CIS 6930/4930: Sparse matrix algorithms

Timothy A. Davis
Computer and Information Science and Engineering Department,
University of Florida. email: davis@cise.ufl.edu.
http://www.cise.ufl.edu/∼davis.

Fall 2007

1 Course Overview

Students in any discipline that uses scientific computing methods will benefit from this course. A wide range of problems in science and engineering require fundamental matrix computations for their solution, and these matrices are often mostly zero. The world's largest sparse matrix arises in Google's web page ranking problem. About once a month, Google finds an eigenvector of a 8 billion by 8 billion sparse matrix that represents the connectivity of the web. Other sparse matrix problems arise in circuit and device simulation, finite element methods, linear programming and other optimization problems, acoustics, electromagnetics, computational fluid dynamics, financial portfolio optimization, structural analysis, and quantum mechanics, to name a few.

Taking effective advantage of the structure of a sparse matrix requires a combination of numerical and combinatorial methods (graph algorithms and related data structures). For example, finding the best ordering to factorize a matrix is an NP-complete graph problem. Topics focus on direct methods, but with some application to iterative methods: sparse matrix-vector multiply, matrix-matrix multiply and transpose, forward/backsolve, LU and Cholesky factorization, singular value decomposition, reordering methods (including the use of graph partitioning methods), and updating/downdating a sparse Cholesky factorization. Projects in the course include programming assignments in C and MATLAB.

Prerequisites: numerical linear algebra, graph theory, data structures and algorithms, C, and MATLAB. This background can be obtained in COT 4501 Numerical Analysis, COT 3100 Discrete Mathematics, and COP 3530

Data Structures and Algorithms, or related courses. Contact the instructor if you have any questions about the prerequisites, or about the course in general.

Time and location: MWF 4th period (10:40am to 11:30am), in CSE E220.

2 About the instructor

The instructor developed and taught this course while on sabbatical at Stanford University in 2002-2003. He is the author of several sparse matrix algorithms (and software packages) used in MATLAB, Mathematica, FEMLAB, Nastran, the semiconductor simulator FLOODS/FLOOPS, and in many other scientific packages. For example, the MATLAB statement

```
x = A \ b
```

when `A` is a square sparse matrix is computed by UMFPACK or CHOLMOD, both written by the instructor. These codes are faster than the original ones by a factor of 3 (for small matrices) to 40 (for large ones). In collaboration with W. Hager, in the UF Math department, he is currently developing a new method for solving large sparse linear programming problems, and a quadratic-programming based method for graph partitioning. A project in collaboration with Gene Golub, Stanford University, will lead to a new method for finding singular values of sparse matrices. With Lawrence Berkeley National Laboratory, he is developing a new sparse matrix ordering algorithm. In a project funded by the Sandia National Laboratories, he is developing a sparse solver to replace the one currently used by the Spice circuit simulation package, which can factorize sparse circuit matrices up to 1000 times faster than the current method used in Spice. With Kermit Sigmon, formerly in the UF Math department, he is a co-author of a popular MATLAB Primer. He collects and maintains the world's largest sparse matrix collection, at http://www.cise.ufl.edu/research/sparse, which is widely used to develop and test sparse matrix algorithms.

3 Textbooks

The following textbook is required: **Direct Methods for Sparse Linear Systems**, T. A. Davis, SIAM, 2006.

The following book is optional: **MATLAB Primer,** 7th edition, Tim Davis and Kermit Sigmon, Chapman & Hall/CRC Press, 2004.

I may include additional papers, if needed.

4 Projects, exams, and such

There will be no exams or quizzes in this course. Grading will be based on projects only.