4.7 Sparse Square Nonsingular Systems of Equations

A. Purpose

For a sparse square nonsingular matrix, A, of order N, and an optional vector or matrix B, this routine will factor A, solve $AX = B$, $A^T X = B$, compute the reciprocal of the condition number, and obtain the determinant and the inverse. This code has similar functionality to the dense codes described in Chapter 4-1. The program in Chapter 19-6 can be used together with the code here to solve problems with the matrix given in Matrix Market format.

B. Usage

With minimal options, this simply solves $A\mathbf{x} = \mathbf{b}$ for a single vector **b**, and does not preserve the factorization. Other features are available through the options in ISPEC.

B.0.a Program Prototype, Single Precision

INTEGER N, ISPEC(?), IA(?)

DOUBLE PRECISION A(?), B(?), OPT(3)

Assign values to N, ISPEC(), IA(), A(), and B().

CALL DSPGE (N, ISPEC, IA, A, B, OPT)

Computed quantities are returned in IA(), A(), B(), and ISPEC.

B.0.b Argument Definitions

N [in] The order of the matrix A and number of rows in the B and X.

ISPEC() [inout] An array used to return a flag, specify dimensions and specify options. Locations are used as follows.

1 [inout] Must be 0 if the matrix is not factored. The value returned on the first call must be preserved on later calls when the matrix is already factored. If the returned value is < 0, then the matrix was not factored. See Section E. on page 2 for details.

2 [in] Declared dimension of IA() (or at least all that you want this routine to know about. This number must be at least the space required for storing the factors of the matrix + twice the space needed for the original matrix + 7N.

3 [in] Declared dimension of A(), this can have a dimension 4N less than that for IA.

4 [out] The total free space in A at the solution. If there was insufficient free space this gives the negative of the number of columns left to be factored.

5 [in] Start of options, see the next Section.

IA() [inout] IA(1) gives the number of nonzeros in the first column. IA(2) to IA(1 + IA(1)) give the row indexes for rows in the first column which must be in increasing order. IA(2+IA(1)) then gives the number of nonzeros in the second column, etc. Locations after this input data are used internally.

A() [inout] For each row index given in IA, the corresponding location in A contains the coefficient corresponding to that row and column. The remaining locations in A are used internally.

B() [inout] Used to hold the right hand side(s) on entry and to hold the solution on exit.

OPT() [out] Used only for some options..
OPT(1) = reciprocal condition number if requested.
OPT(2) $\times 10^{OPT[3]}$ = the determinant if requested.

B.1 Options

First, a summary of the options.

Name	Value	Brief Description
IPS	0	Solve $A\mathbf{x} = \mathbf{b}$
IPST	1	Solve $A^T\mathbf{x} = \mathbf{b}$
IPBDIM	2	Specifies dimensions on B
IPRCON	3	Compute reciprocal condition number
IPDET	4	Compute determinant

Starting in ISPEC(5) there is an index for an action which we denote here by I_a. Each I_a has associated with it a parameter name, a value for that parameter, and a (possibly empty) list of arguments, which follow immediately after I_a in ISPEC. After the last argument for an I_a, the next location in ISPEC contains the next I_a, with its associated arguments. This continues until the last I_a has been specified. An option index ≤ 1 ends the option list (other options must precede these). Setting ISPEC(5) = 0, gives the default action.

It is assumed that B is a one dimensional of length N if option IPBDIM is not used.

All the above options except for IPBDIM, take no arguments, and the above table should be self-explanatory. IPBDIM uses 3 locations in ISPEC, the 2 to indicate this option, next I_2 to indicate the number of columns (default is 1), and then J_2 to indicate the distance between successive columns of B. In addition one can get

the inverse by setting $I_2 = -1$, and DSPGE will initialize B to the identity matrix computing a solution. If no backsolve is desired, set I_2 to 0. In the case of a B with 5 columns one might have ISPEC(5)=2, ISPEC(6)=5, and ISPEC(7)=N. Note that the code always assumes that B is dense.

If you want to compute the inverse matrix and are not an expert, chances are you should ask an expert who is likely to tell you — don't. Given the ability to backsolve using a previous factorization the inverse should very rarely be needed.

B.1.a Program Prototype, Single Precision

Usage is the same as for double precision, except the "DOUBLE PRECISION" statement is changed to "REAL", and the name DSPGE is changed to SSPGE.

C. Examples and Remarks

Program DRDSPGE illustrates the use of DSPGE to solve a system of linear equations and compute the reciprocal condition number of the matrix of the system. Output is shown in ODDSPGE. The data for this problem were chosen so the exact solution has components 2, −5, and 3.

D. Functional Description

To factor the matrix, Gaussian elimination is used on columns to get a factorization of the form $A = UL^{-1}$. Pivot selection permutes rows and columns independently balancing a desire to keep an upper triangular form of the matrix without sacrificing stability.

Internal scaling first computes scale factors to make all the columns have an L_∞ norm of 1, and follows this by doing the same to the rows.

D.1 Estimating reciprocal condition number

The condition number of an $n \times n$ nonsingular matrix, A, is defined as $\kappa = \|A\| \times \|A^{-1}\|$, a quantity that is never less than unity. This is the largest factor by which the relative error in a vector, $\mathbf{y}$, can be amplified as a result of multiplication by A or by A^{-1}. Roughly speaking, if $\kappa = 10^k$ and the components of the vector $\mathbf{b}$ in the problem, $A\mathbf{x} = \mathbf{b}$, are known to d decimal digits, and the components of A are known to more than d decimal digits, and the problem is solved using precision greater than d decimal digits, then the solution will be known to about $d-k$ decimal digits. In particular, if $k \geq d$, the solution may have no reliable digits at all.

We use the procedure described in [1, pp. 128–130] (based on work in [2]), which requires on the order of $3n^2$ additional operations. In a test with 1250 cases reported in [2], the maximum value for the condition number / estimated condition number was just slightly more than 16.

D.2 Computing the determinant of A

We have $Det(A) = Det(U)Det(L^{-1})$. Since all row and column permutations are done on both L and U, and the diagonal of L contains 1's, the determinant is simply the product of the diagonals of U times -1 if the sum of the number of exchanges done on rows and columns of A is odd.

For a matrix of large order it is not uncommon for the determinant to be of extremely large or small magnitude. Consequently we compute and store the determinant as a pair of numbers, permitting a very large exponent range. The second number of the pair is always in integer.

References

1. G. H. Golub and C. F. van Loan, **Matrix Computations, Third Edition**, Johns Hopkins, Baltimore and London (1996) 549 pages.
2. J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, **LINPACK Users' Guide**, Society for Industrial and Applied Mathematics, Philadelphia (1979) 320 pages.

E. Error Procedures and Restrictions

Error messages are printed using the facility described in Chapter 19-03. Options there give one some control over actions taken on errors, and on where the debug print goes. Error indexes < -4 stop by default, those > -4 only return an error flag, and the -4 case will return only if there was enough space to get started on the factorization. The error flags returned are

-1 Matrix appears singular. (Sets OPT(1) = 0.)
-2 Matrix has an empty column.
-3 Matrix has an empty row.
-4 No more space.
-5 N ≤ 0
-6 An unknown option.
-7 Problem with column size.
-8 Problem with row indexes.

F. Supporting Information

The source language is ANSI Fortran 77.

Design and programming by Fred T. Krogh, Math à la Carte, Inc. March 2006. We hope to revisit this in the future and improve performance.

Entry	Required Files
DSPGE	DMESS, DSPGE, MESS
SSPGE	MESS, SMESS, SSPGE

DRDSPGE

```
c>> 2006-03-11 DRDSPGE  Krogh  Initial code.
c--D replaces "?": DR?SPGE, ?SPGE
c Demo driver for DRSPGE
      integer N, LA, LIA
      parameter (N=6, LA=300, LIA=LA+5*N)
      integer IA(LA+5*N), ISPEC(7), I, J, K
      double precision A(LA),OPT(3),SIZE,B(N),BB(N),X(N),ERRX,ERRRES
      data IA(1),A(1), IA(2),A(2), IA(3),A(3) /2,0.D0, 1,-2.D0, 4,1.D0/
      data IA(4),A(4), IA(5),A(5) /1,0.D0, 3,-2.E0/
      data IA(6),A(6), IA(7),A(7), IA(8),A(8) /2,0.D0, 2,-2.D0, 5,-2.D0/
      data IA( 9),A( 9), IA(10),A(10), IA(11),A(11), IA(12),A(12)  /
     1  3,0.D0, 1,-1.D0, 5,-1.D0, 6,2.D0 /
      data IA(13),A(13), IA(14),A(14), IA(15),A(15), IA(16),A(16),
     1  IA(17),A(17), IA(18),A(18) / 5,0.D0, 1,-1.D0, 2,-2.D0, 3,2.D0,
     2  4,-1.D0, 6,-2.D0 /
      data IA(19),A(19), IA(20),A(20), IA(21),A(21), IA(22),A(22) /
     1  3,0.D0,  1,-2.D0, 3,2.D0, 6,1.D0 /
      data X / -2.0D0, -1.0D0, -2.0D0, -1.0D0, -1.0D0, -1.0D0 /
      data B /  8.0D0,  6.0D0, -2.0D0, -1.0D0,  5.0D0, -1.0D0 /
      data ISPEC / 0, LIA, LA, 0, 3, 4, 0 /

      do 100 I = 1, N
         BB(I) = B(I)
 100  continue
c               Call DSPGE to get the solution
      call DSPGE(N, ISPEC, IA, A, B, OPT)
c             Get maximal errors in the residual and the solution.
      ERRRES = 0.D0
      SIZE = 0.D0
      K = 1
      do 280 J = 1, N
        do 270 I = K+1, K + IA(K)
          SIZE = max(SIZE, abs(A(I) * X(J)))
          BB(IA(I)) = BB(IA(I)) - A(I) * B(J)
 270    continue
        K = I
 280  continue
      do 300 J = 1, N
        ERRRES = max(ERRRES, abs(BB(J)))
 300  continue
      ERRRES = ERRRES / SIZE
      ERRX = 0.D0
      do 320 I = 1, N
        ERRX = max(ERRX, abs(X(I) - B(I)))
 320  continue

      I = int(log10(abs(OPT(2))))
      if (I .gt. 0) I = I + 1
      OPT(2) = OPT(2)*10.D0**(-I)
      I = I + nint(OPT(3))
      print '('' Problem      N    RESERR      XERR      Unused      Used'',
     1  ''     RCOND       DET    x10^?'')'
      print '('' DEMOTEST      6'',1P,2E10.2,2I8,E10.2,0PF9.5,I6)',
     1  ERRRES,ERRX,ISPEC(4),ISPEC(3)-ISPEC(4),OPT(1),OPT(2),I
      stop
      end
```

ODDSPGE

Problem	N	RESERR	XERR	Unused	Used	RCOND	DET	x10^?
DEMOTEST	6	0.00E+00	0.00E+00	196	104	5.99E−03	−4.00000	0