**How to compile, link, and use ARPACK in Windows with Microsoft Visual Studio (MSVC)**

I spent a few days trying to make this work, and I eventually succeeded, so I'd like to share the solution with others. The solution was developed with MSVC .NET 2003 and Windows XP. ARPACK is such a great package (robust, powerful, fast) that it is a shame not to use it just because of cross-platform issues.

I used [MinGW](http://www.mingw.org/), a very nice free port of many Unix utilities to the Windows platform. Please see the MinGW webpage for more accurate descriptions of this software.

The same solution was first described [here](http://mithandor.googlepages.com/) (many thanks to David Gleich, the author of that page). The purpose of my page is to explain the whole process in more detail. Some related information is also available on this [MinGW page](http://www.mingw.org/mingwfaq.shtml#faq-msvcdll).

**Step 1: Get the ARPACK Fortran** [**source files.**](http://www.caam.rice.edu/software/ARPACK/)

You can also download ARPACK++, a collection of Fortran-to-C++ wrappers which allow you to call ARPACK routines using an easy C++ interface.

**Step 2: Install MINGW and MSYS**

Install [MinGW](http://www.mingw.org/) (I installed most packages) and then MSYS. MSYS is a Unix-like shell for Windows (similar to Cygwin, but more unrestricted in terms of licensing). It is available on the MinGW webpage. After you install MSYS, there will be a VERY IMPORTANT post-installation step (a script) where it will sycnhronize MinGW and MSYS. I got confused the first time around and it did not get done properly... This resulted in my gcc.exe compiler (and make utility) not working in a very strange way, complaining about not receiving input files where in reality input files were specified. I lost two days trying to fix it, until I realized I simply needed to reinstall.

Also: on the MinGW download page, there is talk about a package that combines most of MinGW for new users, but it was not directly obvious to me that MSYS is this package (I am still not 100% sure). Just giving feedback, I think MinGW & MSYS are great tools.

**Step 3: Compile ARPACK using MinGW's Fortran, in the MSYS shell**

In this step, we will compile ARPACK and create a Windows DLL. You can then link against this DLL from MSVC.

Start a MSYS shell, go into your code directory (in this case your ARPACK directory) and compile the Fortran source files (such as using "make" which will call g77, the MinGW's Fortran compiler). Next, wrap the \*.o files into a DLL:

dllwrap --export-all-symbols BLAS/\*.o LAPACK/\*.o SRC/\*.o UTIL/\*.o -lg2c --output-def arpack\_win32.def -o arpack\_win32.dll

Note that libg2c.a is located in the MinGW's lib directory if you installed MinGW's g77. I used the precompiled version. Also note that you might receive the following warning message. You can safely ignore it:

dllwrap.exe: no export definition file provided.

Creating one, but that may not be what you want

Finally, you need to create an import library for arpack\_win32.dll:

lib /machine:i386 /def:arpack\_win32.def

This will generate a file called "arpack\_win32.lib". This is an import library for arpack\_win32.dll. You need both arpack\_win32.lib and arpack\_win32.dll to link (from MSVC) and run your programs.

Note: I ran all commands in this Step 3 from the MSYS shell.

Note: "lib.exe" is a part of MS Visual Studio. It is usually located in the MS Visual Studio binary directory, in the same directory that also contains the MSVC compiler executable and linker (on my system, the path is C:\Program Files\Microsoft Visual Studio .NET 2003\Vc7\bin). You need to add this directory to the path, or else you will have to type the full path to lib.exe each time you call it in MSYS.

Note: In my case, "lib.exe" at first didn't want to run due to some missing MSVC DLL, which I then found in a MSVC directory and copied it over to the local directory. I am sure there are better solutions too.

**Step 4 (optional): Compile ARPACK++**

With some minor modifications of ARPACK++ header files (I remember I made essentially three small modications, e.g. replacing <iostream.h> with <iostream> and similar), I was able to make these header files compile under MSVC.

**Step 5: Link against ARPACK from MSVC**

You simply need to specify arpack\_win32.lib on the list of libraries in MSVC (either on the library list specific to a particular MSVC project, or on the global MSVC library list), and it should link just fine. When executing your program, you will need arpack\_win32.dll (but not arpack\_win32.lib). Again, I solve this by copying arpack\_win32.dll into the local program directory, but it could go into some canonical place for DLLs.

So, you are "stuck" with having to include arpack\_win32.dll with your programs. I initially tried to make a static library (from Fortran-produced \*.o files) in MinGW and then link against it from MSVC, but did not succeed. So, I don't know of a way to avoid using the dll, but it's not critical for me. DLLs have the advantage that they can be used in many programs simultaneously, saving memory/disk space. Now, "extra credit" would be to make ARPACK link against Intel Math Kernel library as opposed to the standard BLAS and LAPACK that come with ARPACK... (I have not attempted such a Intel MKL linkage yet).