Fourier Transforms

Pierluigi Monaco, attivita` di Abilita` Informatiche, marzo 2019

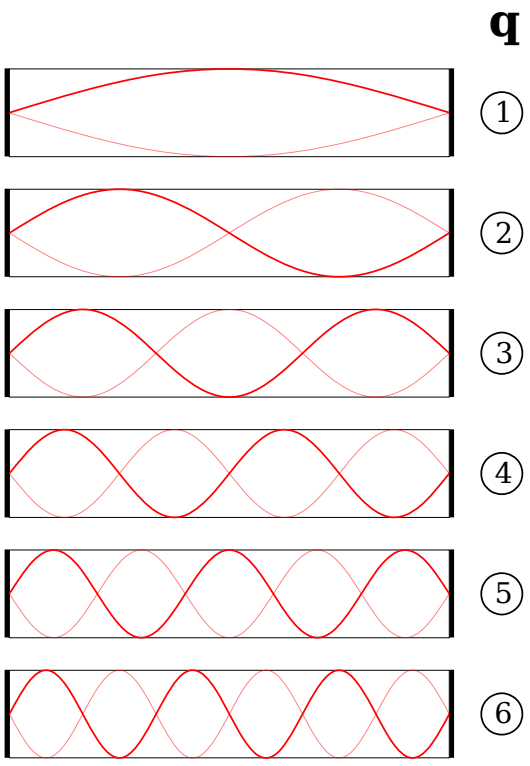
Fourier transforms:

$$k=\frac{2\pi}{\lambda} \qquad \tilde{f}(k)=\int_{-\infty}^{\infty} f(x)e^{-ikx}, \qquad f(x)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\tilde{f}(k)e^{ikx}$$

Discrete Fourier Transform (DFT):

$$f_k=\sum_{n=0}^{N-1} f_x e^{-\frac{2\pi i q n}{N}}, \qquad f_x=\frac{1}{N}\sum_{q=0}^{N-1} f_k e^{\frac{2\pi i q n}{N}}$$

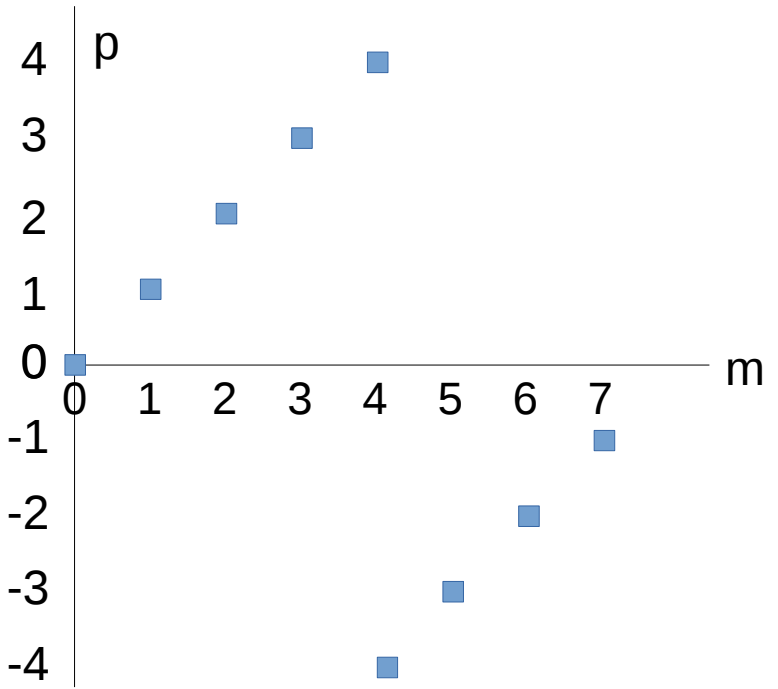
$$D=\frac{L}{N}, \quad x=nD, \quad k=\pi\frac{q}{L}$$



Nyquist frequency: $p=\frac{N}{2}, \qquad k_{Ny}=\pi\frac{N}{2L}=\frac{\pi}{2D}$

Symmetries: $\tilde{f}(-k)=\tilde{f}^{*}(k), \qquad \tilde{f}(k_{Ny})=\tilde{f}(-k_{Ny})$

Modes in memory:



Introduction

FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DCT/DST). We believe that FFTW, which is free software, should become the FFT library of choice for most applications.

The latest official release of FFTW is version **3.3.8**, available from our download page. Version 3.3 introduced support for the AVX x86 extensions, a distributed-memory implementation on top of MPI, and a Fortran 2003 API. Version 3.3.1 introduced support for the ARM Neon extensions. See the release notes for more information.

The FFTW package was developed at MIT by Matteo Frigo and Steven G. Johnson.

Our benchmarks, performed on on a variety of platforms, show that FFTW's performance is typically superior to that of other publicly available FFT software, and is even competitive with vendor-tuned codes. In contrast to vendor-tuned codes, however, FFTW's performance is *portable*: the same program will perform well on most architectures without modification. Hence the name, "FFTW," which stands for the somewhat whimsical title of "**Fastest Fourier Transform in the West**."

Subscribe to the fftw-announce mailing list to receive release announcements (or use the web feed).

Features

FFTW 3.3.8 is the latest official version of FFTW (refer to the release notes to find out what is new). Here is a list of some of FFTW's more interesting features:

- Speed. (Supports SSE/SSE2/Altivec, since version 3.0. Version 3.3.1 supports AVX and ARM Neon.)
- Both one-dimensional and **multi-dimensional** transforms.
- **Arbitrary-size** transforms. (Sizes with small prime factors are best, but FFTW uses O(N log N) algorithms even for prime sizes.)
- Fast transforms of **purely real** input or output data.
- Transforms of real even/odd data: the discrete cosine transform (DCT) and the discrete sine transform (DST), types I-IV. (Version 3.0 or later.)
- Efficient handling of **multiple, strided** transforms. (This lets you do things like transform multiple arrays at once, transform one dimension of a multi-dimensional array, or transform one field of a multi-component array.)
- Parallel transforms: parallelized code for platforms with SMP machines with some flavor of threads (e.g. POSIX) or OpenMP. An MPI version for distributed-memory transforms is also available in FFTW 3.3.
- **Portable** to any platform with a C compiler.
- Documentation in HTML and other formats.
- Both **C** and **Fortran** interfaces.
- Free software, released under the GNU General Public License (GPL—see FFTW license). (Non-free licenses may also be purchased from MIT, for users who do not want their programs protected by the GPL.

Exercise: numerical derivative of a 2D function

$$f(\vec{x}) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \tilde{f}(\vec{k}) e^{i\vec{k}\cdot\vec{x}}$$

$$\vec{\nabla} f = \vec{\nabla} \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \tilde{f}(\vec{k}) e^{i\vec{k}\cdot\vec{x}}$$

$$= \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \tilde{f}(k) \vec{\nabla} e^{i\vec{k}\cdot\vec{x}} = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} (i\vec{k}) \tilde{f}(k) e^{i\vec{k}\cdot\vec{x}}$$