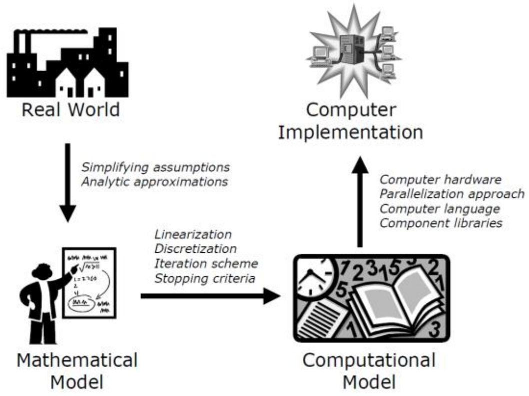
ECE 4960 Spring 2017: Computational and Software Engineering

By Edwin C. Kan
Lecture: MWF 11:15am – 12:05pm (362 Hollister) Laboratory: Tuesday 7:30pm – 9:00pm

Logistics	**Monday** 11:15am – 12:05pm	**Tuesday** 7:30pm – 9:00pm	**Wednesday** 11:15am – 12:05pm	**Friday** 11:15am – 12:05pm
Oliveira Chaps. 6-8 Einarrson Chap. 8 Bindel Chap. 1	**1/23** No class	**1/24** No class	**1/25** Class introduction	**1/27** Software for the real world
Bindel Chap. 2; Einarsson Chaps. 1, 2	**1/30** Source of errors in computing: precision	**1/31** Language tradeoff; structure and objects;	**2/1** Integer and floating point representation standards	**2/3** Exception handling of integers and floating-points
Bindel Chap. 3 Oliveira Chaps. 3,9 Einarrson Chap. 13	**2/6** Conditioning of functions and round-offs	**2/7** Source code control and regression test suites	**2/8** Local analysis and approximation by Taylor series	**2/10** Euler and Richardson methods
Einarsson Chap. 3 *Coding 1: Exception handling*	**2/13** No class	**2/14** Lab 1 practicum	**2/15** Integration and Gaussian quadrature	**2/17** Error estimation; Unit testing
Bindel Chap. 4	**2/20** February break: No class	**2/21** Linear algebra by software: Blas	**2/22** Linear algebra: full and sparse systems	**2/24** Direct and iterative solvers: factorization
Einarsson Chaps. 4 – 6 Chapra Part 3	**2/27** Matrix conditioning and pivoting	**2/28** Modular programming and object design/testing	**3/1** Error analysis: perturbation and noise injection	**3/3** Nonlinear equation and optimization: Jacobian matrix
Bindel Chap. 5; Oliveira Chaps. 14, 15 *Coding 2: Matrix*	**3/6** Nonlinear equation and optimization: Jacobian matrix	**3/7** Lab 2 practicum	**3/8** Iterative methods for local and global optimization	**3/10** Line search and quasi-Newton methods
Bindel Chap. 6 Oliveira Chap. 13	**3/13** Least-square and parametric optimization	**3/14** Management of memory: violation and leaks	**3/15** Least-square and parametric optimization	**3/17** Computational geometry and spline fitting
Bindel Chap. 7 *Coding 3: Parametric optimization*	**3/20** Variations in spline fitting	**3/21** Lab 3 practicum	**3/22** Statistical methods: Monte Carlo	**3/24** Ordinary differential equation and local analysis
Chapra Part 5	**3/27** Euler to Runge Kutta	**3/28** Introduction to SPICE	**3/29** Euler to Runge Kutta	**3/31** No class
	4/3 Spring break: No class	**4/4** Spring break: No class	**4/5** Spring break: No class	**4/7** Spring break: No class
Chapra Part 7 *Coding 4: Nonlinear circuit simulation*	**4/10** Adaptive Runge Kutta by error estimation	**4/11** Lab 4 practicum	**4/12** Multi-step method: TR-BDF2	**4/14** Error estimation and hp adaptivity
Bindel Chap. 8	**4/17** 1D finite-difference PDE solver: elliptic	**4/18** (Reserved for make-up class)	**4/19** 1D finite-difference PDE solver: elliptic	**4/21** Finite-difference parabolic PDE solver: parabolic

Bindel Chap. 9 Chapra Part 8	**4/24** 1D finite-difference PDE solver: hyperbolic	**4/25** Case study for large-scale scientific software	**4/26** 1D finite-element solvers	**4/28** 2D and 3D finite-difference solvers
Coding 5: Poisson solver	**5/1** 2D and 3D finite-difference solvers	**5/2** Lab 5 practicum	**5/3** 2D and 3D finite-element solvers	**5/5** 2D and 3D finite-element solvers
	5/8 Future of Software engineering	**5/9** Hacking 4960	**5/10** No class	**5/12** No class

Course description: This course will introduce the mathematics, design, maintenance and testing practices to computing for interface and simulation of the physical real world. We will introduce mathematical and software techniques to identify and correct the possible error in computing implementation of approximation, matrix manipulation, optimization, geometry and differential equations. Computational conditioning and its relation to nonlinear solvers, as well as stability and error estimation, will be examined. Software design and language characteristics for large-scale computing will be briefly overviewed. Students can choose their most comfortable general-purpose development platforms, and C/C++ with ECE applications will be used for illustration. The Lab section on Tuesday night will be used to introduce programming techniques and project interaction.



Pre-requisites: ECE 2400 or CS 2110. An introductory course in scientific computing will be helpful but not required.

Reference textbooks: (all reading will be provided on-line)

1. D. Bindel and J. Goodman, *Principles of Scientific Computing*, 2009.
2. S. Oliveira and D. Stewart, *Writing Scientific Software: A Guide to Good Style*, Cambridge 2006.
3. B. Einarsson, Ed., *Accuracy and Reliability in Scientific Computing*, SIAM 2005.
4. S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers, 7th Ed.*, McGraw-Hill, 2015.
5. (Optional) S. McConnell, *Code Complete: A Practical Handbook of Software Construction, 2nd Ed.*, Microsoft Press, 2004.
6. (Optional) A. Allain, *Jumping into C++*, Cprogramming.com, 2015.

Program assignments: There are 5 program assignments as the main efforts throughout the semester. Each assignment will contain required smaller modules for design and testing purposes, as well as a culminating program. The grouping regulation will be described in each assignment. Groups of 2 students are the usual cases. Good program practices of version control and object models are strongly encouraged to facilitate code reuse, which not only enhances developer productivity, but more importantly improves reliability and ease of management.

Grading: Reading material reviews (by multiple choice questions on Blackboard): 10%; Coding: 15% each; Final hacking exam: 15%.

Blackboard site: 13047_2017SP