Master Course Syllabus

School of Engineering and Computer Science
Washington State University Vancouver

CS 330
Numerical Computing

3 Semester Hours

Catalog Description

Power and limitation of numerical solutions; design, analysis and implementation of numerical algorithms; visualization and rendering.

Prerequisite Courses

Math 172 – Calculus II

Math 220 – Linear Algebra

CS 121 – Program Design and Development

Prerequisite Topics

- Proficiency with the C programming language
- Use of Unix environment for coding, compilation, debugging and testing
- Differential and integral calculus, Taylor's Theorem
- Systems of linear equations and matrix operations

Measured Course Outcomes

Students taking this course will:

1. Identify the sources of error in numerical programs and algorithms. *(contributes to performance criterion **A-4**)*
2. Design, code, test and debug programs that implement numerical methods. *(contributes to performance criterion **E-2**)*
3. Utilize a graphics or plotting software library to present numerical data visually. *(contributes to performance criterion **K-6**)*

Required Textbooks

Numerical Mathematics and Computing, by Ward Cheney and David Kincaid, Thomson, 5th Edition. (ISBN 0-534-38993-7)

Reference Material

GNU Plotting Utilities online documentation.

Major Topics Covered in the Course

1. Machine representation of numbers
2. Sources of error, solution stability and convergence

3. Curve fitting and approximation
4. Two dimensional plotting
5. Finding roots of equations
6. Interpolation and numerical differentiation
7. Numerical Integration
8. Solving systems of linear equations
9. Solution methods for ordinary and partial differential equations

Laboratory Projects

Programming Project Area	Weeks
Implement bisection and Newton's method for finding roots	1
Display equations and roots visually using plotting library	1
Integrate using trapezoid rule and Simpson's rule, display results and compare accuracy	2
Solve a system of linear equations using Gaussian Elimination and back substitution	2
Compute interpolating polynomials for discrete data sets	1
Solve a system of differential equations and display the results graphically	3

CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	0	0	Computer Organization and Architecture	0	0
Algorithm & Software Design	0	3	Concepts of Programming Languages	0	0

Oral and Written Communications

There are no significant oral or written communications required in this course. Virtually all assignments consist of writing computer programs.

Social and Ethical Issues

This course contains no significant coverage of social and ethical issues beyond the usual proscriptions against plagiarism and cheating. Some coverage of solution accuracy and stability as they relate to critical software applications and liability are presented.

Theoretical Content

Topic	Hours
Machine representation of numbers	2

Problem Analysis

The instructor performs analysis of representative problems in class. All student programming assignments require the student to analyze problem requirements. The instructor analyzes problem solutions (both his own and student's) in class.

Solution Design

This course requires the student to craft 6-8 correctly functioning computer programs. The requirements for each program will necessitate that the student comprehend and apply mathematical knowledge to their program designs. These programs range from 50-100 lines of code at the beginning of the semester, to a final project whose design usually entails >500 lines of program code.

CC2001

This course provides coverage of topics in the following areas (hours listed are minimums):

AR2. Machine representation of data [core]	2
CN1. Numerical Analysis [elective]	30
SE2. Using APIs [core]	4

Course Coordinator:	Dick Lang	
Last Updated:	August 26, 2005	(Approved)
Syllabus Version Number:	1.1	