NAG Library Function Document
nag_shifted_log (s01bac)

1 Purpose
nag_shifted_log (s01bac) returns a value of the shifted logarithmic function, ln(1 + x).

2 Specification
#include <nag.h>
#include <nags.h>
double nag_shifted_log (double x, NagError *fail)

3 Description
nag_shifted_log (s01bac) computes values of ln(1 + x), retaining full relative precision even when |x| is small. The function is based on the Chebyshev expansion

\[ \ln \frac{1 + p^2 + 2px}{1 + p^2 - 2px} = 4 \sum_{k=0}^{\infty} \frac{p^{2k+1}}{2k+1} T_{2k+1}(\bar{x}). \]

Setting \( \bar{x} = \frac{x(1 + p^2)}{2p(x + 2)} \), and choosing \( p = \frac{q - 1}{q + 1}, \; q = \sqrt{2} \) the expansion is valid in the domain \( x \in \left[ \frac{1}{\sqrt{2}} - 1, \sqrt{2} - 1 \right] \).

Outside this domain, ln(1 + x) is computed by the standard logarithmic function.

4 References

5 Arguments
   1: \( x \) – double \hspace{1cm} Input
      On entry: the argument \( x \) of the function.
      Constraint: \( x > -1.0 \).
   2: \( \text{fail} \) – NagError * \hspace{1cm} Input/Output
      The NAG error * argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings
NE_REAL_ARG_LE
   On entry, \( x = \langle \text{value} \rangle \).
   Constraint: \( x > -1.0 \).
7 Accuracy

The returned result should be accurate almost to \textit{machine precision}, with a limit of about 20 significant figures due to the precision of internal constants. Note however that if $x$ lies very close to $-1.0$ and is not exact (for example if $x$ is the result of some previous computation and has been rounded), then precision will be lost in the computation of $1+x$, and hence $\ln(1+x)$, in \texttt{nag_shifted_log (s01bac)}.

8 Parallelism and Performance

Not applicable.

9 Further Comments

Empirical tests show that the time taken for a call of \texttt{nag_shifted_log (s01bac)} usually lies between about 1.25 and 2.5 times the time for a call to the standard logarithm function.

10 Example

The example program 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_shifted_log (s01bac) Example Program. */
/* Copyright 2014 Numerical Algorithms Group. */
/* Mark 7, 2002. */
*
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nags.h>

int main(void)
{
    Integer exit_status = 0;
    double x, y;
    NagError fail;

    INIT_FAIL(fail);

    /* Skip heading in data file */
    #ifdef _WIN32
        scanf_s("%*[\n]");
    #else
        scanf("%*[\n]");
    #endif
    printf("nag_shifted_log (s01bac) Example Program Results
"y
        printf(" x y
";  
    #ifdef _WIN32
        while (scanf_s("%lf", &x) != EOF)
    #else
        while (scanf("%lf", &x) != EOF)
    #endif
    {  
        /* nag_shifted_log (s01bac).
         * ln(1+x)
         */
        y = nag_shifted_log(x, &fail);
        if (fail.code != NE_NOERROR)
        {  
            printf("Error from nag_shifted_log (s01bac).\n%s\n",
                    fail.message);
        }
    }
    return exit_status;
}
```

s01bac

\textit{NAG Library Manual}

s01bac.2

\textit{Mark 25}
exit_status = 1;
goto END;
}
printf("%13.4e %13.4e\n", x, y);
}

END:
return exit_status;
}

10.2 Program Data

nag_shifted_log (s01bac) Example Program Data
2.50e+0
1.25e-1
-9.06e-1
1.29e-3
-7.83e-6
1.00e-9

10.3 Program Results

nag_shifted_log (s01bac) Example Program Results

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5000e+00</td>
<td>1.2528e+00</td>
</tr>
<tr>
<td>1.2500e-01</td>
<td>1.1778e-01</td>
</tr>
<tr>
<td>-9.0600e-01</td>
<td>-2.3645e+00</td>
</tr>
<tr>
<td>1.2900e-03</td>
<td>1.2892e-03</td>
</tr>
<tr>
<td>-7.8300e-06</td>
<td>-7.8300e-06</td>
</tr>
<tr>
<td>1.0000e-09</td>
<td>1.0000e-09</td>
</tr>
</tbody>
</table>