

```

                                TDMA.f
C      TTTTTT   RRr*   iii   DDD.   III   A   ,GGGG
C      T       R  R    I    D  D    I    A  A   '
C      T       Rrr*,  I    D  D    I    AaaaA   G ``g
C      T       R  R    iii   DDD'   III   A    A    `GGGG'
C-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
C
PROGRAM TDMA
IMPLICIT NONE

C-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
C.. Coefficients are defined when system of equations are written as:
C.. In geographical notation,
C..  $AP(i)*x(i)=AE(i)*x(i+1)+AW(i)*x(i-1)+d(i)$ 
C..  $i = 1$  to  $n$ ,  $AW(1)=0$ ,  $AE(n)=0$ 
C..
C.. In algebraic notation,  $a(i)*x(i)=b(i)*x(i+1)+c(i)*x(i-1)+d(i)$ 
C..  $i = 1$  to  $n$ ,  $c(1)=0$ ,  $b(n)=0$ 
C.. Here,  $a(i):: AP(i)$ ,  $b(i):: AE(i)$ ,  $c(i):: AW(i)$ 
C-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
C.. In matrix form, it will look like:
C | AP1 -AE1 0 0 0 ... 0 | |x1| |d1|
C | -AW2 AP2 -AE2 0 0 ... 0 | |x2| |d2|
C | 0 -AW3 AP3 -AE3 0 ... 0 | |x3| |d3|
C | 0 0 -AW4 AP4 -AE4 ... 0 | |x4| = |d4|
C | . . . . . 0 | |xi| |di|
C | . . . . . | |..| |..|
C | 0 0 0 0 . -AWn APn | |xn| |dn|
C
C.. If systems of equations are written (available or derived) as:
C..  $aw(i)*x(i-1) + ap(i)*x(i) + ae(i)*x(i+1) = d(i)$ 
C.. In matrix form, it will look like:
C.. |ap1 ae1 0 0 0 ... 0 | |x1| |d1|
C.. |aw2 ap2 ae2 0 0 ... 0 | |x2| |d2|
C.. | 0 aw3 ap3 ae3 0 ... 0 | |x3| |d3|
C.. | 0 0 aw4 ap4 ae4 ... 0 | |x4| = |d4|
C.. | . . . . . 0 | |xi| |di|
C.. | . . . . . | |..| |..|
C.. | 0 0 0 0 . awn apn | |xn| |dn|
C.. The sign of coefficients  $ap(i)$ ,  $aw(i)$ ,  $ae(i)$  needs to be adjusted.
C..  $ap(i) = AP(i)$ ,  $AW(i) = -aw(i)$ ,  $AE(i) = -ae(i)$ 
C-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
real phi,A,AP,AE,AW,d
integer i,j,NN,solver
dimension phi(100),AP(2000),AE(2000),AW(2000),d(2000)

C..
common A(2000,2001)
common NN

C..
open(33, FILE='TDMA.inp')
open(34, FILE='TDMA.out')

C..
read(33, 51) NN
51 format(I5,I5)
C..
C.. Check for size limits

```

```

if( NN .gt. 2000 .OR. NN .LE. 2) then
  write(*,*)'No. of rows, N: 2 =< N =< 2000. Please check input.'
  stop
endif
C-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
read(33,*) AP(1), AE(1), d(1)
AW(1) = 0.0
do i = 2,NN-1
  read(33,*) AW(i), AP(i), AE(i), d(i)
enddo
read(33,*) AW(NN), AP(NN), d(NN)
AE(NN) = 0.0
write(34,10) (i, AW(i),AP(i), AE(i), d(i), i = 1,NN)
10 format(I5, 5x, 4F10.5)
write(34,*) '-----Input Ends-----'
C..
call tridiag(phi,AP,AE,AW,d,NN)
C..
write(34,11) (i,phi(i), i = 1,NN)
11 format(I5,5x,F20.5)
C..
close(33)
close(34)
stop
end
C-----
subroutine tridiag(phi,AP,AE,AW,d,j)
implicit none
real phi,AP, AE, AW, d, P, Q
integer i, j
dimension phi(j), AP(j), AE(j), AW(j), d(j), P(j), Q(j)
C-----
do i = 1,j
  if(AP(i) .eq. 0.0) then
13   write(6,13) i
      format(/3x,'Diagonal element .eq. 0 in tridiag. at i = ',i2/)
      stop
  end if
end do
C
P(1) = AE(1)/AP(1)    ! b(1)/a(1)
Q(1) = d(1)/AP(1)    ! d(1)/a(1)
do i = 2, j          ! Forward Elimination
  P(i) = AW(i) / (AP(i) - AE(i-1)*P(i-1))
C..  P(i) :: b(i)/[a(i) - c(i)*P(i-1)]
C..
  Q(i) = (d(i)+ AW(i)*Q(i-1)) / (AP(i) - AW(i) * P(i-1))
C..  Q(i) :: [d(i) + c(i)*Q(i-1)] / [a(i) - c(i) * P(i-1)]
enddo
C..
phi(j) = Q(j)        ! Backward Substitution
do i = j-1,1,-1
  phi(i) = P(i)*phi(i+1) + Q(i)
enddo

```

return
end

C-----+-----+-----+-----+-----+-----+-----+-----+-----+-----

C Example-1:

C Original System of Equations:

C	2.04	-1	0	0		X1 =	40.8
C	-1	2.04	-1	0		X2 =	0.80
C	0	-1	2.04	-1		X3 =	0.80
C	0	0	-1	2.04		X4 =	200.8

C Solution:

C X1 = 65.970

C X2 = 93.778

C X3 = 124.54

C X4 = 159.48

C

C Input to TDMA Solver: TDMA.inp

C 4

C 2.04 1 40.8

C 1 2.04 1 0.80

C 1 2.04 1 0.80

C 1 2.04 200.8

C-----+-----+-----+-----+-----+-----+-----+-----+-----+-----

C Example-2:

C Original System of Equations:

C	-2	1	0	0		X1 =	0.04
C	1	-2	1	0		X2 =	0.04
C	0	1	-2	1		X3 =	0.04
C	0	0	1	-2		X4 =	0.04

C

C Solution:

C X1 = -0.08

C X2 = -0.12

C X3 = -0.12

C X4 = -0.08

C

C Input to TDMA Solver: TDMA.inp

C 4

C 2 1 -0.04

C 1 2 1 -0.04

C 1 2 1 -0.04

C 1 2 -0.04