

IRL Installation and Use

Generated by Doxygen 1.8.15

Contents

1 How to Install and Link to IRL

1

Chapter 1

How to Install and Link to IRL

Installing IRL for inclusion and use in your own source code is relatively straight forward. There is only one mandatory external dependency for IRL, the header-only matrix library [Eigen](#).

Additionally, if you would like to help develop IRL you should install [Google Test](#), [GCOVR](#), and [GCOV](#), all of which are detailed below.

Compilation, Installation, and Use of IRL

The compilation of IRL is handled through GNU Make, where all paths and options for compilation are dictated in a `Makefile.in` file. Eventually, CMAKE support will be added.

To get started, `cd` to IRL's root folder, where a `Makefile.in.example` file exists. This is a template which will need to be modified to compile IRL, so first copy this file to the name `Makefile.in`.

Next, the path to the [Eigen](#) root directory needs to be given for the variable `EIGENDIR`. If you do not already have a copy of Eigen, simply go to http://eigen.tuxfamily.org/index.php?title=Main_Page and download the compressed source code using the links on the right side of the page underneath "Get it". You can then `mv` the downloaded source code to a directory you like, uncompress it with `tar -xvf name_of_eigen_file.tar.gz`, and then provide Eigen's root directory for `EIGENDIR` in `Makefile.in`.

Next, the compiler and archiver to use during compilation/linking are given. In the example `Makefile.in`, example, they are given for the GNU compilers, however, IRL should work with other compilers as well as long as C++14 features and Fortran 2003 features are available. The debug flags (`DBGFLAGS`) and optimization flags (`OPTFLAGS`) for the compiler should then be added to the `Makefile.in`. Additional Fortran compiler flags can be supplied as well for debug builds (`FCDBGFLAGS`) and optimized builds (`FCOPTFLAGS`). This is only needed if the Fortran interface is going to be compiled.

At this point, you are ready to compile IRL into a static library and begin using it in your own source code. To do this, in your terminal simply type `make opt` to compile IRL with the optimization flags into a library file, `libirl.a`, which will be placed in `IRL/lib`. If you wish to compile using the debug flags instead, use `make debug`. With the library created, it can now be statically linked to your code by linking to `IRL/lib/libirl.a`. If you are using C or Fortran, please see the details of the [C/Fortran interface in this PDF](#), which discusses the available functions and how to use them.

Unit-Testing and Coverage

If you wish to test your compilation of IRL against its unit-tests or contribute to the development of IRL, it will be necessary to install [Google Test](#).

To install Google Test, clone or download the source code from <https://github.com/google/googletest>. Then in IRL's `Makefile.in`, set the variable `GTEST_DIR` to `/path/to/googletest/googletest/`. You should now be able to run IRL's unit-testing suite by typing `make test` in IRL's home directory. If you wish to run the tests with optimization flags instead of debug flags, use `make opt_test`.

In order to see how much of IRL's source code is covered by the unit-tests, [gcovr](#) will need to be installed. Assuming you have access to a python distribution, this can be done easily according to the [gcovr documentation](#) by using pip:

```
pip install gcovr
```

The path to [gcov](#), the GNU coverage utility that comes with the GCC compiler, then needs to be supplied to the `GCOV` variable in IRL's `Makefile.in`. The path to the newly installed `GCOVR` also needs to be given to `GCOVR` in `Makefile.in`. Lastly, in order for `GCOV` to correctly reference the source files in IRL, the number of forward slashes (/) that exist in the path to `IRL/obj` needs to be included in the variable `NUMBER_OF_SLASHES`.

You now should be able to both run the unit-tests and generate the corresponding test coverage data by typing `make coverage` in the IRL root folder. This will generate coverage reports in html, which can be viewed by opening the file `IRL/tests/coverage/IRL_test.html` in any browser.