NAG Library Function Document

nag_deviates_landau (g01ftc)

1 Purpose

nag_deviates_landau (g01ftc) returns the value of the inverse \( \Phi^{-1}(x) \) of the Landau distribution function.

2 Specification

```c
#include <nag.h>
#include <nagg01.h>

double nag_deviates_landau (double x, NagError *fail)
```

3 Description

nag_deviates_landau (g01ftc) evaluates an approximation to the inverse \( \Phi^{-1}(x) \) of the Landau distribution function given by

\[
\Psi(x) = \Phi^{-1}(x)
\]

(where \( \Phi(\lambda) \) is described in nag_prob_landau (g01etc) and nag_prob_density_landau (g01mtc)), using either linear or quadratic interpolation or rational approximations which mimic the asymptotic behaviour. Further details can be found in Köllbig and Schorr (1984).

It can also be used to generate Landau distributed random numbers in the range \( 0 < x < 1 \).

4 References


5 Arguments

1: \( x \) – double

Input

On entry: the argument \( x \) of the function.

Constraint: \( 0.0 < x < 1.0 \).

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.
Accuracy
At least 5 – 6 significant digits are correct. Such accuracy is normally considered to be adequate for applications in large scale Monte–Carlo simulations.

Parallelism and Performance
Not applicable.

Further Comments
None.

Example
This example evaluates $\Phi^{-1}(x)$ at $x = 0.5$, and prints the results.

Program Text

```c
#include <stdio.h>
#include <nag.h>
#include <nagg01.h>

int main(void)
{
    /* Scalars */
    double x, y;
    Integer exit_status;
    NagError fail;

    INIT_FAIL(fail);
    exit_status = 0;

    printf(" nag_deviates_landau (g01ftc) Example Program Results\n");

    /* Skip heading in data file */
    #ifdef _WIN32
        scanf_s("%*[\n ] ");
    #else
        scanf("%*[\n ] ");
    #endif

    return 0;
}
```


```c
scanf_s("%lf*\n", &x);
#else
scanf("%lf*\n", &x);
#endif

/* nag_deviates_landau (g01ftc).
 * Landau inverse function Psi(x)
 */
y = nag_deviates_landau(x, &fail);
if (fail.code == NE_NOERROR)
{
    printf("\n X Y\n\n");
    printf(" %3.1f %13.4e\n", x, y);
}
else
{
    printf("Error from nag_deviates_landau (g01ftc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}
END:
return exit_status;
```