

Combustion of Packed Pellets of Core-Shell Particle

Generated by Doxygen 1.8.17

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 GivensRotationMatrix< real_t > Class Template Reference	7
4.1.1 Detailed Description	8
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 GivensRotationMatrix()	8
4.1.3 Member Function Documentation	8
4.1.3.1 multiply()	9
4.1.3.2 setupRotationMatrix()	9
4.2 LowerTriangularMatrix< real_t > Class Template Reference	9
4.2.1 Detailed Description	10
4.2.2 Constructor & Destructor Documentation	10
4.2.2.1 LowerTriangularMatrix()	10
4.2.2.2 ~LowerTriangularMatrix()	11
4.2.3 Member Function Documentation	11
4.2.3.1 getElement()	11
4.2.3.2 getIndex()	11
4.2.3.3 indexOfZeroElement()	12
4.2.3.4 printMatrix()	12
4.2.3.5 setElement()	12
4.3 QRSolver Class Reference	13
4.3.1 Detailed Description	14
4.3.2 Constructor & Destructor Documentation	14
4.3.2.1 QRSolver()	14
4.3.2.2 ~QRSolver()	14
4.3.3 Member Function Documentation	15
4.3.3.1 getIndex()	15
4.4 TridiagonalMatrix< real_t > Class Template Reference	15
4.4.1 Detailed Description	16
4.4.2 Constructor & Destructor Documentation	16
4.4.2.1 TridiagonalMatrix()	16
4.4.2.2 ~TridiagonalMatrix()	17
4.4.3 Member Function Documentation	17
4.4.3.1 getElement()	17
4.4.3.2 getIndex()	17

4.4.3.3 <code>indexOfZeroElement()</code>	19
4.4.3.4 <code>print()</code>	19
4.4.3.5 <code>printMatrix()</code>	20
4.4.3.6 <code>setElement()</code>	20
4.4.4 Member Data Documentation	20
4.4.4.1 <code>array</code>	20
4.4.4.2 <code>N</code>	20
5 File Documentation	21
5.1 <code>combustion-packed-pellet-core-shell-particles/examples/Lower_Triangular_Matrix_Example.cpp</code> File Reference	21
5.1.1 Detailed Description	21
5.2 <code>combustion-packed-pellet-core-shell-particles/examples/QR_Solver_Example.cpp</code> File Reference	22
5.2.1 Detailed Description	22
5.3 <code>combustion-packed-pellet-core-shell-particles/include/GivensRotationMatrix.hpp</code> File Reference	22
5.3.1 Detailed Description	23
5.4 <code>combustion-packed-pellet-core-shell-particles/include/Lower_Triangular_Matrix.hpp</code> File Reference	23
5.4.1 Detailed Description	23
5.5 <code>combustion-packed-pellet-core-shell-particles/include/QR_Solver.hpp</code> File Reference	24
5.5.1 Detailed Description	24
5.6 <code>combustion-packed-pellet-core-shell-particles/include/Tridiagonal_Matrix.hpp</code> File Reference	24
5.6.1 Detailed Description	25
5.7 <code>combustion-packed-pellet-core-shell-particles/src/Lower_Triangular_Matrix.cpp</code> File Reference	25
5.7.1 Detailed Description	25
5.8 <code>combustion-packed-pellet-core-shell-particles/src/Tridiagonal_Matrix.cpp</code> File Reference	26
5.8.1 Detailed Description	26
Index	27

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

GivensRotationMatrix< real_t >	7
LowerTriangularMatrix< real_t >	9
TridiagonalMatrix< real_t >	15
QRSolver	13

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

GivensRotationMatrix< real_t >	
Class to implement a memory efficient Givens' Rotation Matrix that can vanish the element $a_{k+1,k}$ of a tridiagonal matrix A	7
LowerTriangularMatrix< real_t >	
Class to implement a memory efficient model of Lower Triangular Square Matrix	9
QRSolver	
Class to implement QR factorization algorithm for solving matrix equations of the $A.x = b$ where A is a $n \times n$ tridiagonal matrix and x and b are $n \times 1$ vectors	13
TridiagonalMatrix< real_t >	
Class to implement a memory efficient 2D Tridiagonal square matrix	15

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

combustion-packed-pellet-core-shell-particles/examples/ Lower_Triangular_Matrix_Example.cpp	
Example to test LowerTriangularMatrix class	21
combustion-packed-pellet-core-shell-particles/examples/ QR_Solver_Example.cpp	
Example cpp file to test out QR_Solver functions	22
combustion-packed-pellet-core-shell-particles/examples/ Tridiagonal_Matrix_Example.cpp	??
combustion-packed-pellet-core-shell-particles/include/ GivensRotationMatrix.hpp	
This header serves the definition of an implementation of Givens' Rotation matrix. The rotation matrix is used to solve matrix equations through QR factorization method, particularly tridiagonal matrix equation	22
combustion-packed-pellet-core-shell-particles/include/ Lower_Triangular_Matrix.hpp	
This header file defines a class for memory efficient implementation of lower triangular square matrices	23
combustion-packed-pellet-core-shell-particles/include/ QR_Solver.hpp	
This header file defines a class for solving 2D matrix equations of the form $A \cdot x = b$ (where A is an $n \times n$ matrix and x and b are $n \times 1$ vectors) using QR factorization technique. Also the class is implemented in such a way that it may be parallelized easily using openmp constructs	24
combustion-packed-pellet-core-shell-particles/include/ Tridiagonal_Matrix.hpp	
This header file serves the definition of an implementation for a Tridiagonal Matrix	24
combustion-packed-pellet-core-shell-particles/scripts/ QR_Factorization.py	??
combustion-packed-pellet-core-shell-particles/src/ Lower_Triangular_Matrix.cpp	25
combustion-packed-pellet-core-shell-particles/src/ main.cpp	??
combustion-packed-pellet-core-shell-particles/src/ QR_Solver.cpp	??
combustion-packed-pellet-core-shell-particles/src/ Tridiagonal_Matrix.cpp	
Implementation of Tridiagonal Matrix	26

Chapter 4

Class Documentation

4.1 GivensRotationMatrix< real_t > Class Template Reference

Class to implement a memory efficient Givens' Rotation Matrix that can vanish the element $a_{k+1,k}$ of a tridiagonal matrix A .

```
#include <GivensRotationMatrix.hpp>
```

Public Member Functions

- [GivensRotationMatrix](#) ()
Construct a new Givens Rotation Matrix.
- void [setupRotationMatrix](#) ([TridiagonalMatrix](#)< real_t > &matrix, unsigned int index)
Sets up the Givens Rotation Matrix to vanish the element at position $(i + 1, i)$ of tridiagonal matrix.
- void [multiply](#) ([TridiagonalMatrix](#)< real_t > &matrix)
Multiplies the Givens Rotation Matrix to the Tridiagonal Matrix and updates the tridiagonal matrix in place.
- void **multiply** ([LowerTriangularMatrix](#)< real_t > &matrix)

Private Attributes

- real_t **element_i_i**
- real_t **element_i_ip1**
- real_t **element_ip1_i**
- real_t **element_ip1_ip1**
- unsigned int **i**

4.1.1 Detailed Description

```
template<typename real_t>
class GivensRotationMatrix< real_t >
```

Class to implement a memory efficient Givens' Rotation Matrix that can vanish the element $a_{k+1,k}$ of a tridiagonal matrix A .

A Givens rotation matrix is an orthogonal such that $G_{N \times N} = [g_{ij}]_{N \times N}$ where

$$g_{ij} = \begin{cases} 1 & i = j \neq k, k+1 \\ \cos \theta & i = j = k, k+1 \\ \sin \theta & i = k, j = k+1 \\ -\sin \theta & i = k+1, j = k \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

When the Givens Rotation Matrix is multiplied to another matrix $B_{N \times N} = [b_{i,j}]_{N \times N}$

$$C_{N \times N} = G_{N \times N} \cdot B_{N \times N} \quad (4.2)$$

$$\Rightarrow c_{ij} = \sum_{l=1}^N g_{i,l} b_{l,j} \quad (4.3)$$

$$\Rightarrow c_{ij} = \begin{cases} g_{k,k} b_{k,j} + g_{k,k+1} b_{k+1,j} & i = k \\ g_{k+1,k} b_{k+1,j} + g_{k+1,k+1} b_{k+1,j} & i = k+1 \\ b_{i,j} & \text{otherwise} \end{cases} \quad (4.4)$$

Template Parameters

<i>real_t</i>	float, double or long double data types to represent real numbers
---------------	---

Definition at line 57 of file GivensRotationMatrix.hpp.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 GivensRotationMatrix()

```
template<typename real_t >
GivensRotationMatrix< real_t >::GivensRotationMatrix ( )
```

Construct a new Givens Rotation Matrix.

4.1.3 Member Function Documentation

4.1.3.1 multiply()

```
template<typename real_t >
void GivensRotationMatrix< real_t >::multiply (
    TridiagonalMatrix< real_t > & matrix )
```

Multiplies the Givens Rotation Matrix to the Tridiagonal Matrix and updates the tridiagonal matrix in place.

Parameters

<i>matrix</i>	Tridiagonal Matrix that will be converted into upper tridiagonal matrix after multiplication
---------------	--

For multiplication with a Tridiagonal matrix

4.1.3.2 setupRotationMatrix()

```
template<typename real_t >
void GivensRotationMatrix< real_t >::setupRotationMatrix (
    TridiagonalMatrix< real_t > & matrix,
    unsigned int index )
```

Sets up the Givens Rotation Matrix to vanish the element at position $(i + 1, i)$ of tridiagonal matrix.

Parameters

<i>matrix</i>	Tridiagonal matrix whose element at position $(i + 1, i)$ needs to be vanished
<i>index</i>	i

The documentation for this class was generated from the following file:

- combustion-packed-pellet-core-shell-particles/include/[GivensRotationMatrix.hpp](#)

4.2 LowerTriangularMatrix< real_t > Class Template Reference

Class to implement a memory efficient model of Lower Triangular Square Matrix.

```
#include <Lower_Triangular_Matrix.hpp>
```

Public Member Functions

- [LowerTriangularMatrix](#) (unsigned int n)
Construct a new Lower Triangular Matrix object.
- [~LowerTriangularMatrix](#) ()
Destroy the Lower Triangular Matrix object.
- [real_t getElement](#) (unsigned int row_index, unsigned int column_index)
Get the i, j th element of the Lower Triangular Matrix.
- void [setElement](#) (unsigned int row_index, unsigned int column_index, real_t value)
Set the value of the i, j th element of the Lower Triangular Matrix.
- void [printMatrix](#) ()
Prints the Lower Triangular Matrix in form of a 2D array.

Private Member Functions

- unsigned int [getIndex](#) (unsigned int row_index, unsigned int column_index)
Get the index of i, j th element of lower triangular matrix.
- bool [indexOfZeroElement](#) (unsigned int row_index, unsigned int column_index)
Checks if the row index i and the column index j belong to a zero element of the lower triangular matrix.

Private Attributes

- real_t * [array](#)
One dimensional array to store only non zero elements of the lower triangular matrix.
- const unsigned int [N](#)
Size of main diagonal of $N \times N$ square matrix.

4.2.1 Detailed Description

```
template<typename real_t>
class LowerTriangularMatrix< real_t >
```

Class to implement a memory efficient model of Lower Triangular Square Matrix.

Template Parameters

<i>real_t</i>	float, double or long double data types to represent real numbers
---------------	---

Definition at line 23 of file Lower_Triangular_Matrix.hpp.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 LowerTriangularMatrix()

```
template<typename real_t >
LowerTriangularMatrix< real_t >::LowerTriangularMatrix (
    unsigned int n )
```

Construct a new Lower Triangular Matrix object.

Parameters

<i>n</i>	Number of rows in the $N \times N$ square matrix
----------	--

Definition at line 17 of file Lower_Triangular_Matrix.cpp.

4.2.2.2 ~LowerTriangularMatrix()

```
template<typename real_t >
LowerTriangularMatrix< real_t >::~~LowerTriangularMatrix
```

Destroy the Lower Triangular Matrix object.

Definition at line 30 of file Lower_Triangular_Matrix.cpp.

4.2.3 Member Function Documentation

4.2.3.1 getElement()

```
template<typename real_t >
real_t LowerTriangularMatrix< real_t >::getElement (
    unsigned int row_index,
    unsigned int column_index )
```

Get the i,j th element of the Lower Triangular Matrix.

Parameters

<i>row_index</i>	Row index i
<i>column_index</i>	Column index j

Returns

Value of the i,j th element of the Lower Triangular Matrix

Definition at line 58 of file Lower_Triangular_Matrix.cpp.

4.2.3.2 getIndex()

```
template<typename real_t >
unsigned int LowerTriangularMatrix< real_t >::getIndex (
    unsigned int row_index,
    unsigned int column_index ) [private]
```

Get the index of i,j th element of lower triangular matrix.

Parameters

<i>row_index</i>	Row index i
<i>column_index</i>	Column index j

Returns

Index of the i,j th element in the flattened array representation

Definition at line 37 of file Lower_Triangular_Matrix.cpp.

4.2.3.3 indexOfZeroElement()

```
template<typename real_t >
bool LowerTriangularMatrix< real_t >::indexOfZeroElement (
    unsigned int row_index,
    unsigned int column_index ) [private]
```

Checks if the row index i and the column index j belong to a zero element of the lower triangular matrix.

Parameters

<i>row_index</i>	Row index i
<i>column_index</i>	Column index j

Returns

true if i,j are indices of zero elements in a lower triangular matrix
false if i,j are indices of non zero elements in a lower triangular matrix

Definition at line 48 of file Lower_Triangular_Matrix.cpp.

4.2.3.4 printMatrix()

```
template<typename real_t >
void LowerTriangularMatrix< real_t >::printMatrix
```

Prints the Lower Triangular Matrix in form of a 2D array.

Definition at line 97 of file Lower_Triangular_Matrix.cpp.

4.2.3.5 setElement()

```
template<typename real_t >
void LowerTriangularMatrix< real_t >::setElement (
    unsigned int row_index,
    unsigned int column_index,
    real_t value )
```

Set the value of the i,j th element of the Lower Triangular Matrix.

Parameters

<i>row_index</i>	Row index i
<i>column_index</i>	Column index j
<i>value</i>	Value to be set at the i,j th element

Definition at line 76 of file Lower_Triangular_Matrix.cpp.

The documentation for this class was generated from the following files:

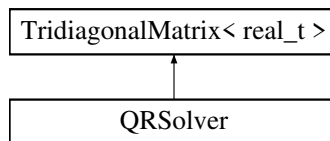
- combustion-packed-pellet-core-shell-particles/include/Lower_Triangular_Matrix.hpp
- combustion-packed-pellet-core-shell-particles/src/Lower_Triangular_Matrix.cpp

4.3 QRSolver Class Reference

Class to implement QR factorization algorithm for solving matrix equations of the $A.x = b$ where A is a $n \times n$ tridiagonal matrix and x and b are $n \times 1$ vectors.

```
#include <QR_Solver.hpp>
```

Inheritance diagram for QRSolver:



Public Member Functions

- [QRSolver](#) (unsigned int N)
Construct a new [QRSolver](#) object.
- [~QRSolver](#) ()
Destroy the [QRSolver](#) object.
- void **QRfactorize** ()
- void **initQ** ()

Private Member Functions

- void [loadR](#) ()
Loads the values of R matrix that will change during multiplication with Givens' rotation matrix to temporary variables.
- void [setupGivensRotationMatrix](#) ()
Setup Givens' rotation matrix.
- void [multiplyGivensMatrixWithR](#) ()
Multiplies the Givens rotation matrix with R matrix and updates its value.
- void [multiplyGivensMatrixWithQ](#) ()
Multiplies the Givens rotation matrix with R matrix and updates its value.
- unsigned int [getIndex](#) (unsigned int row_index, unsigned int column_index)
Get the index in a flattened linear array representation of a 2D matrix.

Private Attributes

- `real_t * Q`
- `real_t G_k_k`
- `real_t G_k_kp1`
- `real_t G_kp1_k`
- `real_t G_kp1_kp1`
- `real_t R_k_k`
- `real_t R_k_kp1`
- `real_t R_kp1_k`
- `real_t R_kp1_kp1`
- `real_t R_kp1_kp2`
- `unsigned int k`

4.3.1 Detailed Description

Class to implement QR factorization algorithm for solving matrix equations of the $A.x = b$ where A is a $n \times n$ tridiagonal matrix and x and b are $n \times 1$ vectors.

Definition at line 27 of file `QR_Solver.hpp`.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 QRSolver()

```
QRSolver::QRSolver (
    unsigned int N )
```

Construct a new [QRSolver](#) object.

Parameters

N	
-----	--

Definition at line 5 of file `QR_Solver.cpp`.

4.3.2.2 ~QRSolver()

```
QRSolver::~QRSolver ( )
```

Destroy the [QRSolver](#) object.

Definition at line 14 of file `QR_Solver.cpp`.

4.3.3 Member Function Documentation

4.3.3.1 getIndex()

```
unsigned int QRSolver::getIndex (
    unsigned int row_index,
    unsigned int column_index ) [inline], [private]
```

Get the index in a flattened linear array representation of a 2D matrix.

Parameters

<i>row_index</i>	Row index i of the desired element
<i>column_index</i>	Column index j of the desired element

Returns

Index in a linear array of the i,j element of a 2D matrix implemented using the linear array

Definition at line 87 of file QR_Solver.hpp.

The documentation for this class was generated from the following files:

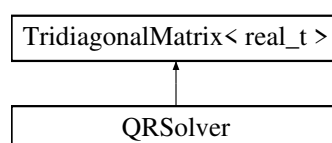
- combustion-packed-pellet-core-shell-particles/include/QR_Solver.hpp
- combustion-packed-pellet-core-shell-particles/src/QR_Solver.cpp

4.4 TridiagonalMatrix< real_t > Class Template Reference

Class to implement a memory efficient 2D Tridiagonal square matrix.

```
#include <Tridiagonal_Matrix.hpp>
```

Inheritance diagram for TridiagonalMatrix< real_t >:



Public Member Functions

- [TridiagonalMatrix](#) (unsigned int n)
Construct a new Tridiagonal Matrix.
- [~TridiagonalMatrix](#) ()
Destroy the Tridiagonal Matrix.
- [real_t getElement](#) (unsigned int row_index, unsigned int column_index)
Get the i,j th element of 2D Tridiagonal Matrix.
- [void setElement](#) (unsigned int row_index, unsigned int column_index, [real_t](#) value)
Set the value of the i,j th element of 2D Tridiagonal Matrix.
- [void printMatrix](#) ()
Prints the Tridiagonal Matrix in form of a 2D array.
- [void print](#) ()
Prints the Tridiagonal Matrix in form of a flattened array.

Private Member Functions

- unsigned int [getIndex](#) (unsigned int row_index, unsigned int column_index)
Get the index of the i,j th element of Tridiagonal 2D Matrix.
- bool [indexOfZeroElement](#) (unsigned int row_index, unsigned int column_index)
Checks if the row index i and the column index j belong to a zero element of the tridiagonal matrix.

Private Attributes

- [real_t * array](#)
*Flattened array of size $4 * N$ to represent 2D Tridiagonal matrix of size $N \times N$.*
- const unsigned int [N](#)
Size of main diagonal of the $N \times N$ 2D Tridiagonal Matrix.

4.4.1 Detailed Description

```
template<typename real_t>
class TridiagonalMatrix< real_t >
```

Class to implement a memory efficient 2D Tridiagonal square matrix.

The class is specifically built for implementation in a QR factorization algorithm for solving matrix equations of the form $A \cdot x = b$. The QR algo converts a normal tridiagonal matrix (a matrix with non zero entries only at indices $(i, i - 1)$, (i, i) and $(i, i + 1)$) to an upper tridiagonal matrix (a matrix with non zero entries only at indices (i, i) , $(i, i + 1)$ and $(i, i + 2)$). Thus only indices $(i, i - 1)$, (i, i) , $(i, i + 1)$ and $(i, i + 2)$ are accessible for this matrix

Definition at line 30 of file Tridiagonal_Matrix.hpp.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 TridiagonalMatrix()

```
template<typename real_t >
TridiagonalMatrix< real_t >::TridiagonalMatrix (
    unsigned int n )
```

Construct a new Tridiagonal Matrix.

Parameters

<i>n</i>	Size of main diagonal of the $N \times N$ 2D Tridiagonal Matrix
----------	---

Definition at line 17 of file Tridiagonal_Matrix.cpp.

4.4.2.2 ~TridiagonalMatrix()

```
template<typename real_t >
TridiagonalMatrix< real_t >::~~TridiagonalMatrix
```

Destroy the Tridiagonal Matrix.

Definition at line 28 of file Tridiagonal_Matrix.cpp.

4.4.3 Member Function Documentation

4.4.3.1 getElement()

```
template<typename real_t >
real_t TridiagonalMatrix< real_t >::getElement (
    unsigned int row_index,
    unsigned int column_index )
```

Get the i,j th element of 2D Tridiagonal Matrix.

Parameters

<i>row_index</i>	Row index i
<i>column_index</i>	Column index j

Returns

Value of the i,j th element of a 2D Tridiagonal Matrix

Definition at line 35 of file Tridiagonal_Matrix.cpp.

4.4.3.2 getIndex()

```
template<typename real_t >
unsigned int TridiagonalMatrix< real_t >::getIndex (
```

```
    unsigned int row_index,  
    unsigned int column_index )    [private]
```

Get the index of the i,j th element of Tridiagonal 2D Matrix.

Parameters

<i>row_index</i>	Row index i
<i>column_index</i>	Column index j

Returns

Returns the index in the the flattened array used to represent the 2D Tridiagonal Matrix

Definition at line 108 of file Tridiagonal_Matrix.cpp.

4.4.3.3 indexOfZeroElement()

```
template<typename real_t >
bool TridiagonalMatrix< real_t >::indexOfZeroElement (
    unsigned int row_index,
    unsigned int column_index ) [private]
```

Checks if the row index i and the column index j belong to a zero element of the tridiagonal matrix.

Parameters

<i>row_index</i>	
<i>column_index</i>	

Returns

true if i,j are indices of zero elements in a Tridiagonal matrix

false if i,j are indices of non-zero elements in a Tridiagonal matrix

Definition at line 124 of file Tridiagonal_Matrix.cpp.

4.4.3.4 print()

```
template<typename real_t >
void TridiagonalMatrix< real_t >::print
```

Prints the Triadiagonal Matrix in form of a flattened array.

Definition at line 93 of file Tridiagonal_Matrix.cpp.

4.4.3.5 printMatrix()

```
template<typename real_t >
void TridiagonalMatrix< real_t >::printMatrix
```

Prints the Tridiagonal Matrix in form of a 2D array.

Definition at line 74 of file Tridiagonal_Matrix.cpp.

4.4.3.6 setElement()

```
template<typename real_t >
void TridiagonalMatrix< real_t >::setElement (
    unsigned int row_index,
    unsigned int column_index,
    real_t value )
```

Set the value of the i,j th element of 2D Tridiagonal Matrix.

Parameters

<i>row_index</i>	Row index i
<i>column_index</i>	Column index j
<i>value</i>	Value to be set at the i,j th element

Definition at line 53 of file Tridiagonal_Matrix.cpp.

4.4.4 Member Data Documentation

4.4.4.1 array

```
template<typename real_t >
real_t* TridiagonalMatrix< real_t >::array [private]
```

Flattened array of size $4 * N$ to represent 2D Tridiagonal matrix of size $N \times N$.

Definition at line 93 of file Tridiagonal_Matrix.hpp.

4.4.4.2 N

```
template<typename real_t >
const unsigned int TridiagonalMatrix< real_t >::N [private]
```

Size of main diagonal of the $N \times N$ 2D Tridiagonal Matrix.

Definition at line 100 of file Tridiagonal_Matrix.hpp.

The documentation for this class was generated from the following files:

- combustion-packed-pellet-core-shell-particles/include/Tridiagonal_Matrix.hpp
- combustion-packed-pellet-core-shell-particles/src/Tridiagonal_Matrix.cpp

Chapter 5

File Documentation

5.1 combustion-packed-pellet-core-shell-particles/examples/Lower_Triangular_Matrix_Example.cpp File Reference

Example to test [LowerTriangularMatrix](#) class.

```
#include <iostream>
#include <stdexcept>
#include "Lower_Triangular_Matrix.hpp"
```

Functions

- `int main (int argc, char const *argv[])`

5.1.1 Detailed Description

Example to test [LowerTriangularMatrix](#) class.

Author

Souritra Garai (souritra.garai@iitgn.ac.in)

Version

0.1

Date

2021-06-27

Copyright

Copyright (c) 2021

5.2 combustion-packed-pellet-core-shell-particles/examples/QR_Solver_Example.cpp File Reference

Example cpp file to test out QR_Solver functions.

```
#include "QR_Solver.hpp"
```

Functions

- QR_Solver **my_solver** (5)
- int **main** (int argc, char const *argv[])

Variables

- real_t **A** [3 *5]

5.2.1 Detailed Description

Example cpp file to test out QR_Solver functions.

Author

Souritra Garai (souritra.garai@iitgn.ac.in)

Version

0.1

Date

2021-06-24

Copyright

Copyright (c) 2021

5.3 combustion-packed-pellet-core-shell-particles/include/Givens_RotationMatrix.hpp File Reference

This header serves the definition of an implementation of Givens' Rotation matrix. The rotation matrix is used to solve matrix equations through QR factorization method, particularly tridiagonal matrix equation.

```
#include <Tridiagonal_Matrix.hpp>
#include <Lower_Triangular_Matrix.hpp>
```

Classes

- class [GivensRotationMatrix< real_t >](#)

Class to implement a memory efficient Givens' Rotation Matrix that can vanish the element $a_{k+1,k}$ of a tridiagonal matrix A .

5.3.1 Detailed Description

This header serves the definition of an implementation of Givens' Rotation matrix. The rotation matrix is used to solve matrix equations through QR factorization method, particularly tridiagonal matrix equation.

Author

Souritra Garai (souritra.garai@iitgn.ac.in)

Version

0.1

Date

2021-06-28

Copyright

Copyright (c) 2021

5.4 combustion-packed-pellet-core-shell-particles/include/Lower_Triangular_Matrix.hpp File Reference

This header file defines a class for memory efficient implementation of lower triangular square matrices.

Classes

- class [LowerTriangularMatrix< real_t >](#)

Class to implement a memory efficient model of Lower Triangular Square Matrix.

5.4.1 Detailed Description

This header file defines a class for memory efficient implementation of lower triangular square matrices.

Author

Souritra Garai (souritra.garai@iitgn.ac.in)

Version

0.1

Date

2021-06-25

Copyright

Copyright (c) 2021

5.5 combustion-packed-pellet-core-shell-particles/include/QR_↵ Solver.hpp File Reference

This header file defines a class for solving 2D matrix equations of the form $A.x = b$ (where A is an $n \times n$ matrix and x and b are $n \times 1$ vectors) using QR factorization technique. Also the class is implemented in such a way that it may be parallelized easily using openmp constructs.

```
#include "Tridiagonal_Matrix.hpp"
```

Classes

- class [QRSolver](#)

Class to implement QR factorization algorithm for solving matrix equations of the $A.x = b$ where A is a $n \times n$ tridiagonal matrix and x and b are $n \times 1$ vectors.

5.5.1 Detailed Description

This header file defines a class for solving 2D matrix equations of the form $A.x = b$ (where A is an $n \times n$ matrix and x and b are $n \times 1$ vectors) using QR factorization technique. Also the class is implemented in such a way that it may be parallelized easily using openmp constructs.

Author

Souritra Garai (souritra.garai@iitgn.ac.in)

Version

0.1

Date

2021-06-23

Copyright

Copyright (c) 2021

5.6 combustion-packed-pellet-core-shell-particles/include/Tridiagonal_↵ Matrix.hpp File Reference

This header file serves the definition of an implementation for a Tridiagonal Matrix.

Classes

- class [TridiagonalMatrix< real_t >](#)

Class to implement a memory efficient 2D Tridiagonal square matrix.

5.6.1 Detailed Description

This header file serves the definition of an implementation for a Tridiagonal Matrix.

Author

Souritra Gari (souritra.garai@iitgn.ac.in)

Version

0.1

Date

2021-06-24

Copyright

Copyright (c) 2021

5.7 combustion-packed-pellet-core-shell-particles/src/Lower_Triangular_Matrix.cpp File Reference

```
#include "Lower_Triangular_Matrix.hpp"
#include "stdexcept"
```

5.7.1 Detailed Description

Author

Souritra Garai (souritra.garai@iitgn.ac.in)

Version

0.1

Date

2021-06-25

Copyright

Copyright (c) 2021

5.8 combustion-packed-pellet-core-shell-particles/src/Tridiagonal_↵ Matrix.cpp File Reference

Implementation of Tridiagonal Matrix.

```
#include "Tridiagonal_Matrix.hpp"  
#include <stdexcept>
```

5.8.1 Detailed Description

Implementation of Tridiagonal Matrix.

Author

Souritra Gari (souritra.garai@iitgn.ac.in)

Version

0.1

Date

2021-06-24

Copyright

Copyright (c) 2021

Index

~LowerTriangularMatrix
 LowerTriangularMatrix< real_t >, 10
~QRSolver
 QRSolver, 14
~TridiagonalMatrix
 TridiagonalMatrix< real_t >, 17
array
 TridiagonalMatrix< real_t >, 20
combustion-packed-pellet-core-shell-particles/examples/Lower_Triangular_Matrix_Example.cpp,
 21
combustion-packed-pellet-core-shell-particles/examples/QR_Solver.cpp,
 22
combustion-packed-pellet-core-shell-particles/include/GivensRotationMatrix.h,
 22
combustion-packed-pellet-core-shell-particles/include/Lower_Triangular_Matrix.hpp,
 23
combustion-packed-pellet-core-shell-particles/include/QR_Solver.hpp,
 24
combustion-packed-pellet-core-shell-particles/include/Tridiagonal_Matrix.hpp,
 24
combustion-packed-pellet-core-shell-particles/src/Lower_Triangular_Matrix.cpp,
 25
combustion-packed-pellet-core-shell-particles/src/Tridiagonal_Matrix.cpp,
 26
getElement
 LowerTriangularMatrix< real_t >, 11
 TridiagonalMatrix< real_t >, 17
getIndex
 LowerTriangularMatrix< real_t >, 11
 QRSolver, 15
 TridiagonalMatrix< real_t >, 17
GivensRotationMatrix
 GivensRotationMatrix< real_t >, 8
GivensRotationMatrix< real_t >, 7
 GivensRotationMatrix, 8
 multiply, 8
 setupRotationMatrix, 9
indexOfZeroElement
 LowerTriangularMatrix< real_t >, 12
 TridiagonalMatrix< real_t >, 19
LowerTriangularMatrix
 LowerTriangularMatrix< real_t >, 10
LowerTriangularMatrix< real_t >, 9
 ~LowerTriangularMatrix, 10
 getElement, 11
 getIndex, 11
 indexOfZeroElement, 12
 LowerTriangularMatrix, 10
 printMatrix, 12
 setElement, 12
multiply
 GivensRotationMatrix< real_t >, 8
N
 TridiagonalMatrix< real_t >, 20
print
 TridiagonalMatrix< real_t >, 19
 printMatrix
 LowerTriangularMatrix< real_t >, 12
 TridiagonalMatrix< real_t >, 19
QRSolver, 13
 QRSolver, 14
 getIndex, 15
 QRSolver, 14
 setElement
 LowerTriangularMatrix< real_t >, 12
 TridiagonalMatrix< real_t >, 20
 setupRotationMatrix
 GivensRotationMatrix< real_t >, 9
TridiagonalMatrix
 TridiagonalMatrix< real_t >, 16
TridiagonalMatrix< real_t >, 15
 ~TridiagonalMatrix, 17
array, 20
getElement, 17
getIndex, 17
indexOfZeroElement, 19
N, 20
print, 19
printMatrix, 19
setElement, 20
TridiagonalMatrix, 16