A Model of 2-Dimensional Freezing Front Movement Using the Complex Variable BE Method

T.V. Hromadka II[1] R.L. Berg[2]

(1) Computational Hydrology International, Irvine, California.
(2) US Army Corp. of Engineers, CRREL, Hannover, New Hampshire, USA.

1. Abstract

The Complex Variable Boundary Element Method or CVBEM is used to develop a computer model (CVBFR1) for estimating the location of the freezing front in soil-water phase change problems. Because the numerical technique is a boundary *integral approach*, the control volume thermal regime is modeled with respect to the boundary values and, therefore, the CVBFR1 data entry requirements are significantly less than that usually required of domain methods such as finite-differences or finite-elements. Soil-water phase change along the freezing front is modeled as a simple balance between computed heat flux and the evolution of soil-water volumetric latent heat of fusion.

2. Key Words

Boundary Element Method, Soil-Water Phase Change, Potential Problems, Complex Variables

3. Overview

3.0 Introduction

The Complex Variable Boundary Element Method or CVBEM is used to develop a computer model for estimating the location of the freezing front in soil-water phase change problems. This computer program, CVBFR1, is based on the following major assumptions:

a) the problem is two-dimensional,
b) the entire soil system is homogeneous and isotropic,
c) the problem thermal boundary conditions are constant values of temperature (or stream function),
d) soil-water flow effects are neglected (the problem is strictly geothermal),
e) all heat flow from the freezing front is within the control volume, there is no heat flux associated with the freezing front from exterior of the control volume,
f) the freezing front movement is sufficiently slow such that heat flux along the moving boundary can be determined by assuming steady state heat flow conditions for small durations of time (i.e., timesteps).

The CVBEM is used to model the thermal regime of the soil system. The theory and development of the CVBEM is given in Hromadka (1984)[1]. Because the numerical technique is a boundary integral approach, the control volume regime is modeled with respect to the boundary values and, therefore, the CVBFR1 data entry requirements are significantly less than that usually required of domain methods such as finite-differences or finite-elements.

Soil-water phase change along the freezing front is modeled as a simple balance between computed heat flux and the evolution of soil-water volumetric latent heat of fusion. To model the displacement of the freezing front, program CVBFR1 provides two options:

a) displace the freezing front coordinates with respect to changes in the y-coordinate only,
b) displace the freezing front coordinates with respect to a vector normal to the freezing front boundary.

4. Modeling Approach

4.0 Introduction

The use of the Complex Variable Boundary Element Method to model soil-water phase change effects is a new numerical approach to this class of problems. In previous work, Hromadka and Guymon (1982)[2] applied the complex variable boundary element method (CVBEM) to the problem of predicting freezing fronts in two-dimensional soil systems. Hromadka et al. (1983)[3] subsequently compare the CVBEM solution to a domain solution method and prototype data for the Deadhorse Airport runway at Prudhoe Bay, Alaska. In another work, the model is further extended to include an approximation of soil-water flow (Hromadka and Guymon, (1984a)[4]. In contrast to the CVBEM approach, an example in the use of real variable boundary element methods (Brebbia, 1978)[5] in the approximation of such moving boundary phase change problems and a review of the pertinent literature is given in O'Niell (1983)[6].

Hromadka and Guymon (1984b)[7] develop an error estimation scheme which exactly evaluates the error distribution on the problem boundary that results from the CVBEM approximator matching the known boundary conditions. This error determination is used to add or delete boundary nodes to improve accuracy. Thus, the CVBEM permits a direct and immediate determination of the approximation error involved in solution of an assumed Laplacian system. The modeling accuracy is evaluated by the model-user in the determination of an approximative boundary upon which the CVBEM provides an exact solution. Although inhomogeneity (and anisotropy) can be included in the CVBEM model, the resulting fully-populated matrix system quickly becomes large. Therefore in this work, the domain is assumed homogeneous and isotropic except for differences in frozen and thawed conduction parameters for freezing and thawing problems, respectively.

A major benefit in the use of the CVBEM over other numerical methods (including real variable boundary element methods and domain methods such as finite-differences and finite-elements) is the accurate and easy-to-use "approximative boundary" error evaluation technique. Other numerical methods can be evaluated for modeling error (where exact mathematical solutions do not exist) by increasing nodal point densities and comparing the resulting changes in predicted nodal values of the governing equation's state variable. In contrast, the CVBEM approximative boundary error evaluation technique is simply the process of locating the (x,y) points where the CVBEM approximative function meets the specified boundary condition values (the approximative boundary), and comparing the resulting plot to the true problem boundary.

A major benefit for using the CVBEM error evaluation technique is that highly accurate solutions for two-dimensional potential problems can be obtained. Often, the CVBEM approximation analysis is terminated when the approximative boundary differs from the true problem boundary to within the construction tolerance of the project, resulting in an exact CVBEM model of a probable constructed version of the engineered plan drawings. Consequently the CVBEM approach can be used directly in engineering applications, or used to provide a wide range of highly accurate approximations for two-dimensional phase change problems (where the freezing front movement is slow; see section 4.2) for checking modeling results produced by other numerical methods.

4.1 Heat Flow Model

For a wide range of soil freezing (or thawing) problems, the freezing front movement is sufficiently slow such that the governing heat flow equation can be modeled using a timestepped steady state heat flow approximation. That is for small durations of time, the heat flux along the freezing front can be computed assuming the temperature distribution within the frozen (or thawed) regions are potential functions (i.e., the Laplace equation applies). Figure 4.1 illustrates a typical two-phase problem definition where the heat flow model solves for heat flux along the freezing front by solving the Laplace equation (by use of potential functions) in both the frozen and thawed regions.

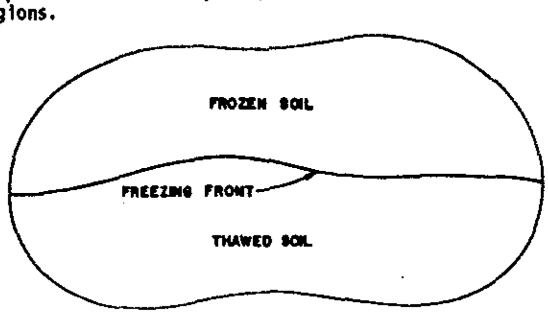

Figure 4.1 Typical Two-Phase Problem Definition

To develop mathematical models of the Laplace equation in each region, a CVBEM approximator is generated which matches specified boundary conditions of either temperature or flux at nodal point locations on the problem boundary and freezing front. The CVBEM approximator exactly satisfies the Laplace equation; consequently there is no modeling error in solving the governing Laplace equation (heat flow model), there is only error in matching the boundary conditions continuously. Figure 4.2 shows an example roadway problem where the freezing front is initially located some known distance below the surface. Boundary conditions for the example problem and a nodal point placement scheme are shown in Fig. 4.3.

The heat flow model in CVBFR1 develops a CVBEM potential function which satisfies the Laplace equation within the boundary of Fig. 4.3. Hromadka (1984)[1] provides the necessary background material for the CVBEM development and the approximative boundary technique.

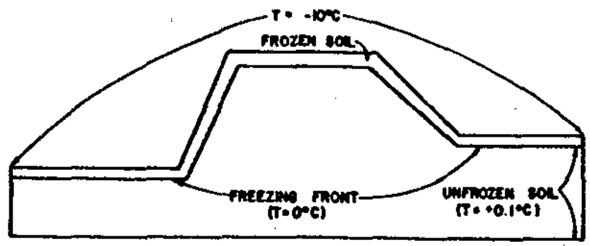

Figure 4.2 Typical Roadway Embankment Problem

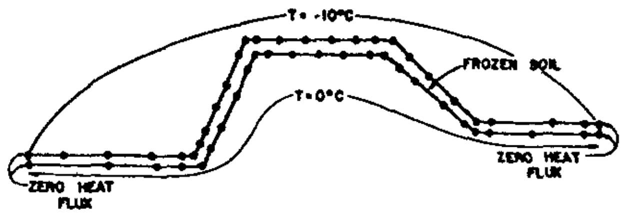

Figure 4.3 Nodal Point Placement and Boundary Conditions for Fig. 4.2 Problem

The usual modeling procedure is to use the approximative boundary technique to analyze the initial condition CVBEM model. After the analyst is satisfied with the CVBEM approximator and its associated level of accuracy then the CVBFR1 program is executed to model the freezing front evolution.

4.2 Phase Change Model

For each timestep, a CVBEM approximator is generated by program CVBFR1 based on the problem geometry and boundary conditions. Heat flux is computed along the freezing front using the CVBEM approximation stream function values. The heat flux estimates are assumed to directly equate to the rate of freezing (or thawing) of a volume of soil at the freezing front. Consequently, a freezing process for the example of Fig. 4.3 results in a downward migration of the freezing front such that the product of the timestep and heat flux equals the latent heat evolved by the change in freezing front coordinates.

Two freezing front displacement models are available in program CVBFR1:

(a) All displacement occurs in the vertical direction. This simplified model is generally appropriate for many roadway problems.
(b) Displacement computed based on an outward normal fector. This model is the most accurate, but requires additional computational effort than the vertical displacement model. Figure 4.4 shows the nodal point displacement in a direction which balances the angles to go between the normal vector and boundary elements.

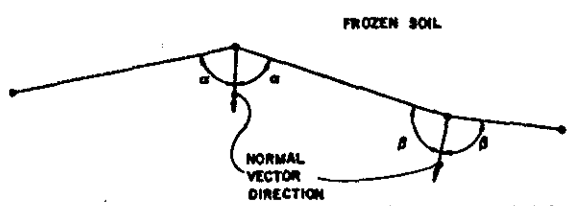

Figure 4.4 Normal Vector Coordinate Displacement Model (note balanced angles for each normal vector)

4.3 Program CVBFR1 Characteristics

(a) Class of Problems Modeled

Program CVBFR1 may be used to model soil-water freezing (or thawing) in two-dimensional, homogeneous, isotropic domains. As illustrated by the example problem in Figs. 4.2 and 4.3, only one region is modeled (i.e., either entirely frozen or entirely thawed) and the freezing front forms part of the control volume's boundary. For example, program CVBFR1 may be used to study the freezing front advancement into a soil system where the soil system is initially close to the freezing poing depression temperature, and negligible heat flow to the freezing front is contributed from the underlying soil system. A schematic of the problem domain and boundary conditions used in CVBFRI are illustrated in Fig. 4.5. Another characteristic of CVBFR1 is that the boundary conditions of the problem are held constant for the entire simulation. Additionally, the initial conditions of the problem are assumed to be near steady state with the freezing front specified some distance below the top of the control volume boundary (control surface).

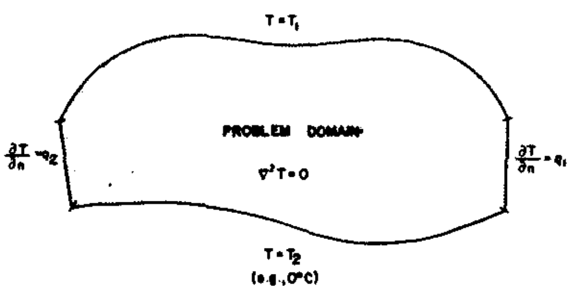

Figure 4.5 Program CVBFR1 Boundary Condition Characteristics

(b) The CVBFR1 Modeling Procedure

The modeling procedure used in the CVBFR1 program is shown schematically in Fig. 4.6 for the case of a soil freezing problem. It is assumed in Fig. 4.6 that the analyst has developed a good CVBEM approximator for the initial conditions fo the problem by using the approximative boundary technique to locate nodal points on the problem boundary. Typically, the most difficult modeling problem occurs when the freezing front is closest to the top of the problem boundary such as shown in Fig. 4.3. Consequently, the CVBEM nodal placement should be concluded based on the problem's smallest anticipated distance to the freezing front. For example, the roadway problem shown in Fig. 4.3 spans a width of 50m; the corresponding distance to the freezing front for initial conditions (freezing problem) is assumed to be 0.25m.

5. Program CVBFR1

5.0 Introduction

CVBFR1 is a CVBEM program with the capability to estimate the moving position of a slow-moving freezing front in soils. The CVBFR1 program uses either subroutine FRT1 or FRT2 to estimate the displacement of the freezing front where subroutine FRT1 is based upon a vertical shifting and FRT2 uses the outer normal direction to calculate the change in nodal point coordinates.

5.1 Problem Set-up

The problem domain is assumed to be a homogeneous isotropic soil mixture enclosed by the problem boundary.

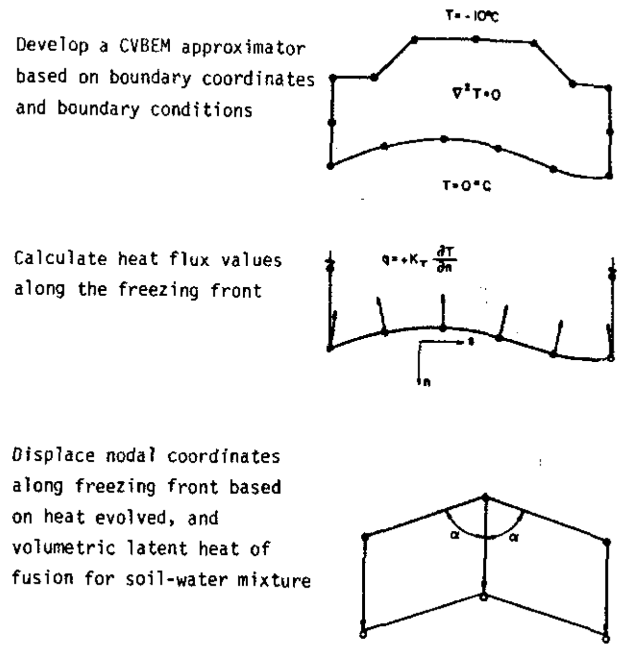

Figure 4.6 CVBFR1 Modeling Procedure

Nodal points are located on the problem boundary and are numbered in sequence in a counterclockwise direction from 1 to NNOD.

Nodal points are generally placed closer together near angle points of the problem boundary, or where boundary condition values (or types of boundary conditions) change. This increase in nodal density reduces the error in integrating a trial function (straight line interpolation functions are used in CVBFR1) which becomes inaccurate near singularities of the potential function, temperature.

The product of the latent heat of fusion for soil-water and the uniform soil porosity value is used as the volumetric latent heat of fusion for the soil-water (or soil-ice) mixture. The thermal conductivity value is used to estimate the normal heat flux values along the freezing front.

5.2 Input Data

Input data for program CVBFR1 is as follows:

VARIABLE	DATA FILE LINE
KODE	Line 1
NNOD, NFRS, NFRE	Line 2
COND, XLAT, POR	Line 3
DELT, SIMUL, OUT, ID	Line 4
X(I), Y(I), KTYPE(I), VALUE(I); I=1 to NNOD	Line 5
	.
	.
	.
X(NNOD), Y(NNOD), KTYPE(NNOD), VALUE(NNOD);	Line NNOD + 4
(END OF FILE)	

where:

VARIABLE

KODE	=	1, For vertical displacement of freezing front coordinates
		2, Use outward normal vector to estimate nodal point displacements
NNOD	=	Total number of nodes on boundary
NFRS	=	First node number of the freezing front contour
NFRE	=	Last node number of the freezing front contour
COND	=	Thermal conductivity of a homogeneous isotropic soil mixture
XLAT	=	Latent heat of fusion for soil-water
POR	=	Porosity of soil
DELT	=	Increment for time advancement model
SIMUL	=	Total simulation time
OUT	=	Output period
ID	=	0, Detailed output (see Example 1)
		1, Summary output (see Example)
X(I), Y(I)	=	(x,y) coorindates of node I in first quadrant
KTYPE(I)	=	1, Prescribed temperature value
		2, Prescribed stream function value
		3, Prescribed flux value
VALUE(I)	=	Prescribed value according to KTYPE(I). For efflux, VALUE(I) = efflux/conductivity

Note: The units of XLAT, COND, DELT, SIMUL, and OUT should be consistent.

5.3 Application

Example 1: Computing the Freezing Front Location in a Roadway Embankment

A roadway embankment (Fig. 5.1) problem is used to issutrate the applicatin of program CVBFR1. The input data and program output (in English units) for the example problem is provided in the following: (note that the first line is a "1" or "2" for using subroutines FRT1 and FRT2, respectively):

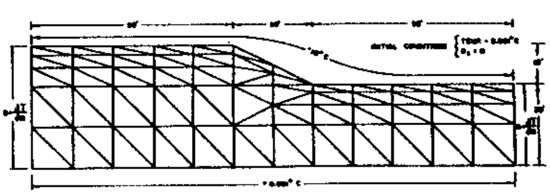

Figure 5.1a Example Problem Roadway Embankment Discretized into Finite Elements (Several node numbers are shown)

Example 2: Nodal Density and Timestep Size Sensitivity Analysis

A sensitivity analysis is prepared examining different time increments and nodal point densities and the resulting effects on CVBFR1 modeling results. Figure 5.1 shows the different nodal densities and Fig. 5.2 shows the

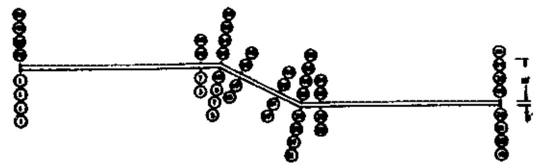

Figure 5.1b Nodal Point Number for 4 CVBEM Nodal Densities

Timestep \ Section	A-A	B-B	C-C	D-D	Number of Nodes
6 hrs	1.3466 (1.3459)	1.4645 (1.4661)	1.2594 (1.2632)	1.3466 (1.3459)	78
12	1.3489 (1.3482)	1.4683 (1.4698)	1.2604 (1.2641)	1.3489 (1.3482)	78
24	1.3537 (1.3530)	1.4764 (1.4770)	1.2625 (1.2660)	1.3537 (1.3529)	78
60	1.3697 (1.3689)	1.5023 (1.4829)	1.2687 (1.2709)	1.3698 (1.3689)	78

Number of Nodes \ Section	A-A	B-B	C-C	D-D	Timestep (hours)
78	1.3466 (1.3459)	1.4645 (1.4661)	1.2594 (1.2632)	1.3466 (1.3459)	6
62	1.3466 (1.3459)	1.4645 (1.4661)	1.2594 (1.2632)	1.3466 (1.3459)	6
62	1.3698 (1.3689)	1.5023 (1.4829)	1.2687 (1.2709)	1.3698 (1.3689)	60
46	1.3461 (1.3454)	1.4649 (1.4667)	1.2591 (1.2630)	1.3467 (1.3459)	6
46	1.3696 (1.3688)	1.5026 (1.4834)	1.2685 (1.2708)	1.3698 (1.3690)	60
30	1.3458 (1.3451)	1.4797 (1.4778)	1.2365 (1.2444)	1.3468 (1.3460)	6
30	1.3693 (1.3686)	1.5241 (1.4887)	1.2392 (1.2472)	1.3699 (1.3690)	60

1.3466: Results from Vertical Displacement Model
(1.3459): Results from Normal Vector Displacement Model

Figure 5.2 Comparison of CVBEM Model Results in Predicting Freezing Front Location (Stefan Solution at 60 hrs. is 1.344 ft.depth)

results from the several CVBEM models. From the analysis, it appears that a small timestep (6-hours) is preferred, but a large timestep such as 60 hours results in an error with respect to the one-dimensional Stefan solution of only 2-percent. Additionally, a relatively sparse nodal density of only 30 nodes results in a satisfactory condition.

Example 3: Comparison to Two-Dimensional Domain Modeling Results

The CVBFR1 modeling results for the previous example are compared to results from a Nodal Domain Integration (NDI) two-dimensional phase change model in Fig. 5.3. The NDI model is based upon an isothermal soil-water phase change approximation, and uses an apparent heat capacity approach to model the freezing front evolution in the fixed grid domain model.

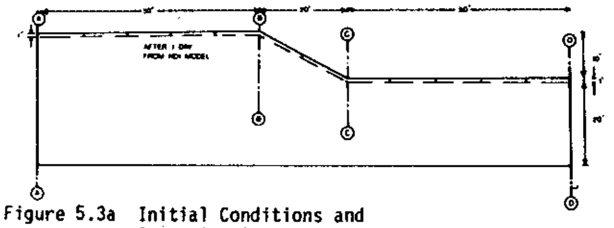

Figure 5.3a Initial Conditions and Cross Section Locations

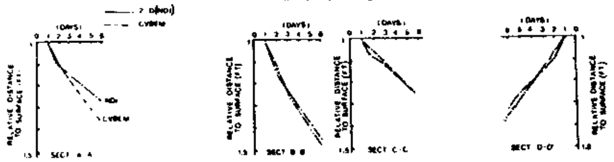

Figure 5.3b Comparison of CVBEM and NDI Modeling Results

6.0 References

[1] Hromadka II, T. V., "The Complex Variable Boundary Element Method," Springer-Verlag, 1984, 250 pgs.

[2] Hromadka II, T. V. and Guymon, G. L., "Application of a Boundary Integral Equation to Prediction of Freezing Fronts in Soils," CRST, (6), 1981, pp. 115-121.

[3] Hromadka II, T. V., Guymon, G. L. and Berg, R. L., "Comparison of Two-Dimensional Domain and Boundary Integral Geothermal Models with Embankment Freeze-Thaw Field Data," Permafrost, Fourth International Conference, Proceedings, National Academy Press, 1983.

Hromadka II, T. V. and Guymon, G. L., "Simple Model of Ice Segregation Using an Analytic Function to Model Heat and Soil Water Flow," Third International Symposium on Offshore Mechanics and Artic Engr., New Orleans, LA. Also republished in special edition form in A.S.M.E. Energy Division Journal, Sept. 1984a.

[5] Brebbia, C. A. "The Boundary Element Method for Engineers," Pentech Press, London, 1978.

[6] O'Niell, K., "Boundary Integral Equation Solution of Moving Boundary Phase Change Problems," Int. J. Num. Mech. Engg., 1983, 19:1825-1850.

[7] Hromadka II, T. V. and Guymon, G. L., "An Algorithm to Reduce Approximation Error from the CVBEM," Numerical Heat Transfer, 1984b.

PROGRAM CVBFR1 Data Input

(Example Problem)

KODE = 1 or 2 (see text)

```
30 1 15
10 80 .4
6 120 120 0
0 10 2 0
25 10 1 0
49 10 1 0
49.9 10 1 0
50 10 1 0
50.1 9.95 1 0
51 9.5 1 0
60 5 1 0
69 .5 1 0
69.9 .05 1 0
70 0 1 0
70.1 0 1 0
71 0 1 0
95 0 1 0
120 0 1 0
120 1 1 -10
95 1 1 -10
71 1 1 -10
70.1 1 1 -10
70 1 1 -10
69.9 1.05 1 -10
69 1.5 1 -10
60 6 1 -10
51 10.5 1 -10
50.1 10.95 1 -10
50 11 1 -10
49.9 11 1 -10
49 11 1 -10
25 11 1 -10
0 11 1 -10
```

- The computer modeling results using FRTI (KODE = 1) are as follows:

TIME INCREMENT = 6.0000
TOTAL SIMULATION TIME = 120.0000
CONDUCTIVITY = 10.0000
LATENT HEAT = 80.0000
POROSITY = 0.4000

NODE NO.	X(I)	Y(I)	KTYPE(I) 1=SV!2=SF 3=EFFLUX	VALUE	ANGLE(I)
1	0.00000	10.00000	2	0.00000	90.00
2	25.00000	10.00000	1	0.00000	180.00
3	49.00000	10.00000	1	0.00000	180.00
4	49.90000	10.00000	1	0.00000	180.00
5	50.00000	10.00000	1	0.00000	206.57
6	50.10000	9.95000	1	0.00000	180.00
7	51.00000	9.50000	1	0.00000	180.00
8	60.00000	5.00000	1	0.00000	180.00
9	69.00000	0.50000	1	0.00000	180.00
10	69.90000	0.05000	1	0.00000	180.00
11	70.00000	0.00000	1	0.00000	153.43
12	70.10000	0.00000	1	0.00000	180.00
13	71.00000	0.00000	1	0.00000	180.00
14	95.00000	0.00000	1	0.00000	180.00
15	120.00000	0.00000	1	0.00000	90.00
16	120.00000	1.00000	1	-10.00000	90.00
17	95.00000	1.00000	1	-10.00000	180.00
18	71.00000	1.00000	1	-10.00000	180.00
19	70.10000	1.00000	1	-10.00000	180.00
20	70.00000	1.00000	1	-10.00000	206.57
21	69.90000	1.05000	1	-10.00000	180.00
22	69.00000	1.50000	1	-10.00000	180.00
23	60.00000	6.00000	1	-10.00000	180.00
24	51.00000	10.50000	1	-10.00000	180.00
25	50.10000	10.95000	1	-10.00000	180.00
26	50.00000	11.00000	1	-10.00000	153.43
27	49.90000	11.00000	1	-10.00000	180.00
28	49.00000	11.00000	1	-10.00000	180.00
29	25.00000	11.00000	1	-10.00000	180.00
30	0.00000	11.00000	1	-10.00000	90.00

Cauchy Program Results

TIME = 120.0000

NODE NUMBER	STATE VARIABLE	STREAM FUNCTION
1	-0.0645	0.0000
2	0.0000	186.8884
3	0.0000	366.0428
4	0.0000	373.0393
5	0.0000	373.7330
6	0.0000	374.6362
7	0.0000	382.6803
8	0.0000	462.5811
9	0.0000	542.9409
10	0.0000	550.4897
11	0.0000	551.2713
12	0.0000	552.0007
13	0.0000	558.4281
14	0.0000	739.2386
15	0.0000	927.1421
16	-10.0000	927.1481
17	-10.0000	739.2394
18	-10.0000	558.3662
19	-10.0000	550.7808
20	-10.0000	549.4852
21	-10.0000	548.0763
22	-10.0000	539.4200
23	-10.0000	459.0154
24	-10.0000	379.2508
25	-10.0000	372.4201
26	-10.0000	372.0577
27	-10.0000	371.7505
28	-10.0000	365.9584
29	-10.0000	186.8909
30	-10.0000	0.0024

CVBEM Approximation Function Nodal Values:

NODE NUMBER	STATE VARIABLE	STREAM FUNCTION
1	-0.0582	-0.0034
2	-0.0123	186.8887
3	-0.0574	366.1034
4	-0.1052	373.1109
5	0.0292	373.7366
6	-0.1064	374.5667
7	-0.0448	382.6476
8	-0.0065	462.5815
9	0.0311	542.9185
10	0.0745	550.4381
11	-0.0335	551.2729
12	0.0556	552.0399
13	0.0402	558.4709
14	-0.0026	739.2393
15	0.0013	927.1403
16	-10.0028	927.1437
17	-10.0216	739.2404
18	-9.9618	558.3270
19	-9.9189	550.7245
20	-9.8962	549.4889
21	-9.9198	548.1329
22	-9.9632	539.4475
23	-9.9878	459.0158
24	-10.0409	379.2812
25	-10.0944	372.4865
26	-10.0653	372.0609
27	-10.0806	371.6945
28	-10.0510	365.9055
29	-9.9580	186.8916
30	-9.9981	-0.0011

Nodal Point Relative Error Values:

1	-0.0063	0.0034
2	0.0123	-0.0004
3	0.0574	-0.0605
4	0.1052	-0.0717
5	-0.0292	-0.0036
6	0.1064	0.0695
7	0.0448	0.0327
8	0.0065	-0.0004
9	-0.0311	0.0225
10	-0.0745	0.0516
11	0.0335	-0.0016
12	-0.0554	-0.0392
13	-0.0402	-0.0428
14	0.0026	-0.0007
15	-0.0013	0.0018
16	0.0028	0.0045
17	0.0216	-0.0010
18	-0.0382	0.0392
19	-0.0811	0.0563
20	-0.1038	-0.0037
21	-0.0802	-0.0566
22	-0.0368	-0.0275
23	-0.0122	-0.0005
24	0.0409	-0.0304
25	0.0944	-0.0664
26	0.0653	-0.0031
27	0.0806	0.0560
28	0.0510	0.0529
29	-0.0420	-0.0007
30	-0.0019	0.0035

New Coordinates of the Freezing Front

Node	X-Coord.	Y-Coord.
1	0.0000	9.6542
2	24.9999	9.6543
3	48.9923	9.6363
4	49.8163	9.5782
5	49.8736	9.5383
6	49.9673	9.5380
7	50.8326	9.1534
8	59.8322	4.6640
9	68.8452	0.1761
10	69.8043	-0.2259
11	69.9464	-0.2389
12	70.0668	-0.2815
13	70.9948	-0.3342
14	95.0000	-0.3469
15	120.0000	-0.3468

- The output (summary) data using FRT2 (KODE = 2) consists of:

```
TIME INCREMENT =   6.0000
TOTAL SIMULATION TIME = 120.0000
CONDUCTIVITY =  10.0000
LATENT HEAT =  80.0000
POROSITY = 0.4000
```

NODE NO.	X(I)	Y(I)	KTYPE(I) 1=SV;2=SF 3=EFFLUX	VALUE	ANGLE(I)
1	0.00000	10.00000	2	0.00000	90.00
2	25.00000	10.00000	1	0.00000	180.00
3	49.00000	10.00000	1	0.00000	180.00
4	49.90000	10.00000	1	0.00000	180.00
5	50.00000	10.00000	1	0.00000	206.57
6	50.10000	9.95000	1	0.00000	180.00
7	51.00000	9.50000	1	0.00000	180.00
8	60.00000	5.00000	1	0.00000	180.00
9	69.00000	0.50000	1	0.00000	180.00
10	69.90000	0.05000	1	0.00000	180.00
11	70.00000	0.00000	1	0.00000	153.43
12	70.10000	0.00000	1	0.00000	180.00
13	71.00000	0.00000	1	0.00000	180.00
14	95.00000	0.00000	1	0.00000	180.00
15	120.00000	0.00000	1	0.00000	90.00
16	120.00000	1.00000	1	-10.00000	90.00
17	95.00000	1.00000	1	-10.00000	180.00
18	71.00000	1.00000	1	-10.00000	180.00
19	70.10000	1.00000	1	-10.00000	180.00
20	70.00000	1.00000	1	-10.00000	206.57
21	69.90000	1.05000	1	-10.00000	180.00
22	69.00000	1.50000	1	-10.00000	180.00
23	60.00000	6.00000	1	-10.00000	180.00
24	51.00000	10.50000	1	-10.00000	180.00
25	50.10000	10.95000	1	-10.00000	180.00
26	50.00000	11.00000	1	-10.00000	153.43
27	49.90000	11.00000	1	-10.00000	180.00
28	49.00000	11.00000	1	-10.00000	180.00
29	25.00000	11.00000	1	-10.00000	180.00
30	0.00000	11.00000	1	-10.00000	90.00

New Coordinates of the Freezing Front

Time = 120.0000

Node	X-Coord.	Y-Coord.
1	0.0000	9.6542
2	25.0000	9.6542
3	49.0000	9.6369
4	49.9000	9.5686
5	50.0000	9.5203
6	50.1000	9.5124
7	51.0000	9.1105
8	60.0000	4.6188
9	69.0000	0.1376
10	69.9000	-0.2377
11	70.0000	-0.2365
12	70.1000	-0.2807
13	71.0000	-0.3338
14	95.0000	-0.3470
15	120.0000	-0.3468

7. Appendix A

```
C
C   MAIN PROGRAM
C
C   THIS CAUCHY PROGRAM ( FREEZING OR THAWING FRONT ADVANCEMENT )
C   USES SUBROUTINES CAUCH1,CAUCH2,CAUCH3,CAUCH4,CAUCH5,HOM,ANG,FRONT
C
C   BASED ON THE APPROXIMATION FUNCTION
C
C       IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        COMMON/BLK 1/X(100)
        COMMON/BLK 2/Y(100)
        COMMON/BLK 3/KTYPE(100)
        COMMON/BLK 4/VALUE(100)
        VIRTUAL P(100,100)
        COMMON/BLK 6/S(100)
        COMMON/BLK 7/ANGLE(100)
        COMMON/BLK 8/NAT(100)
        DIMENSION REX(100),REY(100)
        DIMENSION HIY(100)
C
C   OPEN DATA FILES
C
        NRD=1
        NWT=2
        OPEN(UNIT=NRD,NAME='CAUFRT.DAT',TYPE='OLD')
        OPEN(UNIT=NWT,NAME='CAUCHY.ANS',TYPE='NEW')
```

```
C
C  READ DATA
C
C...NOTE: NODE NUMBER PLUS NUMBER OF EFFLUX B.C.
C          (NNODP=NNOD+NNAT) CAN NOT EXCEED '100'
      READ(NRD,*)KODE
      READ(NRD,*)NNOD,NFRS,NFRE
      READ(NRD,*)COND,XLAT,POR
      READ(NRD,*)DELT,SIMUL,OUT,ID
      WRITE(NWT,601)DELT,SIMUL,COND,XLAT,POR
601   FORMAT(///,6X,'TIME INCREMENT = ',F8.4,/,6X,'TOTAL SIMULATION',
     1' TIME = ',F8.4,/,6X,'CONDUCTIVITY = ',F8.4,/,6X,'LATENT',
     2' HEAT = ',F8.4,/,6X,'POROSITY = ',F6.4,/)
C...VALUE OF EFFLUX B.C = EFFLUX/CONDUCTIVITY
      DO 7 I=1,NNOD
7     READ(NRD,*)(X(I),Y(I),KTYPE(I),VALUE(I))
      CALL ANG(NNOD)
      WRITE(NWT,10)
10    FORMAT(6X,'NODE',6X,'X(I)',6X,'Y(I)',4X,'KTYPE(I)',3X,'VALUE',
     15X,'ANGLE(I)',/,7X,'NO.',24X,'1=SV,2=SF',/,35X,'3=EFFLUX')
      DO 9 I=1,NNOD
      WRITE(NWT,8)I,X(I),Y(I),KTYPE(I),VALUE(I),ANGLE(I)
8     FORMAT(3X,I5,5X,2F10.5,I5,5X,F10.5,F10.2)
9     CONTINUE
      WRITE(NWT,602)
602   FORMAT(72('-'))
C
C  CHECK NATURAL OR EFFLUX BOUNDARY CONDITION
C
      NNAT=0
      DO 3 I=1,NNOD
      X(I)=X(I)
      Y(I)=Y(I)
      IF(KTYPE(I).NE.3)GO TO 3
      NNAT=NNAT+1
      NNODP=NNOD+NNAT
      NAT(I)=NNODP
3     CONTINUE
      IF(NNAT.EQ.0)NNODP=NNOD
C
C  PREPARE GLOBAL MATRICES
C
C...ZERO ARRAYS
      ITER=IFIX(SIMUL/DELT)
      IOUT=IFIX(OUT/DELT)
      KOUT=0
      DO 9999 IIII=1,ITER
      KOUT=KOUT+1
      DO 5 I=1,NNODP
5     S(I)=0.
      DO 6 I=1,NNODP
      DO 6 II=1,NNODP
6     P(I,II)=0.
      DO 1000 J=1,NNOD
C...ACCOMODATE DIAGONAL NODE
      I=J-1
      IF(I.EQ.0)I=NNOD
      K=J+1
      IF(K.GT.NNOD)K=1
      CALL CAUCH1(J,I,K,A,B,C,D)
      AJ=A
      BJ=ANGLE(J)/180.*3.141593
      CALL CAUCH2(J,I,K,A,B,C,D,AJ,BJ,P)
C...ACCOMODATE REMAINING CONTOUR NODAL POINTS
      NELE=NNOD-2
      DO 500 K=1,NELE
      M=J+K
      IF(M.GT.NNOD)M=M-NNOD
      N=M+1
      IF(N.GT.NNOD)N=N-NNOD
      CALL CAUCH1(J,M,N,A,B,C,D)
      CALL CAUCH2(J,M,N,A,B,C,D,AJ,BJ,P)
500   CONTINUE
1000  CONTINUE
C
C  PREPARE RELATIVE ERROR ANALYSIS
C
      CALL CAUCH3(NNODP,NWT,P)
      TIME=DELT*FLOAT(IIII)
      IF(KOUT .EQ. IOUT)CALL CAUCH4(NNOD,NWT,TIME,ID)
C
C  ASSIGN BOUNDARY NODAL POINT VALUES
C
      DO 7010 I=1,NNOD
      IF(KTYPE(I).EQ.2)GO TO 7015
      IF(KTYPE(I).EQ.3)GO TO 7016
      REX(I)=VALUE(I)
      REY(I)=S(I)
      GOTO 7010
7015  REX(I)=S(I)
      REY(I)=VALUE(I)
      GOTO 7010
7016  II=NAT(I)
      REX(I)=S(I)
      REY(I)=S(II)
7010  CONTINUE
C
C  CALCULATE RELATIVE ERROR VALUES
C
      CALL HOM(REX,REY,NNOD,NWT,KOUT,IOUT,H1Y,ID)
C
C  UPDATE THE NEW INTERNAL ANGLES AND POSITIONS
C  OF THE FREEZING OR THAWING FRONT
C
      CALL FRONT(NNOD,NFRS,NFRE,COND,XLAT,POR,DELT,H1Y,KODE)
C
C  OUTPUT THE NEW POSITIONS OF THE FREEZING OR THAWING FRONT
C
      IF(KOUT .NE. IOUT)GO TO 9999
      IF(ID .NE. 0)WRITE(NWT,605)TIME
605   FORMAT(/,6X,'TIME = ',F8.2,/)
      WRITE(NWT,603)
603   FORMAT(/,6X,'NEW COORDINATES OF THE FREEZING OR THAWING FRONT'
     1,//,12X,'NODE',4X,'X-COORD.',5X,'Y-COORD.',/)
      DO 600 I=NFRS,NFRE
      XA=X(I)
      YA=Y(I)
      WRITE(NWT,604)I,XA,YA
604   FORMAT(11X,I3,5X,F8.4,6X,F8.4)
600   CONTINUE
      WRITE(NWT,602)
      KOUT=0
9999  CONTINUE
      CLOSE(UNIT=NRD)
      CLOSE(UNIT=NWT)
      STOP
      END
C-----------------------------------------------------------------------
C  SUBROUTINE CAUCH1
C-----------------------------------------------------------------------
      SUBROUTINE CAUCH1(J,M,N,A,B,C,D)
C     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/BLK 1/X(100)
      COMMON/BLK 2/Y(100)
      COMMON/BLK 3/KTYPE(100)
      COMMON/BLK 7/ANGLE(100)
C
C  SUBROUTINE TO DETERMINE BOUNDARY ELEMENT GEOMETRIC VALUES
C
C...CALCULATE VECTOR LENGTHS
      XLN=SQRT((Y(N)-Y(J))**2+(X(N)-X(J))**2)
      XLM=SQRT((Y(M)-Y(J))**2+(X(M)-X(J))**2)
      XXX=XLN/XLM
      A=ALOG(XXX)
C     A=DLOG(XLN/XLM)
C...DETERMINE ANGLE ARITHMETIC
      ZMX=(X(M)-X(J))/XLM
      ZMY=(Y(M)-Y(J))/XLM
      ZNX=(X(N)-X(J))/XLN
      ZNY=(Y(N)-Y(J))/XLN
      CALL CAUCH5(ZNX,ZNY,ANGLEN)
      CALL CAUCH5(ZMX,ZMY,ANGLEM)
      B=ANGLEN-ANGLEM
C...ACCOMODATE CENTRAL ANGLE DETERMINATION BEING BACKWARDS
      IF(M.EQ.(J-1) .OR. N.EQ.(J+1))GO TO 98
C...ACCOMODATE BRANCH-CUT EFFECTS
      IF(B.LT.-3.14159)B=B+6.2831853
      IF(B.GT. 3.14159)B=B-6.2831853
      GOTO 99
98    CONTINUE
      B=ANGLE(J)
99    CONTINUE
C
C..COMPLEX VARIABLE ARITHMETIC
C
      F=(X(N)-X(M))**2+(Y(N)-Y(M))**2
      C=A*(X(N)-X(M))-B*(Y(N)-Y(M))
      D=B*(X(N)-X(M))+A*(Y(N)-Y(M))
      C=C/F
      D=D/F
      RETURN
      END
C-----------------------------------------------------------------------
C  SUBROUTINE CAUCH2
C-----------------------------------------------------------------------
      SUBROUTINE CAUCH2(J,M,N,A,B,C,D,AJ,BJ,P)
C     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/BLK 1/X(100)
      COMMON/BLK 2/Y(100)
      COMMON/BLK 3/KTYPE(100)
      COMMON/BLK 4/VALUE(100)
      VIRTUAL P(100,100)
      COMMON/BLK 6/S(100)
      COMMON/BLK 7/ANGLE(100)
      COMMON/BLK 8/NAT(100)
C
C  SUBROUINE TO ASSEMBLE BOUNDARY ELEMENTS
C  INTO GLOBAL MATRIX 'P' WITH VECTOR OF CONSTANTS,'S'
C
C
C  SUBROUINE TO ASSEMBLE BOUNDARY ELEMENTS
C  INTO GLOBAL MATRIX 'P' WITH VECTOR OF CONSTANTS,'S'
C
      F=AJ*AJ+BJ*BJ
      AZ=-AJ/F
      BZ=-BJ/F
      JJ=J-1
      IF(M.EQ.JJ)GOTO 100
      JJ=J+1
      IF(N.EQ.JJ)GOTO 100
C...ELEMENT DOES NOT CONTAIN NODE 'J'
      C1=(X(J)-X(M))*C-(Y(J)-Y(M))*D+1.
      C2=(X(J)-X(M))*D+(Y(J)-Y(M))*C
      C3=(X(J)-X(N))*C-(Y(J)-Y(N))*D+1.
      C4=(X(J)-X(N))*D+(Y(J)-Y(N))*C
      CC1=C1*AZ-BZ*C2
      CC2=C1*BZ+C2*AZ
      CC3=C3*AZ-C4*BZ
      CC4=C4*AZ+BZ*C3
      C1=CC1
      C2=CC2
```

```
      C3=CC3
      C4=CC4
C...ASSIGN COEFFICIENTS TO UNKNOWN HARMONIC VARIABLE
      IF(KTYPE(J).EQ.1)GO TO 5
C...DIAGONAL NODAL UNKNOWN HARMONIC IS THE STATE VARIABLE
C...USE REAL EQUATION
      G1=-C3
      G2=C4
      G3=C1
      G4=-C2
      GO TO 8
C...DIAGONAL UNKNOWN HARMONIC IS THE STREAM FUNCTION
C...USE IMAGINARY EQUATION
5     G1=-C4
      G2=-C3
      G3=C2
      G4=C1
8     IF(KTYPE(M).EQ.2)GOTO 10
      IF(KTYPE(M).EQ.3)GOTO 15
C...STATE VARIABLE SPECIFIED FOR NODE 'M'
      S(J)=S(J)-(G1)*VALUE(M)
      P(J,M)=P(J,M)+(G2)
      GO TO 50
C...EFFLUX SPECIFIED FOR NODE 'M'
15    S(J)=S(J)
      P(J,M)=P(J,M)+G1
      MF=NAT(M)
      P(J,MF)=P(J,MF)+G2
      GO TO 50
C...STREAM FUNCTION SPECIFIED FOR NODE 'M'
10    S(J)=S(J)-(G2)*VALUE(M)
      P(J,M)=P(J,M)+(G1)
50    IF(KTYPE(N).EQ.2)GOTO 60
      IF(KTYPE(N).EQ.3)GOTO 65
C...STATE VARIABLE SPECIFIED FOR NODE 'N'
      S(J)=S(J)-(G3)*VALUE(N)
      P(J,N)=P(J,N)+(G4)
      GO TO 250
C...EFFLUX SPECIFIED FOR NODE 'N'
65    S(J)=S(J)
      P(J,N)=P(J,N)+G3
      NF=NAT(N)
      P(J,NF)=P(J,NF)+G4
      GO TO 250
C...STREAM FUNCTION SPECIFIED FOR NODE 'N'
60    S(J)=S(J)-(G4)*VALUE(N)
      P(J,N)=P(J,N)+(G3)
      GO TO 250
C
C  BOUNDARY ELEMENT CONTAINS NODE 'J'
C
100   IF(KTYPE(J).EQ.2 .OR. KTYPE(J).EQ.3)GO TO 110
C...STATE VARIABLE SPECIFIED FOR NODE 'J'
C...USE IMAGINARY EQUATION
      IF(KTYPE(M).EQ.1)P(J,N)=P(J,N)+AZ
      IF(KTYPE(N).EQ.1)S(J)=S(J)-BZ*VALUE(N)
      IF(KTYPE(N).EQ.2)P(J,N)=P(J,N)+BZ
      IF(KTYPE(N).EQ.2)S(J)=S(J)-AZ*VALUE(N)
      IF(KTYPE(N).NE.3)GO TO 113
C...EFFLUX SPECIFIED FOR NODE 'N'
      S(J)=S(J)
      P(J,N)=P(J,N)+BZ
      NF=NAT(N)
      P(J,NF)=P(J,NF)+AZ
113   IF(KTYPE(M).EQ.2)GOTO 115
      IF(KTYPE(M).EQ.3)GOTO 114
      S(J)=S(J)+BZ*VALUE(M)
      P(J,M)=P(J,M)-AZ
      GO TO 200
115   S(J)=S(J)+AZ*VALUE(M)
      P(J,M)=P(J,M)-BZ
      GO TO 200
C...EFFLUX SPECIFIED FOR NODE 'M'
114   S(J)=S(J)
      P(J,M)=P(J,M)-BZ
      MF=NAT(M)
      P(J,MF)=P(J,MF)-AZ
      GO TO 200
C...STREAM FUNCTION SPECIFIED FOR NODE 'J'
110   IF(KTYPE(N).NE.1)GOTO 120
      S(J)=S(J)-AZ*VALUE(N)
      P(J,N)=P(J,N)-BZ
      GO TO 130
120   IF(KTYPE(N).NE.3)GO TO 111
C...EFFLUX SPECIFIED FOR NODE 'N'
      S(J)=S(J)
      P(J,N)=P(J,N)+AZ
      NF=NAT(N)
      P(J,NF)=P(J,NF)-BZ
      GO TO 130
111   S(J)=S(J)+BZ*VALUE(N)
      P(J,N)=P(J,N)+AZ
130   IF(KTYPE(M).NE.1)GO TO 140
      S(J)=S(J)+AZ*VALUE(M)
      P(J,M)=P(J,M)+BZ
      GO TO 200
140   IF(KTYPE(M).NE.3)GO TO 112
C...EFFLUX SPECIFIED FOR NODE 'M'
      S(J)=S(J)
      P(J,M)=P(J,M)-AZ
      MF=NAT(M)
      P(J,MF)=P(J,MF)+BZ
      GO TO 200
112   S(J)=S(J)-BZ*VALUE(M)
      P(J,M)=P(J,M)-AZ
200   IF(KTYPE(J).EQ.3)GO TO 150
      P(J,J)=P(J,J)-1.
      GO TO 250
C...EFFLUX SPECIFICED FOR NODE 'J'
150   JF=NAT(J)
      MF=NAT(M)
      DZZ=(X(J)-X(M))**2+(Y(J)-Y(M))**2
      DZZ=SQRT(DZZ)
      S(JF)=S(JF)-VALUE(J)*DZZ
      P(JF,JF)=1.
      IF(KTYPE(M).NE.3)P(JF,M)=-1.
      IF(KTYPE(M).EQ.3)P(JF,MF)=-1.
      P(J,J)=P(J,J)-1.
250   CONTINUE
      RETURN
      END
C-----------------------------------------------------------------
C  SUBROUTINE CAUCH3
C-----------------------------------------------------------------
      SUBROUTINE CAUCH3(NNOD,NWT,P)
C     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
C  THIS SUBROUTINE SOLVES A NNODxNNOD MATRIX SYSTEM.
C  GAUSSIAN ELIMINATION METHOD USED.
C
      VIRTUAL P(100,100)
      COMMON/BLK 6/S(100)
      N1=NNOD-1
      DO 100 K=1,N1
      K1=K+1
      C=P(K,K)
      IF(ABS(C)-.000001)10,10,70
10    DO 20 J=K1,NNOD
      IF(ABS(P(J,K))-.000001)20,20,15
15    DO 16 L=K,NNOD
      C=P(K,L)
      P(K,L)=P(J,L)
16    P(J,L)=C
      C=S(K)
      S(K)=S(J)
      S(J)=C
      C=P(K,K)
      GO TO 70
20    CONTINUE
30    WRITE(NWT,1)K
1     FORMAT(1X,'>>>>SINGULARITY IN ROW',I5)
      GO TO 300
70    C=P(K,K)
      DO 80 J=K1,NNOD
80    P(K,J)=P(K,J)/C
      S(K)=S(K)/C
      DO 90 I=K1,NNOD
      C=P(I,K)
      DO 99 J=K1,NNOD
99    P(I,J)=P(I,J)-C*P(K,J)
90    S(I)=S(I)-C*S(K)
100   CONTINUE
      IF(ABS(P(NNOD,NNOD))-.000001)30,30,120
120   S(NNOD)=S(NNOD)/P(NNOD,NNOD)
      DO 200 L=1,N1
      K=NNOD-L
      K1=K+1
      DO 200 J=K1,NNOD
200   S(K)=S(K)-P(K,J)*S(J)
300   CONTINUE
      RETURN
      END
C-----------------------------------------------------------------
C   SUBROUTINE CAUCH4
C-----------------------------------------------------------------
      SUBROUTINE CAUCH4(NNOD,NWT,TIME,ID)
C     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/BLK 3/KTYPE(100)
      COMMON/BLK 4/VALUE(100)
      COMMON/BLK 6/S(100)
      COMMON/BLK 8/NAT(100)
C
C     SUBROUTINE FOR OUTPUT
C
      IF(ID .NE. 0) RETURN
      WRITE(NWT,10) TIME
10    FORMAT(//////,40X,'CAUCHY PROGRAM RESULTS',/,6X,'TIME = ',F8.4)
      WRITE(NWT,12)
12    FORMAT(/,6X,'NODE',6X,'STATE',14X,'STREAM',/,5X,'NUMBER',
     C3X,'VARIABLE',12X,'FUNCTION')
      DO 50 I=1,NNOD
      IF(KTYPE(I).NE.3)GO TO 20
      II=NAT(I)
      WRITE(NWT,55)I,S(I),S(II)
20    IF(KTYPE(I).EQ.1)WRITE(NWT,55)I,VALUE(I),S(I)
      IF(KTYPE(I).EQ.2)WRITE(NWT,55)I,S(I),VALUE(I)
55    FORMAT(3X,I5,5X,F10.4,10X,F10.4)
50    CONTINUE
      RETURN
      END
C-----------------------------------------------------------------
C     SUBROUTINE CAUCH5
C-----------------------------------------------------------------
      SUBROUTINE CAUCH5(X,Y,ANGLE)
C     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
C
C     THIS SUBROUTINE DETERMINES THE POSITIVE ANGLE
C     OF COMPLEX POINT X+iY WITH RESPECT TO THE ORIGIN
C
      PI=ACOS(-1.)
      IF(X.EQ.0. .AND. Y.GT.0.)ANGLE=.5*PI
```

```
      IF(X.EQ.0.  .AND. Y.LT.0.)ANGLE=1.5*PI
      IF(X.GT.0.  .AND. Y.GE.0.)ANGLE=ATAN(Y/X)
      IF(X.LT.0.  .AND. Y.GE.0.)ANGLE=PI-ATAN(-Y/X)
      IF(X.LT.0.  .AND. Y.LT.0.)ANGLE=PI+ATAN(Y/X)
      IF(X.GT.0.  .AND. Y.LT.0.)ANGLE=2.*PI-ATAN(-Y/X)
      RETURN
      END
C-----------------------------------------------------------------------
C     SUBROUTINE HOM
C-----------------------------------------------------------------------
      SUBROUTINE HOM(REX,REY,NNOD,NWT,KOUT,IOUT,H1Y,ID)
C     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
C     THIS SUBROUTINE CALCULATES THE LIMITING NODAL POINT VALUES
C     OF THE ANALYTIC H1 APPROXIMATION FUNCTION
C
      COMMON/BLK 1/X(100)
      COMMON/BLK 2/Y(100)
      COMMON/BLK 7/ANGLE(100)
      DIMENSION H1X(100),H1Y(100)
      DIMENSION REX(100),REY(100)
C
C     MAIN LOOP
C
      KP=KOUT
      IF(ID .NE. 0)KOUT=9999
      DO 20 J=1,NNOD
      H1X(J)=0.
20    H1Y(J)=0.
      IF(KOUT .EQ. IOUT)WRITE(NWT,22)
22    FORMAT(//,10X,'CUBEN APPROXIMATION FUNCTION NODAL VALUES!',//,
     C6X,'NODE',6X,'STATE',14X,'STREAM',/,5X,'NUMBER',3X,'VARIABLE',
     C12X,'FUNCTION')
      DO 1000 J=1,NNOD
C
C.....CALCULATE BOUNDARY ELEMENT CONTRIBUTIONS
C
      DO 500 K=1,NNOD
      KK=K+1
      IF(KK.GT.NNOD)KK=1
      IF(K.EQ.J.OR.KK.EQ.J)GOTO 500
      CALL CAUCH1(J,K,KK,A,B,C,D)
      C1=REX(KK)*(X(J)-X(K))-REY(KK)*(Y(J)-Y(K))
     C-REX(K)*(X(J)-X(KK))+REY(K)*(Y(J)-Y(KK))
      C2=REX(KK)*(Y(J)-Y(K))+REY(KK)*(X(J)-X(K))
     C-REX(K)*(Y(J)-Y(KK))-REY(K)*(X(J)-X(KK))
      H1X(J)=H1X(J)+C1*C-C2*D
      H1Y(J)=H1Y(J)+C1*D+C2*C
500   CONTINUE
C
C.....CALCULATE PRINCIPLE VALUE CONTRIBUTIONS
C
      K=J-1
      IF(K.LT.1)K=NNOD
      KK=J+1
      IF(KK.GT.NNOD)KK=1
      XLN=SQRT((Y(KK)-Y(J))**2+(X(KK)-X(J))**2)
      XLM=SQRT((Y(K)-Y(J))**2+(X(K)-X(J))**2)
      XXX=XLN/XLM
      AJ=ALOG(XXX)
C     AJ=DLOG(XLN/XLM)
      BJ=(360.-ANGLE(J))/180.*3.141593
      H1X(J)=H1X(J)+REX(J)*AJ-REY(J)*BJ
      H1Y(J)=H1Y(J)+REX(J)*BJ+REY(J)*AJ
C
C     DIVIDE BY 2*PI*I
C
      TEMP=H1X(J)
      H1X(J)=H1Y(J)/6.28318
      H1Y(J)=-TEMP/6.28318
      IF(KOUT .EQ. IOUT)WRITE(NWT,450)J,H1X(J),H1Y(J)
450   FORMAT(3X,I5,5X,F10.4,10X,F10.4)
1000  CONTINUE
C
C     CALCULATE NODAL POINT RELATIVE ERROR
C
      IF(KOUT .NE. IOUT)GO TO 200
      WRITE(NWT,550)
550   FORMAT(///,10X,'NODAL POINT RELATIVE ERROR VALUES!')
      DO 2000 I=1,NNOD
      DA=REX(I)-H1X(I)
      DB=REY(I)-H1Y(I)
      WRITE(NWT,450)I,DA,DB
2000  CONTINUE
200   IF(KOUT .EQ. 9999)KOUT=KP
      RETURN
      END
C-----------------------------------------------------------------------
C  SUBROUTINE ANGLE
C-----------------------------------------------------------------------
C
      SUBROUTINE ANG(NNOD)
C
      COMMON/BLK 1/X(100)
      COMMON/BLK 2/Y(100)
      COMMON/BLK 7/ANGLE(100)
C
C  THIS SUBROUTINE CALCULATES THE ANGLE BETWEEN EACH NODAL POINT
C
      PI=ACOS(-1.)
      DO 100 I=1,NNOD
      J=I-1
      JJ=I+1
      IF(J.EQ.0)J=NNOD
      IF(JJ.GT.NNOD)JJ=1
      XJ=X(J)-X(I)
      XJJ=X(JJ)-X(I)
      YJ=Y(J)-Y(I)
      YJJ=Y(JJ)-Y(I)
      CALL CAUCH5(XJJ,YJJ,AJJ)
      CALL CAUCH5(XJ,YJ,AJ)
      ANGLE(I)=(AJ-AJJ)*180./PI
      IF(ANGLE(I).LT.0.)ANGLE(I)=ANGLE(I)+360.
100   CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
C  SUBROUTINE FRONT
C-----------------------------------------------------------------------
      SUBROUTINE FRONT(NNOD,NFRS,NFRE,COND,XLAT,POR,DELT,H1Y,KODD)
C
C  THIS SUBROUTINE CALCUALTES THE NEW INTERNAL ANGLES AND NEW POSITIONS
C  OF THE FREEZING OR THAWING FRONT AFTER EACH TIME INCREMENT BY
C  SHIFTING THE POSITIONS VERTICALLY OR NORMALLY.
C
      COMMON/BLK 1/X(100)
      COMMON/BLK 2/Y(100)
      COMMON/BLK 7/ANGLE(100)
      DIMENSION Q(50),XP(50),YP(50)
      DIMENSION H1Y(100)
C
      DO 50 I=1,50
50    Q(I)=0.
C...APPROXIMATE THE EFFLUX ALONG THE FREEZING FRONT
      J=0
      DO 100 I=NFRS,NFRE-1
      J=J+1
      XX=X(I+1)-X(I)
      YY=Y(I+1)-Y(I)
      DIS=SQRT(XX*XX+YY*YY)
      FLUX=.5*COND*(H1Y(I+1)-H1Y(I))/DIS
      Q(J)=Q(J)+FLUX
      Q(J+1)=Q(J+1)+FLUX
100   CONTINUE
C
C  UPDATE THE NEW FREEZING FRONT
C
      J=0
      DO 200 I=NFRS,NFRE
      J=J+1
C...DETERMINE THE NORMAL DIRECTION
      IP1=I+1
      IF(IP1 .GT. NNOD)IP1=1
      IM1=I-1
      IF(IM1 .LT. 1)IM1=NNOD
      PI=ACOS(-1.)
      XJ=X(IM1)-X(I)
      YJ=Y(IM1)-Y(I)
      CALL CAUCH5(XJ,YJ,AJ)
C...CALCULATE THE NEW FREEZING FRONT
      DELS=Q(J)*DELT/(XLAT*POR)
      IF(I.EQ.NFRS .OR. I.EQ.NFRE)GO TO 250
      ANGL=.5*(360-ANGLE(I))*PI/180.+AJ
      IF(KODE .EQ. 2)GO TO 220
      XP(I)=X(I)
      YP(I)=Y(I)-DELS
      GO TO 200
220   XP(I)=X(I)+DELS*COS(ANGL)
      YP(I)=Y(I)+DELS*SIN(ANGL)
      GO TO 200
250   YP(I)=Y(I)-DELS*2.
      XP(I)=X(I)
200   CONTINUE
      DO 300 I=NFRS,NFRE
      X(I)=XP(I)
      Y(I)=YP(I)
300   CONTINUE
C...CALCULATE THE NEW INTERNAL ANGELS
      CALL ANG(NNOD)
      RETURN
      END
C
```