N
414   CONTINUE
C     CALL ANCHO(ICX1)
C     CALL ANCHO(ICX2)
C     CALL MOVREL(-20,0)
      GO TO 465
411   CONTINUE
      IZXA=IFIX(XEND*(900./A))+100
      CALL DRWABS(IZXA,100)
      CALL ANMODE
C     STOP
      RETURN
      END
```

C-3. **Exact Solution Listings.** The following listings are used for the exact solution, which is run using "E___IDEA" since the exact solution also required chaining.

$E IDEA

13 Aug

POT on SCR

RK"A"

```
C       SUBROUTINE POT
C       THIS WILL ACT AS THE MAIN PROGRAM
        DIMENSION TST(2)
        COMMON/W/Y(18),IPSI
        COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
        COMMON/POTV/P(20),DEL,A
        COMMON/DIM/B,IBR
        COMMON/P1/NA,JK,KZ
        COMMON/GRD/NPSIG,NTF
        DATA TST(1),TST(2)/3HTST,4H SRC/
        JK=4
        NA=1
        WRITE(4,601)
        READ(4,600)A
        READ(4,600)B
        IBR=IFIX(10000.*B)
        WRITE(4,2051)
2051    FORMAT(1X,25HSPECIFY NO OF LINES LT 16)
        READ(4,6004)NLINES
6004    FORMAT(I2)
        DELTX=.5/(FLOAT(NLINES))
        DELTA(1)=DELTX
        NTF=NLINES-1
        DO 2050 NT=2,NTF
        DELTA(NT)=DELTA(NT-1)+DELTX
2050    CONTINUE
        DELTA(1)=1./18.
        DELTA(2)=2./18.
        DELTA(14)=8.6/18.
        DELTA(3)=4./18.
        DELTA(4)=5./18.
        DELTA(15)=8.7/18.
        DELTA(5)=7./18.
        DELTA(6)=7.3/18.
        DELTA(7)=7.6/18.
        DELTA(8)=8./18.
        DELTA(9)=8.1/18.
        DELTA(10)=8.2/18.
        DELTA(11)=8.3/18.
        DELTA(12)=8.4/18.
        DELTA(13)=8.5/18.
601     FORMAT(1X,'INPUT A,B,DEL1,DEL2,DEL3,DEL4,DEL5 :F5.2,5F5.4')
900     CONTINUE
697     FORMAT(1X,23HINPUT: A,DEL: F5.2,F5.4)
        WRITE(4,11)DELTA(NA)
        DEL=DELTA(NA)
11      FORMAT(1X,'DEL=',F10.4)
        CR=2.*DEL
        C4=CR*A*(3./6.)
        DX11=A/6.
        DX21=A/6.
        DX31=(A+(6.*C4))/12.
```

```
      DX41=(C4/2.)+((3.*A/6.)-C4)/2.
      DX12=A/6.
      DX22=A/6.
      DX32=A/6.
      DX42=C4
      DX52=(A/2.)-C4
      C1=DX32/DX42
      C2=DX32/DX52
602   FORMAT(1X,6(1X,F10.4))
      P(1)=1.
      P(2)=1.
      P(3)=.01/(DX11*DX12)
      P(4)=SIN(3.14159/6.)
      P(5)=.01/(DX11*DX22)
      P(6)=1.
      P(7)=SIN(3.14159*2./6.)
      P(8)=1.
      P(9)=.01/(DX22*DX21)
      P(10)=.01/(DX32*DX21)
      P(11)=SIN(3.14159*3./6.)
      P(12)=.0I/(DX32*DX31)
      P(13)=1.
      P(14)=DX32/DX42
      P(15)=.01/(DX31*DX42*C1)
      P(16)=DX32/DX52
      P(17)=SIN(3.14159*(1.+CR)/2.)
      P(18)=.01/(DX41*DX42*C1)
      P(19)=DX32/DX42
      P(20)=.01/(DX41*DX52*C2)
      IF(P(14).LT.1.5)GO TO 698
      PTOT1=P(14)*P(15)
      P(14)=1
      P(15)=PTOT1
      PTOT2=P(19)*P(18)
      P(19)=1.
      P(18)=PTOT2
698   CONTINUE
      IF(P(16).LT.1.5)GO TO 699
      PTOT3=P(16)*P(20)
      P(16)=1.
      P(20)=PTOT3
699   CONTINUE
      DO 700 NP=1,20
C     WRITE(4,6010)P(NP)
700   CONTINUE
600   FORMAT(F5.2,F5.4)
6010  FORMAT(1X,F10.4)
      CALL EXACT
      NA=NA+1
      JK=JK+1
      IF(NA.GT.NTF)GO TO 3000
      GO TO 900
3000  CONTINUE
      NTF3=NTF+3
      DO 500 M=1,NTF3
      DO 500 NZ=1,KZ
      WRITE(7,2020)ITM(NZ),ISTA(M,NZ)
```

```
 500    CONTINUE
 2020   FORMAT(1X,9(1X,I7))
2021    FORMAT(1X,T4,'ITM',T14,'X1',T22,'X2',T30,'X3',T38,
     1 'X4',T46,'X5',T54,'X6',T62,'X7',T70,'X8')
C       SEARCH FOR SPECIFIED PSI FOR EQUIPOT PRINTOUT
4003    CONTINUE
        WRITE(4,2006)
2006    FORMAT(1X,11HSPECIFY PSI)
        READ(4,1009)PSI
        READ(4,1009)PSID
        READ(4,1021)NPSI
1021    FORMAT(I2)
        NPSIG=0
4002    CONTINUE
        DO 4000 IZR=1,NPSI
        IPSI=IFIX(PSI*100.)
2022    FORMAT(1X,T9,'PSI',T20,'Y',T31,'X',T42,'XI')
        K=1
        NTF3=NTF+3
        DO 1000 N=1,NTF3
        DO 1001 NA=1,KZ
        IF(ISTA(N,NA).LT.IPSI)GO TO 1002
1001    CONTINUE
1002    NB=NA
        NC=NB-1
        DELSTA=FLOAT(ISTA(N,NC)-ISTA(N,NB))
        DELTM=FLOAT(ITM(NC)-ITM(NB))
        Y(K)=(FLOAT(ITM(NC))-(DELTM*((FLOAT(ISTA(N,NC)-IPSI)/DELSTA))))/
     $ 10000.
        X=XLOC(N)
        XI=A-XLOC(N)
        K=K+1
1000    CONTINUE
1009    FORMAT(2F10.3,I2)
1010    FORMAT(1X,4(1X,F10.3))
        IF(IZR.GT.1)GO TO 4001
        IF(NPSIG.GT.0)GO TO 4001
        CALL DRW
4001    CONTINUE
        CALL DRWA
        PSI=PSI+PSID
4000    CONTINUE
        PSI=PSI-(FLOAT(NPSI)*PSID)
        NPSIG=1
        WRITE(4,1011)
        WRITE(4,1012)
1011    FORMAT(1X,18HNO GRID OR RESTART)
1012    FORMAT(1X,18HTYPE 2 FOR RESTART)
        READ(4,1021)MST
        IF(MST.EQ.2)GO TO 4003
        GO TO 4002
        STOP
        END
```

EXACT on SCR

```
      SUBROUTINE EXACT
      COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),
     1 DELTA(15)
      COMMON/POTV/P(20),DEL,A
      COMMON/DIM/B,IBR
      COMMON/P1/NA,JK,KZ
      PI=3.1415926
      XLOC(1)=A/6.
      XLOC(2)=2.*A/6.
      XLOC(3)=3.*A/6.
      DO 5 I=4,18
      XLOC(I)=XLOC(3)+DELTA(I-3)
5     CONTINUE
      KZ=12
      DO 10 IX=1,18
      DO 20 IYA=1,12
      Y=B*FLOAT(IYA-1)/11.
      ITM(IYA)=IFIX(Y*10000.)
      Q1=PI*XLOC(IX)/A
      Q2=PI*B/A
      Q3=PI*(B-Y)/A
      PSI=100.*SIN(Q1)*(EXP(Q3)-EXP(-Q3))/(EXP(Q2)-EXP(-Q2))
      ISTA(IX,IYA)=IFIX(PSI*100.)
20    CONTINUE
10    CONTINUE
      RETURN
      END
```

DRW on SCR

```
      SUBROUTINE DRW
      COMMON/W/Y(18),IPSI
      COMMON/DR/Z(38),ZY(38)
      COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
      COMMON/POTV/P(20),DEL,A
      COMMON/DIM/B,IBR
      COMMON/GRD/NPSIG,NTF
      NTF3=NTF+3
      DO 99 N=1,NTF3
C     READ(4,98)XLOC(N),Y(N)
99    CONTINUE
98    FORMAT(2F10.5)
      CALL INITT(0)
      CALL ERASE
      CALL MOVABS(100,100)
      CALL DRWABS(100,700)
      CALL DRWABS(1000,700)
      CALL DRWABS(1000,100)
      CALL DRWABS(100,100)
      NLY=48
      LY=100
      NLYT=IFIX(10.*B)+48
      DO 251 N=1,10
      CALL MOVABS(100,LY)
      CALL DRWABS(90,LY)
      CALL MOVABS(50,LY)
      CALL ANCHO(48)
      CALL ANCHO(46)
      CALL ANCHO(NLY)
      NLY=NLY+1
      IF(NLY.GT.NLYT)GO TO 261
      LY=(600/(NLYT-48))+LY
251   CONTINUE
261   CONTINUE
      DO 200 MT=1,NTF3
      XI=A-XLOC(MT)
      KL=IFIX(XLOC(MT)*(900./A))+100
      KLI=IFIX(XI*(900./A))+100
      CALL MOVABS(KL,90)
      CALL DRWABS(KL,700)
      CALL MOVABS(KLI,700)
      CALL DRWABS(KLI,90)
      CALL MOVABS(KL-10,80)
      XLOCX=XLOC(MT)
      ID1=IFIX(XLOCX*10.)
      ID2=IFIX(XLOCX*100.)-(10*ID1)
      XLOCX=XLOCX+A/8.
      IXC2=48
      IXC1=48
      DO 252 NR=1,9
      IF(ID1.EQ.NR)IXC1=IXC1+NR
      IF(ID2.EQ.NR)IXC2=IXC2+NR
252   CONTINUE
      CALL ANCHO(46)
      CALL ANCHO(IXC1)
      CALL ANCHO(IXC2)
```

```
253     CONTINUE
        CALL MOVABS(20,700)
        CALL ANCHO(66)
        CALL MOVABS(1000,30)
        CALL ANCHO(65)
200     CONTINUE
        CALL MOVABS(50,50)
        CALL HOME
        CALL ANMODE
        RETURN
        END
```

DRWA on SCR

```
      SUBROUTINE DRWA
      COMMON/W/Y(18),IPSI
      COMMON/DR/Z(36),ZY(36)
      COMMON/DRWP/XLOC(18),IY(18),ITM(12),ISTA(20,12),IV(4),DELTA(15)
      COMMON/POTV/P(20),DEL,A
      COMMON/DIM/B,IBR
      COMMON/GRD/NPSIG,NTF
      IF(NPSIG.NE.1)GO TO 1000
      CALL MOVABS(100,100)
      CALL DRWABS(100,700)
      CALL DRWABS(1000,700)
      CALL DRWABS(1000,100)
      CALL DRWABS(100,100)
1000  CONTINUE
      NTF3=NTF+3
      DO 100 N=1,NTF3
      Z(N)=XLOC(N)
      Z(N+NTF3)=A-XLOC(N)
      ZY(N)=Y(N)
      ZY(N+NTF3)=Y(N)
100   CONTINUE
      ITOT=1
300   CONTINUE
      IF(ITOT.GT.2000)GO TO 400
      NTFR=(2*NTF3)-1
      DO 220 N=1,NTFR
      IF(Z(N+1).LT.Z(N))GO TO 598
      GO TO 220
598   ZV1=Z(N+1)
      ZV2=Z(N)
      ZY1=ZY(N+1)
      ZY2=ZY(N)
      Z(N)=ZV1
      Z(N+1)=ZV2
      ZY(N)=ZY1
      ZY(N+1)=ZY2
      N=1
      ITOT=ITOT+1
      GO TO 300
220   CONTINUE
400   CONTINUE
301   FORMAT(1X,T5,'X',T15,'Y')
302   FORMAT(1X,2F10.5)
      CALL HOME
      PSI=FLOAT(IPSI)/100.
      X=0.
      DO 498 NP=1,1000
      TPSI=100.*SIN((3.14159*X)/A)
      IF(TPSI.GE.PSI)GO TO 497
      X=X+.005
498   CONTINUE
497   IZX=IFIX(X*(900./A))+100
      XEND=A-X
      CALL MOVABS(IZX,100)
      NTFRA=NTFR+1
```

```
      DO 411 NQ=1,NTFRA
      IF(NQ.EQ.1)GO TO 473
      ZDEL=Z(NQ)-Z(NQ-1)
      IF(ZDEL.LT.(.001))GO TO 411
473   CONTINUE
      KLX=IFIX(Z(NQ)*(900./A))+100
      IF(KLX.LT.IZX)GO TO 411
      IZXE=IFIX(XEND*(900./A))+100
      IF(KLX.GT.IZXE)GO TO 411
      KLY=IFIX(ZY(NQ)*(600./B))+100
C     IF(NQ.EQ.9)GO TO 413
465   CONTINUE
      CALL DRWABS(KLX,KLY)
      GO TO 411
413   CONTINUE
      ID1=IPSI/1000
      ID2=(IPSI-(ID1*1000))/100
      ICX2=48
      ICX1=48
      DO 414 N=1,9
      IF(ID1.EQ.N)ICX1=ICX1+N
      IF(ID2.EQ.N)ICX2=ICX2+N
414   CONTINUE
C     CALL ANCHO(ICX1)
C     CALL ANCHO(ICX2)
C     CALL MOVREL(-20,0)
      GO TO 465
411   CONTINUE
      IZXA=IFIX(XEND*(900./A))+100
      CALL DRWABS(IZXA,100)
      CALL ANMODE
C     STOP
      RETURN
      END
```

C-4. Stored Data for $\psi(x, y)$. The $\psi(y)$-data taken for each x-station during the exact and hybrid solutions are provided as comparison data between solutions.

HYBRID

Hybrid 1, 2, 10 data

3	4993
126	4652
364	3990
545	3504
728	3032
911	2567
1091	2112
1273	1665
1455	1225
1636	787
1819	356
2000	-71
3	8650
126	8048
364	6894
545	6043
728	5210
911	4390
1091	3584
1273	2788
1455	1999
1636	1208
1819	418
2000	-373
3	9987
126	9312
364	8014
545	7055
728	6116
911	5196
1091	4293
1273	3405
1455	2526
1636	1652
1819	788
2000	-72
3	9838
126	9192
364	7849
545	6878
728	5924
911	4985
1091	4057
1273	3137
1455	2222
1636	1305
1819	383
2000	-547
3	9386
126	8781
364	7528
545	6624
728	5736

911	4863
1091	4002
1273	3151
1455	2306
1636	1466
1819	628
2000	-211
3	7648
126	7072
364	6118
545	5372
728	4642
911	3925
1091	3222
1273	2528
1455	1838
1636	1153
1819	473
2000	-209
3	6415
126	5940
364	5122
545	4495
728	3882
911	3280
1091	2690
1273	2110
1455	1533
1636	962
1819	395
2000	-175
3	3414
126	3160
364	2684
545	2338
728	2004
911	1679
1091	1364
1273	1055
1455	750
1636	448
1819	143
2000	-156
3	2920
126	2734
364	2282
545	1984
728	1695
911	1415
1091	1144
1273	878
1455	614
1636	354
1819	96
2000	-162
3	2415
126	2223

364	1874
545	1621
728	1380
911	1148
1091	922
1273	702
1455	484
1636	269
1819	52
2000	-165
3	1731
126	1588
364	1326
545	1140
728	963
911	796
1091	634
1273	475
1455	320
1636	164
1819	9
2000	-149
3	1562
126	1430
364	1189
545	1022
728	860
911	705
1091	559
1273	416
1455	271
1636	134
1819	-7
2000	-152

EXACT

0	5000
181	4495
363	4004
545	3527
727	3061
909	2606
1090	2158
1272	1718
1454	1284
1636	853
1818	426
2000	0
0	8660
181	7785
363	6936
545	6109
727	5303
909	4513
1090	3739
1272	2976
1454	2224
1636	1478
1818	738
2000	0
0	10000
181	8990
363	8009
545	7055
727	6123
909	5212
1090	4317
1272	3437
1454	2568
1636	1707
1818	852
2000	0
0	9848
181	8853
363	7887
545	6947
727	6030
909	5132
1090	4252
1272	3385
1454	2529
1636	1681
1818	839
2000	0
0	9396
181	8447
363	7526
545	6629
727	5754
909	4897
1090	4057
1272	3230

1,2. EXACT 10 delta 8/26/74

1454	2413
1636	1604
1818	800
2000	0
0	7660
181	6886
363	6135
545	5404
727	4690
909	3992
1090	3307
1272	2633
1454	1967
1636	1308
1818	652
2000	0
0	6427
181	5778
363	5148
545	4534
727	3936
909	3350
1090	2775
1272	2209
1454	1650
1636	1097
1818	547
2000	0
0	3420
181	3074
363	2739
545	2412
727	2094
909	1782
1090	1476
1272	1175
1454	878
1636	584
1818	291
2000	0
0	2923
181	2628
363	2341
545	2062
727	1790
909	1523
1090	1262
1272	1005
1454	750
1636	499
1818	249
2000	0
0	2419
181	2174
363	1937
545	1706
727	1481
909	1260

1090	1044
1272	831
1454	621
1636	413
1818	206
2000	0
0	1736

C-5. **Program Control Flow Charts.** The hybrid program is shown in Figure C-1, while the exact program is depicted in Figure C-2.

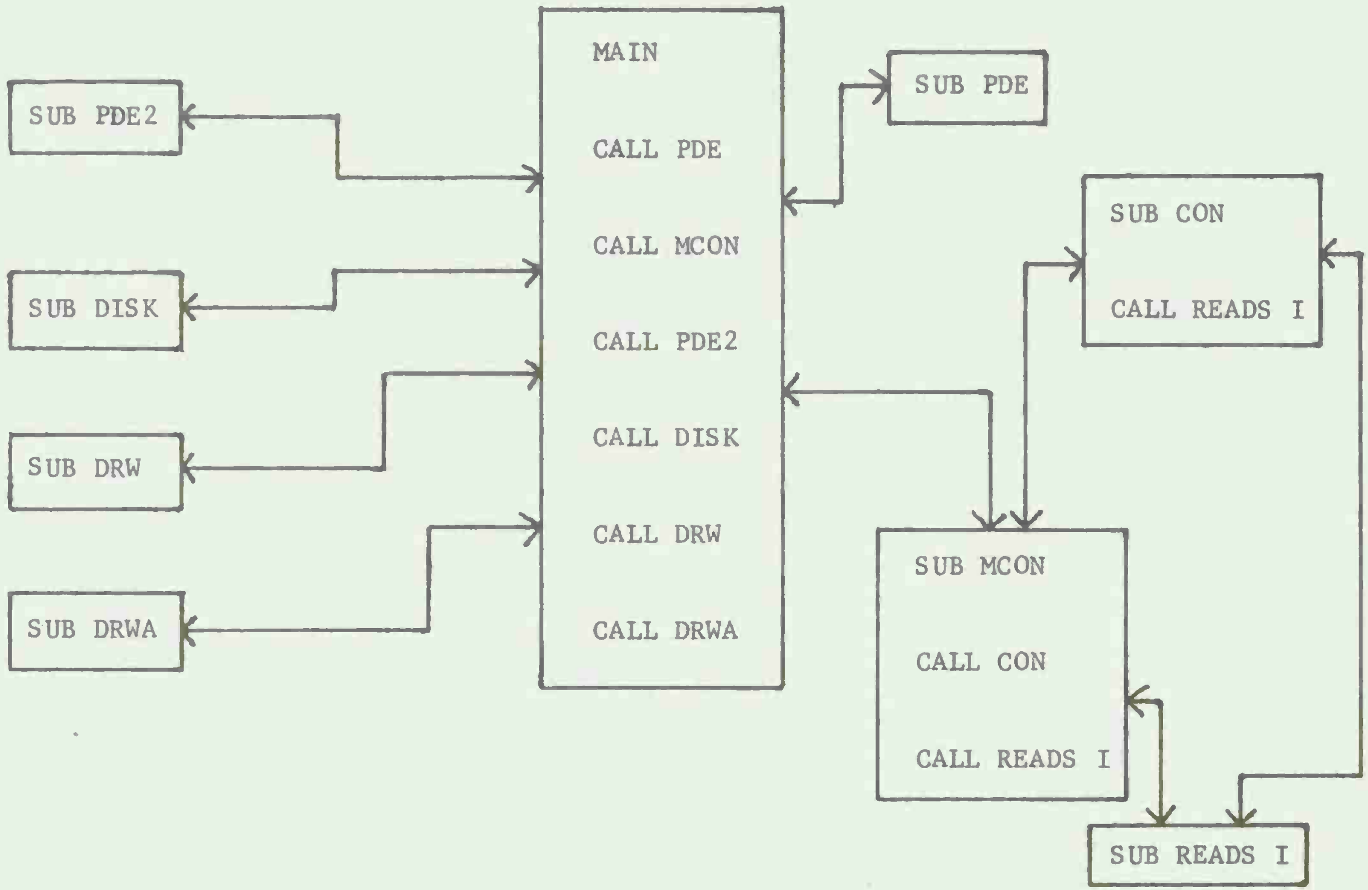


Figure C-1. Block diagram – hybrid program.

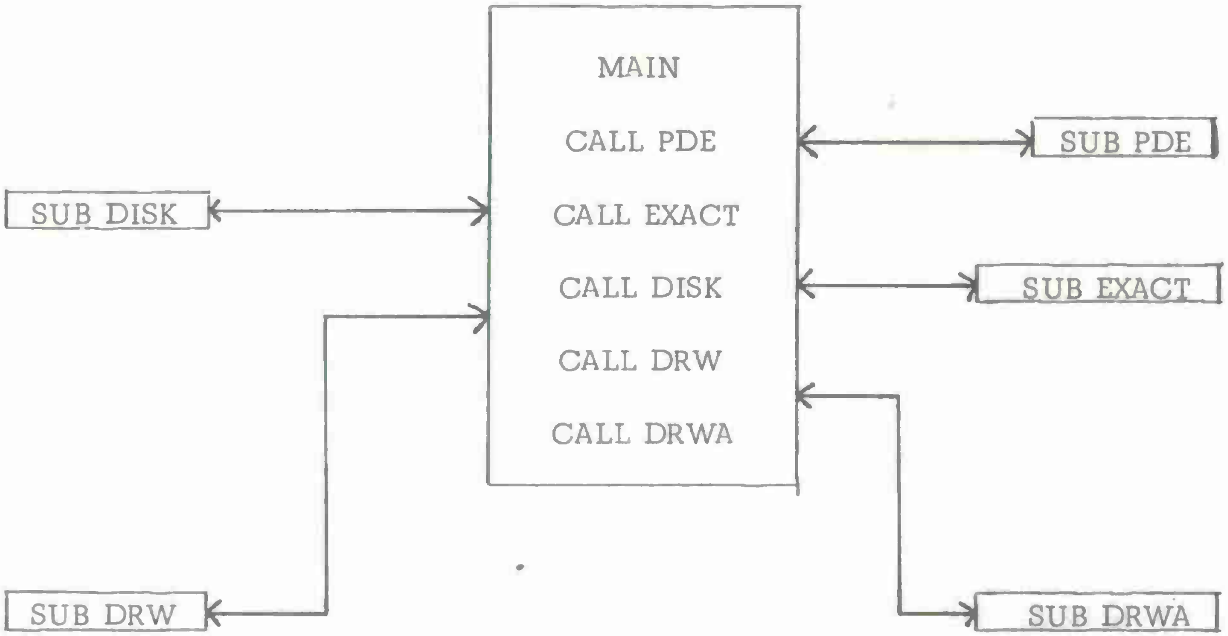


Figure C-2. Block diagram – exact program.

C-6. Chaining Routine. The chaining routine is as follows:

```
 ↑C

SLOGIN RNM

SAVE RK -5

$BANK ON

$K ON

$CHAIN

CHAIN V9A

NAME XCT FILE
>PDR2I
LIST OPTIONS & PARAMETERS
>DKR,16K,SZ
DEFINE RESIDENT CODE
>POT,#INTRV,#INMUX,#ADDR,#RUN,#CONSO,#INITA
DESCRIBE LINKS & STRUCTURE
>L1=PDE
>L2=MCON/CON,READSI
>L3=PDE2
>L4=DISK
>L5=DRW
>L6=DRWA
>L1:L2
>L2:L3
>L3:L4
>L4:L5
>L5:L6
>
LINK TABLE
         37533-37636 00104

RESIDENT CODE
POT      34617-37532 02714
CONSO    34466-34616 00131
INMUX    34427-34465 00037
WAIT     34356-34426 00031
SET      33745-34355 00411
IC       33377-33744 00346
KYSPD    33125-33376 00272
ISTAT    32764-33124 00121
ADDR     32677-32763 00065
IOPKG    32521-32676 00156
GENADD   32277-32520 00222
SETSIZ   32252-32276 00025
INITA    32211-32251 00041
FLOAT    32200-32210 00011
IFIX     31765-31777 00013
SIN      31752-31764 00013
.EE      31650-31751 00102
.EC      31604-31647 00044
```

.DA	31526-31623	00076
ECPIO	25546-31525	03760
STOP	25533-25545	00013
CPUSG	25414-25532	00117
.FLTI	25126-25413	00266
FICPS	24171-25125	00735
RELEAS	23060-24170	01111
OTSER	22650-23057	00210
.CE	22626-22647	00022
V	22561-22625	00045
DRWP	22035-22560	00524
PCTV	21761-22034	00054
MIN	21756-21760	00003
PI	21753-21755	00003
CRD	21751-21752	00002

LINK -- L1

PDE	21163-21752	00570
READ	20570-21162	00373

LINK -- L2

MCOM	21070-21750	00661
COM	16565-17777	01213
READSI	20500-21067	00370
READ	20105-20477	00373
IABS	20071-20104	00014
INTRAE	16431-16564	00134

LINK -- L3

PDE2	20563-21750	01166
READ	20170-20562	00373
GOTO	20142-20167	00026

LINK -- L4

DISK	21550-21750	00201
FILE	21155-21547	00373
.SS	21066-21154	00067
INTRAE	20732-21065	00134

LINK -- L5

DRW	21117-21750	00632
DRWABS	20774-21116	00123
INITT	20545-20773	00227
ERASE	20334-20544	00211
ANCHO	20216-20333	00116
NEWLIN	20207-20215	00007
CARTN	20070-20206	00117
LINEF	17676-17777	00102
HOME	17611-17675	00065
NEWPAG	17463-17610	00126
RESTAT	17225-17462	00236
ANMODE	17126-17224	00077
MOVABS	17031-17125	00075
IOWAIT	16723-17030	00106
VECMOD	16553-16722	00150
SVSTAT	16366-16552	00165
PNTMOD	16253-16365	00113

```
XYCNVT 15641-16252 00412
TINPUT 20027-20067 00041
MOD    15615-15640 00024
GOTO   15567-15614 00026
INTEAE 15433-15566 00134
DP     15233-15432 00200
TYTRNX 15131-15232 00102

LINK -- L6
DRAW   20525-21750 01224
DEVAPS 20402-20524 00123
MODE   20315-20401 00065
ANMODE 20216-20314 00077
MOVAPS 20121-20215 00075
VECMOD 17650-17777 00130
XYCNVT 17236-17647 00412
TINPUT 20060-20120 00041
MOD    20034-20057 00024
INTEAE 17102-17235 00134
DP     16662-17101 00220
TYTRNX 16560-16661 00102

BLANK COMMON
.XY    15075-15130 00034

CORE REQ'D
       15075-37636 22542

DOS-15 V3A
$
```

DISTRIBUTION FOR USAMERDC REPORT 2135

No. Copies	Addressee
	Department of Defense
1	Director, Technical Information Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington, VA 22209
1	Director Defense Nuclear Agency ATTN: STTL Washington, DC 20305
12	Defense Documentation Center Cameron Station Alexandria, VA 22314
	Department of the Army
6	Commander US Army Materiel Command ATTN: AMCRD-WB AMCRD-T AMCRD-J AMCRD-O AMCRD-G AMCRD-FP 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-E Col. C. W. Barker 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-ES Mr. P. O. Langguth 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-E Mr. W. Reuth 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Troop Support Command ATTN: AMSTS-K 4300 Goodfellow Blvd St. Louis, MO 63120

No. Copies	Addressee
1	Commander HQ TRADOC ATTN: ATEN-ME Fort Monroe, VA 23651
1	HQDA (DAMA-AOA-M) Washington, DC 20310
1	HQDA (DAEN-RDL) Washington, DC 20314
1	HQDA (DAEN-MCE-D) Washington, DC 20314
1	Commander US Army Missile Command ATTN: AMSMI-RR Redstone Arsenal, AL 35809
1	Commander Rock Island Arsenal ATTN: SARRI-LPL Rock Island, IL 61201
1	Director Army Materials and Mechanics Research Center ATTN: AMXMR-STL Technical Library Watertown, MA 02172
1	Commander US Army Natick Laboratories ATTN: Technical Library Natick, MA 01760
1	US Army Ballistic Research Labs Technical Library AMXBR-LB (Bldg 305) Aberdeen Proving Ground, MD 21005
1	Commander Edgewood Arsenal ATTN: SAREA-TS-L Aberdeen Proving Ground, MD 21010
1	Director US Army Materiel Systems Analysis Agency ATTN: AMXSY-CM Aberdeen Proving Ground, MD 21005

No. Copies	Addressee
1	Director US Army Engineer Waterways Experiment Station ATTN: Chief, Library Branch Technical Information Center Vicksburg, MS 39180
1	Commander Picatinny Arsenal ATTN: SARPA-TS-S No. 59 Dover, NJ 07801
1	Commander Frankford Arsenal ATTN: Library, K2400, B1 51-2 Philadelphia, PA 19137
1	Learning Resources Center US Army Engineer School Bldg 270 Fort Belvoir, VA 22060
1	President US Army Armor and Engineer Board ATTN: ATZK-AE-TD-E Fort Knox, KY 40121
1	Commandant US Army Command and General Staff College ATTN: ATSW-RI-L Fort Leavenworth, KS 66027
1	Commander US Army Electronics Command ATTN: AMSEL-GG-TD Fort Monmouth, NJ 07703
1	President US Army Aviation Test Board ATTN: STEBG-PO Fort Rucker, AL 36360
1	US Army Aviation School Library P.O. Drawer 0 Fort Rucker, AL 36360
1	HQ, 193d Infantry Brigade (CZ) Directorate of Facilities Engineering Fort Amador, Canal Zone
1	HQ, USAREUR & Seventh Army DCSENGR, ATTN: AEAEN-MO ATTN: Mil Ops Div APO New York 09403

No. Copies	Addressee
2	Engineer Representative US Army Standardization Group, UK Box 65, FPO New York 09510
1	Commander 2nd Engineer Group ATTN: S4 APO San Francisco 96301
1	Commander & Director USAFESA ATTN: FESA-RTD Fort Belvoir, VA 22060
	USAMERDC
1	Commander Technical Director Asst Tech Dir (R&D) Chief, Legal Office CIRCULATE
1	Chief, Lab 1000 Chief, Lab 2000 Chief, Lab 3000 Chief, Lab 4000 Chief, Lab 5000 Chief, Lab 6000 Chief, Lab 7000 Chief, Lab 8000 Chief, Tech & Res Spt Ofc CIRCULATE
1	Lab 3000
20	Electrical Equipment Div
2	Tech Reports Ofc
3	Security Ofc
2	Tech Documents Ctr
1	Operations & Programs Div, PMSO
1	Information Ofc
1	Legal Ofc
	Department of the Navy
1	Director, Physics Program (421) Office of Naval Research Arlington, VA 22217

No. Copies	Addressee
1	Director Naval Research Laboratory ATTN: Code 2627 Washington, DC 20375
1	Commander, Naval Facilities Engineering Command Department of the Navy ATTN: Code 032-A 200 Stovall St Alexandria, VA 22332
1	US Naval Oceanographic Office Library (Code 1600) Washington, DC 20373
1	Officer-in-Charge (Code L31) Civil Engineering Laboratory Naval Construction Battalion Center Port Hueneme, CA 93043
1	Director Earth Physics Program Code 463 Office of Naval Research Arlington, VA 22217
	Department of the Air Force
1	HQ USAF/RDPS (Mr. Allan Eaffy) Washington, DC 20330
1	Mr. William J. Engle Chief, Utilities Branch HQ USAF/PREEU Washington, DC 20332
1	AFAPL/SFL Wright-Patterson AFB, OH 45433
1	ASD/ENCSP Wright-Patterson AFB, OH 45433
	Others
1	Professor Raymond R. Fox Nuclear Defense Design Center School of Engineering and Applied Science The George Washington University Washington, DC 20052
1	Department of Transportation Library, FOB 10A, TAD-494.6 800 Independence Ave., SW Washington, DC 20591

DEPARTMENT OF THE ARMY
U. S. ARMY MOBILITY EQUIPMENT
RESEARCH AND DEVELOPMENT CENTER
FORT BELVOIR, VIRGINIA 22060

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
U. S. DEPARTMENT OF THE ARMY
DOD-314

U.S.MAIL

THIRD CLASS MAIL

Commander
Rock Island Arsenal
ATTN: SARRI-LPL
Rock Island, Il 61201