Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Physics, Computer Science & Mathematics Division

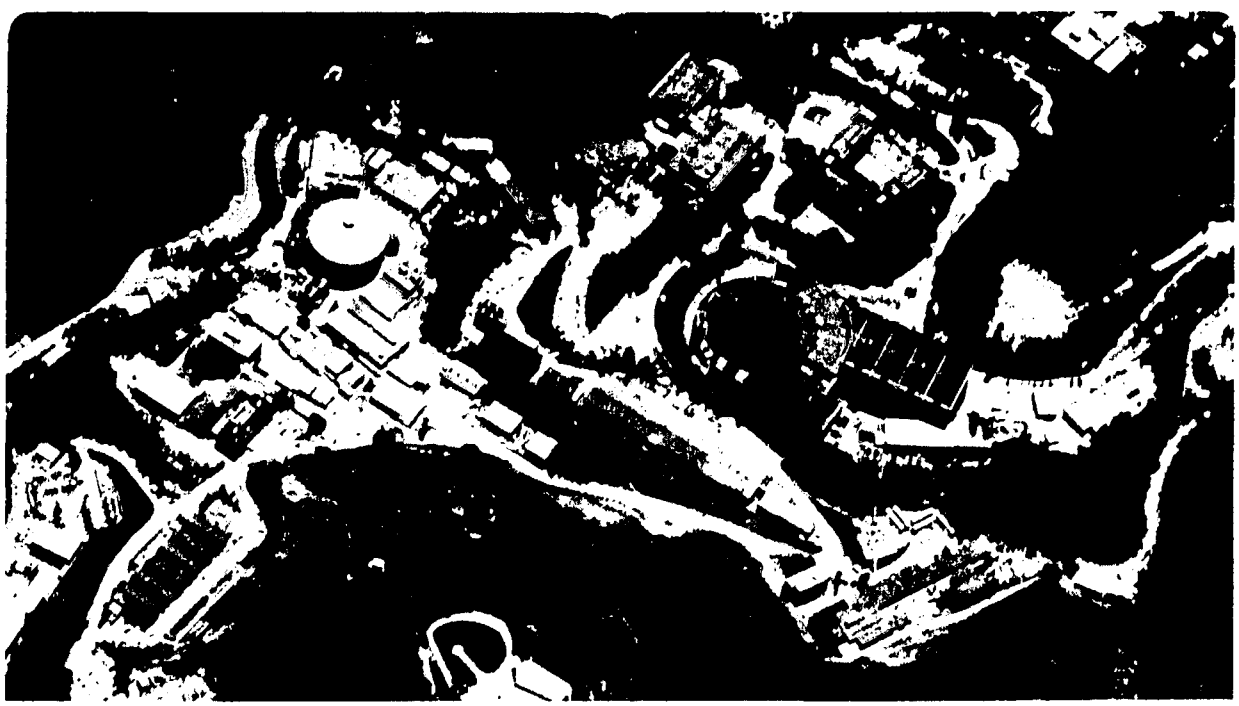
A PROGRAM USING FRONT TRACKING IN THE SOLUTION OF SCALAR HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS

Per Lötstedt

October 1981

RECEIVED
LAWRENCE
BERKELEY LABORATORY
APR 1 1982
LIBRARY AND
DOCUMENTS SECTION

For Reference
Not to be taken from this room



LBID-439 c.1

Prepared for the U.S. Department of Energy under Contract W-7405-ENG-48

DISCLAIMER

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

LBID-439

A PROGRAM USING FRONT TRACKING IN THE SOLUTION OF SCALAR HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS

Per Lötstedt

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

October 1981

This work was supported in part by the Director,
Office of Energy Research, Office of Basic Energy Sciences,
Engineering, Mathematical, and Geosciences Division
of the U.S. Department of Energy under Contract No. W-7405-ENG-48.

A Program Using Front Tracking in the Solution of Scalar Hyperbolic Partial Differential Equations

Per Lötstedt

Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Abstract

An interactive FORTRAN program is described which can follow discontinuity curves in the solution of a scalar hyperbolic partial differential equation. A flow chart of the main program is enclosed and the principal subroutines are documented. In three sessions the interaction between the user and the program is exemplified.

Introduction

We shall give a description of the interactive FORTRAN program that was used to obtain the results in [1]. The main feature of the program is that it is capable of following a discontinuity curve in the solution of scalar hyperbolic partial differential equations. The front tracking technique of the program is developed in [1]. The principle idea is to represent the front by the fraction of fluid F, $0 \leqslant F \leqslant 1$, behind the front in each mesh cell. Consequently, the fraction 1-F is ahead of the front in the same cell. The front is advanced in time by moving the fractions in the mesh with a speed determined by the Rankine-Hugoniot condition. The continuous parts of the solution of the differential equation are solved approximately by Glimm's method or a finite difference scheme based on the upstream difference formula and Godunov's method. The two methods are implemented in two separate subroutines.

The saturation in two-phase porous flow satisfies a scalar hyperbolic PDE. The velocity field is derived from the solution of an elliptic PDE. By calling different subroutines the user has the option to let the velocity field be static (as in Burgers' equation) or time-dependent (as in the two-phase porous flow equations). For a detailed account of the front tracking method and the solution of the differential equations the reader is referred to [1].

First, the major subroutines are described briefly. Then a flow chart of the main program is presented and finally examples are given how to run the program on VAX under VMS. For convenience, the notation here is identical to the notation in [1].

The Subroutines

The solution u is stored in U, the velocity field (q_x, q_y) in QX and QY and the fractions of fluid F for each mesh cell in the matrix F. U, QX, QY and F are two-dimensional arrays.

A short description of some of the subroutines in alphabetical order follows here. The comments in the program listing provide further details.

ACCCG

A solution to the discretized elliptic equation is computed by an accelerated conjugate gradient method. The preconditioning matrix is given by the discrete five-point Laplace operator as described in [2]. Possible error messages are due to PWSCRT, a subroutine from the NCAR-package [3], or too many iterations in the iterative step.

BSWEEP

The points of discontinuity along each grid line in the x- or y-direction are computed and stored in the data structure depicted in fig. 1. The number of discontinuities on line i is NRCSS(i). If NRCSS(i) $\neq$ 0 then NENTR(i) points to the first entry of the line in the tables XYCRS and XYJUMP. Let n satisfy $1 \leqslant n \leqslant$ NRCSS(i) and m = NENTR(i) + n - 1. Then XYCRS(m) *h is the coordinate of the n:th discontinuity along the i:th line and XYJUMP(m) contains its character, i.e. + 1 or - 1.

COMPQ (in file COMPQ)

A stationary velocity field (q_x, q_y) is read from a file. Burgers' equation and a special case of the Buckley-Leverett equation where $\lambda(u)$ = const. have time-independent q_x and q_y.

COMPQ (in file COMPQ1)

The solution p to the elliptic part of the two-phase porous flow equations is determined for a given saturation u by the routine ACCCG. Then the velocity field (q_x, q_y) is calculated by a finite difference formula.

COORD

As input the subroutine requires the points of discontinuity on each grid line in the x- and y-directions as produced by two calls of BSWEEP in the two coordinate directions. The result is NCR coordinate pairs (X(i), Y(i)) where the front curve intersects the grid lines. The idea of the algorithm is briefly as follows. Suppose that a point of discontinuity has been found on one side of a mesh cell. Then search the other three sides for another point of discontinuity. If there is more than one point then choose the nearest point. Mark this point to avoid visiting a point more than once. Continue the search

in the mesh cell which has the side where the last point was found in common with the current cell. If the boundary is found or a closed loop has been completed then stop.

FCOMP

The new fraction of black fluid F at time t_{n+1} in each mesh cell in the i:th row is computed using the old value at t_n, the speed of the front and the information in ICODE from ICDCMX. If F = 0 in a mesh cell then the cell is ignored. When $0 < F < 1$ the slope of the interface is first calculated according to the method in 1 . Then the speed v of the front at the closest discontinuity on a grid line is determined by VELCTY. If a rarefaction wave starts in the mesh cell then a special flag for the cell is set. The speed v is used to move the fraction of the cell. The steps above are simplified if F = 1. The new values of F in the i:th row are stored in FNEW.

FSWEEP

The fractions of fluid in the mesh cells are transported in one coordinate direction following the description in [1]. For each mesh cell in each row the configuration of the fractions in the eight surrounding cells determines a code for the cell in the center. This code is calculated by ICDCMX and stored in ICODE. Based on ICODE, Z, the velocity field (q_x, q_y), and the previous fractions in F at t_n the new fractions at t_{n+1} are computed by FCOMP.

GCOMP

The value of $\lambda(u)$ on mesh lines between mesh points in the x- and y-directions is computed and stored in GX and GY, respectively. The function ALAMBD computes the value $u^2 + \alpha(1 - u)^2$ for the Buckley-Leverett equation and the value $(u + \beta(1 - u))^4$ for the miscible two-phase porous flow.

ICDCMP

Each mesh cell in the i:th row is given a number depending on the fraction of black fluid in the cell and its four immediate neighbors, see fig. 2. The code is stored in the section ICODE for later use by LINTAB.

ICDCMX

Each mesh cell in the i:th row is given a code depending on the fraction of black fluid in the cell and its eight surrounding neighbors, see fig. 3. The code is stored in ICODE for later use by FCOMP.

INTEGR

The hyperbolic PDE is integrated one time-step on the k:th mesh line in the x- or y-direction. The position of the NDISCØ discontinuities on the line before the last advection of the fractions by FSWEEP

is FROLD(i) * h, i = 1,2...NDISCØ. The vector FRNEW holds the positions of the NDISC1 discontinuities after the last call of FSWEEP. NDISCØ may differ from NDISC1. The integration of u at the mesh points on the line commences at the left boundary and continues to the right for all points to the left of the location of the first point of discontinuity in FROLD. The corresponding discontinuity in FRNEW is determined. If the old and the new discontinuity are separated by a mesh point then extrapolation as described in [1] is performed to obtain a correct value of u at the mesh point. Then the integration proceeds until the next discontinuity or the right boundary is reached. The subroutine INTEGR in the file GLIMC1 uses Glimm's method and the version in GLIMC2 uses a finite difference scheme. The new value of u is computed by URIEMN or UDIFF and UGODUN and is stored in the vector UNEW. The subroutine ZBRENT in the IMSL-library [5] is called by URIEMN and UGODUN. The solution procedure is different for points where a rarefaction wave starts at the front curve. The algorithm for finding the discontinuity in FRNEW corresponding to the discontinuity in FROLD is heuristic and may fail (although no error occurred in the examples in [1]). If a matching point of discontinuity is not found then either an error message is printed or positive values of u spill over the front curve to the area of the solution where u = 0.

LINTAB

The points of discontinuity on the i:th grid line in the x- or y-direction are computed using the information on the fractions in the two surrounding rows stored in ICODE and ICODEØ from ICDCMP. If (ICODE(j) = 2 or ICODEØ(j) = 2) and ICODE(j-1) = ICODEØ(j-1) = 1 then there is a discontinuity in the j:th cell. Similarly, if ICODE(j-1) = 0 and ICODE(j) = 3 and ICODEØ(j) $\neq$ 0 (or with the reverse roles of ICODE and ICODEØ) then there is a discontinuity in the j:th cell.

RFTABL

This subroutine determines in which mesh cells on the front curve a rarefaction wave starts. The decision is based on the velocity field (q_x, q_y) and the normal to the front curve defined by X and Y from COORD. If the component of (q_x, q_y) in the normal direction is less than -EPS1 in two adjacent cells then those cells are marked as rarefaction cells. The number of rarefaction cells ℓ in the i:th row in the x-direction is IYNRSQ(i). Suppose that $\ell > 0$ and let k = IYENTR(i). The variable k points to the first entry in the linked list NRSQY. NRSQY(k, 1) is the number of the rarefaction cell counted from the left in the row. If $\ell > 1$ then NRSQY(k, 2) points to the next entry of the row in the list, see fig. 4, otherwise NRSQY (k, 2) = -1. The corresponding arrays in the y-direction are IXNRSQ, IXENTR and NRSQX. This information is utilized by FSWEEP and USWEEP.

SETINI

The values of U and F are initialized at t = 0. The files STDTxy (xy = B1, B2..BC,...Q1,...12,22) contain the code for several different initial conditions for Burgers' equation, the Buckley-Leverett equation and the miscible two-phase porous flow equation. In the table below the

examples in [1] and the file with the version of SETINI that generates the initial data for the example are listed.

Example in [1].	File.
Fig. 5 (Burgers' eq.)	STDTB7
Fig. 6 (Burgers' eq.)	STDTB3
Fig. 7 (Burgers' eq.)	STDTQ1
Fig. 10 (Burgers' eq.)	STDTB6
Fig. 12 (Buckley-Leverett)	STDT12
Fig. 19 (miscible flow)	STDT22

The subroutine ZXGSN in the IMSL-library [5] is called by SETINI for the Buckley-Leverett equation.

USWEEP

The solution u is advanced one fractional time-step by a sweep in the x- or y-direction. For a complete time-step k from t_n to t_{n+1}, two calls of USWEEP are necessary, one in the x-direction and one in the y-direction. USWEEP calls INTEGR for the integration along each mesh line.

VELCTY

Given the coordinates (i, j) of a mesh cell as input the subroutine finds the nearest discontinuity on a mesh line and calculates the front speed by the Rankine-Hugoniot condition. This speed is used by FCOMP to advect the fractions of fluid in the mesh.

The Main Program

The main program contains one major loop which is repeated at each time-step. The code for output of the profiles of the front, the velocity field (q_x, q_y) and the solution u on a Tektronics graphical terminal is also included in the main program. The basic plotting routines are provided by the graphical subroutine package GRAFPAC [4].

A picture of the information flow and the order of the calls of the subroutines is displayed in fig. 5. Each box corresponds to a call of a subroutine. Some of the input arrays to a subroutine are indicated to the left of the box. The arrays containing the result after the return from the subroutine are written to right of the box. BØ is the set of vectors NCRSSØ, NENTRØ, XYCRSØ and XYJMPØ. B1 is the set NCRSS1, NENTR1, XYCRS1 and XYJMP1. R is the set of vectors IXNRSQ, IXENTR, IYNRSQ, IYENTR and the matrices NRSQX and NRSQY.

Session Examples

We shall give three examples of sessions with the program. In the first example Burgers' equation is solved with the initial data from fig. 6 in [1]. In the second example the initialization is made for solution of the equation for miscible two-phase porous flow as in fig. 19 in [1]. Finally, the Buckley-Leverett equation with the same initial data as in fig. 12 in [1] is solved. The program is interactive and is monitored by the user in front of the terminal. The program has been run on a VAX11/780 computer under the operating system VMS connected to a Tektronix 4014 terminal.

The command files for linking the subroutines together are typed in fig. 6. The command file for solving Burgers' equation is LBURG.COM, the file for the Buckley-Leverett equation with static velocity field is LBUCK1.COM and with dynamic velocity field LBUCK2.COM. The file for the miscible two-phase flow is LMISCL.COM.

The program asks questions and the user gives the answers. All answers are sent to the computer by typing an appropriate number and then pressing the "return" button. By replying only "return" the computer interprets the answer as 0.

In the initialization phase of the program the questions and the actions taken by the computer on the answers are:

Data from file.

Yes: The computer asks for the name of the file where the initial data for the problem is.

No: The initial data is provided by SETINI in the file STDTxy linked together with the program.

Give number of squares in each direction.

The dimensions of the arrays in the program are such that 40 is the maximal number. Then the mesh size is $h = 1/40$. In the test runs the values 10, 20 and 40 have been chosen.

Give alpha.

The variable alpha is the parameter α in $f(u)$ and $\lambda(u)$ in the Buckley-Leverett equation. Then the derivative of the initial values of u in the radial direction at the front is chosen. The program then prints the value of the inflection point (PINFL), max $f'(u)$ (SPMAX) and the value of u at the front.

Give mobility ratio.

The program asks for the mobility ratio M when the miscible two phase porous flow equation is solved.

Give dt.

A maximal value of the time-step dt is suggested for a stable computation.

Give z.

The parameter z is used in FSWEEP and FCOMP and its purpose is explained in [1].

Plot?

Yes: The front curves are drawn on a graphical terminal.

No: The solution is printed in numbers on the terminal.

Plot q_x, q_y?

Yes: The velocity field (q_x, q_y) is drawn on a graphical terminal.

No: No action.

Profiles?

If the answer to "Plot?" was "Yes" then this question is asked.

Yes: Only the front curves are drawn and the solution u is not printed. If the reply to the next question "Store profiles?" is "Yes" then the front curves are stored on a file specified by the user and are not displayed on the terminal.

No: The front curve is drawn and the non-zero values of u are printed.

Accuracy.

0: u is printed with two decimal places (except 1.0).

1: u is printed with one decimal place.

Select output?

If the answer to "Plot" was "No" then this question is asked.

Yes: The data printed out is specified by the user.

No: The standard data, the matrices F and U and the data concerning the front, is printed.

Number of steps.

Let i be the answer. Then the program takes i full time-steps before it stops and repeats this question.

Number of steps where printout starts.

Let the answer be j. Then the first drawn or written output is made at the j:th step.

Continue.

When a picture has been studied sufficiently then press the "return" button so that it can be erased for a new picture.

Number of invisible steps.

Suppose that the answer is n. Then the computed results from the next n time-steps will not appear on the terminal.

Save on file?

Yes: The current data in U and F is saved on a file supplied by the user. This data can be used later as initial data.

No: No action.

Continue?

Yes: Continue the session.

No: Finish the session. The CPU-time used is printed and the percentage of that time spent in the subroutines BSWEEP (b), COORD (c), FSWEEP (f), COMPQ (q), RFTABL (r) and USWEEP (u) is displayed.

In the first example, fig. 7, Burgers' equation is solved with the same initial data as in fig. 6 in [1]. The front curves from every second step are stored on the file PROFIL.DAT and are later drawn on the graphical terminal by running the auxiliary program DRAWPR, see fig. 11. The program is restarted with the last data from the previous session stored on SAVE1.DAT as initial data. Then u is printed and the front curve is drawn every 5:th step, figs. 8, 9, 10.

The initial dialogue with the program for solution of the miscible two-phase porous flow equation is shown in fig. 12. The matrices F and U will be printed on the terminal.

In the last example, beginning in fig. 12, the Buckley-Leverett equation is solved with the same initial data as in fig. 12 in [1]. The front curve and u, figs. 13, 15, 17, and the velocity field, figs. 14, 16, are displayed every 10:th time-step.

References

1. P. Lötstedt, A Front Tracking Method Applied to Burgers' Equation and Two-phase Porous Flow, report LBL-13302, Lawrence Berkeley Laboratory, Berkeley, CA, 1981.

2. N. Albright and P. Concus, On Calculating Flows with Sharp Fronts in a Porous Medium, report LBL-9286, Lawrence Berkeley Laboratory, Berkeley, CA, 1979.

3. P. Swartztrauber and R. Sweet, Efficient FORTRAN Subprograms for the Solution of Elliptic Partial Differential Equations, Tech. note NCAR-TN/IA-109, National Center for Atmospheric Research, Boulder, CO, 1975.

4. GRAFPAC manual, Computer Center, Lawrence Berkeley Laboratory, Berkeley, CA, 1980.

5. IMSL library manual, International Mathematical & Statistical Libraries Inc., Houston, TX, June 1980.

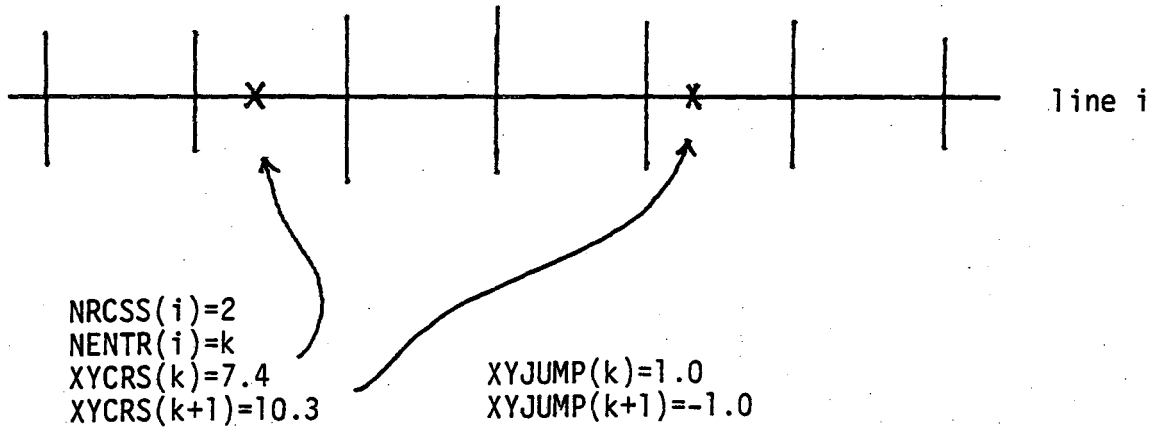


Fig. 1. The datastructure for line i created by BSWEEP. The crosses denote discontinuities on the line.

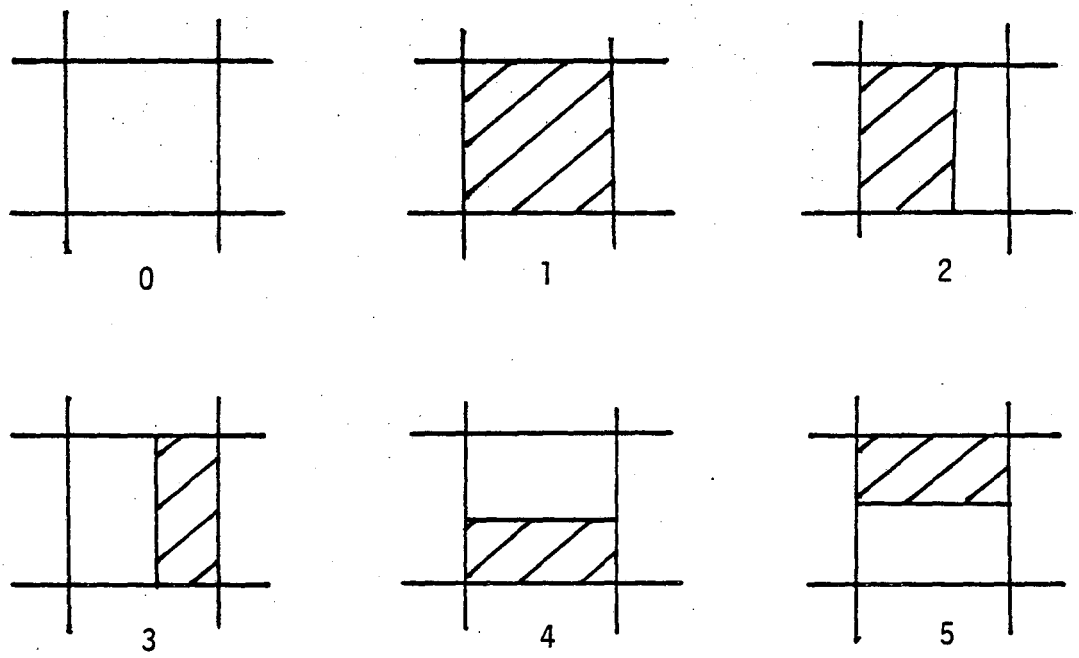


Fig. 2. The codes in ICODE from ICDCMP for each configuration of black and white fluid in a mesh cell.

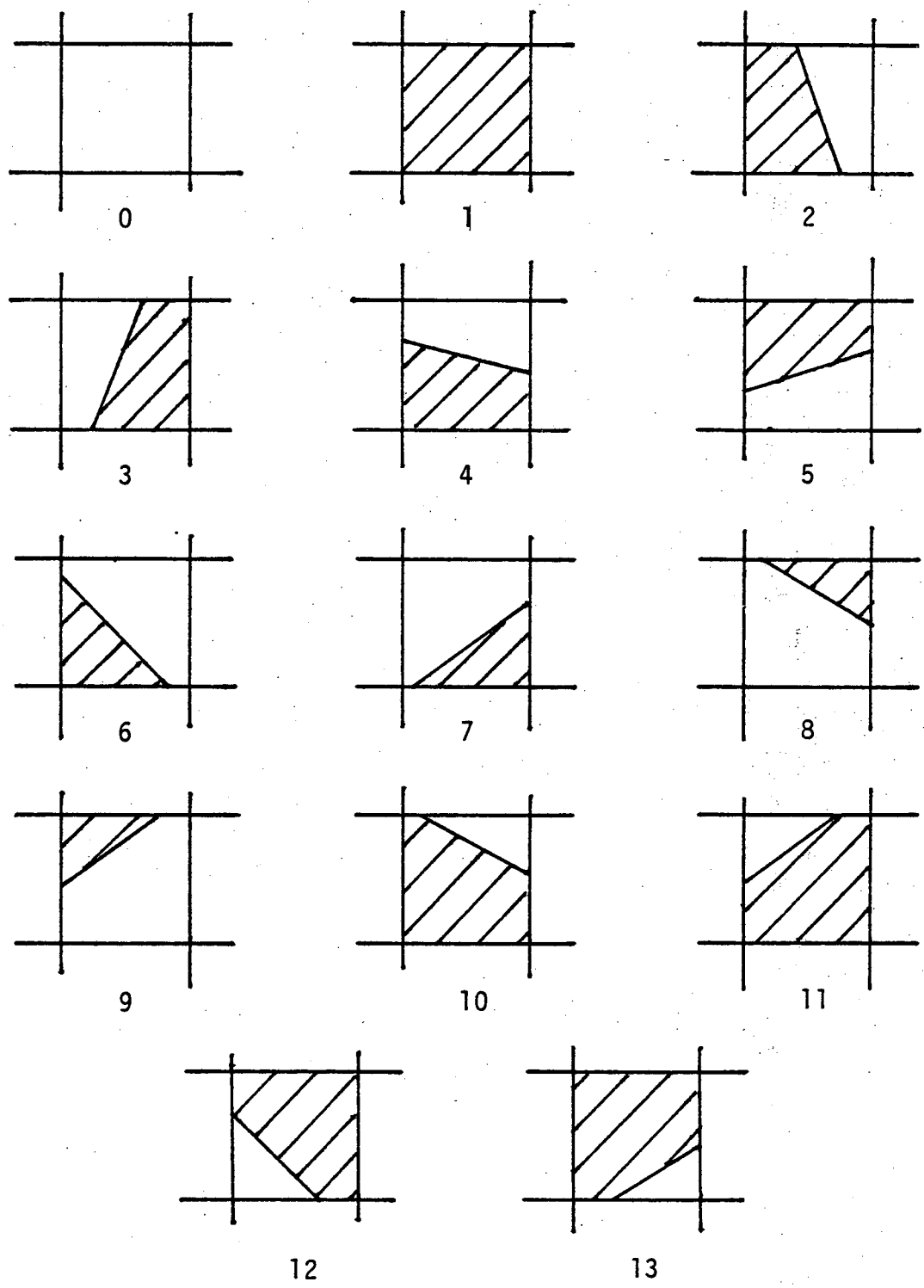


Fig. 3. The codes in ICODE from ICDCMX for each configuration of black and white fluid in a mesh cell.

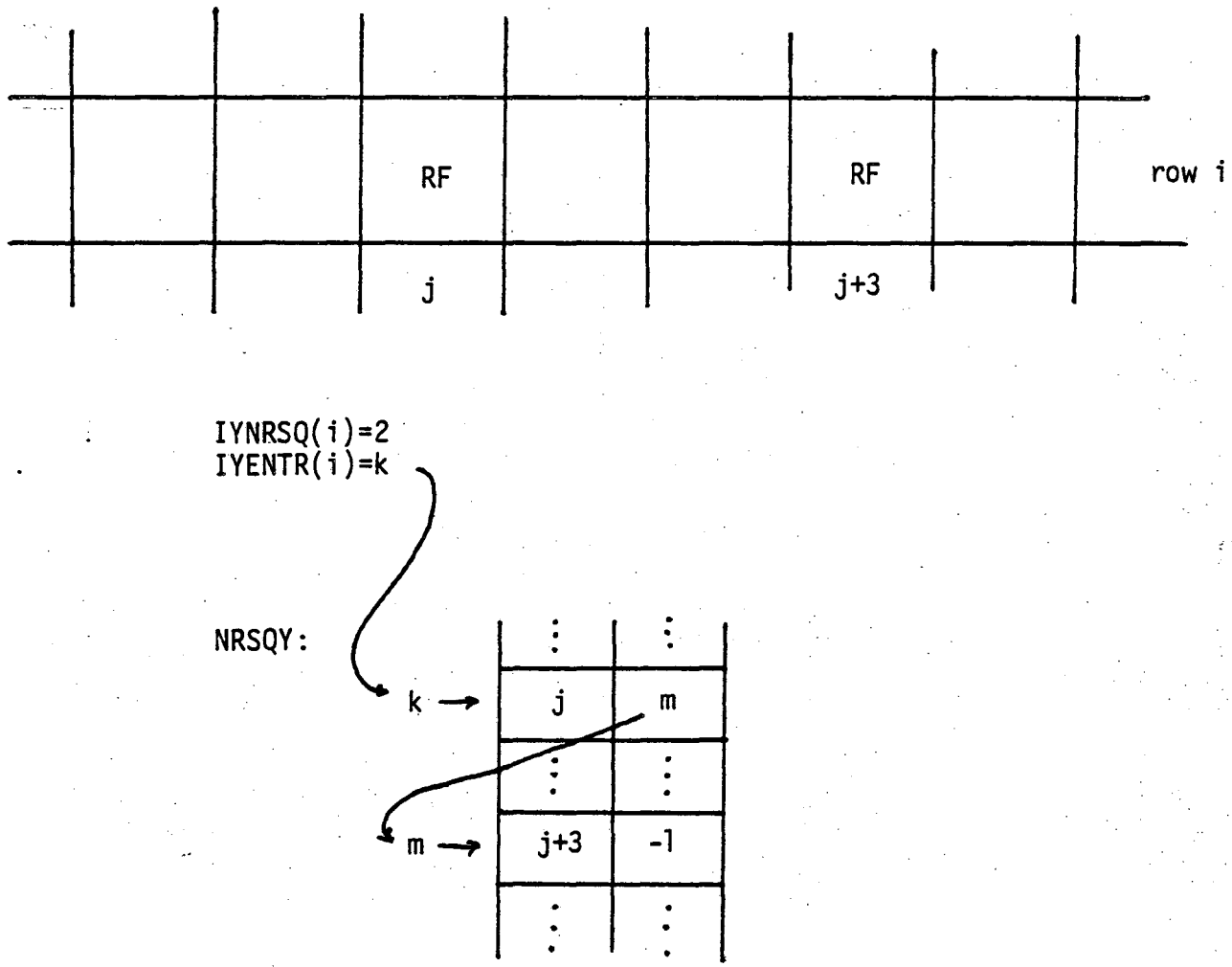


Fig. 4. The datastructure for the i:th horizontal row created by RFTABL. The cells marked by RF are the front cells where a rarefaction wave starts. There are two such cells in row i and their coordinates are (i,j) and (i,j+3).

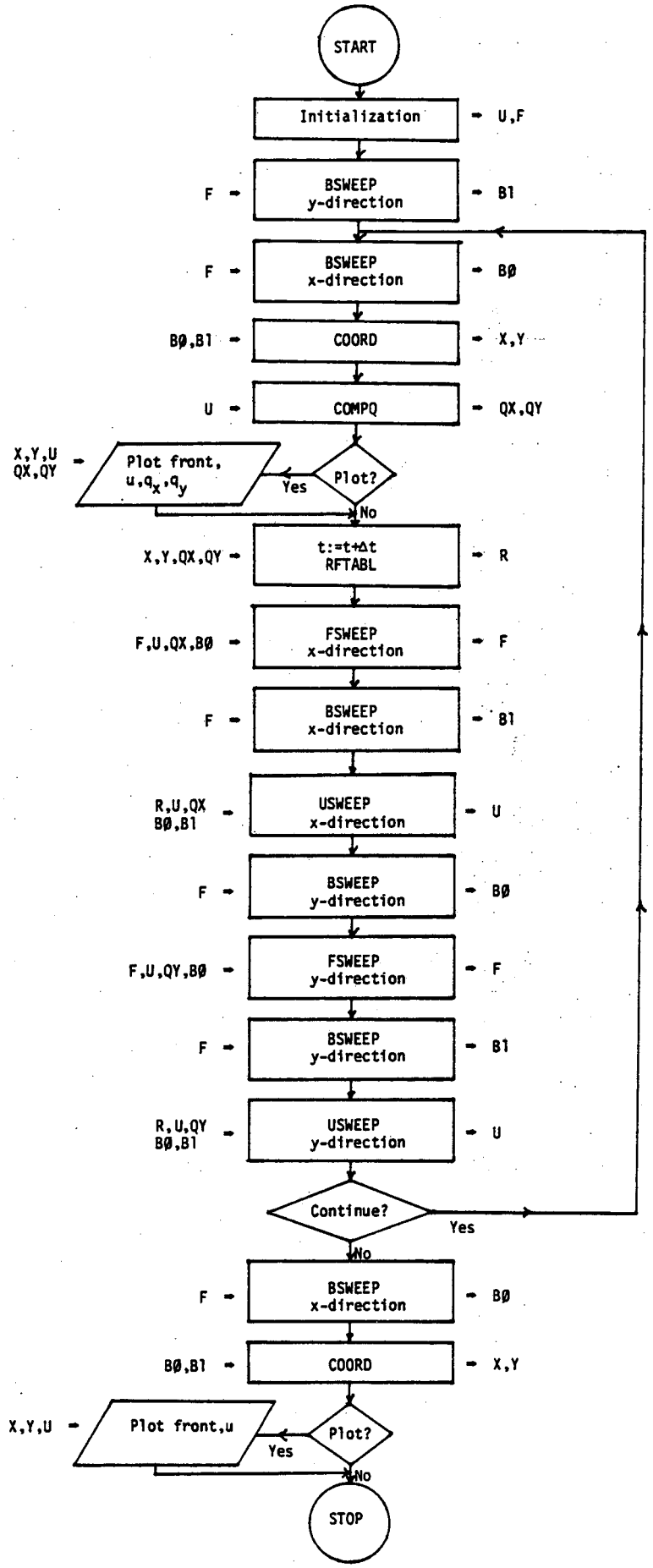


Fig. 5. Flow chart of the main program.

```
$ type lburg.com
lin burger,burout,bswees,fsween,usweep,compq, -
  glimc1,'p1',velcty,coord,rftabl,buriem, -
  dba0:[graphics.grafpac]tk,aid/lib
$ type lbuck1.com
lin burger,fsween,glimc2,burout,'p1',velcty, -
  bswees,usweep,compq,rftabl, -
  coord,blriem,dba0:[graphics.grafpac]tk,aid/lib, -
  dba0:[imsl]imslibs.olb/incl=zbrent/incl=zxgsn/lib
$ type lbuck2.com
lin burger,fsween,glimc2,burout,'p1',velcty, -
  bswees,usweep,compq1,lambd1,rftabl,pwscrt, -
  coord,blriem,dba0:[graphics.grafpac]tk,aid/lib, -
  dba0:[imsl]imslibs.olb/incl=zbrent/incl=zxgsn/lib
$ type lmiscl.com
lin burger,burout,bswees,fsween,usweep,compq1,lambd2, -
  glimc2,'p1',velcty,coord,rftabl,miriem,pwscrt, -
  dba0:[graphics.grafpac]tk,aid/lib
$
```

Fig. 6.

```
$ @lburg  stdtb3
$ r burger
Data from file (No=0,Yes=1):
Give number of squares in each direction (.le.40):40
Give dt (less than 0.0125):0.01
Give z:0.5
Plot? (Yes=1,No=0):1
Plot qx,qy? (Yes=1,No=0):
Profiles? (Yes=1,No=0):1
Store profiles?  (Yes=1,No=0):1
  Give filename:profil.dat
Accuracy ((2 dec pl)=0,(1 dec pl)=1):
Number of steps:10
Number of steps where printout starts:
Continue:
Number of invisible steps:1
Continue:
Number of invisible steps:1
Continue:
Number of invisible steps:1
Continue:
Number of invisible steps:1
Continue:
Number of invisible steps:1
Save on file? (No=0,Yes=1):1
Give filename:save1.dat
Continue?(No=0,Yes=1):
```

```
   TIMING

Percentage of     17.46 sec
     b      c      f      q      r      u
  22.0    1.7   31.2   27.6    0.7   16.8
$
```

```
$ r burger
Data from file (No=0,Yes=1): 1
Give filename:save1.dat
Give z:0.5
Plot? (Yes=1,No=0):1
Plot qx,qy? (Yes=1,No=0):
Profiles? (Yes=1,No=0):
Accuracy ((2 dec pl)=0,(1 dec pl)=1):
Number of steps:10
Number of steps where printout starts:
```

Fig. 7.

Continue:
Number of invisible steps:4

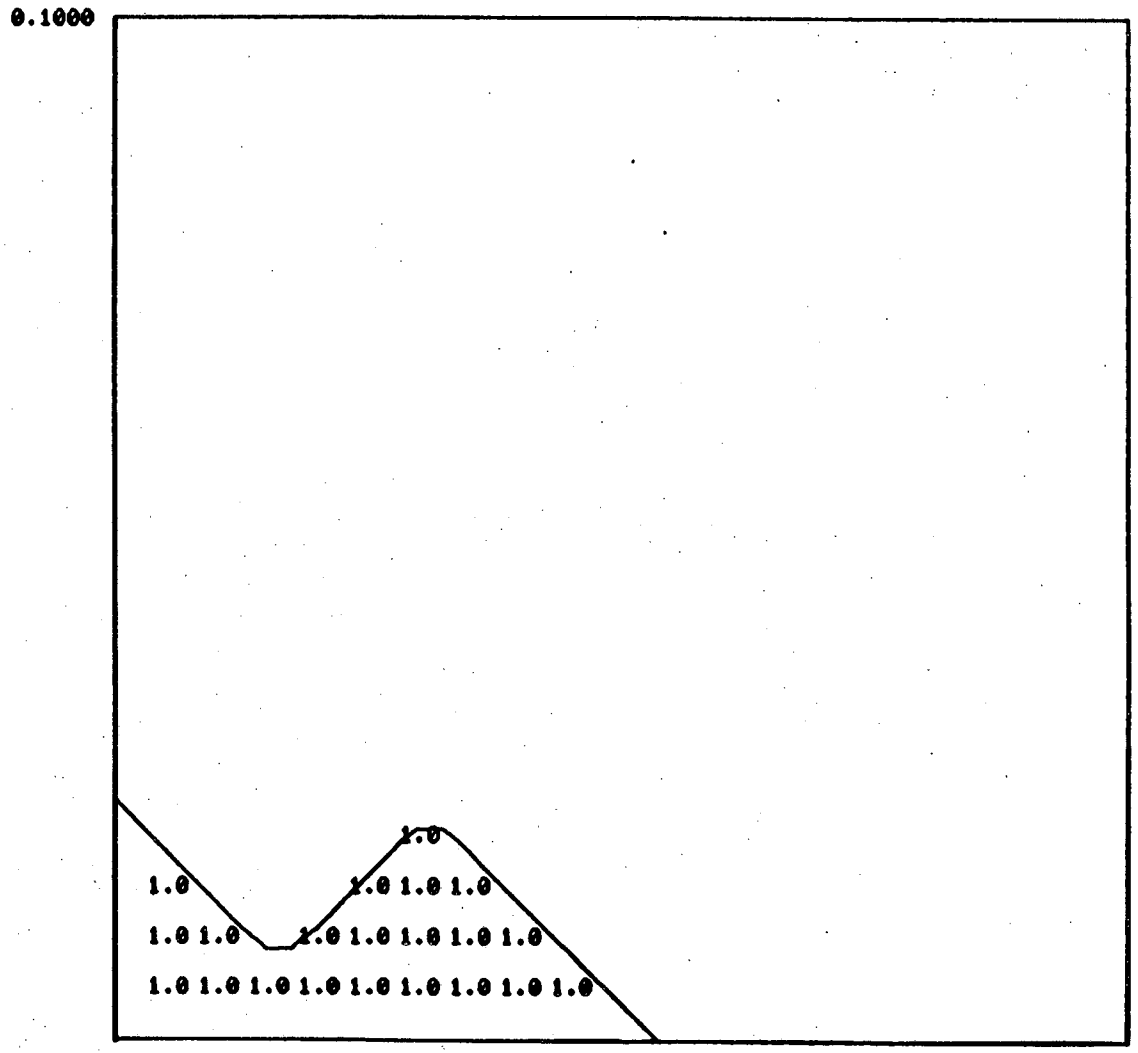


Fig. 8.

```
Continue:
Number of invisible steps:4
Save on file? (No=0,Yes=1):
Continue?(No=0,Yes=1):
```

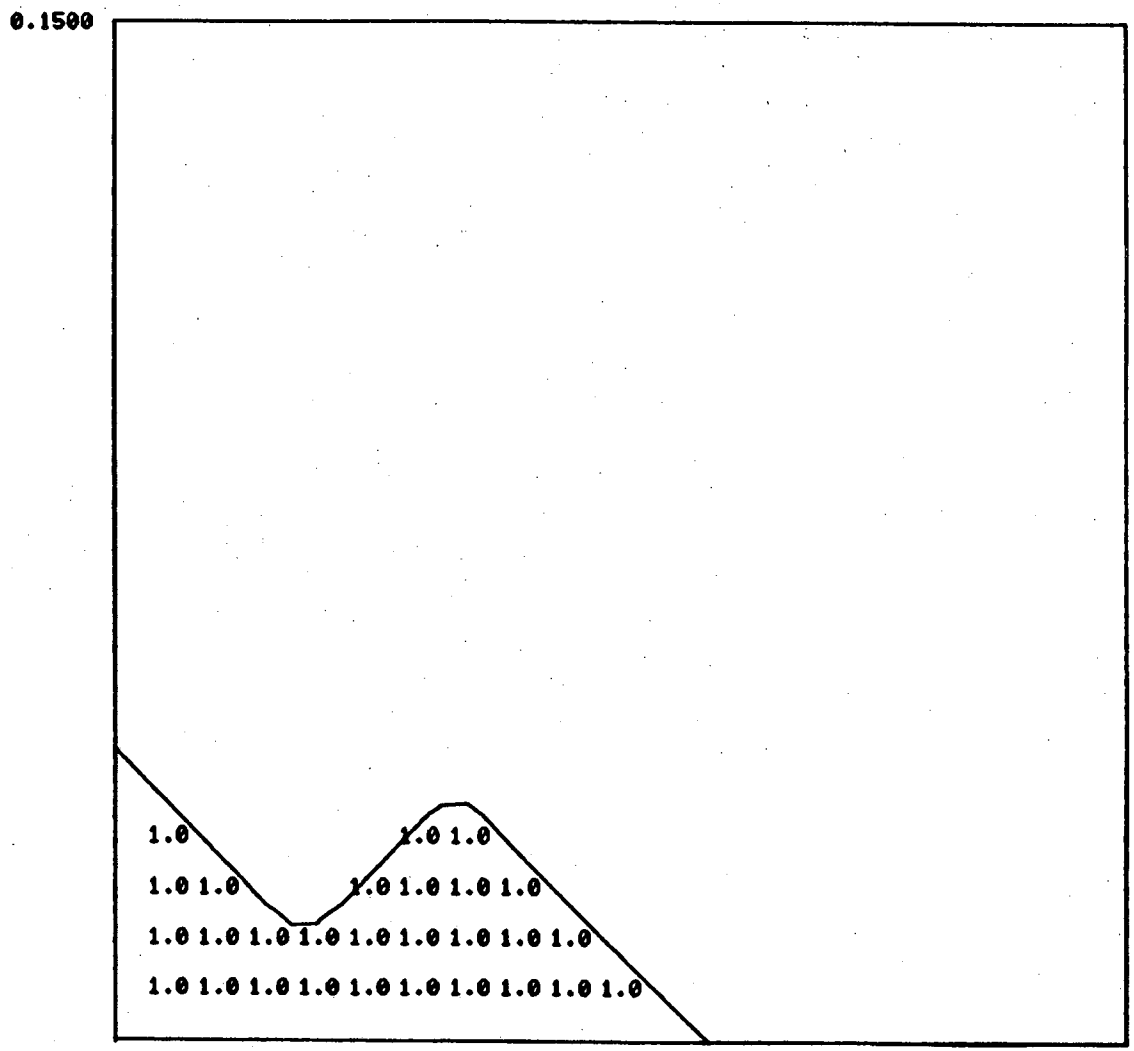


Fig. 9.

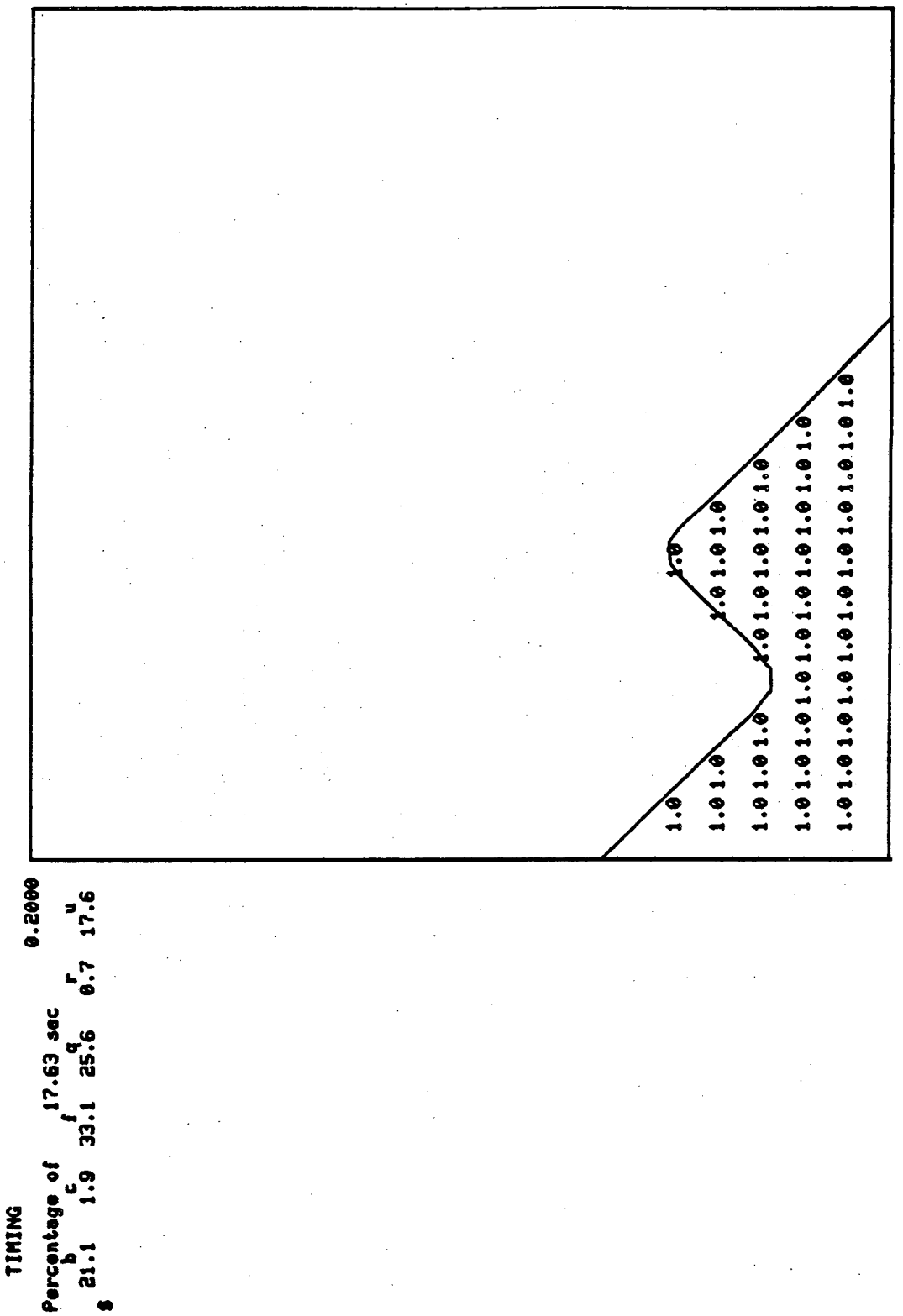
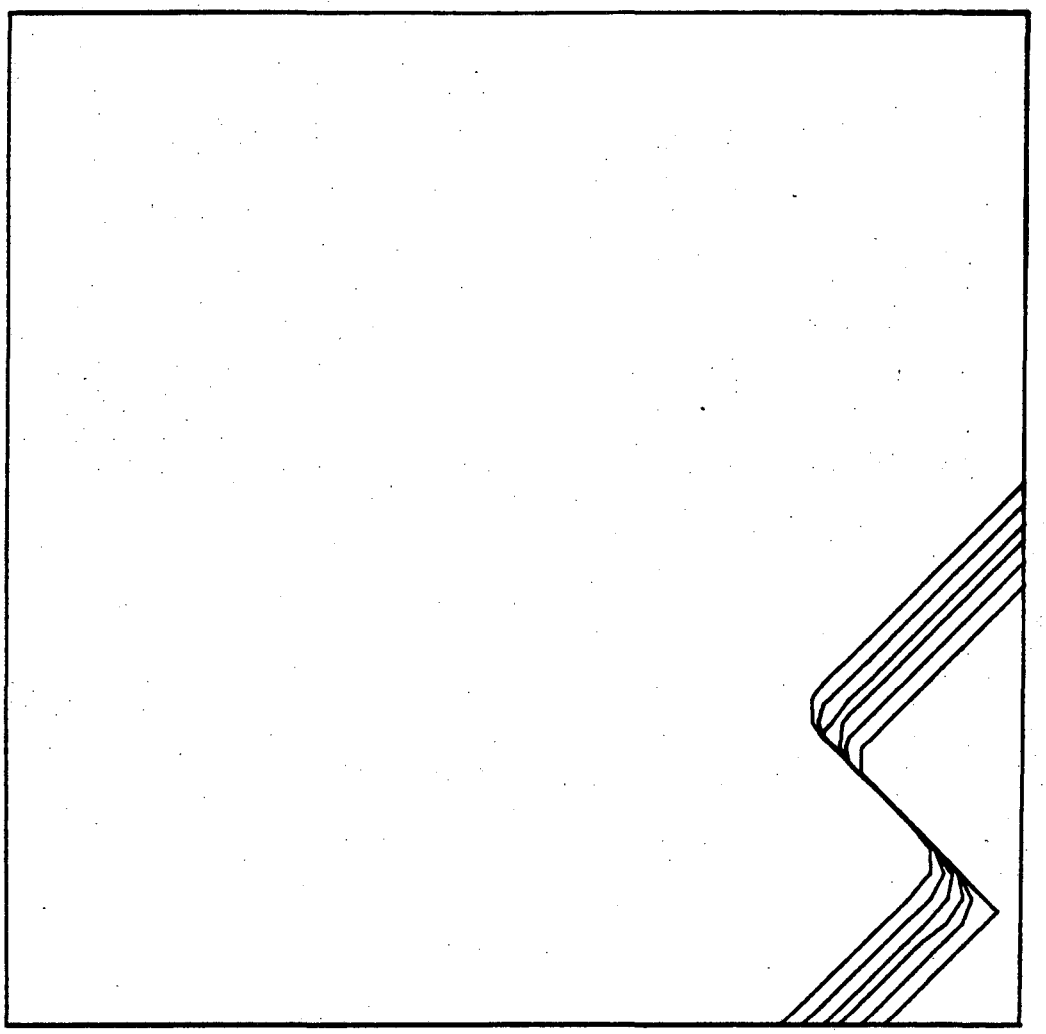

TIMING
Percentage of 17.63 sec
b c f q r u
s 21.1 1.9 33.1 25.6 0.7 17.6
0.2000

Fig. 10.

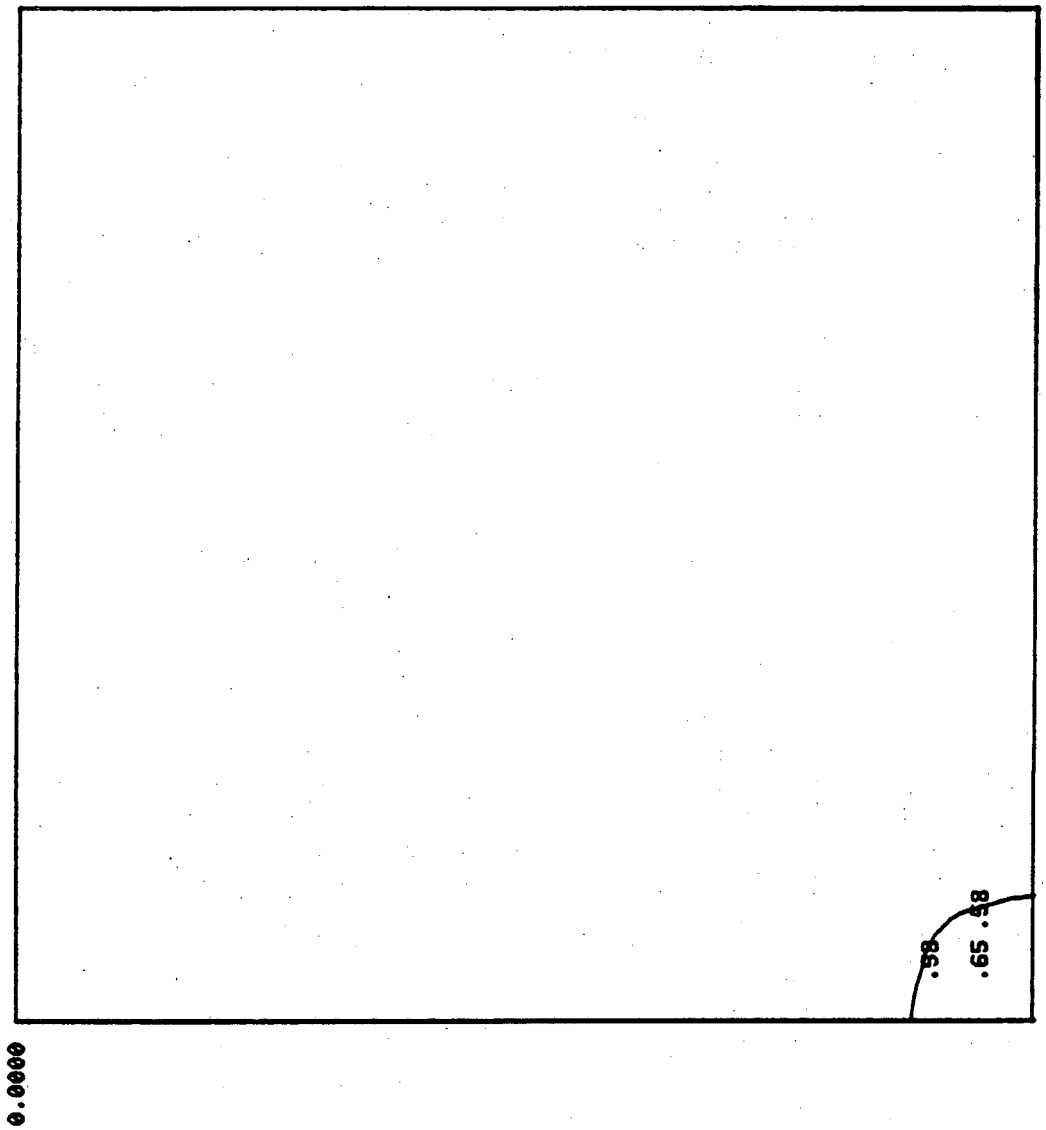


```
More files? (Yes=1,No=0):
FORTRAN STOP
$
```

Fig. 11.

```
$ @lmiscl stdt22
$ r burger
Data from file (No=0,Yes=1):
Give number of squares in each direction (.le.40):40
Give mobility ratio:2.0
Give dt (less than 0.0125):0.01
Give z:1.0
Plot? (Yes=1,No=0):
Plot qx,qy? (Yes=1,No=0):
Select output? (Yes=1,No=0):1
Output of: 1. F-matrix 2. U-matrix 3. Front-data (Yes=1,No=0)
 1 1 0

F-matrix   Time:   0.000

Start

 1.00 1.00 1.00 1.00 1.00 0.95 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 1.00 1.00 1.00 1.00 1.00 0.95 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 1.00 1.00 1.00 1.00 1.00 0.70 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 1.00 1.00 1.00 1.00 1.00 0.40 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 1.00 1.00 1.00 1.00 0.60 0.10 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 0.95 0.95 0.70 0.40 0.10 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00
 ^Y

 $


$ @lbuck2 stdt12
$ r burger
Data from file (No=0,Yes=1):
Give number of squares in each direction (.le.40):40
Give alpha:0.5
Slope for a0=0 :-3.522 New slope:0.0
ier=131
[a,pinfl,b]= 0.38709 0.38710 0.38711 spmax= 2.08079
ufront= 0.57735
Give dt (less than 0.0060):0.005
Give z:0.5
Plot? (Yes=1,No=0):1
Plot qx,qy? (Yes=1,No=0):1
Profiles? (Yes=1,No=0):
Accuracy ((2 dec pl)=0,(1 dec pl)=1):
Number of steps:10
Number of steps where printout starts:
```

Fig. 12.



Continue:

Fig. 13.

```
Number of invisible steps:9
Save on file? (No=0,Yes=1):
Continue?(No=0,Yes=1):1
Number of steps:10
Number of steps where printout starts:
```

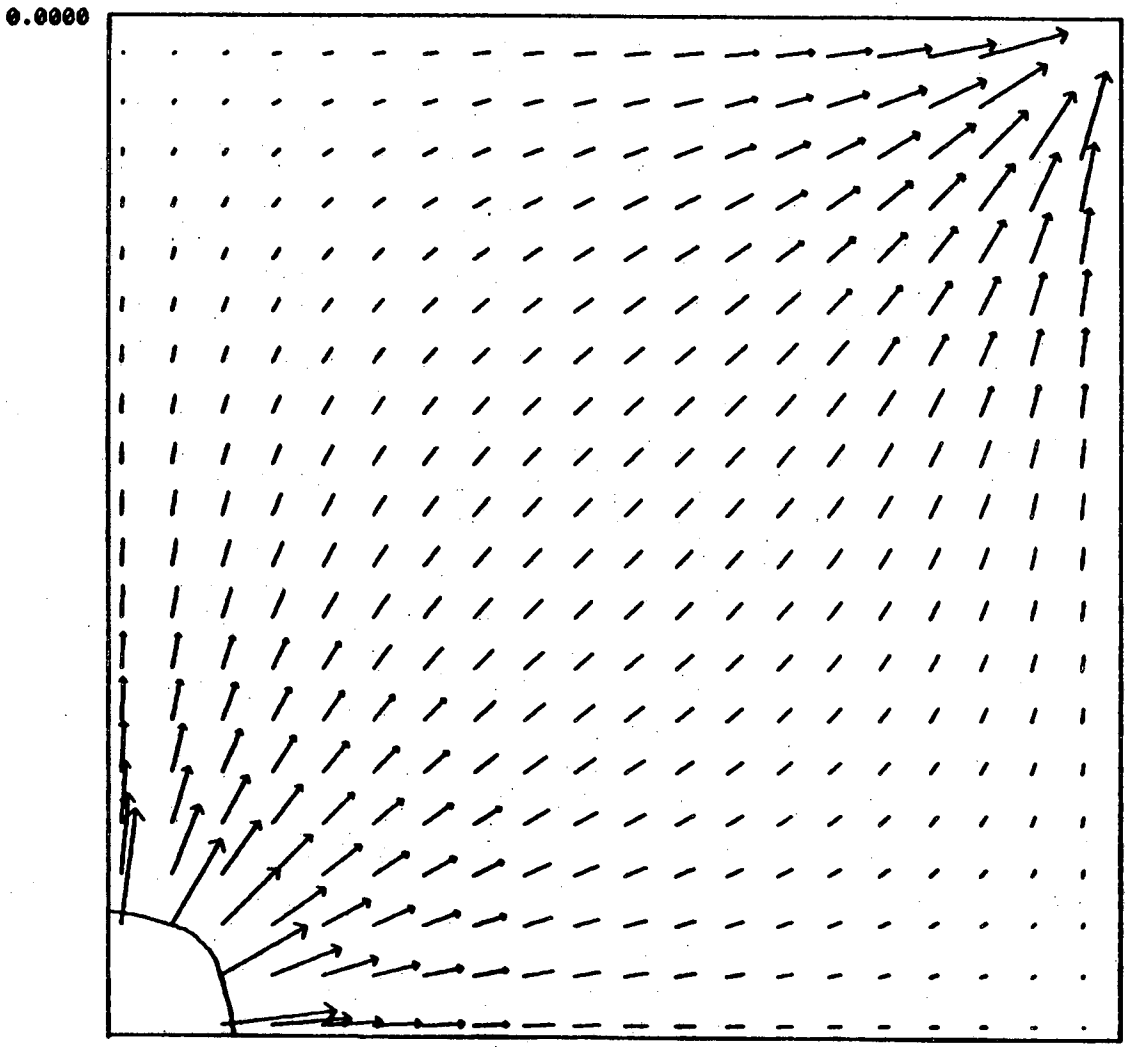
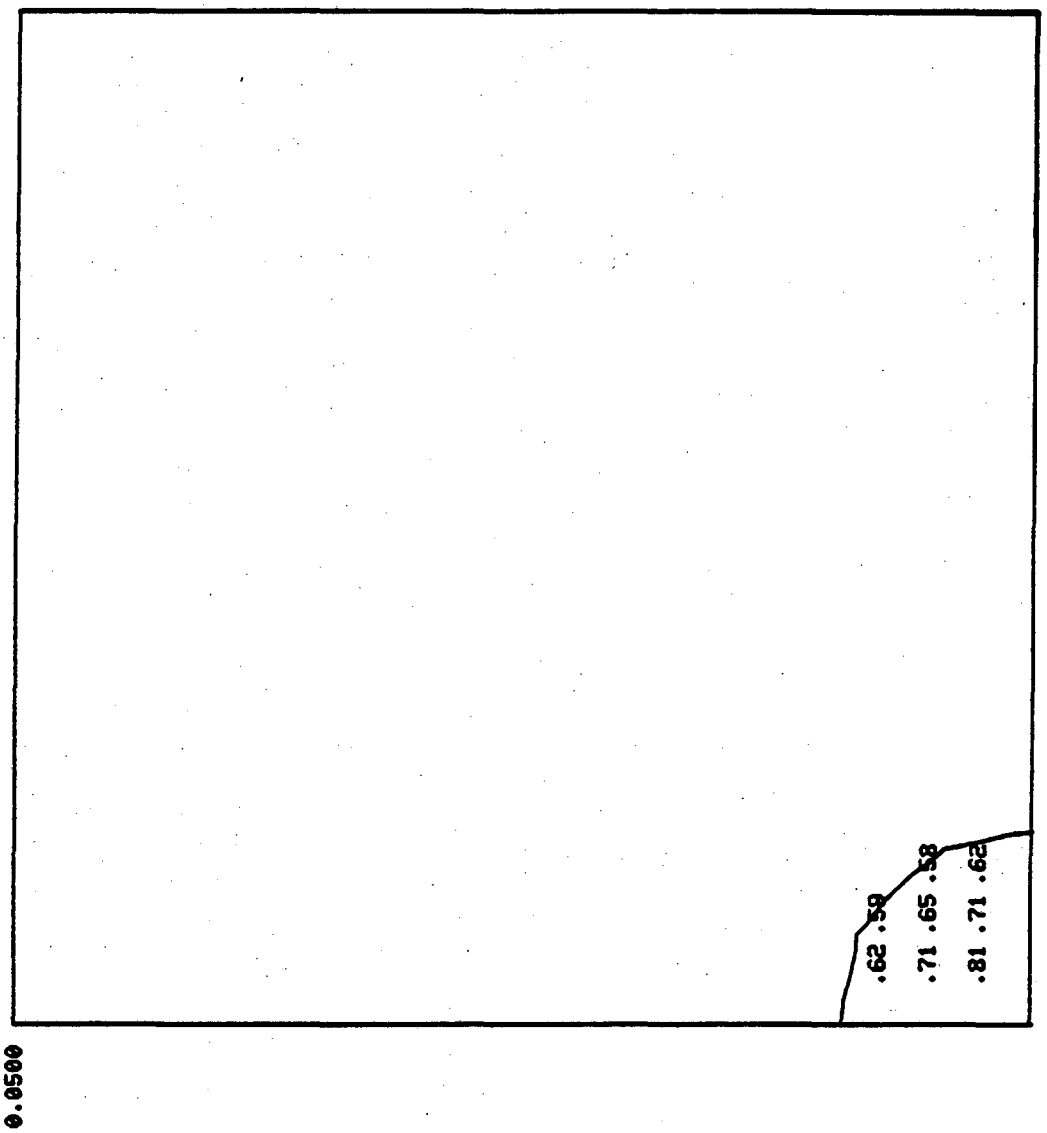
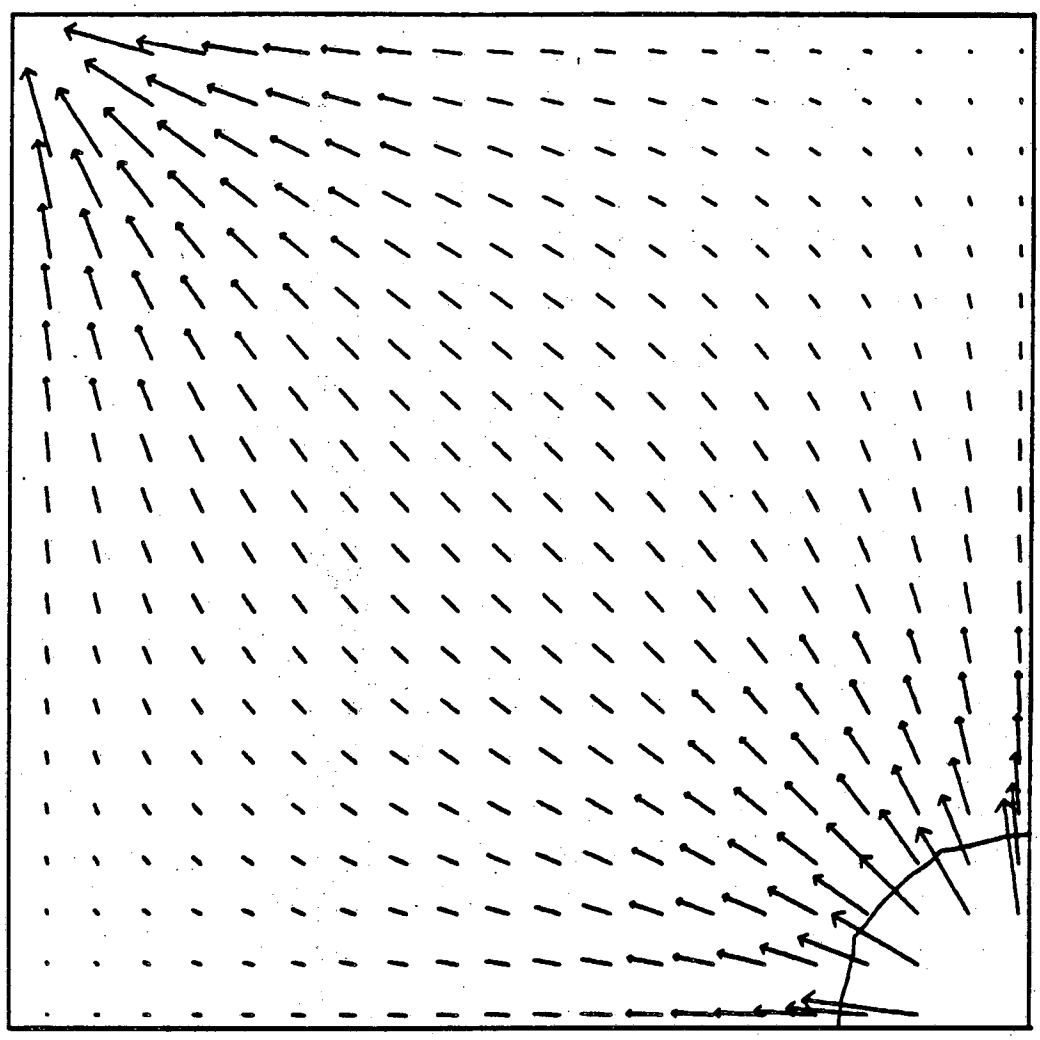


Fig. 14.



Continued

Fig. 15.



0.0500

Number of invisible steps:9
Save on file? (No=0,Yes=1):
Continue?(No=0,Yes=1):

Fig. 16.

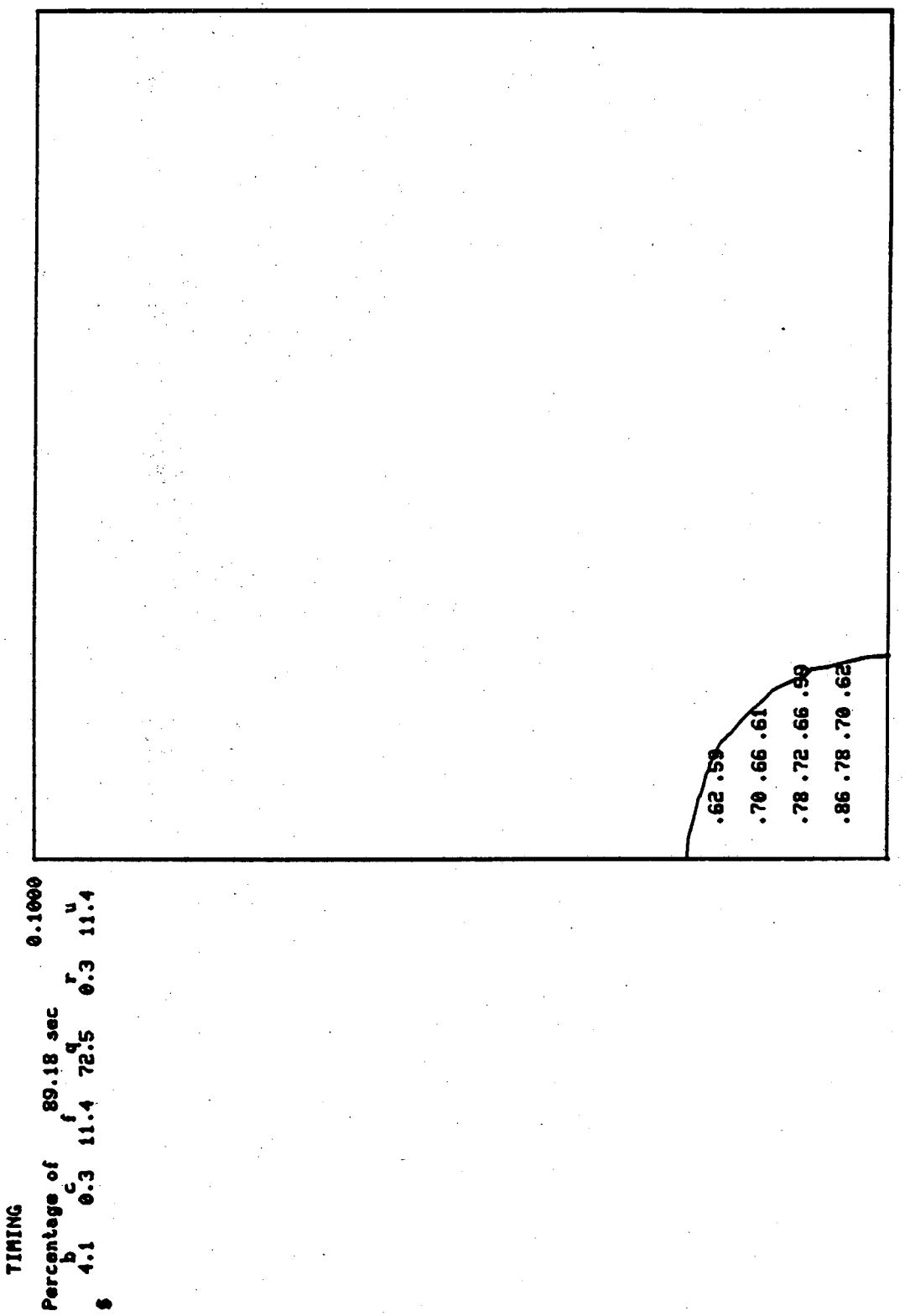

TIMING
Percentage of 89.18 sec
b c f q r u
4.1 0.3 11.4 72.5 0.3 11.4
$
0.1000
.62 .59
.70 .66 .61
.78 .72 .66 .99
.86 .78 .70 .62

Fig. 17.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720