Teaching Computational Physics to Masters' students in Kolkata

Shibaji Banerjee

Department of Physics

St. Xavier's College

Kolkata

B.Sc Situation
Pre Autonomy Period
How computation made its way into the curriculum

- C.U prescribed that the students complete 10 standard assignments in 3 years time.
- C / Fortran must be used to write the programs
- There would be no practical examination, only records need to be displayed at the time of examination.

Responses to the C.U effect

What happened In SXC

- First two years: The students were introduced to the python programming language and how to attack physics problems using a combination of python and gnuplot.
- Last year: Students had to learn how to do the 10 C.U problems right using Fortran.

What happed mostly everywhere else

- After the initial transience, the 10 problems became standardized quickly and digests became available in the market.

Behind the scene

- In SXC, most of the students quickly lost the Python and the Physics.
- Since only a few classes were available, a book was compiled for the students which they could mostly use as a type assist!

The Type Assist

- Sample Programs

Contents

1 Introduction 3

2 Essentials 4
- 2.1 Assignments 4
- 2.2 Bare Bones Solution to some of the assignments 6
  - 2.2.1 Solution to the quadratic equation problem 6
  - 2.2.2 Solution to the perimeter problem 6

3 Arrays and Sorting 6
- 3.1 Assignments 9

4 Statistics with F90 subroutines 9
- 4.1 Assignments 11

5 Sum of series with F90 functions 11
- 5.1 Assignments 12

6 Solution of simple Algebric Equations 12
- 6.1 Assignments 14
- 6.2 Bare Bones Solution to some of the assignments 14

7 Simple Matrix Operations 15
- 7.1 Assignments 17

8 Summation of infinite series 17
- 8.1 Assignments 18

9 Numerical Integration 18
- 9.1 Assignments 19

10 Linear Least Squares Method 19
- 10.1 Assignments 21

A Language Elements 22

B Errors 22

C Style Guide 23

D FORTRAN-77 Vs. F90 24

Listings

1 Hello World 3
2 Square root of a number 4
3 Quadratic Equation, Complex roots 6
4 Perimeter of Ellipse 6
5 Square root of an array 7
6 Largest number of an array 7
7 Bubble Sort 8
8 Mean Median and Mode of a dataset 9
9 Sum of a G.P 11
10 Root finding, Bisection method 12
11 Root finding. Newton's method 14
12 Matrix Multiplication 15
13 Sum of Infinite Series 17
14 Numerical Integration, Simpson's Rule 18
15 Least Squares Method 19

Post Autonomy Period

- We were free to pick a different structure, but were expected to stay in conformity with the existing policy.
- New course contained elements of visualization, LaTeX, Introduction to the F90 Language, Numerics with F90, spread over 5 semesters.
- Method Used:
  - Basic Instruction
  - Factsheet
  - Worksheet

Plotting Labs

Plotting Lab 1

FactSheet

```
plot f(x)
plot f(x),g(x)
plot [][y1:y2] f(x)

set parametric
plot t,t**2,t,t**3
unset parametric

set polar
plot [-2*pi:2*pi] [-3:3] [-3:3] t*sin(t)
unset polar

show samples
set samples 500
```

1 Semester 1

1. **MM1a.Review of Math Methods:** A very brief review of one-variable calculus highlighting the important application areas, and the successful system of differential modelling. Elementary functions and their special properties. Plotting Graphs of simple functions with their derivative and integral curves. Functions of two variables and their visual representation. An overview of the math methods and their relationship to the whole course.

12 Lectures

ASSIGNMENTS

1. Plot the graph of $3x^2 - 8$ for x between -5 and 5 .
2. Plot the same function, but restrict the y range between -20 .. 40.
3. Plot the graph of $x^3 + 1 - \exp(x)$ over x=-8..8. Select suitable y range to show all the four x intercepts.
4. y = sin(x) over two complete periods.
5. Plot $3x^4 - 6x^2$ over the domain [-10,10] with automatic y scaling. After observing the graph, edit the domain and range so that you can see the x-intercepts clearly. Estimate the x-intercepts with the mouse cursor.
6. Define the functions $g(x) = 5exp(-0.5x)$ and $h(x) = x + 1$ then do the following.
   (a) Plot a graph that shows both functions g(x) and h(x). Experiment with different values for domain and range.
   (b) Estimate the coordinates of the point of intersection of these two graphs by using left mouse-button click. Check the answer by any calculator.
7. Define the function $k(x) = x + 3\sin(2x)$, then do the following:
   (a) Plot the graph of this function on the domain [-1,8] .
   (b) Modify your plot from part (a) to include the horizontal line $y = 4$. Use this new plot to estimate the number and approximate values for x such that $k(x) = 4$.
   (c) What single function could you graph that would give you the same information as in part (b)
8. Plot the parametric curve determined by $x = t^2 - t$ and $y = 2t - t^3$ over the t interval [-2,2].
9. Plot the polar equations $r = 1 + \cos\theta$ and $r = \sin 3\theta$ for $\theta = [-\pi, \pi]$
10. Plot $exp(-x/100)\cos(x)$ with a suitable sampling rate.

The F90 Lang Labs

Contents

1 Welcome 3

2 Language Elements 6

3 Preview 8

4 How to use this book 9
- 4.1 Handling Errors . . . 10
- 4.2 Assignments . . . 10

5 Essentials 10
- 5.1 Assignments . . . 12

6 Arrays and Sorting 13
- 6.1 Assignments . . . 17

7 Modular programming with F90 20
- 7.1 Assignments . . . 25

8 The Craft 26
- 8.1 Assignments . . . 26

A F90 Intrinsics 29

B Style Guide 30

C FORTRAN-77 Vs. F90 31

D Solutions to selected problems 33

The new Listings

Listings

1	Hello World	4
2	Square root of a number	12
3	Square root of an array	14
4	Largest number of an array	15
5	Switching	16
6	Bubble Sort	17
7	Mean Median and Mode of a dataset	20
8	Sum of a G.P	24
9	Generate 10 sets of 2 Random numbers	27
10	Quadratic Equation, Complex roots	33
11	Area of Ellipse	33
12	Matrix Multiplication	34
13	Coin Tosses	36
14	Horce Race	37
15	Gambler's Claim, Part 1	38
16	Gambler's Claim, Part 2	39
17	Area Estimation(Monte Carlo))	40
18	Estimation of PI(Monte Carlo)	41

New F90 Numerics Lab

Contents

1 Introduction 3

2 Ball Games or Simple Dynamical Problems 3

3 Exploring Oscilations 7

4 Nonlinear Oscillation, A Case Study 8

5 Numerical Methods 12

5.1 Numerical Integration 13

5.2 Finding Roots of nonlinear equations 14

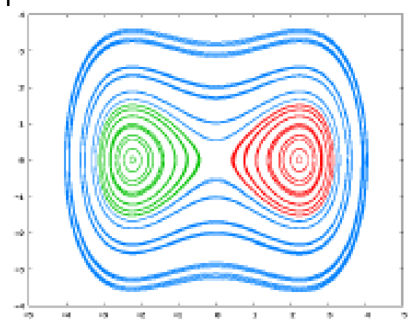
6 F90 Flashback 19

7 Gnuplot Flashback 22

8 Solutions 22

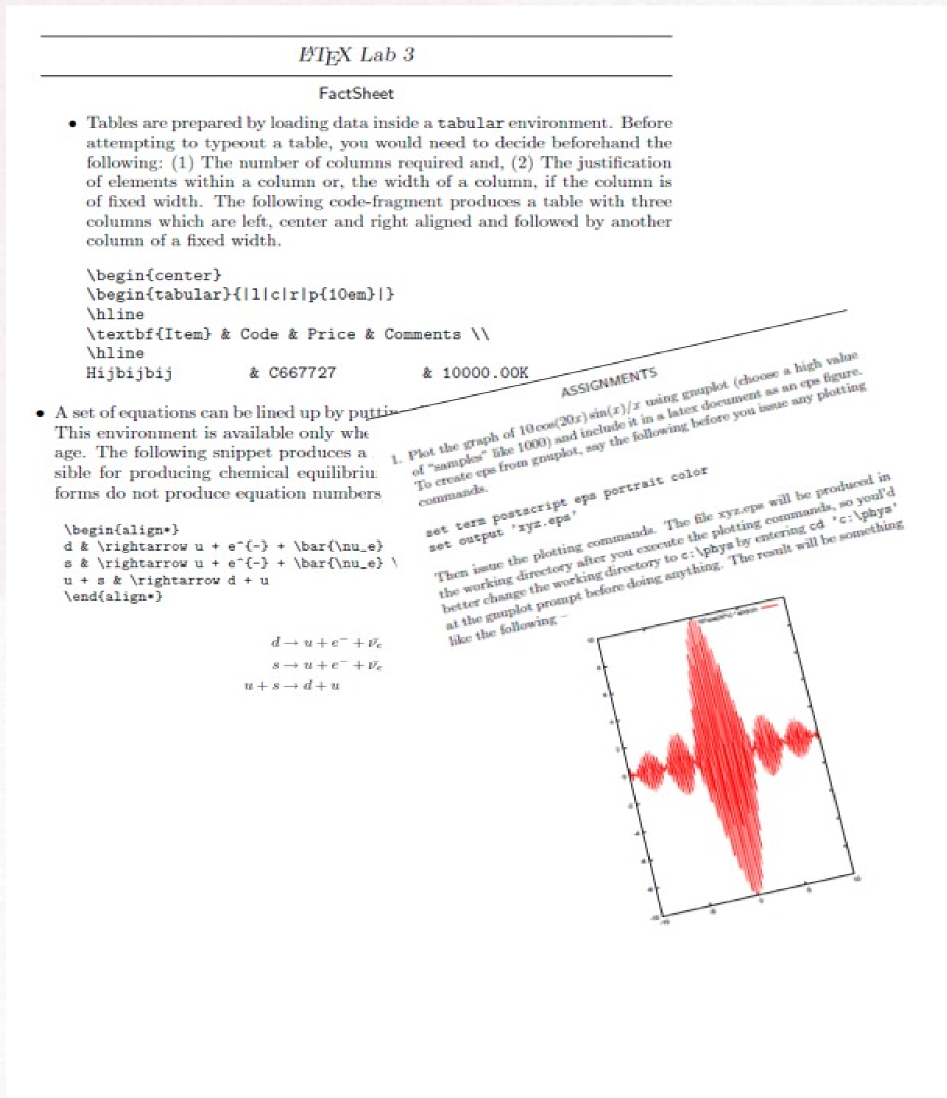
Solution to Ex:15

```
PROGRAM SIMDWELL3
REAL, Dimension(9) :: x0vec
INTEGER :: Nplus,N

Nplus=0
N=9
x0vec=(/0.5,0.8,1.1,1.4,1.7&
&,2.0,2.3,2.6,2.8/)
Call EulerGrind(NPlus,N,x0vec)
NPlus=9
N=9
x0vec =(-1.0)*(/0.5,0.8&
&,1.1,1.4,1.7,2.0,2.3,2.6,2.8/)
Call EulerGrind(NPlus,N,x0vec)
NPlus=18
N=6
x0vec=(/3.2,3.3,3.5,3.6&
&,3.9,4.0,0.,0.,0./)
Call EulerGrind(NPlus,N,x0vec)

Contains
Subroutine EulerGrind(NPlus,N,x0vec)
REAL, Dimension(9), Intent(In) :: x0vec
INTEGER, Intent(In) :: NPlus,N
REAL :: k=1.0,m=1.0,lam=0.2
REAL :: t0=0,v0=0,x0
REAL :: tend=30.
REAL :: Dt=.001 !sec
REAL :: t,x
DO I=1,N
x0=x0vec(I)
t=t0
v=v0
x=x0
Do WHILE (t<tend)
Print*, t,x,v
a = k*x/m - lam*x**3/m
x=x+v*Dt
v=v+a*Dt
t=t+Dt
End DO
Print*, "#Dataset",I+NPlus
Print*,""
Print*,""
END DO
End Subroutine EulerGrind

END PROGRAM SIMDWELL3
```



```
set nokey
plot \
'shmdwell3a.txt' index 0  using 2:3 w l ls 1, \
'shmdwell3a.txt' index 1  using 2:3 w l ls 1, \
...
'shmdwell3a.txt' index 8  using 2:3 w l ls 1, \
'shmdwell3a.txt' index 9  using 2:3 w l ls 2, \
'shmdwell3a.txt' index 10 using 2:3 w l ls 2, \
...
'shmdwell3a.txt' index 17 using 2:3 w l ls 2, \
'shmdwell3a.txt' index 18 using 2:3 w l ls 3, \
'shmdwell3a.txt' index 19 using 2:3 w l ls 3, \
...
'shmdwell3a.txt' index 23 using 2:3 w l ls 3
```

TeX Labs



LaTeX Lab 3

FactSheet

- Tables are prepared by loading data inside a `tabular` environment. Before attempting to typeout a table, you would need to decide beforehand the following: (1) The number of columns required and, (2) The justification of elements within a column or, the width of a column, if the column is of fixed width. The following code-fragment produces a table with three columns which are left, center and right aligned and followed by another column of a fixed width.

```
\begin{center}
\begin{tabular}{|l|c|r|p{10em}|}
\hline
\textbf{Item} & Code & Price & Comments \\
\hline
Hijbijbij       & C667727        & 10000.00K
```

- A set of equations can be lined up by puttin… This environment is available only whe… age. The following snippet produces a… sible for producing chemical equilibriu… forms do not produce equation numbers…

```
\begin{align*}
d & \rightarrow u + e^{-} + \bar{\nu_e}
s & \rightarrow u + e^{-} + \bar{\nu_e} \
u + s & \rightarrow d + u
\end{align*}
```

$$d \rightarrow u + e^{-} + \bar{\nu}_e$$
$$s \rightarrow u + e^{-} + \bar{\nu}_e$$
$$u + s \rightarrow d + u$$

ASSIGNMENTS

1. Plot the graph of $10\cos(20x)\sin(x)/x$ using gnuplot (choose a high value of "samples" like 1000) and include it in a latex document as an eps figure. To create eps from gnuplot, say the following before you issue any plotting commands.

```
set term postscript eps portrait color
set output 'xyz.eps'
```

Then issue the plotting commands. The file xyz.eps will be produced in the working directory after you execute the plotting commands, so you'd better change the working directory to c:\phys by entering cd 'c:\phys' at the gnuplot prompt before doing anything. The result will be something like the following –

Experiences at the P.G Level

- The P.G course is run jointly by SXC and BI(CAPPS) Kolkata.
- The students have to clear a randomized MCQ + Interview before they can get admission.
- The students can apply from anywhere after completing B.Sc
- State of students who choose to study here:
  - seems to be well motivated, but ...
  - more guided by examination reflexes than relying on their rational ability, and
  - lack basic preparation in Physics. Irreversible damage to natural curiosity level ??

Syllabus

2.1.2 Computational Physics

This course will introduce the computational methods used to investigate physical Phenomena.

Numerical Techniques : Finite vs. infinite precision calculations, Coding tools: Languages and Libraries, Handbooks, Numeric Differentiation and Integration, Interpolation and Extrapolation Techniques, Special Functions, Matrices : Inversion, LU decomposition, Tridiagonalization, Eigensystem of a tridiagonal matrix; Linear and non-linear least squares, Monte Carlo Calculations, Finite Difference Solution (RK) of Differential equations, Finite Element Solution to PDEs.

Computer Algebra / Visualization Application of copyrighted and open-source non-proprietary software like Mathematica, Matlab, Maple,Gnuplot/Grace etc for performing rapid calculations, prototyping, visualizations and data analysis.

Applications Applications of computational techniques for the study of a variety of scientific problems like Chaos and Nonlinear Dynamics, Initial and Boundary Value Problems, Simulation of Model Systems, Critical Phenomena, Quantum Mechanical Scattering etc.

Constraints

- Cannot build upon the existing B.Sc framework since many students join from outside.
- Certain amount of repetition is essential
- Feedback taken at the beginning of the course reveals little familiarity / aptitude for computation

Class................................ ROLL...................... Date........................

Name...

Subject...................................... Paper..................... Pass/Honours

Marks will be deducted for not filling this section completely.

Physical Computing
Questionnaire

Hello [Roll: MPH/08/01]

You are attending a course which has a strong emphasis on computation. It is therefore necessary for us to be aware of your initial backgrounds in this field. We have therefore prepared this questionnaire to gather this information and also to indicate where we might be headed. For questions, 7-10, Solve the problems using your favorite computer language. Please hand this questionnaire back to us after an hour. Thank you.

1. Can you handle computers comfortably? (Tick items you can handle, Cross out items you'd feel not so comfortable with)
   1. Copy Files,
   2. Play Media,
   3. Write Text,
   4. Prepare report (with abundant math, tables, graphics and cross references)
   5. Run Applications,
   6. Surf Net
2. Know one or more computer languages (Yes / No) well enough to carry out simple tasks. If yes, tick / mention the language name from the following list.
   1. Logo
   2. Basic
   3. Fortran 77 / 90
   4. C / C++
   5. Java
   6. Other (Please Specify) ____________________
3. Used More than one Operating System
   1. Primary OS: Windows XP, 98.
   2. Secondary OS:
4. Are you familiar with Command Line Interfaces(apart from chat prompts)?
   1. Dos Prompt (?)
   2. Unix Terminals (?)
5. Did you use any applications belonging to the following categories ?
   1. Text / Word Processors: MS WORD, POWERPOINT.
   2. Graphics / Animation: ____________________
   3. Scientific Computation
      1. Plotting data and functions
      2. Fitting data, finding roots of equations, ... (numerical analysis)
      3. Statistical calculations
      4. Symbolic calculations
6. Did you use a scripting language (like Perl or Javascript) ?
   1. Yes, Specify ____________________
   2. No.
7. Ask the user to input his/her name. Print a welcome message in response.
8. Ask user to input his/her weight (in Kg) and height (in feets). Calculate and display the ratio r = (weight in Kg) / (height in m)^2.
9. Ask user to input a number between 1 – 100. Print 'Small', 'Medium' or 'Large' depending on the input.
10. Calculate the sum of first 'n' natural numbers. The number 'n' will be read in at runtime.

Class................................ ROLL 13 .... Date 12/9/08 ....

Name Subhash Bose ....

Subject...................................... Paper..................... Pass/Honours

Marks will be deducted for not filling this section

Physical Computing
Questionnaire

Hello [Roll:]

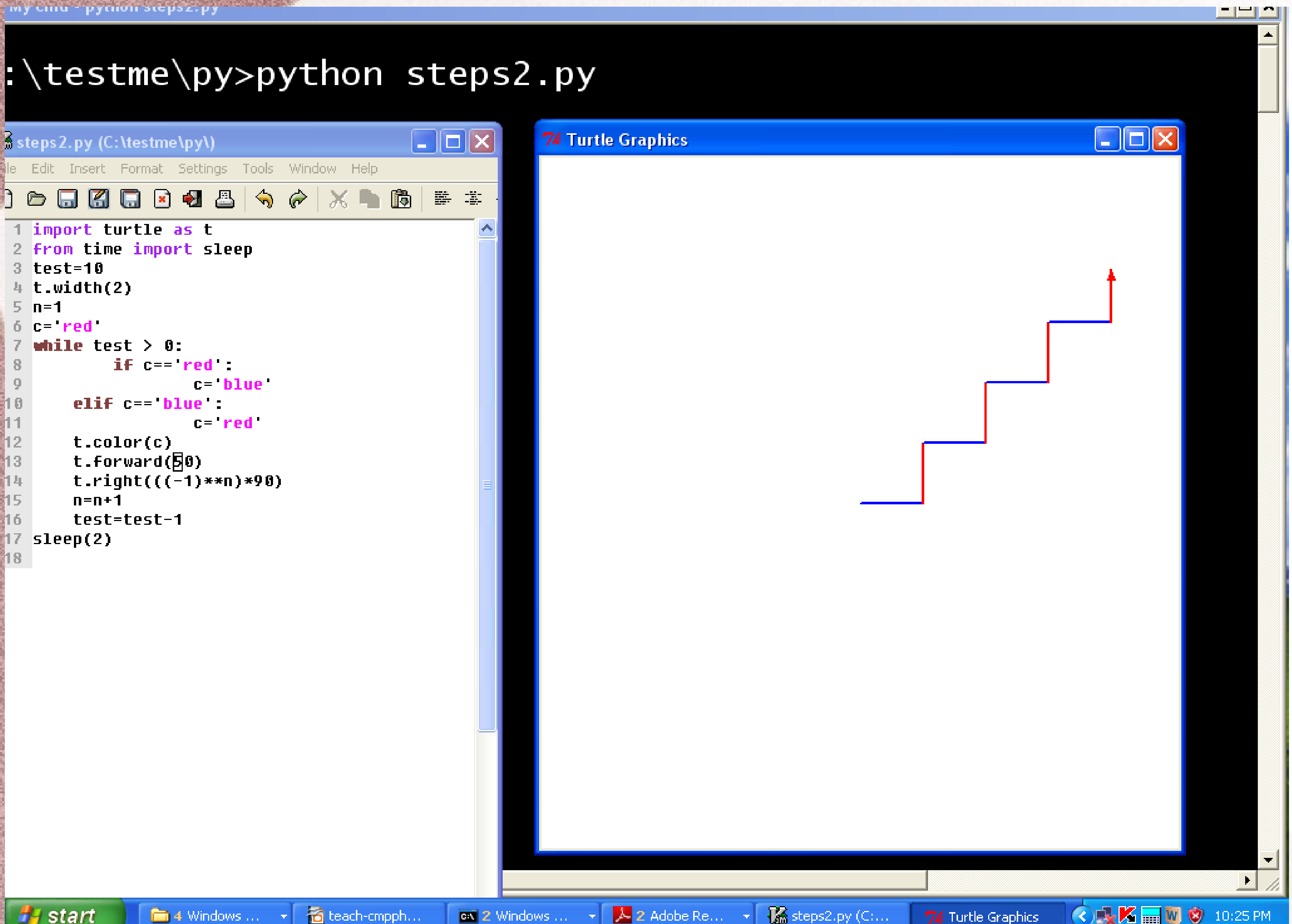
You are attending a course which has a strong emphasis on computation. It is therefore necessary for us to be aware of your initial backgrounds in this field. We have therefore prepared this questionnaire to gather this information and also to indicate where we might be headed. For questions, 7-10, Solve the problems using your favorite computer language. Please hand this questionnaire back to us after an hour. Thank you.

1. Can you handle computers comfortably? (Tick items you can handle, Cross out items you'd feel not so comfortable with)
   1. Copy Files,
   2. Play Media,
   3. Write Text,
   4. Prepare report (with abundant math, tables, graphics and cross references)
   5. Run Applications,
   6. Surf Net
2. Know one or more computer languages (Yes / No) well enough to carry out simple tasks. If yes, tick / mention the language name from the following list.
   1. Logo
   2. Basic
   3. Fortran 77 / 90
   4. C / C++
   5. Java
   6. Other (Please Specify) ____________________
3. Used More than one Operating System
   1. Primary OS: 98, XP (Windows)
   2. Secondary OS:
4. Are you familiar with Command Line Interfaces(apart from chat prompts)?
   1. Dos Prompt (?) Yes
   2. Unix Terminals (?)
5. Did you use any applications belonging to the following categories ?
   1. Text / Word Processors: MS. Word
   2. Graphics / Animation: ____________________
   3. Scientific Computation
      1. Plotting data and functions
      2. Fitting data, finding roots of equations, ... (numerical analysis)
      3. Statistical calculations
      4. Symbolic calculations
6. Did you use a scripting language (like Perl or Javascript) ?
   1. Yes, Specify Javascript, PHP
   2. No.
7. Ask the user to input his/her name. Print a welcome message in response.
8. Ask user to input his/her weight (in Kg) and height (in feets). Calculate and display the ratio r = (weight in Kg) / (height in m)^2.
9. Ask user to input a number between 1 – 100. Print 'Small', 'Medium' or 'Large' depending on the input.

Subsequently

- Students make use of python in nonlinear dynamics.
- Students learn to use python for dynamical problems involving Differential equations and catch up with numerical algorithms later.
- They also use python to talk with the Phoenix Box.
- They learn Matlab to develop awareness of existing toolboxes and rapid development of code.
- They learn to simulate and design analog and digital circuitry.

Say it with pictures!



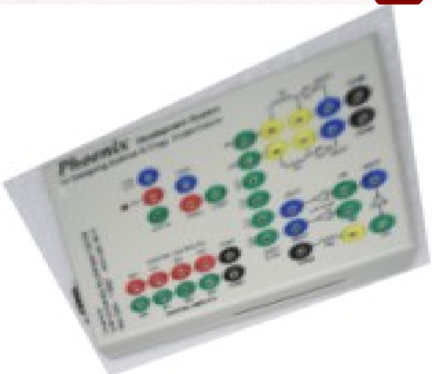
Electronic simulation and elements of Interfacing

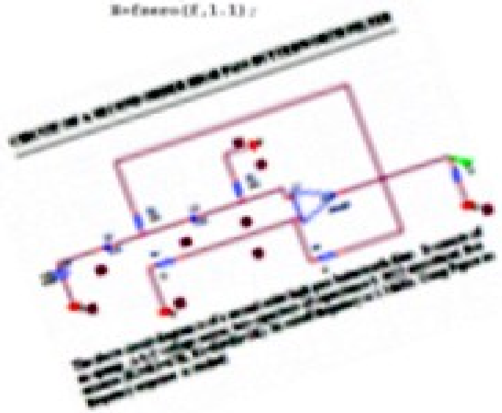
Matlab Session 4
Getting Ahead with Numerics
Finding Roots to nonlinear equations

In many problems we are required to find when a function is zero. In this session we would implement a few common methods for finding that information. We would also learn how to self check the answers using Matlab's `fzero` command.

1. **Initial Estimates:** The graphical capabilities of Gnuplot / Matlab can be used to identify the root (actually the bracketing interval). Create vectors corresponding to sampled x points (use linspace or similar) and the y expression (Use dotted forms of operators to ensure that your expression can work on a point-set. You can also use the `vectorize` function to convert an scalar function to a point-set function, see `help vectorize`). Plot the function, turn on the grids and zoom on the zero crossings of the function to locate the roots approximately (use the magnifier icons on the toolbar). Either turn the *data cursor* on (Tools menu) or use the `ginput` function to locate the roots better than eyeball estimates can do.
   1. Plot the function $x\sin(x)-\sqrt{(x)}$
   2. Use the methods outlined above to estimate the roots of this function.
2. **Self Check for Polynomials:** Use the roots command to find all roots of [illegible] root command takes in the polynomial coefficients taken as a row vector [illegible] power first) and returns the roots in a column vector. No guess values are [illegible]
   1. Use the roots command to find all roots of the polynomial [illegible]
3. **Self Check for any function.** Use the `fzero(fcn, guess)` function [illegible] nearest to the guess value The function fcn is the name of a user defined [illegible] file) or an *anonymous* function.
   Example: Consider the function: $x\sin(x)-\sqrt{(x)}$. Roots lie near x = 1.2, [illegible] fzero(myfcn,1.2) returns 1.1748 as the root. Instead of creating myfcn as [illegible] use it directly in fzero as an anonymous function, viz.
   `fzero(@(x) x.*sin(x)-sqrt(x),1.1)`. Anonymous functions [illegible] name also `f=@(x) x.*sin(x)-sqrt(x); fzero(f,1.1)` will [illegible] root. You can also try an `inline` function approach-

```
x=linspace(0,10);
f=inline('x.*sin(x)-sqrt(x)')
y=f(x)
plot(x,y)
X=fzero(f,1.1);
```





Date:

Interfacing Primer Lab - II

1. In the Phoenix box, the ADC is in charge of reading analog signals and converting it to digital signals which can be processed by the computer. In this experiment, we would charge up a capacitor and let it discharge through a resistor. The Discharge Data would be acquired by the ADC and saved to a file for further analysis. The plan, roughly is in the following order:
   (a) Hookup a capacitor so that it is charged by setting the Dn pin to high, where Dn is the pin connected to the capacitor. Connect the ADC to this point also, so that it can measure the voltage when the capacitor charges.
   (b) Connect the capacitor to +5V through the resistor and wait (sleep!) till it is charged.
   (c) Set the Dn Pin to low and ask the ADC to acquire the data(use a `read_block` statement, take about 200 unipolar samples at 50μs intervals to start with). A question arises right away as to one can synchronize these two events (start the discharge & take the data)? Phoenix allows you to alert the ADC at any point in the code by using a statement of the form `p.enable_set_low(<n>)`. This empowers the ADC to turn pin-n down whenever it begins to acuire data[1].

   Sketch a circuit and construct it on the breadboard. Use a 1K discharge resistor. Note that you'd have to fiddle with the sampling parameters to gather enough data. The value of the capacitor can be determined by fitting the discharge curve with gnuplot which can boast of having a nonlinear fitting routine. First plot the data to determine the range of time over which the discharge is significant. The fitting routine will be glad to accept a working input range. Also determine the (tentative) value of V_0 from the graph and supply it to the routine as well. Data fitting is as much an art as an exact science, so help it as much as you can. Using the measured value of the resistance helps naturally to determine the exact value of the capacitance. *Timed switching and fast measurement of small time ranges are the key features of this experiment.* Note that this experiment can serve as a basis of creating a microcontroller based capacitance measuring equipment. Repeat with another (different) capacitor.

2. The aim of this experiment is to digitize audio signals and calculate the frequency of the signal from the acquired data. Audio signals will be generated by the Piezo Buzzer (flat, short cylindrical piece with a red and a black wire) and taken up by a Condensor Microphone (small cylindrical tablet with three wires coming out of it). Connect the buzzer between the PWG(RED END) and the Ground(BLACK END). The buzzer will emit sound if you set the frequency from python (test it). Connect the microphone between the +5V (RED END) and the Ground (BLACK END). The third end, i.e the one that is connected to the capacitor (to block DC) goes to the ADC channel 0 through two inverting amplifiers in series (Set a net gain ~ 500 to 1000) and the level shifter. Power on the circuit. Run cro.py and set the frequency of the PWG to 3.7KHz. Acquire the data and save it. Plot the data and try to get an independent estimate of the signal frequency from the plot (the time axis is in μs; you can use the gnuplot ruler).

3. For periodic signals, it doesn't matter when you begin to start acquiring the signal. However, if we want to detect a pulse, the acquiring process should know in advance that it should start to take the data when the event occurs. In

Conclusions

- Students should be motivated early to get acquainted with os basics like file-systems and file-operations.
- Python serves very well as a first computing language.
- Pictorial programming can expose the basics of coding very rapidly
- A combination of fact and work sheets works well in all levels, even in the face of resource / time constraints.
- Teaching how to explore phenomena through odes and pdes should really be taught at first, with numerical techniques filling in wherever appropriate. It can also come later as a separate course.
- The students should be exposed to existing code libraries and tools which allow for rapid prototyping.

Acknowledgements

- Dr. Ananda Dasgupta
- Dr. Suparna Roychowdhury
- Dr. Arunabha Adhikari
- Dr. Dibakar Dutta
- Ms. Saswati Chakraborty

The End