EE257/GP 258 Extra Handout LAPACK

Prepared by Ann Chen

Compiling lapack with cmake

Supposing your fortran compiler is called gfortran, you can compile lapack using the following steps:

http://www.netlib.org/lapack/lapack-3.4.2.tgz
tar xvfz lapack-3.4.2.tgz
cd lapack-3.4.2
cmake -DCMAKE_Fortran_COMPILER=gfortran .
make
ls -l lib
The file is called "liblapack.a"

Using lapack from fortran – Matrix inversion

```
program matrixinv
    integer*4 :: ia,ja
    real*4 :: a(4,4),work(4)
    integer*4 :: ipiv(4)
    integer*4 :: info
    ia = 4
    ja = 4
    a = reshape((/ 1,2,3,4,6,7,7,8,9,10,11,10,13,14,15,16/),(/4,4/))

    print *, "a=", a
    call sgetrf(ia,ja,a,ia,ipiv,info)
    call sgetri(min(ia,ja),a,ja,ipiv,work,max(ia,ja),info)
    print *, "ainv=", a
end program matrixinv
```

You can compile your code as:

gfortran -L/path/to/lapack/lib -llapack -lblas test.f90

Note: files it is looking for in /path/to/lapack/lib directory: liblapack.a and libblas.a

Using lapack from C++

Here's an example of using the fortran library from c++. It takes a couple steps -- you have to declare the function prototypes for the functions you want to use (since fortran doesn't use headers, you have to do this manually). You also have to include (possibly) the fortran library. On my system, it is named "libgfortran.so" and it is in /opt/local/lib/gcc45. Compile with:

```
g++ -L/Users/jingyi/lapack-3.4.2/lib -llapack -lblas
-L/opt/local/lib/gcc45 -lgfortran test2.cpp

#include <iostream>

// prototypes for lapack.  Generally the pattern is:
//
//  the name of the function will be the same as in fortran, but lowercase
//
//  the name of the function will have a trailing _ appended
//
//  all arguments to the function are pointers -- even "scalars" in fortran
//  are pointers.
//
extern "C" {
   void sgetrf_( int *M, int *N, float *A, int *LDA, int *IPIV, int *INFO );
   void sgetri_( int *N, float *A, int *LDA, int *IPIV, float *WORK, int *LWORK, int *INFO );
}

int main()
{
  float a[4*4];
  float ainv[4*4];
  float work[4];
  int ipiv[4];
  int ia,ja;
  int info;
  ia = 4;
  ja = 4;
  for( int i=0; i<ia; i++ ) {
    for( int j=0; j<ja; j++ ) {
       a[j*4+i] = (double)std::rand()/(double)RAND_MAX;
    }
  }

std::cout << "a=";
 for( int i=0; i<ia; i++ ) {
   for( int j=0; j<ja; j++ ) {
      std::cout << a[j*4+i] << " ";
```

```
   }
   std::cout << std::endl;
 }
 sgetrf_(&ia,&ja,a,&ia,ipiv,&info);
 sgetri_(&ia,a,&ja,ipiv,work,&ia,&info);
 std::cout << "ainv=";
 for( int i=0; i<ia; i++ ) {
    for( int j=0; j<ja; j++ ) {
       std::cout << a[j*4+i] << " ";
    }
    std::cout << std::endl;
  }
}
```

Please refer to the handout LAPACK summary for more information.