1 SUMMARY

To **solve one or more set of sparse unsymmetric linear equations Ax = B from finite-element applications, using a multifrontal elimination scheme.** The matrix **A** must be input by elements and be of the form

$$\mathbf{A} = \sum_{k=1}^{m} \mathbf{A}^{(k)}$$

where $\mathbf{A}^{(k)}$ is nonzero only in those rows and columns that correspond to variables of the nodes of the k-th element. Optionally, the user may pass an additional matrix $\mathbf{A}_d$ of coefficients for the diagonal. **A** is then of the form

$$\mathbf{A} = \sum_{k=1}^{m} \mathbf{A}^{(k)} + \mathbf{A}_d$$

The right-hand side **B** should be assembled through the summation

$$\mathbf{B} = \sum_{k=1}^{m} \mathbf{B}^{(k)},$$

before calling the solution routine.

ATTRIBUTES — **Version:** 1.0.1. (31 March 2023) **Types:** Real (single, double). **Calls:** _GEMM, _GEMV, _GER, I_AMAX, _SCAL, _SWAP, _TRSM, _TRSV. **Origin:** A.C. Damhaug, Det Norske Veritas Research AS and J.K. Reid, Rutherford Appleton Laboratory. **Original date:** September 1995.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are four routines that can be called by the user:

(a) MA46I/ID sets default values for the control parameters for the other routines.

(b) MA46A/AD accepts the matrix pattern by element-node connectivity lists and chooses diagonal pivots for Gaussian elimination to preserve sparsity while disregarding numerical values. It also constructs information for the numerical factorization to be performed by MA46B/BD. The user may provide a pivot sequence by means of node numbers, in which case the necessary information for MA46B/BD will be generated.

(c) MA46B/BD factorizes the finite-element matrix **A** by NB calls, where NB is the number of assembly steps computed by routine MA46A/AD. For all the elements involving a node, the variables at the node must be in the same order. The actual pivot sequence may differ from that specified by MA46A/AD or provided by the user, due to numerical stability considerations.

(d) MA46C/CD uses the factors generated by MA46B/BD to solve the set of linear equations. The solution overwrites the right-hand side.

Normally, the user will call MA46I/ID prior to the call of any other routine in the package. If non-default values for any of the control parameters are required, they should be set immediately after the call to MA46I/ID. A call to MA46C/CD must be preceded by a call to MA46B/BD, which in turn must be preceded by a call to MA46A/AD. Since the information passed from one routine to the next is not corrupted by the second, several sequences of calls to MA46B/BD for matrices with the same sparsity pattern but different values may follow a single call to MA46A/AD, and similarly MA46C/CD can be used repeatedly to solve for different sets of right-hand sides **B**.

2.1.1 To set default values for control parameters

The single precision version

```
CALL MA46I(CNTL,ICNTL)
```

The double precision version

```
CALL MA46ID(CNTL,ICNTL)
```

CNTL is a REAL (DOUBLE PRECISION in the D version) array of length 2 that need not be set by the user. On return it contains default values. For further information see Section 2.2.

ICNTL is an INTEGER array of length 10 that need not be set by the user. On return it contains default values. For further information see Section 2.2.

2.1.2 To perform ordering and generate assembly tree

The single precision version

```
 CALL MA46A(NELS,NNODS,NEQNS,IPIELT,IELT,LIELT,IVAR,NB,KEEPA,LKEEPA,
$           IW,LIW,ICNTL,RINFO,INFO)
```

The double precision version

```
 CALL MA46AD(NELS,NNODS,NEQNS,IPIELT,IELT,LIELT,IVAR,NB,KEEPA,LKEEPA,
$            IW,LIW,ICNTL,RINFO,INFO)
```

NELS is an INTEGER variable that must be set by the user to the largest integer that is used to index a finite element. It is not altered by the routine.

NNODS is an INTEGER variable that must be set by the user to the largest integer that is used to index a finite-element node. It is not altered by the routine.

NEQNS is an INTEGER variable that must be set by the user to the number of variables. It is not altered by the routine.

IPIELT is an INTEGER array of length NELS+1. It must be set by the user so that the nodes connected to element I are in IELT(IPIELT(I)), IELT(IPIELT(I)+1), ..., IELT(IPIELT(I+1)-1) for I = 1, 2, ... , NELS. It is not altered by the routine.

IELT is an INTEGER array of length LIELT that must be set by the user to contain the lists of nodes in each element. Its length must be at least IPIELT(NELS+1)-1. It is not altered by the routine.

LIELT is an INTEGER variable that must be set by the user to the length of IELT. It is not altered by the routine.

IVAR is an INTEGER array of length NNODS that must be set by the user. It gives the number of variables for each node. It may contain values equal to zero. A node, I, I = 1, 2, ... , NNODS that has IVAR(I)=0 is not processed. It is not altered by the routine.

NB is an INTEGER variable that need not be set by the user. On exit it holds the number of assembly steps needed to factor the matrix. This variable must be preserved between a call to MA46A/AD and a sequence of calls to MA46B/BD.

KEEPA is an INTEGER array of length at least NELS+7*NNODS+55. If the user wishes to provide an ordering for the nodes, the index of the node in position i must be placed in KEEPA(i), i = 1, 2, ... , NNODS and ICNTL(4) must be set to 1. The given order is likely to be replaced by one that is equivalent apart from reordering of additions and subtractions. Otherwise, KEEPA need not be set by the user. On exit, KEEPA contains, in locations KEEPA(51:51+NB), a pointer array into KEEPA for the sequence of finite elements needed in each assembly step. For assembly step, IBL, IBL = 1, 2, ... , NB, the index of the first element required by MA46B/BD is found in location KEEPA(FIRST), where FIRST = KEEPA(50+IBL)+51+NB, and the last element index is found in location KEEPA(LAST), where LAST = KEEPA(51+IBL)+50+NB. The number of elements needed in assembly step IBL is thus KEEPA(51+IBL)-KEEPA(50+IBL). KEEPA must be preserved between a call to MA46A/AD and other routines.

LKEEPA is an INTEGER variable that must be set by the user to the length of KEEPA. It is not altered by the routine.

IW is an INTEGER array of length LIW that need not be set by the user. It is used as workspace by the routine. Its length must be at least max(l_1, l_2), where l_1 = 3*NELS+10*NNODS+4*LIELT+2 (or NELS+NNODS+2*LIELT+2 if the pivot order is specified in KEEPA), and l_2 = NELS+11*NNODS+2*LIELT+5.

LIW is an INTEGER variable that must be set by the user to the length of IW. It is not altered by the routine.

ICNTL is an INTEGER array of length 10 that contains control parameters and must be set by the user. Default values for the components may be set by a call to MA46I/ID. Details of the control parameters are given in Section 2.2. It is not altered by the routine.

RINFO is a REAL (DOUBLE PRECISION in the D version) array of length 6 that need not be set by the user. For the meaning of the values of components of RINFO set by MA46A/AD, see Section 2.2.

INFO is an INTEGER array of length 16 that need not be set by the user. On return from MA46A/AD, a value of zero for INFO(1) indicates that the routine has performed successfully. For nonzero values, see Section 2.3. For the meaning of the value of other components of INFO set by MA46A/AD, see Section 2.2.

2.1.3 To factorize a matrix

To factorize the matrix, MA46B/BD uses 'reverse communication' which means that the routine must be called by the user NB times, where NB is the number of assembly steps determined by MA46A/AD. In each call, the user must pass a specified sequence of finite-element coefficient matrices to the routine.

The single precision version

```
 CALL MA46B(IBL,NELS,NNODS,IPIELT,IELT,LIELT,IVAR,KEEPA,LKEEPA,KEEPB,
$           LKEEPB,ELMAT,A,LA,AD,LAD,IW,LIW,CNTL,ICNTL,RINFO,INFO)
```

The double precision version

```
 CALL MA46BD(IBL,NELS,NNODS,IPIELT,IELT,LIELT,IVAR,KEEPA,LKEEPA,KEEPB,
$            LKEEPB,ELMAT,A,LA,AD,LAD,IW,LIW,CNTL,ICNTL,RINFO,INFO)
```

IBL is an INTEGER variable that must be set by the user to the current assembly step. Calls to the routine must be in the order IBL = 1, 2, . . . , NB . It is not altered by the routine.

NELS , NNODS, IPIELT, IELT, LIELT and IVAR are as in the preceding call to MA46A/AD and their values must not have changed. They are not altered by the routine.

KEEPA is an INTEGER array of length LKEEPA. It must be as on exit from MA46A/AD. It is not altered by the routine.

LKEEPA is an INTEGER variable that must be set by the user to the length of KEEPA. It must be at least as great as INFO(2) as output from MA46A/AD (see Section 2.2). It is not altered by the routine.

KEEPB is an INTEGER array of length at least LKEEPB that need not be set by the user. It is used as workspace by MA46B/BD and on exit holds integer index information on the matrix factors. It must be preserved by the user between the calls to this routine and subsequent calls to MA46C/CD.

LKEEPB is an INTEGER variable that must be set by the user to the length of KEEPB. It must be at least as great as INFO(8) as output from MA46A/AD (see Section 2.2). A greater value is recommended because numerical pivoting may increase storage requirements. It is not altered by the routine.

ELMAT is a REAL (DOUBLE PRECISION in the D version) array that must be set by the user to hold the element coefficient matrices for this assembly step, column by column in the sequence defined by KEEPA(FIRST), KEEPA(FIRST+1), ..., KEEPA(LAST), where FIRST = KEEPA(50+IBL)+51+NB and LAST = KEEPA(51+IBL)+50+NB. It is not altered by the routine.

A is a REAL (DOUBLE PRECISION in the D version) array of length LA that need not be set on the first entry to MA46B/BD. It must be preserved between the calls to MA46B/BD and for subsequent calls to MA46C/CD. On exit

from each intermediate call, A will hold the entries of the factors of the matrix **A** that have been completed. On exit from the final call, A holds the factors needed by MA46C/CD.

LA is an INTEGER variable that must be set by the user to the length of A. It must be at least as great as INFO(9) as output from MA46A/AD (see Section 2.2). It is advisable to allow a greater value because the use of numerical pivoting may increase storage requirements. It is not altered by the routine.

AD is a REAL (DOUBLE PRECISION in the D version) array of length LAD that need not be set if ICNTL(10) has its default value (see Section 2.2). Otherwise, its NEQNS first positions must hold the coefficients for the diagonal of **A**. It is assumed by the routine that variables at nodes are stored consecutively and that nodes are in the initial order. MA46B/BD alters the order of the entries according to the tentative pivot order computed by MA46A/AD.

LAD is an INTEGER variable that must be set by the user to the length of AD. It must be set to at least 1 if ICNTL(10) has its default value. Otherwise, it must be set to a value as least as great as NEQNS as input to MA46A/AD.

IW is an INTEGER array of length LIW that need not be set by the user. It is used as workspace by the routine.

LIW is an INTEGER variable that must be set by the user to the length of IW. It must be at least as great as 3*(NNODS+NEQNS)+1. NNODS and NEQNS are as input to MA46A/AD. It is not altered by the routine.

CNTL is a REAL (DOUBLE PRECISION in the D version) array of length 2 that contains control parameters and must be set by the user. Default values for the components may be set by a call to MA46I/ID. Details of the control parameters are given in Section 2.2. It is not altered by the routine.

ICNTL is an INTEGER array of length 10 that contains control parameters and must be set by the user. Default values for the components may be set by a call to MA46I/ID. Details of the control parameters are given in Section 2.2. It is not altered by the routine.

RINFO is a REAL (DOUBLE PRECISION in the D version) array of length 6 that need not be set by the user. For the meaning of the values of components of RINFO set by MA46B/BD, see Section 2.2.

INFO is an INTEGER array of length 16 that need not be set by the user. On return from MA46B/BD, a value of zero for INFO(1) indicates that the routine has performed successfully. For nonzero values, see Section 2.3. For the meaning of the value of other components of INFO set by MA46B/BD, see Section 2.2.

2.1.4 To solve equations, given the factorization

The single precision version

```
      CALL MA46C(IVAR,NNODS,KEEPA,LKEEPA,KEEPB,LKEEPB,A,LA,B,LDB,NRHS,
     $           IW,LIW,RW,LRW,ICNTL,INFO)
```

The double precision version

```
      CALL MA46CD(IVAR,NNODS,KEEPA,LKEEPA,KEEPB,LKEEPB,A,LA,B,LDB,NRHS,
     $            IW,LIW,RW,LRW,ICNTL,INFO)
```

NNODS and IVAR are as in the preceding call to MA46A/AD and their values must not have changed. They are not altered by the routine.

KEEPA is an INTEGER array of length at least LKEEPA. The first INFO(2) components must be as on exit from MA46B/BD.

LKEEPA is an INTEGER variable that must be set by the user to the length of KEEPA. It must be at least as great as INFO(2) as output from MA46A/AD (see Section 2.2). It is not altered by the routine.

KEEPB is an INTEGER array of length at least LKEEPB. The first INFO(8) components must be as on exit from MA46B/BD.

LKEEPB is an INTEGER variable that must be set by the user to the length of KEEPB. It must be at least as great as

INFO(8) as output from MA46B/BD (see Section 2.2). It is not altered by the routine.

A is a REAL (DOUBLE PRECISION in the D version) array of length LA that must be unchanged since the call to MA46B/BD. It is not altered by the routine.

LA is an INTEGER variable that must be set by the user to the length of A. It must be at least as great as INFO(9) as output from MA46B/BD which may be smaller than predicted in MA46A/AD (see Section 2.2). It is not altered by the routine.

B is a REAL (DOUBLE PRECISION in the D version) array of leading dimension LDB, whose first NRHS columns must be set by the user to hold the right-hand sides. It is assumed that the right-hand side is passed to MA46C/CD in the input node order with the variables at each node stored consecutively. On exit, the solution overwrites the right hand side and the initial nodal order with variables at each node stored consecutively is maintained.

LDB is an INTEGER variable that must be set by the user to the leading dimension of B. It must be at least as great as NEQNS. It is not altered by the routine.

NRHS is an INTEGER variable that must be set by the user to hold the number of right hand sides to be solved in this call to MA46C/CD. It is not altered by the routine.

IW is an INTEGER array of length LIW that need not be set by the user. It is used as workspace by the routine and must be preserved between the calls to the routine.

LIW is an INTEGER variable that must be set by the user to the length of IW. It must be at least as great as NNODS+NEQNS+1. NNODS and NEQNS are as input to MA46A/AD. It is not altered by the routine.

RW is a REAL (DOUBLE PRECISION in the D version) work array of length as least as great as INFO(15)*NRHS, where INFO(15) is as output from MA46B/BD. RW need not be set by the user and is used as workspace by the routine.

LRW is an INTEGER variable that must be set by the user to the length of RW. It is not altered by the routine.

ICNTL is an INTEGER array of length 10 that contains control parameters and must be set by the user. Default values for the components may be set by a call to MA46I/ID. Details of the control parameters are given in Section 2.2. It is not altered by the routine.

INFO is an INTEGER array of length 16 that need not be set by the user. On return from MA46C/CD, a value of zero for INFO(1) indicates that the routine has performed successfully. For nonzero values, see Section 2.3. For the meaning of the value of other components of INFO set by MA46C/CD, see Section 2.2.

2.2 Arrays for control and information

The elements of the arrays CNTL and ICNTL control the action of MA46A/AD, MA46B/BD and MA46C/CD. Default values for the elements are set by MA46I/ID. The elements of the arrays RINFO and INFO provide information on the action of MA46A/AD, MA46B/BD and MA46C/CD.

CNTL(1) has default value 0.1 and is used for pivoting by MA46B/BD. Values greater than 1.0 are treated as 1.0 and less than zero as zero.

CNTL(2) has default value zero. If it is set to a positive value, MA46B/BD will treat any pivot whose modulus is less than CNTL(2) as zero.

ICNTL(1) has default value 6 and holds the unit number to which the error messages are sent.

ICNTL(2) has default value 6 and holds the unit number to which warning messages and additional printing is sent.

ICNTL(3) is used by the routines to control printing. It has default value 1. Possible values are:

- 0 No printing.
- 1 Error messages only.
- 2 Error and warning messages only.

3 Scalar parameters and a few entries of arrays on entry and exit from routines.

4 All parameter values printed on entry and exit from routines.

`ICNTL(4)` has default value 0. It must be set by the user to a value of 1 when calling `MA46A/AD` if a pivot sequence is being supplied by the user in array `KEEPA`.

`ICNTL(5)` has default value 0. This option is related to the node ordering step of `MA46A/AD`. If the value is zero or less, the minimum external degree algorithm is used. Multiple elimination is used when the value is zero. If value is greater than zero, multiple elimination is still in effect, but the minimum external degree condition is relaxed (see Section 4).

`ICNTL(6)` has default value 0. With this value, `MA46A/AD` reorders the assembly steps to reduce the temporary working storage required by `MA46B/BD` while computing the triangular factors. If `ICNTL(6)` is set to 1, `MA46A/AD` uses a standard depth first postordering of the assembly steps.

`ICNTL(7)` has default value 0. It is ignored in the present version, but the intention is for a later version to have the option of amalgamating tree nodes into supernodes even if this introduces additional structural zeros.

`ICNTL(8)` has default value 64. `MA46B/BD` is written to make good use of the cache memory if its size in kBytes is `ICNTL(8)`. Setting the value to zero will mean that the routine assumes that the computer has no cache. Where there are two levels of cache, it is probably the size of the secondary (larger) cache that is important.

`ICNTL(9)` has default value 0. It is ignored in the present version, but the intention is for a later version to have the option of using indirect addressing in the solve step of `MA46C/CD`.

`ICNTL(10)` has default value 0. This means that no diagonal matrix A_d is used to specify the diagonal matrix nonzero coefficients, otherwise `ICNTL(10)` must be set to 1.

`RINFO(1)` gives the number of floating-point additions used to assemble the original finite-element matrix coefficients.

`RINFO(2)` gives the number of floating-point additions used to assemble the generated elements if the tentative pivot sequence calculated by `MA46A/AD` is acceptable numerically.

`RINFO(3)` gives the sum of floating-point additions, multiplications and divisions used to factorize the matrix if it the tentative pivot sequence calculated by `MA46A/AD` is acceptable numerically.

`RINFO(4)` gives the number of floating-point additions used to assemble the generated elements in `MA46B/BD`.

`RINFO(5)` gives the sum of floating-point additions, multiplications and divisions used to factorize the matrix in `MA46B/BD`.

`RINFO(6)` gives the sum of floating-point additions, multiplications and divisions used to solve one set of linear equations in `MA46C/CD`.

`INFO(1)` has the value zero if the call was successful, and a negative value in the event of an error (see Section 2.3).

`INFO(2)` gives the required size of `KEEPA` in `MA46B/BD` and `MA46C/CD` on exit from `MA46A/AD` if `INFO(1)=0`. If `INFO(1)=-1` it gives the required size of `KEEPA` needed in `MA46A/AD`.

`INFO(3)` gives the size of `IW` that has been used in `MA46A/AD` or in `MA46B/BD` if `INFO(1)=0`. If `INFO(1)=-1` it gives the required size of `IW` needed in `MA46A/AD`. If `INFO(1)=-6` it gives the required size of `IW` needed in `MA46B/BD`.

`INFO(4)` gives the number of entries out of range for `INFO(1)=-2`.

`INFO(5)` gives the number of duplicate entries for `INFO(1)=-2`.

`INFO(6)` gives the number of active nodes computed by `MA46A/AD` if `INFO(1)=0`.

`INFO(7)` gives the number of variables computed by `MA46A/AD` if `INFO(1)=0`.

`INFO(8)` gives on exit from `MA46A/AD` with `INFO(1)=0`, the minimum required length of `KEEPB` in `MA46B/BD` on the

assumption that no pivoting is needed. On exit from MA46B/BD with INFO(1)=0, it gives the required length of KEEPB in MA46C/CD. If INFO(1)=−6 on exit from MA46B/BD, it gives the minimum required length of KEEPB for a successful exit.

INFO(9) gives on exit from MA46A/AD with INFO(1)=0, the minimum required length of A in MA46B/BD on the assumption that no pivoting is needed. On exit from MA46B/BD with INFO(1)=0, it gives the required length of A in MA46C/CD. If INFO(1)=−7 on exit from MA46B/BD, it gives the minimum required length of A for a successful exit.

INFO(10) gives on exit from MA46A/AD with INFO(1)=0, the expected order of the largest front matrix on the assumption that no pivoting is needed. On exit from MA46B/BD with INFO(1)=0, it gives the actual order of the largest front matrix.

INFO(11) gives on exit from MA46A/AD with INFO(1)=0, the expected number of indices in the factorized matrix on the assumption that no pivoting is needed. On exit from MA46B/BD with INFO(1)=0, it gives the actual number of indices in the factorized matrix.

INFO(12) gives on exit from MA46A/AD with INFO(1)=0, the expected number entries in the factorized matrix on the assumption that no pivoting is needed. On exit from MA46B/BD with INFO(1)=0, it gives the actual number entries in the factorized matrix.

INFO(13) gives the number of assembly steps if INFO(1)=0.

INFO(14) gives the number of elements if INFO(1)=0.

INFO(15) gives the size of the largest front matrix that occurred in the factorization step if INFO(1)=0.

INFO(16) gives the number of eliminations done by MA46B/BD if INFO(1)=0.

2.3 Error diagnostics

A successful return from MA46A/AD or MA46B/BD is indicated by a value of INFO(1) equal to zero. Possible nonzero values for INFO(1) are given below.

A nonzero flag value is associated with an error message that will be output on unit ICNTL(1).

-1 The length of KEEPA and/or IW is not great enough (MA46A/AD).

-2 Entries in KEEPA are out of range and/or are duplicates (MA46A/AD).

-3 NEQNS less than the number of variables computed by MA46A/AD or the number of variables computed is less than one (MA46A/AD).

-4 Indices out of range in IELT (MA46A/AD).

-5 NELS ≤ 0, and/or NNODS ≤ 0, and/or NEQNS ≤ 0 (MA46A/AD).

-6 The length of KEEPB and/or IW is not great enough. (MA46B/BD).

-7 The length of A is not great enough. (MA46B/BD).

-8 Error from previously called routine is not cleared (MA46B/BD or MA46C/CD).

-9 Error in the symbolic assembly step in MA46B/BD. Signals that KEEPA may have been altered before the call to MA46B/BD.

2.4 Singular systems

If the matrix is singular, MA46B/BD factorizes a nonsingular submatrix. A warning message is written if the right-hand side is not consistent with the factorization.

3 GENERAL INFORMATION

Use of common: None.

Other routines called directly: MA46D/DD, MA46E/ED, MA46F/FD, MA46G/GD, MA46H/HD, MA46J/JD, MA46K/KD, MA46L/LD, MA46M/MD, MA46N/ND, MA46O/OD, MA46P/PD, MA46Q/QD, MA46R/RD, MA46S/SD, MA46T/TD, MA46U/UD, MA46V/VD, MA46W/WD, MA46X/XD, MA46Y/YD, MA46Z/ZD, MA56A/AD, MA56B/BD, MA56C/CD, MA56D/DD, MA56E/ED, MA56F/FD, MA56G/GD, MA56H/HD, MA56I/ID, MA56J/JD, MA56K/KD, MA56L/LD, MA56M/MD, MA56N/ND, MA56O/OD and MA56P/PD.

The package uses the Basic Linear Algebra Subprograms SGEMM/DGEMM, SGEMV/DGEMV, SGER/DGER, SSCAL/DSCAL, ISAMAX/IDAMAX, STRSM/DTRSM and SSWAP/DSWAP.

Input/output: Error messages on unit ICNTL(1). Warning messages and additional printing on unit ICNTL(2). Each has default value 6.

4 METHOD

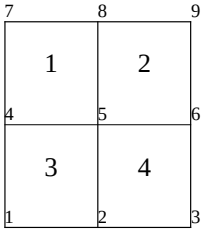
The method used is a direct method using multifrontal sparse Gaussian elimination. The matrix structure is passed to the routine in the form of element-node connectivity lists. The matrix analyse step (MA46A/AD) uses this 'unassembled' form to find the ordering of the nodes, and to build the necessary information for the factorise and solve steps. The ordering is done with the minimum degree heuristic. It is possible to relax this by altering the default value given by ICNTL(5). Setting it greater than zero has the effect of allowing nodes with degree ICNTL(5) greater than the minimum to be eliminated together with the nodes of minimum degree. Sometimes, this helps to reduce the size of the decomposition. The final assembly tree is reordered in an attempt to reduce the size of the working stack. The default value of option ICNTL(6) gives this and is recommended. The factorization step (MA46B/BD) is provided by the analyse step with a tentative pivot sequence, which it uses except when this would be numerically unstable. The numerical stability criterion is the relative pivot tolerance given by CNTL(1), with a default value of 0.1. In general, increasing its value gives a more stable factorization, but increases in the size of the decomposition. A value of 1.0 gives partial pivoting as defined for the dense matrix case.

Reference A.C. Damhaug and J.K. Reid (1996) MA46, a FORTRAN code for direct solution of sparse unsymmetric linear systems of equations from finite-element applications. Rutherford Appleton Laboratory Report RAL-TR-96-010.

5 EXAMPLE OF USE

We give an example of the code required to solve a set of equations using the MA46 package. The example illustrates the use of MA46 when no input order and additional diagonal matrix $\mathbf{A}_d$ is provided by the user. There are two right-hand sides to solve for.

We wish to solve the following simple finite-element problem in which the finite-element mesh consists of four 4-noded elements with two degrees of freedom at each node. The nodes 1, 4, and 7 are assumed constrained, which means that they do not contribute to the matrix system to be solved.



The input to the routine is then:

```
NELS   = 4
NNODS  = 9
```

```
NEQNS  = 12
LIELT  = 16
IVAR   = [0,2,2,0,2,2,0,2,2]
IPIELT = [1,5,9,13,17]
IELT   = [4,5,8,7,5,6,9,8,1,2,5,4,2,3,6,5]
```

The four elemental matrices $\mathbf{A}^{(k)}$ $(1 \le k \le 4)$ are

$$\begin{array}{c} 5 \\ 8 \end{array}\begin{pmatrix} 6. & 2. & 3. & 4. \\ 1. & 5. & 4. & 3. \\ 2. & 3. & 4. & 4. \\ 3. & 2. & 3. & 1. \end{pmatrix}$$

$$\begin{array}{c} 5 \\ 6 \\ 9 \\ 8 \end{array}\begin{pmatrix} 4. & 4. & 3. & 4. & 5. & 4. & 3. & 4. \\ 3. & 1. & 4. & 3. & 4. & 2. & 4. & 3. \\ 2. & 3. & 6. & 2. & 3. & 4. & 7. & 2. \\ 3. & 2. & 1. & 5. & 4. & 3. & 2. & 6. \\ 4. & 3. & 2. & 3. & 4. & 4. & 3. & 4. \\ 3. & 1. & 3. & 2. & 3. & 1. & 4. & 3. \\ 2. & 3. & 6. & 1. & 2. & 3. & 6. & 2. \\ 3. & 2. & 1. & 5. & 3. & 2. & 1. & 5. \end{pmatrix},$$

$$\begin{array}{c} 2 \\ 5 \end{array}\begin{pmatrix} 6. & 2. & 3. & 4. \\ 1. & 5. & 4. & 3. \\ 2. & 3. & 4. & 4. \\ 3. & 2. & 3. & 1. \end{pmatrix}$$

$$\begin{array}{c} 2 \\ 3 \\ 6 \\ 5 \end{array}\begin{pmatrix} 4. & 4. & 3. & 4. & 5. & 4. & 3. & 4. \\ 3. & 1. & 4. & 3. & 4. & 2. & 4. & 3. \\ 2. & 3. & 6. & 2. & 3. & 4. & 7. & 2. \\ 3. & 2. & 1. & 5. & 4. & 3. & 2. & 6. \\ 4. & 3. & 2. & 3. & 4. & 4. & 3. & 4. \\ 3. & 1. & 3. & 2. & 3. & 1. & 4. & 3. \\ 2. & 3. & 6. & 1. & 2. & 3. & 6. & 2. \\ 3. & 2. & 1. & 5. & 3. & 2. & 1. & 5. \end{pmatrix},$$

where the node numbers are indicated by the integers before each matrix (columns are identified symmetrically to rows). The two right-hand side vectors $\mathbf{b}^{(k)}$ $(1 \le k \le 2)$ are

$$\begin{array}{c} 2 \\ 3 \\ 5 \\ 6 \\ 8 \\ 9 \end{array}\begin{pmatrix} 0. \\ 0. \\ 1. \\ 0. \\ 0. \\ 0. \\ 1. \\ 0. \\ 0. \\ 0. \\ 1. \\ 0. \end{pmatrix}$$

$$\begin{array}{c} 2 \\ 3 \\ 5 \\ 6 \\ 8 \\ 9 \end{array}\begin{pmatrix} 0. \\ 0. \\ 0. \\ 1. \\ 0. \\ 0. \\ 0. \\ 2. \\ 0. \\ 0. \\ 0. \\ 1. \end{pmatrix},$$

where the node numbers are indicated by the integers before each vector.

The following program is used to solve this problem.

```
      INTEGER           NELS    , NNODS    , NEQNS     , LIELT
      PARAMETER       ( NELS = 4, NNODS = 9, NEQNS = 12, LIELT = 16 )
      INTEGER           IVAR(NNODS), IPIELT(NELS+1), IELT(LIELT)
      INTEGER           KEEPA(200), KEEPB(200), IW(300), XELMAT(10),
     $                  ELSIZE(10)
      INTEGER           NB    , LKEEPA, LIW   , LKEEPB, LA    , LAD   ,
     $                  IBL   , IPEL  , LDB   , NRHS  , LRW   , LAMAX ,
     $                  LELMAT, L1    , L2    , I     , J     , K     ,
     $                  NODE  , NVAR  , XELSEQ, ELEMNT, ELSEQ ,
     $                  LORDER, ISTRT , ISTOP
      PARAMETER       ( LAMAX = 200, LELMAT = 200 )
      DOUBLE PRECISION  ELMAT(LELMAT), A(LAMAX), AD(NEQNS), B(NEQNS,2),
     $                  RW(NEQNS*2),RELMAT(LELMAT)
      INTEGER           ICNTL(10), INFO(16)
      DOUBLE PRECISION  CNTL(2), RINFO(6)

      NRHS=2
```

```
*     --------------------
*     READ IN THE DATA SET.
*     --------------------
      READ(5,'(10I3)') (IVAR(I),I=1,NNODS)
      READ(5,'(10I3)') (IPIELT(I),I=1,NELS+1)
      READ(5,'(10I3)') (IELT(I),I=1,LIELT)
      READ(5,'(8F5.0)') (ELMAT(I),I=1,160)
      READ(5,'(12F5.0)') ((B(I,J),I=1,NEQNS),J=1,NRHS)

*     ----------------------------------------
*     COMPUTE THE ORDER OF THE ELEMENT MATRICES.
*     ----------------------------------------
      DO 200 I = 1, NELS
         ELSIZE(I) = 0
         DO 100 J = IPIELT(I), IPIELT(I+1)-1
            NODE = IELT(J)
            NVAR = IVAR(NODE)
            IF ( NVAR .GT. 0 )
     $         ELSIZE(I)=ELSIZE(I)+NVAR
  100    CONTINUE
  200 CONTINUE

*     ----------------------------------------
*     CALL MA46ID TO INITIALIZE CONTROL ARRAYS.
*     ----------------------------------------
      CALL MA46ID(CNTL,ICNTL)

*     ------------------------------------------------
*     ANALYSE THE SPARSITY PATTERN BY A CALL TO MA46AD.
*     ------------------------------------------------
      LKEEPA = NELS+7*NNODS+55
      L1     = 3*NELS+2*NNODS+4*LIELT+8*NEQNS+2
      L2     = NELS+11*NNODS+2*LIELT+5
      LIW    = MAX(L1,L2)
      IF ( LKEEPA .GT. 200 .OR.
     $     LIW    .GT. 300      ) GOTO 8000

      CALL MA46AD(NELS,NNODS,NEQNS,IPIELT,IELT,LIELT,IVAR,NB,KEEPA,
     $            LKEEPA,IW,LIW,ICNTL,RINFO,INFO)
      IF ( INFO(1) .NE. 0 ) GOTO 8000

*     --------------------------------
*     STORE THE ELEMENT MATRICES IN THE
*     SEQUENCE DETERMINED BY MA46AD.
*     --------------------------------
      XELSEQ = 50
      ELSEQ  = XELSEQ+NB+1
      IPEL = 1
      XELMAT(1) = IPEL
      DO 600 IBL = 1, NB
         DO 500 I = KEEPA(XELSEQ+IBL), KEEPA(XELSEQ+IBL+1)-1
            ELEMNT = KEEPA(ELSEQ+I)
            LORDER = ELSIZE(ELEMNT)
            K = 0
            DO 300 J = 1, ELEMNT-1
               K = K + ELSIZE(J)*ELSIZE(J)
  300       CONTINUE
            XELMAT(IBL+1) = XELMAT(IBL) + LORDER*LORDER
            DO 400 J = IPEL, IPEL+LORDER*LORDER-1
               K = K + 1
```

```
                RELMAT(J) = ELMAT(K)
  400       CONTINUE
            IPEL = IPEL + LORDER*LORDER
  500    CONTINUE
         XELMAT(IBL+1) = IPEL
  600 CONTINUE

*     ---------------------------------------
*     SET UP THE STORAGE REQUIRED FOR MA46BD.
*     ---------------------------------------
      LKEEPA = INFO(2)
      LKEEPB = INFO(8)
      LA     = INFO(9)
      LAD    = 1
      LIW    = 3*(NNODS+NEQNS) + 1
      IF ( LKEEPA .GT. 200 .OR.
     $     LKEEPB .GT. 200 .OR.
     $     LA     .GT. 200 .OR.
     $     LIW    .GT. 300      ) GOTO 8000

*     -----------------------------------------
*     FACTORIZE THE MATRIX BY NB CALLS TO MA46BD.
*     -----------------------------------------
      DO 700 IBL = 1, NB
         IPEL = XELMAT(IBL)
         CALL MA46BD(IBL,NELS,NNODS,IPIELT,IELT,LIELT,IVAR,KEEPA,LKEEPA,
     $               KEEPB,LKEEPB,RELMAT(IPEL),A,LA,AD,LAD,IW,LIW,CNTL,
     $               ICNTL,RINFO,INFO)
         IF ( INFO(1) .NE. 0 ) GOTO 8000
  700 CONTINUE

*     ---------------------------------------
*     SET UP THE STORAGE REQUIRED FOR MA46CD.
*     ---------------------------------------
      LKEEPB = INFO(8)
      LA     = INFO(9)
      LIW    = NNODS + NEQNS + 1
      LRW    = INFO(15)*NRHS
      IF ( LKEEPB .GT. 200 .OR.
     $     LA     .GT. 200 .OR.
     $     LIW    .GT. 300 .OR.
     $     LRW    .GT. NEQNS*2    ) GOTO 8000
      LDB = NEQNS

*     -------------------------------------
*     SOLVE THE SYSTEMS BY A CALL TO MA46CD.
*     -------------------------------------
*      NRHS=1
      CALL MA46CD(IVAR,NNODS,KEEPA,LKEEPA,KEEPB,LKEEPB,A,LA,B,LDB,NRHS,
     $            IW,LIW,RW,LRW,ICNTL,INFO)

*     --------------------------
*     PRINT THE SOLUTION VECTORS.
*     --------------------------
      ISTRT = 1
      DO 1000 NODE = 1, NNODS
         IF ( IVAR(NODE) .GT. 0 )
     $   THEN
            ISTOP = ISTRT + IVAR(NODE) - 1
            DO 900 J = 1, NRHS
               WRITE(6,'(A,I6,A,I6)')
```

```
     $         'SOLUTION VECTOR ',J,'     FOR NODE :', NODE
               WRITE(6,'(45X,1PE12.3)')
     $         (B(I,J),I=ISTRT,ISTOP)
  900       CONTINUE
            ISTRT = ISTOP + 1
         ELSE
            WRITE(6,'(/A,I6/)')
     $      'NO VARIABLES AT NODE :', NODE
         ENDIF
 1000 CONTINUE

*     ---------------------
*     PRINT SOME STATISTICS:
*     ---------------------
      WRITE(6,'(A,I6)')
     $'NUMBER OF ASSEMBLY STEPS              ',INFO(13)
      WRITE(6,'(A,I6)')
     $'NUMBER OF ELIMINATIONS PERFORMED      ',INFO(16)
      WRITE(6,'(A,I6)')
     $'ORDER OF THE LARGEST FRONT MATRIX     ',INFO(15)
      WRITE(6,'(A,I6)')
     $'NUMBER OF ENTRIES IN THE FACTORS      ',INFO(12)
      WRITE(6,'(A,I6)')
     $'SIZE OF THE INDEX INFORMATION         ',INFO(11)

      STOP
 8000 CONTINUE
*     ------------------------------------
*     ERROR CONDITION, PRINT THE INFO ARRAY.
*     ------------------------------------
      WRITE(6,'(10I5)') (INFO(I),I=1,16)
      END
```

The input data used for this problem is:

```
  0  2  2  0  2  2  0  2  2
  1  5  9 13 17
  4  5  8  7  5  6  9  8  1  2
  5  4  2  3  6  5
   6.   2.   3.   4.   1.   5.   4.   3.
   2.   3.   4.   4.   3.   2.   3.   1.
   4.   4.   3.   4.   5.   4.   3.   4.
   3.   1.   4.   3.   4.   2.   4.   3.
   2.   3.   6.   2.   3.   4.   7.   2.
   3.   2.   1.   5.   4.   3.   2.   6.
   4.   3.   2.   3.   4.   4.   3.   4.
   3.   1.   3.   2.   3.   1.   4.   3.
   2.   3.   6.   1.   2.   3.   6.   2.
   3.   2.   1.   5.   3.   2.   1.   5.
   6.   2.   3.   4.   1.   5.   4.   3.
   2.   3.   4.   4.   3.   2.   3.   1.
   4.   4.   3.   4.   5.   4.   3.   4.
   3.   1.   4.   3.   4.   2.   4.   3.
   2.   3.   6.   2.   3.   4.   7.   2.
   3.   2.   1.   5.   4.   3.   2.   6.
   4.   3.   2.   3.   4.   4.   3.   4.
   3.   1.   3.   2.   3.   1.   4.   3.
   2.   3.   6.   1.   2.   3.   6.   2.
   3.   2.   1.   5.   3.   2.   1.   5.
   0.   0.   1.   0.   0.   0.   1.   0.   0.   0.   1.   0.
   0.   0.   0.   1.   0.   0.   0.   2.   0.   0.   0.   1.
```

The program produces the following output:

```
 NO VARIABLES AT NODE :     1

 SOLUTION VECTOR      1     FOR NODE :     2
                                          -9.806E-01
                                          -8.629E-02
 SOLUTION VECTOR      2     FOR NODE :     2
                                           6.861E-01
                                           9.451E-02
 SOLUTION VECTOR      1     FOR NODE :     3
                                          -6.576E-01
                                          -8.171E-01
 SOLUTION VECTOR      2     FOR NODE :     3
                                           9.307E-01
                                           5.849E-01

 NO VARIABLES AT NODE :     4

 SOLUTION VECTOR      1     FOR NODE :     5
                                           4.808E-01
                                           7.996E-01
 SOLUTION VECTOR      2     FOR NODE :     5
                                          -6.563E-01
                                          -4.634E-01
 SOLUTION VECTOR      1     FOR NODE :     6
                                           9.156E-01
                                           1.178E+00
 SOLUTION VECTOR      2     FOR NODE :     6
                                          -6.225E-01
                                          -9.864E-01

 NO VARIABLES AT NODE :     7

 SOLUTION VECTOR      1     FOR NODE :     8
                                          -7.587E-01
                                          -8.156E-01
 SOLUTION VECTOR      2     FOR NODE :     8
                                           4.504E-01
                                           8.148E-01
 SOLUTION VECTOR      1     FOR NODE :     9
                                          -1.542E+00
                                          -6.427E-01
 SOLUTION VECTOR      2     FOR NODE :     9
                                           1.682E+00
                                           2.914E-01
 NUMBER OF ASSEMBLY STEPS                   2
 NUMBER OF ELIMINATIONS PERFORMED          12
 ORDER OF THE LARGEST FRONT MATRIX          8
 NUMBER OF ENTRIES IN THE FACTORS         112
 SIZE OF THE INDEX INFORMATION             24
```