wiki.chpc.ac.za

Installing Octave 3.6.3 from Source

This guide describes how Octave 3.6.3 would be installed from source, under your home directory. This guides follow on after the GCC 4.7.1 and Python 2.7 guides. i.e

```
$ gcc -v
...
gcc version 4.7.1 (GCC)
$ python -V
Python 2.7.3
```

Octave is to be install under $HOME/local folder, and the build directory used is /tmp/$USER/octave_build

Lets begin by downloading the source for Octave 3.6.3 and reading the install instructions,

```
$ mkdir -p /tmp/$USER/octave_build
$ cd /tmp/$USER/octave_build
$ wget -c ftp://ftp.gnu.org/gnu/octave/octave-3.6.3.tar.bz2
$ tar -xf octave-3.6.3.tar.bz2
$ cd octave-3.6.3
$ less INSTALL
$ less README
```

Prerequisites

GNU make

A recent version of GNU make is required, so

```
$ cd /tmp/$USER/octave_build
$ wget ftp://ftp.gnu.org/gnu/make/make-3.82.tar.gz
$ tar -xf make-3.82.tar.gz
$ mkdir make-3.82/BUILD
$ cd make-3.82/BUILD
$ ../configure --prefix=$HOME/local --libdir=$HOME/local/lib64
$ make
$ make check
$ make install
```

LAPACK and BLAS

Follow the install instruction, as in the numpy subsection of the installing Python from source guide.

Optionals

SuiteSparse

SuiteSparse [http://www.cise.ufl.edu/research/sparse/SuiteSparse/] contains libraries for handling sparse matrixes, include libraries such as * AMD: symmetric approximate minimum degree * CCOLAMD: constrained column approximate

minimum degree * UMFPACK: sparse multifrontal LU factorization

```
$ cd /tmp/$USER/octave_build
$ wget -c http://www.cise.ufl.edu/research/sparse/SuiteSparse/current/SuiteSparse.tar.gz
$ tar -xf SuiteSparse.tar.gz
$ cd SuiteSparse
$ less README.txt
```

#First we need to install # Metis [http://glaros.dtc.umn.edu/gkhome/metis/metis/overview] 4.0.1. # $ wget http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/OLD/metis-4.0.1.tar.gz [http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/OLD/metis-4.0.1.tar.gz] # $ tar -xf metis-4.0.1.tar.gz # $ cd metis-4.0 #edit the metis-4.0/Lib/rename.h file and replace the last line in the file: # #define log2 log2 #with the following: # #define log2 METISlog2 #edit the Makefile.in, changing the CC to gcc, and add the -fPIC flag. Then # $ make #To test, # $ cd Graphs # $ ./mtest test.mgraph

Then edit the *SuiteSparse_config/SuiteSparse_config.mk* as follows

- add the -fPIC flag to the F77FLAGS declaration
- INSTALL_LIB=$(HOME)/local/lib64
- INSTALL_INCLUDE=$(HOME)/local/include
- under the CHOLMOD configuration section, uncomment the line

CHOLMOD_CONFIG = -DNPARTITION

- under the Linux section, uncomment the lines (and add the the -fPIC flag)

#CC = gcc

```
  #CF = $(CFLAGS) -O3 -fexceptions -fPIC
* since no metis-4.0 change
* For CSaprse library, add the -fPIC to n the makefile CSpares/Lib/Makefile
CF declaration?
```

build SuitSparse:

```
$ export CC=gcc
$ make
```

test build

List the build libraries

```
$ find ./ -name \*.a
./SuiteSparse_config/libsuitesparseconfig.a
./SuiteSparse_config/xerbla/libcerbla.a
./COLAMD/Lib/libcolamd.a
./CCOLAMD/Lib/libccolamd.a
./CHOLMOD/Lib/libcholmod.a
./CAMD/Lib/libcamd.a
./AMD/Lib/libamd.a
./metis-4.0/libmetis.a
./CSparse/Lib/libcsparse.a
./BTF/Lib/libbtf.a
./SPQR/Lib/libspqr.a
./CXSparse/Lib/libcxsparse.a
./KLU/Lib/libklu.a
./LDL/Lib/libldl.a
./RBio/Lib/librbio.a
./UMFPACK/Lib/libumfpack.a
```

Intall

```
$ make install
```

Unset environmental variables

```
$ unset CC
$ unset CFLAGS
```

qrupdate

Speeds up the QR & Cholesky updating functions.

```
$ cd /tmp/$USER/octave_build/
$ wget http://tenet.dl.sourceforge.net/project/qrupdate/qrupdate/1.2/qrupdate-1.1.2.tar.gz
$ tar -xf qrupdate-1.1.2.tar.gz
$ cd qrupdate-1.1.2
$ less INSTALL
```

Edit Makeconf as follow

- LIBDIR=lib64
- PREFIX set prefix to your home/local folder

Build it

```
$ make lib
$ make solib
$ make test
...
TOTAL:     PASSED 128     FAILED   0
```

Install it

```
$ make install
```

GraphicsMagick++

Required for the imread function for reading image files to be fully functional.

HDF5 library

Allows Octave will to be able to read or load HDF5 data files.

Qhull

Required for some geometry functions

Configure and Build

```
$ cd /tmp/$USER/octave_build/octave-3.6.3
```

```
$ mkdir BUILD
$ cd BUILD
$ ../configure --help | less
$ ../configure --prefix=$HOME/local --libdir=$HOME/local/lib64 \
   --disable-docs  --enable-docs=no
$ $ cp ../AUTHORS ./
```

If make is run now, the build process would fail giving the following error message

```
../liboctave/.libs/liboctave.so: undefined reference to `SuiteSparse_time'
collect2: error: ld returned 1 exit status
```

Where the SuiteSparse_time prototype is defined in SuiteSparse_config.h header which is installed with SuiteSparse package. And the SuiteSparse_time is declared in both the

```
SuiteSparse_config/SuiteSparse_config.c, and the
CHOLMOD/Tcov/SuiteSparse_config.c
```

files.

Source file which use SuiteSparse_time are,

```
CHOLMOD/Supernodal/t_cholmod_super_numeric.c
CHOLMOD/Supernodal/t_cholmod_gpu.c
UMFPACK/Source/umfpack_tictoc.c:
```

Adding the -lsuitesparseconfig flag to appropriate lib variables in the Makefiles fixs the problem. ie

```
CHOLMOD_LIBS = -lcholmod     ->    CHOLMOD_LIBS = -lcholmod -lsuitesparseconfig -lrt
UMFPACK_LIBS = -lumfpack     ->    UMFPACK_LIBS = -lumfpack -lsuitesparseconfig -lrt
```

-lrt added for undefined reference to `clock_gettime' error.

Alter all the Makefiles using sed and find, as follows

```
$ for MF in `find -name Makefile` ; do
    echo modifying $MF
    sed -i "s|CHOLMOD_LIBS = -lcholmod|CHOLMOD_LIBS = -lcholmod -lsuitesparseconfig -lrt|" $MF
    sed -i "s|UMFPACK_LIBS = -lumfpack|UMFPACK_LIBS = -lumfpack -lsuitesparseconfig -lrt|" $MF
  done
```

Now you are ready to build octave.

Build is going to take a while, so i recommend using screen;

```
$ screen
$ make > make.build 2>&1
# ctrl-a crtl-d
$ tail -f make.build
```

Build took just under an hour to complete, when this guide was written.

Testing

```
$ make check
```

When this guide was written. The following results were obtained

```
scripts/signal/fftfilt.m ............................. PASS    8/9    FAIL   1
Summary:
```

```
  PASS  10020
  FAIL      1
```

Which aint bad, overall :D

Installation

```
$ make install
```

and finally,

```
$ octave -v
GNU Octave, version 3.6.3
...
```

Remember to clean up after yourself

```
$ rm -rf /tmp/$USER/octave_build
```
