

# Steps Taken to install Overture on Windows using VirtualBoxVM

Christopher Jarvis - chjarvis@vt.edu

September 20, 2012

## Install VM and Linux

1. Install VirtualBox (4.1.20)
  - (a) download from <https://www.virtualbox.org/>
2. Set up VM and install OS
  - (a) Ubuntu 10.04 LTS (64 bit, desktop)

## Install Overture on VM

To install Overture in Ubuntu I followed the instructions of Mohammad Abouali found here:

<http://mabouali.wordpress.com/2012/01/16/installing-overture-on-a-freshly-installed-ubuntu/>

1. Begin with fresh install of Ubuntu 10.04 LTS
  - (a) update all packages
  - (b) install
    - i. tcsh
    - ii. emacs
  - (c) set default current folder in path
    - i. set path = (. ~/bin /usr/apps/bin \$path)
  - (d) reboot
2. Preliminary packages
  - (a) sudo apt-get install build-essential manpages-dev gfortran autoconf automake

(b) check compiler versions

i. gcc - 4.4.3

ii. g++ - 4.4.3

iii. gfortran - 4.4.3

### 3. OpenMotif

(a) install

i. libmotif3

ii. libmotif-dev (depends on libmotif3)

### 4. OpenGL and Mesa

(a) sudo apt-get install libgl1-mesa-dev libglu1-xorg libglu1-xorg-dev libglut3 libglut3-dev x11proto-gl-dev x11proto-print-dev libjpeg62-dev libzip-dev libperl-dev libXpm-dev libXp-dev libxmu-dev libxi-dev

From here I used downloaded the tar files and extracted to and installed in “~/apps/<PACKAGE>”. Note: the commands are run from the extracted package directory.

#### 1. HDF5 (hdf5-1.6.9)

(a) sudo ln -s make gmake (symbolic link to gmake from make)

(b) unsetenv CC

(c) unsetenv cc

(d) setenv CC gcc

(e) ./configure -prefix='pwd'

(f) make

(g) make install

#### 2. A++ (A++P++-0.8.0)

(a) I extracted to ~/apps/A++P++-0.8.0

(b) configure -enable-SHARED\_LIBS -prefix 'pwd'

(c) make

(d) make install

(e) make check

### 3. LAPACK (lapack-3.4.1)

- (a) `cp make.inc.example make.inc`
- (b) `make blaslib lapacklib tmglib`
- (c) `cp librefblas.a libblas.a`

### 4. PETSc (2.3.2-p10-lite)

- (a) `setenv PETSC_DIR `pwd``
- (b) `./config/configure.py -with-debugging=0 -with-fortran=0 -with-matlab=0 -with-mpi=0 -with-shared=1 -with-dynamic=1 -prefix=`pwd` -download-c-blas-lapack=1`
- (c) `setenv PETSC_ARCH "recommended"`
- (d) `make all`
- (e) `make install` (may not be necessary install dir is make dir)
- (f) `make test`
- (g) `reboot`

### 5. Overture (Overture.v24g)

- (a) set up `defenv`
  - i. `setenv XLIBS /usr`
  - ii. `setenv MOTIF /usr`
  - iii. `setenv OpenGL /usr`
  - iv. `setenv HDF ~/apps/hdf5-1.6.9`
  - v. `setenv APlusPlus ~/apps/A++P++-0.8.0/A++/install`
  - vi. `setenv Overture ~/apps/Overture.v24`
  - vii. `setenv CG ~/apps/cg.v24`
  - viii. `setenv LAPACK ~/apps/lapack-3.4.1`
- (b) `source defenv`
- (c) in `configure` file change `$FortranDouble` to `$FortranDouble = "-fdefault-real-8 -fdefault-double-8"`; (prevents error during `cg` compile)
- (d) **\*\* IF you are going to install `cg` then correct `cg.v24/sm/src/getRayleighSpeed.C` before compiling \*\***  
change: `printf(getRayleighSpeed: mu=%e, lambda=%e, rho=%e\n",mu,lambda,rho,gamma);`  
to: `cout << "getRayleighSpeed: mu= " << mu << ", lambda=" << lambda << ", rho=" << rho << ", gamma= " << gamma << endl;`

- (e) configure
  - (f) make (or make -j# for parallel using # cores)
  - (g) check.p (all successful)
6. cg (cg.v24)
- (a) make
  - (b) make check
7. Install complete!!
8. Test
- (a) create local test folder
  - (b) copy cilc.cmd from Overture.v24/sampleGrids to test folder
  - (c) copy cylinder.cmd from cg.v24/ins/cmd to test folder
  - (d) cd to test folder
  - (e) ogen cilc (generates default grid)
  - (f) cgins cylinder (solves PDE)

## Comments

I have also installed Overture on Scientific Linux (SL 6.2) using similar instructions. The subtle difference is the repository support is not as vast for SL as for Ubuntu.