

# Assignment #2

Computational Physics I (Phys381)

R. Ouyed

Due Sunday, April 6th, 2014 (by 11:59 pm)

(i) Use Latex to prepare your report; Include your codes in appendices (using the *verbatim* command); (ii) When I refer to *lecture notes* it means the Ouyed&Dobler notes; (iii) If applicable, animations should be shown to the teacher and/or TA before handing in the 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.

→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 them. Include complete figure captions, introduction and conclusion sections.
- Your report must be in a *two-column format*.
- Your report should include your *Fortran codes* and *Gnuplot scripts* in an Appendix using the *verbatim* command.

**Make sure your codes compile and run properly. You might be asked to use them during the exams (midterm and/or final).**

- You must name your report using your first and last name: *firstnamelastname.pdf*.
- email your report (pdf file) to:  
rouyed@ucalgary.ca and Cc it to:  
“amraghoo@ucalgary.ca” AND TO “zachary.a.shand@gmail.com”

Assignments are due (i.e. should be emailed) on the date and time stated above.

Late submissions will be penalized.

# 1 Errors in computation

[Total: 20 marks]

## 1.1 Calculation of $\pi$

[10 marks]

The first 16 digits of correct value of  $\pi$  are known to be

$$\pi_{\text{correct}} = 3.141592653589793 \quad (1)$$

Now let us consider the calculation of  $\pi$  using the following algorithm (due to Madhava of Sangamagrama, Indian Mathematician of the 14th century)

$$\pi_{\text{mad}} = \sqrt{12} \sum_{i=0}^{\infty} (-1)^i \frac{1}{(2i+1)3^i} . \quad (2)$$

The error in Madhava's algorithm is defined as  $\text{error} = \pi_{\text{correct}} - \pi_{\text{mad}}$ .

(i) [5 marks] Write a Fortran 90 code that performs Madhava algorithm. Run it for at least 50 iterations and save your data into a file with the following 3 columns:

Number of iterations, error (single precision), error (double precision)

(ii) [2.5 marks] Use gnuplot to plot the error versus the number of iterations for single and double precision (both in one panel).

(iii) [2.5 marks] Comment on the accuracy of Madhava algorithm and its uses in single and double precision calculations.

## 1.2 Error propagation in series expansion

[10 marks]

In this section you will estimate how the total error

$$\epsilon_{\text{T}} = \epsilon_{\text{round}} + \epsilon_{\text{trunc.}} , \quad (3)$$

propagate during calculations involving series. In the above  $\epsilon_{\text{round}}$  and  $\epsilon_{\text{trunc.}}$  are the round-off and truncation errors, respectively.

As a rule of thumb the truncation (i.e. chopping) error decreases with the number of terms in the series used,  $N$ , as:

$$\epsilon_{\text{trunc.}} \simeq \frac{\alpha}{N^\beta} . \quad (4)$$

Here  $\alpha$  and  $\beta$  are empirical constants that change for different algorithms.

It is often the case that the round-off errors in each step of the algorithm are independent which allows us to model the accumulation of this error as a random walk with step size equal to the machine precision so that:

$$\epsilon_{\text{round}} \simeq \sqrt{N} \epsilon_{\text{m}} , \quad (5)$$

where  $\epsilon_{\text{m}}$  is the machine precision.

(i) [2.5 marks] Show that the total error is minimal when

$$N^{\beta+\frac{1}{2}} = 2 \frac{\alpha\beta}{\epsilon_{\text{m}}} \quad (6)$$

Explain how to derive the expression above and give the corresponding expression for the minimal total error  $\epsilon_{\text{T}}$ .

(ii) [5 marks] Hereafter we set  $\alpha = 1$ . Using gnuplot, make a 3D figure of the minimum total error versus  $\beta$  (with  $1 \leq \beta \leq 10$ ) and  $\epsilon_{\text{m}}$  (with  $10^{-6} \leq \epsilon_{\text{m}} \leq 10^{-11}$ ).

(iii) [2.5 marks] Discuss your results. In particular discuss regimes where you would expect the round-off error to dominate over the truncation error for a given machine precision.

## 2 Evaluation of functions: *Hydrogen orbital function*

[Total: 10 marks]

The radial part of the 3s atomic orbital for hydrogen has the same form as the expression:

$$R(r) = (2r^2 - 18r + 27) \times \exp^{-\left(\frac{r}{3}\right)} . \quad (7)$$

(i) [2.5 marks] Rewrite the expression above by replacing the exponential part of the function,  $\exp^{-\left(\frac{r}{3}\right)}$ , with the first two terms of its power series expansion.

(ii) [2.5 marks] Compare the 2 expressions by plotting the two functions in the range  $0 \leq r \leq 20$ . Use different line-styles and color for each function (include the gnuplot script you wrote).

(iii) [2.5 marks] How does the fit at small radii and large radii compare to the exact expression. Comment and discuss your findings.

(iii) [2.5 marks] How many terms in the exponential power series expansion are needed to get a good (i.e. an acceptable) fit up to  $r = 5$ ?

### 3 Matrix Algebra

[Total: 50 marks]

#### 3.1 The Fortran code

[15 marks]

(i) Start by reading CAREFULLY **Appendix A** below.

(ii) For problems involving matrices below (and for PHYS 381 in general) you will need to develop (and append to assignment # 2) a Fortran code that performs the following tasks:

→ LU decomposition using the Crout algorithm (as a separate subroutine)  
Takes  $A$  as input and outputs  $L$  and  $U$ .

→ A forward substitution and a backward substitution (as a separate subroutine)  
Takes  $L$ ,  $U$  and  $b$  as input and outputs the solution  $x$  of the  $Ax = b$  system.

→ Matrix Inversion (as a separate subroutine)  
Takes  $A$  as input and outputs its inverse  $A^{-1}$

→ Condition Number calculation (as a separate subroutine)  
This subroutine should output the 3 condition numbers as defined in Appendix A below. Takes  $A$  and  $A^{-1}$  as input and outputs  $\kappa_1(A)$ ,  $\kappa_\infty(A)$ ,  $\kappa_E(A)$ .

→ I encourage you to make use of the routines in section 2.3 in the **Numerical Recipes (NR)** textbook. These are the most robust (and best tested) routines in the market. **SEE LAB # 6 FOR MORE INFO.**

### 3.2 The condition number and ill-conditioned systems

[20 marks]

#### 3.2.1 Case A

Consider the following  $Ax = b$  system:

$$\begin{pmatrix} 1 & 10^4 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10^4 \\ 1 \end{pmatrix} \text{ with the exact solution } x_1 = 0.999800\dots \text{ and } x_2 = 0.999900\dots$$

- [3 marks] Using the 1-norm ( $\kappa_1$ ), find the condition number of:  $\begin{pmatrix} 1 & 10^4 \\ -1 & 2 \end{pmatrix}$
- [2 marks] Use LU-decomposition to find the solutions,  $x_1$  and  $x_2$ . Compare your solution to the one given above.

Now, consider the same system with the first equation divided through by  $10^4$ . This of course does not change the solution:  $x_1 = 0.999800\dots$  and  $x_2 = 0.999900\dots$

- [3 marks] Using the 1-norm ( $\kappa_1$ ), find the condition number of:  $\begin{pmatrix} 10^{-4} & 1 \\ -1 & 2 \end{pmatrix}$
- [2 marks] Use LU-decomposition to find the solutions ( $x_1$  and  $x_2$ ) of the new system.
- [2 marks] What do you conclude as far as handling ill-conditioned cases?

#### 3.2.2 Case B

Now consider the system  $Ax = b$  with:

$$A = \begin{pmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix}$$

The exact solutions of this system is :  $x = (1 \ 1 \ 1 \ 1)^T$ .

- [3 marks] Use LU-decomposition to double-check that the solution is  $x = (x_1 \ x_2 \ x_3 \ x_4)^T = (1 \ 1 \ 1 \ 1)^T$ .
- [3 marks] Let us check what happens if we make slight changes to the coefficients in vector  $b = (b_1 \ b_2 \ b_3 \ b_4)^T$ ? Use LU-decomposition to solve for  $x$  when: (i)  $b = (22.9 \ 32.1 \ 32.9 \ 31.1)^T$ ; (iii)  $b = (22.99 \ 32.01 \ 32.99 \ 31.01)^T$ .
- [2 marks] What do you think is happening? Can you make a firmer conclusion by calculating the condition number ( $\kappa_\infty$ ) of the matrix  $A$ ?

### 3.3 Rocket Physics

[15 marks]

The upward velocity of a rocket is given in Table 2 at 3 different times.

Table 2	
Time (s)	Velocity (m/s)
5	106.8
8	177.2
12	279.2

The velocity data is approximated by a polynomial  $v(t) = a_1t^2 + a_2t + a_3$  for the time interval  $5 \leq t \leq 12$ .

(i) [3 marks] Re-write the problem into a linear system  $[Ax = B]$  where  $x = (a_1, a_2, a_3)$ . Write down what the matrices  $A$  and  $B$  are in this case.

(ii) [3 marks] Write a Fortran 90 code that performs the Gauss elimination method (as an internal subroutine) and the LU decomposition method (as another internal subroutine).

(iii) [3 marks] Using the Gauss Elimination method, solve for the vector  $a_G = (a_1, a_2, a_3)$ .

(iv) [3 marks] Using the LU decomposition method, solve for the vector  $a_{LU} = (a_1, a_2, a_3)$ .

(v) [3 marks] How do the  $a_G$  and  $a_{LU}$  solutions compare to each other? Which one is the more accurate solution and why? Give the corresponding velocities at  $t = 6, 7.5, 9, 15$  seconds.

## A What you MUST know about matrices

### A.1 Matrix norms

The norm of a matrix is a measure of how large its elements are. It is a way of determining the size of a matrix that is not necessarily related to how many rows or columns the matrix has. The following are the most common norms used:

(i) **The 1-norm:**  $Norm(A)_1 = \max_{1 \leq j \leq n} \left( \sum_{i=1}^{i=n} |a_{ij}| \right)$ .

→ We sum the absolute values down each column and then take the biggest answer.

(ii) **The  $\infty$ -norm:**  $Norm(A)_\infty = \max_{1 \leq i \leq n} \left( \sum_{j=1}^{j=n} |a_{ij}| \right)$ .

→ We sum the absolute values along each row and then take the biggest answer.

(ii) **The Euclidean norm:**  $Norm(A)_E = \sqrt{\sum_{i=1}^{i=n} \sum_{j=1}^{j=n} (a_{ij})^2}$ .

→ This is the square root of the sum of all the squares.

### A.2 Condition numbers

The condition number gives us information regarding how well conditioned a problem (and its related matrix) is. *The fact that this number is LARGE (meaning  $\kappa(A) \gg 1$ ) is the indication that the problem involving the matrix  $A$  is an ill conditioned one.*

The condition number of an invertible matrix  $A$  is defined to be

$$\kappa(A) = Norm(A) \times Norm(A^{-1}) , \quad (8)$$

with  $A^{-1}$  being the inverse of  $A$  which you can calculate using LU-decomposition. Note that  $\kappa(A)$  is always bigger than (or equal to) 1.

You MUST use the same norm twice on the right-hand side of the above equation. For example if we use the 1-norm we might write

$$\kappa_1(A) = Norm(A)_1 \times Norm(A^{-1})_1 . \quad (9)$$

Using the  $\infty$ -norm it becomes:  $\kappa_\infty(A) = Norm(A)_\infty \times Norm(A^{-1})_\infty$ .