Package: gaussquad (via r-universe)

September 2, 2024

Version 1.0-3

Date 2022-06-14

Title Collection of Functions for Gaussian Quadrature

Author Frederick Novomestky <fnovomes@poly.edu>

Maintainer Frederick Novomestky <fnovomes@poly.edu>

Depends R (>= 2.0.1), orthopolynom

Imports polynom

Description A collection of functions to perform Gaussian quadrature with different weight functions corresponding to the orthogonal polynomials in package orthopolynom. Examples verify the orthogonality and inner products of the polynomials.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-14 08:42:49 UTC

Contents

`chebyshev.c.quadrature`

Perform Gauss Chebyshev quadrature

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Chebyshev C polynomials.

Usage

```
chebyshev.c.quadrature(functn, rule, lower = -2, upper = 2,
weighted = TRUE, ...)
```

Arguments

`functn`	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
`rule`	a data frame containing the order n Chebyshev quadrature rule

lower	numeric value for the lower limit of the integral with a default value of -2
upper	numeric value for the upper limit of the integral with a default value of +2
weighted	boolean value which if true causes the Chebyshev weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n Chebyshev polynomial, weight function and interval $[-2, 2]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Chebyshev quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

chebyshev.c.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Chebyshev C polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 16.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 16.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 16
###
### maximum order plus 1
```

```
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- chebyshev.c.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
```

```
}
###
### construct a list of the Chebyshev C orthogonal polynomials
###
p.list <- chebyshev.c.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- chebyshev.c.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( chebyshev.c.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Chebyshev C orthonormal polynomials
###
p.list <- chebyshev.c.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
```

```
###
rules <- chebyshev.c.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`chebyshev.c.quadrature.rules`

Create list of Chebyshev quadrature rules

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Chebyshev C polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
chebyshev.c.quadrature.rules(n,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

`1`	Quadrature rule data frame for the order 1 Chebyshev polynomial
`2`	Quadrature rule data frame for the order 2 Chebyshev polynomial
...	
`n`	Quadrature rule data frame for the order n Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, chebyshev.c.quadrature

Examples

```
###
### construct the list of quadrature rules for
### the Chebyshev orthogonal polynomials
### of orders 1 to 5
###
orthogonal.rules <- chebyshev.c.quadrature.rules( 5 )
print( orthogonal.rules )
###
### construct the list of quadrature rules for
### the Chebyshev orthonormal polynomials
### of orders 1 to 5
###
orthonormal.rules <- chebyshev.c.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

---

chebyshev.s.quadrature

Perform Gauss Chebyshev quadrature

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Chebyshev S polynomials.

Usage

```
chebyshev.s.quadrature(functn, rule, lower = -2, upper = 2,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n Chebyshev quadrature rule
lower	numeric value for the lower limit of the integral with a default value of -2
upper	numeric value for the upper limit of the integral with a default value of +2
weighted	boolean value which if true causes the Chebyshev weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n Chebyshev polynomial, weight function and interval $[-2, 2]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Chebyshev quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

chebyshev.s.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Chebyshev S polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 16.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 16.  the resultant matrix should be an identity matrix
```

```
###
###
### set the value for the maximum polynomial order
###
    n <- 16
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
```

```
        row.column.integral <- chebyshev.s.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Chebyshev S orthogonal polynomials
###
p.list <- chebyshev.s.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- chebyshev.s.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( chebyshev.s.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Chebyshev S orthonormal polynomials
###
p.list <- chebyshev.s.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
```

```
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- chebyshev.s.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`chebyshev.s.quadrature.rules` *Create list of Chebyshev quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Chebyshev S polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
chebyshev.s.quadrature.rules(n,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 Chebyshev polynomial
2	Quadrature rule data frame for the order 2 Chebyshev polynomial
...	
n	Quadrature rule data frame for the order n Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, chebyshev.s.quadrature

Examples

```
###
### generate a list of quadrature rules for
### the Chebyshev S orthogonal polynomials
### for orders 1 to 5
###
orthogonal.rules <- chebyshev.s.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate a list of quadrature rules for
### the Chebyshev S orthogonal polynomials
### for orders 1 to 5
###
orthonormal.rules <- chebyshev.s.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

chebyshev.t.quadrature

Perform Gauss Chebyshev quadrature

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Chebyshev T polynomials.

Usage

```
chebyshev.t.quadrature(functn, rule, lower = -1, upper = 1,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n Chebyshev quadrature rule
lower	numeric value for the lower limit of the integral with a default value of -1
upper	numeric value for the upper limit of the integral with a default value of +1
weighted	boolean value which if true causes the Chebyshev weight function to be included in the integrand
...	other arguments passed to the given function

Details

The rule argument corresponds to an order n Chebyshev polynomial, weight function and interval $[-1, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Chebyshev quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

chebyshev.t.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Chebyshev T polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 20.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 20.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 20
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
```

```
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- chebyshev.t.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Chebyshev T orthogonal polynomials
###
p.list <- chebyshev.t.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- chebyshev.t.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
```

```
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( chebyshev.t.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Chebyshev T orthonormal polynomials
###
p.list <- chebyshev.t.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- chebyshev.t.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

---

chebyshev.t.quadrature.rules

Create list of Chebyshev quadrature rules

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Chebyshev T polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
chebyshev.t.quadrature.rules(n,normalized=FALSE)
```

Arguments

n	integer value for the highest order
normalized	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name x contains the roots or zeros and the column with name w contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 Chebyshev polynomial
2	Quadrature rule data frame for the order 2 Chebyshev polynomial
...	
n	Quadrature rule data frame for the order n Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, chebyshev.t.quadrature

Examples

```
###
### generate the list of quadrature rules for
### the orthogonal Chebyshev polynomials
### for orders 1 to 5
###
orthogonal.rules <- chebyshev.t.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate the list of quadrature rules for
### the orthonormal Chebyshev polynomials
### for orders 1 to 5
###
orthonormal.rules <- chebyshev.t.quadrature.rules( 5, normalized=TRUE )
print( orthonormal.rules )
```

`chebyshev.u.quadrature`

Perform Gauss Chebyshev quadrature

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Chebyshev U polynomials.

Usage

```
chebyshev.u.quadrature(functn, rule, lower = -1, upper = 1,
weighted = TRUE, ...)
```

Arguments

`functn`	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
`rule`	a data frame containing the order n Chebyshev quadrature rule
`lower`	numeric value for the lower limit of the integral with a default value of -1
`upper`	numeric value for the upper limit of the integral with a default value of +1
`weighted`	a boolean value which if true causes the Chebyshev weight function to be included in the integrand
`...`	other arguments passed to the give function

Details

The rule argument corresponds to an order n Chebyshev polynomial, weight function and interval $[-1, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Chebyshev quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

chebyshev.u.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Chebyshev U polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 20.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 20.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 20
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
```

```
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- chebyshev.u.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Chebyshev U orthogonal polynomials
###
p.list <- chebyshev.u.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
```

```
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- chebyshev.u.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( chebyshev.u.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Chebyshev U orthonormal polynomials
###
p.list <- chebyshev.u.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- chebyshev.u.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
```

```
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`chebyshev.u.quadrature.rules` *Create list of Chebyshev quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Chebyshev U polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
chebyshev.u.quadrature.rules(n,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

`1`	Quadrature rule data frame for the order 1 Chebyshev polynomial
`2`	Quadrature rule data frame for the order 2 Chebyshev polynomial
...	
`n`	Quadrature rule data frame for the order n Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, chebyshev.u.quadrature

Examples

```
###
### generate the list of quadrature rules for
### the Chebyshev U orthogonal polynomials for
### orders 1 to 5
###
orthogonal.rules <- chebyshev.u.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate the list of quadrature rules for
### the Chebyshev U orthonormal polynomials for
### orders 1 to 5
###
orthonormal.rules <- chebyshev.u.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

---

gegenbauer.quadrature *Perform Gauss Gegenbauer quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Gegenbauer polynomials.

Usage

```
gegenbauer.quadrature(functn, rule, alpha = 0, lower = -1, upper = 1,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n Gegenbauer quadrature rule
alpha	numeric value for the Gegenbauer polynomial parameter
lower	numeric value for the lower limit of the integral with a default value of -1
upper	numeric value the upper limit of the integral with a default value of 1
weighted	boolean value which if true causes the Gegenbauer weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n Gegenbauer polynomial, weight function and interval $[-1, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Gegenbauer quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

gegenbauer.quadrature.rules, ultraspherical.quadrature

Examples

```
###
### this example evaluates the quadrature function for
### the Gegenbauer polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 16.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
```

```
### of all pairs of orthonormal polynomials from order 0
### to order 16.  the resultant matrix should be an identity matrix
###
###
### set the polynomial parameter
###
alpha <- 0.25
###
### set the value for the maximum polynomial order
###
    n <- 16
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
```

```
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- gegenbauer.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule, alpha )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Gegenbauer orthogonal polynomials
###
p.list <- gegenbauer.polynomials( n, alpha )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- gegenbauer.quadrature.rules( np1, alpha )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( gegenbauer.inner.products( n,alpha ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=6 ) )
print( apply( p.p.inner.products, 2, round, digits=6 ) )
###
### construct a list of the Gegenbauer orthonormal polynomials
###
p.list <- gegenbauer.polynomials( n, alpha, TRUE )
```

```
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- gegenbauer.quadrature.rules( np1, alpha, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=6 ) )
```

`gegenbauer.quadrature.rules` *Create list of Gegenbauer quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Gegenbauer polynomials for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
gegenbauer.quadrature.rules(n,alpha,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`alpha`	polynomial parameter
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 Gegenbauer polynomial
2	Quadrature rule data frame for the order 2 Gegenbauer polynomial
...	
n	Quadrature rule data frame for the order n Gegenbauer polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

`quadrature.rules`, `gegenbauer.quadrature`

Examples

```
###
### generate the list of quadrature rule data frames for
### the orthogonal Gegenbauer polynomials
### of orders 1 to 5
### polynomial parameter alpha is 1.0
###
orthogonal.rules <- gegenbauer.quadrature.rules( 5, 1 )
print( orthogonal.rules )
###
### generate the list of quadrature rule data frames for
### the orthonormal Gegenbauer polynomials
### of orders 1 to 5
### polynomial parameter alpha is 1.0
###
orthonormal.rules <- gegenbauer.quadrature.rules( 5, 1, TRUE )
print( orthonormal.rules )
```

`ghermite.h.quadrature` *Perform generalized Gauss Hermite quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for generalized Hermite polynomials.

Usage

```
ghermite.h.quadrature(functn, rule, mu = 0, lower = -Inf, upper = Inf,
weighted = TRUE, ...)
```

Arguments

`functn`	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
`rule`	a data frame containing the order n generalized Hermite quadrature rule
`mu`	numeric value for the parameter for the generalized Hermite polynomials
`lower`	numeric value for the lower limit of the integral with a default value of $-\infty$
`upper`	numeric value for the upper limit of the integral with a default value of $+\infty$
`weighted`	a boolean value which if true causes the Hermite weight function to be included in the integrand
`...`	other arguments passed to the give function

Details

The rule argument corresponds to an order n generalized Hermite polynomial, weight function and interval $(-\infty, \infty)$. The lower and upper limits of the integral must be infinite.

Value

The value of definite integral evaluated using Gauss Hermite quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

hermite.h.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the generalized Hermite H polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 10.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 10.  the resultant matrix should be an identity matrix
###
###
### set the polynomial parameter
###
    mu <- 1
###
### set the value for the maximum polynomial order
###
    n <- 10
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
```

```
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- ghermite.h.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule, mu )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the generalized Hermite H orthogonal polynomials
###
p.list <- ghermite.h.polynomials( n, mu )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
```

```
### get the rule table for the order np1 polynomial
###
rules <- ghermite.h.quadrature.rules( np1, mu )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( ghermite.h.inner.products( n, mu ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Hermite H orthonormal polynomials
###
p.list <- ghermite.h.polynomials( n, mu, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- ghermite.h.quadrature.rules( np1, mu, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`ghermite.h.quadrature.rules` *Create list of generalized Hermite quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the generalized Hermite polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
ghermite.h.quadrature.rules(n, mu, normalized=FALSE)
```

Arguments

`n`	integer value for the highest integer order
`mu`	numeric value for the parameter of the generalized Hermite polynomial
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

`1`	Quadrature rule data frame for the order 1 generalized Hermite polynomial
`2`	Quadrature rule data frame for the order 2 generalized Hermite polynomial
...	
`n`	Quadrature rule data frame for the order n generalized Hermite polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, schebyshev.t.quadrature

Examples

```
###
### generate a list of quadrature rule data frames for
### the generalized orthogonal Hermite polynomial
### of orders 1 to 5.
### polynomial parameter mu is 1.0
###
orthogonal.rules <- ghermite.h.quadrature.rules( 5, 1 )
print( orthogonal.rules )
###
### generate a list of quadrature rule data frames for
### the generalized orthonormal Hermite polynomial
### of orders 1 to 5.
### polynomial parameter mu is 1.0
###
orthonormal.rules <- ghermite.h.quadrature.rules( 5, 1, TRUE )
print( orthonormal.rules )
```

---

glaguerre.quadrature *Perform Gauss Laguerre quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for generalized Laguerre polynomials.

Usage

```
glaguerre.quadrature(functn, rule, alpha = 0, lower = 0, upper = Inf,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n generalized Laguerre quadrature rule
alpha	numeric value for the generalized Laguerre polynomial parameter
lower	numeric value for the lower limit of the integral with a default value of 0
upper	numeric value for the upper limit of the integral with a default value of Inf
weighted	a boolean value which if true causes the generalized Laguerre weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n generalized Laguerre polynomial, weight function and interval $[0, \infty)$. The one limit of the integral must be finite and the other must be infinite.

Value

The value of definite integral evaluated using Gauss Laguerre quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

glaguerre.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the generalized Laguerre polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 12.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 12.  the resultant matrix should be an identity matrix
###
###
### set the polynomial parameter
###
alpha <- 1
###
### set the value for the maximum polynomial order
###
    n <- 12
###
### maximum order plus 1
###
    np1 <- n + 1
###
```

```
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- glaguerre.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule, alpha )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the generalized Laguerre orthogonal polynomials
```

```
###
p.list <- glaguerre.polynomials( n, alpha )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- glaguerre.quadrature.rules( np1, alpha )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( glaguerre.inner.products( n,alpha ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=6 ) )
print( apply( p.p.inner.products, 2, round, digits=6 ) )
###
### construct a list of theg generalized Laguerre orthonormal polynomials
###
p.list <- glaguerre.polynomials( n, alpha, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- glaguerre.quadrature.rules( np1, alpha, TRUE )
order.np1.rule <- rules[[np1]]
```

```
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=6 ) )
```

`glaguerre.quadrature.rules` *Create list of generalized Laguerre quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the generalized Laguerre polynomials for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
glaguerre.quadrature.rules(n,alpha,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`alpha`	numeric value for the polynomial parameter
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a quadrature rule data frame

`1`	Quadrature rule for the order 1 generalized Laguerre polynomial
`2`	Quadrature rule for the order 2 generalized Laguerre polynomial
...	
`n`	Quadrature rule for the order n generalized Laguerre polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, glaguerre.quadrature

Examples

```
###
### generate a list of quadrature rule data frames for
### the generalized orthogonal Laguerre polynomials
### of orders 1 to 5
### polynomial parameter is 1.0
###
orthogonal.rules <- glaguerre.quadrature.rules( 5, 1 )
print( orthogonal.rules )
###
### generate a list of quadrature rule data frames for
### the generalized orthonormal Laguerre polynomials
### of orders 1 to 5
### polynomial parameter is 1.0
###
orthonormal.rules <- glaguerre.quadrature.rules( 5, 1, TRUE )
print( orthonormal.rules )
```

---

hermite.h.quadrature *Perform Gauss Hermite quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Hermite H polynomials.

Usage

```
hermite.h.quadrature(functn, rule, lower = -Inf, upper = Inf,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n Hermite quadrature rule
lower	numeric value for the lower limit of the integral with a default value of $-\infty$
upper	numeric value for the upper limit of the integral with a default value of $+\infty$
weighted	boolean value which if true causes the Hermite weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n Hermite polynomial, weight function and interval $(-\infty, \infty)$ The lower and upper limits of the integral must be infinite.

Value

The value of definite integral evaluated using Gauss Hermite quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

hermite.h.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Hermite H polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 8.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 8.  the resultant matrix should be an identity matrix
```

```
###
###
### set the value for the maximum polynomial order
###
    n <- 8
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
```

```
        row.column.integral <- hermite.h.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Hermite H orthogonal polynomials
###
p.list <- hermite.h.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- hermite.h.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( hermite.h.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Hermite H orthonormal polynomials
###
p.list <- hermite.h.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
```

```
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- hermite.h.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`hermite.h.quadrature.rules`

Create list of Hermite quadrature rules

Description

This function returns a list with n elements containing the order n quadrature rule data frame for the Hermite polynomials for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
hermite.h.quadrature.rules(n,normalized=FALSE)
```

Arguments

`n`	integer highest order
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 Hermite polynomial
2	Quadrature rule data frame for the order 2 Hermite polynomial
...	
n	Quadrature rule data frame for the order n Hermite polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, hermite.h.quadrature

Examples

```
###
### generate the list of quadrature rules for
### the Hermite orthogonal polynomials
### of orders 1 to 5
###
orthogonal.rules <- hermite.h.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate the list of quadrature rules for
### the Hermite orthonormal polynomials
### of orders 1 to 5
###
orthonormal.rules <- hermite.h.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

`hermite.he.quadrature` *Perform Gauss Hermite quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for hermite He polynomials.

Usage

```
hermite.he.quadrature(functn, rule, lower = -Inf, upper = Inf,
weighted = TRUE, ...)
```

Arguments

`functn`	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
`rule`	a data frame containing the order n Hermite quadrature rule
`lower`	the lower limit of the integral with a default value of $-\infty$
`upper`	the upper limit of the integral with a default value of $+\infty$
`weighted`	a boolean value which if true causes the Hermite weight function to be included in the integrand
`...`	other arguments passed to the give function

Details

The rule argument corresponds to an order n Hermite polynomial, weight function and interval $(-\infty, \infty)$ The lower and upper limits of the integral must be infinite.

Value

The value of definite integral evaluated using Gauss Hermite quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

hermite.h.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the scaled Hermite He polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 16.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 16.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 12
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
```

```
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- hermite.h.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the scaled Hermite He orthogonal polynomials
###
p.list <- hermite.he.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- hermite.he.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
```

```
###
p.p.inner.products <- diag( hermite.he.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the scaled Hermite H orthonormal polynomials
###
p.list <- hermite.he.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- hermite.he.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`hermite.he.quadrature.rules` *Create list of Hermite quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Hermite polynomials for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
hermite.he.quadrature.rules(n,normalized=FALSE)
```

Arguments

n	integer value for the highest order
normalized	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name x contains the roots or zeros and the column with name w contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 Hermite polynomial
2	Quadrature rule data frame for the order 2 Hermite polynomial
...	
n	Quadrature rule data frame for the order n Hermite polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, hermite.he.quadrature

Examples

```
###
### generate a list of quadrature rule frames for
### the Hermite orthogonal polynomials
### of orders 1 to 5
###
orthogonal.rules <- hermite.he.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate a list of quadrature rule frames for
```

```
### the Hermite orthonormal polynomials
### of orders 1 to 5
###
orthonormal.rules <- hermite.he.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

`jacobi.g.quadrature` *Perform Gauss Jacobi quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Jacobi G polynomials.

Usage

```
jacobi.g.quadrature(functn, rule,  p = 1, q = 1, lower = 0, upper = 1,
weighted = TRUE, ...)
```

Arguments

`functn`	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
`rule`	a data frame containing the order n ultraspherical quadrature rule
`p`	numeric value for the first Jacobi polynomial parameter
`q`	numeric value for the second Jacobi polynomial parameter
`lower`	numeric value for the lower limit of the integral with a default value of 0
`upper`	numeric value for the upper limit of the integral with a default value of 1
`weighted`	boolean value which if true causes the ultraspherical weight function to be included in the integrand
`...`	other arguments passed to the give function

Details

The rule argument corresponds to an order n Jacobi polynomial, weight function and interval $[0, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Jacobi quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

jacobi.p.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Jacobi G polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 10.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 10.  the resultant matrix should be an identity matrix
###
###
### set the polynomial parameter
###
p <- 3
q <- 2
###
### set the value for the maximum polynomial order
###
    n <- 10
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
```

```
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- jacobi.g.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule, p, q )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Jacobi G orthogonal polynomials
###
p.list <- jacobi.g.polynomials( n, p, q )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
```

```
###
### get the rule table for the order np1 polynomial
###
rules <- jacobi.g.quadrature.rules( np1, p, q )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( jacobi.g.inner.products( n,p, q ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Jacobi G orthonormal polynomials
###
p.list <- jacobi.g.polynomials( n, p, q, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- jacobi.g.quadrature.rules( np1, p, q, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`jacobi.g.quadrature.rules` *Create list of Jacobi quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Jacobi G polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
jacobi.g.quadrature.rules(n,p,q,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`p`	numeric value for the first polynomial parameter
`q`	numeric value for the second polynomial parameter
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

`1`	Quadrature rule data frame for the order 1 Jacobi polynomial
`2`	Quadrature rule data frame for the order 2 Jacobi polynomial
...	
`n`	Quadrature rule data frame for the order n Jacobi polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules

Examples

```
###
### generate the list of quadrature rule data frames for
### the orthogonal Jacobi G polynomial
### of orders 1 to 5
### parameter p is 3 and parameter q is 2
###
orthogonal.rules <- jacobi.g.quadrature.rules( 5, 3, 2 )
print( orthogonal.rules )
###
### generate the list of quadrature rule data frames for
### the orthonormal Jacobi G polynomial
### of orders 1 to 5
### parameter p is 3 and parameter q is 2
###
orthonormal.rules <- jacobi.g.quadrature.rules( 5, 3, 2, TRUE )
print( orthonormal.rules )
```

---

jacobi.p.quadrature *Perform Gauss Jacobi quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Jacobi P polynomials.

Usage

```
jacobi.p.quadrature(functn, rule, alpha = 0, beta = 0, lower = -1, upper = 1,
weighted = TRUE,  ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n ultraspherical quadrature rule
alpha	numeric value for the first Jacobi polynomial parameter
beta	numeric value for the second Jacobi polynomial parameter
lower	numeric value for the lower limit of the integral with a default value of -1
upper	numeric value for the upper limit of the integral with a default value of 1
weighted	a boolean value which if true causes the ultraspherical weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n Jacobi polynomial, weight function and interval $[-1, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Jacobi quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

jacobi.p.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Jacobi P polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 12.  the results are compared to
### the diagonal matrix of the inner products for the
```

```
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 12.  the resultant matrix should be an identity matrix
###
###
### set the polynomial parameter
###
alpha <- 1
beta <- -0.6
###
### set the value for the maximum polynomial order
###
    n <- 12
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
```

```
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- jacobi.p.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule, alpha, beta )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Jacobi P orthogonal polynomials
###
p.list <- jacobi.p.polynomials( n, alpha, beta )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- jacobi.p.quadrature.rules( np1, alpha, beta )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( jacobi.p.inner.products( n,alpha, beta ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=6 ) )
print( apply( p.p.inner.products, 2, round, digits=6 ) )
###
### construct a list of the Jacobi P orthonormal polynomials
```

```
###
p.list <- jacobi.p.polynomials( n, alpha, beta, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- jacobi.p.quadrature.rules( np1, alpha, beta, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=6 ) )
```

`jacobi.p.quadrature.rules` *Create list of Jacobi quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Jacobi P polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
jacobi.p.quadrature.rules(n,alpha,beta,normalized=FALSE)
```

Arguments

n	integer value for the highest order
alpha	numeric value for the first polynomial parameter
beta	numeric value for the second polynomial parameter

`normalized` boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a quadrature rule data frame

`1`	Quadrature rule for the order 1 Jacobi polynomial
`2`	Quadrature rule for the order 2 Jacobi polynomial
...	
`n`	Quadrature rule for the order n Jacobi polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

`quadrature.rules`

Examples

```
###
### construct the list of quadrature rules for
### the Jacobi orthogonal polynomials
### of orders 1 to 5
### alpha = 3 and beta = 2
###
orthogonal.rules <- jacobi.p.quadrature.rules( 5, 3, 2 )
print( orthogonal.rules )
###
### construct the list of quadrature rules for
### the Jacobi orthonormal polynomials
### of orders 1 to 5
```

```
### alpha = 3 and beta = 2
###
orthonormal.rules <- jacobi.p.quadrature.rules( 5, 3, 2, TRUE )
print( orthonormal.rules )
```

`laguerre.quadrature` *Perform Gauss Laguerre quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Laguerre polynomials.

Usage

```
laguerre.quadrature(functn, rule, lower = 0, upper = Inf, weighted = TRUE, ...)
```

Arguments

`functn`	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
`rule`	a data frame containing the order n Laguerre quadrature rule
`lower`	numeric value for the lower limit of the integral with a default value of 0
`upper`	numeric value for the upper limit of the integral with a default value of Inf
`weighted`	boolean value which if true causes the Laguerre weight function to be included in the integrand
`...`	other arguments passed to the give function

Details

The rule argument corresponds to an order n Laguerre polynomial, weight function and interval $[0, \infty]$. The one limit of the integral must be finite and the other must be infinite.

Value

The value of definite integral evaluated using Gauss Laguerre quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

laguerre.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Laguerre polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 12.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 12.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 12
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
```

```
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- laguerre.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Laguerre orthogonal polynomials
###
p.list <- laguerre.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- laguerre.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
```

```
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( laguerre.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=6 ) )
print( apply( p.p.inner.products, 2, round, digits=6 ) )
###
### construct a list of the Laguerre orthonormal polynomials
###
p.list <- laguerre.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- laguerre.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=6 ) )
```

---

laguerre.quadrature.rules

Create list of Laguerre quadrature rules

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Laguerre polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
laguerre.quadrature.rules(n,normalized=FALSE)
```

Arguments

n	integer value for the highest order
normalized	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name x contains the roots or zeros and the column with name w contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 Laguerre polynomial
2	Quadrature rule data frame for the order 2 Laguerre polynomial
...	
n	Quadrature rule data frame for the order n Laguerre polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, laguerre.quadrature

Examples

```
###
### generate a list of the quadrature rule frames for
### the orthogonal Laguerre polynomials
### of orders 1 to 5
###
orthogonal.rules <- laguerre.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate a list of the quadrature rule frames for
### the orthonormal Laguerre polynomials
### of orders 1 to 5
###
orthonormal.rules <- laguerre.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

---

legendre.quadrature *Perform Gauss Legendre quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for Legendre polynomials.

Usage

```
legendre.quadrature(functn, rule, lower=-1, upper=1,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n Legendre quadrature rule
lower	numeric value for the lower limit of the integral
upper	numeric value for the upper limit of the integral
weighted	boolean value which if true causes the Legendre weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n Legendre polynomial, weight function and interval $[-1, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Legendre quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

Szego, G., 1939. *Orthogonal Polynomials*, 23, American Mathematical Society Colloquium Publications, Providence, RI.

See Also

legendre.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Legendre polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 20.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 20.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 20
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
```

```
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- legendre.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the Legendre orthogonal polynomials
###
p.list <- legendre.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
```

```
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- legendre.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( legendre.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Legendre orthonormal polynomials
###
p.list <- legendre.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- legendre.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
```

```
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`legendre.quadrature.rules` *Create list of Legendre quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the Legendre polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
legendre.quadrature.rules(n,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

`1`	Quadrature rule data frame for the order 1 Legendre polynomial
`2`	Quadrature rule data frame for the order 2 Legendre polynomial
...	
`n`	Quadrature rule data frame for the order n Legendre polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz and Stegun (1968), Press et. al. (1992)

See Also

quadrature.rules

Examples

```
###
### generate a list of quadrature rule frames for
### the orthogonal Legendre polynomials
### of orders 1 to 5
###
orthogonal.rules <- legendre.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate a list of quadrature rule frames for
### the orthonormal Legendre polynomials
### of orders 1 to 5
###
orthonormal.rules <- legendre.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

---

quadrature.rule.table *Generate the quadrature rule table*

Description

This function returns a column-named data frame with the given quadrature rules combined into a single object

Usage

```
quadrature.rule.table(rules)
```

Arguments

rules	A list of quadrature rule data frames

Details

The column-named data frame has four columns. The first column is called d and it contains the degree of the orthogonal polynomial. The second column is called i and it contains the index of the rule for the given polynomial order. The third column is called x and it contains the abscissas, roots or zeros of the polynomial of given order. The fourth column is called w and it contains the weights associated with the abscissas.

Value

A data frame with the quadrature rules.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules

Examples

```
###
### generate the list of quadrature rules for
### the Chebyshev T polynomials
### of orders 1 to 5
###
rules <- chebyshev.t.quadrature.rules( 5 )
###
### construct the rule table
###
rule.table <- quadrature.rule.table( rules )
print( rule.table )
```

---

quadrature.rules *Create a list of quadrature rule data frames*

Description

This function returns a list with n elements containing the order k quadrature rule data frames for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
quadrature.rules(recurrences, inner.products)
```

Arguments

recurrences	a data frame with the recurrence parameters of a particular orthogonal polynomial
inner.products	a numeric vector of normed squared values of the orthogonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values for the quadrature rule based on an order k orthgonal polynomial.

Value

A list with n elements each of which is a quadrature rule data frame

1	Quadrature rule for the order 1 orthogonal polynomial
2	Quadrature rule for the order 2 orthogonal polynomial
...	
n	Quadrature rule for the order n orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramwitz and Stegun (1968)

Examples

```
###
### get recurrences the Chebyshev T orthgonal polynomials
### of orders 0 to 6\5
###
recurrences <- chebyshev.t.recurrences( 5 )
###
### get the inner products of these polynomials
###
inner.products <- chebyshev.t.inner.products( 5 )
###
### get the quadrature rules
###
rules <- quadrature.rules( recurrences, inner.products )
print( rules )
```

---

schebyshev.t.quadrature

Perform Gauss shifted Chebyshev quadrature

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for shifted Chebyshev T polynomials.

Usage

```
schebyshev.t.quadrature(functn, rule, lower = 0, upper = 1,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n shifted Chebyshev quadrature rule
lower	numeric value for the lower limit of the integral with a default value of 0
upper	numeric value for the upper limit of the integral with a default value of 1
weighted	boolean value which if true causes the Chebyshev weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n shifted Chebyshev polynomial, weight function and interval $[0, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Chebyshev quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

schebyshev.t.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the shifted Chebyshev T polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 10.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 10.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 10
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
```

```
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- schebyshev.t.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the shifted Chebyshev T orthogonal polynomials
###
p.list <- schebyshev.t.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- schebyshev.t.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( schebyshev.t.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=4 ) )
```

```
print( apply( p.p.inner.products, 2, round, digits=4 ) )
###
### construct a list of the shifted Chebyshev T orthonormal polynomials
###
p.list <- schebyshev.t.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- schebyshev.t.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=4 ) )
```

`schebyshev.t.quadrature.rules` *Create list of shifted Chebyshev quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the shifted Chebyshev T polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
schebyshev.t.quadrature.rules(n,normalized=FALSE)
```

Arguments

n	integer value for the highest order
normalized	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name x contains the roots or zeros and the column with name w contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 shifted Chebyshev polynomial
2	Quadrature rule data frame for the order 2 shifted Chebyshev polynomial
...	
n	Quadrature rule data frame for the order n shifted Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, schebyshev.t.quadrature

Examples

```
###
### construct a list of quadrature rule data frames for
### the shifted orthogonal Chebyshev T polynomials
### of orders 1 to 5
###
orthogonal.rules <- schebyshev.t.quadrature.rules( 5 )
print( orthogonal.rules )
###
### construct a list of quadrature rule data frames for
```

```
### the shifted orthonormal Chebyshev T polynomials
### of orders 1 to 5
###
orthonormal.rules <- schebyshev.t.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

---

schebyshev.u.quadrature

Perform Gauss shifted Chebyshev quadrature

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for shifted Chebyshev U polynomials.

Usage

```
schebyshev.u.quadrature(functn, rule, lower = 0, upper = 1,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n shifted Chebyshev quadrature rule
lower	numeric value for the lower limit of the integral with a default value of 0
upper	numeric value for the upper limit of the integral with a default value of 1
weighted	a boolean value which if true causes the Chebyshev weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n shifted Chebyshev polynomial, weight function and interval $[0, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss shifted Chebyshev quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

schebyshev.t.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the shifted Chebyshev U polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 10.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 10.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 10
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
```

```
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- schebyshev.u.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the shifted Chebyshev U orthogonal polynomials
###
p.list <- schebyshev.u.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- schebyshev.u.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
```

```
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( schebyshev.u.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=4 ) )
print( apply( p.p.inner.products, 2, round, digits=4 ) )
###
### construct a list of the shifted Chebyshev U orthonormal polynomials
###
p.list <- schebyshev.u.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- schebyshev.u.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=4 ) )
```

---

schebyshev.u.quadrature.rules

Create list of shifted Chebyshev quadrature rules

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the shifted Chebyshev U polynomial of the first kind for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
schebyshev.u.quadrature.rules(n,normalized=FALSE)
```

Arguments

n	integer value for the highest order
normalized	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name x contains the roots or zeros and the column with name w contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 shifted Chebyshev polynomial
2	Quadrature rule data frame for the order 2 shifted Chebyshev polynomial
...	
n	Quadrature rule data frame for the order n shifted Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, schebyshev.u.quadrature

Examples

```
###
### generate the quadrature rules for
### the shifted Chebyshev U orthogonal polynomials
### of orders 1 to 5
###
orthogonal.rules <- schebyshev.u.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate the quadrature rules for
### the shifted Chebyshev U orthonormal polynomials
### of orders 1 to 5
###
orthonormal.rules <- schebyshev.u.quadrature.rules( 5 )
print( orthonormal.rules )
```

---

slegendre.quadrature *Perform Gauss shifted Legendre quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for shifted Legendre polynomials.

Usage

```
slegendre.quadrature(functn, rule, lower = 0, upper = 1,
weighted = TRUE, ...)
```

Arguments

functn	an R FUnction which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n shifted Legendre quadrature rule
lower	a numeric value for the lower limit of the integral with a default value of 0
upper	a numeric value for the upper limit of the integral with a default value of 1
weighted	a boolean value which if true causes the shifted Legendre weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n shifted Legendre polynomial, weight function and interval $[0, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Legendre quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

slegendre.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the shifted Legendre polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 10.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 10.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 10
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
```

```
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- slegendre.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the shifted Legendre orthogonal polynomials
###
p.list <- slegendre.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
```

```
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- slegendre.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( slegendre.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the shifted Legendre orthonormal polynomials
###
p.list <- slegendre.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- slegendre.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
```

```
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`slegendre.quadrature.rules` *Create list of shifted Legendre quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the shifted Legendre polynomials for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
slegendre.quadrature.rules(n,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

`1`	Quadrature rule data frame for the order 1 shifted Legendre polynomial
`2`	Quadrature rule data frame for the order 2 shifted Legendre polynomial
...	
`n`	Quadrature rule data frame for the order n shifted Legendre polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, slegendre.quadrature

Examples

```
###
### generate the list of shifted Legendre quadrature rules
### for orders 1 to 5 for the orthogonal polynomials
###
orthogonal.rules <- slegendre.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate the list of shifted Legendre quadrature rules
### for orders 1 to 5 for the orthonormal polynomials
###
orthonormal.rules <- slegendre.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

---

spherical.quadrature *Perform Gauss spherical quadrature*

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for spherical polynomials.

Usage

```
spherical.quadrature(functn, rule, lower = -1, upper = 1,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n spherical quadrature rule

lower	numeric value for the lower limit of the integral with a default value of -1
upper	numeric value for the upper limit of the integral with a default value of +1
weighted	boolean value which if true causes the Legendre weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n spherical polynomial, weight function and interval $[-1, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Legendre quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

spherical.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the Chebyshev C polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 16.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
### of all pairs of orthonormal polynomials from order 0
### to order 16.  the resultant matrix should be an identity matrix
###
###
### set the value for the maximum polynomial order
###
    n <- 16
###
### maximum order plus 1
```

```
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- chebyshev.c.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
```

```
}
###
### construct a list of the Chebyshev C orthogonal polynomials
###
p.list <- chebyshev.c.polynomials( n )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- chebyshev.c.quadrature.rules( np1 )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( chebyshev.c.inner.products( n ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=5 ) )
print( apply( p.p.inner.products, 2, round, digits=5 ) )
###
### construct a list of the Chebyshev C orthonormal polynomials
###
p.list <- chebyshev.c.polynomials( n, TRUE )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
```

```
###
rules <- chebyshev.c.quadrature.rules( np1, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=5 ) )
```

`spherical.quadrature.rules` *Create list of shperical quadrature rules*

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the shperical polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
spherical.quadrature.rules(n,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

`1`	Quadrature rule data frame for the order 1 spherical polynomial
`2`	Quadrature rule data frame for the order 2 spherical polynomial
...	
`n`	Quadrature rule data frame for the order n spherical polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

quadrature.rules, spherical.quadrature

Examples

```
###
### generate a list of quadrature rule data frames for
### the orthogonal spherical polynomials
### of orders 1 to 5
###
orthogonal.rules <- spherical.quadrature.rules( 5 )
print( orthogonal.rules )
###
### generate a list of quadrature rule data frames for
### the orthonormal spherical polynomials
### of orders 1 to 5
###
orthonormal.rules <- spherical.quadrature.rules( 5, TRUE )
print( orthonormal.rules )
```

---

ultraspherical.quadrature

Perform Gauss ultraspherical quadrature

Description

This function evaluates the integral of the given function between the lower and upper limits using the weight and abscissa values specified in the rule data frame. The quadrature formula uses the weight function for ultraspherical polynomials.

Usage

```
ultraspherical.quadrature(functn, rule, alpha = 0, lower = -1, upper = 1,
weighted = TRUE, ...)
```

Arguments

functn	an R function which should take a numeric argument x and possibly some parameters. The function returns a numerical vector value for the given argument x.
rule	a data frame containing the order n ultraspherical quadrature rule
alpha	a numeric value for the ultraspherical polynomial parameter
lower	a numeric value for the lower limit of the integral with a default value of -1
upper	a numeric value for the upper limit of the integral with a default value of 1
weighted	a boolean value which if true causes the ultraspherical weight function to be included in the integrand
...	other arguments passed to the give function

Details

The rule argument corresponds to an order n ultraspherical polynomial, weight function and interval $[-1, 1]$. The lower and upper limits of the integral must be finite.

Value

The value of definite integral evaluated using Gauss Gegenbauer quadrature

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

ultraspherical.quadrature.rules

Examples

```
###
### this example evaluates the quadrature function for
### the ultraspherical polynomials.  it computes the integral
### of the product for all pairs of orthogonal polynomials
### from order 0 to order 16.  the results are compared to
### the diagonal matrix of the inner products for the
### polynomials.  it also computes the integral of the product
```

```
### of all pairs of orthonormal polynomials from order 0
### to order 16.  the resultant matrix should be an identity matrix
###
###
### set the polynomial parameter
###
alpha <- 0.25
###
### set the value for the maximum polynomial order
###
    n <- 16
###
### maximum order plus 1
###
    np1 <- n + 1
###
### function to construct the polynomial products by column
###
by.column.products <- function( c, p.list, p.p.list )
{
###
### function to construct the polynomial products by row
###
    by.row.products <- function( r, c, p.list )
    {
        row.column.product <- p.list[[r]] * p.list[[c]]
        return (row.column.product )
    }
    np1 <- length( p.list )
    row.list <- lapply( 1:np1, by.row.products, c, p.list )
    return( row.list )
}
###
### function construct the polynomial functions by column
###
by.column.functions <- function( c, p.p.products )
{
###
### function to construct the polynomial functions by row
###
    by.row.functions <- function( r, c, p.p.products )
    {
        row.column.function <- as.function( p.p.products[[r]][[c]] )
        return( row.column.function )
    }
    np1 <- length( p.p.products[[1]] )
    row.list <- lapply( 1:np1, by.row.functions, c, p.p.products )
    return( row.list )
}
###
### function to compute the integral of the polynomials by column
###
by.column.integrals <- function( c, p.p.functions )
```

```
{
###
### function to compute the integral of the polynomials by row
###
    by.row.integrals <- function( r, c, p.p.functions )
    {
        row.column.integral <- ultraspherical.quadrature(
            p.p.functions[[r]][[c]], order.np1.rule, alpha )
        return( row.column.integral )
    }
    np1 <- length( p.p.functions[[1]] )
    row.vector <- sapply( 1:np1, by.row.integrals, c, p.p.functions )
    return( row.vector )
}
###
### construct a list of the ultraspherical orthogonal polynomials
###
p.list <- ultraspherical.polynomials( n, alpha )
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- ultraspherical.quadrature.rules( np1, alpha )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### construct the diagonal matrix with the inner products
### of the orthogonal polynomials on the diagonal
###
p.p.inner.products <- diag( ultraspherical.inner.products( n,alpha ) )
print( "Integral of cross products for the orthogonal polynomials " )
print( apply( p.p.integrals, 2, round, digits=6 ) )
print( apply( p.p.inner.products, 2, round, digits=6 ) )
###
### construct a list of the ultraspherical orthonormal polynomials
###
p.list <- ultraspherical.polynomials( n, alpha, TRUE )
```

```
###
### construct the two dimensional list of pair products
### of polynomials
###
p.p.products <- lapply( 1:np1, by.column.products, p.list )
###
### compute the two dimensional list of functions
### corresponding to the polynomial products in
### the two dimensional list p.p.products
###
p.p.functions <- lapply( 1:np1, by.column.functions, p.p.products )
###
### get the rule table for the order np1 polynomial
###
rules <- gegenbauer.quadrature.rules( np1, alpha, TRUE )
order.np1.rule <- rules[[np1]]
###
### construct the square symmetric matrix containing
### the definite integrals over the default limits
### corresponding to the two dimensional list of
### polynomial functions
###
p.p.integrals <- sapply( 1:np1, by.column.integrals, p.p.functions )
###
### display the matrix of integrals
###
print( "Integral of cross products for the orthonormal polynomials " )
print(apply( p.p.integrals, 2, round, digits=6 ) )
```

`ultraspherical.quadrature.rules`

Create list of ultraspherical quadrature rules

Description

This function returns a list with n elements containing the order k quadrature rule data frame for the ultraspherical polynomial for orders $k = 1,\ 2,\ \ldots,\ n$.

Usage

```
ultraspherical.quadrature.rules(n,alpha,normalized=FALSE)
```

Arguments

`n`	integer value for the highest order
`alpha`	numeric value for the polynomial parameter
`normalized`	boolean value. if TRUE rules are for orthonormal polynomials, otherwise they are for orthgonal polynomials

Details

An order k quadrature data frame is a named data frame that contains the roots and abscissa values of the corresponding order k orthogonal polynomial. The column with name `x` contains the roots or zeros and the column with name `w` contains the weights.

Value

A list with n elements each of which is a data frame

1	Quadrature rule data frame for the order 1 ultraspherical polynomial
2	Quadrature rule data frame for the order 2 ultraspherical polynomial
...	
n	Quadrature rule data frame for the order n ultraspherical polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

Abramowitz, M. and I. A. Stegun, 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K.

Stroud, A. H., and D. Secrest, 1966. *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, NJ.

See Also

`quadrature.rules`, `ultraspherical.quadrature`

Examples

```
###
### generate a list of quadrature rules for
### the orthogonal ultraspherical polynomial
### of orders 1 to 5
### the polynomial parameter value alpha is 1.0
###
orthogonal.rules <- ultraspherical.quadrature.rules( 5, 1 )
print( orthogonal.rules )
###
### generate a list of quadrature rules for
### the orthonormal ultraspherical polynomial
### of orders 1 to 5
### the polynomial parameter value alpha is 1.0
###
orthonormal.rules <- ultraspherical.quadrature.rules( 5, 1, TRUE )
print( orthonormal.rules )
```

Index