

The ISML C Library Tutorial



Updated: August 2012

Table of Contents

Section 1: Introduction.....	3
Section 2: Getting Started on a Unix Machine	3

Section 1: Introduction

The IMSL library contains over one thousand Fortran subroutines and functions. These routines provide easy access to high quality implementations of numerical methods in mathematics and statistics.

Mathematical routines include: linear algebra, eigensystem analysis, interpolation and approximation, integration and differentiation, solving differential equations, Fourier and Laplace transforms, finding solutions of nonlinear equations, optimization, and special functions.

Statistical routines include: regression correlation, analysis of variance, categorical and discrete data analysis, nonparametric statistics, tests of goodness of fit and randomness, time series analysis and forecasting, covariance structures and factor analysis, discriminant analysis, cluster analysis, sampling, survival analysis, life testing, and reliability, multidimensional scaling, density and hazard estimation, probability distribution functions and inverses, and random number generation.

The IMSL subroutines are called like ordinary Fortran subroutines, then the IMSL libraries are linked into the executable when the calling program is loaded. As discussed below, the method of linking depends on the system you are using.

IMSL is proprietary software and cannot be transferred to a machine that is not licensed to run it.

Section 2: Getting Started on a Unix Machine

The IMSL libraries are not in the default library search path. There is a script which will modify your environment to allow you to call the IMSL libraries, installed on the ITS systems. To run this script type:

```
source /usr/local/imsl/CTT6.0/ctt/bin/cttsetup.csh
```

The command above can be put at the end of your `.cshrc` (or `.profile`) so that it will be executed automatically for you. It is important that you issue this command only after you have set your environment variables such as your `PATH` and `MANPATH`. Once you have executed the script, use:

```
cc [compiler options] $CFLAGS filename $LINK_CNL
```

to compile and load the program in *filename* using the default (shared) version of the IMSL library. It is also possible to explicitly specify the use of the static or shared library. To load the static or shared versions of the IMSL Fortran numeric libraries, respectively, use:

```
cc [compiler options] $CFLAGS filename $LINK_CNL_STATIC
```

or

```
cc [compiler options] $CFLAGS filename $LINK_CNL_SHARED
```

To generate an executable `eigentest` that uses the shared version of the IMSL libraries, compile the example file `eigentest.c` with:

```
cc -o eigentest $CFLAGS eigentest.c $LINK_CNL
```

An example program

You may be using a completely different machine or operating system, but this example might still prove to be helpful. The following program uses the IMSL routines `imsl_f_eig_sym` and `imsl_f_write_matrix`. The discussion below assumes that the contents of the file `eigentest.c` are:

```
#include <imsl.h>

main()
{
    int          n = 3;
    float        a[] = {7.0, -8.0, -8.0,
                       -8.0, -16.0, -18.0,
                       -8.0, -18.0, 13.0};

    float        *eval;
                                /* Compute eigenvalues */
    eval = imsl_f_eig_sym(n, a, 0);
                                /* Print eigenvalues */
    imsl_f_write_matrix ("Eigenvalues", 1, 3, eval, 0);
}
```