

1	INSTALLATION UNDER WINDOWS.....	1
1.1	MODULES	1
1.1.1	What is a module?	1
1.1.2	Modules in GALAHAD	2
1.2	LIBRARIES	3
1.3	COMPILING	4

1 INSTALLATION UNDER WINDOWS

Indeed just copying all the files into a Windows does not make LANCELOT B run under Windows. Modules and archives have to be compiled by a Windows Fortran compiler first. The compiler used was VISUAL DIGITAL Fortran v6 under win2000.

1.1 Modules

1.1.1 What is a module?

Fortran 90 has a new kind of program unit, the module, that is neither part of Fortran 77 nor included in most pre-90 Fortran compilers.

Unlike main programs and external subprograms, modules are not themselves executable program units. Rather, they contain definitions that can be conveniently accessed and used by executable program units. For example, a module might contain interface blocks for a library of external procedures and be used to make the interfaces of these library procedures explicit in an application using that library.

All required facilities (for example, definitions, objects, operators and manipulation procedures) may be packaged together in a MODULE, the programmer simply has to USE this module to have all the features instantly available. The module can now be used like a library unit.

1.1.2 Modules in GALAHAD

The first thing to do is uncompress the tarball file into a directory, galahad for example.

Then a little code changment is needed in cputim.f90 which is locates at galahad/src/aux/cputim.f90

I removed the "!" in front of the following lines :

```
! For general Unix etime (Linux, DEC/Compaq OSF/Ultrix/Tru64, HP, SUN, SGI ...)  
!GEN REAL ( KIND = KIND( 1.0E0 ) ) :: ETIME, DUMMY( 2 )  
!GEN EXTERNAL ETIME  
!GEN time = ETIME( DUMMY )
```

Next step was to generate the appropriate modules. The command to use on the command console is **f90 -c FOO.f90**. Here is the list of files to compile in the same order to get the necessary modules :

- | | | |
|-----------------|-------------|------------------|
| 1. extend.f90 | | 15. cg.f90 |
| 2. cputim.f90 | | 16. hslint.f90 |
| 3. bndsl.f90 | | 17. asmb1.f90 |
| 4. others.f90 | | 18. blas.f90 |
| 5. symbols.f90 | | 19. scu.f90 |
| 6. specfile.f90 | | 20. hsprd.f90 |
| 7. icfs.f90 | | 21. initw.f90 |
| 8. ma61d.f | HSL routine | 22. mdchl.f90 |
| 9. zd01d.f90 | " | 23. precn.f90 |
| 10. fd05d.f | " | 24. frntl.f90 |
| 11. mc71d.f90 | " | 25. strutr.f90 |
| 12. ma57d.f90 | " | 26. lancetot.f90 |
| 13. sort.f90 | | |
| 14. cauchy.f90 | | |

All those files are in galahad/src . The best way to find them is to use Windows' Search tool.

Once those files compiled, I got each time two files: one .mod (the module) and one .obj (the object file). All the .mod files were moved to directory galahad/modules.

1.2 Libraries

An archive (a library) is a single file holding a collection of other files in a structure that makes it possible to retrieve the original individual files (called members of the archive).

The libraries (archives) used by GALAHAD are libgalahad.a, libgalahad_blas.a and libgalahad_hsl.a . Under Linux the command used is **ar**. The GNU ar program creates, modifies, and extracts from archives.

In GALAHAD the archives are made of object files as we can see :

```
[dilipan@gatekeeper double]$ ar -t libgalahad.a
extend.o , cputim.o , bndsl.o , others.o , symbols.o , specfile.o , icfs.o ,
smt.o , sils.o , sort.o , cauchy.o
cg.o , hslint.o , asmb1.o , scu.o , hsprd.o , initw.o , mdchl.o , precn.o ,
frntl.o , strutr.o , lancelot.o
scaln.o , drchg.o , drche.o , rand.o , uselanb.o

[dilipan@gatekeeper double]$ ar -t libgalahad_blas.a
blas.o

[dilipan@gatekeeper double]$ ar -t libgalahad_hsl.a
ma61d.o
ma27d.o
ad01d.o
ad02d.o
kb07ai.o

[dilipan@gatekeeper double]$
```

The equivalent files had to be done for Windows. DIGITAL Fortran has a command called **lib** which provides the same result than **ar**. After placing all the .obj you want to add in the same directory one just need to type:

```
lib *.obj /out:libgalahad.lib
```

And the library compatible with Windows is done.

1.3 Compiling

The hardest is done. I found that the equivalent command to compile was:

```
f90 /exe:lancelot LANCELOT_example.f90
    /module:"V:\galahad\modules"
    /include:"V:\galahad\libraries"
    /libs lancelot.lib
```

If you want to compile with the GUI of DIGITAL Visual Fortran you need to :

- open a new Fortran project
- add your f90 file which uses LANCELOT B to your project
- add the GALAHAD libraries in the project settings
- compile
- execute

To test if the porting from Windows worked I tried the sample code under Windows. At first I had trouble because of the difference of coding the apostrophe (') between Linux and Windows. Such a small reason takes so much time to be found!

Then there was a memory trouble. The allocation of two vectors did not work under Windows. I changed the way they were allocated and the sample program compiled and worked perfectly, giving the same result than under Linux. As we can see below :

```
C:\galahad\code> f90 /exe:lancelot LANCELOT_example.f90
/module:"V:\galahad\modules"
/include:"V:\galahad\libraries"
/libs lancelot.lib
C:\galahad\code> lancelot
14 iterations. Optimal objective value = -6.3129E-01
Optimal solution = 2.4402E-01 -4.0741E-01 1.0000E+00
```