1 SUMMARY

This subroutine **computes the spline, of specified degree and specified knots, which interpolates** n **arbitrarily prescribed function values.** Specifically, given function values $f_1,..,f_n$ at the n data points $x_1<x_2<...<x_n$, and given $n-k$ knots η_i, i=1,2,..,$n-k$, ($1\le k\le n$), in the open interval $x_1<x<x_n$, this subroutine computes the coefficients $a_1,..,a_n$ of the unique spline function

$$S(x)=\sum_{i=1}^{n}a_iN_{k,i}(x)$$

of degree $k-1$ which has the knots $\{\eta_i\}$ and which satisfies the interpolation conditions

$$\sum_{i=1}^{n}a_iN_{k,i}(x_j)=f_j,\quad j=1,2,..,n.$$

N.B. The function $N_{k,i}(x)$ is a normalized B-spline of degree $k-1$.

As a by-product of the calculation of the coefficients $a_1,..,a_n$ the subroutine also computes and returns the value of the integral

$$\int_{x_1}^{x_n} S(x)\,dx$$

ATTRIBUTES — **Version:** 1.0.0. **Types:** `TB06A, TB06AD.` **Calls:** `TG04.` **Original date:** December 1976. **Origin:** P.W.Gaffney*, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list

The single precision version

```
CALL TB06A(N,X,F,K,NK,ETA,IL,AN,ISW,WK,A)
```

The double precision version

```
CALL TB06AD(N,X,F,K,NK,ETA,IL,AN,ISW,WK,A)
```

`N` is an `INTEGER` variable which is set by the user to the number n of data points x_i, i=1,n. **Restriction:** $n>1$. This argument is not altered by the subroutine.

`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 data points x_i, i=1,n. **Restriction:** These must be ordered so that $x_1<x_2<...<x_n$. This argument is not altered by the subroutine.

`F` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n, which must be set by the user to the function values $f_1,..,f_n$. This argument is not altered by the subroutine.

`K` is an `INTEGER` variable which must be set by the user to the number k, where the degree of $S(x)$ is $k-1$. **Restriction:** The value of k must be in the range $1\le k\le n$. This argument is not altered by the subroutine.

`NK` is an `INTEGER` variable which must be set by the user to the length of the array `ETA` . This argument is not altered by the subroutine.

`ETA` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least the maximum value of ($n-k$ and 1), which must be set by the user to the knot values η_i, i=1,n–k when $n \neq k$. If n=k the contents of `ETA` are ignored by the subroutine. This argument is not altered by the subroutine. **Restrictions:** The following restrictions apply

(i) the knots must be ordered so that $\eta_1 \leq \eta_2 \leq ... \leq \eta_{n-k}$.

(ii) the Schoenberg-Whitney (S-W) conditions $x_i < \eta_i < x_{i+k}$, i=1,n–k, must hold.

`IL` is an `INTEGER` array of length at least n, which is used by the subroutine as workspace.

`AN` is a two-dimensional `REAL` (`DOUBLE PRECISION` in the D version) array whose first dimension is n and whose second dimension is k. It is used to store the coefficient matrix $\{N_{k,i}(x_j)\}_{1 \leq i,j \leq n}$, in compact form. On exit from the subroutine `AN` is overwritten by its decomposition, see §4 for further details.

`ISW` is an `INTEGER` variable which the user must set to the length of the array `WK`. This argument is not altered by the subroutine.

`WK` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least $2n$+$3k$+1, which is used as workspace, see §2.3. On exit from the subroutine the value of the integral

$$\int_{x_1}^{x_n} S(x)\, dx$$

is stored in `WK(N+K+1)` .

`A` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n which will be set by the subroutine to the values of the coefficients a_1,.,a_n.

2.2 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

```
COMMON/TB06B/LP,IFAIL
```

The double precision version

```
COMMON/TB06BD/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 $n < 2$,
2 $k < 1$ or $k > n$,
3 $x_i \geq x_{i+1}$ for any i,
4 $\eta_i > \eta_{i+1}$ for any i,
5 $\eta_i \leq x_i$ or $\eta_i \geq x_{i+k}$ for any i.

In the event of an error condition, when `IFAIL` is greater than zero, a diagnostic message is printed, the coefficients a_1,.,a_n are set to zero, and the subroutine returns to the calling program.

2.3 The contents of the workspace

On exit from the subroutine the workspace array `IL` and the first n+k locations of `WK` contain information which may be useful to the user. This information is described below.

The contents of `IL` are the integers which identify the knot intervals where the data points lie. That is `IL(I)`, `I=2,N-1` contain the unique integers, r_i say, such that

$$\eta_{r_i-k} \le x_i < \eta_{r_i+1-k}, i=2,n-1$$

where $\eta_0=x_1$ and $\eta_{n-k+1}=x_n$, and `IL(1)` contains the value k and `IL(N)` the value n.

The first $n+k$ locations of `WK` contain the complete knot set (see §4) which is required in order to write $S(x)$ as a linear combination of B-splines, see §1. They have the values

`WK(J)` $= x_1$, `J=1,K`

`WK(K+J)` $= \eta_j$, `J` $= j=1,n-k$,

and

`WK(N+J)` $= x_n$, `J=1,K`, `N` $= n$.

3 GENERAL INFORMATION

Use of common: the subroutine uses a Common area `TB06B/BD`, see §2.2.

Workspace: the user provides the workspace through the arguments `WK` and `IL`.

Other routines called directly: the library subroutine `TG04A/AD` is called to evaluate B-splines.

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 `LP`, see §2.

Restrictions:

$n \ge 2$,
$1 \le k \le n$,
$x_1 \le x_2 \le ... \le x_n$,
$\eta_1 \le \eta_2 \le ... \le \eta_{n-k}$,
$x_i < \eta_i < x_{i+k}$, $i=1,n-k$.

4 METHOD

We write the spline of interpolation $S(x)$ as a linear combination of n B-splines. In order to do this we introduce an additional $2k$ knots μ_i, $i=1,..,k,n+1,..,n+k$ such that

$$\mu_k = \mu_{k-1} = ... = \mu_1 = x_1, \tag{4.1}$$

and

$$\mu_{n+k} = \mu_{n+k-1} = ... = \mu_{n+1} = x_n, \tag{4.2}$$

and we define intermediate values of μ_i by the equation

$$\mu_{j+k} = \eta_j, \ j=1,n-k. \tag{4.3}$$

We then let $N_{k,i}(x)$ denote the normalized B-spline of degree $k-1$ with knots $\mu_i, \mu_{i+1}, .., \mu_{i+k}$, and write $S(x)$ as the sum

$$S(x) = \sum_{i=1}^{n} a_i N_{k,i}(x). \tag{4.4}$$

The constants $a_1,..,a_n$ are calculated by solving the linear equations

$$\sum_{i=1}^{n} a_i N_{k,i}(x_j) = f_j, \ j=1,n \tag{4.5}$$

Because the knots η_i, $i=1,n-k$ satisfy the conditions

$$x_i < \eta_i < x_{i+k},\ i=1,n-k, \tag{4.6}$$

the coefficient matrix

$$\mathbf{N} = \begin{bmatrix} N_{k,1}(x_1) & N_{k,2}(x_1) & .. & N_{k,n}(x_1) \\ N_{k,1}(x_2) & N_{k,2}(x_2) & .. & N_{k,n}(x_2) \\ : & : & : & : \\ N_{k,1}(x_n) & N_{k,2}(x_n) & .. & N_{k,n}(x_n) \end{bmatrix} \tag{4.7}$$

is nonsingular, and thus equation (4.5) has a unique solution.

Our method for solving the linear equations (4.5) takes advantage of the structure of the coefficient matrix **N**. Because the B-spline $N_{k,i}(x)$ is non-zero only over the interval $\mu_i < x < \mu_{i+k}$, it follows that there are at most k non-zero elements in each row of **N**, and that these occur in adjacent positions. Moreover the first non-zero in each row is either in the same column as, or is to the right of, the first non-zero in the previous row. Because the additional knots satisfy equations (4.1) and (4.2), the first and last rows of **N** always have only one non-zero, namely $N_{k,1}(x_1)$ and $N_{k,n}(x_n)$, which have the value 1. The matrix **N** is thus a band matrix.

We solve equations (4.5) by using Gaussian elimination. Because of the properties of B-splines it is unnecessary to perform any interchanges in the elimination process. Thus by storing only the non-zero elements of the matrix **N** we are able to take full advantage of the sparsity in **N**.