Scientific Programming and Numerical Computation

Course webpage: homepage.ntu.edu.tw/~wttsai/fortran/index.html
1. Introduction

- What is this course for?
- Shall I take this course?
- How is this course taught?
Two fundamental methodologies for scientific or engineering studies

By observing or measuring in the field or in the laboratory ...

By analyzing or computing on papers or using computer ...
Procedure of scientific or engineering study using numerical simulation

(what I was taught, and how we are doing)

1. Identify the physical processes to study
2. Pose a model problem and formulate it using conservation laws
3. Solve the equations numerically or theoretically
4. Check if the equations are solved correctly?
5. Can the solutions explain the physics, or compare correctly with the measurements?
6. Use the “numerical model” to explore mechanisms, make prediction, optimize design, etc.
Assume...

- you know how to use a computer to compute;
- but have not done any serious numerical computations,
- and only know some very basic numerics.

This course will teach you...

- FORTRAN programming language to compute,
- good habits to write structural Fortran programs,
- to write test driver to test the Fortran program written by you or others,
- to use well developed and optimized libraries, such as LAPACK, FFTW,
- to do parallel computing in multi-process computer using OpenMP
- to use these skills to solve partial differential equations governing simple physical systems, such as Poisson equation, wave equation, heat equation.
... and using these basic blocks plus some extra hard work, eventually you are able to develop your own numerical program to simulate the more realistic physical system, such as ...
This course will *not* teach ... 

- advanced Fortran language,
- parallel computing using MPI,
- advanced numerical methods.

So, do not take this course if you think you are advanced student eager to learn cutting-edge computation techniques and numerics.

... and, never, never do this in class!
**Suggested textbooks:**

**Format of class:**
- 1 to 2 hours of lecture plus 1 to 2 hours of programming practice

**Grading:**
- based on the grades of midterm quiz, and/or final quiz, and/or take-home projects

**Exercise (Homework):**
- In-class exercise/homework after the every class
- No need to turn in the exercise/homework

**Teaching assistant:** 呂冠鴻  d01525011@ntu.edu.tw
Some notes about Fortran:

- There exist computer programming languages other than C and C++.

- C/C++ is not the only high level computer programming language being used. (Ada, Basic, COBOL, Fortran, Pascal...)

- Fortran = Formula Translation, is mainly for general numerical computation.

- Is FORTRAN an outdated programming language?
  Of course not, otherwise why there are still so many people using it; and the language has been continuously updated since its first release in 1957!

- Who use Fortran?
  - People who do number crunching, i.e., computing.
  - Many of the HPC (High-Performance Computing) codes have been around for > 20 years and are still in use today.
Evolution of Fortran:

- First released in 1957 by IBM
- Fortran II: 1958 (subroutine and function appeared)
- Fortran IV: 1962
- Fortran 66: 1966 (adapted as an ANSI standard)
- Fortran 77: 1977
- Fortran 90/95: 1990/1997
- ......

Modern structural programming features are introduced.
But many undesirable features from earlier versions are also retained.

For backward compatibility.
Do not recommended to use. (obsolescent features)
**Software used in this course:**

- GNU Fortran compiler or Intel Fortran
- GNU emace editor

GNU = GNU is not UNIX

- GNU project originally intended to develop a free Unix-like computer operating system, with an ultimate goal to be a complete Unix-compatible software system composed wholly of free software.
  (But the current mainstream trend is to use GNU software with Linux kernel.)

- Started in 1984 by Richard Stallman who quit his job at MIT Artificial Intelligence Laboratory.
To use GNU Fortran in Microsoft Windows

- Minimalist GNU for Windows (MinGW)
- a minimalist development environment for native Microsoft Windows applications
- provides a complete Open Source programming tool set, including Fortran
- a zipped folder containing MinGW, GNU Emacs and cmd to start Emacs and Fortran compiler: GNU_emacs_Fortran.zip
- in the folder \GNU_emacs_Fortran
• in the folder \GNU_emacs_Fortran

CMD_emacs_Fortran.bat: open both command prompt console and Emacs editor
CMD_Fortran.bat: open only command prompt console
Open_emacs.bat: open only Emacs editor
Use *command prompt console* to

- move around directories
  
  >dir
  >mkdir homework01
  >cd homework01
  >cd ..

- compile program
  
  >gfortran hw01.f90

- execute program
  
  >a.exe
→ Use *Emacs editor* to
  - create a new program,

then edit the program.
→ Use *Emacs editor* to
open and edit an existing program.
Graphics Software

• Use any graphics software (e.g., Matlab, Tecplot) you prefer for plotting the computation output if needed.

• Two freeware for graphics:
  ◦ **Gnuplot** (a portable command-line driven graphing utility)
  ◦ **Octave** (freeware similar to Matlab)