1 SUMMARY

Given the \( n \) points \( x_1 \leq x_2 \leq \ldots \leq x_n \) and a value of \( x \), this subroutine computes the values of the \( k \) B-splines of degree \( k-1 \)

\[ M_{k,i}(x) \quad 1 \leq i \leq n-k, \]

which have knots at the points \( x_1, x_{i+1}, \ldots, x_{i+k} \) and which cover the point \( x \), and it also computes the values of the corresponding integrals

\[ \int_{x_i}^{x} M_{k,i}(\xi) \, d\xi, \quad 1 \leq i \leq n-k. \]

The subroutine uses the recurrence relation due to De Boor, ‘On Calculating with B-splines’, J. Approx. Theory, 6, (1972), to evaluate the B-spline. The integrals are evaluated using the method due to Gaffney, AERE CSS.10, (1974).


2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

The single precision version:

\[ \text{CALL TG04A}(N, X, K, NORM, XVALUE, JINT, V, VINT) \]

The double precision version:

\[ \text{CALL TG04AD}(N, X, K, NORM, XVALUE, JINT, V, VINT) \]

\( N \) is an INTEGER variable which must be set by the user to \( n \) the number of points \( x_i \), \( i=1,2,\ldots,n \). This argument is not altered by the subroutine. **Restriction:** \( n > 0 \).

\( X \) is a REAL (DOUBLE PRECISION in the D version) array of length at least \( n \), which must be set by the user to the values \( x_i \), \( i=1,2,\ldots,n \). This argument is not altered by the subroutine. **Restriction:** these must be ordered so that \( x_1 \leq x_2 \leq \ldots \leq x_n \) and \( x_i < x_{i+k}, i=1,2,\ldots,n-k \).

\( K \) is an INTEGER variable which must be set by the user to the value \( k \), where \( k \) is the order of the B-spline \( M_{k,i}(x) \). This argument is not altered by the subroutine. **Restriction:** the value of \( k \) must be in the range \( 1 \leq k \leq n-1 \).

\( NORM \) is an INTEGER variable which must be set by the user to the value 1 or 2 to indicate which normalization is to be used when computing the B-spline values.

- When \( NORM=1 \) the subroutine evaluates B-splines \( M_{k,i}(x) \), \( 1 \leq i \leq n-k \), which are normalized so that

\[ \int_{x_i}^{x} M_{k,i}(\xi) \, d\xi = k^{-1} \]

- When \( NORM=2 \) the subroutine evaluates B-splines \( N_{k,i}(x) \), \( 1 \leq i \leq n-k \), which are normalized so that

\[ \sum_i N_{k,i}(x) = 1. \text{ N.B.} \]

\[ N_{k,i}(x) = (x_{i+k} - x_i) M_{k,i}(x) \]

If the value of \( NORM \) on entry to the subroutine is different from 1 or 2, a diagnostic message is printed, NORM...
is set to 1 and the calculation is continued.

XVALUE is a REAL (DOUBLE PRECISION in the D version) variable which is set by the user to the value of the argument \( x \) at which the B-splines are to be evaluated. This argument is not altered by the subroutine.

JINT is an INTEGER variable which is set by the subroutine to indicate the knot interval containing \( x \) and is set in the following way.

(i) For \( x < x_n \), JINT is the unique integer such that
\[
 x_{JINT} \leq x < x_{JINT+1}.
\]

(ii) For \( x = x_n \), JINT is set to \( n \) if \( x_{n-1} < x_n \), otherwise JINT is set so that
\[
 x_{JINT} < x \leq x_{JINT+1} = x_n.
\]

(iii) If \( x < x_1 \) then JINT is set to 1, and if \( x > x_n \) then JINT is set to \( n \).

\( V \) is a REAL (DOUBLE PRECISION in the D version) array of length at least \( k \) which is set by the subroutine so that on return \( V(J), J=1,K \) contains the values of \( M_{k,JINT-kj}^j(x) \), \( j=1,2,\ldots,k \), in the case when \( 1 \leq JINT-kj \leq n-k \), and is set to zero otherwise. All the values of the other B-splines at the point \( x \) are zero.

\( VINT \) is a REAL (DOUBLE PRECISION in the D version) array of length at least \( k \) which is set by the subroutine to the values of the integrals.

When \( x \) is in the interval \( x_1 \leq x \leq x_n \), \( VINT(J), J=1,K \) is set to the value of
\[
 \int_{x_{JINT-kj}}^{x} M_{k,JINT-kj}^j(\xi) \, d\xi \quad j=1,2,\ldots,k
\]

in the case when \( 1 \leq JINT-kj \leq n-k \) and in all other cases is set to zero.

If \( x < x_1 \) then the contents of \( VINT \) are set to zero. If \( x \geq x_n \) the integral
\[
 \int_{x_{JINT-kj}}^{x} M_{k,n-k}^j(\xi) \, d\xi
\]

has the value \( k^{-1} \) and this is stored in \( VINT(1) \), with \( VINT(J)=0, J=2,K \).

### 2.2 Checks on the input parameters

The restrictions imposed on the input parameters are checked by the subroutine. For example, the value of \( k \) is checked to see if it satisfies the inequality \( 1 \leq k \leq n-1 \). If the restrictions are not satisfied an error diagnostic is printed, an error return flag, IFAIL, is set (see §2.3), and a return is made to the calling program.

### 2.3 The Common area and diagnostic messages

The subroutine uses a Common area which the user may also reference. To do this the calling program should include a Common statement of the form

*The single precision version:*

```fortran
COMMON/TG04B/ LP,IFAIL
```

*The double precision version:*

```fortran
COMMON/TG04BD/ LP,IFAIL
```

LP is an INTEGER variable which specifies the Fortran stream number to be used for the error messages. The default value is 6 (line printer). To suppress the printing of error messages set LP to zero.

IFAIL is an INTEGER variable which is always set by the subroutine to indicate success or failure. On exit from the subroutine IFAIL will take one of the following values.
0 successful entry
1 \( n < k + 1 \)
2 \( k < 1 \)
3 the data points are not in ascending order
4 more than \( k \) data points coalesce
5 \( \text{NORM} \) was not equal to 1 or 2 (it was reset to 1)

3 GENERAL INFORMATION

Use of Common: the subroutine uses a Common area TG04B/BD, see §2.3.
Workspace: none.
Other subroutines: none.
Input/Output: In the event of errors diagnostic messages are printed. The output stream for these may be changed or the messages suppressed by altering the Common variable \( \text{LP} \), see §2.3.
System dependence: none.
Restrictions:
\( n > 0, \)
\( 1 \leq k \leq n - 1, \)
\( x_1 \leq x_2 \leq \ldots \leq x_n, \)
\( x_i < x_{i+k}, i=1,2,\ldots,n-k. \)

4 METHOD

To evaluate the B-splines \( M_{i,k}(x) \), \( 1 \leq i \leq n-k \), at the point \( x \) we need only compute \( k \) numbers. To do this we employ the recurrence relation of De Boor (1972) and Cox (1972),

\[
M_{i,k}(x) = \frac{(x-x_i)M_{i+1,k}(x) + (x_{i+k}-x)M_{i+1,k+1}(x)}{x_{i+k}-x_i}
\]

where

\[
M_{i,k}(x) = \begin{cases} (x_{i+1} - x_i)^{-1} & x_i \leq x < x_{i+k} \\ 0 & \text{otherwise} \end{cases}
\]

in the scheme proposed by Cox (1975).

In order to compute the corresponding integrals we use the formula

\[
\int x^n M_{i,k}(x) \, dx = \begin{cases} \sum_{r=0}^{k-1} (x - x_r) M_{i+r,k-r}(x) & x_i \leq x < x_{i+k} \\ x = x_{i+k} \end{cases}
\]

which is due to Gaffney (1974).

References

5 EXAMPLE OF USE

5.1 First example

Suppose we are given the ten data points

\[
\begin{align*}
    x_1 &= x_2 = x_3 = 0.0 \\
    x_4 &= x_5 = 1.0 \\
    x_6 &= 3.0 \\
    x_7 &= 4.0 \\
    x_8 &= x_9 = x_{10} = 6.0
\end{align*}
\]

and we require the elements in the matrix

\[
N = \begin{bmatrix}
    N_{3,1}(t_1) & N_{3,2}(t_1) & N_{3,3}(t_1) & N_{3,4}(t_1) \\
    N_{3,2}(t_2) & N_{3,2}(t_2) & N_{3,3}(t_2) & N_{3,4}(t_2) \\
    \vdots & \vdots & \vdots & \vdots \\
    N_{3,3}(t_7) & N_{3,2}(t_7) & N_{3,3}(t_7) & N_{3,2}(t_7)
\end{bmatrix}
\]

where \(N_j(x)\) is a normalized B-spline of degree 2 with knots at \(x, x_{j+1}, \ldots, x_{j+k}\), and

\[ t_i = \frac{i-1}{4}, \quad i = 1, 2, \ldots, 7 \]

The Fortran code to generate the matrix \(N\) might be as follows

```fortran
REAL X(10), NA(7,4), V(3), VINT(3)
DATA X/0.0, 0.0, 0.0, 1.0, 1.0, 3.0, 4.0, 6.0, 6.0, 6.0/
    \ldots
N=10        \hspace{1cm} \text{number of points} \ x_i \ i=1,2,\ldots,n
NORM=2       \hspace{1cm} \text{set to compute normalized B-splines, i.e.} \ N_j(x)
K=3          \hspace{1cm} \text{the order} \ k \ \text{of the B-spline}
DO 10 I=1,7
    DO 10 J=1,4
10    NA(I,J)=0.
    DO 30 I=1,7
        TI=FLOAT(I-1)/4.
    CALL TG04A(N,X,K,NORM,TI,JINT,V,VINT)
    DO 20 J=1,K
        J1=JINT-K+J
        NA(I,J1)=V(J)
20    CONTINUE
    \ldots
```

...
5.2 The second example

Suppose we are given the data points of the first example and we wish to compute the value of the B-spline \( M_{3,2}(x) \) and the corresponding integral

\[
\int_{x_i}^{x} M_{3,2}(\xi) \, d\xi
\]

at the values of \( x = (i-1)/100 \), \( i = 1,2,\ldots,101 \). The Fortran code for generating these values might be as follows

```fortran
REAL X(10), VALUE(101), VALINT(101), V(3), VINT(3)
DATA X/0.0,0.0,0.0,1.0,1.0,3.0,4.0,6.0,6.0,6.0/

NORM=1
K=3
N1=K+1
DO 20 I=1,101
  XVALUE=.01*FLOAT(I-1)
  CALL TG04A(N1,X(2),K,NORM,XVALUE,JINT,V,VINT)
  J=MAX0(1,N1-JINT)
  VALUE(I)=V(J)
  VALINT(I)=VINT(J)
20...
```

Figure 2. The Fortran code for the second example.

\( M_{3,2}(x) \) and \( \int_{x_i}^{x} M_{3,2}(\xi) \, d\xi \) are stored in \( V(1+K-JINT) \) and \( VINT(1+K-JINT) \), when \( 1+K-JINT \geq 1 \) and in \( V(1) \) and in \( VINT(1) \) otherwise.