NAG Library Function Document

nag_quad_1d_inf_exp_wt (d01ubc)

1 Purpose

nag_quad_1d_inf_exp_wt (d01ubc) returns the Gaussian quadrature approximation for the specific problem $\int_0^\infty \exp\left(-x^2\right) f(x)\, dx$. The degrees of precision catered for are: 1, 3, 5, 7, 9, 19, 29, 39 and 49, corresponding to values of $n = 1$, 2, 3, 4, 5, 10, 15, 20 and 25, where n is the number of weights.

2 Specification

```
#include <nag.h>
#include <nagd01.h>
void nag_quad_1d_inf_exp_wt (
     void (*fun)(const double x[], double f[], Integer n, Nag_Comm *comm,
          Integer *istop),
     Integer n, double *ans, Nag_Comm *comm, NagError *fail)
```

3 Description

nag_quad_1d_inf_exp_wt (d01ubc) uses the weights w_i and the abscissae x_i such that $\int_0^\infty \exp\left(-x^2\right) f(x)$ is approximated by $\sum_{i=1}^{n} w_i f(x_i)$ to maximum precision i.e., it is exact when $f(x)$ is a polynomial of degree $2n - 1$.

4 References

Golub G H and Welsch J H (1969) Calculation of Gauss quadrature rules *Math. Comput.* **23** 221–230

5 Arguments

1: **fun** – function, supplied by the user *External Function*

fun must return the integrands $f(x_i)$ in $\mathbf{f}(i)$ for each x_i in $\mathbf{x}(i)$, for $i = 1, 2, \ldots, \mathbf{n}$ at a given point.

The specification of **fun** is:

```
void fun (const double x[], double f[], Integer n, Nag_Comm *comm,
     Integer *istop)
```

1: **x**[**n**] – const double *Input*

On entry: the points at which the integrand function f must be evaluated.

2: **f**[**n**] – double *Output*

On exit: $\mathbf{f}(i)$ must contain the value of the integrand $f(x_i)$ evaluated at the point $\mathbf{x}(i)$, for $i = 1, 2, \ldots, \mathbf{n}$.

3: **n** – Integer *Input*

On entry: **n** specifies the number of weights and abscissae to be used.

4: **comm** – Nag_Comm *

Pointer to structure of type Nag_Comm; the following members are relevant to **fun**.

user – double *
iuser – Integer *
p – Pointer

The type Pointer will be `void *`. Before calling nag_quad_1d_inf_exp_wt (d01ubc) you may allocate memory and initialize these pointers with various quantities for use by **fun** when called from nag_quad_1d_inf_exp_wt (d01ubc) (see Section 2.3.1.1 in How to Use the NAG Library and its Documentation).

5: **istop** – Integer * *Input/Output*

On entry: **istop** $= 0$.

On exit: you may set **istop** to a negative number if at any time it is impossible to evaluate the function $f(x)$. In this case nag_quad_1d_inf_exp_wt (d01ubc) halts with **fail** set to the value of **istop** and the value returned in **ans** will be that of a non-signalling NaN.

2: **n** – Integer *Input*

On entry: **n** specifies the number of weights and abscissae to be used.

Constraint: **n** $= 1$, 2, 3, 4, 5, 10, 15, 20 or 25.

3: **ans** – double * *Output*

On exit: if **fail.code** $= 0$, **ans** contains an approximation to the integral. Otherwise, **ans** will be a non-signalling NaN.

4: **comm** – Nag_Comm *

The NAG communication argument (see Section 2.3.1.1 in How to Use the NAG Library and its Documentation).

5: **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, **n** $=$ ⟨*value*⟩.
Constraint: $1 \leq$ **n** ≤ 25.

On entry, **n** $=$ ⟨*value*⟩.
n is not one of the allowed values.

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.

NE_USER_STOP

The user has halted the calculation.

7 Accuracy

The weights and abscissae have been calculated using quadruple precision arithmetic.

8 Parallelism and Performance

nag_quad_1d_inf_exp_wt (d01ubc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example computes an approximation to $\int_0^\infty \exp\left(-x^2\right)x\ dx$.

10.1 Program Text

```
/* nag_quad_1d_inf_exp_wt (d01ubc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */
#include <math.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagd01.h>

#ifdef __cplusplus
extern "C"
{
#endif
  static void NAG_CALL fun(const double x[], double f[], const Integer n,
                           Nag_Comm *comm, Integer *istop);
#ifdef __cplusplus
}
#endif

int main(void)
{
  static double ruser[2] = { -1.0, -1.0 };
  /* Scalars */
  Integer       exit_status = 0;
```

```
  double        ans;
  Integer       n;
  /* Nag Types */
  Nag_Comm      comm;
  NagError      fail;

  printf("nag_quad_1d_inf_exp_wt (d01ubc) Example Program Results\n\n");

  /* For communication with user-supplied functions: */
  comm.user = ruser;

  INIT_FAIL(fail);

  n = 10;

  /* Compute the one-dimensional integral, from zero to infinity,
   * of a function weighted by exp(-x*x), using
   * nag_quad_1d_inf_exp_wt (d01ubc).
   */
  nag_quad_1d_inf_exp_wt(fun, n, &ans, &comm, &fail);
  switch (fail.code) {
  case NE_NOERROR:
    {
      /* The definite integral has been estimated. */
      printf("Number of abscissae used  = %5"NAG_IFMT"\n", n);
      printf("approximation to integral = %10.5f\n", ans);
      break;
    }
  case NE_USER_STOP:
    {
      /* A requested exit was made in fun. */
      printf("A stop was requested in fun by setting istop < 0\n\n");
      printf("%s\n", fail.message);
      exit_status++;
      break;
    }
  default:
    {
      /* A solution could not be calculated due to an illegal parameter
       * or other failure.
       */
      printf("%s\n", fail.message);
      exit_status++;
    }
  }
  return exit_status;
}

static void NAG_CALL fun(const double x[], double f[], const Integer n,
                         Nag_Comm *comm, Integer *istop)
{
  Integer i;

  if (comm->user[0] == -1.0) {
    printf("(User-supplied callback fun, first invocation.)\n");
    comm->user[0] = 0.0;
  }
  for (i=0; i<n; i++) {
    f[i] = x[i];
  }
  *istop = 0;
}
```

10.2 Program Data

None.

10.3 Program Results

```
nag_quad_1d_inf_exp_wt (d01ubc) Example Program Results

(User-supplied callback fun, first invocation.)
Number of abscissae used  =    10
approximation to integral =    0.50000
```