

Fast Fourier Transformation

Implementation of FFT

- the web page, link Fast Fourier Transformation (FFT):
[link http://msekcce.karlin.mff.cuni.cz/~dolejsi/Vyuka/NS_source/FFT/index.html](http://msekcce.karlin.mff.cuni.cz/~dolejsi/Vyuka/NS_source/FFT/index.html)
- go through the code `FourierTrans1.f90` to see the algorithmization of FFT
- use code `FourierTrans1.f90` and compare the computational times for DFF, FFT with/without recursion, verify computational costs $O(N^2)$ and $O(N \log_2 N)$
- change the input to observe how the output looks like
- choose a piecewise constant function as input and try to eliminate the fast frequencies to observe how the backward output corresponds to original input, you can use the code `FourierTrans2.f90` from the link above

More practical examples

- install three codes from the archive `fft.tgz`, simply use `make` command and run the codes using drivers routines
- use some of these codes for data from [link www.netlib.org/scilib/fft.dat](http://www.netlib.org/scilib/fft.dat), you should write a simple code which read data, compare results with the results on this link.