



## 1 SUMMARY

To solve the **two point boundary value** problem for the **linear parabolic partial** differential equation

$$\frac{\partial u}{\partial t} = a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial u}{\partial x} + cu + d$$

where  $a$ ,  $b$ ,  $c$  and  $d$  are functions of  $x$  and  $t$ ,  $x_1 \leq x \leq x_n$ , given boundary conditions of the form

$$p \frac{\partial u}{\partial t} = q + ru + s \frac{\partial u}{\partial x}$$

at the points  $x_1$  and  $x_n$ . Given  $\delta t$  and the solution  $u(x, t)$  at  $t$ , on  $n$  equally spaced points  $x_i$ ,  $i=1, 2, \dots, n$  the subroutine advances the integration one time step to obtain  $u(x, t+\delta t)$ .

The Crank-Nicholson integration formula is used to transform to a 2nd order linear ordinary differential equation which is solved by calling DD11.

The user must supply a subroutine to compute values of the functions  $a$ ,  $b$ ,  $c$  and  $d$  given current values of  $x$  and  $t$ .

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** DP11A; DP11AD. **Calls:** DD11, TA03, TD01 and FUNCTS (a user subroutine). **Original date:** September 1965. **Origin:** P.Hallowell. Atlas Lab., Chilton, Oxon.

## 2 HOW TO USE THE PACKAGE

### 2.1 The argument lists

*The single precision version*

```
CALL DP11A(U, X1, XN, N, T0, DT, UX1, UXN, DU, DT, E, LIM, UA, NUM, W)
```

*The double precision version*

```
CALL DP11AD(U, X1, XN, N, T0, DT, UX1, UXN, DU, DT, E, LIM, UA, NUM, W)
```

**U** is a REAL (DOUBLE PRECISION in the D version) array of length at least  $n$ . The user must set the first  $n$  elements of **U** to the solution of the equation at  $t=t_0$ . That is,  $U(i) = u(x_i, t_0)$ ,  $i=1, 2, \dots, n$ . On return, these elements will be overwritten with the solution at  $t_0 + \delta t$ .

**X1** and **XN** are REAL (DOUBLE PRECISION in the D version) variables which must be set by the user to the end points of the range of  $x$ , i.e.  $x_1 \leq x \leq x_n$ .

**N** is an INTEGER variable which must be set by the user to  $n$  the number of  $x$  points covering the range  $x_1 \leq x \leq x_n$ . The  $n$  points will include the end points and be equally spaced.

**T0** is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to the value of  $t_0$  corresponding to the solution provided in **U**. On return **T0** will be set to  $t_0 + \delta t$  ( $\delta t$  passed in **DT**).

**DT** is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to the time step  $\delta t$  which the integration is to be advanced.

**UX1** is a REAL (DOUBLE PRECISION in the D version) array of length 4 which must be set by the user to the boundary conditions at  $x=x_1$ . Given boundary conditions of the form

$$p \frac{\partial u}{\partial t} = q + ru + s \frac{\partial u}{\partial x}$$

the first four elements of **UX1** must be set to:

$$UX1(1) = p, \quad UX1(2) = q, \quad UX1(3) = r, \quad UX1(4) = s.$$

UXN is a REAL (DOUBLE PRECISION in the D version) array of length 4 which must be set by the user to the boundary conditions at  $x=x_n$  (treat as argument UX1).

DUDT is a REAL (DOUBLE PRECISION in the D version) array of length at least  $n$  which is used to pass values of  $\partial u/\partial t$  at  $t_0$  to the subroutine when they are known (see argument NUM for further details). On return DUDT will contain the values of  $\partial u/\partial t$  calculated at  $t_0+\delta t$ .

E is a REAL (DOUBLE PRECISION in the D version) array of length at least  $n$  which is set by the subroutine to the differences between the first approximation to  $u(x_i, t+\delta t)$ ,  $i=1, 2, \dots, n$ , and the final values returned in U. That is, E contains the difference between the solution ignoring 3rd order and higher order differences and the final solution obtained by taking these into account, see DD11 for more details.

LIM is an INTEGER variable which must be set by the user to control the calculation.

If LIM=0 3rd and higher differences are not taken into account; that is the subroutine returns after solving the finite difference equations based on the 1st and 2nd differences. In this case both E and NUM are set equal to zero.

If LIM>0 then LIM is taken to be the limit on the number of times that the correction terms based on the 3rd and 4th differences are calculated. When the required accuracy is reached NUM is set equal to the number of times the correction terms have been calculated. If the correction terms are calculated LIM times without reaching the required accuracy the subroutine returns with NUM set to -1 and DUDT set to  $\partial u/\partial t$  at  $t_0$ .

UA is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to the absolute accuracy required. It may be altered by the routine.

NUM is an INTEGER variable which is set both by the user and the subroutine.

The user must set NUM to a value greater than zero if values of  $\partial u/\partial t$  at  $t_0$  are being supplied in the first  $n$  elements of DUDT. If these values are not known NUM should be set to zero and the subroutine will calculate its own approximations.

On return the subroutine will set NUM to zero or to the number of times correction terms have been calculated depending on the setting of LIM (see argument LIM).

W is a REAL (DOUBLE PRECISION in the D version) array of length at least  $15n+6$  words which is used by the routine for workspace.

## 2.2 Calculating the equation coefficients

The user must provide a subroutine to calculate the equation coefficients  $a$ ,  $b$ ,  $c$  and  $d$  at given values of  $x$  and  $t$ , see section 1. The subroutine must take the form

*The single precision version*

```
SUBROUTINE FUNCTS(A,B,C,D,T,X,IU,IDUDX,W)
REAL A,B,C,D,T,X
REAL W(*)
INTEGER IU,IDUDX
```

*The double precision version*

```
SUBROUTINE FUNCTS(A,B,C,D,T,X,IU,IDUDX,W)
DOUBLE PRECISION A,B,C,D,T,X
DOUBLE PRECISION W(*)
INTEGER IU,IDUDX
```

The subroutine must set  $A=a(x,t)$ ,  $B=b(x,t)$ ,  $C=c(x,t)$  and  $D=d(x,t)$ , given the values of  $x$  and  $t$  in X and T. The variables IU and IDUDX should not be used (they are present only for use by DP12 which calls DP11). The array W is

the work array passed as the last argument to DP11 and the first  $15n+6$  words should not be altered. The subroutine FUNCTS will be called  $n$  times on each call of DP11. Note: the name must be FUNCTS and this must be used for both single and double precision versions.

### 3 GENERAL INFORMATION

**Use of common:** None.

**Workspace:**  $15n+6$  words provided in argument W.

**Other routines called directly:** DD11A/AD, TA03A/AD and TD01A/AD are called. The user must provide a subroutine called FUNCTS to calculate the equation coefficients.

**Input/output:** None.

### 4 METHOD

The term  $\partial u/\partial t$  is replaced using the Crank-Nicolson formula:

$$\frac{1}{2} \left( \frac{\partial u}{\partial t} \Big|_{t_0} + \frac{\partial u}{\partial t} \Big|_{t_0 + \delta t} \right) = \frac{u(t_0 + \delta t) - u(t_0)}{\delta t}$$

The equation is then rewritten as an ordinary second order linear differential equation and solved by calling DD11A/AD (see DD11 for the method).