PHYS3071/7073 - Due Friday 2nd May, 2014 by 11am

Assignment 7 —- Kepler equation

Introduction: The Kepler equation

According to Newton's law of gravity, the position of a body moving in the $x - y$ plane around another fixed body is given by

$$x(t) = r_1 \cos E(t) \tag{1}$$
$$y(t) = r_2 \sin E(t) \ . \tag{2}$$

Here r_1 and r_2 are the semi-major and semi-minor axis of the orbit. They are related to each other through the eccentricity ϵ, which, for elliptic orbits, is given by $\epsilon = \sqrt{1 - \frac{r_2^2}{r_1^2}}$. This equation is valid for $\epsilon < 1$ with the special case $\epsilon = 0$ corresponding to circular orbits. $E(t)$ is the so-called eccentric anomaly. Kepler's equation relates E to the so-called mean anomaly M as

$$M = E - \epsilon \sin E \ . \tag{3}$$

M is given by $M(t) = 2\pi t / T_{orb}$ where t is the time since the last apocenter passage and T_{orb} the orbital time of the body. For known t and T_{orb} it is straightforward to calculate M, however in order to determine the position of the body one needs to solve eq. 3, which cannot be solved analytically.

Problem 1/4: Write down the derivative of the Kepler equation with respect to E and convince yourself that the derivative is monotonically increasing as a function of E for any eccentricity and any value of E. How many roots of this equation do therefore exist for a given M between $0 < M < 4\pi$? If one wants to use the bisection method to solve for E, what would be safe starting conditions for the interval limits a and b in order to make sure that one always finds the solution? Write your answers to these two questions into a file called `as07-studentname-studentid-answers.txt`.

Problem 2/4: Write a program that takes as input the eccentricity of an orbit ϵ and an accuracy parameter Δ and then solves eq. 3 using the bisection method. If an eccentricity $\epsilon < 0$ or $\epsilon \geq 1$ is entered, your program should stop with an error message. If a valid ϵ is entered, your program should solve for 500 values of M equally spaced between $0 < M < 4\pi$. Implement the bisection method as a function in your program that accepts Δ, ϵ and M as input and returns E. Your program should find the roots with an accuracy $|a - b|/2 < \Delta$. For each value of M, your program should print out a line containing t, the corresponding M and E, and x and y calculated according to eqs. 1 and 2 (assume $r_1 = T_{orb} = 1$ for all cases), the distance r between both bodies and the number of iterations necessary to find the root.

Hint: Note that the fixed body is not located at the origin but at a point $(x/y) = (\epsilon/0)$.

Problem 3/4: Run the program for $\epsilon = 0.0, 0.5, 0.9, 0.999$ and put the output into different files. Produce two plots, one showing x vs. y for the different cases and another showing the distance r of the two bodies as a function of time t.

Problem 4/4 (Advanced): Modify your program such that the user can choose whether the bisection method or the Newton-Raphson method should be used to solve Kepler's equation. In case of the Newton-Raphson method, a good starting value for the root is vital as otherwise the program might not converge. Derive a method/formula for the starting values and add a short description of how you found this formula to the file `as07-studentname-studentid-answers.txt`. Stop the iteration for the Newton-Raphson method if the distance between two consecutive estimates for the root is smaller than the accuracy parameter Δ specified by the user. If your program can't find convergence after 30 steps stop with an error message.

Submission

The complete set of files that you should have if you have answered all problems is:

- Your C programs in a file named `as07-studentname-studentid.c`. Your code should have the following information near the top of the program: `COURSENUMBER AS07 SURNAME STUDENTID`
- A file `as07-studentname-studentid-answers.txt`. with the same information at the top.
- Two plots `as07-studentname-studentid-plot1.ps` showing x vs. y and `as07-studentname-studentid-plot2.ps` showing r vs. t for the different cases. The plots should have your name and student ID in the title and should be properly labeled.

To submit your assignment, create a new directory `as07` and upload the above files to your directory, eg. s1234567/as07/. Make sure all files are named correctly. **This information is important for us when marking the assignments and we will have to deduct marks if it is missing or a file has the wrong name.**

PHYS3071/7073 - Due Friday 2nd May, 2014 by 11am

Grading Sheet – Assignment 7: Kepler's equation

A: /30% **Function**: Does the program compile, run and produce the correct output?

B: /10% **Readability**: Is the program easy to read and comprehend? Is it well-commented? If the code is sufficiently complex, has it been broken up into manageable subroutines, each of which is well-documented?

C: /10% **Usability**: Is the output formatting easy to understand?

D: /10% **Efficiency**: Does the program run efficiently? Is the coding clunky or unnecessarily complicated?

E: /10% **Presentation (Plots)**: Do the plots clearly convey the results? Does each plot have an appropriate title? Are the axes and the plot items clearly labeled?

F: /10% **Analysis**: Were correct answers given to the questions asked in the assignment, and was the process used to obtain them reasonable and clearly explained?

G: /20% **Advanced part**: Grading for the advanced part follows the same rules as outlined under points A to D. Full marks will only be given to programs which can solve the Kepler equation using the Newton-Raphson method for all possible values of ϵ and M.

Total Points: $\boxed{\quad /100}$