NAG Toolbox

nag_orthog_real_gram_schmidt (f05aa)

1 Purpose

nag_orthog_real_gram_schmidt (f05aa) applies the Schmidt orthogonalization process to n vectors in m-dimensional space, $n \leq m$.

2 Syntax

```
[a, cc, icol, ifail] = nag_orthog_real_gram_schmidt(a, n1, 'm', m, 'n2', n2)
[a, cc, icol, ifail] = f05aa(a, n1, 'm', m, 'n2', n2)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 22: **m** was made optional.

3 Description

nag_orthog_real_gram_schmidt (f05aa) applies the Schmidt orthogonalization process to n linearly independent vectors in m-dimensional space, $n \leq m$. The effect of this process is to replace the original n vectors by n orthonormal vectors which have the property that the rth vector is linearly dependent on the first r of the original vectors, and that the sum of squares of the elements of the rth vector is equal to 1, for $r = 1, 2, \ldots, n$. Inner-products are accumulated using ***additional precision***.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **a**(lda, **n2**) – REAL (KIND=nag_wp) array

lda, the first dimension of the array, must satisfy the constraint $lda \geq$ **m**.

Columns **n1** to **n2** contain the vectors to be orthogonalized. The vectors are stored by columns in elements 1 to m.

2: **n1** – INTEGER

The indices of the first and last columns of A to be orthogonalized.

Constraint: **n1** $\leq$ **n2**.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the first dimension of the array **a**.

m, the number of elements in each vector.

2: **n2** – INTEGER

Default: For **n2**, the second dimension of the array **a**.

The indices of the first and last columns of A to be orthogonalized.

Constraint: **n1** $\leq$ **n2**.

5.3 Output Parameters

1: **a**(lda, **n2**) – REAL (KIND=nag_wp) array

These vectors store the orthonormal vectors.

2: **cc** – REAL (KIND=nag_wp)

Is used to indicate linear dependence of the original vectors. The nearer **cc** is to 1.0, the more likely vector **icol** is dependent on vectors **n1** to **icol** − 1. See Section 9.

3: **icol** – INTEGER

The column number corresponding to **cc**. See Section 9.

4: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **n1** $>$ **n2**.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Innerproducts are accumulated using ***additional precision*** arithmetic and full machine accuracy should be obtained except when **cc** $>$ 0.99999. (See Section 9.)

8 Further Comments

The time taken by nag_orthog_real_gram_schmidt (f05aa) is approximately proportional to nm^2, where $n =$ **n2** $-$ **n1** $+ 1$.

Arguments **cc** and **icol** have been included to give some indication of whether or not the vectors are nearly linearly independent, and their values should always be tested on exit from the function. **cc** will be in the range $[0.0, 1.0]$ and the closer **cc** is to 1.0, the more likely the vector **icol** is to be linearly dependent on vectors **n1** to **icol** − 1. Theoretically, when the vectors are linearly dependent, **cc** should be exactly 1.0. In practice, because of rounding errors, it may be difficult to decide whether or not a value of **cc** close to 1.0 indicates linear dependence. As a general guide a value of **cc** $>$ 0.99999 usually indicates linear dependence, but examples exist which give **cc** $>$ 0.99999 for linearly independent vectors. If one of the original vectors is zero or if, possibly due to rounding errors, an exactly zero vector is produced by the Gram–Schmidt process, then **cc** is set exactly to 1.0 and the vector is not, of course, normalized. If more than one such vector occurs then **icol** references the last of these vectors.

If you are concerned about testing for near linear dependence in a set of vectors you may wish to consider using function nag_lapack_dgesvd (f08kb).

9 Example

This example orthonormalizes columns 2, 3 and 4 of the matrix:

$$\begin{pmatrix} 1 & -2 & 3 & 1 \\ -2 & 1 & -2 & -1 \\ 3 & -2 & 1 & 5 \\ 4 & 1 & 5 & 3 \end{pmatrix}.$$

9.1 Program Text

```
function f05aa_example

fprintf('f05aa example results\n\n');

a = [ 1, -2,  3,  1;
     -2,  1, -2, -1;
      3, -2,  1,  5;
      4,  1,  5,  3];

% Orthonormalize all but first column of A
n1 = nag_int(2);
[a, cc, icol, ifail] = f05aa(a, n1);

fprintf('Linear dependence measure for column %1d = %6.4f\n', icol, cc);
fprintf('\nFinal orthonormalized columns\n');
disp(a(:,n1:end));
```

9.2 Program Results

```
f05aa example results

Linear dependence measure for column 4 = 0.5822

Final orthonormalized columns
   -0.6325    0.3310   -0.5404
    0.3162   -0.2483    0.2119
   -0.6325   -0.0000    0.7735
    0.3162    0.9104    0.2543
```