AMD Version 1.1 User Guide

Patrick R. Amestoy Timothy A. Davis[†] Iain S. Du[‡]

January 29, 2004

Abstract

AMD is a set of routines that implements the approximate minimum degree ordering algo-

1 Overview

AMD is a set of routines for preordering a sparse matrix prior to numerical factorization. It uses an approximate minimum degree ordering algorithm [1] to find a permutation matrix P so that the Cholesky factorization $PAP^T = LL^T$ has fewer (often much fewer) nonzero entries than the Cholesky factorization of A. The algorithm is typically much faster than other ordering methods and minimum degree ordering algorithms that compute an exact degree [3]. Some methods, such as approximate deficiency [8] and graph-partitioning based methods [4, 6, 7, 9] can produce better orderings, depending on the matrix.

The algorithm starts with an undirected graph representation of a symmetric sparse matrix A. Node i in the graph corresponds to row and column i of the matrix, and there is an edge (i, j) in the graph if a_{ij}

integers, where *e* is the number of nonzeros in $A + A^T$. It computes statistics about the matrix A, such as the symmetry of its nonzero pattern, the number of nonzeros in L, and the number of floating-point operations required for Cholesky and LU factorizations (which are returned in the `Info` array). The user's input matrix is not modified. It returns `AMD_OK` if successful, `AMD_INVALID` if the matrix is invalid, or `AMD_OUT_OF_MEMORY` if out of memory.

- `amd_defaults` (long version: `amd_l_defaults`)

```
#include "amd.h"
double Control b3ble ConθNTROL]ledouble
```

6 Using AMD in a Fortran program

The Demo directory contains an example of how the C version may be called from a Fortran program, but this is highly non-portable. For this reason, it is placed in the Demo directory, not in the primary Source directory.

8 Installation

When you compile your program that uses the C-callable AMD library, you need to add the `AMD/Lib/libamd.a` library and you need to tell your compiler to look in the `AMD/Include` directory for include files. To compile a Fortran program that calls the Fortran AMD library, you need to add the `AMD/Lib/libamdf77.a` library. See `AMD/Demo/Makefile` for an example.

If all you want to use is the AMD mexFunction in MATLAB, you can skip the use of the `make` command entirely. Simply type `amd_make` in MATLAB while in the `AMD/MATLAB` directory. This works on any system with MATLAB, including Windows.

If you lo

9 The AMD routines

```
) ;

/* Input arguments (not modified):
```

*

```
*           output order P.
*
*       Info [AMD_MEMORY]: the amount of memory used by AMD, in bytes.  In the
*           current version, this is 1.2 * Info  [AMD_NZ_A_PLUS_AT] + 9*n
*           times the size of an integer.  This is at most 2.4nz + 9n.  This
*           excludes the size of the input arguments Ai, Ap, and P, which have
*           a total size of nz + 2*n + 1 integers.
*
*       Info [AMD_NCMPA]: the number of garbage collections performed.
*
*       Info [AMD_LNZ]: the number of nonzeros in L (excluding the diagonal).
*           This is a slight upper bound because mass elimination is combined
*           with the approximate degree update.  It is a rough upper bound if
```

```
* sort and remove duplicate entries from a matrix.
*
* Allocates 2*n integer work arrays, and free's them when done.
*
* If you wish to call amd_order, but do not know if your matrix has unsorted
* columns or duplicate entries, then you can use the following code, which is
* fairly efficient.  amd_order will not allocate any internal matrix until
* it checks that the input matrix is valid, so the method below is memory-
* efficient as yhnt.is coer sumecks that and ate
```

References

[1] P. R. Amestoy, T. A. Davis, and I. S. Du□. An approximate minimum degree ordering algorithm. *SIAM J. Matrix Anal. Applic.*