Algorithms of Scientific Computing (Algorithmen des Wissenschaftlichen Rechnens)

Adaptivity, Norms of Functions

1 One-dimensional Sparse Grids—An Adaptive Implementation

Last week we introduced Archimedes' approach to approximate the integral $F(f, a, b) = \int_a^b f(x)\,dx$ of a function $f : \mathbb{R} \to \mathbb{R}$, respectively to approximate the function f itself.

For the one-dimensional case we want to formalize this approach and generalize it in the following ways:

- Let $\phi(x)$ be the "mother of all hat functions" with

$$\phi(x) = \begin{cases} x+1 & \text{for } -1 \leq x < 0 \\ 1-x & \text{for } 0 \leq x < 1 \\ 0 & \text{else} \end{cases} \tag{1}$$

- The data structure used to store the hierarchical coefficients is now called *Sparse Grid*.
- A sparse grid is defined by a particular set of interpolation points $x_{l,i}$ and associated ansatz functions $\phi_{l,i}(x)$ with

$$\phi_{l,i}(x) = \phi(\ 2^l \cdot \left(x - i \cdot \frac{1}{2^l}\right)\ ) = \phi(2^l \cdot x - i)\ , \qquad l \in \mathbb{N}^+, i \in \{1, 3, \ldots, 2^l - 1\} \tag{2}$$

- Archimedes' approach from the lecture corresponds to a *regular* sparse grid.
- To improve the quality of approximation for arbitrary functions f we introduce spatial adaptivity.

The Python source file *worksheet5.py* is a slightly modified version of last week's file *quadrature1d.py*. It contains an additional import (*sparsegrid1d.py*) that makes the two classes *GridPoint* and *SparseGrid1d* known in the global namespace. These classes provide everything needed for an adaptive implementation of a one-dimensional sparse grid.

Your task is to implement the missing parts in the members of the *SparseGrid1d* class. Note that there are comments and class descriptions in the source file providing you with more detailed instructions.

Hint: Only if you do not know how to go on, have a look at the implementation of the *plot* member.

a) The constructor *__init__* creates a grid containing all grid points on levels $l \leq minLevel$. A given function f is then evaluated at those points before *hierarchization* is performed eventually. Implement this behavior.

b) Implement the member function *computeVolume* that computes an approximation for $F(f, 0, 1)$ using the current sparse grid interpolant.

c) Implement the member function *refineAdaptively* that takes a certain refinement criterion (see source code) and inserts new grid points accordingly.

2 Norms of Functions

When representing functions we are interested in the question how "large" a function actually is. Measuring the difference between a function and its interpolant can for example help to draw conclusions about the quality of approximation.

We only consider functions $u : [0, 1] \to \mathbb{R}$ with $u(0) = u(1) = 0$ and will mainly be interested in three norms:

- The infinity norm (German: Maximumsnorm)

$$\|u\|_\infty := \max_{x\in[0,1]} |u(x)|$$

- The L^2 norm

$$\|u\|_2 := \sqrt{\int_0^1 u(x)^2 \, dx},$$

  defined through the L^2 scalar product

$$(u, v)_2 := \int_0^1 u(x)v(x) \, dx$$

- The energy norm $\|u\|_E := \|u'\|_2$

Note: We always assume the existence of maxima, derivatives and integrals.

1. Compute these norms for

$$f_k(x) := \sin(k\pi x), \quad k \in \mathbb{N}$$

   and for

$$\phi_{l,i}(x) := \phi\left(2^l x - i\right) \quad l \in \mathbb{N}, i = 1, \ldots, 2^l - 1$$

   with $\phi(x) := \max\{1 - |x|, 0\}$.

2. For each of these norms prove the triangle inequality

$$\|u + v\| \leq \|u\| + \|v\|.$$

   For the L^2 norm use the Cauchy-Schwarz inequality

$$|(u, v)| \leq \|u\| \cdot \|v\|,$$

   that holds for arbitrary scalar products, i.e. also for the L^2 scalar product.