Instruction to Gopher Fast Fourier Transform Algorithm

Mark Iwen (markiwen@math.msu.edu)

Ruochuan Zhang (zhangr12@math.msu.edu)

How to compile and run the code:

(*>> … : command line)

1. Put ‘GFFT.tgz' in your home directory.

2. Unzip GFFT.tgz'.

3. Open ‘Makefile’ and see ‘Instruction’ in it.

```
####################################################################################################
#Instructions
####################################################################################################
#1. You have to edit line 23 (the line starting with LIBS) to indicate where your 'fftw3 library' is located.
#     For example, the LIBS line can be:
#     LIBS = -L/Users/username/usr/lib -lfftw3 -lm
#     You may need to edit the directory appearing after -L to give the correct path of LIBS.
#2. If you don't have it installed yet, then see the following:
#     1) visit : http://micro.stanford.edu/wiki/Install_FFTW3     to see how to install the library.
#     2) In the command line:
#          >>mkdir $HOME/usr
#          >>mkdir $HOME/soft
#          >>cd ~/soft
#          >>wget http://www.fftw.org/fftw-3.3.3.tar.gz (this is where the 'latest' fftw3 library is located)
#          >>tar -zxvf fftw-3.3.3.tar.gz
#          >>cd fftw-3.3.3
#          >>./configure --prefix=$HOME/usr --enable-shared=yes
#          >>make --jobs=8
#          >>make install
####################################################################################################
```

4. Edit ‘Makefile’ according to the instructions. Mainly, the edition is changing the directory of LIBS.

5. Open ‘GFFT.cpp’ and read the instructions.

```
//**************************************************************************************************************
********************************************
//Instructions
//This code runs numerical experiments for the Gopher Fast Fourier Transform Algorithm (GFFT) algorithm.

//Parameter setting starts at line 28 (after the include and using namespace commands).

//Read the instruction of parameter set up section and set the parameters as you want
//One can include the directory location of fftw3.h in the include command.
//For example, one can change #include <fftw3.h> to #include </User/username/usr/include/fftw3.h> to let the compiler know the correct
path of fftw3.h.

//We have to compile this file (GFFT.cpp) with primes.h using Makefile.
//**************************************************************************************************************
********************************************
```

6. Edit the parameters in ‘GFFT.cpp’ based on the description of each parameter.

```
//**************************************************************************************************************
*****//
//Below is the part of the code for setting the parameters.
//**************************************************************************************************************
*****//
 //N is the bandwidth, later will take the value of N from outside (N is the smallest power of two number that greater than sample size)
 //e.g. if sample size is 1000, then N is 2^10 = 1024
 const int N = pow(2, 22);
 //number of blocks in Fourier space
```

```
const int num_block = 3;
//length of block in Fourier space
const int length_block = pow(2, 4);
//set this to be true of want use FFTW_MEASURE, otherwise, set it to be false
bool fftwMeasure = false;
//number of repeated numerical test
int num_test = 1;
//number of different numerical test
int rounds = 10;
//set this to be true if want to run the numerical experiment in the noiseless case
//otherwise set it to be false
bool noiseless = false;
//Signal to Noise Ratio
double SNR = 30;
//**********************************************************************************************************************
******//
```

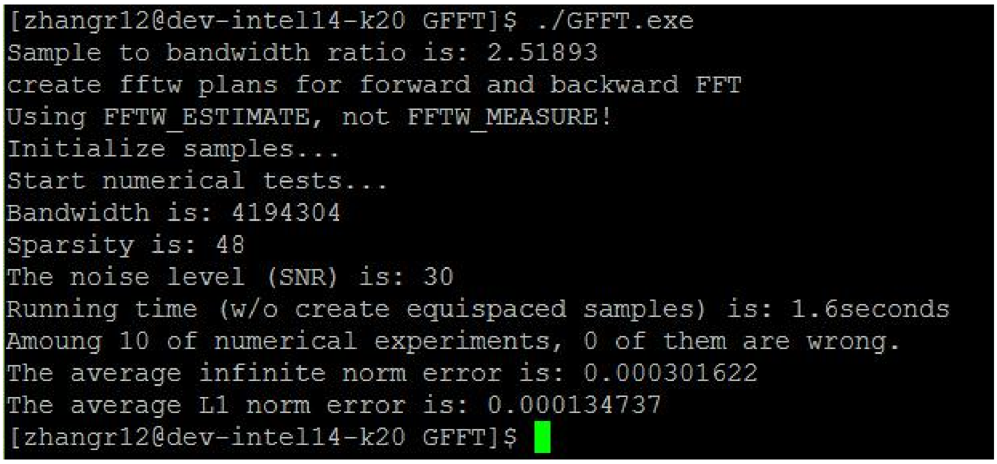
7. Generate ‘GFFT.exe’.

   >>make

   To execute ‘GFFT.exe’.

   >> ./GFFT.exe

8. Output example:



```
[zhangr12@dev-intel14-k20 GFFT]$ ./GFFT.exe
Sample to bandwidth ratio is: 2.51893
create fftw plans for forward and backward FFT
Using FFTW_ESTIMATE, not FFTW_MEASURE!
Initialize samples...
Start numerical tests...
Bandwidth is: 4194304
Sparsity is: 48
The noise level (SNR) is: 30
Running time (w/o create equispaced samples) is: 1.6seconds
Amoung 10 of numerical experiments, 0 of them are wrong.
The average infinite norm error is: 0.000301622
The average L1 norm error is: 0.000134737
[zhangr12@dev-intel14-k20 GFFT]$ 
```