



## 1 SUMMARY

To **evaluate** the **divided differences** of a function  $f(x)$  tabulated at points  $x_1 < x_2 < \dots < x_n$  not necessarily equally spaced.

The values of the function and the points  $x_j$  must be passed to the subroutine in an array. The user specifies the maximum order of the differences and may direct the subroutine to return the difference table in various compact forms. The table is not printed out by the subroutine.

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** TA02A; TA02AD. **Original date:** February 1965. **Origin:** D. Russell, Atlas Lab., Chilton, Oxon.

## 2 HOW TO USE THE PACKAGE

### 2.1 Notation

Explanation is facilitated by means of the following table of divided differences of a function  $y(x)$ :

$x_0$	$y_0$	$\partial y_{\frac{1}{2}}$	$\partial^2 y_1$	$\partial^3 y_{\frac{3}{2}}$	$\partial^4 y_2$	-
$x_1$	$y_1$	$\partial y_{\frac{3}{2}}$	$\partial^2 y_2$	$\partial^3 y_{\frac{5}{2}}$	-	-
$x_2$	$y_2$	$\partial y_{\frac{5}{2}}$	$\partial^2 y_3$	-	-	-
$x_3$	$y_3$	$\partial y_{\frac{7}{2}}$	-	-	-	-
$x_4$	$y_4$	-	-	-	$\partial^4 y_{n-2}$	-
-	-	-	-	$\partial^3 y_{n-1}$	-	-
-	-	$\partial y_{n-1}$	$\partial^2 y_{n-1}$	-	-	-
$x_n$	$y_n$	-	-	-	-	-

where the odd terms are defined by

$$\partial^{2j+1} y_{i+\frac{1}{2}} = \frac{\partial^{2j} y_{i+1} - \partial^{2j} y_i}{x_{i+j+1} - x_{i-j}}$$

and the even terms are defined by

$$\partial^{2j} y_i = \frac{\partial^{2j-1} y_{i+\frac{1}{2}} - \partial^{2j-1} y_{i-\frac{1}{2}}}{x_{i+j} - x_{i-j}}$$

### 2.2 Argument list and calling sequence

*The single precision version*

```
CALL TA02A(X, Y, IY, NL, NS, NU, NDIFF, IOPT)
```

*The double precision version*

```
CALL TA02AD(X, Y, IY, NL, NS, NU, NDIFF, IOPT)
```

X is a REAL (DOUBLE PRECISION in the D version) array which must be set by the user to contain the values of  $x$ . The range of values of  $x$  for which the difference table is to be constructed need not begin at the first element of the array X but is selected using the arguments NL, NS and NU. To be precise,  $x_0, x_1, x_2, \dots, x_n$  should be stored in X(NL), X(NL+NS), X(NL+2\*NS), up to X(NU).

Y is a two-dimensional REAL (DOUBLE PRECISION in the D version) array of first dimension IY in which the user

must put the tabulated values of the function  $y$  and on return will also contain the table of divided differences. The values of  $y_0, y_1, y_2, \dots, y_n$  must be stored in the first column of  $Y$  in positions corresponding to those of the  $x$  values in the array  $X$ , i.e. in  $Y(NL, 1), Y(NL+NS, 1), Y(NL+2*NS, 1),$  up to  $Y(NU, 1)$ . There are several possible storage arrangements for the differences depending on the choice of arguments  $IOPT$  and  $NS$ ; these are described more fully in §2.2.

- $NY$  is an INTEGER variable which must be set by the user to the first dimension of the two-dimensional array  $Y$ .
- $NL$  is an INTEGER variable which must be set by the user to specify where the beginning of the table is in the arrays  $X$  and  $Y$ , i.e.  $x_0$  is stored in  $X(NL)$  and  $y_0$  in  $Y(NL, 1)$ .
- $NS$  is an INTEGER variable which must be set by the user to specify the spacing of the members of the table in the arrays  $X$  and  $Y$ , i.e.  $x_1$  is stored in  $X(NL+NS)$ ,  $x_2$  in  $X(NL+2*NS)$ , and so on. If both odd and even differences are required  $NS$  must be at least two but if only one or the other is required storage can be saved by reducing  $NS$  to one.
- $NU$  is an INTEGER variable which must be set by the user to specify where the end of the table comes in the arrays  $X$  and  $Y$ , i.e.  $x_n$  is stored in  $X(NU)$  and  $y_n$  in  $Y(NU, 1)$ . **Restriction:**  $NU \geq NL$ .
- $NDIFF$  is an INTEGER variable which must be set by the user to the order of the highest order differences required in the table. **Restriction:**  $NDIFF \geq 0$ .
- $IOPT$  is an INTEGER variable which must be set by the user to select one of three storage arrangements for the differences returned in the array  $Y$ . The storage arrangements and the choices for  $IOPT$  and  $NS$  are described in §2.2.

**2.2 The storage arrangements for the differences**

The user is given the option of having the differences returned in the array  $Y$  in a variety of arrangements including: a simple arrangement, more compacted arrangements and arrangements where either the odd or the even differences are omitted to save on storage. The choice of arrangement is controlled by the argument  $IOPT$  and the spacing argument  $NS$ .

The simplest arrangement,  $IOPT=0$  and  $NS=2$ , is illustrated by the following diagram.

$y_{i-1}$		$\partial^2 y_{i-1}$	
	$\partial y_{i-\frac{1}{2}}$		$\partial^3 y_{i-\frac{1}{2}}$
$y_i$		$\partial^2 y_i$	
	$\partial y_{i+\frac{1}{2}}$		$\partial^3 y_{i+\frac{1}{2}}$
$y_{i+1}$		$\partial^2 y_{i+1}$	

A more compact scheme,  $IOPT=1$  and  $NS=2$ , which closes up the differences and hence requires less storage is the following

$y_{i-1}$	$\partial^2 y_{i-1}$	$\partial^4 y_{i-1}$	$\partial^6 y_{i-1}$
	$\partial y_{i-\frac{1}{2}}$	$\partial^3 y_{i-\frac{1}{2}}$	$\partial^5 y_{i-\frac{1}{2}}$
$y_i$	$\partial^2 y_i$	$\partial^4 y_i$	$\partial^6 y_i$
	$\partial y_{i+\frac{1}{2}}$	$\partial^3 y_{i+\frac{1}{2}}$	$\partial^5 y_{i+\frac{1}{2}}$
$y_{i+1}$	$\partial^2 y_{i+1}$	$\partial^4 y_{i+1}$	$\partial^6 y_{i+1}$

The third arrangement, IOPT=2 and NS=2, also compacts the differences but differs slightly from the previous arrangement.

$y_{i-1}$		$\partial^2 y_{i-1}$	$\partial^4 y_{i-1}$
	$\partial y_{i-\frac{1}{2}}$	$\partial^3 y_{i-\frac{1}{2}}$	$\partial^5 y_{i-\frac{1}{2}}$
$y_i$		$\partial^2 y_i$	$\partial^4 y_i$
	$\partial y_{i+\frac{1}{2}}$	$\partial^3 y_{i+\frac{1}{2}}$	$\partial^5 y_{i+\frac{1}{2}}$
$y_{i+1}$		$\partial^2 y_{i+1}$	$\partial^4 y_{i+1}$

The three schemes illustrated are all for the case when NS=2. More storage may be saved by setting NS=1 and closing up the rows, and in two of the cases this means omitting either all odd order differences or all even differences (except order zero). Thus setting NS=1 the three values of IOPT will give

- 0 the i-th row will contain  $y_i, \partial y_{i+\frac{1}{2}}, \partial^2 y_i, \partial^3 y_{i+\frac{1}{2}},$  etc.
- 1 even differences will overwrite odd differences so that the i-th row will contain  $y_i, \partial^2 y_i, \partial^4 y_i,$  etc.
- 2 the odd differences will overwrite the even differences so that the i-th row will contain  $y_i, \partial y_{i+\frac{1}{2}}, \partial^3 y_{i+\frac{1}{2}},$  etc.

The second dimension of the array Y, MY say, must be large enough to accommodate the table of differences and the minimum possible value for MY varies with the setting of IOPT as follows

- 0  $MY \geq \text{NDIFF} + 1.$
- 1  $MY \geq (\text{NDIFF} + 3) / 2.$
- 2  $MY \geq (\text{NDIFF} + 4) / 2$

Note that NS must be greater than or equal to one. If NS>2 there will be empty rows and the row indexes of the rows shown in the examples would be I-NS, I-NS/2, I, I+NS-NS/2, I+NS.

### 3 GENERAL INFORMATION

**Use of common:** none.

**Workspace:** none.

**Other routines called directly:** none.

**Input/output:** none.

**Restrictions:**  $NS \geq 1, NU \geq NL, \text{NDIFF} \geq 0, \text{IOPT}$  must be 0, 1 or 2.