Guidance of library program use

(Numerical calculation: NUMPAC VOL. 1)

1. Basic matrix operations 2. System of linear equations 3. Matrix inversion

4. Eigenvalue analysis 5. Polynomial equation and nonlinear equation

22/9/92

Nagoya University Computer Center

(Supervised by: Ichizo Ninomiya)

I. NUMPAC routine

Library programs of NUMPAC are roughly divided into two cathegories, ie., function subprograms and subroutine subprograms. There are some general rules for each of them and the rules are used in this manual for simple description. Please read the following explanations carefully before using NUMPAC.

(I) Function subprogram

(1) Function name and type

The function name of the real type follows the rule of the implicit type specification of FORTRAN.

Example : BJ0, ACND

The function name of the double precision real type consists of the function name of the corresponding real type with adding D to the head of it. The function name of the quadruple precision real number type (if exists) consists of the function name of the corresponding real type with adding Q to the head of it. However, there are some exceptions.

Example : SINHP, DSINHP, QSINHP

Example of exception : ALOG1, DLOG1, QLOG1

It is severely observed that the function name for double precision begins with D and that for quadruple precision begins with Q. Note that the function name should be declared with a suitable type in each program unit referring to the function.

```
Example : DOUBLE PRECISION DCOSHP, DJ1
          REAL*8 DCELI1,DCELI2
          REAL*16 QSINHP,QASINH
```

Because the function name of double precision always begins with D and that of quadruple precision with Q, it is convenient to use the IMPLICIT statement considering other variables.

```
Example : IMPLICIT REAL*8(D)
          IMPLICIT REAL*8(A-H,O-Z)
```

In this way, you need not declare the function name, separately.

(2) Accuracy of function value

Function routines are created aiming at the accuracy of full working precision as a rule. However, this cannot be achieved completely because of fundamental or technical difficulty [1].

2

Especially, it is not achieved for functions of two variables and functions of complex variable.

(3) Limit of argument

(a) The domain is limited.

Example : ALOG1

This function calculates $log(1+x)$. Therefore, $x>-1$ should be satisfied.

(b) The singular point exists.

Example : TANHP

This function calculates $\tan \pi x/2$. Therefore, an odd integer x is a sungularity.

(c) The function value overflows.

Example : BI0

This function is for modified Bessel function $I_0(x)$, and for big x, e^x is calculated referring to standard function EXP. Therefore, overflow limit $252log_e2 \fallingdotseq 174.673$ of EXP is the upper bound of the argument of this function.

(d) The function value becomes meaningless.

Example : BJ0

This function is for Bessel function $J_0(x)$, and standard functions SIN and COS are referred to for big x. Therefore, the argument limit $|x| \leqq 2^{18}\pi \fallingdotseq 8.23 \cdot 10^5$ of SIN and COS is the limit of the argument of this function.

There are many such examples. Note that the value $2^{18}\pi$ is not a sharp limit and that the number of significant digits for the function decreases gradually as approaching this limit even if within this limit.

When the function value underflows, it is set to 0 without special processing.

(4) Error processing

When the argument exceeds the limit, an message for the error is printed and the calculation is continued with the all function values set as 0. The message consists of the function name, the argument value, the function value (0) and the reason for the error.

Example : ALOG1 ERROR ARG=-0.2000000E+01 VAL=0.0 ARG.LT.-1

The error processing program counts the frequency of the errors and stops the calculation if the frequency exceeds a certain limit, considering the case that the calculation becomes meaningless when the error occurs one after another. Because all users do not want this, you can adopt or reject this processing including the print of the message. Subroutine FNERST is

provided for this purpose and you can use it in the following way.

CALL FNERST(IABORT, MSGPRT, LIMERR)

Argument	Type and kind	Attribute	Content
IABORT	Integer type	Input	IABORT=0 The calculation is not stopped. IABORT≠0 The calculation is stopped.
MSGPRT	Integer type	Input	MSGPRT=0 The message is not printed. MSGPRT≠0 The message is printed.
LIMERR	Integer type	Input	Upper bound of frequency of errors.

If this subroutine is not called, following values are set as a standard value.

IABORT=1, MSGPRT=1, LIMERR=10

(II) Subroutine subprogram

(1) Subroutine name and type

There is no meaning of the type in the head character of the subroutine name. Subroutines with the same purpose and the different type are distinguished by the ending character of the name. The principle is as follows.

Single precision : S Double precision : D Quadruple precision : Q	Complex number : C Double precision complex number : B Quadruple precision complex number : Z	Vector computer single precision : V Vector computer double precision : W Vector computer complex number : X Vector computer double precision complex number : Y

However, there are some exceptions.

Example	Example of exception
LEQLUS/D/Q/C/B RK4S/D/Q/C/B GJMNKS/D/Q	FFTR/FFTRD MINVSP/MINVDP

(2) Argument ••• The following four kinds are distinguished as an attribute of the argument.

Input	Users should set this data before calling the subroutine. As long as it is not especially noticed, the data is preserved as it is at the subroutine exit. This includes the case when the function name and the subroutine name are used as arguments. Note that those names should be declared with EXTERNAL.

Output	This data is created in the subroutine and is significant for the user.
Input/Output	Data is output in the same place as the input to save area. When input/output argument is a single variable, you should not specify a constant as a real argument. For instance, if LEQLUS is called with constant 1 specified in input/output argument and is ended normally, IND=0 is output, but all constants 1 are changed to 0.
Work area	It is an area necessary for calculation in a subroutine, and the content of the subroutine at exit is meaningless for users.

The type and attribute of the argument are explained for each subroutine group. The explanation is for single precision. For others, please read it with exchanging the type for the suitable one.

When a subroutine is called with an argument, but the argument is not used, the area for the argument need not be prepared, and anything can be written in that place. The same area can be allocated for the different arguments, only if it is pointed as it like SVDS. There is an example (FT235R) that special demand is requested for the argument.

It is requested for users to provide the function routine and the subroutine for the numerical integration routine and the routine for solving differential equations. In this case, the number, the type, and the order of the argument should be as specified. If parameters except a regulated argument are necessary, they are allocated in COMMON area to communicate with the main program. Refer to the explanation of an individual routine for the example.

---

1) Ichizo Ninomiya; "Current state, issues of mathematical software", information processing, Vol. 23 and pp. 109-117 (1982).

【 Opening source program to the public 】

The following source programs are published for users requesting them. Calculation can be requested directly, and the source list can be output or can be copied in the shared file. The copied program cannot be given to the third party without the permission of this center.

If you need to copy the source list in the card or the data set, please execute following procedures.

(1) Input the following command for TSS.

NLIBRARY ELM (library name) "DS (data set name)" "SLAVE(ON)"

When you need only the source list, you can omit DS and SLAVE. When SLAVE(ON) is specified, all slave routines of the program will be output.

(2) Execute the following job for BATCH.

//EXEC NLIBRARY, ELM=program names[, DS='data set names'][, SLAVE=ON]

You can have examples of the program usage with the following procedures.

(1) For TSS

EXAMPLE NAME (library name) [DS (data set name)]

(2) For BATCH

//EXEC EXAMPLE, NAME=program names[, DS='data set names']

Four kinds of manual listed below are prepared concerning library program.

Number	Manual title	Content
1	Library program and data list	All library programs and data which can be used in this center are listed. Additionally, "description format of the NUMPAC routine and notes on use", "How to choose the NUMPAC routine", and usage of error processing subroutine "ERRST" are described in this list.
2	Guidance to use library program (General volume : GENERAL VOL. 1)	This volume describes the general use of programs except NUMPAC, which can be used in this center.

3	Guidance to use library program (Numerical calculation : NUMPAC VOL. 1)	This volume describes how to use the following five kinds of programs. 1. Basic matrix operations 2. System of linear equations 3. Matrix inversion 4. Eigenvalue analysis 5. Polynomial equation and nonlinear equation
4	Guidance to use library program (Numerical calculation : NUMPAC VOL. 2)	This volume describes how to use the following five kinds of programs. 6. Interpolation, smoothing, and numerical differentiation and integration 7. Fourier analysis 8. Numerical quadrature 9. Ordinary differential equation 10. Elementary function
5	Guidance to use library program (Numerical calculation : NUMPAC VOL. 3)	This volume describes how to use the following nine kinds of programs. 11. Table functions 12. Orthogonal polynomial 13. Special functions 14. Bessel function and related function 15. Acceleration of convergence of sequences 16. Linear programming 17. Special data processing 18. Figure display application program 19. Others

All these manuals can be output by "MANUAL command". "PICKOUT command" is available if you need part of the usage of individual program.

For NUMPAC users

Please note the following and use NUMPAC effectively.

(1) The user has the responsibility for the result obtained by NUMPAC.

(2) When the trouble is found, please report it to the center program consultation corner (Extension 6530).

(3) Do not use NUMPAC in computer systems other than this center without permission.

(4) To publish the result obtained NUMPAC, the used program names (for instance, *** of NUMPAC) should be referred to.

This manual was translated using Fujitsu's machine translation system ATLAS.

II. Library and program itemized discussion

1. Basic matrix operations

ADDMMV/W/X/Y and SUBMMV/W/X/Y (Addition and Subtraction of Matrices-Vector Version)

Addition and Subtraction of Matrices-Vector Version

Programmed by	Ichizo Ninomiya, July 1987
Format	Subroutine Language: FORTRAN; Size: 70 lines

(1) Outline

The ADDMMV(W, X, Y) and SUBMMV(W, X, Y) calculate the sum $C=A+B$ and difference $C=A-B$ of the two matrices A and B. They are for the single precision real numbers (double precision real number, single precision complex number, and double precision complex number).

(2) Directions

CALL ADDMMV/W/X/Y(A, B, C, KA, KB, KC, M, N, ILL)
CALL SUBMMV/W/X/Y(A, B, C, KA, KB, KC, M, N, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	M×N matrix A
B	Real type Two-dimensional array	Input	M×N matrix B
C	Real type Two-dimensional array	Output	M×N matrix C. A+B or A-B
KA	Integer type	Input	Adjustable dimensions of A. KA≧M
KB	Integer type	Input	Adjustable dimensions of B. KB≧M
KC	Integer type	Input	Adjustable dimensions of C. KC≧M
M	Integer type	Input	Number of rows of A, B, and C. M≧1

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Number of columns of A, B, and C. N≧1
ILL	Integer type	Output	ILL=0: normal termination; ILL=30000: argument error

*1 For ADDMMW(X,Y) and SUBMMW(X,Y), all real types should be changed to double precision real types (complex type and double precision complex type).

(3) Note

1. This routine is for vector computers. However, it can be used also for scalar computers.

(1987.09.18)

MDETS/D/Q/C/B/Z (Calculation of Determinants)

Calculation of Determinants

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 45, 34, 45, 34, and 35 lines respectively

(1) Outline

MDETS/D/Q/C/B/Z calculates the determinant of a given matrix.

(2) Directions

CALL MDETS/D/Q/C/B/Z(A, KA, N, EPS, D, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Matrix whose determinants should be calculated. Destroyed
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Degree of AN≧2
EPS	Real type	Input	Criterion constant for matrix singularity. If the absolute value of pivot elements is smaller than this constant, D = 0 is assumed. EPS>0
D	Real type	Output	The value of determinant is output.
ILL	Integer type	Output	ILL = 3000: Limits on KA, N, and EPS are violated. Otherwise, 0 is output.

*1 For MDETD (Q, C, B, Z), A and D are double precision real types (quadruple precision real type, complex type, double precision complex type, and quadruple precision complex type).

For (Q, C, B, Z), EPS is a double precision real type (quadruple precision real type, real type, double precision real type, and quadruple precision real type).

(3) Performance

Precision depends on problems. Because the LU-decomposition method (Doolittle method) is used, and double precision arithmetic operation is performed to calculate the inner products in MDETS, precision is high. The required computation time is almost the same as that for solving a system of linear equations.

(4) Remarks

1. If the typical size of elements in the matrix A is assumed to be a, the standard value of EPS is $a\times10^{-6}$($a\times10^{-16}$,$a\times10^{-30}$) for MDETS (MDETD).

2. When a system of linear equations is to be solved, and the determinant is to be calculated at the same time, it is recommended to use LEQLUS and LEQLUD.

(1987. 06. 17) (1987. 08. 07)

MNORMS/D/Q/C/B/Z (Normalization of a Matrix)

Normalization of a Matrix

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 20, 21, 20, 23, 24, and 24 lines respectively

(1) Outline

MNORMS/D/Q/C/B/Z divides each row of a given matrix by a number of the form of 2^n to limit the maximum absolute value of elements in each row to the order of 1.

(2) Directions

CALL MNORMS/D/Q/C/B/Z(A, KA, N, M, S, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Matrix to be normalized
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Number of rows in A. N≧2
M	Integer type	Input	Number of columns in A. M≧N
S	Real type One-dimensional array	Output	S (I) (I=1, ···, N) contains a divisor in the form of power of 2 to normalize the row I.
ILL	Integer type	Output	ILL=0: Normal termination ILL=30000: Limits on KA, N, and M are violated. The row number whose elements are all 0

*1 For MNORMD (Q, C, B, Z), A is assumed to be a double precision real type (quadruple precision real number, complex number, double precision complex number, and quadruple precision complex number), and S is assumed to be a double precision (quadruple precision, single precision, double precision, and quadruple precision) real type.

(3) Remarks

1. When normalization is to be done as preprocessing for solving a system of linear equations, merge the right side column in the right of the coefficient matrix, and apply this routine to the augmented matrix. The solution obtained by solving the normalized equation is the solution of the original equation. That is, postprocessing is not required.

2. The inverse matrix of the original matrix is obtained by dividing each I-th column of the inverse matrix of the normalized matrix by the scale factor S (I).

3. In general, normalization changes a symmetric matrix to an asymmetric matrix.

4. For symmetric positive definite matrices, the special-purpose routine such as MNRSPS should be used.

(1987. 06. 17) (1987. 08. 07)

MNRMBS/D/Q/C/B/Z and MNMBSS/D/Q (Normalization of Band Matrices)

Normalization of Band Matrices

Programmed by	Ichizo Ninomiya, May 1982
Format	Subroutine language; FORTRAN77 Size; 25, 26, 26, 26, 27, 27, 25, 26, and 26 lines respectively

(1) Outline

The general band matrix subroutine MNRMBS/D/Q/C/B normalizes each row of a given band matrix by dividing it by a power of 2 that is close to the maximum absolute value of the row.

The symmetric positive definite band matrix subroutine MNMBSS/D/Q normalizes each row and column of a given band matrix by dividing it by a power of 2 that is close to the square root of the diagonal element.

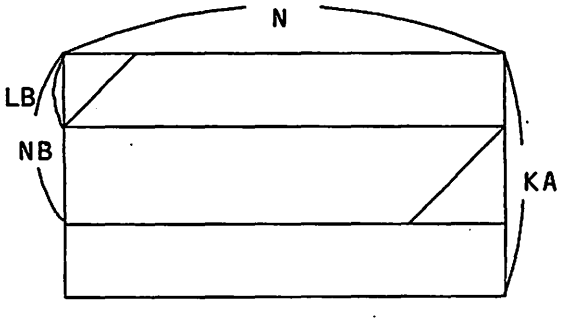


図 1

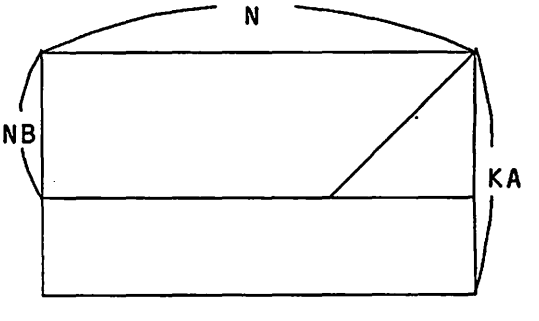


図 2

(2) Directions

CALL MNRMBS/D/Q/C/B/Z(A, KA, N, NB, LB, S, ILL)

CALL MNMBSS/D/Q (A, KA, N, NB, S, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Number of real numbers Two-dimensional array	Input/output	Matrix to be normalized. General matrices are transformed into a rectangular form as shown in Figure 1. That is, the (I, J) elements are stored in A (J - 1 + LB, I). Positive definite symmetric matrices are transformed into a rectangular form as shown in Figure 2. That is, the (I, J) elements are stored in A (I - J + 1, J).
KA	Integer type	Input	Adjustable dimension of A (first subscript in array declaration). KA≧NB
N	Integer type	Input	Degree of A (number of columns). N≧NB
NB	Integer type	Input	Entire band width for general matrices. NB≧LB Half band width for symmetric positive definite matrices. NB≧1
LB	Integer type	Input	Left band width of A. LB≧1
S	Real type One-dimensional array	Output	Normalization factor. Real number of the form of power of 2 used to divide each row (column). One-dimensional array of size N
ILL	Integer type	Output	ILL=0: Normal termination. ILL=K: Normalization is interrupted at Kth step. General: The Kth line is all zero. Symmetric positive definite matrix: The Kth diagonal element is not positive. ILL=30000: Argument error.

*1 For MNRMBD (Q, C, B, Z), A is assumed to be a double precision real type (quadruple precision real type, complex type, double precision complex type, and quadruple precision complex type), and S is assumed to be double precision (quadruple precision, single precision, double precision, and quadruple precision) real type.

For MNMBSD (Q), real types are changed to double (quadruple) precision real types.

(3) Example of use

1. Example of MNRMBS

A quindiagonal matrix (NB = 5, LB = 3) equation (N = 1000) is solved with LEQBDS, after normalizing it with MNRMBS. All diagonal elements are put as $a_{jj}=5j$ and non-diagonal elements as 1, and constant terms are set so that all elements of the solution are 1.

```
DIMENSION A(7,1000),S(1000),X(1000),MAX(1000)
```

```
      N=1000
      KA=7
      NB=5
      LB=3
      EPS=1.E-6
      DO 10 J=1,N
      DO 20 I=1,5
   20 A(I,J)=1.0
      A(3,J)=J*5
      X(J)=A(3,J)+4.0
      IF(J.LE.2) X(J)=X(J)-FLOAT(3-J)
      IF(J.GE.N-1) X(J)=X(J)-FLOAT(J+2-N)
   10 CONTINUE
      CALL MNRMBS(A,KA,N,NB,LB,S,IND)
      DO 25 I=1,N
   25 X(I)=X(I)/S(I)
      IND=0
      CALL LEQBDS(A,KA,N,NB,LB,MB,X,N,1,MAX,EPS,IND)
      EM=0.0
      DO 30 I=1,N
      EM=AMAX1(ABS(X(I)-1.0),EM)
   30 CONTINUE
      WRITE(6,600) EM
  600 FORMAT(10X,E11.3)
      STOP
      END
```

2. Example of MNMBSS

An equation having a positive definite symmetric band matrix (N = 1000, NB = 5) as a coefficient is solved by CHLBDS after normalizing it by MNMBSS. All diagonal elements are put as $a_{jj}=10j$ and non-diagonal elements as 1, and all constant terms are set so that all elements of the solution are 1.

```
      DIMENSION A(5,1000),S(1000),X(1000)
      N=1000
      KA=5
      NB=5
      EPS=1.E-6
      DO 10 J=1,N
      A(1,J)=1*10
      DO 20 I=2,5
   20 A(I,J)=1.0
      X(J)=A(1,J)+8.0
      IF(J.LE.4) X(J)=X(J)-FLOAT(5-J)
      IF(J.GE.N-3) X(J)=X(J)-FLOAT(J+4-N)
   10 CONTINUE
      CALL MNMBSS(A,KA,N,NB,S,IND)
      DO 25 I=1,N
   25 X(I)=X(I)/S(I)
      IND=0
      DET=0.
      CALL CHLBDS(A,KA,N,NB,X,N,1,DET,EPS,IND)
      DO 27 I=1,N
   27 X(I)=X(I)/S(I)
      EM=0.0
      DO 30 I=1,N
      EM=AMAX1(ABS(X(I)-1.0),EM)
```

```
 30 CONTINUE
    WRITE(6,600) EM
600 FORMAT(10X,E11.3)
    STOP
    END
```

(4) Remarks

1. When this routine is used to solve a system of linear equations that has a band matrix as a coefficient, each element of the right side constant vector must be divided by the corresponding normalization factor before the e simultaneous linear equation routine is called. For general matrices, nothing need not be done after a solution is obtained. For a symmetric positive definite matrix, however, each element of the solution must be divided by the corresponding normalization factor. (See the example of use.)

(1987. 06. 17) (1987. 08. 07)

MNRSPS/D/Q (Normalization of a Symmetric Positive Definite Matrix)

Normalization of a Symmetric Positive Definite Matrix

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 20, 21, and 21 lines respectively

(1) Outline

MNRSPS/D/Q limits the maximum absolute value of elements in each row and column to the order of 1 by dividing each row and column of a given symmetric positive definite matrix by a number of the form of 2^n preserving symmetric positive definiteness.

(2) Directions

CALL MNRSPS/D (A, KA, N, M, S, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Matrix to be normalized. Only the upper right half including the diagonal lines is processed. Other part is preserved.
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Number of rows in A. N≧2
M	Integer type	Input	Number of columns in A. M≧N
S	Real type One-dimensional array	Output	S (I) (I = 1, 2, ···, N) contains a divisor in the form of power of 2 to normalize the row I and column I.
ILL	Integer type	Output	ILL=0: Normal termination ILL=30000: Limits on KA, N, and M are violated. The row number of the diagonal element which is not positive

*1 For MNRSPD (Q), all real types are changed to double (quadruple) precision types.

(3) Remarks

1. When normalization is done as a preprocessing for solving a system of linear equations,

every element of solution of normalized equation should be divided by the corresponding scale factor.

2. When normalization is done as a preprocessing for obtaining inverse matrix, every row and column of the inverse of normalized matrix should be divided by the corresponding scale factors.

(1987. 06. 17) (1987. 08. 08)

MULMMV/W/X/Y (Multiplication of Matrices-Vector Version)

Multiplication of Matrices-Vector Version

Programmed by	Ichizo Ninomiya, July 1987
Format	Subroutine Language: FORTRAN; Size: 80 lines

(1) Outline

MULMMV/W/X/Y calculates the product $C=A\cdot B$ of two matrices A and B. MULMMV(W, X, Y) is for the single precision real numbers (double precision real number, single precision complex number, and double precision complex number).

(2) Directions

CALL MULMMV/W/X/Y(A, B, C, KA, KB, KC, L, M, N, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	L×M multiplicand matrix A
B	Real type Two-dimensional array	Input	M×N multiplier matrix B
C	Real type Two-dimensional array	Output	L×N product matrix C
KA	Integer type	Input	Adjustable dimensions of A. KA≧L
KB	Integer type	Input	Adjustable dimension of B. KB≧M
KC	Integer type	Input	Adjustable dimensions of C. KC≧L
L	Integer type	Input	Number of rows of A and C. L≧1

Argument	Type and kind (*1)	Attribute	Content
M	Integer type	Input	Number of columns of A and rows of B. M≧1
N	Integer type	Input	Number of columns of B and C. N≧1
ILL	Integer type	Output	ILL=0: normal termination; ILL=30000: argument error

*1 For MULMMW(X,Y), all real types should be changed to double precision real types (complex type and double precision complex type).

(3) Calculation method

The product of $l \times m$ matrix A and $m \times n$ matrix B is an $l \times n$ matrix.

It is

$$C_{ij}=\sum_{k=1}^{m} a_{ik}b_{kj}, i=1,\cdots,l; j=1,\cdots,n$$

. If A and C are considered as sets of column vectors $A=(a_1,a_2,\cdots,a_m)$ and $C=(c_1,c_2,\cdots,c_n)$ respectively, then C_j can be written as

$$C_j=\sum_{k=1}^{m} b_{kj}a_k, j=1,2,\cdots,n$$

The algorithm of this subroutine is based on this idea.

(4) Note

1. The product of a matrix and a vector can be calculated with this routine with B assumed as a single column matrix. However, it is more reasonable to use the special routine MULMVV.
2. This routine is for vector computers. However, it can be used also for scalar computers.

(1987.08.04)

MULMVV/W/X/Y (Multiplication of a Matrix and a Vector-Vector Version)

Multiplication of a Matrix and a Vector-Vector Version

Programmed by	Ichizo Ninomiya, July 1987
Format	Subroutine Language: FORTRAN; Size: 70 lines

(1) Outline

MULMVV/W/X/Y calculates the product $y=Ax$ of a matrix A and a vector x. MULMVV(W, X, Y) is for the single precision real numbers (double precision real number, single precision complex number, and double precision complex number).

(2) Directions

CALL MULMVV/W/X/Y(A, X, Y, KA, M, N, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	M×N matrix A
X	Real type One-dimensional array	Input	N vector x
Y	Real type One-dimensional array	Output	M vector y
KA	Integer type	Input	Adjustable dimensions of A. KA≧M
M	Integer type	Input	Number of rows of A and order of y. N≧1
N	Integer type	Input	Number of columns of A and order of x. M≧1
ILL	Integer type	Output	ILL=0: Normal termination; ILL=30000: Argument error

*1 For MULMVW(X, Y), all real types should be changed to double precision real types (complex type and double precision complex type).

(3) Calculation method

If the matrix A is considered as a set of the column vector $(a_1, a_2, \cdots, a_m)$, $y=Ax$ can be written as

$$y=\sum_{k=1}^{m} x_k a_k$$

The algorithm of this subroutine is based on this idea.

(4) Note

This routine is for vector computers. However, it can be used also for scalar computers.

(1987. 08. 04)

2. System of linear equations

[Method of choice of linear equation routines]

NUMPAC provides a variety of effective linear equation subroutines that you can select depending on the type, characteristics, and structure of each coefficient matrix. By carefully selecting them based on the guideline shown below, you can enjoy much of their superiority in all aspects of precision, speeds, and storage capacities. To make the following explanation simple, the name of each recommended routine is represented by the one for single precision. The routine marked by * is written in assembly language and recommended specially.

(A) Real coefficient

1. Non-symmetry
   - (1) Dense matrix — LEQLUS*
   - (2) Band matrix — LEQBDS
   - (3) Tridiagonal matrix — TRIDGS
2. Symmetry
   - (1) Dense matrix — BUNCHS
   - (2) Band matrix — BUNCBS
3. Positive-definite symmetric
   - (1) Dense matrix — CHOLFS*, MCHLFS*
   - (2) Band matrix — CHLBDS*, MCHLBS*
   - (3) Variable width band matrix — CHLVBS
   - (4) Tridiagonal matrix — TRDSPS, TDSPCS
4. General system of linear equations — LEQLSS, LSMNS

(B) Complex coefficient

1. Dense matrix — LEQLUC
2. Band matrix — LEQBDC

If there is a great difference between coefficients and between solutions of linear equations, satisfactory precision is not generally expected. It is important to level the coefficients and solutions in advance by means of normalization or variable transformation.

Many users seem to use an inverse matrix routine to solve linear equations because the solution of linear equations $Ax=b$ is theoretically written as $X=A^{-1}b$. However, they shouldn't do this

because it takes three times for calculation that taken by a linear equation routine and the accuracy of the solution is remarkably worse.

To repeat solving equations with the same coefficients by changing the right-hand side column only, it seems reasonable to calculate $x_i=A^{-1}b_i, i=1,2,---$ by calculating an inverse matrix only once. For this, however, it is far more advantageous to exploit the function of reuse of decomposition component available in all Nagoya University routines.

In short, it is nothing but the abuse of inverse matrixes to use them to solve linear equations.

BUNCBS/D (Solution of linear equations with symmetric band matrix of coefficients by bunch's method)

Solution of Linear Equations with Symmetric Band Matrix of Coefficients by Bunch's Method

Programmed by	Ichizo Ninomiya; April 1981
Format	Subroutine language; FORTRAN Size; 200 lines each

(1) Outline

BUNCBS and BUNCBD are single or double precision subroutines used to obtain the following solution using the bunch's LDL^T decomposition method: The solution obtained by the subroutines is $X=A^{-1}B$ of the linear equations $AX=B$ with right-hand side matrix B, and symmetric band matrix A which is not necessarily positive definite as a coefficient.
These subroutines have the function of the reuse of the LDL^T decomposition component.

(2) Directions

CALL BUNCBS/D(A, KA, N, NB, X, KX, M, CHG, EPS, IW, IND)

Argument	Type and Kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	The left lower half containing the diagonal of the symmetric band matrix is transformed to a rectangle as shown in the figure and input. That is, the I, J elements of the matrix are put in A(I-J+1, J). After processing by this routine, the Bunch's decomposition component is output.
KA	Integer type	Input	Adjustable dimension of A (value of the first subscript in array declaration of A). The band width of A will generally increase by pivoting. It is therefore necessary to make KA large enough to meet it. KA≧NB

Argument	Type and Kind (*1)	Attribute	Content
N	Integer type	Input	Number of unknowns in the equation (number of columns of A). N≧1
NB	Integer type	Input/output	Half band width of A (number of rows) is input. Half band width after processing is output. NB≧2
X	Integer type Two-dimensional array	Input/output	The right hand side matrix is input. The solution matrix is generated to the corresponding place.
KX	Integer type	Input	Adjustable dimension of X. KX≧N
M	Integer type	Input	Number of columns in X. When M≦0, only decomposition of A is done .
CHG	Real type One-dimensional array	Output	One-dimensional array of size N or greaterInformation on pivoting and the determinant of the 2×2 diagonal block are generated.
EPS	Real type	Input	When the size of the pivot element becomes smaller than $\|\|A\|\| \cdot EPS$ during decomposition, the coefficient matrix is assumed to be singular and then calculation is interrupted. When EPS≦0.0 is given, default value u is used, where $u=2^{-20}$ (single precision) and $u=2^{-52}$ (double precision).

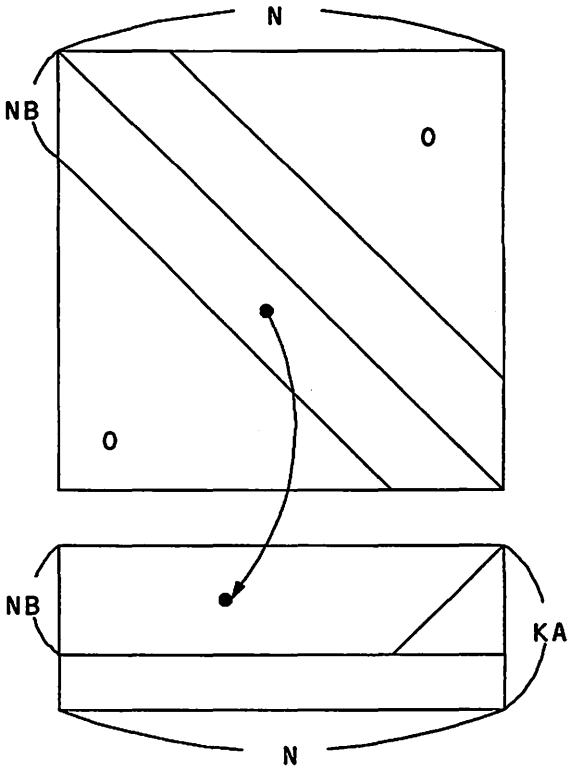
Argument	Type and Kind (*1)	Attribute	Content
IW	Integer type One-dimensional array	Work area	One-dimensional array of size N
IND	Integer type	Input/output	For input, this argument has the following meanings: IND=0: The equation is solved by restarting Bunch's decomposition from the beginning. IND≠0: The equation is solved by using the decomposition component calculated immediately before. To do this, the contents of A and CHG must have been stored. For output, this argument has the following meanings: IND=0: Normal end IND=K: Judged as singular at step K of decomposition or band width exceeded KA. IND=30000: The input argument violated the limit.

*1 For double precision subroutines, real types should be changed to double precision real types.

(3) Calculation method

When coefficient matrix A is a symmetric positive definite, modified Cholesky decomposition $A=LDL^T$ is possible using unit lower triangular matrix L and diagonal matrix D. However, when A is not positive definite, decomposition is generally impossible even if it is symmetric.

However, if D is assumed to be a block diagonal matrix for which the submatrix of 2×2 is permitted as a diagonal block element, decomposition above is possible. Bunch designed an algorithm to perform decomposition $A=LDL^T$ in a numerically stable manner by exchanging rows and columns properly. [1),2)] This routine is based on Bunch's algorithm D.



By using this decomposition, solution $X=A^{-1}B$ of $AX=B$ can be determined by forward substitution $Y=L^{-1}B$ and backward substitution $X=L^{-T}D^{-1}Y$.

(4) Notes

1. The standard value of EPS for BUNCBS or BUNCBD is $10^{-6}(10^{-16})$. If EPS≦0.0 is given, default value $2^{-20}(2^{-52})$ will be used.

2. Argument IND is used for both input and output. Therefore, do not use a constant as an actual argument.

3. The routine's function of reusing decomposition components is very useful to repeatedly solve the equations with the same coefficient matrix and different right hand side matrices. It is superior to the inverse-matrix method in all aspects of accuracy, speed, and storage capacity.

4. When M, the number of columns of B, is 1, a one-dimensional array is acceptable for the actual argument corresponding to X. For this, however, it is necessary to meet the condition KX≧N.

Bibliography

1) J. R. Bunch et al. ; "Decomposition of a Symmetric Matrix " Numer. Math., Bd. 27, pp. 95-109 (1976).

2) J. R. Bunch et al. ; "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems", Math. Comp., Vol. 31, No. 137, pp. 163-179 (1977).

(1987. 06. 16)

BUNCHS/D (Solution of Linear Equations with Symmetric Matrix of Coefficient by Bunch's Method)

Solution of Linear Equations with Symmetric Matrix of Coefficients by Bunch's Method

Programmed by	Ichizo Ninomiya, April 1981
Format	Subroutine language: FORTRAN; size: 200, 200 lines respectively

(1) Outline

BUNCHS (D) is a single (double) precision subroutine for finding $X=A^{-1}B$ or the solution of the simultaneous linear equation $AX=B$ with a symmetric matrix A (not necessarily positive definites) and multiple right side columns B, using the Bunch's U^TDU decomposition method. It has the facility for reusing the U^TDU decomposition elements.

(2) Directions

CALL BUNCHS/D(A, KA, N, X, KX, M, CHG, EPS, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	The upper right half including the diagonal of the coefficient matrix is input. The upper right half is processed with this routine, and the Bunch decomposition elements are output. The lower left half is preserved.
KA	Integer type	Input	Adjustable dimension of A (value of the first subscript in array declaration). KA≧N
N	Integer type	Input	Order of equation. N≧1

Argument	Type and kind (*1)	Attribute	Content
X	Real type Two-dimensional array	Input/output	The right side columns are input. The solution vectors are output in the corresponding place.
KX	Integer type	Input	Adjustable dimension of X. KX≧N
M	Integer type	Input	Number of columns of X. If M≦0, only A is decomposed.
CHG	Real type One-dimensional array	Output	One-dimensional array of size N or greater. Pivoting information and 2 × 2 diagonal block determinants are output.
EPS	Real type	Input	If the size of pivot elements becomes smaller than $\|\|A\|\| \cdot EPS$ during decomposition, the coefficient matrix is decided to be singular, and the calculation is interrupted. If EPS≦0.0 is assigned, the standard value u is used, where $u=2^{-20}$ (single precision) and $u=2^{-52}$ (double precision).
IND	Integer type	Input/output	This argument has the following meaning as an input argument. IND = 0: Solve equation newly starting with Bunch's decomposition. IND ≠ 0: Solve equations, reusing the decomposition elements previously calculated. In this case, A and CHG must be kept unchanged in the states of previous call.

Argument	Type and kind (*1)	Attribute	Content
			This argument has the following meaning as an output argument. IND = 0: Normal termination IND = K: Singularity is decided at Kth step of the decomposition IND = 30000: The input argument violates the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

If the coefficient matrix A is symmetric positive definites, a modified Cholesky decomposition $A=U^TDU$ is possible with an upper unit triangular matrix U and a diagonal matrix D. If the matrix A is not of positive definites even though it is symmetric, the decomposition is generally impossible. However, if D is assumed to be a diagonal block matrix that permits a 2 x 2 submatrix as a diagonal block element, the similar decomposition is possible. Bunch designed a algorithm for calculating the decomposition $A=U^TDU$ with numerical stability by properly interchanging rows and columns. [1),2)]

This routine is based on Bunch's algorithm A.

If this decomposition is applied, the solution $X=A^{-1}B$ of $AX=B$ is found by the forward substitution $Y=U^{-T}B$ and backward substitution $X=U^{-1}D^{-1}Y$.

(4) Remarks

1. The standard value of EPS is $10^{-6}(10^{-16})$ for BUNCHS (D). If EPS≦0.0 is given, the standard value $2^{-20}(2^{-52})$ is used.

2. Because IND is an input/output argument, a constant must not be used as an actual argument.

3. When a solution to the same coefficient matrix is to be repeatedly found changing only the right side columns, the facility for reusing the decomposition components of this routine is extremely useful. As compared with the method by inverse matrices, this calculation method is

excellent in precision, speed, and storage size.

4. If the number of right side columns (M) is 1, an actual argument that corresponds to X can be a one-dimensional array. However, KX≧N must be met.

References

1) J. R. Bunch et al. ;"Decomposition of a Symmetric Matrix"Numer. Math., Bd. 27, pp. 95-109 (1976).

2) J. R. Bunch et al. ;"Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems" and Math. Comp . Vol. 31, No. 137, and pp. 163-179 (1977).

(1987. 06. 16)

CGHTCS/D Solution of a Linear System of Equations with Positive Definite Symmetric Coefficients Matrix by Conjugate Gradient Method (Compressed matrix storage Mode)

Solution of a Linear System of Equations with Positive Definite Symmetric Coefficients Matrix by Conjugate

Gradient Method(Compressed matrix storage mode)

Programmed by	Tsuyako Miyakoda and Tatsuo Torii, February 1982
Format	Subroutine language: FORTRAN; size: 55 and 56 lines respectively

(1) Outline

This is a solution routine with a conjugate gradient method if the upper triangular and diagonal elments of the matrix are stored in a row to set up the storage arrays, where the symmetric positive definite matrix A is a coefficient matrix .

(2) Directions

CALL CGHTCS/D(A, NA, N, B, X, EPS, NMAX, W, IDUMP)

Argument	Type and kind	Attribute	Content
A	Real type One-dimensional array	Input	The upper triangular and diagonal elements of the matrix are stored in a one-dimensional array. Element (I, J) ($I \leq J$) is assumed to be (J*(J-1)/2+I)th element of a one-dimensional array. $A(k)=a_{ij},\ k=j(j-1)/2+i$
NA	Integer type	Input	Length of the vector when the coefficient matrix is made into a one-dimensional array.
N	Integer type	Input	Number of unknowns of the system.

Argument	Type and kind	Attribute	Content
B	Real type One-dimensional array	Input	Right-side vector of the system.
X	Real type One-dimensional array	Input/output	Input: Approximative solution vector (initial value). Output: Corrected solution vector.
EPS	Real type	Input	Convergence criterion. It is assumed to be $8 \cdot u \cdot \|\| b \|\|$ as external page storage if it is too small. u is a unit of the rounding error.
NMAX	Integer type	Input	Maximum number of iterations. When a too large value is input, it is assumed to be $3 \cdot N/2$.
W	Double precision real type One-dimensional array	Work area	Size $N \times 3$.
IDUMP	Integer type	Input/output	It has the following meaning as an input argument. IDUMP<0: No printing of the result on the way. IDUMP=1: Printing of residual (P, AP) of each iteration. IDUMP>2: Printing of residual, A - orthogonal set vectors, and approximative solution of each iteration. It has the following meaning as an output argument. The same as input: Normal termination. IDUMP=3*N : Not settled even for 3*N iterations,. IDUMP=30000: Input parameter error.

* All real types are assumed to be a double precision real type for CGHTCD.

(3) Calculation method

CGHTCS/D finds the solution of a linear system of equations with the symmetric positive definite matrix, $Ax=b$, so that the error function $\varphi(x)=(r,A^{-1}r)$ is minimized if the residual $r=b-Ax$ is assumed. In the conjugate gradient method, both sides of a correction vector (A-orthogonal system) calculation formula of the original version is divided by $|r_{i+1}|^2$, and normalized as a type of formula (Takahashi version). The calculation formula becomes simple, but the speed of residual reduction and the computation time are not much changed. The calculation expression is as follows:

Initial value $x_0=0$, $r_0=b-Ax_0$, $P_0=r_0/|r_0|^2$

$\alpha_i=1/(P_i,AP_i)$

$x_{i+1}=x_i+\alpha_i P_i$

$r_{i+1}=r_i-\alpha_i AP_i$

Convergence decision $|r_{i+1}|^2<(EPS)^2$?

$P_{i+1}=P_i-r_{i+1}/|r_{i+1}|^2$

$i=0,1,\cdots\cdots\cdots$

(4) Example

```
C       MAIN FOR CGHTCS
        DIMENSION AS(5050),X(100),B(100),A(100),X0(100)
        DOUBLE PRECISION W(300),SU
        NR=5
        NW=6
        EPS=0.1E-4
        XX=0.1E8+1.
        N=100
        XI=12345678.00
        DO 10 I=1,N
        X0(I)=0.0
        X(I)=4.*XI/1.E8-2.
        XI=AMOD(23.*XI,XX)
   10   CONTINUE
        NT=0
        DO 40 I=1,N
        SU=0.0D0
        DO 20 J=1,N
        IJ=IABS(I-J)
        A(J)=FLOAT(N-IJ)
   20   SU=A(J)*X(J)+SU
        B(I)=SU
        DO 30 J=1,I
        NT=NT+1
```

```
   30 AS(NT)=A(J)
   40 CONTINUE
      WRITE(NW,1000)N
 1000 FORMAT(1H1,15H EXAMPLE 3-6 N=,I4)
      IDUMP=0
      MAXN=100
      CALL CLOCKM(JTIME1)
      CALL CGHTCS(AS,NT,N,B,XO,EPS,MAXN,W,IDUMP)
      CALL CLOCKM(JTIME2)
      JTIME=JTIME2-JTIME1
      WRITE(6,1010) IDUMP,JTIME
 1010 FORMAT(1H ,' IDUMP =',I5,3X,'TIME(MSEC)=',I5)
      DO 50 I=1,N
      RES=X(I)-XO(I)
      WRITE(NW,1020) I,X(I),XO(I),RES
 1020 FORMAT(I5,2E13.5,E11.3)
   50 CONTINUE
 1030 FORMAT((1H ,5(1PE13.5)))
      STOP
      END
```

```
      EXAMPLE 3-6 N= 100
     IK=  26 ZANSA=    0.53581611D-02
      IDUMP =    0   TIME(MSEC)=  565
        1 -0.15062E+01 -0.15031E+01 -0.303E-02
        2 -0.18420E+01 -0.18510E+01  0.899E-02
        3 -0.19655E+01 -0.19646E+01 -0.883E-03
        4 -0.16055E+01 -0.16043E+01 -0.119E-02
```

(5) Note

The conjugate gradient method is characterized by fast convergence if a coefficient matrix is large in the number of dimensions and sparse. For dense coefficient matrices, it is desirable to use other methods or the conjugate gradient method that includes preprocessing (PRCGFS/D).

References

1) Hayato Togawa: Conjugate Gradient Method, Kyoiku Shuppan, 1977

(1987. 06. 16) (1987. 08. 07) (1987. 08. 10)

CHLBDC/B/Z, MCHLBC/B/Z

(Solution of Hermitian Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Band Matrix))

Solution of Hermitian Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Band Matrix)

Programmed by	Ichizo Ninomiya, December 1983
Format	Subroutine language: FORTRAN; size: 63, 64, 64, 70, 71, and 71 lines respectively

(1) Outline

CHLBDC(B, Z) (MCHLBC(B, Z)) is a single (double or quadruple) precision subroutine for obtaining the solution $X=A^{-1}B$ of the equation $AX=B$ having a Hermitian positive definite band matrix A as coefficient matrix and multiple right side columns B, using modified Cholesky decomposition method. It reuses Cholesky decomposition component.

(2) Directions

```
        CHLBDC/B/Z
CALL                (A, KA, N, NB, X, KX, M, DET, EPS, IND)
        MCHLBC/B/Z
```

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input/output	The lower left half band area containing the diagonal of a coefficient matrix is transformed into a rectangular form and input. That is, the I and J elements of the matrix are input in A(I-J+1, J). These elements are processed with this routine, and Modified Cholesky decomposition elements are output. See the figure.
KA	Integer type	Input	Adjustable dimensions of A (value of the first subscript in the array declaration). KA≧NB

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Order of equations (number of columns of A). N≧1
NB	Integer type	Input	Band width (number of rows of A). 1≦NB≦N
X	Complex type Two-dimensional array	Input/output	The right side columns are input. The solution vectors are output to the corresponding positions.
KX	Integer type	Input	Adjustable dimensions of X. KX≧N
M	Integer type	Input	Number of columns of X. If M≦0, only modified Cholesky decomposition is executed.
DET	Real type	Input/output	If DET≠0.0 is input, coefficient matrix determinants are output. If DET=0.0 is input, 0.0 is output.
EPS	Real type	Input	Coefficient matrix positivity criterion. If the value of a diagonal element becomes smaller than EPS during Cholesky decomposition, it is decided to be not positive definite, and the computation is interrupted. EPS>0

Argument	Type and kind (*1)	Attribute	Content
IND	Integer type	Input/output	This argument has the following meaning as an input argument. IND=0: Equation is solved newly beginning with Cholesky decomposition. IND≠0: Equation is solved reusing the Cholesky decomposition component computed before. This argument has the following meaning as an output argument. IND=0: Computation is normally executed. IND=K: Because the value of a diagonal element becomes smaller than EPS at the K-th step of Cholesky decomposition, computation is interrupted. IND=30000: The input argument exceeded the limit.

1* For double (quadruple) precision subroutines, all single precision types are changed to double (quadruple) precision types.

(3) Calculation method

1. Cholesky decomposition method

The coefficient matrix A is decomposed into $A=LL^*$ with a lower triangular matrix L and its transposition conjugate matrix L^*. The solution $X=A^{-1}B$ is obtained with the forward substitution $Y=L^{-1}B$ and backward substitution $X=(L^*)^{-1}Y$.

2. Modified Cholesky decomposition method

The coefficient matrix A is decomposed into $A=LDL^*$ with a lower unit triangular matrix L, its transposition conjugate matrix $A=LDL^*$, and a diagonal matrix D. The solution $X=A^{-1}B$ is obtained with the forward substitution $Y=L^{-1}B$ and backward substitution $X=(L^*)^{-1}D^{-1}Y$.

(4) Notes

1. If the typical size of coefficient matrix elements is a, the value $10^{-6}a(10^{-16}a, 10^{-30}a)$

is adequate as the standard value of EPS for {MCHLBS(D,Q) CHLBDS(D,Q)}.

2. Because DET and IND are I/O arguments, constants must not be used as an actual argument. Note that DET is a real type.

3. When a solution to the same coefficient matrix is to be repeatedly obtained with only the right side column changed, the function that reuses the Modified Cholesky decomposition elements of this routine is particularly useful. It is more efficient in all of storage size, precision, and speed as compared with the method using the inverse matrix.

4. If the number M of right side columns is 1, the real argument that corresponds to X can be a one-dimensional array. However, KX≧N must be met.

(1987.06.19) (1987.08.07)

CHLBDS/D/Q,MCHLBS/D/Q

(Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method) (Band Matrix)

Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Band Matrix)

Programmed by	Ichizo Ninomiya in April 1981
Format	Subroutine language:Assembler (CHLBDQ and MCHLBQ FORTRAN) Size:233, 239, 64, 202, 199, and 71 lines respectively

(1) Outline

CHLBDS (D, Q) (MCHLBS (D, Q)) is a single (double or quadruple) precision subroutine that finds the solution $X=A^{-1}B$ of the equation $AX=B$ with a symmetric positive definite band matrix A as a coefficient matrix and multiple right sides B, using modified Cholesky decomposition method. It has the facility for reusing Cholesky decomposition components.

(2) Directions

CALL [CHLBDS/D/Q / MCHLBS/D/Q] (A, KA, N, NB, X, KX, M, DET, EPS, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type two-dimensional array	Input/output	Transform the lower left half including the diagonal of the coefficient matrix into a rectangular form, that is, the I and J element of the matrix is stored in A(I-J+1, J). The array is processed by this routine, and modified Cholesky decomposition components are output. See the figure.
KA	Integer type	Input	Adjustable dimension of A (value of the first subscript in the array declaration of A). KA≧NB
N	Integer type	Input	Order of equations (number of columns of A). N≧1
NB	Integer type	Input	Band width (number of rows of A). 1≦NB≦N

X	Real type two-dimensional array	Input/output	The right side columns are input. The solution vectors are output to the corresponding place.
KX	Integer type	Input	Adjustable dimension of X. KX≧N
M	Integer type	Input	Number of columns in X. If M≦0, only (modified) Cholesky decomposition is executed.
DET	Real type	Input/output	If DET≠0.0 is input, coefficient matrix determinant is output. If DET=0.0 is input, 0.0 is output.
EPS	Real type	Input	Constant for determining the positivity of coefficient matrices. If the value of a diagonal element becomes smaller than that of EPS during Cholesky decomposition, the input matrix is decided to be non positive definite, and the calculation is interrupted. EPS>0
IND	Integer type	Input/output	This argument has the following meaning as an input. IND=0: Solve an equation newly starting from Cholesky decomposition. IND≠0: Find the solution of an equation, reusing the Cholesky decomposition elements calculated before. This argument has the following meaning as an output. IND=0: The calculation is normally executed. IND=K: Because the value of a diagonal element becomes smaller than that of EPS at the K-th step of Cholesky decomposition, the calculation is interrupted. IND=30000: The input arguments violate the limit.

*1 For a double (quadruple) precision subroutine, all real types are changed to double (quadruple) precision real types.

(3) Calculation method

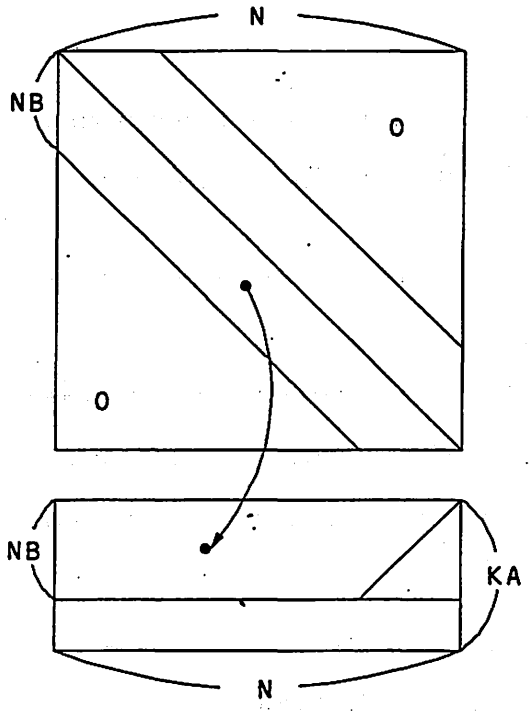
1. Cholesky decomposition method

Decompose $A=LL^T$ with a lower triangular matrix L and its transposition L^T. The solution $X=A^{-1}B$ is found by the forward substitution $Y=L^{-1}B$ and backward substitution $X=L^{-T}Y$.

2. Modified Cholesky decomposition method

Decompose $A=LDL^T$ with a lower unit triangular matrix L, its transposition L^T, and a diagonal matrix D. The solution $X=A^{-1}B$ is found by the forward substitution $Y=L^{-1}B$ and the backward substitution $X=L^{-T}\cdot D^{-1}Y$.

3. Because the partial double precision calculation is used for all inner product calculations in CHLBDS and MCHLBS, the influence of the round-off error is negligible.


N
NB
O
O
NB
KA
N

(4) Remarks

1. If the typical size of elements in a coefficient matrix is assumed to be α,

the value EPS= $10^{-6}\alpha(10^{-10}\alpha, 10^{-30}\alpha)$ is adequate for ┌MCHLBS(D, Q)┐ └CHLBDS(D, Q)┘

2. Because DET and IND are input/output arguments, constants must not be used as actual arguments.

3. If solutions to the same coefficient matrix is to be repeatedly found changing the right side columns, the facility for reusing modified Cholesky decomposition components of this routine is extremely useful. It exceeds in storage size, precision, and speed as compared with the method by inverse matrices.

4. If the number of right side columns M is 1, the actual argument that corresponds to X can be a one-dimensional array. However, KX≧N must be met.

(1987. 06. 16)

CHLBDV/W,MCHLBV/W

(Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Band Matrix) - Vector Version -)

Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method(Band Matrix) -Vector Version-

Programmed by	Ichizo Ninomiya, May 1986
Format	Subroutine language: FORTRAN77; size: 106, 107, 114, and 115 lines respectively

(1) Outline

CHLBDV(W) (MCHLBV(W)) is a single (double) precision subroutine for obtaining the solution $X=A^{-1}B$ of the equation $AX=B$ having a symmetric positive definite band matrix A as coefficient matrix and multiple right side columns B, using modified Cholesky decomposition method. It has the facility of reusing Cholesky decomposition component.

(2) Directions

CALL CHLBDV/W / MCHLBV/W (A, KA, N, NB, X, KX, M, DET, EPS, W, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	The lower left half band area containing the diagonal of a coefficient matrix is transformed into a rectangular form and input. That is, the I and J elements of the matrix are input in A(I-J+1, J). These elements are processed with this routine, and modified Cholesky decomposition elements are output. See the figure.

Argument	Type and kind (*1)	Attribute	Content
KA	Integer type	Input	Adjustable dimensions of A (value of the first subscript in the array declaration). KA≧NB
N	Integer type	Input	Order of equations (number of columns of A). N≧1
NB	Integer type	Input	Band width of A (number of rows of A). 1≦NB≦N
X	Real type two-dimensional array	Input/output	The right side columns are input. The solution vectors are output to the corresponding positions.
KX	Integer type	Input	Adjustable dimensions of X. KX≧N
M	Integer type	Input	Number of columns of X. If M≦0, only modified Cholesky decomposition is executed.
DET	Real type	Input/output	If DET≠0.0 is input, coefficient matrix determinant is output. If DET=0.0 is input, 0.0 is output.
EPS	Real type	Input	Coefficient matrix positivity criterion. If the value of a diagonal element becomes smaller than EPS during Cholesky decomposition, it is decided to be not positive definite, and the computation is interrupted. EPS>0
W	Real type one-dimensional array	Work area	One-dimensional array of size NB.

Argument	Type and kind (*1)	Attribute	Content
IND	Integer type	Input/output	This argument has the following meaning as an input argument. IND=0: An equation is solved newly beginning with Cholesky decomposition. IND≠0: Equation is solved reusing the Cholesky decomposition component calculated before. This argument has the following meaning as an output argument. IND=0: Computation is normally executed. IND=K: Computation is interrupted because the value of a diagonal element becomes smaller than EPS at the K-th step of Cholesky decomposition. IND=30000: The input argument exceeded the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

1. Cholesky decomposition method

The matrix A is decomposed into $A=LL^T$ using a lower triangular matrix L and its transpose L^T. The solution $X=A^{-1}B$ is obtained with the forward substitution $Y=L^{-1}B$ and backward substitution $X=L^{-T}Y$.

2. Modified Cholesky decomposition method

The matrix A is decomposed into $A=LDL^T$ with a lower unit triangular matrix L, its transposition L^T, and a diagonal matrix D. The solution $X=A^{-1}B$ is obtained with the forward substitution $Y=L^{-1}B$ and backward substitution $X=L^{-T}D^{-1}Y$.

3. Because partial double precision calculation is used for all inner sums in CHLBDS and MCHLBS, the influence of rounding errors is small.

(4) Notes

1. If the typical size of coefficient matrix elements is a, the value $10^{-6}\alpha(10^{-16}\alpha)$ is adequate as the standard value of EPS for {MCHLBV(W) CHLBDV(W)}.

2. Because DET and IND are I/O arguments, constants must not be used as an actual argument.

3. When a solution with the same coefficient matrix is to be repeatedly obtained with only the right side column changed, the function of reusing modified Cholesky decomposition component of this routine is particularly useful. It is more efficient in all of storage size, precision, and speed as compared with the method using the inverse matrix.

4. If the number M of right side columns is 1, the actual argument corresponding to X can be a one-dimensional array. However, KX≧N must be met.

(1987.06.19) (1987.08.07)

CHLVBS/D

(Solution of Symmetric Positive Definite Linear Equations by Cholesky Method) (Band Matrices with Variable Bandwidth, Compact Mode)

Solution of Symmetric Positive Definite Linear Equations by Cholesky Method

(Band Matrices with Variable Bandwidth, Compact Mode)

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 94 and 94 lines respectively

(1) Outline

CHLVBS/D finds the solution $X=A^{-1}B$ of the equation $AX=B$ with a symmetric positive definite variable bandwidth matrix A as a coefficient matrix and a multiple right sides B, using Cholesky decomposition method. A is first decomposed as $A=U^TU$ with an upper triangular matrix U and its transpose U^T, then the solution is given by $X=U^{-1}(U^{-T}B)$. $C=U^{-T}B$ is calculated by the forward substitution method for the lower triangular matrix U^T, and $X=U^{-1}C$ is calculated by the backward substitution method for the upper triangular matrix U. This routine posesses facility for reusing Cholesky decomposition components.

(2) Directions

CALL CHLVBS/D(A, NB, X, KX, N, M, EPS, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type One-dimensional array	Input/output	Rearrange the upper right half including the diagonal excluding zero elemnts of a symmetric positive definite band matrix in a line as shown in the figure. These elements are processed by this routine, and Cholesky decomposition components are output.
NB	Integer type One-dimensional array	Input	One-dimensional array with N elements. Input the band width of each column in the upper right half of a coefficient matrix. (See the figure.)

Argument	Type and kind (*1)	Attribute	Content
X	Real type Two-dimensional array	Input/output	The right side columns are input. The solution vector is output to the corresponding place.
KX	Integer type	Input	Value of the first subscript in the array declaration of X. KX≧N
N	Integer type	Input	Order of equations, that is, the number of rows of X. N≧2
M	Integer type	Input	Number of columns of X. M≧0 If M=0, only Cholesky decomposition is executed. If M=1, an actual argument to X can be a one-dimensional array.
EPS	Real type	Input	Constant for determining the positivity of coefficient matrix. If the value of a diagonal element becomes smaller than that of EPS during Cholesky decomposition, the input matrix is decided to be non positive definite, and the calculation is interrupted. EPS>0
IND	Integer type	Input/output	This argument has the following meaning as an input. IND=0: Solve an equation newly starting from Cholesky decomposition. IND≠0: Solve an equation reusing Cholesky decomposition components previously obtained. This argument has the following meaning as an output. IND=0: The calculation is normally executed. IND=30000: Limits on input-output arguments are violated. IND=K: Because the value of a diagonal element becomes smaller than that of EPS at the Kth step, the calculation is interrupted. Because this argument is for both input and output, a constant must not be used as an actual argument.

*1 For CHLVBD, all real types are assumed to be a double precision real type.

```
 1  2  4  7
    3  5  8 11
       6  9 12 15 19 ···
         10 13 16 20 ···
            14 17 21 ···
               18 22 ···
                  23 ···
-------------------------
NB 1  2  3  4  4  4  5 ···
```

(3) Remarks

If the facility for reusing the Cholesky decomposition components of this routine is exploited, it becomes almost unnecessary to calculate inverse matrices. This is because inverse matrices of band matrices are not band matrices any more, thus losing their advantage. The reuse of Cholesky decomposition is more excellent in computation speed and precision than processing via inverse matrices.

(1987. 06. 16)

CHOLCS/D/Q,MCHLCS/D/Q

(Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method) (Full Matrix, Compact Mode)

Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Full Matrix, Compact Mode)

Programmed by	Ichizo Ninomiya, April 1981
Format	Subroutine language: Assembler (CHOLCQ and MCHLCQ are FORTRAN.) Size: 203, 217, 60, 180, 179, and 96 lines respectively

(1) Outline

CHOLCS (D, Q) and (MCHLCS (D, Q)) are single (double or quadruple) precision subroutines that find the solution $X=A^{-1}B$ of the equation $AX=B$ with the compressed symmetric positive definite matrix A as a coefficient matrix and multiple right side columns B, using modified Cholesky decomposition method. It has the facility for reusing modified Cholesky decomposition components.

CALL [CHOLCS/D/Q / MCHLCS/D/Q] (A, N, X, KX, M, DET, EPS, IND)

(2) Directions

Argument	Type and kind (*1)	Attribute	Content
A	Real type One-dimensional array	Input/output	Converts the upper right half including the diagonal of a coefficient matrix column wise into a one dimensional vector before it is input. That is, the I and J element in the original matrix is stored in A ((J - 1))*J/2 + I). The array is processed by this routine, and modified Cholesky decomposition elements are output.

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Order of equations. N≧1
X	Real type Two-dimensional array	Input/output	The right side columns are input. The solution vectors are output to the corresponding place.
KX	Integer type	Input	Adjustable dimension of X. KX≧N
M	Integer type	Input	Number of columns in X. If M≦0, only modified Cholesky decomposition is executed.
DET	Real type	Input/output	If DET≠0.0 is input, coefficient determinants are output. If DET=0.0 is input, 0.0 is output.
EPS	Real type	Input	Constant for determining the positivity of coefficient matrices. If the value of a diagonal element becomes smaller than that of EPS during Cholesky decomposition, the input matrix is decided to be non positive definite and the calculation is interrupted.
IND	Integer type	Input/output	This argument has the following meaning as an input. IND=0: Solve an equation newly starting from Cholesky decomposition. IND≠0: Solve an equation, reusing the Cholesky decomposition components calculated previously. This argument has the following meaning as an output. IND=0: Calculation is normally executed. IND=K: Because the value of diagonal elements becomes smaller than that of EPS at the Kth step of Cholesky decomposition, the calculation is interrupted. IND=30000: The input arguments violate the limit.

*1 For a double (quadruple) precision subroutine, all real types are changed to double (quadruple) precision real types.

(3) Calculation method

1. Cholesky decomposition method

Decompose $A=U^TU$ with an upper triangular matrix U and its transpose U^T. The solution $X=A^{-1}B$ is found by the forward substitution $Y=U^{-T}B$ and backward substitution $Y=U^{-1}Y$.

2. Modified Cholesky decomposition method

Decompose $A=U^TDU$ with a unit upper triangular matrix U, its transpose U^T, and a diagonal matrix D. The solution $X=A^{-1}B$ is found by the forward substitution $Y=U^TB$ and backward substitution $X=U^{-1}D^{-1}Y$.

3. Because CHOLCS and MCHLCS use partial double precision in all product inner calculations, the effects of rounding errors on the results is negligible.

(4) Remarks

1. If the typical size of elements in a coefficient matrix is assumed to be α as the standard value of external page storage, $10^{-6}\alpha(10^{-16}\alpha, 10^{-30}\alpha)$ is adequate for:

[CHOLCS (D, Q)
MCHLCS (D, Q)]

2. Because DET and IND are input-output arguments, a constant must not be used as an actual argument.

3. When a solution to the same coefficient matrix is repeatedly found changing only the right side columns, facility for reusing the modified Cholesky decomposition components of this routine is extremely useful. As compared with the method by inverse matrices, this calculation method is excellent in storage size, precision, and speed.

4. If the number of right side columns M is 1, an actual argument that corresponds to X can be a one-dimensional array. However, KX≧N must be met.

(1987. 06. 17)

CHOLFC/B/Z, MCHLFC/B/Z

(Solution of Hermitian positive definite linear equations by Cholesky and modified Cholesky methods (full matrix))

Solution of Hermitian Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Full Matrix)

Programmed by	Ichizo Ninomiya; December 1983
Format	Subroutine language; FORTRAN Size; 50, 51, 51, 50, 51, and 51 lines respectively

(1) Outline

CHOLFC (B, Z) and MCHLFC (B, Z) are complex single (double, quadruple) precision subroutines, each of which calculates solution $X=A^{-1}B$ of equation $AX=B$ by the Cholesky or modified Cholesky decomposition method, where A is a Hermitian symmetric positive definite matrix and B is a right-hand side matrix. It has the facility of reusing Cholesky decomposition components.

(2) Directions

```
          CHOLFC/B/Z
CALL                  (A, KA, N, X, KX, M, DET, EPS, IND)
          MCHLFC/B/Z
```

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input/output	The upper right half including the diagonal of the coefficient matrix is input. After processing by this routine, Cholesky- or modified Cholesky-decomposed components are output. The lower left half is retained.
KA	Integer type	Input	Adjustable dimension of A (value of the first subscript in array declaration). KA≧N

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Number of unknowns of the equation. N≧1
X	Complex type Two-dimensional array	Input/output	A right-hand side vectors are input, as a matrix. Solution vectors are generated in the corresponding locations.
KX	Integer type	Input	Adjustable dimension of X. KX≧N
M	Integer type	Input	Number of columns of X. When M≦0, only Cholesky or modified Cholesky decomposition is performed.
DET	Real type	Input/output	When a value other than 0.0 is input, the coefficient determinant is output. When 0.0 is input, 0.0 is output as it is.
EPS	Real type	Input	Constant for test of positiveness of coefficient matrix. When the diagonal element becomes smaller than EPS during Cholesky decomposition, it is decided as non positive and calculation is interrupted.

Argument	Type and kind (*1)	Attribute	Content
IND	Integer type	Input/output	This argument has the following meanings for input: IND=0: An equation is solved beginning with Cholesky decomposition. IND≠0: An equation is solved by reusing the Cholesky-decomposed components obtained immediately before. This argument has the following meanings for output: IND=0: Calculation is performed normally. IND=K: Because the diagonal element became smaller than EPS at step K in Cholesky decomposition, calculation was discontinued. IND=30000: The input argument violated the limit.

*1 For double (quadruple) precision subroutines, single precision types are all changed to double (quadruple) precision types.

(3) Calculation method

1. Cholesky decomposition method

A is decomposed as a product of an upper triangular matrix U and its conjugate transpose U^* as $A=U^*U$. Then the solution $X=A^{-1}B$ is determined by forward substitution $Y=(U^*)^{-1}B$ and backward substitution $X=U^{-1}Y$.

2. Modified Cholesky decomposition method

A is decomposed as a product of an upper triangular matrix U and its conjugate transpose U^* and diagonal matrix D as $A=U^*DU$. Then the solution $X=A^{-1}B$ is determined by forward substitution $Y=(U^*)^{-1}B$ and backward substitution $X=U^{-1}D^{-1}Y$.

(4) Notes

1. Let a be the typical size of elements of the coefficient matrix, then $10^{-6}a(10^{-16}a, 10^{-30}a)$ is a reasonable EPS value for {CHOLFS (D, Q) MCHLFS (D, Q)}.

2. Do not use a constant as the actual argument for DET and IND because these are used for both input and output.

3. When equations sharing the same coefficient matrix are solved iteratively with different

right-hand side columns, this routine's facility of reusing the Cholesky- or modified Cholesky-decomposed components is very useful. This routine is superior to the inverse matrix method in every respect of storage capacity, accuracy, and speed.

4. When the number of right-hand side columns M is 1, the actual argument corresponding to X can be a one-dimensional array, where KX≧N must be met.

(1987. 06. 19) (1987. 08. 07) (1987. 08. 10)

CHOLFS/D/Q,MCHLFS/D/Q

(Solution of symmetric positive definite linear equations by Cholesky and modified Cholesky method (full matrix))

Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Full Matrix)

Programmed by	Ichizo Ninomiya; April 1981
Format	Subroutine language; Assembler (CHOLFQ and MCHLFQ are in FORTRAN) Size; 207, 217, 49, 179, 176, and 51 lines respectively

(1) Outline

CHOLFS(D, Q) (MCHLFS(D, Q)) is a subroutine for single (double or quadruple) precision to obtain solution $X=A^{-1}B$ of matrix equation $AX=B$ by the Cholesky or modified Cholesky decomposition method when A is positive definite. It has the facility reusing Cholesky or modified Cholesky decomposition components.

(2) Directions

```
        CHOLFS/D/Q
CALL                (A, KA, N, X, KX, M, DET, EPS, IND)
        MCHLFS/D/Q
```

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	The upper right half of a coefficient matrix including the diagonal is input. After processing by this routine, Cholesky or modified Cholesky composition components are generated. The lower left half is saved.
KA	Integer type	Input	Adjustable dimension of A (value of the first subscript in array declaration). KA≧N
N	Integer type	Input	Number of unknowns of equation. N≧1

Argument	Type and kind (*1)	Attribute	Content
X	Real type two-dimensional array	Input/output	The matrix of right-hand columns are input. Solution vectors are generated at the corresponding columns.
KX	Integer type	Input	Adjustable dimension of X. KX≧N
M	Integer type	Input	Number of columns of X. If M ≦ 0, only Cholesky or modified Cholesky decomposition is done.
DET	Real type	Input/output	When DET≠0.0 is input, coefficient determinant is generated. When DET=0.0 is input, 0.0 is returned.
EPS	Real type	Input	Constant used to check positivity of the coefficient matrix. When the diagonal element becomes smaller than EPS during Cholesky decomposition, it is assumed to be non-positive definite and calculation is interrupted.
IND	Integer type	Input/output	For input, this argument has the following meanings: IND=0: The equation is solved by restarting Cholesky decomposition from the beginning. IND≠0: The equation is solved by reusing the Cholesky decomposition component calculated immediately before. For output, this argument has the following meanings: IND=0: Calculation has been done normally. IND=K: Calculation was terminated because the diagonal element became smaller than EPS at step K in Cholesky decomposition. IND=30000: The input argument violated the limit.

*1 For double or quadruple precision subroutines, all real types are changed to double or quadruple precision real types.

(3) Calculation method

1. Cholesky decomposition method

A is decomposed as $A=U^TU$ by an upper triangular matrix U and its conjugate transpose

matrix U^T. Solution $X=A^{-1}B$ is obtained by forward substitution $Y=U^{-T}B$ and backward substitution $X=U^{-1}Y$.

2. Modified Cholesky decomposition method

A is decomposed as $A=U^TDU$ by an unit upper triangular matrix U, its conjugate transpose matrix U^T, and a diagonal matrix D. Solution $X=A^{-1}B$ is obtained by forward substitution $Y=U^{-T}B$ and backward substitution $X=U^{-1}D^{-1}Y$.

3. Because the partial double precision calculation is used for all inner product calculations in CHOLFS and MCHLFS, rounding errors have little influence on it.

(4) Notes

1. When a typical size of the element of the coefficient matrix is assumed to be a, $10^{-6}a(10^{-16}a, 10^{-30}a)$ is suitable as a standard EPS value for {CHOLFS(D, Q) and MCHLFS(D, Q)}.

2. Do not specify a constant for an actual argument of DET and IND because these arguments are used for both input and output.

3. For the purpose of obtaining solutions of the same equation by simply changing its right-hand side column, this routine's facility to reuse Cholesky or modified Cholesky decomposition components is extremely useful. The Cholesky method is superior to the inverse matrix method in all aspects of the storage capacity, accuracy, and speed.

4. When M, the number of right hand side columns, is 1, a one-dimensional array is acceptable for the actual argument corresponding to X. KX≧N should hold, however.

(1987. 06. 19) (1987. 08. 10)

CHOLFV/W,MCHLFV/W

(Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Full Matrix) - Vector Version-)

Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method(Full Matrix) -Vector Version-

Programmed by	Ichizo Ninomiya, December 1984
Format	Subroutine language: FORTRAN77; size: 141, 142, 141, and 142 lines respectively

(1) Outline

CHOLFV(W) and (MCHLFV(W)) are single (double) precision subroutines for obtaining the solution $X=A^{-1}B$ of the equation $AX=B$ having a symmetric positive definite matrix A as coefficient matrix and multiple right sides B, using modified Cholesky decomposition method. It has the facility of reusing modified Cholesky decomposition component.

CHOLFV(W) and MCHLFV(W) are for single (double) precision.

(2) Directions

```
        CHOLFV/W
CALL              (A, KA, N, X, KX, M, DET, EPS, W, IND)
        MCHLFV/W
```

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	The upper right half containing the diagonal of a coefficient matrix is input. It is processed with this routine, and modified Cholesky decomposition elements are output. The lower left half is retained.
KA	Integer type	Input	Adjustable dimensions of A (value of the first subscript in the array declaration). KA≧N

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Order of equations. N≧1
X	Real type two-dimensional array	Input/output	The right side columns are input. The solution vectors are output to the corresponding positions.
KX	Integer type	Input	Adjustable dimensions of X. KX≧N
M	Integer type	Input	Number of columns of X. If M≦0, only modified Cholesky decomposition is executed.
DET	Real type	Input/output	If DET≠0.0 is input, coefficient matrix determinant is output. If DET=0.0 is input, 0.0 is output.
EPS	Real type	Input	Coefficient matrix positivity criterion. If the value of a diagonal element becomes smaller than EPS during Cholesky decomposition, it is decided to be not positive definite, and the computation is interrupted.
W	Real type one-dimensional array	Work area	One-dimensional array of size N.

Argument	Type and kind (*1)	Attribute	Content
IND	Integer type	Input/output	This argument has the following meaning as an input argument. IND=0: An equation is solved newly beginning with Cholesky decomposition. IND≠0: An equation is solved reusing the Cholesky decomposition elements calculated before. This argument has the following meaning as an output argument. IND=0: Computation is normally executed. IND=K: Computation is interrupted because the value of a diagonal element becomes smaller than EPS at the K-th step of Cholesky decomposition. IND=30000: The input argument exceeded the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

1. Cholesky decomposition method

The coefficient matrix A is decomposed into $A=U^TU$ using an upper triangular matrix U and its transpose U^T. The solution $X=A^{-1}B$ is obtained with the forward substitution $Y=U^{-T}B$ and backward substitution $X=U^{-1}Y$.

2. Modified Cholesky decomposition method

The coefficient matrix A is decomposed into $A=U^TDU$ using an upper unit triangular matrix U, its transpose U^T, and a diagonal matrix D. The solution $X=A^{-1}B$ is obtained with the forward substitution $Y=U^{-T}B$ and backward substitution $X=U^{-1}D^{-1}Y$.

(4) Notes

1. If the typical size of coefficient matrix elements is a, the value $10^{-6}a(10^{-16}a)$ is adequate as the standard value of EPS for {CHOLFV(W) MCHLFV(W)}.

2. Because DET and IND are input-output arguments, constants must not be used as an actual

argument.

3. When a solution with the same matrix is to be repeatedly obtained with only the right side column changed, the function of reusing the modified Cholesky decomposition component of this routine is particularly useful. It is more efficient in all of storage size, precision, and speed as compared with the method using the inverse matrix.

4. If the number M of right side columns is 1, the actual argument corresponding to X can be a one-dimensional array. However, KX≧N must be met.

(1987.06.19)

CHOLSK/CHOLSD (Solution of Symmetric Positive Definite Linear Equations by Cholesky Method)

Solution of Symmetric Positive Definite Linear Equations by Cholesky Method

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 43 and 43 lines respectively

(1) Outline

CHOLSK/CHOLSD solves multiple simultaneous linear equations that share a symmetric positive definite coefficient matrix, using Cholesky decomposition method. In other words, it finds the solution $X=A^{-1}B$ of the matricial equation $AX=B$.

(2) Directions

CALL CHOLSK(A, KA, N, M, EPS, IND)

CALL CHOLSD(A, KA, N, M, EPS, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Input an augmented matrix with multiple right side columns added to the right of a symmetric positive coefficient matrix. Only the upper right half including the diagonal of the coefficient matrix need be input. When the matrix is processed with this routine, Cholesky decomposition components are output in the same place. The corresponding solution vectors are output in the corresponding right side columns. The lower left half of the coefficient matrix is preserved. If Cholesky decomposition components are preserved, the computation time required for Cholesky decomposition can be saved when another equation with the same coefficient matrix is to be solved.
KA	Integer type	Input	Value of the first subscript in the array declaration of A. KA≧N
N	Integer type	Input	Number of rows of A. It is also the order of equation. N≧2

Argument	Type and kind (*1)	Attribute	Content
M	Integer type	Input	Number of columns of A. Sum of the order of equations and the number of right side columns. $M \geqq N_0$ If M = N, only Cholesky decomposition of the coefficient matrix is executed.
EPS	Real type	Input	Constant for deciding the positivity of A. If the value of a pivot element in A is smaller than that of EPS, the input matrix is decided to be non positive definite, and the the calculation is interrupted. EPS>0
IND	Integer type	Input/output	Determines whether to reuse Cholesky decomposition components or not as an input. If IND = 0, normal calculation is executed. If IND≠0, the component is reused.
			Indicates the calculation status in the routine as an output. If limits on KA, N, M, and EPS are violated, 30000 is assumed. If calculation is interrupted at K-th stage of decomposition, K is assumed. If calculation is normally executed, 0 is assumed.

*1 For CHOLSD, all real types are changed to double precision real types.

(3) Remarks

1. Because the argument IND is used for both input and output, a constant must not be written as an actual argument for it.

2. When a number of equations that has the same coefficient matrix, but differ in the right side only are to be solved, computation time can be saved by using the facility for reusing the Cholesky decomposition components of this routine.

3. If the typical size of matrix elements is α, $\alpha \times 10^{-6} (\alpha \times 10^{-16})$ is adequate as the standard value of EPS for CHOLSK (CHOLSD).

4. CHOLFS/D routine that is similar to this routine but has much more facilities, and the CHLBDS/D routine for band matrix are available in NUMPAC.

(1987. 06. 17) (1987. 08. 07)

72

GAUELS/D/Q/C/B (Solution of Linear Equations by LU-Decomposition)

Solution of Linear Equations by LU-Decomposition

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN (GAUELS/D is assembler); size: 180, 183, 53, 43, and 44 lines respectively

(1) Outline

GAUELS/D/Q/C/B solves multiple simultaneous linear equations that share a coefficient matrix, using a modified Doolittle method a version of Gaussian elimination accompanied by row interchange for pivot selection. In other words, it finds the solution $X=A^{-1}B$ of the matricial equation $AX=B$.

(2) Directions

CALL GAUELS/D/Q/C/B(A, KA, N, M, EPS, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Input an augmented matrix in which right side columns are added to the right of the coefficient matrix. The solution vectors are output in the corresponding right side columns.
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Number of rows in A, that is, the order of the equation. N≧2
M	Integer type	Input	The number of columns in A, that is, the order of the equation plus the number of equations to be solved at the same time. M>N
EPS	Real type	Input	Criterion constant for matrix singularity. If the absolute value of a pivot element is smaller than this constant, the input matrix is decided to be singular, and the calculation is interrupted. EPS>0

Argument	Type and kind (*1)	Attribute	Content
ILL	Integer type	Output	ILL=0: Normal termination ILL = 30000: Limits on KA, N, M, and EPS are violated. The number of the pivot element whose absolute value is smaller than EPS.

*1 For GAUELD (Q, C, B), A is a double precision real type (quadruple precision real type, complex type, and double precision complex type).

For GAUELD (Q, C, B), EPS is a double precision real type (quadruple precision real type, real type, and double precision real type).

(3) Remarks

1. If the magnitude of coefficient of equations differ significantly, it is desirable to normalize the equations in advance using MNORMS and MNORMD to insure precision in the results.

2. If the typical absolute value of elements in a coefficient matrix is to be α, $\alpha\times10^{-6}(\alpha\times10^{-16},\alpha\times10^{-30})$ is adequate as the standard value of EPS for GAUELS and (GAUELD, GAUELQ).

3. If there is no special reason, it is recommended to use LEQLUS with high precision and various facilities, as a simultaneous linear equation solver.

(1987. 06. 17) (1987. 08. 07)

GSORSS/D (Solution of linear equations for sparse matrices by SOR method (compact mode))

Solution of Linear Equations for Sparse Matrices by SOR Method(Compact Mode)

Programmed by	Yasuyo Hatano 1977, Revised; Ichizo Ninomiya 1982
Format	Subroutine language; FORTRAN Size; 49 and 50 lines respectively

(1) Outline

Each of these subroutines solves linear equations with coefficient matrices including many 0 elements. To do this, it uses the SOR (successive over-relaxation) method based on the Gauss-Seidel method. This routine is used when only non-zero elements of coefficient matrices are reduced to a one-dimensional system by the compact mode and input.

(2) Directions

CALL GSORSS/D(IJTAB, A, LA, B, N, X, EPS, OMG, IMAX, ILL)

Argument	Type and kind (*1)	Attribute	Content
IJTAB	Two bytes Integer type Two-dimensional array	Input	The numbers of rows and columns with non-zero elements are input to IJTAB (1, K) and IJTAB (2, K) respectively. Suppose $A(K)=a_{ij}$, for instance, then input IJTAB(1, K) = i, IJTAB(2, K) = j. The values are not retained.
A	Real type one-dimensional array	Input	Only non-zero elements of the coefficient matrix are input in a row. The values of i, j are rearranged in ascending order, divided by the pivot element, and put in this argument. Note that the values are not retained.

Argument	Type and kind (*1)	Attribute	Content
LA	Integer type	Input	Number of non-zero elements of coefficient matrix
B	Real type one-dimensional array	Input	One right-hand side column is input. It is then divided by the pivot element and output. The value is not retained.
N	Integer type	Input	Number of unknowns of equation. 2≧N
X	Real type one-dimensional array	Input/output	Input: Initial value of solution vector. Output: Solution vector after iterative calculation. Size N.
EPS	Real type	Input	Tolerance for convergence test in iteration method. When all correction quantities of solution vectors are below EPS, it is assumed that convergence has occurred.
OMG	Real type	Input	Acceleration factor for convergence in iteration method. 1≦OMG<2
IMAX	Integer type	Input/output	Input: Upper limit of the number of iterations. 0<IMAX. Output: Actual number of iterations until convergence of solution vectors
ILL	Integer type	Output	ILL=0: Normal termination ILL=30000: The restrictions on N or OMG are not observed. ILL=25000: IJTAB error ILL=K: Calculation is not done because the diagonal element on Kth row is 0. ILL=IMAX: Convergence does not occur in IMAX iterations.

*1 For double precision subroutines, real types are all assumed to be double precision real types.

(3) Example

```
00010 C....  TEST OF GSORSS ....
00020       DIMENSION V(41,31),IV(41,31),A(4500),X(1000),
00021      1     B(1000)
00030       INTEGER*2 IJTAB(2,4500)
00040       EQUIVALENCE (V,IV)
00050       M=41
00060       N=31
00070       IMAX=1000
00080       EPS=1.E-4
00090       OMG=1.8
00100       DO 20 J=2,30
00110       DO 20 I=2,40
00120    20 IV(I,J)=1
00130       DO 21 J=1,31
00140       V(1,J)=0.
00150    21 V(41,J)=0.
00160       DO 22 I=12,40
00170       V(I,1)=0.
00180    22 V(I,31)=0.
00190       DO 23 I=2,11
00200       DO 23 J=1,11
00210       V(I,J)=0.
00220    23 V(I,J+20)=0.
00230       DO 24 J=11,21
00240       DO 24 I=21,31
00250    24 V(I,J)=100.
00260       WRITE(6,601) ((IV(I,J),J=1,31),I=1,41)
00270       LA=4500
00280       NA=1000
00290       CALL CLOCKM(ITO)
00300       CALL LAPLSS(V,IV,M,M,N,IJTAB,A,LA,B,NA,X,EPS,OMG,
00301      2     IMAX,ILL)
00310       CALL CLOCKM(IT)
00320       IT=IT-ITO
00330       WRITE(6,600) ILL,NA,LA,IT,IMAX,((V(I,J),J=1,16),
00331      3     I=1,41)
00340   600 FORMAT(1H1///10X,5HILL =,I6,5X,4HNA =,I7,5X,4HLA
00341      4     =,I5
00350      *,5X,6HTIME =,I7,5X,6HITER =,I7///(5X,16F8.3))
00360       DO 40 I=1,41
00370       DO 40 J=1,31
00380    40 IV(I,J)=V(I,J)+0.5
00390   555 WRITE(6,601) ((IV(I,J),J=1,31),I=1,41)
00400   601 FORMAT(1H1///(5X,31I4))
00410       STOP
00420       END
00430       SUBROUTINE LAPLSS(V,IV,KV,NR,NC,IJTAB,A,LA,B,NA,X
00440      *,EPS,OMG,IMAX,ILL)
00450       DIMENSION V(KV,NC),IV(KV,NC),A(LA),B(NA),X(NA)
00460       INTEGER*2 IJTAB(2,LA)
00470       IF(NR.LT.3.OR.NC.LT.3.OR.KV.LT.NR.OR.
00480      * EPS.LE.0..OR.OMG.LT.1..OR.OMG.GE.2.) GO TO 130
00490       NR1=NR-1
00500       NC1=NC-1
00510       N=0
00520       L=0
```

```
00530        DO 70 J=2,NC1
00540        LB=1
00550        DO 60 I=2,NR1
00560        IF(IV(I,J).NE.1) GO TO 50
00570        N=N+1
00580        IF(N.GT.NA) GO TO 120
00590        X(N)=0.0
00600        IV(I,J)=N
00610        LEFT=IV(I,J-1)
00620        IF(LEFT.GT.0.AND.LEFT.LT.N) GO TO 10
00630        B(N)=V(I,J-1)
00640        GO TO 20
00650     10 L=L+1
00660        A(L)=-1.0
00670        IJTAB(1,L)=LEFT
00680        IJTAB(2,L)=N
00690     20 IF(LB.EQ.0) GO TO 30
00700        B(N)=V(I-1,J)+B(N)
00710        GO TO 40
00720     30 L=L+1
00730        A(L)=-1.0
00740        IJTAB(1,L)=N-1
00750        IJTAB(2,L)=N
00760     40 L=L+1
00770        IF(L.GT.LA) GO TO 120
00780        A(L)=4.0
00790        IJTAB(1,L)=N
00800        IJTAB(2,L)=N
00810        IF(IV(I+1,J).NE.1) B(N)=V(I+1,J)+B(N)
00820        IF(IV(I,J+1).NE.1) B(N)=V(I,J+1)+B(N)
00830        LB=0
00840        GO TO 60
00850     50 LB=1
00860     60 CONTINUE
00870     70 CONTINUE
00880        NA=N
00890        KM=L
00900        DO 80 K=1,KM
00910        IF(A(K).NE.-1.) GO TO 80
00920        L=L+1
00930        IF(L.GT.LA) GO TO 120
00940        A(L)=-1.0
00950        IJTAB(1,L)=IJTAB(2,K)
00960        IJTAB(2,L)=IJTAB(1,K)
00970     80 CONTINUE
00980        LA=L
00990        CALL GSORSS(IJTAB,A,LA,B,NA,X,EPS,OMG,IMAX,ILL)
01000        IF(ILL.NE.0) GO TO 110
01010        DO 100 J=2,NC1
01020        DO 90 I=2,NR1
01030        L=IV(I,J)
01040        IF(L.LE.0.OR.L.GT.NA) GO TO 90
01050        V(I,J)=X(L)
01060     90 CONTINUE
01070    100 CONTINUE
01080        RETURN
01090    110 ILL=10000
01100        RETURN
01110    120 ILL=20000
01120        RETURN
01130    130 ILL=30000
01140        RETURN
```

```
01150          END
```

(4) Note

If the coefficient matrices are positive definite symmetric, the Gauss-Seidel method (when OMG=1) converges. Furthermore, if the sum of absolute values of non-diagonal elements on each row is smaller than that of diagonal elements, that is, if

$$\sum_{j=1}^{N} |a_{ij}| < |a_{ii}|$$

is met, then convergence occurs. However, this is effective only when the right-hand side is sufficiently larger than the left-hand side and OMG is adequate.

Bibliography

1) Hayato Togawa; "Numerical calculation of matrices," page 64, Ohm-sha (1971)

(1987.06.19)

LAPLBS/VS/SS/CS (Solution of 2-dimensional Laplacian equation)

Solution of 2-Dimensional Laplacian Equation

Programmed by	Ichizo Ninomiya, Yasuyo Hatano, and Tsuyako Miyakoda; September 1982
Format	Subroutine language; FORTRAN77 Size; 07, 69, 73, and 73 lines respectively

(1) Outline

Each of these subroutines solves a Dirichlet boundary value problem of two-dimensional Laplacian equations by five-point difference approximation with uniform orthogonal mesh. When the mesh division of the solution region and the distribution of the boundary values are given, it automatically generates a five-point difference approximation equation, and solves it by the corresponding method as follows:

LAPLBS: Modified Cholesky decomposition method for symmetric band matrix

LAPLVS: Cholesky decomposition method for symmetric band matrix of variable width

LAPLSS: SOR method for sparse matrix

LAPLCS: Conjugate gradient method with preconditioning for sparse matrix

(2) Directions

CALL LAPLBS(V, IV, KV, NR, NC, A, LA, NA, S, IND)

CALL LAPLVS(V, IV, KV, NR, NC, A, LA, NA, NB, S, IND)

CALL LAPLSS(V, IV, KV, NR, NC, IJTAB, A, LA, B, NA, X, EPS, OMG, IMAX, ILL)

CALL LAPLCS(V, IV, KV, NR, NC, IJTAB, A, LA, B, NA, X, EPS, OMG, IMAX, IW, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
V	Real type two-dimensional array	Input/output	Solution region. V and IV are connected by the EQUIVALENCE statement to be assigned to the same region. It is then used appropriately for V or IV depending on purpose. Input: A boundary value is input in V in the boundary point, IV=1 in inner points, and IV=0 in the other points. Output: A solution is output in the interior point as V. (See the example.)
IV	Integer type two-dimensional array	Input/output	
KV	Integer type	Input	Adjustable dimension of V(IV). KV≧NR
NR	Integer type	Input	Number of rows of V(IV). NR≧3
NC	Integer type	Input	Number of columns of V(IV). NC≧3
IJTAB	Integer type one-dimensional array	Output	Numbers of rows and columns of non-zero coefficients of equation. Size LA.
A	Real type one-dimensional array	Output	Equation coefficients (non-zero coefficients for LAPLSS/D and LAPLCS/D) are generated and processed. Size LA.

Argument	Type and kind (*1)	Attribute	Content
LA	Integer type	Input/output	Input: Estimation of the total number of equation coefficients. Output: Total number of equation coefficients.
B	Real type one-dimensional array	Output	Right-hand side vector of equation. Size NA.
NA	Integer type	Input/output	Input: Estimation of the number of unknowns of equations. Output: Number of unknowns of equations.
S	Real type one-dimensional array	Output	Solution vector. Size NA. The same is output also in V.
X	Real type one-dimensional array	Input/output	Input: Initial approximation vector of solution. Output: Solution vector. The same is output also in V. Size NA.
EPS	Real type	Input	Criterion for convergence test. EPS>0
OMG	Real type	Input	Acceleration factor. $1 \leqq OMG < 2$
IMAX	Integer type	Input/output	Input: Upper limit of the number of iterations. Output: Number of iterations.
IW	Integer type one-dimensional array	Work area	Size LA.

Argument	Type and kind (*1)	Attribute	Content
W	Real type one-dimensional array	Work area	The size is 6*NA.
IND	Integer type	Input/output	Input: IND=0: A coefficient matrix is generated and decomposed. A right-hand side vector is generated and solved. IND≠0: Generation and decomposition of a coefficient matrix is omitted, and only a right-hand side vector is generated and solved by using the result of the previous call. It is useful to repeat calculation for the same region with different boundary values. Output: IND=0: Normal termination. IND=10000: Cholesky decomposition was interrupted. IND=20000: LA or NA is too small. IND=30000: Parameter error.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=10000: The number of iterations exceeded the upper limit. ILL=20000: LA or NA is too small. ILL=30000: Parameter error.

(3) Example

The program shown below uses LAPLBS to solve the Laplacian equations for a convex region with a rectangular hole, whose external boundary value is 0, and whose internal boundary value is 100.

KKK=1 (IND=0) indicates an ordinary usage.

KKK=2 (IND=1) indicates how to reuse the decomposed components of a coefficient matrix.

```
      DIMENSION V(41,31),IV(41,31),A(36900),X(1000)
      EQUIVALENCE (V,IV)
      DO 555 KKK=1,2
      DO 20 J=2,30
      DO 20 I=2,40
   20 IV(I,J)=1
      DO 21 J=1,31
      V(1,J)=0.
   21 V(41,J)=0.
      DO 22 I=12,40
      V(I,1)=0.
   22 V(I,31)=0.
      DO 23 I=2,11
      DO 23 J=1,11
      V(I,J)=0.
   23 V(I,J+20)=0.
      DO 24 J=11,21
      DO 24 I=21,31
   24 V(I,J)=100.
      IF(KKK.EQ.1) WRITE(6,601) ((IV(I,J),J=1,31),I=1,41)
      M=41
      N=31
      IND=KKK-1
      NA=1000
      LA=36900
      CALL CLOCKM(ITO)
      CALL LAPLBS(V,IV,M,M,N,A,LA,NA,X,IND)
      CALL CLOCKM(IT)
      IT=IT-ITO
      WRITE(6,600) IND,NA,LA,IT,((V(I,J),J=1,16),I=1,41)
  600 FORMAT(1H1///10X,5HILL =,I6,5X,4HNA =,I7,5X,4HLA =,I7
     *,5X,6HTIME =,I7///(5X,16F8.3))
      DO 40 I=1,41
      DO 40 J=1,31
   40 IV(I,J)=V(I,J)+0.5
  555 WRITE(6,601) ((IV(I,J),J=1,31),I=1,41)
  601 FORMAT(1H1///(5X,31I4))
      STOP
      END
```

A program that solves the same problem by using LAPLSS and the source program of LAPLSS are shown in the example of GSORSS.

(4) Notes

1. For the method of solving the difference equation generated, see the direction of each corresponding subroutine as follows:

LAPLBS·····MCHLBS

LAPLVS·····CHLVBS

LAPLSS·····GSORSS

LAPLCS·····PRCGSS

84

2. It is recommended to allocate a solution area in m x n matrices where m (columns) is larger than n (rows). This can reduce the band width of the coefficient matrices generated and also save the storage capacity and computation time.

3. When calculation is repeated in the same region with different boundary values by using the LAPLBS or LAPLVS subroutine, the subroutine's facility of reusing the Cholesky-decomposed components is very effective to save computation time. Refer to the explanation of IND and the example.

4. When LAPLSS or LAPLCS is used, it is desirable to put a value as close as possible to the true solution into the initial value (X) of the solution vector. If sufficient information for it is not available, however, a zero vector, for instance, or a vector whose components are all equal to the average boundary value can be used.

(1987.06.19)

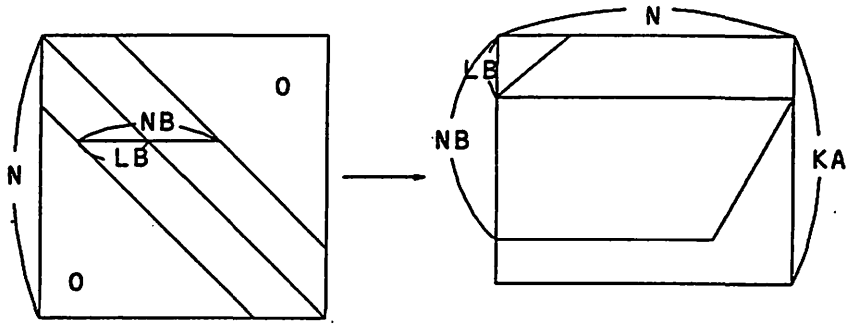
LEQBDS/D/Q/C/B (Solution of Linear Equations with Band Matrix of Coefficients by Gaussian Elimination)

Solution of Linear Equations with Band Matrix of Coefficients by Gaussian Elimination

Programmed by	Ichizo Ninomiya, September 1978
Format	Subroutine language: FORTRAN; size: 80, 80, 80, 81, and 80 lines respectively

(1) Outline

LEQBDS/D/Q/C/B finds the solution $X=A^{-1}B$ of the simultaneous linear equation $AX=B$ with a band matrix A as coefficient matrix and multiple right side columns B using the Gaussian elimination accompanied by row interchange for pivot selection. It has facility for reusing the LU-decomposition elements of A obtained by the elimination.



(2) Directions

CALL LEQBDS/D/Q/C/B(A, KA, N, NB, LB, MB, X, KX, NX, MAX, EPS, IND)

86

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Transform the coefficient band matrix into a rectangle form, that is, the I and J elements in an original matrix is stored in A(J-1+LB, I) (See the figure). The LU decomposition elements processed by this routine are output.
KA	Integer type	Input	Value of the first subscript in the array declaration of A. KA≧NB
N	Integer type	Input	The order of an equation, that is, the number of columns in A. N≧NB
NB	Integer type	Input	Total band width (see the figure). It is also the number of rows in A. NB>LB
LB	Integer type	Input	Left band width (see the figure). LB≧2
MB	Integer type	Output	Number of rows in A after processing. MB≦KA must hold.
X	Real type Two-dimensional array	Input/output	If the right side columns are input, the solution vectors are output in the corresponding place.
KX	Integer type	Input	Value of the first subscript in the array declaration of X. KX≧N
NX	Integer type	Input	Number of columns in X. Only A is processed when NX≦0.
MAX	Integer type One-dimensional array	Output	One-dimensional array containing N elements. It stores information on row interchange, and is used when LU elements are reused.
EPS	Real type	Input	Criterion constant for matrix singularity. If the absolute value of pivot elements is smaller than that of EPS, the input matrix is decided to be singular, and calculation is interrupted. EPS>0

Argument	Type and kind (*1)	Attribute	Content
IND	Integer type	Input/output	This argument has the following meaning as an input. If IND=0, it indicates that an equation should be solved from the beginning starting from the elimination. If IND≠0, it indicates that a solution should be obtained immediately by reusing the LU decomposition elements previously obtained and skipping the elimination. A and MAX must be left unchanged in the state of the previous call. This argument has the following meaning as an output. 0: When calculation terminates normally. 30000: When no calculation is executed because limits on the argument are violated. K: When a matrix is decided to be singular, and the elimination is interrupted at the Kth step

*1 For LEQBDD (Q, C, B), A and X are double precision real types (quadruple precision real type, complex type, and double precision complex type).

For LEQBDD (Q, C, B), EPS is a double precision real type (quadruple precision real type, real type, and double precision real type).

(3) Example of use

This example shows a program for solving an equation with order N=1000, total band width NB=7 and left band width LB=3.

```
C       TEST FOR LEQBDS
        DIMENSION A(10,1000),MAX(1000),X(1000)
        N=1000
        NB=7
        LB=3
        KA=10
        EPS=1.0E-6
        NX=1
        C=1.
        DO 10 I=1,N
        A(1,I)=C
        A(2,I)=-C
        A(3,I)=0.
        A(4,I)=0.
        A(5,I)=C
```

```
      A(6,I)=0.
   10 A(7,I)=-C
      DO 20 L=1,2
      IND=L-1
      DO 11 I=1,N
   11 X(I)=0.0
      X(2)=-C*C
      X(N-2)=C*C
      X(N-3)=C*C
      CALL CLOCKM(KO)
      CALL LEQBDS(A,KA,N,NB,LB,MB,X,N,NX,MAX,EPS,IND)
      CALL CLOCKM(K1)
      KO=K1-KO
      AM=0.0
      DO 12 I=1,N
      AA=ABS(X(I)-C)
      IF(AA.LE.AM) GO TO 12
      AM=AA
      MM=I
   12 CONTINUE
      WRITE(6,600) IND,KO,AM,MM,MB
  600 FORMAT(//10X,'ILL=',I6,5X,'TIME =',I6,5X'ERR=',1PE10.2,
     *5X,'IMAX ='I6,5X,'MB=',I6/)
   20 CONTINUE
      STOP
      END
```

(4) Remarks

1. Since this routine posesses facilities of simultaneous solution of multiple right hand sides and reuse of LU decomposition components, it can play the role of both a linear equation solver and an inverse matrix routine. The reuse of LU decomposition components is especially important. This eliminates the needs for inverse matrix calculation.

2. Because rows are interchanged, if necessary, for pivot selection, the number of columns of the coefficient matrix normally becomes greater than that in the initial state. Thus, KA must be assigned so that MB≦KA. Because MB=NB+LB-1 even in the worst case, KA=NB+LB-1 should be assigned for safety.

3. If equation coefficients differ significantly in size, it is desirable to normalize the coefficient matrix in advance so that the maximum absolute value of each equation coefficient is in the order of 1.

4. If the typical size of elements in a coefficient matrix is a, $a\times10^{-6}(a\times10^{-16})$ is adequate as the standard value of EPS for LEQBDS (LEQBDD).

5. If a coefficient matrix consists of symmetric positive definites, it is wise to use the special routine CHLBDS.

6. Because the argument IND is used for both input and output, a constant must not be used as an actual argument for it.

(1987.06.22) (1987.08.07)

LEQBDV/W/X/Y (Solution of linear equations with band matrix of coefficients by Gaussian elimination - vector version -)

Solution of Linear Equations with Band Matrix of Coefficients by Gaussian Elimination
-Vector Version-

Programmed by	Ichizo Ninomiya; May 1986
Format	Subroutine language; FORTRAN Size; 146, 147, 149, and 150 respectively

(1) Outline

Each of these subroutines determines the solution $X=A^{-1}B$ of the linear equation $AX=B$ having band matrix A as a coefficient matrix and right-hand side matrix B by the Gaussian Elimination involving row interchange for pivoting. It has the facility to reuse LU-decomposition components of A.

(2) Directions

CALL LEQBDV/W/X/Y(A, KA, N, NB, LB, MB, X, KX, NX, MAX, EPS, W, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	A band matrix of coefficients transformed to a rectangular form is input. That is, elements I and J of the original matrix are put in A(J-I+LB, I). (See the figure for LEQBDS.) After processing by this routine, LU-decomposition components are entered.
KA	Integer type	Input	Value of the first subscript in array declaration of A. KA≧NB
N	Integer type	Input	Number of unknowns of the equation, or number of columns of A. N≧NB

Argument	Type and kind (*1)	Attribute	Content
NB	Integer type	Input	Total band width (See the figure.) Number of rows of A. NB>LB
LB	Integer type	Input	Left band width (See the figure.) LB≧2
MB	Integer type	Output	Number of rows of A after processing. MB≦KA must be met.
X	Real type two-dimensional array	Input/output	Right hand side columns are input, and corresponding solution vectors are output.
KX	Integer type	Input	Value of the first subscript in array declaration of X. KX≧N
NX	Integer type	Input	Number of columns of X. Only A is processed when NX≦0.
MAX	Integer type one-dimensional array	Output	One-dimensional array with the N elements. Information concerning row interchange is kept in it. It is needed when LU-decomposition components are reused.
EPS	Real type	Input	Criterion constant of singularity of coefficient matrix. When the absolute value of an pivot element is smaller than EPS, the coefficient matrix is decided as singular and calculation is interrupted. EPS>0
W	Real type one-dimensional array	Work area	One-dimensional array of size LB+NB.

Argument	Type and kind (*1)	Attribute	Content
IND	Integer type	Input/output	Input: IND=0 indicates that the equation will be solved starting with elimination from the beginning. IND≠0 indicates that it will be solved immediately by reuse of the LU-decomposition components obtained previously. For this case, A and MAX must hold the content of the previous call. Output: 0 indicates normal termination of calculation. 30000 indicates that no calculation has been done because the restrictions on the argument were not observed. K indicates that the equation was decided as singular and elimination terminated at step K.

*1 A, X, and W are assumed to be double precision real types (complex type and double precision complex type) for LEQBDW(X,Y).

EPS is changed to a double precision real type for LEQBDW/Y.

(3) Example

A program for solving an equation with 1000 unknowns (N=1000), total band width 7 (NB=7), and left band width 3 (LB=3) is shown below:

```
C       TEST FOR LEQBDV
        DIMENSION A(10,1000),MAX(1000),X(1000),W(1000)
        N=1000
        NB=7
        LB=3
        KA=10
        EPS=1.0E-6
        NX=1
        C=1.
        DO 10 I=1,N
        A(1,I)=C
        A(2,I)=-C
        A(3,I)=0
        A(4,I)=0
        A(5,I)=C
        A(6,I)=0.
   10   A(7,I)=-C
        DO 20 L=1,2
        IND=L-1
        DO 11 I=1,N
   11   X(I)=0.0
```

```
      X(2)=-C*C
      X(N-2)=C*C
      X(N-3)=C*C
      CALL CLOCKM(KO)
      CALL LEQBDV(A,KA,N,NB,LB,MB,X,N,NX,MAX,EPS,W,IND)
      CALL CLOCKM(K1)
      KO=K1-KO
      AM=0.0
      DO 12 I=1,N
      AA=ABS(X(I)-C)
      IF(AA.LE.AM) GO TO 12
      AM=AA
      MM=I
   12 CONTINUE
      WRITE(6,600) IND,KO,AM,MM,MB
  600 FORMAT(//10X,'ILL =',I6,5X,'TIME =',I6,5X,'ERR =
     1 ',1PE10.2, 5X,*'IMAX =',I6,5X,'MB =',I6/)
   20 CONTINUE
      STOP
      END
```

(4) Notes

1. This routine has the facilities of simultaneous processing of multiple right-hand columns and reuse of LU-decomposition components. Therefore, it can work as both a linear equation routine and an inverse matrix routine. Especially, the reuse of LU-decomposed components is important. It almost eliminates the need for calculation of inverse matrices.

2. Because rows are interchanged for pivoting, if necessary, the original number of columns of the coefficient matrix generally increases. Therefore, KA must be prepared to meet the condition MB≦KA.

   Because MB=NB+LB-1 even in the worst case, it is safe to take KA=NB+LB-1.

3. If there is a large difference between the size of the coefficients of the equations, it is desirable to normalize the coefficient matrix beforehand so that the maximum absolute value of the coefficient of each equation becomes the order of 1.

4. The recommended standard value of EPS is about $a\times10^{-6}$ ($a\times10^{-16}$) for LEQBDV/X (LEQBDW/Y) when the typical size of an element of the coefficient matrix is supposed to be a.

5. For a positive symmetric coefficient matrix, it is wiser to use special routines such as CHLBDV.

6. Argument IND is used for both input and output. So, do not use a constant as the actual argument for this.

(1989. 04. 06)

94

LEQLSS/D/Q/C/B

(least squares solution and minimum norm solution of general system of linear equations by Householder transformation)

Least Square and Minimum Norm Solutions of General Simultaneous Linear Equations by Householder Transformation

Programmed	Ichizo Ninomiya March, 1979
Format	Subroutine Language; FORTRAN Size; 94, 94, 94, 94 lines

(1) Outline

When matrix A with m rows n columns ($m \geqq n \geqq 1$) and n as rank and matrix B with m rows l columns are given,

Least squares solution $X=(A^TA)^{-1}A^TB$ (n rows l columns) ($X=(\overline{A}^TA)^{-1}A^{-T}B$ for complex number) of overdetermined system of linear equations $AX=B$ is calculated by A triangulation with Householder transformation. When the similar matrix A and matrix B with n rows k columns are given, minimum norm solution $X=A(A^TA)^{-1}B$ (m rows k columns) ($X=\overline{A}(A^T\overline{A})^{-1}B$) for complex number) of underdetermined system of linear equations $A^TX=B$ is calculated with a similar method.

(2) Directions

CALL LEQLSS/D/Q/C/B(A, KA, M, N, X, KX, NX, EPS, R, Q, ISW, ILL)

Argument	Type and kind *	Attribute	Content
A	Real type Two-dimensional array	Input	Coefficient matrix. Triangulation is done by Householder transformation.

Argument	Type and kind *	Attribute	Content
KA	Integer type	Input	Value of the first subscript in array declaration of A. KA≧M
M	Integer type	Input	Number of rows of A. M≧N
N	Integer type	Input	Number of columns of A. N≧1
X	Real type Two-dimensional array	Input/Output	When right side matrix B is input to call this routine, solution matrix X is generated. Two-dimensional array with M rows NX columns.
KX	Integer type	Input	Value of the first subscript in array declaration of X. KX≧M
NX	Integer type	Input	Number of columns of X. Only triangulation of A is done if NX≦0.
EPS	Real type	Input	The criterion constant ε for the rank degeneration of A. The rank is judged to be degenerated when the absolute value of the diagonal element is smaller than that of $\|A\|_1 \cdot \varepsilon$ in the process of triangulation, and the processing is interrupted. The minimum unit of the round-off error is set as a standard value if EPS≦0.
R	Real type One-dimensional array	Output	One-dimensional array of size NX. The residual norm or the norm of each solution vector (each column of X) is generated.
Q	Real type one-dimensional array	Work area	One-dimensional array of size N.

Argument	Type and kind *	Attribute	Content
ISW	Integer type	Input	The least squares solution is calculated if ISW≧0 and the minimum norm solution is calculated if ISW<0. The triangulation of A is done if |ISW|≦1, and the triangulation of A is omitted if |ISW|>2.
ILL	Integer type	Input	ILL=0: Normal termination. ILL=20000: Rank degeneration. ILL=30000: Input variable error

* All real types are assumed to be a double precision real type if the subroutine is for double precision.

A, X, and Q are assumed to be a (double precision) complex type if the subroutine is for the (double precision) complex number.

(3) Performance

In the current method to calculate the least squares solution (minimum norm solution), A^TA is created from coefficient matrix $A(A^T)$ and system of linear equations with this as a coefficient is solved. Therefore, the condition of the equation deteriorates and it is difficult to obtain the solution with good accuracy.

On the other hand, the condition does not deteriorate in this routine because A is transformed into upper triangular matrix $U=HA$ without creating A^TA by Householder's orthogonal transformation H. Therefore, the accuracy of the solution is excellent though it is a little inferior to the current method in the point of quantity of calculation.

The following table shows the result of the accuracy experiment of the numerical solution when A is the first N column of Hilbert matrix ($a_{ij}=1/(i+j-1)$) of the order M, and B is given so that all elements of the strict solution may become 1.

	Least squares solution N=10		Minimum norm solution N=10	
	M=10	M=20	M=10	M=20

LEQLSD	5 digits	8 digits	6 digits	4 Digits
Current method CHOLFD using	IND=8 interrupte d	IND=9 interrupte d	IND=8 interrupte d	IND=9 interrupte d

(4) Example

The following program is to calculate the least squares solution in the above experiment:

```
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(20,10),X(20),Q(10)
      M=20
      N=10
      EPS=1.0D-17
      ISW=0
      DO 10 I=1,M
      X(I)=0.0D0
      DO 10 J=1,N
      A(I,J)=1.0D0/DFLOAT(I+J-1)
   10 X(I)=A(I,J)+X(I)
      CALL LEQLSD(A,M,M,N,X,M,1,EPS,R,Q,ISW,ICON)
      WRITE(6,600) ICON,R,(X(I),I=1,N)
  600 FORMAT(1H1,I10,D25.17/(1H ,10X,D25.17))
      STOP
      END
```

(5) Calculation method

Matrix M with m rows n columns $(m \geq n \geq 1)$ and vector v of the order m are divided into two parts with $m-n$ rows and n rows and written as follows :

$$M=\begin{pmatrix} M_1 \\ M_2 \end{pmatrix} \quad v=\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

The following explains the case when the right side is vector b.

1. Least squares solution

Norm $\|r\|_2$ of residual $r=Ax-b$ is minimized. Householder's orthogonal transformation matrix H is multiplied to the left of A :

$$HA=\begin{pmatrix} U_1 \\ O_2 \end{pmatrix}$$

, where U_1 is the right upper triangular matrix, and O_2 is zero matrix.

It is sufficient to minimize $\|Hr\|_2$ because $Hr=HAx-Hb$ and $\|Hr\|_2 = \|r\|_2$. Putting

$$Hb=\begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix}$$

, we have

$$Hr=\begin{pmatrix} U_1x-\tilde{b}_1 \\ -\tilde{b}_2 \end{pmatrix}$$

Therefore, the least squares solution is calculated as $x=U_1^{-1}\tilde{b}_1$ by Back-substitution method of the upper triangular matrix and the folowing relation holds :

$$\|r\|_2 = \|Hr\|_2 = \|\tilde{b}_2\|$$

2. Minimum norm solution

Out of the infinitely many solutions of $A^Tx=b$, the one with minimum $\|x\|_2$ is calculated. $(HA)^THx=b$ is obtained by the same conversion as that for the least squares solution. Putting $y=Hx$, we obtain

$$(U_1{}^TO_2{}^T)\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}=b$$

, that is, $U_1{}^Ty_1=b$.

The minimum norm solution of this y is calculated by forward substitution method of lower triangular matrix $U_1{}^T$ and is given by

$$y=\begin{pmatrix} U_1^{-T}b \\ O_2 \end{pmatrix}$$

, where O_2 is assumed to be zero vector of the order $m-n$. When we calculate x by $x=H^Ty$ from y, x is the minimum norm solution because of $\|x\|_2 = \|H^Ty\|_2 = \|y\|_2$.

(6) Note

1. To solve system with identical A and different B many times, it is recommended to set $|ISW|\leq 1$ in the first call, and to set $|ISW|\geq 2$ in the subsequent calls with A and Q preserved, since in this way Householder transformation can be omitted.
2. When the rank of A is smaller than n, this subroutine cannot handle it . In such a case, it is better to use subroutine LSMNS/D based on the singular value decomposition.

(1987. 06. 23) (1987. 08. 07) (1987. 08. 11) (1987. 08. 21)

LEQLUS/D/Q/C/B/Z (Solution of linear equations by LU-decomposition method)

Solution of Linear Equations by LU-Decomposition Method

Programmed by	Ichizo Ninomiya; April 1977
Format	Subroutine language; FORTRAN (assembler for LEQLUS/D only) Size; 293, 241, 74, 75, and 76 lines respectively

(1) Outline

A number of linear equations that share a coefficient matrix are solved by the LU-decomposition method involving a row exchange for pivoting. That is, a solution $X=A^{-1}B$ of matrix equation $AX=B$ is obtained.

(2) Directions

CALL LEQLUS/D/Q/C/B/Z(A, KA, N, X, KX, M, DET, MAX, EPS, IND)

Argument	Type and Kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Specify a coefficient matrix. An LU-decomposition component of the coefficient matrix is determined and is overwritten. When this component is stored, it can be reused to eliminate the need for repeated LU-decomposition and thereby save computation time if an equation with the same coefficient needs to be solved at a later time (see the description of MAX and IND).
KA	Integer type	Input	Value of the first subscript in array declaration of A. KA≧N
N	Integer type	Input	Number of unknowns of the equation, that is, the number of rows in A. N≧2
X	Real type Two-dimensional array	Input/output	Specify several right-hand side columns of the equation. After processing of this routine, solution vectors which correspond to individual columns are overwritten.
KX	Integer type	Input	Value of the first subscript in array declaration of X. KX≧N

Argument	Type and Kind (*1)	Attribute	Content
M	Integer type	Input	Number of right-hand side columns, that is, the number of columns of matrix X. M≧0 If M = 0, only LU-decomposition of the coefficient matrix is done but the equation is not solved. If M = 1, a one-dimensional array is acceptable for the real argument of X.
DET	Real type	Input/output	If DET≠0, a coefficient determinant is output. If DET=0, the value remains unchanged.
MAX	Integer type One-dimensional array	Output	Information concerning the row exchange in LU-decomposition is entered using the name of one-dimensional array having the N number of elements . It is useful to store this information because it can be reused if an equation of the same coefficient needs to be solved later.
EPS	Real type	Input	Criterion constant for singularity of coefficient matrix. The calculation is interrupted because of singularity if the absolute value of a pivot element becomes smaller than this constant. EPS>0
IND	Integer type	Input/output	As an input variable, IND=0 indicates that the equation should be solved by LU-decomposition, and IND≠0 indicates that the equation should be solved immediately by using the result of previous LU-decomposition. For this, A and MAX must be retained the same as those when this subroutine was called previously. As an output variable, 0 indicates that calculation ends successfully, 30000 indicates that no calculation has been done because the restrictive conditions for the argument was violated, and a value K indicates that LU-decomposition was interrupted at the K-th stage of elimination by the singularity test.

*1 A, X, and DET each are a double precision real type (quadruple precision real type, complex type, double precision complex type, or quadruple precision complex type) for LEQLUD, LEQLUQ, LEQLUC, LEQLUB, and LEQLUZ.

EPS is a double precision real type (quadruple precision real type, real type, double precision real type, or quadruple precision real type) for LEQLUD, LEQLUQ, LEQLUC, LEQLUB,

and LEQLUZ.

(3) Performance

Because LEQLUS and LEQLUD are written with the assembler, they run fast and efficiently. Moreover, because double precision operation is partially used for LEQLUS, round-off errors are minimized and accuracy is improved accordingly.

(4) Notes

1. This routine has the functions of simultaneous processing of several right-hand-side columns, calculation of determinants, and reuse of LU-decomposition components. Therefore, it can also work as routines to solve linear equations and calculate determinants and inverse matrices. Especially, the function of reuse of LU-decomposition is most important. This function almost eliminates the need for calculation of inverse matrices.

If there is a substantial difference between the absolute values of coefficients in the equations, it is desirable to normalize the coefficient matrix in advance by MNORMS or MNORMD to secure precision.

3. If a typical absolute value of coefficient matrix elements is assumed to be a, $a\times10^{-6}(a\times10^{-16},a\times10^{-30})$ is adequate for the standard value of EPS for LEQLUS, LEQLUD, or LEQLUQ.

4. When the coefficient matrix is a symmetric positive definite, it is wiser to use special routines CHOLFS and CHOLFD, etc.

5. Arguments DET and IND are used for both input and output. Therefore, do not use constants as real arguments for them.

(1987. 06. 19) (1987. 08. 07)

LEQLUV/W/X/Y (Solution of linear equations by LU-decomposition method -vector version-)

Solution of Linear Equations by LU-Decomposition Method -Vector Version-

Programmed by	Ichizo Ninomiya; May 1986, December 1984
Format	Subroutine language; FORTRAN77 Size; 201, 202, 205, and 206 respectively

(1) Outline

A number of linear equations that share a coefficient matrix are solved by the LU-decomposition method involving row interchange for pivoting. That is, a solution $X=A^{-1}B$ of matrix equation $AX=B$ is determined. LEQLUV is for single precision, LEQLUW is for double precision, LEQLUX is for single precision complex numbers, and LEQLUY is for double precision complex numbers.

(2) Directions

CALL LEQLUV/W/X/Y(A, KA, N, X, KX, M, DET, LIST, EPS, W, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	A coefficient matrix is input. After processing by this routine, the LU-decomposition components of coefficient matrix are entered. These components are stored so that they can be used when it is later needed to solve equations with the same coefficients. This can eliminate the need for repeating LU-decomposition, thus saving computation time (see the descriptions of LIST and IND).
KA	Integer type	Input	Value of the first subscript in array declaration of A. KA≧N
N	Integer type	Input	Number of unknowns of equation, that is, the number of rows in A. N≧2

Argument	Type and kind (*1)	Attribute	Content
X	Real type two-dimensional array	Input/output	One or more right-hand side columns of equations are input. After processing by this routine, corresponding solution vectors are output.
KX	Integer type	Input	Value of the first subscript in array declaration of X. KX≧N
M	Integer type	Input	Number of right-hand side columns, that is, the number of columns of matrix X. M≧0 When M=0, only LU-decomposition of the coefficient matrix is performed. When M=1, the real argument for X can be a one-dimensional array.
DET	Real type	Input/output	When a value other than 0 is input, the coefficient determinant is output. When 0 is input, it is retained as it is.
LIST	Integer type one-dimensional array	Output	A one-dimensional array with N elements. Information concerning row interchange in LU-decomposition is kept in it. If this information is preserved, it can be reused when an equation having the same coefficients needs to be solved.
EPS	Real type	Input	Tolerance for test of singularity of coefficient matrix. If the absolute value of a pivot element becomes smaller than this constant, the matrix is decided as singular and calculation is interrupted. EPS>0

the facility of reuse of LU-decomposition is most important, eliminating the need for calculation of inverse matrix.

2. If a typical absolute value of coefficient matrix elements is assumed to be a, $a\times10^{-6}(a\times10^{-16})$ is reasonable value of EPS for LEQLUV and LEQLUX (LEQLUW and LEQLUY).

3. When the coefficient matrix is symmetric positive definite, it is wiser to use special routines CHOLFV/W and MCHLFV/W etc.

4. Arguments DET and IND are used for both input and output. Therefore, do not use constants as actual arguments for them.

(1987.06.19) (1987.08.07)

LSMNS/D (Least Squares and Minimum Norm Solutions of General Simultaneous Linear Equations by Singular Value Decomposition)

Least Squares and Minimal Norm Solutions of General Simultaneous Linear Equations by Singular Value Decomposition

Programmed by	Ichizo Ninomiya, March 1979
Format	Subroutine language: FORTRAN; size: 194, 194 lines respectively

(1) Outline

LSMNS/D finds an $n \times l$ matrix X that minimizes

$$\| Ax_i - b_i \|_2 \qquad i=1,2,\cdots,l$$

and

$$\| x_i \|_2 \qquad i=1,2,\cdots,l$$

if an $m \times n$ matrix A and an $m \times l$ matrix B are given. Where,

$$B=[b_1, b_2, \cdots, b_l]$$

and

$$X=[x_1, x_2, \cdots, x_l]$$

When this type of least square and minimal norm solutions are to be found, A is first decomposed as

$$A=U\Sigma V^T$$

by the singular value decomposition method, where U is an $m \times n$ matrix, and Σ and V are $n \times n$ matrices,

$$U^TU=V^TV=VV^T=I_n \quad (n \times n \text{ unit matrix})$$

$$\Sigma=diag(q_1, q_2, \cdots, q_n)$$

$q_1 \geqq q_2 \geqq \cdots\cdots \geqq q_n \geqq 0$ are the singular value of A, that is, the positive square roots of eigenvalues of A^TA.

The solution X is then given by

$$X=V\Sigma^+U^TB$$

Where,

$$\Sigma^{+}=diag(q_1^{+},q_2^{+},\cdots,q_n^{+})$$

$$q_i^{+}=\begin{cases}1/q_i & q_i>0 \quad i=1,2,\cdots,n\\ 0 & q_i=0 \quad i=1,2,\cdots,n\end{cases}$$

(2) Directions

CALL LSMNS/D(A, KA, M, N, B, KB, NB, Q, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	If a coefficient matrix is input, the orthogonal matrix V is output to the first N-th row of it. Array of max (M, N) rows and N columns.
KA	Integer type	Input	Value of the first subscript in the array declaration of A. KA≧max(M, N)
M	Integer type	Input	Number of rows of A. M≧1
N	Integer type	Input	Number of columns of A. N≧1
B	Real type Two-dimensional array	Input/output	If a right side matrix is input, the solution matrix X is output to the first N-th row of it. Array of max (M, N) rows and NB columns.
KB	Integer type	Input	Value of the first subscript in the array declaration of B. KB≧max(M, N)
NB	Integer type	Input	Number of columns of B. NB≧1
Q	Real type One-dimensional array	Output	The singular value of A is output in descending order. One-dimensional array of size N.
EPS	Real type	Input	Constant ε used for convergence test and test for singular values. If a double diagonal matrix obtained by bilateral Householder transformation from A is denoted by J, $\|J\|_\infty \cdot \varepsilon + u$ is used as a threshold value for zero test of the non-diagonal element and singular values of A. If EPS≦0, the rounding unit error u is used as ε.

Argument	Type and kind (*1)	Attribute	Content
W	Real type One-dimensional array	Work area.	One-dimensional array of size N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=2000: Singular value decomposition does not converge in 30N iteration. ILL=30000: Input arguments violate the limits.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Performance

The problem described on page 418 in the literature [1] are solved by LSMNS. In that problem, A is an 8 x 5 rank 3 matrix with singular values $\sqrt{1248}, 20, \sqrt{384}, 0, 0$ and B is an 8 x 3 matrix. $EPS=10^{-6}$ is chosen and the transformation matrix V was overwritten on A. The precision of the singular value Q, transformation matrix V, and three least squares and minimal norm solutions was about six digits.

(4) Example of use

The program for the above test is as follows:

```
      DIMENSION A(8,5),B(8,3),Q(5),W(5),R(5)
      M=8
      N=5
      KA=8
      KB=8
      NB=3
      EPS=1.E-6
      R(1)=SQRT(1248.)
      R(2)=20.
      R(3)=SQRT(384.)
      R(4)=0.
      R(5)=0.
      READ(5,500) ((A(I,J),J=1,N),I=1,M)
  500 FORMAT(5F4.0)
      READ(5,510) ((B(I,J),J=1,NB),I=1,M)
  510 FORMAT(3F4.0)
      WRITE(6,600) M,N,NB,((A(I,J),J=1,N),I=1,M)
     *,((B(I,J),J=1,NB),I=1,M)
  600 FORMAT(1H1///10X,'M =',I2,2X,'N =',I2,2X,
     *'NB',I2//8(10X,1P5E13.5/)/(10X,3E13.5))
      CALL LSMNS(A,KA,M,N,B,KB,NB,Q,EPS,W,ICON)
```

```
      WRITE(6,610) EPS,ICON,(Q(J),R(J),J=1,N),
     *((A(I,J),J=1,N),I=1,N),((B(I,J),J=1,NB),I=1,N)
  610 FORMAT(//10X,'EPS =',1PE10.2,2X,'ICON =',I6//
     *5(10X,2E13.5/)/ 5(10X,5E13.5/)/(10X,3E13.5))
      STOP
      END
```

(5) Remarks

1. The constant EPS used for the convergence criterion of singular value decomposition and zero test of singular values must be selected carefully. If EPS that is too small as compared with the precision of A and B is given, unnecessarily precise computation will be executed, or a singular value that should be discarded as 0 may be assumed to be significant. Conversely, too large EPS may cause a small, but meaningful singular value to be discarded as zero.

2. If least squares and minimal norm solution is found only once for the same coefficient matrix A, it is not wise from the standpoint of computation time to find a generalized inverse matrix A^+ using GINVS/D except that A^+ itself is required. By all means, the routine LSMNS/D should be used in this case.

References

1) G. H. Golub, C. Reinsch; "Singular Value Decomposition and Least Squares Solutions", Numerische Mathematik, 14, pp. 403-420 (1970).

(1987. 06. 16)

PRCGFS/D and RECGFS/D (Solution of a linear system of equations with positive definite symmetric coefficient matrix by conjugate gradient method with preconditioning)

Solution of a Linear System of Equations with Positive Definite Symmetric Coefficient Matrix by Conjugate Gradient Method with Preconditioning

Programmed	Tsuyako Miyakoda and Tatsuo Torii; February 1982
Format	Subroutine language; FORTRAN Size; 85 and 86 lines respectively

(1) Outline

Each of these subroutines solves a linear system of equations Ax=b for x, where A is a positive definite symmetric dense matrix and x and b are vectors. To do this, it performs preprocessing to improve convergence conditions and then uses the conjugate gradient method.

It is useful to correct an approximate solution vector which is already known. RECGFS (RECGFD) is provided as an entry name used to perform calculation again by skipping preconditioning after PRCGFS (PRCGFD) is once called.

(2) Conditions

```
CALL PRCGFS/D(A, NA, N, B, X, OMEGA, EPS, NMAX, W, IDUMP)
CALL RECGFS/D(A, NA, N, B, X, OMEGA, EPS, NMAX, W, IDUMP)
```

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Coefficient matrix. This is not retained.
NA	Integer type	Input	Adjustable dimension of A. NA≧N
N	Integer type	Input	Number of unknowns of a system of equations.

Argument	Type and kind (*1)	Attribute	Content
B	Real type one-dimensional array	Input	Right-side vector of the system. Size N
X	Real type one-dimensional array	Input/output	Input: Approximate solution vector Output: Solution vector
OMEGA	Real type	Input	Acceleration factor for convergence in iteration method. 1≦OMEGA<2. If a value outside the range is input, 1 is used for calculation.
EPS	Real type	Input	Tolerance for convergence test. Convergence is assumed when the sum of squares of the residuals is smaller than EPS**2. If EPS is too small, however, 8·u·‖b‖ is used for it. u is the unit of rounding errors.
NMAX	Integer type	Input	Maximum number of iterations. Theoretically, the value of NMAX is N at most. If given NMAX is too large, it will be replaced by 3N/2.
W	Real type one-dimensional array	Work area	Size N*3

Argument	Type and kind (*1)	Attribute	Content
IDUMP	Integer type	Input/output	On entry, this argument has the following meanings: IDUMP≦0: During calculation, no printing is done. IDUMP=1: During calculation, the norm of residuals and (p, Ap) in each iteration are printed. IDUMP≧2: The approximate solution, residuals, and A-orthogonal vectors in each iteration are printed. On return, this argument has the following meanings: IDUMP=0: Normal termination. IDUMP=NMAX: Convergence does not occur in NMAX iterations. IDUMP=30000: Input argument error.

*1 For double precision subroutines, real types are all assumed to be double precision real types.

(3) Calculation method

Regular division is done as A=M-N, where M have the same characteristics as A with symmetric positive definite and permit easy calculation to determine the inverse matrix. We think the system preconditioned by using M as follows:

$$M^{-1}Ax=M^{-1}b$$

Then, we obtain;

$$\tilde{A}=M^{-\frac{1}{2}}AM^{-\frac{1}{2}},\ \tilde{b}=M^{-\frac{1}{2}}b,\ y=M^{\frac{1}{2}}x$$

It is rewritten as

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

This is a positive definite system equivalent to the original system. We then apply an algorithm of the conjugate gradient method to this system.

Matrix M here is formed by the following method by (Nodera and Takahashi)[1] :It is decomposed as follows:

$$A=L_0+D+L_0^T$$

where

L_0 : Lower triangular matrix (diagonal elements 0)

D: Diagonal matrix

We then multiply

$D^{-\frac{1}{2}}$ from both sides of A.

$$A'=D^{-\frac{1}{2}}AD^{-\frac{1}{2}}=L_0'+I+K_0'^T$$

And, we put

$$C=I+\omega L_0'$$

to obtain

$$M^{-1}=(CC^T)^{-1}$$

where, ω is an acceleration parameter of the SOR method, satisfying $0<\omega<2$.

(4) Example

```
00010 C      MAIN FOR PRCGFS
00020        REAL*8 SU,A,X,B,W
00030        DIMENSION AA(100,100),B(100),A(100,100),XY(100),
00031       1    X(100),W(500)
00040        DIMENSION XO(100)
00050        NR=5
00060        NW=6
00070        EPS=1.E-5
00080        NA=100
00090        N=100
00100        XX=1.0E8+1.
00110        DO 1800 I=1,N
00120        DO 1810 J=1,N
00130        IJ=IABS(I-J)
00140  1810  A(I,J)=FLOAT(N-IJ)
00150  1800  CONTINUE
00160        XI=12345678.0
00170        DO 7 I=1,N
00180        XO(I)=0.0
00190        X(I)=4.*XI/1.E8-2.
00200        XI=AMOD(23.*XI,XX)
00210     7  CONTINUE
00220  1100  FORMAT(F12.0)
00230        DO 2000 IK=1,N
00240        SU=0.
00250        DO 2100 I=1,N
00260  2100  SU=SU+A(IK,I)*X(I)
```

```
00270  2000 B(IK)=SU
00280       WRITE(NW,1205)N
00290  1205 FORMAT(1H1,15H EXAMPLE 3-6 N=,I4)
00300       IF(N.GE.10)GO TO 19
00310       DO 5 I=1,N
00320     5 WRITE(NW,200)(A(I,J),J=1,N)
00330    19 CONTINUE
00340       WRITE(NW,203)(B(I),I=1,N)
00370   203 FORMAT(2H B//(4D23.15))
00380   200 FORMAT(2H A//(4D23.15))
00390       DO 20 I=1,N
00400       XY(I)=B(I)
00410       DO 20 J=1,N
00420    20 AA(I,J)=A(I,J)
00430       IDUMP=1
00440       OMEGA=1.00
00450       CALL CLOCKM(JTIME1)
00460       NMAX=100
00470       CALL PRCGFS(AA,NA,N,XY,XO,OMEGA,EPS,NMAX,W,IDUMP)
00480       CALL CLOCKM(JTIME2)
00490       JT=JTIME2-JTIME1
00500       WRITE(NW,300)IDUMP,JT
00510       DO 2200 I=1,N
00520       RES=X(I)-XO(I)
00530       WRITE(NW,303) I,X(I),XO(I),RES
00540  2200 CONTINUE
00550   303 FORMAT(I5,2E15.6,E11.3)
00560   300 FORMAT(7H IDUMP=,I5,3X,'TIME =',I5)
00570   302 FORMAT(2H X//(D23.15))
00580       STOP
00590       END
```

```
 EXAMPLE 3-6 N= 100
IDUMP=    0   TIME =  130
   1  -0.150617E+01  -0.150611E+01 -0.572E-04
   2   0.135802E+01   0.135791E+01  0.116E-03
   3  -0.765456E+00  -0.765381E+00 -0.755E-04
   4  -0.160547E+01  -0.160547E+01 -0.572E-05
   5  -0.925886E+00  -0.925903E+00  0.170E-04
```

Bibliography

1) T. Nodera and H. Takahasi; "Preconditioned Conjugate Gradient Algorithm for Solving Biharmonic Equation" 4th IMACS and International Symposium (1981)

(1987. 06. 19) (1987. 08. 08)

PRCGSS/D, RECGSS/D

(Solution of a linear system of equations with sparse positive definite symmetric coefficient matrix by conjugate gradient method with preconditioning (compressed matrix storage mode))

Solution of a Linear System of Equations with Sparse Positive Definite Symmetric Coefficient Matrix by Conjugate Gradient Method with Preconditioning (Compressed Matrix Storage Mode)

Programmed by	Tsuyako Miyakoda; 1982
Format	Subroutine language; FORTRAN Size; 192 and 193 lines respectively

(1) Outline

Each of these subroutines solves the linear system of equations Ax = b where coefficient matrix A having a relatively small number of nonzero elements is positive definite symmetric. To do this, it performs preconditioning to improve convergence conditions and then uses the conjugate gradient method. This solution routine is used when only non-zero elements of coefficient matrices are stored by rows in a one-dimensional array by the compressed storage mode.

It is useful to correct an approximate solution vector which is already known. RECGSS (RECGSD) is provided as an entry name used to perform calculation again by skipping preconditioning after PRCGSS (PRCGSD) is once called.

(2) Directions

```
CALL PRCGSS/D(IJTAB, A, LA, B, N, X, EPS, OMG, IMAX, IW, W, ILL)
CALL RECGSS/D(IJTAB, A, LA, B, N, X, EPS, OMG, IMAX, IW, W, ILL)
```

Argument	Type and kind (*1)	Attribute	Content
IJTAB	Two bytes Integer type Two-dimensional array	Input	The rows and columns numbers of non-zero elements are input to IJTAB(1, K) and IJTAB(2, K) respectively. Suppose $A(K)=a_{ij}$, for instance, we set: IJTAB(1, K)=i and IJTAB(2, K)=j. The size is 2*LA. Output data in arrays are rearranged in ascending order in the values of i, j.
A	Real type one-dimensional array	Input	The size is 2*LA. Only non-zero elements of the coefficient matrix are stored by row in an array of length LA appearing first. They are rearranged simultaneously with IJTAB. The non-zero elements of the preconditioned matrix are stored in the array of size LA appearing last.
LA	Integer type	Input	Number of non-zero elements of the coefficient matrix. LA≧N
B	Real type one-dimensional array	Input	The right-side vector of a system of equations. Size N
N	Integer type	Input	Number of unknowns of a system of equations. N≧3
X	Real type one-dimensional array	Input/output	Input; Approximate solution vector with size N. (Zero vector at first) Output; Solution vector.

Argument	Type and kind (*1)	Attribute	Content
EPS	Real type	Input/output	Tolerance for convergence test. When the sum of squares of the residuals is smaller than EPS**2, it is assumed that convergence has occurred. If EPS is too small, however, $8 \cdot u \cdot \|b\|$ is used instead. u is the unit of rounding errors.
OMG	Real type	Input	Acceleration factor for convergence in iteration method. $1 \leq OMG < 2$. If a value outside the range is input as OMG, OMG=1 is used for calculation.
IMAX	Integer type	Input/output	Input: Maximum number of iterations. Theoretically, it is N at most. If the given IMAX is too large, however, it is replaced by $3 \cdot N/2$. Output: Actual number of iterations.
IW	Integer type one-dimensional array	Working storage	The size is 2*N.
W	Real type one-dimensional array	Working storage	The size is N*3.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=IMAX: Convergence does not occur in IMAX iterations. ILL=25000: IJTAB error. ILL=30000: Input argument error.

*1 For double precision subroutines, real types are all assumed to be double precision real types.

(3) Example

```
C ... TEST OF PRCGSS ...
      DIMENSION A(40),B(6),X(6),IW(2,6)
      INTEGER*2 IJTAB(2,20)
      REAL*4 W(18)
      DATA N/    6/LA/20/
      DATA A(1),IJTAB(1,1),IJTAB(2,1)/10.0,1,1/
      DATA A(2),IJTAB(1,2),IJTAB(2,2)/-2.0,1,4/
      DATA A(3),IJTAB(1,3),IJTAB(2,3)/-1.0,1,5/
      DATA A(4),IJTAB(1,4),IJTAB(2,4)/-1.0,1,6/
      DATA A(5),IJTAB(1,5),IJTAB(2,5)/12.0,2,2/
      DATA A(6),IJTAB(1,6),IJTAB(2,6)/-3.0,2,3/
      DATA A(7),IJTAB(1,7),IJTAB(2,7)/-1.0,2,4/
      DATA A(8),IJTAB(1,8),IJTAB(2,8)/-2.0,2,6/
      DATA A(9),IJTAB(1,9),IJTAB(2,9)/-3.0,3,2/
      DATA A(10),IJTAB(1,10),IJTAB(2,10)/15.0,3,3/
      DATA A(11),IJTAB(1,11),IJTAB(2,11)/-2.0,4,1/
      DATA A(12),IJTAB(1,12),IJTAB(2,12)/-1.0,4,2/
      DATA A(13),IJTAB(1,13),IJTAB(2,13)/20.0,4,4/
      DATA A(14),IJTAB(1,14),IJTAB(2,14)/-5.0,4,5/
      DATA A(15),IJTAB(1,15),IJTAB(2,15)/-1.0,5,1/
      DATA A(16),IJTAB(1,16),IJTAB(2,16)/-5.0,5,4/
      DATA A(17),IJTAB(1,17),IJTAB(2,17)/1.0,5,5/
      DATA A(18),IJTAB(1,18),IJTAB(2,18)/-1.0,6,1/
      DATA A(19),IJTAB(1,19),IJTAB(2,19)/-2.0,6,2/
      DATA A(20),IJTAB(1,20),IJTAB(2,20)/6.0,6,6/
      DATA (B(I),I=1,6)/10.,-5.,28.5,37.5,-10.0,10./
      WRITE(6,630) (I,A(I),IJTAB(1,I),IJTAB(2,I),I=1,LA)
  630 FORMAT(I5,F10.3,2I3)
      ILL=0
      IMAX=100
      OMG=1.2
      EPS=1.E-4
      DO 100 I=1,N
  100 X(I)=1.0
      CALL PRCGSS(IJTAB,A,LA,B,N,X,EPS,OMG,IMAX,IW,W,ILL)
      WRITE(6,610) ILL,IMAX,N,OMG
  610 FORMAT(1H ,'ILL,IMAX,N,OMG=',3I6,F10.3)
      WRITE(6,620)(X(I),I=1,N)
  620 FORMAT(1H0,3E15.6)
      STOP
      END
```

<Output result>

```
    1     10.000  1  1
    2     -2.000  1  4
    3     -1.000  1  5
    4     -1.000  1  6
    5     12.000  2  2
    6     -3.000  2  3
    7     -1.000  2  4
    8     -2.000  2  6
    9     -3.000  3  2
   10     15.000  3  3
   11     -2.000  4  1
   12     -1.000  4  2
   13     20.000  4  4
   14     -5.000  4  5
   15     -1.000  5  1
   16     -5.000  5  4
```

```
  17      1.000   5   5
  18     -1.000   6   1
  19     -2.000   6   2
  20      6.000   6   6

ILL,IMAX,N,OMG=      0     7      6       1.200

  0.999998E+00    0.500000E+00    0.200000E+01

  0.100000E+01   -0.400000E+01    0.200000E+01
```

(4) Calculation method

See the calculation method for subroutines PRCGFS and PRCGFD which use the conjugate gradient method with the preconditioning.

(5) Notes

If the approximate solution vector is known as argument X, input it. Otherwise, input the zero vector.

Call RECGSS or RECGSD to restart the iterative calculation after PRCGSS or PRCGSD is once called. In this case, do not change the contents of arguments IJTAB, A, L, IW, and W. Also, call RECGSS or RECGSD to determine solutions when only the right-side vector B is changed.

Bibliography

1) Tsuyako Miyakoda; "Consideration on solution of linear equations, and reduction of iterations and quantity of calculation-conjugate gradient method" Osaka University computer center news, Vol. 12, No. 2, pp. 55-69 (1982)

(1987. 06. 19) (1987. 08. 08) (1987. 08. 21)

TRDSPS/D and TDSPCS/D (Solution of Symmetric Positive Definite Tridiagonal Equations)

Solution of Symmetric Positive Definite Tridiagonal Equations

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 29, 29, 45, and 46 lines respectively

(1) Outline

TRDSPS/D or TDSPCS/D solves simultaneous linear equations with a symmetric positive definite tridiagonal matrix as a coefficient matrix, using Cholesky decomposition method that does not use square roots.

TRDSPS/D is used if C(N)=0, and TDSPCS/D handles cyclic type tridiagonal equations, that is, if C(N)≠0. Both routines process multiple right side columns simultaneously, calculate determinants, and can reuse Cholesky decomposition components.

(2) Directions

CALL TRDSPS/D(B, C, N, X, KX, M, DET, EPS, IND)

CALL TDSPCS/D(B, C, D, N, X, KX, M, DET, EPS, IND)

Argument	Type and kind (*1)	Attribute	Content
B	Real type One-dimensional array	Input/output	If coefficient matrix diagonal elements are input, Cholesky decomposition diagonal elements are output.
C	Real type One-dimensional array	Input/output	If coefficient matrix sub-diagonal elements are input, Cholesky decomposition sub-diagonal elements are output. The numbering of sub-diagonal elements is as shown in the figure.
D	Real type One-dimensional array	Output	The Cholesky decomposition elements of a coefficient matrix are output.
N	Integer type	Input	Order of equation. It is also the number of elements in the arrays B, C, and D. N≧3

Argument	Type and kind (*1)	Attribute	Content
X	Real type Two-dimensional array	Input/output	If M right side columns are input in the form a matrix X, the solution vectors are output in the corresponding places.
KX	Integer type	Input	Value of the first subscript in the array declaration of X. KX≧N
M	Integer type	Input	Number of columns in X. If M = 0, only Cholesky decomposition of a coefficient matrix is executed..
DET	Real type	Input/output	If DET≠0 is input, the value of coefficient matrix determinant is output. If DET=0 is input, DET=0 is output.
EPS	Real type	Input	Constant for determining the non-positivity of coefficient matrix. If the value of a pivot element is smaller than this constant, the input matrix is decided to be non positive definite, and the calculation is interrupted. EPS>0
IND	Integer type	Input/output	This argument has the following meaning as an input. IND=0: Solve an equation by newly executing Cholesky decomposition. IND≠0: Solve an equation, reusing the Cholesky decomposition elements previously calculated, and stored in B, C, and D. This argument has the following meaning as an output. IND=0: Calculation terminated normally. IND=30000: Limits on input arguments are violated. IND=I: Calculation is interrupted at the I-th stage of non positivity.

1* For double precision subroutines, all real types are changed to double precision real types.

(3) Performance

Generally, computation time is only proportional to the order of equation. If the same problem is solved with a general simultaneous linear equation routine (LEQLUS, CHOLFS, GAUELS, etc.), it takes very long time because computation time becomes proportional to the cubic power of the order of equations.

$$\begin{bmatrix} B_1 & C_1 & 0 & \cdots & \cdots & \cdots & 0 & C_n \\ C_1 & B_2 & C_2 & \cdots & \cdots & \cdots & 0 & 0 \\ 0 & C_2 & B_3 & \cdots & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & \ddots & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & B_{n-1} & C_{n-1} \\ C_n & 0 & 0 & \cdots & \cdots & \cdots & C_{n-1} & B_n \end{bmatrix}$$

(4) Remarks

1. TRIDGS/D is prepared for tridiagonal equations where coefficients are not symmetric positive definite.

2. When the same equation is to be repeatedly solved with the right side column changed, the calculation time can be saved by using the facility for reusing the Cholesky decomposition components of this routine.

(1987.06.17)

TRIDGS/D (Solution of Tridiagonal Equations)

Solution of Tridiagonal Equations

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 39 and 40 lines respectively

(1) Outline

TRIDGS/D solves a tridiagonal equation or a system of linear equations with a tridiagonal matrix as a coefficient matrix, using the Gauss' elimination accompanied by row interchange for pivot selection.

$$\begin{bmatrix} B_1 & C_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ A_1 & B_2 & C_2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & A_2 & B_3 & C_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & B_{n-2} & C_{n-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & A_{n-2} & B_{n-1} & C_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & A_{n-1} & B_n \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_{n-2} \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{n-2} \\ D_{n-1} \\ D_n \end{bmatrix}$$

(2) Directions

CALL TRIDGS/D(A, B, C, D, N, EPS, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type One-dimensional array	Input	Input N-1 lower diagonal elements of a coefficient matrix in the order of the upper left to the lower right. Destroyed.
B	Real type One-dimensional array	Input	Input N diagonal elements of a coefficient matrix in the order of the upper left to the lower right. Destroyed.
C	Real type One-dimensional array	Input	Input N-1 upper diagonal elements of a coefficient matrix in the order of the upper left to the lower right. Destroyed.

Argument	Type and kind (*1)	Attribute	Content
D	Real type One-dimensional array	Input/output	If the right side column of the equation is input, the solution vector calculated by this routine is output.
N	Integer type	Input	Order of equation. N≧3
EPS	Real type	Input	Criterion constant for singularity. If the absolute value of a pivot element is smaller than this constant, the equation is decided to be singular and the calculation is interrupted. Stored. EPS>0
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: When limits on N and EPS are violated. If an equation is decided to be singular, its pivot element number is output.

1* For double precision subroutines, all real types are changed to be double precision real types.

(3) Performance

Because precision depends on problems, nothing can be said generally. Computation time is only proportional to the order of equation. If the same problem is solved using a general simultaneous linear equation subroutine (LEQLUS, CHOLFS, GAUELS, etc.), it takes very long time because the computation time is proportional to the cubic power of the order of equations.

(4) Remarks

1. Tridiagonal equations can be solved even with a general simultaneous equation routine. However, it is reasonable to use this routine from the standpoint of computation time.

2. If the typical absolute value of elements in a coefficient matrix is a, $a\times10^{-6}(a\times10^{-16})$ is adequate for the standard value of EPS for TRIDGS (TRIDGD).

3. It is more advantageous for a symmetric positive definite tridiagonal equation to use the special-purpose routines TRDSPS and TRDSPD or TDSPCS and TDSPCD.

(1987.06.17)

129

3. Matrix inversion

GINVS/D (Generalized inverse matrix by singular value decomposition)

Generalized Inverses (Pseudo-inverses) by Singular Value Decomposition

Making	Ichizo Ninomiya; March 1979
Form	Subroutine language; FORTRAN, Size; 30 lines each

(1) Outline

Matrix X with n rows m columns which satisfies the following relations is called the generalized inverse to matrix A with m rows n columns.

$$AXA=A$$

$$XAX=X$$

$$(AX)^T=AX$$

$$(XA)^T=XA$$

For a given A, such X is determined uniquely. This X is denoted by A^+. Suppose that singular value decomposition

$$A=U\Sigma V^T$$

of A is given, where U is a matrix with m rows and n columns, Σ and V each are a matrix with n rows and n columns, and the following relation holds:

$$U^TU=V^TV=VV^T=I_n(n\text{-dimensional unit matrix})$$

$$\Sigma=diag(q_1,q_2,\cdots,q_n)$$

$$q_1\geqq q_2\geqq\cdots\geqq q_n\geqq 0$$

and $q_i, i=1,2,\cdots,n$ are singular values of A (positive square root of eigenvalue of A^TA).

Then, A^+ is given by:

$$A^+=V\Sigma^+U^T$$

Where,

$$\Sigma^+=diag(q_1^+,q_2^+,\cdots,q_n^+)$$

and

$$q_i^+=\begin{cases}1/q_i & q_i>0\\ 0 & q_i=0\end{cases}$$

is assumed to be satisfied.

The purpose of this subroutine is to determine A^+ by singular value decomposition when A is given.

(2) Directions

CALL GINVS/D(A, KA, M, N, Q, V, KV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	When A is input, transposed matrix $(A^+)^T$ of the generalized inverse matrix is generated.
KA	Integer type	Input	Value of the first subscript in array declaration of AKA≧N
M	Integer type	Input	Number of rows in AM≧1
N	Integer type	Input	Number of columns in AN≧1
Q	Real type One-dimensional array	Output	Singular values of A are generated in descending order. One-dimensional array of size.
V	Real type Two-dimensional array	Output	Orthogonal transformation matrix V for singular value decomposition is generated. Two-dimensional array with N rows and N columns.
KV	Integer type	Input	Value of the first subscript in array declaration of VKV≧N

Argument	Type and kind (*1)	Attribute	Content
EPS	Real type	Input	Constant ε used for convergence and 0 test $\|J\|_\infty$ determined when A is once converted into double diagonal matrix J by bilateral Householder transformation is used to make $\varepsilon\|J\|_\infty+u$ a threshold of convergence test for singular value decomposition and 0 test of singular values. Where, u denotes the unit of round-off errors. u is used as ε when EPS≦0. 0 is input.
W	Real type One-dimensional array	Work area	One-dimensional array of size N.
ILL	Integer type	Output	ILL=0: Normal end. ILL=30000: The argument violates the limit. ILL=20000: Singular value decomposition of A does not converge even after iteration of 30N times.

*1 For double precision subroutines, all real types should be changed to double precision real types.

(3) Performance

The following is described on page 418 of bibliography [1]: For a problem having matrix B with 8 rows and 3 columns at the right-hand side and using matrix A of rank 3 having 8 rows and 5 columns and singular values $\sqrt{1248}, 20, \sqrt{384}, 0, 0$ as a coefficient, A^+ is first determined by GINVS, and A^+B is then used to calculate the least squares minimal norm solution. When $EPS=10^{-6}$, the accuracy for singular value Q, transformation matrix V, general matrix A^+, and solution vector A^+B was about six decimal digits.

(4) Example

A program to examine the above description is shown below.

```
      DIMENSION A(8,5),B(8,3),V(5,5),Q(5),W(8),R(5)
      M=8
      N=5
      NB=3
      KA=8
      KV=5
      EPS=1.E-6
      R(1)=SQRT(1248.)
      R(2)=20.
      R(3)=SQRT(384.)
      R(4)=0.
      R(5)=0.
      READ(5,500) ((A(I,J),J=1,N),I=1,M)
  500 FORMAT(5F4.0)
      READ(5,510) ((B(I,J),J=1,NB),I=1,M)
  510 FORMAT(3F4.0)
      WRITE(6,600) M,N,NB,((A(I,J),J=1,N),I=1,M)
     *,((B(I,J),J=1,NB),I=1,M)
  600 FORMAT(1H1///10X,'M =',I2,2X,'N =',I2,2X,'NB =',
     *I2//8(10X,1P5E13.5/)/(10X,3E13.5))
      CALL GINVS(A,KA,M,N,Q,V,KV,EPS,W,ICON)
      DO 30 J=1,NB
      DO 10 I=1,M
   10 W(I)=B(I,J)
      DO 30 I=1,N
      S=0.
      DO 20 K=1,M
   20 S=A(K,I)*W(K)+S
   30 B(I,J)=S
      WRITE(6,610) EPS,ICON,(Q(J),R(J),J=1,N)
     *,((V(I,J),J=1,N),I=1,N),((A(I,J),I=1,M),J=1,N)
     *,((B(I,J),J=1,NB),I=1,N)
  610 FORMAT(//10X,'EPS =',1PE10.2,2X,'ICON =',I6//
     *5(10X,2E13.5/)/5(10X,5E13.5/)/5(10X,8E13.5/)
     */(10X,3E13.5))
      STOP
      END
```

(5) Notes

1. The constant EPS used for the convergence test of singular value decomposition and the 0 test of singular values must be specified carefully. If the EPS is too small for the accuracy of data A, unnecessarily and wastefully precise calculation may be done and a singular value which should normally be discarded as 0 may be taken for a significant value. On the contrary, if the EPS is too large, a singular value which is small but significant may be discarded as 0.

2. If the least squares minimal norm solution is calculated only once for a given coefficient matrix A, it is not wise to use this routine GINVS or GINVD to determine A^+ except when A^+ itself is required. This is because the routine requires large quantity of calculation. LSMNS or LSMND should be used for this case.

Bibliography

1) G. H. Golub, C. Reinsch; "Singular Value Decomposition and Least Squares Solutions", Numerische Mathematik, 14, pp. 403-420 (1970).

(1987. 06. 22)

136

MINVS/D/Q/C/B/Z (Inversion of Matrices)

Inversion of Matrices

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 96, 97, 96, 95, 96, and 96 lines respectively

(1) Outline

MINVS/D/Q/C/B/Z generates an inverse matrix of a given matrix in place of the given matrix, using the LU-decomposition method. The rows are interchanged if necessary for pivot selection.

(2) Directions

CALL MINVS/D/Q/C/B/Z(A, KA, N, EPS, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	If a matrix is input, its inverse matrix is output.
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Order of A. 2≦N≦1000
EPS	Real type	Input	Criterion constant for matrix singularity. If the absolute value of a pivot element is smaller than this constant, the input matrix is decided to be singular, and the calculation is interrupted. EPS>0
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: Limits on KA, N, and EPS are violated. The number of the pivot element whose absolute value is smaller than EPS.

*1 For MINVD (MINVQ, MINVC, MINVB, MINVZ), A is a double precision real type (quadruple

precision real type, complex type, double precision complex type, and quadruple precision complex type).

For MINVD (MINVQ, MINVC, MINVB, MINVZ), EPS is a double precision real type (quadruple precision real type, real type, double precision real type, and quadruple precision real type).

(3) Calculation method

1. The permutation matrix P corresponding to row interchanges accompanying to pivoting is applied to A and then PA is decomposed into a lower unit triangular matrix L and an upper triangular matrix. $PA=LU$

2. Generates L^{-1} in place of L.

3. Generates U^{-1} in place of U.

4. Generates $A^{-1}=U^{-1}L^{-1}P$ in place of A.

In case of MINVS, all the necessary inner sum computation are done by partial double precision arithmetic operation.

(4) Remarks

1. If the absolute values of matrix elements differs significantly, it is desirable to normalize the matrix in advance by MNORMS and MNORMD to insure precision in the result. For the required post-processing, see the explanation of MNORMS.

2. If the typical value of matrix elements is a, $a\times10^{-6}(a\times10^{-16}, a\times10^{-30})$ is adequate as the standard value of EPS for MINVS and (MINVD, MINVQ).

3. It is very disadvantageous from the standpoint of computation time and precision to calculate the inverse matrix of A for the calculation of matrix product of the form of $A^{-1}B$. By all means, the simultaneous linear equation routines LEQLUS and LEQLUD should be used.

4. When the inverse matrix of a symmetric positive definite matrix is to be found, it is wise to use the special-purpose routines MINVSP and MINVDP.

(1987. 06. 17) (1987. 08. 07)

MINVSP/MINVDP/MINVQP (Inversion of Symmetric Positive Definite Matrices)

Inversion of Symmetric Positive Definite Matrices

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 41, 41, and 41 lines respectively

(1) Outline

MINVSP/MINVDP/MINVQP generates the inverse matrix of a symmetric positive definite matrix A in place of the input matrix using the Cholesky decomposition method.

(2) Directions

CALL MINVSP/MINVDP/MINVQP(A, KA, N, EPS, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	If a symmetric positive definite matrix is input, its inverse matrix is output. This argument processes only the upper right half including the diagonal lines because of symmetry. The lower left half is preserved.
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Order of A. N≧2
EPS	Real type	Input	Constant for determining the positivity of matrix A. If the value of a pivot element is smaller than this constant, the input matrix is decided to be non positive definite, and the calculation is interrupted. EPS>0
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: Limits on KA, N, and EPS are violated. The number of the pivot element whose absolute value is smaller than EPS

*1 For MINVDP (MINVQP), all real types are changed to double (quadruple) precision real types.

(3) Calculation method

1. Generates the Cholesky decomposition element U of A, that is, the upper triangular matrix U such that $A=U^TU$ in the upper right triangular part of A.

2. Generates the inverse matrix U^{-1} of U, that is, the upper triangular matrix V such that

$UV=I$ in place of U.

3. Generates the upper right half of the inverse matrix $A^{-1}=VV^T$ of A in place of V.

In case of MINVSP, execute all necessary inner sum calculations by partial double precision arithmetic operation.

(4) Remarks

1. If the typical absolute value of matrix elements is a, $a\times10^{-6}(a\times10^{-16},a\times10^{-30})$ is adequate as the standard value of EPS for MINVSP, MINVDP, and MINSQP.

2. It is very disadvantageous from the standpoint of computation time and precision to calculate the inverse matrix of A for the calculation of matrix product of the form of $A^{-1}B$. By all means, the simultaneous linear equation routines CHOLFS and CHOLFD should be used.

(1987.06.16)

MINVV/W/X/Y (Inversion of Matrix - Vector Version -)

Inversion of Matrix -Vector Version-

Programmed by	Ichizo Ninomiya and Yasuyo Hatano, March 1985
Format	Subroutine language: FORTRAN77; size: 112, 113, 111, and 113 lines respectively

(1) Outline

MINVV/W/X/Y obtains an inverse matrix using the Gauss-Jordan elimination. It is for single precision (double precision, single precision complex type, or double precision complex type).

(2) Directions

CALL MINVV/W/X/Y(A, KA, N, EPS, LIST, W, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	The input matrix is processed with this routine, and its inverse is generated.
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Order of A. N≧2
EPS	Real type	Input	Matrix singularity criterion. If the absolute value of pivot elements is smaller than this constant, it is determined to be singular, and the computation is interrupted. EPS>0

Argument	Type and kind (*1)	Attribute	Content
LIST	Integer type one-dimensional array	Work area	One-dimensional array containing N elements.
W	Real type one-dimensional array	Work area	One-dimensional array of size 2N.
IND	Integer type	Output	The value 0 is assumed if computation terminates normally, and 30000 is assumed if computation is not executed at all because limits on the argument are exceeded. Value of K is assumed if computation is stopped at the K-th step because of singularity.

*1 For MINVW(X, Y), A and W are changed to double precision real types (complex type or double precision complex type).
For MINVW/Y, EPS is changed to a double precision real type.

(3) Calculation method

The Gauss-Jordan elimination accompanied by row exchange for partial pivoting is used.

(4) Note

1. If the typical absolute value of matrix elements is a, $a\times10^{-6}$($a\times10^{-16}$) is adequate as the standard value of EPS for MINVV(W).

(1987. 06. 19) (1987. 08. 07)

4. Eigenvalue analysis

[Method of choice of eigenvalue analysis routines]

NUMPAC provides a variety of effective eigenvalue analysis routines that you can select depending on the type, characteristics, and structure of each target matrix. By carefully selecting them based on the guideline shown below, you can enjoy much of their superiority in all aspects of precision, speeds, and storage capacities. To make the following explanation simple, the name of each recommended routine is represented by the one for single precision. In addition to the routines below, the high-speed eigenvalue analysis package NICER is also available.

1. Non-symmetry: HEQRVS

2. Symmetry

   (1) Dense matrix

      (a) To obtain all eigenvalues and eigenvectors: HQQRVS

      (b) To obtain all eigenvalues and all or part of eigenvectors: HQRIIS

      (c) To obtain part of eigenvalues and eigenvectors: HQBSVS

      (d) To obtain a limited part of eigenvalues and eigenvectors: JENNFS

   (2) Band matrix

      (a) To obtain all eigenvalues: RHQRVS

      (b) To obtain a limited part of eigenvalues and eigenvectors: JENNBS

3. General problems of symmetric matrices

   (1) Dense matrix

      (a) To obtain all eigenvalues and eigenvectors: GHQRVS

      (b) To obtain all eigenvalues and part of eigenvectors: GHQRIS

      (c) To obtain part of eigenvalues and eigenvectors: GHBSVS

   (2) Band matrix

      (a) To obtain a limited part of eigenvalues and eigenvectors: GJENBS

4. Singular value decomposition: SVDS

CGHBSS/D/Q (Eigenvalue analysis of the type Ax=λBx by Householder-bisection Method (Hermitian matrices))

Eigenvalue Analysis of the Type $Ax=\lambda Bx$ by Houscholder-Bisection Method (Hermitian Matrices)

Programmed by	Ichizo Ninomiya; December 1983
Format	Subroutine language; FORTRAN. Size; 54, 55, and 55 lines respectively

(1) Outline

When Hermitian symmetric matrix A and Hermitian symmetry positive definite matrix B are given, CGHBSS/D/Q obtains a specified number of eigenvalues and eigenvectors of an eigenvalue problem $Ax=\lambda Bx$ by Householder bisection method. CGHBSS/D/Q is a single (double, quadruple) precision subroutine.

(2) Directions

CALL CGHBSS/D/Q(A, B, KA, N, E, NE, V, NV, EPS, W, Z, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input	The upper right half of the Hermitian symmetric matrix including the diagonal is input. After processing of this routine, $\tilde{A}$ is generated (see the calculation method). The lower left half is preserved.
B	Complex type Two-dimensional array	Input	The upper right half of the Hermitian symmetric positive definite matrix including the diagonal is input. After processing by this routine, Cholesky decomposition component U is generated (see the calculation method). The lower left half is preserved.

Argument	Type and kind (*1)	Attribute	Content
KA	Integer type	Input	Adjustable dimension of A, B, and V (value of the first subscript in array declaration) KA≧N
N	Integer type	Input	Order of A and BThis is also the number of rows of V. N≧2
E	Real type Two-dimensional array	Output	Eigenvalues are generated and arranged. In descending order if NE > 0, and in ascending order if NE < 0
NE	Integer type	Input	The number of the eigenvalue to be obtained is specified by the absolute value. The largest (smallest) |NE| eigenvalues are obtained if NE > 0 (NE < 0). NE≠0
V	Complex type Two-dimensional array	Output	The eigenvector corresponding to eigenvalue E (I) is normalized in the meaning of $x^*Bx=1$ and output in the column I.
NV	Integer type	Input	The number of eigenvectors to be obtained is specified by the absolute value. The eigenvectors corresponding to the first |NV| eigenvalues in the order determined by NE are obtained. 0≦|NV|≦|NE|
EPS	Real type	Input	Convergence criterion constant of bisection methodWhen the tridiagonal matrix generated from $\tilde{A}$ is denoted by T, $\|T\| \cdot \|EPS\|$ is used for convergence test. Cholesky decomposition for B is omitted when EPS < 0. EPS≠0
W	Real type one-dimensional array	Work area	One-dimensional array with the size of 3N or more

Argument	Type and kind (*1)	Attribute	Content
Z	Complex type One-dimensional array	Work area	One-dimensional array with the size of 5N or more
ILL	Integer type	Output	ILL = 0: Normal end ILL = 1: B is decided to be non-positive definite. ILL = 30000: The input arguments violated the limit.

*1 For double or quadruple precision subroutines, all single precision types are changed to double or quadruple precision types.

(3) Calculation method

Positive definite matrix B is Cholesky-decomposed by an upper triangular matrix U as $B=U^*U$. When $\tilde{A}=(U^*)^{-1}AU^{-1}$ is formed from A using U, the generalized eigenvalue problem $Ax=\lambda Bx$ becomes a standard eigenvalue problem $\tilde{A}\tilde{x}=\lambda\tilde{x}$. This problem is solved by Householder-bisection method and eigenvector x is determined by $x=U^{-1}\tilde{x}$.

(4) Notes

1. When all eigenvalues are to be determined, it is more advantageous to use subroutine CGHQRS/D/Q or CGHQIS/D/Q which uses Householder-QR method.

2. For repeated calculation with B fixed and with only A changed from time to time, it is better to reuse the Cholesky-decomposed components of B. Refer to the description of EPS in the argument table.

Bibliography

1) Yoshitaka Beppu and Ichizo Ninomiya; "Comparisons of Matrix Solutions for Standard Eigenvalue Problems," Nagoya University Computer Center News, Vol. 11, No. 3, and pp. 265-274 (1980)

(1987. 08. 07)

CGHQIS/D/Q (Eigenvalue Analysis of the Type Ax=λBx by Householder-QR-Inverse Iteration Method (Hermitian Matrices))

Eigenvalue Analysis of the Type $Ax=\lambda Bx$ by Householder-QR-Inverse Iteration Method (Hermitian Matrices)

Programmed by	Ichizo Ninomiya, December 1983
Format	Subroutine language: FORTRAN; size: 52, 53, and 53 lines respectively.

(1) Outline

CGHQIS/D/Q obtains all of the eigenvalues and a part of the eigenvectors of the eigenvalue problem $Ax=\lambda Bx$ using Householder-QR-inverse iteration method if a Hermitian matrix A and a Hermitian positive definite matrix B are given. It is for single (double or quadruple) precision.

(2) Directions

CALL CGHQIS/D/Q(A, B, KA, N, E, V, NV, EPS, W, Z, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input/output	The upper right half containing the diagonal lines of a Hermitian matrix is input. It is processed and converted to $\tilde{A}$ in this routine. The lower left half is retained.

Argument	Type and kind (*1)	Attribute	Content
B	Complex type Two-dimensional array	Input	The upper right half containing the diagonal of a Hermitian positive definite matrix is input. It is processed and converted to the Cholesky decomposition element U (see the calculation method). The lower left half is retained.
KA	Integer type	Input	Adjustable dimensions of A, B, and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A and B. It also represents the number of rows of V. N≧2
E	Real type One-dimensional array	Output	Eigenvalues are output in the order of size. If NV≧0, eigenvalues are arranged in descending order. If NV<0, eigenvalues are arranged in ascending order.
V	Complex type Two-dimensional array	Output	Eigenvectors corresponding to the eigenvalue E(I) are output to the I-th column. They are normalized in the sense of $x^*Bx=1$.
NV	Integer type	Input	|NV| represents the number of eigenvectors to be obtained. If NV>0 (NV<0), eigenvectors are numbered in algebraically descending (ascending) order from the maximum (minimum). |NV| ≦N
EPS	Real type	Input	Convergence criterion of QR method. If the tridiagonalized matrix is denoted by T, $\|T\| \cdot \|EPS\|$ is used for the criterion. If EPS<0, the Cholesky decomposition of B is omitted. EPS≠0

Argument	Type and kind (*1)	Attribute	Content
W	Real type one-dimensional array	Work area	One-dimensional array of size N.
Z	Complex type One-dimensional array	Work area	One-dimensional array of size 5N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=1: B is decided to be not positive definite. ILL=30000: The input argument exceeded the limit.

1* For double (quadruple) precision subroutines, all single precision types are cahnged to double (quadruple) precision types.

(3) Calculation method

The Hermitian positive definite matrix B is Cholesky-decomposed to $B=U^*U$ with the upper triangular matrix U. If $\tilde{A}=(U^*)^{-1}AU^{-1}$ is generated from A using this U, the generalized eigenvalue problem $Ax=\lambda Bx$ becomes the standard eigenvalue problem $\tilde{A}\tilde{x}=\lambda\tilde{x}$. If this problem is solved using Householder-QR-Inverse iteration method, the eigenvector x is obtained with $x=U^{-1}\tilde{x}$.

(4) Notes

1. When up to about one-fourth of the entire eigenvalues is to be obtained, it is more advantageous to use the subroutine CGHBSS based on Householder bisection method.

2. When the calculation is to be repeated with B kept constant and only A changed, it is better to reuse the Cholesky decomposition elements of B. See the explanation for EPS in the list of arguments.

Bibliography

1) Yoshitaka Beppu and Ichizo Ninomiya: "Comparison of Matrix Solutions of Standard Eigenvalue Problems," Nagoya University Computer Center News, Vol. 11, No. 2, pp. 265-274 (1980)

(1987. 06. 22)

CGHQRS/D/Q (Eigenvalue Analysis of $Ax=\lambda Bx$ by Householder-QR Method (Hermitian Matrices))

Eigenvalue Analysis $Ax=\lambda Bx$ by Householder-QR Method (Hermitian Matrices)

Programmed by	Ichizo Ninomiya, December 1983
Format	Subroutine language: FORTRAN; size: 53, 54, and 54 lines respectively

(1) Outline

CGHQRS/D/Q obtains the entire eigenvalues and, if required, the entire eigenvectors of the eigenvalue problem $Ax=\lambda Bx$ if a Hermitian matrix A and a Hermitian positive definite matrix B are given. It converts A to $\tilde{A}=(U^*)^{-1}AU^{-1}$ by executing Cholesky decomposition with $B=U^*U$, and solves the standard eigenvalue problem $\tilde{A}y=\lambda y$ using Householder-QR method. If eigenvectors are required, it converts the eigenvector y of $\tilde{A}$ by $x=U^{-1}y$.

(2) Directions

CALL CGHQRS/D/Q(A, B, KK, N, E, F, EPS, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input/output	Only the upper right half of a Hermitian matrix is input. It is processed in this routine, and $\tilde{A}$ is generated in the upper right half. If eigenvectors are obtained, they are entered in each column. The vectors are normalized in the sense of $x^*Bx=1$.
B	Complex type Two-dimensional array	Input/output	Only the upper right half of a Hermitian positive definite matrix is input. It is processed in this routine, and the Cholesky decomposition element U of B is entered in the upper right half. The lower left half is retained.
KK	Integer type	Input	Value of the first subscript in the declaration of arrays A and B: KK≧N

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Order of arrays A and B. N≧2
B	Real type One-dimensional array	Input	One-dimensional array containing N elements. Eigenvalues are arranged in algebraically descending order.
F	Complex type One-dimensional array	Work area	One-dimensional array containing N elements.
EPS	Real type	Input	|EPS| is the convergence criterion of the QR method. It is also the positivity criterion for Cholesky decomposition of B. If this routine is called with EPS<0, the Cholesky decomposition elements of B are reused. EPS≠0
IND	Integer type	Input/output	This argument has has the following meaning as an input argument. IND=0: Only eigenvalues are calculated. IND≠0: Eigenvectors are also calculated. This argument has the following meaning as an output argument. IND=0: Calculation is normally executed. IND=1: B is decided to be not positive definite. IND=30000: Limits on the input argument were exceeded. Because this argument is both input and output, constants must not be used.

1* For double (quadruple) precision subroutines, all single precision types are changed to double (quadruple) precision types.

(3) Calculation method

The Hermitian positive definite matrix B is Cholesky-decomposed to $B=U^*U$ with the upper triangular matrix U. If $\tilde{A}=(U^*)^{-1}AU^{-1}$ is generated from A using this U, the generalized eigenvalue problem $Ax=\lambda Bx$ becomes the standard eigenvalue problem $\tilde{A}\tilde{x}=\lambda\tilde{x}$. This problem is solved using Householder QR method, and the eigenvector x is obtained with $x=U^{-1}\tilde{x}$.

(4) Notes

1. If only selected eigenvectors are to be obtained, it may often be advantageous to use Householder-QR-inverse iteration method (CGHQIS).

2. If calculation is iterated with B kept constant and only A changed, it is better to reuse the Cholesky decomposition elements of B. See the explanation for the argument EPS.

(1987. 06. 22) (1987. 08. 07)

CGKLZS/D/Q (Eigenvalue Analysis of the Type Ax=λBx by LZ Method (Complex Matrices))

Eigenvalue Analysis of the Type $Ax=\lambda Bx$ by LZ Method(Complex Matrices)

Programmed by	Ichizo Ninomiya, July 1984
Format	Subroutine language: FORTRAN; size: 256 and 256 lines respectively

(1) Outline

CGKLZS/D/Q obtains all the eignevalues of the eigenvalue problem $Ax=\lambda Bx$ using the LZ method for given complex matrices A and B, and obtains specified eigenvectors using the inverse iteration. It is for single (double or quadruple) precision.

(2) Directions

CALL CGKLZS/D/Q(A, B, KA, N, E, IE, V, NV, EPS, W, Z, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input	Complex matrix A. It is processed with this routine, and transformed to $\tilde{A}$ (see the calculation method).
B	Complex type Two-dimensional array	Input	Complex matrix B. It is processed with this routine, and transformed to $\tilde{B}$ (see the calculation method).
KA	Integer type	Input	Adjustable dimensions of A, B, and V (value of the first subscript in the array declaration). KA≧N

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Order of A and B. It also represents the number of rows of V. N≧2
E	Complex type One-dimensional array	Output	Eigenvalues are output in the order of absolute values. If NV≧0, eigenvalues are arranged in descending order. If NV<0, eigenvalues are arranged in ascending order.
IE	Integer type one-dimensional array	Output	The condition code of the I-th eigenvalue is input in IE(I). IE=0: Normal. IE=1: Eigenvalues do not exist. IE=2: Eigenvalues are indeterminate.
V	Complex type Two-dimensional array	Output	An eigenvector to the eigenvalue E(I) is normalized to a length of 1 and placed to the I-th column.
NV	Integer type	Input	The number of eigenvectors to be obtained is represented by the absolute value, and how to arrange eigenvalues is represented by the sign. (See the item of E.) $0 \leqq \lvert NV \rvert \leqq \lvert N \rvert$
EPS	Real type	Input	Convergence criterion of bisection method. $\max(\lVert A \rVert, \lVert B \rVert) * \lvert EPS \rvert$ is used as the criterion.
W	Complex type one-dimensional array	Work area	One-dimensional array of size N*N.

Argument	Type and kind (*1)	Attribute	Content
Z	Complex type One-dimensional array	Work area	One-dimensional array of size N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=K: K:Number of abnormal eigenvalues. ILL=20000: LZ method does not result in convergence. ILL=30000: The input argument exceeded the limit.

1* For double (quadruple) precision subroutines, all single precision types are changed to double (quadruple) precision types.

(3) Calculation method

1. The matrix A and B are transformed to upper Hessenberg matrix $\tilde{A}=LAM$ and $\tilde{B}=LBM$ respectively using the stabilized elementary row transformation L, and the stabilized elementary column transformation M.

2. All the eigenvalues of the eigenvalue problem $\tilde{A}y=\lambda\tilde{B}y$ are obtained using the LZ method with origin shift, and the specified eigenvectors y are obtained using the inverse iteration.

3. Eigenvectors are obtained by $x=My$, and normalized to length 1.

(4) Note

1. If A is Hermitian, and B is Hermitian positive definite, it is more advantageous to use CGHBSS, CGHQIS, and CGHQRS.

Bibliography

1) Kaufman L; "The LZ Algorithms to Solve the Generalized Eigenvalue Problem", Stanford Computer Science Report PB-222099, p. 103 (1973)

(1987. 08. 07) (1988. 04. 22)

CHEQIS/D/Q (Eigenvalue Analysis for Complex Matrices by QR and Inverse Iteration Method)

Eigenvalue Analysis for Complex Matrices by QR and Inverse Iteration Method

Programmed by	Ichizo Ninomiya, October 1983
Format	Subroutine language: FORTRAN; size: 207 and 208 lines respectively

(1) Outline

CHEQIS/D/Q transformed a complex matrix to an upper Hessenberg matrix using the stabilized elementary transformation, obtains all the eigenvalues using the QR method, and calculates the eigenvectors as many as requested using the inverse iteration method. It is a single (double or quadruple) precision subroutine.

(2) Directions

CALL CHEQIS/D/Q(A, KA, N, E, V, NV, EPS, IW, W, Z, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input	Matrix whose eignevalue analysis is to be executed. It is processed with this routine, and transformed to an upper Hessenberg type.
KA	Integer type	Input	Adjustable dimensions of A and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A. Number of rows of V. It also represents the size of E. N≧1

Argument	Type and kind (*1)	Attribute	Content
E	Complex type One-dimensional array	Output	Eigenvalues. The I-th eigenvalue is E(I).
V	Complex type Two-dimensional array	Output	The I-th eigenvector is output to the I-th column of V. The length is normalized to 1.
NV	Integer type	Input	The number of eigenvectors is represented by the absolute value. If NV≧0, eigenvalues are arranged in the descending order of absolute values. If NV<0, eigenvalues are arranged in the ascending order of absolute values. Then, vectors to the first |NV| eigenvalues are obtained.
EPS	Real type	Input	$\|\|A\|\|$ ·EPS is used as the convergence criterion of QR. EPS>0
IW	Integer type one-dimensional array	Work area	One-dimensional array of size N.
W	Real type one-dimensional array	Work area	One-dimensional array of size 3N.

Argument	Type and kind (*1)	Attribute	Content
Z	Complex type One-dimensional array	Work area	One-dimensional array of size N^2.
ILL	Integer type	Output	Condition code. IND=0: Normal. IND=1: N=1 or the elements in A are all 0. IND=2: The QR method or the inverse iteration does not result in convergence. IND=30000: The input argument exceeded the limit.

*1 For double precision subroutines, all real types are changed to double precision real types, and all complex types to double precision complex types. For quadruple precision subroutines, all real types are changed to quadruple precision real types, and all complex types to quadruple precision complex types.

(3) Calculation method

The complex matrix A is transformed to an upper Hessenberg matrix $H=S^{-1}AS$ using the stabilized elementary transformation S, that is, Gauss's elimination accompanied by row exchange.

All the eigenvalues of H are obtained using the QR method with origin shift.

A specified number of eigenvectors of H are obtained using the inverse iteration. These eigenvectors are placed in U Eigenvectors of A are calculated as $V=SU$ from U.

(4) Notes

1. It is reasonable to process the Hermitian matrix with the special-purpose routine.

2. If an eigenvector is not to be obtained (NV=0), the area to V and Z is not used, and thus need not be prepared. Anything can be written for these arguments.

(1987. 06. 19)

CHEQRS/D/Q (Eigenvalue Analysis for Complex Matrices by QR Method)

Eigenvalue Analysis for Complex Matrices by QR Method

Programmed by	Ichizo Ninomiya, October 1983
Format	Subroutine language: FORTRAN; size: 191 and 192 lines respectively

(1) Outline

CHEQIS/D/Q transforms a complex matrix to an upper Hessenberg matrix using the stabilized elementary transformation, obtains all the eigenvalues using the QR method, and, if required, calculates all the corresponding eigenvectors. This subroutine is for single (double or quadruple) precision.

(2) Directions

CALL CHEQRS/D/Q(A, KA, N, E, V, EPS, IW, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input	Matrix whose eigenvalue analysis is to be executed. It is processed with this routine, and transformed to a Hessenberg type.
KA	Integer type	Input	Adjustable dimensions of A and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A. Number of rows of V. It also represents the size of E. N≧1

Argument	Type and kind (*1)	Attribute	Content
E	Complex type One-dimensional array	Output	Eigenvalue. The I-th eigenvalue is E(I).
V	Complex type Two-dimensional array	Output	The I-th eigenvector is output to the I-th column of V. The length is normalized to 1.
EPS	Real type	Input	‖A‖·EPS is used as the convergence criterion of QR. EPS>0
IW	Integer type one-dimensional array	Work area	One-dimensional array of size N.
IND	Integer type	Input/output	Input: Whether to calculate eigenvectors and how to array eigenvalues are specified. IND=0: Eigenvectors are not calculated. IND≠0: Eigenvectors are calculated.

Argument	Type and kind (*1)	Attribute	Content
			IND≧0: Eigenvalues are arranged in the descending order of absolute values. IND<0: Eigenvalues are arranged in the ascending order of absolute values. Output: Condition code. IND=0: Normal. IND=1: N=1 or the elements of A is all 0. IND=2: The QR method does not result in convergence. IND=30000: The input argument exceeded the limit.

1* For double (quadruple) precision subroutines, all single precision types are changed to double (quadruple) precision types.

(3) Calculation method

The real matrix A is transformed to an upper Hessenberg matrix $H=S^{-1}AS$ using stabilized elementary transformation S, that is, Gauss' elimination accompanied by row exchange. H is converted into an upper triangular matrix using QR method with origin shift. Eigenvalues are given as the diagonal elements. Eigenvectors are obtained from the eigenvectors of the upper triangular matrix using the inverse transformation of the stabilized elementary and QR transformations.

(4) Notes

1. It is reasonable to process the Hermitian matrix with the special-purpose routine.

2. If eigenvector are not to be obtained (IND=0), the area for V is not used, and thus need not be prepared. Anything can be written for it.

(1987.06.19) (1987.08.07)

CHOBSS/D/Q (Eigenvalue Analysis for Hermitian Matrix by Householder-Bisection Method)

Eigenvalue Analysis for Hermitian Matrices by Householder-Bisection Method

Programmed by	Ichizo Ninomiya, October 1983
Format	Subroutine language: FORTRAN; size: 188 and 189 lines respectively

(1) Outline

CHOBSS/D/Q tridiagonalizes a Hermitian matrix using Householder's reflexion transformation, obtains the eigenvalues of the tridiagonalized matrix using the bisection method based on Sturm sequence, and calculates the eigenvectors using the inverse iteration. It is for single (double or quadruple) precision.

(2) Directions

CALL CHOBSS/D/Q(A, KA, N, E, NE, V, NV, EPS, W, Z, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input	The upper right half containing the diagonal of a Hermitian matrix is input. It is processed with this routine. The left lower half is retained.
KA	Integer type	Input	Adjustable dimensions of A and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A. It also represents the number of rows of V. N≧1

Argument	Type and kind (*1)	Attribute	Content
E	Real type One-dimensional array	Output	Eigenvalues are output in the order of size. If NE>0, eigenvalues are arranged in descending order. If NE<0, eigenvalues are arranged in ascending order.
NE	Integer type	Input	The number of eigenvalues to be obtained is represented by the absolute value. If NE>0 (NE<0), eigenvalues are numbered in algebraically descending (ascending) order from the maximum (minimum). NE≠0
V	Complex type Two-dimensional array	Output	Eigenvectors to the eigenvalue E(I) are normalized to length 1, and placed to the I-th column.
NV	Integer type	Input	The number of eigenvectors to be obtained is represented by the absolute value. Eigenvalues are numbered from the end in the order defined by NE. $0 \leq \|NV\| \leq \|NE\|$
EPS	Real type	Input	Convergence criterion of bisection method. If the tridiagonalized matrix is denoted by T, $\|\|T\|\| \cdot EPS$ is used as the criterion. EPS>0
W	Real type One-dimensional array	Work area	One-dimensional array of size 3N.
Z	Complex type One-dimensional array	Work area	One-dimensional array of size 5N.

Argument	Type and kind (*1)	Attribute	Content
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: The input argument exceeded the limit.

1* For double precision subroutines, all real types are changed to double precision real types, and all complex types to double precision complex types. For quadruple precision subroutines, all real types are changed to quadruple precision real types, and all complex types to quadruple precision complex types.

(3) Calculation method

The matrix A is transformed to a tridiagonal matrix $T=H^*AH$ using the Householder transformation H.

The eigenvalues of T are obtained by the bisection method based on Sturm sequence. They are numbered as many as specified from the end in a specified order. The eigenvectors corresponding to the eigenvalues specified as counted from the end are obtained using the inverse iteration. The matrix containing these eigenvectors in columns is denoted by U, then the eigenvector V of A can be obtained by $V=HU$.

(4) Note

When all the eigenvalues of a Hermitian matrix are to be obtained, it is better to use the routine CHOQRS/D based on the QR method than this routine. When all of the eigenvalues and all or part of the corresponding eigenvectors are to be obtained, it is more reasonable to use the routine CHQRIS/D based on the QR-inverse iteration.

(1987. 06. 22)

CHOQRS/D/Q (Eigenvalue Analysis for Hermitian Matrices by Householder-QR Method)

Eigenvalue Analysis for Hermitian Matrices by Householder-QR Method

Programmed by	Ichizo Ninomiya, October 1983
Format	Subroutine language: FORTRAN; size: 133, 134, and 134 lines respectively

(1) Outline

CHOQRS/D/Q obtains all the eigenvalues and, if required, all the corresponding eigenvectors of an Hermitian matrix, using the Householder's tridiagonalization and QR method with origin shift.

(2) Directions

CALL CHOQRS/D/Q(A, KA, N, E, F, EPS, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input/output	The upper right half containing the diagonal lines of a Hermitian matrix is input. Anything can be input in the lower left half. If eigenvectors are to be obtained, eigenvectors are output in A. That is, eigenvectors to the eigenvalues E(I) are normalized to length 1, and is placed in the I-th column of A.
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Order of A. N≧2

Argument	Type and kind (*1)	Attribute	Content
E	Real type One-dimensional array	Output	One-dimensional array containing N elements. Eigenvalues are arranged in algebraically descending order.
F	Complex type One-dimensional array	Work area	One-dimensional array containing N elements.
EPS	Real type	Input	Convergence criterion for QR method. If all the non-diagonal elements become smaller than $\|\|A\|\| \cdot EPS$ in magnitude, convergence is judged to have occurred. EPS>0
ILL	Integer type	Input/output	If ILL=0 is given, only eigenvalues are calculated. If ILL≠0 is given, both eigenvalues and eigenvectors are calculated. If calculation terminates normally, 0 is output. If limits on the input argument are exceeded, 3000 is output. Constants must not be used for the actual argument.

*1 For double precision subroutines, all real types are changed to double precision real types, and all complex types are changed to double precision complex types. For quadruple precision subroutines, all real types are changed to quadruple precision real types, and all complex types are changed to quadruple precision complex types.

(3) Performance

As with real symmetric matrices, this routine is high in speed. It can be used without troubles even for the case of multiple or close eigenvalues.

(4) Notes

1. This routine is optimum when all eigenvalues (and corresponding eigenvectors) are to be obtained with a small storage requirement.

2. If only part of eigenvalues or eigenvectors is to be obtained, Householder-Givens' method (bisection method) is desirable. Subroutines that are currently registered are CHOBSS/D.

(1987.06.22) (1987.08.07)

CHQRIS/D/Q (Eigenvalue Analysis of Hermitian Matrices by Householder-QR-Inverse Iteration Method)

Eigenvalue Analysis of Hermitian Matrices by Householder-QR-Inverse Iteration Method

Programmed by	Ichizo Ninomiya, October 1983
Format	Subroutine language: FORTRAN; size: 188 and 189 lines respectively

(1) Outline

CHQRIS/D/Q obtains all the eigenvalues of a Hermitian matrix using Householder-QR method, and calculates specified eigenvectors using the inverse iteration.

(2) Directions

CALL CHQRIS/D/Q(A, KA, N, E, V, NV, EPS, W, Z, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type Two-dimensional array	Input	The upper right half containing the diagonal of a Hermitian matrix is input. It is processed with this routine. The lower left half is retained.
KA	Integer type	Input	Adjustable dimensions of A and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A. It also represents the number of rows of V. N≧2
E	Real type One-dimensional array	Output	All eigenvalues are output in the order of size. If NV≧0, eigenvalues are arranged in descending order. If NV<0, eigenvalues are arranged in ascending order.

Argument	Type and kind (*1)	Attribute	Content
V	Complex type Two-dimensional array	Output	Eigenvectors to the eigenvalue E(I) are normalized to length 1, and placed to the I-th column.
NV	Integer type	Input	|NV| represents the number of eigenvectors to be obtained. If NV>0 (NV<0), eigenvectors are numbered in algebraically descending (ascending) order from the maximum (minimum). $\|NV\| \leqq N$
EPS	Real type	Input	Convergence criterion of QR method. If the tridiagonalized matrix is denoted by T, $\|\|T\|\| \cdot EPS$ is used as the criterion. EPS>0
W	Real type One-dimensional array	Work area	One-dimensional array of size N.
Z	Complex type One-dimensional array	Work area	One-dimensional array of size 5N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: The input argument exceeded the limit.

*1 For double precision subroutines, all real types are changed to be double precision real types, and all complex types to double precision complex types. For quadruple precision subroutines, all real types are changed to quadruple precision real types, and all complex types to quadruple precision complex types.

(3) Calculation method

The Hermitian matrix A is transformed to a tridiagonal matrix $T=H^*AH$ using the Householder transformation H.

All the eigenvalues of T are calculated using the QR method. A specified number of eigenvectors of T are obtained using the inverse iteration. They are gathered in the matrix U.

The eigenvectors of A are calculated by $V=HU$.

(4) Notes

1. The routine is adequate when all the eigenvalues are obtained quickly, and all or part of the eigenvectors are obtained.

2. If up to about one-fourth of the eigenvalues is to be obtained, it is more advantageous to use CHOBSS/D based on Householder-bisection method.

(1987. 06. 22)

GHBSVS/D (Eigenvalue analysis of the type Ax=λBx by Householder-bisection method)

Eigenvalue Analysis of the Type $Ax=\lambda Bx$ by Householder-Bisection Method

Programmed by	Ichizo Ninomiya; April 1981
Format	Subroutine language; FORTRAN Size; 230 lines each

(1) Outline

When a real symmetric matrix A and a real symmetric positive definite matrix B are given, GHBSVS/D determines the specified number of eigenvalues and eigenvectors of eigenvalue problem $Ax=\lambda Bx$ by using the Householder-bisection method. GHBSVS(D) is for single (double) precision.

(2) Directions

CALL GHBSVS/D(A, B, KK, N, E, V, NV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	The upper right half of the real symmetric matrix including the diagonal is input. After processing by this routine, $\tilde{A}$ is generated (see the calculation method). The lower left half is preserved.
B	Real type Two-dimensional array	Input	The upper right upper of the real symmetry positive definite matrix including the diagonal is input. The matrix is processed by this routine to become Cholesky decomposition component U (see the calculation method). The lower left half is preserved.
KK	Integer type	Input	Adjustable dimensions of A, B, and V (value of the first subscript in array declaration) KK≧N

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Order of A and BThis is also the number of rows of V. N≧2
E	Real type Two-dimensional array	Output	Eigenvalues are generated and arranged. In descending order if NE > 0, and in ascending order if NE < 0
NE	Integer type	Input	The number of eigenvalues to be obtained is specified by the absolute value. The largest (smallest) |NE| eigenvalues are obtained if NE > 0 (NE < 0). NE≠0
V	Real type Two-dimensional array	Output	The eigenvector corresponding to eigenvalue E(I) is normalized in the meaning of $x^TBx=1$ and output in the column I.
NV	Integer type	Input	The number of eigenvectors to be determined is specified by the absolute values. The eigenvectors corresponding to the first |NV| eigenvalues in the order determined by NE are obtained. 0≦|NV|≦|NE|
EPS	Real type	Input	Convergence criterion constant for bisection methodWhen the tridiagonal matrix generated from $\tilde{A}$ is denoted by T, $\|T\|\cdot\|EPS\|$ is used for convergence test. Cholesky decomposition for B is omitted if EPS < 0. EPS≠0
W	Real type one-dimensional array	Work area	One-dimensional array with the size of 6N or more
ILL	Integer type	Output	ILL = 0: Normal end ILL = 1: B is decided to be non-positive definite. ILL = 30000: The input arguments violated the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

Positive definite matrix B is Cholesky-decomposed by an upper triangular matrix U as $B=U^TU$. When $\tilde{A}=U^{-T}AU^{-1}$ is from A using U, the generalized eigenvalue problem $Ax=\lambda Bx$ becomes a standard eigenvalue problem $\tilde{A}\tilde{x}=\lambda\tilde{x}$. This problem is solved by Householder-bisection method and eigenvector x is determined by $x=U^{-1}\tilde{x}$.

(4) Notes

1. When all eigenvalues are to be determined, it is more advantageous to use subroutine CGHQRS/D/Q or CGHQIS/D/Q which uses Householder-QR method.

2. For repeated calculation with B fixed and with only A changed from time to time, it is better to reuse the Cholesky-decomposed components of B. Refer to the description of EPS in the argument table.

Bibliography

1) Yoshitaka Beppu and Ichizo Ninomiya; "Comparisons of Matrix Solutions for Standard Eigenvalue Problems," Nagoya University Computer Center News, Vol.11, No.3, and pp.265-274 (1980)

(1987.08.10) (1988.04.04)

GHBSVV/W (Eigenvalue Analysis of the Type Ax=λBx by Householder-Bisection Method: Vector Version)

Eigenvalue Analysis of the Type $Ax=\lambda Bx$ by Householder-Bisection Method : Vector Version

Programmed by	Ichizo Ninomiya, March 1988
Format	Subroutine Language: FORTRAN; Size: 156 lines

(1) Outline

GHBSVV/W obtains the specified number of eigenvalues and corresponding eigenvectors of the eigenvalue problem $Ax=\lambda Bx$ by the Householder-Bisection method when a real symmetric matrix A and a real symmetric positive definite matrix B are given. GHBSVV(W) is for single (double) precision.

(2) Directions

CALL GHBSVV/W(A, B, KK, N, E, V, NV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	The upper right half containing the diagonal of a real symmetric matrix is input. This routine turns it into $\tilde{A}$ (see "Calculation method"). The lower left half should be used as a work area.
B	Real type Two-dimensional array	Input	The upper right half containing the diagonal of a real symmetric positive definite matrix is input. This routine decomposes it into Cholesky component U (see "Calculation method"). The lower left half is retained.
KK	Integer type	Input	Adjustable dimensions of A, B, and V (value of the first subscript in the array declaration). KK≧N

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Order of A and B or the number of rows of V. $N \geqq 2$
E	Real type Two-dimensional array	Output	Eigenvalues are output in the order of size. If NE>0, they are arranged in decreasing order. If NE<0, they are arranged in increasing order.
NE	Integer type	Input	Represents the number of eigenvalues to be obtained by the absolute value. If NE>0(NE<0), they are numbered from the maximum (minimum) in algebraically decreasing (increasing) order. $NE \neq 0$
V	Real type Two-dimensional array	Output	Eigenvectors to eigenvalues E(I) are normalized and placed to the I-th column in the sense of $x^TBx=1$.
NV	Integer type	Input	Represents the number of eigenvectors to be obtained by the absolute value. Eigenvalues are numbered from the end in the order defined by NE. $0 \leqq \lvert NV \rvert \leqq \lvert NE \rvert$
EPS	Real type	Input	Convergence criterion of bisection method. If the tridiagonalized matrix is denoted by T, $\lVert T \rVert \cdot \lvert EPS \rvert$ is used as the criterion. If EPS<0, the Cholesky decomposition of B is omitted. $EPS \neq 0$
W	Real type one-dimensional array	Work area	One-dimensional array of size 6N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=1: B is decided to be not a positive definite. ILL=30000: Input argument exceeded the limit.

*1 For double precision subroutines, all real types should be changed to double precision real types.

(3) Calculation method

This routine decomposes the symmetric positive definite matrix B into $B=U^TU$ with an upper triangular matrix U by Cholesky decomposition method. If $\tilde{A}=U^{-T}AU^{-1}$ is generated from A by using this U, the generalized eigenvalue problem $Ax=\lambda Bx$ turns into the standard eigenvalue problem $\tilde{A}\tilde{x}=\lambda\tilde{x}$. By solving this problem by the Householder-bisection method, the eigenvector x is obtained by $x=U^{-1}\tilde{x}$.

(4) Note

1. When all eigenvalues are to be obtained, it is more advantageous to use the subroutine GHQRVV/W or GHQRIV/W based on the Householder-QR method.

2. When calculation is to be repeated with only B kept constant and A changed, it is more advantageous to reuse the Cholesky decomposition elements of B. See the explanation for "EPS" in the argument list.

Bibliography

1) Yoshitaka Beppu and Ichizo Ninomiya; "Comparison of Matrix Methods for Standard Eigenvalue Problems", Nagoya University Computer Center News, Vol. 11, No. 3, pp. 265-274(1980).

(1987. 08. 10) (1988. 04. 08)

GHQRIS/D (Eigenvalue Analysis of the Type Ax=λBx by Householder-QR-Inverse Method)

Eigenvalue Analysis of the Type $Ax=\lambda Bx$ by Householder-QR-Inverse Iteration Method

Programmed by	Ichizo Ninomiya, April 1981
Format	Subroutine language: FORTRAN; size: 250 lines

(1) Outline

GHQRIS/D obtains all of the eigenvalues and a part of the corresponding eigenvectors of the eigenvalue problem $Ax=\lambda Bx$ using the Householder-QR-Inverse iteration method when a real symmetric matrix A and a real symmetric positive definite matrix B are given. It is for single (double) precision.

(2) Directions

CALL GHQRIS/D(A, B, KK, N, E, V, NV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Only the upper right half containing the diagonal lines of a real symmetric matrix is input. It is processed with this routine, and converted to A (see "Calculation method"). The lower left half is retained.
B	Real type Two-dimensional array	Input	Only the upper right half containing the diagonal lines of a real symmetric positive definite matrix need be input. It is processed with this routine, and converted to the Cholesky decomposition element U (see "Calculation method"). The left lower half is retained.
KK	Integer type	Input	Adjustable dimensions of A, B, and V (value of the first subscript in the array declaration). KK≧N

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Order of A and B. It also represents the number of rows of V. $N \geqq 2$
E	Real type One-dimensional array	Output	Eigenvalues are output in the order of size. If $NV \geqq 0$, eigenvalues are arranged in descending order. If $NV < 0$, eigenvalues are arranged in ascending order.
V	Real type Two-dimensional array	Output	Eigenvectors corresponding to the eigenvalue E(I) are output to the I-th column. They have been normalized in the meaning of $x^T Bx = 1$.
NV	Integer type	Input	$\|NV\|$ represents the number of eigenvectors to be obtained. If $NV > 0$ ($NV < 0$), eigenvectors are counted in algebraically descending (ascending) order from the maximum (minimum). $\|NV\| \leqq N$
EPS	Real type	Input	Convergence criterion of QR method. If a tridiagonalized matrix is denoted by T, $\|\|T\|\| \cdot \|EPS\|$ is used as the criterion. IF $EPS < 0$, Cholesky decomposition of B is omitted. $EPS \neq 0$
W	Real type one-dimensional array	Work area	One-dimensional array of size 6N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=1: B is decided to be not definite positive. ILL=30000: The input argument exceeded the limit.

*1 For double precision subroutines, all real types should be changed to double precision real types.

(3) Calculation method

The symmetric positive definite matrix B is Cholesky-decomposed to $B=U^TU$ using the upper triangular matrix U. If $\tilde{A}=U^{-T}AU^{-1}$ is made from A by using this U, the generalized eigenvalue problem $Ax=\lambda Bx$ becomes the standard eigenvalue problem $\tilde{A}\tilde{x}=\lambda\tilde{x}$. By solving this problem using Householder-QR-inverse iteration method, the eigenvector x is obtained with $x=U^{-1}\tilde{x}$.

(4) Note

1. If up to about one-fourth of the entire eigenvalues is to be obtained, it is better to use the subroutine GHBSVS based on Householder bisection method.

2. If calculation is to be iterated with B kept constant and only A changed, it is desirable to reuse the Cholesky decomposition elements of B. See the explanation for EPS in the list of arguments.

Bibliography

1) Yoshitaka Beppu and Ichizo Ninomiya; "Comparison of Matrix Solutions of Standard Eigenvalue Problems," Nagoya University Computer Center News, Vol. 11, No. 3, pp. 265-274 (1980)

(1987. 08. 10) (1988. 04. 04)

GHQRIV/W (Eigenvalue Analysis of the Type Ax=λBx by Householder-QR-Inverse Iteration Method: Vector Version)

Eigenvalue Analysis of the Type $Ax=\lambda Bx$ by Householder-QR-Inverse Iteration Method : Vector Version

Programmed by	Ichizo Ninomiya, March 1988
Format	Subroutine Language: FORTRAN; Size: 153 lines

(1) Outline

GHQRIV/W obtains all of eigenvalues and part of the corresponding eigenvectors of the eigenvalue problem $Ax=\lambda Bx$ when a real symmetric matrix A and a real symmetric positive definite matrix B are given. GHQRIV(W) is for single (double) precision.

(2) Directions

CALL GHQRIV/W(A, B, KK, N, E, V, NV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	The upper right half containing the diagonal of a real symmetric matrix is input. This routine turns it into $\tilde{A}$ (see "Calculation Method"). The lower left half is used as a work area.
B	Real type Two-dimensional array	Input	The upper right half containing the diagonal of a real symmetric positive definite matrix is input. This routine turns it into the Cholesky decomposition component U (see "Calculation method"). The left lower half is retained.

Argument	Type and kind (*1)	Attribute	Content
KK	Integer type	Input	Adjustable dimensions of A, B, and V (value of the first subscript in declaration of array). KK≧N
N	Integer type	Input	Order of A and B or the number of rows of V. N≧2
E	Real type One-dimensional array	Output	Eigenvalues are output in the order of size. If NV≧0, they are arranged in decreasing order. If NV<0, they are arranged in increasing order.
V	Real type Two-dimensional array	Output	Eigenvectors to eigenvalues E(I) are output to the I-th column. They are normalized in the sense of $x^TBx=1$.
NV	Integer type	Input	|NV| represents the number of eigenvectors to be obtained. If NV>0(NV<0), they are numbered in algebraically decreasing (increasing) order from the maximum (minimum). |NV| ≦N
EPS	Real type	Input	Convergence criterion of QR method. If the tridiagonalized matrix is denoted by T, $\|\|T\|\| \cdot \|EPS\|$ is used as the criterion. If EPS<0, Cholesky decomposition of B is omitted. EPS≠0
W	Real type one-dimensional array	Work area	One-dimensional array of size 6N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=1: B is decided to be not a positive definite. ILL=30000: Input argument exceeded the limit.

*1 For double precision subroutines, all real types should be changed to double precision real types.

(3) Calculation method

This routine decomposes the symmetric positive definite matrix B into $B=U^TU$ with an upper triangular matrix U by Cholesky decomposition. If $\tilde{A}=U^{-T}AU^{-1}$ is generated from A using U, a generalized eigenvalue problem $Ax=\lambda Bx$ is handled as a standard eigenvalue problem $\tilde{A}\tilde{x}=\lambda\tilde{x}$. By solving this problem with the Householder-QR-iteration method, the eigenvector x is obtained with $x=U^{-1}\tilde{x}$.

(4) Note

1. When up to one-fourth of all eigenvalues are to be obtained, it is more advantageous to use the subroutine GHBSVS/W based on the Householder bisection method.

2. When calculation is to be repeated with only B kept constant and A changed, the Cholesky decomposition elements of B should be reused. See the explanation for "EPS" in the argument list.

Bibliography

1) Yoshitaka Beppu and Ichizo Ninomiya; "Comparison of Matrix Methods for Standard Eigenvalue Problems," Nagoya University Computer Center News, Vol. 11, No. 3, pp. 265-274 (1980)

(1987. 08. 10) (1988. 04. 08)

GHQRVS/D and GHQRUS/D (Eigenvalue Analysis Ax=λBx by Householder-QR Method)

Eigenvalue Analysis $Ax=\lambda Bx$ by Householder-QR Method

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 60, 60, 60, and 60 lines respectively

(1) Outline

GHQRVS/D and GHQRUS/D obtain the entire eigenvalues and, if required, the entire eigenvectors of the eigenvalue problem $Ax=\lambda By$ if a real symmetric matrix A and a real symmetric positive definite matrix B are given. It converts A to $\tilde{A}=U^{-T}AU^{-1}$ by Cholesky decomposition with $B=U^TU$, and solves the standard eigenvalue problem $\tilde{A}y=\lambda y$ using Householder QR method. If eigenvectors are required, the eigenvector y of $\tilde{A}$ is converted as $x=U^{-1}y$.

(2) Directions

CALL GHQRVS/D(A, B, KK, N, E, F, EPS, IND)

CALL GHQRUS/D(A, B, KK, N, E, F, EPS, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	The entire real symmetric matrix (not the upper right half) is input. It is processed in this routine, and $\tilde{A}$ is generated in the upper right half. If eigenvectors are obtained, they are input in each column. The vectors are normalized in the sense of $x^TBx=1$.
B	Real type Two-dimensional array	Input/output	Only the upper right half of a symmetric positive definite matrix is input. It is processed in this routine, and the upper right half contains the Cholesky decomposition element U of B. The lower left half is retained.

Argument	Type and kind (*1)	Attribute	Content
KK	Integer type	Input	Value of the first subscript in the declaration of arrays A and B. KK≧N
N	Integer type	Input	Order of arrays A and B. N≧2
E	Real type One-dimensional array	Output	One-dimensional array containing N elements. In GHQRVS/D, eigenvalues are arranged in algebraically descending order. In GHQRUS/D, they are arranged in descending order of the absolute value.
F	Real type One-dimensional array	Work area	One-dimensional array containing N elements.
EPS	Real type	Input	|EPS| is the convergence criterion of the QR method. It is also the positivity criterion for the Cholesky decomposition of B. If this routine is called with EPS<0, it reuses the Cholesky decomposition elements of B. EPS≠0
IND	Integer type	Input/output	This argument has the following meaning as an input argument. IND=0: Only eigenvalues are calculated. IND≠0: Eigenvectors are also calculated. This argument has the following meaning as an output argument. IND=0: Calculation is normally executed. IND=1: B is decided to be not a positive definite. IND=30000: Limits on the input argument are exceeded. Because this argument is both input and output, constants must not be used as an actual argument.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

The symmetric positive definite matrix B is Cholesky-decomposed to $B=U^TU$ using an upper triangular matrix U. If $\bar{A}=U^{-T}AU^{-1}$ is generated from A by using this U, the generalized eigenvalue problem $Ax=\lambda Bx$ becomes the standard eigenvalue problem $\bar{A}\tilde{x}=\lambda\tilde{x}$. By solving this problem using Householder-QR method, the eigenvector x is obtained with $x=U^{-1}\tilde{x}$.

(4) Notes

1. If only selected eigenvectors are obtained, Householder-QR-inverse iteration (GHQRIS/D) may be advantageous.

2. If calculation is iterated with B kept constant and only A changed, it is better to reuse the Cholesky decomposition elements of B. See the explanation for EPS in the list of arguments.

(1987. 08. 10) (1988. 04. 04)

GHQRVV/W (Eigenvalue Analysis Ax=λBx by Householder-QR Method: Vector Version)

Eigenvalue Analysis $Ax=\lambda Bx$ by Householder-QR Method: Vector Version

Programmed by	Ichizo Ninomiya, March 1988
Format	Subroutine language: FORTRAN; size: 155 lines

(1) Outline

GHQRVV/W obtains all of the eigenvalues and, as required, all of the corresponding eigenvectors of the eigenvalue problem $Ax=\lambda Bx$ when a real symmetric matrix A and a symmetric positive definite matrix B are given. It executes Cholesky decomposition with $B=U^TU$, and solves the standard eigenvalue problem $\tilde{A}y=\lambda y$ using Householder-QR method by converting A to $\tilde{A}=U^{-T}AU^{-1}$.

If eigenvectors are required, the eigenvector y of $\tilde{A}$ is converted using $x=U^{-1}y$.

(2) Directions

CALL GHQRVV/W(A, B, KK, N, E, EPS, W, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Only the upper right half of a real symmetric matrix is entered. It is processed in this routine, and $\tilde{A}$ is generated in the upper right half. The lower left half is used as a work area. If eigenvectors are obtained, they are entered in each column. The vectors are normalized in the meaning of $x^TBx=1$.
B	Real type Two-dimensional array	Input/output	Only the upper right half of a symmetric positive definite matrix is input. It is processed in this routine, and the Cholesky decomposition element U of B is entered in the upper right half. The lower left half is retained.

182

Argument	Type and kind (*1)	Attribute	Content
KK	Integer type	Input	Value of the first subscript in the declaration of arrays A and B. KK≧N
N	Integer type	Input	Order of arrays A and B. N≧2
B	Real type One-dimensional array	Output	One-dimensional array containing N elements. In GHQRVV/W, eigenvalues are arranged in algebraically descending order.
EPS	Real type	Input	|EPS| is the convergence criterion of the QR method. It is also the positivity criterion at the Cholesky decomposition of B. If this routine is called with EPS<0, the Cholesky decomposition elements of B are reused. EPS≠0
W	Real type One-dimensional array	Work area	One-dimensional array containing 2N elements.
IND	Integer type	Input/output	This argument has the following meaning as an input argument. IND=0: Only eigenvalues are calculated. IND≠0: Eigenvectors are calculated. This argument has the following meaning as an output argument. IND=0: Calculation was normally executed. IND=1: B is decided to be not positive definite. IND=30000: Limits on the input argument are exceeded. Because this argument is used for both input and output, constants must not be used as real arguments.

*1 For single precision subroutines, all real types should be changed to double precision real types.

(3) Calculation method

The symmetric positive definite matrix B is Cholesky-decomposed to $B=U^TU$ using the upper triangular matrix U. If $\tilde{A}=U^{-T}AU^{-1}$ is made from A by using this U, the generalized eigenvalue problem $Ax=\lambda Bx$ becomes the standard eigenvalue problem $\tilde{A}\tilde{x}=\lambda\tilde{x}$. By solving this problem using Householder QR method, the eigenvector x is obtained with $x=U^{-1}\tilde{x}$.

(4) Note

1. If only a part of eigenvectors is to be obtained, it may be advantageous to use Householder-QR-inverse iteration method (GHQRIV/W).

2. If calculation is to be iterated with B kept constant and only A changed, it is desirable to reuse the Cholesky decomposition elements of B. See the explanation for argument EPS.

(1987. 08. 10) (1988. 04. 04)

HEQRVS/D/Q (Eigenvalue analysis for real nonsymmetric matrices by double QR method)

Eigenvalue Analysis for Real Nonsymmetric Matrices by Double QR Method

Programmed by	Ichizo Ninomiya; Revised in April 1977, April 1981
Format	Subroutine language; FORTRAN Size; 391 lines each

(1) Outline

A real non-symmetric matrix is transformed into an upper Hessenberg matrix by stabilized elementary transformation. The double QR method is then applied to this to determine all eigenvalues, and a specified number of corresponding eigenvectors are determined by the inverse iteration method. The HEQRVS/D/Q subroutine is used for single (double, quadruple) precision.

(2) Directions

CALL HEQRVS/D(A, KA, N, E, F, G, H, NV, EPS, IW, W, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Matrix subjected to eigenvalue analysis. This matrix is transformed by this routine into an upper Hessenberg type.
KA	Integer type	Input	Adjustable dimensions of A, G, and H (value of the first subscript in array declaration). KA≧N
N	Integer type	Input	Order of A. Number of rows of G and H. It is also the size of E and F. N≧3
E	Real type one-dimensional array	Output	Real part of eigenvalues. The Ith eigenvalue is given by E(I) + iF(I).

Argument	Type and kind (*1)	Attribute	Content
F	Real type one-dimensional array	Output	Imaginary part of eigenvalues. The Ith eigenvalue is given by B(I) + iF(I).
G	Real type two-dimensional array	Output	The real part of the Ith eigenvector is output in the Ith column of G. It must have the area for NV+1 columns.
H	Real type two-dimensional array	Output	The imaginary part of the Ith eigenvector is output in the Ith column of H. Moreover, it is necessary to prepare the region of the size with N rows and N+1 columns for use as a work area.
NV	Integer type	Input	Number of eigenvectors to be determinedBecause conjugate eigenvectors are generated in pairs, number of actually generated vectors can be NV+1. $0 \leqq NV \leqq N$
EPS	Real type	Input	$\|\|A\|\| \cdot EPS/N$ is used as a convergence criterion constant for QR method. EPS>0
IW	Integer type one-dimensional array	Work area	One-dimensional array with the size of 2N or more.
W	Real type one-dimensional array	Work area	One-dimensional array with the size of 2N or more.

Argument	Type and kind (*1)	Attribute	Content
IND	Integer type	Input/output	Input: A mode for arrangement of eigenvalues is specified. IND = 0: Eigenvalues are arranged as they are calculated. IND > 0: Eigenvalues are arranged in descending order of the absolute values. IND < 0: Eigenvalues are arranged in ascending order of the absolute values. Output: Condition code IND = 0: Normal IND = 1: All elements of A are 0. IND = 2: Convergence did not occur even if the QR method was repeated 100N times. IND = 30000: The input argument violated the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

Real matrix A is transformed into an upper Hessenberg matrix $H=S^{-1}AS$ by stabilized elementary transformation S, that is, Gaussian elimination involving row exchange. All eigenvalues of H are determined by the double QR method with origin shift.

The specified number of eigenvectors of H is determined by the inverse iteration method. Let they be grouped into a matrix U. Eigenvectors of A are calculated by $V=SU$ using U.

(4) Note

1. It is reasonable to process symmetric matrices by special routines HOQRVS/D, HQRIIS/D, and HOBSVS/D.

2. If no eigenvectors are to be determined (NV=0), the areas for G and H are not used and need not be prepared, and anything can be written for them.

(1987.07.20)

HEQRVV/W (Eigenvalue Analysis for Real Nonsymmetric Matrices by Double QR Method - Vector Version -)

Eigenvalue Analysis for Real Nonsymmetric Matrices by Double QR Method -Vector Version-

Programmed by	Ichizo Ninomiya, December 1984
Format	Subroutine language: FORTRAN77; size: 485 and 486 lines respectively

(1) Outline

HEQRVV/W transforms a real nonsymmetric matrix to an upper Hessenberg matrix using the stabilized elementary transformation, obtains all the eigenvalues using the double QR method, and calculates the eigenvectors as many as requested using the inverse iteration. It is a single (double) precision subroutine.

(2) Directions

CALL HEQRVV/W(A, KA, N, E, F, G, H, NV, EPS, IW, W, IND)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Matrix whose eigenvalue analysis is to be executed. The matrix is processed with this routine, and transformed to a Hessenberg type.
KA	Integer type	Input	Adjustable dimensions of A, G, and H (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A. Number of rows in G and H. It also represents the size of E and F. N≧3

Argument	Type and kind (*1)	Attribute	Content
E	Real type one-dimensional array	Output	Real part of eigenvalues. The I-th eigenvalue is E(I)+iF(I).
F	Real type one-dimensional array	Output	Imaginary part of eigenvalues. The I-th eigenvalue is E(I)+iF(I).
G	Real type two-dimensional array	Output	Real part of the I-th eigenvector is output to the I-th column of G. Space for NV+1 columns must be provided.
H	Real type two-dimensional array	Output	The imaginary part of the I-th eigenvector is output to the I-th column of H. Because this argument is used as a work area, the area of size N×(N+1) must be provided.
NV	Integer type	Input	Number of eigenvectors to be obtained. Because conjugate eigenvectors are output as a pair, the number of vectors that are actually output may be NV+1. $0 \leq NV \leq N$
EPS	Real type	Input	$\|\|A\|\| \cdot EPS/N$ is used as the convergence criterion of QR. EPS>0
IW	Integer type one-dimensional array	Work area	One-dimensional array of size 2N.

Argument	Type and kind (*1)	Attribute	Content
W	Real type one-dimensional array	Work area	One-dimensional array of size 2N.
IND	Integer type	Input/output	Input: Arrangement of eigenvalues is specified. IND=0: Eigenvalues are kept in the state as calculated. IND>0: Eigenvalues are arranged in the descending order of absolute values. IND<0: Eigenvalues are put in the ascending order of absolute values. Output: Condition code. IND=0: Normal. IND=1: The elements in A are all zeros. IND=2: Convergence may not occur even if the QR method is iterated 100N times. IND=30000: The input argument exceeded the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

The real matrix A is transformed to the upper Hessenberg matrix $H=S^{-1}AS$ using the stabilized elementary transformation S, that is, Gauss' elimination accompanied by row exchange. All the eigenvalues of H are obtained using the double QR method with origin shift.

Specified number of eigenvectors of H are obtained using the inverse iteration and are placed in the matrix U. The eigenvectors of A are calculated by $V=SU$ from U.

(4) Notes

1. It is reasonable to process symmetric matrices with the special-purpose routines HOQRVV/W, HQRIIV/W, and HOBSVV/W.

2. If eigenvectors are not to be obtained (NV=0), the area to G and H is not used, and thus need not be prepared. Anything can be written for these arguments.

(1987. 06. 19)

HOBSVS/D/Q (Eigenvalue Analysis for Real Symmetric Matrices by Householder-Bisection Method)

Eigenvalue Analysis for Real Symmetric Matrices by Householder-Bisection Method

Programmed by	Ichizo Ninomiya, April 1977, revised in April 1981
Format	Subroutine language: FORTRAN; size: 173 and 171 lines respectively

(1) Outline

HOBSVS/D/Q tridiagonalizes a real symmetric matrix using Householder's reflexion transformation, obtains the eigenvalues of the tridiagonalized matrix using the bisection method based on Sturm sequence, and calculates the eigenvectors using the inverse iteration. It is for single (double) precision.

(2) Directions

CALL HOBSVS/D/Q(A, KA, N, E, NB, V, NV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Only the upper right half containing the diagonal lines of a real symmetric matrix is input. It is processed with this routine. The lower left half is retained.
KA	Integer type	Input	Adjustable dimensions of A and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A. It also represents the number of rows of V. N≧1
E	Real type One-dimensional array	Output	Eigenvalues are output in the order of size. If NB>0, eigenvalues are arranged in descending order. If NB<0, eigenvalues are arranged in ascending order.

Argument	Type and kind (*1)	Attribute	Content
NE	Integer type	Input	The number of eigenvalues to be obtained is represented by the absolute value. If NE>0 (NE<0), eigenvalues are numbered in algebraically descending (ascending) order from the maximum (minimum). NE≠0
V	Real type two-dimensional array	Output	Eigenvectors corresponding to the eigenvalue E(I) are normalized to a length of 1 and output to the I-th column.
NV	Integer type	Input	The number of eigenvectors to be obtained is represented by the absolute value. Eigenvalues are numbered from the end in the order defined by NE. $0 \leqq \|NV\| \leqq \|NE\|$
EPS	Real type	Input	Convergence criterion of bisection method. If a tridiagonalized matrix is denoted by T, $\|\|T\|\| \cdot EPS$ is used for the criterion. EPS>0
W	Real type One-dimensional array	Work area	One-dimensional array of size 6N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: The input argument exceeded the limit.

1* For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

The matrix A is transformed to the tridiagonal matrix $T=H^TAH$ using the Householder transformation H.

The eigenvalues of T are obtained by the bisection method based on Sturm sequence. They are numbered as many as specified from the end in a specified order. The eigenvectors corresponding to the eigenvalues are obtained by the inverse iteration. The matrix containing these

eigenvectors in columns is denoted by U. Because U is the eigenvector of T, it is converted to the eigenvector V of A by $V=HU$.

(4) Note

When all the eigenvalues of a symmetric matrix are to be obtained, it is better to use the routine HOQRVS/D based on the QR method. When all of the eigenvalues and all or a part of the eigenvectors are to be obtained, it is more reasonable to use the routine HQRIIS/D based on the QR-inverse iteration method.

Bibliography

1) Yoshitaka Beppu and Ichizo Ninomiya: "Comparison of Matrix Solutions of Standard Eigenvalue Problems," Nagoya University Computer Center News, Vol. 11, No. 3, pp. 265-274 (1980).

(1987. 08. 10) (1987. 08. 21)

HOBSVV/W (Eigenvalue Analysis for Real Symmetric Matrices by Householder-Bisection Method - Vector Version -)

Eigenvalue Analysis for Real Symmetric Matrices by Householder-Bisection Method -Vector Version-

Programmed by	Ichizo Ninomiya, December 1984
Format	Subroutine language: FORTRAN77; size: 345 and 346 lines respectively

(1) Outline

HOBSVV/W tridiagonalizes a real symmetric matrix using Householder's reflexion transformation, obtains the eigenvalue of the resultant tridiagonal matrix using the bisection method based on Sturm sequence, and calculates the eigenvectors using the inverse iteration. It is for single (double) precision.

(2) Directions

CALL HOBSVV/W(A, KA, N, E, NE, V, NV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Whole of a real symmetric matrix is input. It is processed with this routine.
KA	Integer type	Input	Adjustable dimensions of A and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A. It also represents the number of rows of V. N≧2

Argument	Type and kind (*1)	Attribute	Content
E	Real type One-dimensional array	Output	One-dimensional array of size N. Eigenvalues are output in the order of size. If NE>0, eigenvalues are arranged in descending order. If NE<0, eigenvalues are arranged in ascending order.
NE	Integer type	Input	The number of eigenvalues to be obtained is represented by the absolute value. If NE>0 (NE<0), eigenvalues are numbered in algebraically descending order (ascending order) from the maximum (minimum). NE≠0
V	Real type two-dimensional array	Output	Eigenvectors to the eigenvalue E(I) are normalized to length 1 and placed to the I-th column.
NV	Integer type	Input	The number of eigenvectors to be obtained is represented by the absolute value. Eigenvalues are numbered from the end in the order specified by NE. $0 \leq \|NV\| \leq \|NE\|$
EPS	Real type	Input	Convergence criterion of bisection method. If a tridiagonalized matrix is denoted by T, $\|\|T\|\| \cdot$EPS is used as the criterion. EPS>0
W	Real type One-dimensional array	Work area	One-dimensional array of size 6N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: The input argument exceeded the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

The matrix A is transformed to a tridiagonal matrix $T=H^TAH$ using Householder's

transformation H.

The eigenvalues of T are obtained using the bisection method based on Sturm sequence. They are numbered as many as specified from the end in a specified order. The eigenvectors corresponding to the specified number of eigenvalues numbered from the end are obtained using the inverse iteration. A matrix containing these eigenvectors in its columns is denoted by U, then the eigenvector V of A can be obtained by $V=HU$.

(4) Notes

When all the eigenvalues of a symmetric matrix are to be obtained, it is better to use the routine HOQRVV/W based on the QR method than this routine. When all of eignevalues and all or part of eigenvectors are to be obtained, it is more reasonable to use the routine HQRIIV/W based on the QR inverse iteration.

Bibliography

1) Yoshitaka Beppu and Ichizo Ninomiya; "Comparison of Matrix Solutions of Standard Eigenvalue Problems," Nagoya University Computer Center News, Vol. 11, No. 3, pp. 265-274 (1980)

(1987. 06. 19) (1988. 02. 22)

HOQRVS/D/Q and HOQRUS/D/Q (Eigenvalue analysis for real symmetric matrix by Householder-QR method)

Eigenvalue Analysis for Real Symmetric Matrices by Householder-QR Method

Programmed	Ichizo Ninomiya April, 1977
Format	Subroutine Language; FORTRAN Size; 142, 141, 142, 141 lines

(1) Outline

All eigenvalues of the real symmetric matrix and all eigenvectors, if necessary, are calculated by Householder's tridiagonalization and QR method with origin shift.

(2) Directions

CALL HOQRVS/D/Q(A, KA, N, E, F, EPS, ILL)

CALL HOQRUS/D/Q(A, KA, N, E, F, EPS, ILL)

Argument	Type and Kind *	Attribute	Content
A	Real type Two-dimensional array	Input	Only the right upper half which contains the diagonal of the real symmetric matrix need be input. Anything can be input in the left lower half. When eigenvectors are calculated, they are stored in each column of A. Precisely, the eigenvector normalized to the unit length corresponding to the eigenvalue E(I) is stored in the Ith column.
KA	Integer type	Input/Output	Value of the first subscript in the array declaration of A. KA≧N
N	Integer type	Input	Order of A. N≧2

Argument	Type and Kind *	Attribute	Content
E	Real type One-dimensional array	Output	One-dimensional array name with N elements. In HOQRVS/D, eigenvalues are arranged in the decreasing algebraic order, and in HOQRUS/D, they are arranged in the decreasing order of absolute value.
F	Real type One-dimensional array	Work area	One-dimensional array name with N elements.
EPS	Real type	Input	Convergence criterion for QR method. When a nondiagonal element becomes smaller than $\|\|A\|\| \cdot EPS$, it is regarded to have converged to 0. EPS>0
ILL	Integer type	Input/Output	If ILL=0 is input, only eigenvalues are calculated. If ILL≠0 is input, eigenvalues and eigenvectors are calculated. 0 is output for normal end. 30000 is output if the input argument limit is exceeded. Constants should not be used for the actual argument corresponding to this argument.

* All real types should be changed to double precision real types in the case of the subroutine for double precision.

(3) Performance

Compared with Jacobi method, this subroutine has high speed and can be used for the multiple or close eigenvalues without any trouble.

(4) Note

1. This subroutine is optimal to calculate all eigenvalues (and eigenvectors) using small memory space.

2. Householder and Givens' method (bisection) are better in case of calculating only a part of eigenvalues and eigenvectors. The appropriate subroutine is HOBSVS/D.

3. Subroutine GHQRVS/D is recommended to solve generalized eigenvalue problems $A\chi=\lambda B\chi$.

Bibliography

1) Hayato Togawa; "Numerical calculation of matrix", Ohm-sha (1971).

(1987. 08. 10) (1987. 08. 21)

HOQRVV/W (Eigenvalue Analysis for Real Symmetric Matrices by Householder-QR Method - Vector Version -)

Eigenvalue Analysis for Real Symmetric Matrices by Householder-QR Method -Vector Version-

Programmed by	Ichizo Ninomiya, December 1984
Format	Subroutine language: FORTRAN77; size: 233 and 234 lines respectively

(1) Outline

HOQRVV/W obtains all the eigenvalues and, if required, all the corresponding eigenvector of a real symmetric matrix, using the Householder's tridiagonalization and the QR method with origin shift. It is for single (double) precision.

(2) Directions

CALL HOQRVV/W(A, KA, N, E, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Whole of a real symmetric matrix is input. If eigenvectors are obtained, they are entered to columns of A. That is, eigenvectors corresponding to the eigenvalue E(I) is normalized to length 1 and placed to the I-th column.
KA	Integer type	Input/output	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Order of A. N≧2
E	Real type One-dimensional array	Output	One-dimensional array containing N elements. Eigenvalues are arranged in algebraically descending order.

Argument	Type and kind (*1)	Attribute	Content
EPS	Real type	Input	Convergence criterion for QR method. If all the non-diagonal elements become smaller than $\|\|A\|\| \cdot EPS$ in magnitude, convergence is judged to have occurred. EPS>0
W	Real type One-dimensional array	Work area	One-dimensional array with 2N elements.
ILL	Integer type	Input/output	If ILL=0 is given, only eigenvalues are calculated. If ILL≠0, eigenvalues and corresponding eigenvectors are calculated. If calculation terminates normally, 0 is output. If limits on input arguments are exceeded, 30000 is output. Constants must not be used for actual arguments for this argument.

*1 For double precision subroutines, all real types are assumed to be double precision real types.

(3) Calculation method

The symmetric matrix A is transformed to a tridiagonal matrix $T=H^TAH$ using the Householder transformation H. The matrix T is diagonalized to $D=Q^TTQ$ using the QR transformation. The eigenvectors of A are calculated as $V=HQ$.

(4) Notes

1. This routine is optimum if all eigenvalues (and corresponding eigenvectors) are to be obtained with a small size of storage.
2. If only part of eigenvalues and eigenvectors is to be obtained, Householder-Givens' method (bisection) is better. HOBSVV/W is a suitable subroutine.

Bibliography

1) Hayato Togawa: Numerical Calculation of Matrices, Ohm-sha, 1971

(1987.06.19)

HQRIIS/D/Q (Eigenvalue Analysis of Symmetric Matrices by Householder-QR-Inverse Iteration Method)

Eigenvalue Analysis of Symmetric Matrices by Householder-QR-Inverse Iteration Method

Programmed by	Ichizo Ninomiya, April 1981
Format	Subroutine Language: FORTRAN; Size: 198 and 196 lines respectively

(1) Outline

HQRIIS/D/Q obtains all eigenvalues of real symmetric matrices by Householder-QR-method, and calculates specified eigenvectors by the inverse iteration method.

(2) Directions

CALL HQRIIS/D/Q(A, KA, N, E, V, NV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	The right upper half containing the diagonal of a real symmetric matrices is input. It is processed by this routine. The lower left half is retained.
KA	Integer type	Input	Adjustable dimensions of A and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A or the number of rows of V. N≧2
E	Real type One-dimensional array	Output	All eigenvalues are output in the order of size. If NV≧0, they are arranged in decreasing order. If NV<0, they are arranged in increasing order.

Argument	Type and kind (*1)	Attribute	Content
V	Real type Two-dimensional array	Output	Eigenvectors to eigenvalues E(I) are normalized to 1, and placed to the I-th column.
NV	Integer type	Input	|NV| represents the number of eigenvectors to be obtained. If NV>0 (NV<0), the eigenvectors are numbered in algebraically decreasing (or increasing) order from the maximum (or minimum). |NV|≦N
EPS	Real type	Input	Convergence criterion of QR method. If the tridiagonalized matrix is denoted by T, ‖T‖·EPS is used as the criterion. EPS>0
W	Real type One-dimensional array	Work area	One-dimensional array of size 6N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: Input argument exceeded the limit.

*1 For double precision subroutines, all real types should be changed to double precision real types.

(3) Calculation method

Transform the symmetric matrix A to a tridiagonal matrix $T=H^TAH$ by Householder transformation H.

Obtain all eigenvalues of T by the QR method without square root. Obtain a specified number of eigenvectors of T by the inverse iteration method. Put these eigenvectors into the matrix U. The eigenvectors of A are calculated by $V=HU$.

(4) Note

1. It is best to use this routine to obtain all of eigenvalues and all or part of the corresponding eigenvectors quickly.

2. It is better to use HOBSVS/D based on the Householder bisection method to obtain up to one-fourth of all eigenvalues.

Bibliography

1) Parlett and B.N;"The Symmetric Eigenvalue Problem" Prentice-Hall (1980).

(1987.08.10) (1987.08.21) (1988.02.22)

206

HQRIIV/W (Eigenvalue Analysis of Symmetric Matrices by Householder-Inverse-QR Iteration Method - Vector Version -

Eigenvalue Analysis of Symmetric Matrices by Householder-QR-Inverse Iteration Method -Vector Version-

Programmed by	Ichizo Ninomiya, December 1984
Format	Subroutine language: FORTRAN77; size: 345 and 346 lines respectively

(1) Outline

HQRIIV/W obtains all the eigenvalues of real symmetric matrices using the Householder-QR method, and calculates the specified eigenvectors based on the inverse iteration. It is for single (double) precision.

(2) Directions

CALL HQRIIV/W(A, KA, N, E, V, NV, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Whole of a real symmetric matrix is input. It is processed with this routine.
KA	Integer type	Input	Adjustable dimensions of A and V (value of the first subscript in the array declaration). KA≧N
N	Integer type	Input	Order of A. It also represents the number of rows of V. N≧2

Argument	Type and kind (*1)	Attribute	Content
E	Real type One-dimensional array	Output	All eigenvalues are output in the order of size. If NV≧0, eigenvalues are arranged in descending order. If NV<0, eigenvalues are arranged in ascending order.
V	Real type Two-dimensional array	Output	Eigenvectors to the eigenvalue E(I) are normalized to length 1 and placed to the I-th column.
NV	Integer type	Input	|NV| represents the number of eigenvectors to be obtained. If NV>0 (NV<0), eigenvectors are numbered in algebraically descending (ascending) order from the maximum (minimum) value. |NV|≦N
EPS	Real type	Input	Convergence criterion constant of QR method. If a tridiagonalized matrix is denoted by T, ‖T‖·EPS is used as the convergence criterion. EPS>0
W	Real type One-dimensional array	Work area	One-dimensional array of size 6N.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: Input arguments exceeded the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

The symmetric matrix A is transformed to a tridiagonal matrix $T=H^TAH$ using the Householder transformation H.

All the eigenvalues of T are calculated using the square-root-less QR method. A specified number of eigenvectors of T are obtained using the inverse iteration and are placed in a matrix U. The eigenvectors of A are calculated by $V=HU$.

(4) Notes

1. This routine is optimum when all of eigenvalues are obtained quickly, and all or part of corresponding eigenvectors are obtained.

2. When up to about one-fourth of the entire eigenvalues is to be obtained, it is more advantageous to use HOBSVV/W based on Householder-bisection method.

Bibliography

1) Parlett, B. N; "The Symmetric Eigenvalue Problem" Prentice-Hall (1980).

(1987. 06. 19) (1987. 08. 07) (1988. 02. 22)

JACOBS/D (Eigenvalue Analysis for a Real Symmetric Matrix by Threshold Jacobi Method)

Eigenvalue Analysis for a Real Symmetric Matrix by Threshold Jacobi Method

Programmed by	Ichizo Ninomiya, April 1977
Format	Subroutine language: FORTRAN; size: 88 and 88 lines respectively

(1) Outline

JACOB/D calculates all the eigenvalues and eigenvectors of a given real symmetric matrix using the threshold Jacobi method.

(2) Directions

CALL JACOBS/D(A, KA, N, EPS, V, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input/output	Real symmetric matrix. Only the upper right half including the diagonal lines need be given. Eigenvalues are output on the diagonal. The lower left half is preserved.
KA	Integer type	Input	Value of the first subscript in the array-A declaration. KA≧N
N	Integer type	Input	Order of A and V. N≧2
EPS	Real type	Input	Convergence criterion constant. The average absolute value of nondiagonal elements of the input matrix A is assumed as standard. This value multiplied by EPS is used as the standard of convergence decision. EPS>0
V	Real type Two-dimensional array	Output	Each column stores an eigenvector for the corresponding diagonal element A.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: Limits on K, N, and EPS are violated.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Performance

This routine was believed to be advantageous for multiple or close eigenvalues as a method of finding all the eigenvalues and engenvectors of a symmetric matrix. However, since Householder QR method with the same advantage for the same purpose appeared, this routine became obsolete recently.

(4) Remarks

1. Eigenvalues are arranged in descending order along the diagonal of A.

2. A value of $10^{-6}(10^{-16})$ is adequate as the standard value of EPS for JACOBS (JACOBD).

3. Except for small-size problems of about 10, it is advantageous to use the subroutines HOQR and VS using Householder QR method for the same purpose because computation time is significantly saved.

(1987. 06. 17) (1987. 08. 07)

JENNFS/D, JENNBS/D, and GJENBS/D (Eigenvalue analysis of real symmetric matrices by Jennings' simultaneous iteration method)

Eigenvalue Analysis for Real Symmetric Matrices by Jennings' Simultaneous Iteration Method

(Eigenvalue analysis of real symmetric matrices by Jennings' simultaneous iteration method)

Programmed by	Ichizo Ninomiya; April 1981
Format	Subroutine language: FORTRAN. Size; 141, 142, 151, 152, 184, and 185 lines respectively

(1) Outline

A part of eigenvalues and corresponding eigenvectors of real symmetric matrices are determined by the Jennings' simultaneous iteration method accompanying the Jennings' vector acceleration method. JENNFS/D is used to solve standard eigenvalue problem $(A-\lambda I)x=0$ where A is a dense matrix, and JENNBS/D is used to solve the same problem where A is a band matrix. GJENBS/D is used to solve generalized eigenvalue problem $(A-\lambda B)x=0$ where A is a band matrix and B is a positive definite band matrix. JENNFS, JENNBS, and GJENBS are single precision subroutines and JENNFD, JENNBD, and GJENBD are double precision subroutines.

(2) Directions

CALL JENNFS/D(A, KA, N, L, M, V, E, C, W, EPS, ITER, ILL)

CALL JENNBS/D(A, KA, N, NB, L, M, V, KV, E, C, W, EPS, ITER, ILL)

CALL GJENBS/D(A, B, KA, N, NB, L, M, V, KV, E, C, W, EPS, ITER, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	The entire symmetric matrix is input for JENNFS/D. For JENNBS/D and GJENBS/D, the lower left half which contains the diagonal of band matrix is input after it is made to a rectangle as shown in the figure. That is, elements I and J of the matrix are put in A(I-J+1, J). When eigenvalues are determined in ascending order of their absolute values, Cholesky decomposition is done by this routine.
B	Real type Two-dimensional array	Input	A band matrix is input in the same way as for A. When eigenvalues are determined in descending order of their absolute values, Cholesky decomposition is done by this routine.
KA	Integer type	Input	Adjustable dimension of A and V in case of JENNFS/D KA≧N Adjustable dimension of A in case of JENNBS/D Adjustable dimension of A and B in case of GJENBS/D } KA≧NB
N	Integer type	Input	Order of A and B. N≧2
NB	Integer type	Input	Half band width of A and B. NB≧2
L	Integer type	Input	|L| indicates the number of eigenvalues and eigenvectors to be obtained. L>0 (L<0) indicates that they should be arranged in the descending (ascending) order of absolute values. 1≦|L|≦N
M	Integer type	Input	Number of trial vectors |L|≦M≦N
V	Real type Two-dimensional array	Input/output	The M initial eigenvectors are input. Eigenvectors are generated to the first |L| columns.

Argument	Type and kind (*1)	Attribute	Content
KV	Integer type	Input	Adjustable dimension of V. KV≧N
B	Real type One-dimensional array	Output	Eigenvalues are generated in the order specified by L.
C	Real type One-dimensional array	Work area	One-dimensional array with a size of M^2 or more
W	Real type One-dimensional array	Work area	One-dimensional array with a size of 3N or more for JENNFS/D and JENNBS/D and 4N or more for GJENBS/D.
EPS	Real type	Input	Convergence criterion constant. EPS>0
ITER	Integer type	Input/output	Input: Upper bound of repetition numberWhen it is less than N, it is put to 1000. Output: Actual repetition number
ILL	Integer type	Output	ILL=0: Normal. ILL=1: The repetition number exceeded the upper bound. ILL=2: Cholesky's decomposition was impossible. ILL=30000: The input argument violated the limit.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

To simplify description, a standard type problem is abbreviated as T and a general type problem is abbreviated as G. Similarly, L indicates the case of determining an eigenvalue with a large absolute value, and S indicates the case of determining an eigenvalue with a small absolute

value.

1. The initial eigenvectors are grouped into a matrix. V with n rows and m columns, that is, the m vectors are put together into the matrix. Select m such that $l \leqq m \leqq n$ where l is the number of eigenvalues to be obtained (see the notes for selection of initial vectors).

(T, S) A is processed by the modified Cholesky's decomposition to produce $A=R^TDR$.

(G, S) A is processed by the Cholesky's decomposition to produce $A=\tilde{A}^T\tilde{A}$. $\tilde{A}V$ is generated and overwritten on V.

(G, L) B is processed by the Cholesky's decomposition to produce $B=\tilde{B}^T\tilde{B}$. $\tilde{B}V$ is generated and overwritten on V.

2. (T, S) Compute $U=R^{-1}D^{-1}R^{-T}V(=A^{-1}V)$.

(T, L) Compute $U=AV$.

(G, S) Compute $U=\tilde{A}^{-T}B\tilde{A}^{-1}V$.

(G, L) Compute $U=\tilde{B}^{-T}A\tilde{B}^{-1}V$.

3. Form $G=V^TU$. G is a symmetric matrix with m rows and m columns.

4. G is diagonalized into $P^TGP=Q$, where Q is a diagonal matrix with diagonal element $\mu_1, \mu_2 \cdots\cdots \mu_m (|\mu_1| \geqq |\mu_2| \geqq \cdots\cdots \geqq |\mu_m|)$ which is the eigenvalue of G, and P is an orthogonal matrix having eigenvectors as rows.

5. Compute $W=UP$.

6. Compute W^TW and process it by the Cholesky's decomposition to obtain $W^TW=S^TS$. S is an $m \times m$ upper triangular matrix.

7. $V=WS^{-1}$ is formed. V is an orthogonal matrix in the sense of $V^TV=I_m$ (unit matrix of the order m).

8. Convergence test is made (see the notes below). If convergence has not been attained, Jennings' vector acceleration is applied. The result is used as new V and processing returns to step 2.

9. If convergence has been completed:

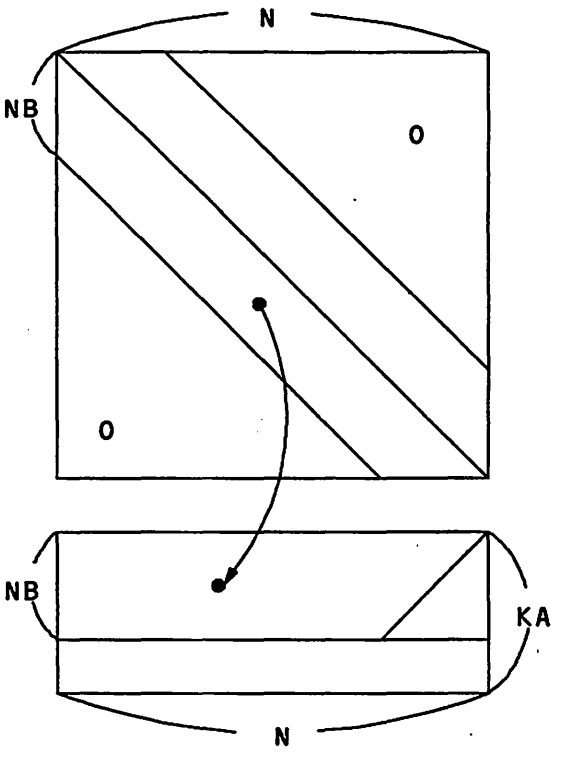
(S) $1/\mu_1, 1/\mu_2, \cdots, 1/\mu_l$ are assumed to be eigenvalues.

(L) $\mu_1, \mu_2, \cdots, \mu_l$ are assumed to be eigenvalues.

(T) The first l columns of V are assumed to be eigenvectors.

(G, S) The first l columns of $\tilde{A}^{-1}V$ are assumed to be eigenvectors.

(G, L) The first l columns of $\tilde{B}^{-1}V$ are assumed to be eigenvectors.



(4) Notes

1. It is desirable that the initial vectors are closed to true eigenvectors. If there is no information available for the initial value, however, a common way is to use a fragment of the unit matrix. For selection of m, it should be as close to l as possible and also satisfy $|\lambda_m| / |\lambda_{m+1}| \gg 1$ (or $|\lambda_m| / |\lambda_{m+1}| \ll 1$) when eigenvalues are arranged as $\lambda_1, \lambda_2, \cdots, \lambda_n$ in order of their absolute values. The quantity of calculation for each iteration is generally proportional to m.

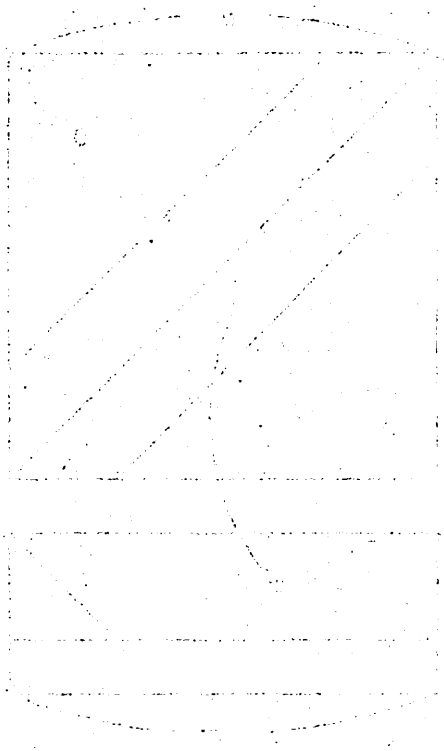
2. A convergence test is made for the components of eigenvectors. In general, eigenvalues converge in much better precision than eigenvectors. Especially, when separation of eigenvalues is good, precision is about twice as good as that for eigenvectors. From this reason, therefore, it is safer to select a little larger value than usual for criterion constant EPS.

3. A standard value of ITER indicating the upper limit of the repetition number is a few hundreds. Do not write a constant as an actual argument for this variable because this variable is used for both input and output.

Bibliography

1) A. Jennings; "Matrix Computation for Engineers and Scientists", John Wiley, London, (1977)

(1989. 01. 20)



NGHOUS/D (Analysis of Av=λBv type eigenvalue by bi-triangular decomposition, Householder, bisection-QR, and inverse iteration methods)

Nicer for Generalized Eigenvalue-Problem by Householder Method

Programmed by	Yoshitaka Beppu and Ichizo Ninomiya; December 1981
Format	Subroutine language; FORTRAN Size; 104 and 105 respectively

(1) Outline

NGHOUS and NGHOUD solve generalized eigenvalue problem concerning real symmetric dense matrix A and real symmetric positive definite dense matrix B by the semi-direct method.

(2) Directions

CALL NGHOUS/D(AB, NMAX, N, NE, NV, EPS, IORD, ICHO, BD, E, V, ILL, W1, W2, W3, W4, W5, W6, W7)

Argument	Type and kind (*1)	Attribute	Content
AB	Real type Two-dimensional array	Input/output	$A_{ij}(i \leq j)$ is input to the upper right half including diagonal elements. The upper right half changes. If ICHO=0, $B_{ij}(i>j)$ is input to the lower left half. If ICHO=1, non-diagonal element $L_{ij}(i>j)$ of Cholesky decomposition component L of B is input to the lower left half. L_{ij} is output to the lower left half.
NMAX	Integer type	Input	Adjustable dimensions of AB and V. N≦NMAX
N	Integer type	Input	Order of A and B. 2≦N

Argument	Type and kind (*1)	Attribute	Content
NE	Integer type	Input	Number of eigenvalues to be determined. $0<NE\leqq N$
NV	Integer type	Input	Number of eigenvectors to be determined. $0\leqq NV\leqq NE\leqq N$
EPS	Real type	Input	Tolerance for convergence test. The default value is 10^{-6} (NGHOUS) or 10^{-10} (N-GHOUD).
IORD	Integer type	Input	The output order of eigenvalues is specified. If IORD>0, they are output in algebraically descending order. If IORD<0, they are output in algebraically ascending order.
ICHO	Integer type	Input	The input mode of real symmetric positive definite matrix B is specified. Refer to the descriptions of AB and BD.
BD	Real type One-dimensional array	Input/output	If ICHO=0, diagonal element B_{ii} of B is input to BD(I). If ICHO=1, the inverse number of L's diagonal element L_{ii} is input to it. L_{ii}^{-1} is output to BD(I).
E	Real type One-dimensional array	Output	The Ith eigenvalue is output to E(I). If IORD is positive, then E(1)>E(2)>……>E(NE). If IORD is negative, then E(1)<E(2)<……<E(NE).
V	Real type Two-dimensional array	Output	The eigenvector which corresponds to E(I) is normalized as $v^TBv=1$ and output to column I.
ILL	Integer type	Output	ILL=0: Normal termination ILL=100: B is a non-positive definite. ILL=300: The argument is abnormal.

Argument	Type and kind (*1)	Attribute	Content
W1~W7	Real type One-dimensional array	Work area	The size must be N or more.

(3) Calculation method

First of all, generalized eigenvalue problem $(Av=\lambda Bv)$ is transformed into standard eigenvalue problem $(\tilde{A}u=\lambda u)$ by the bi-triangular decomposition method.
That is, matrix A is decomposed to the sum of upper right triangular matrix R and lower left triangular matrix R^T $(A=R+R^T)$, and matrix B is decomposed to the product of lower left triangular matrix L and upper right triangular matrix L^T $(B=LL^T)$. $\tilde{A}=L^{-1}(R+R^T)L^{-T}$ can thus be calculated efficiently. Because A is a real symmetric matrix, λ and ortho-normal vector u are determined by NSHOUS/D, and generalized orthogonal vector v is determined by $v=L^{-T}u$.

(4) Notes

1. NGHOUS or NGHOUD is 1.05 times as fast as GHQRID and GHBSVD of NUMPAC. If 1 is specified for ICHO when L is known, it becomes about 5% faster.
2. NGHOUS and NGHOUD are useful when approximate solutions are unknown.
3. These routines are also components of NICER.

(1987.06.16)

NGJENS/D (Analysis of $Av=\lambda Bv$ type eigenvalues by bi-triangular decomposition and Jennings method)

Nicer for Generalized Eigenvalue-Problem by Jennings Method

Programmed by	Yoshitaka Beppu and Ichizo Ninomiya; December 1981
Format	Subroutine language; FORTRAN Size; 89 and 90 lines respectively

(1) Outline

NGJENS and NGJEND solve generalized eigenvalue problems concerning real symmetric dense matrix A and real symmetric positive definite dense matrix B by the simultaneous iteration method.

(2) Directions

CALL NGJENS/D(AB, NMAX, N, NE, NV, EPS, BD, IUV, ITER, ESHIFT, E, V, U, ILL, W1, W2)

Argument	Type and kind (*1)	Attribute	Content
AB	Real type Two-dimensional array	Input/output	$A_{ij}(i \leq j)$ is input to the upper right half including diagonal elements. The upper right half changes. Off diagonal element $L_{ij}(i>j)$ of B's Cholesky-decomposed component L is input to the lower left half. The lower left half does not change.
NMAX	Integer type	Input	Adjustable dimensions of AB, V, and U. $N \leq NMAX$
N	Integer type	Input	Order of A and B. $2 \leq N$

Argument	Type and kind (*1)	Attribute	Content
NB	Integer type	Input	Number of eigenvalues to be determined. They are counted in absolutely descending order. 0<NB<N
NV	Integer type	Input	Number of eigenvectors to be determined. 0<NB≦NV<N
EPS	Real type	Input	Tolerance for convergence test. The default value is 10^{-6} (NSJBNS) or 10^{-10} (NSJBND).
BD	Real type One-dimensional array	Input	Reciprocal L_{ii}^{-1} of L's diagonal element L_{ii} is input to BD(I).
IUV	Integer type	Input	The initial-vector reference mode is specified. If IUV=0, approximate generalized orthogonal vector v_0 input to array V is used as the initial vector for the simultaneous iteration method. If IUV=1, approximate ortho-normal vector u_0 input to array U is used likewise. The content of array U when IUV=0 is not referred, and the content of array V at IUV=1 is not referred.
ITER	Integer type	Input/output	The upper limit for the number of Jennings iterations (standard value ranges from 1 to 10) is input. The number of actual iterations is output.
BSHIFT	Real type	Input	Origin shift σ. In these routines, the simultaneous iteration method is applied not to $\tilde{A}$ but to $\tilde{A}'=\tilde{A}-\sigma\cdot I$. Therefore, eigenvalues which are close to σ are rapidly diminished and the other eigenvalues are rapidly enhanced. The standard value is an approximate value of 0.5* (B(NV+1)+B(N)).

Argument	Type and kind (*1)	Attribute	Content
E	Real type One-dimensional array	Input/output	The approximate value of the eigenvalue whose absolute value is the Ith largest of all in absolute form is input to E(I). The eigenvalue whose absolute value is the Ith largest of all is output to E(I). |E(1)|>|E(2)|>·····>|E(NE)|
V	Real type Two-dimensional array	Input/output	If IUV=0, approximate generalized orthogonal vectors v_0 by the number specified by NV is input. If IUV=1, an arbitrary quantity is input. The generalized orthogonal vector which corresponds to E(I) is normalized to $v^TBv=1$ and output to the Ith column.
U	Real type Two-dimensional array	Input/output	If IUV=0, an arbitrary quantity is input. If IUV=1, approximation ortho-normal vectors u_0 by the number specified by NV is input. The ortho-normal vector which corresponds to E(I) is normalized to $u^Tu=1$ and output to the Ith column.
ILL	Integer type	Output	ILL=0: Normal termination ILL=100: L_{ij} input error ILL=200: Convergence does not occur because of poor precision of approximation vectors. ILL=300: The argument is abnormal.
W1~W2	Real type One-dimensional array	Work area	The size must be N or more.

(3) Calculation method: Approximate solutions are iteratively improved according to the following procedure:

1. $\tilde{A}=L^{-1}(R+R^T)L^{-T}$ is generated by the bi-triangular decomposition method ($A=R+R^T, B=LL^T$), and $Av=\lambda Bv$ is transformed into $\tilde{A}u=\lambda u$.

2. Initial ortho-normal vector u_0 is prepared. If IUV=0, approximation wide-sense orthogonal vector v_0 input to array V is pre-multiplied by L^T to determine u_0. If IUV=1, u_0 input to array U is used without modification.

3. $Au=\lambda u$ is solved by NSJENS or NSJEND by using u_0 as an initial vector.

4. $v=L^{-T}u$ is calculated.

5. The NE number of λ is output to array E, the NV number of v is output to array V, and the NV number of u is output to array V.

(4) Notes

1. NGJEND is faster than NGHOUD when (ITER×NV/N)<0.4. If IUV=1 when initial ortho-normal vector u_0 is known, it becomes faster about 5%.

2. NGJENS and NGJEND are suitable for use if good approximate solutions are known, that is, to diagonalize a lot of similar real symmetric matrices.

<Example of using NICER> The program which solves generalized eigenvalue problem $Av=\lambda Bv$, 10 times, with B fixed and A varied is shown below. This example indicates the calculation procedure of the wave function by the sequential approximation method.

```
C       ITERATIVE COMPUTATION OF A*V=B*V*E    BY NICER
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION  AB(10,10),BD(10)
        DIMENSION  E(10),V(10,10)
        DIMENSION  W1(10),W2(10),W3(10),W4(10),W5(10),W6(10),W7
       *(10)
        DIMENSION  U(10,10)
        NMAX=10
        N=8
        EPS=1.E-10
C
        DO 10 I=1,N
        AB(I,I)=7.2
        BD(I)=N+1-I
        DO 10 J=1,N
        IF(J.GT.I) AB(I,J)=3.0 / (FLOAT(I-J))**2
        IF(I.GT.J) AB(I,J)=N+1-MAX(I,J)
   10 CONTINUE
        CALL NGHOUD(AB,NMAX,N,N,N,EPS,1,0,BD,E,V,ILL,W1,W2,W3,
       *W4,W5,W6,W7)
        WRITE(6,100)  ILL
  100 FORMAT(1H1 //20X,4HILL=I7)
        WRITE(6,200)  (  E(I),I=1,N  )
  200 FORMAT(1H   /  10(2X,10E12.3 /)  )
        WRITE(6,300) (  ( V(I,J),J=1,N ),I=1,N  )
  300 FORMAT(1H  / 10( 8F12.3/ )   )
C
        DO 1000 K=2,10
```

224

```
      FK=0.5*FLOAT(K-1)
      DO 20 I=1,N
      AB(I,I)=7.2 + FK
      DO 20 J=1,N
      IF(J.GT.I) AB(I,J)=(3.0+FK) / (FLOAT(I-J))**2
   20 CONTINUE
      NE=N/2
      NV=N/2
      ESHIFT=0.5*( E(NV+1)+E(N) )
      ITER=10
      CALL NGJEND(AB,NMAX,N,NE,NV,EPS,BD,0,ITER,ESHIFT,E,V,U,
     *ILL,W1,W2)
      WRITE(6,400)  K,  ITER,ILL
  400 FORMAT(1H /10X,3H K=,I2,3X,5HITER=,I5,6H  ILL=,I5)
      WRITE(6,200)  (  E(I),I=1,NE )
      WRITE(6,500)  (  ( V(I,J),J=1,NV ),I=1,N  )
  500 FORMAT(1H  / 10( 4F12.3/ )  )
 1000 CONTINUE
      STOP
      END
```

<Part of result of NICER>

```
                    ILL=      0

    0.126D+02   0.116D+02   0.109D+02   0.102D+02

      0.057       0.355      -0.040      -0.595
      0.005      -0.723      -0.464       0.838
     -0.266       0.049       1.083       0.061
      0.169       0.941      -0.398      -0.209
      0.448      -0.632      -0.617      -0.069
     -0.567      -0.450       0.562      -0.664
     -0.511       0.652      -0.420       0.837
      1.225      -0.156       0.584      -0.006

    0.952D+01   0.729D+01   0.345D+01   0.465D+00

     -0.260       0.545       0.364       0.131
      0.708      -0.216       0.083       0.067
     -0.816      -0.277       0.017       0.076
      0.828      -0.420      -0.133       0.062
     -0.978      -0.087      -0.218       0.052
      0.786       0.197      -0.247       0.039
     -0.440       0.440      -0.217       0.027
      0.295       0.204      -0.083       0.007
```

```
PACKAGE-NAME : NICER(NAGOYA ITERATIVE COMPUTATION EIGENVALUE
               ROUTINES)(VERSION-1,LEVEL-3) MODIFIED ON MARCH
               1981
REFERENCE    : Y.BEPPU AND I.NINOMIYA;QUANTUM CHEMISTRY
               PROGRAM EXCHANGE,NO.409(1980)

      K= 2   ITER=      4 ILL=    0

0.131D+02   0.117D+02   0.112D+02   0.988D+01

  0.099       0.203      -0.456      -0.444
 -0.124      -0.663       0.340       0.973
```

```
-0.202        0.399        0.808       -0.616
 0.328        0.718       -0.665        0.094
 0.281       -0.856       -0.399        0.183
-0.666       -0.245        0.281       -0.790
-0.323        0.718       -0.104        0.946
 1.184       -0.266        0.532       -0.339

      K= 3   ITER=     3  ILL=     0

0.136D+02    0.120D+02    0.115D+02    0.935D+01

 0.093       -0.151       -0.511       -0.297
-0.172       -0.235        0.773        0.897
-0.119        0.740        0.311       -0.800
 0.383        0.190       -0.922        0.192
 0.151       -0.952        0.158        0.173
-0.697       -0.013        0.279       -0.742
-0.199        0.647       -0.371        0.966
 1.156       -0.118        0.592       -0.404

      K= 4   ITER=     6  ILL=     0

0.143D+02    0.126D+02    0.117D+02    0.918D+01

 0.067        0.322        0.395        0.126
-0.180       -0.077       -0.829        0.498
-0.041       -0.770        0.024       -0.884
 0.392        0.137        0.923        0.164
 0.046        0.866       -0.425       -0.054
-0.701       -0.128       -0.245       -0.393
-0.108       -0.555        0.471        0.969
 1.135        0.074       -0.620       -0.288
```

(1987. 06. 16) (1987. 08. 07)

NSHOUS/D (Eigenvalue analysis of $Av=\lambda v$ type by Householder's bisection QR and inverse iteration method)

Nicer for Standard Eigenvalue-Problem by Householder Method

Programmed by	Yoshitaka Beppu and Ichizo Ninomiya; December 1981
Format	Subroutine language; FORTRAN Size; 271 and 272 lines respectively

(1) Outline

NSHOUS and NSHOUD solve standard eigenvalue problems concerning real symmetric dense matrix A by using the semi-direct method.

(2) Directions

CALL NSHOUS/D(A, NMAX, N, NB, NV, EPS, IORD, E, V, ILL, W1, W2, W3, W4, W5, W6, W7)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	$A_{ij}(i \leq j)$ is input to the upper right half including diagonal elements. The lower left half is preserved although the upper right half changes.
NMAX	Integer type.	Input	Adjustable dimensions of A and V. $N \leq NMAX$
N	Integer type	Input	Order of A. $2 \leq N$
NB	Integer type	Input	Number of eigenvalues to be obtained. $0 < NB \leq N$
NV	Integer type	Input	Number of eigenvectors to be obtained. $0 \leq NV \leq NB \leq N$

Argument	Type and kind (*1)	Attribute	Content
EPS	Real type	Input	Tolerance for convergence test. The default value is 10^{-6} (NSHOUS) or 10^{-10} (NSHOUD).
IORD	Integer type	Input	The output order of eigenvalues is specified. When IORD>0, they are output in algebraically descending order. When IORD<0, they are output in algebraically ascending order.
E	Real type One-dimensional array	Output	The Ith eigenvalue is output to E(I). When IORD is positive, E(1)>E(2)>····· >E(NE). When IORD is negative, E(1)<E(2)<·····<E(NE).
V	Real type Two-dimensional array	Output	The eigenvector corresponding to E(I) is normalized as $y^T y=1$ and output to column I.
ILL	Integer type	Output	ILL=0: Normal termination ILL=300: Argument error.
W1~W7	Real type One-dimensional array	Work area	The size must be N or more.

(3) Calculation method

Matrix A is transformed into tridiagonal matrix T by Householder conversion. If (NE/N)<0.12, eigenvalues by the number specified by NE are determined by the bisection method. If (NE/N)≧0.12, the N number of eigenvalues are determined by the square root-free QR method. Then, eigenvectors by the number specified by NV are determined by the inverse iteration method.

(4) Notes

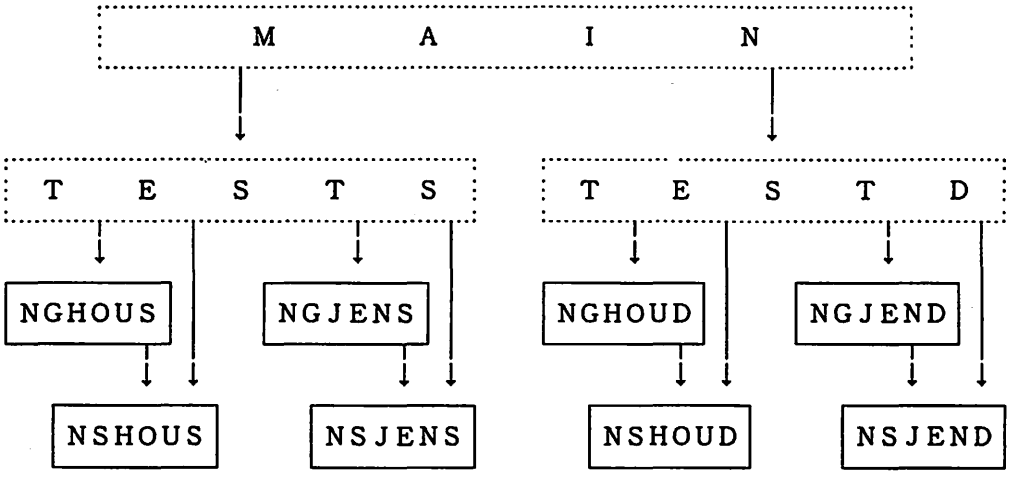
1. This routine is useful when an approximate solution is unknown.

2. As shown in the figure below, NSHOUS/D together with NGHOUS/D, NSJENS/D, and NGJENS/D form the fast eigenvalue routine package NICER (Nagoya Iterative Computation Eigenvalue Routines).

3. The user of NICER should quote the documents listed in the bibliography below.

<Configuration of NICER>

The element enclosed by broken lines is used for calling.



Bibliography

1) Y. Beppu and I. Ninomiya; "Manual of NICER", Quantum Chemistry Program Exchange (Indiana University), No. 409 (1980)

2) Y. Beppu and I. Ninomiya; "NICER-Fast Eigenvalue Routines", Computer Physics Communications, Vol. 23, pp. 123-126 (1981)

3) Y. Beppu and I. Ninomiya; "HQRII-A Fast Diagonalization Subroutine", Computers and Chemistry, Vol. 6, No. 2, pp. 87-91 (1982)

(1987. 06. 16) (1987. 08. 08)

NSJENS/D (Analysis of Av=λv type eigenvalue by Jennings method)

Nicer for Standard Eigenvalue-Problem by Jennings Method

Programmed by	Yoshitaka Beppu and Ichizo Ninomiya; December 1981
Format	Subroutine language; FORTRAN Size; 334 and 335 lines respectively

(1) Outline

NSJENS and NSJEND solve standard eigenvalue problems concerning real symmetric dense matrix A by the simultaneous iteration method.

(2) Directions

CALL NSJENS/D(A, NMAX, N, NE, NV, EPS, ITER, ESHIFT, E, V, ILL, W1, W2, U)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	$A_{ij}(i \leqq j)$ is input to the upper right half including diagonal elements. The upper right half is referred to but not changed. The lower left half is neither referred to nor changed.
NMAX	Integer type	Input	Adjustable dimensions of A, V, and U. N≦NMAX
N	Integer type	Input	Order of A. 2≦N
NE	Integer type	Input	Number of eigenvalues to be determined. They are counted in absolutely descending order. 0<NE<N
NV	Integer type	Input	Number of eigenvectors to be determined. 0<NE≦NV<N

Argument	Type and kind (*1)	Attribute	Content
EPS	Real type	Input	Tolerance for convergence test. The default value is 10^{-6} (NSJBNS) or 10^{-10} (NSJEND).
ITER	Integer type	Input/output	The upper limit of the number of Jennings iterations (standard value ranges from 1 to 10) is input. The number of actual iterations is output.
ESHIFT	Real type	Input	Quantity of origin shift σ. In these routines, the simultaneous iteration method is applied not to A but to $A'=A-\sigma \cdot I$. Therefore, eigenvalues which are close to σ are rapidly diminished and the other eigenvalues are rapidly enhanced. The standard value is an approximate value of 0.5* (E(NV+1)+E(N)).
E	Real type One-dimensional array	Input/output	The approximate value of the eigenvalue whose absolute value is the Ith largest of all is input to E(I). The eigenvalue whose absolute value is the Ith largest of all is output to E(I). |E(1)|>|E(2)|>·····>|E(NE)|
V	Real type Two-dimensional array	Input/output	Approximate ortho-normal vectors by the number specified by NV is input. The eigenvector which corresponds to E(I) is normalized to $v^T v=1$ and output.
ILL	Integer type	Output	ILL=0: Normal termination ILL=200: Conversion does not occur because of poor precision of approximate vectors. ILL=300: The argument is abnormal.
W1~W2	Real type One-dimensional array	Work area	The size must be N or more.

Argument	Type and kind (*1)	Attribute	Content
U	Real type Two-dimensional array	Work area	The row size must be NMAX or more and the column size must be NV or more.

(3) Calculation method

An approximate solution is iteratively improved according to the following procedure:

1. Approximate eigenvector matrix V_0 is prepared.

2. V_0 is pre-multiplied by $A'=A-\sigma I$ to generate X. At this time, the absolutely dominant eigenpairs of A' are enhanced by the principle of the power method. $X=A'V_0=(A-\sigma I)V_0$

3. $G=V_0^TX$ is generated.

4. Eigenvector matrix X and eigenvalue matrix E'_0 of G are determined. $W^TGW=E'_0$

5. $Y=XW$ is generated.

6. $S=Y^TY$ is generated.

7. S is Cholesky-decomposed. $S=Z^TZ$

8. $V'_0=YZ^{-1}$ is generated. V'_0 is nearer V than V_0.

9. If permissible accuracy is reached, the calculation is finished with $V=V'_0$, $E=E'_0$. Conversely, if convergence does not occur, processing returns to 2. with $V_0=V'_0$. Here, I is a unit matrix with N rows and N columns, V_0, X, Y, V'_0 are matrices each with N rows and NV columns, and G, W, S, Z are matrices each with NV rows and NV columns.

(4) Notes

1. NSJEND is faster than NSHOUD when (ITER×NV/N)<0.5.

2. Like JENNFS and JENNFD of NUMPAC, these routines are also suitable for use when good approximate solutions are known.

3. These routines are also components of NICER.

Bibliography

1) A. Jennings; "Matrix Computation for Engineers and Scientists", John Wiley and Sons, London (1977)

2) B. Parlett; "The Symmetric Eigenvalue Problem", Prentice-Hall, New Jersey (1980)

3) Yoshitaka Beppu and Ichizo Ninomiya; "Fast Eigenvalue Routine Package NICER," Kyoto University Data Processing Center Report, Vol. 13, No. 5, pp. 378-386 (1980)

(1987. 06. 16) (1987. 08. 08)

RHBSVS/D (Eigenvalue analysis of symmetric band matrices by Lutishauser-Bisection method)

Eigenvalue Analysis for Symmetric Band Matrices by Lutishauser-Bisection Method

Programmed by	Ichizo Ninomiya; Revised in April 1977; April 1981
Format	Subroutine language; FORTRAN Size; 250 lines each

(1) Outline

RHBSVS or RHBSVD reduces a symmetric band matrix into a tridiagonal from using the Lutishauser-Schwarz method, and applies the bisection and inverse iteration methods to it to perform eigenvalue analysis.

(2) Directions

CALL RHBSVS/D(A, KA, N, NB, E, NE, V, KV, NV, VW, EPS, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	The lower left half including the diagonal of the symmetric band matrix is turned to a rectangle as shown in the figure. That is, elements I and J of the matrix are put in A(I-J+1, J).
KA	Integer type	Input	Adjustable dimension of A (value of the first subscript in array declaration). KA≧NB
N	Integer type	Input	Order of A. N≧3
NB	Integer type	Input	Half band width of A. NB≧2
E	Real type One-dimensional array	Output	Eigenvalues are output in the order of size. If NE>0, in descending order, and in ascending order otherwise.
NE	Integer type	Input	The number of eigenvalues to be determined is indicated by the absolute value. When NE>0 (NE<0), they are counted in algebraically descending (ascending) order from the maximum value (minimum value). NE≠0

Argument	Type and kind (*1)	Attribute	Content
V	Real type Two-dimensional array	Output	The eigenvector for the eigenvalue E(I) is normalized to length 1 and output to column I.
KV	Integer type	Input	Adjustable dimensions of V and VW. KV≧N
NV	Integer type	Input	The number of eigenvectors to be determined is indicated by the absolute value. They are counted starting with an eigenvalue from either side in the order determined by NE. 0≦|NV|≦|NE|
VW	Real type Two-dimensional array	Work area	Two-dimensional array of size N × N. This argument is not needed if no eigenvectors are calculated (NV = 0).
EPS	Real type	Input	Tolerance for convergence test by bisection method. ‖T‖·EPS, where T is a tridiagonal matrix, is used for test. EPS>0
W	Real type One-dimensional array	Work area	One-dimensional array of size 6N or more.
ILL	Integer type	Output	ILL=0: Normal termination ILL=30000: The input argument violated the limit.

*1 For double precision subroutines, real types are all changed to double precision real types.

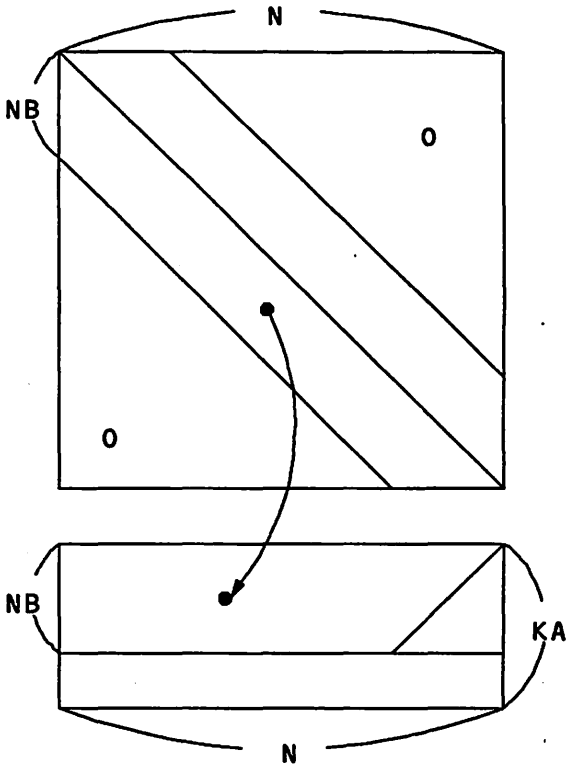
(3) Calculation method

Symmetric band matrix A is transformed into the tridiagonal matrix $T=R^TAR$ by Lutishauser-Schwarz orthogonal transformation R. The eigenvalue problem $Tu=\lambda u$ for T is solved by the bisection and inverse iteration methods. The eigenvector of A is determined as $v=Ru$ from the eigenvector u of T.

The Lutishauser-Schwarz method has the advantage that the calculation can be performed within the interior of a band matrix.

On the other hand, however, it has the disadvantage that when band width expands, quantity of calculation will increase. Furthermore, to determine eigenvectors, transformation matrix R must be saved. This requires a square matrix of $N\times N$ where N is the order. From the above viewpoint, the significance of the existence of this routine is to calculate only eigenvalues of

high dimensional matrices with small band width.



(4) Notes

1. If no eigenvectors need to be calculated, any value can be assigned to V and VW as far as the condition KV≧N is satisfied.

2. If it is desired to save storage capacity when eigenvectors are calculated, A and V can be connected by an equivalence statement. This is because A and V are not used at the same time.

(1987. 06. 16) (1987. 08. 08)

RHQRVS/D (eigenvalue analysis of real symmetric band matrices by Rutishauser-QR method)

Eigenvalue Analysis for Real Symmetric Band Matrices by Rutishauser-QR Method

Programmed by	Ichizo Ninomiya; April 1977
Format	Subroutine language; FORTRAN Size; 150 and 152 lines respectively

(1) Outline

RHQRVS or RHQRVD reduces real symmetric band matrix B to a tridiagonal matrix by using the Lutishauser-Schwarz method, and applies the QR method to this to perform eigenvalue analysis.

(2) Directions

CALL RHQRVS/D(B, KB, N, NB, V, KV, E, F, EPS, IND)

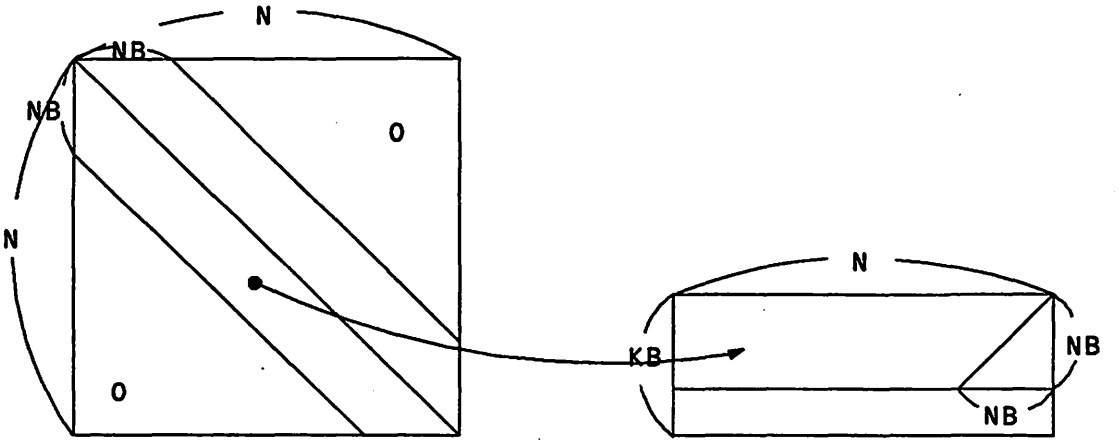
Argument	Type and kind (*1)	Attribute	Content
B	Real type Two-dimensional array	Input	The lower left half including the diagonal of the real symmetric band matrix is reduced to a rectangle shown in the figure. It is not preserved.
KB	Integer type	Input	Value of the first subscript in array declaration of B. KB≧NB
N	Integer type	Input	Order of B (number of columns). This is also the sizes of E and F. 3≦N
NB	Integer type	Input	Half band width of B (number of rows). 3≦NB≦N
V	Real type Two-dimensional array	Output	The eigenvector which corresponds to the eigenvalue (J) is normalized to length 1 and output to column J.
KV	Integer type	Input	Value of the first subscript in array declaration of A. KV≧N
E	Real type One-dimensional array	Output	Eigenvalues are arranged in algebraically descending order from the maximum one and output sequentially.

Argument	Type and kind (*1)	Attribute	Content
F	Real type One-dimensional array	Work area	One dimensional array with N elements.
EPS	Real type	Input	Tolerance for convergence test. When B is turned to tridiagonal T, this argument is used in the form of $\|\|T\|\| \cdot EPS/N$. EPS>0
IND	Integer type	Input/output	When used for input, this argument has the following meanings: IND=0: Eigenvectors are not calculated. IND≠0: All eigenvectors are calculated. When used for output, this argument has the following meanings: IND=0: The calculation ended normally. IND=30000: The limits on the input argument are violated. Note: Do not use a constant as the actual argument for this argument.

*1 For double precision subroutines, real types are all changed to double precision real types.

(3) Performance

Tridiagonalization by Householder's mirror image transformation is a very excellent method. However, it has one problem: even if a real symmetric band matrix is given, it is expanded to a full matrix during transformation. The Lutishauser-Schwarz method can solve the above problem because it reduces a band matrix to a tridiagonal without expanding it from the original position. However, it requires more quantity of calculation as band width is increased. Furthermore, to determine eigenvectors, it is necessary to calculate an orthogonal matrix which represents transformation from a band matrix to a tridiagonal matrix. This requires an additional square matrix of N × N where N is the order. From the viewpoint of the above, the significance of the existence of this routine is to calculate eigenvalues of big matrices with relatively small band width.



(4) Note

If no eigenvectors need to be determined, any value can be assigned to V as far as KV≧N is satisfied.

(1987.06.17)

SVDS/D/Q (Singular value decomposition)

Singular Value Decomposition

Programmed by	Ichizo Ninomiya; March 1979
Format	Subroutine language; FORTRAN Size; 205 lines each

(1) Outline

SVDS, SVDD, or SVDQ uses $m \times n$ orthogonal matrix U, $n \times n$ orthogonal matrix V, and $n \times n$ diagonal matrix Σ to decompose $m \times n$ matrix A $(m \geqq n \geqq 1)$ into

$$A = U\Sigma V^T$$

Where, $U^TU = V^TV = VV^T = I_n$ (n-degree unit matrix),

$$\Sigma = diag(q_1, q_2, \cdots, q_n)$$

U consists of n orthogonal eigenvectors corresponding to the firstlargest n eigenvalues of AA^T, and V is made up of the orthonormal eigenvectors of A^TA. The diagonal elements of Σ are the positive square root of the eigenvalues of A^TA and arranged such that

$$q_1 \geqq q_2 \geqq \cdots \geqq q_n \geqq 0$$

The rank of A is given by the number of q_i which are not 0.

(2) Directions

CALL SVDS/D/Q(A, KA, M, N, ISW, Q, U, KU, V, KV, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type Two-dimensional array	Input	Matrix subjected to singular value decomposition. The value is preserved unless this argument is used as a U or V storage area.
KA	Integer type	Input	Value of the first subscript in array declaration of A. KA≧M
M	Integer type	Input	Number of rows of A. M≧N

Argument	Type and kind (*1)	Attribute	Content
N	Integer type	Input	Number of columns of A. N≧1
ISW	Integer type	Input	0≦ISW≦3 ISW=0: Neither U nor V is calculated. ISW=1: Only V is calculated. ISW=2: Only U is calculated. ISW=3: Both U and V are calculated.
Q	Real type One-dimensional array	Output	Singular values are output in descending order from the largest one. One-dimensional array of size N.
U	Real type Two-dimensional array	Output	Transformation matrix U. This can be written over A. Array with M rows and N columns.
KU	Integer type	Input	Value of the first subscript in array declaration of U. KU≧M
V	Real type Two-dimensional array	Output	Transformation matrix V. This can be written over A. Array with N rows and N columns.
KV	Integer type	Input	Value of the first subscript in array declaration of V. KV≧N
W	Real type One-dimensional array	Work area	One-dimensional array of size N.
ILL	Integer type	Output	ILL=0: Normal termination ILL=20000: Singular value decomposition does not converge in 30N or more iterations. ILL=30000: The argument exceeded the limit.

*1 For double precision subroutines, real types are all changed to double precision real types.

(3) Performance

We experimented with an 8 x 5 matrix of rank 3 with singular values $\sqrt{1248}, 20, \sqrt{384}, 0, 0$ given on page 418 in the bibliography [1]. The precision of the singular value Q and transformation matrix V (the two last columns are two independent solution vectors of homogeneous linear equation $Ax=0$) obtained by SVDS was about six decimal digits.

(4) Example

The program to examine the above test is as follows.

```
      DIMENSION A(8,5),U(8,5),V(5,5),Q(5),W(5),R(5)
      M=8
      N=5
      KA=8
      KU=8
      KV=5
      ISW=3
      R(1)=SQRT(1248,)
      R(2)=20,
      R(3)=SQRT(384,)
      R(4)=0,
      R(5)=0,
      READ(5,500) ((A(I,J),J=1,N),I=1,M)
  500 FORMAT(5F4,0)
      CALL SVDS(A,KA,M,N,ISW,Q,U,KU,V,KV,W,ICON)
      WRITE(6,600) M,N,ISW,EPS,ICON,((A(I,J),J=1,N),I=1,M)
     *,(Q(J),R(J),J=1,N),((U(I,J),J=1,N),I=1,M)
     *,((V(I,J),J=1,N),I=1,N)
  600 FORMAT(1H1///10X,'M=',I2,2X,'N=',I2,2X,'ISW =',I2
     *,2X,'EPS=',1PE10,2,2X,'ICON=',I6//8(10X,5E13,5/)
     */5(10X,2E13,5/)/8(10X,5E13,5/)/5(10X,5E13,5/))
      STOP
      END
```

(5) Notes

1. Even when $M<N$, $A=U\Sigma V^T$ is also obtained if A^T is input instead of A, M is replaced by N, U is replaced by V.

2. Singular value decomposition is a very useful method for matrix A which is of ill condition or suffers a rank deficiency. But its weak point is the need for large quantity of calculation. Therefore, it is desirable to avoid calculating U and V unless they are needed.

3. U or V can be written over A. So, if A need not be retained, it is preferable to write the same data as A as U or V to save the storage capacity.

4. Because the special subroutines are prepared for generalized inverse matrices and least squares minimal norm solutions, select most suitable one for each case.

Bibliography

1) G. H. Golub, C. Reinsch; "Singular Value Decomposition and Least Squares Solutions", Numerische Mathematik, 14, pp. 403-420, (1970)

(1987. 06. 16) (1987. 08. 21)

5. Polynomial equation and nonlinear equation

BROYDS/D (Solution of systems of nonlinear equations by Broyden's method)

Solution of Systems of Nonlinear Equations by Broyden's Method

Programmed by	Ichizo Ninomiya; April 1977
Format	Subroutine language; FORTRAN Size; 59 and 71 lines respectively

(1) Outline

BROYDS and BROYDD are subroutine subprograms to solve non-linear equations $f_i(x_1,\cdots,x_n)=0$ $(i=1,2,\cdots,n)$ using the Broyden's iteration method when an initial solution vector is given.

(2) Directions

CALL BROYDS/D(X, N, H, KH, FN, LF, NF, EPS, FM, ILL)

Argument	Type and kind (*1)	Attribute	Content
X	Real type One-dimensional array	Input/output	When an initial vector is input, the solution vector is generated.
N	Integer type	Input	Number of unknowns of equation. $0<N\leqq 1000$
H	Real type Two-dimensional array	Work area	The size of KH×N is required.
KH	Integer type	Input	Value of the first subscript in array declaration of H. $KH\geqq N$
FN	Subroutine	Input	FN (X, Y) type subroutine used to calculate vector Y which consisting of values of N equations when position vector X is given. The actual argument for this argument needs an EXTERNAL declaration in the program unit which calls this routine.
LF	Integer type	Input	Upper bound of the number of function calls. LF>N+1

Argument	Type and kind (*1)	Attribute	Content
NF	Integer type	Output	Number of function calls.
EPS	Real type	Input	Convergence criterion. EPS>0
FM	Real type	Output	Square root of mean square residuals of equations.
ILL	Integer type	Output	ILL=0: Normal end. ILL=1: No convergence even when NF>LF ILL=30000: The input argument does not satisfy the requirements.

*1 For double precision subroutines, all real types are changed to double precision real types.

(3) Calculation method

Refer to bibliography [1].

(4) Example

The main part of a program to solve the Freudenstein-Roth's problem is shown below.

```
DIMENSION H(2,2),X(2)
EXTERNAL FREUDE
KH=2
N=2
LF=1000
EPS=1.E-5
X(1)=15.0
X(2)= 3.0
CALL BROYDS(X,N,H,KH,FREUDE,LF,NF,EPS,FM,ILL)
  :
END

SUBROUTINE FREUDE(X,F)
DIMENSION X(2),F(2)
F(1)=X(1)-13.+((5.-X(2))*X(2)-2.)*X(2)
F(2)=X(1)-29.+((X(2)+1.)*X(2)-14.)*X(2)
RETURN
END
```

(5) Notes

1. Because non-linear equations generally have a lot of solutions, it must be checked to see if the obtained solution is really the desired one. Good initial values must be selected to ensure

convergence to the target solution.

2. The minimization problem when the minimum value (or the maximum value) is an extremum becomes non-linear equations concerning the gradient vector. On the contrary, non-linear equations $f_i=0$ $(i=1,\cdots,n)$ become a minimization problem if $\sum f_i^2$ is considered. This must be taken into consideration to select a calculation method and program depending on the case.

3. This routine internally calls inverse matrix routine MINVS or MINVD.

Bibliography

1) C. G. Broyden; "A Class of Methods for Solving Nonlinear Simultaneous Equations", Math. Comp., Vol. 19, pp. 577-593 (1965)

(1987. 06. 16) (1987. 08. 07)

246

BROYDV/W (Solution of systems of nonlinear equations by Broyden's method - vector version-)

Solution of Systems of Nonlinear Equations by Broyden's Method -Vector Version -

Programmed by	Ichizo Ninomiya and Yasuyo Hatano; March 1985
Format	Subroutine language; FORTRAN Size; 153 and 154 lines respectively

(1) Outline

BROYDV and BROYDW are the subroutine subprograms used to solve non-linear simultaneous equations $f_i(x_1,\cdots,x_n)=0(i=1,2,\cdots,n)$ by the Broyden's iteration method when an initial value is given. BROYDV is for single precision and BROYDW is for double precision.

(2) Directions

CALL BROYDV/W(X, N, H, KH, FN, LP, NF, EPS, FM, IW, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
X	Real type One-dimensional array	Input/output	When an initial vector is input, the solution vector is output.
N	Integer type	Input	Number of unknowns of equation. $0<N\leqq 1000$
H	Real type Two-dimensional array	Work area	Size N×N is required.
KH	Integer type	Input	Value of the first subscript in array declaration of H. $KH\geqq N$

Argument	Type and kind (*1)	Attribute	Content
FN	Subroutine	Input	Subroutine in the form of FN(X, Y), which calculates vector Y consisting of N equation values when position vector X is given. The real argument for this argument needs to be declared under EXTERNAL in each program that calls this routine.
LF	Integer type	Input	Upper limit of the number of times the function subroutine is called. LF>N+1
NF	Integer type	Output	Number of times the function subroutine is called.
EPS	Real type	Input	Tolerance for convergence test. EPS>0
FM	Real type	Output	Square root of mean square residuals of equations.
IW	Integer type one-dimensional array	Work area	One-dimensional array with N elements.
W	Real type One-dimensional array	Work area	Size 4*N is required.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=1: Convergence does not occur even when NF>LF. ILL=30000: The input arguments violate the limits for them.

*1 For double precision subroutines, real types are all changed to double precision real types.

(3) Calculation method See the bibliography [1].

(4) Example

The major part of a program that solves the Freudenstein-Roth problem (solution: X(1) = 5.0, X(2) = 4.0) is shown below:

```
      DIMENSION H(2,2),X(2),IW(2),W(4,2)
      EXTERNAL FREUDE
      KH=2
      N=2
      LF=1000
      EPS=1.E-5
      X(1)=15.0
      X(2)= 3.0
      CALL BROYDV(X,N,H,KH,FREUDE,LF,NF,EPS,FM,IW,W,ILL)
        :
      END

      SUBROUTINE FREUDE(X,F)
      DIMENSION X(2),F(2)
      F(1)=X(1)-13.+((5.-X(2))*X(2)-2.)*X(2)
      F(2)=X(1)-29.+((X(2)+1.)*X(2)-14.)*X(2)
      RETURN
      END
```

(5) Notes

1. Because non-linear simultaneous equations usually have a lot of solutions, it is needed to check to see if the obtained solution is the target solution. A proper initial value must be given to make calculation converge to the target solution.

2. A minimization problem for a minimum value that is an extremal value (not a boundary value) amounts to non-linear simultaneous equations for gradient vectors. On the contrary, non-linear simultaneous equations $f_i=0(i=1,\cdots,n)$ amount to a minimization problem in terms of $\sum f_i^2$. This should be considered to select a calculation method and program most appropriate to the case in equation.

3. This routine internally calls inverse matrix routine MINVV or MINVW.

Bibliography

1) C. G. Broyden; "A Class of Methods for Solving Nonlinear Simultaneous Equations," Math. Comp., Vol. 19, pp. 577-593 (1965)

(1987. 06. 22) (1987. 08. 07) (1988. 06. 01)

FLPOWS/D (Minimization of functions by Davidon-Fletcher-Powell method)

Minimization of Functions by Davidon-Fletcher-Powell Method

Programmed by	Ichizo Ninomiya; July 1977
Format	Subroutine language; FORTRAN Size; 92 and 105 lines respectively

(1) Outline

FLPOWS and FLPOWD are subroutine subprograms which determine the minimum point of a multivariable function by the Davidon-Fletcher-Powell method when an initial value is given. In addition to a function value, a gradient vector value needs to be given.

(2) Directions

CALL FLPOWS/D(X, N, B, KB, FUNC, GRAD, LF, NF, FLB, EPS, FM, ILL)

Argument	Type and kind (*1)	Attribute	Content
X	Real type One-dimensional array	Input	When an initial vector is input, the solution vector (minimum point) is output.
N	Integer type	Input	Number of variables, or number of elements of X. $0<N\leqq1000$
B	Real type Two-dimensional array	Work area	The size of N×N is required. A unit matrix is first set and, through updating by iterations, it converges to the inverse of the Hessian matrix at the minimum point.
KB	Integer type	Input	Value of the first subscript in array declaration of B. $KB\geqq N$
FUNC	Real type Function subprogram	Input	Target function for minimization. The user prepares this as a function subprogram in the form of FUNC(X). The actual argument name must be declared in an EXTERNAL statement.
GRAD	Subroutine	Input	Subroutine used to calculate gradient vector G of function FUNC. The user prepares this as a subroutine in the form of GRAD(X, G). The actual argument name must be declared in an EXTERNAL statement.
LF	Integer type	Input	Upper limit of the number of the function calls. $LF\geqq N$

Argument	Type and kind (*1)	Attribute	Content
NF	Integer type	Output	Number of the function calls (The number of calls for FUNC and GRAD are the same.)
FLB	Real type	Input	Lower limit of minimum value of function.
EPS	Real type	Input	Tolerance for convergence test. EPS>0
FM	Real type	Output	Minimum value of function.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=1: Convergence does not occur even when NF becomes greater than LF. ILL=30000: The input argument does not satisfy the limits for them.

*1 For double precision subroutines, real types are all changed to double precision real types.

(3) Calculation method

Refer to bibliography [1),2)].

(4) Example

The major part of a program for solving the Rosenbrock problem (minimum point: X(1)=1.0, X(2)=1.0) is shown below:

```
      DIMENSION X(2),B(2,2)
      EXTERNAL ROSEN,GROSEN
      N=2
      KB=2
      LF=1000
      FLB=0.
      EPS=1.E-5
      X(1)=-1.2
      X(2)=1.0
      CALL FLPOWS(X,N,B,KB,ROSEN,GROSEN,LF,NF,FLB,EPS,FM,ILL)
        :
      END

      FUNCTION ROSEN(X)
      DIMENSION X(2)
      ROSEN=100.*(X(1)*X(1)-X(2))**2+(X(1)-1.)**2
      RETURN
      END

      SUBROUTINE GROSEN(X,G)
      DIMENSION X(2),G(2)
      G(2)=200.*(X(2)-X(1)*X(1))
      G(1)=2.*(X(1)-1.-G(2)*X(1))
      RETURN
```

END

(5) Notes

1. This routine can generally obtain a local minimum value only. A proper initial value is required to secure a true minimum value.

2. If it is impossible or very hard to calculate gradient vectors, a method which does not require calculation of gradient vectors should be used.

Bibliography

1) R. Fletcher & M. J. D. Powell; "A Rapidly Convergent Descent Method for Minimization," Computer Journal, Vol. 6, pp. 163-168 (1963)

2) P. J. Reddy, H. J. Zimmermann & Asghar Hussain; "Numerical Experiments on DFP-Method, A Powerful Function Minimization Technique," Journal of Computational & Applied Mathematics, Vol. 1, pp. 255-265 (1975)

(1987. 06. 17)

GJMNKS/D/Q (Solution of polynomial equations with real coefficients by Garside-Jarrat-Mack method)

Solution of Polynomial Equations with Real Coefficients by Garside-Jarrat-Mack Method

Programmed by	Ichizo Ninomiya; April 1977
Format	Subroutine language; FORTRAN Size; 128, 130, and 130 lines respectively

(1) Outline

GJMNKS, GJMNKD, and GJMNKQ are single, double, and quadruple precision subroutines respectively used to determine all roots of a polynomial equation with real coefficients. The Garside-Jarrat-Mack method[1] is widely acknowledged as an effective method for solution of polynomial equations with complex coefficients. These subroutines are created by Ninomiya and Kadowaki [2] by improving it as a solver of equations with real coefficients. These subroutines combine robustness of the original method and the speeds realized by using real numbers for complex calculation. They also incorporate the Cardano's and Ferrari's methods for solutions of third- and fourth-degree equations. They can thus be recommended as general-purpose polynomial equation routines.

(2) Directions

CALL GJMNKS/D/Q(A, N, X, Y, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type One-dimensional array	Input	Coefficients of a polynomial equation is input in descending order of degree. Input values are destroyed. $A(1) \neq 0$
N	Integer type	Input	Degree of polynomial equation. $N \geqq 1$

Argument	Type and kind (*1)	Attribute	Content
X	Real type One-dimensional array	Output	The real parts of roots of a polynomial equation are output. Roots are generally determined in ascending order of their absolute values and stored in reverse order like X(N), X(N-1)...
Y	Real type One-dimensional array	Output	The imaginary parts of roots of a polynomial equation are output. Order of computation and storage method are same as with X.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: N<1 or A(1)=0. ILL=K: Convergence does not occur even after 200 iterations during processing of a deflated Kth-degree equation.

*1 For double (quadruple) precision subroutines, real types are all changed to double (quadruple) precision real types.

(3) Calculation method

These subroutines inherit the advantage of the original method that the convergence rate basically does not change even for multiple roots and adjacent roots. This explains why they practically never fail to solve equations and provide as accurate roots as the condition of equation permits. In this sense, they can be said very robust.

(4) Notes

1. Polynomial equations often fall in ill conditions. Therefore, unless they are in very low degrees, it is safe to use the double precision routine GJMNKD.

2. To solve second-, third-, or fourth-degree equations, use of each specific routine is more advantageous than these routines.

Bibliography

1) G. R. Garside, P. Jarrat and C. Mack; "A New Method for Solving Polynomial Equations," Computer Journal, Vol. 11 (1968)

2) Ichizo Ninomiya and Kohei Kadowaki; "A solution of polynomial equations with real

coefficients," Preprints of the 16th meeting of Information Processing Soc. of Japan, p. 445 (1975)

(1987. 06. 17)

MINSXS/D (Minimization of Functions by Simplex Method)

Minimization of Functions by Simplex Method

Programmed by	Ichizo Ninomiya; July 1977
Form	Subroutine language; FORTRAN Size; 98 and 99 lines respectively

(1) Outline

MINSXS or MINSXD subroutine determines the minimum point of a multivariate function by the Nedler-Mead's simplex method when an initial value is given. It requires only function values. The function values just need to be continuous but need not be smooth.

(2) Directions

CALL MINSXS/D(X, N, P, KP, FUNC, LP, NF, EPS, FM, ILL)

Argument	Type and kind (*1)	Attribute	Content
X	Real type One-dimensional array	Input/output	When an initial value for a minimum point is given, the minimum point is output.
N	Integer type	Input	Number of variables, or number of elements of X. $0<N\leqq100$
P	Real type Two-dimensional array	Work area	N rows and N+1 columns. The coordinates of the N + 1 points which form a simplex are entered in these individual columns. The initial simplex is made of initial value X and N points in which individual coordinate elements of X are increased by 10% (increased by 0.1 if the element is 0). Instead of this, however, the user can prepare it before calling the routine. The option can be specified by argument NF.
KP	Integer type	Input	Value of the first subscript in array declaration of P. $KP\geqq N$
FUNC	Real type Function subprogram	Input	Target function for minimization. The user prepares a function subprogram in the form of FUNC(X) as the actual argument for it. This function name must be declared in an EXTERNAL statement.
LP	Integer type	Input	Upper limit of the number of evaluations of function. LP>N

Argument	Type and kind (*1)	Attribute	Content
NF	Integer type	Input/output	Input: NF≧0 means that formation of the initial simplex is left to the routine, and NF<0 means that the initial simplex is prepared by the user. Output: The number of evaluations of the function is output. Because this argument is used for both input and output, do not specify a constant as the actual argument.
EPS	Real type	Input	Tolerance for convergence test. EPS>0
FM	Real type	Output	Minimum value of function.
ILL	Integer type	Output	ILL=0: Normal termination ILL=1: Convergence does not occur even if NF becomes greater than LF. ILL=30000: The input argument does not satisfy the restrictive conditions.

*1 For double precision subroutines, real types are all changed to double precision real types.

(3) Method of calculation Refer to bibliography [1].

(4) Example

The major part of a program for solving the Rosenbrock problem (minimum point: X(1) = 1.0, X(2) = 1.0) is shown below:

```
      DIMENSION P(2,3),X(2)
      EXTERNAL ROSEN
      N=2
      KP=2
      NF=1
      LF=1000
      EPS=1.E-5
      X(1)=-1.2
      X(2)=1.0
      CALL MINSXS(X,N,P,KP,ROSEN,LF,NF,EPS,FM,ILL)
        :
      END

      FUNCTION ROSEN(X)
      DIMENSION X(2)
      ROSEN=100.*(X(1)*X(1)-X(2))**2+(1.-X(1))**2
      RETURN
      END
```

(5) Notes

1. This routine can generally obtain local minimum values only. A proper initial value is required to secure a true minimum value.

2. Because of slow convergence, this routine is not suitable for problems of higher dimension.

3. The function name as the actual argument must be declared in an EXTERNAL statement in each program which calls this routine.

4. For a smooth function, the calculation of whose gradient is easy, it is more advantageous to use subroutine FLPOWS using the DFP method than to use this routine.

5. When this routine is used for solving non-linear simultaneous equations $f_1=0, f_2=0, \cdots, f_n=0$ as a minimization problem, $F=\sum |f_i|$ is preferable rather than $F=\sum f_i^2$.

Bibliography

1) J. A. Nedler & R. Mead; "A Simplex Method for Function Minimization", Computer Journal, Vol. 7, pp. 308-312 (1965)

(1987. 06. 17)

NOLEQS/D/Q (Solution of Nonlinear Equations)

Solution of Nonlinear Equations

Programmed by	Ichizo Ninomiya, March 1983
Format	Subroutine language: FORTRAN; size: 55, 56, and 56 lines respectively

(1) Outline

If an interval of existence is given, a root of a given nonlinear equation in the interval is obtained.

(2) Directions

CALL NOLEQS/D/Q(A, B, FUN, EPS, NMAX, X, FX, N, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type	Input	Left end of an interval of existence.
B	Real type	Input	Right end of an interval of existence.
FUN	Real type function subprogram	Input	A function program for computing f(x) if the equation to be solved is f(x)=0. The user must prepare it as a function subprogram.
EPS	Real type	Input	Precision criterion for root.
NMAX	Integer type	Input	Upper limit of number of evaluations of function FUN. NMAX≧3
X	Real type	Output	Starting approximation for the root.

Argument	Type and kind (*1)	Attribute	Content
FX	Real type	Output	Value of f(x) for X.
N	Integer type	Output	Number of evaluations of function FUN.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=20000: When convergence is not attained even if the function evaluation count reached NMAX. ILL=30000: When no root exists in the interval (A, B), or NMAX<3.

*1 For double (quadruple) precision subroutines, all real types are changed to double (quadruple) precision real types.

(3) Calculation method

Refer to 1) in Bibliography.

(4) Example of use

This program is used to calculate the root in $(0, \pi)$ of the equation f(x)=cosx-x=0.

```
C       TEST FOR NOLEQS
        EXTERNAL FUN
        EPS=1.E-5
        NMAX=100
        A=0.0
        B=1.5708
        CALL NOLEQS(A,B,FUN,EPS,NMAX,X,FX,N,ILL)
        WRITE(6,600) A,B,EPS,X,FX,N,ILL
  600   FORMAT(1H ,2E13.5,E11.3,E13.5,E11.3,2I6)
        STOP
        END
C       FUNCTION SUBPROGRAM
        FUNCTION FUN(X)
        FUN=COS(X)-X
        RETURN
        END
```

(5) Notes

1. Because the calculation method of this routine is based on the bisection method, convergence

is assured.

2. Because the first or second inverse interpolation is used as required, convergence is fast.

3. The function f(x) must be continuous, but need not be smooth.

4. It is more advantageous for algebraic equations to use the special-purpose subroutine GJMNKS/D.

Bibliography

1) D. B. Popovski; "A Note on King's Method for Finding a Bracketed Root", Computing Vol. 29, pp. 355-359 (1982)

(1987. 06. 22) (1987. 08. 07) (1987. 08. 08)

NOLLS1 (Subroutine for non-linear least squares by quasi Newton method)

Subroutine for Nonlinear Least Squares by a Quasi-Newton Method

Programmed by	Kunio Tanabe and Sumie Ueda; March 1981
Format	Subroutine language; FORTRAN Size; 772 lines

(1) Outline

NOLLS1 obtains $x_i, i=1,\cdots,n$, which minimizes

$$\sum_{j=1}^{m} f_j^2(x_1 \cdots x_n)$$

for the function $f_j(x_1 \cdots x_n), j=1,\cdots,m$, which is nonlinear about the n number of variables $x_i, i=1,\cdots,n$.

It is specially effective for problems involving a high degree of nonlinearlity.

The user is only required to prepare a subroutine (MODELF) to calculate the value of $f_j(x_1,\cdots,x_n), j=1,\cdots,m$. To obtain more accurate results, however, the user is also requested to prepare another subroutine MODELD which calculates the first order derivative $\partial f_j/\partial x_i$ for x_i of f_j.

(2) Directions

CALL NOLLS1(MAXM, MAXN, M, N, X, ITMAX, NFEMAX, FTOL, XTOL, LDERIV, NPRINT, FF2, F, DF, ITER,
NFE, NDE, INFORM, XO, DX, FO, DFO, H, SL, D, S, Y, R, W1, W2, W3, W4)

Argument	Type and kind (*1)	Attribute	Content
MAXM	Integer type	Input	Adjustable dimension of DF (value of the first subscript in array declaration). MAXM≧N
MAXN	Integer type	Input	Adjustable dimension of H. MAXN≧N

Argument	Type and kind (*1)	Attribute	Content
M	Integer type	Input	Number of nonlinear functions $f_j(x)$, m
N	Integer type	Input	Number of unknown parameters x_i, n
X	Real type One-dimensional array	Input/output	When an initial value of unknown parameter x_i is put, the final value is generated. $(i=1,2,\cdots,n)$
ITMAX	Integer type	Input	Upper bound of the number of iterations.
NFEMAX	Integer type	Input	Upper bound of the number of times function evaluation can be done.
FTOL	Real type	Input	Convergence criterion concerning function value. If all values of $f_j(x)$ become FTOL or less, iteration ends. $0 \leqq$ FTOL
XTOL	Real type	Input	Convergence criterion concerning unknown parameter x_j. $0 \leqq$ XTOL
LDERIV	Integer type	Input	Specify whether to prepare subroutine MODELD which gives the first order derivative for x_i of f_j. 1: MODELD is used. 0: MODELD is not used. Even if LDERIV = 0, dummy subroutine MODELD must be prepared.
NPRINT	Integer type	Input	Specify what is to be printed by each iterative calculation. 0: Nothing is printed. 1: Sum of squares and x_i are printed. 2: Sum of square and x_i, f_j are printed. 3: Sum of squares and x_i, f_j, $\partial f_j/x_i$ are printed.
FF2	Real type	Output	Value of sum of squares.
F	Real type One-dimensional array	Output	Value of residual f_j.
DF	Real type Two-dimensional array	Output	Value of first order derivative $\partial f_j/x_i$. DF(MAXM, N).
ITER	Integer type	Output	Number of actual iterations.
NFE	Integer type	Output	Actual number of function evaluations.
NDE	Integer type	Output	Number of evaluations of actual first order derivative (calling frequency of MODELD).

Argument	Type and kind (*1)	Attribute	Content
INFORM	Integer type	Output	Information on the convergence state is generated. When INFORM = 1, the condition in (2) in item (3), "Calculation method", is satisfied. Otherwise, INFORM = 0.
XO, DX	Real type One-dimensional array	Work area	XO(N), DX(N)
FO	Real type One-dimensional array	Work area	FO(M)
DFO	Real type Two-dimensional array	Work area	DFO(MAXM, N)
H, SL	Real type Two-dimensional array	Work area	H(MAXN, N), SL(MAXN, N)
D, S, Y, R, W1, W2	Real type One-dimensional array	Work area	D(N), S(N), Y(N), R(N), W1(N), W2(N)
W3, W4	Real type One-dimensional array	Work area	W3(M), W4(M)

(3) Calculation method

A local minimum value is determined based on the Biggs' quasi-Newton iteration method. The convergence test is controlled by the values of arguments XTOL and FTOL. Conversion ends when one of the following conditions are met:

(1) $|f_j(x)| < max(FTOL, \varepsilon), j=1,2,\cdots,m$

(2) $|f(x^+), \partial_j f(x^+))| \leq \alpha_1 \| f(x^+) \|_2 \| \partial_j f(x^+) \|_2 (j=1,2,\cdots,m)$

And, $\| x^+ - x \|_\infty \leq \alpha_2 max(\| x^+ \|_\infty, 1.0)$

where

$$\alpha_1 = \begin{cases} 10^{-3} & \cdots\cdots(1) \\ 10^{-4/4} & \cdots\cdots(2) \end{cases}$$

$\alpha_2=max(XTOL,\beta)$

$$\beta=\begin{cases}16\varepsilon^{\frac{1}{2}} & \cdots\cdots(1)\\ 32\varepsilon & \cdots\cdots(2)\end{cases}$$

$\partial_i f_j(x)=(\partial f_j/\partial x_i)$

(1) is the treatment when f is given, and (2) is the treatment when f and $f\partial$ are given.

ε is a constant which depends on the machine, and x^+ and x are the values of two continuing xs in the iterative calculation.

(3) The number of iterations exceeds the upper bound value.

(4) The number of operations of function values exceeds the upper bound value.

(5) The value of x does not show a remarkable change.

(4) Example

```
      DIMENSION X(20),F(100),DF(100,20),X0(20),DX(20),F0(100)
      DIMENSION DF0(100,20),H(20,20),SL(20,20),D(20),S(20)
      DIMENSION Y(20),R(20),W1(20),W2(20),W3(100),W4(100)
      MAXM=100
      MAXN=20
      M=2
      N=2
      X(1)=-1.2
      X(2)=1.0
      ITMAX=100
      FTOL=1.0E-5
      XTOL=1.0E-5
      NFEMAX=5000
      LDERIV=1
      NPRINT=3
      WRITE(6,6000) MAXM,MAXN,M,N,ITMAX,NFEMAX,FTOL,XTOL,
     1              LDERIV,NPRINT
      WRITE(6,6100) (X(J),J=1,N)
      CALL NOLLS1(
     -MAXM,MAXN,M,N,X,ITMAX,NFEMAX,FTOL,XTOL,LDERIV,NPRINT
     -,FF,F,DF,ITER,NFE,NDE,INFORM,X0,DX
     -,F0,DF0,H,SL,D,S,Y,R,W1,W2,W3,W4)
      WRITE(6,6200) ITER,NFE,NDE
 6000 FORMAT(1H0,4X,'INITIAL VALUES',/1H ,10X,'MAXM=',I4
     -,'   MAXN=',I4,'   M=',I2,'   N=',I2,'   ITMAX=',I4
     - '   NFEMAX=',I5/1H ,10X,'FTOL=',1PE16.7,'   XTOL='
     -,E16.7/1H ,10X,'LDERIV=',I2,'   NPRINT=',I2)
 6100 FORMAT(1H ,10X,'X=',1P5E16.7/(1H ,10X,5E16.7))
 6200 FORMAT(1H0,10X'ITERATION',I6/1H ,10X,'MODELF-CALL',I4
     -/1H ,10X,'MODELD-CALL',I4)
```

```
      STOP
      END
      SUBROUTINE MODELF(M,N,X,F)
      DIMENSION X(N),F(M)
      F(1)=10.0*(X(1)*X(1)-X(2))
      F(2)=1.0-X(1)
      RETURN
      END
      SUBROUTINE MODELD(MAXM,M,N,X,DF)
      DIMENSION X(N),DF(MAXM,N)
      DF(1,1)=20.0*X(1)
      DF(1,2)=-10.0
      DF(2,1)=-1.0
      DF(2,2)=0.0
      RETURN
      END
```

Output result

```
 INITIAL VALUES
      MAXM= 100 MAXN= 20 M= 2 N= 2 ITMAX= 100 NFEMAX= 5000
      FTOL=   9.9999997E-06  XTOL=   9.9999997E-06
      LDERIV= 1   NPRINT= 3
      X=  -1.1999998E+00   1.0000000E+00
 0       THE SUM OF SQUARES=   2.4199875E+01
      X=  -1.1999998E+00   1.0000000E+00
      F=   4.3999863E+00   2.1999998E+00
     DF=  -2.3999985E+01  -1.0000000E+01
          -1.0000000E+00   0.0
 1       THE SUM OF SQUARES=   2.1258163E+01
      X=  -1.0189848E+00   6.2381876E-01
      F=   4.1451035E+00   2.0189848E+00
     DF=  -2.3999985E+01  -1.0000000E+01
          -1.0000000E+00   0.0
 2       THE SUM OF SQUARES=   3.9795551E+00
      X=  -9.9474800E-01   9.9184918E-01
      F=  -2.3256540E-02   1.9947472E+00
                    :
                    :
       THE SUM OF SQUARES=   1.3669265E-08
      X=   9.9991751E-01   9.9982673E-01
      F=   8.2850456E-05   8.2492828E-05
     DF=   1.9940781E+01  -1.0000000E+01
          -1.0000000E+00   0.0
21       THE SUM OF SQUARES=   3.6948222E-13
      X=   9.9999988E-01   9.9999970E-01
      F=   5.9604645E-07   1.1920929E-07
     DF=   1.9998337E+01  -1.0000000E+01
          -1.0000000E+00   0.0
      ********************************* FINISHED******
      ITERATION    21
      MODELF-CALL  38
      MODELD-CALL  21
```

Bibliography

1) Bartholomew-Biggs, M.C.; "The estimation of Hessian matrix in nonlinear least squares problems with non-zero residuals", Mathematical Programming 12, pp.67-80 (1977)

2) Kunio Tanabe; "Algorithm of nonlinear least squares method," Applied statistics, Vol.9, No.3,

pp. 119-140 (1981)

3) Kunio Tanabe and Sumie Ueda; "NOLLS1, A Fortran subroutine for nonlinear least squares by a quasi-Newton method", Computer Science Monographs, The Institute of Statistical Mathematics (1981).

(1987. 06. 17)

POLEQC/B/Z (Solution of a Polynomial Equation with Complex Coefficients)

Solution of a Polynomial Equation with Complex Coefficients

Programmed by	Tsuyako Miyakoda and Tatsuo Torii, and revised by Ichizo Ninomiya, June 1984
Format	Subroutine language: FORTRAN; size: 172 lines

(1) Outline

POLEQC/B/Z obtains all the roots of an algebraic equation with complex coefficients using the evaluation of the degree-reduced type.

(2) Directions

CALL POLEQC/B/Z(AA, NN, Z, ERR, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
AA	Complex type One-dimensional array	Input	The coefficients of algebraic equations are sequentially input in descending order of degree. AA(1) $\neq 0$ and size NN+1.
NN	Integer type	Input	Degree of algebraic equations. NN$\geqq$1
Z	Complex type One-dimensional array	Output	The roots of algebraic equations are output in the reverse of the searching order.

Argument	Type and kind (*1)	Attribute	Content
ERR	Real type One-dimensional array	Output	Error evaluation for each obtained solution.
W	Complex type One-dimensional array	Work area	The size is 3×(NN+1).
ILL	Integer type	Output	ILL=0: Normal termination. LL=30000: N<1 or AA(1)=0. ILL=K: The convergence may not occur even if the calculation is iterated 50 (100, 200) times when a reduced K-degree equation is processed.

1* For double (quadruple) precision subroutines, all complex types are assumed to be double (quadruple) precision complex types.

(3) Calculation method

Refer to paper (2). The method of obtaining approximate roots is fundamentally the same as in paper (1). And the convergence is improved by distributing the roots of the reduced polynomial evenly inside and outside of a unit circle each time. We obtain the root existing inside of the circle setting the initial value of the iterate as Z=0. When the order of coefficients is reversed and the root of a polynomial whose order is reversed from the original one is obtained, a minus sign is added to ERR.

(4) Example of use

```
C       TEST FOR POLEQB
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*4 XR,XI
```

```
      COMPLEX*16 A(50),B(50),Z(50),X(50),T,WZ(200)
      DIMENSION ERR(50),TER(50)
      DO 60 N=1,10
      DO 10 I=1,N
      A(I+1)=0.D0
      XR=1.0-RANDOM(0)*2.0
      XI=1.0-RANDOM(0)*2.0
   10 X(I)=CMPLX(XR,XI)
      A(1)=1.D0
      DO 50 I=1,N
      DO 30 J=2,I+1
   30 B(J)=A(J)-A(J-1)*X(I)
      DO 40 J=2,I+1
   40 A(J)=B(J)
   50 CONTINUE
      WRITE(6,1010)(I,X(I),I=1,N)
      CALL POLEQB(A,N,Z,WZ,ERR,ILL)
      DO 66 I=1,N-1
      K=I
      DO 70 J=2,N
      IF(CDABS(Z(J)-X(I)).LT.CDABS(Z(K)-X(I))) K=J
   70 CONTINUE
      T=Z(K)
      Z(K)=Z(I)
      Z(I)=T
      SS=ERR(K)
      ERR(K)=ERR(I)
      ERR(I)=SS
   66 CONTINUE
      WRITE(6,1030) ILL
      DO 55 I=1,N
      TER(I)=CDABS(Z(I)-X(I))
   55 WRITE(6,1040)I,Z(I),TER(I),ERR(I)
   60 CONTINUE
 1010 FORMAT(//21X,11HEXACT ROOTS//(I5,2D23.15))
 1030 FORMAT(/25X,5HROOTS,28X,3HTER,8X,3HEST,5H ILL=,I4/)
 1040 FORMAT(1H ,I4,2D23.15,2X,2D11.3)
      END
```

(5) Note

The obtained roots are stored in the reverse order. The error estimation of each root is for the degree-reduced polynominals. Therefore, the evaluation becomes rough gradually. By the by degree reduction, a cubic polynomial is finally obtained. The cubic equation is solved directly, so 0 is input to the error estimation for these 3 roots.

Bibliography

1) Tatsuo Torii and Tsuyako Miyakoda: A Root-finding Method for a Polynomial based upon the Cubic Hermitian Interpolation, Information Processing, Vol. 14, No. 4, pp. 253-259 (1973).

2) Tatsuo Torii and Tsuyako Miyakoda: A Root-finding Algorithm for a Complex Polynomial Based on the Taylor Expansion of Third Order, Information Processing, Vol. 15, No. 8, pp. 644-646 (1974).

(1987.06.22)

270

POLESB/C (Solution of Polynomial Equation with Complex Coefficients by the Model of Electrostatic Field)

Solution of Polynomial Equation with Complex Coefficients by the Model of Electrostatic field

Programmed by	Tetsuya Sakurai, Tatsuo Torii, and Hiroshi Sugiura: September 1986
Format	Subroutine language: FORTRAN; size: 255 lines

(1) Outline

POLESB/C is a single or double precision subroutine for obtaining all the roots of polynomial equations with complex coefficients. Even if the roots include multiple and adjacent roots, they can be obtained in about the same calculation time as for single roots.

(2) Directions

CALL POLESB(A, N, Z, W, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Complex type One-dimensional array	Input	The coefficients of polynomial equations should be entered sequentially starting from the highest order coefficient. Not retained. $A(1) \neq 0$ and size N+1.
N	Integer type	Input	Order of polynomial equations. $N \geq 1$.
Z	Complex type One-dimensional array	Output	The roots of polynomial equations are output.
W	Integer type one-dimensional array	Work area	The size is $3 \times (N+1)$.

Argument	Type and kind (*1)	Attribute	Content
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: N<1 or A(1)=0. ILL=K: If no convergence occurs even if the routine is iterated 50 times while an reduced k-th order equation is processed.

*1 All real and complex types should be of a double precision.

(3) Calculation method

This method solves the equation $f(z)=0$ by approximating $f'(z)/f(x)$ using the rational expression obtained from the electrostatic field model. It has a quaternary convergence characteristic that is independent of the multiplicity of roots, and solves the quadratic equation for each iteration.

(4) Example of use

This is an example of solving $f(z)=z^5-iz^4-3z^3-3iz^2+4z-10i$.

```
*TEST FOR POLESB
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 A(6),Z(5)
      REAL*8 W(18)
*
      N=5
      A(1)=(1.D0,0.D0)
      A(2)=(0.D0,-1.D0)
      A(3)=(-3.D0,0.D0)
      A(4)=(0.D0,-3.D0)
      A(5)=(4.D0,0.D0)
      A(6)=(0.D0,-10.D0)
*
      CALL POLESB(A,N,Z,W,ILL)
*
      WRITE (6,1000) (I,Z(I),I=1,N)
 1000 FORMAT(' ',I10,2F25.15)
      STOP
      END

FORTRAN 77 COMPILER ENTERED
END OF COMPILATION
         1         0.000000000000000         1.000000000000000
         2         2.000000000000000         1.000000000000000
         3        -2.000000000000000         1.000000000000000
         4         1.000000000000000        -1.000000000000000
         5        -1.000000000000000        -1.000000000000000
END OF GO,SEVERITY CODE=00
```

(5) Note

The obtained roots are stored in the reverse order.

Bibliography

1) Tetsuya Sakurai, Tatsuo Torii, and Hiroshi Sugiura; Solution of Polynomial Equations by Electrostatic Field Interpretation, Proceedings of Symposium of 33-rd Information Processing Society of Japan, pp. 1849-1850, 1986

(1987. 07. 28)

QUADRC/B/Z, CUBICC/B/Z, and QUARTC/B/Z (Solution of Low Order Polynomial Equations with Complex Coefficients)

Solution of Low Order Polynomial Equations with Complex Coefficients

Programmed by	Tsuyako Miyakoda and Tatsuo Torii, and revised by Ichizo Ninomiya, June 1984
Format	Subroutine language: FORTRAN; size: 22, 63, and 46 lines respectively

(1) Outline

QUADRC(B, Z), CUBICC(B, Z), and QUARTC(B, Z) are the single (double or quadruple) precision subroutine for calculating all the roots of quadratic, cubic, and quartic polynomial equations with complex coefficients.

(2) Directions

CALL QUADRC/B/Z(C, Z, ILL)

CALL CUBICC/B/Z(C, Z, ILL)

CALL QUARTC/B/Z(C, Z, ILL)

Argument	Type and kind (*1)	Attribute	Content
C	Complex type One-dimensional array	Input	Coefficient of polynomial equations. Coefficients should be input in descending order from the highest.

Argument	Type and kind (*1)	Attribute	Content
Z	Complex type One-dimensional array	Output	Roots of polynomial equations are output.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: C(1)=0.

*1 For double (quadruple) precision subroutines, all complex types should be double (quadruple) precision complex types.

(3) Calculation method

1. Quadratic equations conform to the root formulas and the relationship between the roots and coefficients.

2. Cubic equations conform to the modified Cardano method by Hirano [1].

3. Quartic equations conform to the Ferrari method.

Bibliography

1) Sugayasu Hirano: Numerical Solution of Polynomial Equations by Floating Point Arithemetic, doctorial thesis, 1980.

(1987.07.21)

QUADRS/D/Q/,CUBICS/D/Q,QUARTS/D/Q (Solution of low-order polynomial equations with real coefficients)

Solution of Low Order Polynomial Equations with Real Coefficients

Programmed by	Ichizo Ninomiya; April 1977
Format	Subroutine language; FORTRAN Size; 24, 27, 27, 40, 41, 41, 46, 47, and 47 lines respectively

(1) Outline

QUADRS (D, Q), CUBICS (D, Q), and QUARTS (D, Q) are single (double, quadruple) precision subroutines used to calculate all roots of the quadratic, cubic, and quartic equations (real coefficients) respectively.

(2) Directions

CALL { QUADRS/D/Q / CUBICS/D/Q / QUARTS/D/Q } (A, X, Y, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type One-dimensional array	Input	Coefficients for a polynomial equation is input in descending order of the degree. $A(1) \neq 0$
X	Real type One-dimensional array	Output	The real parts of roots of the polynomial equation are output.
Y	Real type One-dimensional array	Output	The imaginary parts of the roots of the polynomial equation is output.
ILL	Integer type	Output	ILL=0: Normal termination. ILL=30000: A(1) = 0.

*1 For double (quadruple) subroutines, real types are all changed to double (quadruple) precision real types.

(3) Calculation method

1. For a quadratic equation, only the root with the larger absolute value is determined by the quadratic formula and the other root is determined by using the relation between the product of the two roots and the coefficients.

2. A cubic equation is solved by the Cardano's method.

3. A quartic equation is solved by the Ferrari's method.

(4) Note

Fifth or higher degree equations can be solved by using GJMNKS/D/Q.

(1987.07.24) (1987.08.21)

RTFNDS/D (Solution of a nonlinear equation)

Solution of a Nonlinear Equation

Programmed by	Ichizo Ninomiya; August 1984
Format	Subroutine language; FORTRAN Size; 274 lines each

(1) Outline

RTFNDS and RTFNDD calculates all roots in the given interval of the given nonlinear equation.

(2) Directions

CALL RTFNDS(A, B, FUN, C, EPS, EPSZ, L, NR, RT, NF, BD, ILL)

Argument	Type and kind (*1)	Attribute	Content
A	Real type	Input	Left end of an interval. A<B
B	Real type	Input	Right end of an interval. A<B
FUN	Real type Function subprogram	Input	Function subprogram prepared by the user for f(x) when the equation to be solved is f(x)=0.
C	Real type	Input	Constant for Chebyshev test. Default value 3 is given when C≦0.
EPS	Real type	Input	Constant ε for root isolation test. The standard range is 10^{-1}~10^{-4}.
EPSZ	Real type	Input	Constant εz for root precision test.
L	Integer type	Input	Size for arrays RT and BD. About 100 is enough in most cases.
NR	Integer type	Output	Total number of roots

Argument	Type and kind (*1)	Attribute	Content
RT	Real type One-dimensional array	Output	NR roots are output in ascending order. This argument is also used as a work area during calculation.
NF	Integer type	Output	Number of evaluations of function f(x)
BD	Real type One-dimensional array	Work area	Size 4*NR is needed.
ILL	Integer type	Output	Error code. ILL=0: Normal termination. ILL=20000: L was so small that the capacity of array RT or BD was exceeded. Calculation has discontinued. ILL=30000: A≦B or L<2.

*1 For double precision subroutines, real types are all changed to be double precision real types.

(3) Calculation method

1. Sufficiently small intervals, each containing one of all roots in the interval (A, B), are detected by the B. Jones' root isolation method [1]. When each interval is (X1, X2), then $|X2-X1| \leqq \varepsilon \cdot Xm$ holds, where $Xm=\max(|X1+X2|/2, 1)$.

2. The root in each small interval obtained in 1. is calculated by the Popovski's method [2].

(4) Example

```
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RT(100),BD(100)
EXTERNAL FUN
A=0.0D0
B=15.D0
EPS=1.D-2
```

```
      EPSZ=1.D-8
      C=3.DO
      L=100
      CALL RTFNDD(A,B,FUN,C,EPS,EPSZ,L,NR,RT,NF,BD,ILL)
      WRITE(6,600) NR,(RT(I),I=1,NR)
600   FORMAT(5X,'NR=',I4/(5D16.8))
      STOP
      END
      FUNCTION FUN(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA PI /3.1415926535897 94DO/
      FUN=DSIN(PI*X/14.DO)+DSIN(PI*X*1.5DO)
      RETURN
      END
```

(5) Notes

1. Constant C is used in Chebyshev's inequality $m^2 \geqq Cv$ to test that a certain interval contains no root, where m and v are average value and variance of the function values in that interval respectively. If this inequality is satisfied, the statistical hypothesis "a root exists in this interval" is rejected with the level of significance $1/C$ or below. C=3.0 is often a suitable value. If C is too small, there is a danger of misjudging an existent root as "inexistent." Conversely, if C is too large, judgment is done too carefully, increasing the number of function evaluations.

2. Selection of the constant ε for root isolation test is also very important. If ε is too large, roots cannot be isolated completely to one another. If it is too small, the number of function evaluations increases. Once the roots have been isolated, subsequent calculation is done very fast regardless of εz. Therefore, assign ε a large value enough to isolate the roots.

3. If $|\delta z| \leqq \varepsilon z \cdot \max(|z|, 1)$, where z is roots and δz is their correction, is established, convergence is regarded to be completed. Note, therefore, that $|f(z)| \leqq \varepsilon z$ is not always established.

Bibliography

1) B. Jones et al.; "Root Isolation For Transcendental Equations," Computer Journal, Vol. 27, pp. 184-187 (1984)

2) D. B. Popovski; "A Note on King's Method F for Finding a Bracketed Roo," Computing, Vol. 29, pp. 355-359 (1980)

(1987. 06. 24) (1987. 08. 08)