1 SUMMARY

Given a **cubic spline** $S(x)$, an estimate x_0 and a spline value y, the subroutine **finds the point ξ nearest to x_0 such that $s(\xi)=y$.**

The method is to locate the knot interval containing x_0, then examine successive nearest knot intervals for the presence of a real root of $S(x)-y$.

ATTRIBUTES — **Version:** 1.0.0. **Types:** `TC01A; TC01AD.` **Calls:** `FD05`. **Original date:** April 1974. **Origin:** S.Marlow, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

The single precision version

```
CALL TC01A(XN,FN,GN,N,X0,Y)
```

The double precision version

```
CALL TC01AD(XN,FN,GN,N,X0,Y)
```

`XN` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n which must be set by the user to contain the knots x_i, i=1, 2,..., n. **Restriction:** the knots must be ordered and distinct, i.e. $x_1 < x_2 < ... < x_n$.

`FN` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n which must be set by the user to contain the values of the spline at the knots, so that $S(x_i)$, i=1, 2,..., n are found in `FN(I)`, `I=1,N`.

`GN` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n which must be set by the user to contain the values of the first derivative of $S(x)$ at the knots, i.e. $S'(x_i)$, i=1, 2,..., n must be in `GN(I), I=1,N`.

`N` is an `INTEGER` variable which must be set by the user to the value of n, the number of knots. **Restriction:** $n \geq 2$.

`X0` is a `REAL` variable (`DOUBLE PRECISION` in the D version) which must be set by the user to an estimate x_0 of ξ a solution of the equation $S(x)$=y. On return the subroutine will overwrite `X0` with the computed value of ξ. The common variable `IN` is used to indicate whether the root was found and identifies the knot interval which contains it when it is found, see §2.2.

`Y` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which must be set by the user to y, the spline value for which the root ξ is required.

2.2 The common area

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

The single precision version

```
COMMON/TC01B/ LP,IN
```

The double precision version

```
COMMON/TC01BD/ LP,IN
```

`LP` is an `INTEGER` variable (defaulted to 6) and specifies the Fortran stream number used for printing diagnostic messages. The printing can be suppressed by setting `LP` to zero.

`IN` is an `INTEGER` variable which on return from the subroutine will be set to the knot interval in which the root

was found, i.e. the root lies in the interval `XN(IN)` to `XN(IN+1)`. If `IN` is returned set to zero then no root was found and if set to –1 the value of the argument `N` was found to be non-positive.

3 GENERAL INFORMATION

Use of common: Uses a common area called `TC01B/BD`, see §2.2.

Workspace: None.

Other routines called directly: `FD05`.

Input/output: Prints diagnostic messages when errors occur, see §2.2.

Restrictions: $n \geq 2$, $x_1 < x_2 < ... < x_n$.

4 METHOD

On entry to the subroutine a binary search is made to find the knot interval which contains x_0 the estimate of the root ξ.

Simple tests are applied to the interval to detect a possible root. If these tests fail then the adjacent intervals are tested. The interval chosen depends on the users estimate x_0. When the test is successful a symmetric form of the cubic spline is formed and a root in the interval is calculated by using the Newton-Raphson method with a bracket. The other two roots of the cubic are found by solving the quadratic obtained by the deflation of the cubic. Roots outside the interval being rejected. Tests are made to see whether it is possible that adjacent intervals could possibly produce a root nearer to the user's estimate of the root than the one just calculated. If this is not possible then we have finished. Otherwise the root in one of the adjacent intervals is found. The interval chosen depending on the root already found and the user's estimated.