NST 2015-16: Part IA Scientific Computing

Practical Session Three (Lectures 5 & 6)
Programming and 3D plotting

November 2015

1 Session number 3

In this case you will start working directly on the assessed bit. It is thus less directed. You can ask for help if you are blocked, and you can discuss with colleagues. You can also make use of the course book and notes, and consult on line.

Make sure, however that the submission is personal, and distinct, different from the submission of others. There are non-prescriptive tasks so that we expect different entries from the different candidates.

The submission report (pdf file, see details in the appendix below) should be composed of the figures requested in this script of high quality standards, with a short text responding to any queries, **and a clear, concise caption describing each figure**.

Please, **do not include the scripts for Section 2.1** in the submission. There is only need to include the scripts related to the last part (Section 2.2).

2 Assessed Task #3

2.1 Anisotropic oscillation in 3D: Tweaking a given program

Consider a mass (m) experiencing a harmonic restoring force when separating from a given point in space (the origin in this case):

$$F_x = -k_x x$$
$$F_y = -k_y y$$
$$F_z = -k_z z$$

The fact that k_x, k_y and k_z do not need to be equal makes it anisotropic, meaning that pulling in different directions is not equally hard.

We will follow the dynamics of the mass under such force, with different spring constants and initial conditions, using a numerical integration in time of Newton's dynamics, by means of the Verlet algorithm as exemplified in the course book for planetary motion.

2.1.1 Get started

Download the program script, `harmaniso.m` and the function it calls, `harmforce3D`. You find them both in Moodle in "Resources for Practical and Assessed Exercise 3", within "Practical Sessions". Put them in your working directory.

Open Matlab and run `harmaniso`, you should get a 3D plot, with a trajectory plotted in 3D space. Ask for help if you cannot download or run it successfully.

Open the main script (`harmaniso`) with the editor and see the script. It is very similar to the one presented in the lectures for the planetary motion, except for the force (in the `harmforce3D.m` file) which is now harmonic.

2.1.2 First task: Isotropic case

Modify the script so that the force becomes isotropic ($k_x = k_y = k_z$), and run the programme several times with different initial conditions. What kind of shape do you get? Generate one figure for a suitable example and give your answer. It does not need a mathematical description.

Remember to present this figure and all the following ones with the standards of quality demonstrated in the lectures.

2.1.3 Second task: 2D anisotropic

Now change it to different spring constants, and give it initial conditions such that $z = v_z = 0$. Explore different situations by varying the spring constants k_x and k_y and the initial conditions. Generate one figure for a suitable example. Change the 3D point of view towards a clearest display of the trajectory.

2.1.4 Third task: Full 3D anisotropic

Do as before but now for non-coplanar initial conditions ($z \neq 0$ and/or $v_z \neq 0$ in this case).

2.1.5 Fourth task: Quality of the numerics

Going back to the isotropic case, and on a convenient case (an instance of boundary conditions and point of view in which the trajectory is well seen), see what happens when you increase the size of the time step, `dt`. If you keep the same number of time steps, the simulation time trivially extends (same number of steps for larger steps), but what is the effect on the shape of the trajectory, as compared to the converged (correct) one? Illustrate this effect by plotting a clear figure showing it, and briefly explain it in the caption.

2.2 Plotting orbitals

In suitable units, the functions

$$p_x = x\, e^{-\sqrt{x^2+y^2+z^2}} \tag{1}$$

and

$$d_{xy} = x y\, e^{-\sqrt{x^2+y^2+z^2}} \tag{2}$$

give the spatial shape of the p_x and d_{xy} orbitals for electrons in atoms. Considering them just on the xy plane (functions of x and y only, setting $z = 0$), produce the following plots:

- Plot both orbitals in a 3D representation, $f(x, y)$, plotting f on the vertical axis, using either the `mesh` or the `surf` commands.
- Plot their corresponding contour plots, with the `contour` command.
- Plot the d_{xy} orbital along the $x = y$ diagonal in a conventional 2D plot.
- Obtain the numerical derivative, $[f(x+h) - f(x)]/h$, of the above function (as obtained for $x = y$) using the `diff` command and plot it. (Remember that the `diff` command does not divide by h.)

Produce all the plots in high quality standards (fonts, ticks, labels, captions), choosing data ranges so as to optimise the illustration of the shapes of the orbitals.

If that was too easy, produce a plot with isosurfaces (p 117) of the p_x orbital (one for positive value, one for negative) in 3D space, as a function of x, y, and z. (This isosurface plot is not needed for achieving full marks.)

A Appendix: Assessed Task Notes

A.1 Assignments

There are three assessed assignments, to be submitted following lectures 2, 4 and 6. The assignments typically involve writing a small MATLAB program to manipulate some data in order to generate a plot or chart and to give a specific number.

You will be expected to submit two files. The first is a report that should contain a copy of the visualisation, any other piece of information you have been asked to generate within the assignment, and a copy of the program you wrote as an appendix. The second should be the .m program file that you created during the assignment (add more files if there are more .m program files or functions).

A.2 Grading scheme

The primary purpose of the exercise is to generate a visualisation and a small piece of information using a MATLAB script, and all marks reflect this. Each assignment has a maximum mark of 2, to be awarded for a visualisation that matches the quality described in the lectures, a working MATLAB script, and a correct piece of extracted information. A mark of 1 will be given for a partly-completed task. 1.5 will be given for a piece of work that is broadly correct but falls short in some regard, and 0.5 will be given for a poor attempt. 0 will be given for non-submission or submission that indicates little effort and no success. We expect most students will achieve the maximum grade.

Your grade will be returned to you via Moodle.

A.3 Requesting help

The purpose of the assessed assignments is to encourage you to put the required work into the course (and not many months later) so that you gain experience in the use of MATLAB and learn good practices in scientific computing. The goal is not assessment per se, but the successful accomplishment of the tasks. For this reason we encourage seeking help if you are stuck, and to this end we run practical sessions to facilitate discussion. We do not encourage wholesale exchange of program scripts of course, because if someone else effectively does the assignment for you, you will have lost a learning opportunity, and this won't be helpful in the long run.

A.4 Generating the report and program file

You are asked to submit a short report for each assignment, which typically should be around one or two pages maximum length. We would like to discourage you from writing extended prose because the marking scheme doesn't have the scope to reward prose – all marks are only to be assigned for the quality of the visualisation, the plausibility of the Matlab script, and the correct generation of the required information.

The report should be in PDF format only; other formats such as Microsoft Word will not be acceptable. Both Microsoft Office 2010 and OpenOffice/LibreOffice can be used to generate PDF files directly, or you can install a PDF generating package such as CutePDF or Acrobat Distiller (which is also provided on every Desktop Services / PWF computer). On a Mac, you can print to a PDF file through the Print menu. If you are using LaTeX instead, by using pdfLaTeX you will automatically generate the PDF file.

Your MATLAB program script (a .m file) will be run by the person marking the work, to ensure that it gives the result and graph reproduced in the report.

A.5 Submission of work

You need to submit your work through the course Moodle site. Go down to the "Assignments" section and you will see links to the various Assessed Tasks as they are released. The important information against any task is the "Due" date.

To submit a task report, click on the "Submit" link. This will bring up a page with more information, from which you can add your files. Find and select the required files and upload them to the web-page. You need to do this for both the report and the file containing the program script.

When you are sure you have uploaded all the relevant documents, press the “Submit" button on the assignment page. You will them be asked to confirm your submission - until you do so, you will not have actually submitted your work. It is essential to check your work has been submitted correctly.

Moodle will not automatically submit your uploaded work, because it wants to give you chance to replace your submission if you need to. **On pressing the “Submit’ button you will be shown a confirmation page and you will be sent a confirmation email. If you do not receive this email, it is likely that you have not actually submitted your work.** You can also use Moodle to check the status of your assignments, and this should also tell you whether you have properly submitted your work.

Remember not to submit your assignment work until you have uploaded both the report and the program script file.

A.6 Submission Checklist

For each assessment, you are required to submit, through Moodle before the due date (given in Moodle in the Assignments section):

(a) A very brief report (a single document in PDF format) which should contain:

1. Your name, college, and CRSid.
2. An appropriate title.
3. A copy of the figure(s) you have generated.
4. Appropriate figure caption(s).
5. The answers to any questions asked in the assignment.
6. An appendix containing your MATLAB code (cut and pasted).

(b) A copy of the .m program script used (e.g. to plot figures, perform analysis, etc.)

A.7 Deadlines

The deadlines for submission of each assessment are:

Assignment 1: 23:55 on Tuesday 3th November 2015.

Assignment 2: 23:55 on Tuesday 17th November 2015.

Assignment 3: 23:55 on Sunday 17th January 2016.