NUMVEC FORTRAN Library manual

Chapter: Elliptic PDEs

Routine: MGZEB

W.M. Lioen
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract

The NUMVEC FORTRAN Library routine MGZEB is described.
MGZEB solves 7-diagonal linear systems, that arise from 7-point discretizations of elliptic PDEs on a rectangle, using a multigrid technique with zebra 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, zebra relaxation.

Note: The implementation is available in auto-vectorizable ANSI FORTRAN 77

Report NM-R8518
ISSN 0169-0388
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

MGZEB - Routine Document

1. Purpose

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

2. Specification

```
      SUBROUTINE MGZEB(NXF, NYF, NF, LEVELS, NXC, NYC, NM, A, U, RHS, DEC, US,
     +                 ISTART, MAXIT, TOL, P, S, Q, IOUT, RESNO, IFAIL)
C       REAL     A(NM*7), U(NM), RHS(NM), DEC(NM*2), US(NM), TOL, RESNO
C       INTEGER NXF, NYF, NF, LEVELS, NXC, NYC, NM,
C    + START, MAXIT, P, S, Q, IOUT(5), IFAIL
```

3. Description

MGZEB 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 even-odd y-zebra relaxation, symmetric 7-point prolongation and restriction, Galerkin approximation of coarse grid matrices, also on the coarsest grid relaxation is used as the solution process.

However, the user remains unaware of the underlying multigrid method.

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., 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., Wesseling, P. and Zeeuw, P.M. de, 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.

[3] Lioen, W.M. and Hemker P.W., Multigrid methods for elliptic PDEs, I. To appear in: Numerical Aspects of Vector- and Parallel Processors. (H.J.J. te Riele ed.) Procs. 1985 -1986 (CWI, UvA, THD) colloquium.

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

[5] Wesseling, P., A robust and efficient multigrid method. In: Multigrid methods (W. Hackbusch and U. Trottenberg eds.), pp. 614-630, Procs. Koln-porz, 1981. Springer LNM 960, Springer-verlag, 1982.

5. Parameters

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

NF - INTEGER.
On entry, NF must specify the number of grid-points plus the number of horizontal grid-lines on the finest grid.
Unchanged on exit.

LEVELS - INTEGER.
On entry, LEVELS must specify the number of levels in the multigrid method.
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.
Unchanged on exit.

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

Note that the following relations should hold:

$$\text{NXF, NYF, NXC, NYC} \geq 3$$

$$\text{NF} = (\text{NXF}+1)\times\text{NYF}$$

$$1 \leq \text{LEVELS} \leq \min\{\log_2(\text{NXF}-1), \log_2(\text{NYF}-1), 12\}$$

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

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

$$\text{NM} \geq \frac{4}{3}(\text{NXF}-1)(\text{NYF}-1) - \frac{1}{3}(\text{NXC}-1)(\text{NYC}-1) +$$

$$2[(\text{NXF}-1) + 2(\text{NYF}-1)] - [(\text{NXC}-1) + 2(\text{NYC}-1)] +$$

$$2\times\text{LEVELS}$$

For practical purposes: $\text{NM} \geq \frac{4}{3}(\text{NXF}+3)(\text{NYF}+1)$

The program checks the consistency of these data

Examples:

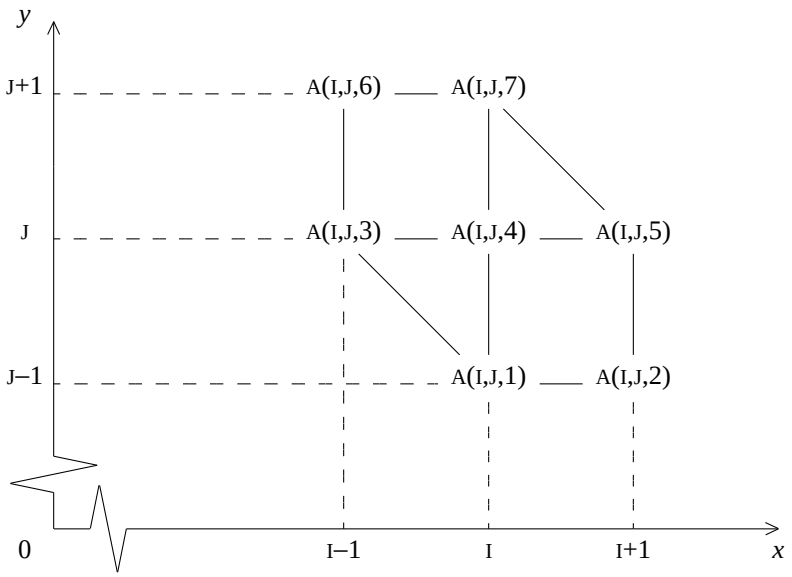
LEVELS	2	3	4	5	6	7	8
NXC	3	3	3	3	3	3	3
NYC	3	3	3	3	3	3	3
NXF	5	9	17	33	65	129	257
NYF	5	9	17	33	65	129	257
NF	30	90	306	1122	4290	16770	66306
NM	42	132	438	1560	5850	22620	88926

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
NF	90	306	1122	4290	16770	66306
NM	120	426	1548	5838	22608	88914

A - REAL array of DIMENSION at least (NM∗7)
Before entry, the first NXF∗NYF∗7 elements of A must contain the matrix corresponding to the finest grid. If the routine has to be re-entered with ISTART = 2, the first NM∗7 elements of A should remain unchanged, otherwise wrong results will be produced!

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 inside 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.

On exit, the first NM∗7 elements of A contain the original information of the user-provided matrix A plus its coarse grid approximations. However the information has in general become worthless to the user (except for the re-entry case), because on exit the information still is stored according to the data-structure internally used by the routine.
Hence, the contents of A are altered on exit.

U - REAL array of DIMENSION at least (NM)
Only if the routine is entered with ISTART = 1 or re-entered 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 equation.
If the routine has to be re-entered with ISTART = 2, the first (NXF+1)∗NYF elements of RHS have to be kept unchanged, otherwise wrong results will be produced!

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.

On exit, the first (NXF+1)∗NYF elements of RHS contain the original information of the user-provided RHS. However, the information has in general become worthless to the user (except for the re-entry case), because on exit the information still is stored according to the data-structure internally used by the routine.
Hence, the contents of RHS are altered on exit.

DEC - REAL array of DIMENSION at least (NM∗2).
Used as workspace.
Only if the routine has to be re-entered with ISTART = 2, the first NM∗2 elements of DEC have to be kept unchanged, otherwise wrong results will be produced!
On exit, the first NM∗2 elements of DEC contain the line LU-decompositions of A on all levels, used for the zebra-relaxation. However, the information is in general worthless to the user (except for the re-entry case), because on exit the information still is stored according to the data-structure internally used by the routine.

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

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 routine is re-entered after a former call of the routine
(e.g. with another initial estimate or to refine the numerical solution found).
In this case:

NXF, NYF, NF, LEVELS, NXC, NYC, NM,
the first NM∗7 elements of A,
the first NM∗2 elements of DEC and

the first (NXF+1)∗NYF elements of RHS

should be kept unchanged, otherwise wrong results will be produced!

Unchanged on exit.

MAXIT - REAL.

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.

Unchanged on exit.

P - INTEGER.

Q - INTEGER.

S - INTEGER.

These parameters allow the user to influence the multigrid strategy. Most probably a good choice is P = 0, Q = S = 1. On entry, P specifies the number of relaxations before the coarse grid correction; S specifies the number of multigrid iterations in the coarse grid correction; and Q specifies the number of relaxations after the coarse grid correction. S defines the cycling structure: V-cycles are obtained by S = 1 and W-cycles by S = 2. By setting P = 0 and Q = S = 1 (the recommended values that can also be found in [1] and [5]) we obtain the so-called sawtooth multigrid cycle, which is also used by MGD1 and MGD5.

The values of P, Q and S are unchanged on exit.

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)	≥	3	norms of residuals, reduction factors, final solution, final residual
	=	2	norms of residuals, reduction factors, final solution
	=	1	norms of residuals, reduction factors (i.e. monitoring the convergence-behaviour the same effect is obtained by setting IFAIL ≥ 100 on entry)
	≤	0	none
IOUT(5)	≥	1	the time spent in various subroutines
	≤	0	none

The contents of IOUT are unchanged on exit.

RESNO - REAL.

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

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 [4], 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
one or more of the input parameters have been incorrectly set.

IFAIL = 2
convergence, but MAXIT iterations performed without reaching TOL. This could be due to a poor convergence-behaviour for this specific problem or to reaching roundoff level. In any case, the user is advised to monitor the convergence-behaviour by choosing IFAIL or IOUT(4) appropriately.

IFAIL = 3
the routine fails to converge. This routine is probably not suitable for this specific problem.

7. Auxiliary routines

This routine calls the following NUMVEC library routines:
ADPROL, ARSPLT, CHANGE, CYCLES, DECOMP, EO2NOR, EORESI, EORSTR, EOZBRA, GALERK, INZBLK, MATRHS, OUTDEC, OUTEOR, OUTMAT, OUTVEC, P01AAF, PROLON, RAP, RESIDU, RESTRI, TIMING, VCCOMB, VCSPLT, VL2NOR, X04AAF, X04ABF and YLDEC.

8. Timing

In scalar-mode the timing (for S = 1) is proportional to NM and approximately proportional to (NXF+3)×(NYF+1). In vector-mode the timing becomes better with increasing NM because of increasing vector-lengths.
If the only concern would be an efficient vector-performance (mflop-rates) then it is advised to take the longest side of the rectangular grid along the x-axis.
If on the other hand one is also interested in a good numerical-performance (convergence-rates) then one has to take full account of the direction-sensitivity of the zebra-relaxation used. (viz. generally y-zebra relaxation is more efficient if a stronger connection exists between points in the y-direction than in the x-direction).
It is obvious, that both requirements can be in conflict with each other and that in some cases the latter requirement is more important in order to obtain a much better overall-performance (efficiency)!

9. Storage

Internally declared arrays contain 9 REAL and 74 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-messages, and also monitoring information about the convergence-behaviour. The user may select the unit numbers on which this output is to appear by calls of X04AAF (for error messages) or X04ABF (for monitoring information) - 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. On the CYBER 205 the compiler-options OPTimize and UNSafe will suffice.

In order to obtain a good (vector-)performance:

1. The data-structure is adapted internally to avoid the CYBER 205 stride problem during the most costly part of a multigrid iteration step: the (zebra-)relaxation. This is done by re-storing the problem during the preparational phase on entry; performing the multigrid cycling phase; and re-storing the numerical solution found according to the normal ordering of array-elements on exit.
2. The (line-wise) recurrencies which occur during the computation of the decompositions and the solution of the tri-diagonal systems, that arise from the zebra-relaxation, are solved simultaneously.
3. The (vector-) division during the zebra-relaxation is avoided by storage of the reciprocal value during the decomposition.
4. Wherever possible, the fact that the residual is zero on half the number of lines after a zebra-relaxation is performed, is exploited (this is possible during the computation of the residual itself, the norm of this special residual and the restriction of this special residual).
5. The number of page-faults is minimized.
6. Wherever possible nested DO-loops are collapsed, (which is not always trivial, think of the matrix-vector multiplications during the residual computation, or even better during the right-hand side computation of the tri-diagonal systems arising from the zebra-relaxation).

In order to conform with ANSI FORTRAN 77:

7. Nested DO-loop collapsing (point 6.) is done by explicit making use of over-indexing.

In order to survive large problems on the CYBER 205:

8. The restriction on the iterative loop count (65535) is taken care of.

In order to fool the (CYBER 205) FORTRAN 200 compiler:

9. Possible recurrencies are taken care of by implicit equivalencing: the same actual argument is passed to two different dummy arguments.
   This possible recurrency occurs when the first index exceeds the maximum rowindex in the actual declaration, which is allowed within FORTRAN and therefore suspected by the CYBER 205 FORTRAN 200 compiler.
   (The CRAY 1 FORTRAN compiler assumes implicitly that the programmer avoids such a situation. The CRAY 1 FORTRAN compiler is also provided with compiler-directives to force generation of vector-instructions or to turn the automatic vectorization off!)

Illustrations of points 5, 6, 7 and 8 are shown in the subroutine "MATRHS" which is part of the example program (see section 13.1).

12. Keywords

Elliptic PDEs,
Galerkin approximation,
Multigrid methods,
Sparse linear systems,
Zebra 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 10 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^{-10} using the former approximation.

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

13.1. Program text

```
      PROGRAM EXAMPL
      INTEGER P, S, Q, IOUT(5)
C
C **********************************************************************
C
C     Actual user provided DIMENSION statements. The arrays are declared
C     in a labelled COMMON block in order to allow large-page mapping
C     on the CYBER 205.
C
      INTEGER NM
      PARAMETER (NM=88926)
      REAL A, RHS, DEC, U, US
      COMMON /BIG/ A(NM*7), RHS(NM), DEC(NM*2), U(NM), US(NM)
C
C     User DATA statements
C
      DATA NXC, NYC, NXF, NYF, NF, LEVELS /3, 3, 257, 257, 66306, 8/
      DATA P, S, Q /0, 1, 1/
      DATA IOUT /1, 0, 0, 1, 1/
C
C **********************************************************************
C
      DATA NOUT /6/
C
      OPEN(UNIT=NOUT, FILE='OUTPUT')
      CALL X04AAF(1, NOUT)
      CALL X04ABF(1, NOUT)
C
      WRITE(NOUT, 99991)
99991 FORMAT('1MGZEB EXAMPLE PROGRAM RESULTS'///)
```

```
C
C       Problem set up
C
        CALL MATRHS(A, RHS, NXF, NYF, NOUT)
C
        CALL TIMING(CPB)
C
C       Approximate the solution of the linear system:
C       perform 10 MGZEB iterations, using the soft fail option.
C
        ISTART = 0
        MAXIT  = 10
        TOL    = 0.0E0
        IFAIL  = 111
C
        CALL MGZEB(NXF, NYF, NF, LEVELS, NXC, NYC, NM, A, U, RHS, DEC, US,
       +           ISTART, MAXIT, TOL, P, S, Q, IOUT, RESNO, IFAIL)
C
C       If no divergence was found, we try to refine the solution until
C       the residual norm < 1.0E-10.
C
        IF (IFAIL.EQ.2) THEN
          ISTART = 2
          MAXIT  = 50
          TOL    = 1.0E-10
          IFAIL  = 110
C
          CALL MGZEB(NXF, NYF, NF, LEVELS, NXC, NYC, NM, A, U, RHS, DEC, US,
       +             ISTART, MAXIT, TOL, P, S, Q, IOUT, RESNO, IFAIL)
        ENDIF
        CALL TIMING(CPE)
        WRITE(NOUT, 99992) CPE-CPB
99992   FORMAT(//' TOTAL CPSECS USED BY MGZEB:', F12.6)
C
C       Test the IFAIL-parameter before printing the final solution.
C
        IF (IFAIL.EQ.0) CALL PRTSOL(U, NXF, NYF, NOUT)
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/FLOAT(NXF+1)
      YH=YSIZE/FLOAT(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

            SUBROUTINE PRTSOL(V, NX, NY, NOUT)
            REAL      V(NX, NY)
            INTEGER NX, NY
C-----------------------------------------------------------------------
C           This routine prints the solution V and is only part
C           of the example program.
C-----------------------------------------------------------------------
            WRITE(NOUT,1)
          1 FORMAT(' 1THE NUMERICAL SOLUTION FOUND:'/)
            IF (NX.LE.33 .AND. NY.LE.33) THEN
              DO 10 J=NY,1,-1
                WRITE(NOUT,2) J,(V(I,J),I=1,NX)
          2     FORMAT(/' Y-INDEX=', I3, 1X, 1P10E12.3/(13X,1P10E12.3))
         10   CONTINUE
            ELSE
              WRITE(NOUT,3)
          3   FORMAT(' IS TOO LARGE TO FIT ON A FEW PAGES.')
            ENDIF
            RETURN
            END
```

13.2. Program data

None.

13.3. Program results

```
MGZEB EXAMPLE PROGRAM RESULTS



POISSON PROBLEM:

XSIZE  =  1.000000E+00
YSIZE  =  1.000000E+00
XH, YH =  3.875969E-03 3.875969E-03



MULTIGRID PROGRAM MGZEB, VERSION 24 MAY 1985

LEVELS    NXC    NYC     NXF     NYF        NF        NM
     8      3      3     257     257     66306     88926
 MAXIT        TOL              P  S  Q
    10        0.000000E+00     0  1  1
ISTART    IOUT            IFAIL
     0    1  0  0  1  1    111


L2-NORM OF INITIAL RESIDUAL= 5.87E+00

ITERATION NUMBER =  1
L2-NORM OF RESIDUAL      =  5.25E-03
REDUCTION FACTOR         =  8.95E-04
AVERAGE REDUCTION FACTOR =  8.95E-04

ITERATION NUMBER =  2
L2-NORM OF RESIDUAL      =  1.61E-03
REDUCTION FACTOR         =  3.06E-01
AVERAGE REDUCTION FACTOR =  1.65E-02

ITERATION NUMBER =  3
L2-NORM OF RESIDUAL      =  5.76E-04
REDUCTION FACTOR         =  3.58E-01
AVERAGE REDUCTION FACTOR =  4.61E-02

ITERATION NUMBER =  4
L2-NORM OF RESIDUAL      =  2.14E-04
REDUCTION FACTOR         =  3.72E-01
AVERAGE REDUCTION FACTOR =  7.77E-02

ITERATION NUMBER =  5
L2-NORM OF RESIDUAL      =  8.07E-05
REDUCTION FACTOR         =  3.77E-01
AVERAGE REDUCTION FACTOR =  1.07E-01
```

```
ITERATION NUMBER =  6
L2-NORM OF RESIDUAL      =  3.06E-05
REDUCTION FACTOR         =  3.79E-01
AVERAGE REDUCTION FACTOR =  1.32E-01

ITERATION NUMBER =  7
L2-NORM OF RESIDUAL      =  1.17E-05
REDUCTION FACTOR         =  3.81E-01
AVERAGE REDUCTION FACTOR =  1.53E-01

ITERATION NUMBER =  8
L2-NORM OF RESIDUAL      =  4.45E-06
REDUCTION FACTOR         =  3.81E-01
AVERAGE REDUCTION FACTOR =  1.72E-01

ITERATION NUMBER =  9
L2-NORM OF RESIDUAL      =  1.70E-06
REDUCTION FACTOR         =  3.82E-01
AVERAGE REDUCTION FACTOR =  1.88E-01

ITERATION NUMBER = 10
L2-NORM OF RESIDUAL      =  6.49E-07
REDUCTION FACTOR         =  3.82E-01
AVERAGE REDUCTION FACTOR =  2.02E-01

MAXIT ITERATIONS PERFORMED, WITHOUT REACHING TOL

ERROR DETECTED BY NUMVEC LIBRARY ROUTINE MGZEB    - IFAIL =  2


TIMINGS:

CHANGE            0.061340

GALERKIN          0.154401

DECOMPOSITION     0.022052

CYCLES (TOTAL)    0.599845

RELAXATION        0.289824

RESIDUAL          0.098288

PROLONGATION      0.101042

RESTRICTION       0.087903

NORM              0.008126


MULTIGRID PROGRAM MGZEB, VERSION 24 MAY 1985
```

```
LEVELS    NXC    NYC     NXF     NYF        NF        NM
     8      3      3     257     257     66306     88926
 MAXIT        TOL              P  S  Q
    50        1.000000E-10     0  1  1
ISTART    IOUT            IFAIL
     2    1  0  0  1  1    110


L2-NORM OF INITIAL RESIDUAL= 6.49E-07

ITERATION NUMBER =  1
L2-NORM OF RESIDUAL      =  2.48E-07
REDUCTION FACTOR         =  3.82E-01
AVERAGE REDUCTION FACTOR =  3.82E-01

ITERATION NUMBER =  2
L2-NORM OF RESIDUAL      =  9.50E-08
REDUCTION FACTOR         =  3.83E-01
AVERAGE REDUCTION FACTOR =  3.83E-01

ITERATION NUMBER =  3
L2-NORM OF RESIDUAL      =  3.64E-08
REDUCTION FACTOR         =  3.83E-01
AVERAGE REDUCTION FACTOR =  3.83E-01

ITERATION NUMBER =  4
L2-NORM OF RESIDUAL      =  1.39E-08
REDUCTION FACTOR         =  3.83E-01
AVERAGE REDUCTION FACTOR =  3.83E-01

ITERATION NUMBER =  5
L2-NORM OF RESIDUAL      =  5.34E-09
REDUCTION FACTOR         =  3.83E-01
AVERAGE REDUCTION FACTOR =  3.83E-01

ITERATION NUMBER =  6
L2-NORM OF RESIDUAL      =  2.05E-09
REDUCTION FACTOR         =  3.84E-01
AVERAGE REDUCTION FACTOR =  3.83E-01

ITERATION NUMBER =  7
L2-NORM OF RESIDUAL      =  7.88E-10
REDUCTION FACTOR         =  3.84E-01
AVERAGE REDUCTION FACTOR =  3.83E-01

ITERATION NUMBER =  8
L2-NORM OF RESIDUAL      =  3.03E-10
REDUCTION FACTOR         =  3.84E-01
AVERAGE REDUCTION FACTOR =  3.83E-01

ITERATION NUMBER =  9
L2-NORM OF RESIDUAL      =  1.17E-10
REDUCTION FACTOR         =  3.85E-01
```

```
AVERAGE REDUCTION FACTOR =  3.84E-01

ITERATION NUMBER = 10
L2-NORM OF RESIDUAL      =  4.49E-11
REDUCTION FACTOR         =  3.85E-01
AVERAGE REDUCTION FACTOR =  3.84E-01


TIMINGS:

CHANGE            0.007268

GALERKIN          0.000000

DECOMPOSITION     0.000000

CYCLES (TOTAL)    0.619218

RELAXATION        0.290756

RESIDUAL          0.118021

PROLONGATION      0.101041

RESTRICTION       0.087911

NORM              0.008118


TOTAL CPSECS USED BY MGZEB:    1.471958

THE NUMERICAL SOLUTION FOUND:

IS TOO LARGE TO FIT ON A FEW PAGES.
```