

## 228A Computing Homework

Generated by Doxygen 1.8.6

Tue Mar 31 2015 17:27:03



# Contents

<b>1</b>	<b>MAT 228A Computing Homework</b>	<b>1</b>
<b>2</b>	<b>Bug List</b>	<b>3</b>
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	Finite_Grid Class Reference . . . . .	9
5.1.1	Member Function Documentation . . . . .	9
5.1.1.1	computation . . . . .	9
5.1.1.2	get_a . . . . .	9
5.1.1.3	get_CFL . . . . .	10
5.1.1.4	get_initial_data . . . . .	10
5.1.1.5	get_local_M . . . . .	10
5.1.1.6	get_local_solution_data . . . . .	10
5.1.1.7	get_M . . . . .	10
5.1.1.8	get_N . . . . .	10
5.1.1.9	get_solution_data . . . . .	10
5.1.1.10	get_T . . . . .	10
5.1.1.11	initial_Data . . . . .	10
5.1.1.12	initialize_Grid . . . . .	10
5.1.1.13	operator() . . . . .	10
5.1.1.14	operator= . . . . .	10
5.1.2	Member Data Documentation . . . . .	10
5.1.2.1	accessed_last . . . . .	10
5.2	Method::initial_condition Struct Reference . . . . .	10
5.2.1	Member Data Documentation . . . . .	11
5.2.1.1	function . . . . .	11
5.2.1.2	name . . . . .	11

5.3	Method Class Reference	11
5.3.1	Member Typedef Documentation	12
5.3.1.1	initial_condition_function	12
5.3.1.2	method_function	12
5.3.2	Member Function Documentation	12
5.3.2.1	FCT	12
5.3.2.2	Fromm	12
5.3.2.3	FvL	12
5.3.2.4	Gaussian_Pulse	12
5.3.2.5	LF	12
5.3.2.6	LW	12
5.3.2.7	LW4	12
5.3.2.8	Semicircle	12
5.3.2.9	Square_Wave	12
5.3.2.10	Upwind	12
5.3.3	Member Data Documentation	12
5.3.3.1	Finite_Grid	12
5.3.3.2	initial_conditions	12
5.3.3.3	methods	13
5.4	Method::method Struct Reference	13
5.4.1	Member Data Documentation	13
5.4.1.1	function	13
5.4.1.2	name	13
5.5	Output Class Reference	13
5.5.1	Member Typedef Documentation	14
5.5.1.1	output_container	14
5.5.2	Member Function Documentation	14
5.5.2.1	index_output_table	14
5.5.2.2	plot	14
5.5.2.3	plot_one	14
5.5.2.4	print_output_table	14
5.5.2.5	print_output_table_latex	14
5.5.2.6	process_output	14
5.6	Output::output_data Struct Reference	14
5.6.1	Member Data Documentation	15
5.6.1.1	convergence_rate_1_norm	15
5.6.1.2	convergence_rate_2_norm	15
5.6.1.3	convergence_rate_sup_norm	15
5.6.1.4	error_1_norm	15
5.6.1.5	error_2_norm	15

---

5.6.1.6	<a href="#">error_sup_norm</a>	15
5.6.1.7	<a href="#">k</a>	15
<b>6</b>	<b>File Documentation</b>	<b>17</b>
6.1	<a href="#">include/Finite_Grid.hpp File Reference</a>	17
6.1.1	<a href="#">Detailed Description</a>	17
6.2	<a href="#">include/Method.hpp File Reference</a>	17
6.3	<a href="#">include/Output.hpp File Reference</a>	18
6.4	<a href="#">main.cpp File Reference</a>	18
6.4.1	<a href="#">Function Documentation</a>	18
6.4.1.1	<a href="#">computation</a>	18
6.4.1.2	<a href="#">main</a>	18
6.5	<a href="#">source/Finite_Grid.cpp File Reference</a>	19
6.6	<a href="#">source/Method.cpp File Reference</a>	19
6.7	<a href="#">source/Output.cpp File Reference</a>	19



# Chapter 1

## MAT 228A Computing Homework

Description of the project; how to install and run.

Computation: C++

[Output](#) data storage: hdf5

Processing: Python + h5py + matplotlib

**Analysis of the computational results:**

*Theory and Computing Homework*

**Author:**

*Ivan Cherkashin*





## Chapter 2

# Bug List

File [Finite\\_Grid.hpp](#)

No known bugs



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Finite_Grid</a>	9
<a href="#">Method::initial_condition</a>	10
<a href="#">Method</a>	11
<a href="#">Method::method</a>	13
<a href="#">Output</a>	13
<a href="#">Output::output_data</a>	14



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.cpp</a>	18
<a href="#">include/Finite_Grid.hpp</a>	
One-dimensional uniform grid with periodic boundary conditions	17
<a href="#">include/Method.hpp</a>	17
<a href="#">include/Output.hpp</a>	18
<a href="#">source/Finite_Grid.cpp</a>	19
<a href="#">source/Method.cpp</a>	19
<a href="#">source/Output.cpp</a>	19



## Chapter 5

# Class Documentation

### 5.1 Finite\_Grid Class Reference

```
#include <Finite_Grid.hpp>
```

#### Public Member Functions

- void [initialize\\_Grid](#) (int m, double A, double s, double t)
- double & [operator\(\)](#) (int i)
- [Finite\\_Grid](#) & [operator=](#) (const [Finite\\_Grid](#) &original)

*Copy a grid. Description.*

- void [initial\\_Data](#) (double(\*initial\_data)(double))
- double \* [get\\_solution\\_data](#) ()
- double \* [get\\_local\\_solution\\_data](#) ()
- double \* [get\\_initial\\_data](#) ()
- int [get\\_M](#) () const
- int [get\\_local\\_M](#) () const
- int [get\\_N](#) () const
- double [get\\_CFL](#) () const
- double [get\\_a](#) () const
- double [get\\_T](#) () const

#### Static Public Member Functions

- static void [computation](#) (double(\*initial\_data)(double), double(\*method)(int, [Finite\\_Grid](#) &), [Finite\\_Grid](#) &u)

#### Public Attributes

- bool [accessed\\_last](#)

#### 5.1.1 Member Function Documentation

5.1.1.1 void [Finite\\_Grid::computation](#) ( double(\*) (double) *initial\_data*, double(\*) (int, [Finite\\_Grid](#) &) *method*, [Finite\\_Grid](#) &  
*u* ) [static]

5.1.1.2 double [Finite\\_Grid::get\\_a](#) ( ) const

- 5.1.1.3 `double Finite_Grid::get_CFL ( ) const`
- 5.1.1.4 `double * Finite_Grid::get_initial_data ( )`
- 5.1.1.5 `int Finite_Grid::get_local_M ( ) const`
- 5.1.1.6 `double * Finite_Grid::get_local_solution_data ( )`
- 5.1.1.7 `int Finite_Grid::get_M ( ) const`
- 5.1.1.8 `int Finite_Grid::get_N ( ) const`
- 5.1.1.9 `double * Finite_Grid::get_solution_data ( )`
- 5.1.1.10 `double Finite_Grid::get_T ( ) const`
- 5.1.1.11 `void Finite_Grid::initial_Data ( double(*) (double) initial_data )`
- 5.1.1.12 `void Finite_Grid::initialize_Grid ( int m, double A, double s, double t )`
- 5.1.1.13 `double & Finite_Grid::operator() ( int i )`
- 5.1.1.14 `Finite_Grid & Finite_Grid::operator= ( const Finite_Grid & original )`

Copy a grid. Description.

Parameters

<i>original</i>	grid to be copied
-----------------	-------------------

Returns

Reference to the new copy.

## 5.1.2 Member Data Documentation

- 5.1.2.1 `bool Finite_Grid::accessed_last`

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

- [include/Finite\\_Grid.hpp](#)
- [source/Finite\\_Grid.cpp](#)

## 5.2 Method::initial\_condition Struct Reference

```
#include <Method.hpp>
```

Public Attributes

- `const char * name`
- `const Method::initial\_condition\_function function`



### 5.2.1 Member Data Documentation

5.2.1.1 `const Method::initial_condition_function Method::initial_condition::function`

5.2.1.2 `const char* Method::initial_condition::name`

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

- `include/Method.hpp`

## 5.3 Method Class Reference

```
#include <Method.hpp>
```

### Classes

- struct `initial_condition`
- struct `method`

### Public Types

- `typedef double(* initial_condition_function)(double)`
- `typedef double(* method_function)(int, Finite_Grid &)`

### Static Public Member Functions

- static double `Square_Wave` (double x)
- static double `Semicircle` (double x)
- static double `Gaussian_Pulse` (double x)
- static double `Upwind` (int m, `Finite_Grid` &u)
- static double `LF` (int m, `Finite_Grid` &u)
- static double `LW` (int m, `Finite_Grid` &u)
- static double `Fromm` (int m, `Finite_Grid` &u)
- static double `FvL` (int m, `Finite_Grid` &u)
- static double `FCT` (int m, `Finite_Grid` &u)
- static double `LW4` (int m, `Finite_Grid` &u)

### Public Attributes

- friend `Finite_Grid`

### Static Public Attributes

- static const `Method::method methods` [7]
- static const `Method::initial_condition initial_conditions` [3]

### 5.3.1 Member Typedef Documentation

5.3.1.1 `typedef double(* Method::initial_condition_function)(double)`

5.3.1.2 `typedef double(* Method::method_function)(int, Finite_Grid &)`

### 5.3.2 Member Function Documentation

5.3.2.1 `double Method::FCT ( int m, Finite_Grid & u ) [static]`

Implementation of Flux-Corrected Transport method

Parameters

<i>m</i>	index of a cell
<i>u</i>	finite grid object on which the computation is performed

Returns

value of the field in the cell for the next time step

5.3.2.2 `double Method::Fromm ( int m, Finite_Grid & u ) [static]`

5.3.2.3 `double Method::FvL ( int m, Finite_Grid & u ) [static]`

5.3.2.4 `double Method::Gaussian_Pulse ( double x ) [static]`

5.3.2.5 `double Method::LF ( int m, Finite_Grid & u ) [static]`

5.3.2.6 `double Method::LW ( int m, Finite_Grid & u ) [static]`

5.3.2.7 `double Method::LW4 ( int m, Finite_Grid & u ) [static]`

5.3.2.8 `double Method::Semicircle ( double x ) [static]`

5.3.2.9 `double Method::Square_Wave ( double x ) [static]`

5.3.2.10 `double Method::Upwind ( int m, Finite_Grid & u ) [static]`

### 5.3.3 Member Data Documentation

5.3.3.1 `friend Method::Finite_Grid`

5.3.3.2 `const Method::initial_condition Method::initial_conditions [static]`

Initial value:

```
= {
{"Square_Wave", &Method::Square_Wave},
{"Semicircle", &Method::Semicircle},
{"Gaussian_Pulse", &Method::Gaussian_Pulse},
}
```

### 5.3.3.3 `const Method::method Method::methods` `[static]`

#### Initial value:

```
= {
    {"Upwind", &Method::Upwind},
    {"LF", &Method::LF},
    {"LW", &Method::LW},
    {"Fromm", &Method::Fromm},
    {"FvL", &Method::FvL},
    {"FCT", &Method::FCT},
    {"LW4", &Method::LW4}
}
```

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

- [include/Method.hpp](#)
- [source/Method.cpp](#)

## 5.4 Method::method Struct Reference

```
#include <Method.hpp>
```

### Public Attributes

- `const char *` [name](#)
- `const` [Method::method\\_function](#) `function`

### 5.4.1 Member Data Documentation

#### 5.4.1.1 `const Method::method_function Method::method::function`

#### 5.4.1.2 `const char* Method::method::name`

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

- [include/Method.hpp](#)

## 5.5 Output Class Reference

```
#include <Output.hpp>
```

### Classes

- `struct` [output\\_data](#)

### Public Types

- `typedef std::map< std::pair  
    < const char *, const char * >  
    , Output::output\_data > output\_container`

## Static Public Member Functions

- static void [plot](#) (const char \*filename, const char \*initial\_condition, const char \*method, std::vector< [Finite\\_Grid](#) > &u)
- static void [plot\\_one](#) (const char \*filename, const char \*initial\_condition, const char \*method, [Finite\\_Grid](#) &u)
- static void [index\\_output\\_table](#) (std::vector< [output\\_container](#) > &output\_table, [Method::initial\\_condition](#) initial\_condition, [Method::method](#) method, std::vector< [Finite\\_Grid](#) > &grid)
- static void [print\\_output\\_table](#) (std::ofstream &output\_stream, std::vector< [output\\_container](#) > &output\_table, double s, [Method::initial\\_condition](#) initial\_condition, [Method::method](#) method)
- static void [print\\_output\\_table\\_latex](#) (std::ofstream &output\_stream, std::vector< [output\\_container](#) > &output\_table, double s, [Method::initial\\_condition](#) initial\_condition, [Method::method](#) method)
- static void [process\\_output](#) (std::vector< [output\\_container](#) > &output\_table, [Method::initial\\_condition](#) initial\_condition, [Method::method](#) method)

### 5.5.1 Member Typedef Documentation

5.5.1.1 typedef std::map<std::pair<const char \*, const char \*>, [Output::output\\_data](#)> [Output::output\\_container](#)

### 5.5.2 Member Function Documentation

- 5.5.2.1 void [Output::index\\_output\\_table](#) ( std::vector< [output\\_container](#) > & *output\_table*, [Method::initial\\_condition](#) *initial\_condition*, [Method::method](#) *method*, std::vector< [Finite\\_Grid](#) > & *grid* ) [static]
- 5.5.2.2 void [Output::plot](#) ( const char \* *filename*, const char \* *initial\_condition*, const char \* *method*, std::vector< [Finite\\_Grid](#) > & *u* ) [static]
- 5.5.2.3 void [Output::plot\\_one](#) ( const char \* *filename*, const char \* *initial\_condition*, const char \* *method*, [Finite\\_Grid](#) & *u* ) [static]
- 5.5.2.4 void [Output::print\\_output\\_table](#) ( std::ofstream & *output\_stream*, std::vector< [output\\_container](#) > & *output\_table*, double *s*, [Method::initial\\_condition](#) *initial\_condition*, [Method::method](#) *method* ) [static]
- 5.5.2.5 void [Output::print\\_output\\_table\\_latex](#) ( std::ofstream & *output\_stream*, std::vector< [output\\_container](#) > & *output\_table*, double *s*, [Method::initial\\_condition](#) *initial\_condition*, [Method::method](#) *method* ) [static]
- 5.5.2.6 void [Output::process\\_output](#) ( std::vector< [output\\_container](#) > & *output\_table*, [Method::initial\\_condition](#) *initial\_condition*, [Method::method](#) *method* ) [static]

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

- include/[Output.hpp](#)
- source/[Output.cpp](#)

## 5.6 [Output::output\\_data](#) Struct Reference

```
#include <Output.hpp>
```

### Public Attributes

- int [k](#)
- double [error\\_1\\_norm](#)
- double [error\\_2\\_norm](#)
- double [error\\_sup\\_norm](#)
- double [convergence\\_rate\\_1\\_norm](#)

- double [convergence\\_rate\\_2\\_norm](#)
- double [convergence\\_rate\\_sup\\_norm](#)

### 5.6.1 Member Data Documentation

5.6.1.1 double Output::output\_data::convergence\_rate\_1\_norm

5.6.1.2 double Output::output\_data::convergence\_rate\_2\_norm

5.6.1.3 double Output::output\_data::convergence\_rate\_sup\_norm

5.6.1.4 double Output::output\_data::error\_1\_norm

5.6.1.5 double Output::output\_data::error\_2\_norm

5.6.1.6 double Output::output\_data::error\_sup\_norm

5.6.1.7 int Output::output\_data::k

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

- include/[Output.hpp](#)



## Chapter 6

# File Documentation

### 6.1 `include/Finite_Grid.hpp` File Reference

One-dimensional uniform grid with periodic boundary conditions.

```
#include <string>
#include <map>
#include <iostream>
#include <fstream>
#include <cmath>
```

#### Classes

- class [Finite\\_Grid](#)

#### 6.1.1 Detailed Description

One-dimensional uniform grid with periodic boundary conditions.

Example paragraph header

Uniform discretization of a one-dimensional interval with periodic boundary conditions. Implemented using `valarray`.

**Bug** No known bugs

### 6.2 `include/Method.hpp` File Reference

```
#include <map>
#include "Finite_Grid.hpp"
```

#### Classes

- class [Method](#)
- struct [Method::method](#)
- struct [Method::initial\\_condition](#)

## 6.3 include/Output.hpp File Reference

```
#include <string>
#include <map>
#include <iostream>
#include <fstream>
#include <cmath>
#include <vector>
#include "Finite_Grid.hpp"
#include "Method.hpp"
```

### Classes

- class [Output](#)
- struct [Output::output\\_data](#)

## 6.4 main.cpp File Reference

```
#include <iostream>
#include <cmath>
#include <fstream>
#include "H5Cpp.h"
#include "boost/lexical_cast.hpp"
#include "boost/thread.hpp"
#include "boost/bind.hpp"
#include "Finite_Grid.hpp"
#include "Method.hpp"
#include "Output.hpp"
```

### Functions

- void [computation](#) (int *k\_min*, int *k\_max*, double *a*, double *S*, double *T*, [Method::initial\\_condition](#) *initial\_condition*, [Method::method](#) *method*)
- int [main](#) (int *argc*, char \*\**argv*)

### 6.4.1 Function Documentation

6.4.1.1 void [computation](#) ( int *k\_min*, int *k\_max*, double *a*, double *S*, double *T*, [Method::initial\\_condition](#) *initial\_condition*, [Method::method](#) *method* )

6.4.1.2 int [main](#) ( int *argc*, char \*\* *argv* )



## 6.5 source/Finite\_Grid.cpp File Reference

```
#include "Finite_Grid.hpp"
#include "Method.hpp"
#include "Output.hpp"
#include <iostream>
#include <fstream>
#include <cmath>
#include "boost/lexical_cast.hpp"
#include "boost/format.hpp"
```

## 6.6 source/Method.cpp File Reference

```
#include "Method.hpp"
#include "Finite_Grid.hpp"
#include "boost/math/special_functions/sign.hpp"
#include <cmath>
#include <iostream>
#include <fstream>
#include <string>
```

## 6.7 source/Output.cpp File Reference

```
#include <string>
#include <map>
#include <iostream>
#include <fstream>
#include <cmath>
#include "boost/lexical_cast.hpp"
#include "boost/format.hpp"
#include "Finite_Grid.hpp"
#include "Method.hpp"
#include "Output.hpp"
#include "plplot/plplot.h"
```

# Index

- accessed\_last
  - Finite\_Grid, [10](#)
- computation
  - Finite\_Grid, [9](#)
  - main.cpp, [18](#)
- convergence\_rate\_1\_norm
  - Output::output\_data, [15](#)
- convergence\_rate\_2\_norm
  - Output::output\_data, [15](#)
- convergence\_rate\_sup\_norm
  - Output::output\_data, [15](#)
- error\_1\_norm
  - Output::output\_data, [15](#)
- error\_2\_norm
  - Output::output\_data, [15](#)
- error\_sup\_norm
  - Output::output\_data, [15](#)
- FCT
  - Method, [12](#)
- Finite\_Grid, [9](#)
  - accessed\_last, [10](#)
  - computation, [9](#)
  - get\_CFL, [9](#)
  - get\_M, [10](#)
  - get\_N, [10](#)
  - get\_T, [10](#)
  - get\_a, [9](#)
  - get\_initial\_data, [10](#)
  - get\_local\_M, [10](#)
  - get\_local\_solution\_data, [10](#)
  - get\_solution\_data, [10](#)
  - initial\_Data, [10](#)
  - initialize\_Grid, [10](#)
  - Method, [12](#)
  - operator(), [10](#)
  - operator=, [10](#)
- Fromm
  - Method, [12](#)
- function
  - Method::initial\_condition, [11](#)
  - Method::method, [13](#)
- FvL
  - Method, [12](#)
- Gaussian\_Pulse
  - Method, [12](#)
- get\_CFL
  - Finite\_Grid, [9](#)
- get\_M
  - Finite\_Grid, [10](#)
- get\_N
  - Finite\_Grid, [10](#)
- get\_T
  - Finite\_Grid, [10](#)
- get\_a
  - Finite\_Grid, [9](#)
- get\_initial\_data
  - Finite\_Grid, [10](#)
- get\_local\_M
  - Finite\_Grid, [10](#)
- get\_local\_solution\_data
  - Finite\_Grid, [10](#)
- get\_solution\_data
  - Finite\_Grid, [10](#)
- include/Finite\_Grid.hpp, [17](#)
- include/Method.hpp, [17](#)
- include/Output.hpp, [18](#)
- index\_output\_table
  - Output, [14](#)
- initial\_Data
  - Finite\_Grid, [10](#)
- initial\_condition\_function
  - Method, [12](#)
- initial\_conditions
  - Method, [12](#)
- initialize\_Grid
  - Finite\_Grid, [10](#)
- k
  - Output::output\_data, [15](#)
- LF
  - Method, [12](#)
- LW
  - Method, [12](#)
- LW4
  - Method, [12](#)
- main
  - main.cpp, [18](#)
- main.cpp, [18](#)
  - computation, [18](#)
  - main, [18](#)
- Method, [11](#)
  - FCT, [12](#)
  - Finite\_Grid, [12](#)

- Fromm, [12](#)
- FvL, [12](#)
- Gaussian\_Pulse, [12](#)
- initial\_condition\_function, [12](#)
- initial\_conditions, [12](#)
- LF, [12](#)
- LW, [12](#)
- LW4, [12](#)
- method\_function, [12](#)
- methods, [12](#)
- Semicircle, [12](#)
- Square\_Wave, [12](#)
- Upwind, [12](#)
- Method::initial\_condition, [10](#)
  - function, [11](#)
  - name, [11](#)
- Method::method, [13](#)
  - function, [13](#)
  - name, [13](#)
- method\_function
  - Method, [12](#)
- methods
  - Method, [12](#)
- name
  - Method::initial\_condition, [11](#)
  - Method::method, [13](#)
- operator()
  - Finite\_Grid, [10](#)
- operator=
  - Finite\_Grid, [10](#)
- Output, [13](#)
  - index\_output\_table, [14](#)
  - output\_container, [14](#)
  - plot, [14](#)
  - plot\_one, [14](#)
  - print\_output\_table, [14](#)
  - print\_output\_table\_latex, [14](#)
  - process\_output, [14](#)
- Output::output\_data, [14](#)
  - convergence\_rate\_1\_norm, [15](#)
  - convergence\_rate\_2\_norm, [15](#)
  - convergence\_rate\_sup\_norm, [15](#)
  - error\_1\_norm, [15](#)
  - error\_2\_norm, [15](#)
  - error\_sup\_norm, [15](#)
  - k, [15](#)
- output\_container
  - Output, [14](#)
- plot
  - Output, [14](#)
- plot\_one
  - Output, [14](#)
- print\_output\_table
  - Output, [14](#)
- print\_output\_table\_latex
  - Output, [14](#)
- process\_output
  - Output, [14](#)
- Semicircle
  - Method, [12](#)
- source/Finite\_Grid.cpp, [19](#)
- source/Method.cpp, [19](#)
- source/Output.cpp, [19](#)
- Square\_Wave
  - Method, [12](#)
- Upwind
  - Method, [12](#)