Computer_Homework2

Generated by Doxygen 1.9.1

Chapter 1

Computer Homework 2

Solve Kepler problem via numerical integration

1.1 Requirements

To install this program, you should have

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

1.2 Installation

- gnu make
  - Type make, then we can see hw2 executable file in bin directory
- cmake
  1. make build directory
  2. go to build directory and type cmake .. -DPRECISION_LEVEL precision_level
     - precision_level 0: float
     - precision_level 1: double
  3. Type make then we can see hw2 executable in build directory

1.3 How To Use

Execute hw2 then, it will interactively read

- inital condition
- number of gird 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

1.5 License

Chapter 2

Numerical Integration

By conservation of energy, we can derive following integral equation.

$$t - t_0 = \int_{\zeta_{min}}^{\zeta(t)} \frac{\zeta'}{\sqrt{-a(\zeta' - \zeta_{min})(\zeta_{max} - \zeta')}} \mathrm{d}\zeta' \tag{2.1}$$

,where

- ζ_{min} is periapsis (minimum value of ζ),
- ζ_{max} is apoapsis (maximum value of ζ)
- α is parameter defined by following relation

$$\begin{aligned} \alpha &= \frac{1}{\zeta_{min}^2} - \frac{2}{\zeta_{min}} \\ &= \frac{1}{\zeta_{max}^2} - \frac{2}{\zeta_{max}} \end{aligned}$$

Using Vieta's Formula, we could simplify above relation.

$$\zeta_{max} = \frac{\zeta_{min}}{2\zeta_{min} - 1} \tag{2.2}$$

$$\alpha = -\frac{1}{\zeta_{min}\zeta_{max}} \tag{2.3}$$

To solve above integral equation (2.1) we need to view time t as a function of ζ with domain $D = [\zeta_{min}, \zeta_{max}]$. Now uniformly divide the domain D into n sub intervals. Let ζ_i be end points of the sub intervals then for $0 \leq \zeta \leq n$,

$$\zeta_i = \zeta_{min} + i\frac{\zeta_{max} - \zeta_{min}}{n} \tag{2.4}$$

Define $t_i = t(\zeta_i)$ then we have following recurrence relation for $i \geq 1$,

$$t_i = t_{i-1} + \int_{\zeta_{i-1}}^{\zeta_i} \frac{\zeta'}{\sqrt{-\alpha(\zeta' - \zeta_{min})(\zeta_{max} - \zeta')}} \mathrm{d}\zeta' \tag{2.5}$$

However, due to the divergence feature of the integrand at $\zeta_0 = \zeta_{min}$ and $\zeta_n = \zeta_{max}$, It is hard to approximate such integral directly. To remove singularity, first consider the following equation.

$$\begin{aligned} \frac{\zeta'}{\sqrt{-\alpha(\zeta' - \zeta_{min})(\zeta_{max} - \zeta')}} &= \frac{\zeta_{min}}{\sqrt{-\alpha}(\zeta_{max} - \zeta_{min})} \frac{\sqrt{\zeta_{max} - \zeta'}}{\sqrt{\zeta' - \zeta_{min}}} \\ &+ \frac{\zeta_{max}}{\sqrt{-\alpha}(\zeta_{max} - \zeta_{min})} \frac{\sqrt{\zeta' - \zeta_{min}}}{\sqrt{\zeta_{max} - \zeta'}} \end{aligned} \tag{2.6}$$

Then we can seperate integral into two parts. Substitute $u = \sqrt{\zeta' - \zeta_{min}}$ and $v = \sqrt{\zeta_{max} - \zeta'}$ to the first and second part of integral respectively, then

$$\text{integral}_i = \frac{2\zeta_{min}}{\sqrt{-\alpha}(\zeta_{max} - \zeta_{min})} \int_{\sqrt{\zeta_{i-1} - \zeta_{min}}}^{\sqrt{\zeta_i - \zeta_{min}}} \sqrt{\zeta_{max} - \zeta_{min} - u^2} \mathrm{d}u \tag{2.7}$$

$$+ \frac{2\zeta_{max}}{\sqrt{-\alpha}(\zeta_{max} - \zeta_{min})} \int_{\sqrt{\zeta_{max} - \zeta_i}}^{\sqrt{\zeta_{max} - \zeta_{i-1}}} \sqrt{\zeta_{max} - \zeta_{min} - v^2} \mathrm{d}v \tag{2.8}$$

by above equation (2.7) , We can deduce

$$\begin{aligned} t_f - t_0 &= \pi \frac{\zeta_{min} + \zeta_{max}}{\sqrt{-\alpha}} \\ &= \pi a^{3/2} \end{aligned} \tag{2.9}$$

,where $a = (\zeta_{min} + \zeta_{max})/2$.

2.1 Approximation

Let $f(u) = \sqrt{\zeta_{max} - \zeta_{min} - u^2}$ and define u_i and v_i as following

$$u_i = \sqrt{\zeta_i - \zeta_{min}} \tag{2.10}$$

$$v_i = \sqrt{\zeta_{max} - \zeta_i} \tag{2.11}$$

then we have following relation

$$u_i = f(v_i) \tag{2.12}$$

$$v_i = f(u_i) \tag{2.13}$$

$$v_{n-i} = u_i \tag{2.14}$$

To exploit above relations (2.12)–(2.14) I use Trapezoidal rule with unequivalently spaced interval. Then,

$$\text{integral}_i = c_1 \frac{u_i - u_{i-1}}{2} (v_i + v_{i-1}) + c_2 \frac{v_i - v_{i-1}}{2} (u_i + u_{i-1}) \tag{2.15}$$

where

$$c_1 = \frac{2\zeta_{min}}{\sqrt{-\alpha}(\zeta_{max} - \zeta_{min})}$$

$$c_2 = \frac{2\zeta_{max}}{\sqrt{-\alpha}(\zeta_{max} - \zeta_{min})}$$

2.2 Complexity

Complex is clearly

$$O(n)$$

2.3 Accuracy

Error bound is given by

$$\begin{aligned}\text{Error bound} &\leq M\sum_{i=1}^{n}(u_i - u_{i-1})^3 \\ &= M\left(\frac{\zeta_{max} - \zeta_{min}}{n}\right)^{3/2}\sum_{i=1}^{n}\left(\frac{1}{\sqrt{i}+\sqrt{i-1}}\right)^3 \\ &< MC\left(\frac{\zeta_{max} - \zeta_{min}}{n}\right)^{3/2}\end{aligned}$$

So, the error bound is

$$O(n^{-3/2})$$

2.4 Convergence

initial condition

$$\begin{aligned}t_0 &= 0 \\ \zeta_{min} &= 0.9\end{aligned}$$

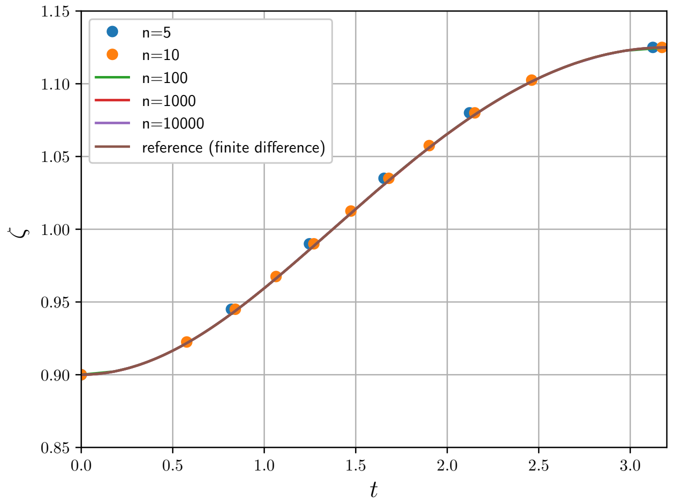


Figure 2.1 Convergence plot: single precision

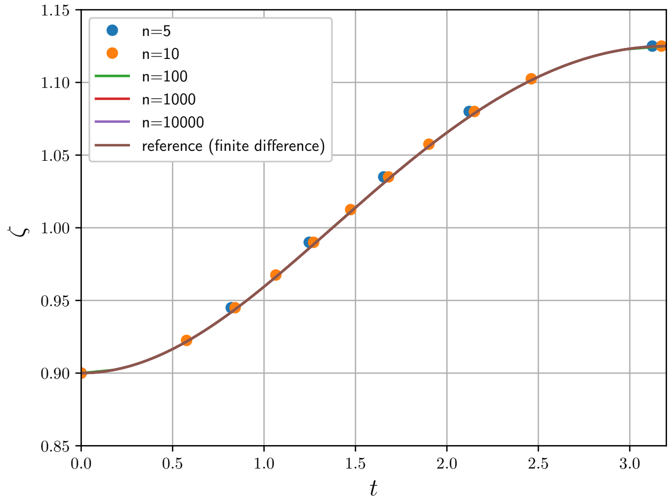


Figure 2.2 Convergence plot:double precision

2.5 Practical Error Bound

We know that the exact value of t_f is $\pi a^{3/2}$. Where $a = (\zeta_{min} + \zeta_{max})/2.0$. So, error can be evaluated by

$$\text{error} = \left| t_n - \pi a^{3/2} \right| \tag{2.16}$$

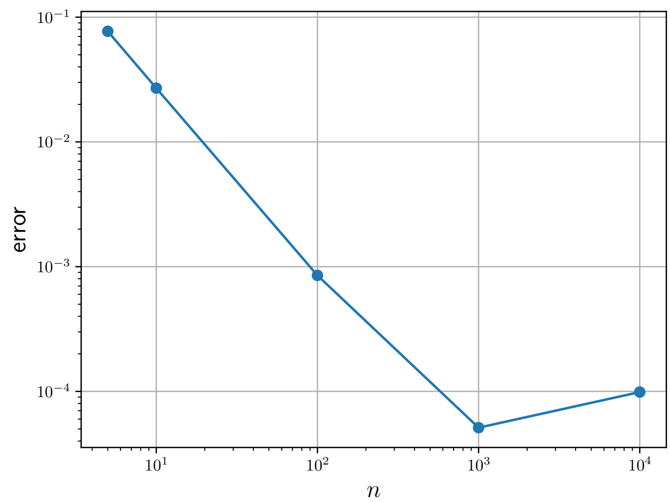


Figure 2.3 Error Analysis Plot: single precision

Due to the floating point turncation error, accuracy got worse at $n > 10^3$.

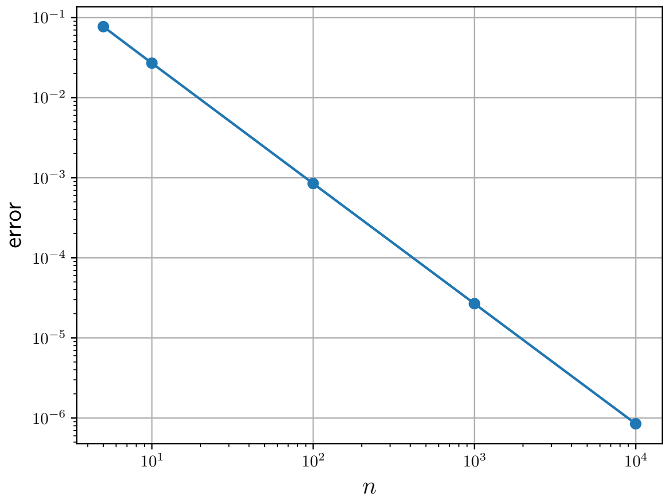


Figure 2.4 Error Analysis Plot: double precision

The practical error bound can be estimated to

$$O(n^{-3/2}) \tag{2.17}$$

It is same as theorical error estimation $O(n^{-3/2})$.

Chapter 3

File Index

3.1 File List

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

Chapter 4

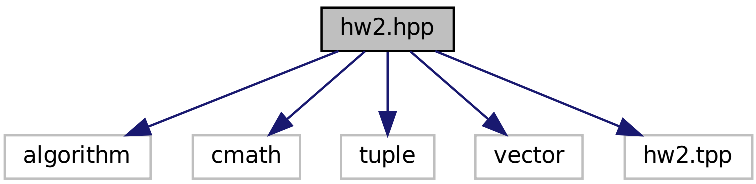
File Documentation

4.1 hw2.hpp File Reference

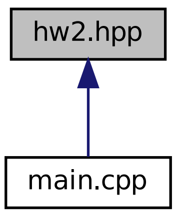
headerfile for homework2 of Computer1 class in Yonsei University Use numerical integration to solve Kepler problem

```
#include <algorithm>
#include <cmath>
#include <tuple>
#include <vector>
#include "hw2.tpp"
```

Include dependency graph for hw2.hpp:



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



Functions

- template< typename T >
  std::tuple< std::vector< T >, std::vector< T > > HW2 (T zeta_min, T t0, int n)
  HW2: Solve Kepler problem via numerical integration from zeta_min to zeta_max If type T is not equal to one of float, double, long double then behavior of HW2 is undefined.

4.1.1 Detailed Description

headerfile for homework2 of Computer1 class in Yonsei University Use numerical integration to solve Kepler problem

Author

pistack (Junho Lee)

Date

2021. 10. 28.

4.1.2 Function Documentation

4.1.2.1 HW2()

```
template<typename T >
std::tuple<std::vector<T>, std::vector<T> > HW2 (
            T zeta_min,
            T t0,
            int n )
```

HW2: Solve Kepler problem via numerical integration from zeta_min to zeta_max If type T is not equal to one of float, double, long double then behavior of HW2 is undefined.

Parameters

zeta_min	minimum value of zeta, for constraint motion $0.5 <$ zeta_min < 1
t0	initial time
n	number of points to evaluate

Returns

tuple of time and zeta

See also

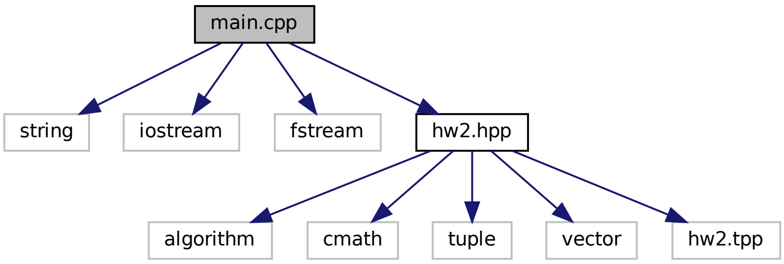
Numerical Integration

4.2 main.cpp File Reference

main program for homework2 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 "hw2.hpp"
```

Include dependency graph for main.cpp:



Macros

- #define **PRECISION** float
- #define **DIGITS** 6

Functions

- int **main** (void)

4.2.1 Detailed Description

main program for homework2 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.

Author

pistack (Junho Lee)

Date

2021. 11. 2.

Index