NAG Fortran Library Routine Document

E02DAF

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

E02DAF forms a minimal, weighted least-squares bicubic spline surface fit with prescribed knots to a given set of data points.

2 Specification

```
 SUBROUTINE E02DAF(M, PX, PY, X, Y, F, W, LAMDA, MU, POINT, NPOINT, DL,
1                  C, NC, WS, NWS, EPS, SIGMA, RANK, IFAIL)
 INTEGER           M, PX, PY, POINT(NPOINT), NPOINT, NC, NWS, RANK, IFAIL
 real              X(M), Y(M), F(M), W(M), LAMDA(PX), MU(PY), DL(NC),
1                  C(NC), WS(NWS), EPS, SIGMA
```

3 Description

This routine determines a bicubic spline fit $s(x,y)$ to the set of data points (x_r, y_r, f_r) with weights w_r, for $r = 1, 2, \ldots, m$. The two sets of internal knots of the spline, $\{\lambda\}$ and $\{\mu\}$, associated with the variables x and y respectively, are prescribed by the user. These knots can be thought of as dividing the data region of the (x,y) plane into panels (see diagram in Section 5). A bicubic spline consists of a separate bicubic polynomial in each panel, the polynomials joining together with continuity up to the second derivative across the panel boundaries.

$s(x,y)$ has the property that Σ, the sum of squares of its weighted residuals ρ_r, for $r = 1, 2, \ldots, m$, where

$$\rho_r = w_r(s(x_r, y_r) - f_r) \tag{1}$$

is as small as possible for a bicubic spline with the given knot sets. The routine produces this minimized value of Σ and the coefficients c_{ij} in the B-spline representation of $s(x,y)$ - see Section 8. E02DEF and E02DFF are available to compute values of the fitted spline from the coefficients c_{ij}.

The least-squares criterion is not always sufficient to determine the bicubic spline uniquely: there may be a whole family of splines which have the same minimum sum of squares. In these cases, the routine selects from this family the spline for which the sum of squares of the coefficients c_{ij} is smallest: in other words, the minimal least-squares solution. This choice, although arbitrary, reduces the risk of unwanted fluctuations in the spline fit. The method employed involves forming a system of m linear equations in the coefficients c_{ij} and then computing its least-squares solution, which will be the minimal least-squares solution when appropriate. The basis of the method is described in Hayes and Halliday (1974). The matrix of the equation is formed using a recurrence relation for B-splines which is numerically stable (see Cox (1972a) and de Boor (1972) – the former contains the more elementary derivation but, unlike de Boor (1972), does not cover the case of coincident knots). The least-squares solution is also obtained in a stable manner by using orthogonal transformations, viz. a variant of Givens rotation (see Gentleman (1973)). This requires only one row of the matrix to be stored at a time. Advantage is taken of the stepped-band structure which the matrix possesses when the data points are suitably ordered, there being at most sixteen non-zero elements in any row because of the definition of B-splines. First the matrix is reduced to upper triangular form and then the diagonal elements of this triangle are examined in turn. When an element is encountered whose square, divided by the mean squared weight, is less than a threshold ϵ, it is replaced by zero and the rest of the elements in its row are reduced to zero by rotations with the remaining rows. The rank of the system is taken to be the number of non-zero diagonal elements in the final triangle, and the non-zero rows of this triangle are used to compute the minimal least-squares solution. If all the diagonal elements are non-zero, the rank is equal to the number of coefficients c_{ij} and the solution obtained is the ordinary least-squares solution, which is unique in this case.

4 References

Cox M G (1972a) The numerical evaluation of B-splines *J. Inst. Math. Appl.* **10** 134–149

de Boor C (1972) On calculating with B-splines *J. Approx. Theory* **6** 50–62

Gentleman W M (1973) Least-squares computations by Givens transformations without square roots *J. Inst. Math. Applic.* **12** 329–336

Hayes J G and Halliday J (1974) The least-squares fitting of cubic spline surfaces to general data sets *J. Inst. Math. Appl.* **14** 89–103

5 Parameters

1: M – INTEGER *Input*

On entry: the number of data points, m.

Constraint: M > 1.

2: PX – INTEGER *Input*
3: PY – INTEGER *Input*

On entry: the total number of knots λ and μ associated with the variables x and y, respectively.

Constraint: PX ≥ 8 and PY ≥ 8.

(They are such that PX $- 8$ and PY $- 8$ are the corresponding numbers of interior knots.) The running time and storage required by the routine are both minimized if the axes are labelled so that PY is the smaller of PX and PY.

4: X(M) – ***real*** array *Input*
5: Y(M) – ***real*** array *Input*
6: F(M) – ***real*** array *Input*

On entry: the co-ordinates of the data point (x_r, y_r, f_r), for $r = 1, 2, \ldots, m$. The order of the data points is immaterial, but see the array POINT, below.

7: W(M) – ***real*** array *Input*

On entry: the weight w_r of the rth data point. It is important to note the definition of weight implied by the equation (1) in Section 3, since it is also common usage to define weight as the square of this weight. In this routine, each w_r should be chosen inversely proportional to the (absolute) accuracy of the corresponding f_r, as expressed, for example, by the standard deviation or probable error of the f_r. When the f_r are all of the same accuracy, all the w_r may be set equal to 1.0.

8: LAMDA(PX) – ***real*** array *Input/Output*

On entry: LAMDA$(i + 4)$ must contain the ith interior knot λ_{i+4} associated with the variable x, for $i = 1, 2, \ldots,$ PX $- 8$. The knots must be in non-decreasing order and lie strictly within the range covered by the data values of x. A knot is a value of x at which the spline is allowed to be discontinuous in the third derivative with respect to x, though continuous up to the second derivative. This degree of continuity can be reduced, if the user requires, by the use of coincident knots, provided that no more than four knots are chosen to coincide at any point. Two, or three, coincident knots allow loss of continuity in, respectively, the second and first derivative with respect to x at the value of x at which they coincide. Four coincident knots split the spline surface into two independent parts. For choice of knots see Section 8.

On exit: the interior knots LAMDA(5) to LAMDA$($PX $- 4)$ are unchanged, and the segments LAMDA$(1 : 4)$ and LAMDA$($PX $- 3 :$ PX$)$ contain additional (exterior) knots introduced by the routine in order to define the full set of B-splines required. The four knots in the first segment are all set equal to the lowest data value of x and the other four additional knots are all set equal to the highest value: there is experimental evidence that coincident end-knots are best for numerical

accuracy. The complete array must be left undisturbed if E02DEF or E02DFF is to be used subsequently.

9: MU(PX – 4) – ***real*** array *Input/Output*

On entry: MU($i+4$) must contain the ith interior knot μ_{i+4} associated with the variable y, $i = 1, 2, \ldots, \text{PY} - 8$.

On exit: the same remarks apply to MU as to LAMDA above, with Y replacing X, and y replacing x.

10: POINT(NPOINT) – INTEGER array *Input*

On entry: indexing information usually provided by E02ZAF which enables the data points to be accessed in the order which produces the advantageous matrix structure mentioned in Section 3. This order is such that, if the (x, y) plane is thought of as being divided into rectangular panels by the two sets of knots, all data in a panel occur before data in succeeding panels, where the panels are numbered from bottom to top and then left to right with the usual arrangement of axes, as indicated in the diagram.

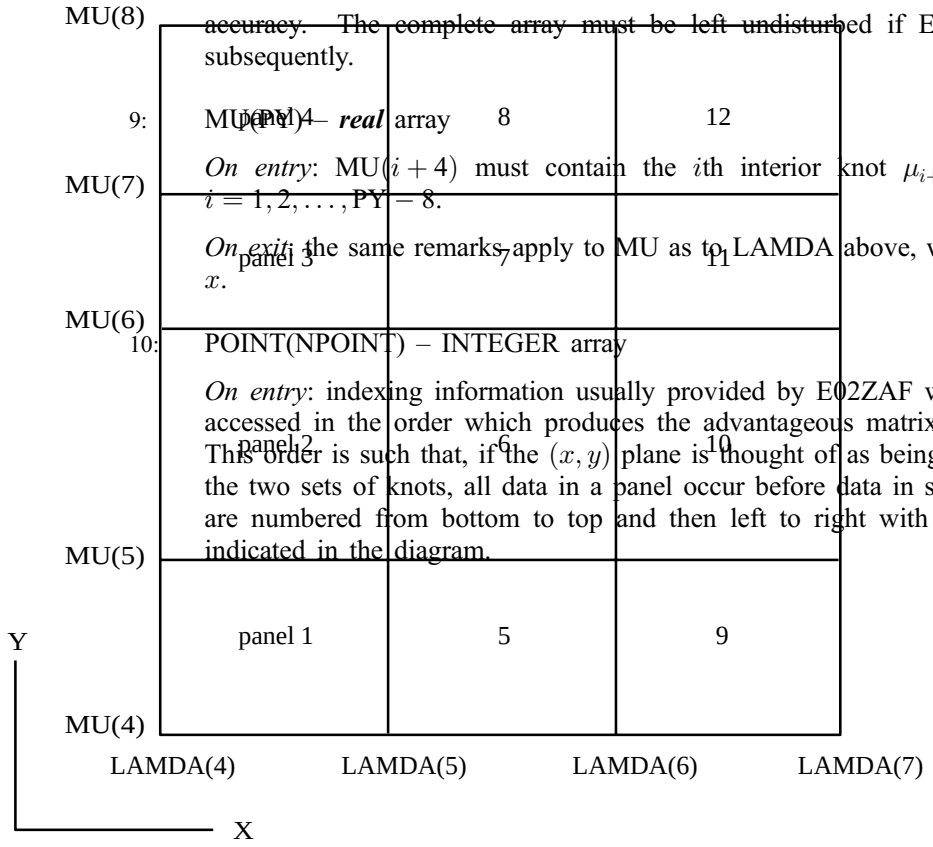

Figure 1

A data point lying exactly on one or more panel sides is considered to be in the highest numbered panel adjacent to the point. E02ZAF should be called to obtain the array POINT, unless it is provided by other means.

11: NPOINT – INTEGER *Input*

On entry: the dimension of the array POINT as declared in the (sub)program from which E02DAF is called.

Constraint: NPOINT $\geq$ M + (PX − 7) × (PY − 7).

12: DL(NC) – ***real*** array *Output*

On exit: DL gives the squares of the diagonal elements of the reduced triangular matrix, divided by the mean squared weight. It includes those elements, less than ϵ, which are treated as zero (see Section 3).

13: C(NC) – ***real*** array *Output*

On exit: C gives the coefficients of the fit. $\mathrm{C}((\mathrm{PY}-4)\times(i-1)+j)$ is the coefficient c_{ij} of Section 3 and Section 8 for $i=1,2,\ldots,\mathrm{PX}-4$ and $j=1,2,\ldots,\mathrm{PY}-4$. These coefficients are used by E02DEF or E02DFF to calculate values of the fitted function.

14: NC – INTEGER *Input*

On entry: the value $(\mathrm{PX}-4)\times(\mathrm{PY}-4)$.

15: WS(NWS) – ***real*** array *Workspace*

16: NWS – INTEGER *Input*

On entry: the dimension of the array WS as declared in the (sub)program from which E02DAF is called.

Constraint: $\mathrm{NWS}\geq(2\times\mathrm{NC}+1)\times(3\times\mathrm{PY}-6)-2$.

17: EPS – ***real*** *Input*

On entry: a threshold ϵ for determining the effective rank of the system of linear equations. The rank is determined as the number of elements of the array DL (see below) which are non-zero. An element of DL is regarded as zero if it is less than ϵ. ***Machine precision*** is a suitable value for ϵ in most practical applications which have only 2 or 3 decimals accurate in data. If some coefficients of the fit prove to be very large compared with the data ordinates, this suggests that ϵ should be increased so as to decrease the rank. The array DL will give a guide to appropriate values of ϵ to achieve this, as well as to the choice of ϵ in other cases where some experimentation may be needed to determine a value which leads to a satisfactory fit.

18: SIGMA – ***real*** *Output*

On exit: Σ, the weighted sum of squares of residuals. This is not computed from the individual residuals but from the right-hand sides of the orthogonally-transformed linear equations. For further details see page 97 of Hayes and Halliday (1974). The two methods of computation are theoretically equivalent, but the results may differ because of rounding error.

19: RANK – INTEGER *Output*

On exit: the rank of the system as determined by the value of the threshold ϵ. When $\mathrm{RANK}=\mathrm{NC}$, the least-squares solution is unique: in other cases the minimal least-squares solution is computed.

20: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: $\mathrm{IFAIL}=0$ unless the routine detects an error (see Section 6).

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, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry $\mathrm{IFAIL}=0$ 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 $= 1$

At least one set of knots is not in non-decreasing order, or an interior knot is outside the range of the data values.

IFAIL $= 2$

More than four knots coincide at a single point, possibly because all data points have the same value of x (or y) or because an interior knot coincides with an extreme data value.

IFAIL $= 3$

Array POINT does not indicate the data points in panel order. Call E02ZAF to obtain a correct array.

IFAIL $= 4$

On entry, $\mathrm{M} \leq 1$,
or $\mathrm{PX} < 8$,
or $\mathrm{PY} < 8$,
or $\mathrm{NC} \neq (\mathrm{PX}-4) \times (\mathrm{PY}-4)$,
or NWS is too small,
or NPOINT is too small.

IFAIL $= 5$

All the weights w_r are zero or rank determined as zero.

7 Accuracy

The computation of the B-splines and reduction of the observation matrix to triangular form are both numerically stable.

8 Further Comments

The time taken by this routine is approximately proportional to the number of data points, m, and to $(3 \times (\mathrm{PY}-4)+4)^2$.

The B-spline representation of the bicubic spline is

$$s(x,y) = \sum_{i,j} c_{ij} M_i(x) N_j(y)$$

summed over $i = 1, 2, \ldots, \mathrm{PX}-4$ and over $j = 1, 2, \ldots, \mathrm{PY}-4$. Here $M_i(x)$ and $N_j(y)$ denote normalised cubic B-splines, the former defined on the knots $\lambda_i, \lambda_{i+1}, \ldots, \lambda_{i+4}$ and the latter on the knots $\mu_j, \mu_{j+1}, \ldots, \mu_{j+4}$. For further details, see Hayes and Halliday (1974) for bicubic splines and de Boor (1972) for normalised B-splines.

The choice of the interior knots, which help to determine the spline's shape, must largely be a matter of trial and error. It is usually best to start with a small number of knots and, examining the fit at each stage, add a few knots at a time at places where the fit is particularly poor. In intervals of x or y where the surface represented by the data changes rapidly, in function value or derivatives, more knots will be needed than elsewhere. In some cases guidance can be obtained by analogy with the case of coincident knots: for example, just as three coincident knots can produce a discontinuity in slope, three close knots can produce rapid change in slope. Of course, such rapid changes in behaviour must be adequately represented by the data points, as indeed must the behaviour of the surface generally, if a satisfactory fit is to be achieved. When there is no rapid change in behaviour, equally-spaced knots will often suffice.

In all cases the fit should be examined graphically before it is accepted as satisfactory.

The fit obtained is not defined outside the rectangle

$$\lambda_4 \leq x \leq \lambda_{\mathrm{PX}-3}, \mu_4 \leq y \leq \mu_{\mathrm{PY}-3}.$$

The reason for taking the extreme data values of x and y for these four knots is that, as is usual in data fitting, the fit cannot be expected to give satisfactory values outside the data region. If, nevertheless, the user requires values over a larger rectangle, this can be achieved by augmenting the data with two artificial data points $(a, c, 0)$ and $(b, d, 0)$ with zero weight, where $a \leq x \leq b$, $c \leq y \leq d$ defines the enlarged rectangle. In the case when the data are adequate to make the least-squares solution unique (RANK = NC), this enlargement will not affect the fit over the original rectangle, except for possibly enlarged rounding errors, and will simply continue the bicubic polynomials in the panels bordering the rectangle out to the new boundaries: in other cases the fit will be affected. Even using the original rectangle there may be regions within it, particularly at its corners, which lie outside the data region and where, therefore, the fit will be unreliable. For example, if there is no data point in panel 1 of the diagram in Section 5, the least-squares criterion leaves the spline indeterminate in this panel: the minimal spline determined by the subroutine in this case passes through the value zero at the point (λ_4, μ_4).

9 Example

This example program reads a value for ϵ, and a set of data points, weights and knot positions. If there are more y knots than x knots, it interchanges the x and y axes. It calls E02ZAF to sort the data points into panel order, E02DAF to fit a bicubic spline to them, and E02DEF to evaluate the spline at the data points.

Finally it prints:

the weighted sum of squares of residuals computed from the linear equations;

the rank determined by E02DAF;

data points, fitted values and residuals in panel order;

the weighted sum of squares of the residuals;

the coefficients of the spline fit.

The program is written to handle any number of data sets.

Note: the data supplied in this example is **not typical** of a realistic problem: the number of data points would normally be much larger (in which case the array dimensions and the value of NWS in the program would have to be increased); and the value of ϵ would normally be much smaller on most machines (see E02DAF; the relatively large value of 10^{-6} has been chosen in order to illustrate a minimal least-squares solution when RANK $<$ NC; in this example NC $= 24$).

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     E02DAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MMAX, MAXPX, MAXPY, NCMAX, IP, NIWS, NWS, NADRES,
     +                 NPTMAX
      PARAMETER        (MMAX=40,MAXPX=10,MAXPY=10,NCMAX=(MAXPX-4)
     +                 *(MAXPY-4),IP=3*(MAXPY-4)+4,NIWS=MAXPY-4,
     +                 NWS=2*NCMAX*(IP+2)+IP,NADRES=(MAXPX-7)*(MAXPY-7),
     +                 NPTMAX=MMAX+NADRES)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             EPS, SIGMA, SUM, TEMP
      INTEGER          I, IADRES, IFAIL, ITEMP, J, M, NC, NP, PX, PY,
     +                 RANK
*     .. Local Arrays ..
      real             C(NCMAX), DL(NCMAX), F(MMAX), FF(MMAX),
     +                 LAMDA(MAXPX), MU(MAXPY), W(MMAX), WS(NWS),
```

```
     +                X(MMAX), Y(MMAX)
      INTEGER         ADRES(NADRES), IWS(NIWS), POINT(NPTMAX)
      CHARACTER*1     LABEL(2)
*     .. External Subroutines ..
      EXTERNAL        E02DAF, E02DEF, E02ZAF
*     .. Data statements ..
      DATA            LABEL/'X', 'Y'/
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E02DAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 READ (NIN,*,END=140) EPS
*     Read data, interchanging X and Y axes if PX.LT.PY
      READ (NIN,*) M
      IF (M.LE.MMAX .AND. M.GT.0) THEN
         READ (NIN,*) PX, PY
         IF (PX.GE.8 .AND. PX.LE.MAXPX .AND. PY.GE.8 .AND. PY.LE.MAXPY)
     +       THEN
            IF (PX.LT.PY) THEN
               ITEMP = PX
               PX = PY
               PY = ITEMP
               ITEMP = 1
               READ (NIN,*) (Y(I),X(I),F(I),W(I),I=1,M)
               IF (PY.GT.8) READ (NIN,*) (MU(J),J=5,PY-4)
               IF (PX.GT.8) READ (NIN,*) (LAMDA(J),J=5,PX-4)
            ELSE
               ITEMP = 0
               READ (NIN,*) (X(I),Y(I),F(I),W(I),I=1,M)
               IF (PX.GT.8) READ (NIN,*) (LAMDA(J),J=5,PX-4)
               IF (PY.GT.8) READ (NIN,*) (MU(J),J=5,PY-4)
            END IF
            NC = (PX-4)*(PY-4)
            NP = (PX-7)*(PY-7)
            WRITE (NOUT,*)
            WRITE (NOUT,99995) 'Interior ', LABEL(ITEMP+1), '-knots'
            DO 40 J = 5, PX - 4
               WRITE (NOUT,99996) LAMDA(J)
   40       CONTINUE
            IF (PX.EQ.8) WRITE (NOUT,*) 'None'
            WRITE (NOUT,*)
            WRITE (NOUT,99995) 'Interior ', LABEL(2-ITEMP), '-knots'
            DO 60 J = 5, PY - 4
               WRITE (NOUT,99996) MU(J)
   60       CONTINUE
            IF (PY.EQ.8) WRITE (NOUT,*) 'None'
*           Sort points into panel order
            IFAIL = 0
*
            CALL E02ZAF(PX,PY,LAMDA,MU,M,X,Y,POINT,NPTMAX,ADRES,NP,
     +                  IFAIL)
*
*           Fit bicubic spline to data points
            IFAIL = 0
*
            CALL E02DAF(M,PX,PY,X,Y,F,W,LAMDA,MU,POINT,NPTMAX,DL,C,NC,
     +                  WS,NWS,EPS,SIGMA,RANK,IFAIL)
*
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'Sum of squares of residual RHS', SIGMA
            WRITE (NOUT,*)
            WRITE (NOUT,99998) 'Rank', RANK
*           Evaluate spline at the data points
            IFAIL = 0
*
            CALL E02DEF(M,PX,PY,X,Y,LAMDA,MU,C,FF,WS,IWS,IFAIL)
*
            SUM = 0
            IF (ITEMP.EQ.1) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'X and Y have been interchanged'
```

```
            END IF
*           Output data points, fitted values and residuals
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        '      X          Y          Data       Fit     Residual'
            DO 100 I = 1, NP
               IADRES = I + M
   80          IADRES = POINT(IADRES)
               IF (IADRES.LE.0) GO TO 100
               TEMP = FF(IADRES) - F(IADRES)
               WRITE (NOUT,99997) X(IADRES), Y(IADRES), F(IADRES),
     +           FF(IADRES), TEMP
               SUM = SUM + (TEMP*W(IADRES))**2
               GO TO 80
  100       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'Sum of squared residuals', SUM
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Spline coefficients'
            DO 120 I = 1, PX - 4
               WRITE (NOUT,99996) (C((I-1)*(PY-4)+J),J=1,PY-4)
  120       CONTINUE
            GO TO 20
         END IF
      END IF
  140 STOP
*
99999 FORMAT (1X,A,1P,e16.2)
99998 FORMAT (1X,A,I5)
99997 FORMAT (1X,4F11.4,e11.2)
99996 FORMAT (1X,6F11.4)
99995 FORMAT (1X,A,A1,A)
      END
```

9.2 Program Data

```
E02DAF Example Program Data
0.000001
  30
   8
  10
   -0.52    0.60    0.93   10.
   -0.61   -0.95   -1.79   10.
    0.93    0.87    0.36   10.
    0.09    0.84    0.52   10.
    0.88    0.17    0.49   10.
   -0.70   -0.87   -1.76   10.
    1.00    1.00    0.33    1.
    1.00    0.10    0.48    1.
    0.30    0.24    0.65    1.
   -0.77   -0.77   -1.82    1.
   -0.23    0.32    0.92    1.
   -1.00    1.00    1.00    1.
   -0.26   -0.63    8.88    1.
   -0.83   -0.66   -2.01    1.
    0.22    0.93    0.47    1.
    0.89    0.15    0.49    1.
   -0.80    0.99    0.84    1.
   -0.88   -0.54   -2.42    1.
    0.68    0.44    0.47    1.
   -0.14   -0.72    7.15    1.
    0.67    0.63    0.44    1.
   -0.90   -0.40   -3.34    1.
   -0.84    0.20    2.78    1.
    0.84    0.43    0.44    1.
    0.15    0.28    0.70    1.
   -0.91   -0.24   -6.52    1.
   -0.35    0.86    0.66    1.
   -0.16   -0.41    2.32    1.
   -0.35   -0.05    1.66    1.
```

```
  -1.00   -1.00   -1.00    1.
 -0.5
  0.0
```

9.3 Program Results

```
 E02DAF Example Program Results

 Interior Y-knots
     -0.5000
      0.0000

 Interior X-knots
 None

 Sum of squares of residual RHS        1.47E+01

 Rank   22

 X and Y have been interchanged

       X          Y          Data       Fit      Residual
    -0.9500    -0.6100    -1.7900    -1.7931  -0.31E-02
    -0.8700    -0.7000    -1.7600    -1.7521   0.79E-02
    -0.7700    -0.7700    -1.8200    -2.4301  -0.61E+00
    -0.6300    -0.2600     8.8800     7.6346  -0.12E+01
    -0.6600    -0.8300    -2.0100    -1.5815   0.43E+00
    -0.5400    -0.8800    -2.4200    -2.6795  -0.26E+00
    -0.7200    -0.1400     7.1500     7.5708   0.42E+00
    -1.0000    -1.0000    -1.0000    -1.0228  -0.23E-01
    -0.4000    -0.9000    -3.3400    -4.6955  -0.14E+01
    -0.2400    -0.9100    -6.5200    -4.7072   0.18E+01
    -0.4100    -0.1600     2.3200     2.7039   0.38E+00
    -0.0500    -0.3500     1.6600     2.2865   0.63E+00
     0.6000    -0.5200     0.9300     0.9441   0.14E-01
     0.8700     0.9300     0.3600     0.3529  -0.71E-02
     0.8400     0.0900     0.5200     0.5024  -0.18E-01
     0.1700     0.8800     0.4900     0.4705  -0.20E-01
     1.0000     1.0000     0.3300     0.6315   0.30E+00
     0.1000     1.0000     0.4800     1.4910   0.10E+01
     0.2400     0.3000     0.6500     0.9241   0.27E+00
     0.3200    -0.2300     0.9200    -0.3692  -0.13E+01
     1.0000    -1.0000     1.0000     1.0835   0.84E-01
     0.9300     0.2200     0.4700     1.4912   0.10E+01
     0.1500     0.8900     0.4900     0.4414  -0.49E-01
     0.9900    -0.8000     0.8400     0.5495  -0.29E+00
     0.4400     0.6800     0.4700     1.5862   0.11E+01
     0.6300     0.6700     0.4400     0.6288   0.19E+00
     0.2000    -0.8400     2.7800     1.7123  -0.11E+01
     0.4300     0.8400     0.4400     0.6888   0.25E+00
     0.2800     0.1500     0.7000     0.7713   0.71E-01
     0.8600    -0.3500     0.6600     0.9347   0.27E+00

 Sum of squared residuals       1.47E+01

 Spline coefficients
    -1.0228   115.4668  -433.5558   -68.1973
    24.8426  -140.1485   258.5042    15.6756
   -29.4878   132.2933  -173.5103    20.0983
     9.9575   -51.6200    67.6666    -5.8765
    10.0577     4.7543   -15.3533    -0.3260
     1.0835    -2.7932     7.7708     0.6315
```