CWI

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.J.F.M. Schlichting, H.A. van der Vorst

NUMVEC FORTRAN Library manual
Chapter: Partial differential equations
Routine: ICCG3D

Department of Numerical Mathematics Report NM-R9018 October

NUMVEC
is a library of NUMerical software for VECtor and parallel computers in FORTRAN.
The documentation conforms as much as possible to that of the NAG - library. A Subsection: *11.1 Vectorization information* mentions, a.o., whether the code is standard FORTRAN 77. If not, information is given about special vector-syntax used and about specific machine(s) to which the code is aimed.
The source code described can be obtained by writing to the NUMVEC Library Manager at the CWI.

NUMVEC FORTRAN Library manual

Chapter: Partial Differential Equations

Routine: ICCG3D

J.J.F.M. Schlichting
Control Data BV
P.O. Box 11, 2285 VL Rijswijk, The Netherlands

H.A. van der Vorst
Mathematical Institute, University of Utrecht
P.O. Box 80010, 3508 TA Utrecht, The Netherlands

Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

This document describes the NUMVEC FORTRAN Library routine ICCG3D for the solution of the 7-diagonal symmetrical linear systems, that arise from the 7-point discretization of a partial differential equation over a 3-dimensional domain. Use is made of the so-called MICCG-hyperplane method with scaled initial diagonal, 'Eisenstat's trick' and Cyber 205 sparse vector instructions. An example shows a performance of nearly 85 MFLOPS for this routine.

1980 Mathematics Subject Classification (1985 revision): 65V05, 65F10, 65NXX.
1987 CR Categories: G.1.3.

Keywords & Phrases: software, ICCG-method, heptadiagonal systems of linear equations.

Note: This software has been developed while the first author was a visiting scientist at CWI.

Report NM-R9018
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

ICCG3D - NUMVEC FORTRAN Library Routine Document

1. Purpose

ICCG3D solves 7-diagonal symmetrical linear systems, that arise from the 7-point discretization of a PDE over a 3-dimensional field, using the MICCG-hyperplane method with scaled initial diagonal, Eisenstat's trick and the Cyber 205 sparse-vector instructions.

2. Specification

```
      SUBROUTINE ICCG3D(NX,NY,NZ,NMX,XM,RHS,X,NIT,RES,EPS,XW,IFAIL)
C     INTEGER NMX,NX,NY,NZ,NIT,IFAIL
C     REAL XM(4*NMX),RHS(NMX),X(NMX),EPS,RES(NIT),XW(5*NMX)
```

3. Description

ICCG3D solves a 7-diagonal linear system, that arises from a 7-points discretization of a PDE on a rectangle. The system is written in the form

$$A \times U = RHS$$

where A and RHS are the user supplied symmetric matrix and the right hand side respectively. Note that only 4 non-zero diagonals are stored. The approximate solution is found by means of a Modified Incomplete Cholesky Conjugate Gradient method. However, the user remains unaware of the underlying method. The user supplies a relative residual tolerance.

4. References

[1] Schlichting, J.J.F.M. and Van der Vorst, H.A., Solving block bidiagonal linear systems, JCAM, Vol 27, 1989, pp. 323-330.

[2] Eisenstat, S.C., Efficient implementation of a class of preconditioned Conjugate Gradient methods, SIAM J. Sci. Stat. Comput. 2, 1981, pp. 1-4.

[3] Van der Vorst, H.A., ICCG and related methods for 3D problems on vector computers, Computer Physics Communications, 53, 1989, pp. 223-235.

5. Parameters

NX - INTEGER.

NY - INTEGER.

NZ - INTEGER.

On entry NX, NY and NZ must contain the dimensions of the three dimensional field in the x, y and z direction respectively.
The total number of elements (NX*NY*NZ) must not exceed 65535.

NMX - INTEGER .

NMX is the maximum number of points of the three dimensional field.
Unchanged on exit.

XM - REAL array of DIMENSION of at least (NMX*4).

Before entry the subarrays XM(1:NMX), XM(NMX+1:2*NMX), XM(2*NMX+1:3*NMX) and XM(3*NMX+1:4*NMX) must contain the main diagonal and the three subdiagonals, respectively.
Changed on exit.

RHS - REAL array of DIMENSION at least (NX*NY*NZ).
On entry RHS must contain the right-hand side.

X - REAL array of DIMENSION at least (NX*NY*NZ).
On exit X will contain the solution array.

NIT - INTEGER.
On entry NIT specifies the maximum number of iteration steps.
On exit NIT contains the actual number of iterations performed. If the initial and returned value of NIT are equal, the required precision may not have been reached.

RES - REAL array of DIMENSION at least (NIT + 1).
On exit RES(1:NIT + 1) will contain the residuals after each iteration. RES(1) is the initial residual, RES(I) is the residual after I − 1 iteration steps.

EPS - REAL.
On entry EPS must contain the relative residual tolerance.

XW - REAL array of DIMENSION at least (5*NMX).
XW is used as working space.

IFAIL - INTEGER.
Before entry, IFAIL must be assigned a value. For users not familiar with this parameter (described in Chapter P01 of [NAG]) the recommended value is 0.
Unless the routine detects an error (see Section 6), IFAIL contains 0 on exit.

6. Error indicators and warnings

Errors detected by the routine:-

IFAIL = 1
no convergence.

7. Auxiliary routines

This routine calls the NUMVEC Library routines PLAN3D and P01AAF.

8. Timing

The time for a single iteration is $0.22N^3 + 50N$ microseconds.

9. Storage

Internally declared arrays contain NX × NY × NZ INTEGERs and 6 × NX × NY × NZ bits.

10. Accuracy

If the process converges the residual becomes less than EPS. The process terminates at the first residual which is less than EPS × RES(1). The routine delivers bitwise the same results as the straightforward standard MICCG procedure.

11. Further comments

Labelled COMMON blocks BITV and INDX are used by the routine and must therefore be avoided by users.

11.1. Vectorization information

The routine is written in FTN200 for the Cyber 205 and and makes heavy use of so-called special calls. Because the routine is based on the Cyber 205 sparse vector instructions it cannot be used on any other system.

12. Keywords

Block diagonal linear system.
ICCG algorithm.
Sparse vector instructions.
Vectorization.

13. Example

The example shows two similar executions of the routine for different problem sizes. The problems have been discussed in ref [3].

13.1. Program text

```
      PROGRAM IC3DU
      SAVE
      COMMON HX(202),HY(202),RESD(751),X(60000),
     1 NXP(10),NYP(10),NZP(10)
      COMMON /MATRIX/ A(-3000:60000,10)
* THIS CODE IS AN ADAPTED VERSION OF
* CODE WRITTEN BY:
*  13  OCT 1986, H.A. VAN DER VORST (DELFT UNIVERSITY)
* SOLUTION OF 3D PARTIAL DIFFERENTIAL EQUATIONS, DISCRETISED
* BY FINITE DIFFERENCES. BOUNDARY CONDITIONS:
*   NEUMANN   ON LOWER Y-BOUNDARY
*   DIRICHLET ON LOWER X-BOUNDARY
*   DIRICHLET ON UPPER Y-BOUNDARY
*   NEUMANN   ON UPPER X-BOUNDARY
*   DIRICHLET U=0 ON LOWER Z-BOUNDARY
*   NEUMANN DU/DN=0  ON UPPER Z-BOUNDARY
*
*  NXP, NYP AND NZP DENOTE THE NUMBER OF GRIDPOINTS
*  IN THE RESPECTIVE DIRECTIONS.
*
*  THE DISCRETISED SYSTEMS ARE ITERATIVELY SOLVED BY
* ICCG EISENSTAT MINCH0 SPARSE VECTORS CY205
      DO 2 I=1,60000
    2   X(I)=0.
      DO 3 J=1,9
        DO 3 I=-3000,60000
    3     A(I,J)=0.
      NXP(7)=21
      NYP(7)=21
      NZP(7)=21
      NXP(8)=36
      NYP(8)=36
```

```
      NZP(8)=36
      DO 100 IP=7,8
      NX=NXP(IP)
      NXX=NX-1
      NY=NYP(IP)
      NYY=NY-1
      NZ=NZP(IP)-1
      NXNY=NX*NY
      NXNYNZ=NZ*NXNY
      DO 10 I=1,NX
   10 HX(I)=5./FLOAT(NX-1)
      DO 20 I=1,NY
   20 HY(I)=5./FLOAT(NY-1)
      HZZ=5./(NZP(IP)-1.)
      EPS=1.E-6
      PRINT 1007,IP
 1007 FORMAT(/' PROBLEM NR',I2,':'/
     1 '       PROBLEM SUGGESTED BY HAYAMI AND HARADA (NEC):'/
     2 '       CUBE (0,5)X(0,5)X(0,5) WITH SPATIALLY INCREASED '/
     3 '       DIFFUSION, PLANES IN VIEW HAVE NEUMANN, OTHERS '/
     4 '       HAVE DIRICHLET BOUNDARY CONDITIONS. '/
     5 '       THE RIGHT-HAND SIDE IS EQUAL TO 500. ')
      PRINT 1000,NXX,NYY,NZ
 1000 FORMAT(' NUMBER OF UNKNOWNS IN THE X-,Y- AND Z-DIRECTIONS',3I4)
      CALL EPSD3D(0.,5.,NX,HX,0.,5.,NY,HY,0.,5.,NZ,HZZ,NXNY,NXNYNZ,
     1 X,NIT,EPS,IERR,RESD)
  100 CONTINUE
      STOP
      END
      SUBROUTINE EPSD3D(XL,XU,NX,HX,YL,YU,NY,HY,ZL,ZU,NZ,HZZ,NXNY,N,
     1 X,NITC,EPSC,IFAIL,RESD)
      SAVE
      DIMENSION RESD(1),X(*),HX(1),HY(1)
      COMMON /MATRIX/ A(-3000:60000,10)
      REAL MFLOPS
      NIT=NITC
      EPS=EPSC
      NXC=NX
      NYC=NY
      NZC=NZ
      NXNYC=NXNY
      NNC=N
      CALL PDEDI3(NXC,NYC,NZC,NXNYC,NNC,XL,XU,YL,YU,ZL,ZU,A(1,1),A(1,2),
     1 A(1,3),A(1,4),A(1,5),HX,HY,HZZ)
      TIMA=SECOND(T)-TIME
       PRINT 991
  991 FORMAT(///,' HYPERPLANE,EISENSTAT,SCALE,SPARSE VECTORS ON CY205')
      TIME=SECOND(T)
        NQ = 63001
        CALL ICCG3D(NXC,NYC,NZC,NQ,A(1,1),A(1,5),X,NIT,RESD,EPS,
```

```
     1    A(1,6),IFAIL)
      TIME=SECOND(T)-TIME
       NFLOPS=((NIT+1)*24+14)*NNC
      MFLOPS = NFLOPS*1.E-6/TIME
      PRINT 30,TIME,MFLOPS,NIT
      PRINT 1505,' FIRST 5 RESIDUALS',(I-1,RESD(I),I=1,5)
      PRINT 1505,' LAST 5 RESIDUALS',(I-1,RESD(I),I=NIT-3,NIT+1)
 1505 FORMAT(A,/,5(' ',I4,3X,E10.3))
  999 CONTINUE
   30 FORMAT(/'   TIME FOR ICCG/INC.DEC.   ',F10.3,
     1      '  MFLOPS=',F8.3,'  NIT= ',I5)
      RETURN
      END
      FUNCTION DC(X,Y,Z)
      DD=1.
      IF (X.GT.5./3.) DD=DD*10.
      IF (X.GT.10./3.) DD=DD*10.
      IF (Y.GT.5./3.) DD=DD*10.
      IF (Y.GT.10./3.) DD=DD*10.
      IF (Z.GT.5./3.) DD=DD*10.
      IF (Z.GT.10./3.) DD=DD*10.
      DC=DD
      RETURN
      END
      SUBROUTINE PDEDI3(NX,NY,NZ,NXNY,N,XL,XU,YL,YU,ZL,ZU,DIAG,DIAGX,
     1    DIAGY,DIAGZ,RHS,HX,HY,HZZ)
      SAVE
      DIMENSION DIAG(*),DIAGX(*),DIAGY(*),DIAGZ(*),RHS(*),
     1    HX(*),HY(*)
*  DISCRETISATION OF 3D PDE.
*  MESH-SIZES ARE ASSUMED TO BE UNIFORM
*  ZC( ) REPRESENTS THE COEFFICIENT OF THE SECOND
*  DERIVATIVE IN Z.
      IP=1
      N=NX*NY*NZ
      DO 10 I=1,N
        DIAG(I)=0.
        DIAGX(I)=0.
        DIAGY(I)=0.
        DIAGZ(I)=0.
        RHS(I)=0.
   10 CONTINUE
      Z=ZL
      Z=ZL+HZZ
      DO 1000 IZ=1,NZ
        NXC=NX
        NYC=NY
        NXNYC=NXNY
        CALL PDEES3(NXC,NYC,NXNYC,XL,XU,YL,YU,HX,HY,
     1      DIAG(IP),DIAGX(IP),DIAGY(IP),
```

```
     2        RHS(IP),Z)
          CALL COMPR(NXC,NYC,NXNYC,DIAG(IP),DIAGX(IP),DIAGY(IP),
     1        RHS(IP))
          IJ=IP
          ZZL=Z-HZZ/2.
          ZZU=Z+HZZ/2.
          YY=YL
          DO 100 IY=1,NYC
            DY=HY(1)
            IF (IY.EQ.1) DY=DY/2.
            XX=XL+HX(1)
            DO 20 IX=1,NXC
            D1=DC(XX,YY,ZZL)
            IF (IZ.EQ.NZ) THEN
            D2=0.
            ELSE
              D2=DC(XX,YY,ZZU)
            ENDIF
              DX=HX(1)
              IF (IX.EQ.NXC) DX=HX(1)/2.
              OPP=DX*DY/(HZZ*HZZ)
              DIAG (IJ)=DIAG(IJ)+OPP*(D1+D2)
              IF (IZ.NE.NZ) DIAGZ(IJ)=-OPP*D2
              IJ=IJ+1
              XX=XX+HX(1)
   20       CONTINUE
            YY=YY+HY(1)
  100     CONTINUE
          IP=IP+NXC*NYC
          Z=Z+HZZ
 1000 CONTINUE
      NX=NXC
      NY=NYC
      NXNY=NXNYC
      N=NX*NY*NZ
      RETURN
      END
      SUBROUTINE PDEES3(NX, NY, NXNY, A, B, C, D, HX, HY,
     1 DIAG, SDIAG, BSDIAG, RHS,Z)
      SAVE
      DIMENSION HX(1), HY(1), DIAG(1), SDIAG(1), BSDIAG(1),RHS(1)
      YIJ = C
      IJ = 0
      DO 120 J=1,NY
        H1 = 1.0
        H3 = 1.0
        IF (J.NE.1) H1 = HY(J-1)
        IF (J.NE.NY) H3 = HY(J)
        XIJ = A
        DO 110 I=1,NX
```

```
      IJ = IJ + 1
      IF (I.NE.1) GO TO 50
      XIJ = A
      D1 = 0.
      D2 = 0.
      E1 = 0.
      E2 = 0.
      F1 = 0.
      F2 = 0.
      COEF1 = 0.
      COEF2 = 0.
      H2 = 1.
50    IF (J.NE.1) GO TO 60
      YIJ = C
      D1 = 0.
      D4 = 0.
      E1 = 0.
      E4 = 0.
      F1 = 0.
      F4 = 0.
      COEF1 = 0.
      COEF4 = 0.
      H1 = 1.
60    IF (I.NE.NX) GO TO 70
      E3 = 0.
      E4 = 0.
      D3 = 0.
      D4 = 0.
      F3 = 0.
      F4 = 0.
      COEF3 = 0.
      COEF4 = 0.
      H4 = 1.
70    IF (J.NE.NY) GO TO 80
      E2 = 0.
      E3 = 0.
      D2 = 0.
      D3 = 0.
      F2 = 0.
      F3 = 0.
      COEF2 = 0.
      COEF3 = 0.
      H3 = 1.
80    CONTINUE
      IF (.NOT.(I.NE.NX .AND. J.NE.NY)) GO TO 90
      H4 = HX(I)
      E3 = DC(XIJ+H4/2.,YIJ+H3/2.,Z)
      D3 = DC(XIJ+H4/2.,YIJ+H3/2.,Z)
      F3 = 500.
      COEF3 = 0.
```

```
          DIAG(IJ+NX) = E3
          SDIAG(IJ+NX) = D3
          BSDIAG(IJ+NX) = F3
          RHS(IJ+NX) = COEF3
   90     IF (.NOT.(I.NE.NX .AND. J.NE.1)) GO TO 100
          H4 = HX(I)
          E4 = DIAG(IJ)
          D4 = SDIAG(IJ)
          F4 = BSDIAG(IJ)
          COEF4 = RHS(IJ)
  100     CONTINUE
          DIAG(IJ) = ((H4*E4+H2*E1)/H1+(H2*E2+H4*E3)/H3+(H1*D1+H3*D2)/
     1     H2+(H3*D3+H1*D4)/H4)*0.5 + (COEF1*H1*H2+COEF2*H2*H3+COEF3*
     2     H3*H4+COEF4*H4*H1)*0.25
          SDIAG(IJ) = -(H3*D3+H1*D4)*0.5/H4
          BSDIAG(IJ) = -(H2*E2+H4*E3)*0.5/H3
          RHS(IJ) = (F1*H1*H2+F2*H2*H3+F3*H3*H4+F4*H4*H1)*0.25
          XIJ = XIJ + H4
          H2 = H4
          E2 = E3
          D2 = D3
          F2 = F3
          COEF2 = COEF3
          E1 = E4
          D1 = D4
          F1 = F4
          COEF1 = COEF4
  110   CONTINUE
        YIJ = YIJ + H3
  120 CONTINUE
*   S-BOUNDARY.
*
      XIJ = A
       YIJ = C
      H2 = 1.
      H3 = HY(1)
      E2 = 0.
      DO 130 I=1,NX
        H4 = 1.
        IF (I.NE.NX) H4 = HX(I)
        E3 = 0.
        IF (I.NE.NX) E3 = DC(XIJ+H4/2.,YIJ+H3/2.,Z)
        E2 = E3
        H2 = H4
        XIJ = XIJ + H4
  130 CONTINUE
*
*   W-BOUNDARY.
*
      XIJ = A
```

```
      YIJ = C
      D4 = 0.0
      H1 = 1.0
      H4 = HX(1)
*
*  DIRICHLET-CONDITION FOR W-BOUNDARY.
*
      IW = 2
      DO 190 I=1,NY
        IW = IW + NX
        IF (I.EQ.NY) GO TO 190
        YIJ = YIJ + HY(I)
  190 CONTINUE
  200 CONTINUE
*
* THE NORTH-BOUNDARY..
*
      XIJ = A
      YIJ = D
      H1 = HY(NY-1)
      E1 = 0.0
      H2 = 1.0
      IJ = NX*(NY-2)
      DO 230 I=1,NX
        IJ = IJ + 1
        BSDIAG(IJ) = 0.0
        IF (I.EQ.NX) GO TO 230
        XIJ = XIJ + HX(I)
  230 CONTINUE
  240 CONTINUE
*
* EAST-BOUNDARY..
*
      XIJ = B
      YIJ = C
      D1 = 0.0
      H1 = 1.0
      H2 = HX(NX-1)
      IE = NX
      DO 250 I=1,NY
        H3 = 1.
        IF (I.NE.NY) H3 = HY(I)
        D2 = 0.0
        D2 = DC(XIJ-H2/2.,YIJ+H3/2.,Z)
        IE = IE + NX
        D1 = D2
        H1 = H3
        YIJ = YIJ + H3
  250 CONTINUE
      NXNY = NX*NY
```

```
      RETURN
      END
      SUBROUTINE COMPR(NX, NY, NXNY, DIAG, SDIAG, BSDIAG, RHS)
      SAVE
      DIMENSION  DIAG(1), SDIAG(1), BSDIAG(1), RHS(1)
   20 CONTINUE
*
* WEST—BOUNDARY ELIMINATED, SYSTEM BECOMES NY SMALLER..
*
      ISCH = 1
      NX = NX - 1
      IJ = 1
      DO 60 I=1,NY
        DO 50 J=1,NX
          IJISCH = IJ + ISCH
          DIAG(IJ) = DIAG(IJISCH)
          SDIAG(IJ) = SDIAG(IJISCH)
          BSDIAG(IJ) = BSDIAG(IJISCH)
          RHS(IJ) = RHS(IJISCH)
          IJ = IJ + 1
   50   CONTINUE
        ISCH = ISCH + 1
   60 CONTINUE
*
* DIRICHLET—CONDITION ON NORTH—BOUNDARY..
*
      NY = NY - 1
      NXNY = NX*NY
      RETURN
      END
```

13.2. Program data

None.

13.3. Program results

```
THE OUTPUT OF THE EXAMPLE PROGRAM IS:


PROBLEM NR 7:
    PROBLEM SUGGESTED BY HAYAMI AND HARADA (NEC):
    CUBE (0,5)X(0,5)X(0,5) WITH SPATIALLY INCREASED
    DIFFUSION, PLANES IN VIEW HAVE NEUMANN, OTHERS
    HAVE DIRICHLET BOUNDARY CONDITIONS.
    THE RIGHT-HAND SIDE IS EQUAL TO 500.
NUMBER OF UNKNOWNS IN THE X-,Y- AND Z-DIRECTIONS  20  20  20
```

```
HYPERPLANE,EISENSTAT,SCALE,SPARSE VECTORS ON CY205

  TIME FOR ICCG/INC.DEC.        0.064  MFLOPS=  61.342  NIT=    19
FIRST 5 RESIDUALS
    0    0.972E+03  1  0.339E+03     2     0.124E+03  1     0.669E+02     4     0.466E+02
LAST 5 RESIDUALS
   15    0.293E-01     16    0.118E-01     17    0.449E-02     18    0.180E-02     19    0.827E-03

 PROBLEM NR 8:
     PROBLEM SUGGESTED BY HAYAMI AND HARADA (NEC):
     CUBE (0,5)X(0,5)X(0,5) WITH SPATIALLY INCREASED
     DIFFUSION, PLANES IN VIEW HAVE NEUMANN, OTHERS
     HAVE DIRICHLET BOUNDARY CONDITIONS.
     THE RIGHT-HAND SIDE IS EQUAL TO 500.
 NUMBER OF UNKNOWNS IN THE X-,Y- AND Z-DIRECTIONS  35  35  35


HYPERPLANE,EISENSTAT,SCALE,SPARSE VECTORS ON CY205

  TIME FOR ICCG/INC.DEC.        0.385  MFLOPS=  84.460  NIT=    30
FIRST 5 RESIDUALS
    0    0.104E+04     1     0.439E+03     2     0.213E+03     3     0.158E+03     4     0.107E+03
LAST 5 RESIDUALS
   26    0.793E-02     27    0.406E-02     28    0.234E-02     29    0.128E-02     30    0.650E-03
```