Package ‘orthopolynom’

February 20, 2015

Version 1.0-5
Date 2013-02-03
Title Collection of functions for orthogonal and orthonormal polynomials
Author Frederick Novomestky <fnovomes@poly.edu>
Maintainer Frederick Novomestky <fnovomes@poly.edu>
Depends R (>= 2.0.1), polynom
Description A collection of functions to construct sets of orthogonal polynomials and their recurrence relations. Additional functions are provided to calculate the derivative, integral, value and roots of lists of polynomial objects.
License GPL (>= 2)
Repository CRAN
Date/Publication 2013-02-04 07:42:24
NeedsCompilation no

R topics documented:

  chebyshev.c.inner.products                      3
  chebyshev.c.polynomials                        5
  chebyshev.c.recurrences                       6
  chebyshev.c.weight                             7
  chebyshev.s.inner.products                    8
  chebyshev.s.polynomials                       10
  chebyshev.s.recurrences                       11
  chebyshev.s.weight                             12
  chebyshev.t.inner.products                    13
  chebyshev.t.polynomials                       15
  chebyshev.t.recurrences                       16
  chebyshev.t.weight                             17
  chebyshev.u.inner.products                    18
  chebyshev.u.polynomials                       20
**R topics documented:**

- chebyshev.u.recurrences ........................................ 21
- chebyshev.u.weight ........................................... 22
- gegenbauer.inner.products .................................... 23
- gegenbauer.polynomials ....................................... 25
- gegenbauer.recurrences ....................................... 26
- gegenbauer.weight ........................................... 27
- hermite.h.inner.products ..................................... 28
- hermite.h.polynomials ........................................ 30
- hermite.h.recurrences ....................................... 31
- hermite.h.weight ............................................ 33
- laguerre.inner.products ...................................... 60
- laguerre.polynomials ......................................... 61
- laguerre.recurrences ........................................ 62
- laguerre.weight ............................................. 63
- legendre.inner.products ..................................... 64
- legendre.polynomials ......................................... 66
- legendre.recurrences ........................................ 67
- legendre.weight ............................................. 68
- lpochhammer .................................................. 69
- monic.polynomial.recurrences ................................ 70
- monic.polynomials ........................................... 71
- orthogonal.polynomials ....................................... 73
- orthonormal.polynomials ..................................... 74
- pochhammer ................................................... 75
- polynomial.coefficients ..................................... 76
- polynomial.derivatives ...................................... 77
- polynomial.functions ......................................... 78
**Description**

This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) Chebyshev polynomial of the first kind, \( C_k(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

**Usage**

```r
chebyshev.c.inner.products(n)
```

**Arguments**

- \( n \) integer value for the highest polynomial order
Details

The formula used to compute the inner products is as follows.

\[ h_n = (C_n | C_n) = \begin{cases} 
4\pi & n \neq 0 \\
8\pi & n = 0 
\end{cases} \]

Value

A vector with \( n + 1 \) elements

1 inner product of order 0 orthogonal polynomial
2 inner product of order 1 orthogonal polynomial
...

\( n+1 \) inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

gegenbauer.inner.products

Examples

```r
###
### generate the inner products vector for the
### C Chebyshev polynomials of orders 0 to 10
###
h <- chebyshev.c.inner.products( 10 )
print( h )
```
Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) Chebyshev polynomials of the first kind, \( C_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

\[
\text{chebyshev.c.polynomials}(n, \text{normalized}=\text{FALSE})
\]

Arguments

- \( n \): integer value for the highest polynomial order
- \( \text{normalized} \): a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function chebyshev.c.recurrences produces a data frame with the recurrence relation parameters for the polynomials. If the \( \text{normalized} \) argument is FALSE, the function orthogonal.polynomials is used to construct the list of orthogonal polynomial objects. Otherwise, the function orthonormal.polynomials is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

- 1: order 0 Chebyshev polynomial
- 2: order 1 Chebyshev polynomial
- ...  
- \( n+1 \): order \( n \) Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Szegö, G., 1939. Orthogonal Polynomials, 23, American Mathematical Society Colloquium Publications, Providence, RI.
chebyshev.c.recurrences

See Also

chebyshev.c.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

###
### generate a list of normalized C Chebyshev polynomials of orders 0 to 10
###
normalized.p.list <- chebyshev.c.polynomials( 10, normalized=TRUE )
print( normalized.p.list )
###
### generate a list of unnormalized C Chebyshev polynomials of orders 0 to 10
###
unnormalized.p.list <- chebyshev.c.polynomials( 10, normalized=FALSE )
print( unnormalized.p.list )

chebyshev.c.recurrences

Recurrence relations for Chebyshev polynomials

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) Chebyshev polynomial of the first kind, \( C_k(x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage

chebyshev.c.recurrences(n, normalized=FALSE)

Arguments

- \( n \) integer value for the highest polynomial order
- \( \text{normalized} \) boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>
References


See Also

chebyshev.c.inner.products

Examples

```r
### generate the recurrences data frame for
### the normalized Chebyshev C polynomials
### of orders 0 to 10.
###
### normalized.r <- chebyshev.c.recurrences(10, normalized=TRUE)
### print( normalized.r )
###
### generate the recurrences data frame for
### the normalized Chebyshev C polynomials
### of orders 0 to 10.
###
### unnormalized.r <- chebyshev.c.recurrences(10, normalized=FALSE)
### print( unnormalized.r )
```

---

**chebyshev.c.weight**

*Weight function for the Chebyshev polynomial*

Description

This function returns the value of the weight function for the order \( k \) Chebyshev polynomial of the first kind, \( C_k(x) \).

Usage

`chebyshev.c.weight(x)`

Arguments

- \( x \) the function argument which can be a vector
Details
The function takes on non-zero values in the interval \((-2, 2)\). The formula used to compute the weight function is as follows.

\[ w(x) = \frac{1}{\sqrt{1-x^2}} \]

Value
The value of the weight function

Author(s)
Frederick Novomestky < fnovomes@poly.edu >

References

Examples
```r
### compute the C Chebyshev weight function for arguments between -3 and 3
###
x <- seq(-3, 3, .01)
y <- chebyshev.c.weight( x )
plot( x, y )
```

Description
This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) Chebyshev polynomial of the second kind, \( S_k(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage
`chebyshev.s.inner.products(n)`
chebyshev.s.inner.products

Arguments

n  integer value for the highest polynomial order

Details

The formula used to compute the inner products is as follows.

\[ h_n = \langle S_n | S_n \rangle = \pi. \]

Value

A vector with \( n + 1 \) elements

1  inner product of order 0 orthogonal polynomial
2  inner product of order 1 orthogonal polynomial
...
\( n+1 \)  inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

```R
###
### generate the inner products vector for the
### S Chebyshev polynomials of orders 0 to 10
###
h <- chebyshev.s.inner.products( 10 )
print( h )
```
chebyshev.s.polynomials

Create list of Chebyshev polynomials

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) Chebyshev polynomials of the second kind, \( S_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

chebyshev.s.polynomials(n, normalized=FALSE)

Arguments

- \( n \) integer value for the highest polynomial order
- \( \text{normalized} \) a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function chebyshev.s.recurrences produces a data frame with the recurrence relation parameters for the polynomials. If the \( \text{normalized} \) argument is FALSE, the function orthogonal.polynomials is used to construct the list of orthogonal polynomial objects. Otherwise, the function orthonormal.polynomials is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

1 order 0 Chebyshev polynomial
2 order 1 Chebyshev polynomial
\ldots
\( n+1 \) order \( n \) Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

See Also

chebyshev.s.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

```r
### generate a list of normalized S Chebyshev polynomials of orders 0 to 10
normalized.p.list <- chebyshev.s.polynomials(10, normalized=TRUE)
print(normalized.p.list)
### generate a list of unnormalized S Chebyshev polynomials of orders 0 to 10
unnormalized.p.list <- chebyshev.s.polynomials(10, normalized=FALSE)
print(unnormalized.p.list)
```

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( e, d, e, \) and \( f \) of the recurrence relations for the order \( k \) Chebyshev polynomial of the second kind, \( S_k(x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage

chebyshev.s.recurrences(n, normalized=FALSE)

Arguments

- \( n \): integer value for the highest polynomial order
- \( \text{normalized} \): boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>
References


See Also

chebyshev.s.inner.products

Examples

```r
### generate the recurrences data frame for
### the normalized Chebyshev S polynomials
### of orders 0 to 10.
###
### normalized.r <- chebyshev.s.recurrences(10, normalized=TRUE)
### print(normalized.r)
###
### generate the recurrences data frame for
### the normalized Chebyshev S polynomials
### of orders 0 to 10.
###
### unnormalized.r <- chebyshev.s.recurrences(10, normalized=FALSE)
### print(unnormalized.r)
```

---

table

### chebyshev.s.weight

Weight function for the Chebyshev polynomial

Description

This function returns the value of the weight function for the order \( k \) Chebyshev polynomial of the second kind, \( S_k(x) \).

Usage

`chebyshev.s.weight(x)`

Arguments

- `x` the function argument which can be a vector
Details
The function takes on non-zero values in the interval $(-2, 2)$. The formula used to compute the weight function is as follows.

\[ w(x) = \sqrt{1 - \frac{x^2}{4}} \]

Value
The value of the weight function.

Author(s)
Frederick Novomestky < fnovomes@poly.edu >

References

Examples
```r
### compute the S Chebyshev weight function for arguments between -2 and 2
### x <- seq(-2, 2, .01)
y <- chebyshev.s.weight( x )
plot( x, y )
```

## chebyshev.t.inner.products

*Inner products of Chebyshev polynomials*

Description
This function returns a vector with $n + 1$ elements containing the inner product of an order $k$ Chebyshev polynomial of the first kind, $T_k(x)$, with itself (i.e. the norm squared) for orders $k = 0, 1, \ldots, n$.

Usage
chebyshev.t.inner.products(n)
Arguments

\( n \)  
integer value for the highest polynomial order

Details

The formula used to compute the inner products is as follows.

\[
h_n = \langle T_n | T_n \rangle = \begin{cases} \frac{\pi}{2} & n \neq 0 \\ \pi & n = 0 \end{cases}
\]

Value

A vector with \( n + 1 \) elements

1  
inner product of order 0 orthogonal polynomial

2  
inner product of order 1 orthogonal polynomial

... 

\( n+1 \)  
inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

```r
###
### generate the inner products vector for the
### T Chybshev polynomials of orders 0 to 10
###
h <- chebyshev.t.inner.products( 10 )
print( h )
```
chebyshev.t.polynomials

Create list of Chebyshev polynomials

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) Chebyshev polynomials of the first kind, \( T_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

chebyshev.t.polynomials(n, normalized=FALSE)

Arguments

\( n \) integer value for the highest polynomial order
\( \text{normalized} \) a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function chebyshev.t.recurrences produces a data frame with the recurrence relation parameters for the polynomials. If the \( \text{normalized} \) argument is FALSE, the function orthogonal.polynomials is used to construct the list of orthogonal polynomial objects. Otherwise, the function orthonormal.polynomials is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

1 order 0 Chebyshev polynomial
2 order 1 Chebyshev polynomial

\ldots

\( n+1 \) order \( n \) Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Szegö, G., 1939. Orthogonal Polynomials, 23, American Mathematical Society Colloquium Publications, Providence, RI.
See Also

chebyshev.u.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

```R
### generate a list of normalized T Chebyshev polynomials of orders 0 to 10
normalized.p.list <- chebyshev.t.polynomials(10, normalized=TRUE)
print(normalized.p.list)

### generate a list of unnormalized T Chebyshev polynomials of orders 0 to 10
unnormalized.p.list <- chebyshev.t.polynomials(10, normalized=FALSE)
print(unnormalized.p.list)
```

Description

This function returns a data frame with $n + 1$ rows and four named columns containing the coefficient vectors $c$, $d$, $e$ and $f$ of the recurrence relations for the order $k$ Chebyshev polynomial of the first kind, $T_k(x)$, for orders $k = 0, 1, \ldots, n$.

Usage

```R
chebyshev.t.recurrences(n, normalized=FALSE)
```

Arguments

- `n` integer value for the highest polynomial order
- `normalized` boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>
chebyshev.t.weight

References


See Also

chebyshev.t.inner.products

Examples

```r
###
### generate the recurrence relations for
### the normalized T Chebyshev polynomials
### of orders 0 to 10
###
normalized.r <- chebyshev.t.recurrences( 10, normalized=TRUE )
print( normalized.r )
###
### generate the recurrence relations for
### the normalized T Chebyshev polynomials
### of orders 0 to 10
###
unnormalized.r <- chebyshev.t.recurrences( 10, normalized=FALSE )
print( unnormalized.r )
```

---

**chebyshev.t.weight**

*Weight function for the Chebyshev polynomial*

**Description**

This function returns the value of the weight function for the order \( k \) Chebyshev polynomial of the first kind, \( T_k(x) \).

**Usage**

`chebyshev.t.weight(x)`

**Arguments**

- `x` the function argument which can be a vector
Details

The function takes on non-zero values in the interval \((-1, 1)\). The formula used to compute the weight function is as follows.

\[ w(x) = \frac{1}{\sqrt{1-x^2}} \]

Value

The value of the weight function.

Author(s)

Frederick Novomestky < fnovomes@poly.edu >

References


Examples

```r
## compute the T Chebyshev function for argument values between -2 and 2
x <- seq(-1, 1, .01)
y <- chebyshev.t.weight(x)
plot(x, y)
```

Description

This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) Chebyshev polynomial of the second kind, \( U_k(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage

`chebyshev.u.inner.products(n)`
Arguments

n integer value for the highest polynomial order

Details

The formula used to compute the inner products is as follows.

\[ h_n = \langle U_n | U_n \rangle = \frac{\pi}{2} \]

Value

A vector with \(n + 1\) elements

1 inner product of order 0 orthogonal polynomial
2 inner product of order 1 orthogonal polynomial
...

\(n+1\) inner product of order \(n\) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

```r
###
### generate the inner products vector for the
### U Chebyshev polynomials of orders 0 to 10
###
h <- chebyshev.u.inner.products( 10 )
print( h )
```
chebyshev.u.polynomials

Create list of Chebyshev polynomials

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) Chebyshev polynomials of the second kind, \( U_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

\[
\text{chebyshev.u.polynomials}(n, \text{normalized}=\text{FALSE})
\]

Arguments

- \( n \): integer value for the highest polynomial order
- \( \text{normalized} \): a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function \text{chebyshev.u.recurrences} produces a data frame with the recurrence relation parameters for the polynomials. If the \( \text{normalized} \) argument is FALSE, the function \text{orthogonal.polynomials} is used to construct the list of orthogonal polynomial objects. Otherwise, the function \text{orthonormal.polynomials} is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

- \( 1 \): order 0 Chebyshev polynomial
- \( 2 \): order 1 Chebyshev polynomial
- \( \ldots \)
- \( n+1 \): order \( n \) Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

See Also

chebyshev.u.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

```r
### generate a list of normalized U Chebyshev polynomials of orders 0 to 10
normalized.p.list <- chebyshev.u.polynomials( 10, normalized=TRUE )
print( normalized.p.list )
### generate a list of unnormalized T Chebyshev polynomials of orders 0 to 10
unnormalized.p.list <- chebyshev.u.polynomials( 10, normalized=FALSE )
print( unnormalized.p.list )
```

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) Chebyshev polynomial of the second kind, \( U_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

chebyshev.u.recurrences(n, normalized=FALSE)

Arguments

- \( n \): integer value for the highest polynomial order
- \( \text{normalized} \): boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>
References


See Also

    chebyshev.u.inner.products

Examples

```r
### generate the recurrence relations for
### the normalized U Chebyshev polynomials
### of orders 0 to 10
###
normalized.r <- chebyshev.u.recurrences( 10, normalized=TRUE )
print( normalized.r )
###
### generate the recurrence relations for
### the unnormalized U Chebyshev polynomials
### of orders 0 to 10
###
unnormalized.r <- chebyshev.u.recurrences( 10, normalized=FALSE )
print( unnormalized.r )
```

---

**chebyshev.u.weight**  
*Weight function for the Chebyshev polynomial*

**Description**

This function returns the value of the weight function for the order $k$ Chebyshev polynomial of the second kind, $U_k(x)$.

**Usage**

```r
chebyshev.u.weight(x)
```

**Arguments**

- `x`  
  the function argument which can be a vector

**Details**

The function takes on non-zero values in the interval $(-1, 1)$. The formula used to compute the weight function is as follows:

$$w(x) = \sqrt{1 - x^2}$$
Value

The value of the weight function.

Author(s)

Frederick Novomestky < fnovomes@poly.edu >

References


Examples

```r
### compute the U Chebyshev function for argument values between -2 and 2
###
x <- seq(-1, 1, .01)
y <- chebyshev.u.weight(x)
plot(x, y)
```

---

**gegenbauer.inner.products**

*Inner products of Gegenbauer polynomials*

Description

This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) Gegenbauer polynomial, \( C_k^{(\alpha)}(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage

```r
gegenbauer.inner.products(n, alpha)
```

Arguments

- \( n \) integer value for the highest polynomial order
- \( \alpha \) numeric value for the polynomial parameter
Details

The formula used to compute the inner products is as follows.

\[ h_n = \left\langle C_n^{(\alpha)} | C_n^{(\alpha)} \right\rangle = \begin{cases} \frac{\pi 2^{1-2\alpha} \Gamma(n+2\alpha)}{n! \Gamma(n+\alpha) \Gamma(2\alpha)} & \alpha \neq 0 \\ \frac{\pi}{n^2} & \alpha = 0 \end{cases} \]

Value

A vector with \( n + 1 \) elements

1  inner product of order 0 orthogonal polynomial
2  inner product of order 1 orthogonal polynomial
...
\( n+1 \)  inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

ultraspherical.inner.products

Examples

###

### generate the inner products vector for the
### Gegenbauer polynomials of orders 0 to 10
### the polynomial parameter is 1.0
###
### h <- gegenbauer.inner.products( 10, 1 )
print( h )
**gegenbauer.polynomials**

*Create list of Gegenbauer polynomials*

---

**Description**

This function returns a list with \( n + 1 \) elements containing the order \( k \) Gegenbauer polynomials, \( C_k^{(\alpha)}(x) \), for orders \( k = 0, 1, \ldots, n \).

**Usage**

```r
gegenbauer.polynomials(n, alpha, normalized=FALSE)
```

**Arguments**

- `n`: integer value for the highest polynomial order
- `alpha`: polynomial parameter
- `normalized`: a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

**Details**

The function `gegenbauer.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the `normalized` argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.

**Value**

A list of \( n + 1 \) polynomial objects

- 1: order 0 Gegenbauer polynomial
- 2: order 1 Gegenbauer polynomial
- ...
- \( n+1 \): order \( n \) Chebyshev polynomial

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>

**References**


See Also

gegenbauer.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

```r
### generate a list of normalized Gegenbauer polynomials of orders 0 to 10
### polynomial parameter is 1.0
###
normalized.p.list <- gegenbauer.polynomials( 10, 1, normalized=TRUE )
print( normalized.p.list )
###
### generate a list of unnormalized Gegenbauer polynomials of orders 0 to 10
### polynomial parameter is 1.0
###
unnormalized.p.list <- gegenbauer.polynomials( 10, 1, normalized=FALSE )
print( unnormalized.p.list )
```

gegenbauer.recurrences

**Recurrence relations for Gegenbauer polynomials**

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) Gegenbauer polynomial, \( C_{k}^{(\alpha)}(x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage

```r
gegenbauer.recurrences(n, alpha, normalized=FALSE)
```

Arguments

- `n`: integer value for the highest polynomial order
- `alpha`: numeric value for polynomial parameter
- `normalized`: boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>
References


See Also

gegenbauer.inner.products

Examples

```r
### generate the recurrences data frame for
### the normalized Gegenbauer polynomials
### of orders 0 to 10.
### polynomial parameter value is 1.0
###
### normalized.r <- gegenbauer.recurrences( 10, 1, normalized=TRUE )
### print( normalized.r )
###
### generate the recurrences data frame for
### the normalized Gegenbauer polynomials
### of orders 0 to 10.
### polynomial parameter value is 1.0
###
### unnormalized.r <- gegenbauer.recurrences( 10, 1, normalized=FALSE )
### print( unnormalized.r )
```

gegenbauer.weight

*Weight function for the Gegenbauer polynomial*

Description

This function returns the value of the weight function for the order \( k \) Gegenbauer polynomial, \( C^{(\alpha)}_k(x) \).

Usage

```r
gegenbauer.weight(x, alpha)
```

Arguments

- **x**: the function argument which can be a vector
- **alpha**: polynomial parameter
Details

The function takes on non-zero values in the interval \((-1, 1)\). The formula used to compute the weight function is as follows.

\[ w(x) = (1 - x^2)^{\alpha - 0.5} \]

Value

The value of the weight function

Author(s)

Frederick Novomestky < fnovomes@poly.edu >

References


Examples

###
### compute the Gegenbauer weight function for argument values between -1 and 1
###
x <- seq(-1, 1, .01)
y <- gegenbauer.weight(x, 1)
plot(x, y)

ghermite.h.inner.products

*Inner products of generalized Hermite polynomials*

Description

This function returns a vector with \(n+1\) elements containing the inner product of an order \(k\) generalized Hermite polynomial, \(H_k^{(\mu)}(x)\), with itself (i.e. the norm squared) for orders \(k = 0, 1, \ldots, n\).

Usage

ghermite.h.inner.products(n, mu)
Arguments

- \( n \) n integer value for the highest polynomial order
- \( \mu \) \( \mu \) polynomial parameter

Details

The parameter \( \mu \) must be greater than -0.5. The formula used to compute the inner products is as follows.

\[
h_n(\mu) = \left\langle H_m^{(\mu)} | H_n^{(\mu)} \right\rangle = 2^n n! \Gamma(\left\lfloor \frac{n+\nu+\frac{1}{2}}{2} \right\rfloor + \mu + \frac{1}{2})
\]

Value

A vector with \( n + 1 \) elements

1 inner product of order 0 orthogonal polynomial
2 inner product of order 1 orthogonal polynomial
...

\( n+1 \) inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

```r
###
### generate the inner products vector for the
### generalized Hermite polynomials of orders 0 to 10
### polynomial parameter is 1
###
h <- ghermite.h.inner.products( 10, 1 )
print( h )
```
germ.h.polynomials

Create list of generalized Hermite polynomials

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) generalized Hermite polynomials, \( H_k^{(\mu)}(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

\[
\text{germ.h.polynomials}(n, \mu, \text{normalized} = \text{FALSE})
\]

Arguments

- \( n \) : integer value for the highest polynomial order
- \( \mu \) : numeric value for the polynomial parameter
- \( \text{normalized} \) : boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Details

The parameter \( \mu \) must be greater than -0.5. The function \text{germ.h.recurrences} produces a data frame with the recurrence relation parameters for the polynomials. If the \text{normalized} argument is FALSE, the function \text{orthogonal.polynomials} is used to construct the list of orthogonal polynomial objects. Otherwise, the function \text{orthonormal.polynomials} is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

1. order 0 generalized Hermite polynomial
2. order 1 generalized Hermite polynomial
   ...
\( n+1 \) order \( n \) generalized Hermite polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>
References


See Also

`ghermite.h.recurrences, orthogonal.polynomials, orthonormal.polynomials`

Examples

```r
### generate a list of normalized generalized Hermite polynomials of orders 0 to 10
### polynomial parameter is 1.0
###
### normalized.p.list <- ghermite.h.polynomials( 10, 1, normalized=TRUE )
### print( normalized.p.list )
###
### generate a list of unnormalized generalized Hermite polynomials of orders 0 to 10
### polynomial parameter is 1.0
###
### unnormalized.p.list <- ghermite.h.polynomials( 10, 1, normalized=FALSE )
### print( unnormalized.p.list )
```

ghermite.h.recurrences

Recurrence relations for generalized Hermite polynomials

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) generalized Hermite polynomial, \( H_k^{(\mu)}(x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage

`ghermite.h.recurrences(n, mu, normalized = FALSE)`
Arguments

- **n**: integer value for the highest polynomial order
- **mu**: numeric value for the polynomial parameter
- **normalized**: normalized boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Details

The parameter $\mu$ must be greater than -0.5.

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

- [ghermite.h.inner.products](#)

Examples

```r
### generate the recurrences data frame for
### the normalized generalized Hermite polynomials
### of orders 0 to 10.
### polynomial parameter value is 1.0
###
normalized.r <- ghermite.h.recurrences( 10, 1, normalized=TRUE )
print( normalized.r )
###
### generate the recurrences data frame for
### the unnormalized generalized Hermite polynomials
### of orders 0 to 10.
### polynomial parameter value is 1.0
###
```
ghermite.h.weight

unnormalized.r <- ghermite.h.recurrences( 10, 1, normalized=FALSE )
print( unnormalized.r )

ghermite.h.weight  Weight function for the generalized Hermite polynomial

Description
This function returns the value of the weight function for the order \( k \) generalized Hermite polynomial, \( H_k^{(\mu)}(x) \).

Usage
ghermite.h.weight(x, mu)

Arguments
- x: a numeric vector function argument
- mu: polynomial parameter

Details
The function takes on non-zero values in the interval \((-\infty, \infty)\). The parameter \( \mu \) must be greater than -0.5. The formula used to compute the generalized Hermite weight function is as follows.

\[
w(x, \mu) = |x|^2 \mu \exp(-x^2)
\]

Value
The value of the weight function

Author(s)
Frederick Novomestky < fnovomes@poly.edu >

References
Examples

###
### compute the generalized Hermite weight function for argument values
### between -3 and 3
###
x <- seq(-3, 3, .01)
y <- ghermite.h.weight(x, 1)

---

**glaguerre.inner.products**

*Inner products of generalized Laguerre polynomials*

**Description**

This function returns a vector with $n + 1$ elements containing the inner product of an order $k$ generalized Laguerre polynomial, $L_n^{(\alpha)}(x)$, with itself (i.e. the norm squared) for orders $k = 0, 1, \ldots, n$.

**Usage**

`glaguerre.inner.products(n, alpha)`

**Arguments**

- `n` integer highest polynomial order
- `alpha` polynomial parameter

**Details**

The formula used to compute the inner products is as follows.

$$ h_n = \left\langle L_n^{(\alpha)} | L_n^{(\alpha)} \right\rangle = \frac{\Gamma(\alpha+n+1)}{n!}. $$

**Value**

A vector with $n + 1$ elements

1. inner product of order 0 orthogonal polynomial
2. inner product of order 1 orthogonal polynomial
   ...
$n+1$: inner product of order $n$ orthogonal polynomial

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>
References


Examples

```r
###
### generate the inner products vector for the
### generalized Laguerre polynomial inner products of orders 0 to 10
### polynomial parameter is 1.
###
### h <- glaguerre.inner.products( 10, 1 )
print( h )
```

---

**glaguerre.polynomials**

Create list of generalized Laguerre polynomials

**Description**

This function returns a list with \( n + 1 \) elements containing the order \( n \) generalized Laguerre polynomials, \( L_n^{(\alpha)}(x) \), for orders \( k = 0, 1, \ldots, n \).

**Usage**

```r
glaguerre.polynomials(n, alpha, normalized=FALSE)
```

**Arguments**

- `n` integer value for the highest polynomial order
- `alpha` numeric value for the polynomial parameter
- `normalized` a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

**Details**

The function `glaguerre.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the `normalized` argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.
**glaguerre.polynomials**

**Value**

A list of \(n + 1\) polynomial objects

1. order 0 generalized Laguerre polynomial
2. order 1 generalized Laguerre polynomial
...
\(n+1\) order \(n\) generalized Laguerre polynomial

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>

**References**


**See Also**

*glaguerre.recurrences, orthogonal.polynomials, orthonormal.polynomials*

**Examples**

```r
### generate a list of normalized generalized Laguerre polynomials of orders 0 to 10
### polynomial parameter is 1.0
###
normalized.p.list <- glaguerre.polynomials( 10, 1, normalized=TRUE )
print( normalized.p.list )
###
### generate a list of unnormalized generalized Laguerre polynomials of orders 0 to 10
### polynomial parameter is 1.0
###
unnormalized.p.list <- glaguerre.polynomials( 10, 1, normalized=FALSE )
```
Description

This function returns a data frame with $n + 1$ rows and four named columns containing the coefficient vectors $c$, $d$, $e$ and $f$ of the recurrence relations for the order $k$ generalized Laguerre polynomial, $L_n^{(\alpha)}(x)$ and for orders $k = 0, 1, \ldots, n$.

Usage

glaguerre.recurrences(n, alpha, normalized=FALSE)

Arguments

  n       integer value for the highest polynomial order
  alpha   numeric value for the polynomial parameter
  normalized boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

glaguerre.inner.products
Examples

###
### generate the recurrences data frame for
### the normalized generalized Laguerre polynomials
### of orders 0 to 10. the polynomial parameter value is 1.0.
###
normalized.r <- glaguerre.recurrences( 10, 1, normalized=TRUE )
print( normalized.r )
###
### generate the recurrences data frame for
### the unnormalized generalized Laguerre polynomials
### of orders 0 to 10. the polynomial parameter value is 1.0.
###
unnormalized.r <- glaguerre.recurrences( 10, 1, normalized=FALSE )
print( unnormalized.r )

---

glaguerre.weight \hspace{1cm} Weight function for the generalized Laguerre polynomial

Description

This function returns the value of the weight function for the order \( k \) generalized Laguerre polynomial, \( L_n^{(\alpha)}(x) \).

Usage

glaguerre.weight(x,alpha)

Arguments

- \( x \) the function argument which can be a vector
- \( \alpha \) polynomial parameter

Details

The function takes on non-zero values in the interval \((0, \infty)\). The formula used to compute the weight function is as follows.

\[
w(x) = e^{-x}x^\alpha
\]

Value

The value of the weight function

Author(s)

Frederick Novomestky < fnovomes@poly.edu >
hermite.h.inner.products

References


Examples

```r
### compute the generalized Laguerre weight function for argument values
### between -3 and 3
### polynomial parameter value is 1.0
###
x <- seq(-3, 3, .01)
y <- glaguerre.weight(x, 1)
```

Description

This function returns a vector with \( n+1 \) elements containing the inner product of an order \( k \) Hermite polynomial, \( H_k(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage

```r
hermite.h.inner.products(n)
```

Arguments

- `n` integer value for highest polynomial order

Details

The formula used to compute the inner product is as follows.

\[
h_n = \langle H_n | H_n \rangle = \sqrt{\pi} \ 2^n \ n!.
\]
Value

A vector with \( n + 1 \) elements

1. inner product of order 0 orthogonal polynomial
2. inner product of order 1 orthogonal polynomial
...
\( n+1 \). inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

```r
###
### generate the inner products vector for the
### Hermite polynomials of orders 0 to 10
###
h <- hermite.h.inner.products(10)
print(h)
```


dermite.h.polynomials Create list of Hermite polynomials

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) Hermite polynomials, \( H_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

```r
hermite.h.polynomials(n, normalized=FALSE)
```

Arguments

\( n \) integer value for the highest polynomial order

\( \text{normalized} \) a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials
hermite.h.polynomials

Details

The function `hermite.h.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the normalized argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct list of orthonormal polynomial objects.

Value

A list of $n + 1$ polynomial objects

1. order 0 Hermite polynomial
2. order 1 Hermite polynomial
... 
$n+1$. order $n$ Hermite polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

`hermite.h.recurrences`, `orthogonal.polynomials`, `orthonormal.polynomials`

Examples

```r
### generate a list of normalized Hermite polynomials of orders 0 to 10
###
normalized.p.list <- hermite.h.polynomials(10, normalized=TRUE)
print(normalized.p.list)
###
### generate a list of unnormalized Hermite polynomials of orders 0 to 10
###
unnormalized.p.list <- hermite.h.polynomials(10, normalized=FALSE)
print(unnormalized.p.list)
```
hermite.h.recurrences

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) Hermite polynomial, \( H_k(x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage

\[
\text{hermite.h.recurrences}(n, \text{normalized}=\text{FALSE})
\]

Arguments

- \( n \) integer value for the highest polynomial order
- \( \text{normalized} \) boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

- hermite.h.inner.products

Examples

```r
### generate the recurrences data frame for
### the normalized Hermite H polynomials
### of orders 0 to 10.
###
normalized.r <- hermite.h.recurrences(10, normalized=TRUE)
print(normalized.r)
```
### generate the recurrences data frame for
### the unnormalized Hermite H polynomials
### of orders 0 to 10.
unnormalized.r <- hermite.h.recurrences(10, normalized=FALSE)
print(unnormalized.r)

---

hermite.h.weight

**Weight function for the Hermite polynomial**

**Description**

This function returns the value of the weight function for the order $k$ Hermite polynomial, $H_k(x)$.

**Usage**

hermite.h.weight(x)

**Arguments**

- **x**: the function argument which can be a vector

**Details**

The function takes on non-zero values in the interval $(-\infty, \infty)$. The formula used to compute the weight function.

$$w(x) = \exp(-x^2)$$

**Value**

The value of the weight function

**Author(s)**

Frederick Novomestky < fnovomes@poly.edu>

**References**


Examples

```r
### compute the Hermite weight function for argument values
### between -3 and 3
x <- seq(-3, 3, .01)
y <- hermite.h.weight(x)
plot(x, y)
```

hermite.he.inner.products

*Inner products of Hermite polynomials*

Description

This function returns a vector with \( n+1 \) elements containing the inner product of an order \( k \) Hermite polynomial, \( He_k(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage

```r
hermite.he.inner.products(n)
```

Arguments

- `n`: integer value for the highest polynomial order

Details

The formula used to compute the inner products is as follows.

\[
h_n = \langle He_n | He_n \rangle = \sqrt{2\pi n!}.
\]

Value

A vector with \( n+1 \) elements

- 1: inner product of order 0 orthogonal polynomial
- 2: inner product of order 1 orthogonal polynomial
- ...
- \( n+1 \): inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>
References


Examples

```r
###
### generate the inner products vector for the
### scaled Hermite polynomials of orders 0 to 10
###
### h <- hermite.he.inner.products(10)
print(h)
```

```
hermite.he.polynomials

*Create list of Hermite polynomials*

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) Hermite polynomials, \( H_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

`hermite.he.polynomials(n, normalized=FALSE)`

Arguments

- `n`: integer value for the highest polynomial order
- `normalized`: a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function `hermite.he.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the `normalized` argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.
### hermite.he.recurrences

**Value**

A list of \( n + 1 \) polynomial objects

1. order 0 Hermite polynomial
2. order 1 Hermite polynomial

... \( n+1 \) order \( n \) Hermite polynomial

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>

**References**


**See Also**

hermite.he.recurrences, orthogonal.polynomials, orthonormal.polynomials

**Examples**

```r
### generate a list of normalized Hermite polynomials of orders 0 to 10
###
normalized.p.list <- hermite.he.polynomials(10, normalized=TRUE)
print(normalized.p.list)

### generate a list of unnormalized Hermite polynomials of orders 0 to 10
###
unnormalized.p.list <- hermite.he.polynomials(10, normalized=FALSE)
print(unnormalized.p.list)
```

**Description**

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) Hermite polynomial, \( H_k(x) \), and for orders \( k = 0, 1, \ldots, n \).
hermite.he.recurrences

Usage

hermite.he.recurrences(n, normalized=FALSE)

Arguments

n integer value for the highest polynomial order
normalized boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

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

See Also

hermite.he.inner.products

Examples

### generate the recurrences data frame for
### the normalized Hermite H polynomials
### of orders 0 to 10.
###
normalized.r <- hermite.he.recurrences(10, normalized=TRUE)
print(normalized.r)
###
### generate the recurrences data frame for
### the unnormalized Hermite H polynomials
### of orders 0 to 10.
###

unnormalized.r <- hermite.he.recurrences(10, normalized=FALSE)
print(unnormalized.r)
**hermite.he.weight**

Weight function for the Hermite polynomial

**Description**

This function returns the value of the weight function for the order \( k \) Hermite polynomial, \( H_k(x) \).

**Usage**

```
hermite.he.weight(x)
```

**Arguments**

- `x` the function argument which can be a vector

**Details**

The function takes on non-zero values in the interval \((-\infty, \infty)\). The formula used to compute the weight function is as follows.

\[
w(x) = \exp\left(-\frac{x^2}{2}\right)
\]

**Value**

The value of the weight function

**Author(s)**

Frederick Novomestky < fnovomes@poly.edu >

**References**


**Examples**

```r
###
### compute the scaled Hermite weight function for argument values
### between -3 and 3
###
x <- seq(-3, 3, .01)
y <- hermite.he.weight(x)
```
Description

This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) Jacobi polynomial, \( G_k(p,q,x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage

\[ \text{jacobi.g.inner.products}(n, p, q) \]

Arguments

- \( n \) integer value for the highest polynomial order
- \( p \) numeric value for the first polynomial parameter
- \( q \) numeric value for the first polynomial parameter

Details

The formula used to compute the inner products is as follows.

\[
h_n = \langle G_n | G_n \rangle = \frac{n! \Gamma(n+q) \Gamma(n+p) \Gamma(n+p-q+1)}{(2n+p) \Gamma(2n+p)}.
\]

Value

A vector with \( n + 1 \) elements

1. inner product of order 0 orthogonal polynomial
2. inner product of order 1 orthogonal polynomial

... 

\( n + 1 \) inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

```r
###
### generate the inner products vector for the
### G Jacobi polynomials of orders 0 to 10
### parameter p is 3 and parameter q is 2
###
### h <- jacobi.g.inner.products(10, 3, 2)
print(h)
```

jacobi.g.polynomials

Create list of Jacobi polynomials

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) Jacobi polynomials, \( G_k(p, q, x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

```r
jacobi.g.polynomials(n, p, q, normalized=FALSE)
```

Arguments

- `n` integer value for the highest polynomial order
- `p` numeric value for the first polynomial parameter
- `q` numeric value for the second polynomial parameter
- `normalized` a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function `jacobi.g.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the `normalized` argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

- `1` order 0 Jacobi polynomial
- `2` order 1 Jacobi polynomial
- `...`
- `n+1` order \( n \) Jacobi polynomial
Author(s)
Frederick Novomestky <fnovomes@poly.edu>

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

See Also
jacobi.g.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

###
### generate a list of normalized Jacobi G polynomials of orders 0 to 10
### first parameter value p is 3 and second parameter value q is 2
###
normalized.p.list <- jacobi.g.polynomials( 10, 3, 2, normalized=TRUE )
print( normalized.p.list )
###
### generate a list of normalized Jacobi G polynomials of orders 0 to 10
### first parameter value p is 3 and second parameter value q is 2
###
unnormalized.p.list <- jacobi.g.polynomials( 10, 3, 2, normalized=FALSE )
print( unnormalized.p.list )

Description
This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) Jacobi polynomial, \( G_k(p, q, x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage
jacobi.g.recurrences(n, p, q, normalized=FALSE)
Arguments

- **n**: integer value for the highest polynomial order
- **p**: numeric value for the first polynomial parameter
- **q**: numeric value for the second polynomial parameter
- **normalized**: boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

- `jacobi.g.inner.products`, `pochhammer`

Examples

```r
### generate the recurrences data frame for
### the normalized Jacobi G polynomials
### of orders 0 to 10.
### parameter p is 3 and parameter q is 2
###
normalized.r <- jacobi.g.recurrences( 10, 3, 2, normalized=TRUE )
print( normalized.r )
###
### generate the recurrences data frame for
### the normalized Jacobi G polynomials
### of orders 0 to 10.
### parameter p is 3 and parameter q is 2
###
unnormalized.r <- jacobi.g.recurrences( 10, 3, 2, normalized=FALSE )
print( unnormalized.r )
```
jacobi.g.weight

Weight function for the Jacobi polynomial

Description

This function returns the value of the weight function for the order \( k \) Jacobi polynomial, \( G_k(p, q, x) \).

Usage

\[ \text{jacobi.g.weight}(x, p, q) \]

Arguments

- \( x \): the function argument which can be a vector
- \( p \): the first polynomial parameter
- \( q \): the second polynomial parameter

Details

The function takes on non-zero values in the interval \((0, 1)\). The formula used to compute the weight function is as follows.

\[ w(x) = (1 - x)^{p-q} x^{q-1} \]

Value

The value of the weight function

Author(s)

Frederick Novomestky < fnovomes@poly.edu >

References


Szegö, G., 1939. Orthogonal Polynomials, 23, American Mathematical Society Colloquium Publications, Providence, RI.
jacobi.matrices

Examples

###
### compute the Jacobi G weight function for argument values
### between 0 and 1
### parameter p is 3 and q is 2
###
x <- seq(0, 1, .01)
y <- jacobi.g.weight(x, 3, 2)

jacobi.matrices  Create list of Jacobi matrices from monic recurrence parameters

Description

Return a list of \( n \times n \) real symmetric, tri-diagonal matrices which are the principal minors of the \( n \times n \) Jacobi matrix derived from the monic recurrence parameters, \( a \) and \( b \), for orthogonal polynomials.

Usage

jacobi.matrices(r)

Arguments

\( r \)  a data frame containing the parameters \( a \) and \( b \)

Value

A list of symmetric, tri-diagonal matrices

1  a 1 \times 1 \ matrix
2  a 2 \times 2 \ matrix
...
n  an \( n \times n \) matrix

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

\( r \) <- chebyshev.t.recurrences(5)
m.r <- monic.polynomial.recurrences(\( r \))j.m <- jacobi.matrices(m.r)
Description

This function returns a vector with $n + 1$ elements containing the inner product of an order $k$ Jacobi polynomial, $P_k^{(\alpha,\beta)}(x)$, with itself (i.e. the norm squared) for orders $k = 0, 1, \ldots, n$.

Usage

`jacobi.p.inner.products(n, alpha, beta)`

Arguments

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

Details

The formula used to compute the inner products is as follows.

$$h_n = \langle P_n^{(\alpha,\beta)} | P_n^{(\alpha,\beta)} \rangle = \frac{2^{\alpha+\beta+1}}{2n^{\alpha+\beta+1}} \frac{\Gamma(n+\alpha+1) \Gamma(n+\beta+1)}{n! \Gamma(n+\alpha+\beta+1)}.$$  

Value

A vector with $n + 1$ elements

- 1 inner product of order 0 orthogonal polynomial
- 2 inner product of order 1 orthogonal polynomial
- ...
- $n+1$ inner product of order $n$ orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

```r
###
### generate the inner product vector for the P Jacobi polynomials of orders 0 to 10
###

h <- jacobi.p.inner.products(10, 2, 2)
print(h)
```

---

**jacobi.p.polynomials**  
*Create list of Jacobi polynomials*

---

**Description**

This function returns a list with \( n + 1 \) elements containing the order \( k \) Jacobi polynomials, \( P_k^{(\alpha,\beta)}(x) \), for orders \( k = 0, 1, ..., n \).

**Usage**

```r
jacobi.p.polynomials(n, alpha, beta, normalized=FALSE)
```

**Arguments**

- `n`  
  integer value for the highest polynomial order

- `alpha`  
  numeric value for the first polynomial parameter

- `beta`  
  numeric value for the second polynomial parameter

- `normalized`  
  a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

**Details**

The function `jacobi.p.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the `normalized` argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.

**Value**

A list of \( n + 1 \) polynomial objects

- \( 1 \)  
  order 0 Jacobi polynomial

- \( 2 \)  
  order 1 Jacobi polynomial

- ...  

- \( n+1 \)  
  order \( n \) Chebyshev polynomial

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>
References


See Also

`jacobi.p.recurrences`, `orthogonal.polynomials`, `orthonormal.polynomials`

Examples

```r
### generate a list of normalized Jacobi P polynomials of orders 0 to 10
### first parameter value a is 2 and second parameter value b is 2
###
normalized.p.list <- jacobi.p.polynomials(10, 2, 2, normalized=TRUE)
print(normalized.p.list)
###
### generate a list of unnormalized Jacobi P polynomials of orders 0 to 10
### first parameter value a is 2 and second parameter value b is 2
###
unnormalized.p.list <- jacobi.p.polynomials(10, 2, 2, normalized=FALSE)
print(unnormalized.p.list)
```

Description

This function returns a data frame with $n + 1$ rows and four named columns containing the coefficient vectors $c$, $d$, $e$ and $f$ of the recurrence relations for the order $k$ Jacobi polynomial, $P_k^{(\alpha,\beta)}(x)$, and for orders $k = 0, 1, \ldots, n$.

Usage

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

Arguments

- `n`: integer value for the highest polynomial order
- `alpha`: numeric value for the first polynomial parameter
- `beta`: numeric value for the second polynomial parameter
- `normalized`: boolean value which, if TRUE, returns recurrence relations for normalized polynomials
A data frame with the recurrence relation parameters.

Author(s)
Frederick Novomestky <fnovomes@poly.edu>

References

See Also
- `jacobi.p.inner.products`, `pochhammer`
**Arguments**

- **x**: the function argument which can be a vector
- **alpha**: the first polynomial parameter
- **beta**: the second polynomial parameter

**Details**

The function takes on non-zero values in the interval $(-1, 1)$. The formula used to compute the weight function is as follows.

$$w(x) = (1 - x)^\alpha (1 + x)^\beta$$

**Value**

The value of the weight function

**Author(s)**

Frederick Novomestky < fnovomes@poly.edu >

**References**


**Examples**

```r
###
### compute the Jacobi P weight function for argument values
### between -1 and 1
###
### x <- seq(-1, 1, .01)
y <- jacobi.p.weight(x, 2, 2)
```
Inner products of Laguerre polynomials

Description

This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) Laguerre polynomial, \( L_n(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage

laguerre.inner.products(n)

Arguments

\( n \) integer value for the highest polynomial order

Details

The formula used to compute the inner products is as follows.

\[
h_n = \langle L_n | L_n \rangle = 1.
\]

Value

A vector with \( n + 1 \) elements

1 inner product of order 0 orthogonal polynomial

2 inner product of order 1 orthogonal polynomial

... 

\( n+1 \) inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Szegö, G., 1939. Orthogonal Polynomials, 23, American Mathematical Society Colloquium Publications, Providence, RI.
Examples

```r
###
### generate the inner products vector for the
### Laguerre polynomial inner products of orders 0 to 10
###
h <- laguerre.inner.products(10)
print(h)
```

---

**Description**

This function returns a list with $n + 1$ elements containing the order $k$ Laguerre polynomials, $L_n(x)$, for orders $k = 0, 1, \ldots, n$.

**Usage**

```r
laguerre.polynomials(n, normalized=FALSE)
```

**Arguments**

- `n`: integer value for the highest polynomial order
- `normalized`: a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

**Details**

The function `laguerre.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the `normalized` argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.

**Value**

A list of $n + 1$ polynomial objects

- 1: order 0 Laguerre polynomial
- 2: order 1 Laguerre polynomial
- ...
- $n+1$: order $n$ Laguerre polynomial

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>
References


See Also

laguerre.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

```r
### generate a list of normalized Laguerre polynomials of orders 0 to 10
###
normalized.p.list <- laguerre.polynomials(10, normalized=TRUE)
print(normalized.p.list)
###
### generate a list of unnormalized Laguerre polynomials of orders 0 to 10
###
unnormalized.p.list <- laguerre.polynomials(10, normalized=FALSE)
print(unnormalized.p.list)
```

Description

This function returns a data frame with \(n + 1\) rows and four named columns containing the coefficient vectors \(c, d, e\) and \(f\) of the recurrence relations for the order \(k\) Laguerre polynomial, \(L_n(x)\), and for orders \(k = 0, 1, \ldots, n\).

Usage

```r
laguerre.recurrences(n, normalized=FALSE)
```

Arguments

- **n**: integer value for the highest polynomial order
- **normalized**: boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.
laguerre.weight

Description

This function returns the value of the weight function for the order $k$ Laguerre polynomial, $L_n(x)$.

Usage

laguerre.weight(x)

Arguments

x the function argument which can be a vector
The function takes on non-zero values in the interval \((0, \infty)\). The formula used to compute the weight function is as follows. \( w(x) = e^{-x} \)

The value of the weight function

Frederick Novomestky < fnovomes@poly.edu >

References


Examples

```r
###
### compute the Laguerre weight function for argument values
### between 0 and 3
x <- seq(-0, 3, .01)
y <- laguerre.weight(x)
plot(x, y)
```

This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) Legendre polynomial, \( P_k(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage

`legendre.inner.products(n)`

Arguments

\( n \) integer value for the highest polynomial order
Details

The formula used to compute the inner products is as follows.

\[ h_n = \langle P_n \mid P_n \rangle = \frac{2}{2n+1}. \]

Value

A vector with \( n + 1 \) elements

1 \hspace{1cm} \text{inner product of order 0 orthogonal polynomial}
2 \hspace{1cm} \text{inner product of order 1 orthogonal polynomial}
...
\( n+1 \) \hspace{1cm} \text{inner product of order \( n \) orthogonal polynomial}

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

*spherical.inner.products*

Examples

```r
### compute the inner product for the
### Legendre polynomials of orders 0 to 1
###
h <- legendre.inner.products(10)
print(h)
```
legendre.polynomials  Create list of Legendre polynomials

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) Legendre polynomials, \( P_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

legendre.polynomials(n, normalized=FALSE)

Arguments

\n
n  integer value for the highest polynomial order
normalized  a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function legendre.recurrences produces a data frame with the recurrence relation parameters for the polynomials. If the normalized argument is FALSE, the function orthogonal.polynomials is used to construct the list of orthogonal polynomial objects. Otherwise, the function orthonormal.polynomials is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

1  order 0 Legendre polynomial
2  order 1 Legendre polynomial
...
\( n+1 \)  order \( n \) Legendre polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

legendre.recurrences

See Also

legendre.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

###
### generate a list of normalized Laguerre polynomials of orders 0 to 10
###
normalized.p.list <- legendre.polynomials( 10, normalized=TRUE )
print( normalized.p.list )
###
### generate a list of unnormalized Laguerre polynomials of orders 0 to 10
###
unnormalized.p.list <- legendre.polynomials( 10, normalized=FALSE )
print( unnormalized.p.list )

---

legendre.recurrences      Recurrence relations for Legendre polynomials

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) Legendre polynomial, \( P_k(x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage

legendre.recurrences(n, normalized=FALSE)

Arguments

- **n** integer value for the highest polynomial order
- **normalized** boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>
References


See Also

legendre.inner.products

Examples

```r
## generate the recurrences data frame for the normalized Legendre polynomials
## of orders 0 to 10.
normalized.r <- legendre.recurrences(10, normalized=TRUE)
print(normalized.r)

## generate the recurrences data frame for the normalized Legendre polynomials
## of orders 0 to 10.
unnormalized.r <- legendre.recurrences(10, normalized=FALSE)
print(unnormalized.r)
```

---

**legendre.weight**

Weight function for the Legendre polynomial

**Description**

This function returns the value of the weight function for the order \( k \) Legendre polynomial, \( P_k(x) \).

**Usage**

```r
legendre.weight(x)
```

**Arguments**

- \( x \) the function argument which can be a vector

**Details**

The function takes on non-zero values in the interval \((-1, 1)\). The formula used to compute the weight function is as follows.

\[
w(x) = 1
\]
Value

The value of the weight function

Author(s)

Frederick Novomestky < fnovomes@poly.edu >

References


Examples

```r
###
### compute the Legendre weight function for argument values
### between -1 and 1
###
x <- seq(-1, 1, .01)
y <- legendre.weight(x)
plot(x, y)
```

------------------------------------------------------------------------

**lpochhammer**

*Calculate the logarithm of Pochhammer's symbol*

Description

`lpochhammer` returns the value of the natural logarithm of Pochhammer’s symbol calculated as

\[
\ln \left( (z)_n \right) = \ln \Gamma (z + n) - \ln \Gamma (z)
\]

where \( \Gamma (z) \) is the Gamma function

Usage

`lpochhammer(z, n)`

Arguments

- `z`: argument of the symbol
- `n`: integer number of terms in the symbol
Value
The value of the logarithm of the symbol

Author(s)
Frederick Novomestky <fnovomes@poly.edu>

See Also
pochhammer

Examples
lpochhammer( pi, 5 )

monic.polynomial.recurrences
Create data frame of monic recurrences

Description
This function returns a data frame with parameters required to construct monic orthogonal polynomials based on the standard recurrence relation for the non-monic polynomials. The recurrence relation for monic orthogonal polynomials is as follows.

\[ q_{k+1}(x) = (x - a_k) q_k(x) - b_k q_{k-1}(x) \]

We require that \( q_{-1}(x) = 0 \) and \( q_0(x) = 1 \). The recurrence for non-monic orthogonal polynomials is given by

\[ c_k p_{k+1}(x) = (d_k + e_k x) p_k(x) - f_k p_{k-1}(x) \]

We require that \( p_{-1}(x) = 0 \) and \( p_0(x) = 1 \). The monic polynomial recurrence parameters, \( a \) and \( b \), are related to the non-monic polynomial parameter vectors \( c, d, e \) and \( f \) in the following manner.

\[ a_k = -\frac{d_k}{e_k} \]

\[ b_k = \frac{c_k - 1}{e_k - 1} \]

with \( b_0 = 0 \).

Usage
monic.polynomial.recurrences(recurrences)

Arguments
recurrences the data frame of recurrence parameter vectors \( c, d, e \) and \( f \)
monic.polynomials

Value

A data frame with $n + 1$ rows and two named columns, \texttt{a} and \texttt{b}.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

\texttt{orthogonal.polynomials}.

Examples

```r
### construct a list of the recurrences for the T Chebyshev polynomials of
### orders 0 to 10
###
r <- chebyshev.t.recurrences(10, normalized=TRUE)
###
### construct the monic polynomial recurrences from the above list
###
m.r <- monic.polynomial.recurrences(r)
```

monic.polynomials Create list of monic orthogonal polynomials

Description

This function returns a list with $n + 1$ elements containing the order $k$ monic polynomials for orders $k = 0, 1, \ldots, n$.

Usage

\texttt{monic.polynomials(monic.recurrences)}
Arguments

monic.recurrences

a data frame containing the parameters a and b

Value

A list with \( n + 1 \) polynomial objects

1 order 0 monic orthogonal polynomial
2 order 1 monic orthogonal polynomial
...
\( n+1 \) order \( n \) monic orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

monic.polynomial.recurrences

Examples

###
### generate the recurrences for the T Chebyshev polynomials
### of orders 0 to 10
###
r <- chebyshev.t.recurrences(10, normalized=TRUE)
###
### get the corresponding monic polynomial recurrences
###
m.r <- monic.polynomial.recurrences(r)
###
### obtain the list of monic polynomials
###
p.list <- monic.polynomials(m.r)
Create orthogonal polynomials

Description

Create list of orthogonal polynomials from the following recurrence relations for \( k = 0, 1, \ldots, n \).

\[
c_k p_{k+1}(x) = (d_k + e_k x) p_k(x) - f_k p_{k-1}(x)
\]

We require that \( p_{-1}(x) = 0 \) and \( p_0(x) = 1 \). The coefficients are the column vectors \( c, d, e \) and \( f \).

Usage

\[
\text{orthogonal.polynomials}(\text{recurrences})
\]

Arguments

- \text{recurrences} a data frame containing the parameters of the orthogonal polynomial recurrence relations

Details

The argument is a data frame with \( n + 1 \) rows and four named columns. The column names are \( c, d, e \) and \( f \). These columns correspond to the column vectors described above.

Value

A list of \( n + 1 \) polynomial objects

- \( 1 \) Order 0 orthogonal polynomial
- \( 2 \) Order 1 orthogonal polynomial
  
- \( \ldots \)
  
- \( n+1 \) Order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Szegö, G., 1939. _Orthogonal Polynomials_, 23, American Mathematical Society Colloquium Publications, Providence, RI.
Examples

###
### generate the recurrence relations for T Chebyshev polynomials of orders 0 to 10
###
r <- chebyshev.t.recurrences( 10, normalized=FALSE )
print( r )
###
### generate the list of orthogonal polynomials
###
p.list <- orthogonal.polynomials( r )
print( p.list )

orthonormal.polynomials

Create orthonormal polynomials

Description

Create list of orthonormal polynomials from the following recurrence relations for \( k = 0, 1, \ldots, n \).

\[
c_k p_{k+1} (x) = (d_k + e_k x) p_k (x) - f_k p_{k-1} (x)
\]

We require that \( p_{-1} (x) = 0 \) and \( p_0 (x) = 1 \). The coefficients are the column vectors \( c, d, e \) and \( f \).

Usage

orthonormal.polynomials(recurrences, p.0)

Arguments

recurrences a data frame containing the parameters of the orthonormal polynomial recurrence relations

p.0 a polynomial object for the order 0 orthonormal polynomial

Details

The argument is a data frame with \( n + 1 \) rows and four named columns. The column names are \( c, d, e \) and \( f \). These columns correspond to the column vectors described above.

Value

A list of \( n + 1 \) polynomial objects

1 Order 0 orthonormal polynomial

2 Order 1 orthonormal polynomial

... 

\( n+1 \) Order \( n \) orthonormal polynomial
pochhammer

Author(s)
Frederick Novomestky <fnovomes@poly.edu>

References


Examples
```r
### generate a data frame with the recurrences parameters for normalized T Chebyshev
### polynomials of orders 0 to 10
###
### r <- chebyshev.t.recurrences( 10, normalized=TRUE )
###
### norm <- sqrt( pi )
###
### p.0 <- polynomial( c( 1 / norm ) )
###
### p.list <- orthonormal.polynomials( r, p.0 )
###
```

---

pochhammer

**Calculate the value of Pochhammer's symbol**

**Description**

pochhammer returns the value of Pochhammer's symbol calculated as

\[
(z)_n = z \ (z+1) \ldots (z+n-1) = \frac{\Gamma(z+n)}{\Gamma(z)}
\]

where \( \Gamma(z) \) is the Gamma function

**Usage**

pochhammer(z, n)
**Arguments**

- \( z \) numeric value for the argument of the symbol
- \( n \) integer value for the number of terms in the symbol

**Value**

The value of Pochhammer's symbol

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>

**Examples**

```plaintext
###
### compute the Pochammer's symbol for z equal to 1 and
### n equal to 5
###
pochhammer(1, 5)
```

**polynomial.coefficients**

Create list of polynomial coefficient vectors

**Description**

This function returns a list with \( n + 1 \) elements containing the vector of coefficients of the order \( k \) polynomials for orders \( k = 0, 1, \ldots, n \). Each element in the list is a vector.

**Usage**

```plaintext
polynomial.coefficients(polynomials)
```

**Arguments**

- polynomials list of polynomial objects

**Value**

A list of \( n + 1 \) polynomial objects where each element is a vector of coefficients.

- 1 Coefficient(s) of the order 0 polynomial
- 2 Coefficient(s) of the order 1 polynomial
- ... 
- \( n+1 \) Coefficient(s) of the order \( n \) polynomial
polynomial.derivatives

Author(s)
Frederick Novomestky <fnovomes@poly.edu>

Examples

###
### generate a list of normalized T Chebyshev polynomials
### of orders 0 to 10
###
p.list <- chebyshev.t.polynomials( 10, normalized=TRUE )
###
### obtain the list of coefficients for these polynomials
###
p.coef <- polynomial.coefficients( p.list )

polynomial.derivatives

Create list of polynomial derivatives

Description
This function returns a list with \( n + 1 \) elements containing polynomial objects which are the derivatives of the order \( k \) polynomials for orders \( k = 0, 1, \ldots, n \).

Usage

polynomial.derivatives(polynomials)

Arguments

polynomials list of polynomial objects

Details
The polynomial objects in the argument polynomials are as follows

- 1order 0 polynomial
- 2order 1 polynomial ...
- \( n+1 \)order \( n \) polynomial

Value
List of \( n + 1 \) polynomial objects

1 derivative of polynomials[[1]]
2 derivative of polynomials[[2]]
...
\( n+1 \) derivative of polynomials[[\( n+1 \)]]
polynomial.functions

Author(s)
Frederick Novomestky <fnovomes@poly.edu>

Examples

```r
### generate a list of normalized T Chebyshev polynomials of
### orders 0 to 10
###
p.list <- chebyshev.t.polynomials( 10, normalized=TRUE )
###
### generate the corresponding list of polynomial derivatives
###
p.deriv <- polynomial.derivatives( p.list )
```

polynomial.functions  
Coerce polynomials to functions

Description
This function returns a list with \(n+1\) elements containing the functions of the order \(k\) polynomials for orders \(k = 0, 1, \ldots, n\) and for the given argument \(x\).

Usage

```r
polynomial.functions(polynomials, ...)
```

Arguments

- **polynomials**: a list of polynomial objects
- **...**: further arguments to be passed to or from methods

Details
The function uses the method as.function.polynomial to coerce each polynomial object to a function object.

Value
A list of \(n + 1\) polynomial objects where each element is the function for the polynomial.

1  Function for the order 0 polynomial
2  Function for the order 1 polynomial
...
\(n+1\)  Function for the order \(n\) polynomial
Author(s)
  Frederick Novomestky <fnovomes@poly.edu>

Examples
  
  ```
  ### generate a list of T Chebyshev polynomials of orders 0 to 10
  ### p.list <- chebyshev.t.polynomials( 10, normalized=FALSE )
  ### create the list of functions for each polynomial
  ### f.list <- polynomial.functions( p.list )
  ```

Description
  This function returns a list with \( n + 1 \) elements containing polynomial objects which are the indefinite integrals of the order \( k \) polynomials for orders \( k = 0, 1, \ldots, n \).

Usage
  polynomial.integrals(polynomials)

Arguments
  polynomials  list of polynomial objects

Details
  The polynomial objects in the argument polynomials are as follows
  - 1order 0 polynomial
  - 2order 1 polynomial ...
  - n+1order n polynomial

Value
  List of \( n + 1 \) polynomial objects
  
  1 integral of polynomials[[1]]
  2 integral of polynomials[[2]]
  ...
  \( n+1 \) integral of polynomials[[\( n+1 \)]]
Author(s)
Frederick Novomestky <fnovomes@poly.edu>

Examples

```r
###
### generate a list of normalized T Chebyshev polynomials
### of orders 0 to 10
###
p.list <- chebyshev.t.polynomials( 10, normalized=TRUE )
###
### generate the corresponding list of polynomial integrals
###
p.int <- polynomial.integrals( p.list )
```

---

**Description**

This function returns a vector with \( n \) elements containing the orders of the polynomials

**Usage**

`polynomial.orders(polynomials)`

**Arguments**

- `polynomials` list of \( n \) polynomial objects

**Value**

A vector of \( n \) values

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Order of polynomials[[1]]</td>
</tr>
<tr>
<td>2</td>
<td>Order of polynomials[[2]]</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
| \( n \) | Order of polynomials[[\( n \)]]

**Author(s)**
Frederick Novomestky <fnovomes@poly.edu>
### Examples

```r
###
### generate a list of normalized T Chebyshev polynomials
### of orders 0 to 10
###
p.list <- chebyshev.t.polynomials( 10, normalized=TRUE )
###
### get the vector of polynomial orders
###
p.order <- polynomial.orders( p.list )
```

---

**polynomial.powers**  
Create a list of polynomial linear combinations

---

**Description**

This function returns a list with $n + 1$ elements containing the vector of linear combinations of the order $k$ polynomials for orders $k = 0, 1, \ldots, n$. Each element in the list is a vector.

**Usage**

```r
polynomial.powers(polynomials)
```

**Arguments**

- `polynomials`  
  A list of polynomials

**Details**

The $j$-th component in the list is a vector of linear combinations of the order $k$ polynomials for orders $k = 0, 1, \ldots, j - 1$ equal to the monomial $x$ raised to the power $j - 1$.

**Value**

A list of $n + 1$ elements where each element is a vector of linear combinations.

- 1  
  Linear combination(s) of polynomials up to order 0
- 2  
  Linear combination(s) of polynomials up to order 1
- ...
- $n+1$  
  Linear combination(s) of polynomials up to order $n$

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>
Examples

```r
###
### generate Legendre polynomials of orders 0 to 10
###
polynomials <- legendre.polynomials( 10 )
###
### generate list of linear combinations of these polynomials
###
alphas <- polynomial.powers( polynomials )
print( alphas )
```

---

**polynomial.roots**  
*Create a list of polynomial roots*

**Description**

This function returns a list with \( n \) elements containing the roots of the order \( k \) monic orthogonal polynomials for orders \( k = 0, 1, \ldots, n \) using a data frame with the monic polynomial recurrence parameter vectors \( a \) and \( b \)

**Usage**

`polynomial.roots(m.r)`

**Arguments**

- `m.r`: monic recurrence data frame with parameters \( a \) and \( b \)

**Details**

The parameter \( m.r \) is a data frame with \( n+1 \) rows and two names columns. The columns which are names \( a \) and \( b \) correspond to the above referenced vectors. Function `jacobi.matrices` is used to create a list of symmetric, tridiagonal Jacobi matrices from these named columns. The eigenvalues of the \( k \times k \) Jacobi matrix are the roots or zeros of the order \( k \) monic orthogonal polynomial.

**Value**

A list with \( n \) elements each of which is a vector of polynomial roots

- 1: roots of the order 1 monic polynomial
- 2: roots of the order 2 monic polynomial
- ...
- \( n \): roots of the order \( n \) monic polynomial

**Author(s)**

Frederick Novomestky `<fnovomes@poly.edu>`
polynomial.values

References


See Also

monic.polynomial.recurrences, jacobi.matrices

Examples

```r
### generate the recurrences data frame for
### the normalized Chebyshev polynomials of
### orders 0 to 10
###
### r <- chebyshev.t.recurrences( 10, normalized=TRUE )
###
### construct the corresponding monic polynomial
### recurrences
###
### m.r <- monic.polynomial.recurrences( r )
###
### obtain the polynomial roots from the monic polynomial
### recurrences
###
### p.roots <- polynomial.roots( m.r )
```

---

polynomial.values Create vector of polynomial values

Description

This function returns a list with $n + 1$ elements containing the values of the order $k$ polynomials for orders $k = 0, 1, \ldots, n$ and for the given argument $x$.

Usage

```r
polynomial.values( polynomials, x )
```

Arguments

- **polynomials**: list of polynomial objects
- **x**: the argument which can be any numeric object
Value

A list of $n + 1$ polynomial objects where each element is the value of the polynomial.

1 Value(s) for the order 0 polynomial
2 Value(s) for the order 1 polynomial
...
$n+1$ Value(s) for the order $n$ polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

Examples

```r
###
### generate a list of T Chebyshev polynomials of
### orders 0 to 10
###
p.list <- chebyshev.t.polynomials(10, normalized=FALSE)
x <- seq(-2, 2, .01)
###
### compute the value of the polynomials for the given range of values in x
###
y <- polynomial.values(p.list, x)
print(y)
```

scaleX

Scale values from $[a,b]$ to $[u,v]$

Description

This function returns a vector of values that have been mapped from the interval $[a,b]$ to the interval $[u,v]$.

Usage

```r
scaleX(x, a = min(x, na.rm = TRUE), b = max(x, na.rm = TRUE), u, v)
```

Arguments

- **x**: A numerical vector of values to be mapped into a target interval
- **a**: A numerical lower bound for the domain interval with min(x) as the default value
- **b**: A numerical upper bound for the domain interval with max(x) as the default value
- **u**: A numerical lower bound for the target interval
- **v**: A numerical upper bound for the target interval
Details

Target lower and/or upper bounds can be $-\infty$ and $\infty$, respectively. This accommodates finite target intervals, semi-infinite target intervals and infinite target intervals.

Value

A vector of transformed values with four attributes. The first attribute is called "a" and it is the domain interval lower bound. The second attribute is called "b" and it is the domain interval upper bound. The third attribute is called "u" and it is the target interval lower bound. The fourth attribute is called "v" and it is the target interval upper bound.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>, Gregor Gorjanc <gregor.gorjanc@bfro-uni-lj.si>

References


Examples

```r
x <- rnorm( 1000, 0, 10 )
y0 <- scaleX( x, u=0 , v=1 )
y1 <- scaleX( x, u=-1, v=1 )
y2 <- scaleX( x, u=-Inf, v=0 )
y3 <- scaleX( x, u=0, v=Inf )
y4 <- scaleX( x, u=-Inf, v=Inf )
```

Description

This function returns a vector with $n + 1$ elements containing the inner product of an order $k$ shifted Chebyshev polynomial of the first kind, $T^*_k (x)$, with itself (i.e. the norm squared) for orders $k = 0, 1, \ldots, n$.

Usage

```r
schebyshev.t.inner.products(n)
```

Arguments

- `$n$` integer value for the highest polynomial order
Details

The formula used to compute the inner products is as follows.

\[ h_n = \langle T_n^* | T_n^* \rangle = \begin{cases} \frac{\pi}{2} & n \neq 0 \\ \frac{\pi}{2} & n = 0 \end{cases} \]

Value

A vector with \( n + 1 \) elements

1. inner product of order 0 orthogonal polynomial
2. inner product of order 1 orthogonal polynomial
...

\( n+1 \). inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

```r
###
### generate the inner products vector for the
### shifted T Chebyshev polynomials of orders 0 to 10
###
h <- schebyshev.t.inner.products( 10 )
print( h )
```

---

*schebyshev.t.polynomials*

Create list of shifted Chebyshev polynomials

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) shifted Chebyshev polynomials of the first kind, \( T_k^* (x) \), for orders \( k = 0, 1, \ldots, n \).
schebyshev.t.polynomials

Usage

schebyshev.t.polynomials(n, normalized)

Arguments

n integer value for the highest polynomial order
normalized a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function schebyshev.t.recurrences produces a data frame with the recurrence relation parameters for the polynomials. If the normalized argument is FALSE, the function orthogonal.polynomials is used to construct the list of orthogonal polynomial objects. Otherwise, the function orthonormal.polynomials is used to construct the list of orthonormal polynomial objects.

Value

A list of $n + 1$ polynomial objects

1 order 0 shifted Chebyshev polynomial
2 order 1 shifted Chebyshev polynomial
...
$n+1$ order $n$ shifted Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

schebyshev.u.recurrences, orthogonal.polynomials, orthonormal.polynomials
Examples

```r
###
## generate a list of normalized shifted T Chebyshev polynomials of orders 0 to 10
###
normalized.p.list <- schebyshev.t.polynomials(10, normalized=TRUE)
print(normalized.p.list)
###
## generate a list of unnormalized shifted T Chebyshev polynomials of orders 0 to 10
###
unnormalized.p.list <- schebyshev.t.polynomials(10, normalized=FALSE)
print(unnormalized.p.list)
```

`schebyshev.t.recurrences`

*Recurrence relations for shifted Chebyshev polynomials*

**Description**

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) shifted Chebyshev polynomial of the first kind, \( T_k^*(x) \), and for orders \( k = 0, 1, \ldots, n \).

**Usage**

`schebyshev.t.recurrences(n, normalized)`

**Arguments**

- `n` integer value for the highest polynomial order
- `normalized` boolean value which, if `TRUE`, returns recurrence relations for normalized polynomials

**Value**

A data frame with the recurrence relation parameters.

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>

**References**


schebyshev.t.weight

See Also
  
  schebyshev.t.inner.products

Examples

```r
### generate the recurrence relations for
### the normalized shifted T Chebyshev polynomials
### of orders 0 to 10
###
normalized.r <- schebyshev.t.recurrences( 10, normalized=TRUE )
print( normalized.r )
### generate the recurrence relations for
### the unnormalized shifted T Chebyshev polynomials
### of orders 0 to 10
###
unnormalized.r <- schebyshev.t.recurrences( 10, normalized=FALSE )
print( unnormalized.r )
```

schebyshev.t.weight

Weight function for the shifted Chebyshev polynomial

Description

This function returns the value of the weight function for the order \( k \) shifted Chebyshev polynomial of the first kind, \( T_k^*(x) \).

Usage

schebyshev.t.weight(x)

Arguments

x
the function argument which can be a vector

Details

The function takes on non-zero values in the interval \((0, 1)\). The formula used to compute the weight function is as follows.

\[
w(x) = \frac{1}{\sqrt{x-x^2}}
\]

Value

The value of the weight function

Author(s)

Frederick Novomestky < fnovomes@poly.edu >
## References


## Examples

```r
### compute the shifted T Chebyshev weight function for argument values
### between 0 and 1
x <- seq(0, 1, .01)
y <- schebyshev.t.weight(x)
plot(x, y)
```

## schebyshev.u.inner.products

### Inner products of shifted Chebyshev polynomials

#### Description

This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) shifted Chebyshev polynomial of the second kind, \( U_k^*(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

#### Usage

`schebyshev.u.inner.products(n)`

#### Arguments

- \( n \) integer value for the highest polynomial order

#### Details

The formula used to compute the inner products is as follows.

\[
h_n = \langle U_n^* U_n^* \rangle = \frac{\pi}{8}.
\]
Value

A vector with \( n + 1 \) elements

- 1 \( \text{inner product of order 0 orthogonal polynomial} \)
- 2 \( \text{inner product of order 1 orthogonal polynomial} \)
- \( \ldots \)
- \( n+1 \) \( \text{inner product of order} \ n \ \text{orthogonal polynomial} \)

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Examples

\[
h <- \text{schebyshev.u.inner.products}(10)
\]

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) shifted Chebyshev polynomials of the second kind, \( U_k^2(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

\[
\text{schebyshev.u.polynomials}(n, \text{normalized})
\]

Arguments

- \( n \) \hspace{1cm} \text{integer value for highest polynomial order}
- \( \text{normalized} \) \hspace{1cm} a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials
Details

The function `schebyshev.u.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the `normalized` argument is `FALSE`, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

\[
\begin{align*}
1 & \quad \text{order 0 shifted Chebyshev polynomial} \\
2 & \quad \text{order 1 shifted Chebyshev polynomial} \\
\vdots & \\
\text{n+1} & \quad \text{order \( n \) shifted Chebyshev polynomial}
\end{align*}
\]

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

`schebyshev.u.recurrences`, `orthogonal.polynomials`, `orthonormal.polynomials`

Examples

```r
### generate a list of normalized shifted U Chebyshev polynomials of orders 0 to 10
###
normalized.p.list <- schebyshev.u.polynomials(10, normalized=TRUE)
p

### generate a list of unnormalized shifted U Chebyshev polynomials of orders 0 to 10
###
unnormalized.p.list <- schebyshev.u.polynomials(10, normalized=FALSE)
p
```
Recurrence relations for shifted Chebyshev polynomials

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) shifted Chebyshev polynomial of the second kind, \( U_k^* (x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage

\[
\text{schebyshev.u.recurrences}(n, \text{normalized})
\]

Arguments

- \( n \) integer value for the highest polynomial order
- \( \text{normalized} \) boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

`schebyshev.u.inner.products`
Examples

```r
###
### generate the recurrence relations for
### the normalized shifted U Chebyshev polynomials
### of orders 0 to 10
###
### normalized.r <- schebyshev.u.recurrences( 10, normalized=TRUE )
### print( normalized.r )
###
### generate the recurrence relations for
### the unnormalized shifted T Chebyshev polynomials
### of orders 0 to 10
### unnormalized.r <- schebyshev.u.recurrences( 10, normalized=FALSE )
### print( unnormalized.r )
```
References


Examples

```r
### compute the shifted U Chebyshev weight function for argument values
### between 0 and 1
###
x <- seq(0, 1, .01)
y <- schebyshev.u.weight(x)
plot(x, y)
```

slegendre.inner.products

*Inner products of shifted Legendre polynomials*

Description

This function returns a vector with \(n+1\) elements containing the inner product of an order \(k\) shifted Legendre polynomial, \(P_k^*(x)\), with itself (i.e. the norm squared) for orders \(k = 0, 1, \ldots, n\).

Usage

`slegendre.inner.products(n)`

Arguments

- `n` integer value for the highest polynomial order

Details

The formula used to compute the inner products is as follows.

\[
h_n = \langle P_n^* | P_n^* \rangle = \frac{1}{2n+1}.
\]
**Value**

A vector with $n+1$ elements

1. inner product of order 0 orthogonal polynomial
2. inner product of order 1 orthogonal polynomial

... 

$n+1$. inner product of order $n$ orthogonal polynomial

**Author(s)**

Frederick Novomestky <fnovomes@poly.edu>

**References**


**Examples**

```r
###
### compute the inner products vector for the
### shifted Legendre polynomials of orders 0 to 10
###
h <- slegendre.inner.products(10)
print(h)
```

---

**slegendre.polynomials**  
*Create list of shifted Legendre polynomials*

**Description**

This function returns a list with $n+1$ elements containing the order $k$ shifted Legendre polynomials, $P_k^* (x)$, for orders $k = 0, 1, \ldots, n$.

**Usage**

```r
slegendre.polynomials(n, normalized=FALSE)
```

**Arguments**

- `n`  
  integer value for the highest polynomial order

- `normalized`  
  a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials
Details

The function `slegendre.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the normalized argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.

Value

A list of \(n+1\) polynomial objects

1. order 0 shifted Legendre polynomial
2. order 1 shifted Legendre polynomial
...
\(n+1\). order \(n\) shifted Legendre polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

`slegendre.recurrences`, `orthogonal.polynomials`, `orthonormal.polynomials`

Examples

```r
### generate a list of normalized shifted Legendre polynomials of orders 0 to 10
###
normalized.p.list <- slegendre.polynomials(10, normalized=TRUE)
print(normalized.p.list)
###
### generate a list of unnormalized shifted Legendre polynomials of orders 0 to 10
###
unnormalized.p.list <- slegendre.polynomials(10, normalized=FALSE)
print(unnormalized.p.list)
```
slegendre.recurrences  

Recurrence relations for shifted Legendre polynomials

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) shifted Legendre polynomial, \( P^*_k(x) \), and for orders \( k = 0, 1, \ldots, n \).

Usage

slegendre.recurrences(n, normalized=FALSE)

Arguments

- \( n \)  
  integer value for the highest polynomial order
- \( \text{normalized} \)  
  boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

slegendre.inner.products,
**Examples**

```r
### generate the recurrence relations for normalized shifted Legendre polynomials
### of orders 0 to 10
###
normalized.r <- slegendre.recurrences(10, normalized=TRUE)
print(normalized.r)

### generate the recurrence relations for normalized shifted Legendre polynomials
### of orders 0 to 10
###
unnormalized.r <- slegendre.recurrences(10, normalized=FALSE)
print(unnormalized.r)
```

---

**slegendre.weight**  
*Weight function for the shifted Legendre polynomial*

**Description**

This function returns the value of the weight function for the order $k$ shifted Legendre polynomial, $P_k^*(x)$.

**Usage**

`slegendre.weight(x)`

**Arguments**

- `x`  
  the function argument which can be a vector

**Details**

The function takes on non-zero values in the interval $(0, 1)$. The formula used to compute the weight function is as follows.

$$w(x) = 1$$

**Value**

The value of the weight function

**Author(s)**

Frederick Novomestky < fnovomes@poly.edu >
References

Examples
###
### compute the shifted Legendre weight function for argument values
### between 0 and 1
###
x <- seq(0, 1, .01)
y <- slegendre.weight(x)

spherical.inner.products

*Inner products of spherical polynomials*

Description
This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) spherical polynomial, \( P_k(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage
spherical.inner.products(n)

Arguments
\( n \) integer value for the highest polynomial order

Details
The formula used to compute the inner products of the spherical orthogonal polynomials is the same as that used for the Legendre orthogonal polynomials.

Value
A vector with \( n + 1 \) elements
1 inner product of order 0 orthogonal polynomial
2 inner product of order 1 orthogonal polynomial
... 
\( n+1 \) inner product of order \( n \) orthogonal polynomial
spherical.polynomials

Author(s)
Frederick Novomestky <fnovomes@poly.edu>

References

See Also
legendre.inner.products

Examples
```r
###
### generate the inner products vector for the spherical polynomials
### of orders 0 to 10.
###
h <- spherical.inner.products( 10 )
print( h )
```

spherical.polynomials Create list of spherical polynomials

Description
This function returns a list with \( n+1 \) elements containing the order \( k \) spherical polynomials, \( P_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage
spherical.polynomials(n, normalized=FALSE)

Arguments
\n n integer value for the highest polynomial order
normalized a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details
The function spherical.recurrences produces a data frame with the recurrence relation parameters for the polynomials. If the normalized argument is FALSE, the function orthogonal.polynomials is used to construct the list of orthogonal polynomial objects. Otherwise, the function orthonormal.polynomials is used to construct the list of orthonormal polynomial objects.
spherical.recurrences

Value

A list of \( n + 1 \) polynomial objects

1  order 0 spherical polynomial
2  order 1 spherical polynomial
...
\( n+1 \) order \( n \) Chebyshev polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

spherical.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

```r
### generate a list of spherical orthonormal polynomials of orders 0 to 10
### normalized.p.list <- spherical.polynomials( 10, normalized=TRUE )
print( normalized.p.list )
###
### generate a list of spherical orthogonal polynomials of orders 0 to 10
###
### unnormalized.p.list <- spherical.polynomials( 10, normalized=FALSE )
print( unnormalized.p.list )
```

spherical.recurrences  Recurrence relations for spherical polynomials

Description

This function returns a data frame with \( n + 1 \) rows and four named columns containing the coefficient vectors \( c, d, e \) and \( f \) of the recurrence relations for the order \( k \) spherical polynomial, \( P_k(x) \), and for orders \( k = 0, 1, \ldots, n \).
spherical.recurrences

Usage

spherical.recurrences(n, normalized=FALSE)

Arguments

  n  integer value for the highest polynomial order
  normalized  boolean value which, if TRUE, returns recurrence relations for normalized polyno-
              mials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


Szegö, G., 1939. *Orthogonal Polynomials*, 23, American Mathematical Society Colloquium Publica-
              tions, Providence, RI.

See Also

  spherical.inner.products

Examples

```r
### generate the recurrence relations for
### the normalized spherical polynomials
### of orders 0 to 10
###
normalized.r <- spherical.recurrences( 10, normalized=TRUE )
print( normalized.r )
###
### generate the recurrence relations for
### the unnormalized spherical polynomials
### of orders 0 to 10
###
unnormalized.r <- spherical.recurrences( 10, normalized=FALSE )
print( unnormalized.r )
```
spherical.weight  Weight function for the spherical polynomial

Description
This function returns the value of the weight function for the order \( k \) spherical polynomial, \( P_k(x) \).

Usage
spherical.weight(x)

Arguments
x  the function argument which can be a vector or matrix

Details
The function takes on non-zero values in the interval \((-1, 1)\). The formula used to compute the weight function is as follows.

\[ w(x) = 1 \]

Value
The value of the weight function

Author(s)
Frederick Novomestky < fnovomes@poly.edu >

References

Examples
```r
### compute the spherical weight function for a sequence of values between -2 and 2
###
x <- seq(-2, 2, .01)
y <- spherical.weight(x)
plot(x, y)
```
ultraspherical.inner.products

Inner products of ultraspherical polynomials

Description

This function returns a vector with \( n + 1 \) elements containing the inner product of an order \( k \) ultraspherical polynomial, \( C_k^{(\alpha)}(x) \), with itself (i.e. the norm squared) for orders \( k = 0, 1, \ldots, n \).

Usage

ultraspherical.inner.products(n,alpha)

Arguments

- \( n \): integer value for the highest polynomial order
- \( \alpha \): numeric value for the polynomial parameter

Details

This function uses the same formula as the function gegenbauer.inner.products.

Value

A vector with \( n + 1 \) elements

1. inner product of order 0 orthogonal polynomial
2. inner product of order 1 orthogonal polynomial
...
\( n+1 \) inner product of order \( n \) orthogonal polynomial

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References

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

See Also

gegenbauer.inner.products
Examples

```r
###
### generate the inner products vector for the
### ultraspherical polynomials of orders 0 to 10.
### the polynomial parameter is 1.0
###
h <- ultraspherical.inner.products(10, 1)
print(h)
```

ultraspherical.polynomials

*Create list of ultraspherical polynomials*

Description

This function returns a list with \( n + 1 \) elements containing the order \( k \) ultraspherical polynomials, \( C^{(\alpha)}_k(x) \), for orders \( k = 0, 1, \ldots, n \).

Usage

```r
ultraspherical.polynomials(n, alpha, normalized=FALSE)
```

Arguments

- \( n \) integer value for the highest polynomial order
- \( \alpha \) polynomial parameter
- \( \text{normalized} \) a boolean value which, if TRUE, returns a list of normalized orthogonal polynomials

Details

The function `ultraspherical.recurrences` produces a data frame with the recurrence relation parameters for the polynomials. If the `normalized` argument is FALSE, the function `orthogonal.polynomials` is used to construct the list of orthogonal polynomial objects. Otherwise, the function `orthonormal.polynomials` is used to construct the list of orthonormal polynomial objects.

Value

A list of \( n + 1 \) polynomial objects

```
1 order 0 ultraspherical polynomial
2 order 1 ultraspherical polynomial
...
n+1 order n ultraspherical polynomial
```
ultraspherical.recurrences

Author(s)
Frederick Novomestky <fnovomes@poly.edu>

References


See Also
gegenbauer.recurrences, orthogonal.polynomials, orthonormal.polynomials

Examples

```r
### generate a list of normalized ultra spherical polynomials
### of orders 0 to 10
###
normalized.p.list <- ultraspherical.polynomials(10, 1, normalized=TRUE)
print(normalized.p.list)
###
### generate a list of unnormalized ultra spherical polynomials
### of orders 0 to 10
###
unnormalized.p.list <- ultraspherical.polynomials(10, 1, normalized=FALSE)
print(unnormalized.p.list)
```

ultraspherical.recurrences

Recurrence relations for ultraspherical polynomials

Description
This function returns a data frame with \(n + 1\) rows and four named columns containing the coefficient vectors \(c\), \(d\), \(e\) and \(f\) of the recurrence relations for the order \(k\) ultraspherical polynomial, \(C_k^{(\alpha)}(x)\), and for orders \(k = 0, 1, \ldots, n\).

Usage
ultraspherical.recurrences(n, alpha, normalized=FALSE)
Arguments

- **n**: integer value for the highest polynomial order
- **alpha**: numeric value for the polynomial parameter
- **normalized**: boolean value which, if TRUE, returns recurrence relations for normalized polynomials

Value

A data frame with the recurrence relation parameters.

Author(s)

Frederick Novomestky <fnovomes@poly.edu>

References


See Also

- *ultraspherical.recurrences*

Examples

```R
###
### generate the recurrence relations for
### the normalized ultraspherical polynomials
### of orders 0 to 10
### polynomial parameter value is 1.0
###
### normalized.r <- ultraspherical.recurrences( 10, 1, normalized=TRUE )
### print( normalized.r )
###
### generate the recurrence relations for
### the normalized ultraspherical polynomials
### of orders 0 to 10
### polynomial parameter value is 1.0
###
### unnormalized.r <- ultraspherical.recurrences( 10, 1, normalized=FALSE )
### print( unnormalized.r )
```
ultraspherical.weight  Weight function for the ultraspherical polynomial

Description
This function returns the value of the weight function for the order $k$ ultraspherical polynomial, $C_k^{(\alpha)}(x)$.

Usage
ultraspherical.weight(x,alpha)

Arguments
x  the function argument which can be a vector
alpha  polynomial parameter

Details
The function takes on non-zero values in the interval $(-1, 1)$. The formula used to compute the weight function is as follows.

$$w(x) = (1 - x^2)^{\alpha - 0.5}$$

Value
The value of the weight function

Author(s)
Frederick Novomestky < fnovomes@poly.edu >

References


Szegö, G., 1939. Orthogonal Polynomials, 23, American Mathematical Society Colloquium Publications, Providence, RI.
Examples

```r
###
### compute the ultraspherical weight function for arguments between -2 and 2
### polynomial parameter is 1.0
###
x <- seq(-2, 2, 0.01)
y <- ultraspherical.weight(x, 1)
plot(x, y)
```
Index

*Topic math

    chebyshev.c.inner.products, 3
    chebyshev.c.polynomials, 5
    chebyshev.c.recurrences, 6
    chebyshev.c.weight, 7
    chebyshev.s.inner.products, 8
    chebyshev.s.polynomials, 10
    chebyshev.s.recurrences, 11
    chebyshev.s.weight, 12
    chebyshev.t.inner.products, 13
    chebyshev.t.polynomials, 15
    chebyshev.t.recurrences, 16
    chebyshev.t.weight, 17
    chebyshev.u.inner.products, 18
    chebyshev.u.polynomials, 20
    chebyshev.u.recurrences, 21
    chebyshev.u.weight, 22
    gegenbauer.inner.products, 23
    gegenbauer.polynomials, 25
    gegenbauer.recurrences, 26
    gegenbauer.weight, 27
    ghermite.h.inner.products, 28
    ghermite.h.polynomials, 30
    ghermite.h.recurrences, 31
    ghermite.h.weight, 33
    laguerre.inner.products, 34
    laguerre.polynomials, 35
    laguerre.recurrences, 37
    laguerre.weight, 38
    hermite.h.inner.products, 39
    hermite.h.polynomials, 40
    hermite.h.recurrences, 42
    hermite.h.weight, 43
    hermite.he.inner.products, 44
    hermite.he.polynomials, 45
    hermite.he.recurrences, 46
    hermite.he.weight, 48
    jacobi.g.inner.products, 49
    jacobi.g.polynomials, 50
    jacobi.g.recurrences, 51
    jacobi.g.weight, 53
    jacobi.matrices, 54
    jacobi.p.inner.products, 55
    jacobi.p.polynomials, 56
    jacobi.p.recurrences, 57
    jacobi.p.weight, 58
    laguerre.inner.products, 60
    laguerre.polynomials, 61
    laguerre.recurrences, 62
    laguerre.weight, 63
    legendre.inner.products, 64
    legendre.polynomials, 66
    legendre.recurrences, 67
    legendre.weight, 68
    lpochhammer, 69
    monic.polynomial.recurrences, 70
    monic.polynomials, 71
    orthogonal.polynomials, 73
    orthonormal.polynomials, 74
    pochhammer, 75
    polynomial.coefficients, 76
    polynomial.derivatives, 77
    polynomial.functions, 78
    polynomial.integrals, 79
    polynomial.orders, 80
    polynomial.powers, 81
    polynomial.roots, 82
    polynomial.values, 83
    scaleX, 84
    schebyshev.t.inner.products, 85
    schebyshev.t.polynomials, 86
    schebyshev.t.recurrences, 88
    schebyshev.t.weight, 89
    schebyshev.u.inner.products, 90
    schebyshev.u.polynomials, 91
    schebyshev.u.recurrences, 93
    schebyshev.u.weight, 94
    slegendre.inner.products, 95

111
Index

schebyshev.u.weight, 94
slegendre.inner.products, 95, 98
slegendre.polynomials, 96
slegendre.recurrences, 97, 98
slegendre.weight, 99

spherical.inner.products, 65, 100, 103
spherical.polynomials, 101
spherical.recurrences, 102, 102
spherical.weight, 104

ultraspherical.inner.products, 24, 105
ultraspherical.polynomials, 106
ultraspherical.recurrences, 107, 108
ultraspherical.weight, 109