Numerical Algorithms

Winter 2023/24
Prof. Dr. Carsten Burstedde
Hannes Brandt

UNIVERSITÄT BONN

Exercise Sheet 11.

Submission: **16.01.2024**

Exercise 1. (Morton Encoding)

Let $d \in \mathbb{N}_{>0}$ and $(\ell, I_\ell), (\ell', J_{\ell'})$ with $\ell, \ell' \in \mathbb{N}, I_\ell \in [0, 2^{d\ell}) \cap \mathbb{N}, J_{\ell'} \in [0, 2^{d\ell'}) \cap \mathbb{N}$ the Morton encodings of two mesh elements.

a) Prove the following properties of the `morton_parent` and `morton_child` functions introduced in the lecture

$$\texttt{morton\_parent}\,(\texttt{morton\_child}\,(\ell, I_\ell, i)) = (\ell, I_\ell) \qquad i = 0, \dots, 2^d - 1.$$

b) Propose a function `morton_sibling` (ℓ, I_ℓ, i) that computes the sibling of (ℓ, I_ℓ) with child id i.

c) Propose a function `morton_compare` $(\ell, I_\ell, \ell', J_{\ell'})$ that returns

- -1, when (ℓ, I_ℓ) precedes $(\ell', J_{\ell'})$ in the Morton order
- 1, when (ℓ', J'_ℓ) precedes (ℓ, I_ℓ) in the Morton order
- 0, when (ℓ, I_ℓ) and $(\ell', J_{\ell'})$ are identical.

If one element is an ancestor of the other, we define it to precede the other element in the Morton order.

d) Derive an algorithm that computes the Morton encoding of the nearest common ancestor of two mesh elements using the `morton_parent` function presented in the lecture.

e) Derive an algorithm that computes the Morton encoding of the nearest common ancestor of two mesh elements that operates in a direct (non-iterative, non-recursive) manner.

(1+2+3+3+3 Points)