```
C      ,GGGGG       A       U     U    .sss     .sss
C     G           A A      U     U    *S.      *S.
C     G  **g     AaaaA     U     U       S        S
C      `GGGG'    A     A     u u u     ssS*     ssS*
C -----------------------------------------------------------------------
      PROGRAM GAUSS_ELIMINATION
      INCLUDE 'Gauss.inc'
C..
      open(33,FILE='Gauss.inp')
      open(34,FILE='Gauss.out')
      read(33,51) NN
51    format(I5,I5)
      if( NN .gt. 5000) then
        write(6,*)'Array size should be < 5000. Please correct input.'
        stop
      endif
C..
      do i = 1,NN
        read(33,*) (A(i,j), j = 1,NN+1)
      enddo
C..
      write(*,*)
      call GAUSS
C..
      write(34,*) 'The upper traingular matrix is: '
      do i = 1,NN
        write(34,12) (A(i,j), j = 1,NN+1)
  12    format(5x,6F10.2)
      enddo
C..
      write(34,*) 'The value of unknowns are: '
      write(34,13) (i, phi(i), i = 1,NN)
  13  format(I5,5x,F20.3)
C..
      close(33)
      close(34)
      stop
      end
C -----------------------------------------------------------------------
      subroutine GAUSS
      INCLUDE 'Gauss_Elimination.inc'
C -----------------------------------------------------------------------
C     Forward Elimination
      do k = 1,NN-1
        if (k .LT. NN) then
          p = k+1
        else
          p = k
        endif
C..
        do i = p,NN
          xkk = A(i,k) / A(k,k)
          do j = k, NN+1
            A(i,j) = A(i,j) - A(k,j) * xkk
```

```
          enddo
C..
        enddo
C..
      enddo
C..
C..   Backward Substitution to Calculate Unknown
      phi(NN) = A(NN,NN+1) / A(NN,NN)
      do i =  NN-1, 1, -1
        sum = 0.0
        do j = i+1, NN
          sum = sum + A(i,j)*phi(j)
        enddo
        phi(i) = (A(i,NN+1) - sum)/A(i,i)
      enddo
      return
      end
C ------------------------------------------------------------------------
C..   Example-1 for Testing the Program
C..   x  +  y  +  z =  6
C..   2x -  y  +  z =  3
C..   x  +  0  +  z =  4
C..   Solution: x=1, Y=2, z=3
C..   The matrix is
C..   1     1     1    6
C..   2    -1     1    3
C..   1     0     1    4
C ------------------------------------------------------------------------
C..   Example-1 for Testing the Program
C..
C..  -2   1   0   0   0    -800.0
C..   1  -4   1   0   0    -1600.0
C..   0   1  -4   2   0    -50.0
C..   0   0   1  -4   1    -850.0
C..   0   0   0   2  -4    -850.0
C..   Solution is: 741.56, 683.12, 390.91, 415.26, 420.13
```

```
IMPLICIT NONE
integer i,j,k,p,NN,N, N1
parameter (N = 5000)
parameter (N1 = 5001)
real A(N,N1), xkk, sum, phi(N)
common A, phi, NN
```

```
Public Sub Gauss(n, ByRef A() As Double, ByRef phi() As Double)
  Dim i, j, k, n_1, p As Integer
  Dim X As Double       'By default, an array begins with zero
  'The extra column is for the Load Vector.
Rem
  'Convert the matrix to a upper triangular matrix
  n_1 = n - 1
  For k = 0 To n_1
    p = k + 1
    For i = p To n
      X = A(i, k) / A(k, k)
      For j = k To n + 1
        A(i, j) = A(i, j) - X * A(k, j)
      Next j
    Next i
  Next k
Rem
  'Back-substitution
  phi(n) = A(n, n + 1) / A(n, n)
  For i = n_1 To 0 Step -1
    Sum = 0#
    For j = i + 1 To n
      Sum = Sum + A(i, j) * phi(j)
    Next j
    phi(i) = (A(i, n + 1) - Sum) / A(i, i)
  Next i
End Sub
'--------------------------------------------------------------------------------------
----
Sub checkGauss()
  Dim m, n As Integer
  Dim A() As Double
  Dim phi() As Double
Rem
   ' myArray = Range("A1:F10").Value - Read value from worksheets
Rem  or
   ' Specify matrix value directly
   inMtx = Array(Array(-2.467, 0.5, 1#, 0, -196.7), Array(0.5, -2.267, 0, 1#,
-126.7), _
                Array(1#, 0#, -4#, 1#, -700#), Array(0#, 1#, 1#, -4#, -600#))
   ReDim A(0 To UBound(inMtx), 0 To UBound(inMtx(1)))
   'UBound(inMtx)    = number of inner arrays
   'UBound(inMtx(1)) = number of elements in inner array
Rem
   n = UBound(A, 1)  'Number of row vectors
   m = UBound(A, 2)  'number of column vectors
   ReDim phi(n)
   ReDim A(n, m)
Rem
   For i = 0 To n
     For j = 0 To m
       A(i, j) = inMtx(i)(j)
     Next j
   Next i
```

```
Rem
   'Call subroutine
   Gauss n, A(), phi()
   For i = 0 To n
     For j = 0 To n
       Worksheets("SOW").Cells(16 + i, j + 3) = A(i, j)
       If Abs(A(i, j)) < 0.001 Then
         Worksheets("SOW").Cells(16 + i, j + 3).NumberFormat = "0"
       Else
         Worksheets("SOW").Cells(16 + i, j + 3).NumberFormat = "0.00"
       End If
     Next j
     Worksheets("SOW").Cells(16 + i, m + 3) = phi(i)
     Worksheets("SOW").Cells(16 + i, m + 3).NumberFormat = "0.00"
   Next i
End Sub
'-------------------------------------------------------------------------------------------
----
Public Sub triDiag(n, ByRef A() As Double, ByRef phi() As Double)
Rem
  'A(0,i) : AW(i), A(1,i): AP(i), A(2,i): AE(i), A(3,i): d(i)
 Dim i As Integer
 Dim p() As Double
 Dim q() As Double
 ReDim p(n)
 ReDim q(n)
Rem
    p(0) = A(2, 0) / A(1, 0)     ' b(1)/a(1) :: AE(0)/AP(0)
    q(0) = A(3, 0) / A(1, 0)     ' d(1)/a(1)
      For i = 1 To n             ' Forward Elimination
        p(i) = A(0, i) / (A(1, i) - A(2, i - 1) * p(i - 1))
Rem       P(i) :: b(i)/[a(i) - c(i)*P(i-1)]
Rem
        q(i) = (A(3, i) + A(0, i) * q(i - 1)) / (A(1, i) - A(0, i) * p(i - 1))
Rem     Q(i) :: [d(i) + c(i)*Q(i-1)] / [a(i) - c(i) * P(i-1)]
      Next i
Rem
      phi(n) = q(n)        ' Backward Substitution
      For i = n - 1 To 0 Step -1
        phi(i) = p(i) * phi(i + 1) + q(i)
      Next i
End Sub
'-------------------------------------------------------------------------------------------
----
Sub checkTDMA()
  Dim n, iRow As Integer
  Dim A() As Double
  Dim phi() As Double
   ' myArray = Range("A1:F10").Value - Read value from worksheets
Rem
   ' Specify matrix value directly
   inMtx = Array(Array(0, 1#, 1#, 1#), Array(2.04, 2.04, 2.04, 2.04), _
                Array(1#, 1#, 1#, 0#), Array(40.8, 0.8, 0.8, 200.8))
Rem
```

```
   ReDim A(0 To UBound(inMtx), 0 To UBound(inMtx(1)))
   n = UBound(inMtx(1))
   ReDim phi(n)
   ReDim A(3, n)
Rem
   iRow = 26
   For i = 0 To 3
     For j = 0 To n
       A(i, j) = inMtx(i)(j)
       Worksheets("SOW").Cells(iRow + i, 3 + j) = A(i, j)
     Next j
   Next i
Rem
   'Call subroutine
   triDiag n, A(), phi()
   For i = 0 To n
       Worksheets("SOW").Cells(iRow + i, n + 3) = phi(i)
       If Abs(A(i, 3)) < 0.001 Then
         Worksheets("SOW").Cells(iRow + i, n + 3).NumberFormat = "0"
       Else
         Worksheets("SOW").Cells(iRow + i, n + 3).NumberFormat = "0.00"
       End If
   Next i
End Sub
```

Gaussian Elimination:

		1v	2u	2v	3u	4v	5u	5v	6u			
	1v	**1.46E+08**	2.20E+07	-5.78E+07	0.00E+00	-1.12E+08	-2.36E+07	2.36E+07	0.00E+00	v1		7.07E+03
	2u	2.20E+07	2.34E+08	-1.21E+07	-5.78E+07	0.00E+00	-1.52E+08	-2.52E+07	0.00E+00	u2		1.00E+04
	2v	-5.78E+07	-1.21E+07	2.34E+08	2.20E+07	0.00E+00	-2.52E+07	-1.52E+08	0.00E+00	v2		1.00E+04
	3u	0.00E+00	-5.78E+07	2.20E+07	1.46E+08	0.00E+00	2.36E+07	-2.36E+07	-1.12E+08	u3	=	7.07E+03
	4v	-1.12E+08	0.00E+00	0.00E+00	0.00E+00	1.13E+08	1.33E+07	-1.14E+06	0.00E+00	v4		0.00E+00
	5u	-2.36E+07	-1.52E+08	-2.52E+07	2.36E+07	1.33E+07	1.56E+08	3.91E+07	-1.14E+06	u5		0.00E+00
	5v	2.36E+07	-2.52E+07	-1.52E+08	-2.36E+07	-1.14E+06	3.91E+07	1.56E+08	1.33E+07	v5		0.00E+00
	6u	0.00E+00	0.00E+00	0.00E+00	-1.12E+08	0.00E+00	-1.14E+06	1.33E+07	1.13E+08	u6		0.00E+00

		1v	2u	2v	3u	4v	5u	5v	6u			
Step-1:	1v	**1.46E+08**	2.20E+07	-5.78E+07	0.00E+00	-1.12E+08	-2.36E+07	2.36E+07	0.00E+00	v1		7.07E+03
	2u	0	**2.31E+08**	-3.44E+06	-5.78E+07	1.69E+07	-1.48E+08	-2.88E+07	0.00E+00	u2		8.94E+03
	2v	0	-3.44E+06	2.11E+08	2.20E+07	-4.43E+07	-3.46E+07	-1.43E+08	0.00E+00	v2		1.28E+04
	3u	0	-5.78E+07	2.20E+07	1.46E+08	0.00E+00	2.36E+07	-2.36E+07	-1.12E+08	u3	=	7.07E+03
	4v	0	1.69E+07	-4.43E+07	0.00E+00	2.74E+07	-4.73E+06	1.69E+07	0.00E+00	v4		5.42E+03
	5u	0	-1.48E+08	-3.46E+07	2.36E+07	-4.73E+06	1.52E+08	4.28E+07	-1.14E+06	u5		1.14E+03
	5v	0	-2.88E+07	-1.43E+08	-2.36E+07	1.69E+07	4.28E+07	1.52E+08	1.33E+07	v5		-1.14E+03
	6u	0	0.00E+00	0.00E+00	-1.12E+08	0.00E+00	-1.14E+06	1.33E+07	1.13E+08	u6		0.00E+00

		1v	2u	2v	3u	4v	5u	5v	6u			
Step-2:	1v	1.46E+08	2.20E+07	-5.78E+07	0.00E+00	-1.12E+08	-2.36E+07	2.36E+07	0.00E+00	v1		7.07E+03
	2u	0	**2.31E+08**	-3.44E+06	-5.78E+07	1.69E+07	-1.48E+08	-2.88E+07	0.00E+00	u2		8.94E+03
	2v	0	0	**2.11E+08**	2.12E+07	-4.40E+07	-3.68E+07	-1.43E+08	0.00E+00	v2		1.29E+04
	3u	0	0	2.12E+07	1.32E+08	4.22E+06	-1.36E+07	-3.08E+07	-1.12E+08	u3	=	9.31E+03
	4v	0	0	-4.40E+07	4.22E+06	2.61E+07	6.12E+06	1.90E+07	0.00E+00	v4		4.76E+03
	5u	0	0	-3.68E+07	-1.36E+07	6.12E+06	5.69E+07	2.43E+07	-1.14E+06	u5		6.88E+03
	5v	0	0	-1.43E+08	-3.08E+07	1.90E+07	2.43E+07	1.49E+08	1.33E+07	v5		-2.40E+01
	6u	0	0	0.00E+00	-1.12E+08	0.00E+00	-1.14E+06	1.33E+07	1.13E+08	u6		0.00E+00

		1v	2u	2v	3u	4v	5u	5v	6u			
Step-3:	1v	1.46E+08	2.20E+07	-5.78E+07	0.00E+00	-1.12E+08	-2.36E+07	2.36E+07	0.00E+00	v1		7.07E+03
	2u	0	2.31E+08	-3.44E+06	-5.78E+07	1.69E+07	-1.48E+08	-2.88E+07	0.00E+00	u2		8.94E+03
	2v	0	0	**2.11E+08**	2.12E+07	-4.40E+07	-3.68E+07	-1.43E+08	0.00E+00	v2		1.29E+04
	3u	0	0	0	**1.30E+08**	8.63E+06	-9.88E+06	-1.65E+07	-1.12E+08	u3	=	8.01E+03
	4v	0	0	0	8.63E+06	1.69E+07	-1.54E+06	-1.08E+07	0.00E+00	v4		7.46E+03
	5u	0	0	0	-9.88E+06	-1.54E+06	5.05E+07	-5.41E+05	-1.14E+06	u5		9.13E+03
	5v	0	0	0	-1.65E+07	-1.08E+07	-5.41E+05	5.19E+07	1.33E+07	v5		8.73E+03
	6u	0	0	0	-1.12E+08	0.00E+00	-1.14E+06	1.33E+07	1.13E+08	u6		0.00E+00

		1v	2u	2v	3u	4v	5u	5v	6u			
Step-4:	1v	1.46E+08	2.20E+07	-5.78E+07	0.00E+00	-1.12E+08	-2.36E+07	2.36E+07	0.00E+00	v1		7.07E+03
	2u	0	2.31E+08	-3.44E+06	-5.78E+07	1.69E+07	-1.48E+08	-2.88E+07	0.00E+00	u2		8.94E+03
	2v	0	0	2.11E+08	2.12E+07	-4.40E+07	-3.68E+07	-1.43E+08	0.00E+00	v2		1.29E+04
	3u	0	0	0	**1.30E+08**	8.63E+06	-9.88E+06	-1.65E+07	-1.12E+08	u3	=	8.01E+03
	4v	0	0	0	0	**1.64E+07**	-8.85E+05	-9.68E+06	7.46E+06	v4		6.92E+03
	5u	0	0	0	0	-8.85E+05	4.98E+07	-1.79E+06	-9.68E+06	u5		9.74E+03
	5v	0	0	0	0	-9.68E+06	-1.79E+06	4.98E+07	-8.85E+05	v5		9.74E+03
	6u	0	0	0	0	7.46E+06	-9.68E+06	-8.85E+05	1.64E+07	u6		6.92E+03

		1v	2u	2v	3u	4v	5u	5v	6u			
Step-5:	1v	1.46E+08	2.20E+07	-5.78E+07	0.00E+00	-1.12E+08	-2.36E+07	2.36E+07	0.00E+00	v1		7.07E+03
	2u	0	2.31E+08	-3.44E+06	-5.78E+07	1.69E+07	-1.48E+08	-2.88E+07	0.00E+00	u2		8.94E+03
	2v	0	0	2.11E+08	2.12E+07	-4.40E+07	-3.68E+07	-1.43E+08	0.00E+00	v2		1.29E+04
	3u	0	0	0	1.30E+08	8.63E+06	-9.88E+06	-1.65E+07	-1.12E+08	u3	=	8.01E+03
	4v	0	0	0	0	**1.64E+07**	-8.85E+05	-9.68E+06	7.46E+06	v4		6.92E+03
	5u	0	0	0	0	0	**4.97E+07**	-2.32E+06	-9.28E+06	u5		1.01E+04
	5v	0	0	0	0	0	-2.32E+06	4.41E+07	3.53E+06	v5		1.38E+04
	6u	0	0	0	0	0	-9.28E+06	3.53E+06	1.30E+07	u6		3.77E+03

		1v	2u	2v	3u	4v	5u	5v	6u			
Step-6:	1v	1.46E+08	2.20E+07	-5.78E+07	0.00E+00	-1.12E+08	-2.36E+07	2.36E+07	0.00E+00	v1		7.07E+03
	2u		2.31E+08	-3.44E+06	-5.78E+07	1.69E+07	-1.48E+08	-2.88E+07	0.00E+00	u2		8.94E+03
	2v	0	0	2.11E+08	2.12E+07	-4.40E+07	-3.68E+07	-1.43E+08	0.00E+00	v2		1.29E+04
	3u	0	0	0	1.30E+08	8.63E+06	-9.88E+06	-1.65E+07	-1.12E+08	u3	=	8.01E+03
	4v	0	0	0	0	1.64E+07	-8.85E+05	-9.68E+06	7.46E+06	v4		6.92E+03
	5u	0	0	0	0	0	**4.97E+07**	-2.32E+06	-9.28E+06	u5		1.01E+04
	5v	0	0	0	0	0	0	**4.40E+07**	3.10E+06	v5		1.43E+04
	6u	0	0	0	0	0	0	3.10E+06	1.12E+07	u6		5.66E+03

		1v	2u	2v	3u	4v	5u	5v	6u			
Step-7:	1v	1.46E+08	2.20E+07	-5.78E+07	0.00E+00	-1.12E+08	-2.36E+07	2.36E+07	0.00E+00	v1		7.07E+03
	2u		2.31E+08	-3.44E+06	-5.78E+07	1.69E+07	-1.48E+08	-2.88E+07	0.00E+00	u2		8.94E+03
	2v	0	0	2.11E+08	2.12E+07	-4.40E+07	-3.68E+07	-1.43E+08	0.00E+00	v2		1.29E+04
	3u	0	0	0	1.30E+08	8.63E+06	-9.88E+06	-1.65E+07	-1.12E+08	u3	=	8.01E+03
	4v	0	0	0	0	1.64E+07	-8.85E+05	-9.68E+06	7.46E+06	v4		6.92E+03
	5u	0	0	0	0	0	4.97E+07	-2.32E+06	-9.28E+06	u5		1.01E+04
	5v	0	0	0	0	0	0	**4.40E+07**	3.10E+06	v5		1.48E+04
	6u	0	0	0	0	0	0	0	1.10E+07	u6		4.65E+03

```
C      TTTTTTT    RRr*     iii    DDD.    III       A         ,GGGGG
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
          write(6,13) i
13        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
```