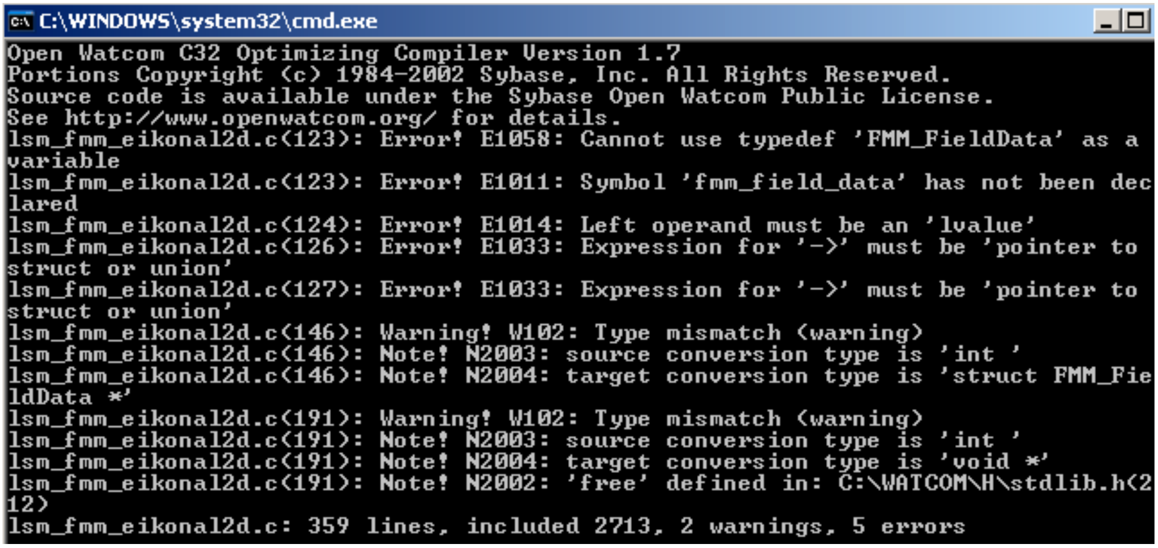
LSMLIB for Windows with MINGW and MSYS

It is difficult to choose environment/compiler to compile LSMLIB (http://www.princeton.edu/~ktchu/software/lsmlib/) for Windows PCs. I have used Open Watcom C/C++ and FORTRAN compiler (http://www.openwatcom.org/) and successfully compiled LSMLIB for Windows and used it in my research. But when I tried Watcom on the most recent LSMLIB (v0.9.0), I got errors like



```
Open Watcom C32 Optimizing Compiler Version 1.7
Portions Copyright (c) 1984-2002 Sybase, Inc. All Rights Reserved.
Source code is available under the Sybase Open Watcom Public License.
See http://www.openwatcom.org/ for details.
lsm_fmm_eikonal2d.c(123): Error! E1058: Cannot use typedef 'FMM_FieldData' as a
variable
lsm_fmm_eikonal2d.c(123): Error! E1011: Symbol 'fmm_field_data' has not been dec
lared
lsm_fmm_eikonal2d.c(124): Error! E1014: Left operand must be an 'lvalue'
lsm_fmm_eikonal2d.c(126): Error! E1033: Expression for '->' must be 'pointer to
struct or union'
lsm_fmm_eikonal2d.c(127): Error! E1033: Expression for '->' must be 'pointer to
struct or union'
lsm_fmm_eikonal2d.c(146): Warning! W102: Type mismatch (warning)
lsm_fmm_eikonal2d.c(146): Note! N2003: source conversion type is 'int '
lsm_fmm_eikonal2d.c(146): Note! N2004: target conversion type is 'struct FMM_Fie
ldData *'
lsm_fmm_eikonal2d.c(191): Warning! W102: Type mismatch (warning)
lsm_fmm_eikonal2d.c(191): Note! N2003: source conversion type is 'int '
lsm_fmm_eikonal2d.c(191): Note! N2004: target conversion type is 'void *'
lsm_fmm_eikonal2d.c(191): Note! N2002: 'free' defined in: C:\WATCOM\H\stdlib.h(2
12)
lsm_fmm_eikonal2d.c: 359 lines, included 2713, 2 warnings, 5 errors
```

I still believe that these errors are easy to fix. Since LSMLIB uses standard C and FORTRAN programming, Watcom should be able to make it running. On the other hand, dealing with all these compiling details is kind of waste of your time and it might be easier to just switch the environment/compiler you are working with. If you would, you may want to try Linux systems such as Debian with gcc/g++/g77 (http://www.debian.org/). The compiling and installing of the compiled library usually takes two simple steps (1) configure (2) make. Then you can focus on writing your own programs to use LSMLIB instead of spending time on compiling LSMLIB itself.

You can still use Windows while working in a Linux environment at the same time by using virtual machine software such as Virtual Box (http://www.virtualbox.org/).

Cygwin (http://www.cygwin.com/) is a choice if you do not mind distributing your application with Cygwin libraries.

I would recommend using MINGW (http://www.mingw.org/) and MSYS together to compile LSMLIB for Windows. In plain words, MINGW is the Windows version of Linux kernel for developers; MSYS is the Windows version of Linux Shell. By combining the two together, you will be able to compile LSMLIB and generate Windows native executable files without the need of relying on third-party library as is the case for Cygwin. The major drawback of this approach, of course, is to shift your development environment to MINGW and MSYS and your compiler would be

gcc/g++/g77. It is also possible to use Watcom compiler (and other compilers) inside MSYS, but that would not solve the compiling error problem mentioned above.

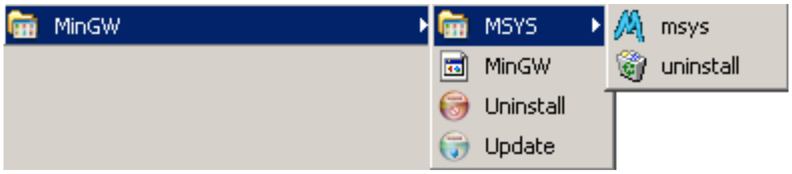
Now I am going to give details on setting up MINGW and MSYS and compiling LSMLIB for Windows using the two.

Step 1: You need to download the two installation files: 1. MinGW-5.1.3.exe, 2. MSYS-1.0.10.exe from http://sourceforge.net/project/showfiles.php?group_id=2435.

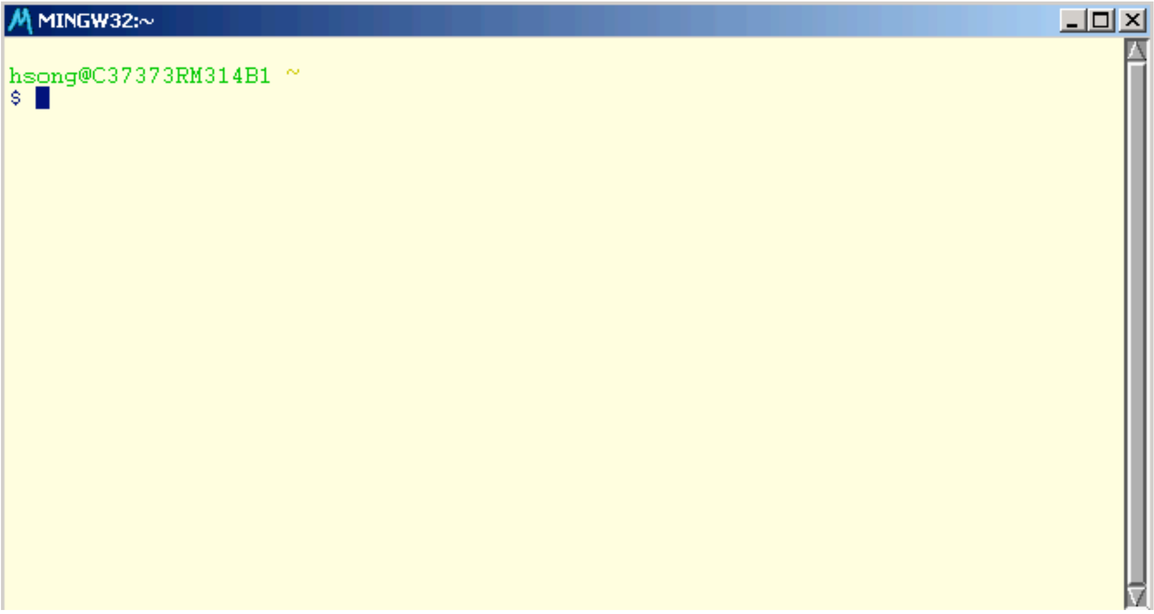
Step 2: Install MinGW first. Note that MinGW-5.1.3.exe will need to download files from Internet. Read and follow on-screen instructions.

Step 3: Install MSYS. Read and follow on-screen instructions.

You should be able to find the following in your start menu



Step 4: Click **, and you will see the following screen**

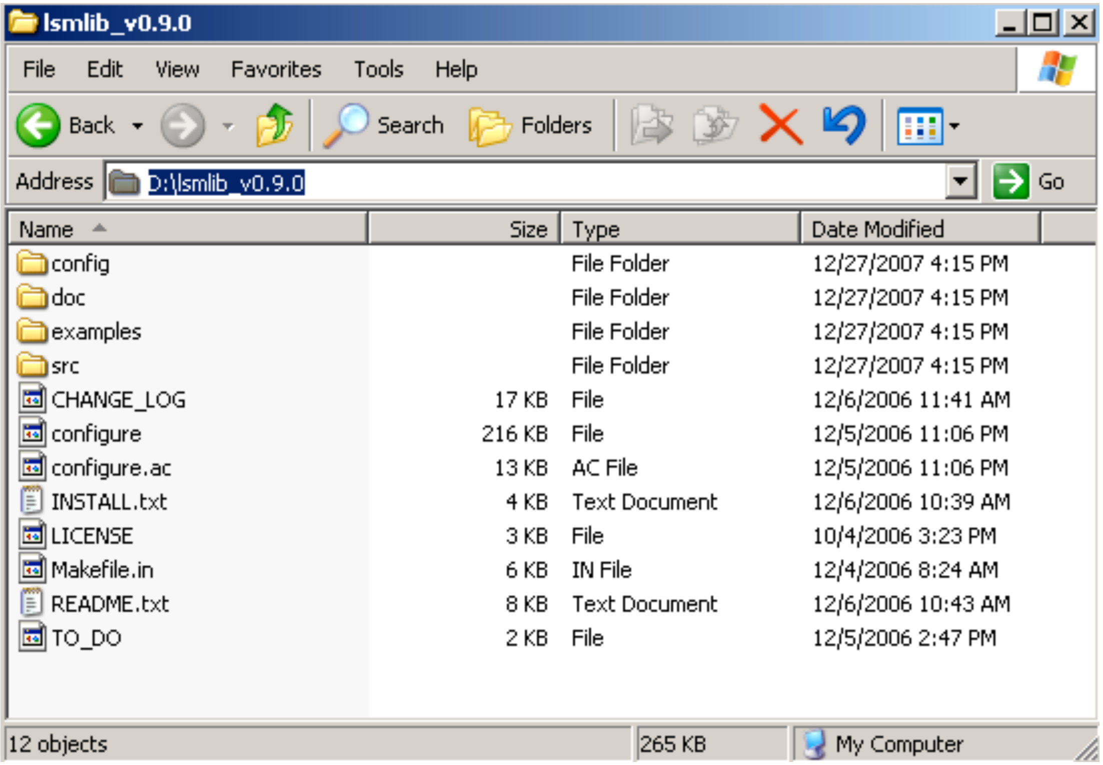


```
hsong@C37373RM314B1 ~
$
```

Now you are in the MSYS environment with MinGW.

Step 5: Download LSMLIB (http://www.princeton.edu/~ktchu/software/lsmlib/) and save the zip file (lsmlib_v0.9.0.zip) to disk.

Step 6: Decompress the zip file using WinZip or other software to a folder, for example, “D:\lsmlib_v0.9.0”. Now you should have the following folder structure

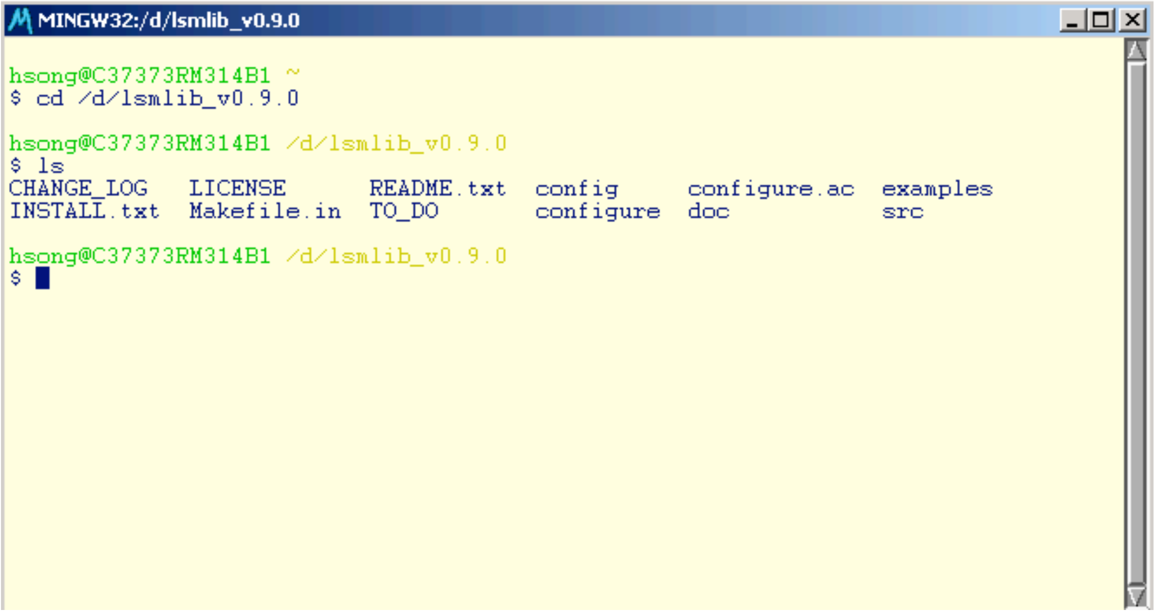


Address: D:\lsmlib_v0.9.0

Name	Size	Type	Date Modified
config		File Folder	12/27/2007 4:15 PM
doc		File Folder	12/27/2007 4:15 PM
examples		File Folder	12/27/2007 4:15 PM
src		File Folder	12/27/2007 4:15 PM
CHANGE_LOG	17 KB	File	12/6/2006 11:41 AM
configure	216 KB	File	12/5/2006 11:06 PM
configure.ac	13 KB	AC File	12/5/2006 11:06 PM
INSTALL.txt	4 KB	Text Document	12/6/2006 10:39 AM
LICENSE	3 KB	File	10/4/2006 3:23 PM
Makefile.in	6 KB	IN File	12/4/2006 8:24 AM
README.txt	8 KB	Text Document	12/6/2006 10:43 AM
TO_DO	2 KB	File	12/5/2006 2:47 PM

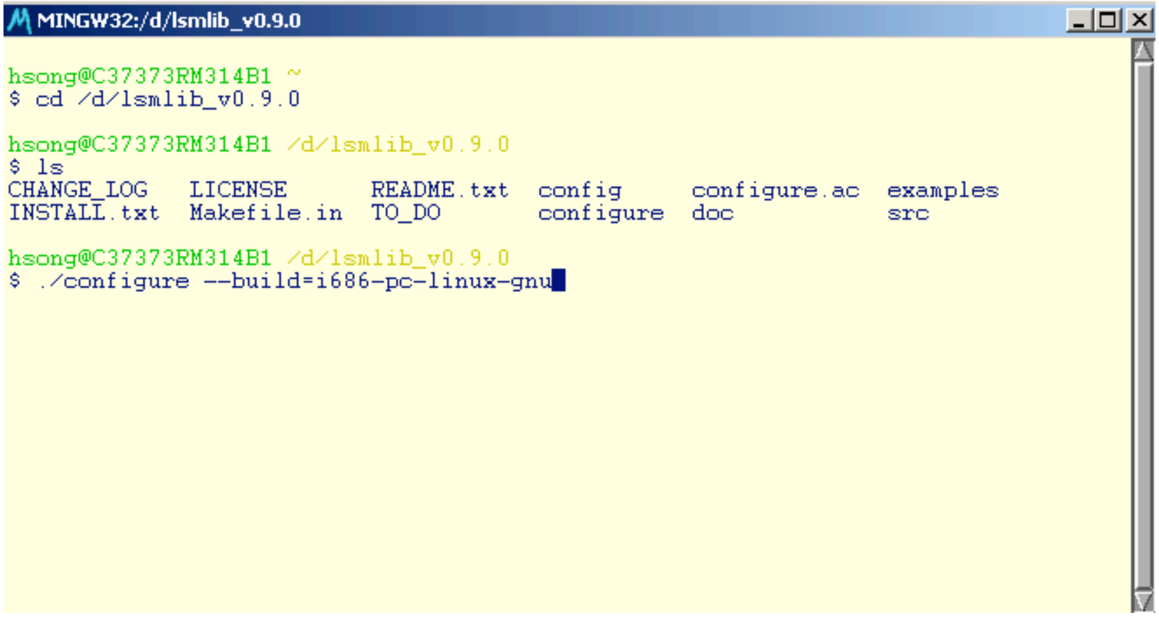
12 objects | 265 KB | My Computer

Step 7: Change current path in MSYS to the LSMLIB folder.



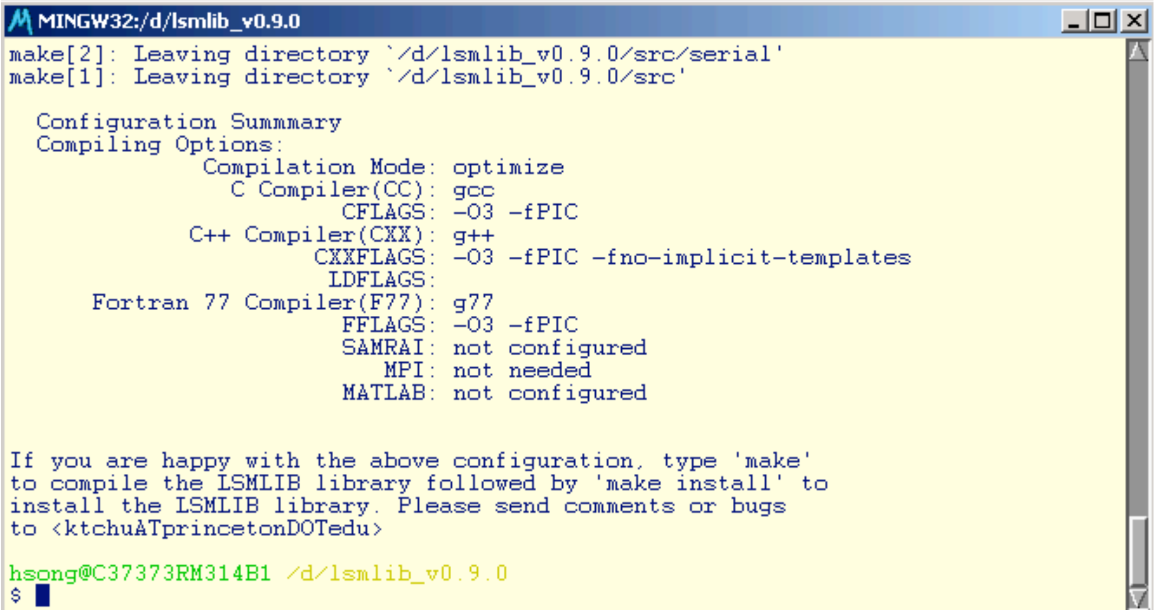
```
hsong@C37373RM314B1 ~
$ cd /d/lsmlib_v0.9.0

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$ ls
CHANGE_LOG   LICENSE      README.txt  config     configure.ac  examples
INSTALL.txt  Makefile.in  TO_DO       configure  doc           src

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$
```

Step 8: Run “configure” to generate Makefiles.



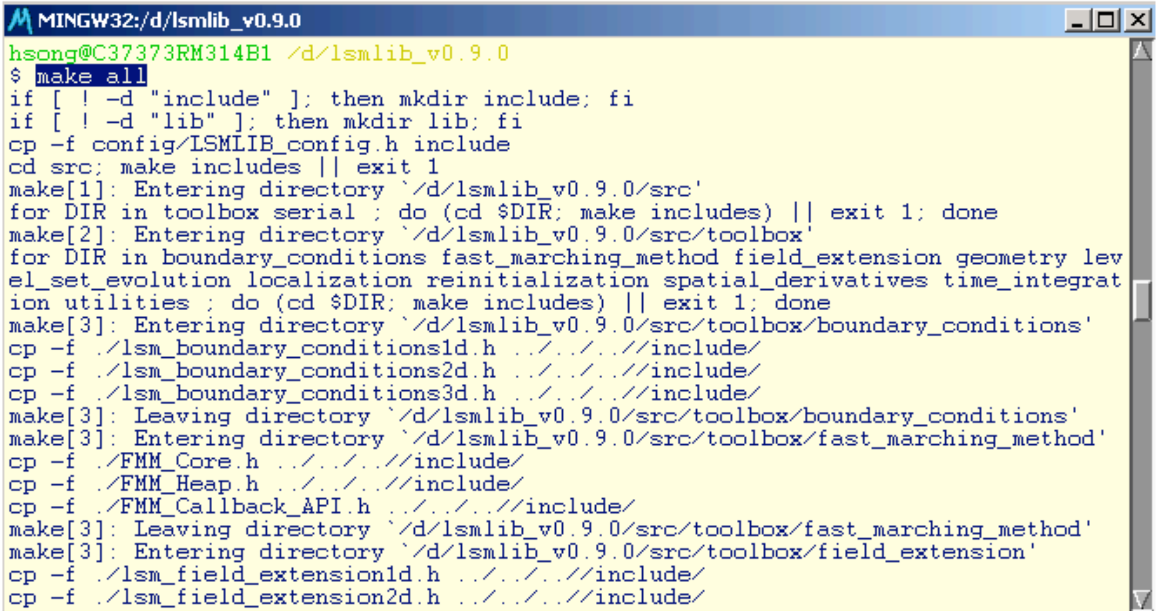
```
hsong@C37373RM314B1 ~
$ cd /d/lsmlib_v0.9.0

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$ ls
CHANGE_LOG   LICENSE      README.txt  config     configure.ac  examples
INSTALL.txt  Makefile.in  TO_DO       configure  doc           src

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$ ./configure --build=i686-pc-linux-gnu
```

You should see the following screen after a successful configure.



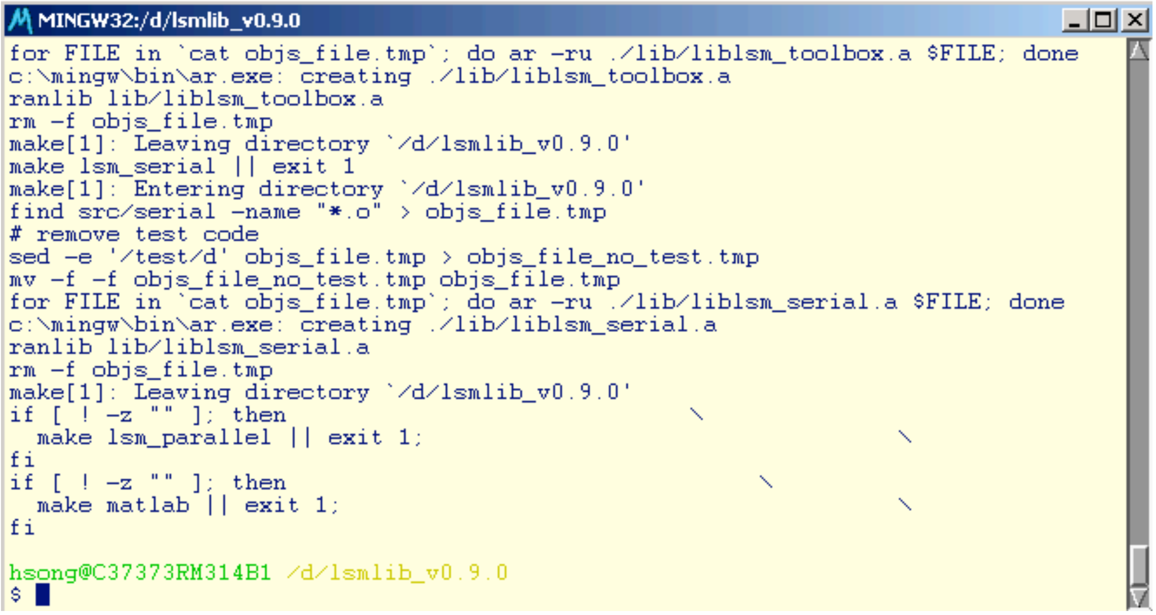
```
make[2]: Leaving directory `/d/lsmlib_v0.9.0/src/serial'
make[1]: Leaving directory `/d/lsmlib_v0.9.0/src'

  Configuration Summary
  Compiling Options:
               Compilation Mode: optimize
                C Compiler(CC): gcc
                        CFLAGS: -O3 -fPIC
              C++ Compiler(CXX): g++
                      CXXFLAGS: -O3 -fPIC -fno-implicit-templates
                       LDFLAGS:
      Fortran 77 Compiler(F77): g77
                        FFLAGS: -O3 -fPIC
                        SAMRAI: not configured
                           MPI: not needed
                        MATLAB: not configured


If you are happy with the above configuration, type 'make'
to compile the LSMLIB library followed by 'make install' to
install the LSMLIB library. Please send comments or bugs
to <ktchuATprincetonDOTedu>

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$
```

Step 9: Run “make” to execute the Makefiles.



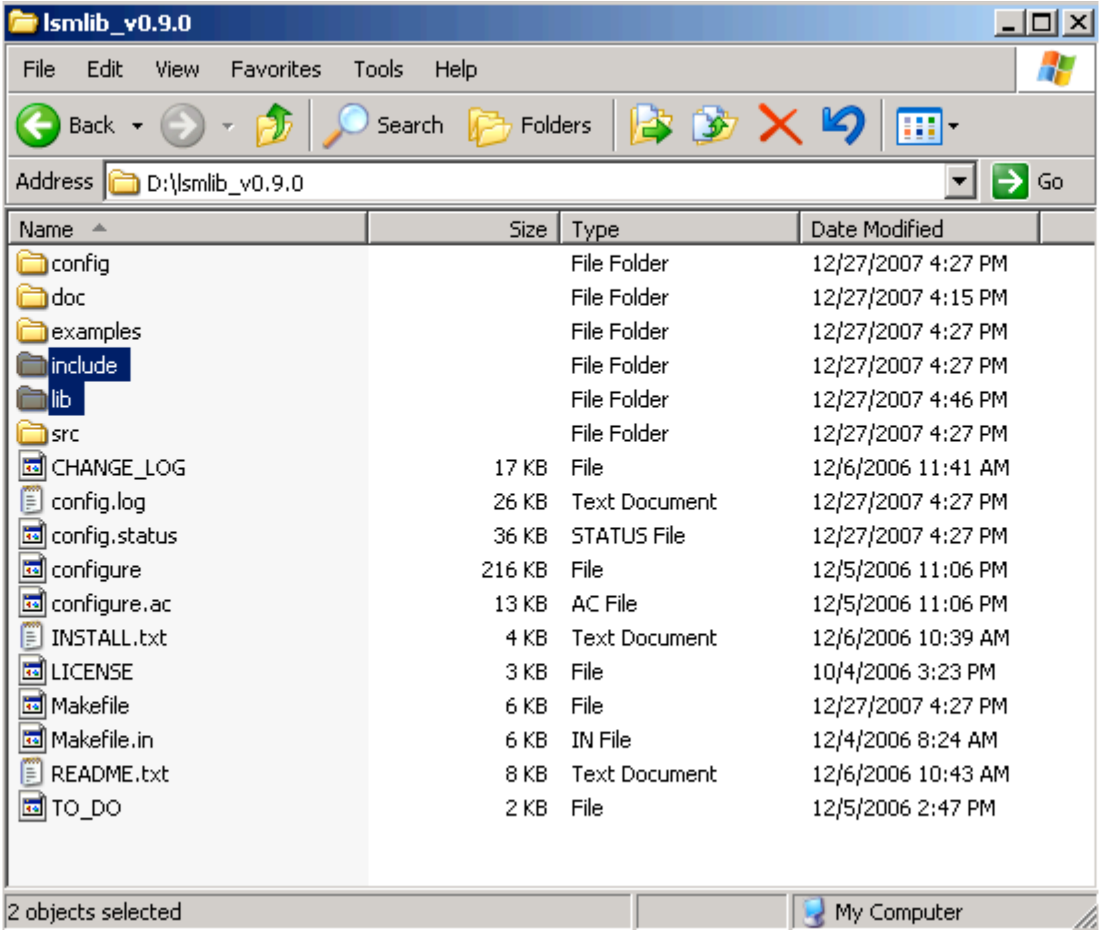
```
hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$ make all
if [ ! -d "include" ]; then mkdir include; fi
if [ ! -d "lib" ]; then mkdir lib; fi
cp -f config/LSMLIB_config.h include
cd src; make includes || exit 1
make[1]: Entering directory `/d/lsmlib_v0.9.0/src'
for DIR in toolbox serial ; do (cd $DIR; make includes) || exit 1; done
make[2]: Entering directory `/d/lsmlib_v0.9.0/src/toolbox'
for DIR in boundary_conditions fast_marching_method field_extension geometry lev
el_set_evolution localization reinitialization spatial_derivatives time_integrat
ion utilities ; do (cd $DIR; make includes) || exit 1; done
make[3]: Entering directory `/d/lsmlib_v0.9.0/src/toolbox/boundary_conditions'
cp -f ./lsm_boundary_conditions1d.h ../../..//include/
cp -f ./lsm_boundary_conditions2d.h ../../..//include/
cp -f ./lsm_boundary_conditions3d.h ../../..//include/
make[3]: Leaving directory `/d/lsmlib_v0.9.0/src/toolbox/boundary_conditions'
make[3]: Entering directory `/d/lsmlib_v0.9.0/src/toolbox/fast_marching_method'
cp -f ./FMM_Core.h ../../..//include/
cp -f ./FMM_Heap.h ../../..//include/
cp -f ./FMM_Callback_API.h ../../..//include/
make[3]: Leaving directory `/d/lsmlib_v0.9.0/src/toolbox/fast_marching_method'
make[3]: Entering directory `/d/lsmlib_v0.9.0/src/toolbox/field_extension'
cp -f ./lsm_field_extension1d.h ../../..//include/
cp -f ./lsm_field_extension2d.h ../../..//include/
```

You should see the following screen after a successful make.



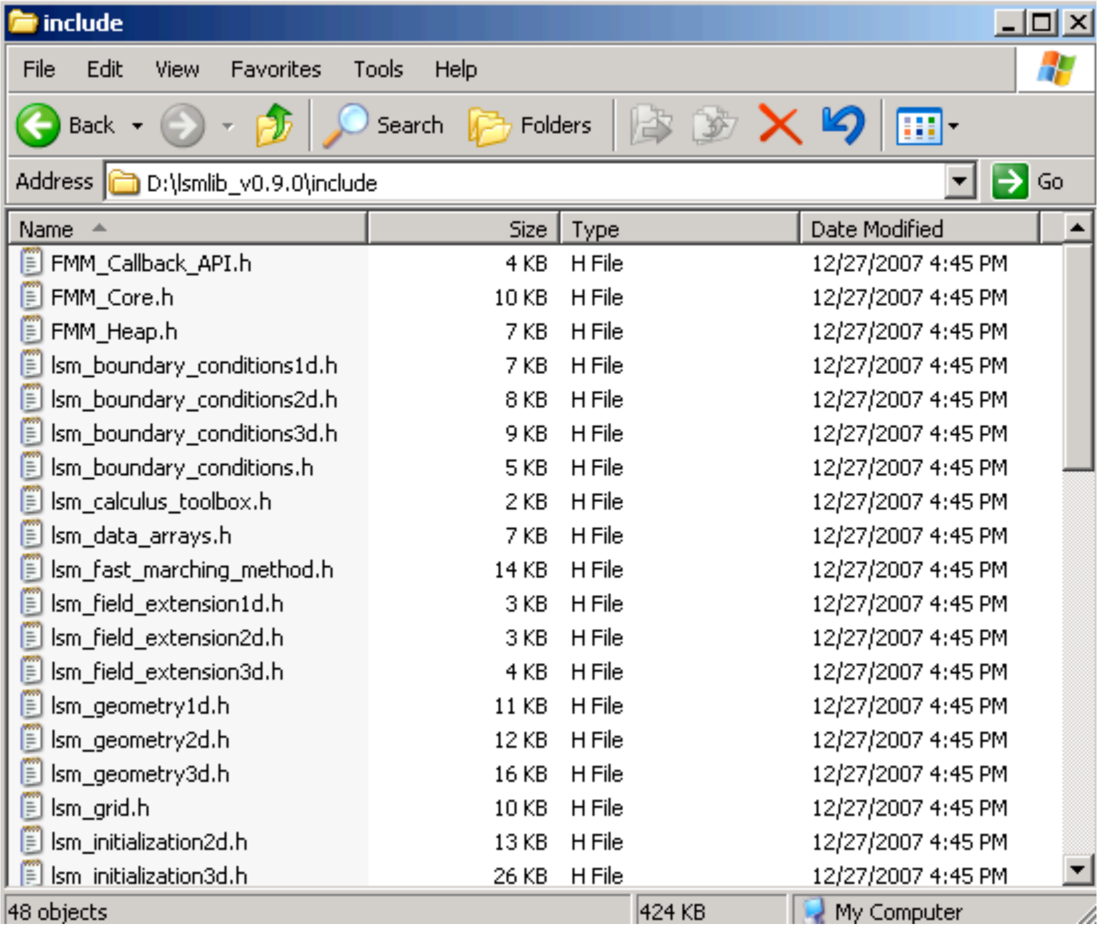
```
for FILE in `cat objs_file.tmp`; do ar -ru ./lib/liblsm_toolbox.a $FILE; done
c:\mingw\bin\ar.exe: creating ./lib/liblsm_toolbox.a
ranlib lib/liblsm_toolbox.a
rm -f objs_file.tmp
make[1]: Leaving directory `/d/lsmlib_v0.9.0'
make lsm_serial || exit 1
make[1]: Entering directory `/d/lsmlib_v0.9.0'
find src/serial -name "*.o" > objs_file.tmp
# remove test code
sed -e '/test/d' objs_file.tmp > objs_file_no_test.tmp
mv -f -f objs_file_no_test.tmp objs_file.tmp
for FILE in `cat objs_file.tmp`; do ar -ru ./lib/liblsm_serial.a $FILE; done
c:\mingw\bin\ar.exe: creating ./lib/liblsm_serial.a
ranlib lib/liblsm_serial.a
rm -f objs_file.tmp
make[1]: Leaving directory `/d/lsmlib_v0.9.0'
if [ ! -z "" ]; then                         \
  make lsm_parallel || exit 1;                              \
fi
if [ ! -z "" ]; then                              \
  make matlab || exit 1;                                    \
fi

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$ 
```

Step 10: Check the final results of compiled LSMLIB. Supposedly, the header files in “include” and library files in “lib” are the only files you will need for using LSMLIB. Also note that we only compiled the “serial” and “toolbox” portions of LSMLIB, if you also want the “matlab” and “parallel” parts, please work it out by yourself. You will need to make sure that the related libraries are installed and available to the compilers.



Name	Size	Type	Date Modified
config		File Folder	12/27/2007 4:27 PM
doc		File Folder	12/27/2007 4:15 PM
examples		File Folder	12/27/2007 4:27 PM
include		File Folder	12/27/2007 4:27 PM
lib		File Folder	12/27/2007 4:46 PM
src		File Folder	12/27/2007 4:27 PM
CHANGE_LOG	17 KB	File	12/6/2006 11:41 AM
config.log	26 KB	Text Document	12/27/2007 4:27 PM
config.status	36 KB	STATUS File	12/27/2007 4:27 PM
configure	216 KB	File	12/5/2006 11:06 PM
configure.ac	13 KB	AC File	12/5/2006 11:06 PM
INSTALL.txt	4 KB	Text Document	12/6/2006 10:39 AM
LICENSE	3 KB	File	10/4/2006 3:23 PM
Makefile	6 KB	File	12/27/2007 4:27 PM
Makefile.in	6 KB	IN File	12/4/2006 8:24 AM
README.txt	8 KB	Text Document	12/6/2006 10:43 AM
TO_DO	2 KB	File	12/5/2006 2:47 PM

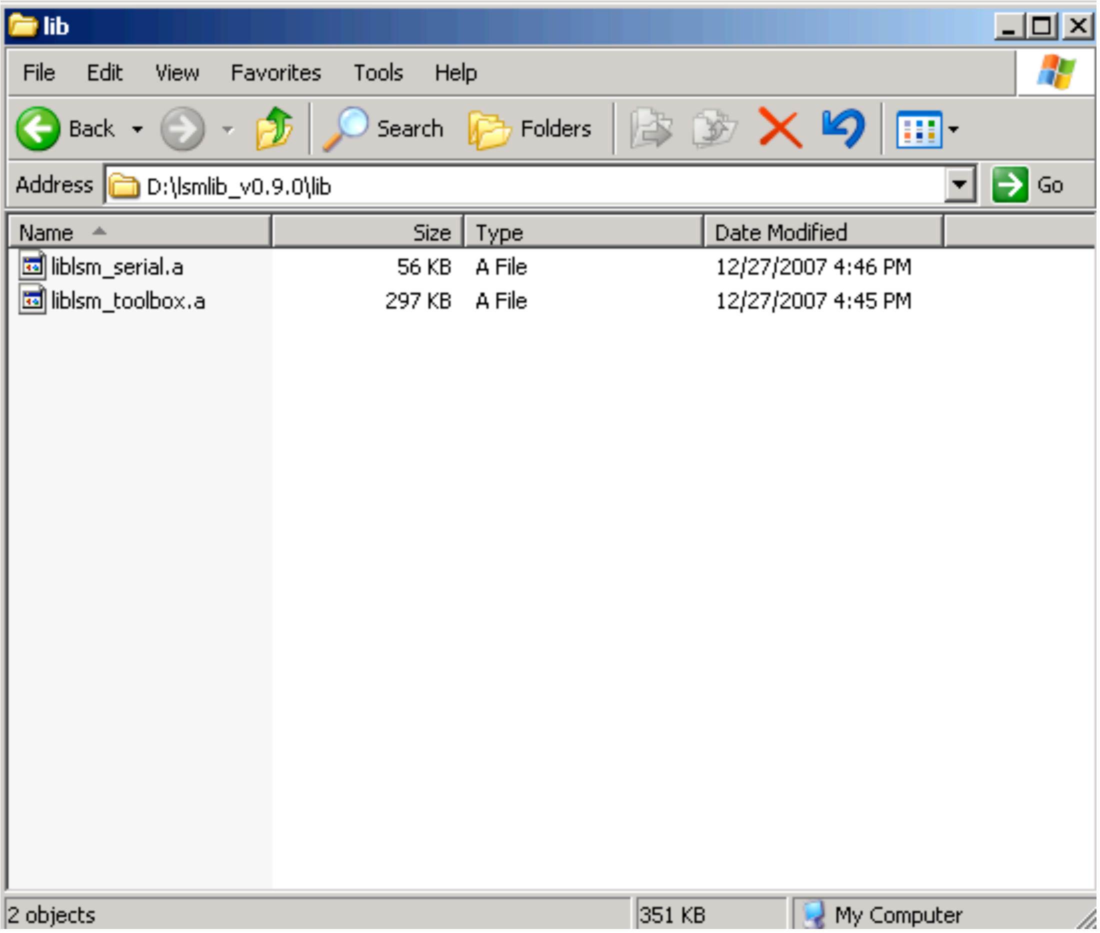
2 objects selected



Address: D:\lsmlib_v0.9.0\include

Name	Size	Type	Date Modified
FMM_Callback_API.h	4 KB	H File	12/27/2007 4:45 PM
FMM_Core.h	10 KB	H File	12/27/2007 4:45 PM
FMM_Heap.h	7 KB	H File	12/27/2007 4:45 PM
lsm_boundary_conditions1d.h	7 KB	H File	12/27/2007 4:45 PM
lsm_boundary_conditions2d.h	8 KB	H File	12/27/2007 4:45 PM
lsm_boundary_conditions3d.h	9 KB	H File	12/27/2007 4:45 PM
lsm_boundary_conditions.h	5 KB	H File	12/27/2007 4:45 PM
lsm_calculus_toolbox.h	2 KB	H File	12/27/2007 4:45 PM
lsm_data_arrays.h	7 KB	H File	12/27/2007 4:45 PM
lsm_fast_marching_method.h	14 KB	H File	12/27/2007 4:45 PM
lsm_field_extension1d.h	3 KB	H File	12/27/2007 4:45 PM
lsm_field_extension2d.h	3 KB	H File	12/27/2007 4:45 PM
lsm_field_extension3d.h	4 KB	H File	12/27/2007 4:45 PM
lsm_geometry1d.h	11 KB	H File	12/27/2007 4:45 PM
lsm_geometry2d.h	12 KB	H File	12/27/2007 4:45 PM
lsm_geometry3d.h	16 KB	H File	12/27/2007 4:45 PM
lsm_grid.h	10 KB	H File	12/27/2007 4:45 PM
lsm_initialization2d.h	13 KB	H File	12/27/2007 4:45 PM
lsm_initialization3d.h	26 KB	H File	12/27/2007 4:45 PM

48 objects | 424 KB | My Computer



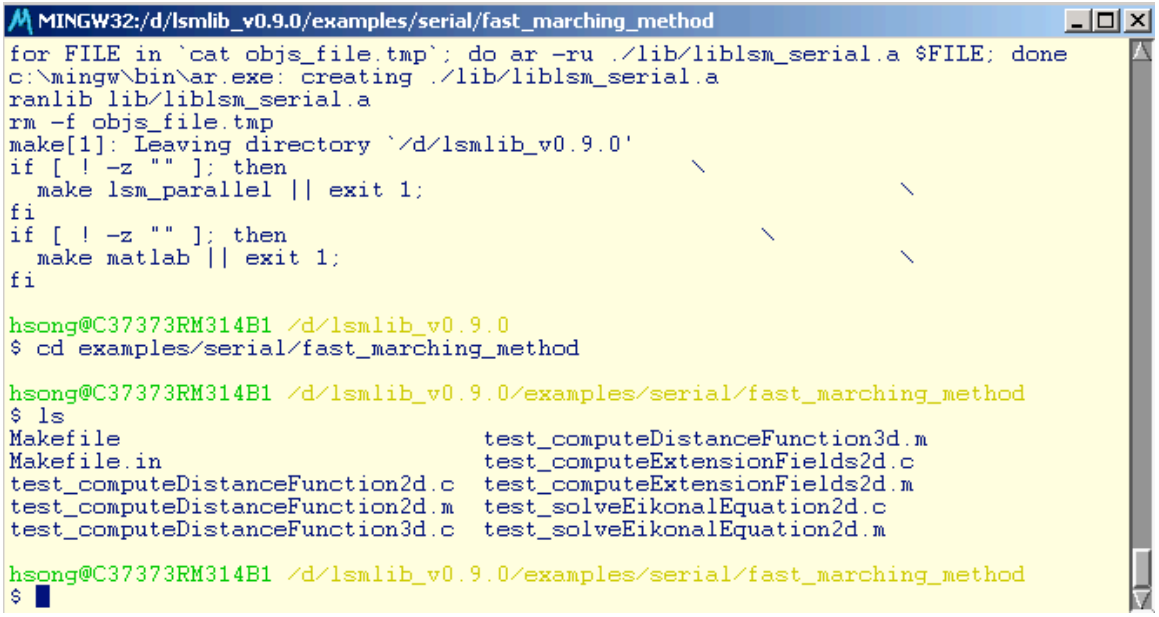
Address D:\lsmlib_v0.9.0\lib

Name	Size	Type	Date Modified
liblsm_serial.a	56 KB	A File	12/27/2007 4:46 PM
liblsm_toolbox.a	297 KB	A File	12/27/2007 4:45 PM

2 objects | 351 KB | My Computer

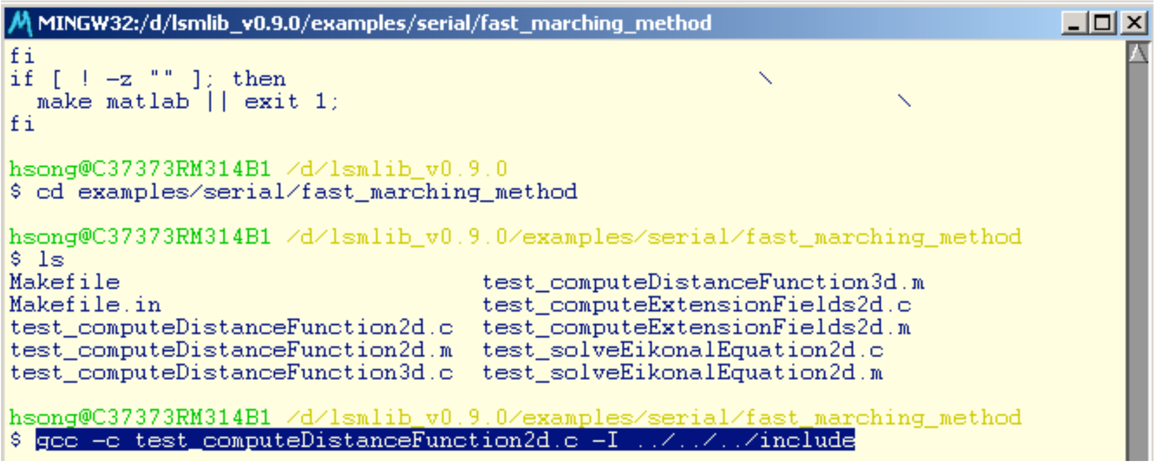
Step 11: Start using LSMLIB. We will use an example (D:\lsmlib_v0.9.0\examples\serial\fast_marching_method\test_computeDistanceFunction2d.c) from LSMLIB. Note that if you want to write FORTRAN programs to use LSMLIB, you can call most of the FORTRAN subroutines in LSMLIB directly and only need to interface the fast marching method subroutines (since these are written in C). You can search the internet to learn how to implement mix programming with gcc/g++/g77.

1. Change path to "D:\lsmlib_v0.9.0\examples\serial\fast_marching_method" in MSYS.


MINGW32:/d/lsmlib_v0.9.0/examples/serial/fast_marching_method

```
for FILE in `cat objs_file.tmp`; do ar -ru ./lib/liblsm_serial.a $FILE; done
c:\mingw\bin\ar.exe: creating ./lib/liblsm_serial.a
ranlib lib/liblsm_serial.a
rm -f objs_file.tmp
make[1]: Leaving directory `/d/lsmlib_v0.9.0'
if [ ! -z "" ]; then                                    \
  make lsm_parallel || exit 1;                                        \
fi
if [ ! -z "" ]; then                                         \
  make matlab || exit 1;                                              \
fi

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$ cd examples/serial/fast_marching_method

hsong@C37373RM314B1 /d/lsmlib_v0.9.0/examples/serial/fast_marching_method
$ ls
Makefile                          test_computeDistanceFunction3d.m
Makefile.in                       test_computeExtensionFields2d.c
test_computeDistanceFunction2d.c  test_computeExtensionFields2d.m
test_computeDistanceFunction2d.m  test_solveEikonalEquation2d.c
test_computeDistanceFunction3d.c  test_solveEikonalEquation2d.m

hsong@C37373RM314B1 /d/lsmlib_v0.9.0/examples/serial/fast_marching_method
$ 
```

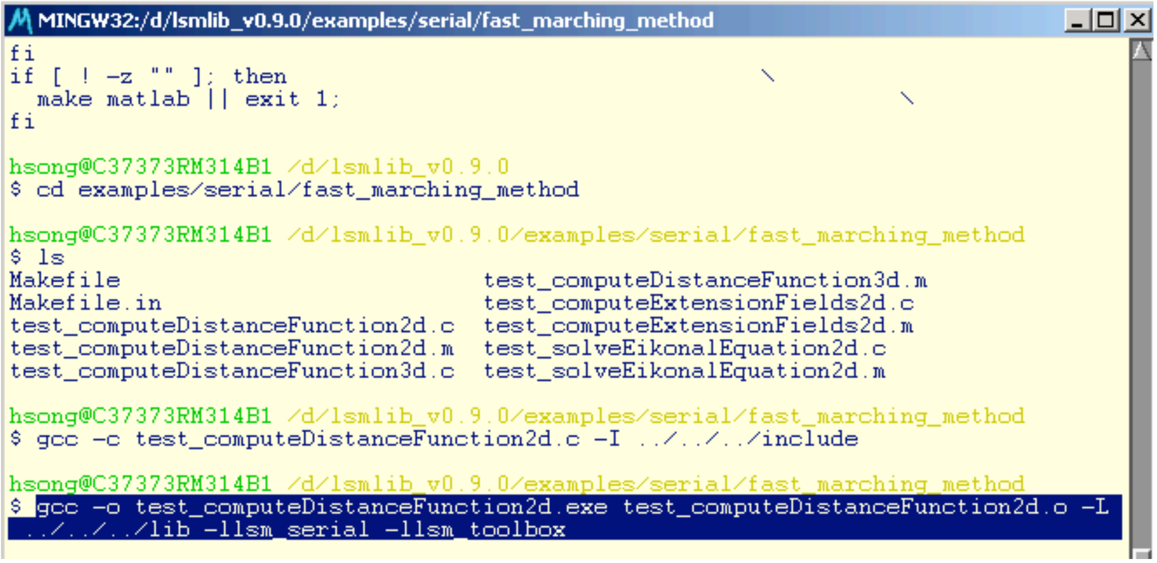
2. Compile "test_computeDistanceFunction2d.c" using gcc.


MINGW32:/d/lsmlib_v0.9.0/examples/serial/fast_marching_method

```
fi
if [ ! -z "" ]; then                                         \
  make matlab || exit 1;                                              \
fi

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$ cd examples/serial/fast_marching_method

hsong@C37373RM314B1 /d/lsmlib_v0.9.0/examples/serial/fast_marching_method
$ ls
Makefile                          test_computeDistanceFunction3d.m
Makefile.in                       test_computeExtensionFields2d.c
test_computeDistanceFunction2d.c  test_computeExtensionFields2d.m
test_computeDistanceFunction2d.m  test_solveEikonalEquation2d.c
test_computeDistanceFunction3d.c  test_solveEikonalEquation2d.m

hsong@C37373RM314B1 /d/lsmlib_v0.9.0/examples/serial/fast_marching_method
$ gcc -c test_computeDistanceFunction2d.c -I ../../../include
```

You should find the object file "test_computeDistanceFunction2d.o" after a successful compilation.

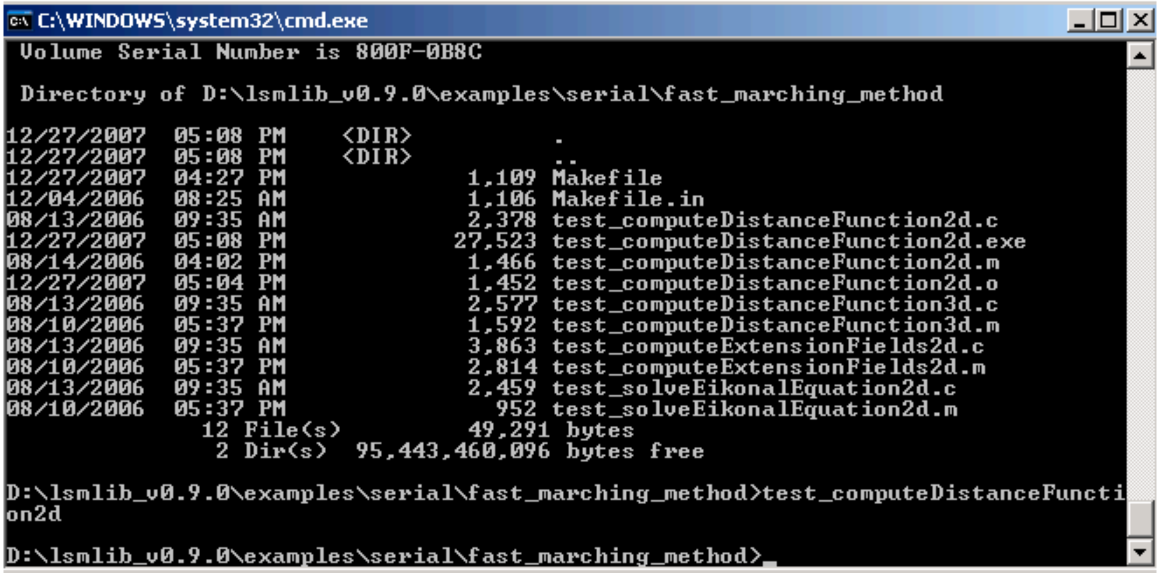
3. Link the object file "test_computeDistanceFunction2d.o" with LSMLIB.


MINGW32:/d/lsmlib_v0.9.0/examples/serial/fast_marching_method

```
fi
if [ ! -z "" ]; then                                    \
  make matlab || exit 1;                                          \
fi

hsong@C37373RM314B1 /d/lsmlib_v0.9.0
$ cd examples/serial/fast_marching_method

hsong@C37373RM314B1 /d/lsmlib_v0.9.0/examples/serial/fast_marching_method
$ ls
Makefile                          test_computeDistanceFunction3d.m
Makefile.in                       test_computeExtensionFields2d.c
test_computeDistanceFunction2d.c  test_computeExtensionFields2d.m
test_computeDistanceFunction2d.m  test_solveEikonalEquation2d.c
test_computeDistanceFunction3d.c  test_solveEikonalEquation2d.m

hsong@C37373RM314B1 /d/lsmlib_v0.9.0/examples/serial/fast_marching_method
$ gcc -c test_computeDistanceFunction2d.c -I ../../../include

hsong@C37373RM314B1 /d/lsmlib_v0.9.0/examples/serial/fast_marching_method
$ gcc -o test_computeDistanceFunction2d.exe test_computeDistanceFunction2d.o -L
 ../../../lib -llsm_serial -llsm_toolbox
```

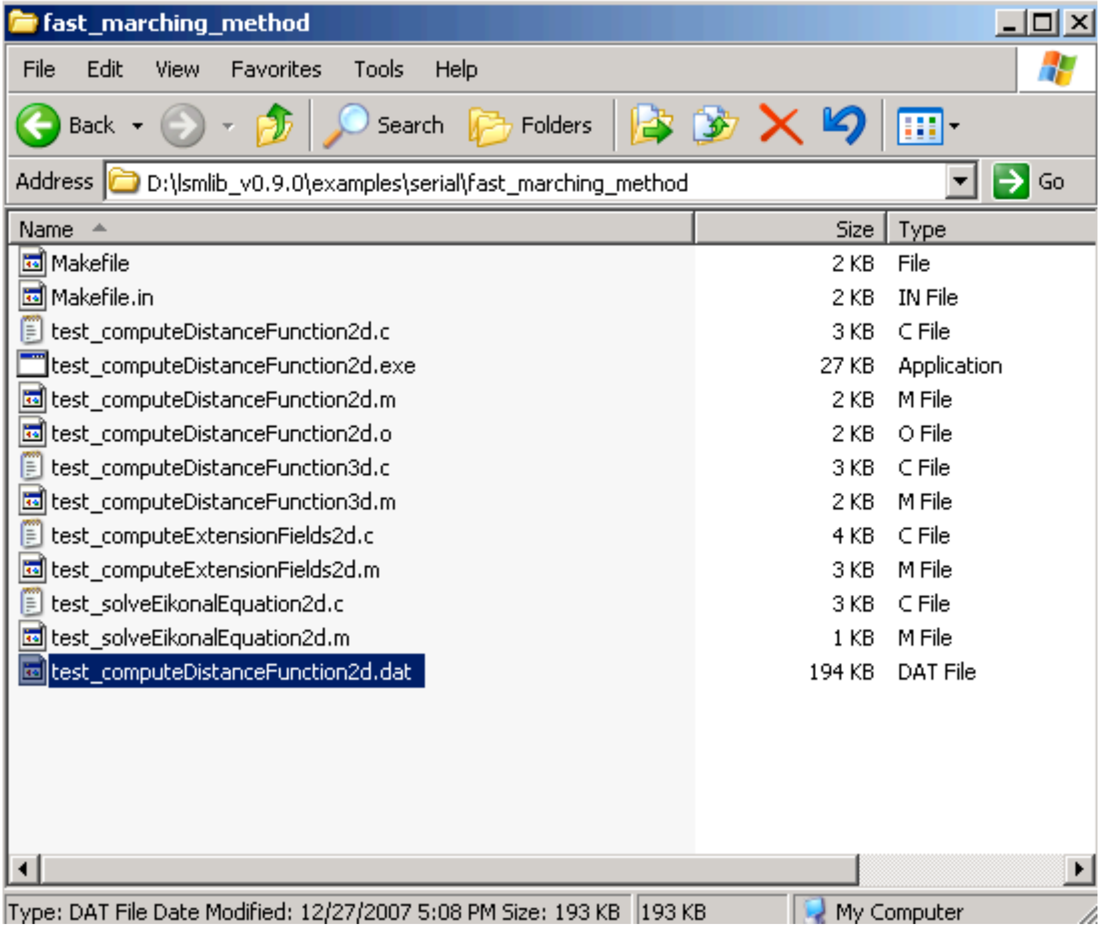
After a successful link, you should find the executable file “test_computeDistanceFunction2d.exe”.

4. Test the executable file “test_computeDistanceFunction2d.exe”.



```
 Volume Serial Number is 800F-0B8C

 Directory of D:\lsmlib_v0.9.0\examples\serial\fast_marching_method

12/27/2007  05:08 PM    <DIR>          .
12/27/2007  05:08 PM    <DIR>          ..
12/27/2007  04:27 PM             1,109 Makefile
12/04/2006  08:25 AM             1,106 Makefile.in
08/13/2006  09:35 AM             2,378 test_computeDistanceFunction2d.c
12/27/2007  05:08 PM            27,523 test_computeDistanceFunction2d.exe
08/14/2006  04:02 PM             1,466 test_computeDistanceFunction2d.m
12/27/2007  05:04 PM             1,452 test_computeDistanceFunction2d.o
08/13/2006  09:35 AM             2,577 test_computeDistanceFunction3d.c
08/10/2006  05:37 PM             1,592 test_computeDistanceFunction3d.m
08/13/2006  09:35 AM             3,863 test_computeExtensionFields2d.c
08/10/2006  05:37 PM             2,814 test_computeExtensionFields2d.m
08/13/2006  09:35 AM             2,459 test_solveEikonalEquation2d.c
08/10/2006  05:37 PM               952 test_solveEikonalEquation2d.m
              12 File(s)         49,291 bytes
               2 Dir(s)  95,443,460,096 bytes free

D:\lsmlib_v0.9.0\examples\serial\fast_marching_method>test_computeDistanceFuncti
on2d

D:\lsmlib_v0.9.0\examples\serial\fast_marching_method>_
```

This example generates a text file “test_computeDistanceFunction2d.dat”

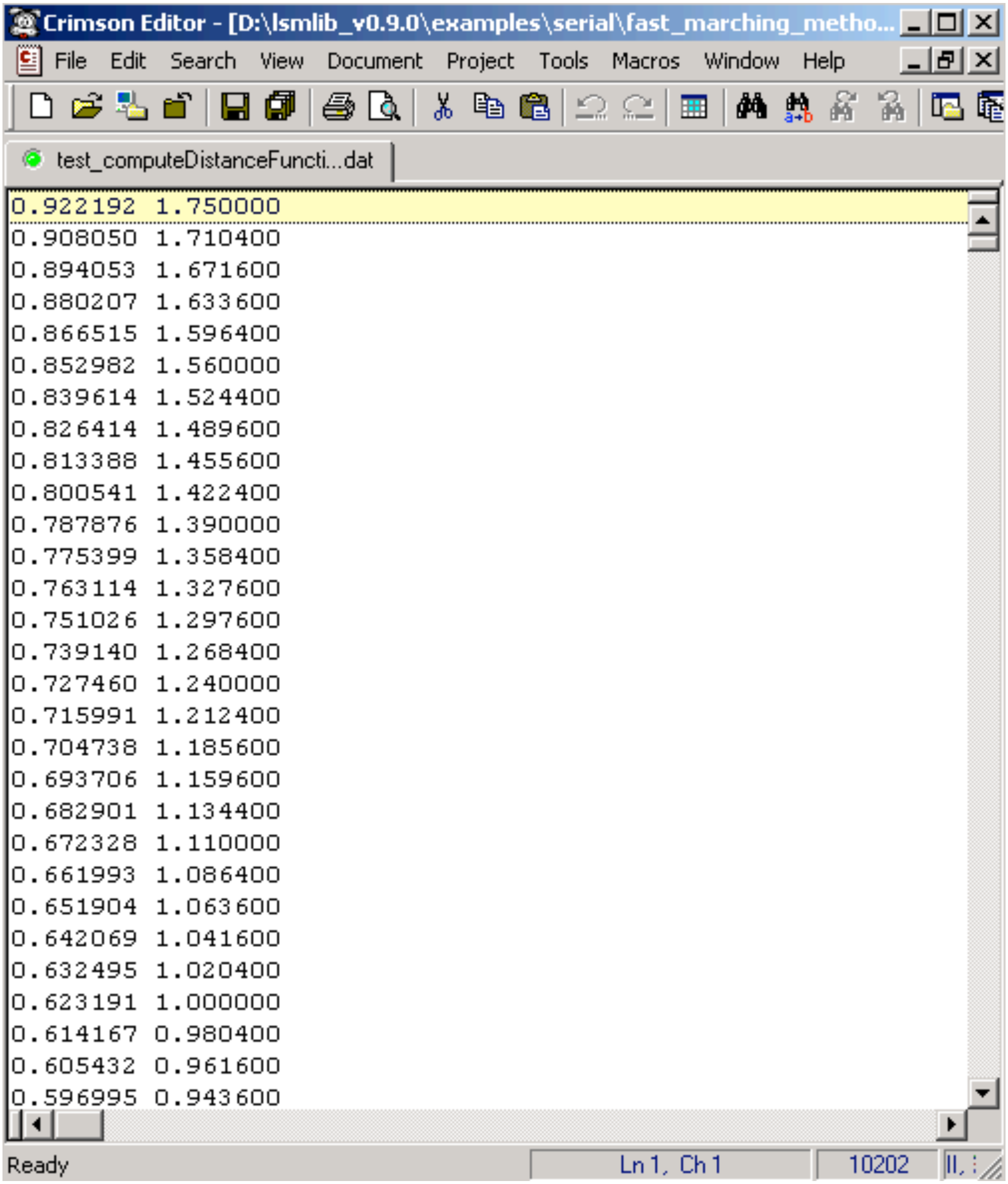


Address D:\lsmlib_v0.9.0\examples\serial\fast_marching_method

Name	Size	Type
Makefile	2 KB	File
Makefile.in	2 KB	IN File
test_computeDistanceFunction2d.c	3 KB	C File
test_computeDistanceFunction2d.exe	27 KB	Application
test_computeDistanceFunction2d.m	2 KB	M File
test_computeDistanceFunction2d.o	2 KB	O File
test_computeDistanceFunction3d.c	3 KB	C File
test_computeDistanceFunction3d.m	2 KB	M File
test_computeExtensionFields2d.c	4 KB	C File
test_computeExtensionFields2d.m	3 KB	M File
test_solveEikonalEquation2d.c	3 KB	C File
test_solveEikonalEquation2d.m	1 KB	M File
test_computeDistanceFunction2d.dat	194 KB	DAT File

Type: DAT File Date Modified: 12/27/2007 5:08 PM Size: 193 KB | 193 KB | My Computer

And this is its content


Crimson Editor - [D:\lsmlib_v0.9.0\examples\serial\fast_marching_metho...
File Edit Search View Document Project Tools Macros Window Help
test_computeDistanceFuncti...dat
Ready
Ln 1, Ch 1
10202

```
0.922192 1.750000
0.908050 1.710400
0.894053 1.671600
0.880207 1.633600
0.866515 1.596400
0.852982 1.560000
0.839614 1.524400
0.826414 1.489600
0.813388 1.455600
0.800541 1.422400
0.787876 1.390000
0.775399 1.358400
0.763114 1.327600
0.751026 1.297600
0.739140 1.268400
0.727460 1.240000
0.715991 1.212400
0.704738 1.185600
0.693706 1.159600
0.682901 1.134400
0.672328 1.110000
0.661993 1.086400
0.651904 1.063600
0.642069 1.041600
0.632495 1.020400
0.623191 1.000000
0.614167 0.980400
0.605432 0.961600
0.596995 0.943600
```

If you can follow till this step, congratulations, you can start programming with LSMLIB!

Another comment, learn “Configure”, “Makefile”, “gcc/g++/g77” to get yourself familiar with important programming tools and make life easier and easier in the future!