nag_fft_init_trig (c06gzc)

1. Purpose

nag_fft_init_trig (c06gzc) calculates the trigonometric coefficients required for the computation of discrete Fourier transforms.

2. Specification

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

void nag_fft_init_trig(Integer n, double trig[], NagError *fail)
```

3. Description

This is a utility function for use in conjunction with one or more of nag_fft_multiple_real (c06fpc), nag_fft_multiple_hermitian (c06fqc), nag_fft_multiple_complex (c06frc), nag_fft_2d_complex (c06fuc), nag_fft_multiple_sine (c06hac), nag_fft_multiple_cosine (c06hbc), nag_fft_multiple_qtr_sine (c06hcc) and nag_fft_multiple_qtr_cosine (c06hdc). nag_fft_init_trig initialises the array **trig** with trigonometric coefficients according to the value of **n** and must be called prior to the first call of one of the above listed functions.

4. Parameters

n

Input: the value of n in the Fourier transform function being called.
Constraint: $\mathbf{n} \geq 1$.

trig[2∗n]

Output: the trigonometric coefficients are stored in **trig**.

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

6.1. Accuracy

Exact.

7. See Also

nag_fft_multiple_real (c06fpc)
nag_fft_multiple_hermitian (c06fqc)
nag_fft_multiple_complex (c06frc)
nag_fft_2d_complex (c06fuc)
nag_fft_multiple_sine (c06hac)
nag_fft_multiple_cosine (c06hbc)
nag_fft_multiple_qtr_sine (c06hcc)
nag_fft_multiple_qtr_cosine (c06hdc)

8. Example

The program reads in 3 real data sequences and prints their discrete Fourier transforms in Hermitian format as calculated by nag_fft_multiple_real (c06fpc). A call is made to nag_fft_init_trig to initialise the array **trig** prior to calling nag_fft_multiple_real (c06fpc). The transforms are then printed out in full complex form after a call to nag_multiple_hermitian_to_complex (c06gsc).

8.1. Program Text

```
/* nag_fft_init_trig(c06gzc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

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

#define MMAX  5
#define NMAX 20

main()
{
  double trig[2*NMAX];
  Integer i, j, m, n;
  double u[MMAX*NMAX], v[MMAX*NMAX];
  double x[MMAX*NMAX];

  Vprintf("c06gzc Example Program Results\n");
  Vscanf("%*[^\n]");   /* Skip heading in data file */
  while (scanf("%ld%ld", &m, &n)!=EOF)
    if (m<=MMAX && n<=NMAX)
      {
        Vprintf("\n\nm = %2ld  n = %2ld\n", m, n);
        /* Read in data and print out. */
        for (j = 0; j<m; ++j)
          for (i = 0; i<n; ++i)
            Vscanf("%lf", &x[j*n + i]);
        Vprintf("\nOriginal data values\n\n");
        for (j = 0; j<m; ++j)
          {
            Vprintf("    ");
            for (i = 0; i<n; ++i)
              Vprintf("%10.4f%s", x[j*n + i],
                      (i%6==5 && i!=n-1 ? "\n    " : ""));
            Vprintf("\n");
          }
        c06gzc(n, trig, NAGERR_DEFAULT);  /* Initialize trig array */
        /* Calculate transform */
        c06fpc(m, n, x, trig, NAGERR_DEFAULT);
        Vprintf("\nDiscrete Fourier transforms in Hermitian format\n\n");
        for (j = 0; j<m; ++j)
          {
            Vprintf("    ");
            for (i = 0; i<n; ++i)
              Vprintf("%10.4f%s", x[j*n + i],
                      (i%6==5 && i!=n-1 ? "\n     " : ""));
            Vprintf("\n");
          }
        /* Convert Hermitian form to full complex */
        c06gsc(m, n, x, u, v, NAGERR_DEFAULT);
        Vprintf("\nFourier transforms in full complex form\n\n");
        for (j = 0; j<m; ++j)
          {
            Vprintf("Real");
            for (i = 0; i<n; ++i)
              Vprintf("%10.4f%s", u[j*n + i],
                      (i%6==5 && i!=n-1 ? "\n    " : ""));
            Vprintf("\nImag");
            for (i = 0; i<n; ++i)
              Vprintf("%10.4f%s", v[j*n + i],
                      (i%6==5 && i!=n-1 ? "\n     " : ""));
            Vprintf("\n\n");
          }
```

```
      }
    else
      {
        Vfprintf(stderr,"\nInvalid value of m or n.\n");
        exit(EXIT_FAILURE);
      }
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
c06gzc Example Program Data
     3     6
     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
```

8.3. Program Results

```
c06gzc Example Program Results


m =  3  n =  6

Original data values

        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

Discrete Fourier transforms in Hermitian format

        1.0737   -0.1041    0.1126   -0.1467   -0.3738   -0.0044
        1.3961   -0.0365    0.0780   -0.1521   -0.0607    0.4666
        1.1237    0.0914    0.3936    0.1530    0.3458   -0.0508

Fourier transforms in full complex form

Real    1.0737   -0.1041    0.1126   -0.1467    0.1126   -0.1041
Imag    0.0000   -0.0044   -0.3738    0.0000    0.3738    0.0044

Real    1.3961   -0.0365    0.0780   -0.1521    0.0780   -0.0365
Imag    0.0000    0.4666   -0.0607    0.0000    0.0607   -0.4666

Real    1.1237    0.0914    0.3936    0.1530    0.3936    0.0914
Imag    0.0000   -0.0508    0.3458    0.0000   -0.3458    0.0508
```