Lecturer : Christina Christara, `ccc@cs`
Lectures : Tuesday 1-3 PM, Room BA B024
Tutor : Kirill Ignatiev, `ikirill@cs`
Tutorial : Thursday 1-2 PM, Room BA B024
Office Hours : Monday 3:30-4:30 PM, or by appointment, Room BA 4226
Textbook : Michael Heath, Scientific Computing: an introductory survey, McGraw-Hill Inc., selected chapters copied and printed by Custom Printing, UT bookstore.
Web site : `http://www.cs.toronto.edu/~ccc/Courses/350.html`
Bulletin board : `https://csc.cdf.toronto.edu/csc350h1f`

Aims of course

Introduce numerical methods for solving linear and nonlinear algebraic equations and systems.
Evaluate numerical methods with respect to their accuracy, time and memory complexity.
Develop and practice computer skills in implementing numerical methods efficiently on the computer.
Use high level software for studying numerical methods.

Skills / Knowledge testing in the course

Apply basic principles, not recall lecture notes in detail
Problem recognition
Method recognition
Apply a given method correctly
Solve a numerical problem efficiently and reliably using numerical software.

Prerequisite Mathematics

Ability to handle notation and to do algebraic manipulation

Induction

Calculus including:
Differentiation and integration of polynomial, trigonometric, exponential, logarithmic and rational functions, continuity, limits, graphs of functions, Taylor series, Rolle's theorem, mean-value theorem, de l' Hospital's rule, etc. Basic multivariate calculus.

Elementary Linear Algebra including:
Matrix and vector addition and multiplication, elementary row operations, linear (in)dependence, inverse matrix, etc.

Programming: knowledge of basic programming constructs, such as for loops and if-then-else statements; manipulation of vectors and matrices; knowledge of (or will to learn) MATLAB, or knowledge of some conventional programming language, such as FORTRAN or C.

Computer accounts

You will get a computer account on the CDF Unix system. Consoles/workstations are located in the Bahen Center for Information Technology building. Most computer assignments **must** be run on that system. You must log-in frequently and read mail, news and other messages relating to the course through your account.

Marks distribution

Assignment 1	Due Tuesday, October 8	12%
Term test	Thursday, October 24	24%
Assignment 2	Due Thursday, November 7	12%
Assignment 3	Due Tuesday, December 3	12%
Final exam		40%

- Must get at least 33% in the final exam.
- Must get at least 33% average in the computer assignments.
- **Midterm test and Final exam** : Calculators are the only aids permitted.

Problem sets / Computer assignments

problem sets: please write as clearly as possible.
Indicate your last (family) name by capitalisation or underlining in the front page of your paper.
computer assignments: don't leave it to the last minute - think of the following
  − the machine being down, when you need it.
  − the workstation room being crowded.

− the printer being stuck, when you are just at the time to get your final listing.
− accidentally deleting an important file.
overcome this by using backup procedures (for the source and data files only).

The above are not good reasons for extension of the assignment due date.

Late assignment policy

Assignments are due the day posted, at lecture/tutorial time. Assignments submitted late have a reduction of marks based on the maximum total marks the assignment could get, had it been submitted on time (and not the total marks the assignment actually got). Each day costs 10%, to a maximum of 5 days. Assignments submitted later than 5 days after the due date do not receive any marks. Weekends and holidays count as regular days for the purpose of late assignment policy.

Topics to be covered

- Introduction (Ch. 1)
  Computer representation of numbers, computer arithmetic
  Round-off error, error propagation, conditioning and stability
- Square linear systems - direct methods (Ch. 2)
  Gauss elimination, LU factorisation, pivoting, scaling, forward and back substitution
  Symmetric and symmetric positive definite matrices, Choleski factorisation
  Special cases: tridiagonal, banded, general sparse systems
- More on matrices (Ch. 2 − §2.3, §2.4, Ch. 4 − §4.1)
  Vector and matrix norms
  Condition number of a matrix
  Iterative refinement
  Eigenvalues and eigenvectors
- Non-square linear systems − Linear least squares solution (Ch. 3)
  Least squares solution, normal equations
  QR factorisation, Gram-Schmidt method for matrices
- Computing eigenvalues and eigenvectors (Ch. 4)
  The power method; the QR iteration
- Nonlinear equations and systems (Ch. 5)
  Bisection; fixed point iteration, Newton's method; secant
  Convergence, accuracy and efficiency
  Newton's method for systems, Jacobian matrix
  Modifications to Newton's method, approximations to Jacobian
  Quasi-Newton methods; Broyden's method
- Unconstrained optimisation (Ch. 6, time permitting)
  one-dimensional: Golden section search; Newton's method
  multi-dimensional: Steepest descent; Newton's; BFGS; Gradient methods

Other references

William W. Hager
Applied Numerical Linear Algebra
Prentice Hall 1988

Conte, S. D. and Carl de Boor
Elementary Numerical Analysis
McGraw-Hill Inc.

Johnson, L. W. and R. D. Riess
Numerical Analysis
Addison Wesley

David Kincaid and Ward Cheney
Numerical Analysis
Brooks/Cole

Richard L. Burden and J. Douglas Faires
Numerical Analysis
Brooks/Cole

Dahlquist, G. and A. Bjorck (trans. N. Anderson)
Numerical Methods
Prentice Hall

Stoer, J. and R. Bulirsch
Introduction to Numerical Analysis
Springer Verlag

Golub, G. and Charles Van Loan
Matrix computations
John Hopkins University Press

Ciarlet, Philippe G.
Introduction to numerical linear algebra and optimisation
Cambridge University Press