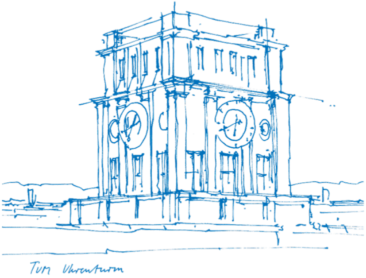
# Algorithms of Scientific Computing

## Discrete Sine Transform (DST)

Michael Bader
Technical University of Munich

Summer 2018



# DFT and Symmetry

| | INPUT | | TRANSFORM |
|---|---|---|---|
| **real symmetry** | $f_n \in \mathbb{R}$ | $\rightarrow$ | Real DFT (RDFT) |
| **even symmetry** | $f_n = f_{-n}$ | $\rightarrow$ | Discrete Cosine Transform (DCT) |
| **odd symmetry** | $f_n = -f_{-n}$ | $\rightarrow$ | Discrete Sine Transform (DST) |

| "QUARTER-WAVE" | INPUT | | TRANSFORM |
|---|---|---|---|
| **even symmetry** | $f_n = f_{-n-1}$ | $\rightarrow$ | QW-DCT |
| **odd symmetry** | $f_n = -f_{-n-1}$ | $\rightarrow$ | QW-DST |

# Real-valued Input Data with “Odd” Symmetry

Given: $2N$ input data $f_{-N+1}, \ldots, f_N$, all $f_n \in \mathbb{R}$, with

$$f_{-n} = -f_n, \qquad \text{in particular} \quad f_0 = f_N = f_{-N} = 0$$

The DFT then has the following form:

$$\begin{aligned} F_k &= \frac{1}{2N} \sum_{n=-N+1}^{N} f_n \omega_{2N}^{-nk} \\ &= \frac{1}{2N} \left( \underbrace{f_0}_{=0} + \sum_{n=1}^{N-1} \left( f_n \omega_{2N}^{-nk} + f_{-n} \omega_{2N}^{nk} \right) + \underbrace{f_N}_{=0} \omega_{2N}^{-Nk} \right) \\ &= \frac{1}{2N} \sum_{n=1}^{N-1} f_n \left( \omega_{2N}^{-nk} - \omega_{2N}^{nk} \right) = \frac{-i}{N} \sum_{n=1}^{N-1} f_n \sin\left( \frac{\pi nk}{N} \right). \end{aligned}$$

# Symmetry in the Coefficients

Transform to $f_n$ with symmetry $f_{-n} = -f_n$ gives:

$$F_k = \frac{-i}{N} \sum_{n=1}^{N-1} f_n \sin\left(\frac{\pi nk}{N}\right) \qquad \text{for } k = -N+1, \ldots, N.$$

Same symmetry in the coefficients $F_k$:

$$F_{-k} = \frac{-i}{N} \sum_{n=1}^{N-1} f_n \sin\left(\frac{\pi n(-k)}{N}\right) = \frac{-i}{N} \sum_{n=1}^{N-1} f_n \left(-\sin\frac{\pi nk}{N}\right) = -F_k$$

$\Rightarrow$ leads to the same (up to scaling) **"discrete sine transform"**

# Discrete Sine Transform (DST)

From DFT of real-valued, odd symmetric data:

$$F_k = -\frac{i}{N} \sum_{n=1}^{N-1} f_n \sin\left(\frac{\pi nk}{N}\right), \quad k = 1, \ldots N-1 \; .$$

Analogue calculation for IDFT gives:

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

$\Rightarrow$ definition of the discrete sine transform ($\widehat{F}_k := iF_k$):

$$\widehat{F}_k = \frac{1}{N} \sum_{n=1}^{N-1} f_n \sin\left(\frac{\pi nk}{N}\right), \qquad f_n = 2 \sum_{k=1}^{N-1} \widehat{F}_k \sin\left(\frac{\pi nk}{N}\right),$$

# Computation of the Discrete Sine Transform

Via pre-/postprocessing:

**(1)** generate $2N$ vector with odd symmetry

$$x_{-k} = -x_k \qquad \text{for } k = 1, \ldots N-1$$
$$x_0 = x_N = 0$$

**(2)** coefficients $X_k$ via fast, real-valued FFT on vector $x$

**(3)** postprocessing: $\widehat{X}_k = -\operatorname{Im}\{X_k\}$ for $k = 1, \ldots, N-1$.

**(4)** if necessary: scaling

# Recall: New Transforms and Symmetries

We obtain a new pair of transforms:

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

- both transforms work on **data sets** that **are neither symmetric nor periodic**
- for the particular case of the DST, we have in mind that $f_0 = f_N = 0$
- if we extend the data sets according to the symmetry rules, then the reflected (and thus symmetric) sets become periodic
- the two transforms are connected to the DFT and iDFT via a 3-step procedure:
  1. extend/duplicate the data set in a symmetric way
  2. apply the DFT/iDFT
  3. extract the symmetric half of the transformed data set
- this equivalence has two important consequences:
  1. we may compute the sine transforms ($N-1$ numbers that require sums over $N-1$ terms $\Rightarrow \mathcal{O}(N^2)$ operations) by using an FFT in step 2 $\Rightarrow$ reduces work to $\mathcal{O}(N \log N)$
  2. we prove that DST and iDST are inverse operations to each other (as the DFT and iDFT are inverse to each other on the respective symmetric data)

# Summary: Survey on DCT/DST Variants

Symmetry properties $\Leftrightarrow$ how is data continued at boundaries:

| beg. \ end | even | odd |
|---|---|---|
| even | x | x |
| odd | x | x |

$\Rightarrow$ 4 possibilities

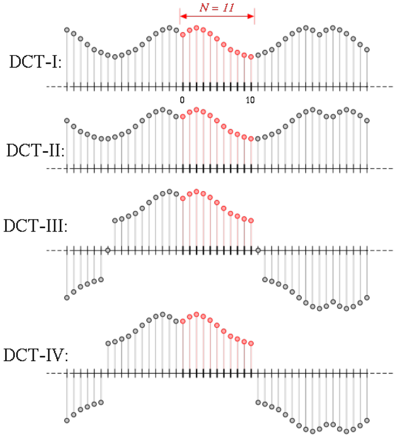
| beg. \ end | mirror | copy |
|---|---|---|
| mirror | x | x |
| copy | x | x |

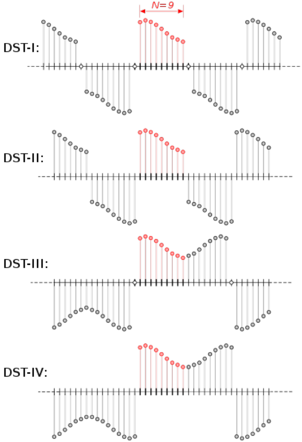
$\Rightarrow$ 4 possibilities

$\Rightarrow$ in total: 16 possibilities (8 DCT, 8 DST)

# Summary: Survey on DCT/DST Variants (2)

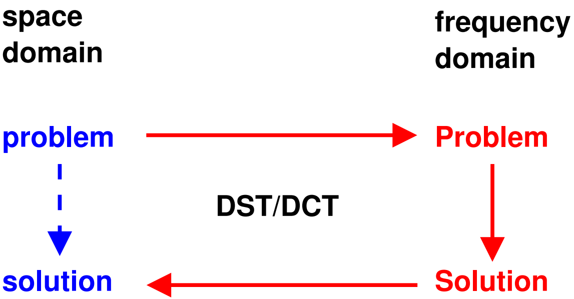
Common schemes of DCT (left) and DST (right) (images: Wikipedia):



# Application: DCT/DST for PDE (Spectral Methods)

nice application: DST for Fast Poisson Solver




**Attention**: limits/problems for using DFT with PDE include

- irregular (i.e. non-rectangular) domains
- variable coefficients in problem

⇒ other methods: FVM, FEM (fast linear solvers, multigrid, etc.)