NAG Library Function Document

nag_superlu_matrix_norm (f11mlc)

1 Purpose

nag_superlu_matrix_norm (f11mlc) computes the 1-norm, the ∞-norm or the maximum absolute value of the elements of a real, square, sparse matrix which is held in compressed column (Harwell–Boeing) format.

2 Specification

```
#include <nag.h>
#include <nagf11.h>
void nag_superlu_matrix_norm (Nag_NormType norm, double *anorm, Integer n,
     const Integer icolzp[], const Integer irowix[], const double a[],
     NagError *fail)
```

3 Description

nag_superlu_matrix_norm (f11mlc) computes various quantities relating to norms of a real, sparse n by n matrix A presented in compressed column (Harwell–Boeing) format.

4 References

None.

5 Arguments

1: **norm** – Nag_NormType *Input*

On entry: specifies the value to be returned in **anorm**.

norm = Nag_RealOneNorm

The 1-norm $\|A\|_1$ of the matrix is computed, that is $\max\limits_{1\le j\le n}\sum\limits_{i=1}^{n}|A_{ij}|$.

norm = Nag_RealInfNorm

The ∞-norm $\|A\|_\infty$ of the matrix is computed, that is $\max\limits_{1\le i\le n}\sum\limits_{j=1}^{n}|A_{ij}|$.

norm = Nag_RealMaxNorm

The value $\max\limits_{1\le i,j\le n}|A_{ij}|$ (not a norm).

Constraint: **norm** = Nag_RealOneNorm, Nag_RealInfNorm or Nag_RealMaxNorm.

2: **anorm** – double * *Output*

On exit: the computed quantity relating the matrix.

3: **n** – Integer *Input*

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

Constraint: **n** ≥ 0.

4: **icolzp**[dim] – const Integer *Input*

Note: the dimension, *dim*, of the array **icolzp** must be at least $\mathbf{n} + 1$.

On entry: **icolzp**[$i - 1$] contains the index in A of the start of a new column. See Section 2.1.3 in the f11 Chapter Introduction.

5: **irowix**[dim] – const Integer *Input*

Note: the dimension, *dim*, of the array **irowix** must be at least **icolzp**[**n**] $- 1$, the number of nonzeros of the sparse matrix A.

On entry: the row index array of sparse matrix A.

6: **a**[dim] – const double *Input*

Note: the dimension, *dim*, of the array **a** must be at least **icolzp**[**n**] $- 1$, the number of nonzeros of the sparse matrix A.

On entry: the array of nonzero values in the sparse matrix A.

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

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

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

NE_INT

On entry, $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{n} \geq 0$.

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 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_superlu_matrix_norm (f11mlc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example computes norms and maximum absolute value of the matrix A, where

$$A = \begin{pmatrix} 2.00 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & -1.00 & 0 \\ 4.00 & 0 & 1.00 & 0 & 1.00 \\ 0 & 0 & 0 & 1.00 & 2.00 \\ 0 & -2.00 & 0 & 0 & 3.00 \end{pmatrix}.$$

10.1 Program Text

```
/* nag_superlu_matrix_norm (f11mlc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf11.h>

int main(void)
{
  double anorm;
  Integer exit_status = 0, i, n, nnz;
  double *a = 0;
  Integer *icolzp = 0, *irowix = 0;
  /* Nag types */
  Nag_NormType norm;
  NagError fail;

  INIT_FAIL(fail);

  printf("nag_superlu_matrix_norm (f11mlc) Example Program Results\n\n");
  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif
  /* Read order of matrix and number of right hand sides */
#ifdef _WIN32
  scanf_s("%" NAG_IFMT "%*[^\n] ", &n);
#else
  scanf("%" NAG_IFMT "%*[^\n] ", &n);
#endif
  /* Read the matrix A */
  if (!(icolzp = NAG_ALLOC(n + 1, Integer)))
  {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }
  for (i = 1; i <= n + 1; ++i)
#ifdef _WIN32
    scanf_s("%" NAG_IFMT "%*[^\n] ", &icolzp[i - 1]);
#else
    scanf("%" NAG_IFMT "%*[^\n] ", &icolzp[i - 1]);
#endif
```

```
  nnz = icolzp[n] - 1;
  /* Allocate memory */
  if (!(a = NAG_ALLOC(nnz, double)) || !(irowix = NAG_ALLOC(nnz, Integer)))
  {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }
  for (i = 1; i <= nnz; ++i)
#ifdef _WIN32
    scanf_s("%lf%" NAG_IFMT "%*[^\n] ", &a[i - 1], &irowix[i - 1]);
#else
    scanf("%lf%" NAG_IFMT "%*[^\n] ", &a[i - 1], &irowix[i - 1]);
#endif
  /* Calculate 1-norm */
  norm = Nag_RealOneNorm;
  /* nag_superlu_matrix_norm (f11mlc).
   * 1-norm, infinity-norm, largest absolute element, real
   * general matrix
   */
  nag_superlu_matrix_norm(norm, &anorm, n, icolzp, irowix, a, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_superlu_matrix_norm (f11mlc).\n%s\n",
           fail.message);
    exit_status = 1;
    goto END;
  }

  /* Output norm */
  printf("%s\n%7.3f\n", "One-norm", anorm);

  /* Calculate M-norm */
  norm = Nag_RealMaxNorm;
  /* nag_superlu_matrix_norm (f11mlc), see above. */
  nag_superlu_matrix_norm(norm, &anorm, n, icolzp, irowix, a, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_superlu_matrix_norm (f11mlc).\n%s\n",
           fail.message);
    exit_status = 1;
    goto END;
  }

  /* Output norm */
  printf("\n");
  printf("%s\n%7.3f\n", "Max", anorm);

  /* Calculate I-norm */
  norm = Nag_RealInfNorm;
  /* nag_superlu_matrix_norm (f11mlc), see above. */
  nag_superlu_matrix_norm(norm, &anorm, n, icolzp, irowix, a, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_superlu_matrix_norm (f11mlc).\n%s\n",
           fail.message);
    exit_status = 1;
    goto END;
  }

  /* Output norm */
  printf("\n");
  printf("%s\n%7.3f\n", "Infinity-norm", anorm);

END:
  NAG_FREE(a);
  NAG_FREE(icolzp);
  NAG_FREE(irowix);

  return exit_status;
}
```

10.2 Program Data

```
nag_superlu_matrix_norm (f11mlc) Example Program Data
  5  n
 1
 3
 5
 7
 9
 12   icolzp(i) i=0..n
  2.   1
  4.   3
  1.   1
 -2.   5
  1.   2
  1.   3
 -1.   2
  1.   4
  1.   3
  2.   4
  3.   5    a(i) irowix(i) i=0..nnz-1
```

10.3 Program Results

```
nag_superlu_matrix_norm (f11mlc) Example Program Results

One-norm
  6.000

Max
  4.000

Infinity-norm
  6.000
```