Programming with Nonequispaced FFT

Solution 2 **C Library Hands On**

Exercise 1:
The routine `simple_test_nfft_1d()` initialises a plan for a one-dimensional nfft with $N = 14$ Fourier coefficients and $M = 19$ nodes, generates random nodes $x_j \in [-\frac{1}{2}, \frac{1}{2}]$, $j = 0, \dots, 18$, and precomputes the matrix $\mathbf{B} \in \mathbb{R}^{19\times 32}$. Random Fourier coefficients $\hat{f}_k \in \mathbb{C}$, $k = -7, \dots, 6$ are generated as well and the trigonometric polynomial $f(x) = \sum_{k=-7}^{6} \hat{f}_k \mathrm{e}^{-2\pi\mathrm{i}kx}$ is evaluated at the nodes x_j by the routine `ndft_trafo` (direct way), and by `nfft_trafo` using the approximation scheme as outlined in the lecture. Coefficients and function values are displayed. Also, the adjoint transforms are executed - clearly showing that the nfft is, in contrast to the fft, a non-unitary transform.

The output on JUMP looks like:

```
1) computing an one dimensional ndft, nfft and an adjoint nfft

 given Fourier coefficients, vector f_hat, adr=110025af0
     0. +3.9E-01+2.9E-01i,+7.4E-01+3.0E-01i,+7.6E-02+4.0E-01i,+8.6E-01+9.4E-01i,
     4. +6.6E-01+8.5E-01i,+2.8E-03+4.6E-01i,+5.3E-01+7.9E-01i,+2.7E-01+9.8E-01i,
     8. +3.1E-01+6.0E-01i,+6.1E-01+2.1E-01i,+8.9E-01+3.0E-01i,+1.5E-01+3.4E-01i,
    12. +3.9E-01+6.4E-01i,+7.5E-01+6.0E-01i,

 ndft, vector f, adr=110025bf0
     0. -8.3E-01-3.6E-01i,+6.5E-01+2.4E+00i,+5.0E-01+7.8E-01i,+2.6E+00+2.9E+00i,
     4. +6.4E-02+1.8E+00i,-4.3E-01+2.3E-02i,+1.5E-01+2.9E-02i,-2.6E+00-6.2E-01i,
     8. +2.3E+00-1.2E+00i,-5.8E-01-2.4E-01i,+3.9E-01+1.1E+00i,-1.2E+00+2.8E-01i,
    12. -7.6E-01-4.9E-01i,+2.5E+00-1.2E+00i,+2.2E+00-1.2E+00i,+3.6E+00+5.9E+00i,
    16. -3.1E+00-1.6E+00i,-2.5E+00-5.0E-01i,-2.2E+00+1.5E+00i,
 took 1.400000e-04 seconds.

 nfft, vector f, adr=110025bf0
     0. -8.3E-01-3.6E-01i,+6.5E-01+2.4E+00i,+5.0E-01+7.8E-01i,+2.6E+00+2.9E+00i,
     4. +6.4E-02+1.8E+00i,-4.3E-01+2.3E-02i,+1.5E-01+2.9E-02i,-2.6E+00-6.2E-01i,
     8. +2.3E+00-1.2E+00i,-5.8E-01-2.4E-01i,+3.9E-01+1.1E+00i,-1.2E+00+2.8E-01i,
    12. -7.6E-01-4.9E-01i,+2.5E+00-1.2E+00i,+2.2E+00-1.2E+00i,+3.6E+00+5.9E+00i,
    16. -3.1E+00-1.6E+00i,-2.5E+00-5.0E-01i,-2.2E+00+1.5E+00i,

 adjoint ndft, vector f_hat, adr=110025af0
     0. +8.6E+00-2.5E+00i,+2.4E+01+2.7E+00i,+1.4E+00+1.9E+00i,+1.1E+01+2.1E+01i,
     4. +1.4E+01+1.2E+01i,-8.8E+00+3.4E+00i,-2.8E+00+1.5E+01i,+8.5E-01+9.2E+00i,
     8. -5.0E+00+1.0E+00i,+7.0E+00+3.8E+00i,+1.3E+01-8.1E+00i,-3.5E+00-3.0E+00i,
    12. +9.4E+00+1.7E+01i,+1.5E+01+5.2E+00i,

 adjoint nfft, vector f_hat, adr=110025af0
     0. +8.6E+00-2.5E+00i,+2.4E+01+2.7E+00i,+1.4E+00+1.9E+00i,+1.1E+01+2.1E+01i,
     4. +1.4E+01+1.2E+01i,-8.8E+00+3.4E+00i,-2.8E+00+1.5E+01i,+8.5E-01+9.2E+00i,
     8. -5.0E+00+1.0E+00i,+7.0E+00+3.8E+00i,+1.3E+01-8.1E+00i,-3.5E+00-3.0E+00i,
    12. +9.4E+00+1.7E+01i,+1.5E+01+5.2E+00i,
```

Exercise 2:
The initialisation with no precomputation of $\mathbf{B}$, compressed storage, and explicit storage are given by

```
nfft_init_guru(&p, 2, N, N[0]*N[1], n, 4,
               PRE_PHI_HUT| MALLOC_F_HAT| MALLOC_X| MALLOC_F |
               FFTW_INIT| FFT_OUT_OF_PLACE,
               FFTW_ESTIMATE| FFTW_DESTROY_INPUT);
```

```
nfft_init_guru(&p, 2, N, N[0]*N[1], n, 4,
               PRE_PHI_HUT| PRE_PSI| MALLOC_F_HAT| MALLOC_X| MALLOC_F |
               FFTW_INIT| FFT_OUT_OF_PLACE,
               FFTW_ESTIMATE| FFTW_DESTROY_INPUT);
```

and

```
nfft_init_guru(&p, 2, N, N[0]*N[1], n, 4,
               PRE_PHI_HUT| PRE_FULL_PSI| MALLOC_F_HAT| MALLOC_X| MALLOC_F |
               FFTW_INIT| FFT_OUT_OF_PLACE,
               FFTW_ESTIMATE| FFTW_DESTROY_INPUT);
```

For a transform with $N_0 = 70$ and $N_1 = 90$, i.e. $N_0 N_1 = 6300$ Fourier coefficients, $n_0 = 140$, $n_1 = 180$ and $M = 6300$ evaluation nodes, the computation times are something like

Results on JUMP:

transform	cpu time (secs.)
ndft	9.88e+00
nfft, no precomputation	3.957500e-02
nfft, `PRE_PSI`	9.604000e-03
nfft, `PRE_FULL_PSI`	3.064000e-03