NAG C Library Function Document

nag_1d_quad_gen (d01ajc)

1 Purpose

nag_1d_quad_gen (d01ajc) is a general purpose integrator which calculates an approximation to the
integral of a function \( f(x) \) over a finite interval \([a, b]\):

\[
I = \int_a^b f(x) \, dx.
\]

2 Specification

#include <nag.h>
#include <nagd01.h>

void nag_1d_quad_gen (double (*f)(double x),
          double a, double b, double epsabs, double epsrel,
          Integer max_num_subint, double *result, double *abserr,
          Nag_QuadProgress *qp, NagError *fail)

3 Description

This function is based upon the QUADPACK routine QAGS (Piessens et al. (1983)). It is an adaptive
function, using the Gauss 10-point and Kronrod 21-point rules. The algorithm, described by De Doncker
(1978), incorporates a global acceptance criterion (as defined by Malcolm and Simpson (1976)) together
with the \( c \)-algorithm (Wynn (1956)) to perform extrapolation. The local error estimation is described by
Piessens et al. (1983).

The function is suitable as a general purpose integrator, and can be used when the integrand has
singularities, especially when these are of algebraic or logarithmic type.

This function requires the user to supply a function to evaluate the integrand at a single point.

4 Parameters

1: \( f \) – function supplied by user

The function \( f \), supplied by the user, must return the value of the integrand \( f \) at a given point.

The specification of \( f \) is:

\[
double f(double x)
\]

1: \( x \) – double

On entry: the point at which the integrand \( f \) must be evaluated.

2: \( a \) – double

On entry: the lower limit of integration, \( a \).

3: \( b \) – double

On entry: the upper limit of integration, \( b \). It is not necessary that \( a < b \).

4: \( \text{epsabs} \) – double

On entry: the absolute accuracy required. If \( \text{epsabs} \) is negative, the absolute value is used. See
Section 6.1.
5: \textbf{epsrel} – double \hspace{1cm} \textit{Input}
\textit{On entry:} the relative accuracy required. If \textbf{epsrel} is negative, the absolute value is used. See Section 6.1.

6: \textbf{max\_num\_subint} – Integer \hspace{1cm} \textit{Input}
\textit{On entry:} the upper bound on the number of sub-intervals into which the interval of integration may be divided by the function. The more difficult the integrand, the larger \textbf{max\_num\_subint} should be.

\textit{Suggested values:} a value in the range 200 to 500 is adequate for most problems.
\textit{Constraint:} \textbf{max\_num\_subint} \geq 1.

7: \textbf{result} – double * \hspace{1cm} \textit{Output}
\textit{On exit:} the approximation to the integral \( I \).

8: \textbf{abserr} – double * \hspace{1cm} \textit{Output}
\textit{On exit:} an estimate of the modulus of the absolute error, which should be an upper bound for \( |I - \textbf{result}| \).

9: \textbf{qp} – Nag\_QuadProgress *

Pointer to structure of type \textbf{Nag\_QuadProgress} with the following members:

\textbf{num\_subint} – Integer \hspace{1cm} \textit{Output}
\textit{On exit:} the actual number of sub-intervals used.

\textbf{fun\_count} – Integer \hspace{1cm} \textit{Output}
\textit{On exit:} the number of function evaluations performed by \textbf{nag\_1d\_quad\_gen}.

\textbf{sub\_int\_beg\_pts} – double * \hspace{1cm} \textit{Output}
\textbf{sub\_int\_end\_pts} – double * \hspace{1cm} \textit{Output}
\textbf{sub\_int\_result} – double * \hspace{1cm} \textit{Output}
\textbf{sub\_int\_error} – double * \hspace{1cm} \textit{Output}

\textit{On exit:} these pointers are allocated memory internally with \textbf{max\_num\_subint} elements. If an error exit other than \textbf{NE\_INT\_ARG\_LT} or \textbf{NE\_ALLOC\_FAIL} occurs, these arrays will contain information which may be useful. For details, see Section 6.

Before a subsequent call to \textbf{nag\_1d\_quad\_gen} is made, or when the information contained in these arrays is no longer useful, the user should free the storage allocated by these pointers using the NAG macro \textbf{NAG\_FREE}.

10: \textbf{fail} – NagError * \hspace{1cm} \textit{Input/Output}

The NAG error parameter (see the Essential Introduction).

Users are recommended to declare and initialise \textbf{fail} and set \textbf{fail\_print} = TRUE for this function.

\section{Error Indicators and Warnings}

\textbf{NE\_INT\_ARG\_LT}

\textit{On entry,} \textbf{max\_num\_subint} must not be less than 1: \textbf{max\_num\_subint} = <\textit{value}>

\textbf{NE\_ALLOC\_FAIL}

Memory allocation failed.
NE_QUAD_MAX_SUBDIV

The maximum number of subdivisions has been reached: \(\text{max\_num\_subint} = \text{<value>}.\)

The maximum number of subdivisions has been reached without the accuracy requirements being achieved. Look at the integrand in order to determine the integration difficulties. If the position of a local difficulty within the interval can be determined (e.g., a singularity of the integrand or its derivative, a peak, a discontinuity, etc.) you will probably gain from splitting up the interval at this point and calling the integrator on the sub-intervals. If necessary, another integrator, which is designed for handling the type of difficulty involved, must be used. Alternatively, consider relaxing the accuracy requirements specified by \(\text{epsabs}\) and \(\text{epsrel}\), or increasing the value of \(\text{max\_num\_subint}\).

NE_QUAD_ROUNDOff_TOL

Round-off error prevents the requested tolerance from being achieved: \(\text{epsabs} = \text{<value>}, \text{epsrel} = \text{<value>}.\)

The error may be underestimated. Consider relaxing the accuracy requirements specified by \(\text{epsabs}\) and \(\text{epsrel}\).

NE_QUAD_BAD_SUBDIV

Extremely bad integrand behaviour occurs around the sub-interval \((\text{<value>}, \text{<value>}).\)

The same advice applies as in the case of \text{NE\_QUAD\_MAX\_SUBDIV}.

NE_QUAD_ROUNDOff_EXTRAPL

Round-off error is detected during extrapolation.

The requested tolerance cannot be achieved, because the extrapolation does not increase the accuracy satisfactorily; the returned result is the best that can be obtained.

The same advice applies as in the case of \text{NE\_QUAD\_MAX\_SUBDIV}.

NE_QUAD_NO_CONV

The integral is probably divergent or slowly convergent.

Please note that divergence can occur with any error exit other than \text{NE\_INT\_ARG\_LT} and \text{NE\_ALLOC\_FAIL}.

6 Further Comments

The time taken by \text{nag\_1d\_quad\_gen} depends on the integrand and the accuracy required.

If the function fails with an error exit other than \text{NE\_INT\_ARG\_LT} or \text{NE\_ALLOC\_FAIL}, then the user may wish to examine the contents of the structure \text{qp}. These contain the end-points of the sub-intervals used by \text{nag\_1d\_quad\_gen} along with the integral contributions and error estimates over the sub-intervals.

Specifically, for \(i = 1, 2, \ldots, n\), let \(r_i\) denote the approximation to the value of the integral over the sub-interval \([a_i, b_i]\) in the partition of \([a, b]\) and \(e_i\) be the corresponding absolute error estimate.

Then, \(\int_a^b f(x) \, dx \simeq r_i\) and \(\text{result} = \sum_{i=1}^n r_i\), unless the function terminates while testing for divergence of the integral (see Section 3.4.3 of Piessens \textit{et al.} (1983)). In this case, \(\text{result}\) (and \text{abserr}) are taken to be the values returned from the extrapolation process. The value of \(n\) is returned in \text{num\_subint}, and the values \(a_i, b_i, r_i\) and \(e_i\) are stored in the structure \text{qp} as:

\[
\begin{align*}
  a_i &= \text{sub\_int\_beg\_pts}[i - 1], \\
  b_i &= \text{sub\_int\_end\_pts}[i - 1], \\
  r_i &= \text{sub\_int\_result}[i - 1] \text{ and} \\
  e_i &= \text{sub\_int\_error}[i - 1].
\end{align*}
\]
6.1 Accuracy
The function cannot guarantee, but in practice usually achieves, the following accuracy:

\[ |I - \text{result}| \leq tol \]

where

\[ tol = \max\{|\text{epsabs}|, |\text{epsrel}| \times |I|\} \]

and \text{epsabs} and \text{epsrel} are user-specified absolute and relative error tolerances. Moreover it returns the quantity \text{abserr} which, in normal circumstances, satisfies

\[ |I - \text{result}| \leq \text{abserr} \leq tol. \]

6.2 References
Wynn P (1956) On a device for computing the \(e_m(S_n)\) transformation Math. Tables Aids Comput. 10 91–96

7 See Also
nag_1d_quad_osc (d01akc)
nag_1d_quad_brkpts (d01alc)

8 Example
To compute

\[ \int_0^{2\pi} \frac{x \sin(30x)}{\sqrt{1 - (\frac{x}{\pi})^2}} \, dx. \]

8.1 Program Text

/* nag_1d_quad_gen(d01ajc) Example Program */
/* Copyright 1991 Numerical Algorithms Group. */
/* Mark 2, 1991. */
/* Mark 6 revised, 2000. */
/
#include <nag.h>
#include <stdio.h>
#include <math.h>
#include <nagd01.h>
#include <nagx01.h>

static double f(double x);

main()
```c
{  
double a, b;
double epsabs, abserr, epsrel, result;
Nag_QuadProgress qp;
Integer max_num_subint;
static NagError fail;
double pi = X01AAC;

Vprintf("d01ajc Example Program Results\n");
epsabs = 0.0;
epsrel = 0.0001;
a = 0.0;
b = pi*2.0;
max_num_subint = 200;
d01ajc(f, a, b, epsabs, epsrel, max_num_subint, &result, &abserr,
   &qp, &fail);
Vprintf("a - lower limit of integration = %10.4f\n", a);
Vprintf("b - upper limit of integration = %10.4f\n", b);
Vprintf("epsabs - absolute accuracy requested = %9.2e\n", epsabs);
Vprintf("epsrel - relative accuracy requested = %9.2e\n", epsrel);
if (fail.code != NE_NOERROR)
   Vprintf("%s\n", fail.message);
if (fail.code != NE_INT_ARG_LT && fail.code != NE_ALLOC_FAIL)
{
   Vprintf("result - approximation to the integral = %9.5f\n", result);
   Vprintf("abserr - estimate of the absolute error = %9.2e\n", abserr);
   Vprintf("qp.fun_count - number of function evaluations = %4ld\n",
      qp.fun_count);
   Vprintf("qp.num_subint - number of subintervals used = %4ld\n",
      qp.num_subint);
   /* Free memory used by qp */
   NAG_FREE(qp.sub_int_beg_pts);
   NAG_FREE(qp.sub_int_end_pts);
   NAG_FREE(qp.sub_int_result);
   NAG_FREE(qp.sub_int_error);
   exit(EXIT_SUCCESS);
}
else
   exit(EXIT_FAILURE);
}

static double f(double x)
{
   double pi = X01AAC;
   return (x*sin(x*30.0)/sqrt(1.0-x*x/(pi*pi*4.0)));
}

8.2 Program Data

None.

[NP3491/6]
```
8.3 Program Results

d0lajc Example Program Results
a - lower limit of integration = 0.0000
b - upper limit of integration = 6.2832
epsabs - absolute accuracy requested = 0.00e+00
epsrel - relative accuracy requested = 1.00e-04

result - approximation to the integral = -2.54326
abserr - estimate of the absolute error = 1.28e-05
qp.fun_count - number of function evaluations = 777
qp.num_subint - number of subintervals used = 19