GNU Scientific Library (GSL)

GSL is the go-to library for programs involving deep mathematical expressions.

We'll cover the following

The GNU Scientific Library provides a broad range of numerical and mathematical functions. If you want to do something in C that you have been doing in MATLAB, or Python/NumPy/SciPy, or R, then probably you will need to look at something like GSL for functions you need (or code them up yourself).

There is a list of main categories here. Things you might be interested are:

- Vectors and Matrices
- Linear Algebra
- FFTs
- Random Number Generation & Random Number Distributions
- Statistics
- ODEs
- Interpolation
- Numerical Differentiation
- Least-Squares Fitting
- Nonlinear Least-Squares Fitting
- Multidimensional Minimization

Linking to GSL

Of course before you can link to the gsl library, you have to make sure it is installed on your computer. See the gsl documentation for how to install it.

To link your code to the gsl you will have to do two things. First you will have to use the `#include` directive to include the relevant header files. In the gsl documentation for each function, it tells you which header file(s) you will need

documentation for each function, it tells you which header file(s) you will need. Second, you will have to use a flag when compiling to tell the compiler to include the library files themselves. Here is an example program from the gsl documentation that uses linear algebra routines to solve a linear system $A\_\_x = b$:

```
[ 0.18 0.60 0.57 0.96 ] [x0]   [1.0]
[ 0.41 0.24 0.99 0.58 ] [x1] = [2.0]
[ 0.14 0.30 0.97 0.66 ] [x2]   [3.0]
[ 0.51 0.13 0.19 0.85 ] [x3]   [4.0]
```

```
// gcc -o go go.c -lgsl -lgslcblas

#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 };

  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;
}
```

```
plg@wildebeest:~$ gcc -o go go.c -lgsl -lgslcblas
plg@wildebeest:~$ ./go
x =
-4.05205
-12.6056
1.66091
8.69377
```

After GSL, we have the BLAS/LAPACK libraries for algebraic computations.