

Algorithms of Scientific Computing (Algorithmen des Wissenschaftlichen Rechnens)

Multi-dimensional Quadrature

In order to find an alternative way to approximate the d -dimensional integral

$$F_d := \int_{[0,1]^d} u(x_1, \dots, x_d) d(x_1, \dots, x_d)$$

we use a regular d -dimensional grid of step size $h = 2^{-n}$ stored in an array with indices in $[0, 2^n]^d$.

We don't want to be bothered with special boundary treatment so we simplify this task assuming our function is 0 on the boundary, i.e.

$$\forall \mathbf{x} = [x_1, \dots, x_d]^T : \exists j \text{ such that } (x_j = 0 \vee x_j = 1) \Rightarrow u(\mathbf{x}) = 0$$

In the next step we compute the d -dimensional hierarchical surpluses:

loop $j = 1, \dots, d$ **over the dimensions**
loop over all 1d subgrids discretizing spatial direction j
(fix all coordinates except x_j , what you'll get is a 1d array
of grid points like on worksheet 4 — for $d = 2$ this comes down
to processing all rows and all columns once)
Compute 1d surpluses in dimension j **(in place) for each subarray**

Once we have the surpluses $v_{l,i}$ (indexing with level l and point index i as introduced in the lecture) we only need to multiply them with the volume of the associated pagoda and sum everything up. The volume of the pagoda associated with grid point $x_{l,i}$ is $2^{-|l|_1}$.

1. Implement this method!

For $d = 2$ and $n = 3$ this can be done easily in a spreadsheet (using copy and paste is not only ok but also encouraged for thorough understanding).

2. Let

$$u(x_1, x_2) = 16x_1(1-x_1)x_2(1-x_2) = \left(1 - 4\left(x_1 - \frac{1}{2}\right)^2\right) \cdot \left(1 - 4\left(x_2 - \frac{1}{2}\right)^2\right).$$

Write the summed up volumes in descending order (compute with your program or derive from results of worksheet 5). Imagine n to be large.

How many volumes are at least needed to approximate the integral ($4/9$) with an absolute error not larger than $1/144$ (this choice is not random, it will give you nice results).

3. Spot and draw the used grid points in the unit square!