Numerical Mathematics Assignment 7 Peter-Jan Derks 11065540

Exercise 1

PLANETARIUM shows the movement of planets. It calculates the orbits of planets in two or three dimensions by using an ODE solver to solve a system of first-order differential equations derived from Newton's laws of gravitation.

The function accepts 4 input parameters. M is row vector with the masses of the planets. IP is a matrix IP of initial positions with size n x p n being the amount of dimensions and p being the amount of planets. IV is a matrix of initial velocities with again size n x p. T should be a value for the length of time over which the movements should be computed. The function makes a picture of the orbits and returns as output a column vector TIME of points in time and a matrix Z of corresponding positions and velocities of the planets.

An example of an input for the function is:

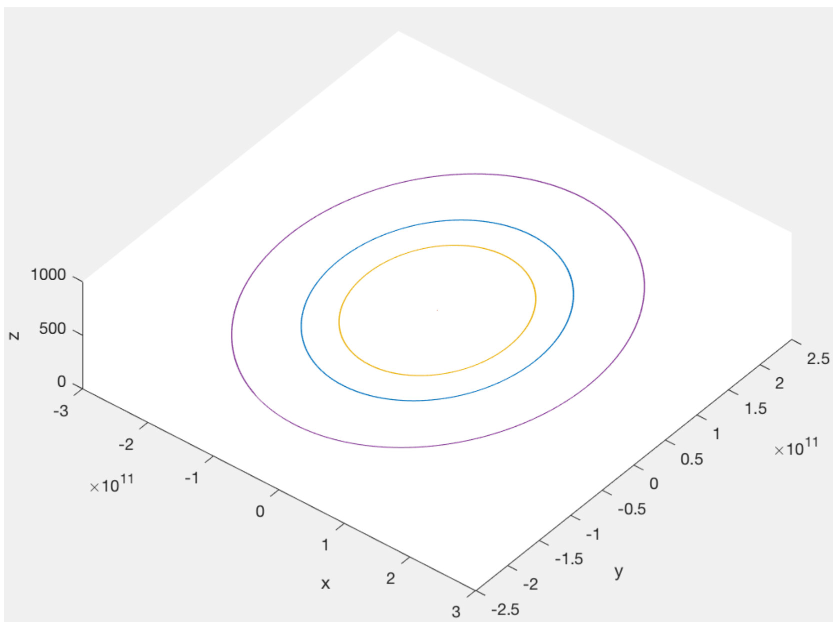
```
>> planetarium([5.972e+24,1.989e+30,4.868e+24,6.418e+23],[149.6e+9, 11,
108.2e+9,227.9e+9;12,13,14,15; 500,501,502,503],
[1,3,4,5;29780,5,35000,24076;0,0,0,0],100000000)
```

This gives output:

```
time =

   1.0e+08 *

         0
    0.0020
    0.0040
    0.0060
    0.0080
    0.0100
    0.0120
    0.0140
    0.0160
    0.0180
    0.0200
    etc....
```



Z =

1.0e+11 *

Columns 1 through 8

1.4960	0.0000	0.0000	0.0000	0.0000	0.0000	1.0820	0.0000
1.4948	0.0597	0.0000	0.0000	0.0000	0.0000	1.0797	0.0701
1.4912	0.1192	0.0000	0.0000	0.0000	0.0000	1.0729	0.1399
1.4853	0.1786	0.0000	0.0000	0.0000	0.0000	1.0616	0.2091
1.4770	0.2377	0.0000	0.0000	0.0000	0.0000	1.0458	0.2774
1.4663	0.2964	0.0000	0.0000	0.0001	0.0000	1.0256	0.3446
1.4533	0.3547	0.0000	0.0000	0.0001	0.0000	1.0011	0.4103
1.4380	0.4123	0.0000	0.0000	0.0001	0.0000	0.9724	0.4743
1.4205	0.4694	0.0000	0.0000	0.0001	0.0000	0.9396	0.5363

etc…

Description

The built-in odesolver ode45 calls a nested function: ‘system of equations’. Take n to be [number dimensions · number of planets]. System of equations forms a total amount of 2 · n equations. It is split up into two parts. First, one for loop is used to make n equations for the velocities. The velocities are the integral of the accelerations. For three dimensions, each planet has an equation for velocity in the x y and z direction.

Secondly, three for loops are used to make n equations for the accelerations. If there are 3 planets in two dimension, with masses m1, m2, m3 the acceleration of the first planet can be derived from Newton’s laws of gravitation:

$$a_1 = \frac{Gm_2}{(x_1 - x_2)^2 + (y_1 - y_2)^2}\hat{r} + \frac{Gm_3}{(x_1 - x_3)^2 + (y_1 - y_3)^2}\hat{r}$$

Error measurement

The tolerance of ode45 is changed using options to make the function more precise. The function will never plot the orbit as they are in reality because therefore I would need to use all the planets in the entire universe because the all exert a gravitational force on eachother.

Questions

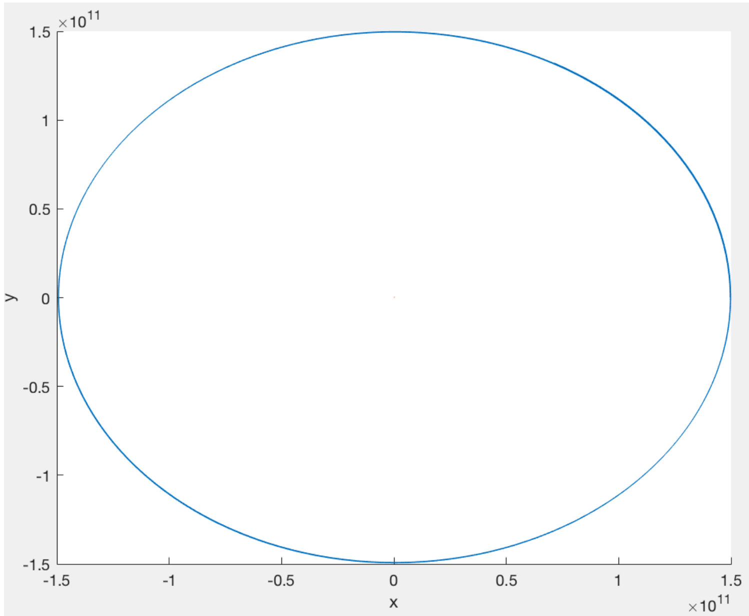
The following values were used source: (Wikipedia).

	Sun	Earth	Venus	Mars
Mass in kg	$1.989 \cdot 10^{30}$	$5.972 \cdot 10^{24}$	$4.868 \cdot 10^{24}$	$6.417 \cdot 10^{23}$
Orbital speed around the sun in meters/second	x	$2.9780 \cdot 10^{4}$	$3.5000 \cdot 10^{4}$	$2.4076 \cdot 10^{4}$
Distance to Sun in meters	x	$1.496 \cdot 10^{11}$	$1.082 \cdot 10^{11}$	$2.2279 \cdot 10^{11}$

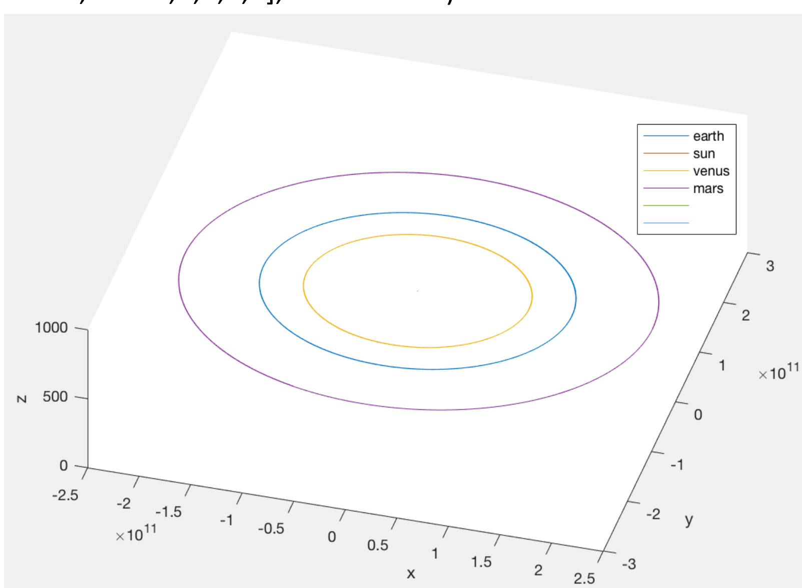
Pictures

The earth circling around the sun in 2 dimensions

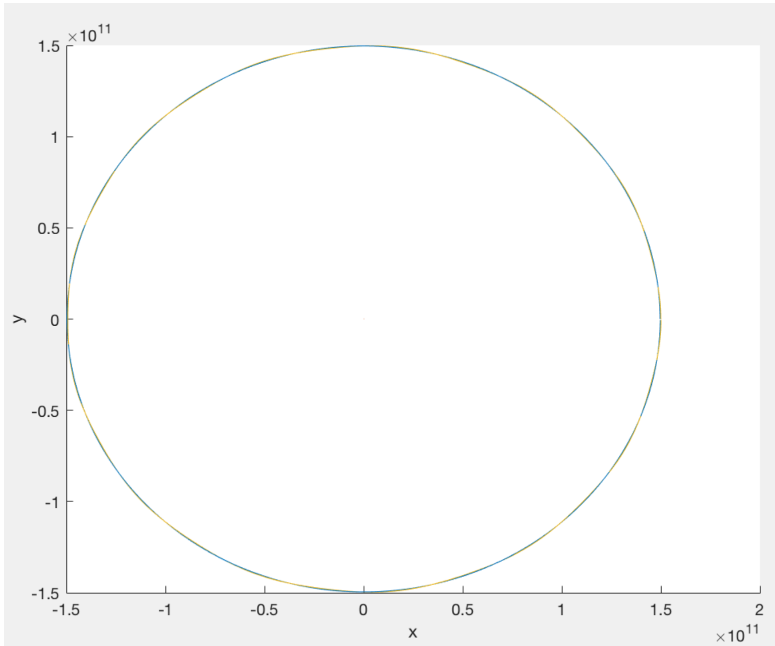
```
>> planetarium ([5.972e+24,1.989e+30],[149.6e+9, 11;12,13], [1,3;29780,5],100000000)
```



Venus, Earth and Mars orbiting around a sun in 3 dimensions
>> planetarium([5.972e+24,1.989e+30,4.8675e+24,6.418e+23],[149.6e+9, 11, 108.2e+9,227.9e+9;12,13,14,15; 3e+5,4e+5,5e+5,6e+5], [1,3,4,5;29780,5,35000,24076;0,0,0,0],1000000000)



A sun-earth-moon system; >>planetarium([5.972e+24,1.989e+30,7.342e+22],[149.6e+9, 11, 1.4998e+11;12,13,14; 3,4,5], [1,3,4;29780,5,30803;0,0,0],3.155e+7)



Exercise 2

[TIME,X,U] = KortewegdeVries(U0, L, T, S, N,) shows a moderatly high waves in a shallow chanel by solving the Korteweg-de Vries equation by discretizing the spacial variable.

The function accepts 6 input parameters of which two are optional. U0 is the (almost) periodic initial conditions U0 (a function handle). L is half of the length of the interval of the wave as the interval is -L < x < L. T is the length of time. S is the number of time steps, S, N is an optional input and determines the number of spatial discretization points. P is an optional input and determines if a 3-dimensional plot is outputted or a 2-dimensional wave animation.

The function gives three outputs and can plot either a 2-dimensional wave animationoutput or a 3-dimensional plot. TIME is a vector of points in time. X is a vector of discretization points. U is a matrix containing the profile of the solution at points in time and space, such that it is easy to plot.

An example of an input for the function is:
>> [Time,dx,u] = KortewegdeVries(@(x) 2./((cosh(x)).^2),5,100,100,100,1)

This gives output:

Time =

    1
    2

3
4
5
6
7
8
9
etc

dx =

Columns 1 through 8

-5.0000	-4.8990	-4.7980	-4.6970	-4.5960	-4.4949	-4.3939	-4.2929

etc

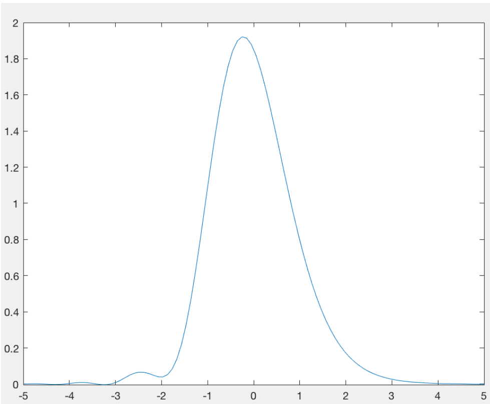
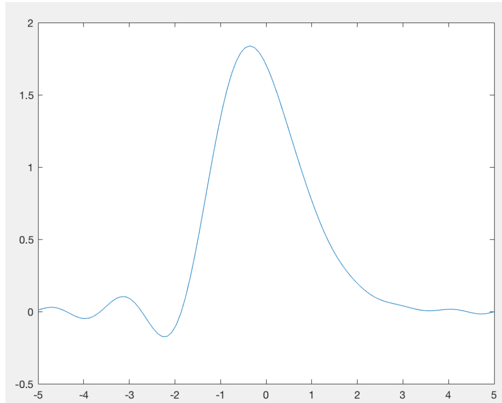
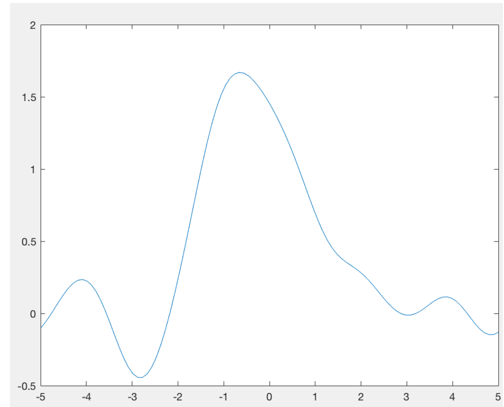
u =

Columns 1 through 8

0.0004	0.0004	0.0005	0.0007	0.0008	0.0010	0.0012	0.0015
0.0004	0.0005	0.0005	0.0007	0.0008	0.0010	0.0012	0.0015
0.0004	0.0005	0.0005	0.0007	0.0008	0.0010	0.0012	0.0014
0.0004	0.0005	0.0005	0.0007	0.0008	0.0010	0.0012	0.0014
0.0004	0.0005	0.0005	0.0006	0.0008	0.0009	0.0011	0.0014
0.0004	0.0005	0.0005	0.0006	0.0008	0.0009	0.0011	0.0014
0.0004	0.0005	0.0005	0.0006	0.0008	0.0009	0.0011	0.0014
0.0004	0.0005	0.0005	0.0006	0.0007	0.0009	0.0011	0.0013
0.0005	0.0005	0.0005	0.0006	0.0007	0.0009	0.0011	0.0013
0.0005	0.0005	0.0005	0.0006	0.0007	0.0009	0.0011	0.0013
0.0005	0.0005	0.0005	0.0006	0.0007	0.0009	0.0011	0.0013
0.0005	0.0005	0.0005	0.0006	0.0007	0.0009	0.0010	0.0012

Picture

The input produces additionally produces the following animation (of which I took screenshots at different times):

Description

$$\frac{\partial u}{\partial t} = -\frac{\partial^3 u}{\partial x^3} - 6u\frac{\partial u}{\partial x} = -\frac{\partial}{\partial x}\left(\frac{\partial^2 u}{\partial x^2} + 3u^2\right).$$

To solve the Korteweg-de Vries equation with periodic boundary conditions the entire solution u(x) needs to be calculated in one go. The function does this by discretizing the spatial variable x. Using centered differences two equations can be found that represent i) the first order derivative of u:

$$\frac{u_{i+i} - u_{i-1}}{2\Delta x}$$

and the third order derivative

$$\frac{-u_{i-2} + 2u_{i-1} - 2u_{i+1} + u_{i+2}}{2\Delta x^3}$$

These equations can be rewritten as matrices. Two boundary conditions are required for the first order derivative and five boundary conditions are required for the third order derivative.

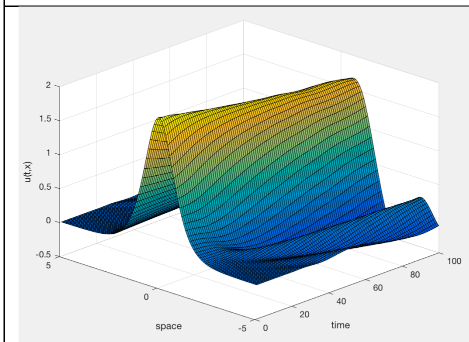
The function is precise up to an order of Δx^2.

Questions

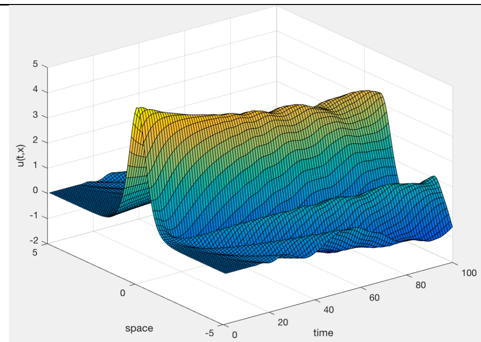
For all functions the following values were used:
L = 100 , T = 100, S = 100, N = 100

$$u_0(x) = f(x;\lambda) \stackrel{\text{def}}{=} \frac{\lambda}{2}\frac{1}{\cosh^2(\sqrt{\lambda}x/2)},$$

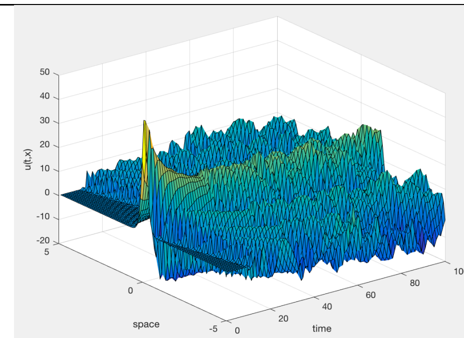
$\lambda = 4$



$\lambda = 10$

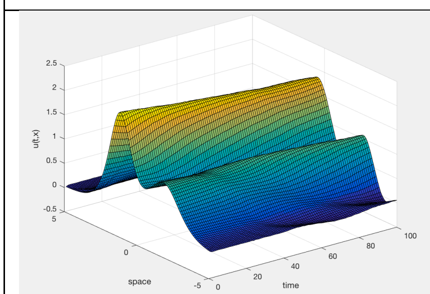


$\lambda = 100$

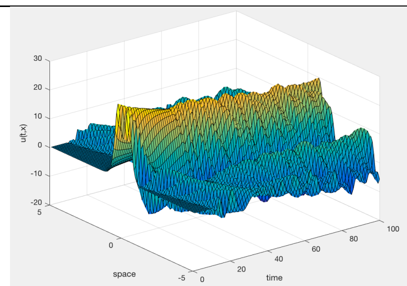


$$u_0(x) = f(x - \tfrac{L}{2}; \lambda_1) + f(x + \tfrac{L}{2}; \lambda_2)$$

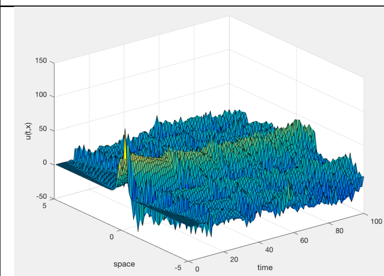
$\lambda_1 = 4,\ \lambda_2 = 2$



$\lambda_1 = 50,\ \lambda_2 = 50$

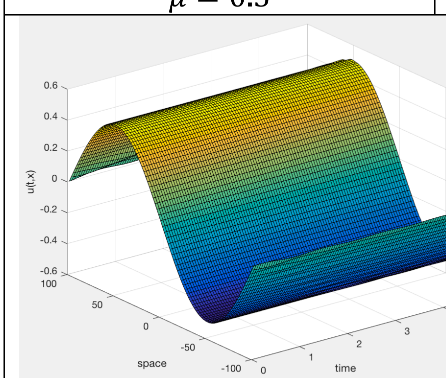


$\lambda_1 = 100,\ \lambda_2 = 200$

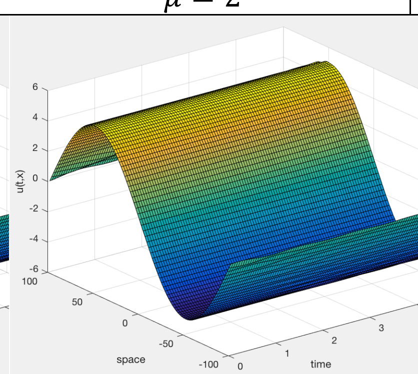


$$u_0(x) = \mu \sin(\pi x / L)$$

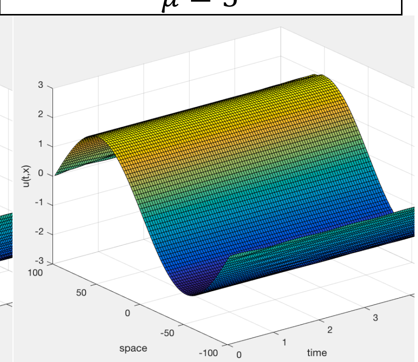
$\mu = 0.5$



$\mu = 2$



$\mu = 5$



Something seems to be going wrong here, either my function is not right or my input is not. This is my input:

```
>> u = @(x) (5 * sin((pi * x)/100))
[Time,dx,u] = KortewegdeVries(u,100,5,100,100,2)
```