SKi
STATENS KÄRNKRAFTINSPEKTION
Swedish Nuclear Power Inspectorate

Datum - Date
1987-03-24
Ert datum - Your date

Vår ref - Our ref
HYDROCOIN (87)4
Er ref - Your ref

HYDROCOIN Project Secretariat

To HYDROCOIN Participants

Sixth HYDROCOIN workshop

The sixth HYDROCOIN workshop and Coordinating Group meeting will be held in Washington D.C. on May 26-29, 1987. The main part of the workshop will be devoted to presentation and discussion on level 2 and level 3 results. There may also be invited talks on uncertainty and sensitivity analysis. Attached to this letter there is a participation form. For the planning of the workshop program the HYDROCOIN participants are kindly requested to fill in this form and send it to the Secretariat before April 25.

The Washington meeting is organised jointly by the U.S. DOE and U.S. NRC in cooperation with the Project Secretariat. Further information on the meeting concerning hotel accomodation, field trip etc. will shortly be distributed.

Complementary specifications on levels 2 and 3

Attached to this letter you find the following complements to two case specifications which have been submitted by Paul Davis:

1. A revision of the output format for level 2, case 4.
2. An addendum to level 3, case 7 containing a list of a Fortran subprogram for quantification of the error of predicted flow paths.

With best regards,

Kjell Andersson
Secretary to the Coordinating Group

Postadress - Postal address
Box 27106
S-102 52 STOCKHOLM

Gatuadress-Office
Sehlstedtsgatan 11

Telefon-Telephone
08-63 55 60

Telegram-Cable
SWEDATOM, Stockholm

Telex
11961 SWEATOM S
Telefax
08-61 90 86

PARTICIPATION FORM

SIXTH HYDROCOIN WORKSHOP, Washington D.C., May 26-29, 1987

---

1. Organisation

2. Participant(s)

3. Expected new results on level 2 cases

case 1 (heat transfer) ☐ case 2 (lab. exp.) ☐ case 3 (Chalk River) ☐

case 4 (Piceance basin) ☐ case 5 (Cental Valley) ☐

4. Expected level 3 preliminary results

case 1 ☐ case 2 ☐ case 3 ☐ case 4 ☐

case 5 ☐ case 6 ☐ case 7 ☐

5. We plan to present a short contribution on following item(s):

level 2 results ☐ level 3 results ☐ other ☐

6. Titles and authors of short contributions

Please return this form no later than April 25 to:

Kjell Andersson
Swedish Nuclear Power Inspectorate
Box 271 06
S-102 52 STOCKHOLM
SWEDEN

The output format for Case 4 Level 2 has been revised by Mr Paul Davis at Sandia National Laboratories. For changes and additions, please see attached pages.

To: All HYDROCOIN participants

Fm: P. A. Davis, Sandia National Laboratories

Re: Level2/Case4 output format change

The following output format for model error and particle trajectories supercedes the format given on page 4 of the problem definition:

Output Format for Model Error:

Line 1: Line number as shown in figure 20, FORMAT(I5)

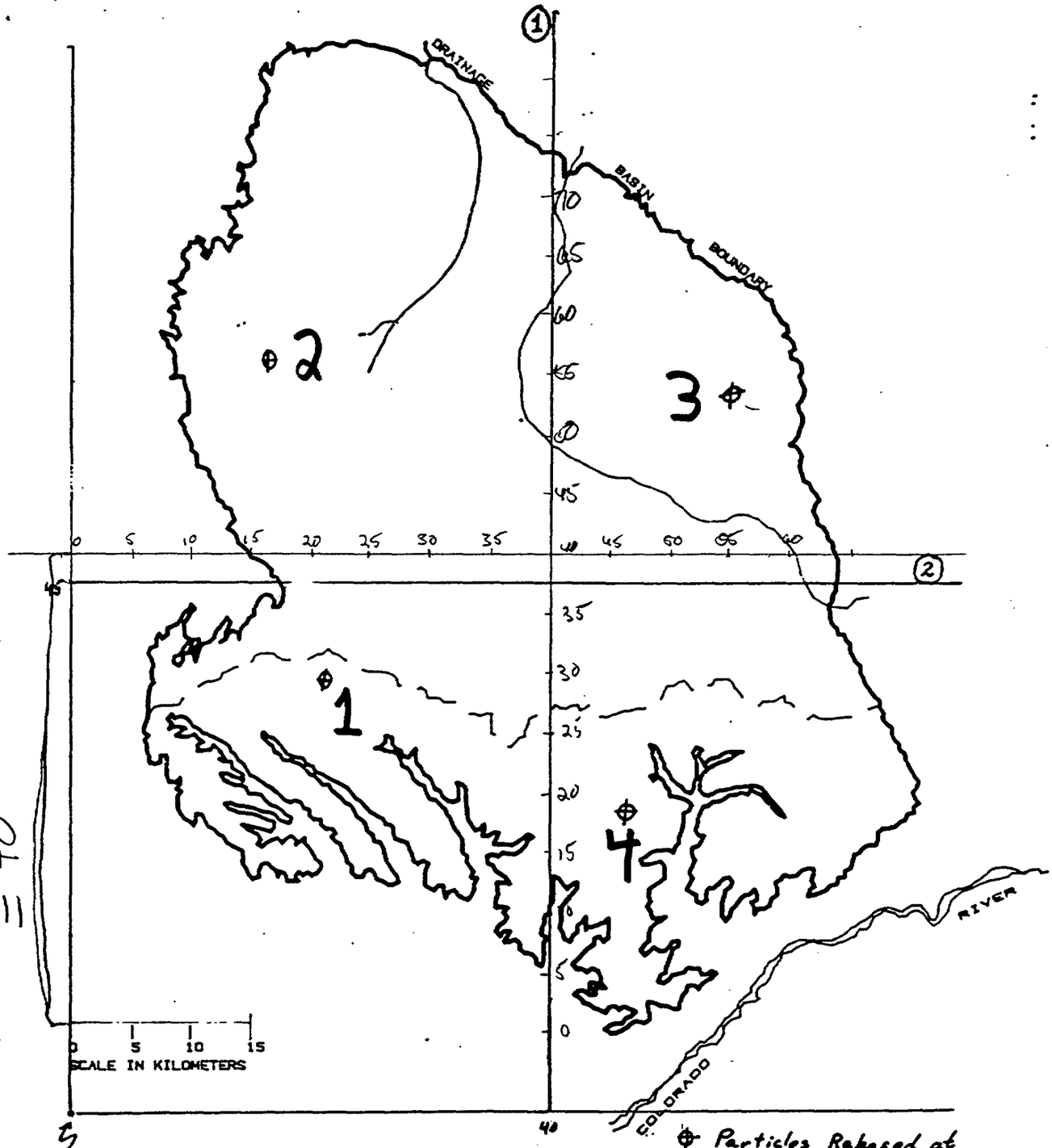
Line 2-: x or y (meters), error(1),error(2),error(4),error(5) where 1,2,4,5 refer to the model layer numbers with layer 1 at the bottom, FORMAT(5(3X,E12.5))

Output Format for Particle Trajectories:

Line 1: Layer number, Trajectory number (see enclosed figure), Number of points, FORMAT(3I5)

Line 2- x(meters), y(meters), z(meters), accumulated length (meters), and accumulated time(seconds), FORMAT(5(3X,E12.5))

The tape characteristics should follow the specifications in the problem description.



= 40

origin

- Figure 20.

Output Requirements

⊕ Particles Released at each point for layers 1, 3, 4, 5.

Enclosed is an Addendum to Case 7 at Level 3 as suggested by Mr Davis and Mr Beyeler at Sandia National Laboratories. It contains a list of a Fortran subprogram in order for the Project Teams to be able to quantify the error of predicted flow paths.

Addendum to HYDROCOIN Level 3 Case 7

COMPARISON OF CALCULATED FLOW PATHS TO AN ANALYTICAL SOLUTION: QUANTIFICATION OF THE ERROR OF PREDICTED FLOW PATHS

by
P. DAVIS
W. BEYELER

Sandia National Laboratories
Albuquerque, New Mexico
U.S.A.

1. INTRODUCTION

The case 7 problem is intended to be a way of evaluating different particle tracking algorithms, and of examining the sensitivity of these algorithms to the resolution of the discretization used to calculate potential (and velocity) fields. The following supplemental procedure gives a quantitative estimate of the accuracy of calculated trajectories.

2. SYSTEM DESCRIPTION

An analytical solution for both the potential and stream-function is available for a pumping well in an infinite confined aquifer (see Bear). In the first part of case 7, samples of potential and velocity were used directly for estimating flow paths. Because discrete samples were used, and because of deviations of the system from the assumptions necessary to a general-purpose tracking algorithm, the estimated flow path will deviate from the analytical flow path.

The analytical expression for the streamfunction value at any point in the upper half-plane is:

$$\Psi = .01y + \tan^{-1}(y/x) \qquad (1)$$

solving this expression for x along a given streamline where $\Psi = \Psi_1$ gives

$$x_1 = y_1/\tan(\Psi_1 - .01y_1) \qquad (2)$$

The squared distance between two points (x1,y1) and (x2,y2) is

$$d^2 = (x_2-x_1)^2+(y_2-y_1)^2 \qquad (3)$$

or when x1 and y1 are on the streamline associated with Ψ_1

$$d^2 = (x_2 - y_1/\tan(\Psi_1 - .01y_1))^2+(y_2-y_1)^2 \qquad (4)$$

Equation (4) can be minimized with respect to y_1, giving the y location of the point on streamline Ψ_1 closest to (x2,y2). The minimum distance to the analytical streamline can then be found from (3).

This distance is the proposed measure of error of the calculated trajectories. HYDROCOIN participants are requested to supply a table of minimum distance vs. path length for paths 3 through 7, along with the trajectories and velocity component profiles requested in the original problem definition.

3. ERROR CALCULATION

The FORTRAN subprograms listed in the appendix calculate the accumulated length along a supplied flow path and the minimum distance from each point in the flow path to the streamline passing through the first point in the flow path. The user must supply arrays containing the x and y coordinates of the calculated flow path, the number of points in the flow path, and arrays for receiving the accumulated length and error at each point in the path.

Equation (4) is solved iteratively by a univariate Newton-Rhapson procedure. This algorithm has been tested using the trajectories calculated by the program TRACKER, and found to converge for all points except those that overshoot the well (for example, flow paths having negative x values originating at release point 7). For these points, we considered the distance to the well to be an appropriate measure of error. The supplied subroutine automatically returns this distance when the iterative procedure fails to converge.

The subprograms may be entered from the listing in the appendix, or may be copied from an ASCII source file available from the directorate.

4. OUTPUT SPECIFICATION

The required problem output consists of the path length and cumulative error evaluated at each point in the flow path for release points 3, 4, 5, 6, and 7 for each griding scheme.

Tape characteristics

- Nine track magnetic tape
- 1600 bits per inch
- EBCDIC or 8 bit ASCII
- Fixed record length
- 80 characters per logical record
- 40 records per block
- Unlabelled tape

Required output format

Repeated for both potential and velocity samples:

Repeated for each grid:

Line 1: Grid number (coarsest=1,finest=3) FORMAT (I5)

Repeated for each release point (3, 4, 5, 6, and 7):

Line 2: Release point number from Table 1.
(original problem definition) FORMAT (I5)

Line 3: Number of trajectory points in the path. FORMAT (I5)

Repeated for each point in the path:

Line 4..: cumulative path length, cumulative error
at the point. FORMAT (2(2x,E12.5))

4. EXAMPLE RESULTS

Figure 1 shows flow paths from points 5 and 6 estimated using the Sandia program TRACKER with an 80 meter grid spacing. Velocities were obtained by sampling potentials. Figure 2 shows plots of the error associated with these paths vs. accumulated path length.

5. REFERENCE

BEAR, JACOB,
Dynamics of Fluids in Porous Media,
American Elsevier (1972). pp 323,324

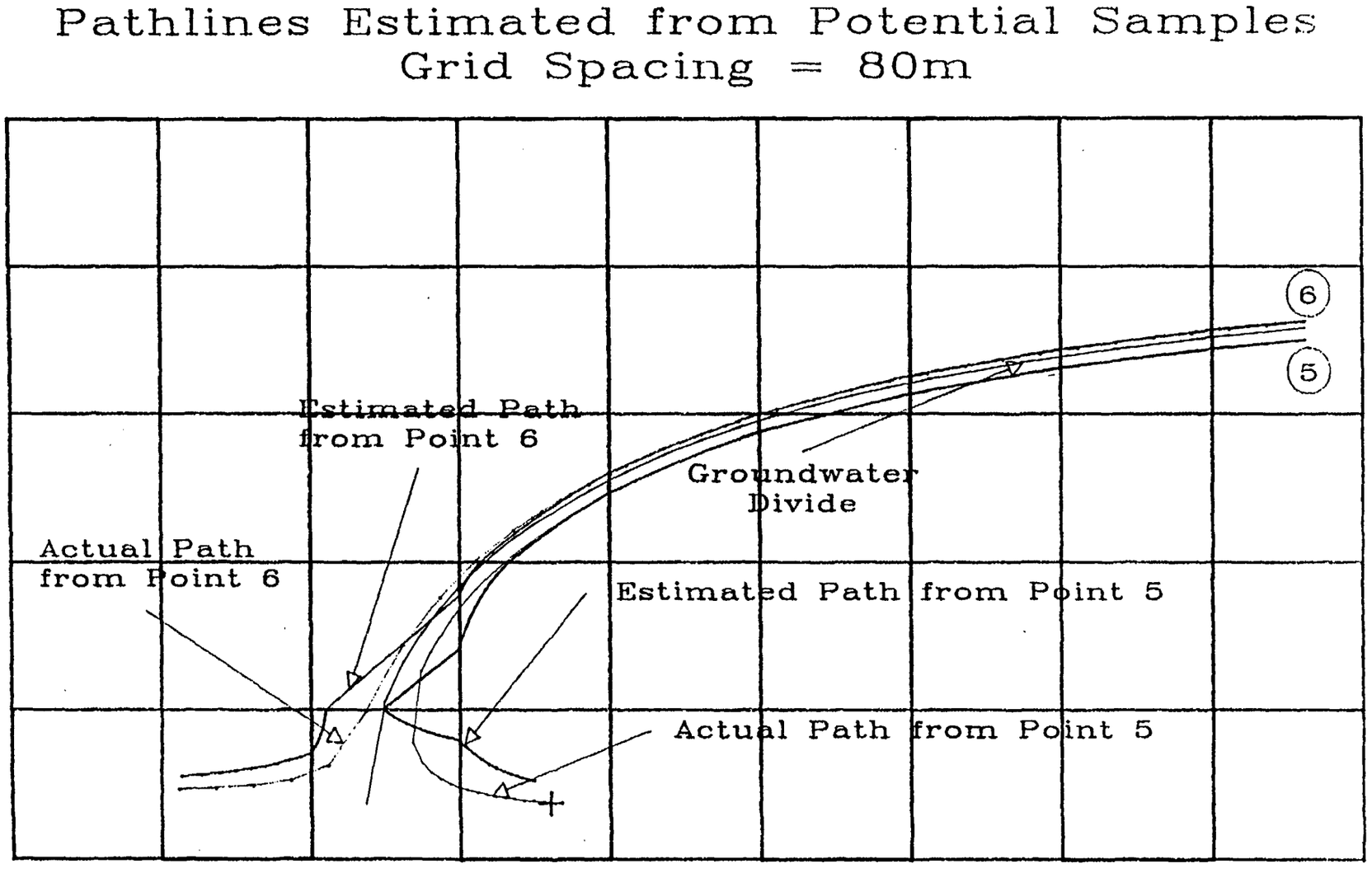
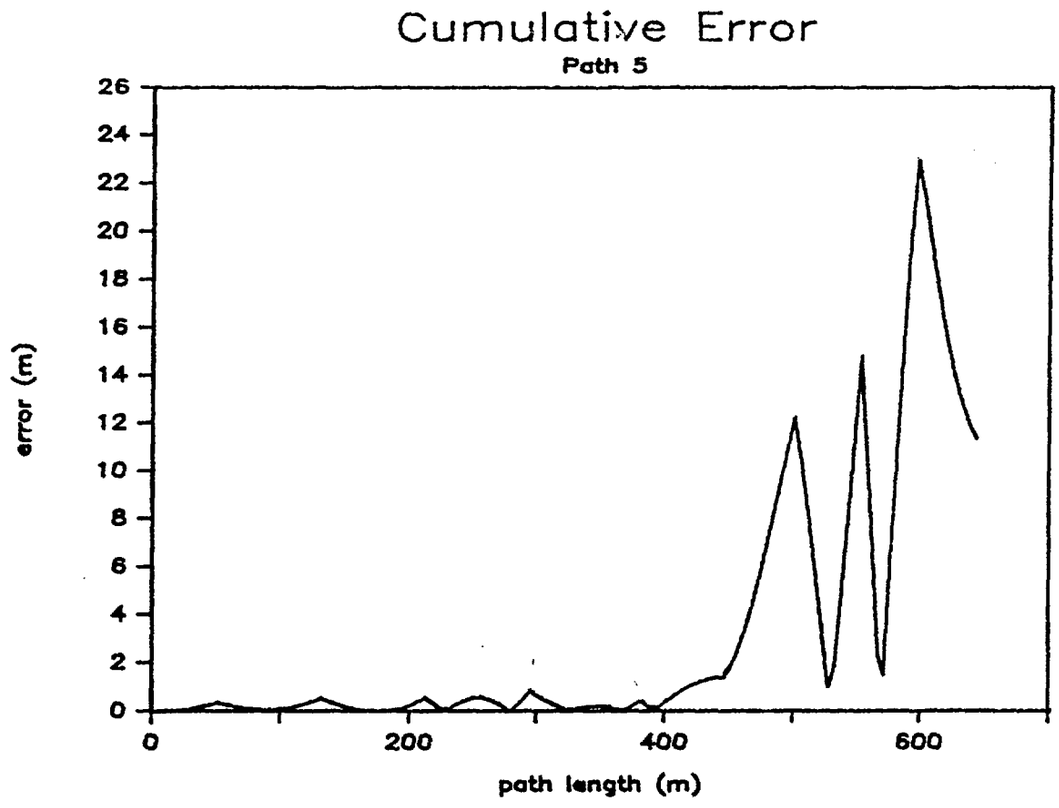


Figure 1 – Calculated Pathlines for Level 3 Case 7



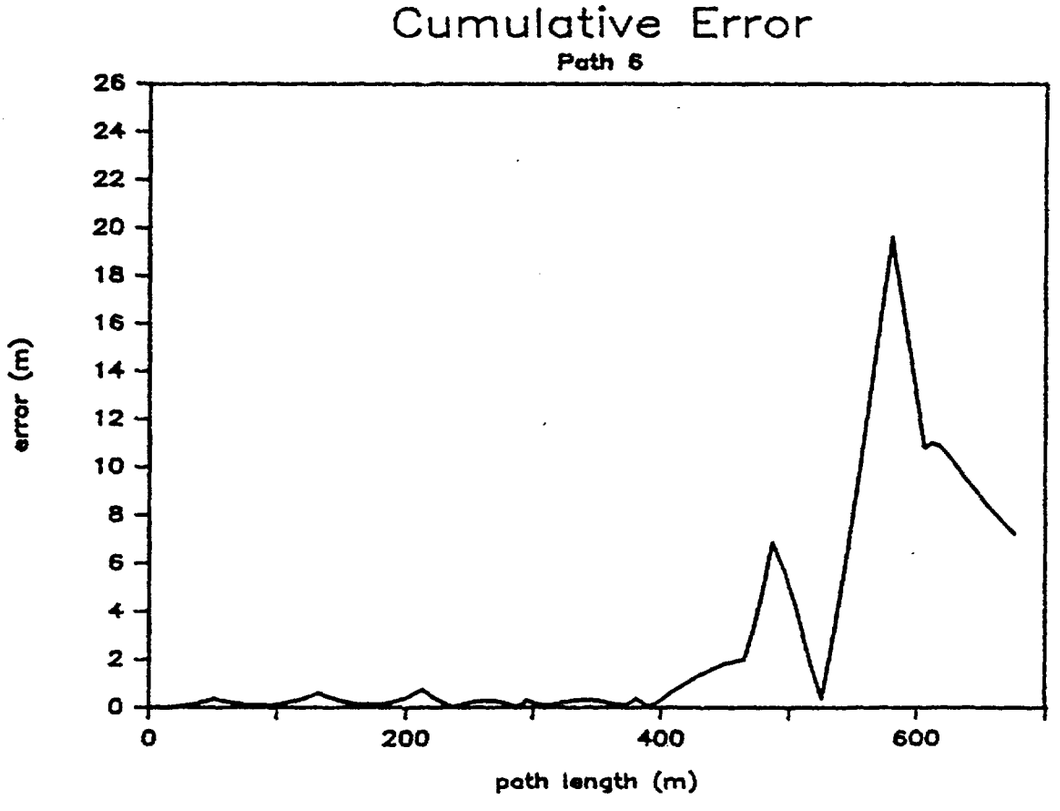


Figure 2 – Error associated with Pathlines Shown in Figure 1

Appendix

The following FORTRAN subprograms evaluate the accumulated length and error at each point in a supplied flow path. The user must provide a calling program which reads the flow path data, translates it if needed so that the pumping well is at the origin, and prints the returned length and error.

```
      subroutine errest(x,y,npts,dist,accerr)
c
c returns the accumulated error estimate at each point in the
c flow path.  The required parameters are:
c
c   input
c
c     (x,y) - arrays containing the calculated flow path
c             coordinates
c
c      npts - the number of points in 'x' and 'y'
c
c
c  output
c
c      dist - array containing the accumulated distance at each point
c
c    accerr - array containing the accumulated error at each point
c
c
cc
      implicit double precision (a-h,o-z)
      double precision x(*),y(*),dist(*),accerr(*)

c initialize

      dist(1)=0.0
      accerr(1)=0.0
      psi0=psi(x(1),y(1))
      tol=1.d-6
      nit=200
      ya=y(1)

c loop through the flow path

      do 10 i=2,npts
        dist(i)=dist(i-1)+dsqrt((x(i)-x(i-1))**2+(y(i)-y(i-1))**2)
c
c find the closest point on curve C
c
        if(ya.ne.-1.d0) then
          yest=ya
          ymax=ya
        else
          yest=y(i)
        endif
        y1a=y1(psi0,x(i),y(i),yest,tol,nit,ymax)
        ya=y1a
```

```
c
c  find the error if function y1 converged, otherwise assume that
c  the flow path has overshot the well
c
        if(y1a.ne.-1.d0) then
          x1a=y1a/dtan(psi0-.01*y1a)
          accerr(i)=dsqrt((x1a-x(i))**2+(y1a-y(i))**2)

        else
          accerr(i)=dsqrt(x(i)**2+y(i)**2)
        endif

 10   continue


      return
      end


      double precision function y1(psi1,x2,y2,yest,tol,nit,ymax)

      implicit double precision (a-h,o-z)

      y1e1=yest
      it=0
      ymx=ymax
      ymn=0.d0
 10   y1e2=y1e1-cnewt(y1e1,x2,y2,psi1)
      if(dabs(y1e1-y1e2).gt.tol) then
        y1e1=y1e2
        it=it+1
        if(y1e1.gt.ymx) then
          y1e1=ymx
          ymx=ymx-tol
        else if(y1e1.lt.ymn) then
          y1e1=ymn
          ymn=ymn+tol
        endif
        if(it.gt.nit) go to 20
        go to 10
      endif

      if(y1e2.lt.0.d0) go to 20
      y1=y1e2
      return

 20   y1=-1.d0
      return

      end
```

```
      double precision function cnewt(y,x2,y2,psi1)

      implicit double precision (a-h,o-z)

      alpha=psi1-.01*y
      sina=dsin(alpha)
      cosa=dcos(alpha)
      sin4a=sina**4

      cnum=(sina*(y*cosa-x2*sina)*(cosa*sina+.01*y)-(y2-y)*sin4a)

      cdenom=((sina*cosa+.01*y)**2+.02*(y*cosa-x2*sina)*
     &(sina+.01*y*cosa)+sin4a)
      cnewt=cnum/cdenom

      return
      end



      double precision function psi(x,y)
c
c evaluates the stream function at (x,y)
c
      double precision x,y
      psi=.01*y+datan2(y,x)
      return
      end
```