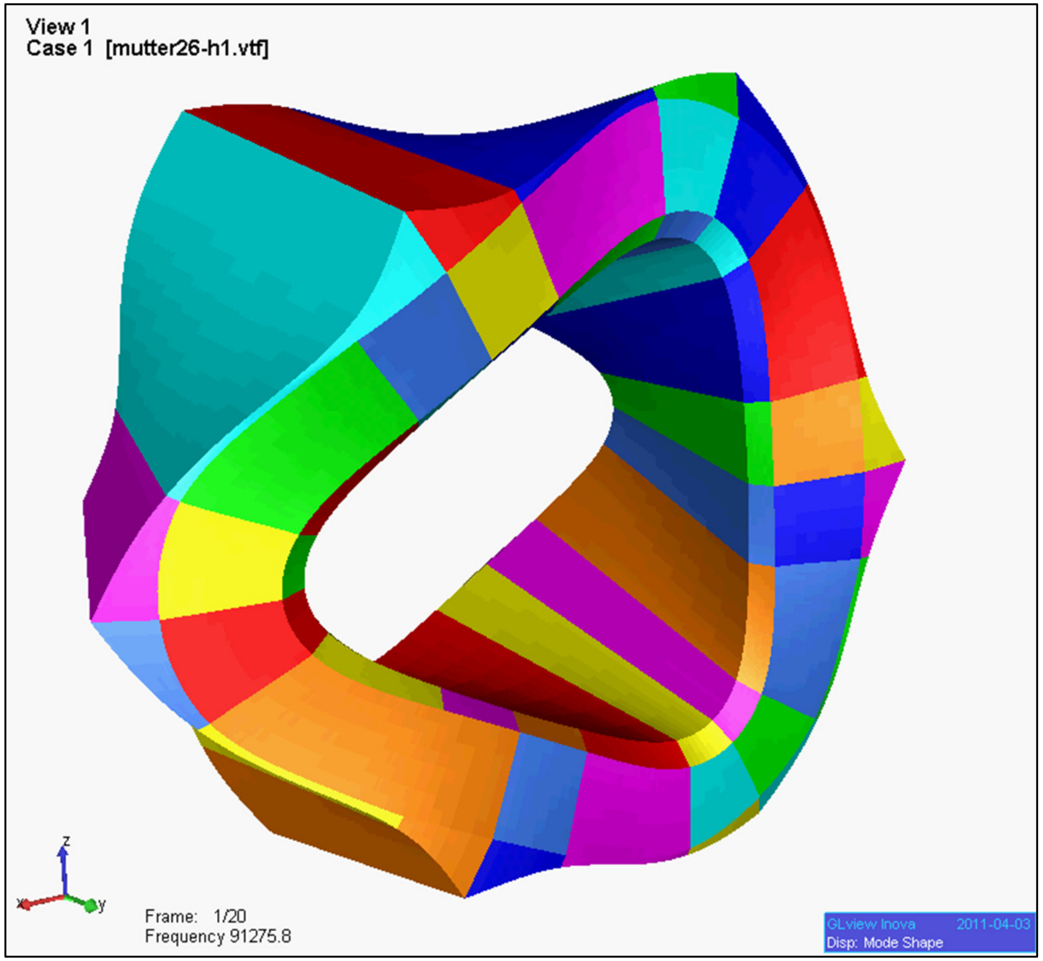
Locally Refined B-splines for isogeometric representation and Isogeometric Analysis

Tor Dokken
SINTEF, Oslo, Norway
CMA, University of Oslo

Free vibration of a Nut – 3-variate NURBS elements



Example by: Knut Morten Okstad, SINTEF IKT

Introductory remarks

The idea of LR B-splines is inspired by T-splines

- The work on LR B-splines started in June 2009 when I wanted to understand T-splines better. However, I did this without looking at the details of the T-spline approach, contemplating the concept "Local Refinement of B-spline represented functions".
- Comparing my thoughts on "Local Refinement of B-spline represented functions" with T-splines, we realized we had an alternative approach. The LR B-spline refinement is specified in the parameter domain, while for T-splines the B-splines are inferred from T-splines vertex grid,
- The LR B-splines can be given a T-spline type interface for specifying refinement. Having such an interface the data provided will be used directly to refine in the parameter domain, rather than reason in the T-spline type vertex grid.

Introductory remarks - continued

- We prefer to refine directly in the mesh in the parameter domain, as we then deal with only one mesh that reflects the piecewise polynomial structure, rather than the multiple meshes of T-splines.
- It was also obvious that the idea of local refinement posed many new questions related to what we now denote B-splines over Box-partitions. So rather than first proposing new algorithms we: Prof Tom Lyche University of Oslo, Tor Dokken and Kjell Fredrik Pettersen from SINTEF wanted to build a theoretical foundation.
- The theoretical work took much longer time than originally anticipated. The preprint (March 2012) of the paper can be downloaded from:
  - http://www.sintef.no/upload/IKT/9011/geometri/LR-splines%20SINTEF%20Preprint%20-%20signatures.pdf

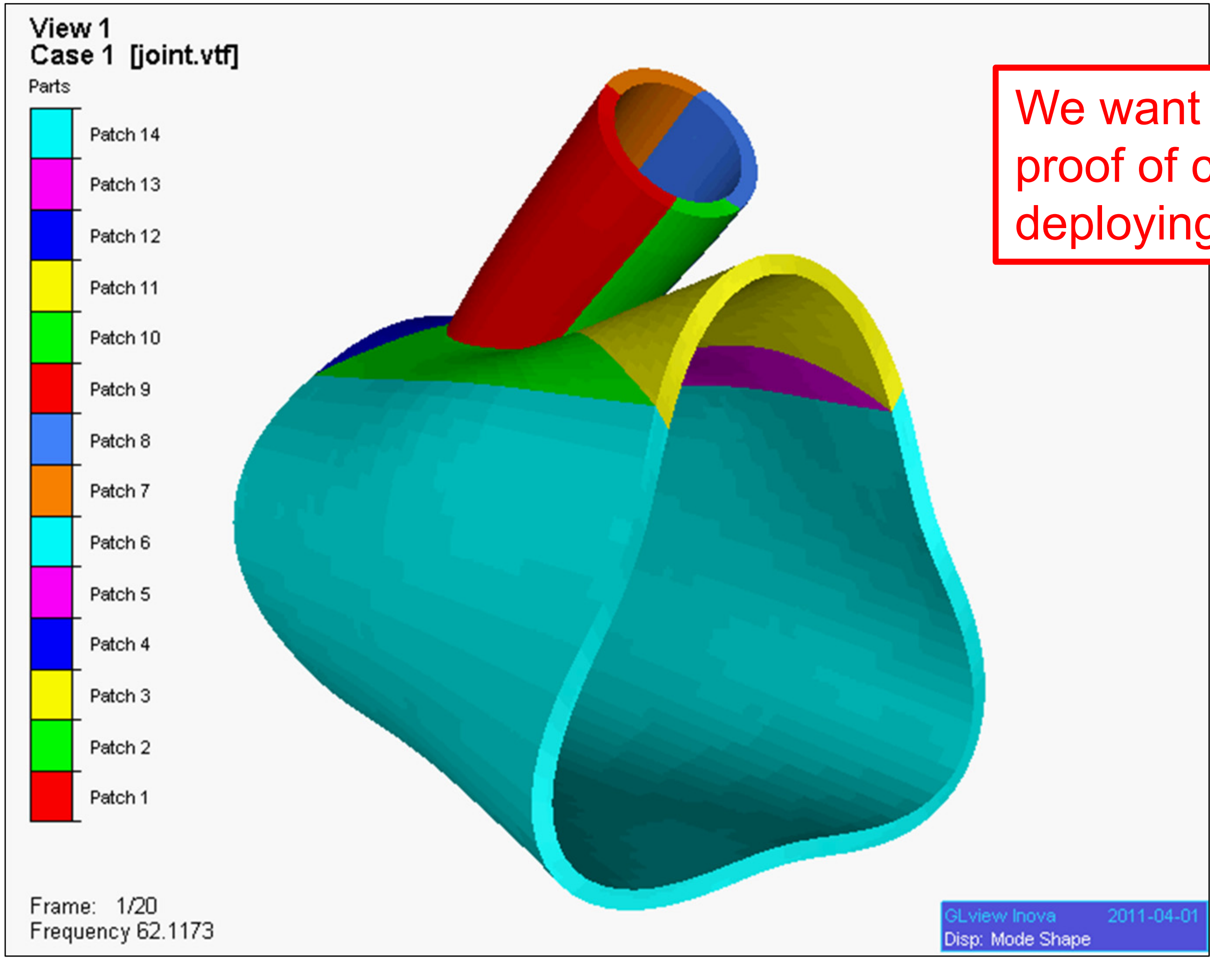
Structure of the 4 lectures

- Hour 1 givens and introduction and sets the scene.
- Hour 2 will be focused on spline spaces over box-partitions and their relevance for FEA. The lectures are thus relaxant for both:
  - T-splines
  - PHT-splines
  - LR B-splines
  - Hierarchical splines
- Hour 3 and 4 will address LR B-splines

Introduction - The larger picture

Some of you will have seen some/many of these slides before, but I feel it important to paint a larger picture.

Free vibration of a Tubular Joint – 3-variate NURBS elements



We want to move from proof of concept of IGA to deploying IGA in industry.

Example by: Knut Morten Okstad, SINTEF ICT

Independent evolution of Computer Aided Design (CAD) and Finite Element Analysis (FEA)

- CAD and FEA evolved in different communities before electronic data exchange
  - FEA developed to improve analysis in engineering
  - CAD developed to improve the design process
  - Information exchange was drawing based, consequently the mathematical representation used posed no problems
    - Manual modelling of the element grid
  - Implementations used approaches that best exploited the limited computational resources and memory available.
- FEA was developed before the NURBS theory
  - FEA evolution started in the 1940s and was given a rigorous mathematical foundation around 1970 (1973: Strang and Fix's *An Analysis of The Finite Element Method)*
  - B-splines: 1972: DeBoor-Cox Algorithm, 1980: Oslo Algorithm

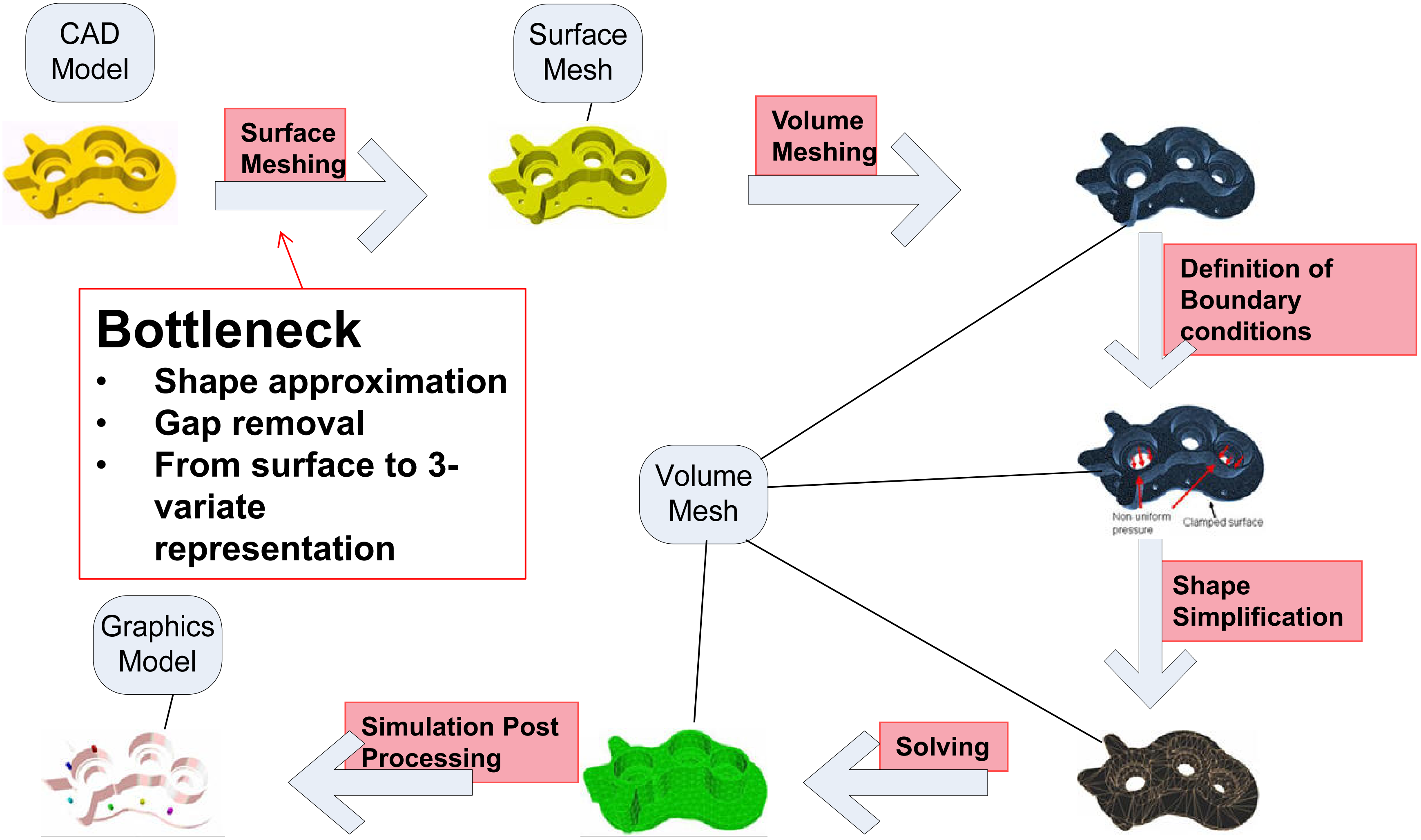
Why have NURBS not been used in FEA?

- Shape representation in CAD and FEA
  - Finite Elements are 3-variate polynomials most often of degree 1 or 2 and cannot represent many shapes accurately.
  - NURBS (NonUniform Rational B-Spline) surfaces used in CAD can represent elementary curves and surfaces exactly. (circle, ellipse, cylinder, cone…) in addition to free form shapes.
  - Mathematical representation in CAD and FEA chosen based on what was computationally feasible in the early days of the technology development.
- We needed someone with high standing in FEA to promote the idea of splines in analysis
  - Prof. Tom Hughes, University of Texas at Austin, did this in 2005
  - This has triggered a new drive in spline research after a quiet period.

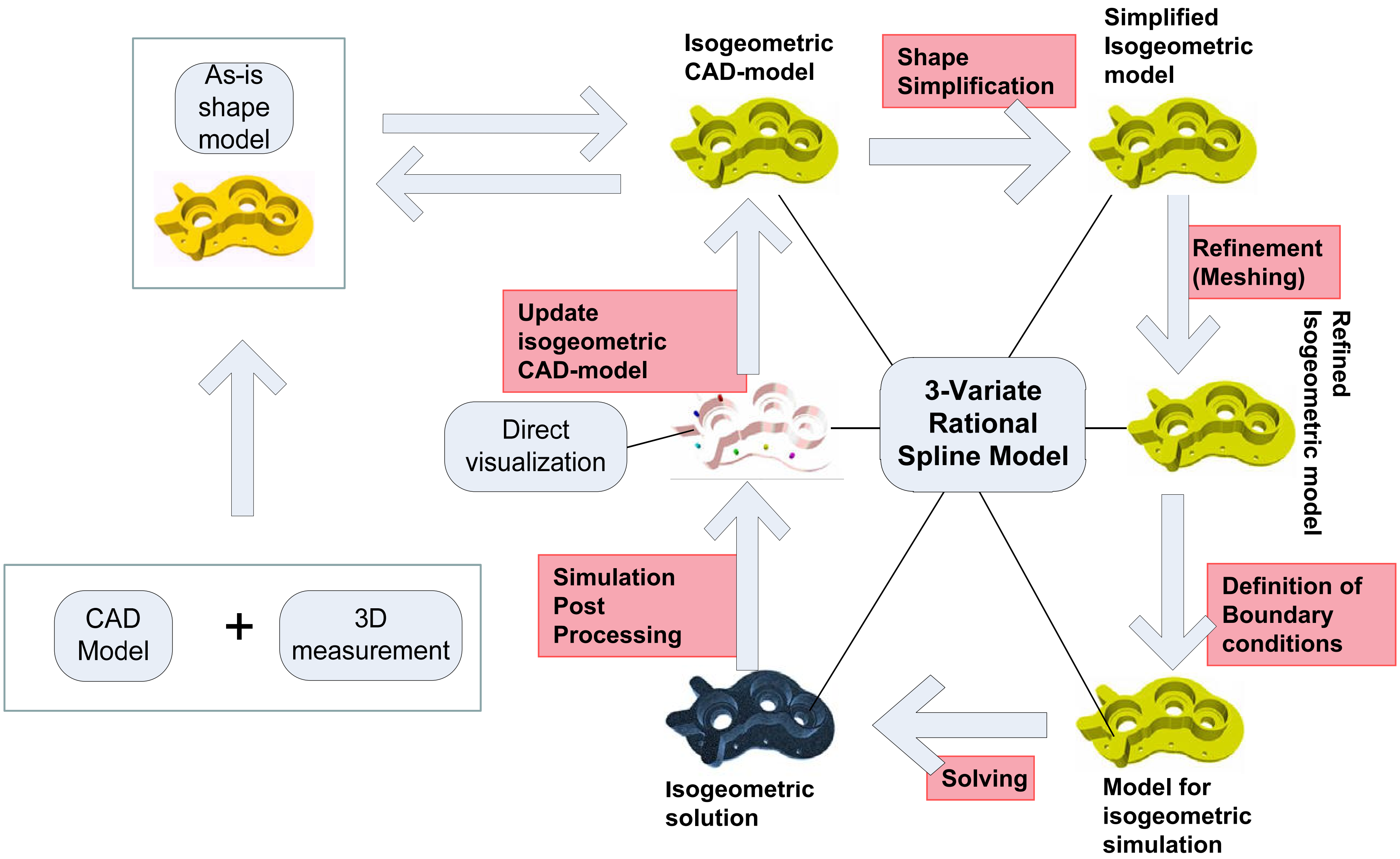
Why are splines important to isogeometric analysis?

- Splines are polynomial, same as Finite Elements
- B-Splines are very stable numerically
- B-splines represent regular piecewise polynomial structure in a more compact way than Finite Elements
- NonUniform rational B-splines can represent elementary curves and surfaces exactly. (circle, ellipse, cylinder, cone…)
- Efficient and stable methods exist for refining the piecewise polynomials represented by splines
  - Knot insertion (Boehms Algorithm, and Oslo Algorithm, 1980)
  - B-spline has a rich set of refinement methods

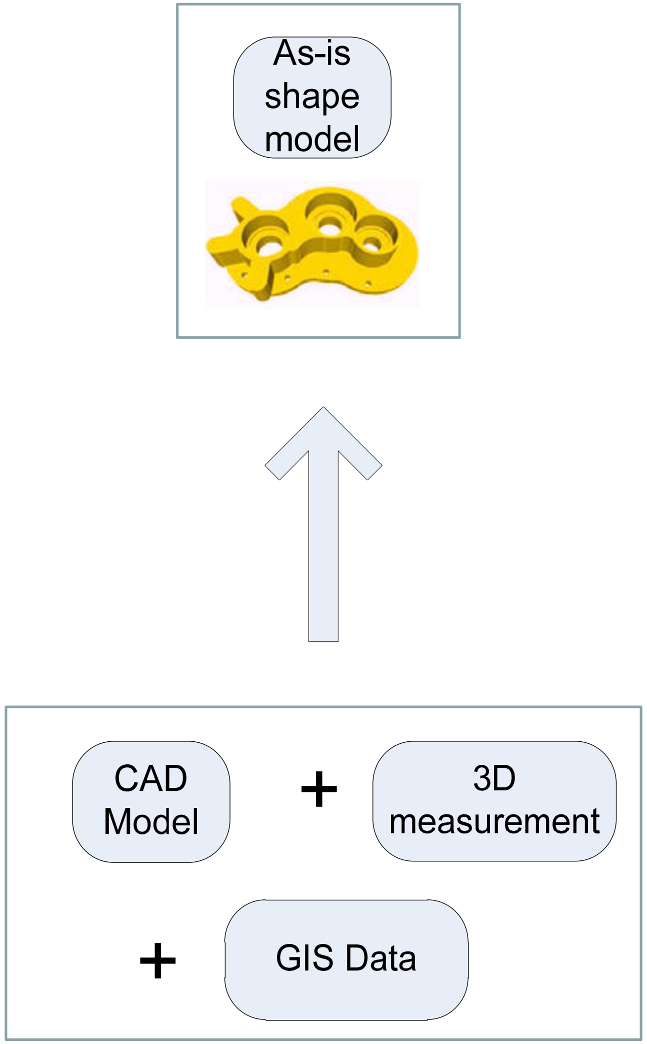
Traditional simulation pipeline



Simulation – Future Information flow

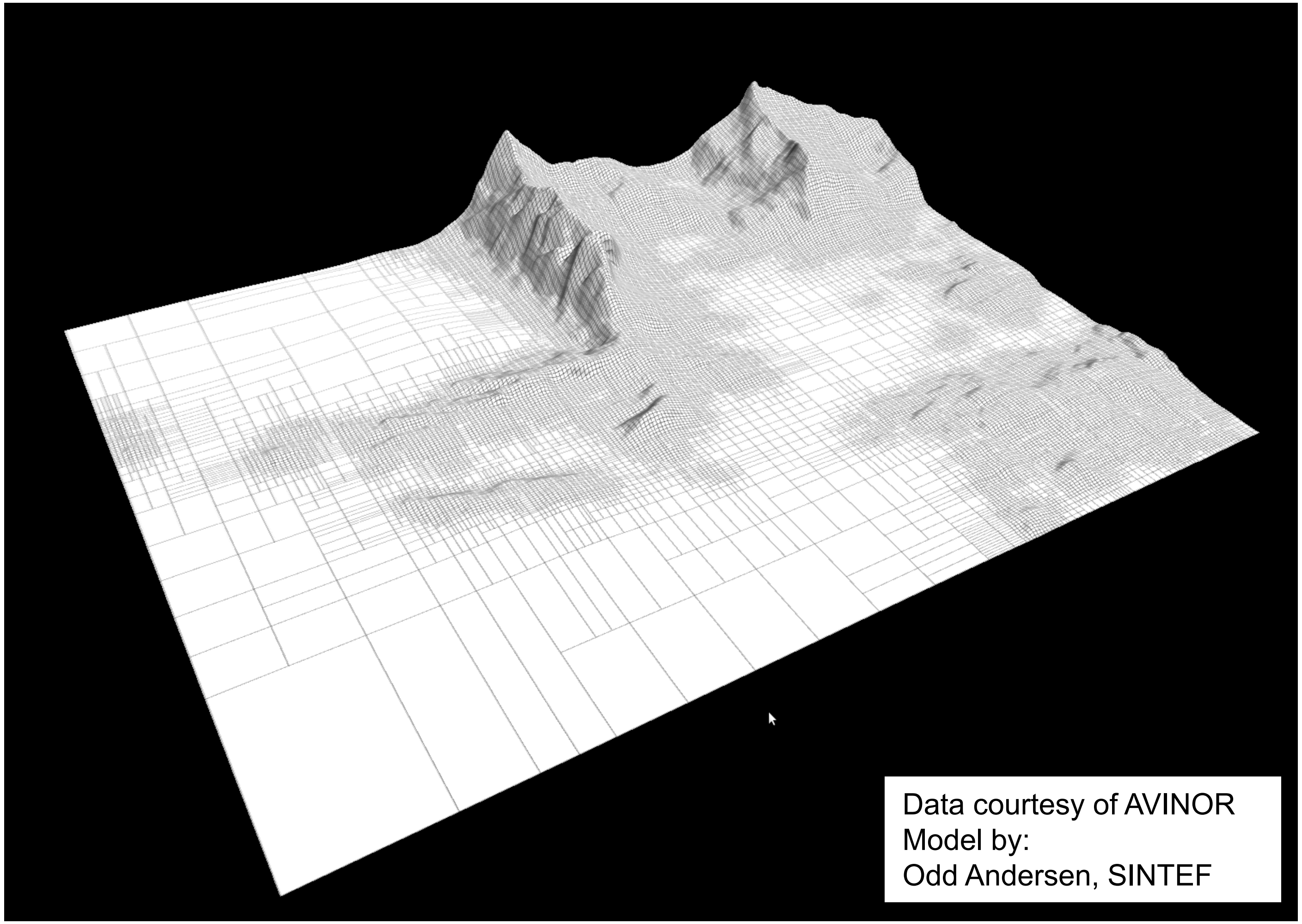


Challenge 1: Create “as-is” model



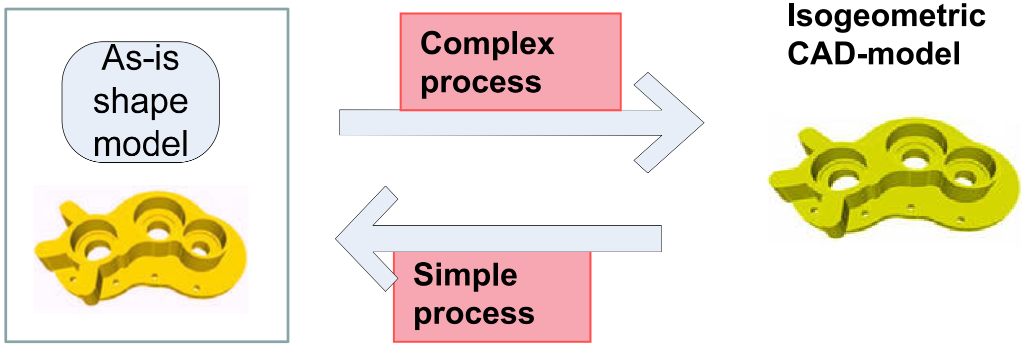
- CAD-models describes the object as planned
  - Combines elementary surfaces (plane, cylinder, cone, sphere, torus and NURBS)
- Models aimed at visual purpose most often represent shape by (texture mapped) triangulations
- Laser scanning efficiently produce millions of points on the geometry
  - Extracting information from 3D datasets is complex
  - A industry is established related to model building from laser scans
  - Using the datasets for validation and updating of 3D models (CAD) is challenging
- The projects following projects address these issue partly using LR B-splines:
  - “3D Airports for Remotely Operated Towers” in SESAR JU (EU)
  - The STREP TERRIFIC – EU ICT-program Factory of the Future
  - New IP- IQmulus on large GIS data sets (currently under contract negotiations) (October 2012-September 2016)

LR B-spline representation of area around Svolvær airport, Norway



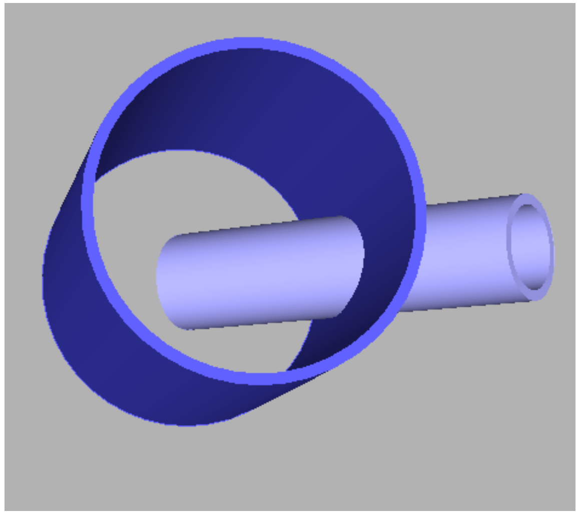
Data courtesy of AVINOR
Model by:
Odd Andersen, SINTEF

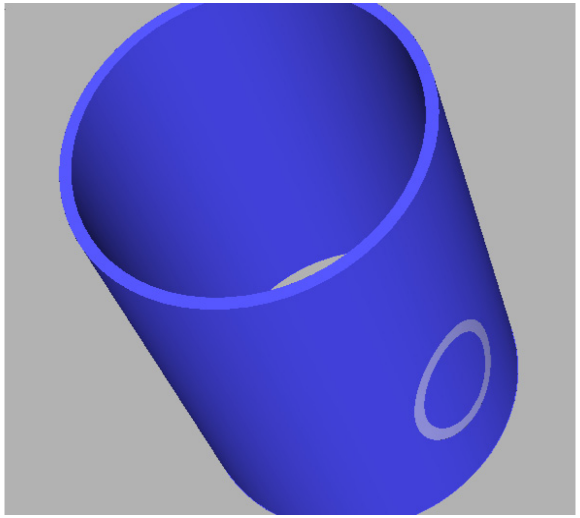
Challenge 2: Create 3-variate isogeometric model

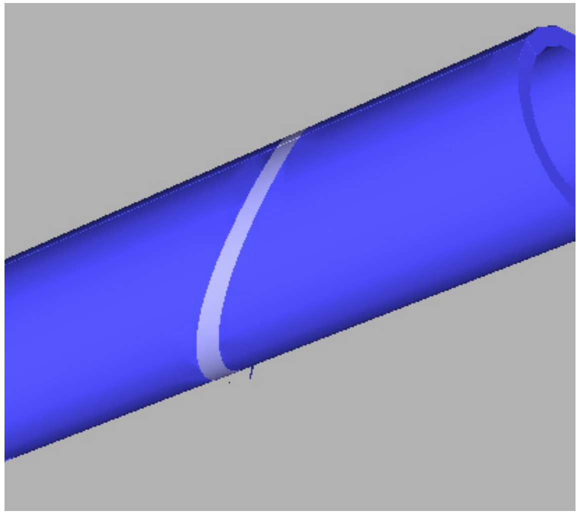


- The “As-is” shape model describes mathematically only the inner and outer hulls (surfaces) of the object using triangulations, elementary surfaces or NURBS surfaces.
- The isogeometric model is analysis/simulation suitable and describes the volumes mathematically by watertight structures of blocks of 3-variate rational splines
- Building an isogeometric model is a challenge:
  - There is a mismatch between the surface patch structure of the “As-is” model, and a suited block structure of an Isogeometric 3-variate rational spline model.
  - Augmented spline technology is needed such as the novel Locally Refined Splines.
- Projects addressing this
  - Isogeometry (2008-2012)- KMB project funded by the Norwegian Research council
  - TERRIFIC (2011-2014) – STREP funded by the EU ICT

Example: Isogeometric tube joint - Intersection







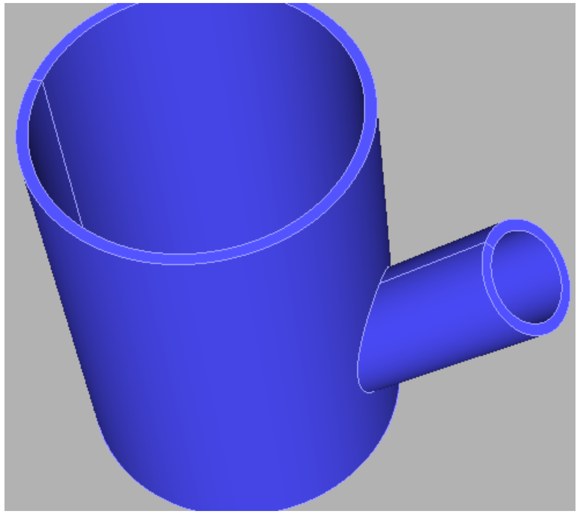
- Two independent pipes coming from CAD and described as 3-variate volumes

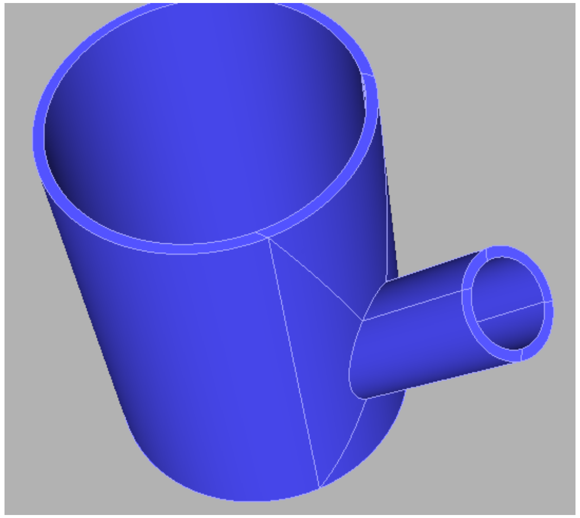
- The intersection of the pipes calculated.
- The original large pipe is split in 3 volumes

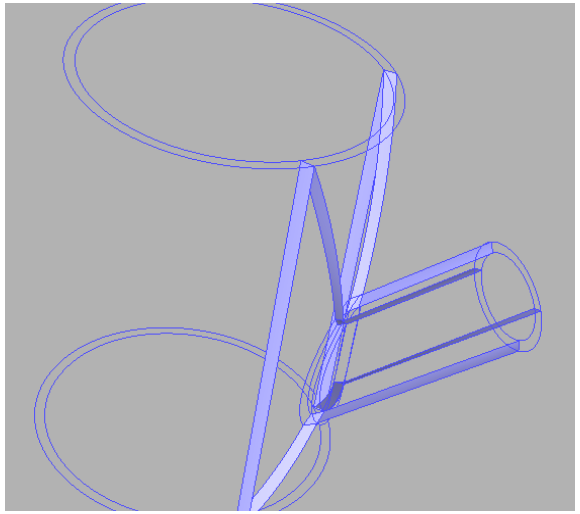
- The intersection of the pipes calculated.
- The original small pipe is split in 3 volumes

Example by:
Vibeke Skytt,
SINTEF ICT

Example: Isogeometric tube joint – Composing volumes







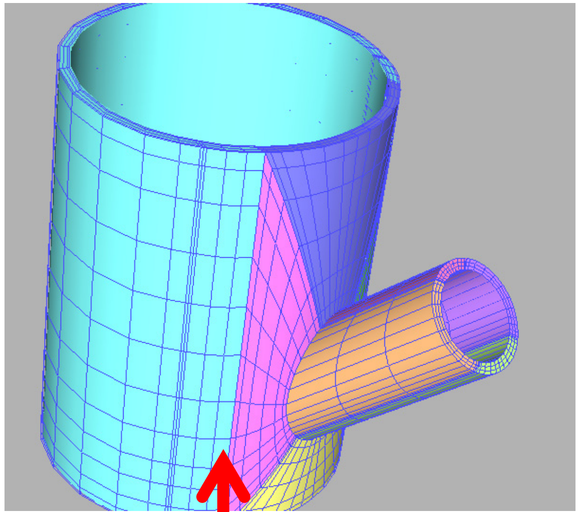
- The relations between the sub volumes produced by the intersection are established
- These volumes do not satisfy the hexahedral (box structure) of the need isogeometric sub volumes

- The volumes split to produce hexahedral volumes

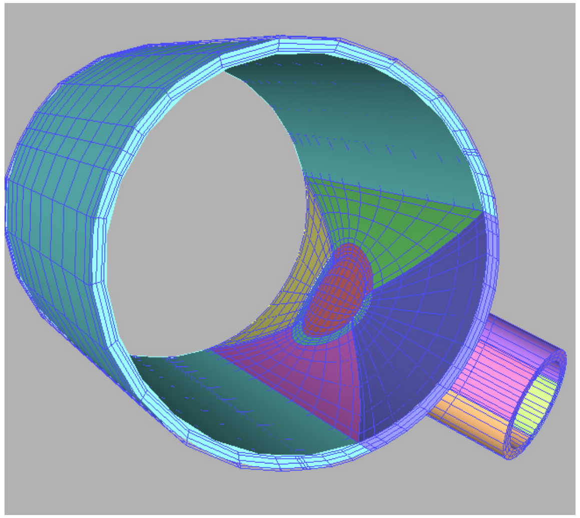
- The internal faces produced by the splitting process

Example by:
Vibeke Skytt,
SINTEF ICT

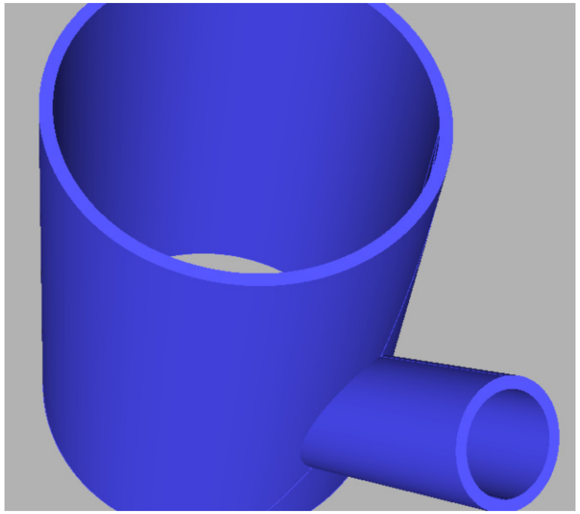
Example: Isogeometric tube joint – match spline spaces(B-splines)



- Spline space refined to have matching lines in each hexahedral NURBS-block to produce a watertight representation



- Same as to the left, different view



- The final isogeometric tube joint.

Matching spline spaces of adjacent tensor product B-splines volumes drastically increases the number of vertices. (Local refinement needed)

Example by:
Vibeke Skytt,
SINTEF ICT

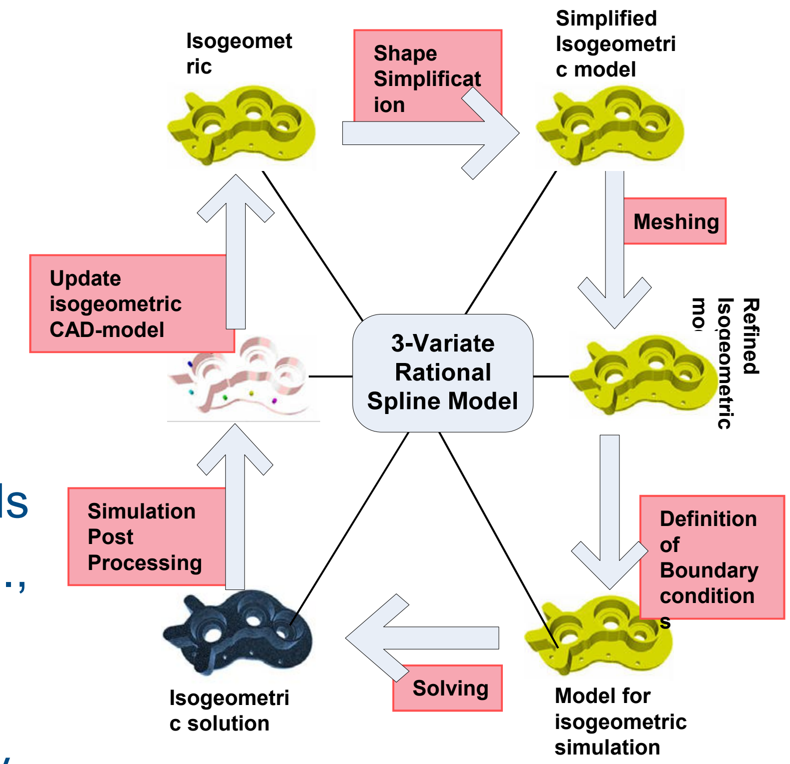
Challenge 3: Isogeometric analysis

First introduced in 2005 by T.J.R. Hughes

- Replace traditional Finite Elements by NURBS - NonUniform Rational B-splines
- Accurate representation of shape
- Allows higher order methods
- Perform much better that traditional Finite Elements on benchmarks
- Refinement of analysis models without remeshing
- Exact coupling of stationary and rotating grids
- Augmented spline technology is needed, e.g., Locally Refined Splines

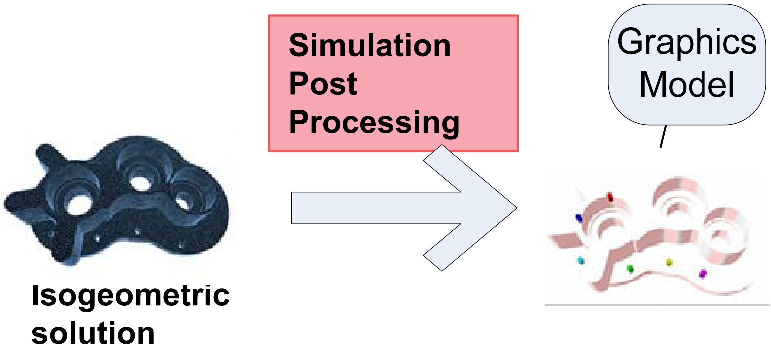
Projects:

- ICADA (2009-2014)– KMB Project funded by Norwegian Research Council and Statoil
- Exciting (2008-2012)
- TERRIFIC (2011-2014) – STREP EU ICT-program

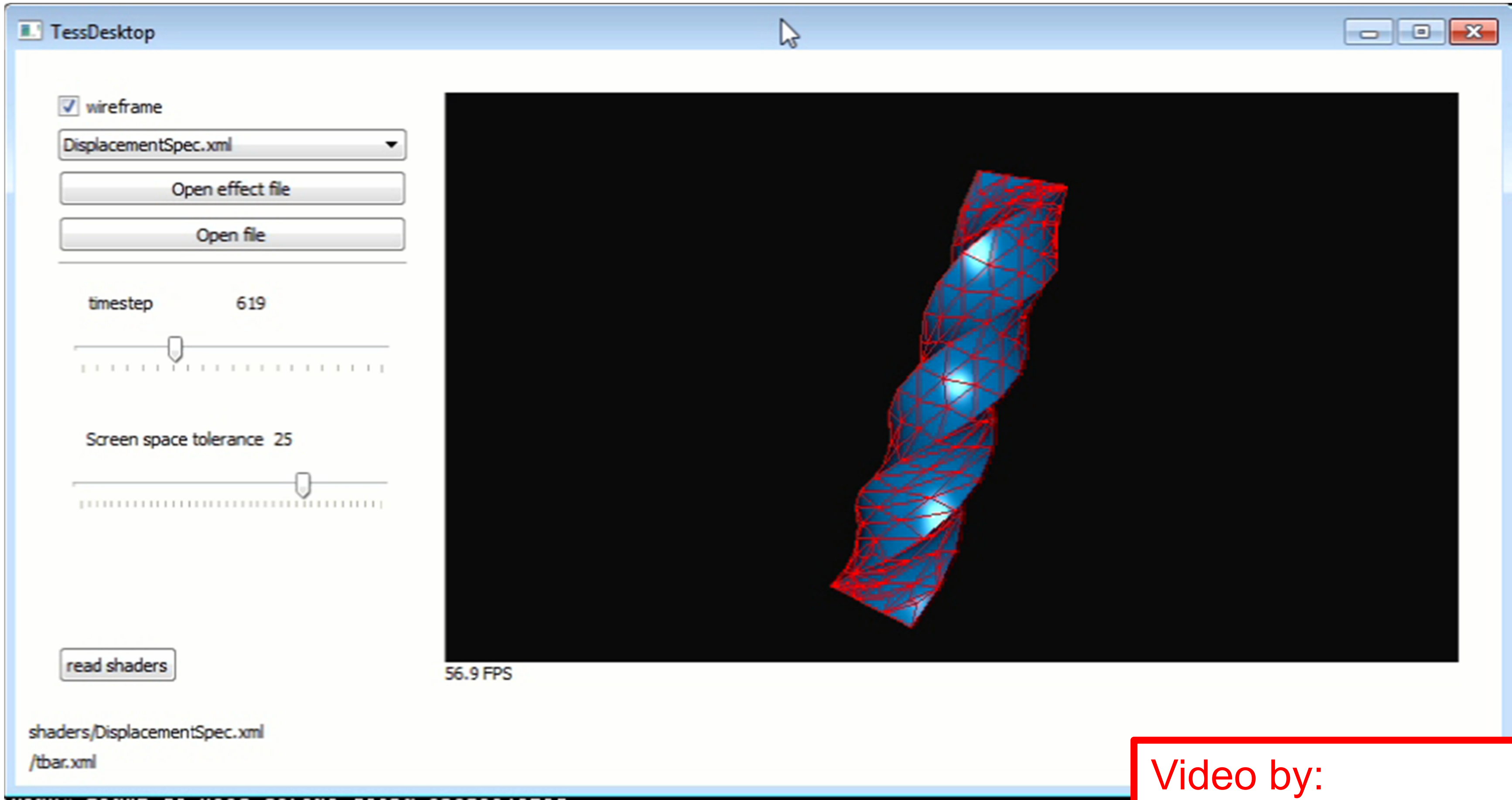


Challenge 4: Isogeometric visualization

- Traditional visualization technology is triangle based (tesselation)
  - The isogeometric model has to be approximated with triangles for visualization
  - Results are degraded and information lost
- Need for visualization solutions exploiting the higher order representations
  - Higher order representations are more advanced and can better represent singularities in the solution
  - Create view dependent tessellation of splines on the GPU
  - Direct ray tracing on the GPU:
- Project: Cloudviz (2011-2014)



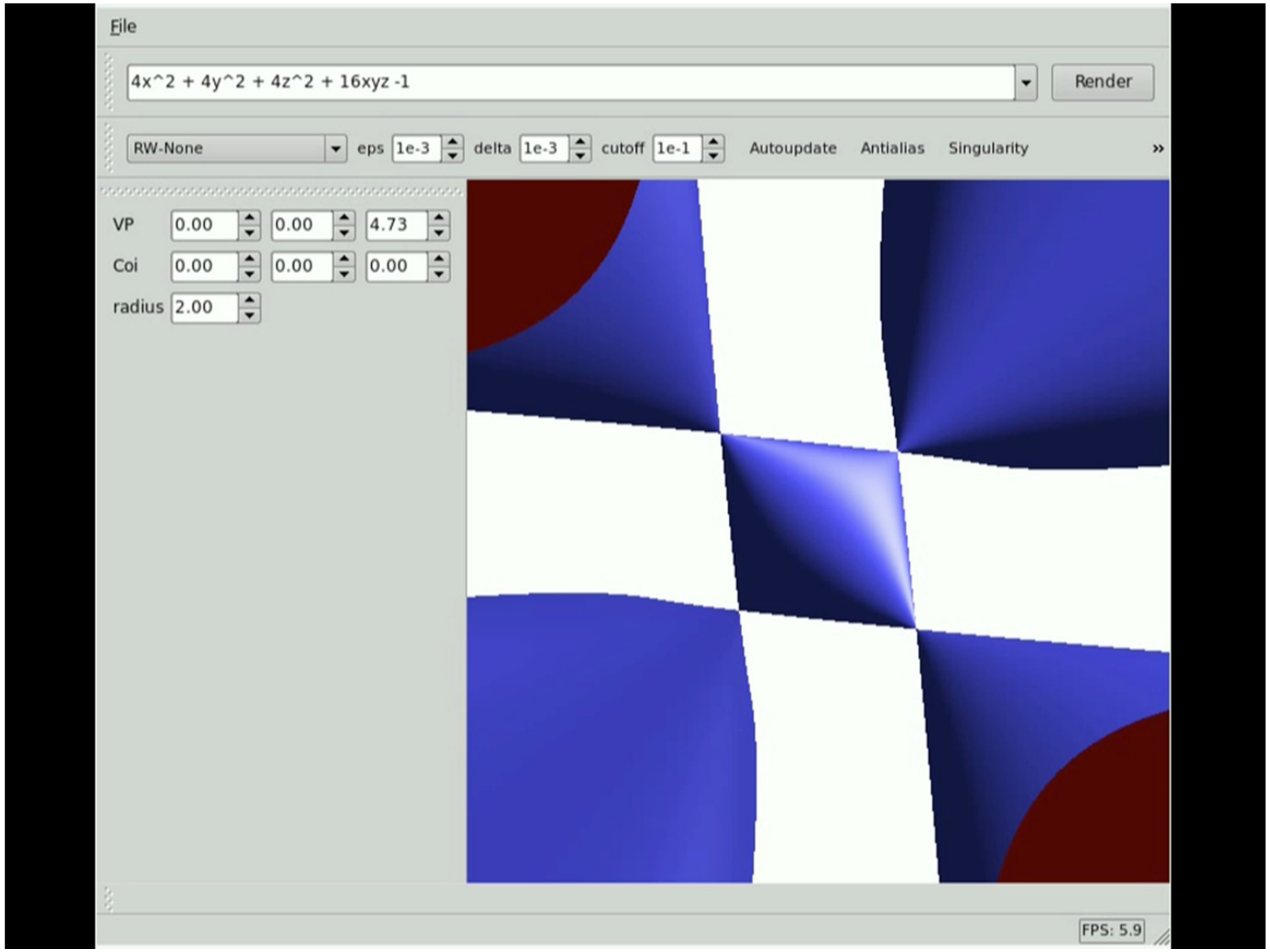
Isogeometric view dependent tesselation of the spline model



Video by:
Jon Hjelmervik,
SINTEF ICT

Click her for video:
http://www.youtube.com/watch?v=KOsDBx8yEt0&list=UU_GWvrs307jzpjIWvQxWwHA&index=1&feature=plcp

Direct ray-casting on the GPU



Video by:
Johan S. Seland,
SINTEF ICT

Click her for video: http://www.youtube.com/watch?v=ORFhU3diakA

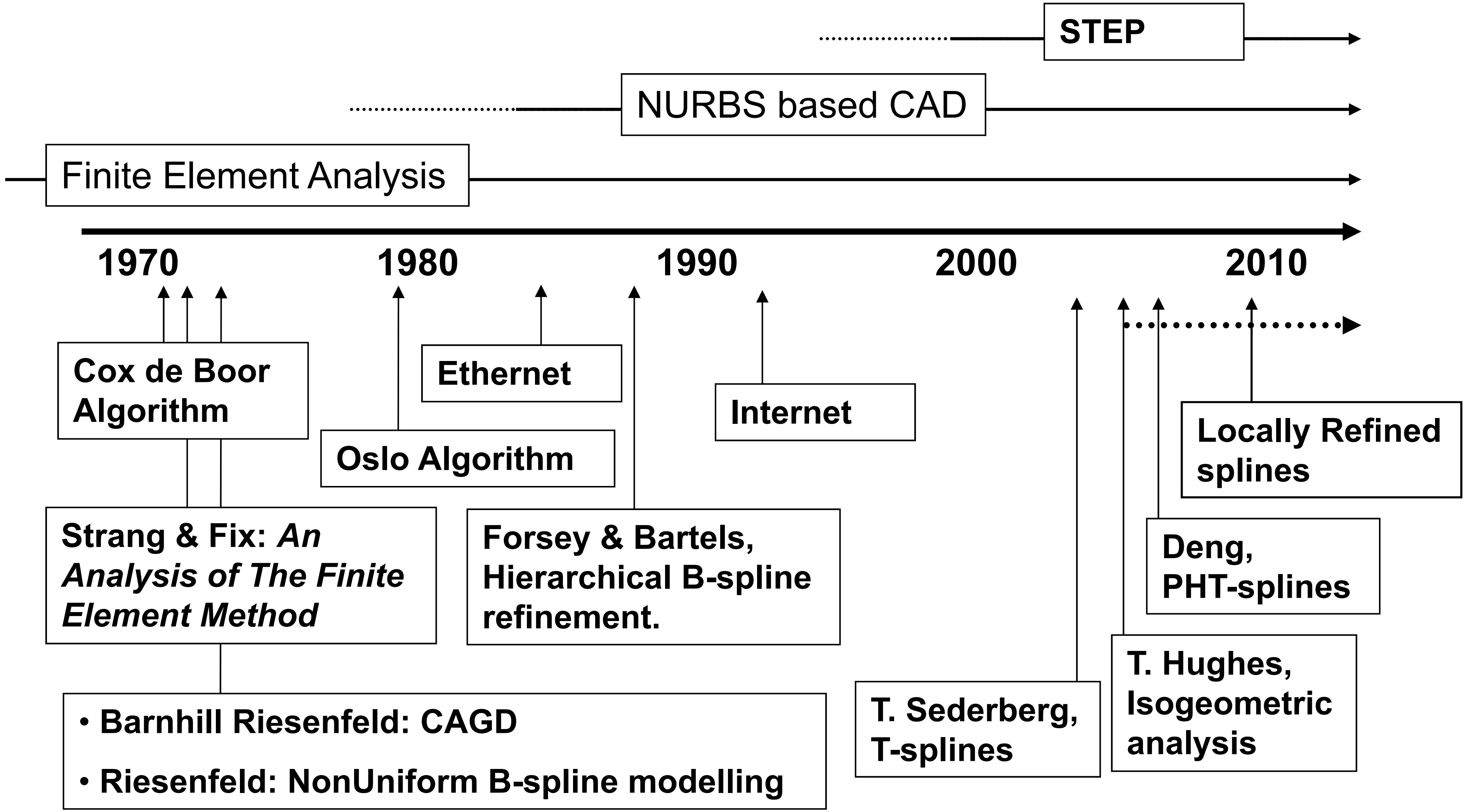
From stand alone computers and systems to integrated information flows

- As long as communication between computers was hard, information exchange remained paper based
  - The Ethernet invented by Xerox Parc in 1973-1975,
  - ISO/IEEE 802/3 standard in 1984
  - Deployment in industry started, simple communication between computers
- CAD Data Exchange introduced
  - IGES - *Initial Graphics Exchange Specification*

    Version 1.0 in 1980
  - STEP - *Standard for the Exchange of Product model data*

    started in 1984 as a successor of IGES, SET and VDA-FS, Initial Release in 1994/1995, deployment in industry started
- The Internet opened to all 1991
  - Start of deployment of data exchange between processes over the Internet

Timeline important events related to FEA and Isogeometric Analysis

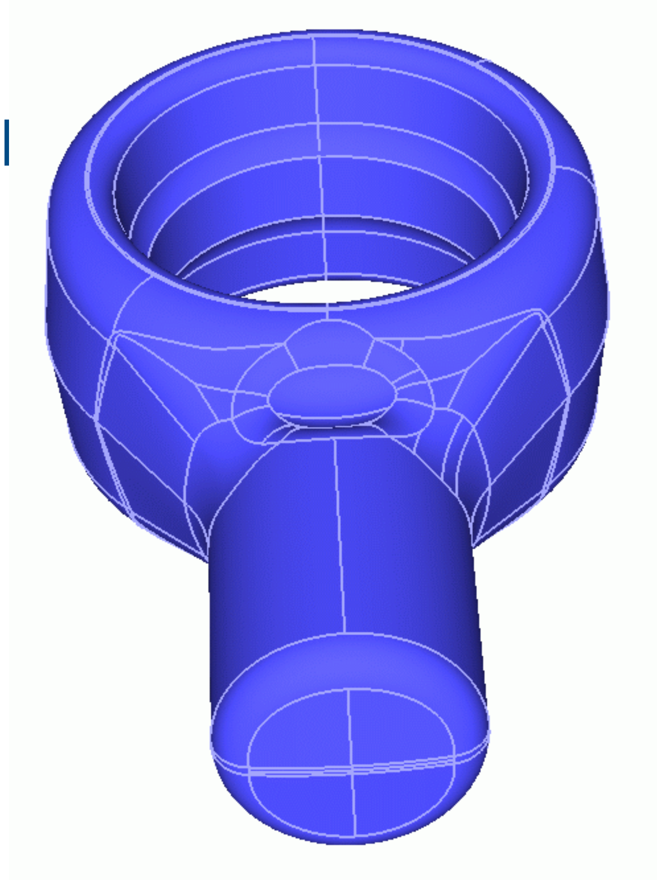


CAD has to change to support isogeometric analysis

- Example: Patch structure of a fairly simple CAD-object
  - Object designed patch by patch to match the desired shape
  - Shape designed for production

CAD patch structure not an obvious guide to isogeometric block structure

- We would like considerably fewer NURBS blocks than the number of surfaces patches in the CAD-model
- The object has three main parts
  - The “torus” like part
  - The “cylindrical” handle
  - The transition between these
- Not obvious how this can be represented as a composition of NURBS blocks
  - Acute angles
  - Extraordinary points
  - Singular points



Current CAD technology is here to stay

- The major part of revenue of CAD vendors comes from industries that don't suffer from the CAD to analysis bottleneck. However, advanced industry suffers.
- Current CAD is standardized in ISO 10303 STEP.
- The driving force for isogeometric CAD has to be industries that has the most to gain from the novel approach, e.g.,
  - aeronautics, defense, space and automotive industries
- Isogeometric CAD: A next natural step in CAD evolution?
- ISO 10303 STEP should also include isogeometric CAD

Two approaches to isogeometric CAD

1. Build the block structure one block at the time
   - User responsible for block interfaces and interfaces to outer and inner hulls.
   - Similar to surface modeling without trimming
   - Can be template based
2. Design the tri-variate block structure in an already existing ISO 10303 STEP type CAD model
   - The user controls the block structure. The blocks snap together and to outer and inner hulls.
   - Similar to designing surfaces into a point cloud in reverse engineering
   - Last week (June 2012) my colleague Vibeke Skytt participated in an ISO meeting in Stockholm to find out how ISO 10303 STEP can better support the idea of isogeometric analysis.

NURBS lack local refinement

- The regular structure of tensor product NURBS does not allow local refinement
- 1988: Forsey & Bartels: Hierarchical B-spline refinement.
  - $f(s,t) = f_1(s,t) + f_2(s,t) + \ldots + f_n(s,t)$
  - The spline space of f_{i+1} is a refinement of the spline space of f_i
- 1998: Rainer Kraft, Adaptive und linear unabhängige multilevel B-splines und ihre Anwendungen. PhD Thesis
- 2003: T. Sederberg, T-splines
  - Compact one level of hierarchical B-splines in the surface control grid. Generalization based on the control grid of B-spline surfaces
- 2006: Deng, PHT-splines
  - C^1 Patchwork of bi-cubic Hermite surface patches allowing T-joints between patches
- 2009: Locally Refined B-splines (LR B-splines), addressing local refinement from the viewpoint of CAGD and Analysis
- 2010 Hierarchical B-splines and IGA – Jüttler, Giannelli, et. al.

Second lecture
Splines spaces over box partitions

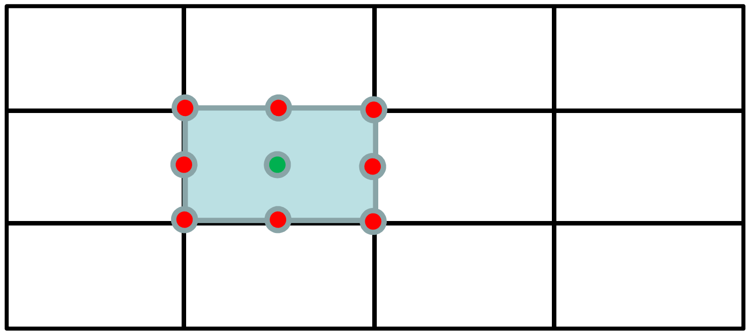
Block structured grids, local refinement and spline bases

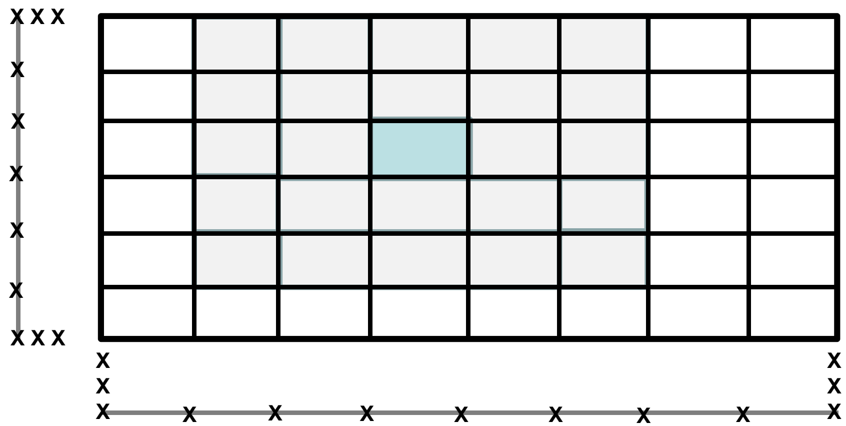
Tor Dokken

Isogeometric Analysis challenge traditional FEA approaches to block structured grids.

- Block structured FEA
  - Composed of coupled blocks of meshes with a regular structure
- Spline function represented by tensor product B-splines
  - The B-splines space defined by univariate knot vectors $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ one in each parameter direction and the polynomial degrees p_1, p_2. A regular structure of not lines
  - Continuity defined by multiplicity of values
  - A tensor product B-spline not local to an element
- Higher order continuity a challenge when blocks meet in an extraordinary point
  - Simple when four surfaces meet
  - Challenge independent of T-splines, LR B-splines….

Bi-quadratic example





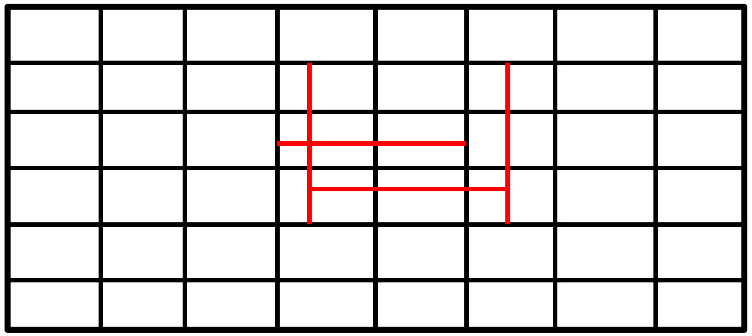
Bi-quadratic finite element

- Shape functions local to each element
- C^0- continuity by matching edge nodes
- C^1- continuity imposes many relations between nodes

Bi-quadratic B-spline

- Shape functions (B-splines) not local to a single element
- Continuity embedded in the B-spline definition through the knot vectors
- C^1- continuity straight forward

Local refinement of a block

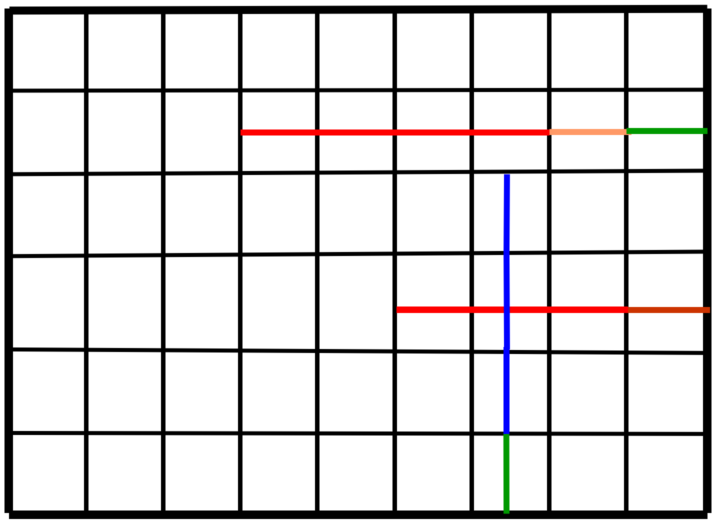


- We want to insert new local line segment in the grid that result in new elements (and degrees of freedom) located where they are needed.
- Questions:
  - What is the dimension of the resulting spline space?
  - How to find a suitable basis?
  - Which restrictions has to be imposed on the refinement?
- The questions are independent of the approach is T-splines Locally Refined B-splines, PHT-splines or Hierarchical B-splines.

Dimension increase for local refinement in the 2-variate case well understood.

Example dimension increase of spline space when

- Bi-cubic $p = 3$.
- C^2-continuity across interior mesh lines



Mesh line segment length 1 starting at the boundary, T-joint at other end. Dimension increase 1.

Mesh line segment length 1 extending existing mesh line segment of length 4 or more, T-joint at other end. Dimension increase 1.

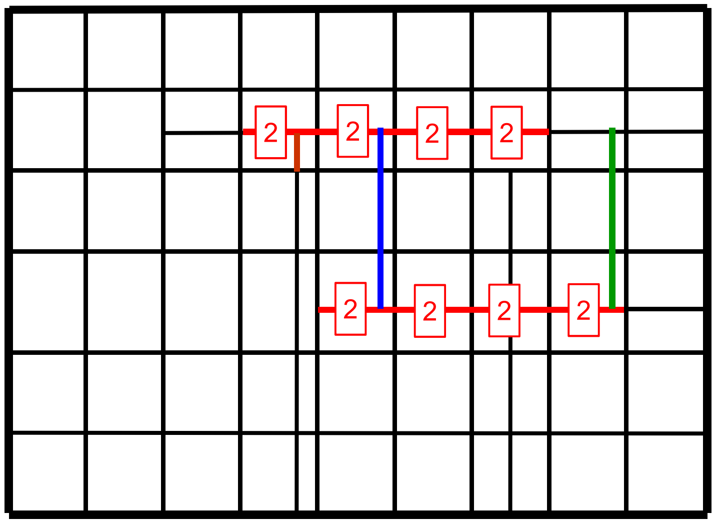
Interior mesh line segment length 4, T-joints at both ends. Dimension increase 1.

Mesh line segment length 1 gap filling. Dimension increase 4.

Mesh line segment length 1 extension of existing mesh line segment to the boundary. Dimension increase 4

Increasing interior multiplicity in the bi-cubic case

- When multiplicity is 2 we have C^1-continuity across interior mesh lines in the bi-cubic case.
- The edges on the boundary have multiplicity 4.



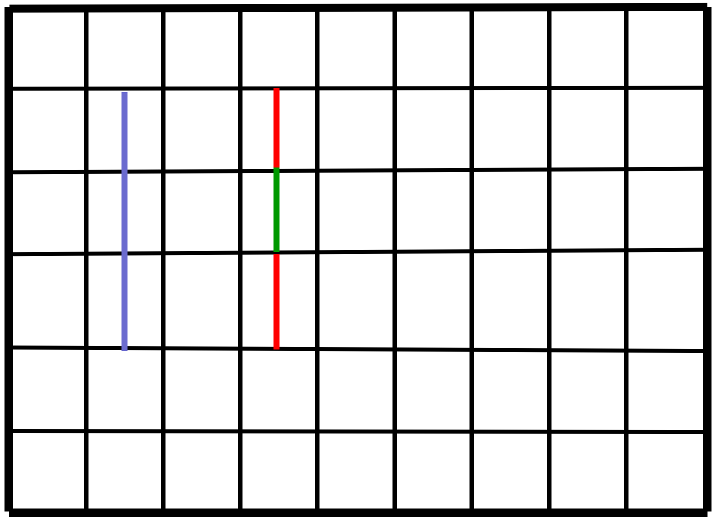
Interior mesh line segment length 4, increase multiplicity to 2, lower multiplicity at both ends, dimension increase 1.

Interior mesh line segment length 3, ending in T-joints with orthogonal mesh line segment , one with multiplicity 1,and one with multiplicity 2, dimension increase 1.

Interior mesh line segment length 3, ending in T-joints with orthogonal mesh line segment of multiplicity 2, dimension increase 2.

Extend existing mesh line segment by length 1, ending in T-joint with orthogonal mesh line segment with multiplicity 2, dimension increase 2,

Short lines do not give dimension increase bi-cubic example



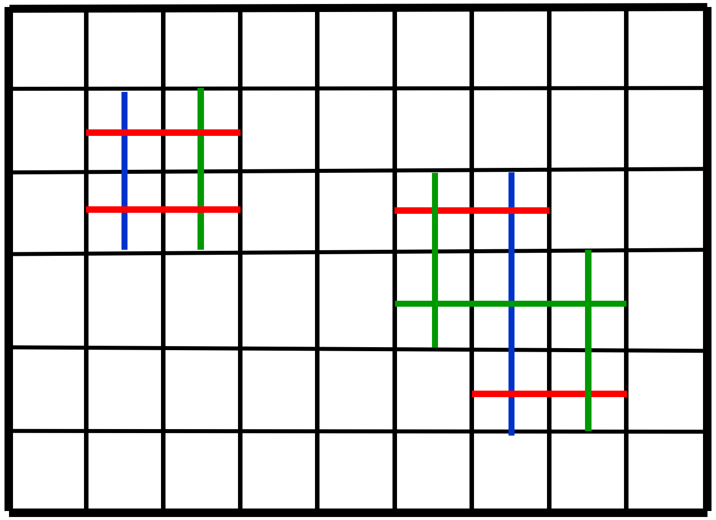
Interior mesh line segment length 3 (or shorter), T-joints at both ends. No dimension increase.

Interior mesh line segment length 3 (or shorter), T-joints at both ends. No dimension increase.

Gap filling mesh line segment length 1, resulting mesh line segment less than 4 (or shorter), T-joints at both ends. No dimension increase.

- Mesh line segment ending in T-joints has to have length $(p + 1)$ to give dimension increase by 1.

Using short lines to localize refinement bi-cubic example



Interior mesh line segment length 2, T-joints at both ends. No dimension increase.

Interior mesh line segment length 3, T-joints at both ends. No dimension increase.

Interior mesh line segment length 4, T-joints at both ends. Dimension increase 1.

- Note: we are only addressing refinement of the spline space. Currently no relation to T-spline, LR B-splines, PHT-splines, Hierarchical B-splines etc.

Bernard Mourrain: *On the dimension of spline spaces on planer T-subdivisions*

- A T-subdivision is a 2-variate specialization of what we will denote an μ-extended box-mesh.
- Dimension of space of bivariate functions, piecewise polynomial of bidegree (p_1, p_2) and class C^{r_1,r_2} over a planar T-subdivision:

$$\begin{aligned}\dim \mathcal{S}_{p_1,p_2}^{r_1,r_2} &= (p_1+1)(p_2+1)f_2 - (p_1+1)(r_2+1)f_1^v \\ &\quad -(p_2+1)(r_1+1)f_1^h + (r_1+1)(r_2+1)f_0 + h_{p_1,p_2}^{r_1,r_2}\end{aligned}$$

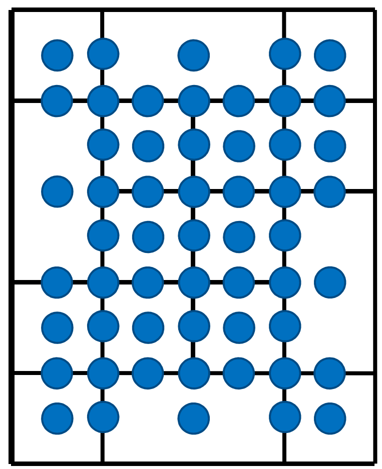
- f_2, f_1^v, f_1^h, f_0 are respectively the number of 2-faces(elements), horisontal interior edges, vertical interior edges and interior vertices
- $h_{p_1,p_2}^{r_1,r_2}$ is a homology term that is zero provided all sequences of interior edges that are not attached to the boundary resepectively consists of at least p_1 or p_2 consequtive edges.

Dimension formula, example

- Dimension of T-mesh space, $(p_1 = p_2 = 3,\ r_1 = r_2 = 2)$:

$$\dim \mathcal{S}_{p_1,p_2}^{r_1,r_2} = (p_1+1)(p_2+1)f_2 - (p_1+1)(r_2+1)f_1^v - (p_2+1)(r_1+1)f_1^h + (r_1+1)(r_2+1)f_0$$

- f_2 = number of 2-faces
- f_1^h = number of interior horisontal edges
- f_1^v = number of interior vertical edges
- f_0 = number of interior vertices



f_1^h=12

$$\dim \mathcal{S}_{3,3}^{2,2} = 4 \times 4 \times 16 - 4 \times 3 \times 13 - 4 \times 3 \times 14 + 3 \times 3 \times 12 = 40$$

Find a piecewise polynomial basis for the locally refined block mesh (1)

- Using Mourrain's dimension formula when it applicable or the more general formula to be presented soon enables in the cases where the homology terms are zero to calculate the spline space over the extended box-mesh.
- We now want to find a basis that span this box splines.
- Some candidate approaches for bases:
  - Hierarchical B-splines
  - PHT-splines
  - T-splines
  - LR B-splines

Find a piecewise polynomial basis for the locally refined block mesh (2)

- **Hierarchical B-splines** have nested levels of refinement, with a nested selection of regions to use at each level.
  - The spline spaced by the extended box-mesh produced by merging the trimmed knotlines of all levels (the element structure) is not spanned by the Hierarchical B-spline basis.
- Hierarchical B-splines do not offer a basis than spans the spline space of the extended box-mesh.

- **PHT-splines** are special basis produced for specific degrees and continuities for very general box-meshes.
  - There exists, e.g., a $C^{1,1}$ bi-cubic PHT-spline basis.
  - PHT-splines do not, as far as I know, offer a general solution.

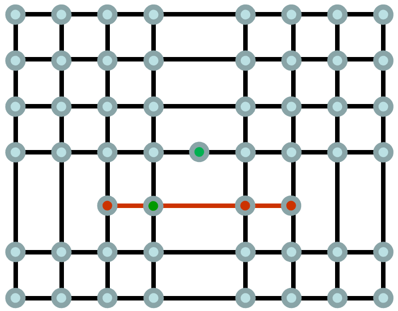
Find a piecewise polynomial basis for the locally refined block mesh (3)

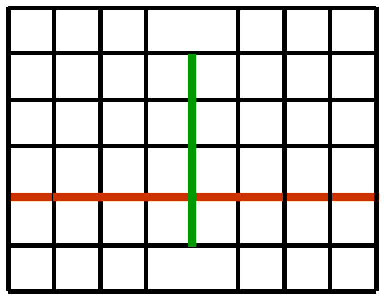
- **T-splines**
  - Refinement defined in the T-spline vertex T-grid following the T-spline rules, recently new extension.
  - Box-mesh (Bezier surfaces) inferred from T-grid
  - Rational scaling of B-splines to form partition of unity
- **LR B-splines**
  - Refinement directly on B-splines in box-mesh, mesh-lines projected on to the surface provides a T-grid structure
  - More choices of refinements than T-splines
  - Scaled B-splines to form partition of unity
- Same challenges of both methods to ensure:
  - Spanning of spline space defined by the Box-mesh
  - Linear independence of the B-splines

Direct modeling by vertices

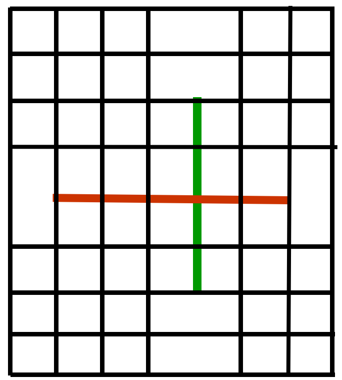
- Both T-splines an LR B-splines have bases that are a partition of unity.
  - Consequently the coefficient/vertices have a geometric interpretation, and the surfaces can be directly manipulated by these.
- The refinement allowed for T-splines is restricted by the requirement that the vertex grid of T-splines, T- joint, recently extensions by L-joints and I-joints.
  - No such structure is imposed on the coefficients of the general LR B-splines. The only restriction, at least one B-spline has to be refined.
  - Direct modeling by vertices is difficult, so in CAD the location of vertices is most often calculated by algorithms.

The refinement approaches of T-spline and LR-spline, bi-cubic example

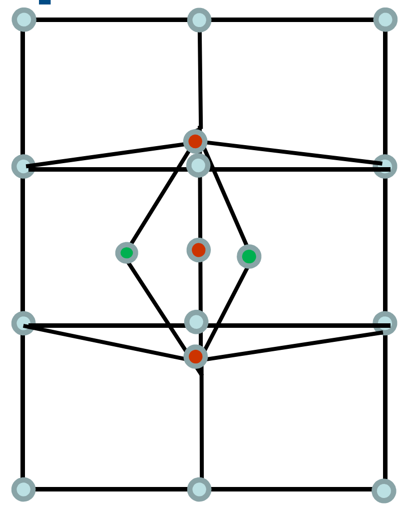




Parameter domain,
Polynomial segments



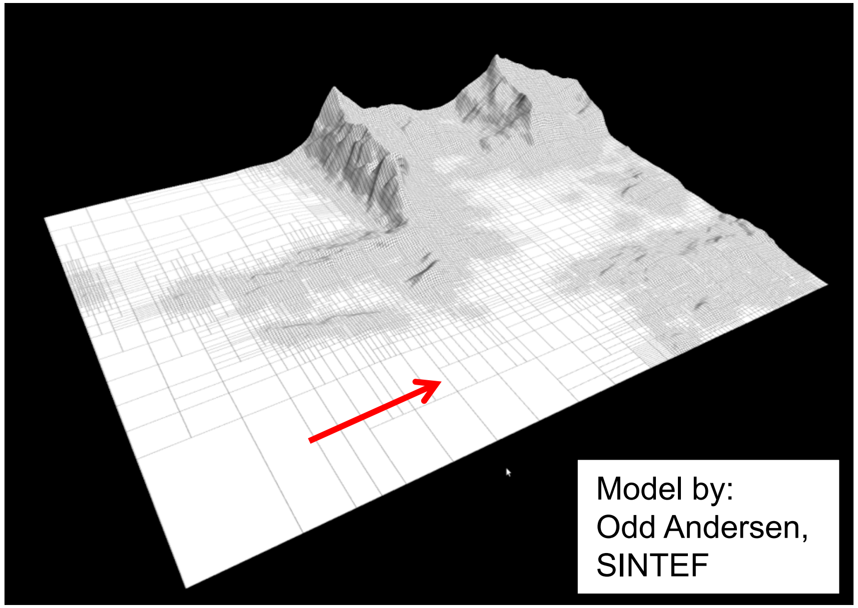
Parameter domain

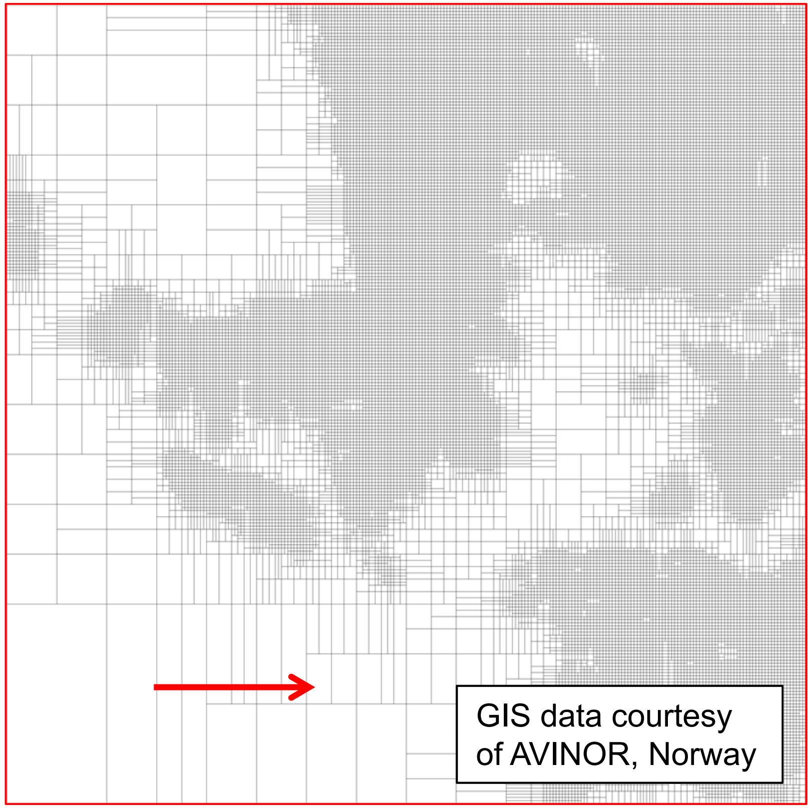


- Adding two close vertices by T-spline refinement creates 11 new polynomial segments and 5 vertices (original T-spline rules)

- Adding a minimal “+” structure by LR B-splines creates 2 vertices and 8 new polynomial segments
- Position of vertices in parameter domain average of internal knots

Local refinement for LR B-splines

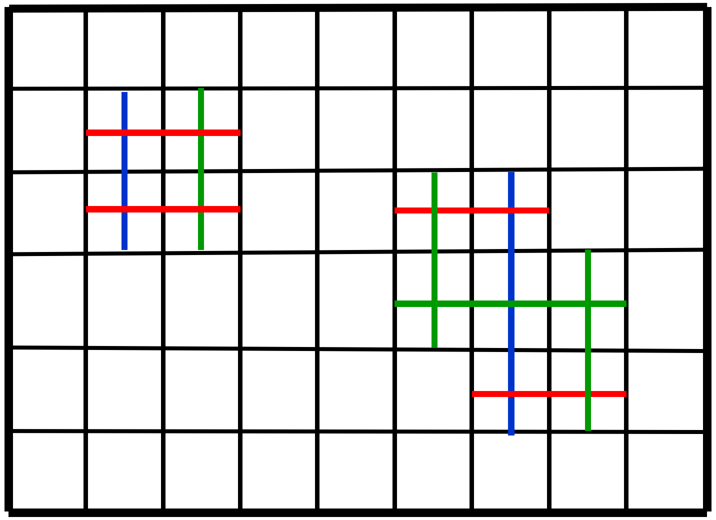




- Refine in mesh project on 3D surface
- Refine in 2D mesh in parameter domain
- Vector specifiying refinement knotline

- Automatic checking:
  - Spline space filled?
  - LR B-spline basis exists?
  - Automatic corrections possible

Support of short lines to localize refinement



- In the $C^{1,1}$ bi-cubic case possibly supported by PHT-splines.
- Currently not supported by T-splines or LR B-splines
  - Challenge: How to extend LR B-spline theory to allow minimal support B-splines in the mesh that are not a result of B-spline refinement.

Summary

Method:	Use B-splines	Fill the spline space of the mesh	Guaranteed basis	Geometric approach to refinement
PHT-splines	No	For specific continuities	For specific continuities	In parameter domain (mapped on surface)
Hierarchal B-splines	Yes	No	Yes	Nested regions
T-splines	Yes	Often	If analysis suitable	In vertex T-grid
LR B-splines	Yes	Often	Tools exist ensue that a basis is made	In parameter domain (mapped on surface)

The theory of LR B-splines applicable to T-splines

Lecture 3
Box-partitions and dimension of spline spaces over Box-partition

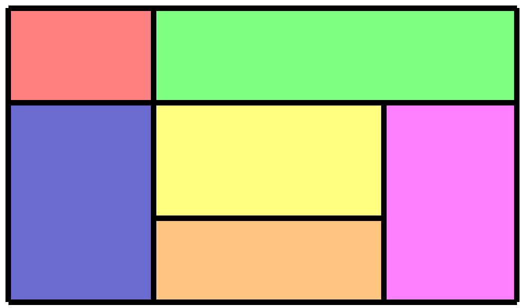
Definition of LR B-splines
some geometric properties

Tor Dokken

Box-partitions

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

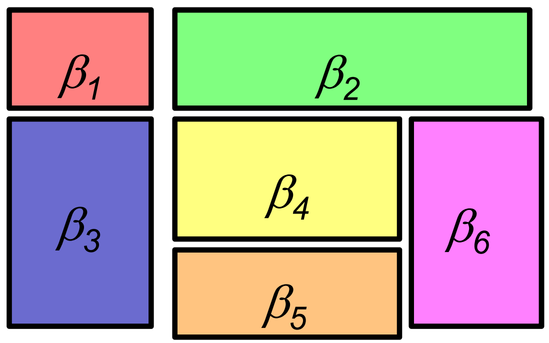
$$\Omega \subseteq \mathbb{R}^d$$
$$\Omega = [a_1, b_1] \times \cdots \times [a_d, b_d]$$
$$a_i < b_i,\ 1 \leq i \leq d$$



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

- Subdivision of Ω into smaller d -boxes

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



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

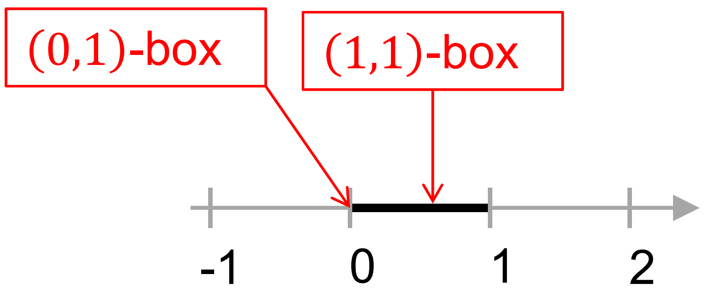
A box in $\mathbb{R}^d$

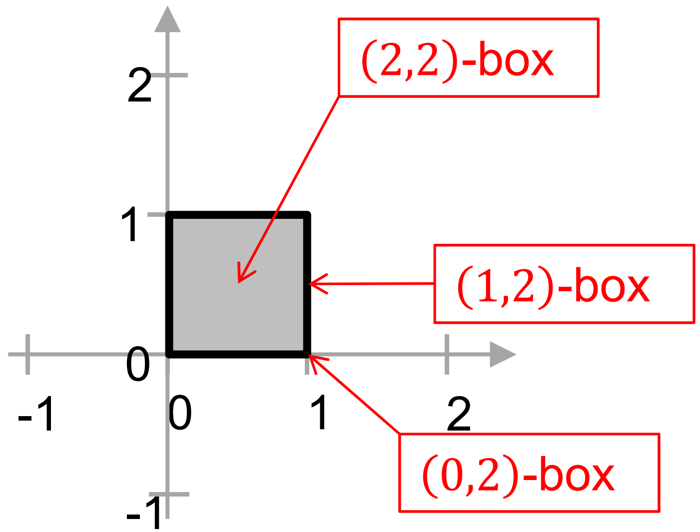
Given an integer $d \geq 0$. A box in $\mathbb{R}^d$ is a Cartesian product

$$\beta = J_1 \times \cdots \times J_d \subseteq \mathbb{R}^d,$$

where each $J_k = [a_k, b_k]$ with $a_k \leq b_k$ is a closed finite interval in $\mathbb{R}^d$. We also write $\beta = [\boldsymbol{a}, \boldsymbol{b}]$, where $\boldsymbol{a} = [a_1, \ldots, a_d]$, and $\boldsymbol{b} = [b_1, \ldots, b_d]$.

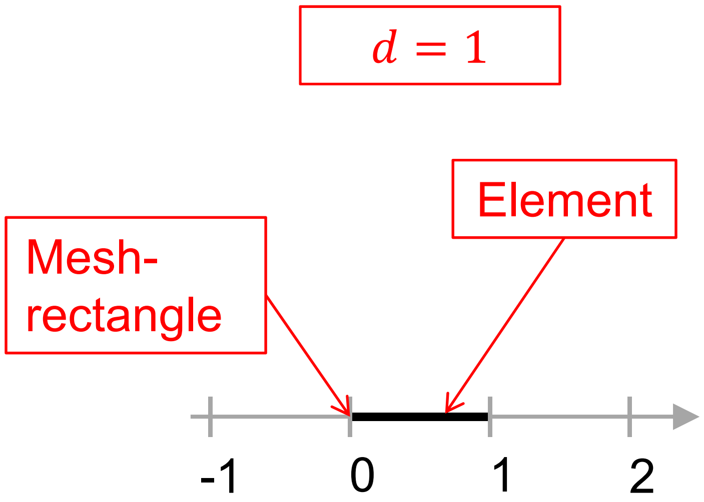
- The interval J_k is said to be **trivial** if $a_k = b_k$ and **non-trivial** otherwise.
- The **dimension** of β, denoted $\dim \beta$, is the number of non-trivial intervals J_k in β.
  - We call β an l-box or and (l, d)-box if $\dim \beta = l$.

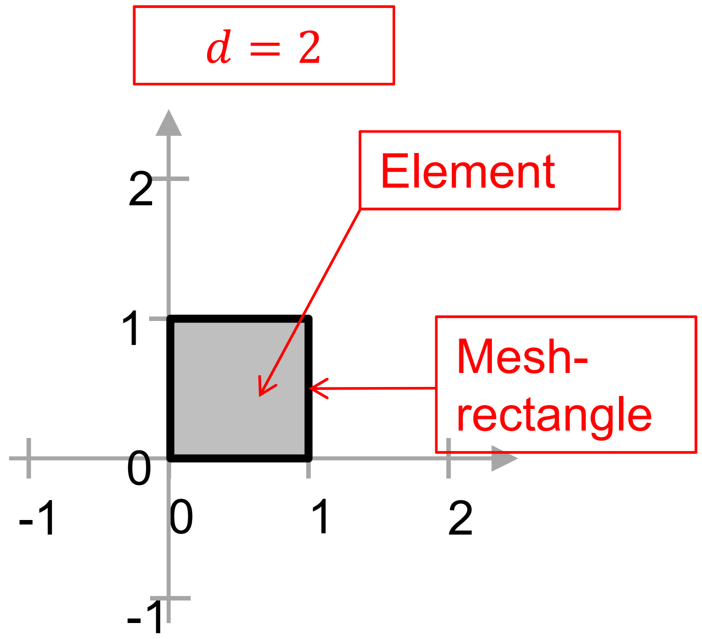




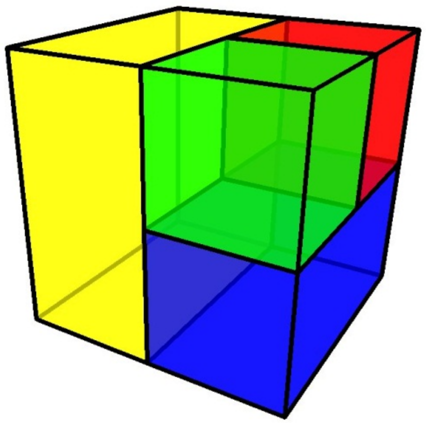
Important boxes

- If $\dim \beta = d$ then β is called an **element**.
- If $\dim \beta = d - 1$ there exists exactly one k such that $J_k = [a]$ is trivial. Then β is called a **mesh-rectangle**, a k-**mesh-rectangle** or a (k, a)-**mesh-rectangle**

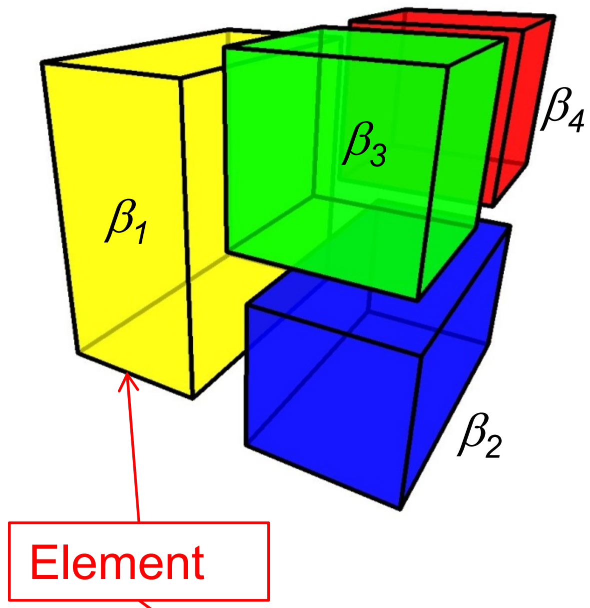




Example: 3D-mesh



$\Omega \subseteq \mathbb{R}^3$



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

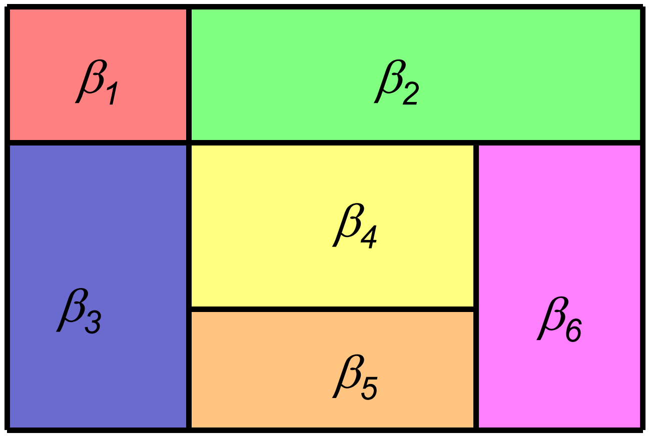
Illustration by: Kjell Fredrik Pettersen, SINTEF

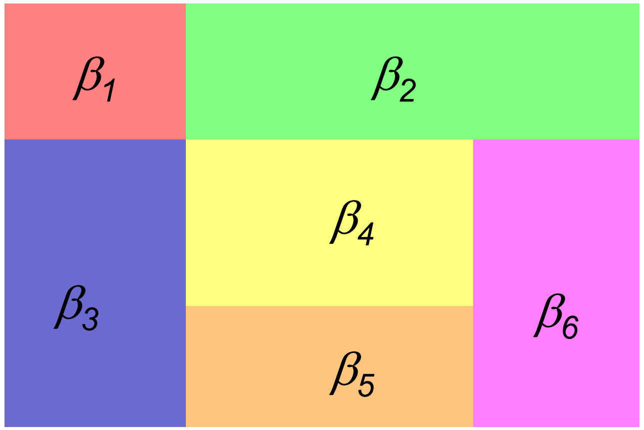
Lower-dimensional boxes in the mesh

- $\mathcal{F}_l(\mathcal{E})$ is the set of l -boxes describing the mesh topology, $0 \leq l \leq d$
- $\mathcal{F}_d(\mathcal{E})$ is the same as $\mathcal{E}$
- For $l < d$: $\mathcal{F}_l(\mathcal{E})$ is the set of l-boxes where higher-dimensional boxes in $\mathcal{E}$ intersect, or at boundary of Ω

Example: 2D-mesh

dim = 2 (Elements)



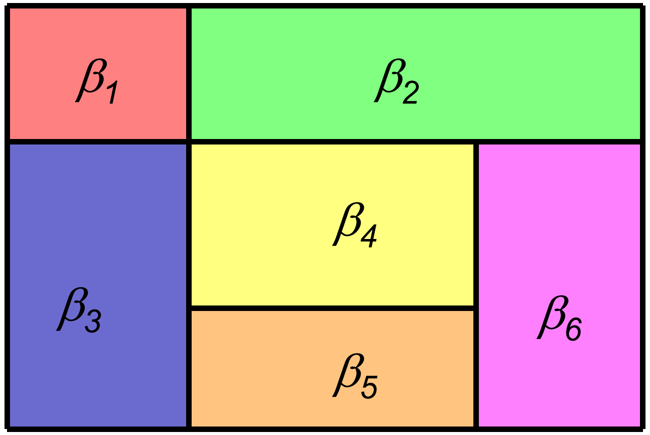


$$\mathcal{E} = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$$

$$\mathcal{F}_2(\mathcal{E}) = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\} = \mathcal{E}$$

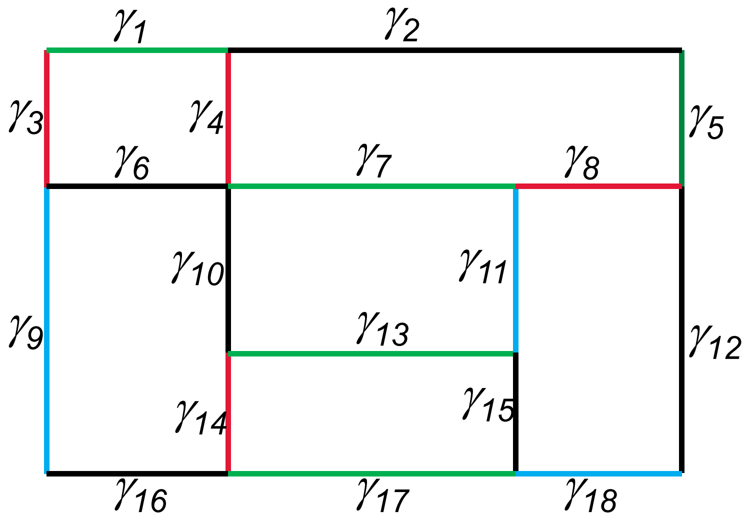
Illustration by: Kjell Fredrik Pettersen, SINTEF

Example: 2D-mesh



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

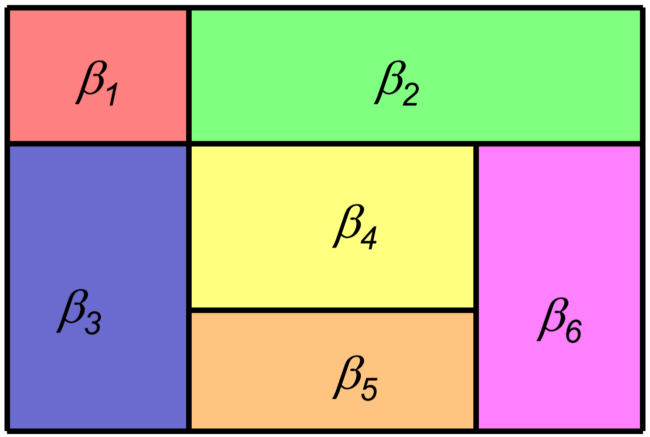
dim = 1 (Mesh-rectangles)



$$\mathcal{F}_1(\mathcal{E}) = \{\gamma_1, \gamma_2, \gamma_3, \ldots, \gamma_{18}\}$$

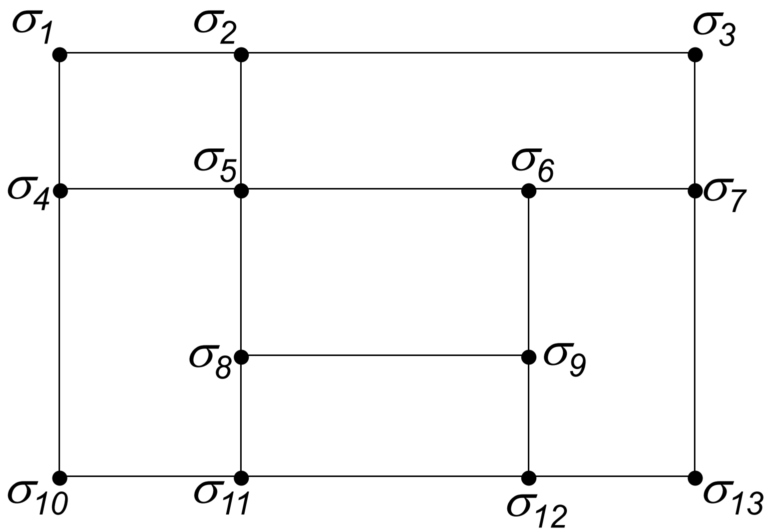
Illustration by: Kjell Fredrik Pettersen, SINTEF

Example: 2D-mesh



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

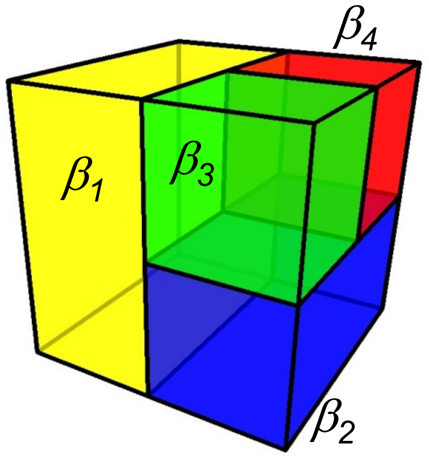
dim = 0



$$\mathcal{F}_0(\mathcal{E}) = \{\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{13}\}$$

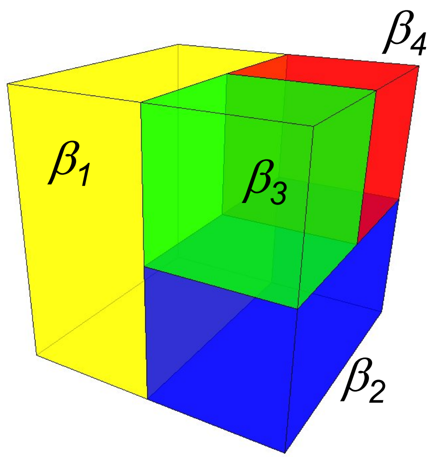
Illustration by: Kjell Fredrik Pettersen, SINTEF

Example: 3D-mesh



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

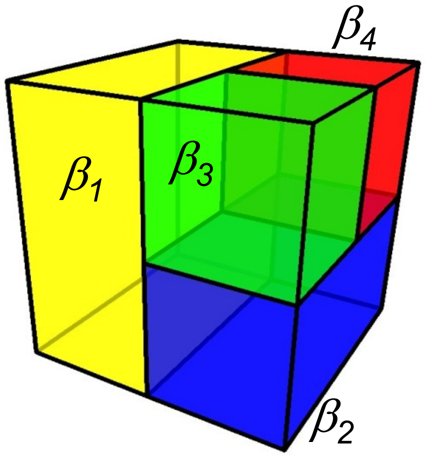
dim = 3 (Elements)



$\mathcal{F}_3(\mathcal{E}) = \{\beta_1, \beta_2, \beta_3, \beta_4\} = \mathcal{E}$

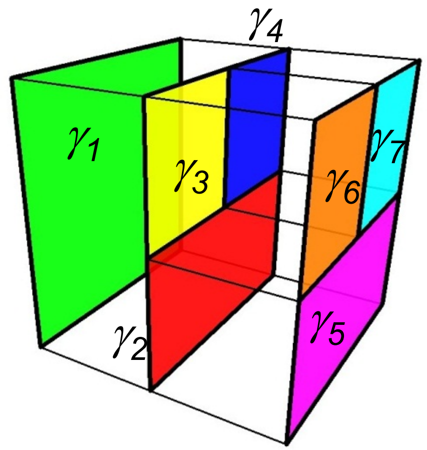
Illustration by: Kjell Fredrik Pettersen, SINTEF

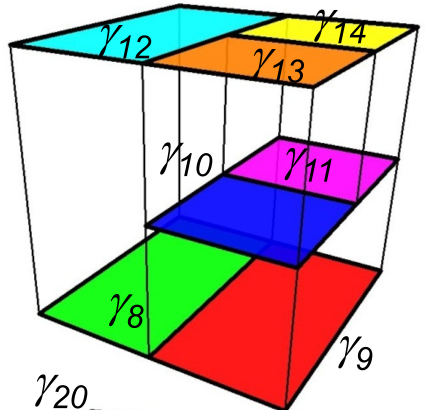
Example: 3D-mesh



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

dim = 2 (Mesh-rectangles)





$$\mathcal{F}_2(\mathcal{E}) = \{\gamma_1, \ldots, \gamma_{21}\}$$

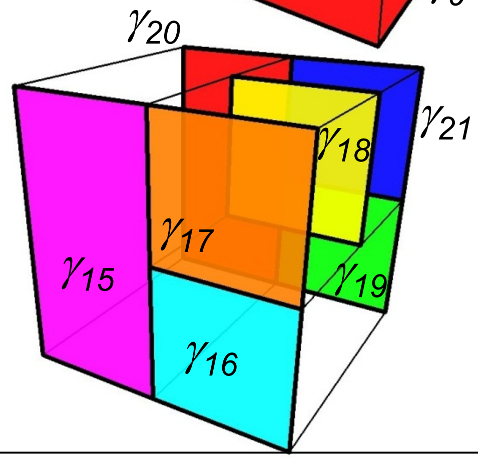
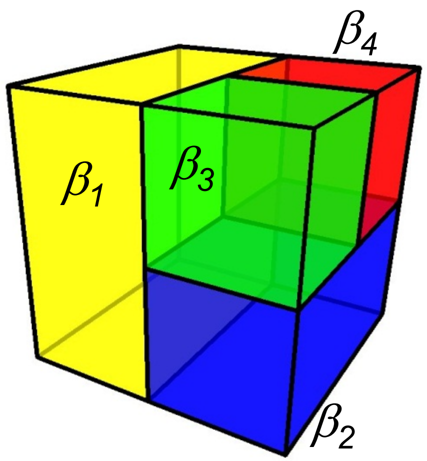


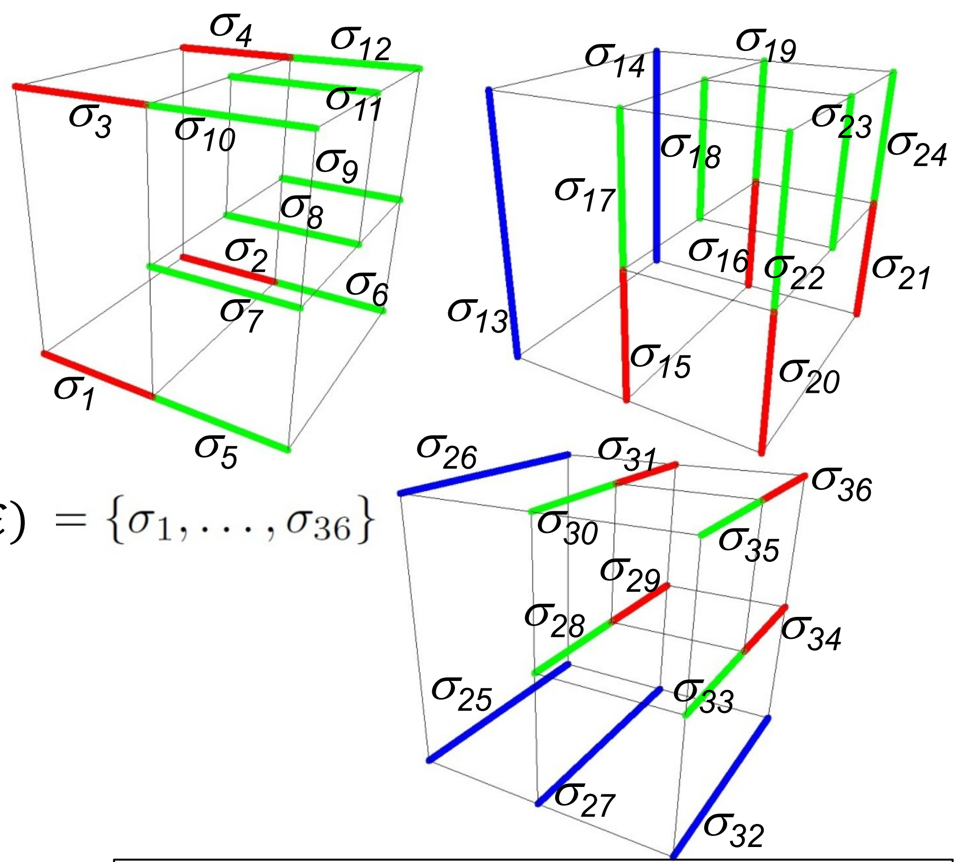
Illustration by: Kjell Fredrik Pettersen, SINTEF

Example: 3D-mesh



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

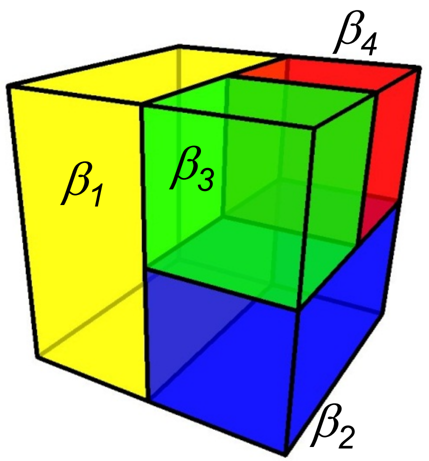
dim = 1



$$\mathcal{F}_1(\mathcal{E}) = \{\sigma_1, \ldots, \sigma_{36}\}$$

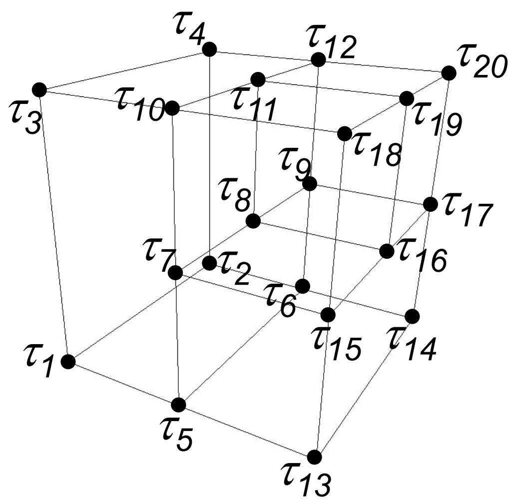
Illustration by: Kjell Fredrik Pettersen, SINTEF

Example: 3D-mesh



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

dim = 0

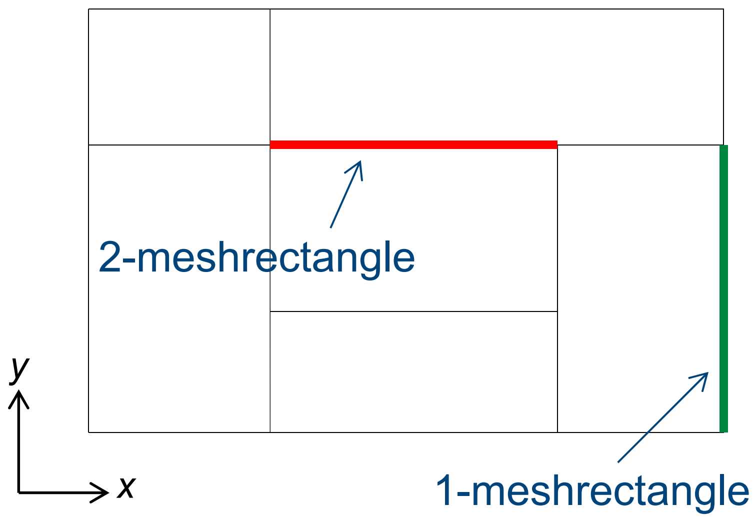


$$\mathcal{F}_0(\mathcal{E}) = \{\tau_1, \ldots, \tau_{20}\}$$

Illustration by: Kjell Fredrik Pettersen, SINTEF

Mesh-rectangles

Example, 2D



Example, 3D

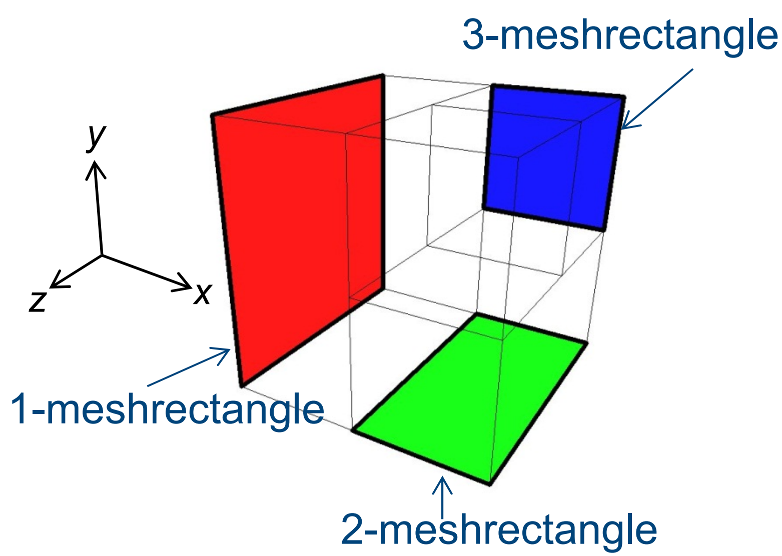
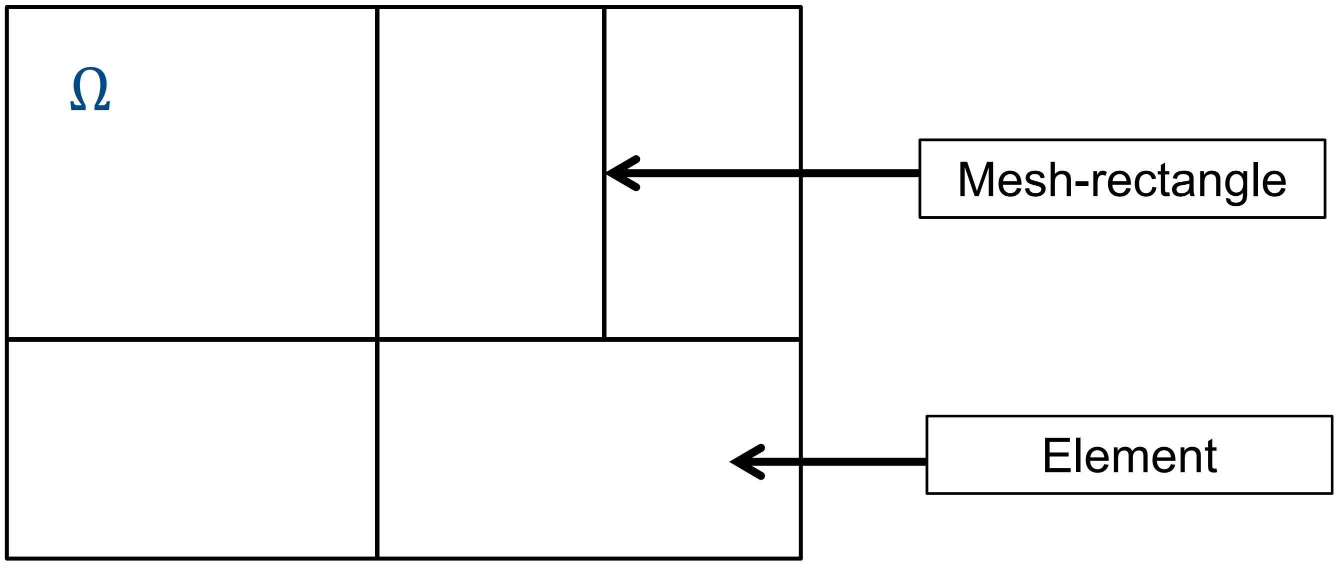


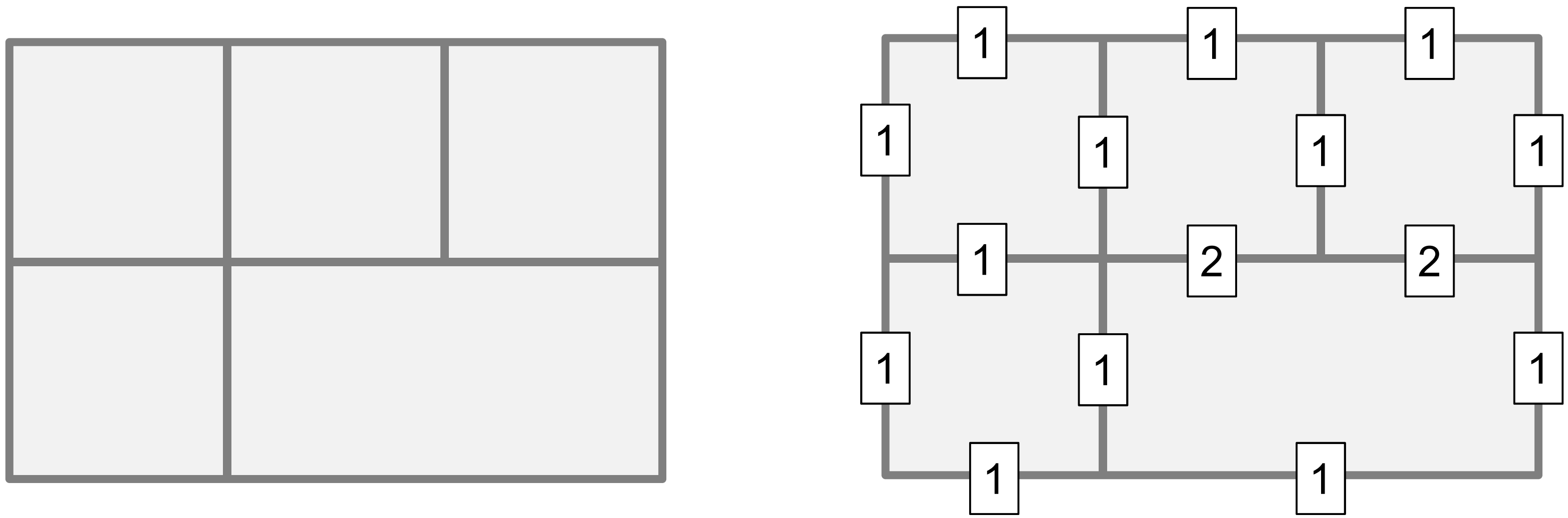
Illustration by: Kjell Fredrik Pettersen, SINTEF

Box-partition

- $\Omega \subseteq \mathbb{R}^d$ a d-box in $\mathbb{R}^d$.
- A finite collection $\mathcal{E}$ of d-boxes in $\mathbb{R}^d$ is said to be a **box partition** of Ω if
  1. $\beta_1^o \cap \beta_2^o = \emptyset$ for any $\beta_1^o, \beta_2^o \in \mathcal{E}$, where $\beta_1^o \neq \beta_2^o$.
  2. $\bigcup_{\beta \in \mathcal{E}} \beta = \Omega$.



μ-extended box-mesh (adding multiplicities)



- 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.

Comment

- When our work started out we used the index grid for knots.
  - However, this posed challenges with respect to mesh-rectangles of multiplicity higher than one and the uniqueness of refinement.
  - The generalized dimension formula uses multiplicity, thus to ensure
  - Not straight forward to understand what happens with spline space dimensionality when two knotline segments converge and optain the same knot value.
- To make a consistent theory we discarded the index grid.

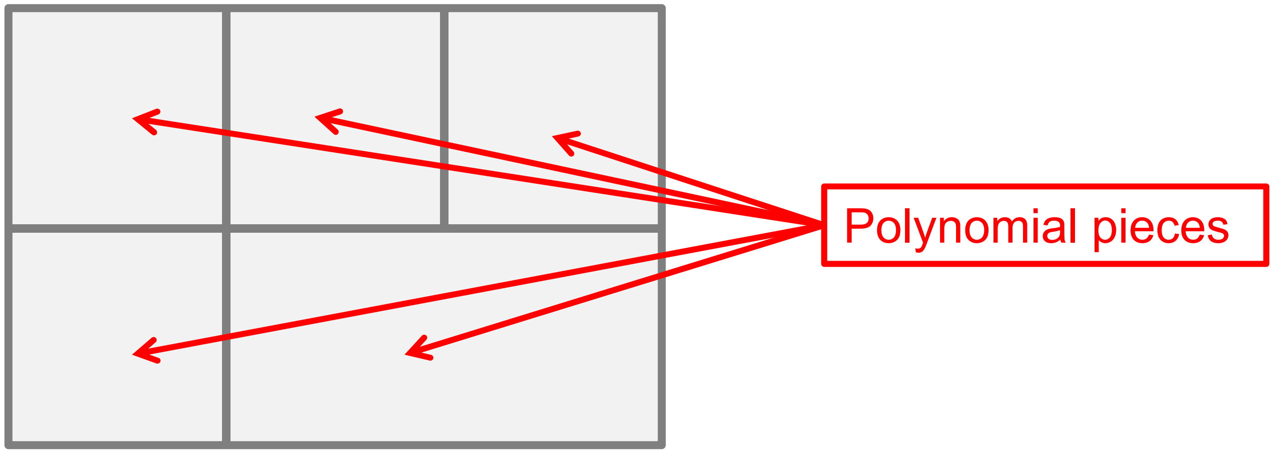
Polynomials of component degree

On each element the spline is a polynomial.

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

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

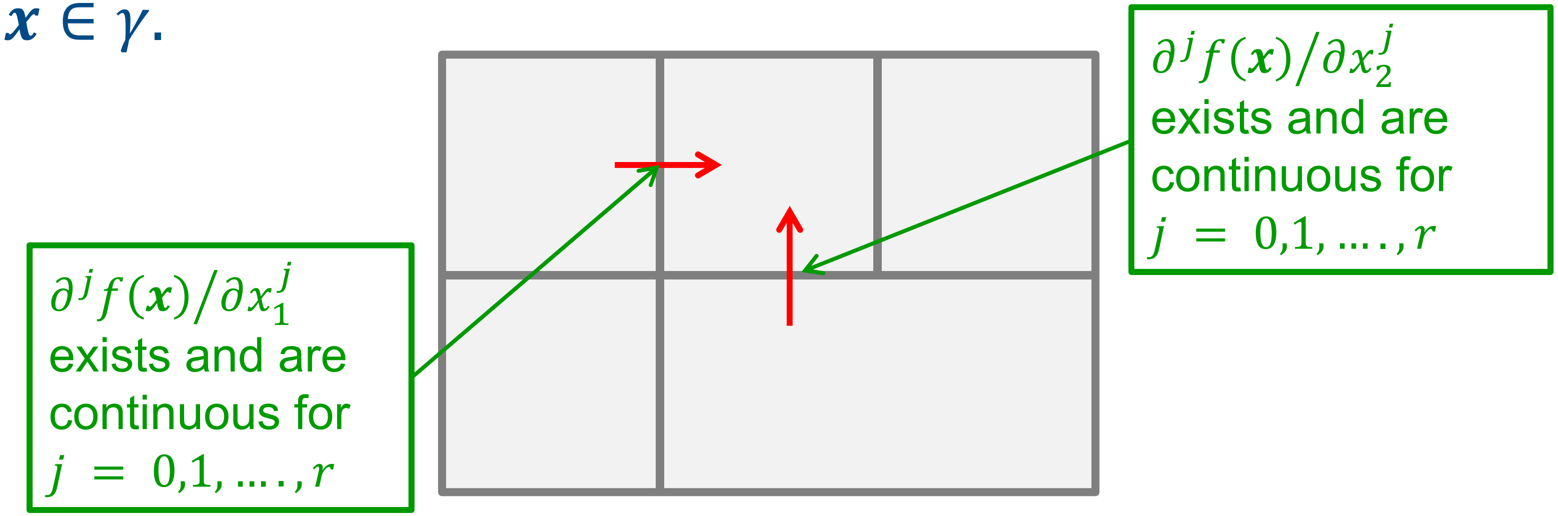
$\boldsymbol{p} = (p_1, \dots, p_d)$
$\boldsymbol{i} = (i_1, \dots, i_d)$



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 $1 \le k \le d$.

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 $\boldsymbol{x} \in \gamma$.

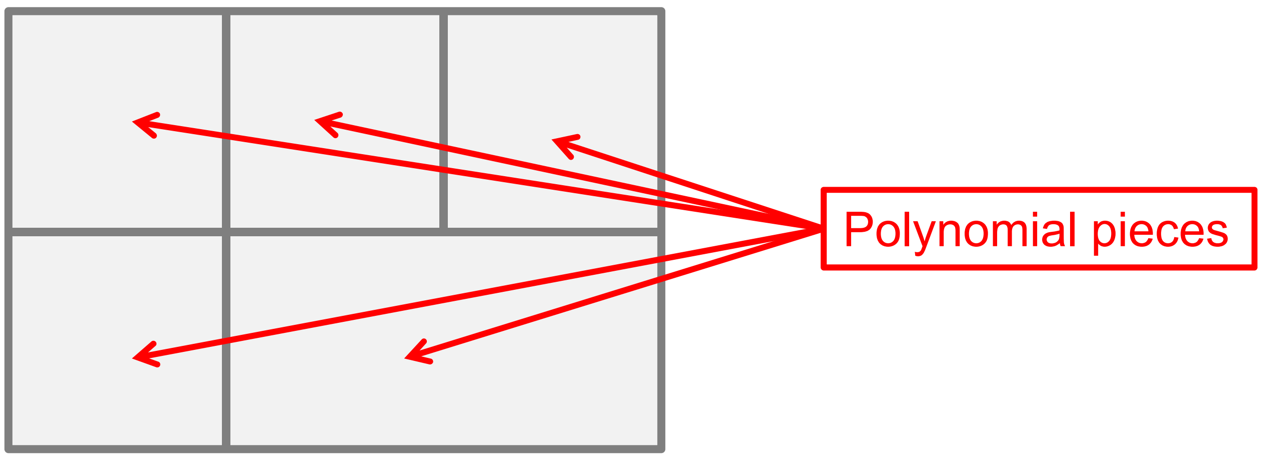


Piecewise polynomial space

We define the piecewise polynomial space

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

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



Spline space

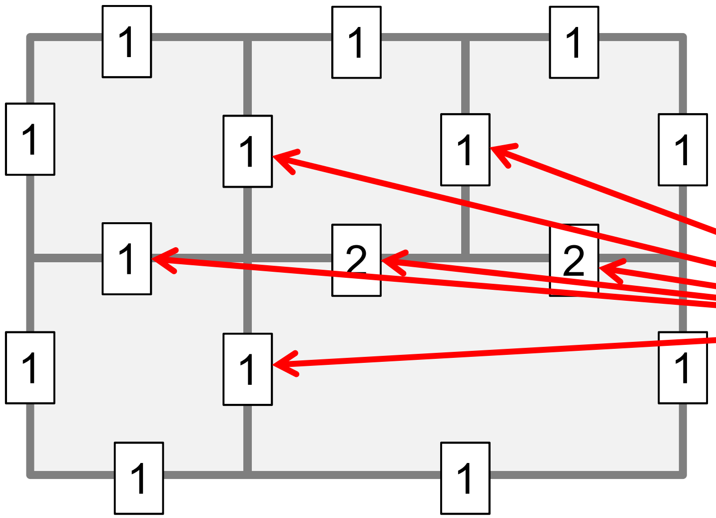
Polynomial degree in direction k

Continuity across k-mesh-rectangle γ

We define the spline space

$$\mathbb{S}_{\boldsymbol{p}}(\mathcal{M},\mu) = \Big\{f \in \Pi_{\boldsymbol{p}}^{d}\big(\mathcal{E}(\mathcal{M})\big) : f \in C^{p_k-\mu(\gamma)}(\gamma),$$
$$\forall \gamma \in \mathcal{F}_{d-1,k}\big(\mathcal{E}(\mathcal{M})\big), k = 1, \dots, d\Big\}$$

All k-mesh-rectangles



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

How to measure dimensional of spline space of degree p over a μ-extended box partition $(\mathcal{M}, \mu)$.

- Dimension formula developed (Mourrain, Pettersen)

$$\dim \mathbb{S}_{\boldsymbol{p}}(\mathcal{M},\boldsymbol{\mu}) = \boxed{\sum_{\ell=0}^{d}(-1)^{d-\ell}\left(\sum_{\beta\in\mathcal{F}_\ell(\mathcal{M})}\prod_{k=1}^{d}(p_k-\mu_k(\beta)+1)\right)} \boxed{-\sum_{q=0}^{d-1}(-1)^{d-q}\dim H_q(\widetilde{\mathfrak{S}}(\mathcal{N}))}$$

Summing over all l-boxes of all dimensions

Dimension influenced by mesh-rectangle multiplicity

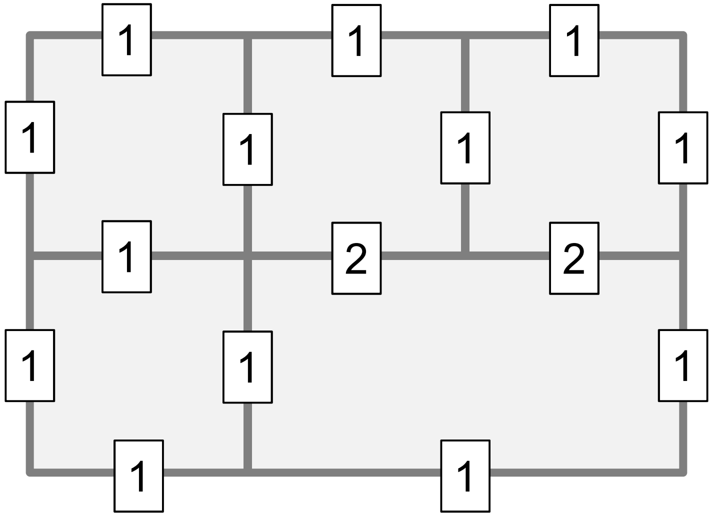
Combinatorial values calculated from topological structure

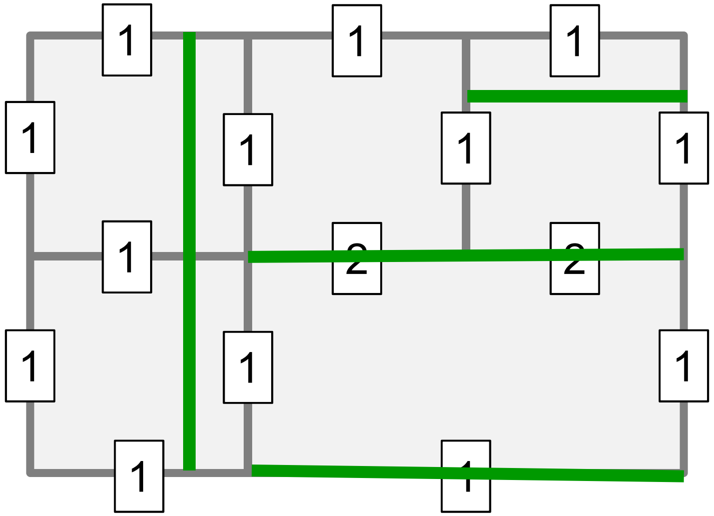
Homology terms

- In the case of 2-variate LR-splines always zero

LR B-splines over μ-extended box-mesh

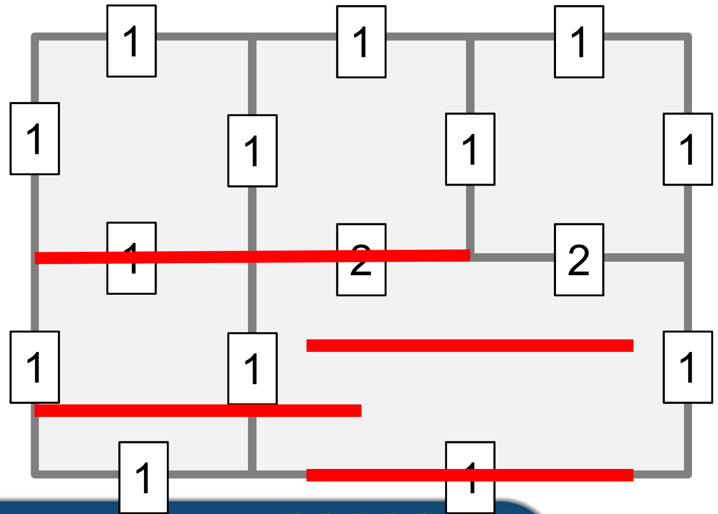
Refinement by inserting mesh-rectangles giving a constant split





Constant split

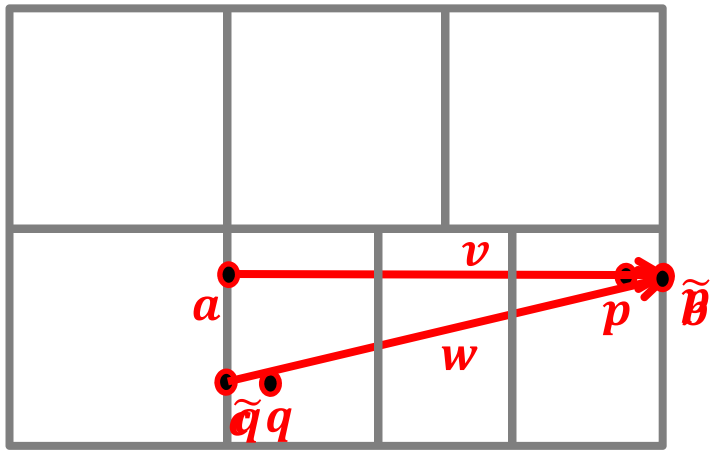
Not constant split



Interactively specifying mesh-rectangles – 2-variate case

Idea developed together with Peter Nørtoft Nielsen and Odd Andersen, SINTEF

- A mesh-rectangle is defined by the two the extreme corners
  - $(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{v})$
  - Where $\boldsymbol{v} = (v_1, v_2)$, with just one of v_1, v_2 zero
- Given two points $\boldsymbol{p}$ and $\boldsymbol{q}$ as input. Snap $\boldsymbol{p}$ to the nearest mesh-rectangle to create $\tilde{\boldsymbol{p}}$, remember the class of k-mesh-rectangles snapped to, $k \in \{1,2\}$. Snap $\boldsymbol{q}$ to the nearest parallel k-meshrectangle to create $\tilde{\boldsymbol{q}}$.
  - Make $\boldsymbol{c}$ where $c_j = \min(\tilde{p}_j, \tilde{q}_j)$
  - Make $\boldsymbol{w}$ where $w_j = |\tilde{p}_j - \tilde{q}_j|$
  - Define $\boldsymbol{a} = (a_1, a_2)$, $a_j = \begin{cases} p_k, j \neq k \\ c_j, j = k \end{cases}$
  - Define $\boldsymbol{v} = (v_1, v_2)$, $v_j = \begin{cases} 0, j \neq k \\ w_j, j = k \end{cases}$

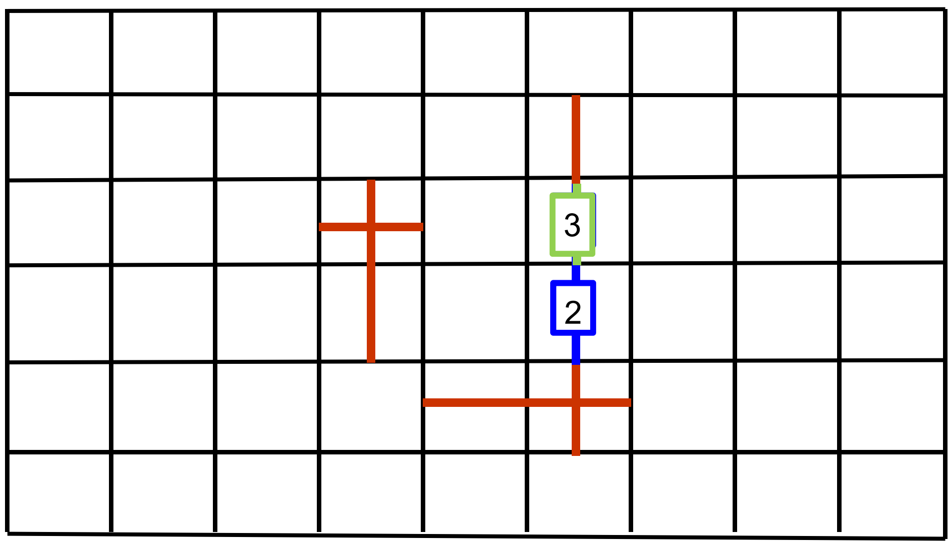


- Provided $v_j > 0$, for $j = k$, we have a well defined mesh-rectangle define by$(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{v})$
- When creating mesh-rectangles automatic checks can be run with respect to the increase of dimensionality, and if the resulting B-splines form a basis.

μ-extended LR-mesh

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

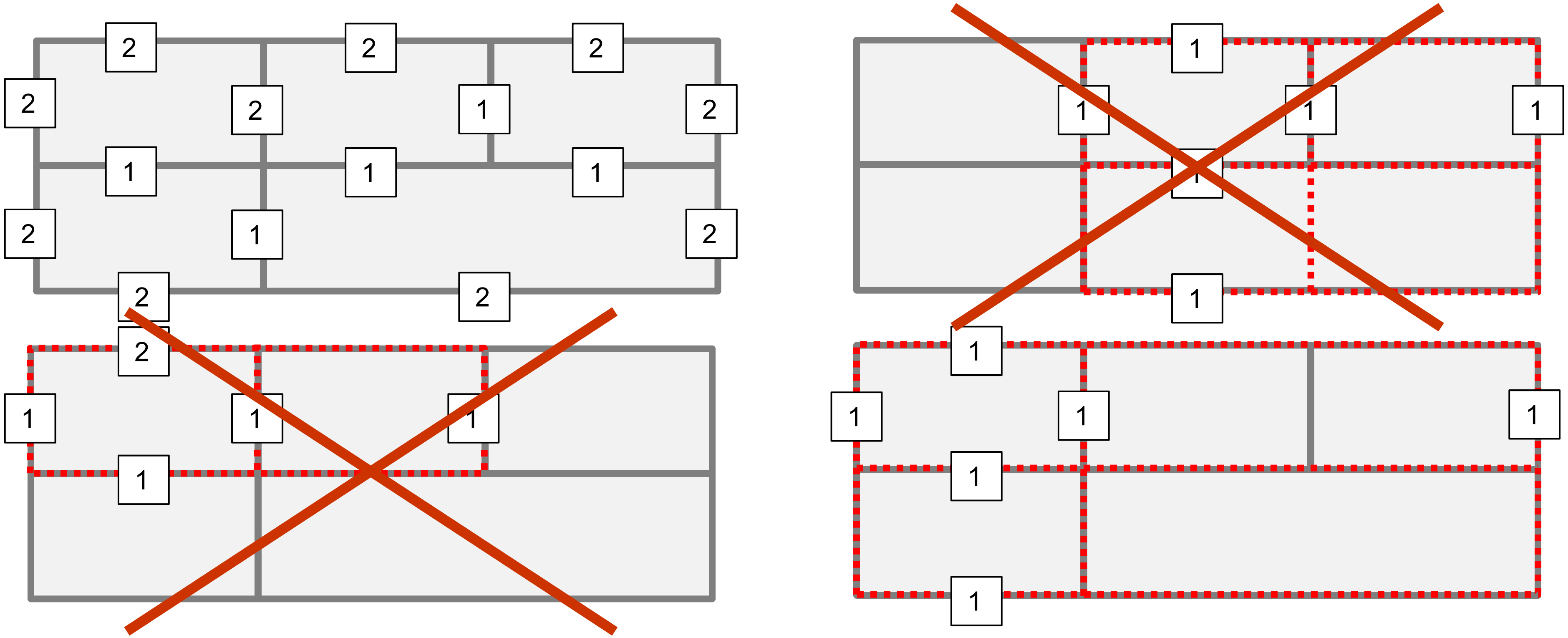
1. $(\mathcal{M}, \mu)$ is a tensor-mesh with knot multiplicities or
2. $(\mathcal{M}, \mu) = (\widetilde{\mathcal{M}} + \gamma, \tilde{\mu}_\gamma)$ where $(\widetilde{\mathcal{M}}, \tilde{\mu})$ is a μ -extended LR-mesh and γ is a constant split of $(\widetilde{\mathcal{M}}, \tilde{\mu})$.



All multiplicities not shown are 1.

LR B-spline

Let (M, μ) be an μ-extended LR-mesh in $\mathbb{R}^d$. A function $B: \mathbb{R}^d \to \mathbb{R}$ 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

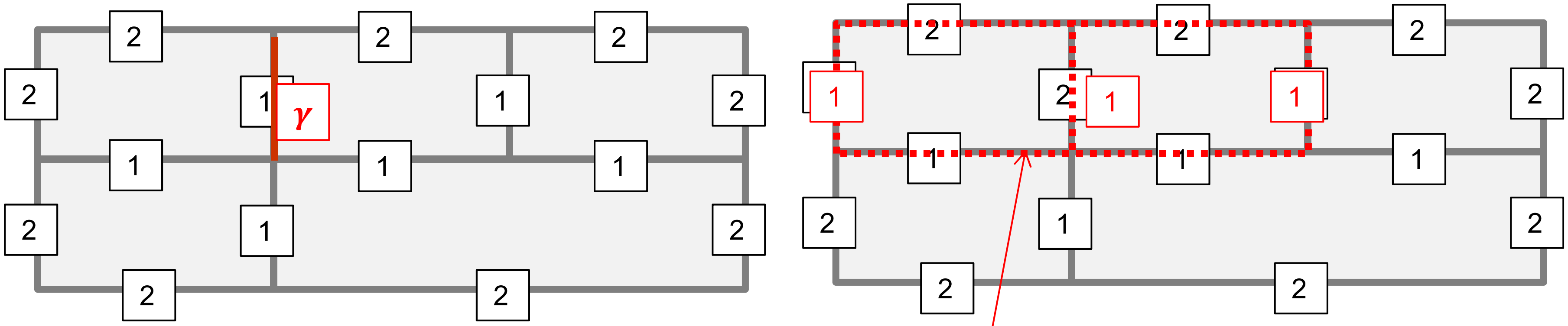
We define as sequence of $\boldsymbol{\mu}$-extended LR-meshes $(\mathcal{M}_1, \mu_1), \dots, (\mathcal{M}_q, \mu_q)$ with corresponding collections of minimal support B-splines $\mathcal{B}_1, \dots, \mathcal{B}_q$.

For $j = 1, \dots, q-1$ creating $(\mathcal{M}_{j+1}, \mu_{j+1}) = (\mathcal{M}_j + \gamma_j, \mu_{j,\gamma_j})$ from $(\mathcal{M}_j, \mu_j)$ involves inserting a mesh-rectangles γ_j that increases the number of B-splines. More specifically:

- γ_j splits $(\mathcal{M}_j, \mu_j)$ in a constant split.
- at least on B-spline in $\mathcal{B}_j$ does not have minimal support in $(\mathcal{M}_{j+1}, \mu_{j+1})$.

After inserting γ_j we start a process to generate a collection of minimal support B-splines $\mathcal{B}_{j+1}$ over $(\mathcal{M}_{j+1}, \mu_{j+1})$ from $\mathcal{B}_j$.

Going from $(\mathcal{M}_j, \mu_j)$ to $(\mathcal{M}_{j+1}, \mu_{j+1})$

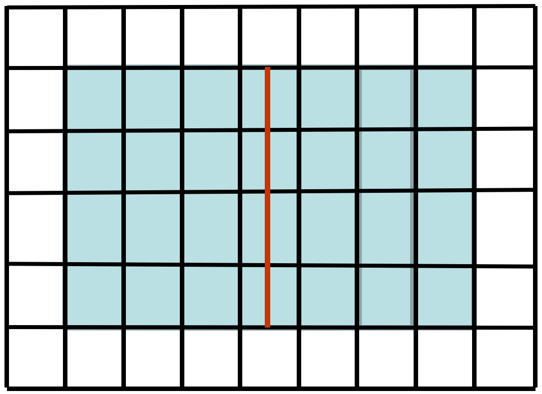
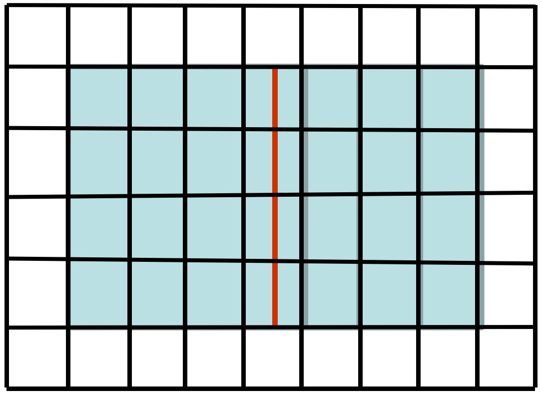


$(\mathcal{M}_j, \mu_j)$ $(\mathcal{M}_{j+1}, \mu_{j+1})$

B-spline from $\mathcal{B}_j$ that has to be split to generate $\mathcal{B}_{j+1}$

LR B-Spline Refinement step
Cubic example: One line

- Insert knot line segments that at least span the width of one basis function

Four B-splines functions that do not have minimal Support in the refined mesh

- 4 B-splines to be removed
- 5 B-splines to be added
- Dimension increase 1

Refinement of LR B-splines is focused on the tensor product B-splines

Let d be a positive integer, suppose $\boldsymbol{p} = (p_1, \ldots, p_d)$ has nonnegative components (the degrees), and let $\boldsymbol{y}_k; = \left(y_{k,1}, \ldots, y_{k,p_k+2}\right)$ be a nondecreasing (knot) sequence $k = 1, \ldots, d$. We define a tensor product B-spline $B[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_d]\colon \mathbb{R}^d \to \mathbb{R}$ by

$$\boxed{B[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_d](x_1, \ldots, x_d)} := \boxed{\prod_{k=1}^{d} B[\boldsymbol{y}_k]\,(x_k\,).}$$

Tensor product B-spline.

Product of Univariate B-splines.

Refinement of a tensor product B-spline

- The support of B is given by the cartesian product

$$\mathrm{supp}(B) \coloneqq [y_{1,1}, \dots, y_{1,p_k+2}] \times \cdots \times [y_{d,1}, \dots, y_{d,p_k+2}].$$

- Suppose we insert a knot z in $(y_{k,1}, \dots, y_{k,p_k+2})$ for some $1 \le k \le d$. Then

$$B[\boldsymbol{Y}] = \alpha_1 B[\boldsymbol{Y}_1] + \alpha_2 B[\boldsymbol{Y}_2]$$

- Where $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ are the knot vectors of the resulting tensor product B-splines, and

$$\alpha_1 \coloneqq \begin{cases} 1 & y_{k,p_k+1} \le z < y_{k,p_k+2} \\ \dfrac{z - y_{k,1}}{y_{k,p_k+1} - y_{k,1}} & y_{k,1} < z < y_{k,p_k+1} \end{cases}$$

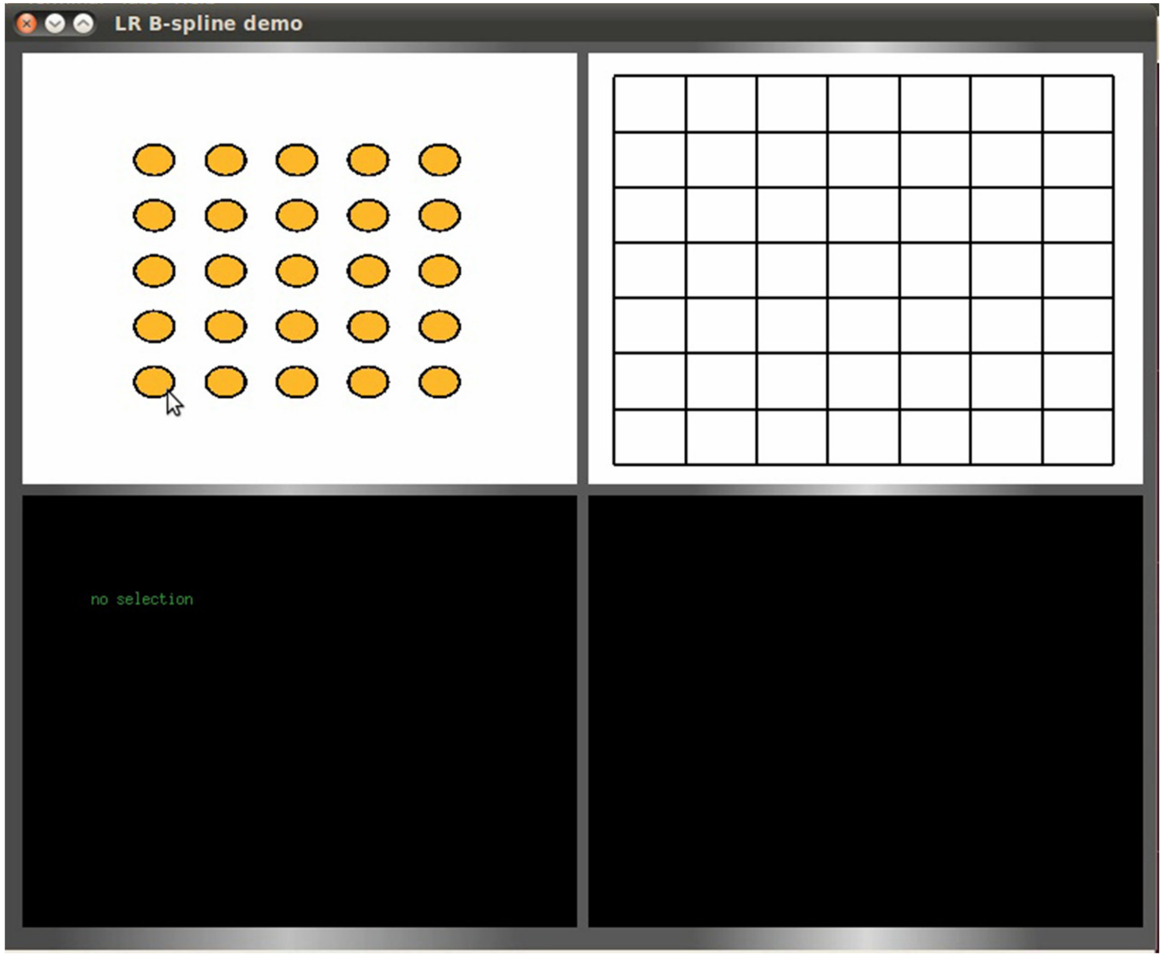
$$\alpha_2 \coloneqq \begin{cases} 1 & y_{k,1} \le z \le y_{k\ 2} \\ \dfrac{y_{k,p_k+2} - z}{y_{k,p_k+2} - y_{k,2}} & y_{k,2} < z < y_{k,p_k+2} \end{cases}$$

- α_1 and α_2 calculated by Oslo Algorithm/Boehms algorithm

LR B-splines and partition of unity

- The LR B-spline refinement starts from a partition of unity tensor product B-spline basis.
- By accumulating the weights α_1 and α_2 as scaling factors for the LR B-splines, partition of unity is maintained throughout the refinement for the scaled collection of tensor product B-splines
- The partition of unity properties gives the coefficients of LR B-splines the same geometric interpretation as B-splines and T-splines.
  - However, the spatial interrelation of the coefficients is more intricate than for T-splines as the refinement strategies are more generic than for T-splines.
  - This is, however, no problem as in general algorithms calculate the coefficients both in FEA and CAD.

Example LR B-spline refinement



Video by PhD fellow Kjetil A. Johannessen, NTNU, Trondheim, Norway.

Lecture 4
LR B-splines and linear independence

+ examples

Tor Dokken

Spline space and μ-extended LR-mesh

- We introduced in Lecture 3:
  - The μ-extended box-mesh
  - The dimension formula
  - The μ-extended LR-mesh
  - The LR B-splines
- In Lecture 2 we focused on the importance of:
  - Spanning the spline space over the μ-extended box-mesh
  - Finding a basis for the spline space
- In this Lecture we focus approaches for ensuring that the LR B-splines is a basis for the spline space defined by μ-extended LR-mesh by:
  - Defining a hand in-hand-property between the LR B-splines and the spline space over the μ-extended LR-mesh
  - When the LR B-splines is a basis for the spline space over the μ-extended LR-mesh

Ensuring linear independence

Spline space spanned by B-splines before refinement

We say that $(\mathcal{M}_{j+1}, \mu_{j+1}, \boldsymbol{p})$ goes hand-in-hand with $(\mathcal{M}_j, \mu_j, \boldsymbol{p})$ if

Spline space before refinement

- $\boxed{\text{span}\,(B)_{B \in \mathcal{B}_j}} = \boxed{\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_j, \mu_j)}$ and
- $\boxed{\text{span}\,(B)_{B \in \mathcal{B}_{j+1}}} = \boxed{\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1}, \mu_{j+1}).}$

Spline space spanned by B-splines after refinement

Spline space after refinement

If $(\mathcal{M}_{j+1}, \mu_{j+1}, \boldsymbol{p})$ and $(\mathcal{M}_{j+1}, \mu_{j+1}, \boldsymbol{p})$ goes hand-in-hand and

$$\boxed{\#\mathcal{B}_{j+1} = \dim \mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1}, \mu_{j+1})}$$

A basis if hand-in-hand and the number of B-splines matches the spline space dimenison

then the B-splines of $\mathcal{B}_{j+1}$ form a basis for $\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1}, \mu_{j+1})$.

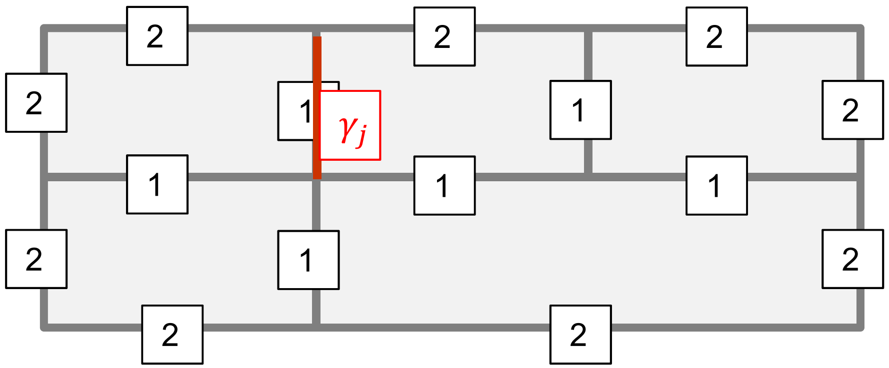
To ensure linear independence we have to

1. Determine $\dim \mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1}, \mu_{j+1})$
2. Determine if $\mathcal{B}_{j+1}$ spans $\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1}, \mu_{j+1})$
3. Check that $\#\mathcal{B}_{j+1} = \dim \mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1}, \mu_{j+1})$

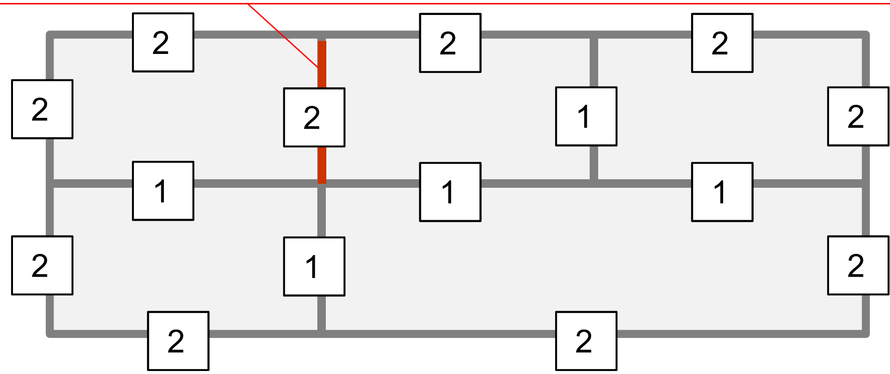
Difference in spanning properties between $\mathcal{B}_j$ and $\mathcal{B}_{j+1}$

- The only B-splines in $\mathcal{B}_{j+1}$ that model the discontinuity introduced by inserting the mesh-rectangle γ_j are those that have γ_j with multiplicity $\mu(\gamma_j)$ as part of the knot structure.
- By restricting these B-splines to γ_j we get a set of B-splines $\mathcal{B}_\gamma$ restricted to γ_j with dimension one lower than the dimension of the B-splines of $\mathcal{B}_{j+1}$.

Pick out the B-splines with multiplicity 2 over γ_j. Intersect with γ_j to select trimmed univariate B-splines $\mathcal{B}_\gamma$.



$(\mathcal{M}_j, \mu_j)$



$(\mathcal{M}_{j+1}, \mu_{j+1})$

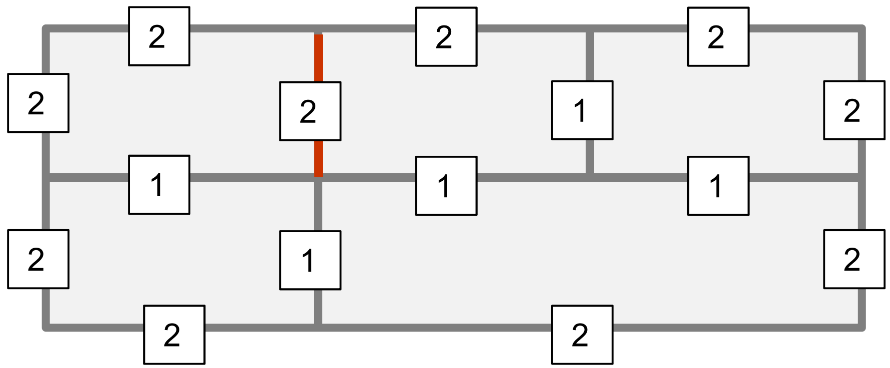
The use of $\mathcal{B}_\gamma$

- A theorem for general dimensions and degrees states

$$\dim \operatorname{span}\ (B_\gamma)_{B\in\mathcal{B}_\gamma} \le \dim\ \mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1},\mu_{j+1}) - \dim\ \mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_j,\mu_j)$$

- Further it is stated that $\mathcal{B}_{j+1}$ spans $\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1},\mu_{j+1})$ if

$$\dim \operatorname{span}\ (B_\gamma)_{B\in\mathcal{B}_\gamma} = \dim\ \mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1},\mu_{j+1}) - \dim\ \mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_j,\mu_j)$$



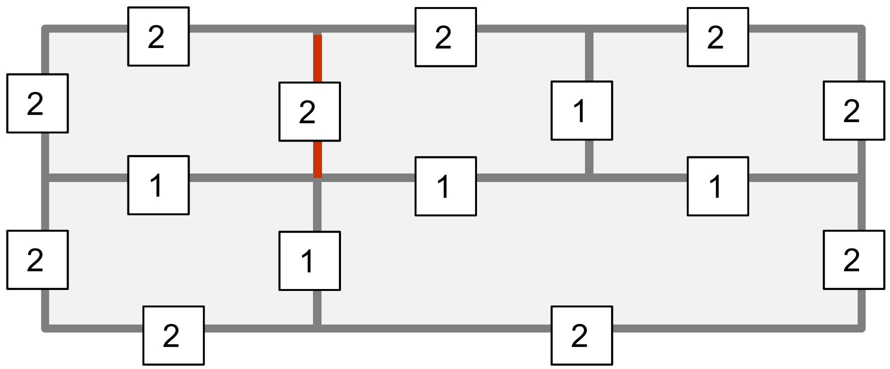
$(\mathcal{M}_{j+1},\mu_{j+1})$

We can find $\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_j,\mu_j)$ and $\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1},\mu_{j+1})$ using the dimension formula provided the homology terms are zero.

Checking the dimension of the space spanned by $\mathcal{B}_\gamma$ is a constructive tool to check if $\mathcal{B}_{j+1}$ spans the spline space required.

Observations

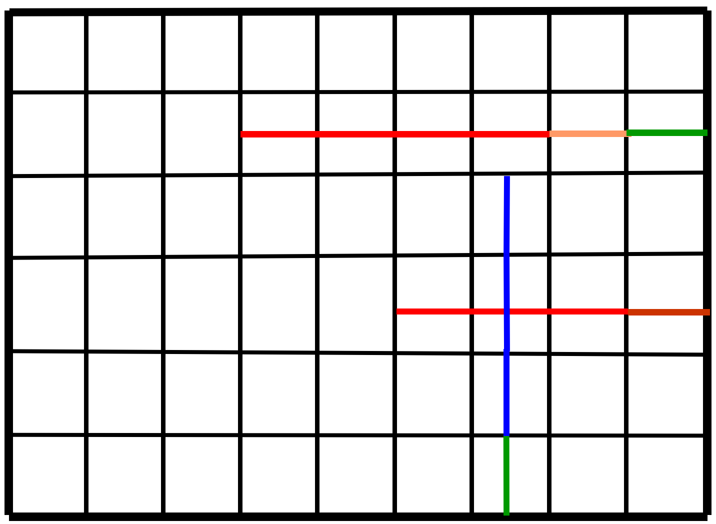
- To find the dimension of a spline space with many B-splines is more complex than finding the dimension of a spline space with few B-splines
- When assessing the B-splines $\mathcal{B}_\gamma$ over γ_j we see if the refinement can be broken into a sequence of LR B-spline refinements with as low dimension increase as possible.
  - As a legal LR-spline refinement always introduces at least one B-spline linearly independent from the pre-existing, a dimension increase by just one will ensure that we go hand-in-hand.
  - If the dimension increase is greater than 1 we have to resort to assessing the B-splines $\mathcal{B}_\gamma$ over γ_j.



Example: C^2 bi-cubic refinement configurations

Dimension increase of spline space over the box-partition

- All boundary knots mesh-rectangles have multiplicity 4
- All interior mesh-rectangles have multiplicity 1



Mesh-rectangle length 1 starting at the boundary, T-joint at other end. Dimension increase 1. Trivial

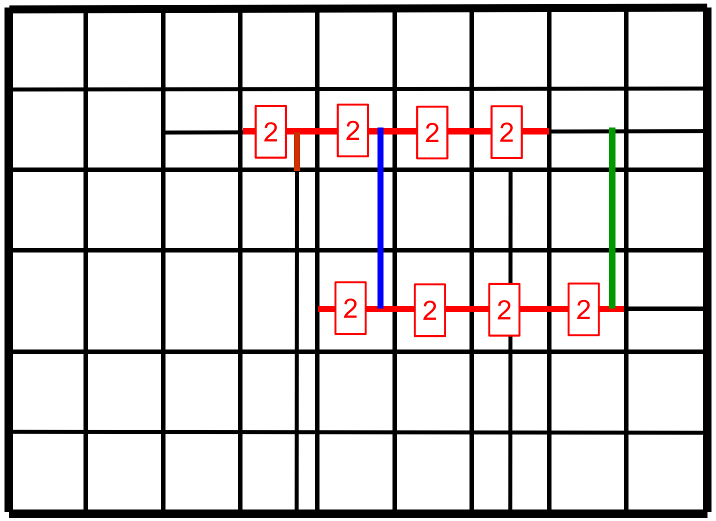
Mesh-rectangle length 1 extending existing mesh-rectangle, T-joint at other end. Dimension increase 1. Trivial.

Interior mesh-rectangle length 4, T-joints at both ends. Dimension increase 1. Trivial.

Mesh-rectangle length 1 gap filling. Dimension increase 4, $\mathcal{B}_\gamma$ spans a polynomial space, Trivial to check

Mesh-rectangle length 1 extension of existing mesh-rectangle to the boundary. Dimension increase 4, $\mathcal{B}_\gamma$ spans a polynomial space, Trivial to check.

Increasing interior multiplicity in the bi-cubic case



Interior mesh-rectangle length 4, increase multiplicity to 2, lower multiplicity at both ends, dimension increase 1. Trivial.

Interior mesh-rectangle length 3, ending in T-joints with orthogonal mesh rectangles, one with multiplicity 1,and one with multiplicity 2, dimension increase 1. Trivial.

Extend existing mesh by length 1, ending in T-joint with orthogonal mesh rectangles with multiplicity 2, dimension increase 2, $\mathcal{B}_\gamma$ spans a polynomial space. Trivial to check.

Interior mesh-rectangle length 3, ending in T-joints with orthogonal mesh rectangles of multiplicity 2, dimension increase 2, two new B-splines. To decide if $\mathcal{B}_{j+1}$ is a basis check if $\dim \operatorname{span} \left(B_\gamma\right)_{B\in\mathcal{B}_\gamma} = 2$.

C^2 bi-cubic refinement configurations Cases: Dimension increase 1

The start point is a bi-cubic tensor product B-spline basis spanning the spline space over a tensor-mesh.

Assume that before the refinement that the B-splines in $\mathcal{B}_j$ are linear independent and span $\mathbb{S}_{(3,3)}(\mathcal{M}_j, \mu_j)$

- Assume that a B-spline in $\mathcal{B}_{j+1}$ has a knotline containing the new mesh rectangle γ_j. This B-spline will be linearly independent from the B-splines in $\mathcal{B}_j$. Consequently the whole spline space defined over $\mathbb{S}_{(3,3)}(\mathcal{M}_{j+1}, \mu_{j+1})$ is spanned by $\mathcal{B}_{j+1}$.
- If the number of B-splines in $\mathcal{B}_{j+1}$ corresponds to the dimension of the bi-cubic spline space over $\mathcal{M}_{j+1}$ then the B-splines in $\mathcal{B}_{j+1}$ are linearly independent.

C^2 bi-cubic refinement configurations Cases: Dimension increase 4

Questions:

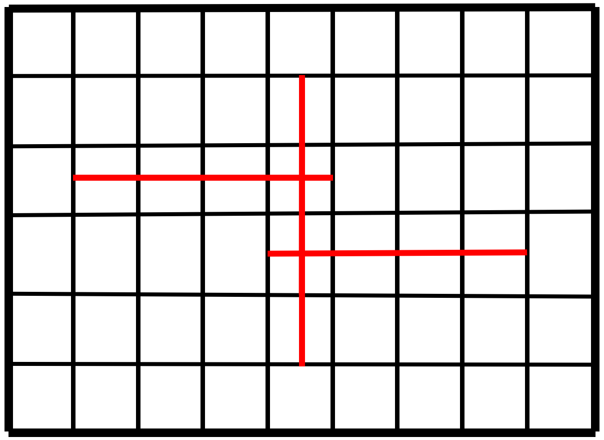
1. Do the B-splines in $\mathcal{B}_{j+1}$ span $\mathbb{S}_{(3,3)}(\mathcal{M}_j, \mu_j)$?
2. Is $\mathcal{B}_{j+1}$ a basis for $\mathbb{S}_{(3,3)}(\mathcal{M}_j, \mu_j)$?

If we can answer yes to question 1, and the number of B-splines in $\mathcal{B}_{j+1}$ corresponds to the dimension of $\mathbb{S}_{(3,3)}(\mathcal{M}_j, \mu_j)$ then $\mathcal{B}_{j+1}$ is a basis for $\mathbb{S}_{(3,3)}(\mathcal{M}_j, \mu_j)$.

C^2 bi-cubic refinement Cases: Dimension increase 4

- We have to determine if $\dim \operatorname{span} \mathcal{B}_{\gamma} = 4$ or less than 4.
  - If the B-spline have a structure known for univariate B-splines, trivial to check.
  - If a more complex b-spline configuration, perform knot insertion such that the knot multiplicity at both ends of γ is 4, e.g., convert to a Bernstein basis. Check if the rank of the knot insertion matrix is 4.

Possible to increase dimension without refining B-splines

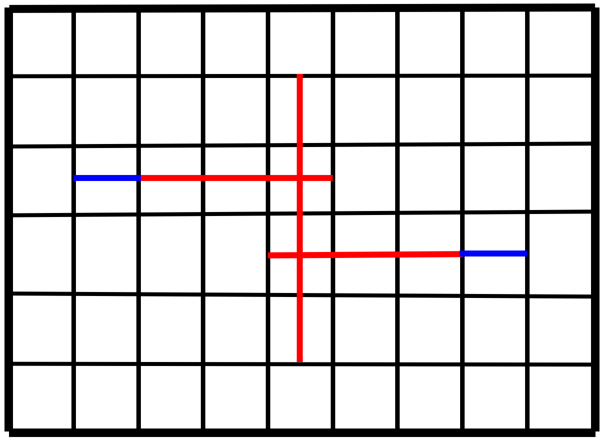


Dimension increase 1, one new B-splines (+5, -4)

Dimension increase 1, one new B-splines (+5, -4)

Dimension increase 1, no new B-splines
Dimension increase 3, three new B-splines (+ 9, -6)

- To decide if $\mathcal{B}_{j+1}$ is a basis check if $\dim \operatorname{span} \left(B_\gamma\right)_{B\in\mathcal{B}_\gamma} = 3.$



Alternative refinement sequence

Dimension increase 1, one new B-spline (+5, -4)

Dimension increase 1, one new B-spline (+2, -1)

Dimension increase 1, one new B-spline (+2, -1)

Dimension increase 1, one new B-spline (+5, -4)

Dimension increase 1, one new B-spline (+5, -4)

How to guarantee that $\mathcal{B}_{j+1}$ is a basis for $\left(\mathcal{M}_{j+1}, \mu_{j+1}\right)$ in the general case?

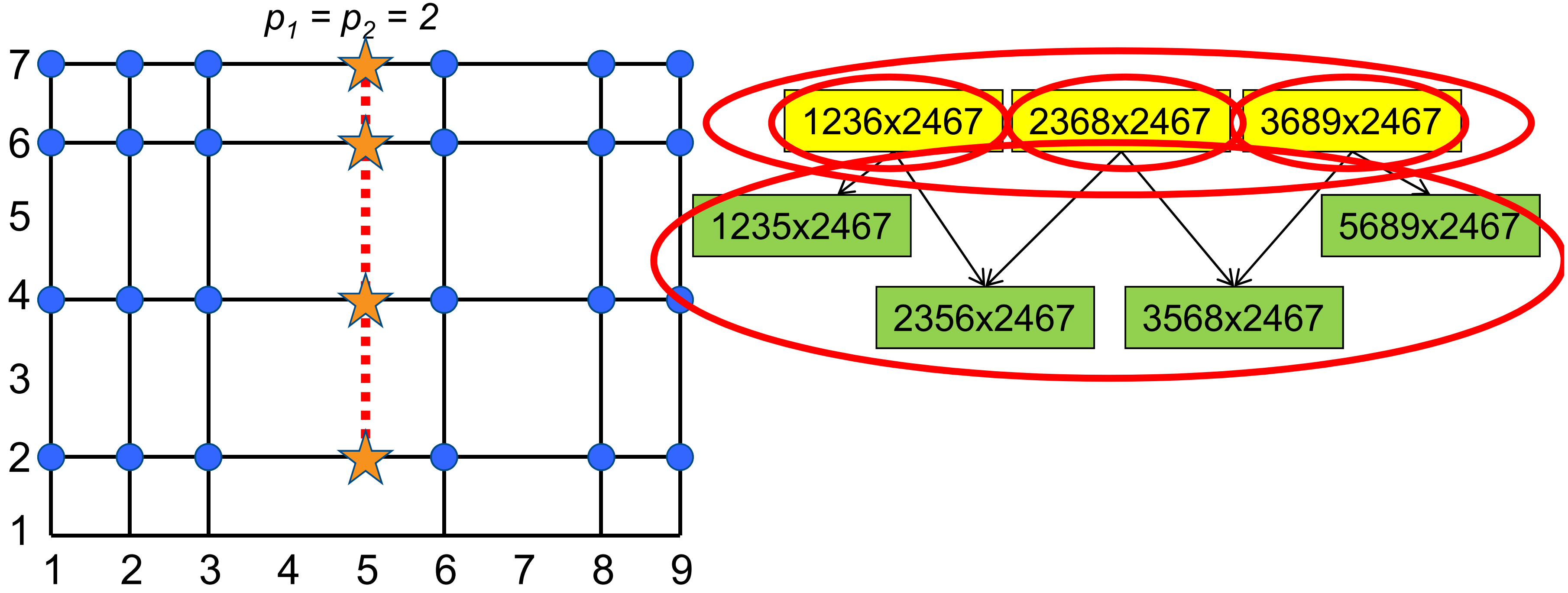
- Assume that $\mathcal{B}_j$ is a basis for $\mathbb{S}_{\boldsymbol{p}}\left(\mathcal{M}_j, \mu_j\right)$, .
- Make $\left(\mathcal{M}_{j+1}, \mu_{j+1}\right) = \left(\mathcal{M}_j + \gamma_j, \mu_{j,\gamma_j}\right)$
- $\mathcal{B}_{j+1}$ is a basis for $\mathbb{S}_{\boldsymbol{p}}\left(\mathcal{M}_{j+1}, \mu_{j+1}\right)$ if
  - The B-splines of $\mathcal{B}_{j+1}$ spans $\mathbb{S}_{\boldsymbol{p}}\left(\mathcal{M}_j, \mu_j\right)$ **(Goes hand in hand)**
  - $\#\mathcal{B}_{j+1} = dim\ \ \mathbb{S}_{\boldsymbol{p}}\left(\mathcal{M}_{j+1}, \mu_{j+1}\right)$
    - The number of B-splines corresponds to the dimension of $\mathbb{S}_{\boldsymbol{p}}\left(\mathcal{M}_{j+1}, \mu_{j+1}\right)$

How to determine if the collection of B-splines goes hand in hand with the spline space?

- The study of when two μ-extended meshes go hand-in-hand is simplified by considering the restriction $\mathcal{B}_\gamma$ of a B-spline $\mathcal{B}$ to a mesh-rectangle γ.
  - In the 2-variate case we can look at the B-splines of $\mathcal{B}_{j+1}$ that have γ with multiplicity $\mu(\gamma)$ as a knotline, and determine the dimension of the univarate spline space spanned by $\mathcal{B}_\gamma$
    $\dim \operatorname{span} \left(\mathcal{B}_\gamma\right)_{\mathcal{B}\in\mathcal{B}_\gamma}$
  - If $\dim \operatorname{span} \left(B_\gamma\right)_{B\in\mathcal{B}_\gamma} = \dim \mathbb{S}_{\boldsymbol{p}}\left(\mathcal{M}_{j+1},\mu_{j+1}\right) - \dim \mathbb{S}_{\boldsymbol{p}}\left(\mathcal{M}_j,\mu_j\right)$
    then the spline space go hand in hand

Linear dependency example

= lost RMSBF = new RMSBF lin. indep. ⇔ (#) = (#) + 1



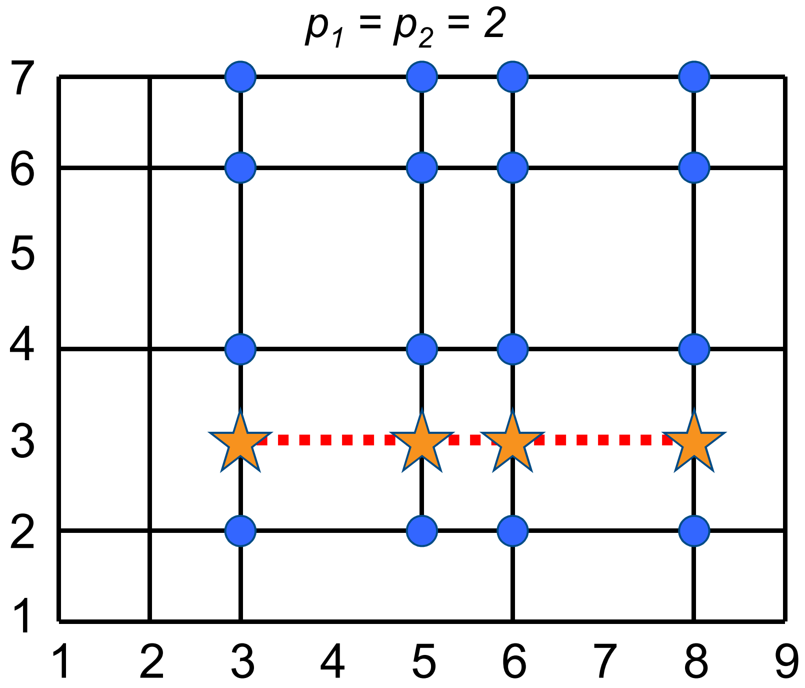
= 3 = 4 → Linear independecy

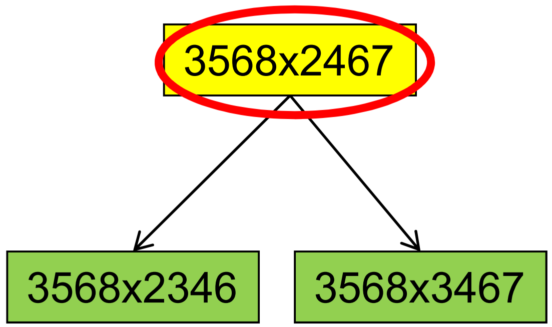
Example by: Kjell Fredrik Pettersen, SINTEF

Linear dependency example

= lost RMSBF = new RMSBF

lin. indep. ⇔ (#) = (#) + 1

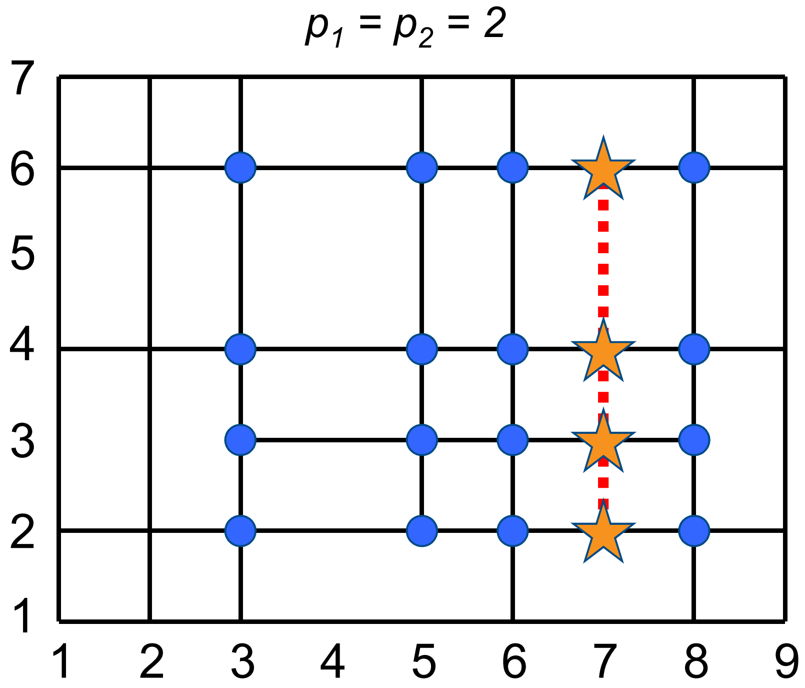


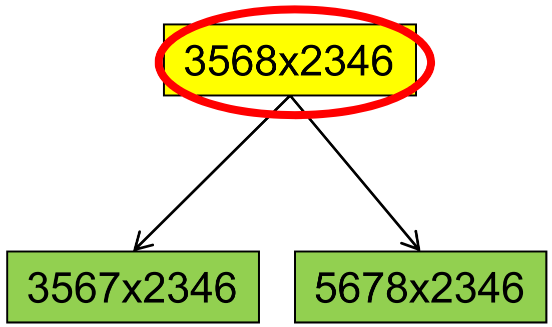


= 1 = 2 → Linear independecy

Linear dependency example

= lost RMSBF = new RMSBF lin. indep. ⇔ (#) = (#) + 1



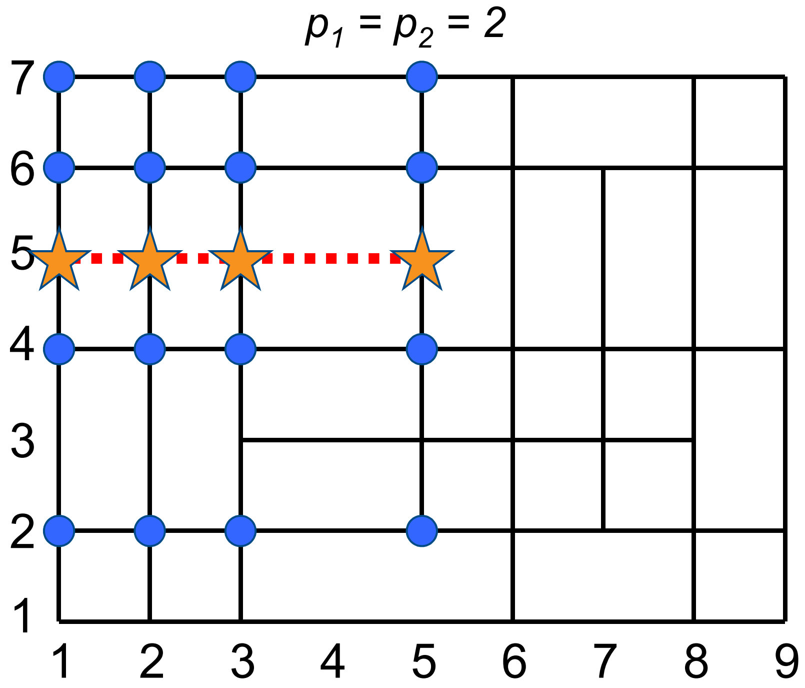


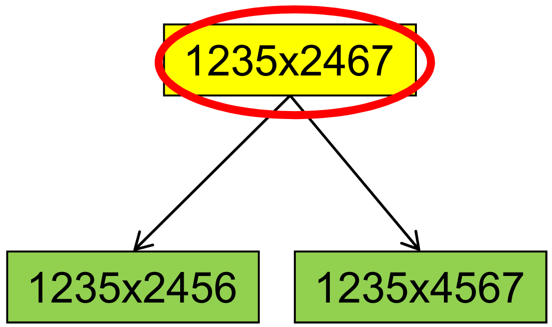
= 1 = 2 Linear independecy

Linear dependency example

= lost RMSBF = new RMSBF

lin. indep. ⇔ (#) = (#) + 1





= 1 = 2

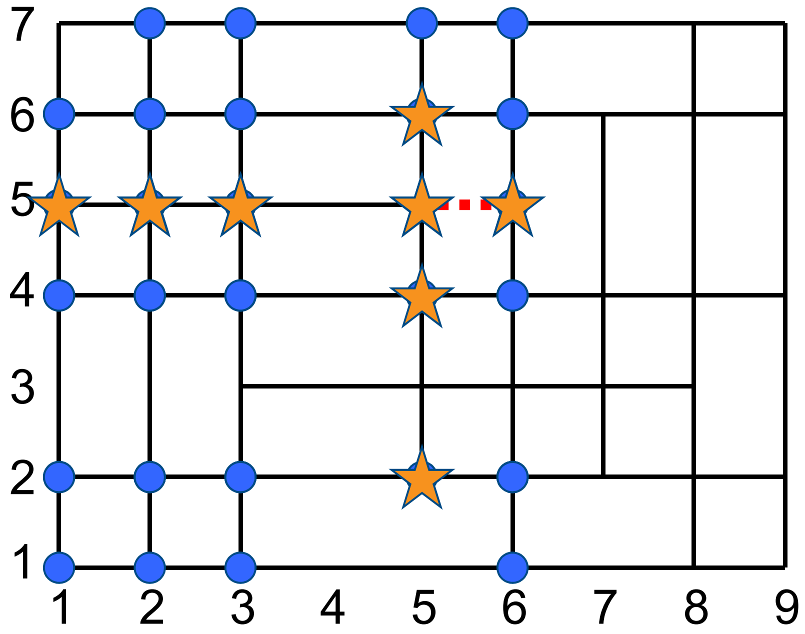
Linear independecy

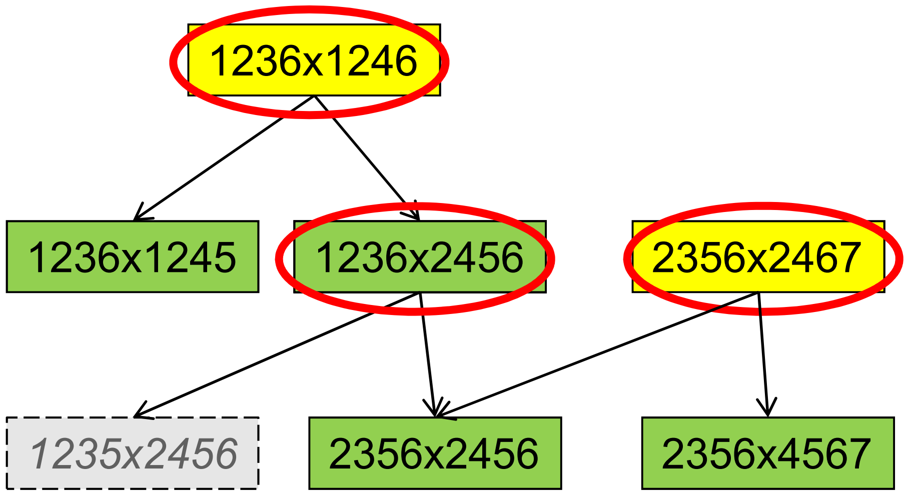
Linear dependency example

[yellow] = lost RMSBF [green] = new RMSBF

lin. indep. ⇔ (# [green]) = (# [yellow]) + 1

$p_1 = p_2 = 2$

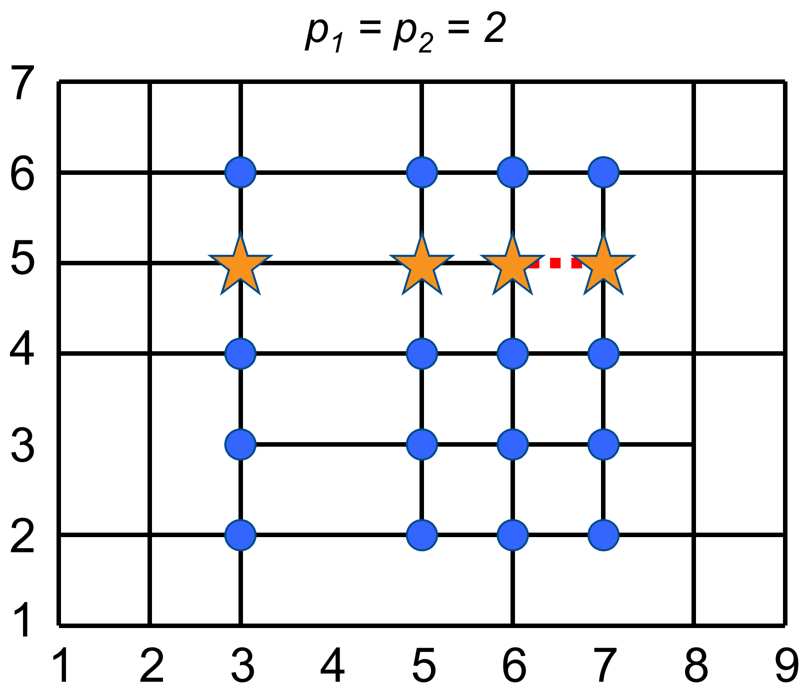


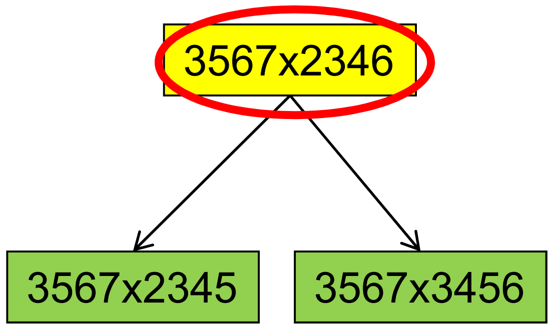


[yellow] = 2 [green] = 3 → Linear independecy

Linear dependency example

= lost RMSBF = new RMSBF lin. indep. ⇔ (#) = (#) + 1

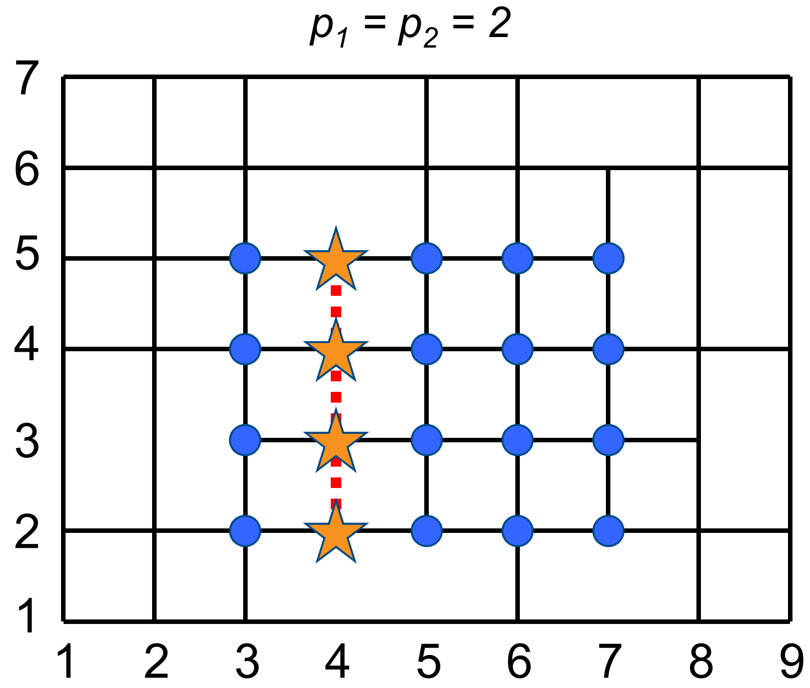


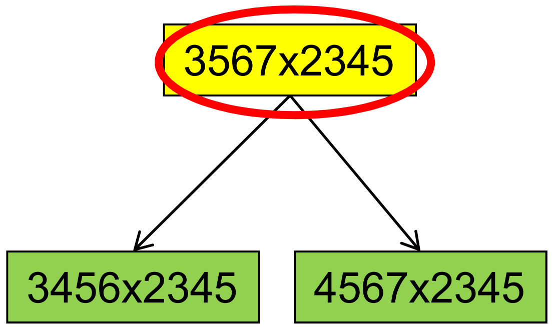


= 1 = 2 → Linear independecy

Linear dependency example

= lost RMSBF = new RMSBF lin. indep. ⇔ (#) = (#) + 1

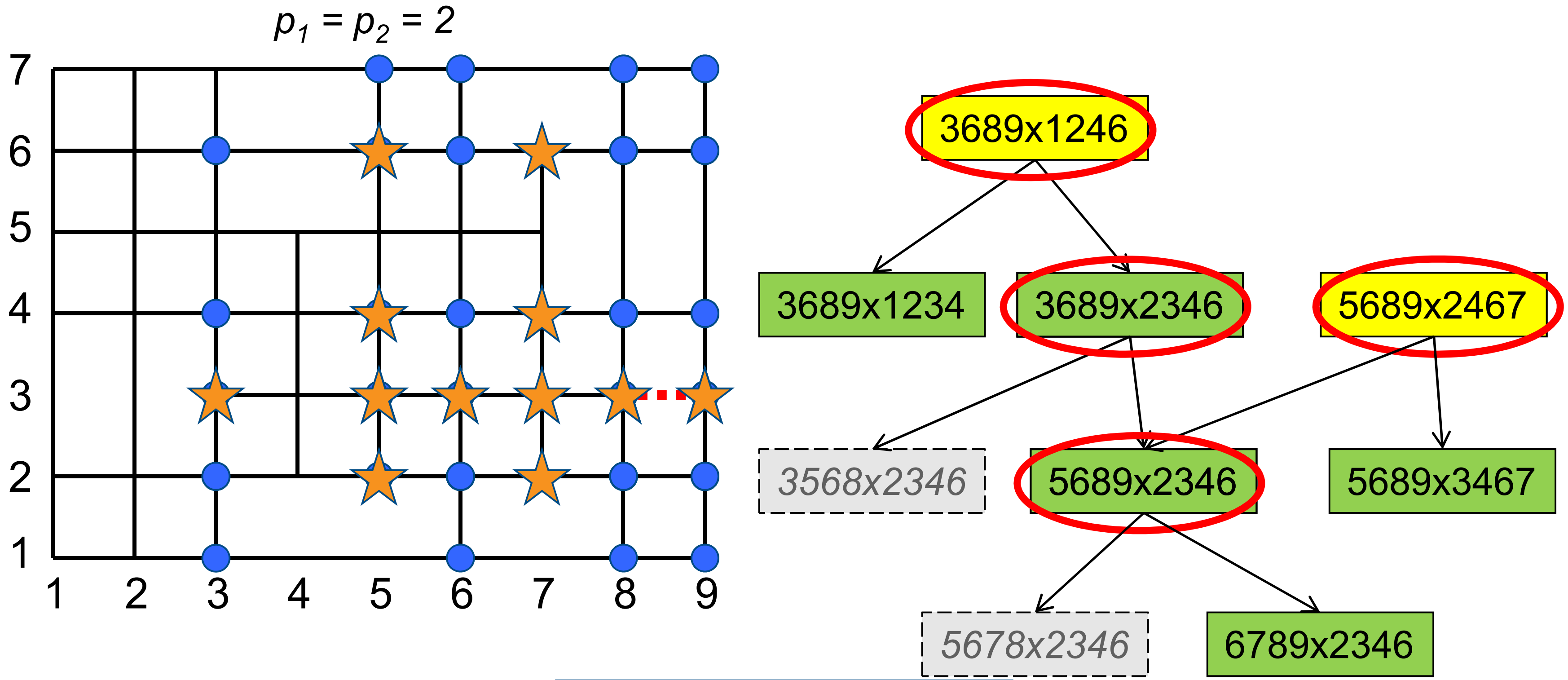




= 1 = 2 Linear independecy

Linear dependency example

= lost RMSBF = new RMSBF lin. indep. ⇔ (#) = (#) + 1



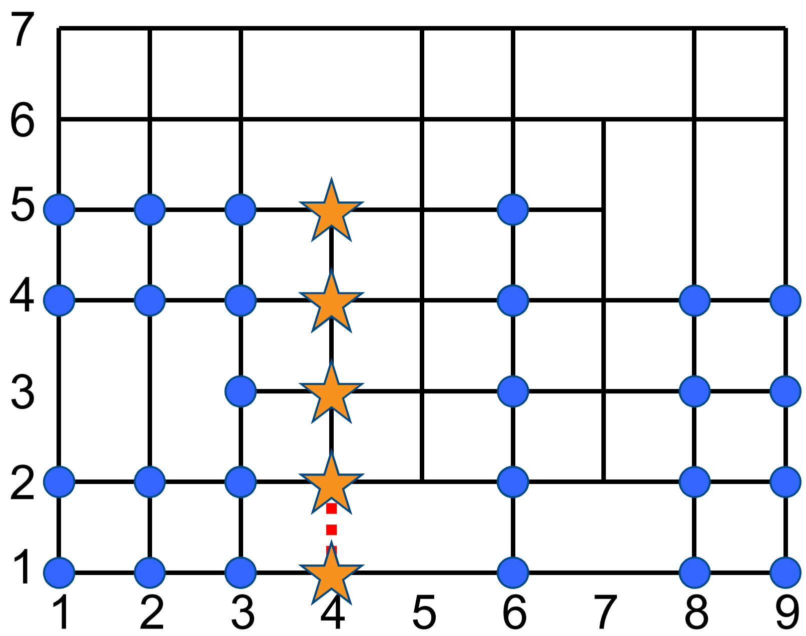
= 2 = 3

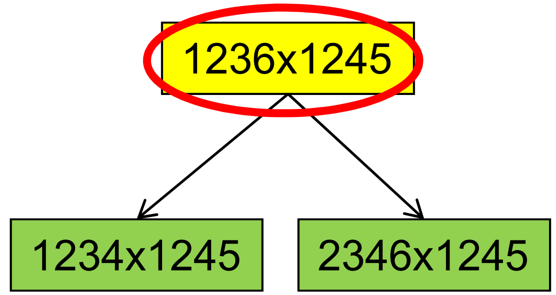
Linear independecy

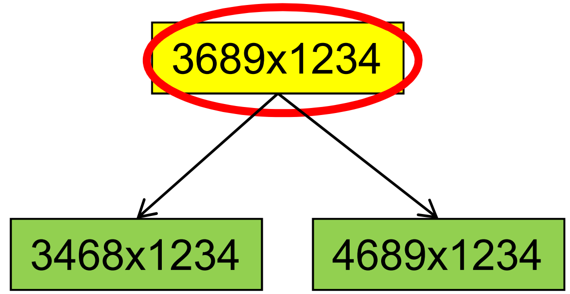
Linear dependency example

= lost RMSBF = new RMSBF lin. indep. ⇔ (#) = (#) + 1

$p_1 = p_2 = 2$



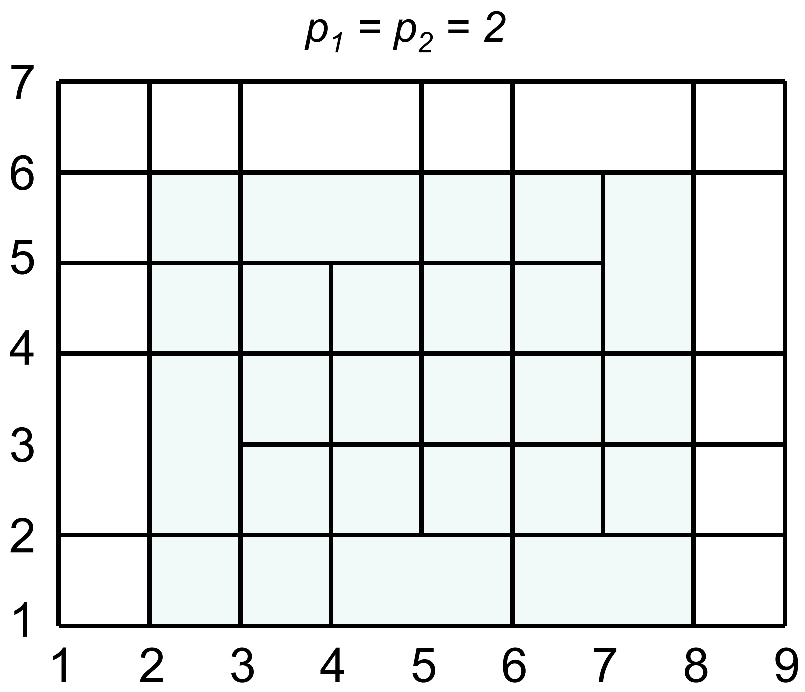




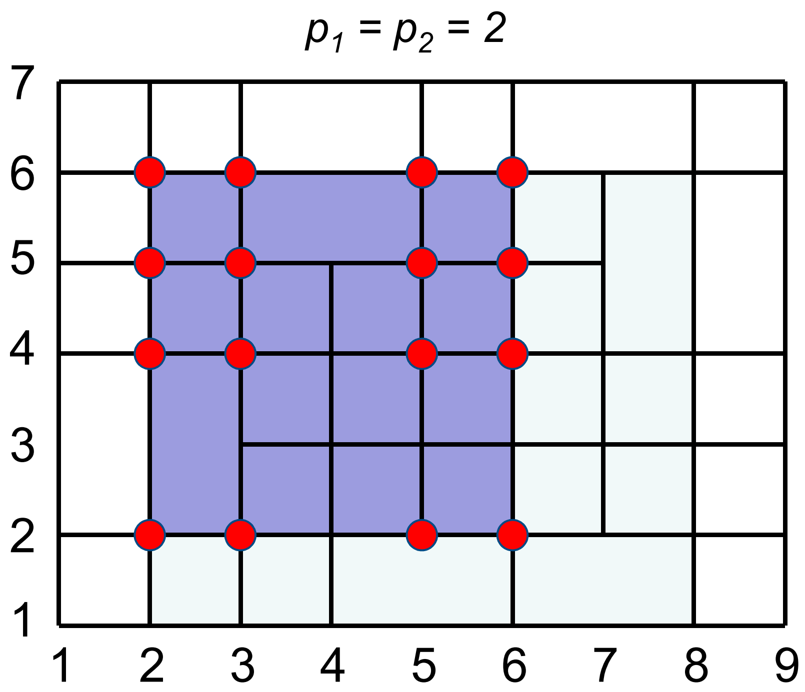
= 2 = 4

Linear dependecy!!!

Linear dependency example

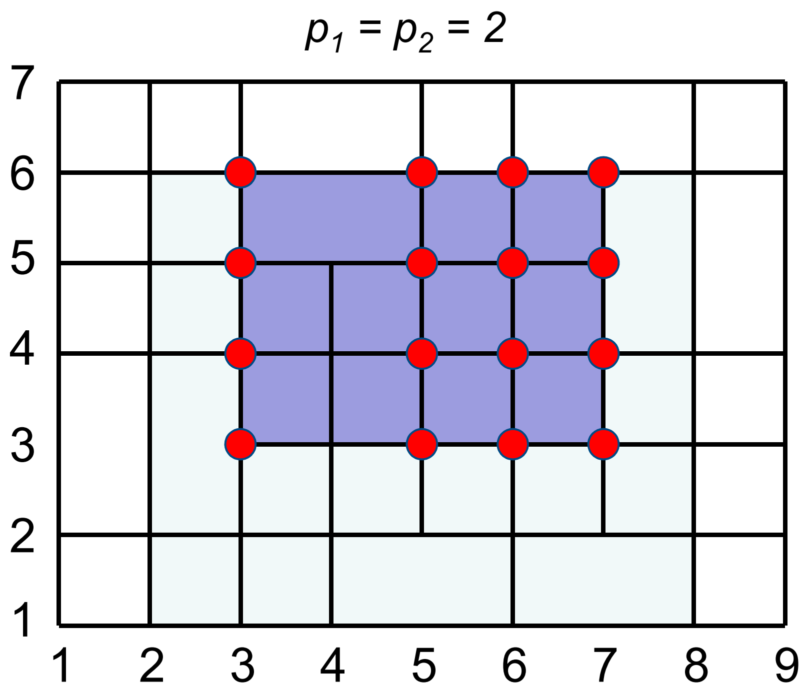


Linear dependency example



2356x2456

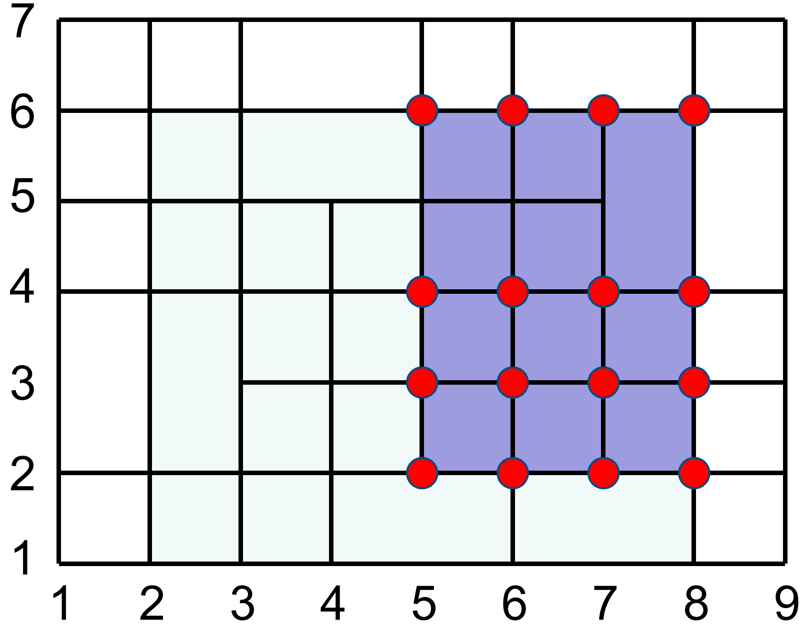
Linear dependency example



3567x3456

Linear dependency example

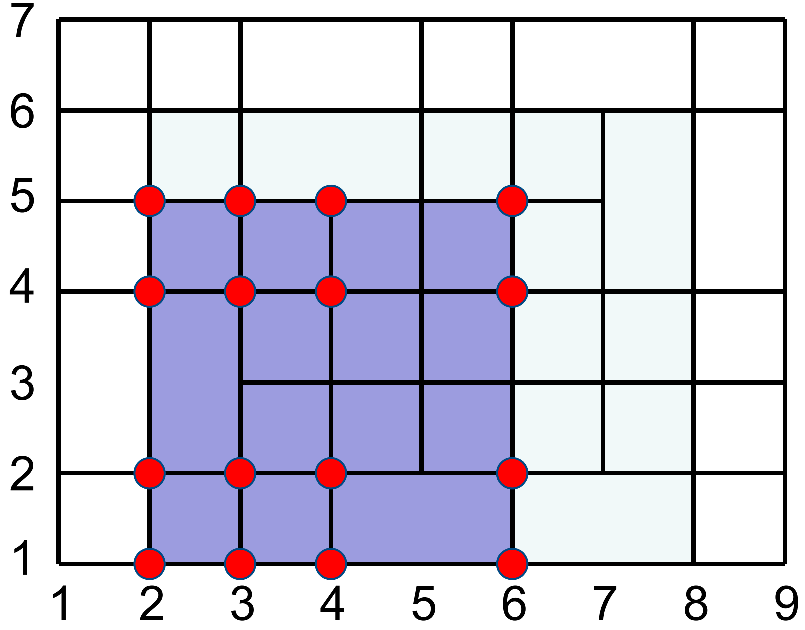
$p_1 = p_2 = 2$



5678x2346

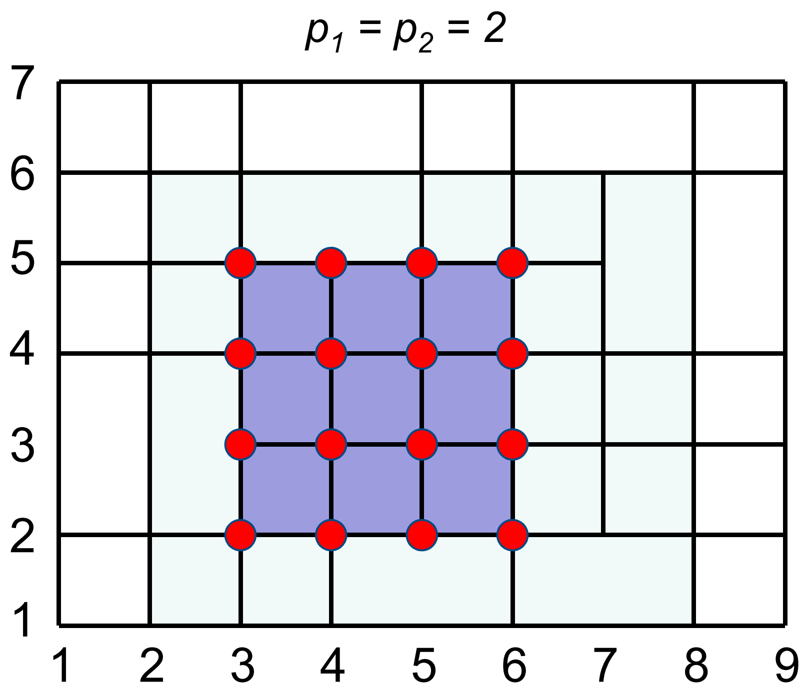
Linear dependency example

$p_1 = p_2 = 2$



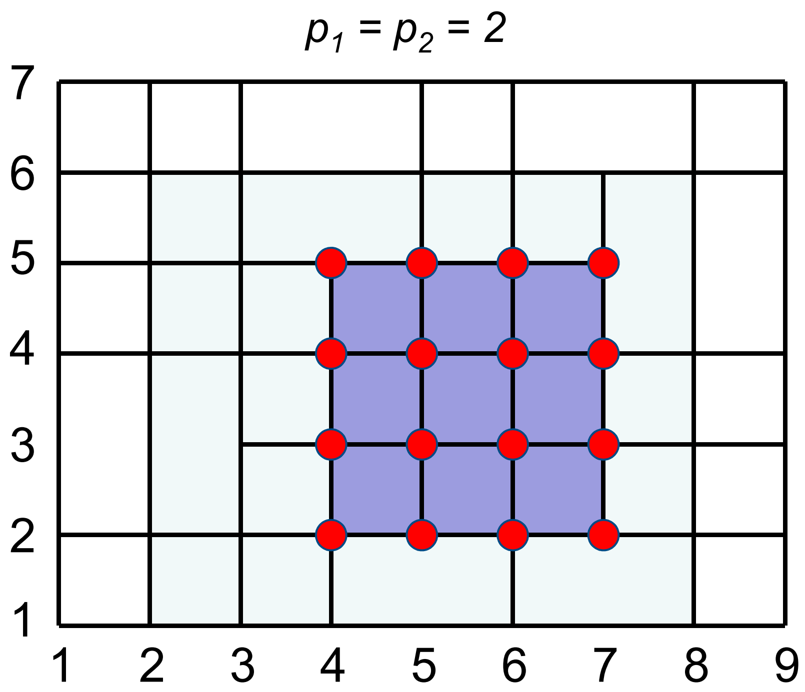
2346x1245

Linear dependency example



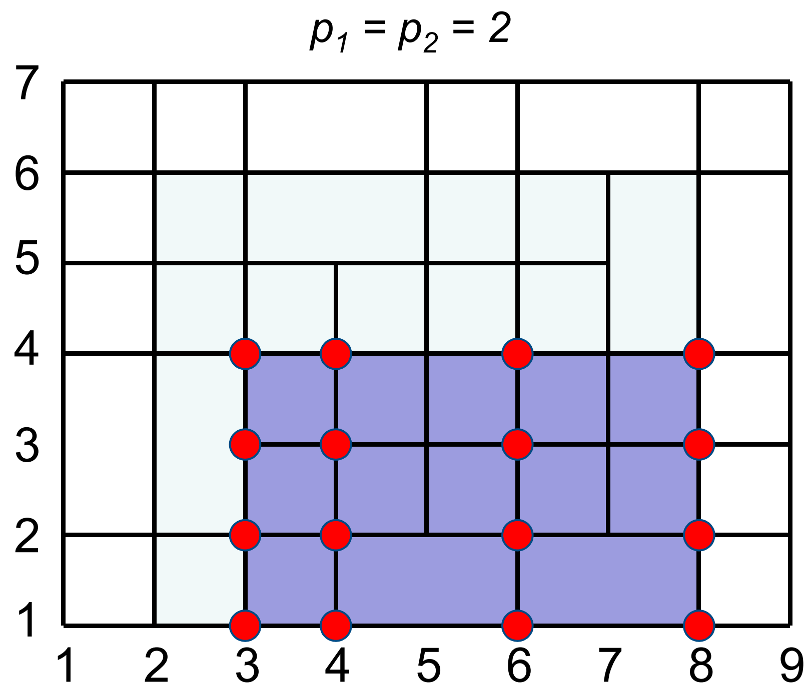
3456x2345

Linear dependency example



4567x2345

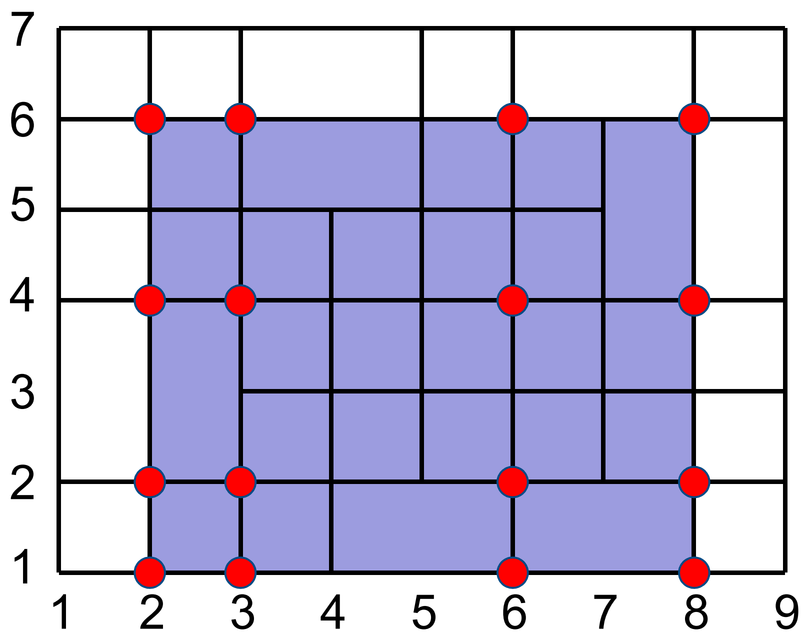
Linear dependency example



3468x1234

Linear dependency example

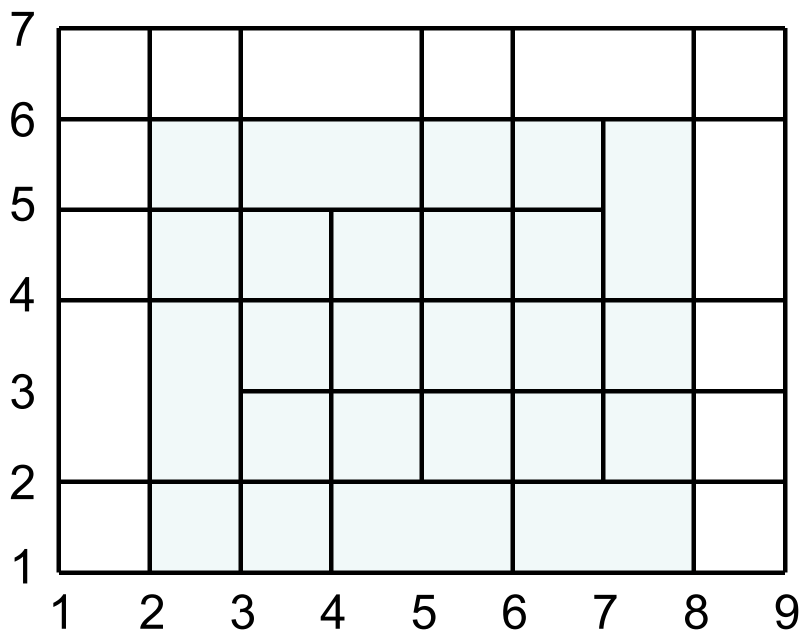
$p_1 = p_2 = 2$



2368x1246

Linear dependency example

$p_1 = p_2 = 2$



Linear relation

(knot value = knot position)

$$
\begin{aligned}
& 108 \cdot (5678)\text{x}(2346) \\
+ & 135 \cdot (2356)\text{x}(2456) \\
+ & 108 \cdot (3567)\text{x}(3456) \\
+ & 268 \cdot (3456)\text{x}(2345) \\
+ & 324 \cdot (4567)\text{x}(2345) \\
+ & 360 \cdot (2346)\text{x}(1245) \\
+ & 384 \cdot (3468)\text{x}(1234) \\
\\
= & 720 \cdot (2368)\text{x}(1246)
\end{aligned}
$$

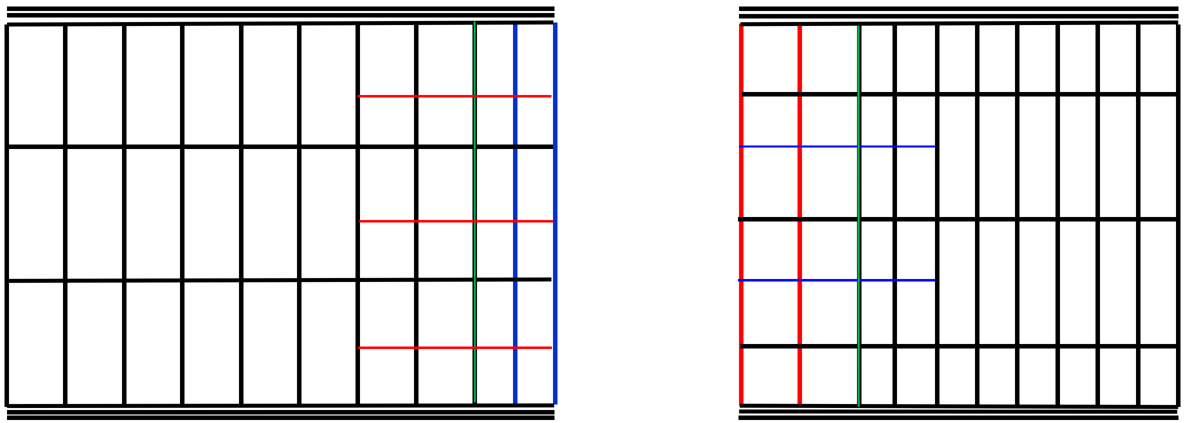
What to do to handle the situation when we produce too many B-splines to have a basis?

- We can eliminate one of the B-splines
  - We may end up with a collection of scaled B-splines that are only a partition of unity, but not a not a positive partition of unit.
  - Discard elimination strategy if the result is not a positive partiton of unity.
- Discard the problematic refinement and perform an alternative refinement close by.
- We perform additional refinements to solve the problem.

Some examples of use of LR B-splines

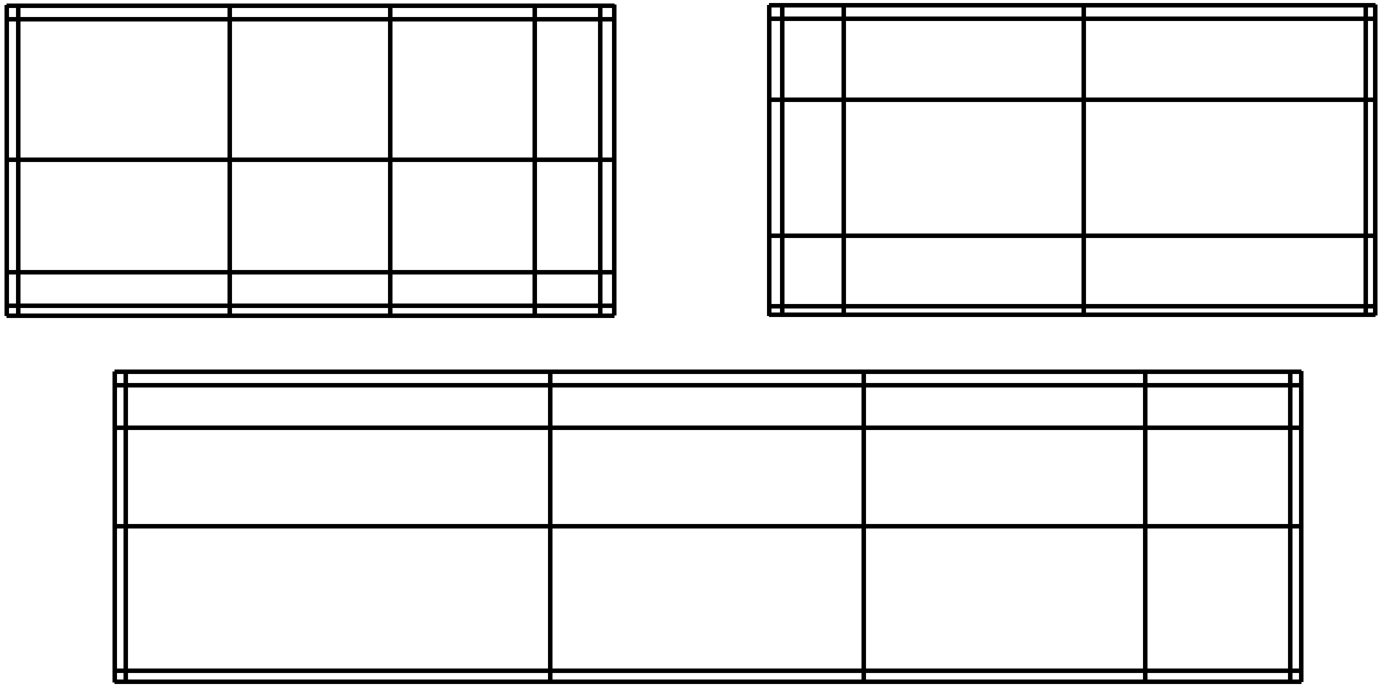
- Stitching of B-spline patches
- Approximation of large data sets

C^1 Stiching of 2-variate B-splines Bi-quadratic case using LR B-splines

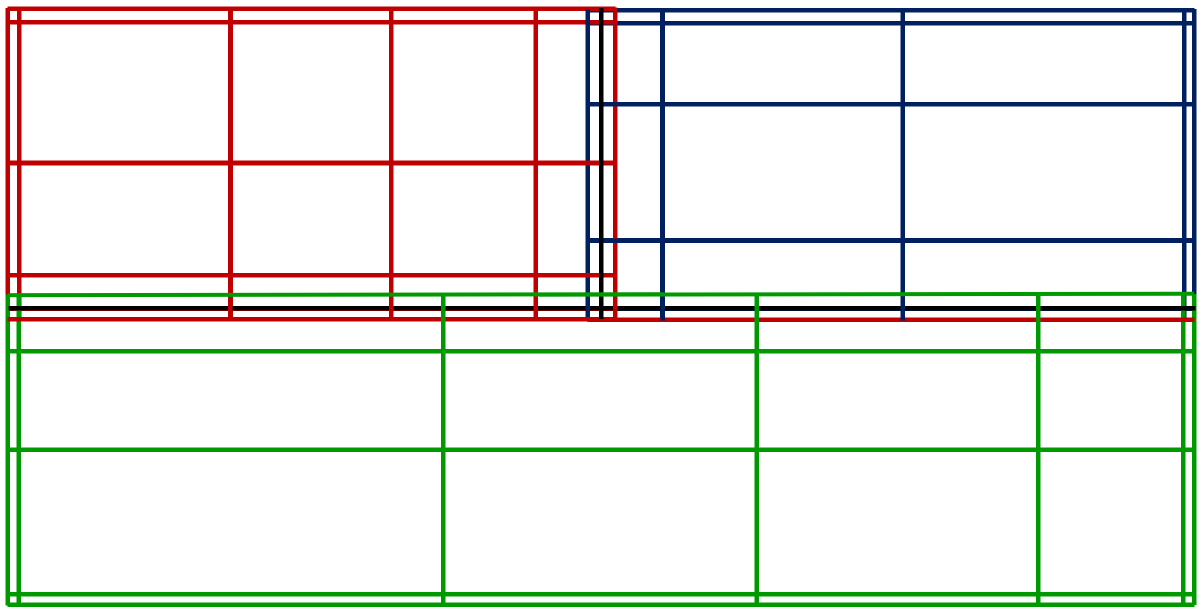


1. Adapt the edge knotlines of A to B
2. Adapt the edge knotlines of B to A
3. Insert horisontal knotline segement from B in A
4. Insert horisontal knotline segement from A in B
5. Merge the parameter domains

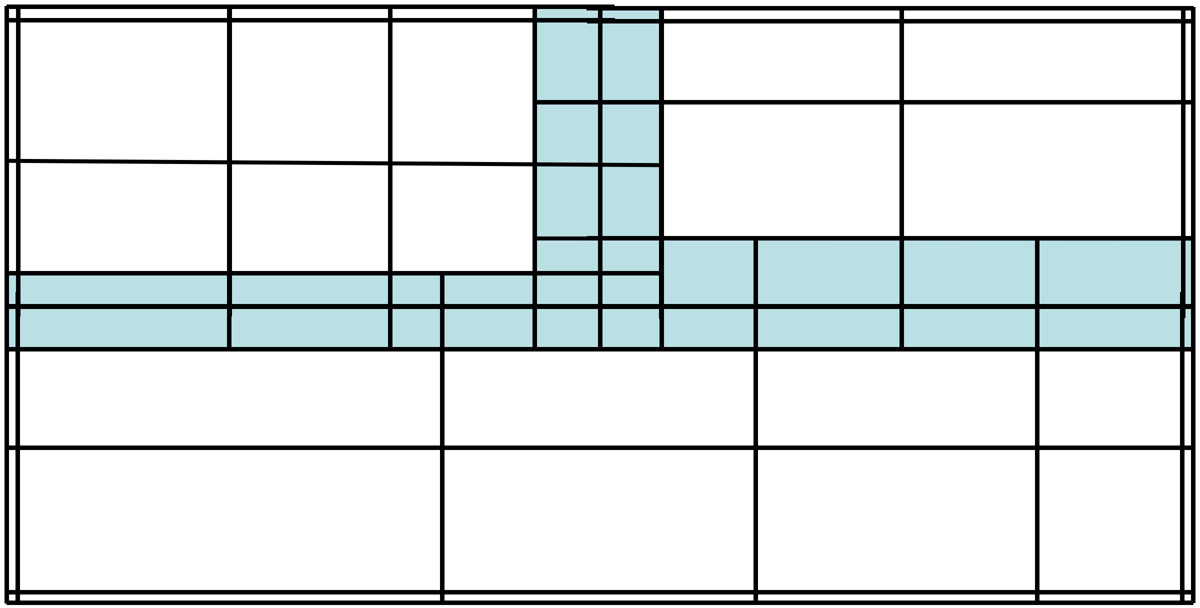
Multi-block T-joints (1) match parametrizaton



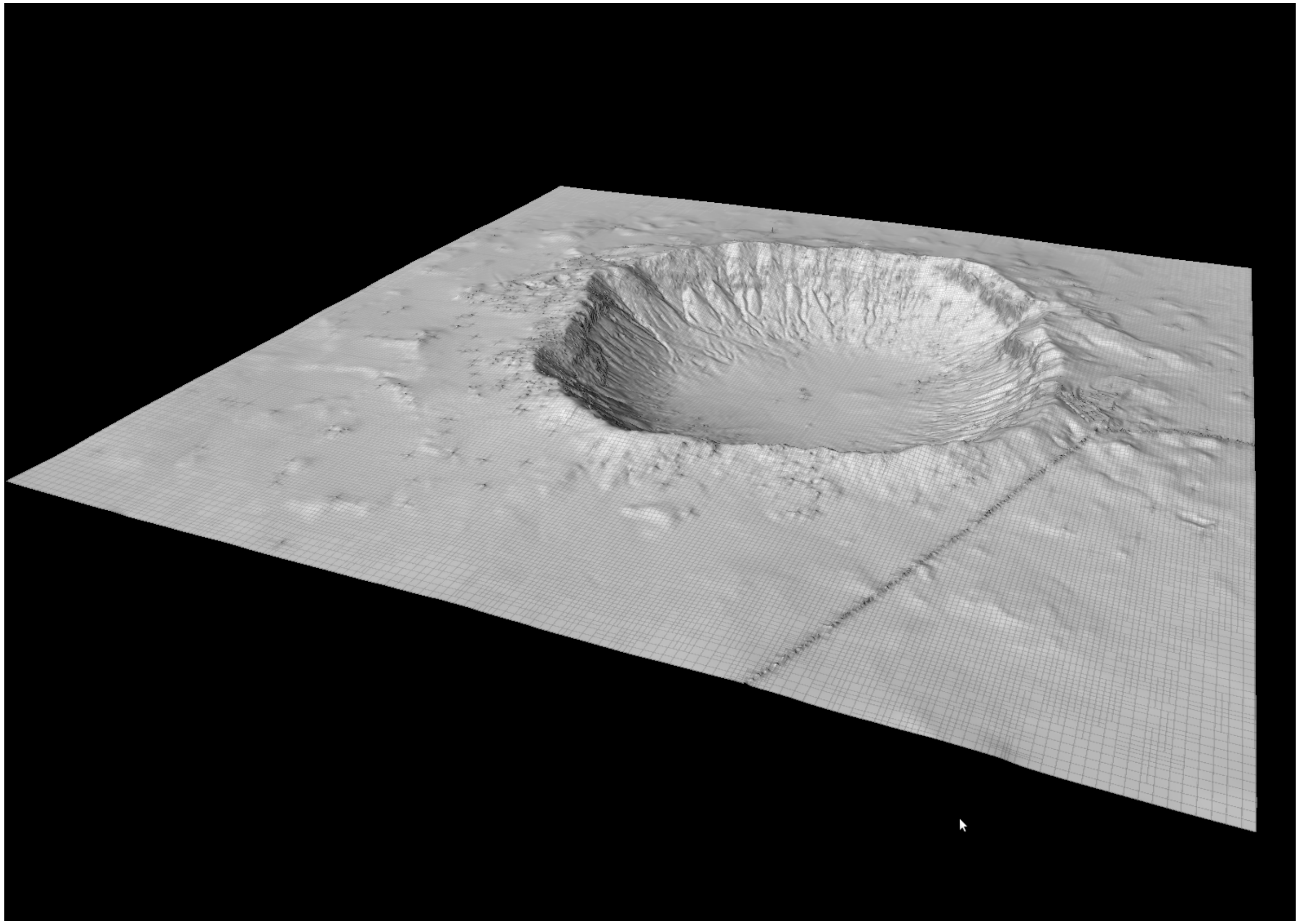
Multi-block T-joints(2) adjust boundary knotlines



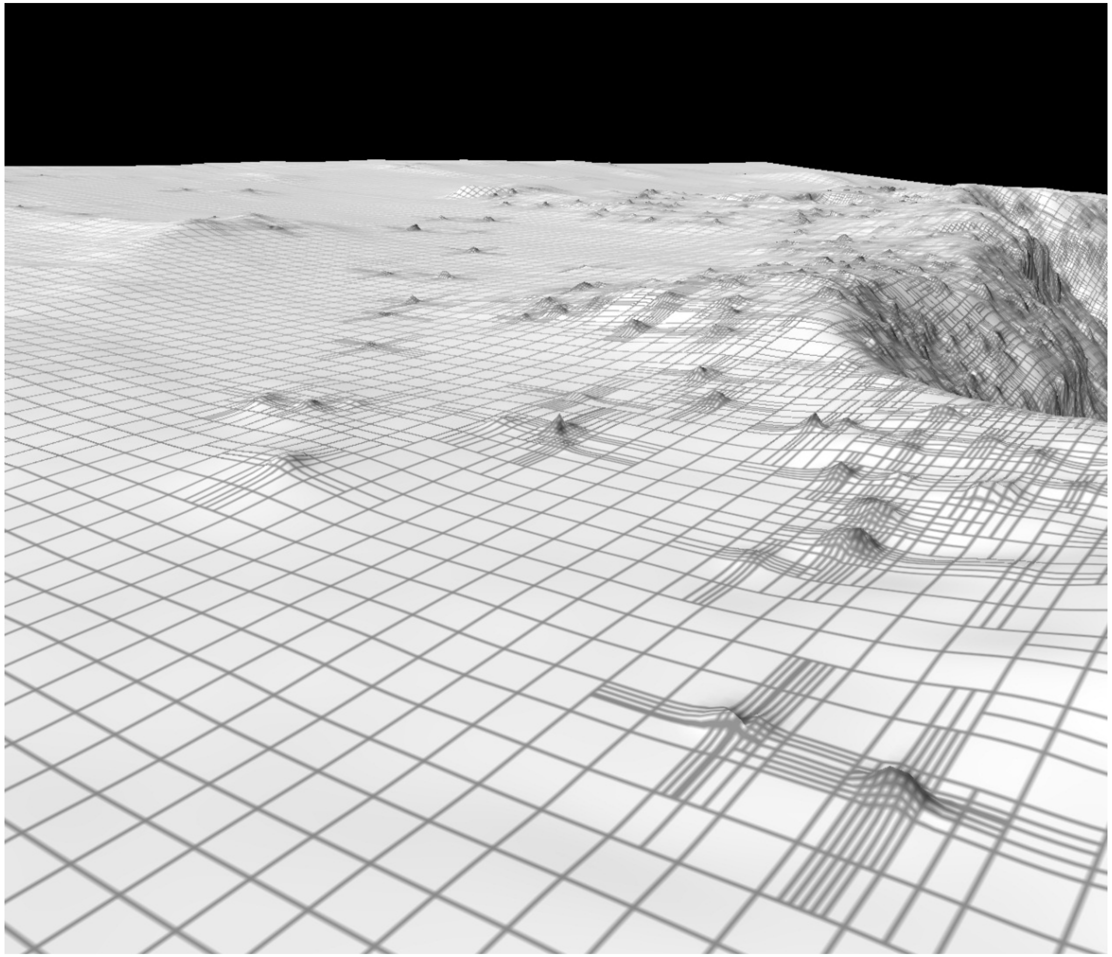
Multi-block T-joints(3)
identify + split transition B-splines



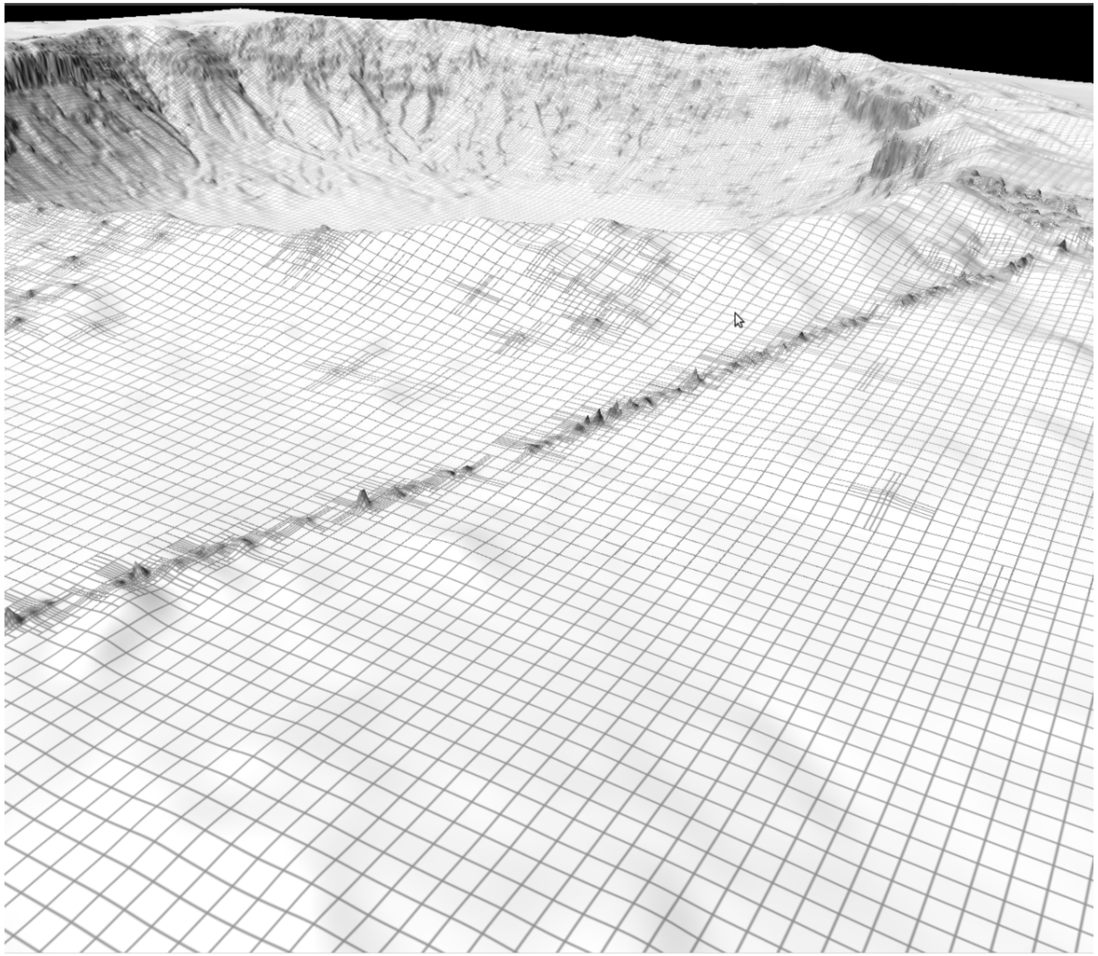
Approximation of large data set Barringer crater Arizona



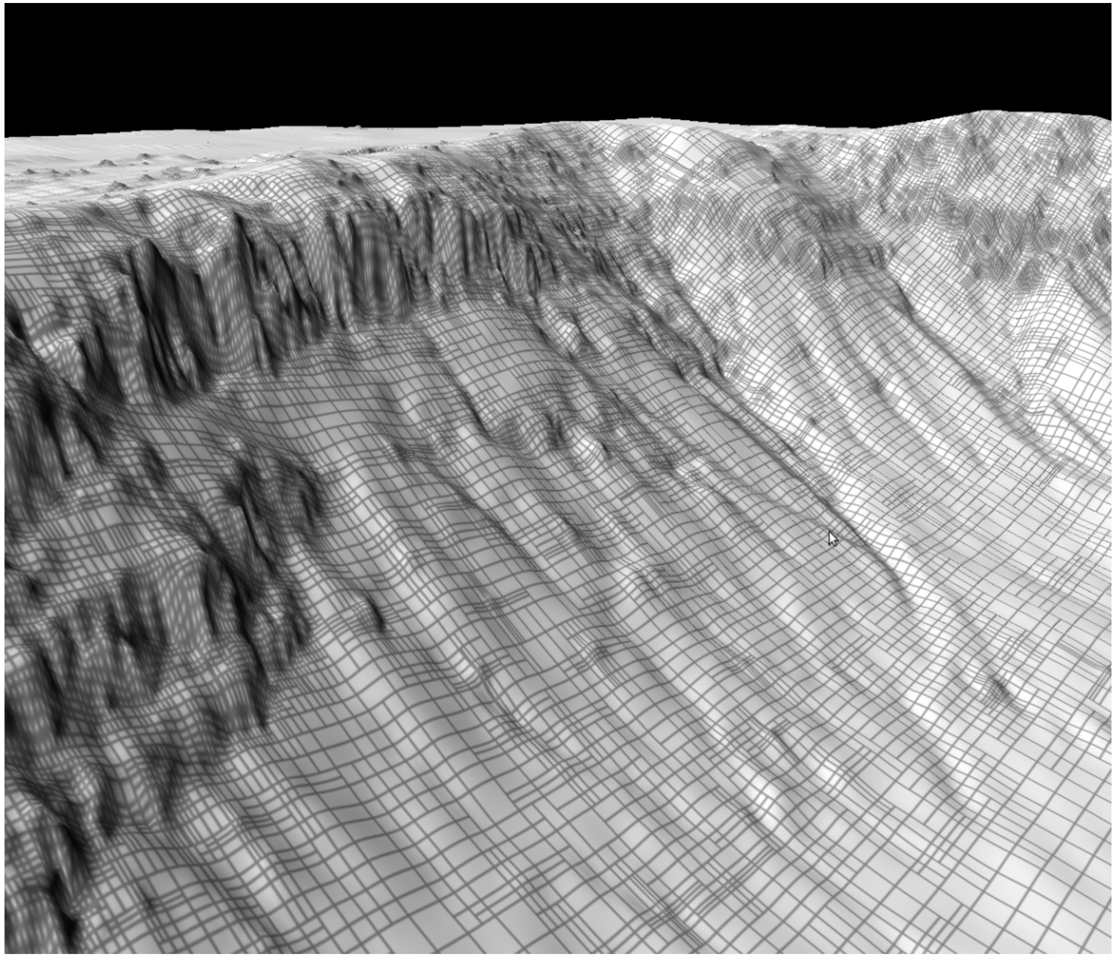
Local refinement to adapt to fine details



Data along powerline? reproduced



Details along inside slope



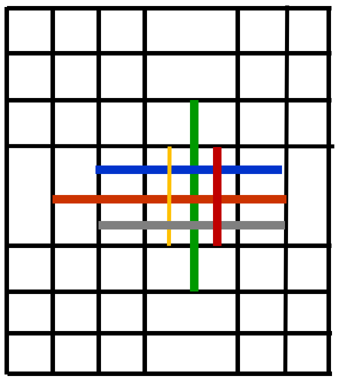
T-spline vertex grid as interface to LR B-splines

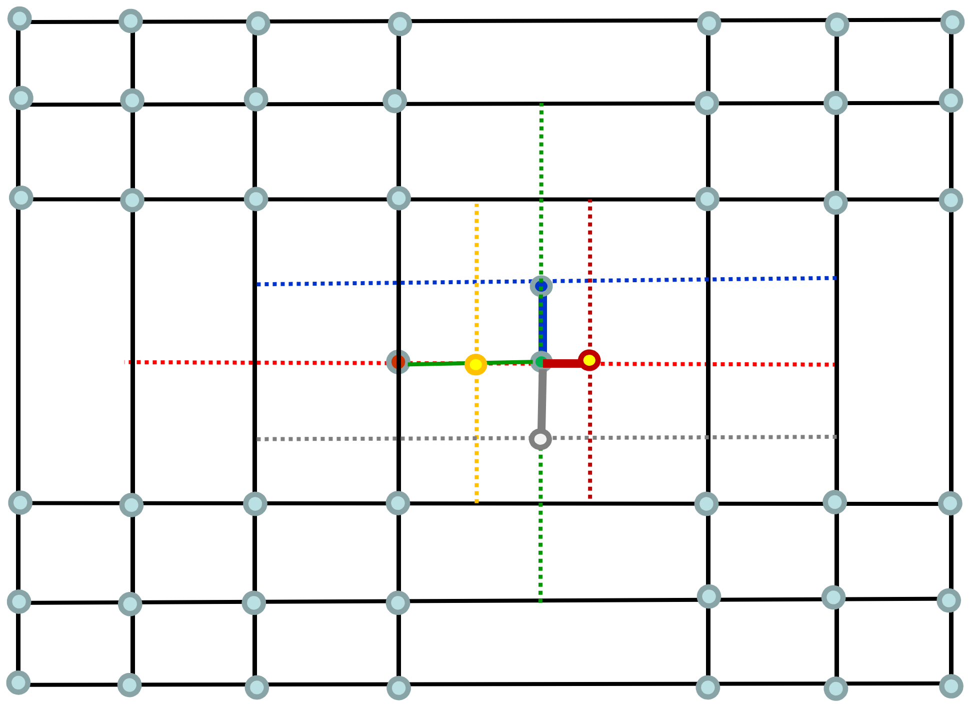
- The insertion of a vertex in a T-spline vertex grid (single vertex, T-, I-, L-joint) can be regarded as:
  - A specification of the parameter direction in which to refine
  - The parameter value to be used for the refinement
  - The location of the center of the new B-spline:
    - For odd degrees the location of the middle knotlines of the new B-spline
    - For even degrees the location of the mid-knot interval of the new B-splines
- This information is sufficient for performing refinement directly in the μ-extended box mesh
  - The hand-in-hand principle can be used for check linear independence

T-spline vertex grid as interface to LR B-splines - properties

- Andrea Bressan, University of Pavia, has compared T-spline and LR B-spline refinement in the case where exactly $(p+1) \times (p+1)$ B-splines overlap the elements of the box partition and found that in most cases the B-splines are the same.
  - Difference observed related to lines with multiplicities
- T-spline compatible LR B-spline can be defined
  - Restriction imposed on which refinement are allowed for LR B-splines
- The vertices and lines in the T-spline T-mesh have all a well defined location in the parameter domain.
  - Projecting the T-mesh/Dual mesh on to the LR-spline surface a T-spline type refinement can be specified directly in the parameter domain of the LR B-spline by specifying the location of the center point of a new B-spline in the mesh.

Draft of concept: T-spline type vertex mesh driving LR B-spline refinement in parameter domain





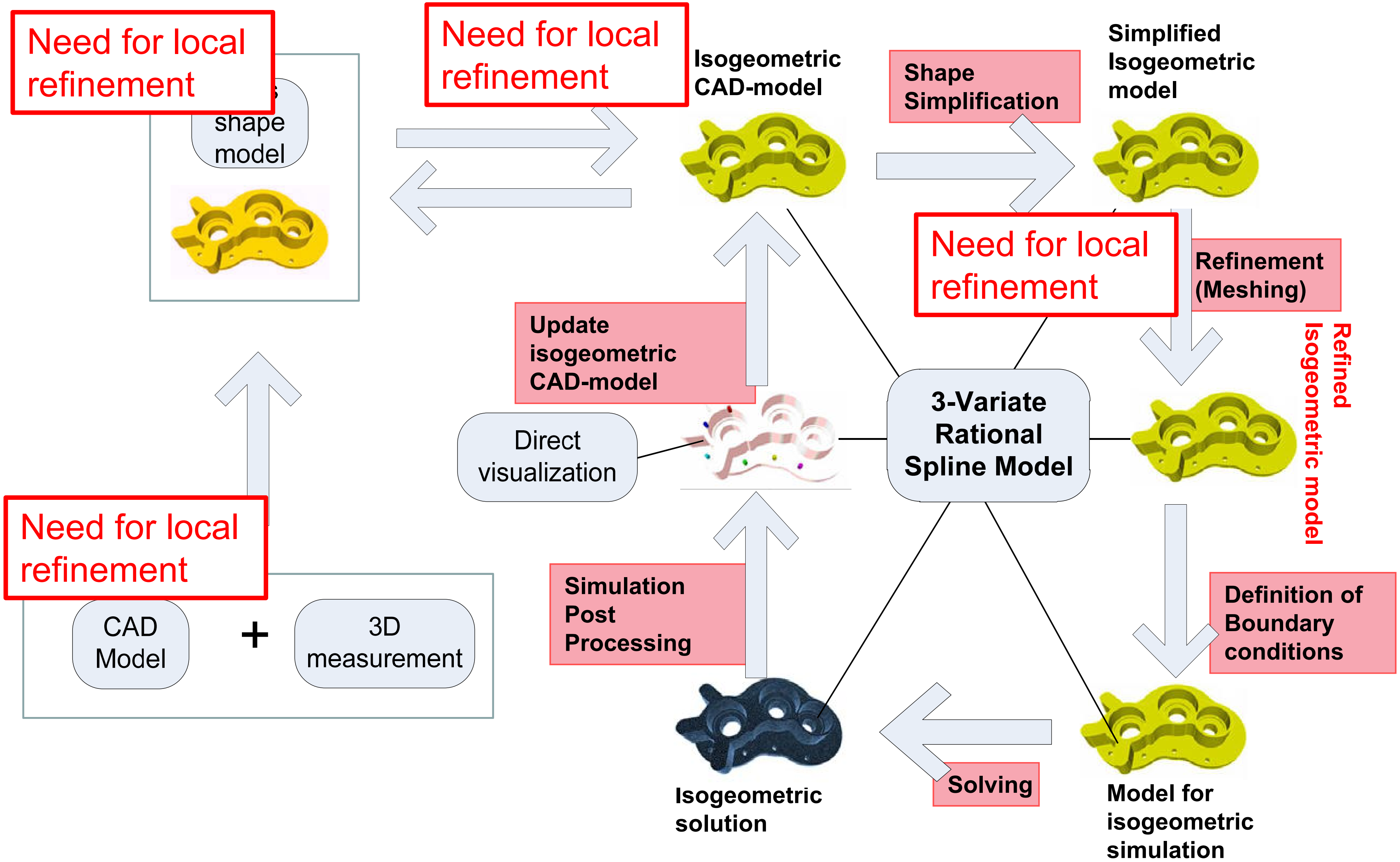
Concluding remarks

- T-splines and LR B-splines are two related approaches to Local Refinement of B-splines
- Refinement of LR B-splines can be performed by
  - Insertion of mesh-rectangles
  - T-spline like refinement approaches somewhat restricting the allowed refinements.
- The possibility of using a T-spline like interface also opens up the possibility to replace the T-spline rules for creating B-splines by the LR B-spline approach thus opening up for the use of the LR B-spline results on dimensionality and linear independence.

Current work on LR B-splines at SINTEF

- LR Splines extensions to the SINTEF GoTools C++ library is under way. EU-project: TERRIFIC.
- 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 (New EU-project starting October 1.
- We look at LR B-splines in design optimization. ITN Network SAGA.

Simulation – Future Information flow



The end

- Click here for video of the isogeometric dancing queen.
- http://www.youtube.com/watch?v=7LGpiptQ1u4