

NAG Library Routine Document

C06GBF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

C06GBF forms the complex conjugate of a Hermitian sequence of n data values.

2 Specification

```
SUBROUTINE C06GBF (X, N, IFAIL)
  INTEGER          N, IFAIL
  REAL (KIND=nag_wp) X(N)
```

3 Description

This is a utility routine for use in conjunction with C06EAF, C06EBF, C06FAF or C06FBF to calculate inverse discrete Fourier transforms (see the C06 Chapter Introduction).

4 References

None.

5 Parameters

- 1: X(N) – REAL (KIND=nag_wp) array *Input/Output*
On entry: if the data values z_j are written as $x_j + iy_j$ and if X is declared with bounds $(0 : N - 1)$ in the subroutine from which C06GBF is called, 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. (See also Section 2.1.2 in the C06 Chapter Introduction.)
On exit: the imaginary parts y_j are negated. The real parts x_j are not referenced.
- 2: N – INTEGER *Input*
On entry: n , the number of data values.
Constraint: $N \geq 1$.
- 3: IFAIL – INTEGER *Input/Output*
On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.
 For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**
On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry `IFAIL = 0` or `-1`, explanatory error messages are output on the current error message unit (as defined by `X04AAF`).

Errors or warnings detected by the routine:

`IFAIL = 1`

On entry, $N < 1$.

`IFAIL = -99`

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

See Section 3.8 in the Essential Introduction for further information.

`IFAIL = -399`

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

See Section 3.7 in the Essential Introduction for further information.

`IFAIL = -999`

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Exact.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by `C06GBF` is negligible.

10 Example

This example reads in a sequence of real data values, calls `C06EAF` followed by `C06GBF` to compute their inverse discrete Fourier transform, and prints this after expanding it from Hermitian form into a full complex sequence.

10.1 Program Text

```

Program c06gbfe

!      C06GBF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
Use nag_library, Only: c06eaf, c06gbf, c06gsf, nag_wp
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Integer                    :: ieof, ifail, j, m, n
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: a(:), b(:), x(:)

```

```

!      .. Executable Statements ..
      Write (nout,*) 'C06GBF Example Program Results'
!      Skip heading in data file
      Read (nin,*)
loop: Do
      Read (nin,*,Iostat=ieof) n
      If (ieof<0) Exit loop

      Allocate (a(0:n-1),b(0:n-1),x(0:n-1))
      Read (nin,*) x(0:n-1)

!      ifail: behaviour on error exit
!      =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call c06eaf(x,n,ifail)

      Call c06gbf(x,n,ifail)

      Write (nout,*)
      Write (nout,*) 'Components of inverse discrete Fourier transform'
      Write (nout,*)
      Write (nout,*) '          Real          Imag'
      Write (nout,*)

!      Convert x to separated real and imaginary parts for printing.
      ifail = 0
      m = 1
      Call c06gsf(m,n,x,a,b,ifail)
      Write (nout,99999)(j,a(j),b(j),j=0,n-1)
      Deallocate (a,b,x)
End Do loop

99999 Format (1X,I6,2F10.5)
      End Program c06gbfe

```

10.2 Program Data

```

C06GBF Example Program Data
  7          : n
  0.34907
  0.54890
  0.74776
  0.94459
  1.13850
  1.32850
  1.51370          : x

```

10.3 Program Results

C06GBF 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
