GSL: GNU Scientific Library

What is GSL?

The GNU Scientific Library (GSL) is a collection of routines for numerical computing. The routines have been written from scratch by the GSL team in ANSI C.

GSL is free software distributed under the terms of the GNU General Public License.

Functionality and Documentation

GSL provides a modern Applications Programming Interface (API) for C programmers, while allowing wrappers to be written for very high level languages.

The library covers the following areas:

<table>
<thead>
<tr>
<th>Complex Numbers</th>
<th>Roots of Polynomials</th>
<th>Special Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vectors and Matrices</td>
<td>Permutations</td>
<td>Sorting</td>
</tr>
<tr>
<td>BLAS Support</td>
<td>Linear Algebra</td>
<td>Eigensystems</td>
</tr>
<tr>
<td>Fast Fourier Transforms</td>
<td>Quadrature</td>
<td>Random Numbers</td>
</tr>
<tr>
<td>Quasi-Random Sequences</td>
<td>Random Distributions</td>
<td>Statistics</td>
</tr>
<tr>
<td>Histograms</td>
<td>N-Tuples</td>
<td>Monte Carlo Integration</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>Differential Equations</td>
<td>Interpolation</td>
</tr>
<tr>
<td>Numerical Differentiation</td>
<td>Chebyshev Approximation</td>
<td>Series Acceleration</td>
</tr>
<tr>
<td>Discrete Hankel Transforms</td>
<td>Root-Finding</td>
<td>Minimization</td>
</tr>
<tr>
<td>Large Least squares</td>
<td>running statistics</td>
<td>spline linear algebra</td>
</tr>
<tr>
<td>spline matrix routines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Least-Squares Fitting</td>
<td>Physical Constants</td>
<td>IEEE Floating-Point</td>
</tr>
</tbody>
</table>

Further information can be found in the GSL Reference Manual (version 1.16, version 2.4, or version 2.3, or online HTML for the current version). Important Note: the 1.x and 2.x releases have significantly different APIs and are therefore source and binary incompatible.

Please also refer to the GSL Home Page for downloads, design documentation and up-to-date releases of the Library.

Usage of GSL on LRZ computing platforms

Accessing the GSL installations

Access to the GSL installations is provided via the environment module gsl:

```
module load gsl
```

This will suitably set all required environment variables. The include files are accessed via the GSL_INC variable and should be referenced as

```
#include <gsl/gsl_foo.h>
```

in your source files (where "foo" is a placeholder for the various functionality areas covered by GSL). Hence, compilation is done via

```
[gcc|icc|icpc] <options> $GSL_INC foo.c
```

static linkage via

```
[gcc|icc|icpc] <options> myprog <objects> $GSL_LIB $GSL_BLAS_LIB
```

and dynamic linkage via

```
[gcc|icc|icpc] <options> myprog <objects> $GSL_SHLIB $GSL_BLAS_SHLIB
```

The GSL CBLAS implementation can be replaced by $MKL_LIB or $MKL_SHLIB on all Intel-based Linux systems after loading the module mkl for improved performance of the Intel Math Kernel Library.
The Fortran interface is available on all HPC systems and can be used by performing

```
module load mkl
module load gsl
module load fgsl
```

To access the module information file for FGSL any source code referencing routines must have a `use fgsl` statement; also note that

- any name beginning with `gsl_` from the C API must be replaced by the corresponding `fgsl_` name, and
- there is no need to differentiate between pointers and passed-by-value since this is properly attended to by the wrapper.

Compiling and linking then proceeds as follows:

```
ifort $FGSL_INC -c foo.f90
...
ifort $FGSL_INC -c myprog.f90
ifort -o myprog.exe foo.o ... myprog.o $FGSL_LIB $GSL_LIB $MKL_LIB
```

Further Information

Please consult the GSL examples for how to use GSL from C, and the FGSL examples for how to use FGSL from Fortran. These are contained in the source packages for GSL and FGSL, respectively.

The source code for the Fortran GSL interface is available from the LRZ web server.