Technische Universität München
Lehrstuhl für Informatik V
Dr. T. Neckel, Dr. D. Pflüger

SS 2011
Sheet 7
29.6.2011

Algorithms of Scientific Computing (Algorithmen des Wissenschaftlichen Rechnens)

Hierarchization in Higher Dimensions, Combination Technique

Exercise 1: Hierarchization in Higher Dimensions

In this exercise we will implement the multi-recursive algorithm for hierarchization of a multi-dimensional regular sparse grid. The *SparseGrid* class in the Python source file *sparsegrid.py* provides a code skeleton in which some parts are missing.

(i) Fill the gaps in *__init__*, *__hierarchizeMainAxisRecursively* and *__hierarchizeRecursively*. For the specifications look at the functions' document strings and follow the instructions in the comments.
Hint: The basic algorithm is very similar to *__insertGridPointsRecursively*.

(ii) Fill in the function body of *computeVolume*.

Exercise 2: The Combination Technique – A Different View on Sparse Grids

Dealing with hierarchical bases often turns out to be sophisticated. On this worksheet we will therefore see how the so-called *combination technique* finds a sparse grid interpolant, that approximates a function on a number of full grids, each consisting only of a "relatively small" number of grid points.

Let $u_{\underline{l}}$ ($\underline{l} \in \mathbb{N}^2$) for a $u : [0,1]^2 \to \mathbb{R}$ the interpolant in $V_{\underline{l}}$ (interpolating piecewise bilinearly at the inner grid points, at the boundary u is assumed to be zero again).

(i) $V_{\underline{l}}$ can be decomposed into a set of subspaces $W_{\underline{l}}$. Accordingly, the interpolant $u_{\underline{l}} \in V_{\underline{l}}$ can be written as a sum of $w_{\underline{l}} \in W_{\underline{l}}$.

Spot the grid associated with $u_{(3,2)}$ in the right part of Figure 1. Identify those subspaces in the left part that are needed to reconstruct $u_{(3,2)}$.

(ii) Use the result from (i) to rewrite

$$\sum_{|\underline{l}|_1 = n+1} u_{\underline{l}}\,, \qquad n \in \mathbb{N}$$

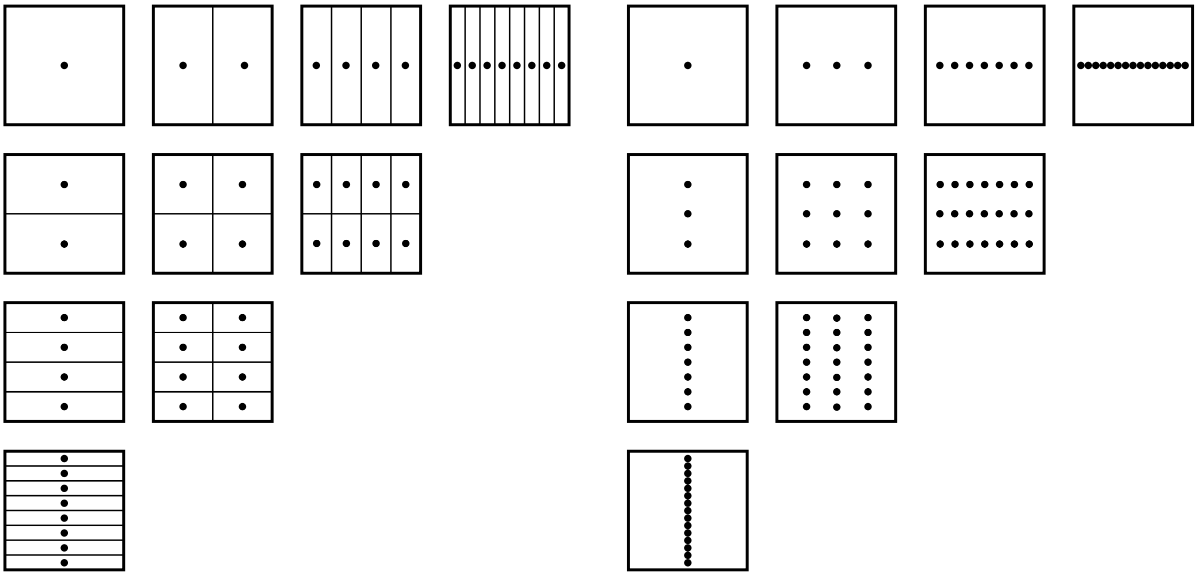

Figure 1: The two parts in the picture show the grid points and supports associated with interpolants $w_{\underline{l}}$ (left) and $u_{\underline{l}}$ (right) up to level 4 for the 2d case.

for the two-dimensional case as a weighted sum of $w_{\underline{l}}$.

Hint: Look at the subspace scheme in Figure 1 and count the occurrences of each subspace in the sum. What do you notice when comparing $w_{\underline{l}}$ with common level $n = |\underline{l}|_1 + dim - 1$?

(iii) In the final step use the previous results to give a representation of the sparse grid interpolant

$$u_n^D := \sum_{|\underline{l}|_1 \leq n+1} w_{\underline{l}}$$

as a weighted sum of $u_{\underline{l}}$. Again, count the occurrences of the $w_{\underline{l}}$.

(iv) Assume you are talking to a person who knows how to approximate the volume $F_2(u)$ through the trapezoidal rule (in 2d) with respect to $u_{\underline{l}}$. Give instructions on how to write a program that implements a sparse grid approximation of $F_2(u)$. Remember Archimedes quadrature.

(v) Compare this method with Archimedes quadrature — what are the (dis-)advantages?