

NAG Toolbox

nag_sum_convert_herm2complex_sep (c06gs)

1 Purpose

nag_sum_convert_herm2complex_sep (c06gs) takes m Hermitian sequences, each containing n data values, and forms the real and imaginary parts of the m corresponding complex sequences.

Note: This function is scheduled to be withdrawn, please see c06gs in Advice on Replacement Calls for Withdrawn/Superseded Routines..

2 Syntax

```
[u, v, ifail] = nag_sum_convert_herm2complex_sep(m, n, x)
[u, v, ifail] = c06gs(m, n, x)
```

3 Description

This is a utility function for use in conjunction with nag_sum_fft_real_1d_multi_rfmt (c06fp) and nag_sum_fft_hermitian_1d_multi_rfmt (c06fq) (see the C06 Chapter Introduction).

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **m** – INTEGER

m , the number of Hermitian sequences to be converted into complex form.

Constraint: $\mathbf{m} \geq 1$.

2: **n** – INTEGER

n , the number of data values in each Hermitian sequence.

Constraint: $\mathbf{n} \geq 1$.

3: **x**($\mathbf{m} \times \mathbf{n}$) – REAL (KIND=nag_wp) array

The data must be stored in **x** as if in a two-dimensional array of dimension ($1 : \mathbf{m}, 0 : \mathbf{n} - 1$); each of the m sequences is stored in a **row** of the array in Hermitian form. If the n data values z_j^p are written as $x_j^p + iy_j^p$, then for $0 \leq j \leq n/2$, x_j^p is contained in **x**(p, j), and for $1 \leq j \leq (n - 1)/2$, y_j^p is contained in **x**($p, n - j$). (See also Section 2.1.2 in the C06 Chapter Introduction.)

5.2 Optional Input Parameters

None.

5.3 Output Parameters

- 1: **u**($\mathbf{m} \times \mathbf{n}$) – REAL (KIND=nag_wp) array
 2: **v**($\mathbf{m} \times \mathbf{n}$) – REAL (KIND=nag_wp) array

The real and imaginary parts of the m sequences of length n , are stored in **u** and **v** respectively, as if in two-dimensional arrays of dimension ($1 : \mathbf{m}, 0 : \mathbf{n} - 1$); each of the m sequences is stored as if in a **row** of each array. In other words, if the real parts of the p th sequence are denoted by x_j^p , for $j = 0, 1, \dots, n - 1$ then the mn elements of the array **u** contain the values

$$x_0^1, x_0^2, \dots, x_0^m, x_1^1, x_1^2, \dots, x_1^m, \dots, x_{n-1}^1, x_{n-1}^2, \dots, x_{n-1}^m$$

- 3: **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, **m** < 1.

ifail = 2

On entry, **n** < 1.

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

Exact.

8 Further Comments

None.

9 Example

This example reads in sequences of real data values which are assumed to be Hermitian sequences of complex data stored in Hermitian form. The sequences are then expanded into full complex form using nag_sum_convert_herm2complex_sep (c06gs) and printed.

9.1 Program Text

```
function c06gs_example

fprintf('c06gs example results\n\n');

% 3 Hermitian sequences stored as rows in compact form
m = nag_int(3);
n = nag_int(6);
```

```

x = [0.3854  0.6772  0.1138  0.6751  0.6362  0.1424;
      0.5417  0.2983  0.1181  0.7255  0.8638  0.8723;
      0.9172  0.0644  0.6037  0.6430  0.0428  0.4815];

disp('Original values in compact Hermitian form:');
disp(x);

% Put x in full complex form
[u, v, ifail] = c06gs(m, n, x);

nd = [m,n];
z = reshape(u + i*v,nd);
disp(' ');
title = 'Original data in full complex form';
[ifail] = x04da('General','Non-unit', z, title);

```

9.2 Program Results

c06gs example results

```

Original values in compact Hermitian form:
 0.3854    0.6772    0.1138    0.6751    0.6362    0.1424
 0.5417    0.2983    0.1181    0.7255    0.8638    0.8723
 0.9172    0.0644    0.6037    0.6430    0.0428    0.4815

Original data in full complex form
      1         2         3         4         5         6
 1  0.3854  0.6772  0.1138  0.6751  0.1138  0.6772
 0.0000  0.1424  0.6362  0.0000 -0.6362 -0.1424

 2  0.5417  0.2983  0.1181  0.7255  0.1181  0.2983
 0.0000  0.8723  0.8638  0.0000 -0.8638 -0.8723

 3  0.9172  0.0644  0.6037  0.6430  0.6037  0.0644
 0.0000  0.4815  0.0428  0.0000 -0.0428 -0.4815

```
