Solve a (parameterized) system of differential equations with boundary conditions at two points, using a variable order, variable step size finite difference method with deferred corrections.

Usage

CALL BVPFD (FCNEQN, FCNJAC, FCNBC, FCNPEQ, FCNPBC, N,
NLEFT, NCUPBC, TLEFT, TRIGHT, PISTEP, TOL,
NINIT, TINIT, YINIT, LDYINI, LINEAR, PRINT,
MXGRID, NFINAL, TFINAL, YFINAL, LDYFIN, ERREST)

Arguments

FCNEQN — User-supplied `SUBROUTINE` to evaluate derivatives. The usage is `CALL FCNEQN (N, T, Y, P, DYDT)`, where

`N` — Number of differential equations. (Input)
`T` — Independent variable, *t*. (Input)
`Y` — Array of size `N` containing the dependent variable values, *y*(t). (Input)
`P` — Continuation parameter, *p*. (Input)
See Comment 3.
`DYDT` — Array of size `N` containing the derivatives *y′*(*t*). (Output)

The name `FCNEQN` must be declared `EXTERNAL` in the calling program.

FCNJAC — User-supplied `SUBROUTINE` to evaluate the Jacobian. The usage is `CALL FCNJAC (N, T, Y, P, DYPDY)`, where

`N` — Number of differential equations. (Input)
`T` — Independent variable, *t*. (Input)
`Y` — Array of size `N` containing the dependent variable values. (Input)
`P` — Continuation parameter, *p*. (Input)
See Comments 3.
`DYPDY` — `N` by `N` array containing the partial derivatives $a_{i,j} = \partial f_i / \partial y_j$ evaluated at (*t*, *y*). The values $a_{i,j}$ are returned in `DYPDY(i, j)`. (Output)

The name `FCNJAC` must be declared `EXTERNAL` in the calling program.

FCNBC — User-supplied `SUBROUTINE` to evaluate the boundary conditions. The usage is `CALL FCNBC (N, YLEFT, YRIGHT, P, H)`, where

`N` — Number of differential equations. (Input)
`YLEFT` — Array of size `N` containing the values of the dependent variable at the left endpoint. (Input)
`YRIGHT` — Array of size `N` containing the values of the dependent variable at the right endpoint. (Input)
`P` — Continuation parameter, *p*. (Input)
See Comment 3.
`H` — Array of size `N` containing the boundary condition residuals. (Output)
The boundary conditions are defined by $h_i = 0$; for $i = 1, \ldots, N$. The left endpoint conditions must be defined first, then, the conditions involving both endpoints, and finally the right endpoint conditions.

The name `FCNBC` must be declared `EXTERNAL` in the calling program.

FCNPEQ — User-supplied `SUBROUTINE` to evaluate the partial derivative of *y′* with respect to the parameter *p*. The usage is `CALL FCNPEQ (N, T, Y, P, DYPDP)`, where

`N` — Number of differential equations. (Input)
`T` — Dependent variable, *t*. (Input)

Y — Array of size N containing the dependent variable values. (Input)
P — Continuation parameter, *p*. (Input)
See Comment 3.
DYPDP — Array of size N containing the partial derivatives $a_{i,j} = \partial f_i / \partial y_j$ evaluated at (*t*, *y*). The values $a_{i,j}$ are returned in DYPDY(i, j). (Output)

The name FCNPEQ must be declared EXTERNAL in the calling program.

FCNPBC — User-supplied SUBROUTINE to evaluate the derivative of the boundary conditions with respect to the parameter *p*. The usage is CALL FCNPBC (N, YLEFT, YRIGHT, P, H), where

N — Number of differential equations. (Input)
YLEFT — Array of size N containing the values of the dependent variable at the left endpoint. (Input)
YRIGHT — Array of size N containing the values of the dependent variable at the right endpoint. (Input)
P — Continuation parameter, *p*. (Input)
See Comment 3.
H — Array of size N containing the derivative of f_i with respect to *p*. (Output)

The name FCNPBC must be declared EXTERNAL in the calling program.

N — Number of differential equations. (Input)

NLEFT — Number of initial conditions. (Input)
The value NLEFT must be greater than or equal to zero and less than N.

NCUPBC — Number of coupled boundary conditions. (Input)
The value NLEFT + NCUPBC must be greater than zero and less than or equal to N.

TLEFT — The left endpoint. (Input)

TRIGHT — The right endpoint. (Input)

PISTEP — Initial increment size for *p*. (Input)
If this value is zero, continuation will not be used in this problem. The routines FCNPEQ and FCNPBC will not be called.

TOL — Relative error control parameter. (Input)
The computations stop when ABS(ERROR(J, I))/MAX(ABS(Y(J, I)), 1.0).LT.TOL for all J = 1, . . . , N and I = 1, . . . , NGRID. Here, ERROR(J, I) is the estimated error in Y(J, I).

NINIT — Number of initial grid points, including the endpoints. (Input)
It must be at least 4.

TINIT — Array of size NINIT containing the initial grid points. (Input)

YINIT — Array of size N by NINIT containing an initial guess for the values of Y at the points in TINIT. (Input)

LDYINI — Leading dimension of YINIT exactly as specified in the dimension statement of the calling program. (Input)

LINEAR — Logical .TRUE. if the differential equations and the boundary conditions are linear. (Input)

PRINT — Logical .TRUE. if intermediate output is to be printed. (Input)

MXGRID — Maximum number of grid points allowed. (Input)

NFINAL — Number of final grid points, including the endpoints. (Output)

TFINAL — Array of size MXGRID containing the final grid points. (Output)
Only the first NFINAL points are significant.

YFINAL — Array of size N by MXGRID containing the values of Y at the points in TFINAL. (Output)

LDYFIN — Leading dimension of YFINAL exactly as specified in the dimension statement of the calling program. (Input)

ERREST — Array of size N. (Output)
ERREST(J) is the estimated error in Y(J).

Comments

1. Automatic workspace usage is

BVPFD N(3N * MXGRID + 4N + 1) + MXGRID * (7N + 2) + 2N * MXGRID + N + MXGRID

DBVPFD 2N(3N * MXGRID + 4N + 1) + 2 * MXGRID(7N + 2) + 2N * MXGRID + N + MXGRID

Workspace may be explicitly provided, if desired, by use of B2PFD/DB2PFD. The reference is

```
CALL B2PFD (FCNEQN, FCNJAC, FCNBC, FCNPEQ, FCNPBC,
            N, NLEFT, NCUPBC, TLEFT, TRIGHT, PISTEP,
            TOL, NINIT, TINIT, YINIT, LDYINI,
            LINEAR, PRINT, MXGRID, NFINAL, TFINAL,
            YFINAL, LDYFIN, ERREST, RWORK, IWORK)
```

The additional arguments are as follows:

RWORK — Floating-point work array of size N(3N * MXGRID + 4N + 1) + MXGRID * (7N + 2).

IWORK — Integer work array of size 2N * MXGRID + N + MXGRID.

2. Informational errors

Type	Code	
4	1	More than MXGRID grid points are needed to solve the problem.
4	2	Newton's method diverged.
3	3	Newton's method reached roundoff error level.

3. If the value of PISTEP is greater than zero, then the routine BVPFD assumes that the user has embedded the problem into a one-parameter family of problems:

$y' = y'(t, y, p)$

$h(y\text{tleft}, y\text{tright}, p) = 0$

such that for $p = 0$ the problem is simple. For $p = 1$, the original problem is recovered. The routine BVPFD automatically attempts to increment from $p = 0$ to $p = 1$. The value PISTEP is the beginning increment used in this continuation. The increment will usually be changed by routine BVPFD, but an arbitrary minimum of 0.01 is imposed.

4. The vectors TINIT and TFINAL may be the same.

5. The arrays YINIT and YFINAL may be the same.

Algorithm

The routine BVPFD is based on the subprogram PASVA3 by M. Lentini and V. Pereyra (see Pereyra 1978). The basic discretization is the trapezoidal rule over a nonuniform mesh. This mesh is chosen adaptively, to make the local error approximately the same size everywhere.

Higher-order discretizations are obtained by deferred corrections. Global error estimates are produced to control the computation. The resulting nonlinear algebraic system is solved by Newton's method with step control. The linearized system of equations is solved by a special form of Gauss elimination that preserves the sparseness.

Example 1

This example solves the third-order linear equation

$$y''' - 2y'' + y' - y = \sin t$$

subject to the boundary conditions $y(0) = y(2\pi)$ and $y'(0) = y'(2\pi) = 1$. (Its solution is $y = \sin t$.) To use BVPFD, the problem is reduced to a system of first-order equations by defining $y1 = y$, $y2 = y'$ and $y3 = y''$. The resulting system is

$$\begin{aligned} y_1' &= y_2 & \qquad y_2(0) - 1 &= 0 \\ y_2' &= y_3 & \qquad y_1(0) - y_1(2\pi) &= 0 \\ y_3' &= 2y_3 - y_2 + y_1 + \sin t & \qquad y_2(2\pi) - 1 &= 0 \end{aligned}$$

Note that there is one boundary condition at the left endpoint $t = 0$ and one boundary condition coupling the left and right endpoints. The final boundary condition is at the right endpoint. The total number of boundary conditions must be the same as the number of equations (in this case 3).

Note that since the parameter p is not used in the call to BVPFD, the routines FCNPEQ and FCNPBC are not needed. Therefore, in the call to BVPFD, FCNEQN and FCNBC were used in place of FCNPEQ and FCNPBC.

```
C                                SPECIFICATIONS FOR PARAMETERS
      INTEGER    LDYFIN, LDYINI, MXGRID, NEQNS, NINIT
      PARAMETER  (MXGRID=45, NEQNS=3, NINIT=10, LDYFIN=NEQNS,
     &           LDYINI=NEQNS)
C                                SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER    I, J, NCUPBC, NFINAL, NLEFT, NOUT
      REAL        ERREST(NEQNS), PISTEP, TFINAL(MXGRID), TINIT(NINIT),
     &           TLEFT, TOL, TRIGHT, YFINAL(LDYFIN,MXGRID),
     &           YINIT(LDYINI,NINIT)
      LOGICAL    LINEAR, PRINT
C                                SPECIFICATIONS FOR INTRINSICS
      INTRINSIC  FLOAT
      REAL       FLOAT
C                                SPECIFICATIONS FOR SUBROUTINES
      EXTERNAL   BVPFD, SSET, UMACH
C                                SPECIFICATIONS FOR FUNCTIONS
      EXTERNAL   CONST, FCNBC, FCNEQN, FCNJAC
      REAL       CONST, FCNBC, FCNEQN, FCNJAC
C                                Set parameters
      NLEFT  = 1
      NCUPBC = 1
      TOL    = .001
      TLEFT  = 0.0
      TRIGHT = 2.0*CONST('PI')
      PISTEP = 0.0
      PRINT  = .FALSE.
      LINEAR = .TRUE.
C                                Define TINIT
      DO 10  I=1, NINIT
         TINIT(I) = TLEFT + (I-1)*(TRIGHT-TLEFT)/FLOAT(NINIT-1)
   10 CONTINUE
```

```
C                                 Set YINIT to zero
      DO 20  I=1, NINIT
         CALL SSET (NEQNS, 0.0, YINIT(1,I), 1)
   20 CONTINUE
C                                 Solve problem
      CALL BVPFD (FCNEQN, FCNJAC, FCNBC, FCNEQN, FCNBC, NEQNS, NLEFT,
     &            NCUPBC, TLEFT, TRIGHT, PISTEP, TOL, NINIT, TINIT,
     &            YINIT, LDYINI, LINEAR, PRINT, MXGRID, NFINAL,
     &            TFINAL, YFINAL, LDYFIN, ERREST)
C                                 Print results
      CALL UMACH (2, NOUT)
      WRITE (NOUT,99997)
      WRITE (NOUT,99998) (I,TFINAL(I),(YFINAL(J,I),J=1,NEQNS),I=1,
     &                  NFINAL)
      WRITE (NOUT,99999) (ERREST(J),J=1,NEQNS)
99997 FORMAT (4X, 'I', 7X, 'T', 14X, 'Y1', 13X, 'Y2', 13X, 'Y3')
99998 FORMAT (I5, 1P4E15.6)
99999 FORMAT (' Error estimates', 4X, 1P3E15.6)
      END
      SUBROUTINE FCNEQN (NEQNS, T, Y, P, DYDX)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       T, P, Y(NEQNS), DYDX(NEQNS)
C                                 SPECIFICATIONS FOR INTRINSICS
      INTRINSIC  SIN
      REAL       SIN
C                                 Define PDE
      DYDX(1) = Y(2)
      DYDX(2) = Y(3)
      DYDX(3) = 2.0*Y(3) - Y(2) + Y(1) + SIN(T)
      RETURN
      END
      SUBROUTINE FCNJAC (NEQNS, T, Y, P, DYPDY)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       T, P, Y(NEQNS), DYPDY(NEQNS,NEQNS)
C                                 Define d(DYDX)/dY
      DYPDY(1,1) = 0.0
      DYPDY(1,2) = 1.0
      DYPDY(1,3) = 0.0
      DYPDY(2,1) = 0.0
      DYPDY(2,2) = 0.0
      DYPDY(2,3) = 1.0
      DYPDY(3,1) = 1.0
      DYPDY(3,2) = -1.0
      DYPDY(3,3) = 2.0
      RETURN
      END
      SUBROUTINE FCNBC (NEQNS, YLEFT, YRIGHT, P, F)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       P, YLEFT(NEQNS), YRIGHT(NEQNS), F(NEQNS)
C                                 Define boundary conditions
      F(1) = YLEFT(2) - 1.0
      F(2) = YLEFT(1) - YRIGHT(1)
      F(3) = YRIGHT(2) - 1.0
      RETURN
      END
```

Output

```
 I       T             Y1             Y2             Y3
 1   0.000000E+00  -1.123191E-04   1.000000E+00   6.242319E05
 2   3.490659E-01   3.419107E-01   9.397087E-01  -3.419580E01
 3   6.981317E-01   6.426908E-01   7.660918E-01  -6.427230E-01
 4   1.396263E+00   9.847531E-01   1.737333E-01  -9.847453E-01
 5   2.094395E+00   8.660529E-01  -4.998747E-01  -8.660057E-01
 6   2.792527E+00   3.421830E-01  -9.395474E-01  -3.420648E-01
 7   3.490659E+00  -3.417234E-01  -9.396111E-01   3.418948E-01
 8   4.188790E+00  -8.656880E-01  -5.000588E-01   8.658733E-01
 9   4.886922E+00  -9.845794E-01   1.734571E-01   9.847518E-01
10   5.585054E+00  -6.427721E-01   7.658258E-01   6.429526E-01
11   5.934120E+00  -3.420819E-01   9.395434E-01   3.423986E-01
12   6.283185E+00  -1.123186E-04   1.000000E+00   6.743190E-04
Error estimates     2.840430E-04   1.792939E-04   5.588399E-04
```

Example 2

In this example, the following nonlinear problem is solved:

$y'' - y3 + (1 + \sin 2\, t) \sin t = 0$

with $y(0) = y(\pi) = 0$. Its solution is $y = \sin t$. As in Example 1, this equation is reduced to a system of first-order differential equations by defining $y1 = y$ and $y2 = y'$. The resulting system is

$$
\begin{aligned}
y_1' &= y_2 & \qquad y_1(0) &= 0 \\
y_2' &= y_1^3 - \left(1 + \sin^2 t\right)\sin t & \qquad y_1(\pi) &= 0
\end{aligned}
$$

In this problem, there is one boundary condition at the left endpoint and one at the right endpoint; there are no coupled boundary conditions.

Note that since the parameter p is not used, in the call to BVPFD the routines FCNPEQ and FCNPBC are not needed. Therefore, in the call to BVPFD, FCNEQN and FCNBC were used in place of FCNPEQ and FCNPBC.

```
C                                  SPECIFICATIONS FOR PARAMETERS
      INTEGER    LDYFIN, LDYINI, MXGRID, NEQNS, NINIT
      PARAMETER  (MXGRID=45, NEQNS=2, NINIT=12, LDYFIN=NEQNS,
     &           LDYINI=NEQNS)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER    I, J, NCUPBC, NFINAL, NLEFT, NOUT
      REAL        ERREST(NEQNS), PISTEP, TFINAL(MXGRID), TINIT(NINIT),
     &           TLEFT, TOL, TRIGHT, YFINAL(LDYFIN,MXGRID),
     &           YINIT(LDYINI,NINIT)
      LOGICAL    LINEAR, PRINT
C                                  SPECIFICATIONS FOR INTRINSICS
      INTRINSIC  FLOAT
      REAL       FLOAT
C                                  SPECIFICATIONS FOR SUBROUTINES
      EXTERNAL   BVPFD, UMACH
C                                  SPECIFICATIONS FOR FUNCTIONS
      EXTERNAL   CONST, FCNBC, FCNEQN, FCNJAC
      REAL       CONST
C                                  Set parameters
      NLEFT  = 1
      NCUPBC = 0
```

```
      TOL    = .001
      TLEFT  = 0.0
      TRIGHT = CONST('PI')
      PISTEP = 0.0
      PRINT  = .FALSE.
      LINEAR = .FALSE.
C                                Define TINIT and YINIT
      DO 10  I=1, NINIT
         TINIT(I)   = TLEFT + (I-1)*(TRIGHT-TLEFT)/FLOAT(NINIT-1)
         YINIT(1,I) = 0.4*(TINIT(I)-TLEFT)*(TRIGHT-TINIT(I))
         YINIT(2,I) = 0.4*(TLEFT-TINIT(I)+TRIGHT-TINIT(I))
   10 CONTINUE
C                                Solve problem
      CALL BVPFD (FCNEQN, FCNJAC, FCNBC, FCNEQN, FCNBC, NEQNS, NLEFT,
     &            NCUPBC, TLEFT, TRIGHT, PISTEP, TOL, NINIT, TINIT,
     &            YINIT, LDYINI, LINEAR, PRINT, MXGRID, NFINAL,
     &            TFINAL, YFINAL, LDYFIN, ERREST)
C                                Print results
      CALL UMACH (2, NOUT)
      WRITE (NOUT,99997)
      WRITE (NOUT,99998) (I,TFINAL(I),(YFINAL(J,I),J=1,NEQNS),I=1,
     &                  NFINAL)
      WRITE (NOUT,99999) (ERREST(J),J=1,NEQNS)
99997 FORMAT (4X, 'I', 7X, 'T', 14X, 'Y1', 13X, 'Y2')
99998 FORMAT (I5, 1P3E15.6)
99999 FORMAT (' Error estimates', 4X, 1P2E15.6)
      END
      SUBROUTINE FCNEQN (NEQNS, T, Y, P, DYDT)
C                                SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       T, P, Y(NEQNS), DYDT(NEQNS)
C                                SPECIFICATIONS FOR INTRINSICS
      INTRINSIC  SIN
      REAL       SIN
C                                Define PDE
      DYDT(1) = Y(2)
      DYDT(2) = Y(1)**3 - SIN(T)*(1.0+SIN(T)**2)
      RETURN
      END
      SUBROUTINE FCNJAC (NEQNS, T, Y, P, DYPDY)
C                                SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       T, P, Y(NEQNS), DYPDY(NEQNS,NEQNS)
C                                Define d(DYDT)/dY
      DYPDY(1,1) = 0.0
      DYPDY(1,2) = 1.0
      DYPDY(2,1) = 3.0*Y(1)**2
      DYPDY(2,2) = 0.0
      RETURN
      END
      SUBROUTINE FCNBC (NEQNS, YLEFT, YRIGHT, P, F)
C                                SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       P, YLEFT(NEQNS), YRIGHT(NEQNS), F(NEQNS)
C                                Define boundary conditions
      F(1) = YLEFT(1)
      F(2) = YRIGHT(1)
      RETURN
```

```
      END
```

Output

```
 I       T              Y1             Y2
 1   0.000000E+00   0.000000E+00   9.999277E-01
 2   2.855994E-01   2.817682E-01   9.594315E-01
 3   5.711987E-01   5.406458E-01   8.412407E-01
 4   8.567980E-01   7.557380E-01   6.548904E-01
 5   1.142397E+00   9.096186E-01   4.154530E-01
 6   1.427997E+00   9.898143E-01   1.423307E-01
 7   1.713596E+00   9.898143E-01  -1.423307E-01
 8   1.999195E+00   9.096185E-01  -4.154530E-01
 9   2.284795E+00   7.557380E-01  -6.548903E-01
10   2.570394E+00   5.406460E-01  -8.412405E-01
11   2.855994E+00   2.817683E-01  -9.594313E-01
12   3.141593E+00   0.000000E+00  -9.999274E-01
Error estimates     3.906105E-05   7.124186E-05
```

Example 3

In this example, the following nonlinear problem is solved:

$$y'' - y^3 = \frac{40}{9}\left(t - \frac{1}{2}\right)^{2/3} - \left(t - \frac{1}{2}\right)^8$$

with *y*(0) = *y*(1) = π/2. As in the previous examples, this equation is reduced to a system of first-order differential equations by defining *y*1 = *y* and *y*2 = *y*′. The resulting system is

$$y_1' = y_2 \qquad y_1(0) = \pi / 2$$

$$y_2' = y_1^3 - \frac{40}{9}\left(t - \frac{1}{2}\right)^{2/3} + \left(t - \frac{1}{2}\right)^8 \qquad y_1(1) = \pi / 2$$

The problem is embedded in a family of problems by introducing the parameter *p* and by changing the second differential equation to

$$y_2' = py_1^3 + \frac{40}{9}\left(t - \frac{1}{2}\right)^{2/3} - \left(t - \frac{1}{2}\right)^8$$

At *p* = 0, the problem is linear; and at *p* = 1, the original problem is recovered. The derivatives ∂*y*′/∂*p* must now be specified in the subroutine FCNPEQ. The derivatives ∂*f*/∂*p* are zero in FCNPBC.

```
C                                  SPECIFICATIONS FOR PARAMETERS
      INTEGER    LDYFIN, LDYINI, MXGRID, NEQNS, NINIT
      PARAMETER  (MXGRID=45, NEQNS=2, NINIT=5, LDYFIN=NEQNS,
     &           LDYINI=NEQNS)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER    NCUPBC, NFINAL, NLEFT, NOUT
      REAL       ERREST(NEQNS), PISTEP, TFINAL(MXGRID), TLEFT, TOL,
     &           XRIGHT, YFINAL(LDYFIN,MXGRID)
      LOGICAL    LINEAR, PRINT
C                                  SPECIFICATIONS FOR SAVE VARIABLES
      INTEGER    I, J
      REAL       TINIT(NINIT), YINIT(LDYINI,NINIT)
      SAVE       I, J, TINIT, YINIT
C                                  SPECIFICATIONS FOR SUBROUTINES
```

```
      EXTERNAL   BVPFD, UMACH
C                                 SPECIFICATIONS FOR FUNCTIONS
      EXTERNAL   FCNBC, FCNEQN, FCNJAC, FCNPBC, FCNPEQ
C
      DATA TINIT/0.0, 0.4, 0.5, 0.6, 1.0/
      DATA ((YINIT(I,J),J=1,NINIT),I=1,NEQNS)/0.15749, 0.00215, 0.0,
     &     0.00215, 0.15749, -0.83995, -0.05745, 0.0, 0.05745,
0.83995/
C                                 Set parameters
      NLEFT  = 1
      NCUPBC = 0
      TOL    = .001
      TLEFT  = 0.0
      XRIGHT = 1.0
      PISTEP = 0.1
      PRINT  = .FALSE.
      LINEAR = .FALSE.
C
      CALL BVPFD (FCNEQN, FCNJAC, FCNBC, FCNPEQ, FCNPBC, NEQNS,
NLEFT,
     &            NCUPBC, TLEFT, XRIGHT, PISTEP, TOL, NINIT, TINIT,
     &            YINIT, LDYINI, LINEAR, PRINT, MXGRID, NFINAL,
     &            TFINAL, YFINAL, LDYFIN, ERREST)
C                                 Print results
      CALL UMACH (2, NOUT)
      WRITE (NOUT,99997)
      WRITE (NOUT,99998) (I,TFINAL(I),(YFINAL(J,I),J=1,NEQNS),I=1,
     &                  NFINAL)
      WRITE (NOUT,99999) (ERREST(J),J=1,NEQNS)
99997 FORMAT (4X, 'I', 7X, 'T', 14X, 'Y1', 13X, 'Y2')
99998 FORMAT (I5, 1P3E15.6)
99999 FORMAT (' Error estimates', 4X, 1P2E15.6)
      END
      SUBROUTINE FCNEQN (NEQNS, T, Y, P, DYDT)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       T, P, Y(NEQNS), DYDT(NEQNS)
C                                 Define PDE
      DYDT(1) = Y(2)
      DYDT(2) = P*Y(1)**3 + 40./9.*((T-0.5)**2)**(1./3.) - (T-
0.5)**8
      RETURN
      END
      SUBROUTINE FCNJAC (NEQNS, T, Y, P, DYPDY)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       T, P, Y(NEQNS), DYPDY(NEQNS,NEQNS)
C                                 Define d(DYDT)/dY
      DYPDY(1,1) = 0.0
      DYPDY(1,2) = 1.0
      DYPDY(2,1) = P*3.*Y(1)**2
      DYPDY(2,2) = 0.0
      RETURN
      END
      SUBROUTINE FCNBC (NEQNS, YLEFT, YRIGHT, P, F)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       P, YLEFT(NEQNS), YRIGHT(NEQNS), F(NEQNS)
```

```
C                                 SPECIFICATIONS FOR LOCAL VARIABLES
      REAL       PI
C                                 SPECIFICATIONS FOR FUNCTIONS
      EXTERNAL   CONST
      REAL       CONST
C                                 Define boundary conditions
      PI   = CONST('PI')
      F(1) = YLEFT(1) - PI/2.0
      F(2) = YRIGHT(1) - PI/2.0
      RETURN
      END
      SUBROUTINE FCNPEQ (NEQNS, T, Y, P, DYPDP)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       T, P, Y(NEQNS), DYPDP(NEQNS)
C                                 Define d(DYDT)/dP
      DYPDP(1) = 0.0
      DYPDP(2) = Y(1)**3
      RETURN
      END
      SUBROUTINE FCNPBC (NEQNS, YLEFT, YRIGHT, P, DFDP)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER    NEQNS
      REAL       P, YLEFT(NEQNS), YRIGHT(NEQNS), DFDP(NEQNS)
C                                 SPECIFICATIONS FOR SUBROUTINES
      EXTERNAL   SSET
C                                 Define dF/dP
      CALL SSET (NEQNS, 0.0, DFDP, 1)
      RETURN
      END
```

Output

```
 I       T              Y1             Y2
 1   0.000000E+00   1.570796E+00  -1.949336E+00
 2   4.444445E-02   1.490495E+00  -1.669567E+00
 3   8.888889E-02   1.421951E+00  -1.419465E+00
 4   1.333333E-01   1.363953E+00  -1.194307E+00
 5   2.000000E-01   1.294526E+00  -8.958461E-01
 6   2.666667E-01   1.243628E+00  -6.373191E-01
 7   3.333334E-01   1.208785E+00  -4.135206E-01
 8   4.000000E-01   1.187783E+00  -2.219351E-01
 9   4.250000E-01   1.183038E+00  -1.584200E-01
10   4.500000E-01   1.179822E+00  -9.973146E-02
11   4.625000E-01   1.178748E+00  -7.233893E-02
12   4.750000E-01   1.178007E+00  -4.638248E-02
13   4.812500E-01   1.177756E+00  -3.399763E-02
14   4.875000E-01   1.177582E+00  -2.205547E-02
15   4.937500E-01   1.177480E+00  -1.061177E-02
16   5.000000E-01   1.177447E+00  -1.479182E-07
17   5.062500E-01   1.177480E+00   1.061153E-02
18   5.125000E-01   1.177582E+00   2.205518E-02
19   5.187500E-01   1.177756E+00   3.399727E-02
20   5.250000E-01   1.178007E+00   4.638219E-02
21   5.375000E-01   1.178748E+00   7.233876E-02
22   5.500000E-01   1.179822E+00   9.973124E-02
23   5.750000E-01   1.183038E+00   1.584199E-01
24   6.000000E-01   1.187783E+00   2.219350E-01
```

25	6.666667E-01	1.208786E+00	4.135205E-01
26	7.333333E-01	1.243628E+00	6.373190E-01
27	8.000000E-01	1.294526E+00	8.958461E-01
28	8.666667E-01	1.363953E+00	1.194307E+00
29	9.111111E-01	1.421951E+00	1.419465E+00
30	9.555556E-01	1.490495E+00	1.669566E+00
31	1.000000E+00	1.570796E+00	1.949336E+00
Error estimates		3.448358E-06	5.549869E-05