CSci 8314 Spring 2019
Sparse Matrix Computations

General Information

This course is an introduction to sparse matrix techniques with an emphasis on solving sparse linear systems of equations and eigenvalue problems. The course will begin with a general discussion of sparse matrices, their origins and how they are stored and exploited. Then it will cover direct methods and iterative methods for solving linear systems of equations. It will also cover other topics related to sparsity, e.g., graph-based algorithms in machine learning and algorithms for eigenvalue problems. A tentative list of topics is shown at the end of this syllabus. This will be a practical course. Students will learn about the algorithms and their complexity or convergence theory, and they will also get an understanding on how to implement them and how to work with sparse matrices in general. One of the requirements of the course is the completion of a term project, possibly using available software packages.
The course will consist of lectures given by the instructor and in class discussions. It qualifies as a 50% project toward a computer science plan C (coursework only) Masters.

- **Class Schedule:** 09:45 AM - 11:00 AM; M-W Akerman Hall 227
- **Instructor:** Yousef Saad `http://www.cs.umn.edu/~saad`
  [e-mail: saad@umn.edu]
  Office: 5 -225B EE/CS bldg – Office Phone: (612) 624 – 7804.
- **Prerequisite:** csci 5304 (or equivalent).
- **Office hours:** See class web-page

Textbook

Part of the lectures will be based on my book: *"Iterative methods for sparse linear systems (2nd edition)"* which is posted (free download) – see

`http://www.cs.umn.edu/~saad/books`.

This will be supplemented by articles and lecture notes.
Supplemental reading: *Direct methods for sparse linear systems*, by T. A. Davis, SIAM publishing, 2006. (see also accompanying software CSparse noted in the resources link in the class web-site (see below). *You need not purchase the books listed above.* My own book is posted for free and will be only used for part of the material.
Note: Matlab/Octave will be often used for writing short programs. See the matlab section of the class web-page for information and some documentation.

Lecture Notes

In addition to the text, lecture notes on some of the chapters will be posted regularly. These will be posted on the class web-site:
`http://www-users.cselabs.umn.edu/classes/Spring-2019/csci8314/`
Click on the "Lect. Notes" icon in the menu. These may include material not available in the text-book. The lecture notes will be posted by topic rather than lecture by lecture. I try to have these posted in advance.

Project

To pass this course, each student will need to complete a class project. This will generally be on the use of sparse matrix techniques in specific application areas. You can for example write a comprehensive survey on a selected topic related to the course. But you can also write a paper on how to exploit a specific class of sparse matrix methods in your own research. In the past the projects ranged from exhaustive survey articles on a specific topic (e.g., 'sparse matrix techniques in text mining') to implementations /comparisons of (a) specific method (s) say for solving sparse linear systems; or a specific theme in your own research that illustrates the use of sparse matrix methods. A few sample topics along with additional details, including grading criteria, what and how to present, and so on, will be given toward the middle of the semester. Minimal requirements, as well as the details on grading criteria, will be specified later.
Given the class size, not everyone will present her/his project at the end of the semester – Details on this will be provided later.

Evaluation

The evaluation of your performance in this class will be based on the following:

- Two written assignments at 7 % each – these may include a mix of small programming assignments (matlab), and theoretical questions/
- Class participation and attendance 6 %
- Two tests at 15 % each: 30% [actually two best out of three given tests.]
- Term Project : 50 %. Projects will be presented in the last 3 lectures.

There will be no make-up Tests: if you miss any one of the 3 exams then you still have 2 other exams that can be used to calculate the full exam part (30%) of the score. Note also that the tests may be 45mn, or or 60mn, or 75mn long. Final grades will *not* be based on a competitive curve. They will be assigned based on the following scale, where T is the total score (out of 100) you achieved in the class.

A : $100 \geq T \geq 92$	**A-** : $92 > T \geq 87$	**B+** : $87 > T \geq 83$
B : $82 > T \geq 77$	**B-** : $77 > T \geq 72$	**C+** : $72 > T \geq 65$
C : $65 > T \geq 60$	**C-** : $60 > T \geq 55$	**D+** : $55 > T \geq 50$
D : $50 > T \geq 40$	**F** : $40 > T$	

For example, to get a B you will need a grade between 77 (inclusive) and 82 (exclusive).

Overview of topics to be covered (tentative)

1. Background. Introduction and background in linear algebra. Linear algrebra problems. Sparse matrices. Sparsity - Origins of sparse problems. Discretization of Partial Differential Equations.
2. Finite Difference and Finite Element Discretization. Other origins of sparse matrices. Electrical networks, Information retrieval, ... Graph representation of sparse matrices.
3. Storage schemes for sparse matrices. Regular and irregular structures. Focus: Sparse matrices in matlab. Background on direct solution methods; Variants of Gaussian Elimination. Permutations and orderings. Mesh problems.

4. Sparse direct methods. Band and envelope methods. Cuthill-Mc Kee and reverse Cuthill-Mc Kee orderings.

5. Sparse direct methods - General techniques. Graph representation. Elimination tree. The frontal approach. Multifrontal approach.

6. Minimal degree ordering, Nested Dissection ordering. Static methods (SPD case). Dynamic methods (multifrontal). Sparse direct solvers - existing software. SUPERLU, MUMPF, UMFPACK,..

7. More on Graphs: Graph Laplaceans, graph partitioning, clustering; Application: image segmentation; Networks and graphs; measure of node centrality.

8. Iterative methods. Projection methods. One-dimensional case: steepest descent, minimal residual methods. Krylov subspace methods. Krylov subspaces and Arnoldi's method. Conjugate gradient (CG) method, basic theory. Connection to Lanczos tridiagonalization and to orthogonal polynomials.

9. The GMRES variants: GMRES, FGMRES, ORTHOMIN, ... Nonsymmetric Lanczos and related methods. The two-sided Lanczos algorithm. Bi-Conjugate gradient. CGS and bi CGSTAB.

10. Preconditioning. Preconditioned iterations: left, right, split preconditioning, PCG. SSOR preconditioner. ILU(0) and ILU(k). ILUT. Block preconditioners. Software: SPARSKIT, ITSOL, ILUPACK.

11. Eigenvalue problems. Types of problems. Background - Tools. The power method. Simple projection techniques. Subspace iteration. Krylov methods

12. Krylov methods (cont.). Arnoldi's method. The Lanczos algorithm. Nonsymmetric Lanczos.

13. Krylov methods (cont.). Arnoldi's method. Implicit restarts. Software: ARPACK. Sample applications: Quantum mechanics (Schrödinger equation), Applications in page ranking.

If time permits (or to be covered by student projects:)

14. Other applications of sparsity. Nonlinear equations, optimization. Least-squares problems. SVD, and Principal Component Analysis. Application. Latent Semantic Indexing in information retrieval. Others.

15. Overview of Domain Decomposition and/or Multigrid Methods

16. Multipole, H-/H2 matrices, hierarchical matrices