

Super Flow 3000

Generated by Doxygen 1.8.1

Fri Nov 8 2013 14:57:01



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Array< T, size > Class Template Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Constructor & Destructor Documentation . . . . .	6
3.1.2.1	Array . . . . .	6
3.1.2.2	~Array . . . . .	6
3.1.3	Member Function Documentation . . . . .	6
3.1.3.1	operator[] . . . . .	6
3.1.3.2	operator[] . . . . .	6
3.2	Gridfunction Class Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Constructor & Destructor Documentation . . . . .	7
3.2.2.1	Gridfunction . . . . .	7
3.2.2.2	Gridfunction . . . . .	7
3.2.2.3	~Gridfunction . . . . .	7
3.3	IO Class Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Constructor & Destructor Documentation . . . . .	8
3.3.2.1	IO . . . . .	8
3.3.2.2	~IO . . . . .	8
3.3.3	Member Function Documentation . . . . .	8
3.3.3.1	getSimparam . . . . .	8
3.3.3.2	writeVTKFile . . . . .	8
3.4	Simparam Struct Reference . . . . .	8
3.4.1	Detailed Description . . . . .	9
3.4.2	Member Data Documentation . . . . .	9

3.4.2.1	<a href="#">alpha</a>	9
3.4.2.2	<a href="#">deltaVec</a>	9
3.4.2.3	<a href="#">eps</a>	9
3.4.2.4	<a href="#">GX</a>	9
3.4.2.5	<a href="#">GY</a>	9
3.4.2.6	<a href="#">iMax</a>	9
3.4.2.7	<a href="#">iterMax</a>	10
3.4.2.8	<a href="#">jMax</a>	10
3.4.2.9	<a href="#">omg</a>	10
3.4.2.10	<a href="#">PI</a>	10
3.4.2.11	<a href="#">RE</a>	10
3.4.2.12	<a href="#">tau</a>	10
3.4.2.13	<a href="#">tEnd</a>	10
3.4.2.14	<a href="#">UI</a>	10
3.4.2.15	<a href="#">UV</a>	10
3.4.2.16	<a href="#">xLength</a>	10
3.4.2.17	<a href="#">yLength</a>	10
3.5	<a href="#">Solver Class Reference</a>	11
3.5.1	<a href="#">Detailed Description</a>	11
3.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	11
3.5.2.1	<a href="#">Solver</a>	11
3.5.2.2	<a href="#">~Solver</a>	11
3.5.3	<a href="#">Member Function Documentation</a>	11
3.5.3.1	<a href="#">computeResidual</a>	11
3.5.3.2	<a href="#">SORCycle</a>	11
3.6	<a href="#">Stencil Class Reference</a>	12
3.6.1	<a href="#">Detailed Description</a>	12
3.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	12
3.6.2.1	<a href="#">Stencil</a>	12
3.6.3	<a href="#">Member Function Documentation</a>	12
3.6.3.1	<a href="#">ApplyStencilOperator</a>	12
3.6.3.2	<a href="#">setFxxStencil</a>	12
3.6.4	<a href="#">Member Data Documentation</a>	12
3.6.4.1	<a href="#">h</a>	12
3.6.4.2	<a href="#">stencil</a>	13
3.6.4.3	<a href="#">stencilwidth</a>	13
<b>4</b>	<b><a href="#">File Documentation</a></b>	<b>15</b>
4.1	<a href="#">Debug/Grid/Gridfunction.d File Reference</a>	15
4.2	<a href="#">Release/Grid/Gridfunction.d File Reference</a>	15

4.3	Debug/IO/IO.d File Reference . . . . .	15
4.4	Release/IO/IO.d File Reference . . . . .	15
4.5	Debug/main.d File Reference . . . . .	15
4.6	Release/main.d File Reference . . . . .	15
4.7	Debug/Stencil/stencil.d File Reference . . . . .	15
4.8	Release/Stencil/stencil.d File Reference . . . . .	15
4.9	Grid/Gridfunction.cpp File Reference . . . . .	15
4.10	Grid/Gridfunction.h File Reference . . . . .	15
4.11	IO/IO.cpp File Reference . . . . .	16
4.11.1	Macro Definition Documentation . . . . .	16
4.11.1.1	Element . . . . .	16
4.12	IO/IO.hpp File Reference . . . . .	16
4.13	main.cpp File Reference . . . . .	16
4.13.1	Function Documentation . . . . .	16
4.13.1.1	main . . . . .	16
4.14	Misc/template.h File Reference . . . . .	17
4.15	Misc/typedef.h File Reference . . . . .	17
4.15.1	Typedef Documentation . . . . .	17
4.15.1.1	GridFunctionType . . . . .	17
4.15.1.2	IndexType . . . . .	17
4.15.1.3	MultIndexType . . . . .	17
4.15.1.4	PointType . . . . .	17
4.15.1.5	RealType . . . . .	18
4.15.1.6	StencilType . . . . .	18
4.16	Solver/solver.h File Reference . . . . .	18
4.17	Stencil/stencil.cpp File Reference . . . . .	18
4.17.1	Function Documentation . . . . .	18
4.17.1.1	ApplyStencilOperator . . . . .	18
4.18	Stencil/stencil.h File Reference . . . . .	19
4.19	Structs/simparam.h File Reference . . . . .	19



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Array&lt; T, size &gt;</a>	
Typedefs for the application . . . . .	5
<a href="#">Gridfunction</a> . . . . .	6
<a href="#">IO</a>	
The class implements the <a href="#">IO</a> . . . . .	7
<a href="#">Simparam</a> . . . . .	8
<a href="#">Solver</a> . . . . .	11
<a href="#">Stencil</a> . . . . .	12





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.cpp</a>	16
Debug/ <a href="#">main.d</a>	15
Debug/Grid/ <a href="#">Gridfunction.d</a>	15
Debug/IO/ <a href="#">IO.d</a>	15
Debug/Stencil/ <a href="#">stencil.d</a>	15
Grid/ <a href="#">Gridfunction.cpp</a>	15
Grid/ <a href="#">Gridfunction.h</a>	15
IO/ <a href="#">IO.cpp</a>	16
IO/ <a href="#">IO.hpp</a>	16
Misc/ <a href="#">template.h</a>	17
Misc/ <a href="#">typedef.h</a>	17
Release/ <a href="#">main.d</a>	15
Release/Grid/ <a href="#">Gridfunction.d</a>	15
Release/IO/ <a href="#">IO.d</a>	15
Release/Stencil/ <a href="#">stencil.d</a>	15
Solver/ <a href="#">solver.h</a>	18
Stencil/ <a href="#">stencil.cpp</a>	18
Stencil/ <a href="#">stencil.h</a>	19
Structs/ <a href="#">simpparam.h</a>	19



## Chapter 3

# Class Documentation

### 3.1 `Array< T, size >` Class Template Reference

Typedefs for the application.

```
#include <template.h>
```

#### Public Member Functions

- `Array` (void)  
*Constructor.*
- `~Array` (void)  
*Destructor.*
- `T & operator[]` (size\_t index)  
*Operator returns the element with the index index.*
- `const T & operator[]` (size\_t index) const  
*Operator returns the element with the index index.*

#### 3.1.1 Detailed Description

```
template<class T, size_t size>class Array< T, size >
```

Typedefs for the application.

#### Author

diehlpk

#### Date

2012 Template for a basic array data type.

#### Parameters

<i>T</i>	The datatype or class of the elemtens in the array.
<i>size</i>	The size of the array.

Definition at line 18 of file template.h.

### 3.1.2 Constructor & Destructor Documentation

3.1.2.1 `template<class T, size_t size> Array< T, size >::Array ( void ) [inline]`

Constructor.

Definition at line 26 of file `template.h`.

3.1.2.2 `template<class T, size_t size> Array< T, size >::~~Array ( void ) [inline]`

Destructor.

Definition at line 33 of file `template.h`.

### 3.1.3 Member Function Documentation

3.1.3.1 `template<class T, size_t size> T& Array< T, size >::operator[] ( size_t index ) [inline]`

Operator returns the element with the index `index`.

Parameters

<i>index</i>	The index of the element.
--------------	---------------------------

Definition at line 41 of file `template.h`.

3.1.3.2 `template<class T, size_t size> const T& Array< T, size >::operator[] ( size_t index ) const [inline]`

Operator returns the element with the index `index`.

Parameters

<i>index</i>	The index of the element.
--------------	---------------------------

Definition at line 51 of file `template.h`.

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

- [Misc/template.h](#)

## 3.2 Gridfunction Class Reference

```
#include <Gridfunction.h>
```

### Public Member Functions

- [Gridfunction](#) (int DimX, int DimY)
- [Gridfunction](#) (const [MultiIndexType](#) griddimension)
- [~Gridfunction](#) ()

*Destructor.*

### 3.2.1 Detailed Description

Definition at line 13 of file `Gridfunction.h`.

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 Gridfunction::Gridfunction ( int *DimX*, int *DimY* )

Konstruktor

Parameters

<i>DimX</i>	Dimension in X-Direction
<i>DimY</i>	Dimension in Y-Direction

Definition at line 11 of file Gridfunction.cpp.

#### 3.2.2.2 Gridfunction::Gridfunction ( const MultiIndexType *griddimension* )

Definition at line 20 of file Gridfunction.cpp.

#### 3.2.2.3 Gridfunction::~~Gridfunction ( )

Destruktor.

Definition at line 25 of file Gridfunction.cpp.

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

- [Grid/Gridfunction.h](#)
- [Grid/Gridfunction.cpp](#)

## 3.3 IO Class Reference

The class implements the [IO](#).

```
#include <IO.hpp>
```

### Public Member Functions

- [IO](#) (char \*input, char \*output)
- [~IO](#) ()  
*Destructor.*
- void [writeVTKFile](#) (const [MultiIndexType](#) &griddimension, [GridFunctionType](#) &u, [GridFunctionType](#) &v, [GridFunctionType](#) &p, const [PointType](#) &delta, int step)
- [Simparam](#) [getSimparam](#) ()  
*Method that returns private member variable simparam.*

#### 3.3.1 Detailed Description

The class implements the [IO](#).

Author

diehlpk

Date

2012

Definition at line 17 of file IO.hpp.

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 `IO::IO ( char * input, char * output )`

Konstruktor

Parameters

<i>input</i>	Path to the file with the simulation parameters.
<i>output</i>	Path to the directory for the vtk files.

Definition at line 7 of file IO.cpp.

#### 3.3.2.2 `IO::~~IO ( )`

Destructor.

Definition at line 15 of file IO.cpp.

### 3.3.3 Member Function Documentation

#### 3.3.3.1 `Simparam IO::getSimparam ( ) [inline]`

Method that returns private member variable simparam.

Definition at line 46 of file IO.hpp.

#### 3.3.3.2 `void IO::writeVTKFile ( const MultiIndexType & griddimension, GridFunctionType & u, GridFunctionType & v, GridFunctionType & p, const PointType & delta, int step )`

Method writes the GridFunctions u,v,p in the vtk data format to the hard disk. The files are named in the following convention: field\_(step).vts.

Parameters

<i>griddimension</i>	The dimension of the gridfunctions.
<i>u</i>	The gridfunction u.
<i>v</i>	The gridfunction v.
<i>p</i>	The gridfunction p.
<i>delta</i>	The mesh width in x direction and y direction
<i>step</i>	The number of the timestep.

Definition at line 187 of file IO.cpp.

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

- [IO/IO.hpp](#)
- [IO/IO.cpp](#)

## 3.4 Simparam Struct Reference

```
#include <simparam.h>
```

## Public Attributes

- [RealType xLength](#)
- [RealType yLength](#)
- [int iMax](#)
- [int jMax](#)
- [int tEnd](#)
- [RealType tau](#)
- [int deltaVec](#)
- [int iterMax](#)
- [RealType eps](#)
- [RealType omg](#)
- [RealType alpha](#)
- [int RE](#)
- [int GX](#)
- [int GY](#)
- [int UI](#)
- [int UV](#)
- [int PI](#)

### 3.4.1 Detailed Description

Definition at line 12 of file `simparam.h`.

### 3.4.2 Member Data Documentation

#### 3.4.2.1 [RealType Simparam::alpha](#)

Definition at line 23 of file `simparam.h`.

#### 3.4.2.2 [int Simparam::deltaVec](#)

Definition at line 19 of file `simparam.h`.

#### 3.4.2.3 [RealType Simparam::eps](#)

Definition at line 21 of file `simparam.h`.

#### 3.4.2.4 [int Simparam::GX](#)

Definition at line 25 of file `simparam.h`.

#### 3.4.2.5 [int Simparam::GY](#)

Definition at line 26 of file `simparam.h`.

#### 3.4.2.6 [int Simparam::iMax](#)

Definition at line 15 of file `simparam.h`.

#### 3.4.2.7 int Simparam::iterMax

Definition at line 20 of file `simparam.h`.

#### 3.4.2.8 int Simparam::jMax

Definition at line 16 of file `simparam.h`.

#### 3.4.2.9 RealType Simparam::omg

Definition at line 22 of file `simparam.h`.

#### 3.4.2.10 int Simparam::PI

Definition at line 29 of file `simparam.h`.

#### 3.4.2.11 int Simparam::RE

Definition at line 24 of file `simparam.h`.

#### 3.4.2.12 RealType Simparam::tau

Definition at line 18 of file `simparam.h`.

#### 3.4.2.13 int Simparam::tEnd

Definition at line 17 of file `simparam.h`.

#### 3.4.2.14 int Simparam::UI

Definition at line 27 of file `simparam.h`.

#### 3.4.2.15 int Simparam::UV

Definition at line 28 of file `simparam.h`.

#### 3.4.2.16 RealType Simparam::xLength

Definition at line 13 of file `simparam.h`.

#### 3.4.2.17 RealType Simparam::yLength

Definition at line 14 of file `simparam.h`.

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

- Structs/[simparam.h](#)



## 3.5 Solver Class Reference

```
#include <solver.h>
```

### Public Member Functions

- [RealType](#) **computeResidual** ([GridFunctionType](#) &sourcegridfunction, [GridFunctionType](#) &rhs, const [PointType](#) &h)  
*Function to compute the global residual.*
- void **SORCycle** ([Gridfunction](#) \*gridfunction, [GridFunctionType](#) &rhs, const [PointType](#) &delta, [RealType](#) omega)  
*Function to compute the global residual.*
- [Solver](#) (void)  
*Constructor (does not have to do anything)*
- [~Solver](#) ()  
*Destructor.*

### 3.5.1 Detailed Description

Definition at line 15 of file solver.h.

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 Solver::Solver ( void )

Constructor (does not have to do anything)

#### 3.5.2.2 Solver::~~Solver ( )

Destructor.

### 3.5.3 Member Function Documentation

#### 3.5.3.1 RealType Solver::computeResidual ( GridFunctionType & sourcegridfunction, GridFunctionType & rhs, const PointType & h )

Function to compute the global residual.

##### Parameters

<i>sourcegridfunction</i>	?The discretized solution
<i>rhs</i>	The right hand side of the discretized local PDE
<i>h</i>	?what are these two RealTypes for?

#### 3.5.3.2 void Solver::SORCycle ( Gridfunction \* gridfunction, GridFunctionType & rhs, const PointType & delta, RealType omega )

Function to compute the global residual.

## Parameters

<i>gridfunction</i>	Pointer on the discretized solution
<i>rhs</i>	The right hand side of the discretized local PDE
<i>delta</i>	\$\$ are the gridwidths in x- and y-direction
<i>omega</i>	The \$\$ -parameter of the SOR-cycle

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

- Solver/[solver.h](#)

## 3.6 Stencil Class Reference

```
#include <stencil.h>
```

### Public Member Functions

- [Stencil](#) (int *stencilwidth\_input*, [PointType](#) &*h\_input*)
- void [ApplyStencilOperator](#) (const [MultiIndexType](#) &*gridreadbegin*, const [MultiIndexType](#) &*gridreadend*, const [Gridfunction](#) *sourcegridfunction*, [Gridfunction](#) *imagegridfunction*)
- void [setFxxStencil](#) ()

### Public Attributes

- [StencilType](#) *stencil*
- int *stencilwidth*
- [PointType](#) & *h*

#### 3.6.1 Detailed Description

Definition at line 15 of file `stencil.h`.

#### 3.6.2 Constructor & Destructor Documentation

##### 3.6.2.1 `Stencil::Stencil ( int stencilwidth_input, PointType & h_input )`

Definition at line 11 of file `stencil.cpp`.

#### 3.6.3 Member Function Documentation

##### 3.6.3.1 `void Stencil::ApplyStencilOperator ( const MultiIndexType & gridreadbegin, const MultiIndexType & gridreadend, const Gridfunction sourcegridfunction, Gridfunction imagegridfunction )`

##### 3.6.3.2 `void Stencil::setFxxStencil ( )`

Definition at line 25 of file `stencil.cpp`.

#### 3.6.4 Member Data Documentation

##### 3.6.4.1 `PointType& Stencil::h`

Definition at line 23 of file `stencil.h`.

#### 3.6.4.2 StencilType Stencil::stencil

Definition at line 21 of file stencil.h.

#### 3.6.4.3 int Stencil::stencilwidth

Definition at line 22 of file stencil.h.

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

- Stencil/[stencil.h](#)
- Stencil/[stencil.cpp](#)



## Chapter 4

# File Documentation

### 4.1 Debug/Grid/Gridfunction.d File Reference

### 4.2 Release/Grid/Gridfunction.d File Reference

### 4.3 Debug/IO/IO.d File Reference

### 4.4 Release/IO/IO.d File Reference

### 4.5 Debug/main.d File Reference

### 4.6 Release/main.d File Reference

### 4.7 Debug/Stencil/stencil.d File Reference

### 4.8 Release/Stencil/stencil.d File Reference

### 4.9 Grid/Gridfunction.cpp File Reference

```
#include "Gridfunction.h"
```

### 4.10 Grid/Gridfunction.h File Reference

```
#include "../Misc/typedef.h"
```

## Classes

- class [Gridfunction](#)

## 4.11 IO/IO.cpp File Reference

```
#include "IO.hpp"
#include <iostream>
#include <stdio.h>
```

### Macros

- `#define Element(field, ic) ((field)[(ic)[0]][(ic)[1]])`

### 4.11.1 Macro Definition Documentation

4.11.1.1 `#define Element( field, ic ) ((field)[(ic)[0]][(ic)[1]])`

Definition at line 82 of file IO.cpp.

## 4.12 IO/IO.hpp File Reference

```
#include "../Misc/typedef.h"
#include <iostream>
#include <fstream>
#include "../Structs/simparam.h"
```

### Classes

- class [IO](#)

*The class implements the [IO](#).*

## 4.13 main.cpp File Reference

```
#include <iostream>
#include "Misc/template.h"
#include "Misc/typedef.h"
#include "IO/IO.hpp"
#include "Stencil/stencil.h"
#include "Grid/Gridfunction.h"
```

### Functions

- int [main](#) ()

### 4.13.1 Function Documentation

4.13.1.1 `int main ( )`

Definition at line 14 of file main.cpp.

## 4.14 Misc/template.h File Reference

```
#include <cstdlib>
#include <cassert>
```

### Classes

- class [Array< T, size >](#)  
*Typedefs for the application.*

## 4.15 Misc/typedef.h File Reference

```
#include "template.h"
```

### Typedefs

- typedef double [RealType](#)  
*Typedefs for the application.*
- typedef int [IndexType](#)
- typedef [Array< IndexType, 2 >](#) [MultiIndexType](#)
- typedef [RealType](#) \*\* [GridFunctionType](#)
- typedef [GridFunctionType](#) [StencilType](#)
- typedef [Array< RealType, 2 >](#) [PointType](#)

### 4.15.1 Typedef Documentation

#### 4.15.1.1 typedef [RealType](#)\*\* [GridFunctionType](#)

Creates a type name for [GridFunctionType](#)

Definition at line 22 of file [typedef.h](#).

#### 4.15.1.2 typedef int [IndexType](#)

Creates a type name for [IndexType](#)

Definition at line 16 of file [typedef.h](#).

#### 4.15.1.3 typedef [Array< IndexType, 2 >](#) [MultiIndexType](#)

Creates a type name for [MultiIndexType](#)

Definition at line 19 of file [typedef.h](#).

#### 4.15.1.4 typedef [Array< RealType, 2 >](#) [PointType](#)

Creates a type name for [PointType](#)

Definition at line 28 of file [typedef.h](#).

#### 4.15.1.5 typedef double RealType

Typedefs for the application.

##### Author

diehlpk

##### Date

2012

Creates a type name for RealType

Definition at line 13 of file typedef.h.

#### 4.15.1.6 typedef GridFunctionType StencilType

Creates a type name for StencilType

Definition at line 25 of file typedef.h.

### 4.16 Solver/solver.h File Reference

```
#include "../Misc/typedef.h"
#include "../Grid/gridfunction.h"
```

#### Classes

- class [Solver](#)

### 4.17 Stencil/stencil.cpp File Reference

```
#include "stencil.h"
#include <iostream>
```

#### Functions

- void [ApplyStencilOperator](#) (const [MultiIndexType](#) &gridreadbegin, const [MultiIndexType](#) &gridreadend, const [MultiIndexType](#) &gridwritebegin, const [MultiIndexType](#) &gridwriteend, const [Gridfunction](#) sourcegridfunction, [Gridfunction](#) imagegridfunction)

#### 4.17.1 Function Documentation

4.17.1.1 void [ApplyStencilOperator](#) ( const [MultiIndexType](#) & *gridreadbegin*, const [MultiIndexType](#) & *gridreadend*, const [MultiIndexType](#) & *gridwritebegin*, const [MultiIndexType](#) & *gridwriteend*, const [Gridfunction](#) *sourcegridfunction*, [Gridfunction](#) *imagegridfunction* )

Definition at line 15 of file stencil.cpp.



## 4.18 Stencil/stencil.h File Reference

```
#include "../Misc/typedef.h"  
#include "../Grid/Gridfunction.h"
```

### Classes

- class [Stencil](#)

## 4.19 Structs/simparam.h File Reference

```
#include "../Misc/template.h"
```

### Classes

- struct [Simparam](#)

# Index

- ~Array
  - Array, [6](#)
- ~Gridfunction
  - Gridfunction, [7](#)
- ~IO
  - IO, [8](#)
- ~Solver
  - Solver, [11](#)
- alpha
  - Simparam, [9](#)
- ApplyStencilOperator
  - Stencil, [12](#)
  - stencil.cpp, [18](#)
- Array
  - ~Array, [6](#)
  - Array, [6](#)
- Array < T, size >, [5](#)
- computeResidual
  - Solver, [11](#)
- Debug/Grid/Gridfunction.d, [15](#)
- Debug/IO/IO.d, [15](#)
- Debug/Stencil/stencil.d, [15](#)
- Debug/main.d, [15](#)
- deltaVec
  - Simparam, [9](#)
- Element
  - IO.cpp, [16](#)
- eps
  - Simparam, [9](#)
- GX
  - Simparam, [9](#)
- GY
  - Simparam, [9](#)
- getSimparam
  - IO, [8](#)
- Grid/Gridfunction.cpp, [15](#)
- Grid/Gridfunction.h, [15](#)
- GridFunctionType
  - typedef.h, [17](#)
- Gridfunction, [6](#)
  - ~Gridfunction, [7](#)
  - Gridfunction, [7](#)
- h
  - Stencil, [12](#)
- iMax
  - Simparam, [9](#)
- IO, [7](#)
  - ~IO, [8](#)
  - getSimparam, [8](#)
  - IO, [8](#)
  - IO, [8](#)
  - writeVTKFile, [8](#)
- IO.cpp
  - Element, [16](#)
- IO/IO.cpp, [16](#)
- IO/IO.hpp, [16](#)
- IndexType
  - typedef.h, [17](#)
- iterMax
  - Simparam, [9](#)
- jMax
  - Simparam, [10](#)
- main
  - main.cpp, [16](#)
- main.cpp, [16](#)
  - main, [16](#)
- Misc/template.h, [17](#)
- Misc/typedef.h, [17](#)
- MultiIndexType
  - typedef.h, [17](#)
- omg
  - Simparam, [10](#)
- PI
  - Simparam, [10](#)
- PointType
  - typedef.h, [17](#)
- RE
  - Simparam, [10](#)
- RealType
  - typedef.h, [17](#)
- Release/Grid/Gridfunction.d, [15](#)
- Release/IO/IO.d, [15](#)
- Release/Stencil/stencil.d, [15](#)
- Release/main.d, [15](#)
- SORCycle
  - Solver, [11](#)
- setFxxStencil
  - Stencil, [12](#)
- Simparam, [8](#)

- alpha, [9](#)
- deltaVec, [9](#)
- eps, [9](#)
- GX, [9](#)
- GY, [9](#)
- iMax, [9](#)
- iterMax, [9](#)
- jMax, [10](#)
- omg, [10](#)
- PI, [10](#)
- RE, [10](#)
- tEnd, [10](#)
- tau, [10](#)
- UI, [10](#)
- UV, [10](#)
- xLength, [10](#)
- yLength, [10](#)
- Solver, [11](#)
  - ~Solver, [11](#)
  - computeResidual, [11](#)
  - SORCycle, [11](#)
  - Solver, [11](#)
- Solver/solver.h, [18](#)
- Stencil, [12](#)
  - ApplyStencilOperator, [12](#)
  - h, [12](#)
  - setFxxStencil, [12](#)
  - Stencil, [12](#)
  - stencil, [12](#)
  - stencilwidth, [13](#)
- stencil
  - Stencil, [12](#)
- stencil.cpp
  - ApplyStencilOperator, [18](#)
- Stencil/stencil.cpp, [18](#)
- Stencil/stencil.h, [19](#)
- StencilType
  - typedef.h, [18](#)
- stencilwidth
  - Stencil, [13](#)
- Structs/simparam.h, [19](#)
- tEnd
  - Simparam, [10](#)
- tau
  - Simparam, [10](#)
- typedef.h
  - GridFunctionType, [17](#)
  - IndexType, [17](#)
  - MultiIndexType, [17](#)
  - PointType, [17](#)
  - RealType, [17](#)
  - StencilType, [18](#)
- UI
  - Simparam, [10](#)
- UV
  - Simparam, [10](#)
- writeVTKFile
  - IO, [8](#)
- xLength
  - Simparam, [10](#)
- yLength
  - Simparam, [10](#)