Lab #2

Computational Physics I (**Phys381**)
R. Ouyed
Due Jan. 30, 2014 (at the end of class)

[Total: 100 marks including 20 marks for the report]

- The latexed report is worth **20%** of the total mark. Only well documented and neatly presented reports will be worth that much! It is not sufficient to give only numerical results and show plots, you should also discuss your results. Include complete figure captions, introduction and conclusion sections.
- Your report must be in a *two-column format*.
- Your report should include your *Fortran code* and *Gnuplot scripts* in an Appendix using the *verbatim* command.
- If applicable, animations should be shown to the teacher and/or TA before handing in the lab report. The animation should be well documented and contains necessary information (student name, assignment number, run time etc ...). Basically, the information should be included in each frame before they are put together.
- You must name your report using the names (last names only): *student1-student2-phys381-lab#.pdf*.
- **Procedure for Handing in your lab report (see instructions on phys381 wbsite)**:

  1) Set permission to your PDF report as 644. It means:
  chmod 644 *student1-student2-phys381-lab#.pdf*
  2) cp -a *student1-student2-phys381-lab#.pdf* **/home/ambrish/phys381/labs/lab#**
  3) Copy a second time to ensure that your exam copied correctly. If you are prompted as to whether or not you would like to replace the existing file, then you report has been successfully submitted.

- **You must check with your TAs (Ambrish or Zach) that your report was received and is readable BEFORE you leave the lab.**

1 Fractals

The purpose of this laboratory is to introduce you to the basics of **Fortran 90/95** and get you more acquainted with **Gnuplot**. This laboratory is based on **section A.5.3** of Appendix A in Ouyed&Dobler.

The study of fractal is related to chaos theory and the fact that inphysical systems (weather is one example) are inherently unpredictable on large time scales because small perturbations to the starting conditions will cause large changes over time.

The most famous of all fractal images is the Mandelbrot set. To generate the Mandelbrot set, we begin by considering a complex number, $c = x + i\ y$ (see appendix A below for complex numbers in Fortran 90/95). We then apply the following algorithm:

- set $z = 0$ to start
- then repeatedly compute $z = z \times z + c = z^2 + c$
- until $|z| > 2$ OR the number of iterations exceeds some threshold
- then output the number of iterations (let us call it $n_{\rm c}$)

(a) [**40 Marks**]: Write a program that performs the algorithm above and show that for:

(i) [**10 Marks**]: $c = 0.3 + 0.3i$ (or $x = 0.3$ and $y = 0.3$) the first 4 iterations give

1st iteration:	$z = 0.30 + 0.30i$	$\|z\| = 0.42$
2nd iteration:	$z = 0.30 + 0.48i$	$\|z\| = 0.57$
3rd iteration:	$z = 0.16 + 0.59i$	$\|z\| = 0.61$
4th iteration:	$z = -0.02 + 0.49i$	$\|z\| = 0.49$

In this case, explain why z will remain bounded even after an infinite number of iterations.

(ii) [**10 Marks**]: For $c = 0.5 + 1.0i$ (or $x = 0.5$ and $y = 1.0$) the first 5 iterations give

1st iteration:	$zz = 0.50 + 1.00i$	$\|z\| = 1.1$
2nd iteration:	$zz = -0.25 + 2.00i$	$\|z\| = 2.0$
3rd iteration:	$zz = -3.44 + 0.00i$	$\|z\| = 3.4$
4th iteration:	$zz = 12.32 + 1.00i$	$\|z\| = 12.4$
5th iteration:	$zz = 151.19 + 25.63i$	$\|z\| = 153.4$

and the size of z explodes toward infinite values. By the 10th iteration it will exceed the floating point range of most computers (*check this on your computer if you have time*).

However, we can stop computing z once its absolute value exceeds 2 because it can be shown that divergence is guaranteed at this point.

(iii) [**20 Marks**]: Repeat this calculation for a series of different values of c (i.e. different x and y) and save into a file containing the following rows: x, y and n_c. **It means for each c you need to find n_c.** [Do not save z!]

Use 1000 as an upper limit for the number of iterations.

(b) [**40 Marks**]: You now have to plot the resulting output (n_c) as a function of the location of c on the complex plane. To do so, follow the structure in the Gnuplot script file *gnuplot-fractal.gp* included in Appendix H in Ouyed&Dobler (section H includes the gnuplot **codes**). You should then edit the file and modify the name of the data file to yours. Run the script and save the output.

N.B.: I suggest you make the fractals as impressive as possible by using color when and where necessary. I also suggest to use the *multiplot* command to show multiple fractals in one figure (be creative!).

A Complex numbers in Fortran 90/95

```
The syntax for a complex number, COMPLEX(real, imag) is

Syntax:   C = CMPLX(X,Y,KIND)  [if KIND=2 the double-precision is used]


Example:

            program test_cmplx
              integer :: i = 42
              real :: x = 3.14
              complex :: z
              z = cmplx(i, x)
              print *, z, cmplx(x)
          end program test_cmplx
```