Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

P.M. de Zeeuw

NUMVEC FORTRAN library manual
Chapter: Elliptic PDEs
Routine: MGD1V and MGD5V

Department of Numerical Mathematics Report NM-R8624 November

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

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: Elliptic PDEs

Routine: MGD1V and MGD5V

P.M. de Zeeuw

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

The NUMVEC FORTRAN Library routines MGD1V and MGD5V are described.
MGD1V solves 7-diagonal linear systems, that arise from 7-point discretizations of elliptic PDEs on a rectangle, using a multigrid technique with ILU relaxation as smoothing process.
MGD5V solves 7-diagonal linear systems, that arise from 7-point discretizations of elliptic PDEs on a rectangle, using a multigrid technique with ILLU relaxation as smoothing process.

1980 Mathematics subject classification (1985 revision): 65V05, 65N20, 65F10.
1980 CR Categories: 5.17.

Keywords & Phrases: Elliptic PDEs, Galerkin approximation, multigrid methods, software, sparse linear systems, ILU relaxation, ILLU relaxation.

Note: The implementations are available in auto-vectorizable ANSI FORTRAN 77.

Report NM-R8624
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

MGD1V - NUMVEC FORTRAN Library Routine Document

1. Purpose

MGD1V solves 7-diagonal linear systems, that arise from 7-point discretizations of elliptic PDEs on a rectangle, using a multigrid technique.

2. Specification

```
      SUBROUTINE MGD1V(LEVELS, NXC, NYC, NXF, NYF, NM,
     +                 A, U, RHS, UB, US, WORK, RESNO,
     +                 IOUT, ISTART, IPREP, MAXIT, TOL, IFAIL)
C     INTEGER LEVELS, NXC, NYC, NXF, NYF, NM, IOUT(5), ISTART, IPREP,
C    +        MAXIT, IFAIL
C     REAL    A(NM*7), U(NM), RHS(NM), UB(NXF*NYF), US(NM), WORK(NXF),
C    +        RESNO, TOL
```

3. Description

MGD1V solves a 7-diagonal linear system, that arises from a 7-point discretization of an elliptic PDE on a rectangle. The system is written in the form

$$A \times U = RHS, \tag{1}$$

where A and RHS are the user-supplied matrix and right-hand side respectively. Note, that only the 7 non-zero diagonals are stored. The approximate solution is found by means of the multigrid correction storage algorithm

with: smoothing by ILU relaxation, symmetric 7-point prolongation and restriction, Galerkin approximation of coarse-grid matrices, also on the coarsest grid ILU relaxation is used instead of an exact solver.

The relation between a coarse grid and a fine grid (two subsequent levels) is the following:

let $C = \{ (x_i,y_j) | x_i = (i-1)h+o_1,\ i = 1\ (1)\ NXC;$

$$y_j = (j-1)h+o_2,\ j = 1\ (1)\ NYC\ ) \}$$

be the coarse grid, then the next fine grid is

$$F = \{ (x_i, y_j) | x_i = (i-1)\frac{h}{2}+o_1,\ i = 1\ (1)\ (2\ NXC-1);$$

$$y_j = (j-1)\frac{h}{2}+o_2,\ j = 1\ (1)\ (2\ NYC-1)\ \}$$

The user needs no knowledge of the multigrid method, it suffices to comply to the specification of the parameters.

The user supplies a bound of the l_2-norm of the residual vector. The user may supply an initial approximation of the solution (alternatively, the zero solution is used as initial approximation).

4. References

[1] Hemker, P.W., On the comparison of line-Gauß Seidel and ILU relaxations in multigrid algorithms. In: Computational and asymptotic methods for boundary and interior layers.

(J.J.H. Miller ed.) pp. 269-277, Boole press, 1982.

[2] Hemker, P.W., Kettler, R., Wesseling, P., de Zeeuw, P.M., Multigrid Methods: Development of fast solvers, Appl. Math. Comp. 13, pp. 311-326 (1983)

[3] Hemker, P.W., Wesseling, P., de Zeeuw, P.M., A portable vector code for autonomous multigrid modules. In: PDE software: modules, interfaces and systems. (B. Engquist and T. Smedsaas eds.), pp. 29-40, Procs. IFIP WG 2.5 working conference, North-Holland, 1984.

[4] Hemker, P.W., de Zeeuw, P.M., Some implementations of multigrid linear system solvers. Lectures held at the University of Bristol, England, sept.1983. In: D.J. Paddon and Holstein (eds.), Multigrid methods for integral and differential equations. The Institute of Mathematics and its Applications Conference Series, Oxford University Press, 1985, pp.85-116.

[5] Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[6] Sonneveld, P., Wesseling, P., de Zeeuw, P.M., Multigrid and conjugate gradient methods as convergence acceleration techniques. Lectures held at the University of Bristol, England, sept.1983. In: D.J. Paddon and H. Holstein (eds.), Multigrid methods for integral and differential equations. The Institute of Mathematics and its Applications Conference Series, Oxford University Press, 1985, pp.117-167.

5. Parameters

LEVELS - INTEGER.
On entry, LEVELS must specify the number of levels in the multigrid method.
$1 \leqslant$ LEVELS $\leqslant 12$.
Unchanged on exit.

NXC - INTEGER.
NYC - INTEGER.
On entry, NXC and NYC must specify the number of vertical and horizontal grid-lines respectively, on the coarsest grid.
NXC, NYC $\geqslant 3$.
Unchanged on exit.

NXF - INTEGER.
NYF - INTEGER.
On entry, NXF and NYF must specify the number of vertical and horizontal grid-lines respectively, on the finest grid.
NXF $\leqslant 65535$.
Unchanged on exit.

NM - INTEGER.
On entry, NM must specify the number of grid-points on all grids together.
Unchanged on exit.

Note that the following relations should hold:

$$\text{NXF} = (\text{NXC}-1) \times 2^{\text{LEVELS}-1} + 1$$

$$\text{NYF} = (\text{NYC}-1) \times 2^{\text{LEVELS}-1} + 1$$

$$\text{NM} = \sum_{\text{L}=1}^{\text{LEVELS}} ((\text{NXC}-1) \times 2^{\text{L}-1}+1)((\text{NYC}-1) \times 2^{\text{L}-1}+1)$$

The program checks the consistency of these data

Examples:

LEVELS	2	3	4	5	6	7
NXC	5	5	5	5	5	5
NYC	5	5	5	5	5	5
NXF	9	17	33	65	129	257
NYF	9	17	33	65	129	257
NM	106	395	1484	5709	22350	88399

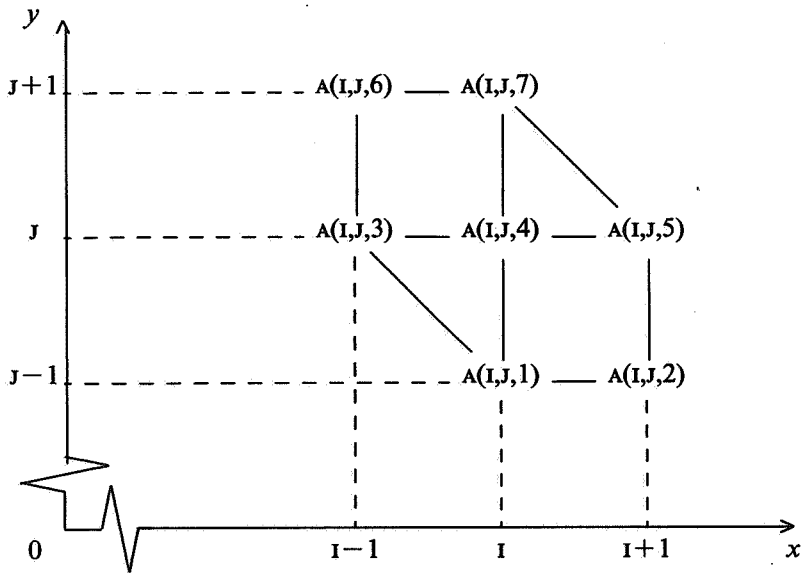
LEVELS	2	3	4	5	6	7
NXC	5	5	5	5	5	5
NYC	3	3	3	3	3	3
NXF	9	17	33	65	129	257
NYF	5	9	17	33	65	129
NM	60	213	774	2919	11304	44457

A - REAL array of DIMENSION at least (NM*7)
Before entry, if IPREP = 0 then the first NXF*NYF*7 elements of A must contain the 7-diagonal matrix of the linear system (1). (If IPREP = 0 then the contents of A will be overwritten by the program.)
Before entry, if IPREP = 1 then the first NM*7 elements of A must contain the unaltered contents of A after the last previous call of MGD1V. (If IPREP=1 those contents are assumed to be the decompositions of the matrix of (1) and its coarse-grid approximations.)

The following holds for the case that IPREP = 0:

The easiest way for the user to fill the matrix A is writing a subroutine where the actual argument A is handled as an adjustable array with dimensions (NXF, NYF, 7).

The 7-point difference molecule at the point with subscripts (I, J) is positioned in the x,y -plane as follows:



Important: the user has to provide the matrix A only on the finest grid. The coarse-grid matrices are computed internally by the routine by means of Galerkin approximation.

Important: the user has to take care that parts of the molecules outside the domain are initialized to zero, otherwise wrong results are produced.

Remark: problems with topologically non-rectangular regions can also be solved using this routine by surrounding the region by a circumscribing topological rectangle. Let (ie,je) denotate a nodal point external to the region of interest, then set A(ie,je,k)=0 for k=1,2,3,5,6,7 and A(ie,je,4)=eps with eps small, e.g. 10^{-9} (further set RHS(ie,je)=0).

On exit, the first NM*7 elements of A contain the incomplete Crout decompositions of the user-provided matrix A and its coarse-grid approximations.
Hence, the contents of A are altered on exit.

U - REAL array of DIMENSION at least (NM)

Before entry, but only if the routine is entered with ISTART = 1 or with ISTART = 2, the first NXF*NYF elements of U must contain an initial estimate for the iterative process. If ISTART = 0 no initialization of U is necessary (a zero initial estimate is assumed).

The easiest way for the user to fill the initial estimate is writing a subroutine where the actual argument U is handled as an adjustable array with dimensions (NXF, NYF).

On successful exit, the first NXF*NYF elements of U contain the (approximate) numerical solution.

RHS - REAL array of DIMENSION at least (NM)

Before entry, the first NXF*NYF elements of RHS must contain the right-hand side of the linear system (1), the other elements are used as workspace.

The easiest way for the user to fill the right-hand side is writing a subroutine where the actual argument RHS is handled as an adjustable array with dimensions (NXF, NYF).

Important: the user has to provide the right-hand side of the discretized equation only on the finest grid.

The first NXF*NYF elements of RHS are unchanged on exit.

UB - REAL array of DIMENSION at least (NXF*NYF)
Before entry, but only if the routine is entered with ISTART = 2, the first NXF*NYF elements of UB must contain the residual RHS−A*U of the initial estimate U.
On successful exit, the first NXF*NYF elements of UB contain the residual of the approximate numerical solution U.

US - REAL array of DIMENSION at least (NM).
Used as workspace.

WORK - REAL array of DIMENSION at least (NXF).
Used as workspace.

RESNO - REAL.
On exit, RESNO contains the l_2 -norm of the residual.

IOUT - INTEGER array of DIMENSION at least (5)
IOUT governs the amount of information about the solution process delivered to the user. Smaller IOUT-values mean less output. The user may select the unit-number on which this output is to appear by a call of X04ABF. Before entry, the first 5 elements of IOUT must contain one of the following values:

IOUT(1) ≥ 1 confirmation of input data
≤ 0 none

IOUT(2) ≥ 2 matrices on all levels and right hand side on highest level
$= 1$ matrix and right-hand side on highest level
≤ 0 none

IOUT(3) ≥ 2 matrix-decompositions on all levels
$= 1$ matrix-decomposition on highest level
≤ 0 none

IOUT(4) ≥ 4 norms of residuals, reduction factors, final solution, final residual
$= 3$ norms of residuals, reduction factors, final residual
$= 2$ norms of residuals, reduction factors (i.e. monitoring the convergence-behaviour)
$= 1$ final norm of residual, number of iterations
≤ 0 none

IOUT(5) ≥ 1 the time spent in various subroutines
≤ 0 none

The contents of IOUT are unchanged on exit.

ISTART - INTEGER.

On entry, ISTART must be set to 0, 1 or 2.

ISTART = 0 means that the initial estimate is zero, initialization of U is not necessary;
= 1 means that the user provides an initial estimate of the solution in U;
= 2 means that the user provides an initial estimate of the solution in U and the residual of U in UB.

Unchanged on exit.

IPREP - INTEGER.

On entry, IPREP must be set to 0 or 1.

IPREP = 1 if MGD1V has been called before and the information in A after that call has not been changed. (This is useful if the user wants to refine the numerical solution by re-entering the routine or if the user desires to solve a sequence of problems with identical matrices but different right-hand sides, this option then prevents a new setup of coarse-grid matrices and decompositions which have already been computed.)
= 0 if MGD1V has not been called before, or if after a call of MGD1V the information in A has been altered.

Unchanged on exit.

MAXIT - INTEGER.

TOL - REAL.

On entry, MAXIT must specify the maximum number of allowed multigrid iterations and TOL must specify the tolerance desired by the user, where TOL is a bound of the l_2 -norm of the residual.

If during the multigrid process either MAXIT iterations have been performed or the tolerance has been reached, multigrid cycling is stopped.

For MAXIT a value of about 10 is adequate for many problems, clearly TOL should not be less than a reasonable multiple of NXF*NYF times the machine accuracy.

Both MAXIT and TOL are unchanged on exit.

IFAIL - INTEGER.

For this routine, the normal use of IFAIL is extended to control the printing of error and warning messages as well as specifying hard or soft failure (see [5], chapter P01). On entry IFAIL must be set to a value with the decimal expansion cba, where each of the decimal digits c, b and a must have the value 0 or 1.

a = 0 specifies hard failure, otherwise soft failure;
b = 0 suppresses error messages, otherwise error messages will be printed (see section 6);
c = 0 suppresses warning messages, otherwise warning messages will be printed (see section 6).

For users not familiar with this parameter the recommended value is 110 (i.e. hard failure with all messages printed).

Unless the routine detects an error (see section 6), IFAIL contains 0 on exit.

6. Error indicators and warnings

Errors detected by the routine:-

For some errors the routine outputs an explanatory message on the current error message unit (see routine X04AAF), unless suppressed by the value of IFAIL on entry.

IFAIL = 1
On entry, LEVELS is out of bounds.

IFAIL = 2
On entry, NXC or NYC is less than 3.

IFAIL = 3
On entry, no consistency among LEVELS, NXC and NXF.

IFAIL = 4
On entry, no consistency among LEVELS, NYC and NYF.

IFAIL = 5
On entry, NM is wrong.

IFAIL = 6
divergence, maybe due to this routine or to user discretization.

IFAIL = 7
poor convergence, maybe due to this routine or to user discretization or to reaching roundoff level.

IFAIL = 8
convergence, but MAXIT iterations performed without reaching TOL. This could be due to a value of MAXIT which is too small or to reaching roundoff level. The user is advised to monitor the convergence behaviour by choosing IOUT(4) $\geqslant$ 2.

7. Auxiliary routines

This routine calls the following NUMVEC library routines:
COPY, CTUMV, CTUPF, CYCLES, DECOMP, FOURTH, GALERK, ILUDEC, OUTMAT, OUTVEC, PREPAR, PROLON, P01AAF, RAP, RESTRI, SOLVE, TIMING, VL2NOR, X04AAF, X04ABF and ZERO.

8. Timing

The time taken per multigrid iteration is proportional to NXF $\times$ NYF.
However, the vector performance (mflop-rate) increases with increasing NXF, therefore it is advised to take the longest side of the rectangular grid along the x-axis assuming that for many problems there is not much orientation-sensitivity in the algorithm used.
According to multigrid theory it is plausible that the convergence rate is independent of the chosen meshwidth.

9. Storage

Internally declared arrays contain 3 REAL and 50 INTEGER elements.

10. Accuracy

If the process converges, and MAXIT is large enough, the l_2 -norm of the residual becomes less than TOL.

11. Further comments

Labelled common blocks CPU and POI are used by the routine and must therefore be avoided by users.
The user is strongly recommended to set IFAIL to obtain self-explanatory error- and advisory-messages. The user may select the unit numbers on which this output is to appear by calls of X04AAF (for error messages) or X04ABF (for advisory messages) - see section 13 for an example. Otherwise the default unit numbers will be used.

11.1. Vectorization information

This program is entirely written in ANSI FORTRAN 77 and auto-vectorizes both on the CYBER 205 and the CRAY-1. On the CYBER 205 the compiler options OPTimize and UNSafe are recommended. The UNSafe option can be used safely, because throughout the whole program all vectorizable DO-loops with possibly more than 65535 iterations are subdivided in loops which do not violate the constraint on the loop count.

12. Keywords

Elliptic PDEs,
Galerkin approximation,
Multigrid methods,
Sparse linear systems,
ILU relaxation.

13. Example

We solve the Poisson equation on the unit square with Dirichlet boundary conditions and the right-hand side constructed according to the exact solution $x\,(1-x) + y\,(1-y)$. In this example the boundary conditions are eliminated.

First we try 3 iterations with a zero initial approximation and if no divergence is found (using the soft fail option) we try to refine the solution until the residual norm becomes lesser than 10^{-9} using the former approximation.

Note the calls to X04AAF and X04ABF prior to the call of MGD1V.

13.1. Program text

```
      PROGRAM EXAMPL
C
      PARAMETER( LEVELS = 6, NXC =   5, NYC =   5,
     +                       NXF = 129, NYF = 129, NM = 22350)
C
      REAL RHS, A, VS, WORK, V, VB
      COMMON /ARRAYS/ RHS(NM), A(7*NM), VS(NM), WORK(NXF), V(NM),
     +                VB(NXF*NYF)
C     These arrays are declared in a labelled common block in order to
C     allow large-page mapping on the CYBER205.
C
      INTEGER IOUT(5)
C
      DATA IOUT/1,0,0,2,1/
```

```
C
      DATA NOUT/6/
C
      OPEN(UNIT=NOUT,FILE='OUTPUT')
      CALL X04AAF(1,NOUT)
      CALL X04ABF(1,NOUT)
C
      WRITE(NOUT,999)
  999 FORMAT('1 MGD1V EXAMPLE PROGRAM RESULTS.')
C
C     Problem set up.
C     subroutine MATRHS is an example of a subroutine which fills the
C     matrix and the right-hand side.
C
      CALL MATRHS(A,RHS,NXF,NYF,NOUT)
C
C     Approximate the solution of the linear system by performing 3
C     MGD1V iterations, using the soft fail option.
C
      ISTART= 0
      IPREP = 0
      MAXIT = 3
      TOL   = 0.0
      IFAIL = 111
C
      CALL MGD1V(LEVELS, NXC, NYC, NXF, NYF, NM,
     +           A, V, RHS, VB, VS, WORK, RESNO,
     +           IOUT, ISTART, IPREP, MAXIT, TOL, IFAIL)
C
C     Possible refinement until residual norm .LT. 1.0E-9
C
      IF( IFAIL.GT.0 .AND. IFAIL.GE.7 ) THEN
        ISTART= 2
        IPREP = 1
        MAXIT = 100
        TOL   = 1.0E-9
        IFAIL = 111
C
        CALL MGD1V(LEVELS, NXC, NYC, NXF, NYF, NM,
     +             A, V, RHS, VB, VS, WORK, RESNO,
     +             IOUT, ISTART, IPREP, MAXIT, TOL, IFAIL)
      END IF
C
      STOP
      END
```

```
      SUBROUTINE MATRHS(A, RHS, NXF, NYF, NOUT)
      REAL     A(NXF, NYF, 7), RHS(NXF, NYF)
      INTEGER NXF, NYF, NOUT
C-----------------------------------------------------------------------
C     MATRHS is a subroutine which fills the matrix and the right-hand
C     side. It is part of the example program.
C
C     The example is the Poisson equation on the unit square with
C     Dirichlet boundary conditions and the exact solution is:
C     X * (XSIZE - X) + Y * (YSIZE - Y).
C     In this example the boundary conditions are eliminated.
C-----------------------------------------------------------------------
      REAL XSIZE, YSIZE, XH, YH, X, Y, B
      REAL ZERO, ONE, TWO, FOUR
      DATA ZERO, ONE, TWO, FOUR /0.0E0, 1.0E0, 2.0E0, 4.0E0/
      XSIZE=ONE
      YSIZE=ONE
      XH=XSIZE/REAL(NXF+1)
      YH=YSIZE/REAL(NYF+1)
C     /(...+1) Because of elimination of boundary conditions
      WRITE(NOUT, 99991) XSIZE, YSIZE, XH, YH
99991 FORMAT(' POISSON PROBLEM:'//' XSIZE  = ', 1PE13.6/' YSIZE  = ',
     +       E13.6/' XH, YH = ', 2E13.6)
      XYH4=FOUR*XH*YH
      XY=XH/YH
      YX=YH/XH
C
C     Initial filling of the matrix and the right-hand side neglecting
C     the boundaries.
C
C     DO 10 J=1,NYF
C       DO 10 I=1,NXF
C         A(I,J,1)=-XY
C         A(I,J,2)=ZERO
C         A(I,J,3)=-YX
C         A(I,J,4)=TWO*(YX+XY)
C         A(I,J,5)=-YX
C         A(I,J,6)=ZERO
C         A(I,J,7)=-XY
C         RHS(I,J)=XYH4
C 10  CONTINUE
C
C     In order to obtain maximum vector-performance we reformulate
C     loop 10 keeping the following points in mind:
C
C       . the nested DO-loop 10 is collapsible (by making use of
C         explicit over-indexing)
C       . we have to satisfy the CYBER 205 restriction on the
C         iterative loop count
```

```
C          . by not taking one single outer DO-loop: DO 10 KK=1,NF,65535
C            for all inner DO-loops 11 up to and including 18 we possibly
C            avoid a few page-faults
C
      NF=NXF*NYF
C
      DO 11 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 11 K=KK,KKE
          A(K,1,1)=-XY
   11 CONTINUE
      DO 12 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 12 K=KK,KKE
          A(K,1,2)=ZERO
   12 CONTINUE
      DO 13 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 13 K=KK,KKE
          A(K,1,3)=-YX
   13 CONTINUE
      DO 14 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 14 K=KK,KKE
          A(K,1,4)=TWO*(YX+XY)
   14 CONTINUE
      DO 15 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 15 K=KK,KKE
          A(K,1,5)=-YX
   15 CONTINUE
      DO 16 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 16 K=KK,KKE
          A(K,1,6)=ZERO
   16 CONTINUE
      DO 17 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 17 K=KK,KKE
          A(K,1,7)=-XY
   17 CONTINUE
      DO 18 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 18 K=KK,KKE
          RHS(K,1)=XYH4
   18 CONTINUE
```

```
C
C     Correction for the Dirichlet boundary conditions corresponding
C     to the exact solution, X * (XSIZE - X) + Y * (YSIZE - Y)
C
C     Note, that after this correction-process all parts of the
C     difference-molecules outside the domain are initialized to zero!
C
C-----------------------------------------------------------------------
C     Lower boundary
C-----------------------------------------------------------------------
      X=ZERO
      DO 20 I=1,NXF
        X=X+XH
        B=X*(XSIZE-X)
        RHS(I,1)=RHS(I,1)-A(I,1,1)*B
        A(I,1,1)=ZERO
   20 CONTINUE
C-----------------------------------------------------------------------
C     Left and right-hand boundary
C-----------------------------------------------------------------------
      Y=ZERO
      DO 30 J=1,NYF
        Y=Y+YH
        B=Y*(YSIZE-Y)
        RHS(1,J)=RHS(1,J)-A(1,J,3)*B
        A(1,J,3)=ZERO
        RHS(NXF,J)=RHS(NXF,J)-A(NXF,J,5)*B
        A(NXF,J,5)=ZERO
   30 CONTINUE
C-----------------------------------------------------------------------
C     Upper boundary
C-----------------------------------------------------------------------
      X=ZERO
      DO 40 I=1,NXF
        X=X+XH
        B=X*(XSIZE-X)
        RHS(I,NYF)=RHS(I,NYF)-A(I,NYF,7)*B
        A(I,NYF,7)=ZERO
   40 CONTINUE
      RETURN
      END
```

13.2. Program data

None.

13.3. Program results

```
MGD1V EXAMPLE PROGRAM RESULTS.
POISSON PROBLEM:

XSIZE  =  1.000000E+00
YSIZE  =  1.000000E+00
XH, YH =  7.692308E-03 7.692308E-03

MULTIGRID PROGRAM MGD1V, VERSION 10 OCTOBER 1986

LEVELS     NXC     NYC      NXF      NYF        NM
     6       5       5      129      129     22350
 MAXIT          TOL
     3          0.000000E+00
                IOUT  ISTART  IPREP  IFAIL
     1  0  0  2  1        0      0    111

L2-NORM OF INITIAL RESIDUAL=   0.417E+01

MGD1V, ITERATION NUMBER =       1
L2-NORM OF RESIDUAL     =   0.145E-02
CURRENT REDUCTION FACTOR=   0.347E-03
AVERAGE REDUCTION FACTOR=   0.347E-03

MGD1V, ITERATION NUMBER =       2
L2-NORM OF RESIDUAL     =   0.869E-04
CURRENT REDUCTION FACTOR=   0.601E-01
AVERAGE REDUCTION FACTOR=   0.457E-02

MGD1V, ITERATION NUMBER =       3
L2-NORM OF RESIDUAL     =   0.599E-05
CURRENT REDUCTION FACTOR=   0.689E-01
AVERAGE REDUCTION FACTOR=   0.113E-01

MAXIT ITERATIONS PERFORMED WITHOUT REACHING TOL.

GOOD CONVERGENCE RATE, SO V AND VB ARE VALUABLE,
AT THIS POINT ONE MAY RESTART MGD1V WITH ISTART=2 AND IPREP=1

ERROR DETECTED BY NUMVEC LIBRARY ROUTINE MGD1V    - IFAIL =  8


CP-TIMES USED (SEC.)

GALERKIN         0.053

DECOMPOSE        0.053

MG-CYCLES        0.116
```

```
MULTIGRID PROGRAM MGD1V, VERSION 10 OCTOBER 1986

LEVELS     NXC     NYC      NXF      NYF        NM
     6       5       5      129      129     22350
 MAXIT         TOL
   100         0.100000E-08
                 IOUT  ISTART  IPREP  IFAIL
     1  0  0   2  1        2       1    111

L2-NORM OF INITIAL RESIDUAL=   0.599E-05

MGD1V, ITERATION NUMBER =       1
L2-NORM OF RESIDUAL     =   0.391E-06
CURRENT REDUCTION FACTOR=   0.653E-01
AVERAGE REDUCTION FACTOR=   0.653E-01

MGD1V, ITERATION NUMBER =       2
L2-NORM OF RESIDUAL     =   0.266E-07
CURRENT REDUCTION FACTOR=   0.681E-01
AVERAGE REDUCTION FACTOR=   0.667E-01

MGD1V, ITERATION NUMBER =       3
L2-NORM OF RESIDUAL     =   0.176E-08
CURRENT REDUCTION FACTOR=   0.660E-01
AVERAGE REDUCTION FACTOR=   0.664E-01

MGD1V, ITERATION NUMBER =       4
L2-NORM OF RESIDUAL     =   0.121E-09
CURRENT REDUCTION FACTOR=   0.686E-01
AVERAGE REDUCTION FACTOR=   0.670E-01


CP-TIMES USED (SEC.)

GALERKIN         0.000

DECOMPOSE        0.000

MG-CYCLES        0.154
```

MGD5V - NUMVEC FORTRAN Library Routine Document

1. Purpose

MGD5V solves 7-diagonal linear systems, that arise from 7-point discretizations of elliptic PDEs on a rectangle, using a multigrid technique.

2. Specification

```
      SUBROUTINE MGD5V(LEVELS, NXC, NYC, NXF, NYF, NM,
     +                 A, U, RHS, UB, WORK, LDU, RESNO,
     +                 IOUT, ISTART, IPREP, MAXIT, TOL, IFAIL)
C     INTEGER LEVELS, NXC, NYC, NXF, NYF, NM, IOUT(5), ISTART, IPREP,
C    +        MAXIT, IFAIL
C     REAL    A(NM*7), U(NM), RHS(NM), UB(NM), WORK(NXF*9),
C    +        LDU(NM*3), RESNO, TOL
```

3. Description

MGD5V solves a 7-diagonal linear system, that arises from a 7-point discretization of an elliptic PDE on a rectangle. The system is written in the form

$$A \times U = RHS, \tag{1}$$

where A and RHS are the user-supplied matrix and right-hand side respectively. Note, that only the 7 non-zero diagonals are stored. The approximate solution is found by means of the multigrid correction storage algorithm

with: smoothing by ILLU relaxation, symmetric 7-point prolongation and restriction, Galerkin approximation of coarse-grid matrices, also on the coarsest grid ILLU relaxation is used instead of an exact solver.

The relation between a coarse grid and a fine grid (two subsequent levels) is the following:

let $C = \{ (x_i,y_j) | x_i = (i-1)h+o_1,\ i = 1\ (1)\ NXC;$

$$y_j = (j-1)h+o_2,\ j = 1\ (1)\ NYC\ ) \}$$

be the coarse grid, then the next fine grid is

$$F = \{ (x_i, y_j) | x_i = (i-1)\frac{h}{2}+o_1,\ i = 1\ (1)\ (2\ NXC-1);$$

$$y_j = (j-1)\frac{h}{2}+o_2,\ j = 1\ (1)\ (2\ NYC-1)\ \}$$

The user needs no knowledge of the multigrid method, it suffices to comply to the specification of the parameters.

The user supplies an absolute residual tolerance in the form of a bound of its l_2-norm. The user may also supply an initial approximation (alternatively, the zero solution is used as initial approximation).

4. References

[1] Hemker, P.W., de Zeeuw, P.M., Some implementations of multigrid linear system solvers. Lectures held at the University of Bristol, England, sept.1983. In: D.J. Paddon and Holstein (eds.), Multigrid methods for integral and differential equations. The Institute of Mathematics and its Applications Conference Series, Oxford University Press, 1985, pp.85-116.

[2] Meijerink, J.A., Iterative methods for the solution of linear equations based on incomplete factorization of the matrix. Publication 643, Shell Research B.V., Kon. Shell Expl. and Prod. Lab., Rijswijk, The Netherlands, July 1983.

[3] Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[4] Sonneveld, P., Wesseling, P., de Zeeuw, P.M., Multigrid and conjugate gradient methods as convergence acceleration techniques. Lectures held at the University of Bristol, England, sept.1983. In: D.J. Paddon and H. Holstein (eds.), Multigrid methods for integral and differential equations. The Institute of Mathematics and its Applications Conference Series, Oxford University Press, 1985, pp.117-167.

5. Parameters

LEVELS - INTEGER.
On entry, LEVELS must specify the number of levels in the multigrid method.
$1 \leqslant$ LEVELS $\leqslant 12$.
Unchanged on exit.

NXC - INTEGER.
NYC - INTEGER.
On entry, NXC and NYC must specify the number of vertical and horizontal grid-lines respectively, on the coarsest grid.
NXC, NYC $\geqslant 3$.
Unchanged on exit.

NXF - INTEGER.
NYF - INTEGER.
On entry, NXF and NYF must specify the number of vertical and horizontal grid-lines respectively, on the finest grid.
NXF $\leqslant 65535$.
Unchanged on exit.

NM - INTEGER.
On entry, NM must specify the number of grid-points on all grids together.
Unchanged on exit.

Note that the following relations should hold:

$$\text{NXF} = (\text{NXC}-1) \times 2^{\text{LEVELS}-1} + 1$$

$$\text{NYF} = (\text{NYC}-1) \times 2^{\text{LEVELS}-1} + 1$$

$$\text{NM} = \sum_{\text{L}=1}^{\text{LEVELS}} ((\text{NXC}-1) \times 2^{\text{L}-1}+1)((\text{NYC}-1) \times 2^{\text{L}-1}+1)$$

The program checks the consistency of these data

Examples:

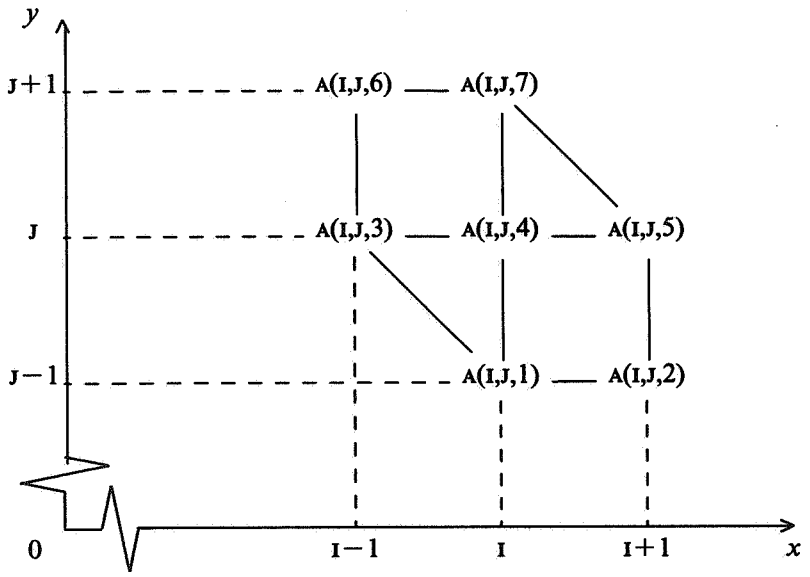
LEVELS	2	3	4	5	6	7
NXC	5	5	5	5	5	5
NYC	5	5	5	5	5	5
NXF	9	17	33	65	129	257
NYF	9	17	33	65	129	257
NM	106	395	1484	5709	22350	88399

LEVELS	2	3	4	5	6	7
NXC	5	5	5	5	5	5
NYC	3	3	3	3	3	3
NXF	9	17	33	65	129	257
NYF	5	9	17	33	65	129
NM	60	213	774	2919	11304	44457

A - REAL array of DIMENSION at least (NM*7)
Before entry, the first NXF*NYF*7 elements of A must contain the 7-diagonal matrix of the linear system (1).

The easiest way for the user to fill the matrix A is writing a subroutine where the actual argument A is handled as an adjustable array with dimensions (NXF, NYF, 7).

The 7-point difference molecule at the point with subscripts (I, J) is positioned in the x,y -plane as follows:



Important: the user has to provide the matrix A only on the finest grid. The coarse-grid matrices are computed internally by the routine by means of Galerkin approximation.

Important: the user has to take care that parts of the molecules outside the domain are initialized to zero, otherwise wrong results are produced.

Remark: problems with topologically non-rectangular regions can also be solved using this routine by surrounding the region by a circumscribing topological rectangle. Let (ie,je) denotate a nodal point external to the region of interest, then set A(ie,je,k)=0 for k=1,2,3,5,6,7 and A(ie,je,4)=eps with eps small, e.g. 10^{-9} (further set RHS(ie,je)=0).

The first NXF*NYF*7 elements of A are unchanged on exit.

U - REAL array of DIMENSION at least (NM)
Before entry, but only if the routine is entered with ISTART = 1 or with ISTART = 2, the first NXF*NYF elements of U must contain an initial estimate for the iterative process. If ISTART = 0 no initialization of U is necessary (a zero initial estimate is assumed).

The easiest way for the user to fill the initial estimate is writing a subroutine where the actual argument U is handled as an adjustable array with dimensions (NXF, NYF).

On successful exit, the first NXF*NYF elements of U contain the (approximate) numerical solution.

RHS - REAL array of DIMENSION at least (NM)
Before entry, the first NXF*NYF elements of RHS must contain the right-hand side of the linear system (1), the other elements are used as workspace.

The easiest way for the user to fill the right-hand side is writing a subroutine where the actual argument RHS is handled as an adjustable array with dimensions (NXF, NYF).

Important: The user has to provide the right-hand side of the discretized equation only on the finest grid.

The first NXF*NYF elements of RHS are unchanged on exit.

UB - REAL array of DIMENSION at least (NM)
Before entry, but only if the routine is entered with ISTART = 2, the first NXF*NYF elements of UB must contain the residual RHS−A*U of the initial estimate U.
On successful exit, the first NXF*NYF elements of UB contain the residual of the approximate numerical solution U.

WORK - REAL array of DIMENSION at least (NXF*9).
Used as workspace.

LDU - REAL array of DIMENSION at least (NM*3).
Before entry, if IPREP = 0, the elements of LDU need not to be initialized.
Before entry, if IPREP = 1, then the first NM*3 elements of LDU must contain the unaltered contents of LDU after the last previous call of MGD5V. (If IPREP = 1 those contents are assumed to be the ILLU-decompositions of the matrix of (1) and its coarse-grid approximations.)
On exit, if IPREP = 0, the first NM*3 elements of LDU contain the ILLU-decompositions of the userprovided matrix A and its coarse-grid approximations.
If IPREP = 1 then the contents of LDU are unchanged on exit.

RESNO - REAL.
On exit, RESNO contains the l_2 -norm of the residual.

IOUT - INTEGER array of DIMENSION at least (5)
IOUT governs the amount of information about the solution process delivered to the user. Smaller IOUT-values mean less output. The user may select the unit-number on which this output is to appear by a call of X04ABF. Before entry, the first 5 elements of IOUT must contain one of the following values:

IOUT(1) ≥ 1 confirmation of input data
≤ 0 none

IOUT(2) ≥ 2 matrices on all levels and right-hand side on highest level
$= 1$ matrix and right-hand side on highest level
≤ 0 none

IOUT(3) ≥ 2 matrix-decompositions on all levels
$= 1$ matrix-decomposition on highest level
≤ 0 none

IOUT(4) ≥ 4 norms of residuals, reduction factors, final solution, final residual
$= 3$ norms of residuals, reduction factors, final residual
$= 2$ norms of residuals, reduction factors (i.e. monitoring the convergence-behaviour)
$= 1$ final norm of residual, number of iterations
≤ 0 none

IOUT(5) ≥ 1 the time spent in various subroutines
≤ 0 none

The contents of IOUT are unchanged on exit.

ISTART - INTEGER.
On entry, ISTART must be set to 0, 1 or 2.

ISTART $= 0$ means that the initial estimate is zero, initialization of U is not necessary;
$= 1$ means that the user provides an initial estimate of the solution in U;
$= 2$ means that the user provides an initial estimate of the solution in U and the residual of U in UB.

Unchanged on exit.

IPREP - INTEGER.
On entry, IPREP must be set to 0 or 1.

IPREP $= 1$ if MGD5V has been called before and the information in LDU after that call has not been changed. (This is useful if the user wants to refine the numerical solution by re-entering the routine or if the user desires to solve a sequence of problems with identical matrices but different right-hand sides, this option then prevents a new setup of coarse-grid matrices and decompositions which have already been computed.)
$= 0$ if MGD5V has not been called before or if after a call of MGD5V the information in LDU has been altered.

Unchanged on exit.

MAXIT - INTEGER.

TOL - REAL.

On entry, MAXIT must specify the maximum number of allowed multigrid iterations and TOL must specify the tolerance desired by the user, where TOL is a bound of the l_2 -norm of the residual.

If during the multigrid process either MAXIT iterations have been performed or the tolerance has been reached, multigrid cycling is stopped.

For MAXIT a value of about 10 is adequate for many problems, clearly TOL should not be less than a reasonable multiple of NXF*NYF times the machine accuracy.

Both MAXIT and TOL are unchanged on exit.

IFAIL - INTEGER.

For this routine, the normal use of IFAIL is extended to control the printing of error and warning messages as well as specifying hard or soft failure (see [3], chapter P01). On entry IFAIL must be set to a value with the decimal expansion cba, where each of the decimal digits c, b and a must have the value 0 or 1.

a = 0 specifies hard failure, otherwise soft failure;
b = 0 suppresses error messages, otherwise error messages will be printed (see section 6);
c = 0 suppresses warning messages, otherwise warning messages will be printed (see section 6).

For users not familiar with this parameter the recommended value is 110 (i.e. hard failure with all messages printed).

Unless the routine detects an error (see section 6), IFAIL contains 0 on exit.

6. Error indicators and warnings

Errors detected by the routine:-

For some errors the routine outputs an explanatory message on the current error message unit (see routine X04AAF), unless suppressed by the value of IFAIL on entry.

IFAIL = 1
On entry, LEVELS is out of bounds.

IFAIL = 2
On entry, NXC or NYC is less than 3.

IFAIL = 3
On entry, no consistency among LEVELS, NXC and NXF.

IFAIL = 4
On entry, no consistency among LEVELS, NYC and NYF.

IFAIL = 5
On entry, NM is wrong.

IFAIL = 6
divergence, maybe due to this routine or to user discretization.

IFAIL = 7
poor convergence, maybe due to this routine or to user discretization or to reaching roundoff level.

IFAIL = 8
convergence, but MAXIT iterations performed without reaching TOL. This could be due to a value of MAXIT which is too small or to reaching roundoff level. The user is advised to monitor the convergence behaviour by choosing IOUT(4) $\geq$ 2

7. Auxiliary routines

This routine calls the following NUMVEC library routines:
BLOCKS, COPY, CYCLES, DECOMP, FOURTH, GALERK, ILLUDC, OUTMAT, OUTVEC, PREPAR, PROLON, P01AAF, RAP, RESIDU, RESTRI, SMOOTH, SOLVE, TIMING, TRIDEC, VL2NOR, X04AAF, X04ABF, and ZERO.

8. Timing

The time taken per multigriditeration is proportional to NXF $\times$ NYF.
However, the vector performance (mflop-rate) increases with increasing NXF, therefore it is advised to take the longest side of the rectangular grid along the x-axis assuming that for many problems there is not much orientation-sensitivity in the algorithm used.
According to multigrid theory it is plausible that the convergence rate is independent of the chosen meshwidth.

9. Storage

Internally declared arrays contain 3 REAL and 50 INTEGER elements.

10. Accuracy

If the process converges, and MAXIT is large enough, the l_2 -norm of the residual becomes less than TOL.

11. Further comments

Labelled common blocks CPU and POI are used by the routine and must therefore be avoided by users.
The user is strongly recommended to set IFAIL to obtain self-explanatory error- and advisory-messages. The user may select the unit numbers on which this output is to appear by calls of X04AAF (for error messages) or X04ABF (for advisory messages) - see section 13 for an example. Otherwise the default unit numbers will be used.

11.1. Vectorization information

This program is entirely written in ANSI FORTRAN 77 and auto-vectorizes both on the CYBER 205 and the CRAY-1. On the CYBER 205 the compiler options OPTimize and UNSafe are recommended. The UNSafe option can be used safely, because throughout the whole program all vectorizable DO-loops with possibly more than 65535 iterations are subdivided in loops which do not violate the constraint on the loop count.

12. Keywords

Elliptic PDEs,
Galerkin approximation,
Multigrid methods,
Sparse linear systems,

ILLU relaxation.

13. Example

We solve the Poisson equation on the unit square with Dirichlet boundary conditions and the right-hand side constructed according to the exact solution $x\ (1-x) + y\ (1-y)$. In this example the boundary conditions are eliminated.

First we try 3 iterations with a zero initial approximation and if no divergence is found (using the soft fail option) we try to refine the solution until the residual norm becomes lesser than 10^{-9} using the former approximation.

Note the calls to X04AAF and X04ABF prior to the call of MGD5V.

13.1. Program text

```
      PROGRAM EXAMPL
C
      PARAMETER( LEVELS = 6, NXC =   5, NYC =   5,
     +                       NXF = 129, NYF = 129, NM = 22350)
C
      REAL A, V, RHS, VB, WORK, LDU
      COMMON /ARRAYS/ A(NM*7), V(NM), RHS(NM), VB(NM), WORK(NXF*9),
     +                LDU(NM*3)
C     These arrays are declared in a labelled common block in order to
C     allow large-page mapping on the CYBER205.
C
      INTEGER IOUT(5)
C
      DATA IOUT/1,0,0,2,1/
C
      DATA NOUT/6/
C
      OPEN(UNIT=NOUT,FILE='OUTPUT')
      CALL X04AAF(1,NOUT)
      CALL X04ABF(1,NOUT)
C
      WRITE(NOUT,999)
  999 FORMAT('1 MGD5V EXAMPLE PROGRAM RESULTS.')
C
C     Problem set up.
C     subroutine MATRHS is an example of a subroutine which fills the
C     matrix and the right-hand side.
C
      CALL MATRHS(A,RHS,NXF,NYF,NOUT)
C
C     Approximate the solution of the linear system by performing 3
C     MGD5V iterations, using the soft fail option.
C
      ISTART= 0
      IPREP = 0
      MAXIT = 3
```

```
      TOL   = 0.0
      IFAIL = 111
C
      CALL MGD5V(LEVELS, NXC, NYC, NXF, NYF, NM,
     +           A, V, RHS, VB, WORK, LDU, RESNO,
     +           IOUT, ISTART, IPREP, MAXIT, TOL, IFAIL)
C
C     Possible refinement until residual norm .LT. 1.0E-9
C
      IF( IFAIL.GT.0 .AND. IFAIL.GE.7 ) THEN
        ISTART= 2
        IPREP = 1
        MAXIT = 100
        TOL   = 1.0E-9
        IFAIL = 111
C
        CALL MGD5V(LEVELS, NXC, NYC, NXF, NYF, NM,
     +             A, V, RHS, VB, WORK, LDU, RESNO,
     +             IOUT, ISTART, IPREP, MAXIT, TOL, IFAIL)
      END IF
C
      STOP
      END

      SUBROUTINE MATRHS(A, RHS, NXF, NYF, NOUT)
      REAL    A(NXF, NYF, 7), RHS(NXF, NYF)
      INTEGER NXF, NYF, NOUT
C-----------------------------------------------------------------------
C     MATRHS is a subroutine which fills the matrix and the right-hand
C     side. It is part of the example program.
C
C     The example is the Poisson equation on the unit square with
C     Dirichlet boundary conditions and the exact solution is:
C     X * (XSIZE - X) + Y * (YSIZE - Y).
C     In this example the boundary conditions are eliminated.
C-----------------------------------------------------------------------
      REAL XSIZE, YSIZE, XH, YH, X, Y, B
      REAL ZERO, ONE, TWO, FOUR
      DATA ZERO, ONE, TWO, FOUR /0.0E0, 1.0E0, 2.0E0, 4.0E0/
      XSIZE=ONE
      YSIZE=ONE
      XH=XSIZE/REAL(NXF+1)
      YH=YSIZE/REAL(NYF+1)
C     /(...+1) Because of elimination of boundary conditions
      WRITE(NOUT, 99991) XSIZE, YSIZE, XH, YH
99991 FORMAT(' POISSON PROBLEM:'//' XSIZE = ', 1PE13.6/' YSIZE = ',
     +       E13.6/' XH, YH = ', 2E13.6)
      XYH4=FOUR*XH*YH
      XY=XH/YH
```

```
      YX=YH/XH
C
C     Initial filling of the matrix and the right-hand side neglecting
C     the boundaries.
C
C     DO 10 J=1,NYF
C       DO 10 I=1,NXF
C         A(I,J,1)=-XY
C         A(I,J,2)=ZERO
C         A(I,J,3)=-YX
C         A(I,J,4)=TWO*(YX+XY)
C         A(I,J,5)=-YX
C         A(I,J,6)=ZERO
C         A(I,J,7)=-XY
C         RHS(I,J)=XYH4
C 10  CONTINUE
C
C     In order to obtain maximum vector-performance we reformulate
C     loop 10 keeping the following points in mind:
C
C       . the nested DO-loop 10 is collapsible (by making use of
C         explicit over-indexing)
C       . we have to satisfy the CYBER 205 restriction on the
C         iterative loop count
C       . by not taking one single outer DO-loop: DO 10 KK=1,NF,65535
C         for all inner DO-loops 11 up to and including 18 we possibly
C         avoid a few page-faults
C
      NF=NXF*NYF
C
      DO 11 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 11 K=KK,KKE
          A(K,1,1)=-XY
   11 CONTINUE
      DO 12 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 12 K=KK,KKE
          A(K,1,2)=ZERO
   12 CONTINUE
      DO 13 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 13 K=KK,KKE
          A(K,1,3)=-YX
   13 CONTINUE
      DO 14 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 14 K=KK,KKE
          A(K,1,4)=TWO*(YX+XY)
```

```
   14 CONTINUE
      DO 15 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 15 K=KK,KKE
          A(K,1,5)=-YX
   15 CONTINUE
      DO 16 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 16 K=KK,KKE
          A(K,1,6)=ZERO
   16 CONTINUE
      DO 17 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 17 K=KK,KKE
          A(K,1,7)=-XY
   17 CONTINUE
      DO 18 KK=1,NF,65535
        KKE=(KK-1)+MIN0(65535,NF-(KK-1))
        DO 18 K=KK,KKE
          RHS(K,1)=XYH4
   18 CONTINUE
C
C     Correction for the Dirichlet boundary conditions corresponding
C     to the exact solution, X * (XSIZE - X) + Y * (YSIZE - Y)
C
C     Note, that after this correction-process all parts of the
C     difference-molecules outside the domain are initialized to zero!
C
C-----------------------------------------------------------------------
C     Lower boundary
C-----------------------------------------------------------------------
      X=ZERO
      DO 20 I=1,NXF
        X=X+XH
        B=X*(XSIZE-X)
        RHS(I,1)=RHS(I,1)-A(I,1,1)*B
        A(I,1,1)=ZERO
   20 CONTINUE
C-----------------------------------------------------------------------
C     Left and right-hand boundary
C-----------------------------------------------------------------------
      Y=ZERO
      DO 30 J=1,NYF
        Y=Y+YH
        B=Y*(YSIZE-Y)
        RHS(1,J)=RHS(1,J)-A(1,J,3)*B
        A(1,J,3)=ZERO
        RHS(NXF,J)=RHS(NXF,J)-A(NXF,J,5)*B
        A(NXF,J,5)=ZERO
```

```
   30 CONTINUE
C-------------------------------------------------------------------
C     Upper boundary
C-------------------------------------------------------------------
      X=ZERO
      DO 40 I=1,NXF
        X=X+XH
        B=X*(XSIZE-X)
        RHS(I,NYF)=RHS(I,NYF)-A(I,NYF,7)*B
        A(I,NYF,7)=ZERO
   40 CONTINUE
      RETURN
      END
```

13.2. Program data

None.

13.3. Program results

```
MGD5V EXAMPLE PROGRAM RESULTS.
POISSON PROBLEM:

XSIZE  =  1.000000E+00
YSIZE  =  1.000000E+00
XH, YH =  7.692308E-03 7.692308E-03

MULTIGRID PROGRAM MGD5V, VERSION  1 SEPT.    1986

LEVELS    NXC    NYC     NXF     NYF       NM
     6      5      5     129     129    22350
 MAXIT        TOL
     3        0.000000E+00
              IOUT  ISTART  IPREP  IFAIL
     1  0  0  2  1       0      0    111

L2-NORM OF INITIAL RESIDUAL=   0.417E+01

MGD5V, ITERATION NUMBER =       1
L2-NORM OF RESIDUAL     =   0.591E-03
CURRENT REDUCTION FACTOR=   0.142E-03
AVERAGE REDUCTION FACTOR=   0.142E-03

MGD5V, ITERATION NUMBER =       2
L2-NORM OF RESIDUAL     =   0.218E-04
CURRENT REDUCTION FACTOR=   0.369E-01
AVERAGE REDUCTION FACTOR=   0.229E-02
```

```
MGD5V, ITERATION NUMBER =       3
L2-NORM OF RESIDUAL     =   0.852E-06
CURRENT REDUCTION FACTOR=   0.391E-01
AVERAGE REDUCTION FACTOR=   0.589E-02

MAXIT ITERATIONS PERFORMED WITHOUT REACHING TOL.

GOOD CONVERGENCERATE, SO V AND VB ARE VALUABLE,
AT THIS POINT ONE MAY RESTART MGD5V WITH ISTART=2 AND IPREP=1

ERROR DETECTED BY NUMVEC LIBRARY ROUTINE MGD5V    - IFAIL =  8


CP-TIMES USED (SEC.)

GALERKIN        0.053

DECOMPOSE       0.092

MG-CYCLES       0.265



MULTIGRID PROGRAM MGD5V, VERSION  1 SEPT.    1986

LEVELS    NXC    NYC     NXF     NYF        NM
     6      5      5     129     129     22350
 MAXIT         TOL
   100         0.100000E-08
               IOUT  ISTART  IPREP  IFAIL
     1  0  0  2  1       2      1    111

L2-NORM OF INITIAL RESIDUAL=   0.852E-06

MGD5V, ITERATION NUMBER =       1
L2-NORM OF RESIDUAL     =   0.347E-07
CURRENT REDUCTION FACTOR=   0.408E-01
AVERAGE REDUCTION FACTOR=   0.408E-01

MGD5V, ITERATION NUMBER =       2
L2-NORM OF RESIDUAL     =   0.148E-08
CURRENT REDUCTION FACTOR=   0.426E-01
AVERAGE REDUCTION FACTOR=   0.417E-01

MGD5V, ITERATION NUMBER =       3
L2-NORM OF RESIDUAL     =   0.661E-10
CURRENT REDUCTION FACTOR=   0.446E-01
AVERAGE REDUCTION FACTOR=   0.426E-01
```

```
CP-TIMES USED (SEC.)

GALERKIN        0.000

DECOMPOSE       0.000

MG-CYCLES       0.264
```