Chemnitz University of Technology
Michael Pippig and Daniel Potts

Jülich Research Center
Heraeus Summer School

Programming with Nonequispaced FFT

Lab 2 **C Library Hands On**

Exercise 1: (Installation and testing of NFFT)
Browse through the NFFT homepage

`http://www.tu-chemnitz.de/~potts/nfft`

Then, download the NFFT package and build the library in your home directory, i.e.,

1. `tar xfvz nfft-3.1.3.tar.gz`
2. `cd nfft-3.1.3`
3. `./configure`
4. `make`

Alternatively, you can first do Exercise 1 from Lab 3 and install NFFT on JUMP, i.e.,

1. `cd ~/src`
2. `wget http://www.tu-chemnitz.de/~potts/nfft/download/nfft-3.1.3.tar.gz`
3. `gtar xzvf nfft-3.1.3.tar.gz`
4. `cd nfft-3.1.3`
5. `export CC="xlc" CFLAGS="-O3 -q64 -qhot" AR="ar -X64"`
6. `export CPPFLAGS="-I$HOME/FFTW3_3_MPI/include"`
7. `export LDFLAGS="-L$HOME/FFTW3_3_MPI/lib"`
8. `./configure --with-window=gaussian --disable-shared --prefix=$HOME/NFFT_GAUSS`
9. `gmake all install`

Lookup and open the source file `simple_test.c` found in `nfft-3.1.0/examples/nfft`.
Skim through the subroutine `simple_test_nfft_1d()`. Try to understand what it does.
Then, run the actual executable `simple_test`.

Exercise 2: (Exploring precomputation of NFFT)
Using matrix-vector notation as in the lecture, the NFFT algorithm corresponds to using the approximation

$$\mathbf{A}\hat{\mathbf{f}} \approx \mathbf{BFD}\hat{\mathbf{f}},$$

where $\mathbf{B}$ denotes the real $M \times n$ sparse matrix

$$\mathbf{B} := \left(\tilde{\psi}\left(x_j - \frac{l}{n}\right)\right)_{j=0,\ldots,M-1;\, l=-n/2,\ldots,n/2-1}.$$

We propose different methods for the compressed storage and application of the matrix $\mathbf{B}$ which are all available in the NFFT library by choosing different precomputation flags [1]. These methods do not yield a different asymptotic performance but yet lower the constant hidden in the $\mathcal{O}$ notation.

Compare the situation with no precomputation (that is, no precomputation flags set) with the usage of the flags `PRE_PSI` and `PRE_FULL_PSI` in the routine `simple_test_nfft_2d`. Modify the call to `nfft_init_guru` as necessary. There should be an observeable performance difference.

References

[1] J. Keiner, S. Kunis, and D. Potts. Using NFFT3 - a software library for various nonequispaced fast Fourier transforms. *ACM Trans. Math. Software*, 36:Article 19, 1 – 30, 2009.