Scientific C-Libraries

GNU Scientific Library - GSL

GNU have released their C Scientific Library: http://www.gnu.org/software/gsl/

It is recommended to use GSL on a free operating system such as GNU/Linux

However, a compiled version of GSL can be obtained as part of Cygwin on Windows: http://gnuwin32.sourceforge.net/packages/gsl.htm

Licensing: “GSL can be used internally (‘in-house’) without restriction, but only redistributed in other software that is under the GNU GPL”

Example

```
#include <stdio.h>
#include <assert.h>
#include <gsl/gsl_sf_bessel.h>
int main() {
        double x = 0.0;
        printf("Enter x : ");
        assert( scanf("%lf", &x) == 1 );
        printf("J0(%g) = %.18e\n", x, gsl_sf_bessel_J0(x));
        return 0;
}
```

Compile this using:

```
gcc gsl_example.c -lgsl -lgslcblas -lm -o gsl_example
```

NAGLIB and Netlib

NAGLIB

The Numerical Algorithms Group, based in Oxford, maintain a software library called NAGLIB. They have a 30 day trial (check the NAG Software License Agreement).

- Certain departments in Imperial College have a license for this.
- Check with the Software Shop; you might be covered.

NETLIB

The Netlib repository contains a lot of very useful general numerical code and papers:

`www.netlib.org`

There are two general purpose libraries you should be aware of if you want to write reasonably portable C that requires access to lower-level operating system functionality or require more functionality than the standard library provides:

APR The Apache Portable Runtime[1] provides a consistent interface to many OS-specific functions (e.g. threads, networking, opening libraries at runtime) and other useful tools (e.g. memory pools to simplify memory management and hash tables).

GLib GLib[2] is similar to APR, but has a stronger focus on user-level applications. It contains functionality for character conversion, regular expressions, multiple container types, dynamically sized strings and more.