BASIC LINEAR ALGEBRA MODULES

ISAMAX	-	index if the largest element of a vector
SASUM	-	1-norm of a vector
SAXPY	-	add multiple of one vector to another
SDOT	-	dot product of two vectors
SSCAL	-	scale a vector
SSWAP	-	interchange two vectors

Purpose: ISAMAX looks through a vector to find the (first) component with maximum magnitude. The integer position of that component in the vector is returned.

Type: Integer function

Usage: <answer> = ISAMAX(N, X, INCX)

N	$\longrightarrow$	the number of elements to be compared
X	$\longrightarrow$	the vector of elements
INCX	$\longrightarrow$	the elements are spaced at intervals of INCX in X: X(1), X(1+INCX), ..., X(1+(N–1)∗INCX)
<answer>	$\longleftarrow$	M, the position of the (first) component of maximum magnitude: If INCX = 1, $\lvert X(M) \rvert$ is largest. In general, $\lvert X(1 + (M - 1) * INCX) \rvert$ is largest.

Note 1: Since Fortran stores arrays in column-wise order we can use ISAMAX to deal with the rows of a 2-dimensional array as in the example below.

Note 2: If N=0, ISAMAX=0 is returned.

Error situations: (All errors in this subprogram are fatal — see *Error Handling*, Framework Chapter)

Number	Error
1	N < 0
2	INCX ≤ 0

Double-precision version: IDAMAX with X declared double precision

Complex version: ICAMAX with X declared complex

See also: SAMAX, ISMAX

Author: Linda Kaufman

Reference: Lawson, C. L., Hanson, R. J., Kincaid, D. R., and Krogh, F. T., Basic linear algebra subprograms for Fortran usage, *ACM Trans. Math. Software 5*, 3 (1979), 308-323.

Example: In this example the columns of an \$m times n\$ matrix A, dimensioned (IA,N) are permuted so that the (1,1) element of A is the largest in modulus of the elements of the first row in A. The subroutine SSWAP interchanges two vectors.

```
J=ISAMAX(N,A,IA)
CALL SSWAP(M,A,1,A(1,J),1)
```

SASUM — 1-norm of a vector

Purpose: SASUM computes the sum of the absolute values of a vector: $\sum_{i=1}^{n} | x_i |$

Type: Real function

Usage: <answer> = SASUM(N, X, INCX)

N	$\longrightarrow$	the number of elements to be summed
X	$\longrightarrow$	the vector of elements
INCX	$\longrightarrow$	the elements are spaced at intervals of INCX in X: X(1), X(1+INCX), ..., X(1+(N−1)INCX)
<answer>	$\longleftarrow$	| X(1) | + | X(1+INCX) | + ... + | X(1+(N−1)INCX) |

Note 1: Since Fortran stores arrays in column-wise order we can use SASUM to deal with the rows of a 2-dimensional array as in the example below.

Note 2: If N=0, SASUM=0.0 is returned.

Note 3: For complex vectors, SCASUM computes $\sum_{i=1}^{n} (|Re(x_i)|+|Im(x_i)|)$.

Note 4: No attempt is made to prevent or give warning of underflow or overflow.

Error situations: (All errors in this subprogram are fatal — see *Error Handling*, Framework Chapter)

Number	Error
1	N<0
2	INCX≤0

Double-precision version: DASUM with X declared double precision

Complex version: SCASUM with X declared complex (see Note 3).

Author: Linda Kaufman

Reference: Lawson, C. L., Hanson, R. J., Kincaid, D. R., and Krogh, F. T., Basic linear algebra subprograms for Fortran Usage, Report SAND77-0898, Sandia Laboratories, Albuquerque, New Mexico 87115, October 1977.

Examples: The following program fragment computes the 1-norm of an $m times n$ matrix A. The 1-norm is defined by $max from 1<=j<=n sum from i=1 to m | a sub i,j^^|$

```
   ANORM1=0.0
   DO 10 J=1,N
      ANORMJ = SASUM(M,A(1,J),1)
      IF (ANORMJ .GT. ANORM1) ANORM1 = ANORMJ
10 CONTINUE
```

The next program fragment computes the infinity norm of an $m times n$ matrix A, dimensioned (IA,N). The infinity norm is defined as $max from 1<=i<=m sum from j=1 to n | a sub i,j^^|$

```
   ANORM=0.0
   DO 10 I=1,M
      ANORMI = SASUM(N,A(I,1),IA)
      IF (ANORMI .GT. ANORM) ANORM=ANORMI
10 CONTINUE
```

SAXPY — add multiple of one vector to another

Purpose: SAXPY scales a vector x by a scalar a and adds the result to a vector y.

Usage: CALL SAXPY (N, A, X, INCX, Y, INCY)

N	$\longrightarrow$	the number of affected elements in X and Y
A	$\longrightarrow$	the scalar variable
X	$\longrightarrow$	the vector which is to be scaled
INCX	$\longrightarrow$	the elements are spaced at intervals of INCX in X: X(1), X(1+INCX), ..., X(1+(N−1)∗INCX)
Y	$\longrightarrow$	the vector which is to be added
	$\longleftarrow$	AX + Y
INCY	$\longrightarrow$	the elements are space at intervals of INCY in Y: Y(1), Y(1+INCY), ..., Y(1+(N−1)∗INCY)

Note: If N=0, no action is performed.

Error situations: (All errors in this subprogram are fatal — see *Error Handling*, Framework Chapter)

Number	Error
1	N<0
2	INCX≤0
3	INCY≤0

Double-precision version: DAXPY with X and Y declared double precision

Complex version: CAXPY with X and Y declared complex

See also: SSCAL

Author: Linda Kaufman

Reference: Lawson, C. L., Hanson, R. J., Kincaid, D. R., and Krogh, F. T., Basic linear algebra subprograms for Fortran usage, *ACM Trans. Math. Software 5*, 3 (1979), 308-323.

Example: The following program fragment forms the product Ax where A is an $m times n$ matrix and puts the result in an array Y:

```
      DO 10 I=1,M
         Y(I)=0.0
   10 CONTINUE
      DO 20 I=1,N
         CALL SAXPY(M,X(I),A(1,I),1,Y,1)
   20 CONTINUE
```

Matrix by vector multiplication is usually done using inner products as in the example in SDOT, but on a paged machine using the above program fragment can be preferable because FORTRAN stores two-dimensional arrays column-wise and this program refers to the array A one column at a time.

SDOT — dot product of two vectors

Purpose: SDOT determines the inner product of two vectors x and y, $sum from i=1 to n x sub i y sub i$

Type: Real function

Usage: <answer> = SDOT(N, X, INCX, Y, INCY)

N	$\longrightarrow$	the number of elements to be summed
X	$\longrightarrow$	the first vector
INCX	$\longrightarrow$	the elements are spaced at intervals of INCX in X: X(1), X(1+INCX), ..., X(1+(N–1)∗INCX)
Y	$\longrightarrow$	the second vector
INCY	$\longrightarrow$	the elements are spaced at intervals of INCY in Y: Y(1), Y(1+INCY), ..., Y(1+(N–1)∗INCY)
<answer>	$\longleftarrow$	X(1)∗Y(1)+X(1+INCX)∗Y(1+INCY)+...+ X(1+(N–1)∗INCX)∗Y(1+(N–1)∗INCY)

Note 1: Since Fortran stores arrays in column-wise order we can use SDOT to deal with the rows of a 2-dimensional array as in the example below.

Note 2: If N=0, SDOT=0.0 is returned.

Note 3: No attempt is made to prevent underflow or overflow in the subroutine.

Error situations: (All errors in this subprogram are fatal — see *Error Handling*, Framework Chapter)

Number	Error
1	N<0
2	INCX≤0
3	INCY≤0

Double-precision version: DDOT with X and Y declared double precision

Complex versions: CDOTU with X and Y declared complex. CDOTC with X and Y declared complex. CDOTC $= \sum_{i=1}^{n} \bar{x}_i y_i$, i.e. the conjugate of the elements of X are used.

Author: Linda Kaufman

Reference: Lawson, C. L., Hanson, R. J., Kincaid, D. R., and Krogh, F. T., Basic linear algebra subprograms for Fortran usage, *ACM Trans. Math. Software 5*, 3 (1979), 308-323.

Example: The following program fragment forms the product $A\mathbf{x}$ where A is an $m \times n$ matrix dimensioned (IA,N), and puts the result in an array Y:

```
   DO 10 I=1,M
      Y(I)=SDOT(N,A(I,1),IA,X,1)
10 CONTINUE
```

Because of page faults, the execution of this program fragment on certain machines might require an excessive amount of time. The program fragment given in the example in SAXPY, which accesses the elements of A one column at a time, would be preferable in this case.

SSCAL — scale a vector

Purpose: SSCAL multiplies a vector x by a scalar A

Usage: CALL SSCAL(N, A, X, INCX)

N	$\longrightarrow$	the number of affected elements in X
A	$\longrightarrow$	the scaling factor
X	$\longrightarrow$	the vector to be scaled
	$\longleftarrow$	the scaled vector
INCX	$\longrightarrow$	the elements are spaced at intervals of INCX in X: X(1), X(1+INCX), ..., X(1+(N−1)∗INCX)

Note 1: If N=0, no action is performed.

Note 2: Since Fortran stores arrays in column-wise order, we can use SSCAL to deal with the rows of a 2-dimensional array as in the example below.

Error situations: (All errors in this subprogram are fatal — see *Error Handling*, Framework Chapter)

Number	Error
1	N<0
2	INCX≤0

Double-precision version: DSCAL with X and A declared double precision

Complex versions: CSCAL with X and A declared complex
CSSCAL with X declared complex and A declared real

See also: SAXPY

Author: Linda Kaufman

Reference: Lawson, C. L., Hanson, R. J., Kincaid, D. R., and Krogh, F. T., Basic linear algebra subprograms for Fortran usage, *ACM Trans. Math. Software 5*, 3 (1979), 308-323.

Example: In this example the rows of an $m times n$ matrix A, dimensioned (IA,N) are scaled so that the sum of the modulus of the elements in each row is 1.0. The function SASUM returns the sum of the absolute values of the elements of a vector.

```
      DO 10 J=1,M
         SC=1.0/SASUM(N,A(J,1),IA)
         CALL SSCAL(N,SC,A(J,1),IA)
   10 CONTINUE
```

SSWAP — interchange two vectors

Purpose: SSWAP interchanges two vectors

Usage: CALL SSWAP(N, X, INCX, Y, INCY)

N	$\longrightarrow$	the number of affected elements in X and Y
X	$\longrightarrow$	the first vector
	$\longleftarrow$	the vector Y
INCX	$\longrightarrow$	the elements are spaced at intervals of INCX in X: X(1), X(1+INCX), ..., X(1+(N−1)INCX)
Y	$\longrightarrow$	the second vector
	$\longleftarrow$	the vector X
INCY	$\longrightarrow$	the elements are spaced at intervals of INCY in Y: Y(1), Y(1+INCY), ..., Y(1+(N−1)*INCY)

Note 1: If N=0, no action is performed.

Note 2: Since Fortran stores arrays in column-wise order, we can use SSWAP to deal with the rows of a 2-dimensional array as in the example below.

Error situations: (All errors in this subprogram are fatal — see *Error Handling*, Framework Chapter)

Number	Error
1	N<0
2	INCX≤0
3	INCY≤0

Double-precision version: DSWAP with X and Y declared double precision

Complex version: CSWAP with X and Y declared complex

See also: MOVExx (Utility Chapter)

Author: Linda Kaufman

Reference: Lawson, C. L., Hanson, R. J., Kincaid, D. R., and Krogh, F. T., Basic linear algebra subprograms for Fortran usage, *ACM Trans. Math. Software 5*, 3 (1979), 308-323.

Example: In this example the rows of an $m times n$ matrix A, dimensioned (IA,N) are permuted so that the (1,1) element of A is the largest in modulus of the elements in the first column of A. The subroutine ISAMAX computes the index of the element of a vector having maximum modulus.

```
J=ISAMAX(M,A,1)
CALL SSWAP(N,A,IA,A(J,1),IA)
```