Algorithm 8xx: COLAMD, a column approximate minimum degree ordering algorithm *

Timothy A. Davis † John R. Gilbert ‡
Stefan I. Larimore § Esmond G. Ng ¶

October 16, 2000

Technical Report TR-00-006. Department of Computer and Information Science and Engineering, University of Florida. October, 2000.

Abstract

Two codes are discussed, `colamd` and `symamd`, that compute approximate minimum degree orderings for sparse matrices in two contexts: (1) sparse partial pivoting, which requires a sparsity preserving column pre-ordering prior to numerical factorization, (2) sparse Cholesky factorization, which requires a symmetric permutation of both the rows and columns of the matrix being factorized. These orderings are computed by `colamd` and `symamd`, respectively. The ordering from `colamd` is also suitable for sparse QR factorization, and the factorization of matrices of the form $\mathbf{A}^\mathsf{T}\mathbf{A}$ and $\mathbf{A}\mathbf{A}^\mathsf{T}$, such as

*This work was supported in part by the National Science Foundation under grant number DMS-9803599; in part by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract number DE-AC03-76SF00098; and in part by DARPA under contract DABT63-95-C-0087.

†Computer and Info. Sci. and Eng. Dept., University of Florida, Gainesville, FL, USA. email: davis@cise.ufl.edu. http://www.cise.ufl.edu/~davis.

‡Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304-1314. email: gilbert@parc.xerox.com.

§Microsoft, Inc. email: slarimor@microsoft.com.

¶Lawrence Berkeley National Laboratory, One Cyclotron Road, Mail Stop 50F, Berkeley, CA 94720. email: EGNg@lbl.gov.

those that arise in least-squares problems and interior point methods for linear programming problems. The two routines are available both in Matlab and C-callable forms. They appear as built-in routines in Matlab Version 6.0.

Categories and Subject Descriptors: G.1.3 [Numerical Analysis]: Numerical Linear Algebra – *linear systems (direct methods), sparse and very large systems* G.4 [Mathematics of Computing]: Mathematical Software – *algorithm analysis, efficiency*

General terms: Algorithms, Experimentation, Performance

Keywords: sparse unsymmetric matrices, linear equations, ordering methods

1 Overview

Sparse partial pivoting methods compute the factorization $\mathbf{PAQ} = \mathbf{LU}$ by first finding a column ordering $\mathbf{Q}$ [3, 4, 5]. The column ordering $\mathbf{Q}$ is selected without regard to the numerical values. The row permutation $\mathbf{P}$ is found via standard partial pivoting, without regard to sparsity. The goal is to find $\mathbf{Q}$ to limit the worst-case fill-in, regardless of how $\mathbf{P}$ is subsequently chosen. Such an ordering $\mathbf{Q}$ also limits the fill-in in the Cholesky factorization of the matrix $(\mathbf{AQ})^{\mathsf{T}}(\mathbf{AQ})$ as compared to the unpermuted matrix $\mathbf{A}^{\mathsf{T}}\mathbf{A}$. The matrix $\mathbf{A}^{\mathsf{T}}\mathbf{A}$ is not constructed by our ordering methods, however. Sparse partial pivoting is used extensively in Matlab, by both `lu` and the matrix division operators \ and /. The methods used by `colamd` and `symamd` are discussed in a companion paper [2].

2 `colamd`: a column ordering algorithm

`Colamd` computes a column ordering $\mathbf{Q}$, given the nonzero pattern of the sparse m-by-n matrix $\mathbf{A}$. It is based on an approximate minimum degree method [1]. The code can be used in two contexts: as a Matlab mexFunction, and as a C-callable routine. In Matlab, the usage is

```
Q = colamd (A)
```

where `A` is an m-by-n Matlab sparse matrix. The permutation `Q` is returned as a size n integer vector. The permuted matrix $\mathbf{AQ}$ can be computed

in Matlab as `A (:,Q)`. There is an optional input argument that modifies how `colamd` treats rows and columns with many nonzero entries, and an optional output argument that gives statistics on the ordering. The Matlab `colamd` function first calls the `colamd.c` code via the `colamdmex.c` routine, and then computes a column elimination tree post-ordering, via Matlab's built-in `coletree` function.

3 `Symamd`: a symmetric ordering algorithm

A symmetric permutation $\mathbf{PAP}^{\mathsf{T}}$ is typically performed prior to sparse Cholesky factorization to limit the fill-in in $\mathbf{L}$. This can be computed directly (as in [1]) or it can be derived from a column ordering method such as `colamd`. The `symamd` algorithm first constructs a matrix $\mathbf{M}$ such that the nonzero pattern of $\mathbf{M}^{\mathsf{T}}\mathbf{M}$ is the same as the pattern of $\mathbf{A}$, and then orders the columns of $\mathbf{M}$ via `colamd`.

4 Contents of the `colamd` package

The `colamd` package consists of the following files and routines:

1. *Matlab routines and drivers:*

   - `colamd.m`: The Matlab-callable `colamd` routine. This provides help information (via typing `help colamd` inside Matlab). It checks the arguments and calls `colamdmex.c`.
   - `colamdmex.c`: The Matlab mexFunction for interfacing Matlab to the `colamd.c` computational kernel.
   - `symamd.m`: The Matlab-callable `symamd` routine. This provides help information (via `help symamd`). It calls `symamdmex.c`.
   - `symamdmex.c`: The Matlab interface for the `symamd` routine.

2. *C-callable routines and drivers:*

   The following user-callable routines are all located in the file `colamd.c`, with required definitions in `colamd.h`. They contain no Matlab-specific code, so they can be used in any ANSI C environment. The calling sequence of each user-callable routine is fully documented in the `colamd.c` file. These include:

- `colamd_recommended`: Returns the recommended size of the working array used by `colamd`. Also available as a macro `COLAMD_RECOMMENDED` in `colamd.h`.
- `colamd_set_defaults`: Sets default parameters for `colamd` and `symamd`.
- `colamd`: The primary column ordering kernel. The sparse input matrix **A** is in a compressed-column format, such as that used by Matlab.
- `symamd`: The symmetric ordering routine. This sets up a matrix **M** and then calls `colamd`.
- `colamd_report`: Prints the error status and statistics for `colamd`.
- `symamd_report`: Prints the error status and statistics for `symamd`.

3. *Testing environment:*

   Two files, `colamd_demo.m` and `startup.m`, provide a simple demonstration of `colamd` and `symamd` within Matlab. Simply starting Matlab within the `colamd` directory will compile the mexFunctions and run a simple example. A main program in C, `colamd_example.c`, shows how to call `colamd`, and tests its use.

4. *Support files:*

   We provide a `Makefile` for compiling an example main program (`colamd_example.c`) and the Matlab mexFunctions, `colamdmex.c` and `symamdmex.c`.

5 Availability

The `colamd` and `symamd` routines are written in ANSI/ISO C, with Matlab-callable interfaces. Version 2.0 of the code is freely available from the following sources:

1. University of Florida, `http://www.cise.ufl.edu/research/sparse`.
2. Netlib, `http://www.netlib.org/linalg/colamd/`.
3. The MathWorks, Inc., for user-contributed contributions to Matlab, `http://www.mathworks.com`. `Colamd` and `symamd` are built-in functions in Matlab Version 6.0.

4. The collected algorithms of the ACM, as Algorithm 8xx.

References

[1] P. R. Amestoy, T. A. Davis, and I. S. Duff. An approximate minimum degree ordering algorithm. *SIAM J. Matrix Anal. Applic.*, 17(4):886–905, 1996.

[2] T. A. Davis, J. R. Gilbert, S. I. Larimore, and E. G. Ng. A column approximate minimum degree ordering algorithm. Technical Report TR-00-005, Univ. of Florida, CISE Department, Gainesville, FL, October 2000. (submitted to ACM Trans. on Mathematical Software).

[3] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Anal. Applic.*, 20(3):720–755, 1999.

[4] J. R. Gilbert, C. Moler, and R. Schreiber. Sparse matrices in MATLAB: design and implementation. *SIAM J. Matrix Anal. Applic.*, 13(1):333–356, 1992.

[5] J. R. Gilbert and T. Peierls. Sparse partial pivoting in time proportional to arithmetic operations. *SIAM J. Sci. Statist. Comput.*, 9:862–874, 1988.