Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
Dipl.-Inf. G. Buse, M.Sc. K. Rahnema

SS 2012
Sheet 3
9.5.2012

Algorithms of Scientific Computing

Discrete Cosine Transformation

In Exercise 4 from the last worksheet, we showed that the Fourier coefficients

$$F_k = \frac{1}{2N} \sum_{n=-N+1}^{N} f_n \omega_{2N}^{-kn} \qquad (1)$$

are real values only and can be written as:

$$F_k = \frac{1}{N} \left(\frac{1}{2} f_0 + \sum_{n=1}^{N-1} f_n \cos\left(\frac{\pi n k}{N}\right) + \frac{1}{2} f_N \cos(\pi k) \right). \qquad (2)$$

if our data set $f_{-N+1}, \ldots, f_N$, fulfills the following symmetry constraint:

$$f_{-n} = f_n \qquad \text{for } n = 1, \ldots, N-1$$

Exercise 1: Discrete Cosine Transformation (continued)

Let `FFT(f,N)` be a procedure that computes the coefficients F_k efficiently (according to equation (1)) from a field `f` which consists of $2N$ values f_n. (The result is written back to `f`)

Write a short procedure `DCT(g,N)` which uses procedure `FFT` to compute the coefficients F_k for $k = 0, \ldots, N$ from equation (2) for the (non-symmetrical) data $f_0, \ldots, f_N$, stored in the parameter field g.

Excercise 2: Fast Discrete Cosine Transformation

Formulate the butterfly scheme for equation (1) from excercise 1. Divide the dataset f_n of length $2N$ into a dataset $g_n := f_{2n}$, containing all values with an even index, and a dataset $h_n := f_{2n-1}$, with all values with odd index. Which symmetries can be found in g_n and h_n? Of which kind (Cosine/Sine Transformation, DFT with real data) are the according DFTs of length N? Which symmetries can be found if the dataset f_n fulfills the following symmetry constraint:

$$f_{-n} = f_{n+1}.$$

Fast Poisson Solver

We will focus in this excercise on systems of linear equations, which are created, amongst others, during the discretization of differential equations (in this worksheet we will use the Poisson equation $-\Delta u = f$).

In the one-dimensional case we have the unknowns $u_1, \ldots, u_{N-1}$ in the equations

$$-u_{n+1} + 2u_n - u_{n-1} = f_n, \quad n = 1, \ldots, N-1 \tag{3}$$

With boundary values $u_0 = 0, u_N = 0$.

In the two-dimensional case we have the unknowns $u_{n,m}$ with $n, m = 1, \ldots N-1$ in the equations

$$-u_{n,m+1} - u_{n+1,m} + 4u_{n,m} - u_{n-1,m} - u_{n,m-1} = f_{n,m}, \quad n, m = 1, \ldots, N-1 \tag{4}$$

Here we need a whole bunch of virtual boundary values, which are all assumed to be zero:

$$u_{n,0} = u_{n,N} = u_{0,n} = u_{N,n} = 0, \quad n = 1, \ldots, N-1.$$

Excercise 3: Fast Poisson Solver for the 2d Problem

In 1d the Fast Poisson Solver is not very helpfull, since the tridiagonal system of equations (3) can be solved by a Gauß elimination in $\mathcal{O}(N)$ operations. So we will try to extend this method to the 2d case.

Like in 1d we insert the transformed values

$$u_{n,m} = 2\sum_{k=1}^{N-1}\sum_{l=1}^{N-1} U_{k,l} \sin\frac{\pi nk}{N} \sin\frac{\pi ml}{N}, \qquad f_{n,m} = 2\sum_{k=1}^{N-1}\sum_{l=1}^{N-1} F_{k,l} \sin\frac{\pi nk}{N} \sin\frac{\pi ml}{N} \tag{5}$$

into the system of equations (4).

Show that this time you can set the $U_{k,l}$, depending on the $F_{k,l}$, like this:

$$U_{k,l} = \frac{F_{k,l}}{4 - 2\cos\frac{\pi k}{N} - 2\cos\frac{\pi l}{N}} \qquad \text{für} \quad k, l = 1, \ldots, N-1 \tag{6}$$

So, again the $U_{k,l}$ can be retrieved directly form the $F_{k,l}$. Formulate an algorithm for 2d, which solves the system of equations (4) efficiently by using the Fast Sine Transform and the given dependency.

Discussion

What happens if we have nonhomogeneous Dirichlet boundary conditions? How about Neuman boundary conditions?