NAME

qglqf − Gauss-Laguerre logarithmic Quadrature with Function values

SYNOPSIS

Fortran (77, 90, 95, HPF):

f77 *[flags] file(s) . . . -L/usr/local/lib -lgjl*

```
SUBROUTINE qglqf(x, w, wxm1, y, z, alpha, nquad, ierr)
INTEGER        ierr,     nquad
REAL*16        alpha,    w(*),     wxm1(*),    x(*)
REAL*16        y(*),     z(*)
```

C (K&R, 89, 99), C++ (98):

cc *[flags] -I/usr/local/include file(s) . . . -L/usr/local/lib -lgjl*

Use

#include <gjl.h>

to get this prototype:

```
void qglqf(fortran_quadruple_precision x_[],
    fortran_quadruple_precision w_[],
    fortran_quadruple_precision wxm1_[],
    fortran_quadruple_precision y_[],
    fortran_quadruple_precision z_[],
    const fortran_quadruple_precision * alpha_,
    const fortran_integer * nquad_,
    fortran_integer * ierr_);
```

NB: The definition of C/C++ data types **fortran_** *xxx*, and the mapping of Fortran external names to C/C++ external names, is handled by the C/C++ header file. That way, the same function or subroutine name can be used in C, C++, and Fortran code, independent of compiler conventions for mangling of external names in these programming languages.

DESCRIPTION

Compute the nodes and weights for the evaluation of the integral

\int_0^\infty x^\alpha e^{−x} \ln(x) f(x) dx

as the quadrature sum

\sum_{i=1}^N[W_i(\alpha)(x_i(\alpha) − 1)f(x_i(\alpha)) − Z_i(\alpha)f(y_i(\alpha))]

The nonlogarithmic integral

\int_0^\infty x^\alpha e^{−x} f(x) dx

can be computed from the quadrature sum

\sum_{i=1}^N[W_i(\alpha) f(x_i(\alpha))]

The quadrature is exact to machine precision for f(x) of polynomial order less than or equal to 2***nquad** − 2 (logarithmic) or 2***nquad** − 1 (nonlogarithmic).

This form of the quadrature requires only values of the function at 2***nquad** points. For a faster, and slightly more accurate, quadrature that requires values of the function and its derivative at **nquad** points, see the companion routine, qglqfd().

On entry:

alpha Power of x in the integrand (**alpha** > −1).

nquad Number of quadrature points to compute. It must be less than the limit MAXPTS defined in the header file, *maxpts.inc*. The default value chosen there should be large enough for any realistic application.

On return:

x(1. .**nquad**)	Nodes of the first part of the quadrature, denoted x_i(\alpha) above.
w(1. .**nquad**)	Weights of the first part of the quadrature, denoted W_i(\alpha) above.
wxm1(1. .**nquad**)	Scaled weights of the first part of the quadrature, **wxm1**(i) = **w**(i)*(x(i) − 1).
y(1. .**nquad**)	Nodes of the second part of the quadrature, denoted y_i(\alpha) above.
z(1. .**nquad**)	Weights of the second part of the quadrature, denoted −Z_i(\alpha) above.
ierr	Error indicator: = 0 (success), 1 (eigensolution could not be obtained), 2 (destructive overflow), 3 (**nquad** out of range), 4 (**alpha** out of range).

The logarithmic integral can then be computed by code like this:

```
          sum = 0.0q+00
          do 10 i = 1,nquad
              sum = sum + wxm1(i)*f(x(i)) - z(i)*f(y(i))
       10 continue
```

The nonlogarithmic integral can be computed by:

```
          sum = 0.0q+00
          do 20 i = 1,nquad
              sum = sum + w(i)*f(x(i))
       20 continue
```

SEE ALSO

qglqfd(3), **qglqrc**(3).

AUTHORS

The algorithms and code are described in detail in the paper

Fast Gaussian Quadrature for Two Classes of Logarithmic Weight Functions

in ACM Transactions on Mathematical Software, Volume ??, Number ??, Pages ????--???? and ????--????, 20xx, by

Nelson H. F. Beebe
Center for Scientific Computing
University of Utah
Department of Mathematics, 110 LCB
155 S 1400 E RM 233
Salt Lake City, UT 84112-0090
Tel: +1 801 581 5254
FAX: +1 801 581 4148
Email: beebe@math.utah.edu, beebe@acm.org, beebe@computer.org
WWW URL: http://www.math.utah.edu/~beebe

and

James S. Ball
University of Utah
Department of Physics
Salt Lake City, UT 84112-0830
USA
Tel: +1 801 581 8397
FAX: +1 801 581 6256
Email: ball@physics.utah.edu
WWW URL: http://www.physics.utah.edu/people/faculty/ball.html