

Computer_Homework1

Generated by Doxygen 1.9.1

1 Computer Homework 1	1
1.1 Requirements	1
1.2 Installation	1
1.3 How To Use	1
1.4 Copyright	1
1.5 License	1
2 Finite difference Method	3
2.1 Complexity	4
2.2 Accuracy	4
2.3 Convergence	4
3 File Index	5
3.1 File List	5
4 File Documentation	7
4.1 hw1.cpp File Reference	7
4.1.1 Detailed Description	8
4.1.2 Function Documentation	8
4.1.2.1 HW1()	8
4.2 hw1.hpp File Reference	8
4.2.1 Detailed Description	9
4.2.2 Function Documentation	10
4.2.2.1 HW1()	10
4.3 main.cpp File Reference	10
4.3.1 Detailed Description	11
Index	13

Chapter 1

Computer Homework 1

Solve Kepler problem via finite difference Method

1.1 Requirements

To install this program, you should have

- C++ compiler like g++
- gnu make

1.2 Installation

Type make, then we can see hw1 executable file in bin directory

1.3 How To Use

Execute hw1 then, it will interactively read

- initial condition
- number of grid points to evaluate
- output file name

Then it computes and saves solution to file. You can plot the result using usual plotting software like gnuplot

1.4 Copyright

Copyright 2021 pistack (Junho Lee). All rights reserved.

1.5 License

This project is released under the GNU Lesser General Public License v3.0.

Chapter 2

Finite difference Method

To solve Kepler problem, we need to solve

$$\frac{d^2\zeta}{dt^2} = \frac{1}{\zeta^3} - \frac{1}{\zeta^2} \quad (2.1)$$

with initial condition

$$\begin{aligned}\zeta(t_0) &= \zeta_0 \\ \zeta'(t_0) &= \zeta'_0\end{aligned}$$

To solve above 2nd order ordinary differential equation (2.1) numerically, we need to approximate 2nd derivative as finite difference. Suppose that the solution $\zeta(t)$ has continuous 4th order derivative in the Domain $[t_0, t_f]$. then

$$\zeta(x+h) = \zeta(x) + \zeta'(x)h + \frac{1}{2!}\zeta''(x)h^2 + \frac{1}{3!}\zeta'''(x)h^3 + \frac{1}{4!}\zeta^{(4)}(\eta)h^4 \quad (2.2)$$

for some $\eta(x, h) \in (t_0, t_f)$. Using 4th order Taylor approximation (2.2), we can get following equation

$$\zeta(x-h) - 2\zeta(x) + \zeta(x+h) = h^2\zeta''(x) + O(h^4) \quad (2.3)$$

Next uniformly divide the domain $[t_0, t_f]$ into n sub intervals. let x_i be the end points of the sub intervals then for $0 \leq i \leq n$,

$$x_i = t_0 + ih \quad (2.4)$$

, where $h = (t_f - t_0)/n$. Now for $0 \leq i \leq n$, define ζ_i as following

$$\zeta_i = \zeta(x_i) \quad (2.5)$$

Then we can rewrite finite difference equation (2.3) as following

$$\zeta_{i-1} - 2\zeta_i + \zeta_{i+1} = h^2\zeta''_i + O(h^4) \quad (2.6)$$

for $1 \leq i \leq n-1$. Plug this equation (2.6) into 2nd order ode (2.1), then we have following recurrence relation

$$\zeta_{i-1} - 2\zeta_i + \zeta_{i+1} = h^2 \left(\frac{1}{\zeta_i^3} - \frac{1}{\zeta_i^2} \right) \quad (2.7)$$

In above equation (2.7), we truncate, so local truncation error is $O(h^4) = O(n^{-4})$. Therefore global truncation error can be roughly estimated to $O(n^{-3})$. To solve recurrence relation, we need to know both ζ_0 and ζ_1 . However only ζ_0 is explicitly given by the initial condition. To approximate ζ_1 with $O(n^{-3})$ error bound, I use 2nd order Taylor expansion.

$$\zeta_1 \approx \zeta_0 + \zeta'_0 h + \frac{1}{2!}\zeta''_0 h^2 \quad (2.8)$$

ζ''_0 can be derived by 2nd order ode (2.1)

$$\zeta''_0 = \frac{1}{\zeta_0^3} - \frac{1}{\zeta_0^2} \quad (2.9)$$

2.1 Complexity

Clearly

$$O(n).$$

2.2 Accuracy

Global turncation error is roughly estimated by

$$O(n^{-3}).$$

2.3 Convergence

- Initial Condition

$$\zeta(0) = 0.9$$

$$\zeta'(0) = 0$$

- Initial time: 0
- Final time: 10

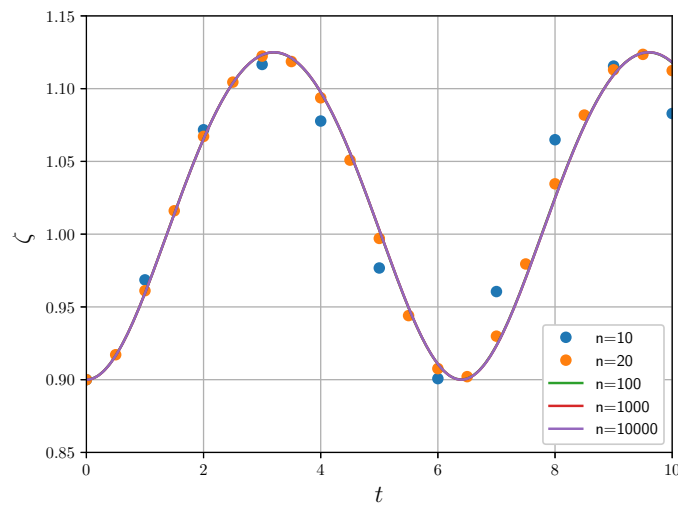


Figure 2.1 Convergence plot

Chapter 3

File Index

3.1 File List

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

hw1.cpp	Code for homework1 of Computer1 class in Yonsei University Use finite difference method to solve Kepler problem	7
main.cpp	Main program for homework1 of Computer1 class in Yonsei University Interactively reads initial condition, number of grid points to evaluate and output file name then computes and saves solution	10
hw1.hpp	Header file for homework1 of Computer1 class in Yonsei University Use explicit mid point Method to solve Kepler problem	8

Chapter 4

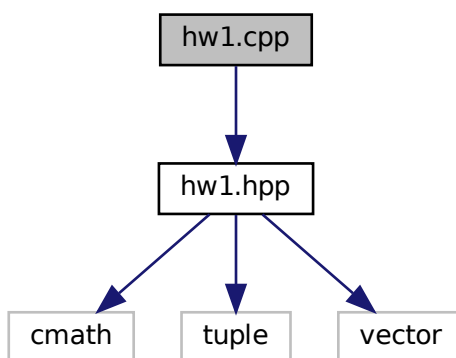
File Documentation

4.1 hw1.cpp File Reference

code for homework1 of Computer1 class in Yonsei University Use finite difference method to solve Kepler problem

```
#include "hw1.hpp"
```

Include dependency graph for hw1.cpp:



Functions

- `tuple< vector< double >, vector< double > > HW1 (double t0, double t1, int n, double y0, double y0p)`
HW1: Solve Kepler problem via finite difference Method.

4.1.1 Detailed Description

code for homework1 of Computer1 class in Yonsei University Use finite difference method to solve Kepler problem

Author

pistack (Junho Lee)

Date

2021. 10. 10.

4.1.2 Function Documentation

4.1.2.1 HW1()

```
tuple<vector<double>, vector<double> > HW1 (
    double t0,
    double t1,
    int n,
    double y0,
    double y0p )
```

HW1: Solve Kepler problem via finite difference Method.

Parameters

<i>t0</i>	initial time
<i>t1</i>	final time
<i>n</i>	number of gird points to evaluate
<i>y0</i>	initial condition for zeta
<i>y0p</i>	intial condition for derivative of zeta

Returns

tuple of time and zeta

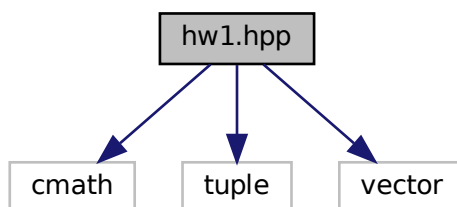
4.2 hw1.hpp File Reference

Header file for homework1 of Computer1 class in Yonsei University Use explicit mid point Method to solve Kepler problem.

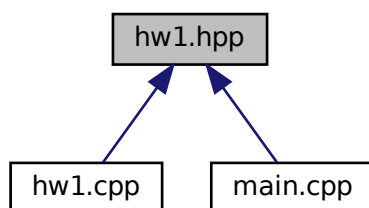
```
#include <cmath>
#include <tuple>
```

```
#include <vector>
```

Include dependency graph for hw1.hpp:



This graph shows which files directly or indirectly include this file:



Functions

- `std::tuple< std::vector< double >, std::vector< double > > >` [HW1](#) (double t0, double t1, int n, double y0, double y0p)

HW1: Solve Kepler problem via finite difference Method.

4.2.1 Detailed Description

Header file for homework1 of Computer1 class in Yonsei University Use explicit mid point Method to solve Kepler problem.

Author

pistack (Junho Lee)

Date

2021. 10. 10.

4.2.2 Function Documentation

4.2.2.1 HW1()

```
std::tuple<std::vector<double>, std::vector<double> > HW1 (
    double t0,
    double t1,
    int n,
    double y0,
    double y0p )
```

HW1: Solve Kepler problem via finite difference Method.

Parameters

<i>t0</i>	initial time
<i>t1</i>	final time
<i>n</i>	number of gird points to evaluate
<i>y0</i>	initial condition for zeta
<i>y0p</i>	intial condition for derivative of zeta

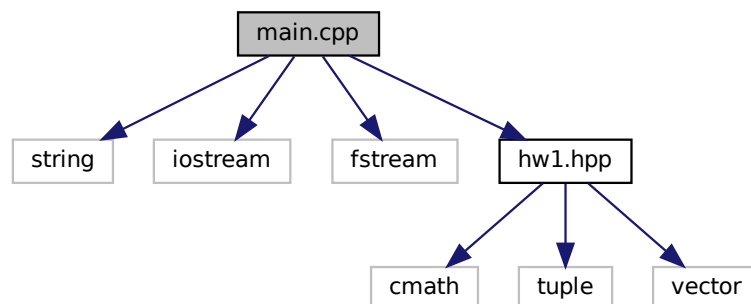
Returns

tuple of time and zeta

4.3 main.cpp File Reference

main program for homework1 of Computer1 class in Yonsei University Interactively reads inital condition, number of gird points to evaluate and output file name then computes and saves solution.

```
#include <string>
#include <iostream>
#include <fstream>
#include "hw1.hpp"
Include dependency graph for main.cpp:
```



Functions

- `int main` (void)

4.3.1 Detailed Description

main program for homework1 of Computer1 class in Yonsei University Interactively reads initial condition, number of grid points to evaluate and output file name then computes and saves solution.

Author

pistack (Junho Lee)

Date

2021. 10. 10.

Index

HW1
 hw1.cpp, [8](#)
 hw1.hpp, [10](#)
hw1.cpp, [7](#)
 HW1, [8](#)
hw1.hpp, [8](#)
 HW1, [10](#)

main.cpp, [10](#)