

GSL on HPC

What is GSL?

The GNU Scientific Library (GSL) is a collection of routines for numerical computing. The routines have been written from scratch in C and present a modern Applications Programming Interface (API) for C programmers, allowing wrappers to be written for very high-level languages. The source code is distributed under the GNU General Public License. [\[Source\]](#)

Links:

[Official Website](#)

[Documentation](#)

Versions Available:

The following versions are available on the cluster:

- GSL 1.15
- GSL 2.4
- GSL 2.7.1

How to load GSL?

To load software, use the following commands:

```
#Load the GSL module  
module load gsl/2.7.1
```

To verify if the module and dependencies are loaded correctly, use the following command.

```
#Show all the modules loaded
module list
```

This should list all the dependencies that are loaded. In this case, only GSL will be loaded since it is a standalone library.

How to use GSL?

The numerical library can be used on the following cases,

Complex Numbers	Roots of Polynomials	Special Functions
Vectors and Matrices	Permutations	Combinations
Sorting	BLAS Support	Linear Algebra
CBLAS Library	Fast Fourier Transforms	Eigensystems
Random Numbers	Quadrature	Random Distributions
Quasi-Random Sequences	Histograms	Statistics
Monte Carlo Integration	N-Tuples	Differential Equations
Simulated Annealing	Numerical Differentiation	Interpolation
Series Acceleration	Chebyshev Approximations	Root-Finding
Discrete Hankel Transforms	Least-Squares Fitting	Minimization
IEEE Floating-Point	Physical Constants	Basis Splines
Wavelets	Sparse BLAS Support	Sparse Linear Algebra

To demonstrate the use of the GSL library, use the following code snippet:

```
#include <stdio.h>
#include <gsl/gsl_linalg.h>

int
main (void)
{

double a_data[] = { 0.18, 0.60, 0.57, 0.96,
0.41, 0.24, 0.99, 0.58,
0.14, 0.30, 0.97, 0.66,
```

```

0.51, 0.13, 0.19, 0.85 };

double b_data[] = { 1.0, 2.0, 3.0, 4.0 };

#Solving a matrix system AX = B using gsl
gsl_matrix_view m = gsl_matrix_view_array (a_data, 4, 4);
gsl_vector_view b = gsl_vector_view_array (b_data, 4);
gsl_vector *x = gsl_vector_alloc (4);

int s;

gsl_permutation * p = gsl_permutation_alloc (4);

gsl_linalg_LU_decomp (&m.matrix, p, &s);

gsl_linalg_LU_solve (&m.matrix, p, &b.vector, x);

printf ("x = \n");

gsl_vector_fprintf (stdout, x, "%g");

gsl_permutation_free (p);

gsl_vector_free (x);

return 0;
}

```

Copy and paste the code in a test.c file, and load a gcc compiler (11 version) along with gsl module. After loading, the shared library will be in the cpath and to link the shared library, use the following compilation syntax.

```

# Compile and link
gcc test.c -lgsl -lm

```

Execute to get the result, in this case

```

x =
-4.05205
-12.6056
1.66091
8.69377

```

See more about the linking process and library itself in the gsl documentation.

Where to find help?

If you are stuck on some part or need help at any point, please contact OIT at the following address.

<https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp>