Lecturer : Christina C. Christara
Lectures : Wednesday 1-3pm, Room Virtual
Tutorial : Thursday 3-4pm, Room Virtual (tutorials will be used for lectures)
Office Hours : Monday 3:30-4:30pm, Room Virtual, other hours by appointment
Web site : `http://www.cs.toronto.edu/~ccc/Courses/456.html`
Electr. forum : `https://bb-2021-01.teach.cs.toronto.edu/c/csc456`

Topics to be covered

- Introduction
  - Parallel architectures, communication complexity; Speedup, efficiency
  - Simple examples: inner product, matrix-vector multiplication, total exchange
  - Performance study
- Linear systems - Direct methods
  - Gauss elimination, LU factorisation, Cholesky decomposition, back substitution
  - Banded systems; Cyclic reduction; Partitioning methods
- Linear systems - Iterative methods
  - Jacobi, Gauss Seidel, SOR, SSOR and conjugate gradient methods
  - Preconditioning; Sparse linear systems; Multicolouring
  - Asynchronous iterations
- Partial Differential Equations
  - Schur complement - domain decomposition method
  - Schwarz splitting - domain decomposition method
  - Multigrid method
  - Fast Fourier Transform methods
- Interpolation
  - Deboor decomposition

Aims of course

Introduce the basic concepts in parallel computation and state-of-the-art scientific computing.
Formulate parallel numerical methods.
Implement the above methods on specific parallel architectures.
Study the performance of methods and machines.
Offer lots of fun.

Prerequisites

- Elementary calculus: Taylor series, Rolle's theorem, mean value theorem, graphs of functions, continuity, convergence, de l' Hospital's rule, partial derivatives, etc.
- Numerical Linear Algebra (included in CSC336): rough knowledge of direct methods for solving linear systems; some familiarity with sparse matrices; fluency in matrix and vector manipulation.
- Interpolation (included in CSC336 or CSC436): some knowledge on interpolation.
- Partial Differential Equations: minimal knowledge on PDEs.
- Theory of Computer Algorithms: some knowledge on data structures, computer algorithms and computational complexity.
- Programming: proficiency in some conventional programming language, preferably C/C++ or FORTRAN; knowledge of MATLAB is useful but not necessary.

Tentative marks distribution

Assignment 0	Due Wednesday, February 3	8%
Assignment 1	Due Thursday, February 11	12%
Term test 1	Thursday, March 4	20%
Assignment 2	Due Thursday, March 18	20%
Term test 2	Thursday, March 25	20%
Assignment 3	Due Thursday, April 8	20%

- Must get at least 30% in **each** of the assessments; can't skip any
- Term tests (and final exam, if any): Calculators and this course's materials are the only aids permitted.
- The assignments include substantial computer work.
- Assignments are expected to look like short reports, i.e., the presentation of the subject counts too.

The final marks distribution will be confirmed in 3 weeks.

References

Christina C. Christara
CSC456-2306 Lecture Notes on the website

James M. Ortega
Introduction to Parallel and Vector Solution
of Linear Systems
Plenum Press 1988

Yousef Saad
Iterative Methods for Sparse Linear Systems
PWS 1996 or SIAM 2003
http://www-users.cs.umn.edu/~saad/books.html

Ian Foster
Designing and Building Parallel Programs
Addison Wesley 1995 and
http://www.mcs.aln.gov/dbpp

George Em Karniadakis and Robert M. Kirby II
Parallel Scientific Computing in C++ and MPI
A Seamless Approach to Parallel Algorithms and
their Implementation
Cambridge 2003

J. M. Bahi, S. Contassot-Vivier and R. Couturier
Parallel Iterative Algorithms:
from sequential to grid computing
Chapman & Hall/CRC 2007

William Gropp, Ewing Lusk and Anthony Skjellum
Using MPI: portable parallel programming
with the message-passing interface
MIT Press 2014
see also
http://wgropp.cs.illinois.edu/usingmpiweb/

Michael J. Quinn
Parallel Programming in C with MPI and OpenMP
McGraw Hill 2004

Jianping Zhu
Solving Partial Differential Equations
on Parallel Computers
World Scientific 1994

Ananth Grama, Anshul Gupta, George Karypis
and Vipin Kumar
Introduction to Parallel Computing:
Design and Analysis of Algorithms
Addison Wesley 2003

Jeffrey D. Ullman
Computational Aspects of VLSI
Computer Science Press 1984

Dimitri P. Bertsekas and John N. Tsitsiklis
Parallel and Distributed Computation;
Numerical Methods
Prentice Hall 1989
see also
https://dspace.mit.edu/handle/1721.1/3719#files-area

William W. Hager
Applied Numerical Linear Algebra
Prentice Hall 1988

Gene H. Golub and Charles Van Loan
Matrix computations
John Hopkins University Press 1996

Uri Ascher and Chen Greif
A first course in Numerical Methods
SIAM 2011 (e-book on library)

Samuel D. Conte and Carl de Boor
Elementary Numerical Analysis
SIAM 2018 (also McGraw-Hill Inc.)

David Kincaid and Ward Cheney
Numerical Analysis
Brooks/Cole

Michael Heath
Scientific Computing: an introductory survey
McGraw-Hill Inc.

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

John C. Strikwerda
Finite Difference schemes and
Partial Differential Equations
Wadsworth and Brooks/Cole 1989

William F. Ames
Numerical Methods for Partial Differential Equations
Academic Press 1977 3rd edition (or 2nd edition)
(or Thomas Nelson & Sons)

P. M. Prenter
Splines and Variational Methods
John Wiley & Sons 1975

William L. Briggs, Van Emden Henson, Steve McCormic
A multigrid tutorial
SIAM 2000

Selected papers

Late assignment policy

Assignments are due the day and time posted. 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 on the total marks the assignment actually got). Each day costs 10%, to a maximum of 3 days. Assignments submitted later than 3 days after the due date do not receive any marks. Weekends and holidays count as regular days for the purpose of late assignment policy.

Academic integrity

Assignments, homeworks and exams must be your own individual work and using only course materials. While students at your level are well aware of what academic integrity means, please note that violating academic integrity includes more things than presenting others' work as one's own. For example, *not taking reasonable measures to protect your work from being plagiarized by others is also a violation of academic integrity*. This is becoming particularly important now that so many things are online.

You should *never post anywhere or share with anyone* assignments, exams or solutions, *even after the deadline*.

Additional information

Assignments and exams will be submitted electronically; details to be given with each assessment.

Assignments will be typed in latex. A template is given in the course website.

Exams will be handwritten and scanned, or, alternatively, typed in latex.

Tests/exams will be synchronous. The default time for term tests is the teaching time of the day, but, depending on timezones and availability, there may be another time chosen.

Must get at least 30% in **each** of the assessments; can't skip any

Lectures/tutorials will be offered online (through zoom). I expect to be able to post recordings too. Links will be posted in the course website or e-mailed.

Office hours will be available with pre-arrangement, either at default times posted, or at other mutually agreeable times. Office hours will be for individual students, not for a group of students.