Chemnitz University of Technology
Michael Pippig and Daniel Potts

Jülich Research Center
CECAM Summer School

Programming with Nonequispaced FFT

Lab 2 **Parallel C Library Hands On**

The following steps must be done, before you can start with the exercises:

1. Login to Judge via `ssh userid@judge.fz-juelich.de`
2. Copy the tests from `/usr/local/scafacos/wednesday/pfft_tests` to your home directory via `cp -r /usr/local/scafacos/wednesday/pfft_tests $HOME`
3. Go into the test directory and build the tests via `cd ~/pfft_tests && make`

Exercise 1 (Testing PFFT):
Lookup and open the source file `pfft_check.c` found in `~/pfft_tests/`. Skim through the `main` routine. Try to understand what it does. Then, run the actual executable `pfft_check.x` found in `~/pfft_tests/build` with 8 processes, i.e.,

```
mpiexec -np 8 ~/pfft_tests/build/pfft_check.x
```

Exercise 2 (Testing PNFFT):
Lookup and open the source file `pnfft_test.c` found in `~/pfft_tests/`. Skim through the `main` routine. Try to understand what it does. Then, run the actual executable `pnfft_test.x` found in `~/pfft_tests/build` with 8 processes, i.e.,

```
mpiexec -np 8 ~/pfft_tests/build/pnfft_test.x
```

Where is the difference to Exercise 1?

Exercise 3 (Accuracy check for PFFT):
Since the approach from exercise 2 did not yield a check for the PNFFT, we try something different.

Lookup and open the source file `pnfft_check.c` found in `~/pfft_tests/`. Skim frist through the subroutine `pnfft_perform_guru`. Try to understand what it does. Then, have a look at the two calls of this subroutine in the `main` routine. What is the idea behind this error check? Then, run the actual executable `pnfft_check.x` found in `~/pfft_tests/build` with 8 processes, i.e.,

```
mpiexec -np 8 ~/pfft_tests/build/pnfft_check.x
```

Explore the behavior of the PNFFT error for different parameters. Therefore, you can start the executable `pnfft_check.x` again with some additional command line arguments:

- Change the real space cutoff with the argument `-pnfft_m` followed by one number between 1 and 8.
- Change the number of Fourier coefficients with the argument `-pnfft_N` followed by 3 even numbers.
- Change the size of the FFT grid with the argument `-pnfft_n` followed by three even numbers.
- Change the process mesh with the argument `-pnfft_np` followed by the three dimensions of the mesh.

For example you can call

```
mpiexec -np 2 ~/pfft_tests/build/pnfft_check.x -pnfft_m 3 -pnfft_N 8 8 8 -pnfft_np 1 2 1
```

If the check fails, think about the choice of parameters.

Exercise 4 (Advanced: Fix the check from Exercise 2):
In exercise 2 we found out, that the subroutine `init_random_nodes` in `pnfft_test.c` was used to initialize random nodes to calculate the NFFT.

Lookup and open the source file `pnfft_test_adv.c` found in `~/pfft_tests/`. There is little difference to `pnfft_test.c`. Now subroutine `init_equispaced_nodes` is used to initialize the nodes, but the body of this subroutine is missing. Add the missing lines to complete the check. Don't forget to call `make` after every change in the source file.

Hint: We have to choose the nodes equal to the grid points of an equispaced FFT. But how can we compute the needed parameters form the inputs of `init_equispaced_nodes`?