1 SUMMARY

To compute at a given point x_p **function values and the 1st, 2nd and 3rd derivative values** of a **cubic spline** $S(x)$ defined by knots ξ_j, function values $S_j = S(\xi_j)$ and first derivative values $g_j = dS(\xi_j)/dx$, j=1,2,...,n, $n \geq 2$.

The spline and its derivatives are defined to be zero outside the range of the knot points. An option is provided for reducing the search time for the knot interval containing x_p making tabulation of the spline and its derivatives economical. Also the knot interval which was found to contain x_p is returned in Common.

The subroutine is not restricted to splines and may be used with any piece-wise cubic with continuous first derivative defined by its values and derivative values at the joins. Symmetric formulae are used for the spline and its derivatives.

ATTRIBUTES — **Version:** 1.0.0. **Types:** `TG02A; TG02AD`. **Calls:** `FD05`. **Original date:** March 1974. **Origin:** M.J.Hopper, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

The single precision version:

```
CALL TG02A(I,N,XI,S,G,XP,VALS)
```

The double precision version:

```
CALL TG02AD(I,N,XI,S,G,XP,VALS)
```

`I` is an `INTEGER` variable which must be set by the user to specify the method which the subroutine should use when initially estimating which of the knot intervals contains x_p. There are two possibilities.

- `I<0` no previous knowledge is assumed and an initial choice is made based on the assumption that the knots ξ_j j=1,2,...,n are equally spaced. If this does not give the required interval a sequential search is made starting from the chosen interval and continued until the interval containing x_p is located.
- `I≥0` here the subroutine assumes that the value of x_p is very near to the value used on the previous call and it starts a sequential search from that point. It also assumes that it is still dealing with the same spline and if the current x_p value is found to lie within the same knot interval as the previous value the coefficients of the cubic are not recalculated. This means that this option must only be used after at least one call using the first option for every new spline presented to the subroutine.

The user is advised to use `I<0` except when tabulating when a use of `I≥0` on subsequent entries would be more efficient. This argument is not altered by the subroutine.

`N` is an `INTEGER` variable which must be set by the user to n the number of knots. This argument is not altered by the subroutine.

`XI` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n which must be set by the user to the knot values ξ_j j=1,2,...,n. These must be ordered and distinct so that $\xi_1 < \xi_2 < ... < \xi_n$. This argument is not altered by the subroutine.

`S` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n which must be set by the user to the spline values at the knots, $S(\xi_j)$ j=1,2,...,n. This argument is not altered by the subroutine.

`G` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n which must be set by the user to the first derivative values, g_j j=1,2,...,n at the knots.

`XP` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which must be set by the user to the point x_p at which the spline and its derivatives are to be evaluated. If $x_p < \xi_1$ or $x_p > \xi_n$ the spline and its derivatives are defined to be zero. This argument is not altered by the subroutine.

`VALS` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least 4 which will be set by the subroutine to the values of the spline and its derivatives, i.e.

`VALS(1)` = $S(x_p)$

`VALS(2)` = $dS(x_p)/dx$

`VALS(3)` = $d^2S(x_p)/dx^2$

`VALS(4)` = $d^3S(x_p)/dx^3$

2.2 The Common area

The user may reference a Common area called `TG02B/BD` in which the subroutine identifies the knot interval which contains x_p.

The single precision version:

```
COMMON/TG02B/ K
```

The double precision version:

```
COMMON/TG02BD/ K
```

`K` is an `INTEGER` variable which is set by the subroutine to the knot interval k containing x_p, i.e. $\xi_k \le x_p \le \xi_{k+1}$. If $x_p < \xi_1$, `K` is set to zero and if $x_p > \xi_n$, `K` is set to n.

3 GENERAL INFORMATION

Use of Common: reference to Common area `TG02B/BD`, see §2.2.

Workspace: none.

Other subroutines: `FD05`.

Input/Output: none.

Restrictions:

$n \ge 2$,

$\xi_1 < \xi_2 < \ldots < \xi_n$.

4 METHOD

We are given the spline in terms of knots ξ_j j=1,2,...,n, spline values $S_j = S(\xi_j)$ j=1,2,...,n and derivatives $g_j = S'(\xi_j)$ j=1,2,...,n. If we let $h = \xi_{k+1} - \xi_k$ the cubic for a given x in (ξ_k, ξ_{k+1}) can be written in the symmetric form

$$S(x) = \phi(S_k - \theta\phi\alpha) + \theta(S_{k+1} + \theta\phi\beta)$$

where $\theta = (x - \xi_k)/h$, $\phi = 1 - \theta$, $\alpha = S_{k+1} - S_k - hg_k$ and $\beta = S_{k+1} - S_k - hg_{k+1}$.

The derivatives are similarly given by

$S'(x) = \phi(g_k + 3\alpha\theta/h) + \theta(g_{k+1} + 3\beta\phi/h)$

$S''(x) = \phi(4\alpha+2\beta)/h^2 - \theta(4\beta+2\alpha)/h^2$

$S'''(x) = -6(\alpha+\beta)/h^3$.

These symmetric forms have been chosen so as to obtain a finer accuracy when x lies towards either of the knot points. However, it is still possible for cancellation to occur and in the few occasions this is likely to happen it cannot be avoided, no matter what formula is used, but when it does it will tend to affect the second and third derivative values most.

5 EXAMPLE OF USE

Calling `TG02A/AD` should present no problems but we show here a simple example where an interpolating spline obtained through `TB04A` is to be tabulated.

Suppose we have some function values f_i i=1,2,...,n at points x_i i=1,2,...,n and have chosen, we hope on sound numerical grounds, to obtain the cubic spline that interpolates the values with a view to tabulating the function and its derivative. Then code such as shown in figure 1 might be used.

It should be appreciated that the example was chosen mainly for its simplicity and to illustrate the calling sequence of `TG02A/AD.` Numerically, the degree to which the spline is a good approximation to the function will depend both upon the accuracy in the original values and the spacing. Even if the fit is a good one the spline derivatives will not necessarily be good approximations to the function derivatives and it is likely that a smoothing process would produce better results. Expert advice should be sought if there is any doubt as to which method to use.

```
spline value and derivatives
      REAL  VALS(4)
knots, values and derivatives at knots
      REAL  X(100),F(100),G(100)
workspace for TB04A
      REAL  W(300)
no. of values and no. of print points.
   10 READ(5,1,END=99) N,M
    1 FORMAT(2I5)
function values
      READ(5,2,END=99) (X(I),F(I),I=1,N)
    2 FORMAT(8F10.0)
obtain interpolation spline
      CALL TB04A(N,X,F,G,W)
tabular interval
      DX=(X(N)-X(1))/(M-1)
      DO 3 I=1,M
tabular point
      XP=X(1)+(I-1)*DX
evaluate spline and derivatives
      CALL TG02A(I-1,N,X,F,G,XP,VALS)
print them
    3 WRITE(6,4) XP,(VALS(J)),J=1,4)
    4 FORMAT(2X,5E16.5)
go and do another set
      GO TO 10
   99 STOP
      END
```

Figure 1. Tabulating a spline