



1 SUMMARY

To **compute the knots of the optimal spline interpolation formula**. Specifically, the function $\tilde{S}(x)$ which interpolates values of a function of one variable $f(x)$ at n distinct points $x_1 < x_2 < \dots < x_n$, and which, whenever the k -th derivative of $f(x)$ is bounded and the value of the bound is unknown, provides the smallest possible value of $B(x)$ in the error bound

$$|f(x) - \tilde{S}(x)| \leq B(x) \max_{x_1 \leq x \leq x_n} |f^{(k)}(x)| \quad 1 \leq k \leq n.$$

$\tilde{S}(x)$ is a unique spline function of degree $k-1$ with $n-k$ knots $\eta_1, \dots, \eta_{n-k}$. The subroutine computes the knots $\{\eta_i\}$, given the data points x_1, \dots, x_n and given a value of k in the range $1 \leq k \leq n-1$.

The method is described in the reports Gaffney, AERE CSS.52 (1977) and Gaffney, AERE R.8781 (1977).

ATTRIBUTES — **Version:** 1.0.0. **Types:** TB08A, TB08AD. **Calls:** FD05. **Original date:** June 1977. **Origin:** P.W.Gaffney, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list

The single precision version

CALL TB08A(N, X, K, AJ, NK, WK, ISW, L, NMK, ETA)

The double precision version

CALL TB08AD(N, X, K, AJ, NK, WK, ISW, L, NMK, ETA)

- N** is an INTEGER variable which must be set by the user to n , the number of points x_i , $i=1, 2, \dots, n$. **N** is not altered by the subroutine. **Restriction:** $n \geq 2$.
- 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 of the points x_i , $i=1, 2, \dots, n$. **X** is not altered by the subroutine. **Restriction:** these must be ordered and distinct so that $x_1 < x_2 < \dots < x_n$.
- K** is an INTEGER variable which must be set by the user to the value k , where $k-1$ is the degree of the spline $\tilde{S}(x)$. **K** is not altered by the subroutine. **Restriction:** $1 \leq k \leq n-1$.
- AJ** is a two-dimensional REAL (DOUBLE PRECISION in the D version) array with first dimension $n-k$ and second dimension equal to the minimum of k and $n-k$. It is used by the subroutine as workspace.
- NK** is an INTEGER variable which must be set by the user to the second dimension of the array **AJ**. **NK** is not altered by the subroutine.
- WK** is a REAL (DOUBLE PRECISION in the D version) array of length at least $4n-2k$, which is used by the subroutine as workspace.
- ISW** is an INTEGER variable which must be set by the user to the length of the array **WK**. **ISW** is not altered by the subroutine.
- L** is an INTEGER array of length at least $n-k$. On return from the subroutine the i -th element of **L** is the unique integer λ_i such that

$$x_{\lambda_i} \leq \eta_i < x_{\lambda_i+1}, \quad i=1, 2, \dots, n-k.$$

NMK is an INTEGER variable which the user must set to $n-k$, the number of knots η_i , $i=1, 2, \dots, n-k$. **NMK** is not altered

by the subroutine.

ETA is a REAL (DOUBLE PRECISION in the D version) array of length at least $n-k$, in which the subroutine returns the values of η_i , $i=1, 2, \dots, n-k$.

2.2 Checks on the arguments

The restrictions imposed on the arguments are checked by the subroutine. If the restrictions are not satisfied an error diagnostic is printed, an error flag, IFAIL, is set, see §2.3, and a return is made to the calling program.

2.3 The common area and error messages

The subroutine uses a common area which the user may also reference.

The single precision version

```
COMMON/TB08B/ IMAX, LP, IFAIL
```

The double precision version

```
COMMON/TB08BD/ IMAX, LP, IFAIL
```

IMAX is an INTEGER variable which is initially defaulted to the value 1,000 and is the limit imposed on the number of Newton iterations, see §4.

LP is an INTEGER variable which is initially defaulted to 6 (line-printer), and specifies the Fortran stream number to be used for error messages. The printing can be suppressed by setting LP to zero.

IFAIL is an INTEGER variable which is always set by the subroutine to an error code. The possible values are

- 0 successful entry.
- 1 $NMK \neq n-k$.
- 2 $k < 1$ or $k > n-1$.
- 3 $x_i \geq x_{i+1}$ for some i .
- 4 IMAX is too small.
- 5 The continuation method failed to start. This is usually because k is too large or the problem is ill-conditioned.

3 GENERAL INFORMATION

Use of common: Uses a common area TB08B/BD, see §2.3.

Workspace: The user provides the workspace in the arrays AJ and WK.

Other routines called directly: FD05, and it uses private subroutines called TB08C/CD and TB08D/DD.

Input/output: Diagnostic messages are printed on Fortran stream 6, see LP in §2.3.

Restrictions: $n \geq 2$, $x_1 < x_2 < \dots < x_n$, $1 \leq k \leq n-1$.

4 METHOD

The knots η_i , $i=1, 2, \dots, n-k$ are calculated by solving the equations

$$\sum_{j=0}^{n-k} (-1)^j \int_{\hat{\eta}_j}^{\hat{\eta}_{j+1}} M_{k,i}(x) dx = 0, \quad i=1, 2, \dots, n-k \quad (4.1)$$

where $\hat{\eta}_0 = x_1$, $\hat{\eta}_{n-k-1} = x_n$, and $M_{k,i}(x)$ is a B-spline of order k with knots x_i, \dots, x_{i+k} .

The integral in the middle of equation (4.1) is calculated by using Gaffney's (1976a) formula together with Cox's (1975) algorithm for evaluating B-splines.

When $k=1$ equation (4.1) is linear and in this case the knots η_i , $i=1, 2, \dots, n-1$ have the values

$$\eta_i = \frac{1}{2}(x_i + x_{i+1}), \quad i=1, 2, \dots, n-1. \quad (4.2)$$

When $k \geq 2$ equation (4.1) is nonlinear. In this case we use a continuation method with a Newton iteration to solve (4.1). Further details are given by Gaffney (1977a) and (1977b).

References

- Cox.M.G. 1975. "An Algorithm for Spline Interpolation". J. Instit. Maths. Applics. **15**, pp. 95-108.
- Gaffney.P.W. 1976a. "The Calculation of Indefinite Integrals of B-splines". J. Instit. Maths. Applics. **17**, pp. 37-41.
- Gaffney.P.W. 1977a. "To Compute the Optimal Interpolation Formula". Harwell report CSS.52.
- Gaffney.P.W. 1977b. "Fortran Subroutines for Computing the Optimal Interpolation Formula". Harwell report R.8781.