Assignment 1: Linear Solvers

Dominik Dahlem

January 7, 2008

1 General Approach

- GNU autotools is used to maintain this project:
  autoconf version 2.61 automake version 1.10
- For all but the last task getopt is used to parse command-line parameters. A help message can be displayed by specifying "-?" or "-h" to the command-line of the executable.
- The GSL library was chosen to provide the API for the vector and matrix operations. These entail simple operators within the `gsl_vector` and `gsl_matrix` packages, but also more complex operators in the gsl-cblas library. Consequently, the `gsl_vector` and `gsl_matrix` containers are heavily used to specify vectors and matrices in the assignment respectively.
- The cunit library was used to implement the unit tests.

2 Configuration

- The assignment ships with a configure script generated by autotools. The following options are supported:
  - –enable-test : enables testing using cunit. The configure script will check whether the cunit library is present and set a automake variables appropriately.
  - –enable-gcov : enables the coverage analysis using gcov and lcov. The configure script will check whether those libraries is present and set a automake variables appropriately.

3 Make

- Installing the application was not considered.
- To build the project with MPI support do:

1. No gcov, no tests
   ./configure
   make
   ./src/c/main/linsolvmain -f test.dat -s q -e
   Execute the main using the matrix and vector specifications in the test.dat file. The solver being invoked is the QR decomposition and the eigenvalues are calculated as well.
2. No gcov
   ./configure –enable-test
   make check
   This command will compile everything and run the cunit tests.
3. Enable gcov and testing
   ./configure –enable-test –enable-gcov
   make lcov This command will compile everything, run the cunit tests, and generate a snazzy HTML coverage report using lcov [1].

4 Assignment Layout

The assignment contains the octave-code in src/octave and the c-code in src/c. The octave library is packaged into an octave library called linearsolv using the octave-forge conventions for packages. The documentation for the octave code is done with texinfo within the sources following the conventions of octave-forge. However, for that reason the texinfo documentation is not generated, because this package is not part of the octave-forge distribution and I did not copy the build environment over.

```
- src
   - c
   - octave
- report
```

A Doxygen configuration file is provided to generate the code documentation in HTML. doxygen support is integrated into the makefiles. Run: make doxygen-doc

```
- doc
   - doxygen
      - html
```

The generated doxygen report details the inter-relationships between the implemented modules and the source files.

The lcov coverage report is provided in the coverage folder.

[1] http://ltp.sourceforge.net/coverage/lcov.php

5 Remarks on the Approach

The application accepts a number of command-line arguments (see help output) to configure the file with the matrix and vector specifications; the linear solver to be used; whether eigenvalues should be generated; whether the matrix should be generated with a given dimension; and whether the generated matrix with its solution should be written into a file called output.dat.

The command-line arguments are parsed with getopt and verified once they are parsed. The verification makes sure that either a filename or the dimension of a matrix is specified. Further checks include whether the dimension is negative, and that the specified solver exists. Any mistake on the command-line will be reported and the help will be displayed.

The c-sources for the linear solvers commented out the check whether the given matrix A is positive definite, because I found an inconsistency between the GSL library and octave. The `matrix_type` method of octave returned that the matrix in test.dat in the root directory is SPD. However, the GSL library would report an error. I followed Gilbert Strang's checks in his "Introduction to Linear Algebra" book that all upper left determinants are positive.

The eigenvalue solver uses the direct QR decomposition method without any pre-processing step such as tridiagonalisation of the matrix A using the Householder algorithm. Since the QR decomposition can be used as well to solve the linear system, this is an additional method provided which can be specified on the command-line.

Since the matrices and vectors are in-memory representations using the `gsl_vector` and `gsl_matrix` containers the respective sizes are limited by the constraints of the available computer main memory.

I extended the I/O module a little to read/write the matrices A, x, and b. The c module is compatible with the octave text output.