

nag_conjugate_hermitian (c06gbc)

1. Purpose

nag_conjugate_hermitian (c06gbc) forms the complex conjugate of a Hermitian sequence of n data values.

2. Specification

```
#include <nag.h>
#include <nagc06.h>
```

```
void nag_conjugate_hermitian(Integer n, double x[], NagError *fail)
```

3. Description

This is a utility function for use in conjunction with **nag_fft_real (c06eac)** and **nag_fft_hermitian (c06ebc)**, to calculate inverse discrete Fourier transforms.

4. Parameters

n

Input: the number of data values, n .

Constraint: $n \geq 1$.

x[n]

Input: if the data values z_j are written as $x_j + iy_j$, then for $0 \leq j \leq n/2$, **x[j]** must contain $x_j (= x_{n-j})$, while for $n/2 < j \leq n-1$, **x[j]** must contain $-y_j (= y_{n-j})$. In other words, **x** must contain the Hermitian sequence in Hermitian form.

Output: the imaginary parts y_j are negated. The real parts x_j are not referenced.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_INT_ARG_LT

On entry, **n** must not be less than 1: **n** = *<value>*.

6. Further Comments

The time taken by the function is negligible.

6.1. Accuracy

Exact.

7. See Also

nag_fft_real (c06eac)
nag_fft_hermitian (c06ebc)

8. Example

This program reads in a sequence of real data values, calls **nag_fft_real (c06eac)** followed by **nag_conjugate_hermitian** to compute their inverse discrete Fourier transform, and prints this after expanding it from Hermitian form into a full complex sequence.

8.1. Program Text

```
/* nag_conjugate_hermitian(c06gbc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
```

```

*
* Mark 1, 1990.
*/

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagc06.h>

#define NMAX 20

main()
{
    Integer j, n, n2, nj;
    double a[NMAX], b[NMAX], x[NMAX];

    Vprintf("c06gbc Example Program Results\n");
    /* Skip heading in data file */
    Vscanf("%*[^\\n]");
    while (scanf("%ld", &n)!=EOF)
        if (n>1 && n<=NMAX)
        {
            for (j = 0; j<n; j++)
                Vscanf("%lf", &x[j]);
            /* Calculate inverse transform */
            /* Calculate transform of data */
            c06eac(n, x, NAGERR_DEFAULT);
            /* Calculate conjugates of Hermitian result to */
            /* give inverse transform */
            c06gbc(n, x, NAGERR_DEFAULT);
            /* Expand conjugated Hermitian sequence to full complex */
            a[0] = x[0];
            b[0] = 0.0;
            n2 = (n-1)/2;
            for (j = 1; j<=n2; j++)
            {
                nj = n - j;
                a[j] = x[j];
                a[nj] = x[j];
                b[j] = x[nj];
                b[nj] = -x[nj];
            }
            if (n % 2==0)
            {
                a[n2+1] = x[n2+1];
                b[n2+1] = 0.0;
            }
            Vprintf("\\nComponents of inverse discrete Fourier transform\\n");
            Vprintf("\\n          Real          Imag \\n\\n");
            for (j = 0; j<n; j++)
                Vprintf("%3ld %10.5f %10.5f\\n", j, a[j], b[j]);
        }
    else
    {
        Vfprintf(stderr, "Invalid value of n.\\n");
        exit(EXIT_FAILURE);
    }
    exit(EXIT_SUCCESS);
}

```

8.2. Program Data

```

c06gbc Example Program Data
7
0.34907
0.54890
0.74776
0.94459
1.13850
1.32850
1.51370

```

8.3. Program Results

c06gbc Example Program Results

Components of inverse discrete Fourier transform

	Real	Imag
0	2.48361	0.00000
1	-0.26599	-0.53090
2	-0.25768	-0.20298
3	-0.25636	-0.05806
4	-0.25636	0.05806
5	-0.25768	0.20298
6	-0.26599	0.53090
