



1 SUMMARY

This module **solves a system of linear equations whose matrix is banded, symmetric and positive definite**. It uses block Cholesky factorization, taking advantage of any variation in bandwidth. It has an option for storing the matrix itself as well as its factorization. A secondary entry provides for further systems with the same matrix but different right-hand sides to be solved economically. The secondary entry can also be used to calculate residuals, needed for iterative refinement, provided the matrix itself has been stored. For very large systems or if restart facilities are desired, MA55 uses a direct-access file for the factors and a sequential file for the matrix.

The user must first specify all the row lengths. The matrix itself is supplied by blocks of rows using reverse communication. The blocking for input is chosen by the user and is independent of the blocking used for solution.

MC60 may be used to get a good ordering.

ATTRIBUTES — **Version:** 1.2.0. (26th Sept 2016) **Types:** Real (single, double). **Language:** Fortran 95, plus allocatable dummy arguments and allocatable components of derived types. **Calls:** `_GEMM`, `_SYRK`, `_TRSM`, `_TRSV`. **Helpful:** MC60. **Original date:** July 1999. **Origin:** J.K.Reid, Rutherford Appleton Laboratory.

2 HOW TO USE THE PACKAGE

Access to the package requires a USE statement whose simplest form is

Single precision version

```
USE HSL_MA55_SINGLE
```

Double precision version

```
USE HSL_MA55_DOUBLE
```

If it is required to use the two modules at the same time, the type `MA55_DATA` (Section 2.1) must be renamed on one of the USE statements. The procedures described in Section 2.2 are generic so do not need to be renamed.

For each problem, the user must declare a structure `D` of the derived type `MA55_DATA` defined by the module. This structure holds data for the problem and must be passed as an argument on every call to the procedures.

The following procedures are available to the user:

- (a) For each matrix, the initialization subroutine `MA55_START` must first be called.
- (b) `MA55_ROWS` must be called repeatedly for each block of rows to specify the lower-triangular part of the rows and (optionally) the right-hand side entries. The solution is completed on the call that includes the final row.
- (c) `MA55_SOLVE` uses the factors to rapidly solve further systems with the same matrix.
- (d) `MA55_RESID` uses the stored matrix to calculate residuals.
- (e) For each matrix, the subroutine `MA55_FINISH` should be called when all required calls have been made. This closes any files holding the matrix or its factors and optionally writes all the data in the structure `D` to a file.
- (e) For each matrix, the subroutine `MA55_RESTART` should be called to restore all the data written by `MA55_FINISH` to a file. This allows subsequent calls of `MA55_SOLVE` and `MA55_RESID` to be made without calling `MA55_START` and `MA55_ROWS`.

2.1 The derived data type

For each problem, the user is required to declare a structure of type `MA55_DATA` thus:

```
USE HSL_MA55_SINGLE
```

```

      :
      TYPE (MA55_DATA) D

```

or

```

      USE HSL_MA55_DOUBLE
      :
      TYPE (MA55_DATA) D

```

Components whose values are of interest to the user are documented in Section 2.2.7.

2.2 Procedures

2.2.2 To start a computation

For each problem, the user must first call the subroutine MA55_START to specify the row lengths, initialize a structure of type MA55_DATA, open a direct-access file to hold the factors, open a sequential file to hold the matrix (if desired), and specify a block size for solution. The structure must not be changed except through calls to MA55 procedures.

```

CALL MA55_START(D,N,LEN,ERROR &
               [,MATRIX,BSIZE,NRHS,DFILENAME,SFILENAME,RECL,PIVTOL,LP])

```

D is a scalar of INTENT(OUT) and type MA55_DATA. It is initialized in preparation for calls to MA55_ROWS.

N is a scalar of INTENT(IN) and type default INTEGER. It must be set by the user to the matrix order. **Restriction:** $N \geq 0$.

LEN is an array of shape (N), INTENT(IN), and type default INTEGER. For $i = 1, 2, \dots, N$, $LEN(i)$ must be set by the user to hold the length of row i from the first entry in the row to the diagonal. **Restrictions:** $1 \leq LEN(i) \leq i$, $i = 1, 2, \dots, N$.

ERROR is a scalar of INTENT(OUT) and type default INTEGER. On successful return, it has the value zero. After an unsuccessful return, it has one of the following values: -1, -2, -3, -4, -5, -6, -7, -8, -9, -10, -11; the meanings are given in Section 2.2.9.

Optional arguments

Since we reserve the right to add additional optional arguments in future releases of the code, **we strongly recommend that the optional arguments be called by keyword, not by position.**

MATRIX is an optional scalar of INTENT(IN) and type default LOGICAL. If present, it specifies whether the matrix is to be stored. If absent, the matrix is not stored.

BSIZE is an optional scalar of INTENT(IN) and type default INTEGER. If present, it specifies the block size desired during solution. If absent, the value 4 is employed. For advice on choosing the value, see Section 2.2.8. **Restriction:** $BSIZE \geq 1$.

NRHS is an optional scalar of INTENT(IN) and type default INTEGER. If present, it specifies the number of right-hand sides to be processed during calls to MA55_ROWS, which may be 0. If absent, the value 1 is employed. **Restriction:** $NRHS \geq 0$.

DFILENAME is an optional scalar of INTENT(IN) and type default CHARACTER with assumed character length. If present, it specifies the name to be used for the direct-access file holding the factors. The file must not be altered by the user while it is in use by HSL_MA55. If there is already a file with this name, the file must not be open and it is replaced. If DFILENAME is absent, a direct-access file with status SCRATCH is opened; in this case, a successful call to MA55_RESTART is not possible.

SFILENAME is an optional scalar of INTENT(IN) and type default CHARACTER with assumed character length. If MATRIX is present with the value .TRUE., a sequential file is opened to hold the matrix. If SFILENAME is present, it specifies the name to be used for this file. The file must not be altered by the user while it is in use by HSL_MA55. If there is already a file with this name, the file must not be open and it is replaced. If SFILENAME is

absent, a sequential file with status SCRATCH is opened; in this case, residuals are not available following a call to MA55_RESTART.

RECL is an optional scalar of INTENT(IN) and type default INTEGER. If present, it specifies the number of default REAL (double precision REAL in the DOUBLE version) values to be output in each direct-access record. If the argument is absent, the value $2*N$ is employed. **Restriction:** $RECL \geq 1$.

PIVTOL is an optional scalar of INTENT(IN) and type default REAL (double precision REAL in the DOUBLE version). If present, any pivot with absolute value less than PIVTOL is regarded as zero. If PIVTOL is absent or present with a negative value, the value zero is used.

LP is an optional scalar of INTENT(IN) and type default INTEGER. If present and positive, it specifies the unit for diagnostic messages. If present and non-positive, it specifies that no messages are wanted. If absent, the value 6 is employed.

2.2.3 To specify a block of rows

The user must call MA55_ROWS repeatedly to specify the lower-triangular part of the matrix in blocks of rows and (optionally) the corresponding right-hand side entries. The solution is completed on the call that includes the final row.

The block sizes are completely independent of the target block size for the solution, that is, the value of BSIZE of MA55_START. The user is free to specify the rows one at a time, but specifying more than one at once will reduce procedure call overheads and the overheads of several statements that are executed once for each call.

Each call is as follows:

```
CALL MA55_ROWS(D, LA, A, NROWS, LENR, ROWS, ERROR[, B, LB])
```

D is a scalar of INTENT(INOUT) and type MA55_DATA. It records data about the problem being solved. On the first call, it must be as on return from MA55_START, and it must not be changed between calls to MA55_ROWS or before calling MA55_SOLVE.

LA is a scalar of INTENT(IN) and type default INTEGER. It must be set by the user to the length of array A. The minimum possible value is $D\%MINLA$ (see Section 2.2.7). If the matrix is not being stored, the minimum possible value of LA without using a direct-access file is $D\%STORAGE$. If the matrix is being stored, the minimum possible value of LA without using a direct-access file $D\%STORAGE + D\%MAXBLK$ and the minimum possible value of LA without using a sequential file is $2 * D\%STORAGE$. Any scratch files opened by MA55_START but not needed are closed. LA must not be changed between calls to MA55_ROWS or before calling MA55_SOLVE. **Restriction:** $LA \geq D\%MINLA$.

A is an array of shape (LA), INTENT(INOUT), and type default REAL (double precision REAL in the DOUBLE version). A need not be initialized by the user and must not be changed between calls to MA55_ROWS or before calling MA55_SOLVE.

NROWS is a scalar of INTENT(IN) and type default INTEGER. It must be set by the user to the number of rows being presented on this call. The index of the final row presented on the previous call is available as $D\%IP$ (see Section 2.2.7). The rows presented here are $D\%IP + i$, $i = 1, 2, \dots, NROWS$. **Restrictions:** $NROWS \geq 1$, $NROWS \leq N - D\%IP$.

LENR is an array of shape (NROWS), INTENT(IN), and type default INTEGER. For $i = 1, 2, \dots, NROWS$, $LENR(i)$ must be set by the user to hold the length, from the first entry in the row to the diagonal, of the i -th being provided. This will usually be a section of the array LEN of MA55_START, but the only requirement is that the rows fit within the blocks chosen by MA55_START. **Restrictions:** $LENR(i) \geq 1$, $LENR(i) \leq i + D\%IP$, $i = 1, 2, \dots, NROWS$.

ROWS is an assumed-shape array of rank-one, INTENT(IN), and type default REAL (double precision REAL in the DOUBLE version). It must be set by the user to hold the entries in the rows of the block from the first nonzero to the diagonal, including embedded zeros. There must be no gaps between the diagonal of a row and the first entry of the next row.

ERROR is a scalar of INTENT(OUT) and type default INTEGER. On successful return, it has the value zero. The value 1 warns that the pivot in row i is zero (or has absolute value less than PIVTOL if this was specified in the call of MA55_START); the value of i is available as D%I (see Section 2.2.7). After an unsuccessful return, it has one of the following values: -12, -13, -14, -15, -16, -17, -22, -23, -24, -26; the meanings are given in Section 2.2.9.

Optional arguments

- B is an optional array of shape (N) or (LB, NRHS), INTENT(INOUT), and type default REAL (double precision REAL in the DOUBLE version), where N and NRHS are the matrix order and number of right-hand sides, as specified by the call to MA55_START. It must be present if NRHS > 0. It may have rank 1 only if NRHS == 1. It must either be set by the user to hold the right-hand side vector(s) before the first call of MA55_ROWS for the matrix and not changed thereafter, or the components of the block (that is, B(K+1:K+NROWS) or B(K+1:K+NROWS, :), where K = D%IP, see Section 2.2.7) may be set by the user before the call and the array otherwise not altered. After the final call, it holds the solution.
- LB is an optional scalar of INTENT(IN) and type default INTEGER. It must be present if B is present with rank 2.
Restriction: LB ≥ D%N.

2.2.4 To solve another equation or calculate residuals

Further systems with the same matrix but different right-hand sides may be solved or residuals calculated. The number of right-hand sides may be different from the number specified in the call to MA55_START.

```
CALL MA55_SOLVE(D, LA, A, ERROR, B[, LB, NRHS])
```

```
CALL MA55_RESID(D, LA, A, ERROR, B, X[, LB, NRHS, LX])
```

- D is a scalar of INTENT(INOUT) and type D%DATA. It records data about the problem being solved, and must not have been changed since the final call to MA55_ROWS for the matrix or a previous call to MA55_SOLVE for the matrix.
- LA is a scalar of INTENT(IN) and type default INTEGER. It must be set by the user to the length of array A. It must not have been changed since the final call to MA55_ROWS for the matrix or a previous call to MA55_SOLVE for the matrix.
- A is the array used in the calls to MA55_ROWS, and here is of INTENT(INOUT). If no direct-access file is in use (D%NREC = 0, see Section 2.2.7), it must not have been changed since the final call to MA55_ROWS for the matrix or a previous call to MA55_SOLVE for the matrix. Otherwise, it is used only as workspace.
- ERROR is a scalar of INTENT(OUT) and type default INTEGER. On successful return, it has the value zero. After an unsuccessful return, it has one of the following values: -4, -14, -19, -24 (MA55_SOLVE); -4, -18, -19, -20, -24, -25, -27 (MA55_RESID); the meanings are given in Section 2.2.9.
- B is an array of shape (N) or (LB, NRHS), INTENT(INOUT), and type default REAL (double precision REAL in the DOUBLE version), where N is the matrix order specified by the call to MA55_START. It must have rank 1 if NRHS is absent and rank 2 if NRHS is present. It must be set by the user to hold the right-hand side vector(s) and on return it holds the solution(s) (MA55_SOLVE) or the residuals (MA55_RESID).

For residuals only

Given vectors x and b or matrices X and B , the residual $b - Ax$ or $B - AX$ may be found, where A is the matrix presented to MA55_ROWS.

- X is an array of shape (N) or (LX, NRHS), INTENT(IN), and type default REAL (double precision REAL in the DOUBLE version). It must be set by the user to hold the vector x or matrix X .

Optional arguments (both cases)

LB is an optional scalar of `INTENT(IN)` and type default `INTEGER`. It must be present if B has rank 2. **Restriction:** $LB \geq D\%N$.

NRHS is an optional scalar of `INTENT(IN)` and type default `INTEGER`. It specifies the number of right-hand sides. It must be present if B has rank 2. **Restriction:** $NRHS \geq 0$.

Optional argument (residuals only)

LX is an optional scalar of `INTENT(IN)` and type default `INTEGER`. It must be present if X is present with rank 2. **Restriction:** $LX \geq D\%N$.

2.2.5 To indicate that processing for a matrix has finished

Calling `MA55_FINISH` causes the open files to be closed and permits their unit numbers to be used for other purposes. It has an option for writing all the data in the structure D to a file.

```
CALL MA55_FINISH(D, [FILENAME, ERROR])
```

D is a scalar of `INTENT(INOUT)` and type `MA55_DATA`. It records data about the problem being solved.

Optional arguments

FILENAME is an optional scalar of `INTENT(IN)` and type default `CHARACTER` with assumed character length. If present, it specifies the name of a sequential file to be used to hold the data in D. This file is opened, written, and closed.

ERROR is an optional scalar of `INTENT(OUT)` and type default `INTEGER`. It must be present if FILENAME is present. On successful return, it has the value zero. After an unsuccessful return, it has one of the following values: -7, -8, -10, -19, -21; the meanings are given in Section 2.2.9.

2.2.6 To restart processing

Calling `MA55_RESTART` restores all the data written by `MA55_FINISH` to a file and opens the files.

```
CALL MA55_RESTART(D, FILENAME, ERROR)
```

D is a scalar of `INTENT(OUT)` and type `MA55_DATA`. It records data about the problem being solved and is restored to its value when `MA55_FINISH` was called, except that if a scratch sequential file was in use, `D%MATRIX` is set to `.FALSE.`

FILENAME is a scalar of `INTENT(IN)` and type default `CHARACTER` with assumed character length. It specifies the name of the sequential file used to hold the data in D. This file is opened, read, and closed.

ERROR is a scalar of `INTENT(OUT)` and type default `INTEGER`. On successful return, it has the value zero. The value 2 warns that the named sequential file was not in use so residuals are no longer available. After an unsuccessful return, it has one of the following values: -6, -7, -8, -9, -10, -11, -18, -19, -21; the meanings are given in Section 2.2.9.

2.2.7 Data in D

Each of the following components of the scalar D of type `MA55_DATA` are scalars that are given values by `MA55_START` or `MA55_RESTART`. They are altered by other calls only where this is stated.

`D%FILENAME` is of type default `CHARACTER` with length 255 and holds the direct-access file name or null if no file name has been specified.

`D%FLOPS` is of type default `REAL` (double precision `REAL` in the `DOUBLE` version) and holds the number of floating-point operations performed in factorizing the matrix.

`D%IP` is of type default `INTEGER`. After a call to `MA55_START`, it holds the value zero. After a call to `MA55_ROWS`, it

holds the index of the last row processed.

D%I is of type default INTEGER. After a return from MA55_START or MA55_ROWS with ERROR having the value -2, it holds the index i .

D%LP is of type default INTEGER and holds the unit number for messages.

D%MATRIX is of type default LOGICAL and has the value .TRUE. if the matrix is stored.

D%MAXBLK is of type default INTEGER and holds the size of the largest block row.

D%MINLA is of type default INTEGER and holds an estimate of the least value of LA for a successful factorization.

D%N is of type default INTEGER and holds the matrix order.

D%ND is of type default INTEGER and holds the unit number of the direct-access file or zero if no file is open. The value may change following a call to MA55_ROWS, MA55_SOLVE, or MA55_FINISH.

D%ND2 is of type default INTEGER and holds the unit number of the sequential-access file or zero if no file is open. The value may change following a call to MA55_ROWS, MA55_SOLVE, or MA55_FINISH.

D%NREC is of type default INTEGER and holds the number of direct-access records written. The value may change following a call to MA55_ROWS.

D%NRHS is of type default INTEGER and holds the number of right-hand side vectors. The value may change following a call to MA55_SOLVE.

D%PIVTOL is of type default REAL (double precision REAL in the DOUBLE version) and holds the pivot threshold.

D%PROFILE is of type default REAL (double precision REAL in the DOUBLE version) and holds the number of entries in the factors, excluding extra entries added for the blocking.

D%RECL is of type default INTEGER and holds the array size of each record.

D%SBAND is of type default INTEGER and holds the semi-band width, that is, MAXVAL(LEN).

D%FILENAME is of type default CHARACTER with length 255 and holds the sequential file name or null if no file name has been specified.

D%STAT is of type default INTEGER and holds the STAT value following an unsuccessful allocation or deallocation and the IOSTAT value following an unsuccessful read or write. The value may change following a call to MA55_SOLVE.

D%STORAGE is of type default REAL (double precision REAL in the DOUBLE version) and holds the number of reals needed to store the matrix or its factorization (including the additional entries needed to make each block row rectangular).

D%ZERO is of type default INTEGER and holds the number of zero pivots found. The value may change following a call to MA55_ROWS.

2.2.8 Choice of the block size BSIZE

The best choice for BSIZE will vary with the matrix structure and the relative efficiency of the BLAS on the computer in use. As BSIZE increases, more floating-point operations and storage are needed, since the matrix is held as a set of rectangular blocks, each corresponding to the smallest rectangular submatrix that contains all the lower-triangular entries of the block of rows, including the diagonal. Calling MA55_START is not expensive, so the user may choose to call it for several values of BSIZE and look at the values returned by D%FLOPS and D%STORAGE (see Section 2.2.7). A modest increase in D%FLOPS is likely to be justified by the increase of speed caused by giving larger blocks to the level-3 BLAS.

2.2.9 Values of the argument `ERROR`

An unsuccessful call to any of the procedures is flagged with a negative value of the argument `ERROR`. The meanings are as follows:

- 1 $N < 0$.
- 2 $LEN(i) < 1$ or $LEN(i) > i$. The value of i is available as `D%I` (see Section 2.2.7).
- 3 $BSIZE < 1$.
- 4 $NRHS < 0$.
- 5 $RECL < 1$.
- 6 Direct-access file is already open.
- 7 Sequential file is already open.
- 8 Unable to find a unit for a file.
- 9 Unable to open the direct-access file. `D%STAT` (see Section 2.2.7) gives the `IOSTAT` value.
- 10 Unable to open a sequential file. `D%STAT` (see Section 2.2.7) gives the `IOSTAT` value.
- 11 Allocation failure. `D%STAT` (see Section 2.2.7) gives the `STAT` value.
- 12 $LA < D\%MINLA$ (see Section 2.2.7).
- 13 $NROWS < 1$ or $NROWS > N - D\%IP$ (see Section 2.2.7).
- 14 Unable to read the direct-access file. `D%STAT` (see Section 2.2.7) gives the `IOSTAT` value.
- 15 Unable to write to the direct-access file. `D%STAT` (see Section 2.2.7) gives the `IOSTAT` value.
- 16 The pivot in row i is negative (or less than $-PIVTOL$ if this was specified in the call of `MA55_START`). The value of i is available as `D%I` (see Section 2.2.7).
- 17 Unable to write to the sequential file. `D%STAT` (see Section 2.2.7) gives the `IOSTAT` value.
- 18 Unable to read the sequential file. `D%STAT` (see Section 2.2.7) gives the `IOSTAT` value.
- 19 The sequence of calls to `MA55_ROWS` has not been completed successfully.
- 20 Residual calculation requested but `MATRIX` was not specified with the value `.TRUE.` in the `MA55_START` call.
- 21 Named direct-access file was not in use so the factors will be lost.
- 22 `B` absent on call to `MA55_ROWS` when $NRHS > 0$.
- 23 `B` present with rank 1 on call to `MA55_ROWS` when $NRHS \neq 1$.
- 24 $LB < D\%N$ (see Section 2.2.7).
- 25 $LX < D\%N$ (see Section 2.2.7).
- 26 $LENR(i) < 1$ or $LENR(i) > i + D\%IP$. The value of i is available as `D%I` (see Section 2.2.7).
- 27 Unable to rewind to the sequential file. `D%STAT` (see Section 2.2.7) gives the `IOSTAT` value.

3 GENERAL INFORMATION

Other routines called directly: The BLAS `SGEMM/DGEMM`, `SSYRK/DSYRK`, `STRSM/DTRSM`, `STRSV/DTRSV`.

Input/output: Diagnostic messages on unit `LP` (see §2.2.2). A direct-access file is used, if necessary, for the factors. A sequential file is used for the matrix if this is requested. Another sequential file is used if it is required to preserve the factors for later use.

Restrictions: $N \geq 1$; $LEN(i) \geq 1$, $LEN(i) \leq i$, $i = 1, 2, \dots, N$; $BFSIZE \geq 1$; $NRHS \geq 0$; $RECL \geq 1$; $LA \geq D\%MINLA$; $NROWS \geq 1$, $NROWS \leq N-D\%IP$; and $LENR(i) \geq 1$, $LENR(i) \leq i+D\%IP$, $i = 1, 2, \dots, NROWS$.

4 METHOD

Cholesky block LL^T decomposition without interchanges is used. The elimination proceeds within the array A as long as possible, but if space runs out then a set of block rows that are known not to be required later in the elimination are written to disk. The remaining space in A is employed in *wrap-round* fashion, with efficiency maintained by keeping each block row contiguous.

If $BFSIZE$ is given a value $b > 1$, the block sizes are chosen in sequence by $MA55_START$. If rows 1 to i have been grouped into blocks and row $i+b$ has its first entry ahead of the first entry of the previous $b-1$ rows, the blocksize $b-1$ is used. Otherwise, if row $i+b+1$ has its first entry ahead of the first entry of the previous b rows, the blocksize b is used. Otherwise, the blocksize $b+1$ is used.

If a very small ($\leq D\%PIVTOL$) or zero pivot is encountered, the factorization is probably invalid. If this has happened ($D\%ZERO > 0$), we recommend that the solution be checked by using iterative refinement.

5 EXAMPLE OF USE

Suppose that it is required to solve the equation

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{T} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{T} & -\mathbf{I} \\ & -\mathbf{I} & \mathbf{T} \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} 4 & -1 & & & & & \\ -1 & 4 & -1 & & & & \\ & -1 & 4 & -1 & & & \\ & & -1 & 4 & -1 & & \\ & & & -1 & 4 & & \\ & & & & -1 & 4 & \\ & & & & & -1 & 4 \end{pmatrix}, \quad b_i = \begin{cases} 1 & i=7 \\ 0 & \text{otherwise,} \end{cases}$$

and estimate the error by finding the largest correction from one iterative correction.

Suitable code, using the file `ma55_temp` is as follows:


```

program example
  use hsl_ma55_double
  implicit none
  integer, parameter :: la=60, n=12
  double precision :: a(la), b(n), x(n), row(5)
  integer :: error,i,lrow,len(12)=(/1,2,2,2, 5,5,5,5, 5,5,5,5/)
  type(ma55_data) :: d

  call ma55_start(d,n,len,error,matrix=.true.,bsize=1,sfilename='ma55_temp')
  if( error /= 0) then
    write(6,'(a,i3,a,i3)') 'Error',error,' from ma55_start'
    stop
  end if
  b=0.0d0
  b(7)=1.0d0
  x = b
  do i = 1,n
    row = (/ -1.0d0,0.0d0,0.0d0,-1.0d0,4.0d0 /)
    ! Allow for rows that correspond to first row of T.
    if (modulo(i,4) == 1) row(4)=0.0d0
    lrow = 5
    ! Allow for the first block of rows.
    if (i <= 4) lrow = min(i,2)
    call ma55_rows(d,la,a,lrow,len(i:i),row(6-lrow:5),error,x)
    if( error /= 0) then
      write(6,'(a,i3,a,i3)') 'Error',error,' for row',i
      stop
    end if
  end do
  write(6,'(a/, (4f12.6))') ' Solution:',x
  write(6,'(a,f8.0)') ' Number of flops:      ',d%flops
  write(6,'(a,i3)') ' Number of negative pivots:',d%zero
  write(6,'(a,i3)') ' Maximum semi-band width: ',d%band
  call ma55_resid(d,la,a,error,b,x) ! Find residual
  call ma55_solve(d,la,a,error,b)  ! Find correction
  if( error /= 0) then
    write(6,'(a,i3,a,i3)') 'Error',error,' from ma55_solve'
    stop
  end if
  if (maxval(abs(b)) < 1e-15) then
    write(6, '(a)') ' Largest correction < 1e-15'
  else
    write(6,'(a,e12.4)') ' Largest correction:',maxval(abs(b))
  end if
  call ma55_finish(d)
end program example

```

This produces the output:

```

Solution:
  0.033167    0.078752    0.132516    0.065425
  0.053915    0.149326    0.385885    0.129184
  0.033167    0.078752    0.132516    0.065425
Number of flops:      205.
Number of negative pivots:  0
Maximum semi-band width:   5
Largest correction < 1e-15

```