NAG Library Function Document

nag_zpttrf (f07jrc)

1 Purpose

nag_zpttrf (f07jrc) computes the modified Cholesky factorization of a complex n by n Hermitian positive definite tridiagonal matrix A.

2 Specification

```
#include <nag.h>
#include <nagf07.h>
```

```
void nag_zpttrf (Integer n, double d[], Complex e[], NagError *fail)
```

3 Description

nag_zpttrf (f07jrc) factorizes the matrix A as

$$A = LDL^{\mathrm{H}},$$

where L is a unit lower bidiagonal matrix and D is a diagonal matrix with positive diagonal elements. The factorization may also be regarded as having the form $U^{\mathrm{H}}DU$, where U is a unit upper bidiagonal matrix.

4 References

None.

5 Arguments

1: **n** – Integer *Input*

On entry: n, the order of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

2: **d**[dim] – double *Input/Output*

Note: the dimension, *dim*, of the array **d** must be at least $\max(1, \mathbf{n})$.

On entry: must contain the n diagonal elements of the matrix A.

On exit: is overwritten by the n diagonal elements of the diagonal matrix D from the LDL^{H} factorization of A.

3: **e**[dim] – Complex *Input/Output*

Note: the dimension, *dim*, of the array **e** must be at least $\max(1, \mathbf{n} - 1)$.

On entry: must contain the $(n-1)$ subdiagonal elements of the matrix A.

On exit: is overwritten by the $(n-1)$ subdiagonal elements of the lower bidiagonal matrix L. (**e** can also be regarded as containing the $(n-1)$ superdiagonal elements of the upper bidiagonal matrix U.)

4: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_INT

On entry, **n** = ⟨*value*⟩.
Constraint: **n** ≥ 0.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_MAT_NOT_POS_DEF

The leading minor of order n is not positive definite, the factorization was completed, but **d**[**n** $- 1] \leq 0$.

The leading minor of order ⟨*value*⟩ is not positive definite, the factorization could not be completed.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

7 Accuracy

The computed factorization satisfies an equation of the form

$$A + E = LDL^{\mathrm{H}},$$

where

$$\|E\|_\infty = O(\epsilon)\|A\|_\infty$$

and ϵ is the ***machine precision***.

Following the use of this function, nag_zpttrs (f07jsc) can be used to solve systems of equations $AX = B$, and nag_zptcon (f07juc) can be used to estimate the condition number of A.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The total number of floating-point operations required to factorize the matrix A is proportional to n.

The real analogue of this function is nag_dpttrf (f07jdc).

10 Example

This example factorizes the Hermitian positive definite tridiagonal matrix A given by

$$A = \begin{pmatrix} 16.0 & 16.0 - 16.0i & 0 & 0 \\ 16.0 + 16.0i & 41.0 & 18.0 + 9.0i & 0 \\ 0 & 18.0 - 9.0i & 46.0 & 1.0 + 4.0i \\ 0 & 0 & 1.0 - 4.0i & 21.0 \end{pmatrix}.$$

10.1 Program Text

```
/* nag_zpttrf (f07jrc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 23, 2011.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf07.h>

int main(void)
{

  /* Scalars */
  Integer  exit_status = 0, i, n;

  /* Arrays */
  Complex  *e = 0;
  double   *d = 0;

  /* Nag Types */
  NagError fail;

  INIT_FAIL(fail);

  printf("nag_zpttrf (f07jrc) Example Program Results\n\n");
  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif

#ifdef _WIN32
  scanf_s("%"NAG_IFMT"%*[^\n]", &n);
#else
  scanf("%"NAG_IFMT"%*[^\n]", &n);
#endif
  if (n < 0)
    {
      printf("Invalid n\n");
      exit_status = 1;
      goto END;
    }
  /* Allocate memory */
  if (!(e = NAG_ALLOC(n-1, Complex)) ||
      !(d = NAG_ALLOC(n, double)))
    {
      printf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }

  /* Read the lower bidiagonal part of the tridiagonal matrix A from */
  /* data file */
#ifdef _WIN32
```

```
  for (i = 0; i < n; ++i) scanf_s("%lf", &d[i]);
#else
  for (i = 0; i < n; ++i) scanf("%lf", &d[i]);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
#ifdef _WIN32
  for (i = 0; i < n - 1; ++i) scanf_s(" ( %lf , %lf )", &e[i].re, &e[i].im);
#else
  for (i = 0; i < n - 1; ++i) scanf(" ( %lf , %lf )", &e[i].re, &e[i].im);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif

  /* Factorize the tridiagonal matrix A using nag_zpttrf (f07jrc). */
  nag_zpttrf(n, d, e, &fail);
  if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_zpttrf (f07jrc).\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }

  /* Print details of the factorization */
  printf("Details of factorization\n\n");
  printf(" The diagonal elements of D\n");
  for (i = 0; i < n; ++i) printf("%9.4f%s", d[i], i%8 == 7?"\n":" ");

  printf("\n\n Sub-diagonal elements of the Cholesky factor L\n");
  for (i = 0; i < n-1; ++i)
    printf("(%8.4f, %8.4f)%s", e[i].re, e[i].im, i%8 == 7?"\n":" ");
  printf("\n");

 END:
  NAG_FREE(e);
  NAG_FREE(d);

  return exit_status;
}
```

10.2 Program Data

```
nag_zpttrf (f07jrc) Example Program Data
    4                                                   : n
   16.0          41.0          46.0          21.0   : diagonal d
 ( 16.0, 16.0) ( 18.0, -9.0) (  1.0, -4.0)         : sub-diagonal e
```

10.3 Program Results

```
nag_zpttrf (f07jrc) Example Program Results

Details of factorization

 The diagonal elements of D
  16.0000    9.0000    1.0000    4.0000

 Sub-diagonal elements of the Cholesky factor L
(  1.0000,   1.0000) (  2.0000,  -1.0000) (  1.0000,  -4.0000)
```