NAG Library Routine Document

D03RAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

D03RAF integrates a system of linear or nonlinear, time-dependent partial differential equations (PDEs) in two space dimensions on a rectangular domain. The method of lines is employed to reduce the PDEs to a system of ordinary differential equations (ODEs) which are solved using a backward differentiation formula (BDF) method. The resulting system of nonlinear equations is solved using a modified Newton method and a Bi-CGSTAB iterative linear solver with ILU preconditioning. Local uniform grid refinement is used to improve the accuracy of the solution. D03RAF originates from the VLUGR2 package (see Blom and Verwer (1993) and Blom *et al.* (1996)).

2 Specification

```
SUBROUTINE D03RAF (NPDE, TS, TOUT, DT, XMIN, XMAX, YMIN, YMAX, NX, NY,     &
                   TOLS, TOLT, PDEDEF, BNDARY, PDEIV, MONITR, OPTI,        &
                   OPTR, RWK, LENRWK, IWK, LENIWK, LWK, LENLWK, ITRACE,    &
                   IND, IFAIL)
INTEGER            NPDE, NX, NY, OPTI(4), LENRWK, IWK(LENIWK), LENIWK,     &
                   LENLWK, ITRACE, IND, IFAIL
REAL (KIND=nag_wp) TS, TOUT, DT(3), XMIN, XMAX, YMIN, YMAX, TOLS, TOLT,    &
                   OPTR(3,NPDE), RWK(LENRWK)
LOGICAL            LWK(LENLWK)
EXTERNAL           PDEDEF, BNDARY, PDEIV, MONITR
```

3 Description

D03RAF integrates the system of PDEs:

$$F_j\big(t, x, y, u, u_t, u_x, u_y, u_{xx}, u_{xy}, u_{yy}\big) = 0, \quad j = 1, 2, \ldots, \text{NPDE}, \tag{1}$$

for x and y in the rectangular domain $x_{\min} \leq x \leq x_{\max}$, $y_{\min} \leq y \leq y_{\max}$, and time interval $t_0 \leq t \leq t_{\text{out}}$, where the vector u is the set of solution values

$$u(x, y, t) = [u_1(x, y, t), \ldots, u_{\text{NPDE}}(x, y, t)]^{\text{T}},$$

and u_t denotes partial differentiation with respect to t, and similarly for u_x etc.

The functions F_j must be supplied by you in PDEDEF. Similarly the initial values of the functions $u(x, y, t)$ must be specified at $t = t_0$ in PDEIV.

Note that whilst complete generality is offered by the master equations (1), D03RAF is not appropriate for all PDEs. In particular, hyperbolic systems should not be solved using this routine. Also, at least one component of u_t must appear in the system of PDEs.

The boundary conditions must be supplied by you in BNDARY in the form

$$G_j\big(t, x, y, u, u_t, u_x, u_y\big) = 0, \tag{2}$$

for all y when $x_{\min}$ or $x_{\max}$ and for all x when $y = y_{\min}$ or $y = y_{\max}$ and $j = 1, 2, \ldots, \text{NPDE}$

The domain is covered by a uniform coarse base grid of size $n_x \times n_y$ specified by you, and nested finer uniform subgrids are subsequently created in regions with high spatial activity. The refinement is controlled using a space monitor which is computed from the current solution and a user-supplied space tolerance TOLS. A number of optional parameters, e.g., the maximum number of grid levels at any time, and some weighting factors, can be specified in the arrays OPTI and OPTR. Further details of the refinement strategy can be found in Section 9.

The system of PDEs and the boundary conditions are discretized in space on each grid using a standard second-order finite difference scheme (centred on the internal domain and one-sided at the boundaries), and the resulting system of ODEs is integrated in time using a second-order, two-step, implicit BDF method with variable step size. The time integration is controlled using a time monitor computed at each grid level from the current solution and a user-supplied time tolerance TOLT, and some further optional user-specified weighting factors held in OPTR (see Section 9 for details). The time monitor is used to compute a new step size, subject to restrictions on the size of the change between steps, and (optional) user-specified maximum and minimum step sizes held in DT. The step size is adjusted so that the remaining integration interval is an integer number times Δt. In this way a solution is obtained at $t = t_{\text{out}}$.

A modified Newton method is used to solve the nonlinear equations arising from the time integration. You may specify (in OPTI) the maximum number of Newton iterations to be attempted. A Jacobian matrix is calculated at the beginning of each time step. If the Newton process diverges or the maximum number of iterations is exceeded, a new Jacobian is calculated using the most recent iterates and the Newton process is restarted. If convergence is not achieved after the (optional) user-specified maximum number of new Jacobian evaluations, the time step is retried with $\Delta t = \Delta t/4$. The linear systems arising from the Newton iteration are solved using a Bi-CGSTAB iterative method, in combination with ILU preconditioning. The maximum number of iterations can be specified by you in OPTI.

The solution at all grid levels is stored in the workspace arrays, along with other information needed for a restart (i.e., a continuation call). It is not intended that you extract the solution from these arrays, indeed the necessary information regarding these arrays is not included. The user-supplied monitor MONITR should be used to obtain the solution at particular levels and times. MONITR is called at the end of every time step, with the last step being identified via the input argument TLAST.

Within PDEIV, PDEDEF, BNDARY and MONITR the data structure is as follows. Each point on a particular grid is given an index (ranging from 1 to the total number of points on the grid) and all coordinate or solution information is stored in arrays according to this index, e.g., $\mathrm{X}(i)$ and $\mathrm{Y}(i)$ contain the x- and y coordinate of point i, and $\mathrm{U}(i,j)$ contains the jth solution component u_j at point i.

Further details of the underlying algorithm can be found in Section 9 and in Blom and Verwer (1993) and Blom *et al.* (1996) and the references therein.

4 References

Adjerid S and Flaherty J E (1988) A local refinement finite element method for two-dimensional parabolic systems *SIAM J. Sci. Statist. Comput.* **9** 792–811

Blom J G, Trompert R A and Verwer J G (1996) Algorithm 758. VLUGR2: A vectorizable adaptive grid solver for PDEs in 2D *Trans. Math. Software* **22** 302–328

Blom J G and Verwer J G (1993) VLUGR2: A vectorized local uniform grid refinement code for PDEs in 2D *Report NM-R9306* CWI, Amsterdam

Brown P N, Hindmarsh A C and Petzold L R (1994) Using Krylov methods in the solution of large scale differential-algebraic systems *SIAM J. Sci. Statist. Comput.* **15** 1467–1488

Trompert R A (1993) Local uniform grid refinement and systems of coupled partial differential equations *Appl. Numer. Maths* **12** 331–355

Trompert R A and Verwer J G (1993) Analysis of the implicit Euler local uniform grid refinement method *SIAM J. Sci. Comput.* **14** 259–278

5 Arguments

1: NPDE – INTEGER *Input*

On entry: the number of PDEs in the system.

Constraint: NPDE ≥ 1.

2: TS – REAL (KIND=nag_wp) *Input/Output*

On entry: the initial value of the independent variable t.

On exit: the value of t which has been reached. Normally TS = TOUT.

Constraint: TS < TOUT.

3: TOUT – REAL (KIND=nag_wp) *Input*

On entry: the final value of t to which the integration is to be carried out.

4: DT(3) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the initial, minimum and maximum time step sizes respectively.

DT(1)
: Specifies the initial time step size to be used on the first entry, i.e., when IND = 0. If DT(1) = 0.0 then the default value DT(1) = 0.01 × (TOUT − TS) is used. On subsequent entries (IND = 1), the value of DT(1) is not referenced.

DT(2)
: Specifies the minimum time step size to be attempted by the integrator. If DT(2) = 0.0 the default value DT(2) = 10.0 × ***machine precision*** is used.

DT(3)
: Specifies the maximum time step size to be attempted by the integrator. If DT(3) = 0.0 the default value DT(3) = TOUT − TS is used.

On exit: DT(1) contains the time step size for the next time step. DT(2) and DT(3) are unchanged or set to their default values if zero on entry.

Constraints:

if IND = 0, DT(1) ≥ 0.0;
if IND = 0 and DT(1) > 0.0,
10.0 × ***machine precision*** × max(|TS|, |TOUT|) ≤ DT(1) ≤ TOUT − TS and
DT(2) ≤ DT(1) ≤ DT(3), where the values of DT(2) and DT(3) will have been reset to their default values if zero on entry;
0 ≤ DT(2) ≤ DT(3).

5: XMIN – REAL (KIND=nag_wp) *Input*
6: XMAX – REAL (KIND=nag_wp) *Input*

On entry: the extents of the rectangular domain in the x-direction, i.e., the x coordinates of the left and right boundaries respectively.

Constraint: XMIN < XMAX and XMAX must be sufficiently distinguishable from XMIN for the precision of the machine being used.

7: YMIN – REAL (KIND=nag_wp) *Input*
8: YMAX – REAL (KIND=nag_wp) *Input*

On entry: the extents of the rectangular domain in the y-direction, i.e., the y coordinates of the lower and upper boundaries respectively.

Constraint: YMIN < YMAX and YMAX must be sufficiently distinguishable from YMIN for the precision of the machine being used.

9: NX – INTEGER *Input*

On entry: the number of grid points in the x-direction (including the boundary points).

Constraint: NX ≥ 4.

10: NY – INTEGER *Input*

On entry: the number of grid points in the y-direction (including the boundary points).

Constraint: NY ≥ 4.

11: TOLS – REAL (KIND=nag_wp) *Input*

On entry: the space tolerance used in the grid refinement strategy (σ in equation (4)). See Section 9.2.

Constraint: TOLS > 0.0.

12: TOLT – REAL (KIND=nag_wp) *Input*

On entry: the time tolerance used to determine the time step size (τ in equation (7)). See Section 9.3.

Constraint: TOLT > 0.0.

13: PDEDEF – SUBROUTINE, supplied by the user. *External Procedure*

PDEDEF must evaluate the functions F_j, for $j = 1, 2, \ldots, \text{NPDE}$, in equation (1) which define the system of PDEs (i.e., the residuals of the resulting ODE system) at all interior points of the domain. Values at points on the boundaries of the domain are ignored and will be overwritten by BNDARY. PDEDEF is called for each subgrid in turn.

The specification of PDEDEF is:

```
SUBROUTINE PDEDEF (NPTS, NPDE, T, X, Y, U, UT, UX, UY, UXX, UXY,     &
                   UYY, RES)

INTEGER            NPTS, NPDE
REAL (KIND=nag_wp) T, X(NPTS), Y(NPTS), U(NPTS,NPDE),                &
                   UT(NPTS,NPDE), UX(NPTS,NPDE), UY(NPTS,NPDE),      &
                   UXX(NPTS,NPDE), UXY(NPTS,NPDE),                   &
                   UYY(NPTS,NPDE), RES(NPTS,NPDE)
```

1: NPTS – INTEGER *Input*

On entry: the number of grid points in the current grid.

2: NPDE – INTEGER *Input*

On entry: the number of PDEs in the system.

3: T – REAL (KIND=nag_wp) *Input*

On entry: the current value of the independent variable t.

4: X(NPTS) – REAL (KIND=nag_wp) array *Input*

On entry: X(i) contains the x coordinate of the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$.

5: Y(NPTS) – REAL (KIND=nag_wp) array *Input*

On entry: Y(i) contains the y coordinate of the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$.

6: U(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: U(i, j) contains the value of the jth PDE component at the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$ and $j = 1, 2, \ldots, \text{NPDE}$.

7: UT(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UT(i,j) contains the value of $\dfrac{\partial u}{\partial t}$ for the jth PDE component at the ith grid point, for $i = 1,2,\ldots,$NPTS and $j = 1,2,\ldots,$NPDE.

8: UX(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UX(i,j) contains the value of $\dfrac{\partial u}{\partial x}$ for the jth PDE component at the ith grid point, for $i = 1,2,\ldots,$NPTS and $j = 1,2,\ldots,$NPDE.

9: UY(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UY(i,j) contains the value of $\dfrac{\partial u}{\partial y}$ for the jth PDE component at the ith grid point, for $i = 1,2,\ldots,$NPTS and $j = 1,2,\ldots,$NPDE.

10: UXX(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UXX(i,j) contains the value of $\dfrac{\partial^2 u}{\partial x^2}$ for the jth PDE component at the ith grid point, for $i = 1,2,\ldots,$NPTS and $j = 1,2,\ldots,$NPDE.

11: UXY(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UXY(i,j) contains the value of $\dfrac{\partial^2 u}{\partial x \partial y}$ for the jth PDE component at the ith grid point, for $i = 1,2,\ldots,$NPTS and $j = 1,2,\ldots,$NPDE.

12: UYY(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UYY(i,j) contains the value of $\dfrac{\partial^2 u}{\partial y^2}$ for the jth PDE component at the ith grid point, for $i = 1,2,\ldots,$NPTS and $j = 1,2,\ldots,$NPDE.

13: RES(NPTS, NPDE) – REAL (KIND=nag_wp) array *Output*

On exit: RES(i,j) must contain the value of F_j, for $j = 1,2,\ldots,$NPDE, at the ith grid point, for $i = 1,2,\ldots,$NPTS, although the residuals at boundary points will be ignored (and overwritten later on) and so they need not be specified here.

PDEDEF must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub) program from which D03RAF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

14: BNDARY – SUBROUTINE, supplied by the user. *External Procedure*

BNDARY must evaluate the functions G_j, for $j = 1,2,\ldots,$NPDE, in equation (2) which define the boundary conditions at all boundary points of the domain. Residuals at interior points must **not** be altered by this subroutine.

The specification of BNDARY is:

```
SUBROUTINE BNDARY (NPTS, NPDE, T, X, Y, U, UT, UX, UY, NBPTS,        &
                   LBND, RES)

INTEGER            NPTS, NPDE, NBPTS, LBND(NBPTS)
REAL (KIND=nag_wp) T, X(NPTS), Y(NPTS), U(NPTS,NPDE),                &
                   UT(NPTS,NPDE), UX(NPTS,NPDE), UY(NPTS,NPDE),      &
                   RES(NPTS,NPDE)
```

1: NPTS – INTEGER *Input*

On entry: the number of grid points in the current grid.

2: NPDE – INTEGER *Input*

On entry: the number of PDEs in the system.

3: T – REAL (KIND=nag_wp) *Input*

On entry: the current value of the independent variable t.

4: X(NPTS) – REAL (KIND=nag_wp) array *Input*

On entry: X(i) contains the x coordinate of the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$.

5: Y(NPTS) – REAL (KIND=nag_wp) array *Input*

On entry: Y(i) contains the y coordinate of the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$.

6: U(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: U(i, j) contains the value of the jth PDE component at the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$ and $j = 1, 2, \ldots, \text{NPDE}$.

7: UT(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UT(i, j) contains the value of $\dfrac{\partial u}{\partial t}$ for the jth PDE component at the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$ and $j = 1, 2, \ldots, \text{NPDE}$.

8: UX(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UX(i, j) contains the value of $\dfrac{\partial u}{\partial x}$ for the jth PDE component at the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$ and $j = 1, 2, \ldots, \text{NPDE}$.

9: UY(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input*

On entry: UY(i, j) contains the value of $\dfrac{\partial u}{\partial y}$ for the jth PDE component at the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$ and $j = 1, 2, \ldots, \text{NPDE}$.

10: NBPTS – INTEGER *Input*

On entry: the number of boundary points in the grid.

11: LBND(NBPTS) – INTEGER array *Input*

On entry: LBND(i) contains the grid index for the ith boundary point, for $i = 1, 2, \ldots, \text{NBPTS}$. Hence the ith boundary point has coordinates X(LBND(i)) and Y(LBND(i)), and the corresponding solution values are U(LBND(i), NPDE), etc.

12: RES(NPTS, NPDE) – REAL (KIND=nag_wp) array *Input/Output*

On entry: RES(i, j) contains the value of F_j, for $i = 1, 2, \ldots, \text{NPDE}$, at the ith grid point, for $i = 1, 2, \ldots, \text{NPTS}$, as returned by PDEDEF. The residuals at the boundary points will be overwritten and so need not have been set by PDEDEF.

On exit: RES(LBND(i), j) must contain the value of G_j, for $j = 1, 2, \ldots, \text{NPDE}$, at the ith boundary point, for $i = 1, 2, \ldots, \text{NBPTS}$.

Note: elements of RES corresponding to interior points must **not** be altered.

BNDARY must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which D03RAF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

15: PDEIV – SUBROUTINE, supplied by the user. *External Procedure*

PDEIV must specify the initial values of the PDE components u at all points in the grid. PDEIV is not referenced if, on entry, IND $= 1$.

The specification of PDEIV is:

```
SUBROUTINE PDEIV (NPTS, NPDE, T, X, Y, U)
INTEGER            NPTS, NPDE
REAL (KIND=nag_wp) T, X(NPTS), Y(NPTS), U(NPTS,NPDE)
```

1: NPTS – INTEGER *Input*

On entry: the number of grid points in the grid.

2: NPDE – INTEGER *Input*

On entry: the number of PDEs in the system.

3: T – REAL (KIND=nag_wp) *Input*

On entry: the (initial) value of the independent variable t.

4: X(NPTS) – REAL (KIND=nag_wp) array *Input*

On entry: X(i) contains the x coordinate of the ith grid point, for $i = 1, 2, \ldots,$ NPTS.

5: Y(NPTS) – REAL (KIND=nag_wp) array *Input*

On entry: Y(i) contains the y coordinate of the ith grid point, for $i = 1, 2, \ldots,$ NPTS.

6: U(NPTS, NPDE) – REAL (KIND=nag_wp) array *Output*

On exit: U(i, j) must contain the value of the jth PDE component at the ith grid point, for $i = 1, 2, \ldots,$ NPTS and $j = 1, 2, \ldots,$ NPDE.

PDEIV must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub) program from which D03RAF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

16: MONITR – SUBROUTINE, supplied by the user. *External Procedure*

MONITR is called by D03RAF at the end of every successful time step, and may be used to examine or print the solution or perform other tasks such as error calculations, particularly at the final time step, indicated by the argument TLAST. The input arguments contain information about the grid and solution at all grid levels used.

MONITR can also be used to force an immediate tidy termination of the solution process and return to the calling program.

The specification of MONITR is:

```
SUBROUTINE MONITR (NPDE, T, DT, DTNEW, TLAST, NLEV, NGPTS, XPTS,     &
                   YPTS, LSOL, SOL, IERR)
INTEGER            NPDE, NLEV, NGPTS(NLEV), LSOL(NLEV), IERR
REAL (KIND=nag_wp) T, DT, DTNEW, XPTS(*), YPTS(*), SOL(*)
LOGICAL            TLAST
```

1: NPDE – INTEGER *Input*

On entry: the number of PDEs in the system.

2: T – REAL (KIND=nag_wp) *Input*

On entry: the current value of the independent variable t, i.e., the time at the end of the integration step just completed.

3: DT – REAL (KIND=nag_wp) *Input*

On entry: the current time step size Δt, i.e., the time step size used for the integration step just completed.

4: DTNEW – REAL (KIND=nag_wp) *Input*

On entry: the step size that will be used for the next time step.

5: TLAST – LOGICAL *Input*

On entry: indicates if intermediate or final time step. TLAST = .FALSE. for an intermediate step, TLAST = .TRUE. for the last call to MONITR before returning to your program.

6: NLEV – INTEGER *Input*

On entry: the number of grid levels used at time T.

7: NGPTS(NLEV) – INTEGER array *Input*

On entry: NGPTS(l) contains the number of grid points at level l, for $l = 1, 2, \ldots,$ NLEV.

8: XPTS($*$) – REAL (KIND=nag_wp) array *Input*

On entry: contains the x coordinates of the grid points in each level in turn, i.e., X(i), for $i = 1, 2, \ldots,$ NGPTS(l) and $l = 1, 2, \ldots,$ NLEV.

So for level l, X(i) = XPTS($k + i$), where $k =$ NGPTS(1) + NGPTS(2) + $\cdots$ + NGPTS($l - 1$), for $i = 1, 2, \ldots,$ NGPTS(l) and $l = 1, 2, \ldots,$ NLEV.

9: YPTS($*$) – REAL (KIND=nag_wp) array *Input*

On entry: contains the y coordinates of the grid points in each level in turn, i.e., Y(i), for $i = 1, 2, \ldots,$ NGPTS(l) and $l = 1, 2, \ldots,$ NLEV.

So for level l, Y(i) = YPTS($k + i$), where $k =$ NGPTS(1) + NGPTS(2) + $\cdots$ + NGPTS($l - 1$), for $i = 1, 2, \ldots,$ NGPTS(l) and $l = 1, 2, \ldots,$ NLEV.

10: LSOL(NLEV) – INTEGER array *Input*

On entry: LSOL(l) contains the pointer to the solution in SOL at grid level l and time T. (LSOL(l) actually contains the array index immediately preceding the start of the solution in SOL.)

11: SOL($*$) – REAL (KIND=nag_wp) array *Input*

On entry: contains the solution U(NGPTS(l), NPDE) at time T for each grid level l in turn, positioned according to LSOL, i.e., for level l, U(i, j) = SOL(LSOL(l) + ($j - 1$) $\times$ NGPTS(l) + i), for $i = 1, 2, \ldots,$ NGPTS(l), $j = 1, 2, \ldots,$ NPDE and $l = 1, 2, \ldots,$ NLEV.

12: IERR – INTEGER *Input/Output*

On entry: will be set to 0.

On exit: should be set to 1 to force a tidy termination and an immediate return to the calling program with IFAIL $= 4$. IERR should remain unchanged otherwise.

MONITR must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which D03RAF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

17: OPTI(4) – INTEGER array *Input*

On entry: may be set to control various options available in the integrator.

OPTI$(1) = 0$
All the default options are employed.

OPTI$(1) > 0$
The default value of OPTI(i), for $i = 2, 3, 4$, can be obtained by setting OPTI$(i) = 0$.

OPTI(1)
Specifies the maximum number of grid levels allowed (including the base grid). OPTI$(1) \geq 0$. The default value is OPTI$(1) = 3$.

OPTI(2)
Specifies the maximum number of Jacobian evaluations allowed during each nonlinear equations solution. OPTI$(2) \geq 0$. The default value is OPTI$(2) = 2$.

OPTI(3)
Specifies the maximum number of Newton iterations in each nonlinear equations solution. OPTI$(3) \geq 0$. The default value is OPTI$(3) = 10$.

OPTI(4)
Specifies the maximum number of iterations in each linear equations solution. OPTI$(4) \geq 0$. The default value is OPTI$(4) = 100$.

Constraint: OPTI$(1) \geq 0$ and if OPTI$(1) > 0$, OPTI$(i) \geq 0$, for $i = 2, 3, 4$.

18: OPTR$(3, \text{NPDE})$ – REAL (KIND=nag_wp) array *Input*

On entry: may be used to specify the optional vectors $u^{\max}$, w^{s} and w^{t} in the space and time monitors (see Section 9).

If an optional vector is not required then all its components should be set to 1.0.

OPTR$(1, j)$, for $j = 1, 2, \ldots, \text{NPDE}$, specifies $u_j^{\max}$, the approximate maximum absolute value of the jth component of u, as used in (4) and (7). OPTR$(1, j) > 0.0$, for $j = 1, 2, \ldots, \text{NPDE}$.

OPTR$(2, j)$, for $j = 1, 2, \ldots, \text{NPDE}$, specifies w_j^{s}, the weighting factors used in the space monitor (see (4)) to indicate the relative importance of the jth component of u on the space monitor. OPTR$(2, j) \geq 0.0$, for $j = 1, 2, \ldots, \text{NPDE}$.

OPTR$(3, j)$, for $j = 1, 2, \ldots, \text{NPDE}$, specifies w_j^{t}, the weighting factors used in the time monitor (see (6)) to indicate the relative importance of the jth component of u on the time monitor. OPTR$(3, j) \geq 0.0$, for $j = 1, 2, \ldots, \text{NPDE}$.

Constraints:

OPTR$(1, j) > 0.0$, for $j = 1, 2, \ldots, \text{NPDE}$;
OPTR$(i, j) \geq 0.0$, for $i = 2, 3$ and $j = 1, 2, \ldots, \text{NPDE}$.

19: RWK(LENRWK) – REAL (KIND=nag_wp) array *Communication Array*

20: LENRWK – INTEGER *Input*

On entry: the dimension of the array RWK as declared in the (sub)program from which D03RAF is called.

The required value of LENRWK cannot be determined exactly in advance, but a suggested value is

$$\text{LENRWK} = maxpts \times \text{NPDE} \times (5 \times l + 18 \times \text{NPDE} + 9) + 2 \times maxpts,$$

where $l = \text{OPTI}(1)$ if $\text{OPTI}(1) \neq 0$ and $l = 3$ otherwise, and $maxpts$ is the expected maximum number of grid points at any one level. If during the execution the supplied value is found to be too small then the routine returns with IFAIL $= 3$ and an estimated required size is printed on the current error message unit (see X04AAF).

Constraint: LENRWK $\geq$ NX $\times$ NY $\times$ NPDE $\times (14 + 18 \times \text{NPDE}) + 2 \times$ NX $\times$ NY (the required size for the initial grid).

21: IWK(LENIWK) – INTEGER array *Communication Array*

On entry: if IND $= 0$, IWK need not be set. Otherwise IWK must remain unchanged from a previous call to D03RAF.

On exit: the following components of the array IWK concern the efficiency of the integration. Here, m is the maximum number of grid levels allowed ($m = \text{OPTI}(1)$ if $\text{OPTI}(1) > 1$ and $m = 3$ otherwise), and l is a grid level taking the values $l = 1, 2, \ldots, nl$, where nl is the number of levels used.

IWK(1)
Contains the number of steps taken in time.

IWK(2)
Contains the number of rejected time steps.

IWK$(2 + l)$
Contains the total number of residual evaluations performed (i.e., the number of times PDEDEF was called) at grid level l.

IWK$(2 + m + l)$
Contains the total number of Jacobian evaluations performed at grid level l.

IWK$(2 + 2 \times m + l)$
Contains the total number of Newton iterations performed at grid level l.

IWK$(2 + 3 \times m + l)$
Contains the total number of linear solver iterations performed at grid level l.

IWK$(2 + 4 \times m + l)$
Contains the maximum number of Newton iterations performed at any one time step at grid level l.

IWK$(2 + 5 \times m + l)$
Contains the maximum number of linear solver iterations performed at any one time step at grid level l.

Note: the total and maximum numbers are cumulative over all calls to D03RAF. If the specified maximum number of Newton or linear solver iterations is exceeded at any stage, then the maximums above are set to the specified maximum plus one.

22: LENIWK – INTEGER *Input*

On entry: the dimension of the array IWK as declared in the (sub)program from which D03RAF is called.

The required value of LENIWK cannot be determined exactly in advance, but a suggested value is

$$\text{LENIWK} = maxpts \times (14 + 5 \times m) + 7 \times m + 2,$$

where $maxpts$ is the expected maximum number of grid points at any one level and $m = \text{OPTI}(1)$ if $\text{OPTI}(1) > 0$ and $m = 3$ otherwise. If during the execution the supplied value is

found to be too small then the routine returns with IFAIL $= 3$ and an estimated required size is printed on the current error message unit (see X04AAF).

Constraint: LENIWK $\geq 19 \times$ NX $\times$ NY $+ 9$ (the required size for the initial grid).

23: LWK(LENLWK) – LOGICAL array *Workspace*

24: LENLWK – INTEGER *Input*

On entry: the dimension of the array LWK as declared in the (sub)program from which D03RAF is called.

The required value of LENLWK cannot be determined exactly in advanced, but a suggested value is

$$\text{LENLWK} = maxpts + 1,$$

where $maxpts$ is the expected maximum number of grid points at any one level. If during the execution the supplied value is found to be too small then the routine returns with IFAIL $= 3$ and an estimated required size is printed on the current error message unit (see X04AAF).

Constraint: LENLWK $\geq$ NX $\times$ NY $+ 1$ (the required size for the initial grid).

25: ITRACE – INTEGER *Input*

On entry: the level of trace information required from D03RAF. ITRACE may take the value -1, 0, 1, 2 or 3.

ITRACE $= -1$
No output is generated.

ITRACE $= 0$
Only warning messages are printed.

ITRACE > 0
Output from the underlying solver is printed on the current advisory message unit (see X04ABF). This output contains details of the time integration, the nonlinear iteration and the linear solver.

If ITRACE < -1, then -1 is assumed and similarly if ITRACE > 3, then 3 is assumed.

The advisory messages are given in greater detail as ITRACE increases. Setting ITRACE $= 1$ allows you to monitor the progress of the integration without possibly excessive information.

26: IND – INTEGER *Input/Output*

On entry: must be set to 0 or 1, alternatively 10 or 11.

IND $= 0$
Starts the integration in time. PDEDEF is assumed to be serial.

IND $= 1$
Continues the integration after an earlier exit from the routine. In this case, only the following parameters may be reset between calls to D03RAF: TOUT, DT, TOLS, TOLT, OPTI, OPTR, ITRACE and IFAIL. PDEDEF is assumed to be serial.

IND $= 10$
Starts the integration in time. PDEDEF is assumed to have been parallelized by you, as described in Section 8. In all other respects, this is equivalent to IND $= 0$.

IND $= 11$
Continues the integration after an earlier exit from the routine. In this case, only the following parameters may be reset between calls to D03RAF: TOUT, DT, TOLS, TOLT, OPTI, OPTR, ITRACE and IFAIL. PDEDEF is assumed to have been parallelized by you, as described in Section 8. In all other respects, this is equivalent to IND $= 1$.

Constraint: $0 \leq$ IND ≤ 1 or $10 \leq$ IND ≤ 11.

On exit: IND $= 1$, if IND on input was 0 or 1, or IND $= 11$, if IND on input was 10 or 11.

Note: for users of serial versions of the NAG Library, it is recommended that you only use IND $= 0$ or 1. See Section 8 for more information on the use of IND.

27: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, NPDE < 1,
or TOUT $\leq$ TS,
or TOUT is too close to TS,
or IND $= 0$ and DT(1) < 0.0,
or DT(i) < 0.0, for $i = 2$ or 3,
or DT(2) $>$ DT(3),
or IND $= 0$ and $0.0 <$ DT(1) $< 10 \times$ ***machine precision*** $\times \max(|\text{TS}|, |\text{TOUT}|)$,
or IND $= 0$ and DT(1) $>$ TOUT $-$ TS,
or IND $= 0$ and DT(1) $<$ DT(2) or DT(1) $>$ DT(3),
or XMIN $\geq$ XMAX,
or XMAX too close to XMIN,
or YMIN $\geq$ YMAX,
or YMAX too close to YMIN,
or NX or NY < 4,
or TOLS or TOLT ≤ 0.0,
or OPTI(1) < 0,
or OPTI(1) > 0 and OPTI(j) < 0, for $j = 2$, 3 or 4,
or OPTR$(1, j) \leq 0.0$, for some $j = 1, 2, \ldots,$ NPDE,
or OPTR$(2, j) < 0.0$, for some $j = 1, 2, \ldots,$ NPDE,
or OPTR$(3, j) < 0.0$, for some $j = 1, 2, \ldots,$ NPDE,
or LENRWK, LENIWK or LENLWK too small for initial grid level,
or IND $\neq 0$ or 1,
or IND $= 1$ on initial entry to D03RAF.

IFAIL $= 2$

The time step size to be attempted is less than the specified minimum size. This may occur following time step failures and subsequent step size reductions caused by one or more of the following:

the requested accuracy could not be achieved, i.e., TOLT is too small,

the maximum number of linear solver iterations, Newton iterations or Jacobian evaluations is too small,

ILU decomposition of the Jacobian matrix could not be performed, possibly due to singularity of the Jacobian.

Setting ITRACE to a higher value may provide further information.

In the latter two cases you are advised to check their problem formulation in PDEDEF and/or BNDARY, and the initial values in PDEIV if appropriate.

IFAIL $= 3$

One or more of the workspace arrays is too small for the required number of grid points. An estimate of the required sizes for the current stage is output, but more space may be required at a later stage.

IFAIL $= 4$

IERR was set to 1 in MONITR, forcing control to be passed back to calling program. Integration was successful as far as T $=$ TS.

IFAIL $= 5$

The integration has been completed but the maximum number of levels specified in OPTI(1) was insufficient at one or more time steps, meaning that the requested space accuracy could not be achieved. To avoid this warning either increase the value of OPTI(1) or decrease the value of TOLS.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

There are three sources of error in the algorithm: space and time discretization, and interpolation (linear) between grid levels. The space and time discretization errors are controlled separately using the arguments TOLS and TOLT described in the following section, and you should test the effects of varying these arguments. Interpolation errors are generally implicitly controlled by the refinement criterion since in areas where interpolation errors are potentially large, the space monitor will also be large. It can be shown that the global spatial accuracy is comparable to that which would be obtained on a uniform grid of the finest grid size. A full error analysis can be found in Trompert and Verwer (1993).

8 Parallelism and Performance

D03RAF is not thread safe and should not be called from a multithreaded user program. Please see Section 3.12.1 in How to Use the NAG Library and its Documentation for more information on thread safety.

D03RAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

D03RAF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

D03RAF requires a user-supplied routine PDEDEF to evaluate the functions F_j, for $j = 1, 2, \ldots, \mathrm{NPDE}$. The parallelism within D03RAF will be more efficient if PDEDEF can also be parallelized. This is often the case, but you must add some OpenMP directives to your version of PDEDEF to implement the parallelism. For example, if the body of code for PDEDEF is as follows (adapted from the first test case in the document for D03RAF):

```
res(1:npts,1:npde) = ut(1:npts,1:npde) - diffusion*(uxx(1:npts,1:          &
   npde)+uyy(1:npts,1:npde)) - damkohler*(one+heat_release-u(1:npts,       &
    1:npde))*exp(-activ_energy/u(1:npts,1:npde))
```

This example can be parallelized, as the updating of RES for each value in the range $1, \ldots, \mathrm{NPTS}$ is independent of every other value. Thus this should be parallelized in OpenMP (using an explicit loop rather than Fortran array syntax) as follows:

```
!$OMP DO
    Do i = 1, npts
       res(i,1:npde) = ut(i,1:npde) -diffusion*(uxx(i,1:npde)+uyy(i,1:npde &
        )) - damkohler*(1.0E0_nag_wp+heat_release-u(i,1:npde))*exp(-       &
         activ_energy/u(i,1:npde))
    End Do
 !$OMP END DO
```

Note that the OpenMP PARALLEL directive must **not** be specified, as the OpenMP DO directive will bind to the PARALLEL region within the D03RAF code. Also note that this assumes the default OpenMP behaviour that all variables are SHARED, except for loop indices that are PRIVATE.

To avoid problems for existing library users, who will not have specified any OpenMP directives in their PDEDEF routine, the default assumption of D03RAF is that PDEDEF has not been parallelized, and executes calls to PDEDEF in serial mode. You must indicate that PDEDEF has been parallelized by setting IND to 10 or 11 as appropriate. See Section 5 for details.

If the code within PDEDEF cannot be parallelized, you must **not** add any OpenMP directives to your code, and must **not** set IND to 10 or 11. If IND is set to 10 or 11 and PDEDEF has not been parallelized, results on multiple threads will be unpredictable and may give rise to incorrect results and/or program crashes or deadlocks. Please contact NAG for advice if required. Overloading IND in this manner is not entirely satisfactory, consequently it is likely that replacement interfaces for D03RAF will be included in a future NAG Library release.

9 Further Comments

9.1 Algorithm Outline

The local uniform grid refinement method is summarised as follows:

1. Initialize the course base grid, an initial solution and an initial time step.
2. Solve the system of PDEs on the current grid with the current time step.
3. If the required accuracy in space and the maximum number of grid levels have not yet been reached:
   (a) Determine new finer grid at forward time level.
   (b) Get solution values at previous time level(s) on new grid.
   (c) Interpolate internal boundary values from old grid at forward time.
   (d) Get initial values for the Newton process at forward time.
   (e) Go to 2.
4. Update the coarser grid solution using the finer grid values.
5. Estimate error in time integration. If time error is acceptable advance time level.

6. Determine new step size then go to 2 with coarse base as current grid.

9.2 Refinement Strategy

For each grid point i a space monitor μ_i^{s} is determined by

$$\mu_i^{\mathrm{s}} = \max_{j=1,\mathrm{NPDE}} \left\{ \gamma_j \left(\left| \Delta x^2 \frac{\partial^2}{\partial x^2} u_j(x_i, y_i, t) \right| + \left| \Delta y^2 \frac{\partial^2}{\partial y^2} u_j(x_i, y_i, t) \right| \right) \right\}, \tag{3}$$

where Δx and Δy are the grid widths in the x and y directions; and x_i, y_i are the x and y coordinates at grid point i. The argument γ_j is obtained from

$$\gamma_j = \frac{w_j^{\mathrm{s}}}{u_j^{\max} \sigma}, \tag{4}$$

where σ is the user-supplied space tolerance; w_j^s is a weighting factor for the relative importance of the jth PDE component on the space monitor; and $u_j^{\max}$ is the approximate maximum absolute value of the jth component. A value for σ must be supplied by you. Values for w_j^{s} and $u_j^{\max}$ must also be supplied but may be set to the value 1.0 if little information about the solution is known.

A new level of refinement is created if

$$\max_i \{\mu_i^{\mathrm{s}}\} > 0.9 \quad \text{or} \quad 1.0, \tag{5}$$

depending on the grid level at the previous step in order to avoid fluctuations in the number of grid levels between time steps. If (5) is satisfied then all grid points for which $\mu_i^s > 0.25$ are flagged and surrounding cells are quartered in size.

No derefinement takes place as such, since at each time step the solution on the base grid is computed first and new finer grids are then created based on the new solution. Hence derefinement occurs implicitly. See Section 9.1.

9.3 Time Integration

The time integration is controlled using a time monitor calculated at each level l up to the maximum level used, given by

$$\mu_l^t = \sqrt{\frac{1}{N} \sum_{j=1}^{\mathrm{NPDE}} w_j^t \sum_{i=1}^{\mathrm{NGPTS}(l)} \left(\frac{\Delta t}{\alpha_{ij}} u_t(x_i, y_i, t) \right)^2} \tag{6}$$

where $\mathrm{NGPTS}(l)$ is the total number of points on grid level l; $N = \mathrm{NGPTS}(l) \times \mathrm{NPDE}$; Δt is the current time step; u_t is the time derivative of u which is approximated by first-order finite differences; w_j^{t} is the time equivalent of the space weighting factor w_j^{s}; and α_{ij} is given by

$$\alpha_{ij} = \tau \left(\frac{u_j^{\max}}{100} + |u(x_i, y_i, t)| \right) \tag{7}$$

where $u_j^{\max}$ is as before, and τ is the user-specified time tolerance.

An integration step is rejected and retried at all levels if

$$\max_l \{\mu_l^t\} > 1.0. \tag{8}$$

10 Example

For this routine two examples are presented, with a main program and two example problems given in Example 1 (EX1) and Example 2 (EX2).

Example 1 (EX1)

This example stems from combustion theory and is a model for a single, one-step reaction of a mixture of two chemicals (see Adjerid and Flaherty (1988)). The PDE for the temperature of the mixture u is

$$\frac{\partial u}{\partial t} = d\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + D(1 + \alpha - u)\exp\left(-\frac{\delta}{u}\right)$$

for $0 \leq x, y \leq 1$ and $t \geq 0$, with initial conditions $u(x, y, 0) = 1$ for $0 \leq x, y \leq 1$, and boundary conditions

$$u_x(0, y, t) = 0, u(1, y, t) = 1 \quad \text{for } 0 \leq y \leq 1,$$
$$u_y(x, 0, t) = 0, u(x, 1, t) = 1 \quad \text{for } 0 \leq x \leq 1.$$

The heat release argument $\alpha = 1$, the Damkohler number $D = R\exp(\delta)/(\alpha\delta)$, the activation energy $\delta = 20$, the reaction rate $R = 5$, and the diffusion argument $d = 0.1$.

For small times the temperature gradually increases in a circular region about the origin, and at about $t = 0.24$ 'ignition' occurs causing the temperature to suddenly jump from near unity to $1 + \alpha$, and a reaction front forms and propagates outwards, becoming steeper. Thus during the solution, just one grid level is used up to the ignition point, then two levels, and then three as the reaction front steepens.

Example 2 (EX2)

This example is taken from a multispecies food web model, in which predator-prey relationships in a spatial domain are simulated (see Brown *et al.* (1994)). In this example there is just one species each of prey and predator, and the two PDEs for the concentrations c_1 and c_2 of the prey and the predator respectively are

$$\frac{\partial c_1}{\partial t} = c_1(b_1 + a_{11}c_1 + a_{12}c_2) + d_1\left(\frac{\partial^2 c_1}{\partial x^2} + \frac{\partial^2 c_1}{\partial y^2}\right),$$
$$0 = c_2(b_2 + a_{21}c_1 + a_{22}c_2) + d_2\left(\frac{\partial^2 c_2}{\partial x^2} + \frac{\partial^2 c_2}{\partial y^2}\right),$$

with

$$a_{11} = a_{22} = -1,$$
$$a_{12} = -0.5 \times 10^{-6}, \text{ and}$$
$$a_{21} = 10^4, \text{ and}$$
$$b_1 = 1 + \alpha xy + \beta\sin(4\pi x)\sin(4\pi y),$$

where $\alpha = 50$ and $\beta = 300$, and $b_2 = -b_1$.

The initial conditions are taken to be simple peaked functions which satisfy the boundary conditions and very nearly satisfy the PDEs:

$$c_1 = 10 + (16x(1 - x)y(1 - y))^2,$$
$$c_2 = b_2 + a_{21}c_1,$$

and the boundary conditions are of Neumann type, i.e., zero normal derivatives everywhere.

During the solution a number of peaks and troughs develop across the domain, and so the number of levels required increases with time. Since the solution varies rapidly in space across the whole of the domain, refinement at intermediate levels tends to occur at all points of the domain.

10.1 Program Text

```
!   D03RAF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module d03rafe_mod

!     D03RAF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
```

```
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                             :: bndry1, bndry2,                    &
                                            compute_wkspace_lens, monit1,      &
                                            monit2, monit_dummy, pdedef1,      &
                                            pdedef2, pdeiv1, pdeiv2,           &
                                            print_statistics
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter    :: alpha = 50.0_nag_wp
      Real (Kind=nag_wp), Parameter    :: beta = 300.0_nag_wp
      Real (Kind=nag_wp), Parameter, Public :: one = 1.0_nag_wp
      Real (Kind=nag_wp), Parameter, Public :: xmax = one
      Real (Kind=nag_wp), Parameter, Public :: ymax = one
      Real (Kind=nag_wp), Parameter, Public :: zero = 0.0_nag_wp
      Real (Kind=nag_wp), Parameter, Public :: xmin = zero
      Real (Kind=nag_wp), Parameter, Public :: ymin = zero
      Integer, Parameter, Public       :: itrace = 0, nin = 5, nout = 6,     &
                                          npde1 = 1, npde2 = 2
    Contains
      Subroutine pdeiv1(npts,npde,t,x,y,u)

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: t
        Integer, Intent (In)           :: npde, npts
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out) :: u(npts,npde)
        Real (Kind=nag_wp), Intent (In) :: x(npts), y(npts)
!       .. Executable Statements ..
        u(1:npts,1:npde) = one
        Return
      End Subroutine pdeiv1
      Subroutine pdedef1(npts,npde,t,x,y,u,ut,ux,uy,uxx,uxy,uyy,res)

!       .. Parameters ..
        Real (Kind=nag_wp), Parameter  :: activ_energy = 20.0_nag_wp
        Real (Kind=nag_wp), Parameter  :: diffusion = 0.1_nag_wp
        Real (Kind=nag_wp), Parameter  :: heat_release = 1.0_nag_wp
        Real (Kind=nag_wp), Parameter  :: reaction_rate = 5.0_nag_wp
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: t
        Integer, Intent (In)           :: npde, npts
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out) :: res(npts,npde)
        Real (Kind=nag_wp), Intent (In) :: u(npts,npde), ut(npts,npde),       &
                                           ux(npts,npde), uxx(npts,npde),     &
                                           uxy(npts,npde), uy(npts,npde),     &
                                           uyy(npts,npde), x(npts), y(npts)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: damkohler
        Integer                        :: i
!       .. Intrinsic Procedures ..
        Intrinsic                      :: exp
!       .. Executable Statements ..
        damkohler = reaction_rate*exp(activ_energy)/                          &
          (heat_release*activ_energy)
        !$Omp Do
        Do i = 1, npts
          res(i,1:npde) = ut(i,1:npde) - diffusion*(uxx(i,1:npde)+uyy(i,1:npde &
            )) - damkohler*(1.0E0_nag_wp+heat_release-u(i,1:npde))*exp(       &
            -activ_energy/u(i,1:npde))
        End Do
        !$Omp End Do
        Return
      End Subroutine pdedef1
      Subroutine bndry1(npts,npde,t,x,y,u,ut,ux,uy,nbpts,lbnd,res)

!       .. Use Statements ..
        Use nag_library, Only: x02ajf
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: t
```

```
        Integer, Intent (In)           :: nbpts, npde, npts
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Inout) :: res(npts,npde)
        Real (Kind=nag_wp), Intent (In) :: u(npts,npde), ut(npts,npde),        &
                                           ux(npts,npde), uy(npts,npde),        &
                                           x(npts), y(npts)
        Integer, Intent (In)           :: lbnd(nbpts)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: tol
        Integer                        :: i, j
!       .. Intrinsic Procedures ..
        Intrinsic                      :: abs
!       .. Executable Statements ..
        tol = 10._nag_wp*x02ajf()

        Do i = 1, nbpts
          j = lbnd(i)

          If (abs(x(j))<=tol) Then
            res(j,1:npde) = ux(j,1:npde)
          Else If (abs(x(j)-one)<=tol) Then
            res(j,1:npde) = u(j,1:npde) - one
          Else If (abs(y(j))<=tol) Then
            res(j,1:npde) = uy(j,1:npde)
          Else If (abs(y(j)-one)<=tol) Then
            res(j,1:npde) = u(j,1:npde) - one
          End If
        End Do

        Return
      End Subroutine bndry1
      Subroutine monit1(npde,t,dt,dtnew,tlast,nlev,ngpts,xpts,ypts,lsol,sol,   &
        ierr)

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: dt, dtnew, t
        Integer, Intent (Inout)        :: ierr
        Integer, Intent (In)           :: nlev, npde
        Logical, Intent (In)           :: tlast
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In) :: sol(*), xpts(*), ypts(*)
        Integer, Intent (In)           :: lsol(nlev), ngpts(nlev)
!       .. Local Scalars ..
        Integer                        :: i, ipsol, k, level, npts
!       .. Intrinsic Procedures ..
        Intrinsic                      :: sum
!       .. Executable Statements ..
        If (tlast) Then

!         Print solution

          level = nlev
          Write (nout,99999) level, t
          Write (nout,99998)
          npts = ngpts(level)
          ipsol = lsol(level)
          k = sum(ngpts(1:nlev-1))
          Do i = 1, npts, 4
            Write (nout,99997) xpts(k+i), ypts(k+i), sol(ipsol+i)
          End Do

          Write (nout,*)

        End If

        Return
99999   Format (1X,'Solution at every 4th grid point in level',I10,            &
          ' at time ',F8.4,':')
99998   Format (1X,/,7X,'x',10X,'y',8X,'approx u',/)
99997   Format (1X,1P,E11.4,2(1X,1P,E11.3))
      End Subroutine monit1
```

```
      Subroutine pdeiv2(npts,npde,t,x,y,u)

!       .. Use Statements ..
        Use nag_library, Only: x01aaf
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: t
        Integer, Intent (In)           :: npde, npts
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out) :: u(npts,npde)
        Real (Kind=nag_wp), Intent (In) :: x(npts), y(npts)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: fourpi
!       .. Intrinsic Procedures ..
        Intrinsic                      :: sin
!       .. Executable Statements ..
        fourpi = 4.0_nag_wp*x01aaf(fourpi)
        u(1:npts,1) = 10.0_nag_wp + (16.0_nag_wp*x(1:npts)*(one-x(1:npts))*y(1 &
          :npts)*(one-y(1:npts)))**2
        u(1:npts,2) = -one - alpha*x(1:npts)*y(1:npts) -                       &
          beta*sin(fourpi*x(1:npts))*sin(fourpi*y(1:npts)) +                   &
          1.0E4_nag_wp*u(1:npts,1)
        Return
      End Subroutine pdeiv2
      Subroutine pdedef2(npts,npde,t,x,y,u,ut,ux,uy,uxx,uxy,uyy,res)

!       .. Use Statements ..
        Use nag_library, Only: x01aaf
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: t
        Integer, Intent (In)           :: npde, npts
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out) :: res(npts,npde)
        Real (Kind=nag_wp), Intent (In) :: u(npts,npde), ut(npts,npde),        &
                                           ux(npts,npde), uxx(npts,npde),      &
                                           uxy(npts,npde), uy(npts,npde),      &
                                           uyy(npts,npde), x(npts), y(npts)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: b1, fourpi
        Integer                        :: i
!       .. Intrinsic Procedures ..
        Intrinsic                      :: sin
!       .. Executable Statements ..
        fourpi = 4.0_nag_wp*x01aaf(fourpi)
        Do i = 1, npts
          b1 = 1.0E0_nag_wp + alpha*x(i)*y(i) + beta*sin(fourpi*x(i))*sin(     &
            fourpi*y(i))
          res(i,1) = ut(i,1) - (uxx(i,1)+uyy(i,1)) -                           &
            u(i,1)*(b1-u(i,1)-0.5E-6_nag_wp*u(i,2))
          res(i,2) = -0.05E0_nag_wp*(uxx(i,2)+uyy(i,2)) -                      &
            u(i,2)*(-b1+1.0E4_nag_wp*u(i,1)-u(i,2))
        End Do
        Return
      End Subroutine pdedef2
      Subroutine bndry2(npts,npde,t,x,y,u,ut,ux,uy,nbpts,lbnd,res)

!       .. Use Statements ..
        Use nag_library, Only: x02ajf
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: t
        Integer, Intent (In)           :: nbpts, npde, npts
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Inout) :: res(npts,npde)
        Real (Kind=nag_wp), Intent (In) :: u(npts,npde), ut(npts,npde),        &
                                           ux(npts,npde), uy(npts,npde),       &
                                           x(npts), y(npts)
        Integer, Intent (In)           :: lbnd(nbpts)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: tol
        Integer                        :: i, j
!       .. Intrinsic Procedures ..
        Intrinsic                      :: abs
```

```
!       .. Executable Statements ..
        tol = 10.0_nag_wp*x02ajf()

        Do i = 1, nbpts
          j = lbnd(i)

          If (abs(x(j))<=tol .Or. abs(x(j)-one)<=tol) Then
            res(j,1:npde) = ux(j,1:npde)
          Else If (abs(y(j))<=tol .Or. abs(y(j)-one)<=tol) Then
            res(j,1:npde) = uy(j,1:npde)
          End If
        End Do

        Return
      End Subroutine bndry2
      Subroutine monit2(npde,t,dt,dtnew,tlast,nlev,ngpts,xpts,ypts,lsol,sol,   &
        ierr)

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: dt, dtnew, t
        Integer, Intent (Inout)         :: ierr
        Integer, Intent (In)            :: nlev, npde
        Logical, Intent (In)            :: tlast
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In) :: sol(*), xpts(*), ypts(*)
        Integer, Intent (In)            :: lsol(nlev), ngpts(nlev)
!       .. Local Scalars ..
        Integer                         :: i, ipsol, k, level, npts
!       .. Intrinsic Procedures ..
        Intrinsic                       :: sum
!       .. Executable Statements ..
        If (tlast) Then

!         Print solution

          level = nlev
          Write (nout,99999) level, t
          Write (nout,99998)
          npts = ngpts(level)
          ipsol = lsol(level)
          k = sum(ngpts(1:nlev-1))

          Do i = 1, npts, 2
            Write (nout,99997) xpts(k+i), ypts(k+i), sol(ipsol+i),             &
              sol(ipsol+npts+i)
          End Do

          Write (nout,*)

        End If

        Return
99999   Format (1X,'Solution at every 2nd grid point in level',I10,            &
          ' at time ',F8.4,':')
99998   Format (1X,/,7X,'x',10X,'y',9X,'approx c1',3X,'approx c2',/)
99997   Format (1P,2(1X,E11.3),2X,E11.3,2X,E11.3)
      End Subroutine monit2
      Subroutine monit_dummy(npde,t,dt,dtnew,tlast,nlev,ngpts,xpts,ypts,lsol,  &
        sol,ierr)

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: dt, dtnew, t
        Integer, Intent (Inout)         :: ierr
        Integer, Intent (In)            :: nlev, npde
        Logical, Intent (In)            :: tlast
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In) :: sol(*), xpts(*), ypts(*)
        Integer, Intent (In)            :: lsol(nlev), ngpts(nlev)
!       .. Executable Statements ..
        Return
      End Subroutine monit_dummy
```

```
      Subroutine compute_wkspace_lens(maxlev,npde,maxpts,lenrwk,leniwk,lenlwk)

!       Returns suitable workspace lengths for the two problems
!       being solved, based on trial-and-error.

!       .. Scalar Arguments ..
        Integer, Intent (Out)          :: leniwk, lenlwk, lenrwk
        Integer, Intent (In)           :: maxlev, maxpts, npde
!       .. Executable Statements ..
        lenrwk = 2*maxpts*npde*(5*maxlev+18*npde+9) + 2*maxpts
        leniwk = 2*maxpts*(14+5*maxlev) + 7*maxlev + 2
        lenlwk = 2*maxpts + 400
        Return
      End Subroutine compute_wkspace_lens
      Subroutine print_statistics(ts,iwk,maxlev)

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: ts
        Integer, Intent (In)           :: maxlev
!       .. Array Arguments ..
        Integer, Intent (In)           :: iwk(6*maxlev+2)
!       .. Local Scalars ..
        Integer                        :: i, j
!       .. Local Arrays ..
        Integer                        :: istats(4)
!       .. Executable Statements ..
        Write (nout,'(1X,A)') 'Statistics:'
        Write (nout,99999) 'Time = ', ts
        Write (nout,99998) 'Total number of accepted timesteps =', iwk(1)
        Write (nout,99998) 'Total number of rejected timesteps =', iwk(2)
        Write (nout,'(1X,4(/,A))')                                              &
          '            Total  number (rounded) of     ',                      &
          '          Residual  Jacobian    Newton   Lin sys',                  &
          '             evals     evals     iters     iters', ' At level '

        Do j = 1, maxlev
          If (iwk(j+2)/=0) Then
            istats(1:4) = iwk(j+2:j+2+3*maxlev:maxlev)
            Call round_statistics(istats)
            Write (nout,99997) j, istats(1:4)
          End If
        End Do

        Write (nout,'(1X,3(/,A))') '             Maximum  number of   ',       &
          '              Newton iters    Lin sys iters ', ' At level '

        Do j = 1, maxlev
          If (iwk(j+2)/=0) Then
            Write (nout,99996) j, (iwk(j+2+i*maxlev),i=4,5)
          End If
        End Do
        Write (nout,*)

        Return
99999   Format (1X,A,F8.4)
99998   Format (1X,A,I5)
99997   Format (I8,4I10)
99996   Format (I8,2I14)
      End Subroutine print_statistics
      Subroutine round_statistics(istat)

!       .. Array Arguments ..
        Integer, Intent (Inout)        :: istat(4)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: lt
        Integer                        :: i, k
!       .. Intrinsic Procedures ..
        Intrinsic                      :: int, log, real
!       .. Executable Statements ..
        lt = log(10.0_nag_wp)
        Do i = 1, 4
```

```
          k = int(log(real(istat(i),kind=nag_wp))/lt)
          k = 10**k
          istat(i) = k*((istat(i)+k/2)/k)
        End Do
      End Subroutine round_statistics
    End Module d03rafe_mod
    Program d03rafe

!     D03RAF Example Main Program

!     .. Use Statements ..
      Use d03rafe_mod, Only: nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Executable Statements ..
      Write (nout,*) 'D03RAF Example Program Results'

      Call ex1

      Call ex2

    Contains
      Subroutine ex1

!       .. Use Statements ..
        Use nag_library, Only: d03raf, nag_wp
        Use d03rafe_mod, Only: bndry1, compute_wkspace_lens, itrace, monit1,   &
                               monit_dummy, nin, npde1, one, pdedef1, pdeiv1,  &
                               print_statistics, xmax, xmin, ymax, ymin, zero
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: tols, tolt, tout, ts
        Integer                        :: i, ifail, ind, leniwk, lenlwk,       &
                                          lenrwk, maxlev, npde, npts, nx, ny
!       .. Local Arrays ..
        Real (Kind=nag_wp)             :: dt(3), twant(2)
        Real (Kind=nag_wp), Allocatable :: optr(:,:), rwk(:)
        Integer, Allocatable           :: iwk(:)
        Integer                        :: opti(4)
        Logical, Allocatable           :: lwk(:)
!       .. Intrinsic Procedures ..
        Intrinsic                      :: max
!       .. Executable Statements ..
        Write (nout,*)
        Write (nout,*)
        Write (nout,*) 'Example 1'
        Write (nout,*)
!       Skip heading in data file
        Read (nin,*)
        Read (nin,*) npts

        npde = npde1

        dt(1:3) = (/0.1E-2_nag_wp,zero,zero/)
        twant(1:2) = (/0.24_nag_wp,0.25_nag_wp/)
        ts = zero

!       Specify that we are starting the integration in time (ind = 0
!       normally).
!       Note: we have parallelized the loop in the function pdedef1 using
!       OpenMP so set alternative value of ind to indicate that this can be
!       run in parallel if we are using a multithreaded implementation.
!       Either option is OK for serial NAG Library implementations from
!       Mark 25 onwards.
        ind = 10

        nx = 41
        ny = 41
        tols = 0.5_nag_wp
        tolt = 0.01_nag_wp
        opti(1) = 6
        opti(2:4) = 0
```

```
        maxlev = max(opti(1),3)

        Call compute_wkspace_lens(maxlev,npde,npts,lenrwk,leniwk,lenlwk)

        Allocate (rwk(lenrwk),iwk(leniwk),lwk(lenlwk),optr(3,npde))
        optr(1:3,1:npde) = one

        Do i = 1, 2
          tout = twant(i)

!         ifail: behaviour on error exit
!                =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
          ifail = 0
          If (i==1) Then
!           Use monit_dummy to avoid output first time around
            Call d03raf(npde,ts,tout,dt,xmin,xmax,ymin,ymax,nx,ny,tols,tolt,   &
              pdedef1,bndry1,pdeiv1,monit_dummy,opti,optr,rwk,lenrwk,iwk,      &
              leniwk,lwk,lenlwk,itrace,ind,ifail)
          Else
            Call d03raf(npde,ts,tout,dt,xmin,xmax,ymin,ymax,nx,ny,tols,tolt,   &
              pdedef1,bndry1,pdeiv1,monit1,opti,optr,rwk,lenrwk,iwk,leniwk,    &
              lwk,lenlwk,itrace,ind,ifail)
          End If

          Call print_statistics(ts,iwk,maxlev)

        End Do

        Return
      End Subroutine ex1
      Subroutine ex2

!       .. Use Statements ..
        Use nag_library, Only: d03raf, nag_wp
        Use d03rafe_mod, Only: bndry2, compute_wkspace_lens, itrace, monit2,   &
                               monit_dummy, nin, npde2, one, pdedef2, pdeiv2,  &
                               print_statistics, xmax, xmin, ymax, ymin, zero
!       .. Parameters ..
        Integer, Parameter             :: opti(4) = (/4,0,0,0/)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: tols, tolt, tout, ts
        Integer                        :: i, ifail, ind, leniwk, lenlwk,       &
                                          lenrwk, maxlev, npde, npts, nx, ny
!       .. Local Arrays ..
        Real (Kind=nag_wp)             :: dt(3), twant(2)
        Real (Kind=nag_wp), Allocatable :: optr(:,:), rwk(:)
        Integer, Allocatable           :: iwk(:)
        Logical, Allocatable           :: lwk(:)
!       .. Intrinsic Procedures ..
        Intrinsic                      :: max
!       .. Executable Statements ..
        Write (nout,*)
        Write (nout,*)
        Write (nout,*) 'Example 2'
        Write (nout,*)
        Read (nin,*)
        Read (nin,*) npts

        npde = npde2
        dt(1:3) = (/0.5E-3_nag_wp,1.0E-6_nag_wp,zero/)
        twant(1:2) = (/0.01_nag_wp,0.025_nag_wp/)
        ts = zero

!       Specify that we are starting the integration in time (ind = 0
!       normally).
!       Note: In this second example we have not added OpenMP directives to
!       parallelize the loop in the function pdedef2. Thus the alternative
!       ind=10 must not be specified here, as this will not function correctly
!       if a multithreaded implementation of the NAG Library is used. Adding
!       OpenMP to pdedef2, that would enable ind=10 to be used here safely, is
!       left as an exercise for the reader.
```

```
        ind = 0

        nx = 11
        ny = 11
        tols = 0.1_nag_wp
        tolt = 0.1_nag_wp
        maxlev = max(opti(1),3)

        Call compute_wkspace_lens(maxlev,npde,npts,lenrwk,leniwk,lenlwk)

        Allocate (rwk(lenrwk),iwk(leniwk),lwk(lenlwk),optr(3,npde))
        optr(1,1:npde) = (/250.0_nag_wp,1.5E6_nag_wp/)
        optr(2:3,1:npde) = one

        Do i = 1, 2
          tout = twant(i)

          ifail = 0
          If (i==1) Then
!           Use monit_dummy to avoid output first time around
            Call d03raf(npde,ts,tout,dt,xmin,xmax,ymin,ymax,nx,ny,tols,tolt,   &
              pdedef2,bndry2,pdeiv2,monit_dummy,opti,optr,rwk,lenrwk,iwk,      &
              leniwk,lwk,lenlwk,itrace,ind,ifail)
          Else
            Call d03raf(npde,ts,tout,dt,xmin,xmax,ymin,ymax,nx,ny,tols,tolt,   &
              pdedef2,bndry2,pdeiv2,monit2,opti,optr,rwk,lenrwk,iwk,leniwk,    &
              lwk,lenlwk,itrace,ind,ifail)
          End If

          Call print_statistics(ts,iwk,maxlev)

        End Do

        Return
      End Subroutine ex2
    End Program d03rafe
```

10.2 Program Data

```
D03RAF Example Program Data
 2000                   : npts

 2000                   : npts
```

10.3 Program Results

```
 D03RAF Example Program Results


 Example 1

 Statistics:
 Time =   0.2400
 Total number of accepted timesteps =   77
 Total number of rejected timesteps =    0

             Total  number (rounded) of
          Residual  Jacobian    Newton   Lin sys
             evals     evals     iters     iters
 At level
       1       600        80       200       200

             Maximum  number of
              Newton iters    Lin sys iters
 At level
       1             2             3

 Solution at every 4th grid point in level         2 at time   0.2500:

       x          y        approx u
```

```
 6.7500E-01   0.000E+00   2.000E+00
 7.2500E-01   0.000E+00   1.980E+00
 7.7500E-01   0.000E+00   1.342E+00
 8.2500E-01   0.000E+00   1.190E+00
 7.1250E-01   1.250E-02   1.996E+00
 7.6250E-01   1.250E-02   1.468E+00
 8.1250E-01   1.250E-02   1.213E+00
 7.0000E-01   2.500E-02   1.999E+00
 7.5000E-01   2.500E-02   1.625E+00
 8.0000E-01   2.500E-02   1.240E+00
 6.8750E-01   3.750E-02   2.000E+00
 7.3750E-01   3.750E-02   1.870E+00
 7.8750E-01   3.750E-02   1.283E+00
 6.7500E-01   5.000E-02   2.000E+00
 7.2500E-01   5.000E-02   1.980E+00
 7.7500E-01   5.000E-02   1.342E+00
 8.2500E-01   5.000E-02   1.190E+00
 7.1250E-01   6.250E-02   1.996E+00
 7.6250E-01   6.250E-02   1.468E+00
 8.1250E-01   6.250E-02   1.213E+00
 7.0000E-01   7.500E-02   1.999E+00
 7.5000E-01   7.500E-02   1.625E+00
 8.0000E-01   7.500E-02   1.240E+00
 6.8750E-01   8.750E-02   2.000E+00
 7.3750E-01   8.750E-02   1.870E+00
 7.8750E-01   8.750E-02   1.283E+00
 6.7500E-01   1.000E-01   2.000E+00
 7.2500E-01   1.000E-01   1.980E+00
 7.7500E-01   1.000E-01   1.342E+00
 8.2500E-01   1.000E-01   1.190E+00
 7.1250E-01   1.125E-01   1.996E+00
 7.6250E-01   1.125E-01   1.468E+00
 8.1250E-01   1.125E-01   1.213E+00
 7.0000E-01   1.250E-01   1.999E+00
 7.5000E-01   1.250E-01   1.625E+00
 8.0000E-01   1.250E-01   1.240E+00
 6.8750E-01   1.375E-01   2.000E+00
 7.3750E-01   1.375E-01   1.870E+00
 7.8750E-01   1.375E-01   1.283E+00
 6.7500E-01   1.500E-01   2.000E+00
 7.2500E-01   1.500E-01   1.980E+00
 7.7500E-01   1.500E-01   1.341E+00
 8.2500E-01   1.500E-01   1.190E+00
 7.1250E-01   1.625E-01   1.995E+00
 7.6250E-01   1.625E-01   1.467E+00
 8.1250E-01   1.625E-01   1.213E+00
 7.0000E-01   1.750E-01   1.999E+00
 7.5000E-01   1.750E-01   1.624E+00
 8.0000E-01   1.750E-01   1.240E+00
 6.8750E-01   1.875E-01   2.000E+00
 7.3750E-01   1.875E-01   1.869E+00
 7.8750E-01   1.875E-01   1.282E+00
 6.7500E-01   2.000E-01   2.000E+00
 7.2500E-01   2.000E-01   1.980E+00
 7.7500E-01   2.000E-01   1.341E+00
 8.2500E-01   2.000E-01   1.189E+00
 7.1250E-01   2.125E-01   1.995E+00
 7.6250E-01   2.125E-01   1.465E+00
 8.1250E-01   2.125E-01   1.212E+00
 7.0000E-01   2.250E-01   1.999E+00
 7.5000E-01   2.250E-01   1.621E+00
 8.0000E-01   2.250E-01   1.240E+00
 6.8750E-01   2.375E-01   2.000E+00
 7.3750E-01   2.375E-01   1.869E+00
 7.8750E-01   2.375E-01   1.282E+00
 6.7500E-01   2.500E-01   2.000E+00
 7.2500E-01   2.500E-01   1.980E+00
 7.7500E-01   2.500E-01   1.340E+00
 8.2500E-01   2.500E-01   1.189E+00
 7.1250E-01   2.625E-01   1.995E+00
```

```
  7.6250E-01   2.625E-01   1.462E+00
  8.1250E-01   2.625E-01   1.212E+00
  7.0000E-01   2.750E-01   1.999E+00
  7.5000E-01   2.750E-01   1.616E+00
  8.0000E-01   2.750E-01   1.239E+00
  6.8750E-01   2.875E-01   2.000E+00
  7.3750E-01   2.875E-01   1.866E+00
  7.8750E-01   2.875E-01   1.281E+00
  6.7500E-01   3.000E-01   2.000E+00
  7.2500E-01   3.000E-01   1.979E+00
  7.7500E-01   3.000E-01   1.337E+00
  8.2500E-01   3.000E-01   1.189E+00
  7.1250E-01   3.125E-01   1.995E+00
  7.6250E-01   3.125E-01   1.455E+00
  8.1250E-01   3.125E-01   1.212E+00
  7.0000E-01   3.250E-01   1.999E+00
  7.5000E-01   3.250E-01   1.605E+00
  8.0000E-01   3.250E-01   1.239E+00
  6.8750E-01   3.375E-01   2.000E+00
  7.3750E-01   3.375E-01   1.862E+00
  7.8750E-01   3.375E-01   1.279E+00
  6.7500E-01   3.500E-01   2.000E+00
  7.2500E-01   3.500E-01   1.977E+00
  7.7500E-01   3.500E-01   1.332E+00
  8.2500E-01   3.500E-01   1.189E+00
  7.1250E-01   3.625E-01   1.993E+00
  7.6250E-01   3.625E-01   1.442E+00
  8.1250E-01   3.625E-01   1.211E+00
  7.0000E-01   3.750E-01   1.999E+00
  7.5000E-01   3.750E-01   1.585E+00
  8.0000E-01   3.750E-01   1.236E+00
  6.8750E-01   3.875E-01   2.000E+00
  7.3750E-01   3.875E-01   1.849E+00
  7.8750E-01   3.875E-01   1.274E+00
  6.7500E-01   4.000E-01   2.000E+00
  7.2500E-01   4.000E-01   1.972E+00
  7.7500E-01   4.000E-01   1.324E+00
  8.2500E-01   4.000E-01   1.187E+00
  7.1250E-01   4.125E-01   1.990E+00
  7.6250E-01   4.125E-01   1.420E+00
  8.1250E-01   4.125E-01   1.209E+00
  7.0000E-01   4.250E-01   1.997E+00
  7.5000E-01   4.250E-01   1.549E+00
  8.0000E-01   4.250E-01   1.233E+00
  6.8750E-01   4.375E-01   1.999E+00
  7.3750E-01   4.375E-01   1.813E+00
  7.8750E-01   4.375E-01   1.267E+00
  6.7500E-01   4.500E-01   2.000E+00
  7.2500E-01   4.500E-01   1.952E+00
  7.7500E-01   4.500E-01   1.310E+00
  8.2500E-01   4.500E-01   1.185E+00
  7.1250E-01   4.625E-01   1.976E+00
  7.6250E-01   4.625E-01   1.387E+00
  8.1250E-01   4.625E-01   1.206E+00
  6.7500E-01   4.750E-01   2.000E+00
  7.2500E-01   4.750E-01   1.924E+00
  7.7500E-01   4.750E-01   1.301E+00
  8.2500E-01   4.750E-01   1.184E+00
  6.8750E-01   4.875E-01   1.999E+00
  7.3750E-01   4.875E-01   1.714E+00
  7.8750E-01   4.875E-01   1.257E+00
  6.5000E-01   5.000E-01   2.000E+00
  7.0000E-01   5.000E-01   1.991E+00
  7.5000E-01   5.000E-01   1.454E+00
  8.0000E-01   5.000E-01   1.224E+00
  6.6250E-01   5.125E-01   2.000E+00
  7.1250E-01   5.125E-01   1.932E+00
  7.6250E-01   5.125E-01   1.346E+00
  6.5000E-01   5.250E-01   2.000E+00
  7.0000E-01   5.250E-01   1.986E+00
  7.5000E-01   5.250E-01   1.414E+00
```

```
 8.0000E-01   5.250E-01   1.219E+00
 6.8750E-01   5.375E-01   1.996E+00
 7.3750E-01   5.375E-01   1.545E+00
 7.8750E-01   5.375E-01   1.243E+00
 6.5000E-01   5.500E-01   2.000E+00
 7.0000E-01   5.500E-01   1.978E+00
 7.5000E-01   5.500E-01   1.373E+00
 8.0000E-01   5.500E-01   1.214E+00
 6.6250E-01   5.625E-01   1.999E+00
 7.1250E-01   5.625E-01   1.843E+00
 7.6250E-01   5.625E-01   1.302E+00
 6.2500E-01   5.750E-01   2.000E+00
 6.7500E-01   5.750E-01   1.995E+00
 7.2500E-01   5.750E-01   1.545E+00
 7.7500E-01   5.750E-01   1.256E+00
 6.3750E-01   5.875E-01   2.000E+00
 6.8750E-01   5.875E-01   1.954E+00
 7.3750E-01   5.875E-01   1.392E+00
 7.8750E-01   5.875E-01   1.226E+00
 6.2500E-01   6.000E-01   2.000E+00
 6.7500E-01   6.000E-01   1.984E+00
 7.2500E-01   6.000E-01   1.443E+00
 7.7500E-01   6.000E-01   1.243E+00
 6.1250E-01   6.125E-01   2.000E+00
 6.6250E-01   6.125E-01   1.988E+00
 7.1250E-01   6.125E-01   1.531E+00
 7.6250E-01   6.125E-01   1.263E+00
 5.5000E-01   6.250E-01   2.000E+00
 6.0000E-01   6.250E-01   2.000E+00
 6.5000E-01   6.250E-01   1.993E+00
 7.0000E-01   6.250E-01   1.577E+00
 7.5000E-01   6.250E-01   1.280E+00
 8.0000E-01   6.250E-01   1.194E+00
 5.8750E-01   6.375E-01   2.000E+00
 6.3750E-01   6.375E-01   1.992E+00
 6.8750E-01   6.375E-01   1.672E+00
 7.3750E-01   6.375E-01   1.300E+00
 4.7500E-01   6.500E-01   2.000E+00
 5.2500E-01   6.500E-01   2.000E+00
 5.7500E-01   6.500E-01   2.000E+00
 6.2500E-01   6.500E-01   1.993E+00
 6.7500E-01   6.500E-01   1.680E+00
 7.2500E-01   6.500E-01   1.314E+00
 7.7500E-01   6.500E-01   1.218E+00
 5.1250E-01   6.625E-01   2.000E+00
 5.6250E-01   6.625E-01   1.999E+00
 6.1250E-01   6.625E-01   1.988E+00
 6.6250E-01   6.625E-01   1.721E+00
 7.1250E-01   6.625E-01   1.331E+00
 7.6250E-01   6.625E-01   1.229E+00
 2.5000E-02   6.750E-01   2.000E+00
 7.5000E-02   6.750E-01   2.000E+00
 1.2500E-01   6.750E-01   2.000E+00
 1.7500E-01   6.750E-01   2.000E+00
 2.2500E-01   6.750E-01   2.000E+00
 2.7500E-01   6.750E-01   2.000E+00
 3.2500E-01   6.750E-01   2.000E+00
 3.7500E-01   6.750E-01   2.000E+00
 4.2500E-01   6.750E-01   2.000E+00
 4.7500E-01   6.750E-01   2.000E+00
 5.2500E-01   6.750E-01   2.000E+00
 5.7500E-01   6.750E-01   1.995E+00
 6.2500E-01   6.750E-01   1.932E+00
 6.7500E-01   6.750E-01   1.447E+00
 7.2500E-01   6.750E-01   1.277E+00
 7.7500E-01   6.750E-01   1.205E+00
 3.7500E-02   6.875E-01   2.000E+00
 8.7500E-02   6.875E-01   2.000E+00
 1.3750E-01   6.875E-01   2.000E+00
 1.8750E-01   6.875E-01   2.000E+00
 2.3750E-01   6.875E-01   2.000E+00
```

```
 2.8750E-01   6.875E-01   2.000E+00
 3.3750E-01   6.875E-01   2.000E+00
 3.8750E-01   6.875E-01   2.000E+00
 4.3750E-01   6.875E-01   1.999E+00
 4.8750E-01   6.875E-01   1.999E+00
 5.3750E-01   6.875E-01   1.996E+00
 5.8750E-01   6.875E-01   1.954E+00
 6.3750E-01   6.875E-01   1.672E+00
 6.8750E-01   6.875E-01   1.343E+00
 7.3750E-01   6.875E-01   1.244E+00
 2.5000E-02   7.000E-01   1.999E+00
 7.5000E-02   7.000E-01   1.999E+00
 1.2500E-01   7.000E-01   1.999E+00
 1.7500E-01   7.000E-01   1.999E+00
 2.2500E-01   7.000E-01   1.999E+00
 2.7500E-01   7.000E-01   1.999E+00
 3.2500E-01   7.000E-01   1.999E+00
 3.7500E-01   7.000E-01   1.999E+00
 4.2500E-01   7.000E-01   1.997E+00
 4.7500E-01   7.000E-01   1.994E+00
 5.2500E-01   7.000E-01   1.986E+00
 5.7500E-01   7.000E-01   1.941E+00
 6.2500E-01   7.000E-01   1.577E+00
 6.7500E-01   7.000E-01   1.336E+00
 7.2500E-01   7.000E-01   1.247E+00
 1.2500E-02   7.125E-01   1.996E+00
 6.2500E-02   7.125E-01   1.996E+00
 1.1250E-01   7.125E-01   1.995E+00
 1.6250E-01   7.125E-01   1.995E+00
 2.1250E-01   7.125E-01   1.995E+00
 2.6250E-01   7.125E-01   1.995E+00
 3.1250E-01   7.125E-01   1.995E+00
 3.6250E-01   7.125E-01   1.993E+00
 4.1250E-01   7.125E-01   1.990E+00
 4.6250E-01   7.125E-01   1.976E+00
 5.1250E-01   7.125E-01   1.932E+00
 5.6250E-01   7.125E-01   1.843E+00
 6.1250E-01   7.125E-01   1.531E+00
 6.6250E-01   7.125E-01   1.331E+00
 7.1250E-01   7.125E-01   1.250E+00
 2.5000E-02   7.250E-01   1.980E+00
 7.5000E-02   7.250E-01   1.980E+00
 1.2500E-01   7.250E-01   1.980E+00
 1.7500E-01   7.250E-01   1.980E+00
 2.2500E-01   7.250E-01   1.980E+00
 2.7500E-01   7.250E-01   1.979E+00
 3.2500E-01   7.250E-01   1.978E+00
 3.7500E-01   7.250E-01   1.975E+00
 4.2500E-01   7.250E-01   1.965E+00
 4.7500E-01   7.250E-01   1.924E+00
 5.2500E-01   7.250E-01   1.772E+00
 5.7500E-01   7.250E-01   1.545E+00
 6.2500E-01   7.250E-01   1.366E+00
 6.7500E-01   7.250E-01   1.277E+00
 7.2500E-01   7.250E-01   1.222E+00
 3.7500E-02   7.375E-01   1.870E+00
 8.7500E-02   7.375E-01   1.870E+00
 1.3750E-01   7.375E-01   1.870E+00
 1.8750E-01   7.375E-01   1.869E+00
 2.3750E-01   7.375E-01   1.868E+00
 2.8750E-01   7.375E-01   1.866E+00
 3.3750E-01   7.375E-01   1.862E+00
 3.8750E-01   7.375E-01   1.849E+00
 4.3750E-01   7.375E-01   1.813E+00
 4.8750E-01   7.375E-01   1.714E+00
 5.3750E-01   7.375E-01   1.545E+00
 5.8750E-01   7.375E-01   1.392E+00
 6.3750E-01   7.375E-01   1.300E+00
 6.8750E-01   7.375E-01   1.244E+00
 2.5000E-02   7.500E-01   1.625E+00
 7.5000E-02   7.500E-01   1.625E+00
```

```
 1.2500E-01   7.500E-01   1.625E+00
 1.7500E-01   7.500E-01   1.624E+00
 2.2500E-01   7.500E-01   1.621E+00
 2.7500E-01   7.500E-01   1.616E+00
 3.2500E-01   7.500E-01   1.605E+00
 3.7500E-01   7.500E-01   1.585E+00
 4.2500E-01   7.500E-01   1.549E+00
 4.7500E-01   7.500E-01   1.491E+00
 5.2500E-01   7.500E-01   1.414E+00
 5.7500E-01   7.500E-01   1.337E+00
 6.2500E-01   7.500E-01   1.280E+00
 6.7500E-01   7.500E-01   1.237E+00
 1.2500E-02   7.625E-01   1.468E+00
 6.2500E-02   7.625E-01   1.468E+00
 1.1250E-01   7.625E-01   1.468E+00
 1.6250E-01   7.625E-01   1.467E+00
 2.1250E-01   7.625E-01   1.465E+00
 2.6250E-01   7.625E-01   1.462E+00
 3.1250E-01   7.625E-01   1.455E+00
 3.6250E-01   7.625E-01   1.442E+00
 4.1250E-01   7.625E-01   1.420E+00
 4.6250E-01   7.625E-01   1.387E+00
 5.1250E-01   7.625E-01   1.345E+00
 5.6250E-01   7.625E-01   1.302E+00
 6.1250E-01   7.625E-01   1.263E+00
 6.6250E-01   7.625E-01   1.229E+00
 2.5000E-02   7.750E-01   1.342E+00
 7.5000E-02   7.750E-01   1.342E+00
 1.2500E-01   7.750E-01   1.342E+00
 1.7500E-01   7.750E-01   1.341E+00
 2.2500E-01   7.750E-01   1.340E+00
 2.7500E-01   7.750E-01   1.339E+00
 3.2500E-01   7.750E-01   1.335E+00
 3.7500E-01   7.750E-01   1.329E+00
 4.2500E-01   7.750E-01   1.317E+00
 4.7500E-01   7.750E-01   1.301E+00
 5.2500E-01   7.750E-01   1.280E+00
 5.7500E-01   7.750E-01   1.256E+00
 6.2500E-01   7.750E-01   1.231E+00
 6.7500E-01   7.750E-01   1.205E+00
 3.7500E-02   7.875E-01   1.283E+00
 8.7500E-02   7.875E-01   1.283E+00
 1.3750E-01   7.875E-01   1.283E+00
 1.8750E-01   7.875E-01   1.282E+00
 2.3750E-01   7.875E-01   1.282E+00
 2.8750E-01   7.875E-01   1.281E+00
 3.3750E-01   7.875E-01   1.279E+00
 3.8750E-01   7.875E-01   1.274E+00
 4.3750E-01   7.875E-01   1.267E+00
 4.8750E-01   7.875E-01   1.257E+00
 5.3750E-01   7.875E-01   1.243E+00
 5.8750E-01   7.875E-01   1.226E+00
 0.0000E+00   8.000E-01   1.240E+00
 5.0000E-02   8.000E-01   1.240E+00
 1.0000E-01   8.000E-01   1.240E+00
 1.5000E-01   8.000E-01   1.240E+00
 2.0000E-01   8.000E-01   1.240E+00
 2.5000E-01   8.000E-01   1.240E+00
 3.0000E-01   8.000E-01   1.239E+00
 3.5000E-01   8.000E-01   1.238E+00
 4.0000E-01   8.000E-01   1.235E+00
 4.5000E-01   8.000E-01   1.230E+00
 5.0000E-01   8.000E-01   1.224E+00
 5.5000E-01   8.000E-01   1.214E+00
 6.0000E-01   8.000E-01   1.202E+00
 1.2500E-02   8.125E-01   1.213E+00
 6.2500E-02   8.125E-01   1.213E+00
 1.1250E-01   8.125E-01   1.213E+00
 1.6250E-01   8.125E-01   1.213E+00
 2.1250E-01   8.125E-01   1.212E+00
 2.6250E-01   8.125E-01   1.212E+00
```

```
 3.1250E-01   8.125E-01   1.212E+00
 3.6250E-01   8.125E-01   1.211E+00
 4.1250E-01   8.125E-01   1.209E+00
 4.6250E-01   8.125E-01   1.206E+00
 0.0000E+00   8.250E-01   1.190E+00
 5.0000E-02   8.250E-01   1.190E+00
 1.0000E-01   8.250E-01   1.190E+00
 1.5000E-01   8.250E-01   1.190E+00
 2.0000E-01   8.250E-01   1.189E+00
 2.5000E-01   8.250E-01   1.189E+00
 3.0000E-01   8.250E-01   1.189E+00
 3.5000E-01   8.250E-01   1.189E+00
 4.0000E-01   8.250E-01   1.187E+00
 4.5000E-01   8.250E-01   1.185E+00
 5.0000E-01   8.250E-01   1.182E+00

Statistics:
Time =   0.2500
Total number of accepted timesteps =  161
Total number of rejected timesteps =    2

            Total  number (rounded) of
         Residual  Jacobian    Newton   Lin sys
            evals     evals     iters     iters
At level
      1      1000       200       400       400
      2       200        20        50        50

            Maximum  number of
             Newton iters    Lin sys iters
At level
      1             4             3
      2             4             1



Example 2

Statistics:
Time =   0.0100
Total number of accepted timesteps =   14
Total number of rejected timesteps =    0

            Total  number (rounded) of
         Residual  Jacobian    Newton   Lin sys
            evals     evals     iters     iters
At level
      1       200        10        30        40
      2       200        10        20        30
      3        40         3         6         9

            Maximum  number of
             Newton iters    Lin sys iters
At level
      1             2             2
      2             2             2
      3             2             2

Solution at every 2nd grid point in level         3 at time   0.0250:

      x          y         approx c1   approx c2

  0.000E+00   0.000E+00    6.997E+01    6.996E+05
  5.000E-02   0.000E+00    7.282E+01    7.281E+05
  1.000E-01   0.000E+00    7.567E+01    7.566E+05
  1.500E-01   0.000E+00    6.957E+01    6.956E+05
  2.000E-01   0.000E+00    5.387E+01    5.386E+05
  2.500E-01   0.000E+00    3.536E+01    3.536E+05
  3.000E-01   0.000E+00    2.107E+01    2.108E+05
  3.500E-01   0.000E+00    1.370E+01    1.371E+05
  4.000E-01   0.000E+00    1.300E+01    1.301E+05
```

4.500E-01	0.000E+00	1.859E+01	1.859E+05
5.000E-01	0.000E+00	3.003E+01	3.003E+05
5.500E-01	0.000E+00	4.434E+01	4.433E+05
6.000E-01	0.000E+00	5.473E+01	5.472E+05
6.500E-01	0.000E+00	5.469E+01	5.468E+05
7.000E-01	0.000E+00	4.413E+01	4.413E+05
7.500E-01	0.000E+00	2.943E+01	2.944E+05
8.000E-01	0.000E+00	1.712E+01	1.713E+05
8.500E-01	0.000E+00	9.594E+00	9.601E+04
9.000E-01	0.000E+00	6.033E+00	6.040E+04
9.500E-01	0.000E+00	4.825E+00	4.829E+04
1.000E+00	0.000E+00	4.607E+00	4.609E+04
2.500E-02	2.500E-02	7.150E+01	7.150E+05
7.500E-02	2.500E-02	7.646E+01	7.646E+05
1.250E-01	2.500E-02	7.586E+01	7.585E+05
1.750E-01	2.500E-02	6.394E+01	6.394E+05
2.250E-01	2.500E-02	4.495E+01	4.494E+05
2.750E-01	2.500E-02	2.736E+01	2.736E+05
3.250E-01	2.500E-02	1.628E+01	1.629E+05
3.750E-01	2.500E-02	1.231E+01	1.232E+05
4.250E-01	2.500E-02	1.481E+01	1.481E+05
4.750E-01	2.500E-02	2.364E+01	2.364E+05
5.250E-01	2.500E-02	3.764E+01	3.764E+05
5.750E-01	2.500E-02	5.170E+01	5.169E+05
6.250E-01	2.500E-02	5.776E+01	5.775E+05
6.750E-01	2.500E-02	5.161E+01	5.160E+05
7.250E-01	2.500E-02	3.731E+01	3.731E+05
7.750E-01	2.500E-02	2.274E+01	2.275E+05
8.250E-01	2.500E-02	1.266E+01	1.266E+05
8.750E-01	2.500E-02	7.348E+00	7.358E+04
9.250E-01	2.500E-02	5.237E+00	5.245E+04
9.750E-01	2.500E-02	4.739E+00	4.742E+04
0.000E+00	5.000E-02	7.282E+01	7.281E+05
5.000E-02	5.000E-02	7.745E+01	7.744E+05
1.000E-01	5.000E-02	8.280E+01	8.278E+05
1.500E-01	5.000E-02	7.633E+01	7.631E+05
2.000E-01	5.000E-02	5.756E+01	5.755E+05
2.500E-01	5.000E-02	3.603E+01	3.603E+05
3.000E-01	5.000E-02	2.036E+01	2.037E+05
3.500E-01	5.000E-02	1.273E+01	1.275E+05
4.000E-01	5.000E-02	1.209E+01	1.210E+05
4.500E-01	5.000E-02	1.799E+01	1.800E+05
5.000E-01	5.000E-02	3.070E+01	3.071E+05
5.500E-01	5.000E-02	4.762E+01	4.761E+05
6.000E-01	5.000E-02	6.050E+01	6.048E+05
6.500E-01	5.000E-02	6.048E+01	6.047E+05
7.000E-01	5.000E-02	4.750E+01	4.749E+05
7.500E-01	5.000E-02	3.026E+01	3.026E+05
8.000E-01	5.000E-02	1.679E+01	1.681E+05
8.500E-01	5.000E-02	9.182E+00	9.200E+04
9.000E-01	5.000E-02	5.901E+00	5.918E+04
9.500E-01	5.000E-02	5.010E+00	5.021E+04
1.000E+00	5.000E-02	4.957E+00	4.961E+04
2.500E-02	7.500E-02	7.646E+01	7.646E+05
7.500E-02	7.500E-02	8.585E+01	8.583E+05
1.250E-01	7.500E-02	8.738E+01	8.735E+05
1.750E-01	7.500E-02	7.236E+01	7.234E+05
2.250E-01	7.500E-02	4.806E+01	4.805E+05
2.750E-01	7.500E-02	2.701E+01	2.702E+05
3.250E-01	7.500E-02	1.491E+01	1.493E+05
3.750E-01	7.500E-02	1.091E+01	1.094E+05
4.250E-01	7.500E-02	1.361E+01	1.363E+05
4.750E-01	7.500E-02	2.348E+01	2.349E+05
5.250E-01	7.500E-02	4.057E+01	4.056E+05
5.750E-01	7.500E-02	5.914E+01	5.912E+05
6.250E-01	7.500E-02	6.756E+01	6.754E+05
6.750E-01	7.500E-02	5.918E+01	5.916E+05
7.250E-01	7.500E-02	4.047E+01	4.047E+05
7.750E-01	7.500E-02	2.296E+01	2.297E+05
8.250E-01	7.500E-02	1.211E+01	1.213E+05
8.750E-01	7.500E-02	7.095E+00	7.120E+04

```
9.250E-01   7.500E-02    5.522E+00    5.542E+04
9.750E-01   7.500E-02    5.505E+00    5.512E+04
0.000E+00   1.000E-01    7.567E+01    7.566E+05
5.000E-02   1.000E-01    8.280E+01    8.278E+05
1.000E-01   1.000E-01    9.189E+01    9.187E+05
1.500E-01   1.000E-01    8.513E+01    8.510E+05
2.000E-01   1.000E-01    6.222E+01    6.220E+05
2.500E-01   1.000E-01    3.676E+01    3.676E+05
3.000E-01   1.000E-01    1.947E+01    1.949E+05
3.500E-01   1.000E-01    1.168E+01    1.170E+05
4.000E-01   1.000E-01    1.114E+01    1.116E+05
4.500E-01   1.000E-01    1.738E+01    1.739E+05
5.000E-01   1.000E-01    3.167E+01    3.167E+05
5.500E-01   1.000E-01    5.212E+01    5.211E+05
6.000E-01   1.000E-01    6.856E+01    6.853E+05
6.500E-01   1.000E-01    6.864E+01    6.861E+05
7.000E-01   1.000E-01    5.228E+01    5.226E+05
7.500E-01   1.000E-01    3.165E+01    3.165E+05
8.000E-01   1.000E-01    1.682E+01    1.684E+05
8.500E-01   1.000E-01    9.289E+00    9.317E+04
9.000E-01   1.000E-01    6.561E+00    6.590E+04
9.500E-01   1.000E-01    6.305E+00    6.322E+04
1.000E+00   1.000E-01    6.594E+00    6.600E+04
2.500E-02   1.250E-01    7.586E+01    7.585E+05
7.500E-02   1.250E-01    8.738E+01    8.735E+05
1.250E-01   1.250E-01    9.027E+01    9.024E+05
1.750E-01   1.250E-01    7.438E+01    7.436E+05
2.250E-01   1.250E-01    4.841E+01    4.840E+05
2.750E-01   1.250E-01    2.662E+01    2.663E+05
3.250E-01   1.250E-01    1.465E+01    1.467E+05
3.750E-01   1.250E-01    1.087E+01    1.090E+05
4.250E-01   1.250E-01    1.355E+01    1.358E+05
4.750E-01   1.250E-01    2.351E+01    2.352E+05
5.250E-01   1.250E-01    4.148E+01    4.147E+05
5.750E-01   1.250E-01    6.178E+01    6.175E+05
6.250E-01   1.250E-01    7.124E+01    7.121E+05
6.750E-01   1.250E-01    6.205E+01    6.203E+05
7.250E-01   1.250E-01    4.183E+01    4.182E+05
7.750E-01   1.250E-01    2.364E+01    2.365E+05
8.250E-01   1.250E-01    1.302E+01    1.304E+05
8.750E-01   1.250E-01    8.572E+00    8.603E+04
9.250E-01   1.250E-01    7.650E+00    7.675E+04
9.750E-01   1.250E-01    8.236E+00    8.245E+04
0.000E+00   1.500E-01    6.957E+01    6.956E+05
5.000E-02   1.500E-01    7.633E+01    7.631E+05
1.000E-01   1.500E-01    8.513E+01    8.510E+05
1.500E-01   1.500E-01    7.927E+01    7.924E+05
2.000E-01   1.500E-01    5.845E+01    5.844E+05
2.500E-01   1.500E-01    3.536E+01    3.536E+05
3.000E-01   1.500E-01    1.982E+01    1.984E+05
3.500E-01   1.500E-01    1.289E+01    1.292E+05
4.000E-01   1.500E-01    1.245E+01    1.248E+05
4.500E-01   1.500E-01    1.809E+01    1.811E+05
5.000E-01   1.500E-01    3.108E+01    3.108E+05
5.500E-01   1.500E-01    4.985E+01    4.983E+05
6.000E-01   1.500E-01    6.509E+01    6.506E+05
6.500E-01   1.500E-01    6.532E+01    6.529E+05
7.000E-01   1.500E-01    5.043E+01    5.041E+05
7.500E-01   1.500E-01    3.175E+01    3.175E+05
8.000E-01   1.500E-01    1.852E+01    1.853E+05
8.500E-01   1.500E-01    1.204E+01    1.207E+05
9.000E-01   1.500E-01    1.005E+01    1.007E+05
9.500E-01   1.500E-01    1.043E+01    1.045E+05
1.000E+00   1.500E-01    1.106E+01    1.106E+05
2.500E-02   1.750E-01    6.394E+01    6.394E+05
7.500E-02   1.750E-01    7.236E+01    7.234E+05
1.250E-01   1.750E-01    7.438E+01    7.436E+05
1.750E-01   1.750E-01    6.250E+01    6.248E+05
2.250E-01   1.750E-01    4.296E+01    4.295E+05
2.750E-01   1.750E-01    2.634E+01    2.635E+05
3.250E-01   1.750E-01    1.704E+01    1.706E+05
```

```
3.750E-01   1.750E-01    1.408E+01    1.410E+05
4.250E-01   1.750E-01    1.619E+01    1.621E+05
4.750E-01   1.750E-01    2.393E+01    2.394E+05
5.250E-01   1.750E-01    3.768E+01    3.767E+05
5.750E-01   1.750E-01    5.302E+01    5.300E+05
6.250E-01   1.750E-01    6.022E+01    6.020E+05
6.750E-01   1.750E-01    5.361E+01    5.359E+05
7.250E-01   1.750E-01    3.870E+01    3.869E+05
7.750E-01   1.750E-01    2.512E+01    2.513E+05
8.250E-01   1.750E-01    1.719E+01    1.721E+05
8.750E-01   1.750E-01    1.407E+01    1.409E+05
9.250E-01   1.750E-01    1.381E+01    1.383E+05
9.750E-01   1.750E-01    1.479E+01    1.480E+05
0.000E+00   2.000E-01    5.387E+01    5.386E+05
5.000E-02   2.000E-01    5.756E+01    5.755E+05
1.000E-01   2.000E-01    6.222E+01    6.220E+05
1.500E-01   2.000E-01    5.845E+01    5.844E+05
2.000E-01   2.000E-01    4.603E+01    4.602E+05
2.500E-01   2.000E-01    3.208E+01    3.208E+05
3.000E-01   2.000E-01    2.249E+01    2.250E+05
3.500E-01   2.000E-01    1.815E+01    1.817E+05
4.000E-01   2.000E-01    1.783E+01    1.785E+05
4.500E-01   2.000E-01    2.125E+01    2.126E+05
5.000E-01   2.000E-01    2.913E+01    2.913E+05
5.500E-01   2.000E-01    4.033E+01    4.032E+05
6.000E-01   2.000E-01    4.935E+01    4.933E+05
6.500E-01   2.000E-01    4.973E+01    4.972E+05
7.000E-01   2.000E-01    4.145E+01    4.144E+05
7.500E-01   2.000E-01    3.086E+01    3.086E+05
8.000E-01   2.000E-01    2.340E+01    2.341E+05
8.500E-01   2.000E-01    1.997E+01    1.998E+05
9.000E-01   2.000E-01    1.916E+01    1.918E+05
9.500E-01   2.000E-01    1.973E+01    1.974E+05
1.000E+00   2.000E-01    2.031E+01    2.031E+05
2.500E-02   2.250E-01    4.495E+01    4.494E+05
7.500E-02   2.250E-01    4.806E+01    4.805E+05
1.250E-01   2.250E-01    4.841E+01    4.840E+05
1.750E-01   2.250E-01    4.296E+01    4.295E+05
2.250E-01   2.250E-01    3.441E+01    3.440E+05
2.750E-01   2.250E-01    2.736E+01    2.736E+05
3.250E-01   2.250E-01    2.372E+01    2.373E+05
3.750E-01   2.250E-01    2.262E+01    2.263E+05
4.250E-01   2.250E-01    2.316E+01    2.317E+05
4.750E-01   2.250E-01    2.585E+01    2.586E+05
5.250E-01   2.250E-01    3.130E+01    3.130E+05
5.750E-01   2.250E-01    3.770E+01    3.769E+05
6.250E-01   2.250E-01    4.091E+01    4.090E+05
6.750E-01   2.250E-01    3.863E+01    3.862E+05
7.250E-01   2.250E-01    3.321E+01    3.321E+05
7.750E-01   2.250E-01    2.881E+01    2.881E+05
8.250E-01   2.250E-01    2.704E+01    2.704E+05
8.750E-01   2.250E-01    2.685E+01    2.685E+05
9.250E-01   2.250E-01    2.695E+01    2.696E+05
9.750E-01   2.250E-01    2.714E+01    2.714E+05
0.000E+00   2.500E-01    3.536E+01    3.536E+05
5.000E-02   2.500E-01    3.603E+01    3.603E+05
1.000E-01   2.500E-01    3.676E+01    3.676E+05
1.500E-01   2.500E-01    3.536E+01    3.536E+05
2.000E-01   2.500E-01    3.208E+01    3.208E+05
2.500E-01   2.500E-01    2.934E+01    2.934E+05
3.000E-01   2.500E-01    2.875E+01    2.875E+05
3.500E-01   2.500E-01    2.932E+01    2.932E+05
4.000E-01   2.500E-01    2.914E+01    2.914E+05
4.500E-01   2.500E-01    2.808E+01    2.808E+05
5.000E-01   2.500E-01    2.782E+01    2.782E+05
5.500E-01   2.500E-01    2.935E+01    2.935E+05
6.000E-01   2.500E-01    3.135E+01    3.135E+05
6.500E-01   2.500E-01    3.187E+01    3.187E+05
7.000E-01   2.500E-01    3.106E+01    3.105E+05
7.500E-01   2.500E-01    3.106E+01    3.106E+05
8.000E-01   2.500E-01    3.316E+01    3.316E+05
```

```
8.500E-01   2.500E-01    3.594E+01    3.594E+05
9.000E-01   2.500E-01    3.702E+01    3.702E+05
9.500E-01   2.500E-01    3.612E+01    3.612E+05
1.000E+00   2.500E-01    3.540E+01    3.540E+05
2.500E-02   2.750E-01    2.736E+01    2.736E+05
7.500E-02   2.750E-01    2.701E+01    2.702E+05
1.250E-01   2.750E-01    2.662E+01    2.663E+05
1.750E-01   2.750E-01    2.634E+01    2.635E+05
2.250E-01   2.750E-01    2.736E+01    2.736E+05
2.750E-01   2.750E-01    3.081E+01    3.080E+05
3.250E-01   2.750E-01    3.542E+01    3.541E+05
3.750E-01   2.750E-01    3.761E+01    3.760E+05
4.250E-01   2.750E-01    3.522E+01    3.521E+05
4.750E-01   2.750E-01    3.021E+01    3.020E+05
5.250E-01   2.750E-01    2.614E+01    2.614E+05
5.750E-01   2.750E-01    2.446E+01    2.447E+05
6.250E-01   2.750E-01    2.447E+01    2.448E+05
6.750E-01   2.750E-01    2.568E+01    2.569E+05
7.250E-01   2.750E-01    2.908E+01    2.908E+05
7.750E-01   2.750E-01    3.567E+01    3.566E+05
8.250E-01   2.750E-01    4.371E+01    4.371E+05
8.750E-01   2.750E-01    4.876E+01    4.875E+05
9.250E-01   2.750E-01    4.818E+01    4.817E+05
9.750E-01   2.750E-01    4.500E+01    4.499E+05
0.000E+00   3.000E-01    2.107E+01    2.108E+05
5.000E-02   3.000E-01    2.036E+01    2.037E+05
1.000E-01   3.000E-01    1.947E+01    1.949E+05
1.500E-01   3.000E-01    1.982E+01    1.984E+05
2.000E-01   3.000E-01    2.249E+01    2.250E+05
2.500E-01   3.000E-01    2.875E+01    2.875E+05
3.000E-01   3.000E-01    3.792E+01    3.791E+05
3.500E-01   3.000E-01    4.533E+01    4.531E+05
4.000E-01   3.000E-01    4.534E+01    4.532E+05
4.500E-01   3.000E-01    3.784E+01    3.783E+05
5.000E-01   3.000E-01    2.840E+01    2.840E+05
5.500E-01   3.000E-01    2.183E+01    2.184E+05
6.000E-01   3.000E-01    1.914E+01    1.915E+05
6.500E-01   3.000E-01    1.976E+01    1.977E+05
7.000E-01   3.000E-01    2.408E+01    2.409E+05
7.500E-01   3.000E-01    3.336E+01    3.336E+05
8.000E-01   3.000E-01    4.678E+01    4.676E+05
8.500E-01   3.000E-01    5.863E+01    5.861E+05
9.000E-01   3.000E-01    6.202E+01    6.200E+05
9.500E-01   3.000E-01    5.724E+01    5.723E+05
1.000E+00   3.000E-01    5.355E+01    5.354E+05
2.500E-02   3.250E-01    1.628E+01    1.629E+05
7.500E-02   3.250E-01    1.491E+01    1.493E+05
1.250E-01   3.250E-01    1.465E+01    1.467E+05
1.750E-01   3.250E-01    1.704E+01    1.706E+05
2.250E-01   3.250E-01    2.372E+01    2.373E+05
2.750E-01   3.250E-01    3.542E+01    3.541E+05
3.250E-01   3.250E-01    4.849E+01    4.847E+05
3.750E-01   3.250E-01    5.457E+01    5.454E+05
4.250E-01   3.250E-01    4.867E+01    4.865E+05
4.750E-01   3.250E-01    3.559E+01    3.558E+05
5.250E-01   3.250E-01    2.375E+01    2.376E+05
5.750E-01   3.250E-01    1.711E+01    1.713E+05
6.250E-01   3.250E-01    1.548E+01    1.551E+05
6.750E-01   3.250E-01    1.855E+01    1.857E+05
7.250E-01   3.250E-01    2.761E+01    2.762E+05
7.750E-01   3.250E-01    4.362E+01    4.361E+05
8.250E-01   3.250E-01    6.229E+01    6.226E+05
8.750E-01   3.250E-01    7.345E+01    7.342E+05
9.250E-01   3.250E-01    7.114E+01    7.112E+05
9.750E-01   3.250E-01    6.278E+01    6.277E+05
0.000E+00   3.500E-01    1.370E+01    1.371E+05
5.000E-02   3.500E-01    1.273E+01    1.275E+05
1.000E-01   3.500E-01    1.168E+01    1.170E+05
1.500E-01   3.500E-01    1.289E+01    1.292E+05
2.000E-01   3.500E-01    1.815E+01    1.817E+05
2.500E-01   3.500E-01    2.932E+01    2.932E+05
```

```
3.000E-01   3.500E-01    4.533E+01    4.531E+05
3.500E-01   3.500E-01    5.835E+01    5.832E+05
4.000E-01   3.500E-01    5.854E+01    5.851E+05
4.500E-01   3.500E-01    4.577E+01    4.576E+05
5.000E-01   3.500E-01    2.979E+01    2.979E+05
5.500E-01   3.500E-01    1.865E+01    1.867E+05
6.000E-01   3.500E-01    1.387E+01    1.390E+05
6.500E-01   3.500E-01    1.455E+01    1.458E+05
7.000E-01   3.500E-01    2.128E+01    2.130E+05
7.500E-01   3.500E-01    3.607E+01    3.607E+05
8.000E-01   3.500E-01    5.785E+01    5.783E+05
8.500E-01   3.500E-01    7.728E+01    7.725E+05
9.000E-01   3.500E-01    8.238E+01    8.235E+05
9.500E-01   3.500E-01    7.359E+01    7.357E+05
1.000E+00   3.500E-01    6.699E+01    6.698E+05
2.500E-02   3.750E-01    1.231E+01    1.232E+05
7.500E-02   3.750E-01    1.091E+01    1.094E+05
1.250E-01   3.750E-01    1.087E+01    1.090E+05
1.750E-01   3.750E-01    1.408E+01    1.410E+05
2.250E-01   3.750E-01    2.262E+01    2.263E+05
2.750E-01   3.750E-01    3.761E+01    3.760E+05
3.250E-01   3.750E-01    5.457E+01    5.454E+05
3.750E-01   3.750E-01    6.259E+01    6.256E+05
4.250E-01   3.750E-01    5.507E+01    5.504E+05
4.750E-01   3.750E-01    3.835E+01    3.834E+05
5.250E-01   3.750E-01    2.342E+01    2.343E+05
5.750E-01   3.750E-01    1.514E+01    1.517E+05
6.250E-01   3.750E-01    1.306E+01    1.309E+05
6.750E-01   3.750E-01    1.671E+01    1.673E+05
7.250E-01   3.750E-01    2.779E+01    2.780E+05
7.750E-01   3.750E-01    4.774E+01    4.773E+05
8.250E-01   3.750E-01    7.139E+01    7.136E+05
8.750E-01   3.750E-01    8.551E+01    8.548E+05
9.250E-01   3.750E-01    8.219E+01    8.216E+05
9.750E-01   3.750E-01    7.107E+01    7.105E+05
0.000E+00   4.000E-01    1.300E+01    1.301E+05
5.000E-02   4.000E-01    1.209E+01    1.210E+05
1.000E-01   4.000E-01    1.114E+01    1.116E+05
1.500E-01   4.000E-01    1.245E+01    1.248E+05
2.000E-01   4.000E-01    1.783E+01    1.785E+05
2.500E-01   4.000E-01    2.914E+01    2.914E+05
3.000E-01   4.000E-01    4.534E+01    4.532E+05
3.500E-01   4.000E-01    5.854E+01    5.851E+05
4.000E-01   4.000E-01    5.887E+01    5.885E+05
4.500E-01   4.000E-01    4.616E+01    4.615E+05
5.000E-01   4.000E-01    3.019E+01    3.018E+05
5.500E-01   4.000E-01    1.904E+01    1.905E+05
6.000E-01   4.000E-01    1.426E+01    1.429E+05
6.500E-01   4.000E-01    1.499E+01    1.502E+05
7.000E-01   4.000E-01    2.182E+01    2.183E+05
7.500E-01   4.000E-01    3.676E+01    3.676E+05
8.000E-01   4.000E-01    5.874E+01    5.872E+05
8.500E-01   4.000E-01    7.834E+01    7.831E+05
9.000E-01   4.000E-01    8.350E+01    8.347E+05
9.500E-01   4.000E-01    7.465E+01    7.463E+05
1.000E+00   4.000E-01    6.800E+01    6.799E+05
2.500E-02   4.250E-01    1.481E+01    1.481E+05
7.500E-02   4.250E-01    1.361E+01    1.363E+05
1.250E-01   4.250E-01    1.355E+01    1.358E+05
1.750E-01   4.250E-01    1.619E+01    1.621E+05
2.250E-01   4.250E-01    2.316E+01    2.317E+05
2.750E-01   4.250E-01    3.522E+01    3.521E+05
3.250E-01   4.250E-01    4.867E+01    4.865E+05
3.750E-01   4.250E-01    5.507E+01    5.504E+05
4.250E-01   4.250E-01    4.937E+01    4.935E+05
4.750E-01   4.250E-01    3.636E+01    3.635E+05
5.250E-01   4.250E-01    2.455E+01    2.456E+05
5.750E-01   4.250E-01    1.793E+01    1.795E+05
6.250E-01   4.250E-01    1.637E+01    1.639E+05
6.750E-01   4.250E-01    1.956E+01    1.958E+05
7.250E-01   4.250E-01    2.884E+01    2.885E+05
```

```
 7.750E-01   4.250E-01    4.515E+01    4.514E+05
 8.250E-01   4.250E-01    6.413E+01    6.411E+05
 8.750E-01   4.250E-01    7.551E+01    7.548E+05
 9.250E-01   4.250E-01    7.322E+01    7.320E+05
 9.750E-01   4.250E-01    6.474E+01    6.473E+05
 0.000E+00   4.500E-01    1.859E+01    1.859E+05
 5.000E-02   4.500E-01    1.799E+01    1.800E+05
 1.000E-01   4.500E-01    1.738E+01    1.739E+05
 1.500E-01   4.500E-01    1.809E+01    1.811E+05
 2.000E-01   4.500E-01    2.125E+01    2.126E+05
 2.500E-01   4.500E-01    2.808E+01    2.808E+05
 3.000E-01   4.500E-01    3.784E+01    3.783E+05
 3.500E-01   4.500E-01    4.577E+01    4.576E+05
 4.000E-01   4.500E-01    4.616E+01    4.615E+05
 4.500E-01   4.500E-01    3.890E+01    3.889E+05
 5.000E-01   4.500E-01    2.959E+01    2.959E+05
 5.500E-01   4.500E-01    2.313E+01    2.314E+05
 6.000E-01   4.500E-01    2.056E+01    2.057E+05
 6.500E-01   4.500E-01    2.133E+01    2.135E+05
 7.000E-01   4.500E-01    2.585E+01    2.586E+05
 7.500E-01   4.500E-01    3.540E+01    3.539E+05
 8.000E-01   4.500E-01    4.914E+01    4.913E+05
 8.500E-01   4.500E-01    6.130E+01    6.128E+05
 9.000E-01   4.500E-01    6.485E+01    6.483E+05
 9.500E-01   4.500E-01    6.004E+01    6.003E+05
 1.000E+00   4.500E-01    5.629E+01    5.629E+05
 2.500E-02   4.750E-01    2.364E+01    2.364E+05
 7.500E-02   4.750E-01    2.348E+01    2.349E+05
 1.250E-01   4.750E-01    2.351E+01    2.352E+05
 1.750E-01   4.750E-01    2.393E+01    2.394E+05
 2.250E-01   4.750E-01    2.585E+01    2.586E+05
 2.750E-01   4.750E-01    3.021E+01    3.020E+05
 3.250E-01   4.750E-01    3.559E+01    3.558E+05
 3.750E-01   4.750E-01    3.835E+01    3.834E+05
 4.250E-01   4.750E-01    3.636E+01    3.635E+05
 4.750E-01   4.750E-01    3.166E+01    3.165E+05
 5.250E-01   4.750E-01    2.787E+01    2.787E+05
 5.750E-01   4.750E-01    2.648E+01    2.648E+05
 6.250E-01   4.750E-01    2.674E+01    2.675E+05
 6.750E-01   4.750E-01    2.814E+01    2.815E+05
 7.250E-01   4.750E-01    3.170E+01    3.170E+05
 7.750E-01   4.750E-01    3.846E+01    3.845E+05
 8.250E-01   4.750E-01    4.672E+01    4.672E+05
 8.750E-01   4.750E-01    5.199E+01    5.197E+05
 9.250E-01   4.750E-01    5.155E+01    5.154E+05
 9.750E-01   4.750E-01    4.841E+01    4.840E+05
 0.000E+00   5.000E-01    3.003E+01    3.003E+05
 5.000E-02   5.000E-01    3.070E+01    3.071E+05
 1.000E-01   5.000E-01    3.167E+01    3.167E+05
 1.500E-01   5.000E-01    3.108E+01    3.108E+05
 2.000E-01   5.000E-01    2.913E+01    2.913E+05
 2.500E-01   5.000E-01    2.782E+01    2.782E+05
 3.000E-01   5.000E-01    2.840E+01    2.840E+05
 3.500E-01   5.000E-01    2.979E+01    2.979E+05
 4.000E-01   5.000E-01    3.019E+01    3.018E+05
 4.500E-01   5.000E-01    2.959E+01    2.959E+05
 5.000E-01   5.000E-01    2.982E+01    2.982E+05
 5.500E-01   5.000E-01    3.188E+01    3.188E+05
 6.000E-01   5.000E-01    3.438E+01    3.437E+05
 6.500E-01   5.000E-01    3.522E+01    3.522E+05
 7.000E-01   5.000E-01    3.448E+01    3.448E+05
 7.500E-01   5.000E-01    3.442E+01    3.441E+05
 8.000E-01   5.000E-01    3.647E+01    3.647E+05
 8.500E-01   5.000E-01    3.934E+01    3.934E+05
 9.000E-01   5.000E-01    4.064E+01    4.064E+05
 9.500E-01   5.000E-01    4.000E+01    4.000E+05
 1.000E+00   5.000E-01    3.940E+01    3.940E+05
 2.500E-02   5.250E-01    3.764E+01    3.764E+05
 7.500E-02   5.250E-01    4.057E+01    4.056E+05
 1.250E-01   5.250E-01    4.148E+01    4.147E+05
 1.750E-01   5.250E-01    3.768E+01    3.767E+05
```

```
2.250E-01   5.250E-01    3.130E+01    3.130E+05
2.750E-01   5.250E-01    2.614E+01    2.614E+05
3.250E-01   5.250E-01    2.375E+01    2.376E+05
3.750E-01   5.250E-01    2.342E+01    2.343E+05
4.250E-01   5.250E-01    2.455E+01    2.456E+05
4.750E-01   5.250E-01    2.787E+01    2.787E+05
5.250E-01   5.250E-01    3.410E+01    3.409E+05
5.750E-01   5.250E-01    4.134E+01    4.133E+05
6.250E-01   5.250E-01    4.519E+01    4.518E+05
6.750E-01   5.250E-01    4.309E+01    4.308E+05
7.250E-01   5.250E-01    3.744E+01    3.744E+05
7.750E-01   5.250E-01    3.265E+01    3.265E+05
8.250E-01   5.250E-01    3.066E+01    3.066E+05
8.750E-01   5.250E-01    3.056E+01    3.057E+05
9.250E-01   5.250E-01    3.106E+01    3.106E+05
9.750E-01   5.250E-01    3.169E+01    3.169E+05
0.000E+00   5.500E-01    4.434E+01    4.434E+05
5.000E-02   5.500E-01    4.762E+01    4.761E+05
1.000E-01   5.500E-01    5.212E+01    5.211E+05
1.500E-01   5.500E-01    4.985E+01    4.983E+05
2.000E-01   5.500E-01    4.033E+01    4.032E+05
2.500E-01   5.500E-01    2.935E+01    2.935E+05
3.000E-01   5.500E-01    2.183E+01    2.184E+05
3.500E-01   5.500E-01    1.865E+01    1.867E+05
4.000E-01   5.500E-01    1.904E+01    1.905E+05
4.500E-01   5.500E-01    2.313E+01    2.314E+05
5.000E-01   5.500E-01    3.188E+01    3.188E+05
5.500E-01   5.500E-01    4.419E+01    4.418E+05
6.000E-01   5.500E-01    5.427E+01    5.425E+05
6.500E-01   5.500E-01    5.523E+01    5.521E+05
7.000E-01   5.500E-01    4.679E+01    4.678E+05
7.500E-01   5.500E-01    3.558E+01    3.558E+05
8.000E-01   5.500E-01    2.753E+01    2.754E+05
8.500E-01   5.500E-01    2.390E+01    2.391E+05
9.000E-01   5.500E-01    2.342E+01    2.344E+05
9.500E-01   5.500E-01    2.472E+01    2.473E+05
1.000E+00   5.500E-01    2.575E+01    2.575E+05
2.500E-02   5.750E-01    5.170E+01    5.169E+05
7.500E-02   5.750E-01    5.914E+01    5.912E+05
1.250E-01   5.750E-01    6.178E+01    6.175E+05
1.750E-01   5.750E-01    5.302E+01    5.300E+05
2.250E-01   5.750E-01    3.770E+01    3.769E+05
2.750E-01   5.750E-01    2.446E+01    2.447E+05
3.250E-01   5.750E-01    1.711E+01    1.713E+05
3.750E-01   5.750E-01    1.514E+01    1.517E+05
4.250E-01   5.750E-01    1.793E+01    1.795E+05
4.750E-01   5.750E-01    2.648E+01    2.648E+05
5.250E-01   5.750E-01    4.134E+01    4.133E+05
5.750E-01   5.750E-01    5.798E+01    5.796E+05
6.250E-01   5.750E-01    6.625E+01    6.622E+05
6.750E-01   5.750E-01    5.995E+01    5.993E+05
7.250E-01   5.750E-01    4.455E+01    4.454E+05
7.750E-01   5.750E-01    3.017E+01    3.018E+05
8.250E-01   5.750E-01    2.172E+01    2.174E+05
8.750E-01   5.750E-01    1.871E+01    1.873E+05
9.250E-01   5.750E-01    1.927E+01    1.929E+05
9.750E-01   5.750E-01    2.131E+01    2.132E+05
0.000E+00   6.000E-01    5.473E+01    5.472E+05
5.000E-02   6.000E-01    6.050E+01    6.048E+05
1.000E-01   6.000E-01    6.856E+01    6.853E+05
1.500E-01   6.000E-01    6.509E+01    6.506E+05
2.000E-01   6.000E-01    4.935E+01    4.933E+05
2.500E-01   6.000E-01    3.135E+01    3.135E+05
3.000E-01   6.000E-01    1.914E+01    1.915E+05
3.500E-01   6.000E-01    1.387E+01    1.390E+05
4.000E-01   6.000E-01    1.426E+01    1.429E+05
4.500E-01   6.000E-01    2.056E+01    2.057E+05
5.000E-01   6.000E-01    3.438E+01    3.437E+05
5.500E-01   6.000E-01    5.427E+01    5.425E+05
6.000E-01   6.000E-01    7.076E+01    7.073E+05
6.500E-01   6.000E-01    7.189E+01    7.186E+05
```

```
7.000E-01   6.000E-01    5.714E+01    5.712E+05
7.500E-01   6.000E-01    3.796E+01    3.795E+05
8.000E-01   6.000E-01    2.415E+01    2.416E+05
8.500E-01   6.000E-01    1.758E+01    1.760E+05
9.000E-01   6.000E-01    1.630E+01    1.633E+05
9.500E-01   6.000E-01    1.813E+01    1.815E+05
1.000E+00   6.000E-01    1.967E+01    1.967E+05
2.500E-02   6.250E-01    5.776E+01    5.775E+05
7.500E-02   6.250E-01    6.756E+01    6.754E+05
1.250E-01   6.250E-01    7.124E+01    7.121E+05
1.750E-01   6.250E-01    6.022E+01    6.020E+05
2.250E-01   6.250E-01    4.091E+01    4.090E+05
2.750E-01   6.250E-01    2.447E+01    2.448E+05
3.250E-01   6.250E-01    1.548E+01    1.551E+05
3.750E-01   6.250E-01    1.306E+01    1.309E+05
4.250E-01   6.250E-01    1.637E+01    1.639E+05
4.750E-01   6.250E-01    2.674E+01    2.675E+05
5.250E-01   6.250E-01    4.519E+01    4.518E+05
5.750E-01   6.250E-01    6.625E+01    6.622E+05
6.250E-01   6.250E-01    7.677E+01    7.673E+05
6.750E-01   6.250E-01    6.856E+01    6.853E+05
7.250E-01   6.250E-01    4.883E+01    4.882E+05
7.750E-01   6.250E-01    3.068E+01    3.068E+05
8.250E-01   6.250E-01    2.013E+01    2.015E+05
8.750E-01   6.250E-01    1.635E+01    1.638E+05
9.250E-01   6.250E-01    1.700E+01    1.702E+05
9.750E-01   6.250E-01    1.949E+01    1.950E+05
0.000E+00   6.500E-01    5.469E+01    5.468E+05
5.000E-02   6.500E-01    6.048E+01    6.047E+05
1.000E-01   6.500E-01    6.864E+01    6.861E+05
1.500E-01   6.500E-01    6.532E+01    6.529E+05
2.000E-01   6.500E-01    4.973E+01    4.972E+05
2.500E-01   6.500E-01    3.187E+01    3.187E+05
3.000E-01   6.500E-01    1.976E+01    1.977E+05
3.500E-01   6.500E-01    1.455E+01    1.458E+05
4.000E-01   6.500E-01    1.499E+01    1.502E+05
4.500E-01   6.500E-01    2.133E+01    2.135E+05
5.000E-01   6.500E-01    3.522E+01    3.522E+05
5.500E-01   6.500E-01    5.523E+01    5.521E+05
6.000E-01   6.500E-01    7.189E+01    7.186E+05
6.500E-01   6.500E-01    7.323E+01    7.320E+05
7.000E-01   6.500E-01    5.869E+01    5.867E+05
7.500E-01   6.500E-01    3.968E+01    3.968E+05
8.000E-01   6.500E-01    2.603E+01    2.604E+05
8.500E-01   6.500E-01    1.963E+01    1.966E+05
9.000E-01   6.500E-01    1.860E+01    1.863E+05
9.500E-01   6.500E-01    2.076E+01    2.078E+05
1.000E+00   6.500E-01    2.250E+01    2.250E+05
2.500E-02   6.750E-01    5.161E+01    5.160E+05
7.500E-02   6.750E-01    5.918E+01    5.916E+05
1.250E-01   6.750E-01    6.205E+01    6.203E+05
1.750E-01   6.750E-01    5.361E+01    5.359E+05
2.250E-01   6.750E-01    3.863E+01    3.862E+05
2.750E-01   6.750E-01    2.568E+01    2.569E+05
3.250E-01   6.750E-01    1.855E+01    1.857E+05
3.750E-01   6.750E-01    1.671E+01    1.673E+05
4.250E-01   6.750E-01    1.956E+01    1.958E+05
4.750E-01   6.750E-01    2.814E+01    2.815E+05
5.250E-01   6.750E-01    4.309E+01    4.308E+05
5.750E-01   6.750E-01    5.995E+01    5.993E+05
6.250E-01   6.750E-01    6.856E+01    6.853E+05
6.750E-01   6.750E-01    6.271E+01    6.269E+05
7.250E-01   6.750E-01    4.782E+01    4.781E+05
7.750E-01   6.750E-01    3.394E+01    3.394E+05
8.250E-01   6.750E-01    2.597E+01    2.598E+05
8.750E-01   6.750E-01    2.344E+01    2.346E+05
9.250E-01   6.750E-01    2.457E+01    2.459E+05
9.750E-01   6.750E-01    2.718E+01    2.719E+05
0.000E+00   7.000E-01    4.413E+01    4.413E+05
5.000E-02   7.000E-01    4.750E+01    4.749E+05
1.000E-01   7.000E-01    5.228E+01    5.226E+05
```

```
1.500E-01   7.000E-01    5.043E+01    5.041E+05
2.000E-01   7.000E-01    4.145E+01    4.144E+05
2.500E-01   7.000E-01    3.106E+01    3.105E+05
3.000E-01   7.000E-01    2.408E+01    2.409E+05
3.500E-01   7.000E-01    2.128E+01    2.130E+05
4.000E-01   7.000E-01    2.182E+01    2.183E+05
4.500E-01   7.000E-01    2.585E+01    2.586E+05
5.000E-01   7.000E-01    3.448E+01    3.448E+05
5.500E-01   7.000E-01    4.679E+01    4.678E+05
6.000E-01   7.000E-01    5.714E+01    5.712E+05
6.500E-01   7.000E-01    5.869E+01    5.867E+05
7.000E-01   7.000E-01    5.114E+01    5.113E+05
7.500E-01   7.000E-01    4.103E+01    4.102E+05
8.000E-01   7.000E-01    3.413E+01    3.414E+05
8.500E-01   7.000E-01    3.154E+01    3.155E+05
9.000E-01   7.000E-01    3.193E+01    3.194E+05
9.500E-01   7.000E-01    3.397E+01    3.398E+05
1.000E+00   7.000E-01    3.539E+01    3.539E+05
2.500E-02   7.250E-01    3.731E+01    3.731E+05
7.500E-02   7.250E-01    4.047E+01    4.047E+05
1.250E-01   7.250E-01    4.183E+01    4.182E+05
1.750E-01   7.250E-01    3.870E+01    3.869E+05
2.250E-01   7.250E-01    3.321E+01    3.321E+05
2.750E-01   7.250E-01    2.908E+01    2.908E+05
3.250E-01   7.250E-01    2.761E+01    2.762E+05
3.750E-01   7.250E-01    2.779E+01    2.780E+05
4.250E-01   7.250E-01    2.884E+01    2.885E+05
4.750E-01   7.250E-01    3.170E+01    3.170E+05
5.250E-01   7.250E-01    3.744E+01    3.744E+05
5.750E-01   7.250E-01    4.455E+01    4.454E+05
6.250E-01   7.250E-01    4.883E+01    4.882E+05
6.750E-01   7.250E-01    4.782E+01    4.781E+05
7.250E-01   7.250E-01    4.392E+01    4.391E+05
7.750E-01   7.250E-01    4.136E+01    4.136E+05
8.250E-01   7.250E-01    4.158E+01    4.159E+05
8.750E-01   7.250E-01    4.315E+01    4.316E+05
9.250E-01   7.250E-01    4.458E+01    4.458E+05
9.750E-01   7.250E-01    4.567E+01    4.567E+05
0.000E+00   7.500E-01    2.943E+01    2.944E+05
5.000E-02   7.500E-01    3.026E+01    3.026E+05
1.000E-01   7.500E-01    3.165E+01    3.165E+05
1.500E-01   7.500E-01    3.175E+01    3.175E+05
2.000E-01   7.500E-01    3.086E+01    3.086E+05
2.500E-01   7.500E-01    3.106E+01    3.106E+05
3.000E-01   7.500E-01    3.336E+01    3.336E+05
3.500E-01   7.500E-01    3.607E+01    3.607E+05
4.000E-01   7.500E-01    3.676E+01    3.676E+05
4.500E-01   7.500E-01    3.540E+01    3.539E+05
5.000E-01   7.500E-01    3.442E+01    3.441E+05
5.500E-01   7.500E-01    3.558E+01    3.558E+05
6.000E-01   7.500E-01    3.796E+01    3.795E+05
6.500E-01   7.500E-01    3.968E+01    3.968E+05
7.000E-01   7.500E-01    4.103E+01    4.102E+05
7.500E-01   7.500E-01    4.429E+01    4.428E+05
8.000E-01   7.500E-01    5.034E+01    5.034E+05
8.500E-01   7.500E-01    5.664E+01    5.664E+05
9.000E-01   7.500E-01    5.974E+01    5.973E+05
9.500E-01   7.500E-01    5.934E+01    5.933E+05
1.000E+00   7.500E-01    5.860E+01    5.860E+05
2.500E-02   7.750E-01    2.274E+01    2.275E+05
7.500E-02   7.750E-01    2.296E+01    2.297E+05
1.250E-01   7.750E-01    2.364E+01    2.365E+05
1.750E-01   7.750E-01    2.512E+01    2.513E+05
2.250E-01   7.750E-01    2.881E+01    2.881E+05
2.750E-01   7.750E-01    3.567E+01    3.566E+05
3.250E-01   7.750E-01    4.362E+01    4.361E+05
3.750E-01   7.750E-01    4.774E+01    4.773E+05
4.250E-01   7.750E-01    4.515E+01    4.514E+05
4.750E-01   7.750E-01    3.846E+01    3.845E+05
5.250E-01   7.750E-01    3.265E+01    3.265E+05
5.750E-01   7.750E-01    3.017E+01    3.018E+05
```

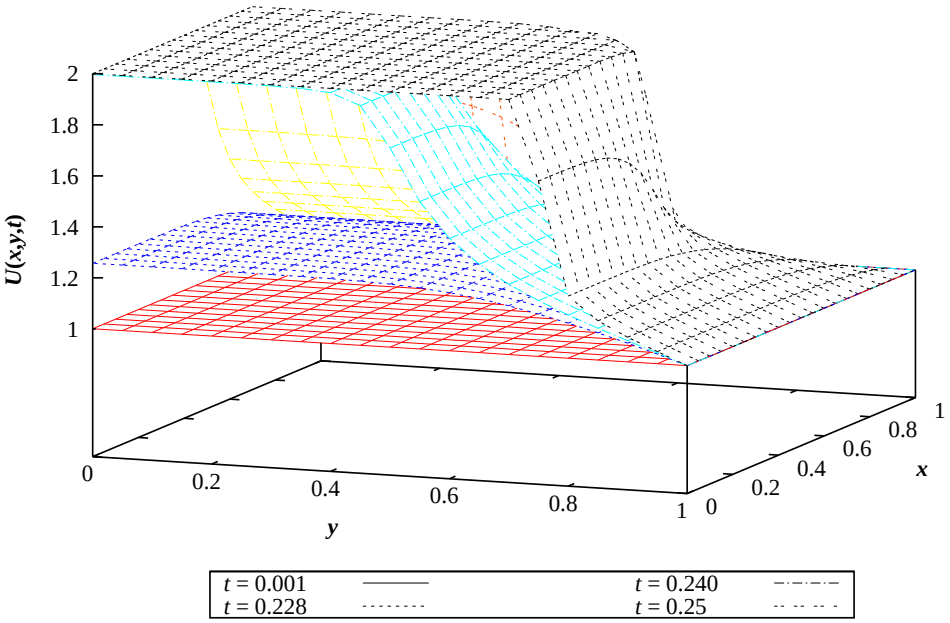
```
6.250E-01   7.750E-01    3.068E+01    3.068E+05
6.750E-01   7.750E-01    3.394E+01    3.394E+05
7.250E-01   7.750E-01    4.136E+01    4.136E+05
7.750E-01   7.750E-01    5.388E+01    5.387E+05
8.250E-01   7.750E-01    6.826E+01    6.825E+05
8.750E-01   7.750E-01    7.755E+01    7.754E+05
9.250E-01   7.750E-01    7.785E+01    7.784E+05
9.750E-01   7.750E-01    7.362E+01    7.361E+05
0.000E+00   8.000E-01    1.712E+01    1.713E+05
5.000E-02   8.000E-01    1.679E+01    1.681E+05
1.000E-01   8.000E-01    1.682E+01    1.684E+05
1.500E-01   8.000E-01    1.852E+01    1.853E+05
2.000E-01   8.000E-01    2.340E+01    2.341E+05
2.500E-01   8.000E-01    3.316E+01    3.316E+05
3.000E-01   8.000E-01    4.678E+01    4.676E+05
3.500E-01   8.000E-01    5.785E+01    5.783E+05
4.000E-01   8.000E-01    5.874E+01    5.872E+05
4.500E-01   8.000E-01    4.914E+01    4.913E+05
5.000E-01   8.000E-01    3.647E+01    3.647E+05
5.500E-01   8.000E-01    2.753E+01    2.754E+05
6.000E-01   8.000E-01    2.415E+01    2.416E+05
6.500E-01   8.000E-01    2.603E+01    2.604E+05
7.000E-01   8.000E-01    3.413E+01    3.414E+05
7.500E-01   8.000E-01    5.034E+01    5.034E+05
8.000E-01   8.000E-01    7.292E+01    7.291E+05
8.500E-01   8.000E-01    9.274E+01    9.272E+05
9.000E-01   8.000E-01    9.928E+01    9.926E+05
9.500E-01   8.000E-01    9.289E+01    9.288E+05
1.000E+00   8.000E-01    8.753E+01    8.752E+05
2.500E-02   8.250E-01    1.266E+01    1.266E+05
7.500E-02   8.250E-01    1.211E+01    1.213E+05
1.250E-01   8.250E-01    1.302E+01    1.304E+05
1.750E-01   8.250E-01    1.719E+01    1.721E+05
2.250E-01   8.250E-01    2.704E+01    2.704E+05
2.750E-01   8.250E-01    4.371E+01    4.371E+05
3.250E-01   8.250E-01    6.229E+01    6.226E+05
3.750E-01   8.250E-01    7.139E+01    7.136E+05
4.250E-01   8.250E-01    6.413E+01    6.411E+05
4.750E-01   8.250E-01    4.672E+01    4.672E+05
5.250E-01   8.250E-01    3.066E+01    3.066E+05
5.750E-01   8.250E-01    2.172E+01    2.174E+05
6.250E-01   8.250E-01    2.013E+01    2.015E+05
6.750E-01   8.250E-01    2.597E+01    2.598E+05
7.250E-01   8.250E-01    4.158E+01    4.159E+05
7.750E-01   8.250E-01    6.826E+01    6.825E+05
8.250E-01   8.250E-01    9.880E+01    9.878E+05
8.750E-01   8.250E-01    1.174E+02    1.174E+06
9.250E-01   8.250E-01    1.151E+02    1.151E+06
9.750E-01   8.250E-01    1.029E+02    1.029E+06
0.000E+00   8.500E-01    9.594E+00    9.601E+04
5.000E-02   8.500E-01    9.182E+00    9.200E+04
1.000E-01   8.500E-01    9.289E+00    9.317E+04
1.500E-01   8.500E-01    1.204E+01    1.207E+05
2.000E-01   8.500E-01    1.997E+01    1.998E+05
2.500E-01   8.500E-01    3.594E+01    3.594E+05
3.000E-01   8.500E-01    5.863E+01    5.861E+05
3.500E-01   8.500E-01    7.728E+01    7.725E+05
4.000E-01   8.500E-01    7.834E+01    7.831E+05
4.500E-01   8.500E-01    6.130E+01    6.128E+05
5.000E-01   8.500E-01    3.934E+01    3.934E+05
5.500E-01   8.500E-01    2.390E+01    2.391E+05
6.000E-01   8.500E-01    1.758E+01    1.760E+05
6.500E-01   8.500E-01    1.963E+01    1.966E+05
7.000E-01   8.500E-01    3.154E+01    3.155E+05
7.500E-01   8.500E-01    5.664E+01    5.664E+05
8.000E-01   8.500E-01    9.274E+01    9.272E+05
8.500E-01   8.500E-01    1.247E+02    1.246E+06
9.000E-01   8.500E-01    1.341E+02    1.340E+06
9.500E-01   8.500E-01    1.215E+02    1.215E+06
1.000E+00   8.500E-01    1.115E+02    1.115E+06
2.500E-02   8.750E-01    7.348E+00    7.358E+04
```

```
 7.500E-02   8.750E-01    7.095E+00    7.120E+04
 1.250E-01   8.750E-01    8.572E+00    8.603E+04
 1.750E-01   8.750E-01    1.407E+01    1.409E+05
 2.250E-01   8.750E-01    2.685E+01    2.685E+05
 2.750E-01   8.750E-01    4.876E+01    4.875E+05
 3.250E-01   8.750E-01    7.345E+01    7.342E+05
 3.750E-01   8.750E-01    8.551E+01    8.548E+05
 4.250E-01   8.750E-01    7.551E+01    7.548E+05
 4.750E-01   8.750E-01    5.199E+01    5.197E+05
 5.250E-01   8.750E-01    3.056E+01    3.057E+05
 5.750E-01   8.750E-01    1.871E+01    1.873E+05
 6.250E-01   8.750E-01    1.635E+01    1.638E+05
 6.750E-01   8.750E-01    2.344E+01    2.346E+05
 7.250E-01   8.750E-01    4.315E+01    4.316E+05
 7.750E-01   8.750E-01    7.755E+01    7.754E+05
 8.250E-01   8.750E-01    1.174E+02    1.174E+06
 8.750E-01   8.750E-01    1.416E+02    1.416E+06
 9.250E-01   8.750E-01    1.379E+02    1.379E+06
 9.750E-01   8.750E-01    1.211E+02    1.211E+06
 0.000E+00   9.000E-01    6.033E+00    6.040E+04
 5.000E-02   9.000E-01    5.901E+00    5.918E+04
 1.000E-01   9.000E-01    6.561E+00    6.590E+04
 1.500E-01   9.000E-01    1.005E+01    1.007E+05
 2.000E-01   9.000E-01    1.916E+01    1.918E+05
 2.500E-01   9.000E-01    3.702E+01    3.702E+05
 3.000E-01   9.000E-01    6.202E+01    6.200E+05
 3.500E-01   9.000E-01    8.238E+01    8.235E+05
 4.000E-01   9.000E-01    8.350E+01    8.347E+05
 4.500E-01   9.000E-01    6.485E+01    6.483E+05
 5.000E-01   9.000E-01    4.064E+01    4.064E+05
 5.500E-01   9.000E-01    2.342E+01    2.344E+05
 6.000E-01   9.000E-01    1.630E+01    1.633E+05
 6.500E-01   9.000E-01    1.860E+01    1.863E+05
 7.000E-01   9.000E-01    3.193E+01    3.194E+05
 7.500E-01   9.000E-01    5.974E+01    5.973E+05
 8.000E-01   9.000E-01    9.928E+01    9.926E+05
 8.500E-01   9.000E-01    1.341E+02    1.340E+06
 9.000E-01   9.000E-01    1.446E+02    1.446E+06
 9.500E-01   9.000E-01    1.316E+02    1.316E+06
 1.000E+00   9.000E-01    1.211E+02    1.210E+06
 2.500E-02   9.250E-01    5.237E+00    5.245E+04
 7.500E-02   9.250E-01    5.522E+00    5.542E+04
 1.250E-01   9.250E-01    7.650E+00    7.675E+04
 1.750E-01   9.250E-01    1.381E+01    1.383E+05
 2.250E-01   9.250E-01    2.695E+01    2.695E+05
 2.750E-01   9.250E-01    4.818E+01    4.817E+05
 3.250E-01   9.250E-01    7.114E+01    7.112E+05
 3.750E-01   9.250E-01    8.219E+01    8.216E+05
 4.250E-01   9.250E-01    7.322E+01    7.320E+05
 4.750E-01   9.250E-01    5.155E+01    5.154E+05
 5.250E-01   9.250E-01    3.106E+01    3.106E+05
 5.750E-01   9.250E-01    1.927E+01    1.929E+05
 6.250E-01   9.250E-01    1.700E+01    1.702E+05
 6.750E-01   9.250E-01    2.457E+01    2.459E+05
 7.250E-01   9.250E-01    4.458E+01    4.458E+05
 7.750E-01   9.250E-01    7.785E+01    7.784E+05
 8.250E-01   9.250E-01    1.151E+02    1.151E+06
 8.750E-01   9.250E-01    1.379E+02    1.379E+06
 9.250E-01   9.250E-01    1.362E+02    1.361E+06
 9.750E-01   9.250E-01    1.225E+02    1.225E+06
 0.000E+00   9.500E-01    4.825E+00    4.829E+04
 5.000E-02   9.500E-01    5.010E+00    5.021E+04
 1.000E-01   9.500E-01    6.305E+00    6.322E+04
 1.500E-01   9.500E-01    1.043E+01    1.045E+05
 2.000E-01   9.500E-01    1.973E+01    1.974E+05
 2.500E-01   9.500E-01    3.612E+01    3.612E+05
 3.000E-01   9.500E-01    5.724E+01    5.723E+05
 3.500E-01   9.500E-01    7.359E+01    7.357E+05
 4.000E-01   9.500E-01    7.465E+01    7.463E+05
 4.500E-01   9.500E-01    6.004E+01    6.003E+05
 5.000E-01   9.500E-01    4.000E+01    4.000E+05
```

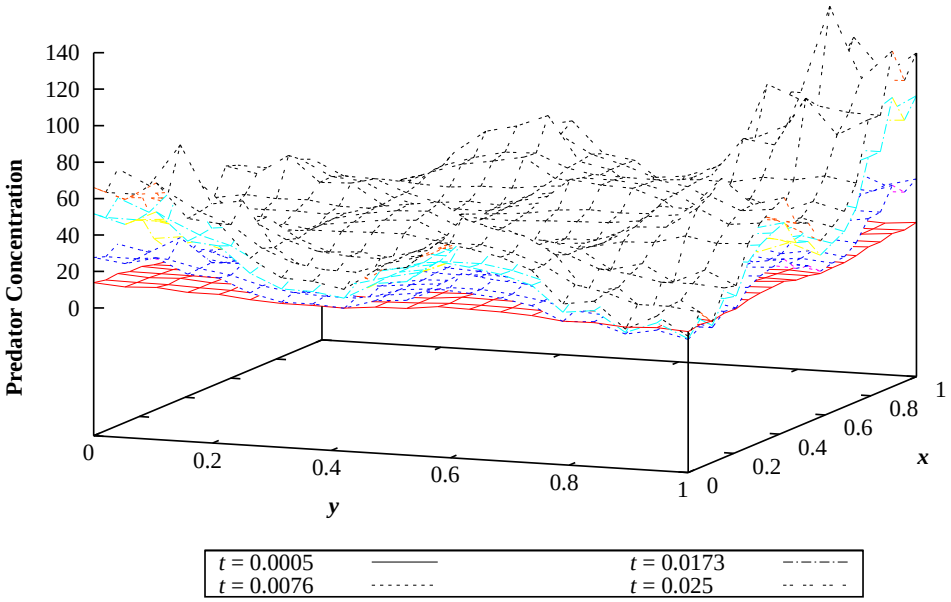
```
  5.500E-01   9.500E-01    2.472E+01    2.473E+05
  6.000E-01   9.500E-01    1.813E+01    1.814E+05
  6.500E-01   9.500E-01    2.076E+01    2.078E+05
  7.000E-01   9.500E-01    3.397E+01    3.398E+05
  7.500E-01   9.500E-01    5.934E+01    5.933E+05
  8.000E-01   9.500E-01    9.289E+01    9.288E+05
  8.500E-01   9.500E-01    1.215E+02    1.215E+06
  9.000E-01   9.500E-01    1.316E+02    1.316E+06
  9.500E-01   9.500E-01    1.240E+02    1.240E+06
  1.000E+00   9.500E-01    1.173E+02    1.173E+06
  2.500E-02   9.750E-01    4.739E+00    4.742E+04
  7.500E-02   9.750E-01    5.505E+00    5.512E+04
  1.250E-01   9.750E-01    8.236E+00    8.245E+04
  1.750E-01   9.750E-01    1.479E+01    1.480E+05
  2.250E-01   9.750E-01    2.714E+01    2.714E+05
  2.750E-01   9.750E-01    4.500E+01    4.499E+05
  3.250E-01   9.750E-01    6.278E+01    6.277E+05
  3.750E-01   9.750E-01    7.107E+01    7.105E+05
  4.250E-01   9.750E-01    6.474E+01    6.473E+05
  4.750E-01   9.750E-01    4.841E+01    4.840E+05
  5.250E-01   9.750E-01    3.169E+01    3.169E+05
  5.750E-01   9.750E-01    2.131E+01    2.132E+05
  6.250E-01   9.750E-01    1.949E+01    1.950E+05
  6.750E-01   9.750E-01    2.718E+01    2.719E+05
  7.250E-01   9.750E-01    4.567E+01    4.567E+05
  7.750E-01   9.750E-01    7.362E+01    7.361E+05
  8.250E-01   9.750E-01    1.029E+02    1.029E+06
  8.750E-01   9.750E-01    1.211E+02    1.211E+06
  9.250E-01   9.750E-01    1.225E+02    1.224E+06
  9.750E-01   9.750E-01    1.154E+02    1.154E+06
  0.000E+00   1.000E+00    4.607E+00    4.608E+04
  5.000E-02   1.000E+00    4.957E+00    4.961E+04
  1.000E-01   1.000E+00    6.594E+00    6.600E+04
  1.500E-01   1.000E+00    1.106E+01    1.106E+05
  2.000E-01   1.000E+00    2.031E+01    2.031E+05
  2.500E-01   1.000E+00    3.540E+01    3.540E+05
  3.000E-01   1.000E+00    5.355E+01    5.354E+05
  3.500E-01   1.000E+00    6.699E+01    6.698E+05
  4.000E-01   1.000E+00    6.800E+01    6.799E+05
  4.500E-01   1.000E+00    5.629E+01    5.629E+05
  5.000E-01   1.000E+00    3.940E+01    3.940E+05
  5.500E-01   1.000E+00    2.575E+01    2.575E+05
  6.000E-01   1.000E+00    1.967E+01    1.967E+05
  6.500E-01   1.000E+00    2.250E+01    2.250E+05
  7.000E-01   1.000E+00    3.539E+01    3.539E+05
  7.500E-01   1.000E+00    5.860E+01    5.860E+05
  8.000E-01   1.000E+00    8.753E+01    8.752E+05
  8.500E-01   1.000E+00    1.115E+02    1.115E+06
  9.000E-01   1.000E+00    1.211E+02    1.210E+06
  9.500E-01   1.000E+00    1.173E+02    1.173E+06
  1.000E+00   1.000E+00    1.132E+02    1.132E+06

 Statistics:
 Time =   0.0250
 Total number of accepted timesteps =   29
 Total number of rejected timesteps =    0

            Total  number (rounded) of
          Residual  Jacobian    Newton   Lin sys
             evals     evals     iters     iters
 At level
       1       400        30        60        90
       2       400        30        50        80
       3       300        20        40        50

             Maximum  number of
              Newton iters    Lin sys iters
```

```
At level
      1              2              2
      2              2              2
      3              2              2
```

Example Program 1
Model for a Single, One-step Reaction of a Mixture of Two Chemicals



Example Program 2
Multispecies Food Web Model
Concentrations of Predator



Multispecies Food Web Model
Concentrations of Prey