PHY 305 Classical Mechanics

Instructor: Sebastian Wüster

Assignments, numerics component:

Assignments may contain a component of numerical work. This is essential in modern science.

For numerics, I will provide almost ready code templates into which you have to just insert some small but essential contribution then run them, and analyse with tools I provide.

(1) Codes will be provided as a matlab script. Matlab allows a large variety of computation, programming and visualisation of which we shall use a very small introductory set. You can learn most from the template I give you. See back page for some essential starter info.

(2) Now at beginning of the semester, please install the campus license matlab from CC at https://www.iiserb.ac.in/cc/download. To run it you shall need a network connection and VPN access to the campus net (see the same download page). Alternatively, you can try using octave (https://www.gnu.org/software/octave/) which is a free version of matlab. Matlab scripts will work with no or minimal changes in octave.

(3) Please also get access to a text editor that can do “matlab syntax highlighting” (google that), which means it will correctly color matlab keywords, comments etc..

(4) To run a matlabscript (program) SCRIPT.m, you navigate within matlab to the directory DIR containing the script (using cd DIR) and then type SCRIPT. It will execute the code and write some output in form of a .mat file. You then load that output as instructed in the script. You now have all the data in the matlab “workspace”. You can output variable A by typing “A” or plot it using e.g. “plot(A)”, however I shall always provide template plotting scripts as well. *(advanced note: When you run a script that does not define a top-level function, there is no need to save and load the output, it will be directly in your workspace).*

(5) For your hand-in of this part, please include a printout of the part of code modified by you (ONLY) and the plots I had asked you to make.

(6) Like with other pieces of assignment the policy will be full marks for **a good attempt**. However I invest a lot of time preparing the numerics part and would hence strongly encourage you to do it well.

MATLAB in 5 min:

- The command:

```
position = 0.5;
```

defines a variable called “position” and gives it the value 0.5. When a command is followed by ;, it does not generate output. If you forget the ”;”, matlab will print the result of the assignment onto the screen.

- Arithmetic operations can be done as expected:
  `A=B+C;  A=B-C;  A=B*C;  A=B/C;  A=B^c` (B to the power c).

- You can define a vector

```
vec = [0.5   0.3  0.2];
```

and look at a component, e.g. `vec(2)` gives 0.3

- or a matrix

```
mat = [1 3;
       5 7];
```

Access matrix element, e.g. `mat(2,2)` gives 6. Or the first column “`mat(:,1)`” gives [1; 5] (i.e. “:” means ALL of that row/column)

- All arithmetic operation have a component-wise version `(.+   .-   .*   ./   .^)` that works on vectors or matrices, e.g. if `v= [v1 v2 v3];`   `w= [w1  w2  w3];` then

```
v.*w = [v1*w1    v2*w2    v3*w3]
```
  etc.

- Define a string with quotation marks `‘name_of_sth’`

- careful: `mat1*mat2` calculates the actual matrix multiplication of mat1 and mat2, while `mat1.*mat2` just does componentwise multiplication

- The most basic command to plot a vector is plot(time,function), where e.g. time is a discrete set of time points `time=[0 0.2 0.4 0.6 0.8 1.0 1.2 1.4….];` and *function=[0 0.2^2 0.4^2 0.6^2 0.8^2 1.0^2 1.2^2 1.4^2….]* would be the function f(t)=t^2.

-Text preceded by “%” is a comment. This does not have to be in matlab form, you write there whatever you like so that you can understand the code better.