

nag_conjugate_complex (c06gcc)

1. Purpose

nag_conjugate_complex (c06gcc) forms the complex conjugate of a sequence of n data values.

2. Specification

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

```
void nag_conjugate_complex(Integer n, double y[], NagError *fail)
```

3. Description

This is a utility function for use in conjunction with **nag_fft_complex** (c06ecc) to calculate inverse discrete Fourier transforms.

4. Parameters

n

Input: the number of data values, n .

Constraint: $n \geq 1$.

y[n]

Input: **y[j]** must contain the imaginary part of the j th data value, for $0 \leq j \leq n - 1$.

Output: these values are negated.

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: $n = \langle value \rangle$.

6. Further Comments

The time taken by the function is negligible.

6.1. Accuracy

Exact.

7. See Also

nag_fft_complex (c06ecc)

8. Example

This program reads in a sequence of complex data values and prints their inverse discrete Fourier transform as computed by calling **nag_conjugate_complex**, followed by **nag_fft_complex** (c06ecc) and **nag_conjugate_complex** again.

8.1. Program Text

```
/* nag_conjugate_complex(c06gcc) 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 ;
    double x[NMAX], y[NMAX];

    Vprintf("c06gcc Example Program Results\n");
    /* Skip heading in data file */
    Vscanf("%*[^\\n]");
    while (scanf("%ld", &n)!=EOF)
        if (n>1 && n<=NMAX)
            {
                /* Read in complex data */
                for (j = 0; j<n; ++j)
                    Vscanf("%lf%lf", &x[j], &y[j]);
                /* Compute inverse transform */
                /* Calculate conjugates of data */
                c06gcc(n, y, NAGERR_DEFAULT);
                /* Calculate transform of conjugated data */
                c06ecc(n, x, y, NAGERR_DEFAULT);
                /* Conjugate to give inverse transform */
                c06gcc(n, y, NAGERR_DEFAULT);
                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, x[j], y[j]);
            }
        else
            {
                Vfprintf(stderr, "\nInvalid value of n\n");
                exit(EXIT_FAILURE);
            }
    exit(EXIT_SUCCESS);
}

```

8.2. Program Data

```

c06gcc Example Program Data
7
0.34907  -0.37168
0.54890  -0.35669
0.74776  -0.31175
0.94459  -0.23702
1.13850  -0.13274
1.32850   0.00074
1.51370   0.16298

```

8.3. Program Results

```

c06gcc Example Program Results

Components of inverse discrete Fourier transform

          Real          Imag
0    2.48361   -0.47100
1    0.01983   -0.56496
2   -0.14825   -0.30840
3   -0.22506   -0.17477
4   -0.28767   -0.05865
5   -0.36711    0.09756
6   -0.55180    0.49684

```