Using Standard Tools to Package and Distribute Scientific Software C and Fortran Libraries: a Demonstration with the General Purpose Timing Library (GPTL)

Edward Hartnett 1,2

1 CIRES, University of Colorado, Boulder, CO 80309, USA
2 NOAA/ESRL/GSD, Boulder, CO 80305, USA

Motivation
- Improved portability.
- Reduced maintenance.
- Support shared libraries.
- Support standard build targets.

Plan
1. Create autotools build for C library, alongside existing build system.
2. Create new repository, copy all Fortran code and tests.
3. Create autotools build for Fortran library.
4. Once both new builds are demonstrated to work, delete old build system and remove Fortran code and tests from original repository.
5. Create combined distribution, containing both C and Fortran libraries.

Results
- Build is more portable.
- Build system is less complex.

<table>
<thead>
<tr>
<th>Build System</th>
<th>Files</th>
<th>Lines of Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy</td>
<td>30</td>
<td>4807</td>
</tr>
<tr>
<td>Autotools (combined C/Fortran)</td>
<td>14</td>
<td>593</td>
</tr>
</tbody>
</table>

C Library Build

- **configure.ac**
  - Initialize Autotools

- **bin/Makefile.am**
  - Setup Config.h

- **include/Makefile.am**
  - Include Files

- **src/Makefile.am**
  - Main Makefile

- **test/Makefile.am**
  - Check for x86

Fortran Library Build

- **configure.ac**
  - Initialize Autotools

- **include/Makfile.am**
  - Find Fortran Compiler

- **src/Makfile.am**
  - Set up test code.

- **test/Makfile.am**
  - Set Flags

Combined C/Fortran Build

- **configure.ac**
  - Initialize Autotools

- **src/Makfile.am**
  - Build C Library

- **src/FortranMakefile.am**
  - Build Fortran Library

- **test/Makefile.am**
  - Optional Tests

- **src/Makefile.am**
  - Configure Outputs