# Package 'chebInterp'

February 9, 2019

**Type** Package

**Title** Chebyshev Polynomial Interpolation

**Version** 0.1.0

**Maintainer** Walter Zhang <walterwzhang@chicagobooth.edu>

**Description** Chebyshev polynomial interpolation routines

**License** GPL-3

**Depends** R (>= 3.1.0)

**Suggests** parallel, knitr, rmarkdown, reshape

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 6.1.1

## R topics documented:



---

`calculateChebyshevCoefficients` *Computes the Chebyshev coefficients from a given function and cheb list*

---

### Description

Also checks to ensure the cheb$T matrix is orthogonal The rounding down to 0 in the beginning is to account for numerical precision and is controlled by the tolerance parameter The function f only takes one argument

### Usage

```
calculateChebyshevCoefficients(f, cheb, tolerance = 1e-12)
```

### Arguments

| | |
|---|---|
| f | Function to be approximated (function) |
| cheb | List of item from initalizeChebyshevApproximator (list) |
| tolerance | Numerical Tolerance for rounding down |

### Value

A list of Chebyshev coefficients (matrix)

---

## calculateChebyshevPolynomials

*Computes the polynomials for a given degree and vector of values.*

---

### Description

Resultant matrix of polynomials is of size length(x) by N + 1

### Usage

```
calculateChebyshevPolynomials(x, N)
```

### Arguments

| | |
|---|---|
| x | Vector of values to compute the polynomials at (numeric) |
| N | Highest Degree of the Polynomial (Integer) |

### Value

A matrix of the polynomials (matrix)

---

## evaluateChebyshev

*Evaluates the Chebyshev Approximation for a matrix (or a vector) of points*

---

### Description

Option for parallelized evaluation for many points to evaluate

### Usage

```
evaluateChebyshev(x, cheb, parallel = FALSE, numcores = 1L)
```

### Arguments

| | |
|---|---|
| x | Points to evaluate with size Points by Dimensions (matrix) |
| cheb | List of item from initalizeChebyshevApproximator (list) |
| parallel | Boolean flag for parallelization (logical) |
| numcores | Cores for parallelization (integer) |

## Value

A vector of predictions for each point of x

---

`evaluateChebyshev_T` *Evaluates the Chebyshev Approximation for a matrix (or a vector) of points and returns the underlying basis function values instead of the interpolation values*

---

## Description

Option for parallelized evaluation for many points to evaluate

## Usage

```
evaluateChebyshev_T(x, cheb, parallel = FALSE, numcores = 1L)
```

## Arguments

| | |
|---|---|
| `x` | Points to evaluate with size Points by Dimensions (matrix) |
| `cheb` | List of item from initalizeChebyshevApproximator (list) |
| `parallel` | Boolean flag for parallelization (logical) |
| `numcores` | Cores for parallelization (integer) |

## Value

A matrix of the underlying basis function values

---

`initializeChebyshevApproximator` *Initializes the Chebyshev Approximation*

---

## Description

Initializes the Chebyshev Approximation

## Usage

```
initializeChebyshevApproximator(D, N, M = N + 1, bounds = NULL,
  upper_b = NULL, lower_b = NULL)
```

## Arguments

| | |
|---|---|
| `D` | Dimensions of the Problem (integer) |
| `N` | Highest Degree of the Polynomial (integer) |
| `M` | Number of Interpolation Nodes in each dimension (integer) |
| `bounds` | Bounds of the rectangle on which the function is approximated (list) |
| `upper_b` | A vector of upper bounds (numeric) |
| `lower_b` | A vector of lower bounds (numeric) |

## Value

A list of the initialized approximation

# Index