PY502, Computational Physics

Instructor: Prof. Anders Sandvik

Office: SCI 316, phone: 353-3843, e-mail: sandvik@bu.edu

Lectures: Tuesday/Thursday 11-12:30 in CAS 327
• Programming tutorials, Fridays Sep. 6, 13, 1-2 PM
• Occasional make-up classes Friday 1-2:30 PM

CAS 327 reserved for PY502 students’ use Fridays 1-3 PM

Homework: ≈7 assignments
- Adam Iaizzi (iaizzi@bu.edu) is the grader

Grade: 100% homeworks
Course web site
http://physics.bu.edu/py502
- Lecture notes
- Lecture presentations
- Example program
- Homework assignments and solutions
- Messages (“Course News”)

Submitting homework

Use e-mail to submit programs: py502@buphy.bu.edu
- Include the program(s) as attachment(s)!
- Name your programs: hwx_lastname.f90

Turn in write-up (hardcopy) report to instructor or grader

Some discussion/collaboration on homework problems is allowed, but each student has to turn in her/his independently written programs and reports.
Computers and programming language

- Student accounts on the CAS 327 workstations
- Fortran 90 will be used in lectures, homework, exam
- Introduction to Fortran 90 will be given (3 lectures)
- Extensive background in programming not needed
- Some Unix/Linux knowledge assumed (e.g., text editing)
- Come to office hours if you need help!
Lecture notes

• Will be posted on the web site ahead of the lectures
• No additional required text

Recommended reading

• Computational Physics, by J. M. Thijssen
• Computational Physics, by N. J. Giordano and H. Nakanishi
• Fortran 90/95 Explained, by M. Metcalf and J. Reid
• Fortran 90/95 For Scientists and Engineers, by S. Chapman
• Numerical Recipes, by W. H. Press et al.
  (free on-line with codes in many languages: http://www.nr.com/)
What is computational physics?

- “Scientific computing” in physics

- Studies of models of physical systems using computers
  - Numerical solutions of equations that cannot be done analytically
  - Direct studies of models to “simulate” a system

- Most subfields of physics use some computations, e.g.,
  - Dynamics of solar systems, galaxies, etc
  - Studies of mechanical models of earthquakes
  - Fluid dynamics; turbulence
  - Molecular dynamics of gases, fluids
  - Electrostatics and dynamics (Maxwell’s equations)
  - Electronic structure of materials
  - Statistical mechanics of polymers, magnetic systems, etc.
  - Lattice gauge theory (numerical QCD)

- Some times considered third “branch” of physics
  - Experimental, theoretical, computational

- Most physicists need to do some computational work
Topics covered in PY502

- The Fortran 90/95 programming language
- Numerical integration
- Numerical solution of differential equations
  - classical and quantum mechanics problems
- Monte Carlo simulations (statistical mechanics)
- Basic methods for quantum many-body (lattice) systems

Goals

- Learn the basics of the above techniques
- Gain proficiency in scientific computing in general
Teaser: The last topic of the course will combine several of the previous methods we have learned to study: Quantum Annealing (a paradigm for quantum computing). You will learn what is going on (supposedly…) in machines made by D-wave, Google,….
Work station accounts

- All registered students will be given accounts
  - accounts should be set up if you are registered
  - use your BU Kerberos ID and password
- guest account: id=student, password=… (ask instructor)
- The workstations can be booted in Windows or Linux
  - if Windows is running, restart and select BU Linux

Download material from course web site

- Open web browser
- Go to http://physics.bu.edu/py502
- Download Fortran 90 lecture notes (pdf file)
- Open Linux terminal (“console”)
- Create a directory (mkdir f90examples) for programs
- Click the “programs” link next to the lecture title
- Download the tar-archive; save it to f90examples
- Go to f90examples, do “tar -xvf f90examples.tar”
The Fortran 90 programming language

- Fortran has evolved since the early days of computing
- Fortran 90/95 is a modern programming language
- Many useful features for scientific (numerical) computing
- Widely used language in computational science
  - also widely used in finance, engineering, etc.
- Many “canned” subroutines available (often Fortran 77)
- Relatively easy to learn

Fortran 90 compilers

- Many commercial compilers available; often expensive
  - f90 available on buphy (Physics Dept. server)
- gfortran; free open source Fortran 90/95/2003/2008 compiler
  - available on CAS workstations
  - part of gcc (gnu compiler collection)
  - installed on Physics Dept server buphy (and buphy0)
- g95; other open source product
Fortran 90 language tutorial

- Introduction to basic elements needed to get started

- Simple examples used to illustrate concepts

- Example programs also available on the web site

- For more complete language description, see, e.g.,
  - *Fortran 90/95 explained*, by M. Metcalf and J. Reid
  - *Fortran 90/95 for Scientists and Engineers*, by S. Chapman
  - Links on course web site (On-line Fortran resources)

Discussion and practice at (some) Friday tutorials
To create a Fortran 90 program:

- Write program text to a file, using, e.g., Emacs

```fortran
[program program-name]
  program statements
  ...
end [program program-name]
```

- Compile & link using Fortran 90 compiler
  - creates “object” (.o) files
  - object files linked to form executable (.out or .x) file
Compilation/linking (using gfortran)

> gfortran program.f90
    (gives executable program a.out)

> gfortran program.f90 -o program.x
    (gives executable named program.x)

> gfortran -O program.f90
    (turns on code optimization)

> gfortran -O program1.f90 program2.f90
    (program written in more than one file)

> gfortran -O program1.f90 program2.o
    (unit program2 previously compiled)
Variables and declarations

Intrinsic variable types
- real (floating-point)
- integer
- complex
- logical (boolean)
- character, character string (“text”)
- arrays of all of these (up to 7-dimensional)

- A declaration is used to state the type of a variable
- Without declaration, real assumed, except for variables with names starting with i,...,n, which are integer
- Declarations forced with implicit none statement
  - always use this; eliminates 99% of programming errors!
- Fortran 90 is case-insensitive

[we discussed how numbers are represented in the computer]