Polynomial Splines over locally refined box-partitions

Tor Dokken*, Tom Lyche+, Kjell Fredrik Pettersen*

*SINTEF and +University of Oslo

The work is partly funded by the European Union through

- The TERRIFIC project (EU Contract 284981) www.terrific-project.eu
- The IQmulus project (EU Contract 318787) www.iqmulus.eu

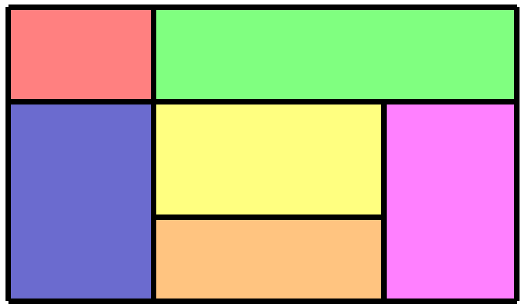
Structure of talk

- Polynomial Splines over locally refined box-partitions
- LR-mesh, constant split and LR B-spline
- The relation between T-splines and Locally Refined B-splines
- Peeling algorithm for linear independence (both valid for LR B-splines and Semi-Standard T-splines)

Box-partitions

- Box-partitions - Rectangular subdivision of regular domain d-box $\mathbb{R}^d$

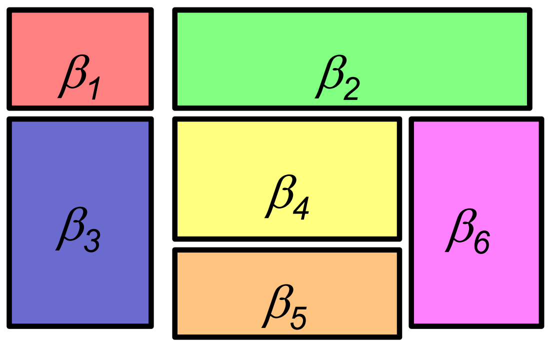
$$\Omega \subseteq \mathbb{R}^d$$
$$\Omega = [a_1, b_1] \times \cdots \times [a_d, b_d]$$
$$a_i < b_i,\ 1 \leq i \leq d$$



$\Omega \subseteq \mathbb{R}^2$

- Subdivision of Ω into smaller d -boxes

$$\mathcal{E} = \{\beta_1, \ldots, \beta_1\}$$
$$\beta_1 \cup \beta_2 \cup \cdots \cup \beta_n = \Omega$$
$$\beta_i^o \cap \beta_j^o = \emptyset,\ i \neq j$$



$\mathcal{E} = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$

Why introduce Box-partitions?

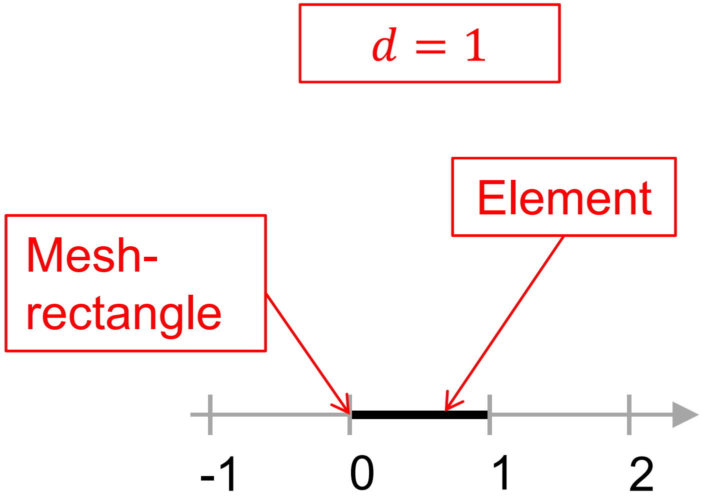
The different approaches to local refinement of splines over regular grids seem all be related to Box-partitions.

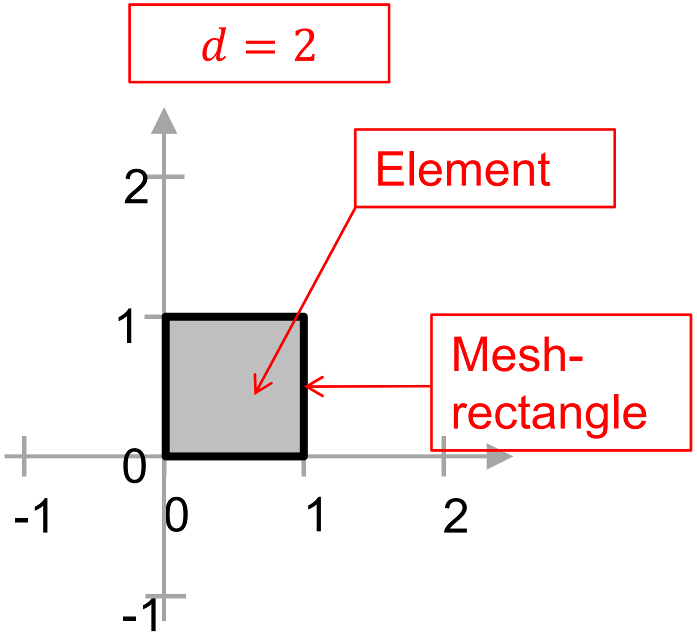
- The structure of Bezier elements of T-splines over a rectangular domain is a box-partition
- Locally Refined B-splines (LR B-splines) are defined over a box partition with multiplicities
- The knot lines of Hierarchical B-splines define a box-partition
- PHT-splines are defined over a box-partition

Local refinement in any of the approaches above lead to refinement of the Box-partition

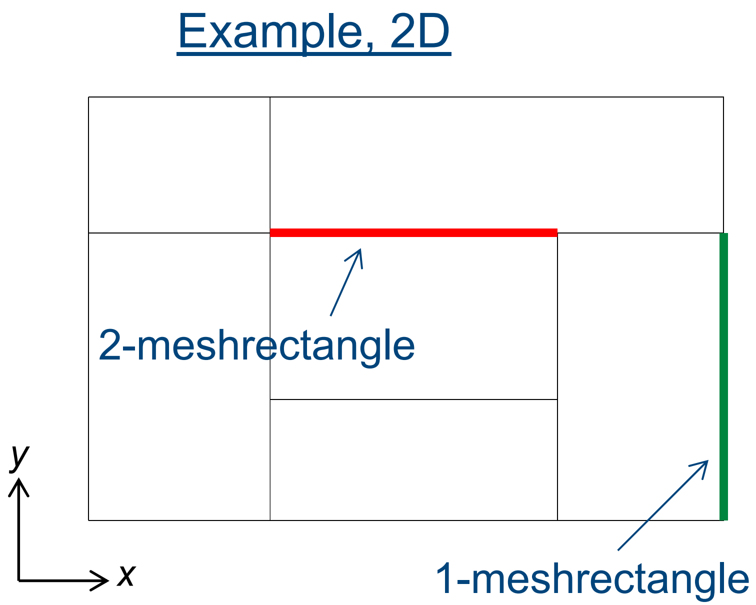
Important boxes

- If $\dim \beta = d$ then β is called an **element**.
- If $\dim \beta = d - 1$ then β is called a **mesh-rectangle**, a k-**mesh-rectangle**





Mesh-rectangles



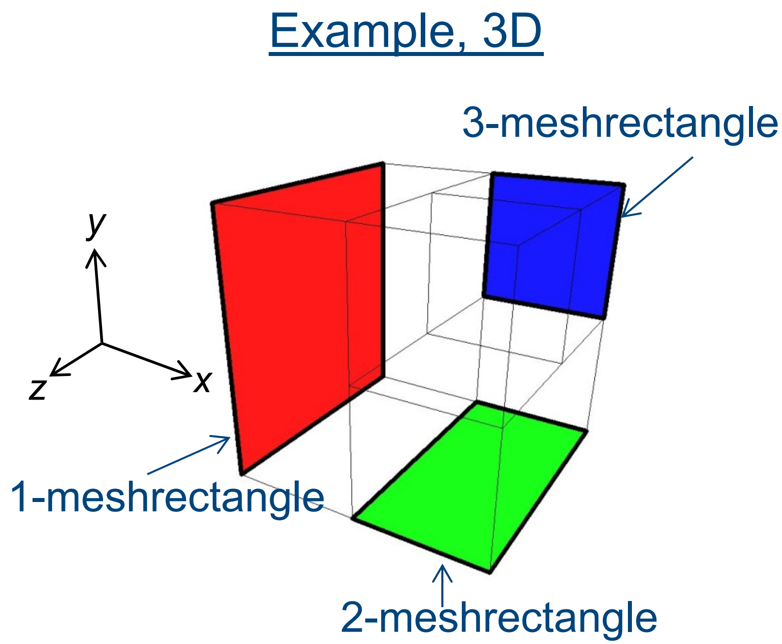


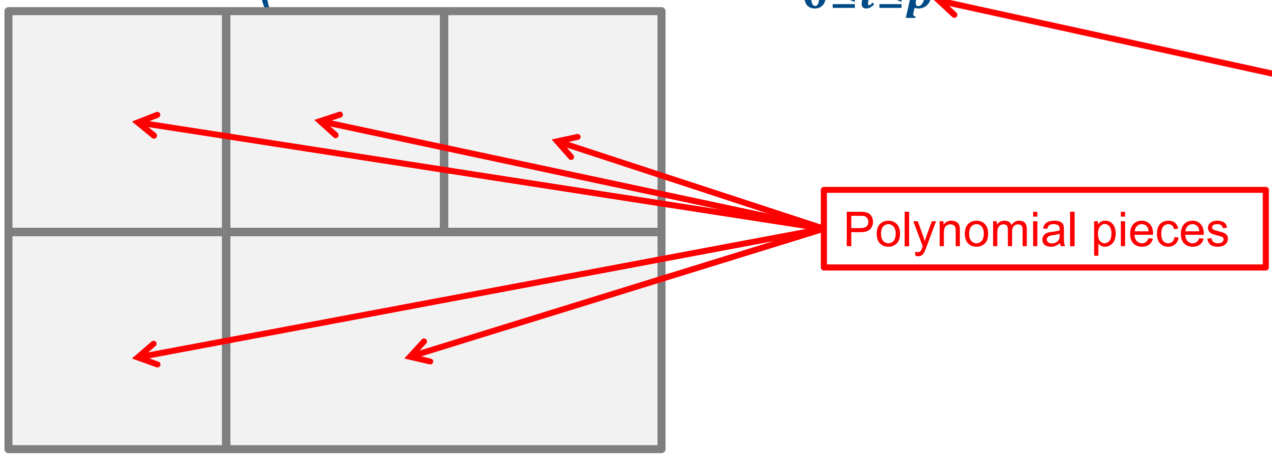
Illustration by: Kjell Fredrik Pettersen, SINTEF

Polynomials of component degree

On each element of the Box-partition the spline is a polynomial.

We define polynomials of component degree at most $p_k, k = 1, \dots, d$ by:

$$\Pi_{\boldsymbol{p}}^d = \left\{ f: \mathbb{R}^d \rightarrow \mathbb{R}: f(\boldsymbol{x}) = \sum_{\boldsymbol{0} \le \boldsymbol{i} \le \boldsymbol{p}} c_{\boldsymbol{i}} \boldsymbol{x}^{\boldsymbol{i}}, c_{\boldsymbol{i}} \text{ in } \mathbb{R} \text{ for all } \boldsymbol{i} \right\}.$$



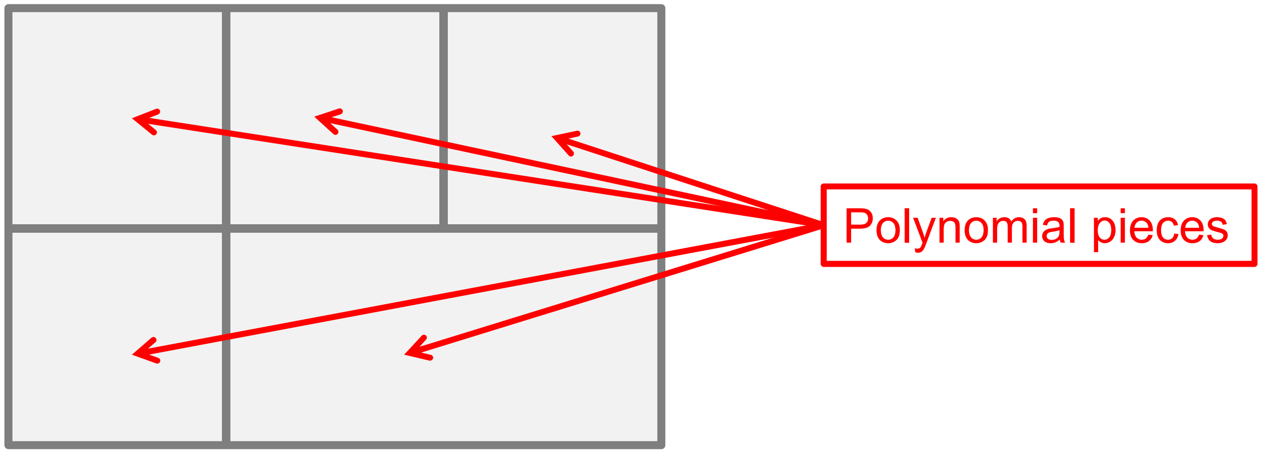
$\boldsymbol{p} = (p_1, .., p_d)$
$\boldsymbol{i} = (i_1, .., i_d)$

Piecewise polynomial space

We define the piecewise polynomial space

$$\mathbb{P}_{\boldsymbol{p}}(\boldsymbol{\mathcal{E}}) = \left\{f: [\boldsymbol{a}, \boldsymbol{b}] \rightarrow \mathbb{R}: f|_{\beta} \in \Pi_{\boldsymbol{p}}^{d}, \beta \in \tilde{\mathcal{E}}\right\},$$

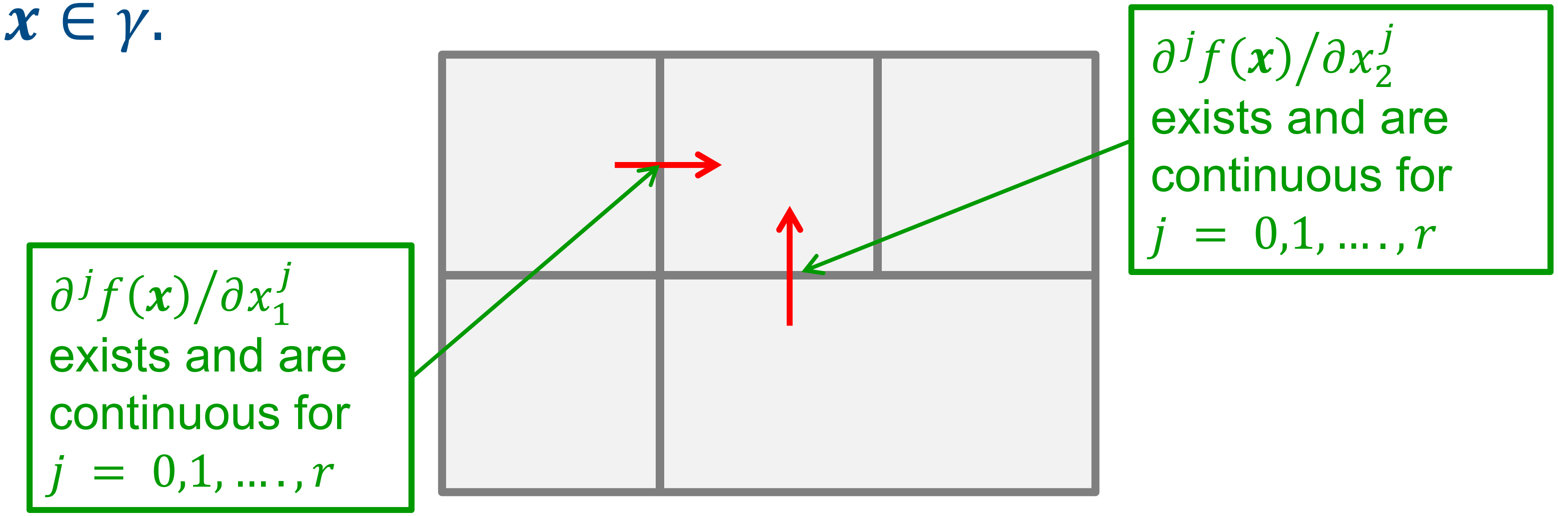
where $\tilde{\mathcal{E}}$ is obtained from $\boldsymbol{\mathcal{E}}$ using half-open intervals as for univate B-splines.



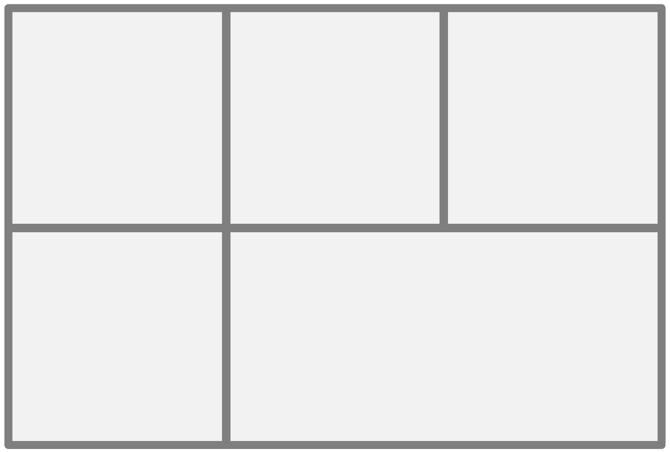
Continuity across mesh-rectangles

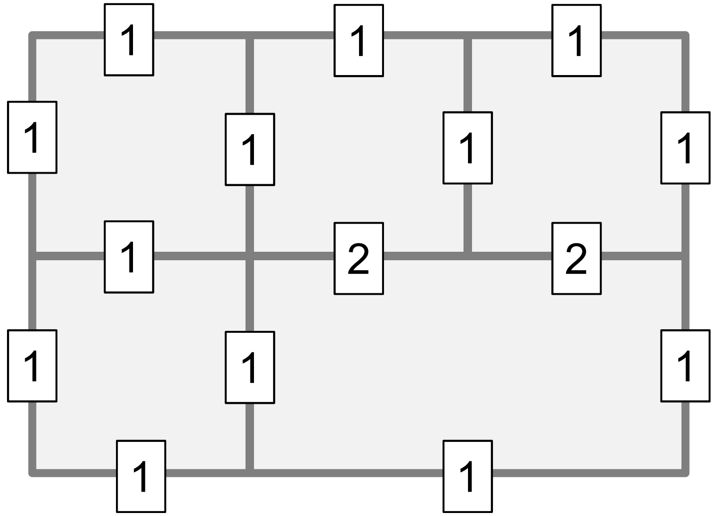
Given a function $f\colon [\boldsymbol{a}, \boldsymbol{b}] \to \mathbb{R}$, and let $\gamma \in \mathcal{F}_{d-1,k}(\mathcal{E})$ be any k-mesh-rectangle in $[\boldsymbol{a}, \boldsymbol{b}]$ for some $1 \leq k \leq d$.

We say that $f \in C^r(\gamma)$ if the partial derivatives $\partial^j f(\boldsymbol{x})/\partial x_k^j$ exists and are continuous for $j = 0,1, \ldots, r$ and all $\boldsymbol{x} \in \gamma$.



μ-extended box-mesh (multiplicities added to model continuity)





- A multiplicity μ is assigned to each mesh-rectangle
- Supports variable knot multiplicity for Locally Refined B-splines, and local lower order continuity across mesh-rectangles.
- Compatible with nonuniform univariate B-splines

Spline space

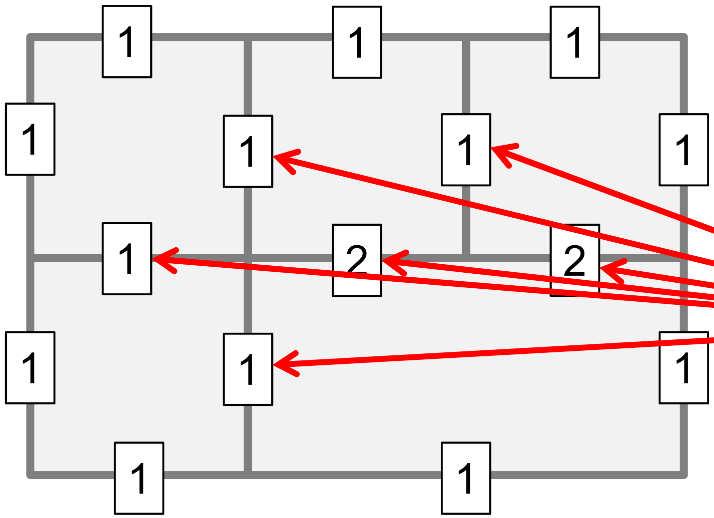
Polynomial degree in direction k

Continuity across k-mesh-rectangle γ

We define the spline space

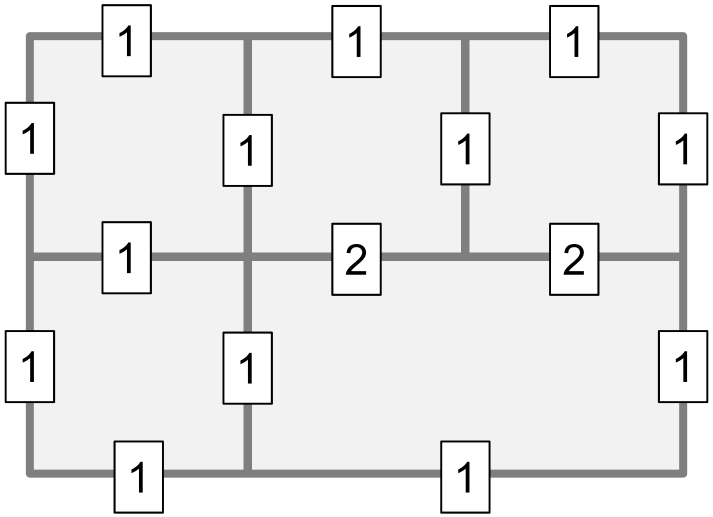
$$\mathbb{S}_{\boldsymbol{p}}(\mathcal{M},\mu) = \left\{ f \in \Pi^d_{\boldsymbol{p}}\left(\mathcal{E}(\mathcal{M})\right): f \in C^{p_k-\mu(\gamma)}(\gamma), \right.$$
$$\left. \forall \gamma \in \mathcal{F}_{d-1,k}\left(\mathcal{E}(\mathcal{M})\right), k = 1, \dots, d \right\}$$

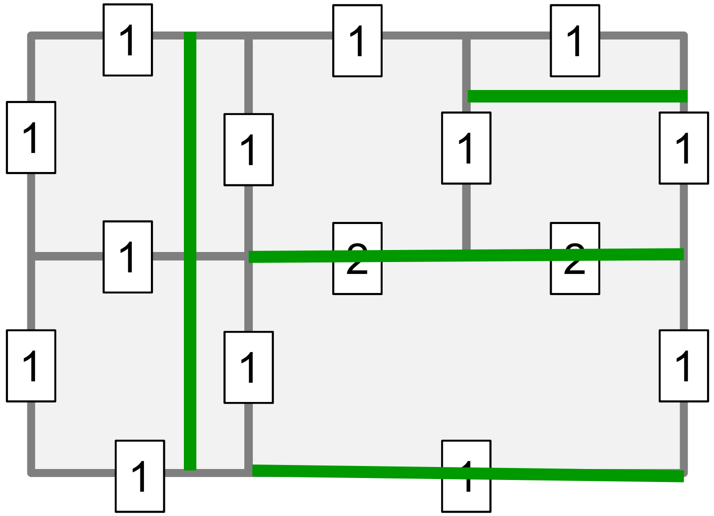
All k-mesh-rectangles



Specify multiplicity, e.g., continuity across mesh-rectangle

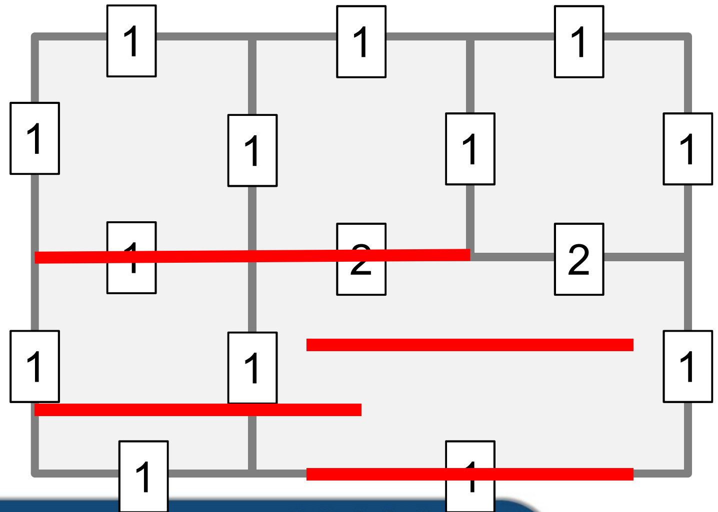
Refinement by inserting mesh-rectangles giving a constant split





Constant split

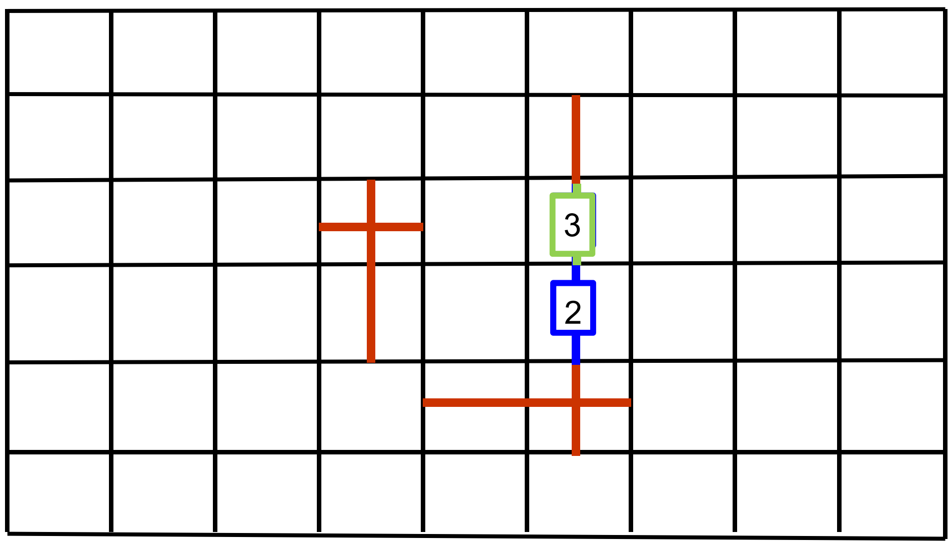
Not constant split



μ-extended LR-mesh

A μ -extended LR-mesh is a μ-extended box-mesh $(\mathcal{M},\mu)$ where either

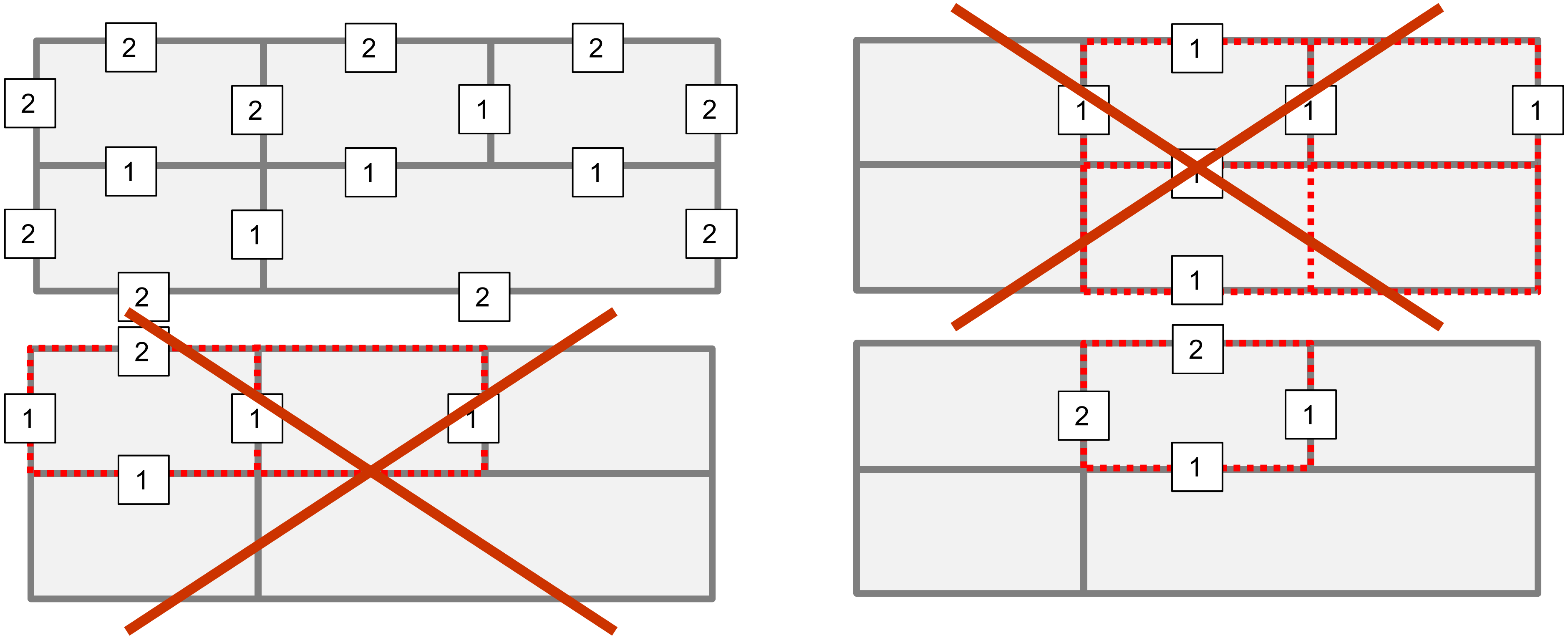
1. $(\mathcal{M},\mu)$ is a tensor-mesh with knot multiplicities or
2. $(\mathcal{M},\mu) = \left(\widetilde{\mathcal{M}} + \gamma, \tilde{\mu}_\gamma\right)$ where $\left(\widetilde{\mathcal{M}}, \tilde{\mu}\right)$ is a μ -extended LR-mesh and γ is a constant split of $\left(\widetilde{\mathcal{M}}, \tilde{\mu}\right)$.



All multiplicities not shown are 1.

LR B-spline

Let (M, μ) be a μ-extended LR-mesh in $\mathbb{R}^d$. A function $B: \mathbb{R}^d \rightarrow \mathbb{R}$ is called an LR B-spline of degree $\boldsymbol{p}$ on $(\mathcal{M}, \mu)$ if B is a tensor-product B-spline with minimal support in $(\mathcal{M}, \mu)$.



Splines on a μ-extended LR-mesh

We define as sequence of $\boldsymbol{\mu}$-extended LR-meshes $(\mathcal{M}_1, \mu_1), \dots, (\mathcal{M}_q, \mu_q)$ with corresponding collections of minimal support B-splines $\mathcal{B}_1, \dots, \mathcal{B}_q$.

$(\mathcal{M}_1, \mu_1)$,	$(\mathcal{M}_2, \mu_2)$,	...	$(\mathcal{M}_j, \mu_j)$,	$(\mathcal{M}_{j+1}, \mu_{j+1})$	...	$(\mathcal{M}_q, \mu_q)$
$\mathcal{B}_1$,	$\mathcal{B}_2$,	...	$\mathcal{B}_j$,	$\mathcal{B}_{j+1}$,	...	$\mathcal{B}_q$

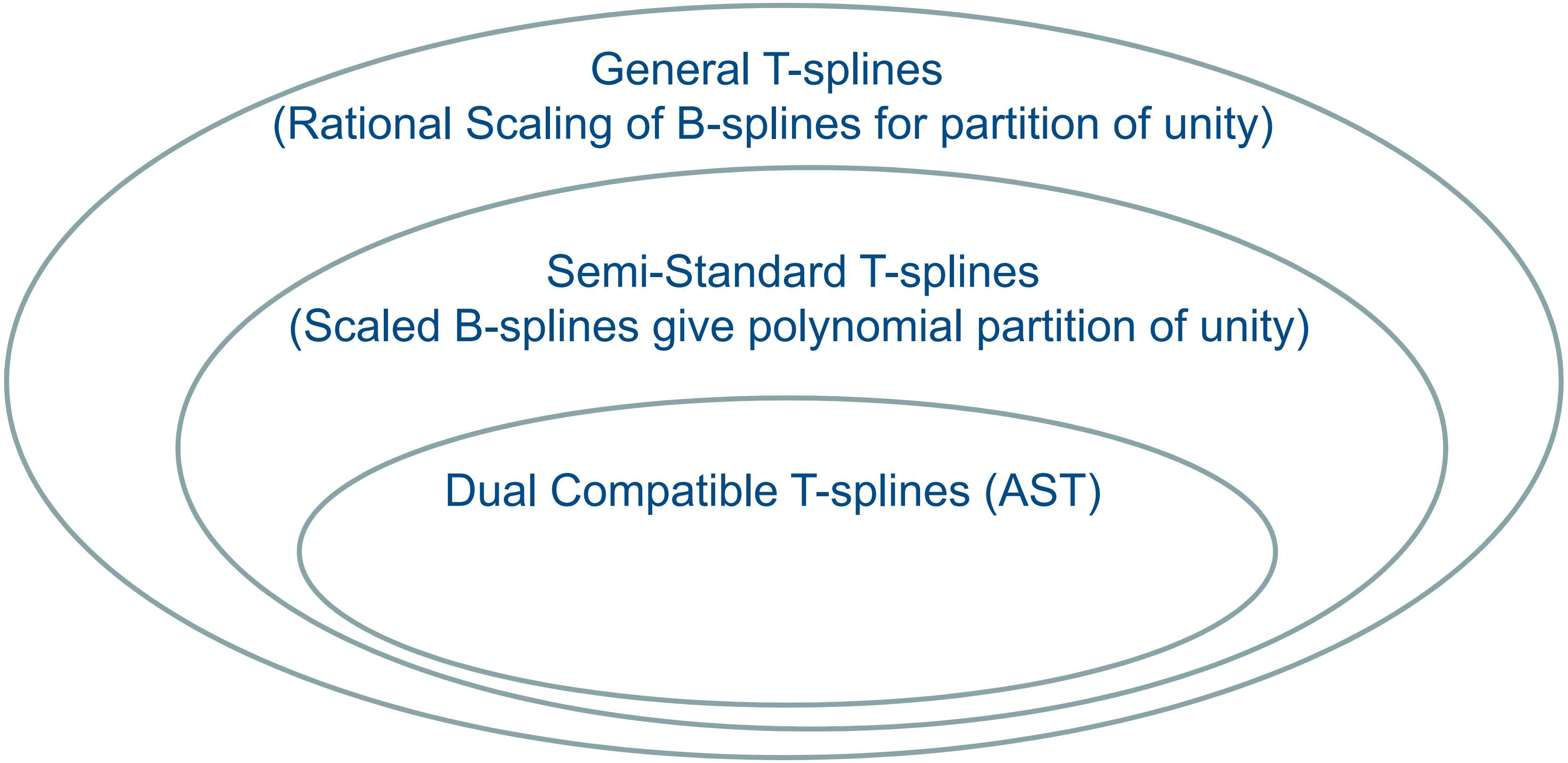
The LR B-spline rules

- Starting point tensor product B-spline basis
- Incrementally refine the spline space by splitting the support of selected B-splines by inserting mesh-rectangles. For each refinement:
  - Perform additional refinements if some B-splines do not have minimal support
  - Optional check for Linear Independence at each refinement step
    - Check dimension increase by dimension formula for spline space over box partitions in $\mathbb{R}^d,\ d \geq 2$.
    - Check that the spline space is spanned by the B-splines
    - Check that we have a basis, e.g., number of B-splines corresponds to the dimension of the spline space
- Alternative linear independency check: Run the collection of LR B-splines through the peeling Algorithm

Important properties for splines in Isogeometric Analysis

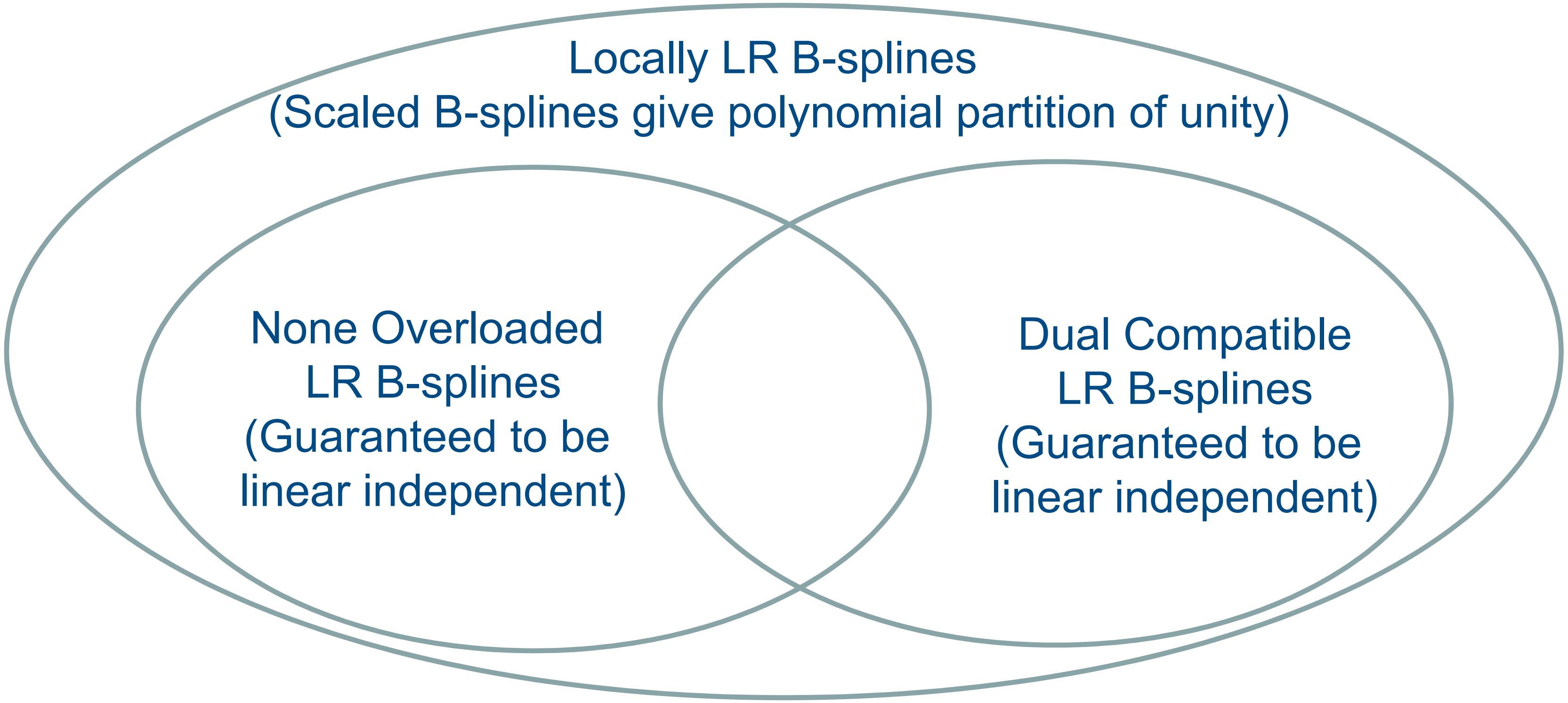
- Local refinement should give nested spline spaces
- Mesh completion to keep mesh properties
  - Does the B-spline functions span the spline space over the Box-partition
- Geometric interpretation of coefficients
  - Scaled or rational scaled shape functions form a partition of unity

T-splines



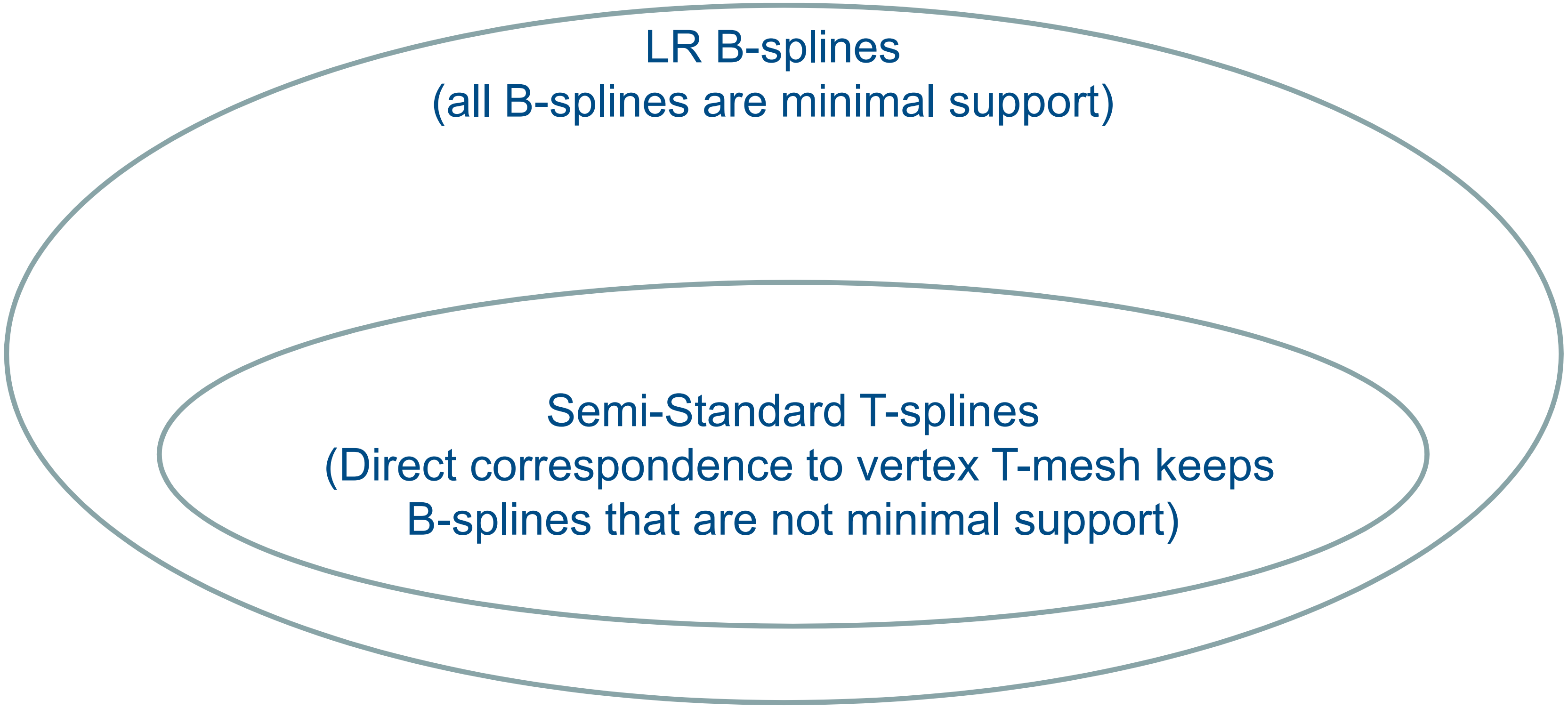
- Nestedness of splines spaces
  - General T-splines;: No
  - Semi-Standard T-splines: Yes

Locally Refined B-splines



- LR B-splines guarantee nested spline spaces
- None overloaded B-splines are linearly independent
- Dual Compatible LR B-splines and Dual Compatible T-splines (AST) are closely related.

Splines spaces of Semi-Standard T-splines seem to be included in the spline spaces of LR B-splines

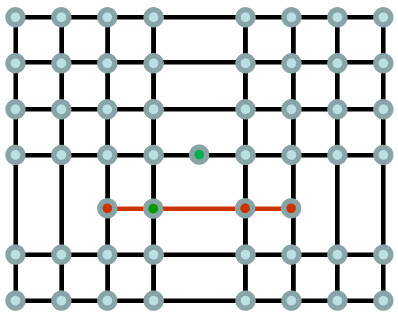


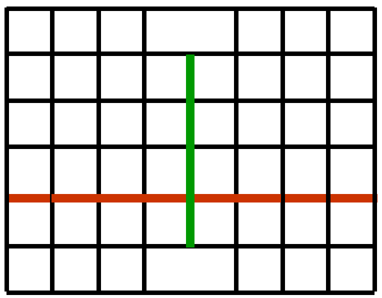
- LR B-splines keep pace with the dimension of the spline space over the box partion

Refinement approaches of T-splines and LR B-splines

T-splines

- Refinement in vertex T-grid



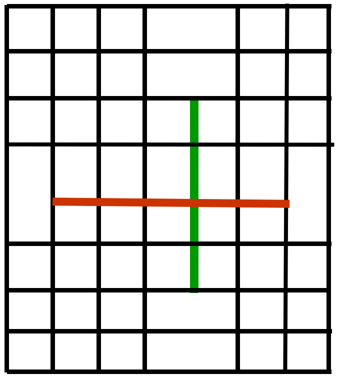


Parameter domain,
Polynomial segments

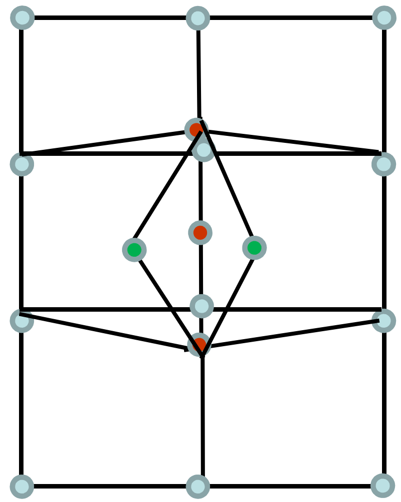
- Adding two close vertices by refinement in the T-spline vertex grid (T-grid)

LR B-splines

- Refinement in parameter domain

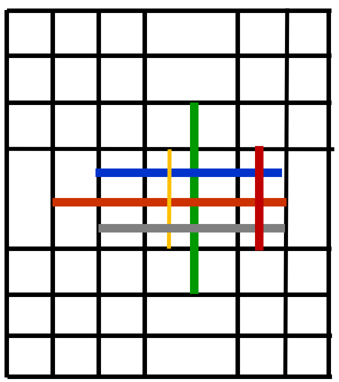


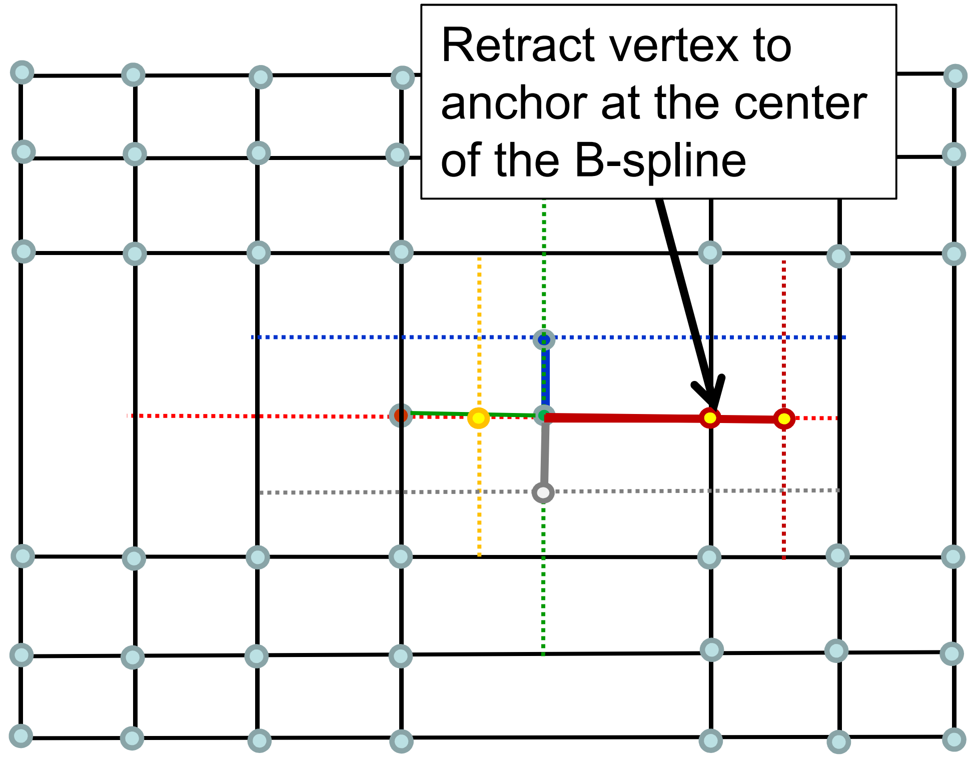
Parameter domain



- Adding a minimal "+" in the parameter domain of LR B-splines
- Position of vertices in parameter domain average of internal knots

Specify LR B-spline refinement in the vertex mesh (Bi-cubic example)

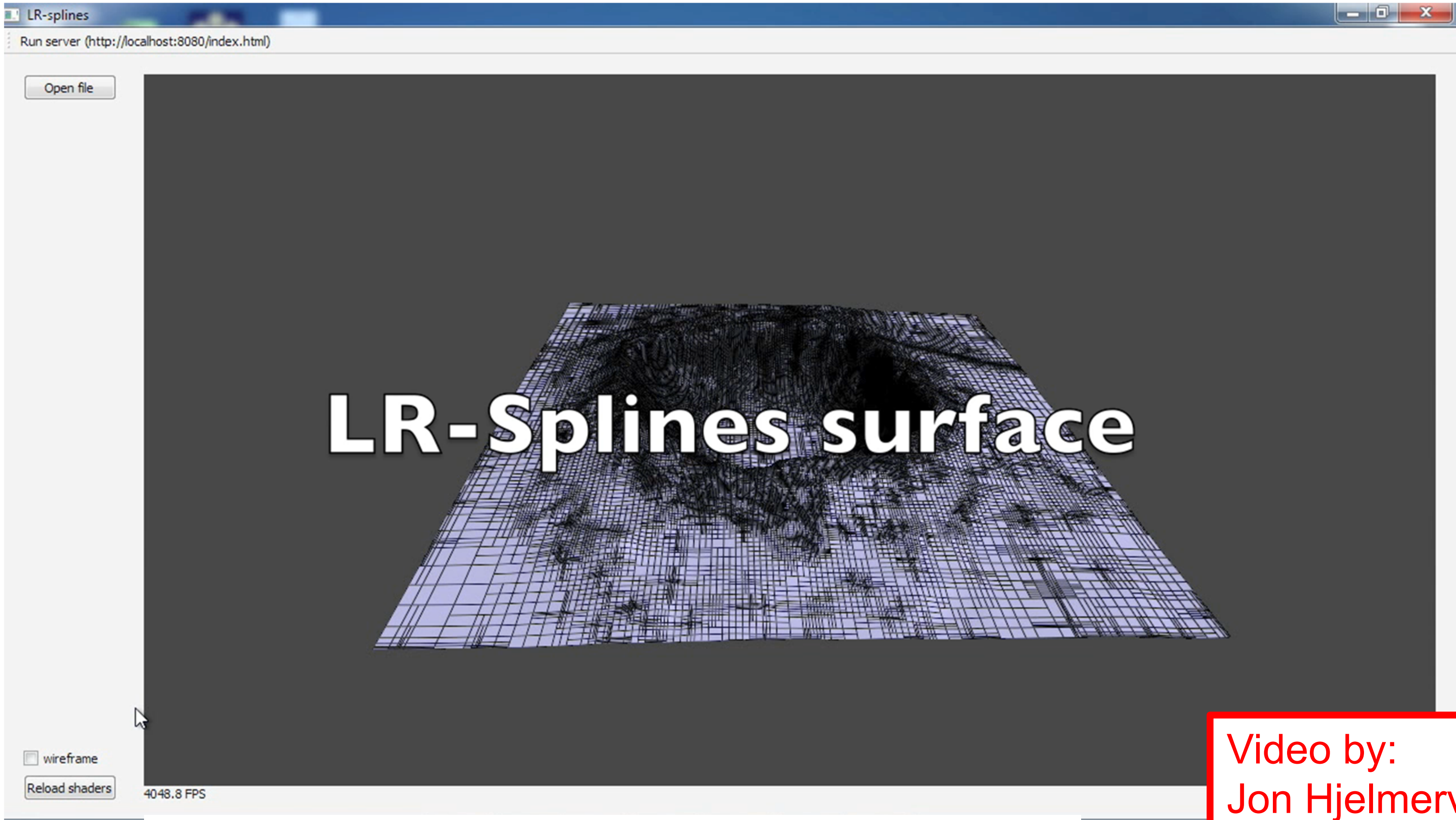




Dotted lines added to visualize partition into elements (polynomial pieces) and suggest possible locations for refinement specification.

- The vertex mesh augmented with all knotlines (extended T-mesh)

Direct Visualization of LR B-splines on the GPU, view dependent tessellation



Data courtesy of http://www.opentopography.org/

Video by:
Jon Hjelmervik,
SINTEF ICT

Peeling for Ensuring Linear Independence
(Valid for LR B-splines and Semi-Standard T-splines)

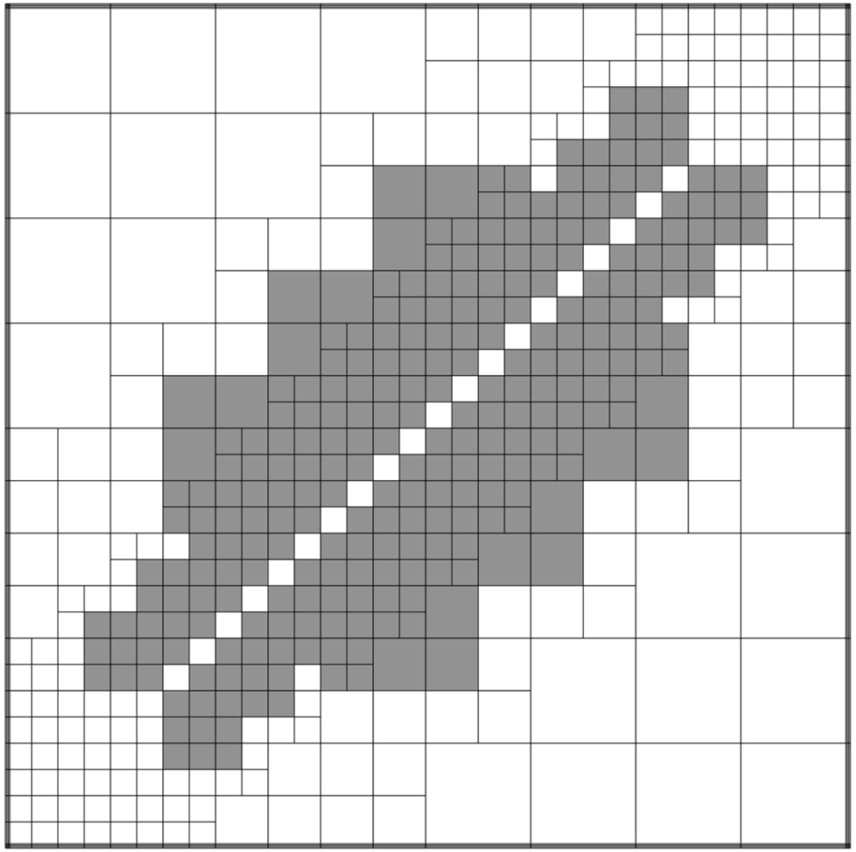
The refinement starts from a tensor product B-spline space with $(p_1 + 1)(p_2 + 1) \dots (p_d + 1)$ B-splines covering an element spanning the polynomial space of degree $(p_1, p_2, \dots, p_d)$ over the element.

- A refinement cannot reduce the polynomial space spanned over an element.
- An extra B-spline in a linear dependency relation can be removed without changing spanning properties over its elements
  - Before the removal of a B-spline there must consequently be more than $(p_1 + 1)(p_2 + 1) \dots (p_d + 1)$ B-splines covering all elements of the removed B-spline.

Overloaded elements and B-splines

- We call an element overloaded if there are more than $(p_1 + 1)(p_2 + 1) \dots (p_d + 1)$ B-splines covering the element.
- We call a B-spline overloaded if all its elements are overloaded.

Illustration by:
Kjetil A. Johannessen,
SINTEF



The support of overloaded B-splines colored grey.

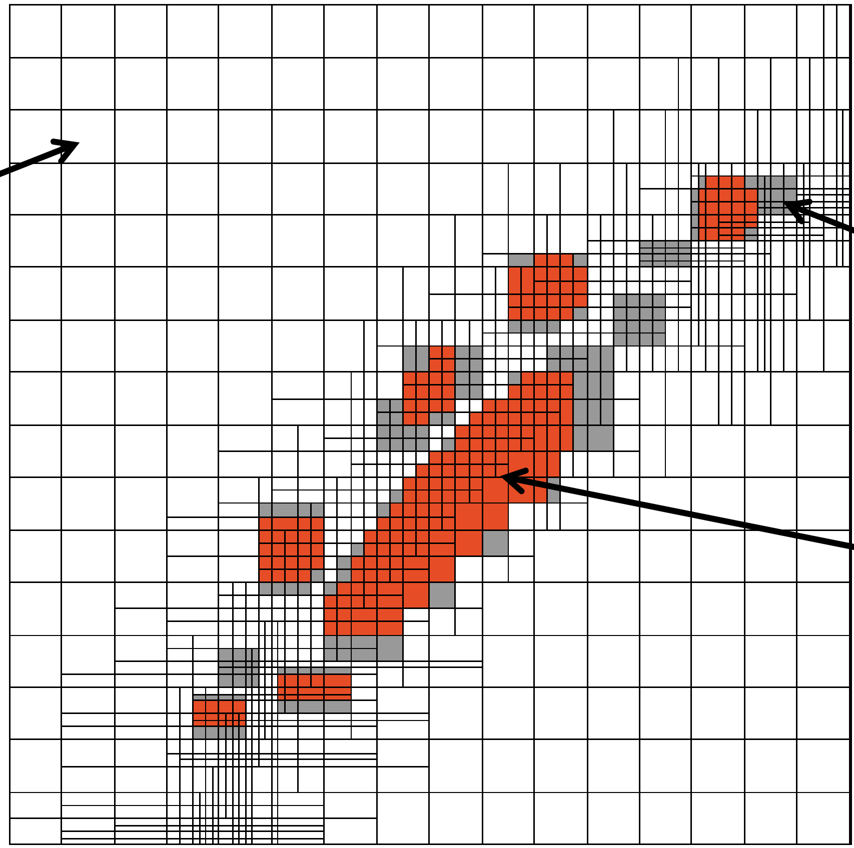
Observations

- If there is no overloaded B-spline then the B-splines are locally (and globally) linearly independent
  - All overloaded elements not part of an overloaded B-spline can be disregarded
- Only overloaded B-splines can occur in linear dependency relations
- A linear dependency relation has to include at least two* overloaded B-splines.
  - Elements with only one overloaded B-spline cannot be part of linear dependency relation. Thus overloaded B-spline having such an element cannot be part of linear dependency relation.

* The number is actually higher, at least: $2^l + 1$ in the l-variate case.

Example peeling algorithm for overloaded B-splines.

White: Elements covered by no overloaded B-spline



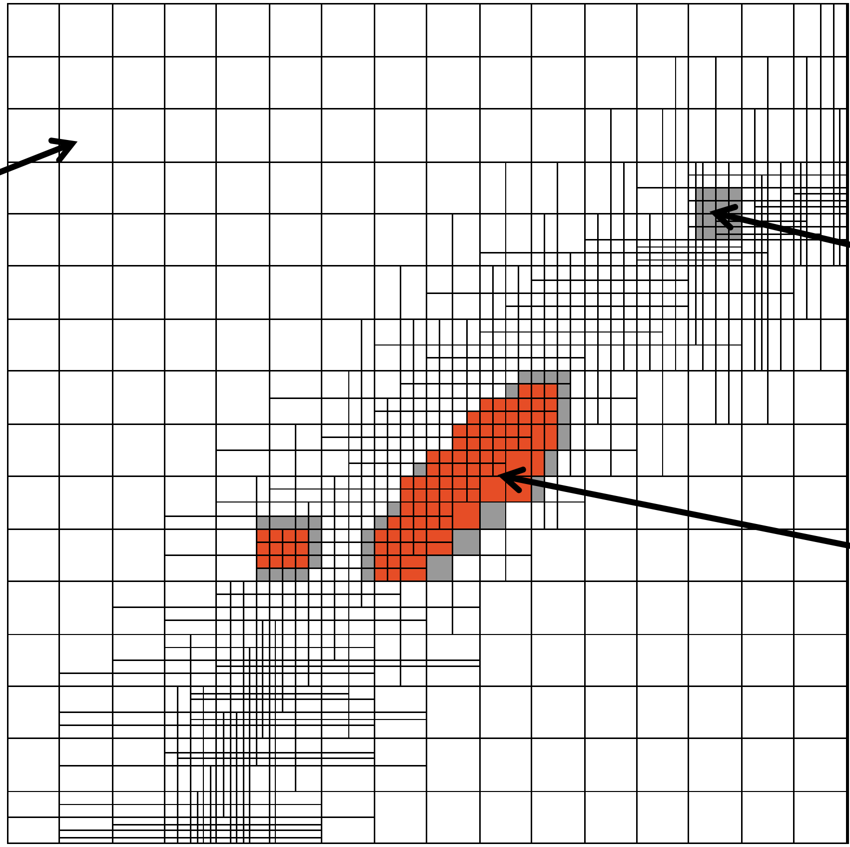
Grey: Elements covered by only one overloaded B-spline (can be removed)

Red: Elements covered by more than one overloaded B-spline

Illustration by: Kjetil A. Johannessen, NTNU

Example, Continued.

White: Elements covered by no overloaded B-spline

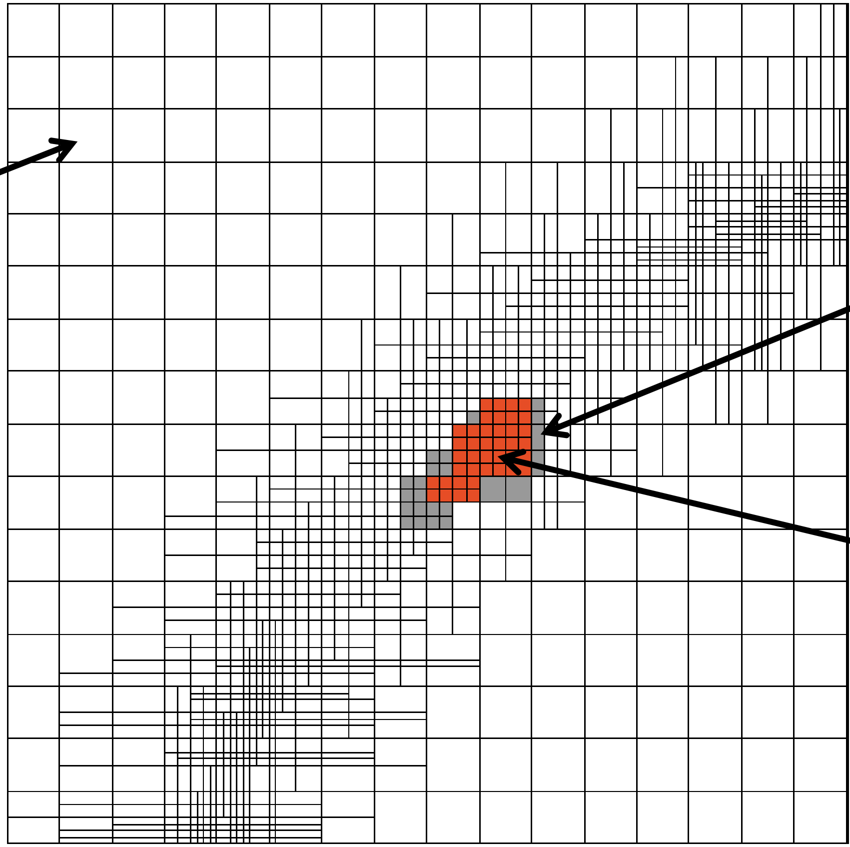


Grey: Elements covered by only one overloaded B-spline (can be removed)

Red: Elements covered by more than one overloaded B-spline

Example, Continued.

White: Elements covered by no overloaded B-spline



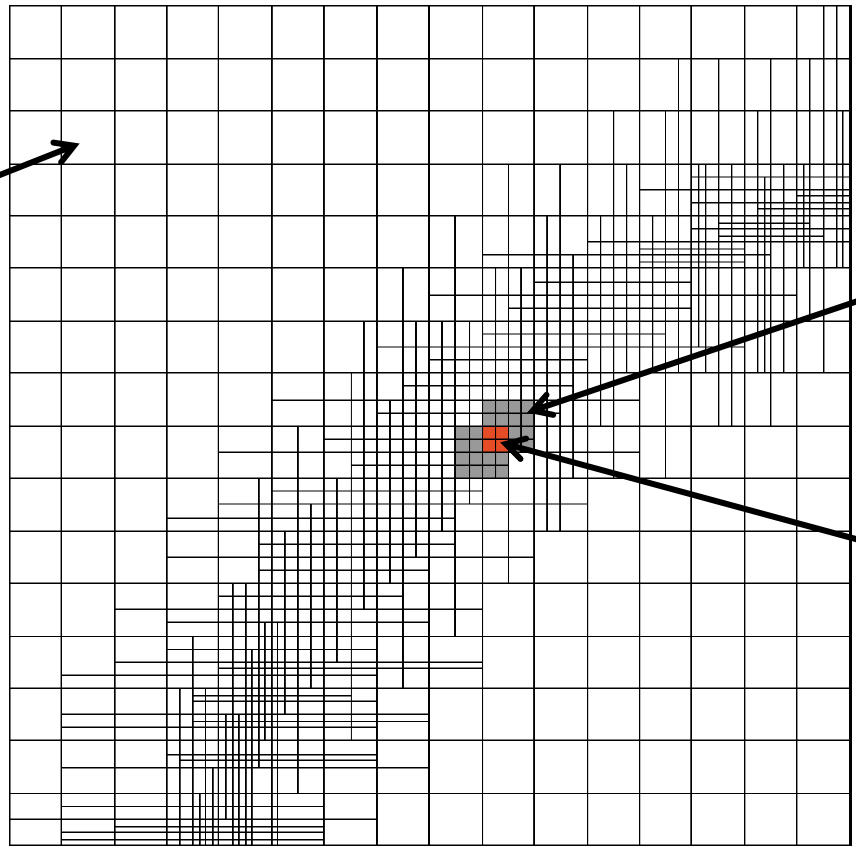
Grey: Elements covered by only one overloaded B-spline (can be removed)

Red: Elements covered by more than one overloaded B-spline

Illustration by: Kjetil A. Johannessen, NTNU

All B-splines remaining some elements that are overloaded only once. Linear dependency not possible.

White: Elements covered by no overloaded B-spline



Grey: Elements covered by only one overloaded B-spline (can be removed)

Red: Elements covered by more than one overloaded B-spline

Illustration by: Kjetil A. Johannessen, NTNU

No overloaded B-splines remaining. Linear dependency not possible.

White:
Elements covered by no overloaded B-spline

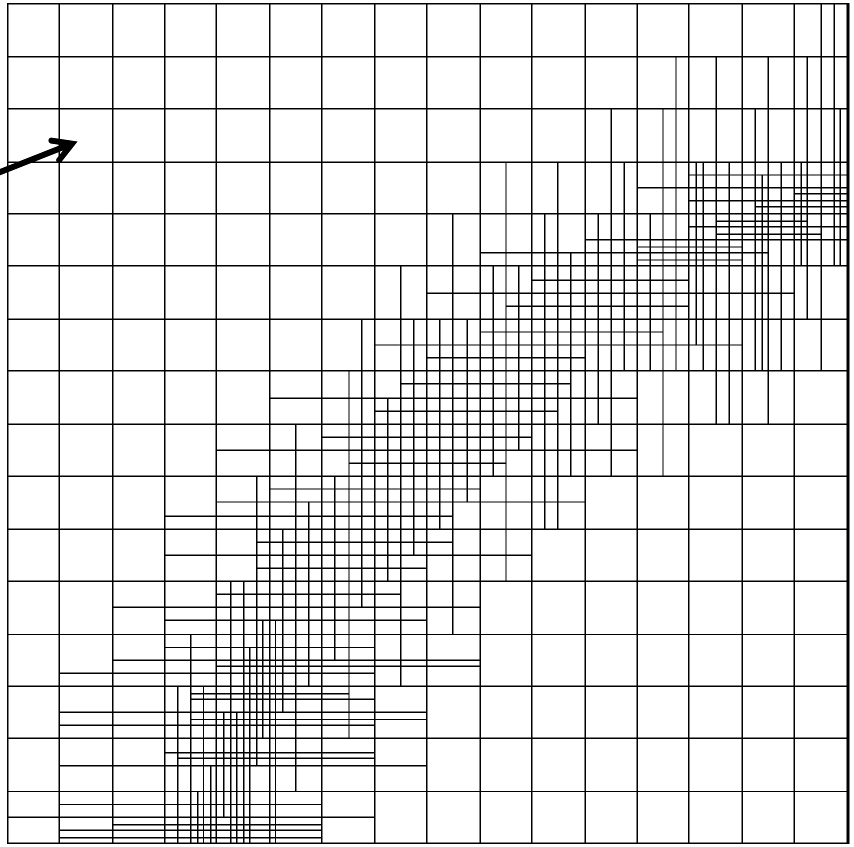


Illustration by:
Kjetil A. Johannessen, NTNU

Current work on LR B-splines at SINTEF

- LR Splines extensions to the SINTEF GoTools C++ library is under way. EU-project: TERRIFIC.
  - www.terrific-project.eu
- We work an efficient computation of stiffness matrices for LR Spline represented IGA on multi-core and many core CPUs
- We work on IgA based on LR B-splines
- We work on efficient LR B-spline visualization on GPUs
- We address representation of geographic information using LR B-splines in EU-project Iqmulus
  - www.iqmulus.eu

Simulation – Future Information flow