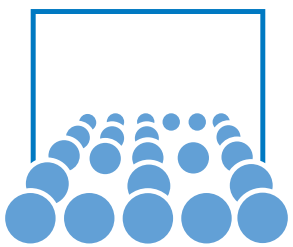
Algorithms of Scientific Computing

FFT on Real-valued Data

Tobias Neckel, Dirk Pflüger

Summer Term 2010



DFT and Symmetry

	INPUT		TRANSFORM
real symmetry	f_n is real	$\longrightarrow$	Real-valued DFT (RDFT)
even symmetry	$f_n = f_{-n}$	$\longrightarrow$	Discrete Cosine Transform (DCT)
odd symmetry	$f_n = -f_{-n}$	$\longrightarrow$	Discrete Sine Transform (DST)

Real-valued DFT (RDFT)

For real-valued input data $f_n \in \mathbb{R}$ (i.e. $f_n^* := \overline{f_n} = f_n$):

$$F_k = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n e^{-i2\pi nk/N} = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \left(\cos\left(\frac{2\pi nk}{N}\right) - i \sin\left(\frac{2\pi nk}{N}\right)\right).$$

Properties:

- $\mathrm{Re}\{F_k\} = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \cos\left(\frac{2\pi nk}{N}\right)$,

  $\mathrm{Im}\{F_k\} = -\frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \sin\left(\frac{2\pi nk}{N}\right)$
- only N independent, real-valued coefficients necessary, since:

$$F_k^* = \frac{1}{N} \sum f_n^* \left\{\omega_N^{-nk}\right\}^* = \frac{1}{N} \sum f_n \omega_N^{-n(-k)} = F_{-k}$$

Real-valued DFT (RDFT) (2)

$$\left\{ f_{-\frac{N}{2}-1}, \ldots, \quad f_0, \quad \ldots, \quad f_{\frac{N}{2}} \right\}$$

$$\text{DFT} \Downarrow \quad \Uparrow \text{IDFT}$$

$$\left\{ F_0, \mathrm{Re}\{F_1\}, \mathrm{Im}\{F_1\}, \quad \ldots, \quad \mathrm{Re}\{F_{\frac{N}{2}-1}\}, \mathrm{Im}\{F_{\frac{N}{2}-1}\}, F_{\frac{N}{2}} \right\}$$

Real-valued DFT (RDFT) (3)

Situation:

- only N real-valued input data (since N times imaginary part 0)
- only N independent, **real-valued** coefficients due to symmetry $F_{-k} = F_k^*$

Wanted: new transformation:

$$N \text{ real-valued input data} \rightarrow N \text{ real-valued coefficients}$$

Hence: Insert symmetry $F_{-k} = F_k^*$ in IDFT!

Real-valued DFT (RDFT) (4)

"Real-valued discrete Fourier transform"

- formulation 1

$$F_k = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \left(\cos\left(\frac{2\pi nk}{N}\right) - i \sin\left(\frac{2\pi nk}{N}\right) \right) , k = 0, \ldots, \frac{N}{2}$$

- formulation 2

$$\mathrm{Re}\{F_k\} = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \cos\left(\frac{2\pi nk}{N}\right) , k = 0, \ldots, \frac{N}{2}$$

$$\mathrm{Im}\{F_k\} = -\frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \sin\left(\frac{2\pi nk}{N}\right) , k = 1, \ldots, \frac{N}{2} - 1$$

Inverse Real-valued DFT

We start with an input vector (Fourier coefficients) of length $2N$; then:

$$
\begin{aligned}
f_n &= \sum_{k=-N+1}^{N} F_k e^{i2\pi nk/2N} \\
&= F_0 + \sum_{k=1}^{N-1} \left(F_k e^{i2\pi nk/2N} + F_{-k} e^{-i2\pi nk/2N} \right) + F_N e^{i2\pi nN/2N} \\
&= F_0 + \sum_{k=1}^{N-1} \left(F_k e^{i2\pi nk/2N} + \left\{ F_k e^{i2\pi nk/2N} \right\}^* \right) + F_N e^{i\pi n} \\
&= F_0 + 2 \sum_{k=1}^{N-1} \mathrm{Re} \left\{ F_k e^{i2\pi nk/2N} \right\} + F_N e^{i\pi n} \\
&= F_0 + 2 \sum_{k=1}^{N-1} \left(\mathrm{Re}\{F_k\} \cos\left(\frac{\pi nk}{N} \right) - \mathrm{Im}\{F_k\} \sin\left(\frac{\pi nk}{N} \right) \right) + F_N \cos(\pi n)
\end{aligned}
$$

Inverse Real-valued DFT (2)

Set $a_k := 2\mathrm{Re}\{F_k\}$ and $b_k := -2\mathrm{Im}\{F_k\}$ (but $a_0 := \mathrm{Re}\{F_0\}$ and $a_N := \mathrm{Re}\{F_N\}$) to get :

$$f_n = a_0 + \sum_{k=1}^{N-1} \left(a_k \cos\left(\frac{\pi nk}{N}\right) + b_k \sin\left(\frac{\pi nk}{N}\right)\right) + a_N \cos(\pi n)$$

"Real-valued inverse discrete Fourier transform"

Using the (real-valued) formula for F_k:

$$a_k = \frac{1}{N} \sum_{n=-N+1}^{N} f_n \cos\left(\frac{\pi nk}{N}\right), \qquad b_k = \frac{1}{N} \sum_{n=-N+1}^{N} f_n \sin\left(\frac{\pi nk}{N}\right)$$

Real-valued Trigonometric Interpolation

Interpretation of the real-valued DFT as an interpolation problem:

- $2N$ ansatz functions:

$$\begin{aligned} g_k(x) &:= \cos(kx) \qquad k = 0, \ldots, N \\ h_k(x) &:= \sin(kx) \qquad k = 1, \ldots, N-1 \end{aligned}$$

- $2N$ supporting points: $x_n := \dfrac{2\pi n}{2N} = \dfrac{\pi n}{N} \qquad n = -N+1, \ldots, N$
- $2N$ interpolation conditions:

$$f_n = a_0 + \sum_{k=1}^{N-1} \left(a_k \cos\left(\frac{\pi nk}{N}\right) + b_k \sin\left(\frac{\pi nk}{N}\right)\right) + a_N \cos(\pi n)$$

Fast Real-valued DFT

Computation of the real-valued DFT using complex FFT is inefficient: N redundant components (symmetrie)

Possibilities to ameliorate the efficiency:

- compute two real-valued DFT from one complex FFT
- compute a real-valued DFT of length $2N$ from one complex FFT of length N
- “compact” real-valued FFT – use symmetry of the data directly in the algorithm

Two Real-valued DFT from one complex FFT

Idea: for real-valued g_n and h_n: compute DFT of $f_n := g_n + ih_n$:

$$F_k = \frac{1}{N} \sum_n (g_n + ih_n) \omega_N^{-nk}$$

Comparison with coefficients G_k and H_k of the two real-valued DFT:

$$G_k = \frac{1}{N} \sum_n g_n \omega_N^{-nk} \qquad H_k = \frac{1}{N} \sum_n h_n \omega_N^{-nk}$$

Due to linearity of the Fourier transform:

$$F_k = G_k + iH_k$$

Two Real-valued DFT from one complex FFT (2)

Since g_n and h_n are real-valued data, the following symmetry holds:

$$G_k = G^*_{-k} \qquad H_k = H^*_{-k} \ .$$

Hence, for F^*_{-k} follows

$$F^*_{-k} = (G_{-k} + iH_{-k})^* = \left(G^*_{-k} + i^* H^*_{-k}\right) = G_k - iH_k \ .$$

Together with $F_k = G_k + iH_k$, we obtain

$$G_k = \frac{1}{2}\left(F_k + F^*_{-k}\right) \qquad \text{and} \qquad H_k = -\frac{i}{2}\left(F_k - F^*_{-k}\right) \ .$$

Two real-valued DFT from one complex FFT – Algorithm

Algorithm to compute two real-valued DFT:

(1) set $f_n := g_n + ih_n$

(2) compute F_k from FFT (using a library, e.g.)

(3) compute G_k and H_k according to

$$G_k = \frac{1}{2}\left(F_k + F^*_{-k}\right) \qquad \text{and} \qquad H_k = -\frac{i}{2}\left(F_k - F^*_{-k}\right)$$

$\Rightarrow$ "half" of the costs for a real-valued FFT

but: additional operations for pre- and postprocessing

Real-valued DFT of length $2N$ from complex FFT of length N

Compute DFT of a real-valued vector $(f_{-N+1}, \ldots, f_N)$:

$$F_k = \frac{1}{2N} \sum_{-N+1}^{N} f_n \omega_{2N}^{-nk} \qquad \text{for} \quad k = -\frac{N}{2} + 1, \ldots, \frac{N}{2}$$

Split up in $g_n := f_{2n}$ and $h_n := f_{2n-1}$; leads to butterfly scheme:

$$\begin{aligned} F_k &= \tfrac{1}{2}\left(G_k + \omega_{2N}^k H_k\right) , \\ F_{k\pm N} &= \tfrac{1}{2}\left(G_k - \omega_{2N}^k H_k\right) \end{aligned}$$

for $k = -\frac{N}{2} + 1, \ldots, \frac{N}{2}$, respectively.

Real-valued $2N$-DFT from complex N-FFT

Compute G_k and H_k from complex FFT applied to $z_n := g_n + ih_n = f_{2n} + if_{2n-1}$:

$$G_k = \frac{1}{2}\left(Z_k + Z_{-k}^*\right) \qquad \text{and} \qquad H_k = -\frac{i}{2}\left(Z_k - Z_{-k}^*\right)$$

Combine both schemes to:

$$\begin{aligned} F_k &= \frac{1}{4} Z_k \left(1 - i\omega_{2N}^k\right) + \frac{1}{4} Z_{-k}^* \left(1 + i\omega_{2N}^k\right), & k &= 0, \ldots, \frac{N}{2} \\ F_{k+N} &= \frac{1}{4} Z_k \left(1 + i\omega_{2N}^k\right) + \frac{1}{4} Z_{-k}^* \left(1 - i\omega_{2N}^k\right), & k &= -\frac{N}{2} + 1, \ldots, 0 \end{aligned}$$

Real-valued $2N$-DFT from complex N-FFT – Algorithm

Algorithm for a real-valued $2N$-DFT:

(1) set $z_n := f_{2n} + if_{2n-1}$

(2) compute Z_k from FFT applied on z_n (using a library, e.g.)

(3) compute F_k according to

$$\begin{aligned} F_k &= \tfrac{1}{4} Z_k \left(1 - i\omega_{2N}^k\right) + \tfrac{1}{4} Z_{-k}^* \left(1 + i\omega_{2N}^k\right), & k = 0, \ldots, \tfrac{N}{2} \\ F_{k+N} &= \tfrac{1}{4} Z_k \left(1 + i\omega_{2N}^k\right) + \tfrac{1}{4} Z_{-k}^* \left(1 - i\omega_{2N}^k\right), & k = -\tfrac{N}{2} + 1, \ldots, 0 \end{aligned}$$

$\Rightarrow$ Complexity determined by complex N-FFT

plus: additional operations for pre- and postprocessing

Compact Real-valued FFT

Compute DFT of a real-valued vector $(f_{-N+1}, \ldots, f_N)$:

$$F_k = \frac{1}{2N} \sum_{-N+1}^{N} f_n \omega_{2N}^{-nk} \qquad \text{for} \quad k = 0, \ldots, N$$

Split up in $g_n := f_{2n}$ and $h_n := f_{2n-1}$; leads to butterfly scheme:

$$\begin{aligned} F_k &= \tfrac{1}{2}\left(G_k + \omega_{2N}^k H_k\right) && \text{für } k = 0, \ldots, \tfrac{N}{2}\ , \\ F_{k+N} &= \tfrac{1}{2}\left(G_k - \omega_{2N}^k H_k\right) && \text{für } k = -\tfrac{N}{2}+1, \ldots, 0\ . \end{aligned}$$

G_k and H_k are coefficients of a real-valued DFT of length N; hence:

$$G_k = G_{-k}^* \qquad \text{and} \qquad H_k = H_{-k}^* \qquad \text{for } k = 0, \ldots, \tfrac{N}{2} - 1$$

Compact Real-valued FFT (2)

Use symmetry of G_k and H_k for the computation of F_k:

$$\begin{aligned} F_k &= \frac{1}{2}\left(G_k + \omega_{2N}^k H_k\right) && \text{für } k = 0, \ldots, \frac{N}{2}\,, \\ F_{N-k} &= \frac{1}{2}\left(G_{-k} - \omega_{2N}^{-k} H_{-k}\right) \\ &= \frac{1}{2}\left(G_k - \omega_{2N}^k H_k\right)^* && \text{for } k = 0, \ldots, \frac{N}{2} - 1 \end{aligned}$$

$\Rightarrow$ Computation of F_k (for $k = 0, \ldots, N$) reduced to the computation of G_k and H_k (for $k = 0, \ldots, \frac{N}{2}$, respectively).

“Edson’s algorithm” (1968)