NAG Library Function Document

nag_erfcx (s15agc)

1 Purpose

nag_erfcx (s15agc) returns the value of the scaled complementary error function $\text{erfcx}(x)$.

2 Specification

```c
#include <nag.h>
#include <nags.h>
double nag_erfcx (double x, NagError *fail)
```

3 Description

nag_erfcx (s15agc) calculates an approximate value for the scaled complementary error function $\text{erfcx}(x) = e^x \text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} \, dt = e^x (1 - \text{erf}(x))$.

Let $\hat{x}$ be the root of the equation $\text{erfc}(x) - \text{erf}(x) = 0$ (then $\hat{x} \approx 0.46875$). For $|x| \leq \hat{x}$ the value of $\text{erfcx}(x)$ is based on the following rational Chebyshev expansion for $\text{erf}(x)$:

$$\text{erf}(x) \approx x R_{\ell,m}(x^2),$$

where $R_{\ell,m}$ denotes a rational function of degree $\ell$ in the numerator and $m$ in the denominator.

For $|x| > \hat{x}$ the value of $\text{erfcx}(x)$ is based on a rational Chebyshev expansion for $\text{erfc}(x)$: for $\hat{x} < |x| \leq 4$ the value is based on the expansion

$$\text{erfc}(x) \approx e^x R_{\ell,m}(x);$$

and for $|x| > 4$ it is based on the expansion

$$\text{erfc}(x) \approx e^x \left( \frac{1}{\sqrt{\pi}} + \frac{1}{x^2 R_{\ell,m}(1/x^2)} \right).$$

For each expansion, the specific values of $\ell$ and $m$ are selected to be minimal such that the maximum relative error in the expansion is of the order $10^{-d}$, where $d$ is the maximum number of decimal digits that can be accurately represented for the particular implementation (see nag_decimal_digits (X02BEC)).

Asymptotically, $\text{erfcx}(x) \sim 1/(\sqrt{\pi}|x|)$. There is a danger of setting underflow in $\text{erfcx}(x)$ whenever $x \geq x_{\text{hi}} = \min(x_{\text{huge}}, 1/(\sqrt{\pi}x_{\text{tiny}}))$, where $x_{\text{huge}}$ is the largest positive model number (see nag_real_largest_number (X02ALC)) and $x_{\text{tiny}}$ is the smallest positive model number (see nag_real_smallest_number (X02AKC)). In this case nag_erfcx (s15agc) exits with fail.code = NW_HI and returns $\text{erfcx}(x) = 0$. For $x$ in the range $1/(2\sqrt{\epsilon}) \leq x < x_{\text{hi}}$, where $\epsilon$ is the machine precision, the asymptotic value $1/(\sqrt{\pi}|x|)$ is returned for $\text{erfcx}(x)$ and nag_erfcx (s15agc) exits with fail.code = NW_REAL.

There is a danger of setting overflow in $e^x$ whenever $x < x_{\text{neg}} = -\sqrt{\log(x_{\text{huge}}/2)}$. In this case nag_erfcx (s15agc) exits with fail.code = NW_NEG and returns $\text{erfcx}(x) = x_{\text{huge}}$.

The values of $x_{\text{hi}}, 1/(2\sqrt{\epsilon})$ and $x_{\text{neg}}$ are given in the Users’ Note for your implementation.
4 References

5 Arguments
1:  \( x \) – double
    *Input*
    \( On \ entry: \) the argument \( x \) of the function.

2:  \( \text{fail} \) – 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_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_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.

**NW_HI**
On entry, \( x = \langle \text{value} \rangle \) and the constant \( x_{hi} = \langle \text{value} \rangle \).
Constraint: \( x < x_{hi} \).

**NW_NEG**
On entry, \( x = \langle \text{value} \rangle \) and the constant \( x_{neg} = \langle \text{value} \rangle \).
Constraint: \( x \geq x_{neg} \).

**NW_REAL**
On entry, \( |x| \) was in the interval \( [\langle \text{value} \rangle, \langle \text{value} \rangle] \) where \( \text{erfcx}(x) \) is approximately \( 1/(\sqrt{\pi} * |x|) \):
\( x = \langle \text{value} \rangle \).

7 Accuracy
The relative error in computing \( \text{erfcx}(x) \) may be estimated by evaluating
\[
E = \frac{\text{erfcx}(x) - e^{2} \sum_{n=1}^{\infty} I^n \text{erfc}(x)}{\text{erfcx}(x)},
\]
where \( I^n \) denotes repeated integration. Empirical results suggest that on the interval \( (\tilde{x}, 2) \) the loss in base \( b \) significant digits for maximum relative error is around 3.3, while for root-mean-square relative...
error on that interval it is 1.2 (see nag_real_base (X02BHC) for the definition of the model parameter \( b \)).
On the interval (2,20) the values are around 3.5 for maximum and 0.45 for root-mean-square relative
errors; note that on these two intervals \( \text{erfc}(x) \) is the primary computation. See also Section 7 in nag_erfc
(s15adc).

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

This example reads values of the argument \( x \) from a file, evaluates the function at each value of \( x \) and
prints the results.

10.1 Program Text

```c
/* nag_erfcx (s15agc) Example Program. */
* Copyright 2014 Numerical Algorithms Group.
* Mark 9, 2009. */
/* Pre-processor includes */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nags.h>

int main(void)
{
    /*Integer scalar and array declarations */
    Integer exit_status = 0;
    /*Double scalar and array declarations */
    double x, y;
    NagError fail;
    const char *str_fail;
    INIT_FAIL(fail);

    printf("nag_erfcx (s15agc) Example Program Results\n");
    /* Skip heading in data file*/
    #ifdef _WIN32
    scanf_s("%*[\n] ");
    #else
    scanf("%*[\n] ");
    #endif
    #ifdef _WIN32
    while (scanf_s("%lf%*[\n] ", &x) != EOF)
    #else
    while (scanf("%lf%*[\n] ", &x) != EOF)
    #endif
    {
        /*
        * nag_erfcx (s15agc)
        * Scaled complement of error function, erfcx(x)
        */
        y = nag_erfcx(x, &fail);
        if (fail.code != NE_NOERROR)
        {
            if (fail.code == NW_HI || fail.code == NW_NEG ||
                fail.code == NW_REAL)
```
nag_erfcx (s15agc) Example Program Data

-30.0
-6.0
-4.5
-1.0
1.0
4.5
6.0
70000000.0

10.3 Program Results

nag_erfcx (s15agc) Example Program Results

<table>
<thead>
<tr>
<th>x</th>
<th>erfcx(x)</th>
<th>fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.00000e+01</td>
<td>1.79769e+308</td>
<td>NW_NEG</td>
</tr>
<tr>
<td>-6.00000e+00</td>
<td>8.62246e+15</td>
<td></td>
</tr>
<tr>
<td>-4.50000e+00</td>
<td>1.24593e+09</td>
<td></td>
</tr>
<tr>
<td>-1.00000e+00</td>
<td>5.00898e+00</td>
<td></td>
</tr>
<tr>
<td>1.00000e+00</td>
<td>4.27584e-01</td>
<td></td>
</tr>
<tr>
<td>4.50000e+00</td>
<td>1.22485e-01</td>
<td></td>
</tr>
<tr>
<td>6.00000e+00</td>
<td>9.27766e-02</td>
<td></td>
</tr>
<tr>
<td>7.00000e+07</td>
<td>8.05985e-09</td>
<td>NW_REAL</td>
</tr>
</tbody>
</table>