

## **GEO XXX: COMPUTER PROGRAMMING AND NUMERICAL TECHNIQUES FOR THE GEOSCIENCES**

INSTRUCTOR: ANDREW RIDGWELL  
PHONE 951 XXX XXXX, GEOLOGY XXXX  
E-MAIL: ANDREW.RIDGWELL@UCR.EDU

Foundation course, lecture/lab. Geo xxx. Pierce xxxx.

### **Units:**

4 unit course

### **Hours per week:**

Lecture: 3

Lab: 3

### **Description:**

This course will provide an introduction to computer programming and numerical modelling for Earth and Environmental Science problems. It will provide a chance to learn a computer programming language and all the elements that constitute it, including number bases and types, logical constructs, debugging, etc. The course will develop programming skills step-wise, applying them at each point to practical questions and outcomes, such as data progressing and visualization. How complex environmental processes can be encapsulated and approximated, and numerical models constructed, will be illustrated. Guided opportunities will be provided to build a 'DIY' climate model and in doing so, further develop programming and modelling skills at the same time as reinforcing basic concepts in climate dynamics through 'practice' rather than just theory. The cumulating objectives of the course are to develop an understanding of how computers and the internet 'work', how computer programs are written and numerical models constructed, and hence foster a critical understanding of modern technology and its application in the Earth and Environmental Sciences.

### **Prerequisites:**

TBA

### **Course syllabus:**

The weekly format of the Class would be of one, 3-hour computer practical session per week, plus two 1½ hour lectures. The computer practical is the central element and these would consist of structured exercises leading step-by-step through the components of computer programming and numerical model construction, debugging, and testing, plus applications to common geosciences problems.

The first 1½ hour lecture each week will outline the basics and introduce the key

concepts of the week. The 1½ hour session ending the week would be to ensure all the concepts are understood and misconceptions resolved and would be a mix of presentation and worked-through examples, plus questions and discussion.

### **Week 1**

*Lecture 1: Introduction to the course.*

Lab: Basic skills and model building in Excel.

*Lecture 2: Numbers and bases. Logic. Basic constituents and functioning of computers. Programs.*

### **Week 2**

*Lecture 1: Introduction to MATLAB.  
Matrix maths.*

Lab: MATLAB basics, including variables and matrixes, data I/O. Data processing in MATLAB.

*Lecture 2: Worked examples.*

### **Week 3**

*Lecture 1: Data visualization.*

Lab: Data visualization – types of 1-D plots. 2-D plots and interpolation. Re-gridding. Spectral analysis. Basic statistics.

*Lecture 2: Worked examples.*

### **Week 4**

*Lecture 1: Fundamentals of computer programs.*

Lab: Loops and conditionals. Further data processing techniques.

*Lecture 2: Best practice in computer programming: structure and debugging.*

### **Week 5**

*Lecture 1: Introduction to numerical models.*

Lab: Time-stepping and (zero-D) box models. Geochemical reservoir dynamics, numerical stability.

*Lecture 2: Worked examples.*

### **Week 6**

*Lecture 1: 1-D and reaction-transport models.*

Lab: Simple 1-D gridded modelling.

*Lecture 2: Applications of reaction-transport models to environmental problems.*

### **Week 7**

*Lecture 1: 2/3-D and global environmental numerical models.*

Lab: DIY climate model #1 – surface energy budget and greenhouse gases.

*Lecture 2: Fundamentals of the global climate system and surface energy balance.*

### **Week 8**

*Lecture 1: Empirical and statistic models.*

Lab: DIY climate model #2 – feedbacks and dynamics.

Lecture 2: *Climate modelling discussion.*

### **Week 9**

Lecture: *Review of other programming languages and applications.*

Lab: Basic linux commands and scripting. Introduction to programming in FORTRAN and compiling.

Lecture 2: *Worked examples.*

### **Week 10**

Lecture 1: *The internet – what it is, how it works. Finding, accessing, and visualizing geosciences resources on the web.*

Lab: html programming and webpage construction.

Lecture 2: *Discussion and course wrap-up.*

### **iLearn:**

Lecture handouts (PDF format) will be made available prior to the start of the lecture in a compact printable and annotatable format.

### **Course goals:**

Upon completion of the course, students will be expected to have gained through hands-on practical exploration, factual knowledge and an understanding of:

- Number bases, how computers 'work' plus computer programs and their basic building blocks. (*Learning Outcome 2*)
- Numerical models and the representation of time. Construction and application of a variety of models spanning box models of biogeochemical cycles and population dynamics, through 1D reaction-diffusion models of surface Earth processes, to 3D gridded global models. (*Learning Outcomes 1 and 2*)
- The Greenhouse effect and basic climate feedbacks. (*Learning Outcome 1*)
- The linux operating system; compiled languages and FORTRAN, plus how webpages and the internet 'work'. (*Learning Outcome 2*)

as well as:

- The use of numerical models in addressing scientific questions and testing hypotheses as well as the limitations of numerical models. (*Learning Outcomes 2 and 4*)

The course will provide transferable skills in:

- Written communication and presentation. (*Learning Outcome 3*)
- Problem solving and quantitative analysis together with logic and fault-finding. (*Learning Outcomes 4 and 5*)
- Computer programming. (*Learning Outcomes 2 and 4*)
- Effective internet use and website construction. (*Learning Outcomes 2 and 4*)

## **Assessment:**

The primary assessment of the course will be in the form of a student-produced website displaying the graphical and tabulated results of a set of simulations made using a model built by the student. The scope of the model would be restricted and prescribed but with flexibility in terms of what is done and explored using it. The website will be presented in a short (5-10 minute) interactive demonstration. This will count for 60%. In addition, there would be an earlier, shorter, mid-term assessment, for which a computer program must be written in MATLAB to solve a specific problem, and the code handed in. The code will be 'run' to judge the overall success of the program, with additional marking of the code itself (including structure and commenting). This will count for 40%.

## **Course textbook:**

**TBA**

## **Recommended reading:**

**TBA**