Chair of Scientific Computing in Computer Science
Department of Informatics
Technical University of Munich

Personal sticker

S0031

Compliance to the code of conduct
I hereby assure that I solve and submit this exam myself under my own name by only using the allowed tools listed below.

Signature or full name if no pen input available

Algorithms for Scientific Computing

Exam:	IN2001 / Endterm	**Date:**	Tuesday 27th July, 2021
Examiner:	Michael Bader	**Time:**	11:00 – 13:00

Working instructions

- This exam consists of **16 pages** with a total of **6 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 45 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - all printed or electronic scripts, textbooks or lecture material
  - your own handwritten notes
- The working time for the exam is 120 minutes plus 20 minutes submission time.
- Subproblems marked by * can be solved without results of previous subproblems.
- The small scale of boxes printed next to each subproblem indicates the maximum number of credits awarded for this subproblem. **Do not write into these boxes!**
- 17 credits will be sufficient to pass the exam. Grades will be computed relative to a maximum score of 42 points.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors.
- During the exam, no questions concerning the content of the exam questions will be answered.
- If you think that a question or exercise text contains an error or ambiguity, then choose a correction or variant that allows you to complete the solution – state how you have corrected or interpreted the exercise.

By submitting this exam you confirm that you have completed all exercises yourself, without third-party help and only using the allowed helping material.

Problem 1 **Real Quarter Wave Fourier Transformation (9 credits)**

Given is a real data-set $f_n \in \mathbb{R}$ and the corresponding Fourier coefficients F_k for the Quarter Wave Fourier transformation of length 2N:

$$F_k := \frac{1}{2N} \sum_{n=-N+1}^{N} f_n w_{2N}^{-k(n+\frac{1}{2})} \qquad f_n := \sum_{k=-N+1}^{N} F_k w_{2N}^{k(n+\frac{1}{2})} \tag{1.1}$$

The following symmetry holds for the Quarter Wave Fourier transformation

$$F_{-k} = (F_k)^*, \tag{1.2}$$

as it is also known from the lecture for the normal real Fourier transformation.

0 1 2 3

a)* Show that for the Quarter Wave coefficients F_k from equation (1) the following modified periodic condition holds:

$$F_{k+2N} = -F_k \tag{1.3}$$

Furthermore show that F_N is thus purely imaginary:

b)* Compute a purely real representation for f_n, in which only real values and functions, such as sin() and cos() appear.

0
1
2
3
4
5
6

Problem 2 Piecewise Constant Function Approximation in 1-D (6 credits)

We want to approximate a one-dimensional function $u : \mathbb{R} \to \mathbb{R}$ over the unit interval $\Omega = (0, 1)$ using a hierarchical basis $\Phi = \{\phi_{l,i}\}$ of *piecewise constant functions*

$$\phi(x) = \begin{cases} 1 & \text{for } 0 \le x < 1 \\ 0 & \text{else} \end{cases} \tag{2.1}$$

$$\phi_{l,i} = \phi(2^l \cdot x - i) \tag{2.2}$$

with $l \in \mathbb{N}_0$ and $0 \le i < 2^l$. The degree of freedom $\beta_{l,i}$ of function $\phi_{l,i}$ is located at the center of $\phi_{l,i}$'s support

$$x_{l,i} = \frac{2 \cdot i + 1}{2^{l+1}}. \tag{2.3}$$

0 1 2

a)* Give a formula how to compute a basis function $\phi_{l,i}$ as a weighted sum of basis functions $\phi_{l+1,k}$, i.e., give coefficients such that $\phi_{l,i}(x) = \sum_k p_k \phi_{l+1,k}(x)$.

Hint: No computation required. Drawing a plot can be helpful.

b) Implement in *pseudo code* (e.g. Python-like code) the recursive function `transform` (see below), which turns sampled function values (given in a hash map) into hierarchical coefficients – i.e., overwrites all function values by hierarchical coefficients in the hash map.

0 1 2 3 4

Assume the following:

- `map` is a hash map holding function values for a consistent hierarchy of grid points
- `map[(l,i)]` returns the coefficient stored for the grid point with key `(l,i)`
- `map[(l,i)] = 3` sets this grid point's coefficient to the value 3
- `map.has(l,i)` returns `true` if point `(l,i)` exists
- the level indexing is **zero-based**, so the top-level call is: `transform(0, 0, 0.0)`

There is no maximum level in this scenario: the recursive descent stops when there are no more children.

```
def transform(map, l, i, f_parent):
    '''
    @param map hash map of function values / coefficients
    @param l (zero-based) index of current level
    @param i index within level: 0 <= i < 2^l
    @param f_parent the parent's coefficient: map((l-1, i/2)
    '''
```

Problem 3 Haar Wavelets: The Discrete Wavelet Transform (7 credits)

A scaling function or *father wavelet* ϕ satisfies the dilation equation.

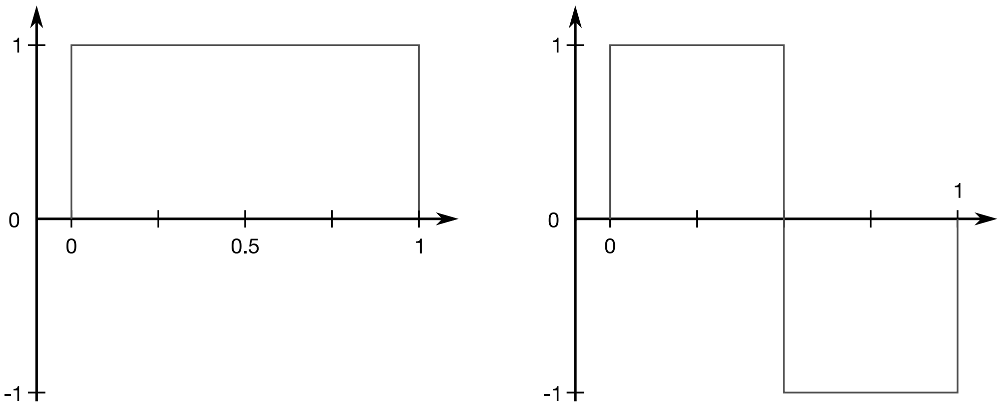
$$\phi(t) = \sum_{k\in\mathbb{Z}} h_k \cdot \phi(2t - k). \tag{3.1}$$

for certain coefficients h_k.

In this problem we look at the Discrete Wavelet Transform for the *Haar Wavelet Basis* with father and mother wavelets $\phi(x)$ and $\psi(x)$ as given in Figure 3.1. The unnormalized *refinement coefficients* h_k for this basis are

$$h_0 = h_1 = 1. \tag{3.2}$$

Let $\vec{c} = [c_0\ c_1\ \dots\ c_{n-2}\ c_{n-1}]^T$ be a signal vector and $\vec{v} = DWT(\vec{c})$ its Haar wavelet transform.



$$\phi(x) = \begin{cases} 1 & \text{for } 0 \le x \le 1 \\ 0 & \text{else} \end{cases} \qquad \psi(x) = \begin{cases} 1 & \text{for } 0 \le x \le 0.5 \\ -1 & \text{for } 0.5 < x \le 1 \\ 0 & \text{else} \end{cases}$$

Figure 3.1: Father wavelet $\phi(x)$ and mother wavelet $\psi(x)$ for the Haar wavelet basis.

a)* Compute the DWT for these Haar wavelets for the example given below using the pyramidal algorithm. For every stage of the transform mark the involved vectors $\vec{c}^{(J)}$ (average, obtained by applying the low pass filters) and $\vec{d}^{(J)}$ (detail, obtained by applying the high pass filters) that were defined in the lecture.

0 1 2 3 4

Hint: There is no need to copy parts of the vectors that do not change from one stage to the next, instead you can use an asterix (*) or cross (+) to mark such values. Also note that no $\sqrt{2}$-factors occur in this exercise.

$$\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{c^{(5)}}$$

6	-8	-2	-8	11/2	3/2	5	2	-10	4	7/2	9/2	2	4	-9	5

0 1 2 3

b)* With an $\mathbb{R}^{n\times n}$ matrix H, we can write the transformations between $\vec{c}$ and $\vec{v}$ as $\vec{c} = H\vec{v}$ and $\vec{v} = H^{-1}\vec{c}$. For $n = 8$ give the unnormalized inverse of the transformation matrix H ignoring any normalizing factor $1/\sqrt{2}$.

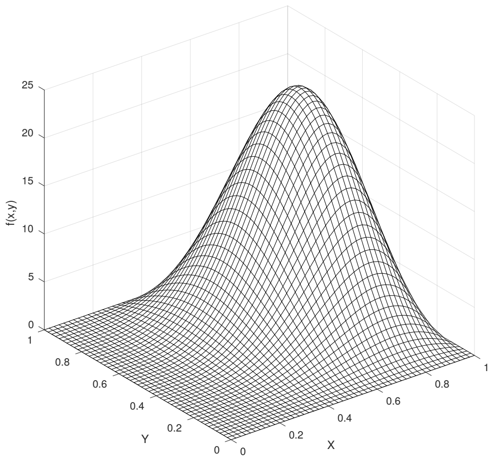
Hint: Here, $H \in \mathbb{Z}^{n\times n}$ is integer-valued and contains the basis vectors $\vec{\psi_{l,i}}$ in its columns. It is sufficient to give only the non-zero entries.

Problem 4 2D Hierarchization and Sparse Grids (10 credits)

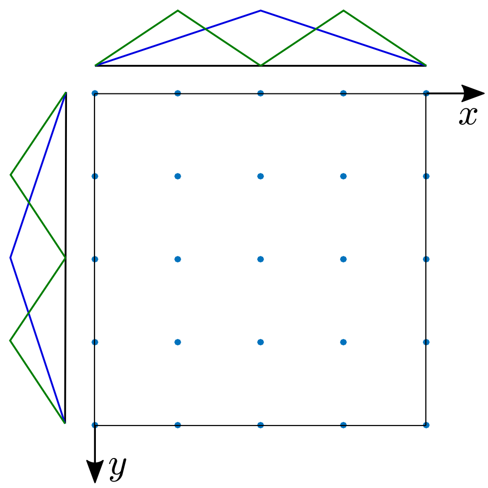
In this problem we want to interpolate the two-dimensional function $f(x, y)$ given by

$$f(x, y) = 128\, x^3 \cdot \sin(\pi x) \cdot y \cdot \sin(\pi y) \tag{4.1}$$

in the domain $\Omega = [0, 1]^2$ using the hierarchical hat function basis, as discussed in the lectures. As you can see in Figure 4.1, the function is 0 on the boundaries and the most important features are located in the area of $x \geq 0.5$ and $y \geq 0.5$.



Function to interpolate



Full grid of level (2,2)

Figure 4.1: Function and full grid

0 1 2 3

a)*

2D Hierarchization: We first use a hierarchical basis on a full grid to interpolate function f. Grid points (x_i, y_j) and hierarchical basis functions are illustrated in Figure 4.1. Table 4.1 shows the function evaluation $f(x, y)$ at each grid point (x_i, y_j). Perform 2D-hierarchization to transform the function values into hierarchical surpluses (round to **two digits** after the decimal point).

Use Tables 4.2 and 4.3 (extra tables are given on the right in case you want to correct an error) or redraw these tables on an extra sheet to compute your solution.

Table 4.1: Function values $f(x_i, y_j)$.

$y_j \backslash x_i$	0.00	0.25	0.50	0.75	1.00
0.00	0.00	0.00	0.00	0.00	0.00
0.25	0.00	0.25	2.83	6.75	0.00
0.50	0.00	0.71	8.00	19.09	0.00
0.75	0.00	0.75	8.49	20.25	0.00
1.00	0.00	0.00	0.00	0.00	0.00

Table 4.2: Hierarchization (intermediate results; use right table if you need to correct errors)

$y_j \backslash x_i$	0.00	0.25	0.50	0.75	1.00
0.00					
0.25					
0.50					
0.75					
1.00					

$y_j \backslash x_i$	0.00	0.25	0.50	0.75	1.00
0.00					
0.25					
0.50					
0.75					
1.00					

Table 4.3: Hierarchization (results; use right table if you need to correct errors)

$y_j \backslash x_i$	0.00	0.25	0.50	0.75	1.00
0.00					
0.25					
0.50					
0.75					
1.00					

$y_j \backslash x_i$	0.00	0.25	0.50	0.75	1.00
0.00					
0.25					
0.50					
0.75					
1.00					

b)*

0
1
2

2D Hierarchical Surpluses: Consider the hierarchical interpolant on $(N+1) \times (N+1)$ grids for increasing number of grid points $N = 2^L$. Characterize how the size of the surpluses decreases for sufficiently smooth functions f (state at least two properties).

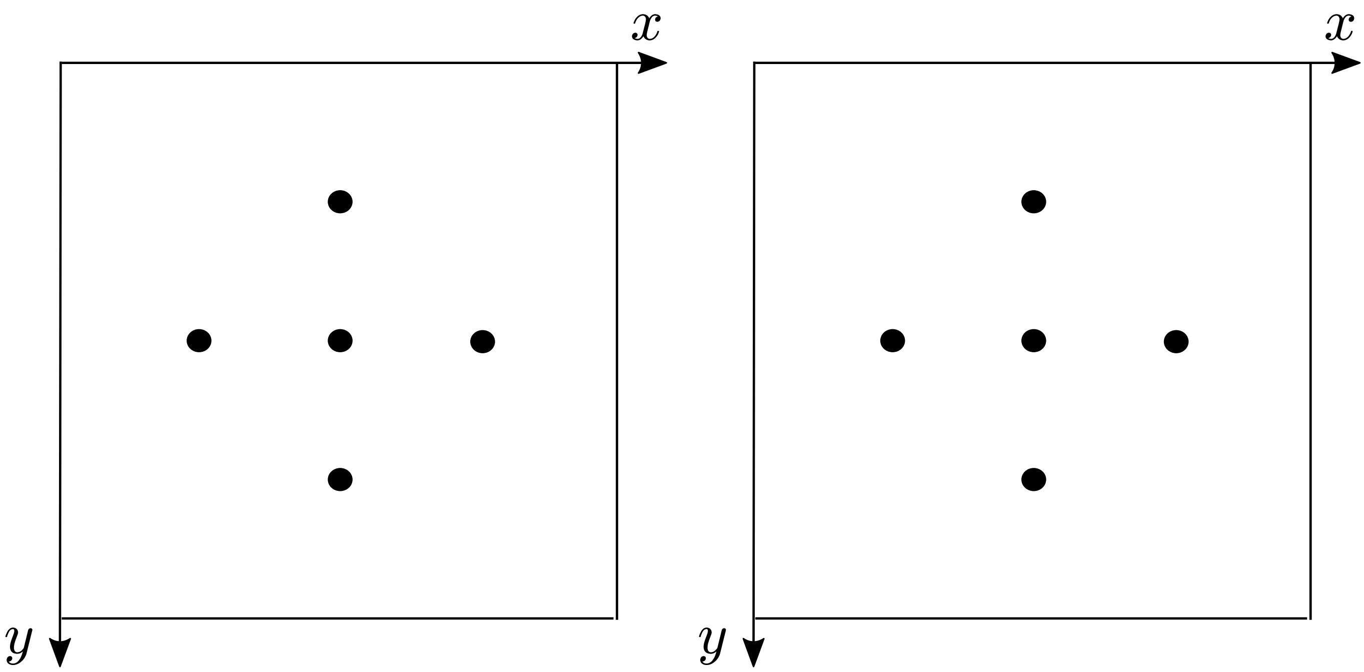
Page empty

c)*

Sparse Grid: Now we use a sparse grid as shown in Figure 4.2 to interpolate function f. Grid points have been hierarchized. A grid point can be uniquely identified by a level-index-vector pair $\vec{l}, \vec{i} := (l_x, l_y), (i_x, i_y)$.

Table 4.4: Level 2 regular sparse grid points

Point	1	2	3	4	5
$\vec{l}, \vec{i}$	$(1,1),(1,1)$	$(2,1),(1,1)$	$(2,1),(3,1)$	$(1,2),(1,1)$	$(1,2),(1,3)$
$\alpha_{\vec{l},\vec{i}}$					



Sparse grid

Backup sparse grid

Figure 4.2: Sparse grid

i) Label the grid points in the picture with numbers $1, 2, ...$ according to the first 2 lines in Table 4.4.

ii) Fill in the hierarchical surpluses $\alpha_{\vec{l},\vec{i}}$ in the third line. Use the function values provided in task a). Round to **two digits** after the decimal point.

iii) Refine the grid point with the highest hierarchical surplus value. Use the "$\times$" symbol to mark the added grid points in Figure 4.2.

iv) Refine grid point $(2,2),(3,3)$. Use the "$\triangle$" symbol to mark the added grid points in Figure 4.2.

Hint: Do not forget to check for missing parents in grid refinement.

Problem 5 Space-filling Curves – H-Index (6 credits)

0
1
2
3
4
5
6

In Figure 5.1, you see the first 3 iterations of the so called H-index space-filling curve (the arrow of the curve shows the direction). It maps the unit interval $[0, 1]$ onto the unit square $[0, 1] \times [0, 1]$.

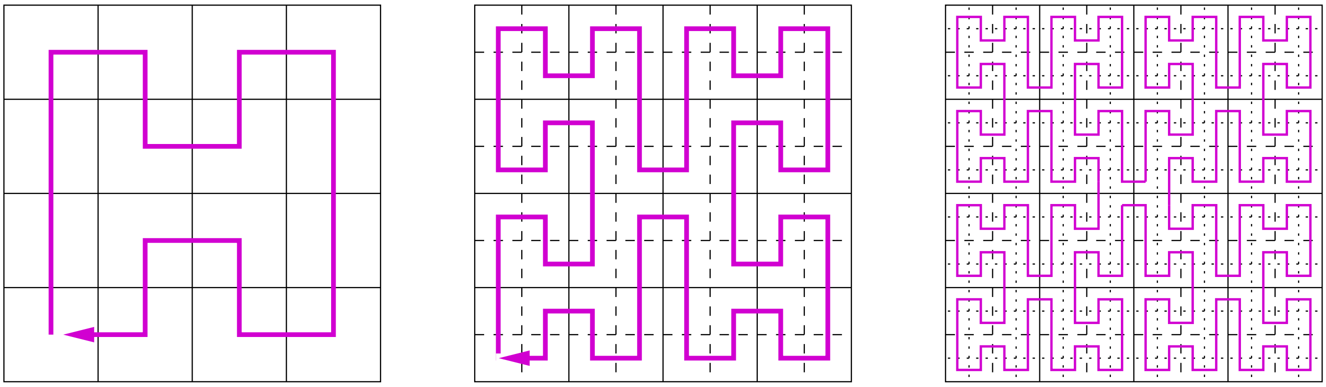


Figure 5.1: The first three iterations of the H-index curve

The iterations of the H-index curve can be described by a grammar construction, similar to the grammars discussed in the lectures. The grammar consists of the following four "terminal" productions

$$\begin{aligned} A &\longleftarrow \downarrow\rightarrow\uparrow\rightarrow\downarrow\downarrow\downarrow \\ B &\longleftarrow \downarrow\downarrow\downarrow\leftarrow\uparrow\leftarrow\downarrow \\ C &\longleftarrow \uparrow\uparrow\uparrow\rightarrow\downarrow\rightarrow\uparrow \\ D &\longleftarrow \uparrow\leftarrow\downarrow\leftarrow\uparrow\uparrow\uparrow \end{aligned}$$

plus four non-terminal productions that define the recursive construction of the H-index iterations.

(1) Mark the occurrences of the terminal productions in the iterations of Figure 5.1.

(2) State the missing four non-terminal productions for A, B, C and D.

(3) State an additional production $H \longleftarrow \ldots$ with starting symbol H that is needed to initiate the iteration.

(In the rightmost image, only mark those occurrences which you need to determine the productions.)

Problem 6 Space-filling Curves – Meurthe Order (7 credits)

0
1
2
3
4

a)*
The mapping of the so-called *Meurthe order* space-filling curve is defined via the following arithmetisation:

$$m(0_9.n_1 n_2 n_3 n_4 \ldots) = M_{n_1} \circ M_{n_2} \circ M_{n_3} \circ M_{n_4} \circ \cdots \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

with the operators M_ν given as:

$$M_0 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3}y + 0 \\ \frac{1}{3}x + 0 \end{pmatrix} \qquad M_1 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -\frac{1}{3}y + \frac{1}{3} \\ \frac{1}{3}x + \frac{1}{3} \end{pmatrix} \qquad M_2 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3}x + 0 \\ \frac{1}{3}y + \frac{2}{3} \end{pmatrix}$$

$$M_3 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3}y + \frac{1}{3} \\ -\frac{1}{3}x + 1 \end{pmatrix} \qquad M_4 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -\frac{1}{3}y + \frac{2}{3} \\ -\frac{1}{3}x + \frac{2}{3} \end{pmatrix} \qquad M_5 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3}x + \frac{1}{3} \\ -\frac{1}{3}y + \frac{1}{3} \end{pmatrix}$$

$$M_6 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3}y + \frac{2}{3} \\ \frac{1}{3}x + 0 \end{pmatrix} \qquad M_7 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -\frac{1}{3}y + 1 \\ \frac{1}{3}x + \frac{1}{3} \end{pmatrix} \qquad M_8 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3}x + \frac{2}{3} \\ \frac{1}{3}y + \frac{2}{3} \end{pmatrix}$$

Draw the first and second iteration of the Meurthe order curve.

b)* Determine a parameter t that is mapped to the centre of the unit square, i.e., find a t for which $m(t) = (\frac{1}{2}, \frac{1}{2})$.
Make sure to justify your solution! (i.e., perform the respective calculation, give a complete argumentation, or similar)

0
1
2
3

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.