Package 'custom.gauss.quad'

May 8, 2026

Title Custom Made Gauss Quadrature Nodes and Weights

Version 1.0.0

Maintainer Paul Kabaila <p.kabaila@latrobe.edu.au>

Description Use the high-precision arithmetic provided by the R package 'Rmpfr' to compute a custom-made Gauss quadrature nodes and weights, with up to 33 nodes, using a moment-based method via moment determinants. Paul Kabaila (2022) <doi:10.48550/arXiv.2211.04729>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

Imports Rmpfr

NeedsCompilation no

Author Paul Kabaila [aut, cre] (ORCID: <https://orcid.org/0000-0002-9205-668X>)

Repository CRAN

Date/Publication 2022-11-16 14:50:05 UTC

Contents

`custom` *Custom-Made Gauss Quadrature Nodes and Weights*

Description

For the nonnegative weight function specified by `which.f` and given number `n` of nodes, the function `custom` computes the Gauss quadrature nodes `asNumeric.nodes` and corresponding weights `asNumeric.weights` which are double precision vectors.

Usage

```
custom(which.f, n)
```

Arguments

which.f	a list specifying the nonnegative integrable weight function f, with the following three components: (i) name (in the form of a character string), (ii) support specified by a 2-vector of the endpoints of the interval, (iii) parameter vector when f belongs to a family of weight functions and is specified by the value of this parameter vector (if f is already fully specified then the parameter vector is set to NULL)
n	number of Gauss quadrature nodes

Details

Suppose that we wish to evaluate

$$\int_{-\infty}^{\infty} g(x)f(x)dx,$$

where f is a specified nonnegative integrable weight function. The Gauss quadrature approximation to this integral has the form

$$\sum_{i=1}^{n} \lambda_i \, g(\tau_i),$$

where $\tau_1, \ldots, \tau_n$ are called the nodes and $\lambda_1, \ldots, \lambda_n$ are called the corresponding weights. This approximation is exact whenever g is a polynomial of degree less than or equal to $2n-1$.

If f takes a form that leads to Gauss quadrature rules with nodes that are the roots of classical orthogonal polynomials of a continuous variable then these rules (such as Gauss Legendre, Gauss Hermite and Gauss Laguerre) are readily accessible to statisticians via R packages such as statmod. If, however, f does not take one of these particular forms then the Gauss quadrature nodes and weights need to be custom-made.

custom computes the Gauss quadrature nodes and weights, for given n, using a user-supplied formula for the r'th moment

$$\int_{-\infty}^{\infty} x^r f(x)dx$$

for all nonnegative integers r. This formula must be inserted by the user into the code for the function moments and must be able to be computed to an arbitrary number of bits (nbits) of precision using the R package Rmpfr.

To obtain some assurance that the Gauss quadrature nodes and weights are computed to sufficient precision for subsequent double precision computations in R, these nodes and weights are computed for the increasing numbers of bits of precision given in the 5-vector nbits.vec and the results compared. This comparison results in the criteria max.abs.diffs.nodes, sum.abs.diffs.weights, L.nodes and L.weights described in detail by Kabaila (2022). The execution times for various parts of the code are stored in mat.timings whose components are described by Kabaila (2022).

list.Gauss.nodes[[i]] and list.Gauss.weights[[i]] are the n-vectors of Gauss quadrature nodes and weights, respectively, computed using nbits.vec[i] bits of precision (i=1,...,5).

The most accurate approximations to the Gauss quadrature nodes and weights are `list.Gauss.nodes[[5]]` and `list.Gauss.weights[[5]]`. These are converted to double precision by applying the `asNumeric` function from the `R` package `Rmpfr`, resulting in `asNumeric.nodes` and `asNumeric.weights`, respectively.

Value

A list with the following elements: `which.f`, `n`, `nbits.vec`, `list.Gauss.nodes`, `list.Gauss.weights`, `mat.timings`, `max.abs.diffs.nodes`, `sum.abs.diffs.weights`, `L.nodes`, `L.weights`, `asNumeric.nodes`, `asNumeric.weights`.

References

Kabaila, P. (2022) Custom-made Gauss quadrature for statisticians. arXiv:2211.04729

See Also

`moments`

Examples

```
# Suppose that the weight function f is the probability density
# function of a random variable with the same probability
# distribution as R divided by the square root of m, where R has a
# chi distribution with m degrees of freedom.
# Also suppose that we wish to compute the Gauss quadrature nodes
# and weights, for number of nodes n = 5, when the parameter m = 160.
# The r th moment can be computed to an arbitrary number of bits of
# precision using the R package Rmpfr. We describe the weight function
# f using the following R commands:

m <- 160
which.f <- list(name="scaled.chi.pdf", support=c(0, Inf),
parameters=m)

# Here, "scaled.chi.pdf" is the name (a character string) that we
# have given to the weight function f. The R function moments includes
# the code needed to compute the r th moment to an arbitrary number
# of bits of precision using the R package Rmpfr.
# We compute the Gauss quadrature node and weight, for the toy example
# with number of nodes n=1, using the following R commands:

n <- 1
gauss.list <- custom(which.f, n)
old <- options(digits = 17)
gauss.list$asNumeric.nodes
gauss.list$asNumeric.weights
options(old)

# These commands take less than 1 second to run. The resulting
# of node and corresponding weight in double precision are:
# > gauss.list$asNumeric.nodes
```

```
# [1] 0.99843873022375829
# > gauss.list$asNumeric.weights
# [1] 1

# The computation times for number of nodes n=5, 17 and 33 are roughly
# 160 seconds, 31 minutes and 5 hours,respectively.
#
# We compute the Gauss quadrature nodes and weights, for number of
# nodes n=5, using the following R commands:

n <- 5
gauss.list <- custom(which.f, n)
old <- options(digits = 17)
gauss.list$asNumeric.nodes
gauss.list$asNumeric.weights
options(old)

# These commands take roughly 3 minutes to run. The resulting vectors
# of nodes and corresponding weights in double precision are:
# > gauss.list$asNumeric.nodes
# [1] 0.84746499810651410 0.92785998378868118 1.00262691212158761
# [4] 1.07930375924992528 1.16628363226782716
# > gauss.list$asNumeric.weights
# [1] 0.0144433732487188448 0.2483585328946608384 0.5305446123744097520
# [4] 0.1977278905956056654 0.0089255908866048821
```

`moments` *Moments Computed in Multiple Precision Using the Package Rmpfr*

Description

This module computes the `r`'th moment

$$\int_{-\infty}^{\infty} x^r f(x) dx,$$

where f is the weight function (specified by the list `which.f`), for any nonnegative integer `r` using `nbits` bits of precision for its computation, via the `R` package `Rmpfr`.

Usage

```
moments(which.f, r, nbits)
```

Arguments

`which.f`	a list specifying the nonnegative integrable weight function f, with the following 3 components: (i) name (in the form of a character string), (ii) support specified by a 2-vector of the endpoints of the interval, (iii) parameter vector when f

belongs to a family of weight functions and is specified by the value of this parameter vector (if f is already fully specified then the parameter vector is set to NULL)

r — nonnegative integer, specifying that it is the r'th moment for the weight function f that is to be computed

nbits — number of bits in the multiple precision numbers used by the R package Rmpfr to carry out the computation of the r'th moment

Details

Suppose, for example, that we wish to find the Gauss quadrature nodes and weights for the weight function f that is the probability density function of a random variable with the same distribution as $R/m^{1/2}$ where R has a χ_m distribution (i.e. R^2 has a χ^2_m distribution). In this case, the r'th moment is

$$\int_{-\infty}^{\infty} x^r f(x)dx = \left(\frac{2}{m}\right)^{r/2} \frac{\Gamma((r+m)/2)}{\Gamma(m/2)},$$

which can be computed to an arbitrary number of bits of precision nbits using the R package Rmpfr. In this case, we specify this weight function f by first assigning the value of m and then using the R command

```
which.f <- list(name="scaled.chi.pdf", support=c(0, Inf), parameters=m)
```

The code within the function moments used to compute the r'th moment, to an arbitrary number of bits of precision nbits using the package Rmpfr, is listed in the Examples section.

Value

The r'th moment with number of bits of precision nbits used in its computation, via the R package Rmpfr

See Also

custom

Examples

```
# The code for the function moments must include a section
# that computes the r th moment to an arbitrary number of bits
# of precision nbits using the R package Rmpfr for the particular
# weight function f of interest.
# Suppose that the weight function f is the probability density
# function of a random variable with the same probability
# distribution as R divided by the square root of m, where R has a
# chi distribution with m degrees of freedom.
# The code for the function moments includes the following:
#
#    if (which.f$name == "scaled.chi.pdf"){
#    m <- which.f$parameters
#    if (r == 0){
#    return(mpfr(1, nbits))
#    }
```

```
#    mp.2 <- mpfr(2, nbits)
#    mp.r <- mpfr(r, nbits)
#    mp.m <- mpfr(m, nbits)
#    term1 <- (mp.r/ mp.2) * log(mp.2 / mp.m)
#    term2 <- lgamma((mp.r + mp.m) / mp.2)
#    term3 <- lgamma(mp.m / mp.2)
#    return(exp(term1 + term2 - term3))
#    }
```

Index