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

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

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

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

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 $\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1}, \mu_{j+1})$.

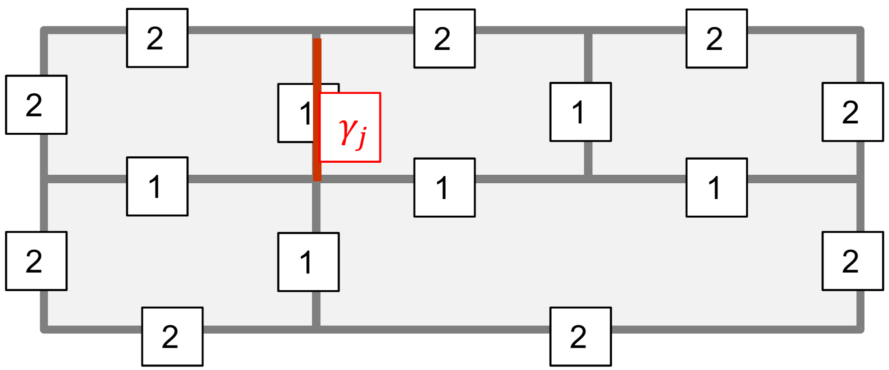
To ensure linear independence we have to

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

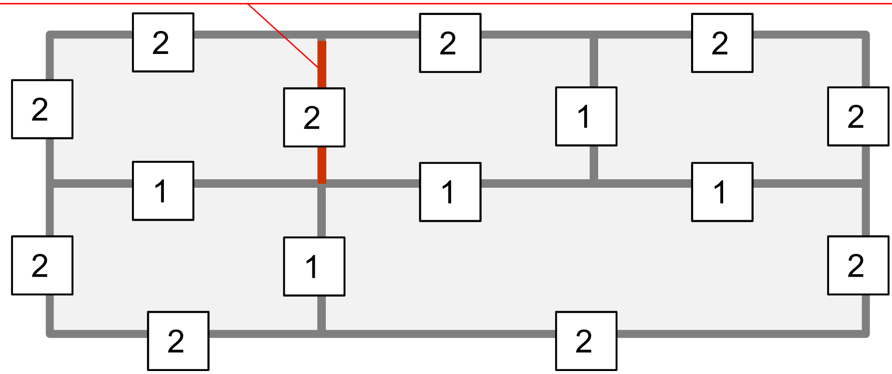
Difference in spanning properties between $\mathcal{B}_j$ and $\mathcal{B}_{j+1}$

- 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 $\mathcal{B}_\gamma$.



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



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

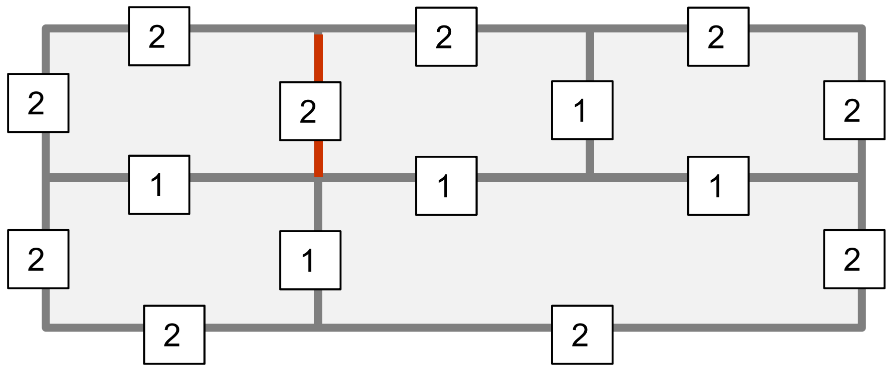
The use of $\mathcal{B}_\gamma$

- 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

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



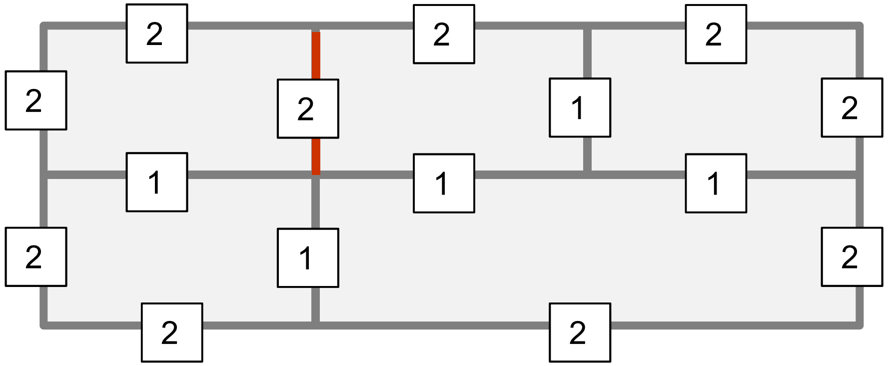
$(\mathcal{M}_{j+1}, \mu_{j+1})$

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

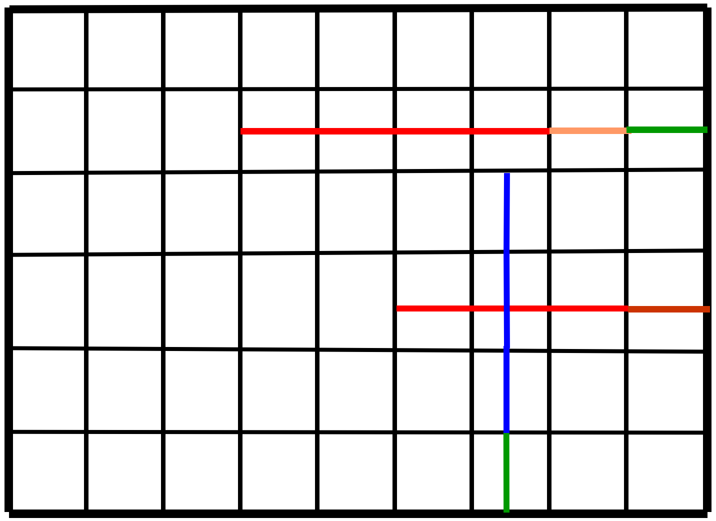
- To find the dimension of a spline space with many B-splines is more complex than finding the dimension of a spline space with few B-splines
- When assessing the B-splines $\mathcal{B}_\gamma$ over γ_j we see if the refinement can be broken into a sequence of LR B-spline refinements with as low dimension 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

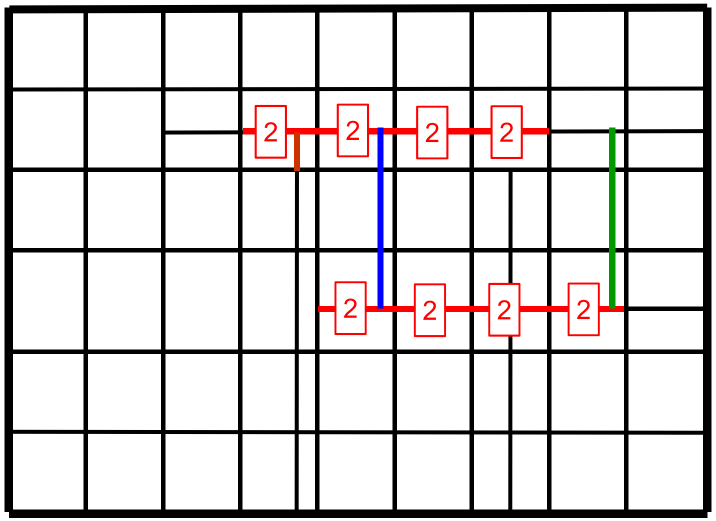
Mesh-rectangle length 1 extending existing mesh-rectangle, T-joint at other end. Dimension increase 1. Trivial.

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:

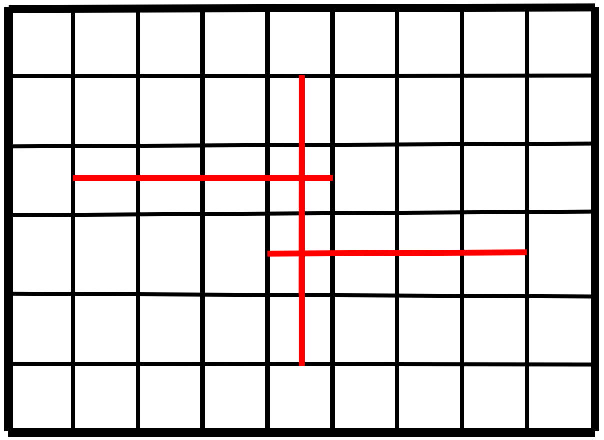
1. Do the B-splines in $\mathcal{B}_{j+1}$ span $\mathbb{S}_{(3,3)}(\mathcal{M}_j, \mu_j)$?
2. Is $\mathcal{B}_{j+1}$ a basis for $\mathbb{S}_{(3,3)}(\mathcal{M}_j, \mu_j)$?

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



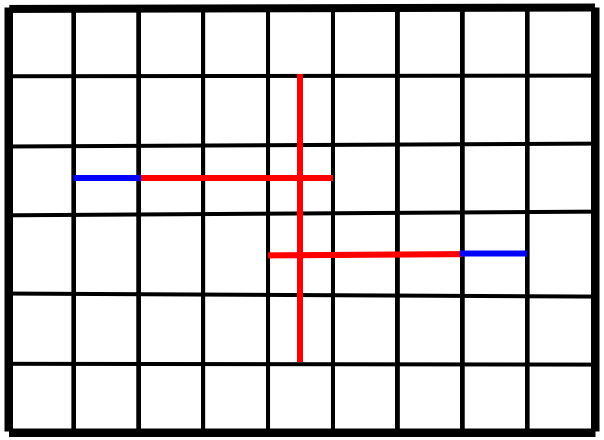
Dimension increase 1, one new B-splines (+5, -4)

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 $(\mathcal{M}_{j+1}, \mu_{j+1})$ in the general case?

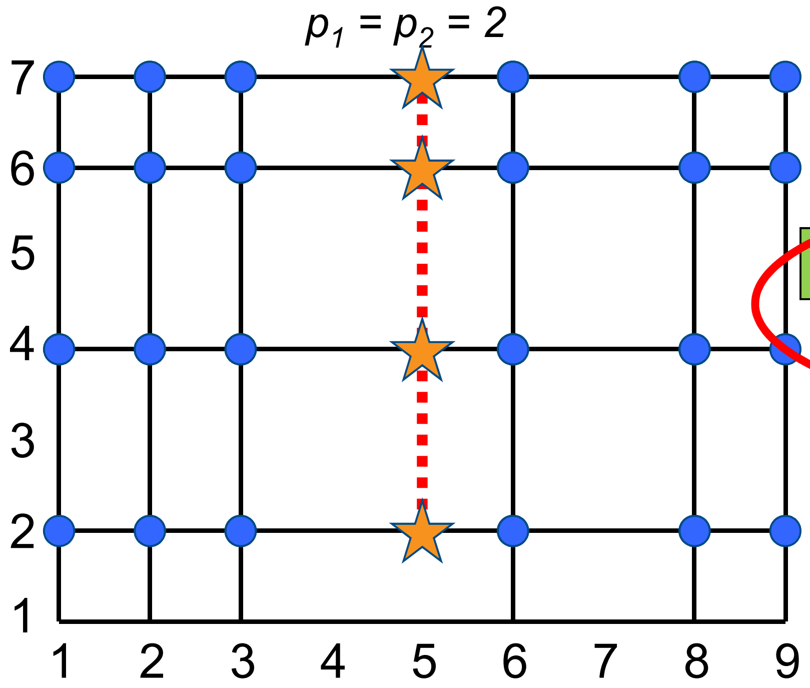
- Assume that $\mathcal{B}_j$ is a basis for $\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_j, \mu_j)$, .
- Make $(\mathcal{M}_{j+1}, \mu_{j+1}) = (\mathcal{M}_j + \gamma_j, \mu_{j,\gamma_j})$
- $\mathcal{B}_{j+1}$ is a basis for $\mathbb{S}_{\boldsymbol{p}}(\mathcal{M}_{j+1}, \mu_{j+1})$ if
  - The B-splines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

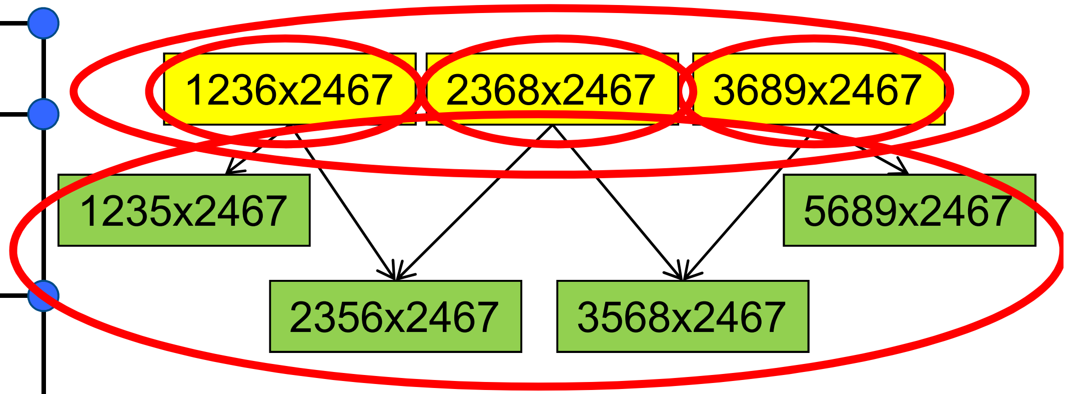
How to determine if the collection of B-splines goes hand in hand with the spline space?

- 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

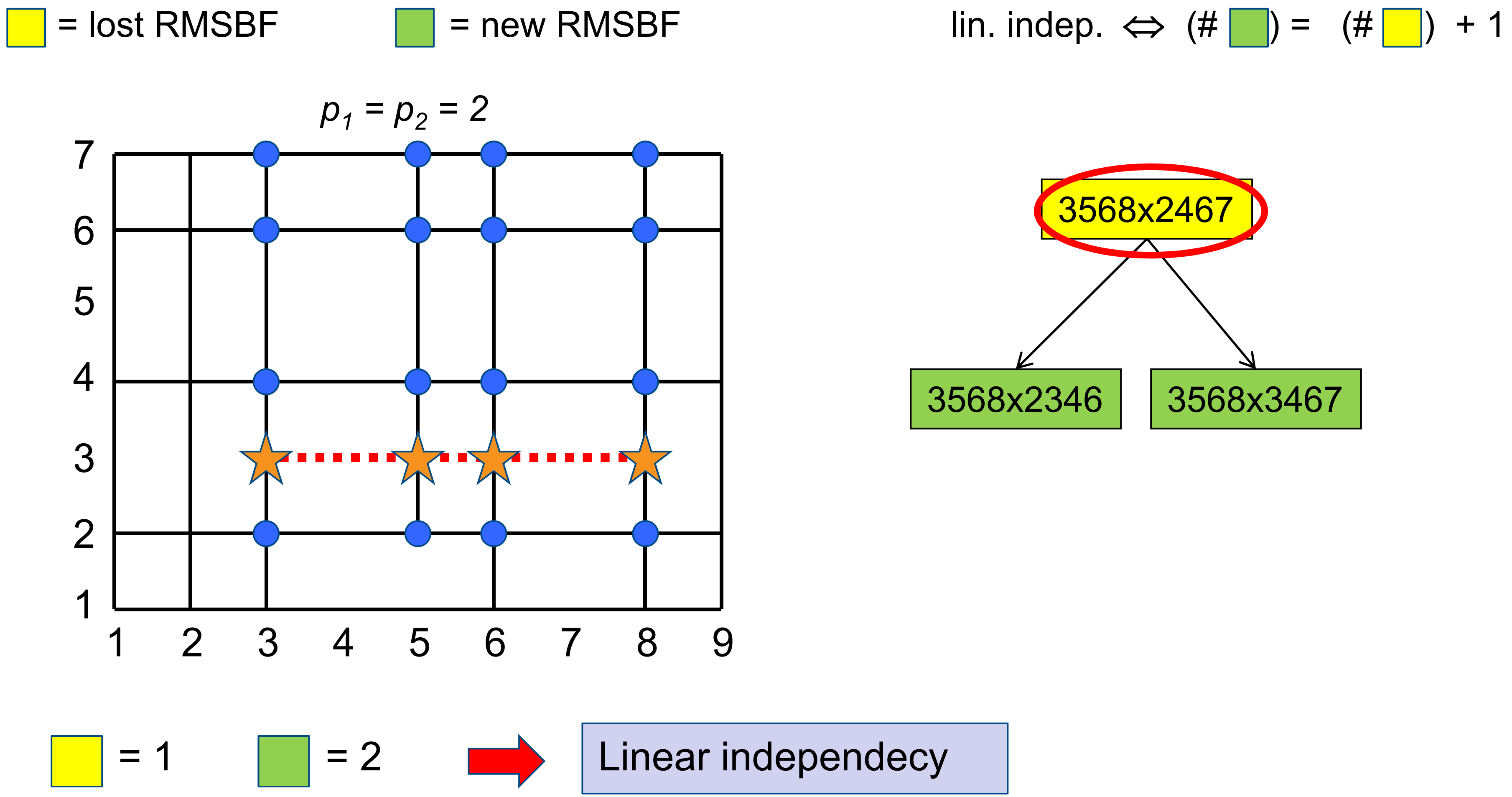




= 3 = 4 → Linear independecy

Example by: Kjell Fredrik Pettersen, SINTEF

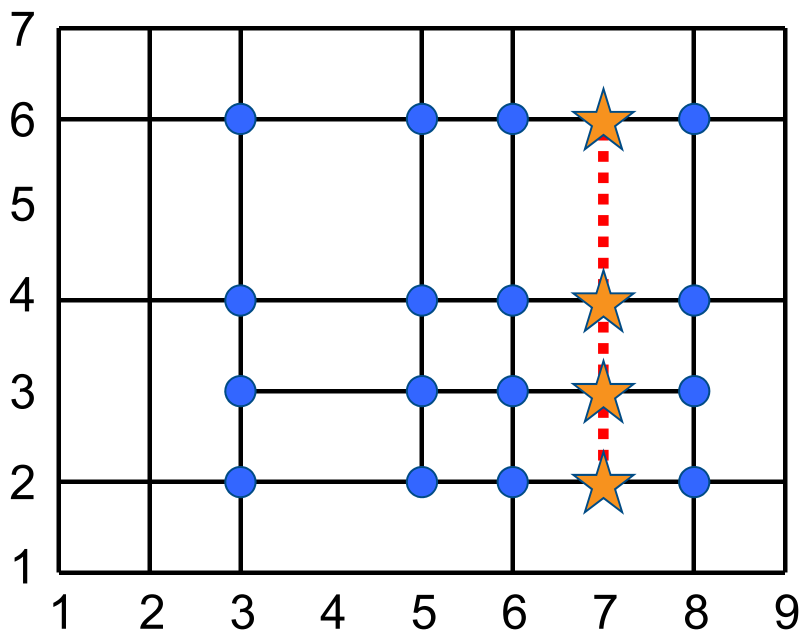
Linear dependency example

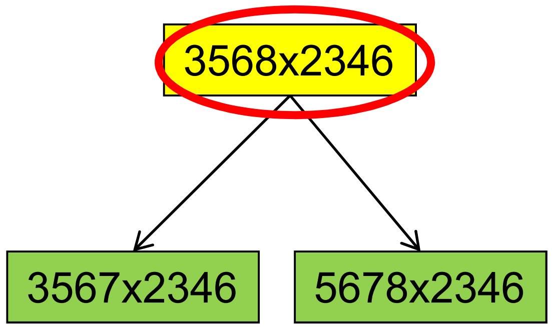


Linear dependency example

[yellow box] = lost RMSBF   [green box] = new RMSBF   lin. indep. ⇔ (# [green box]) = (# [yellow box]) + 1

$p_1 = p_2 = 2$





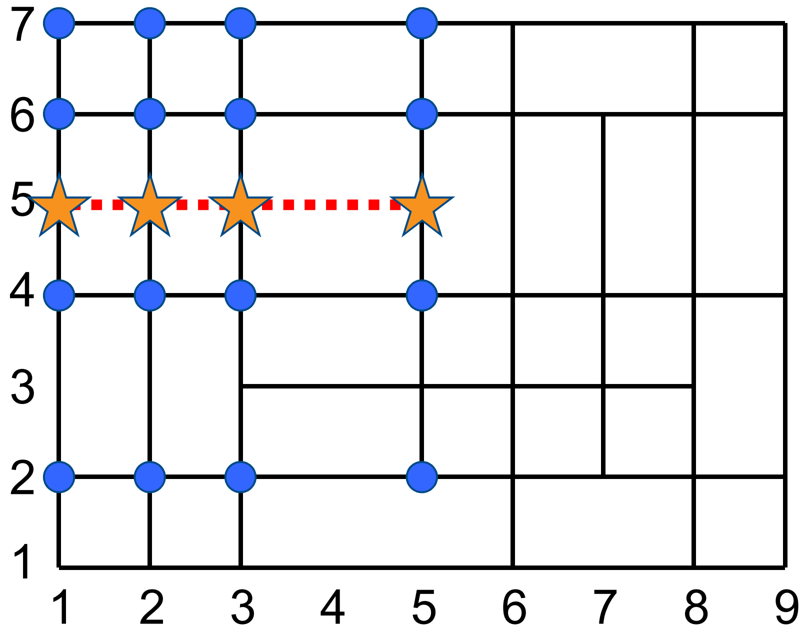
[yellow box] = 1   [green box] = 2   → Linear independecy

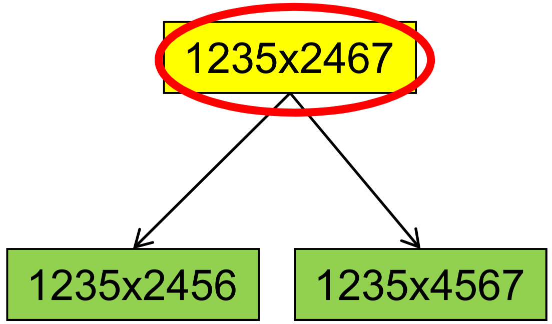
Linear dependency example

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

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

$p_1 = p_2 = 2$



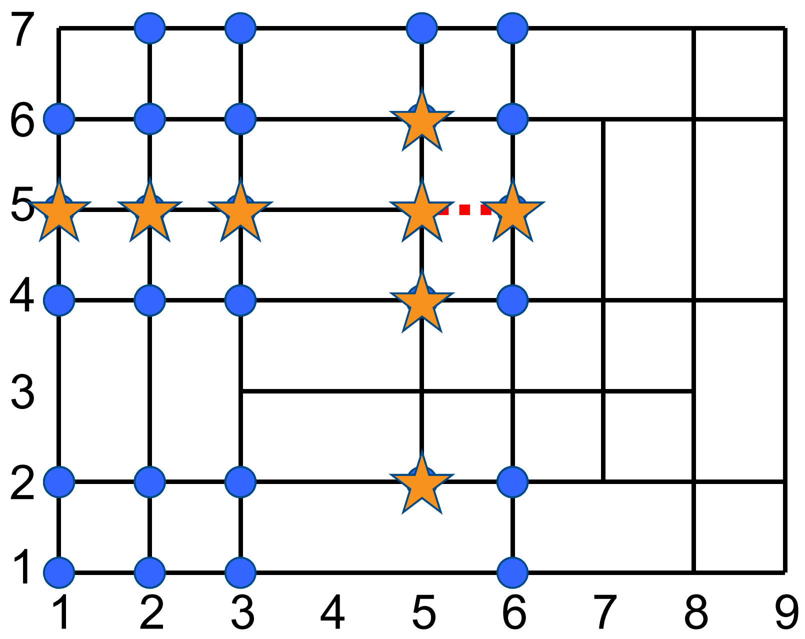


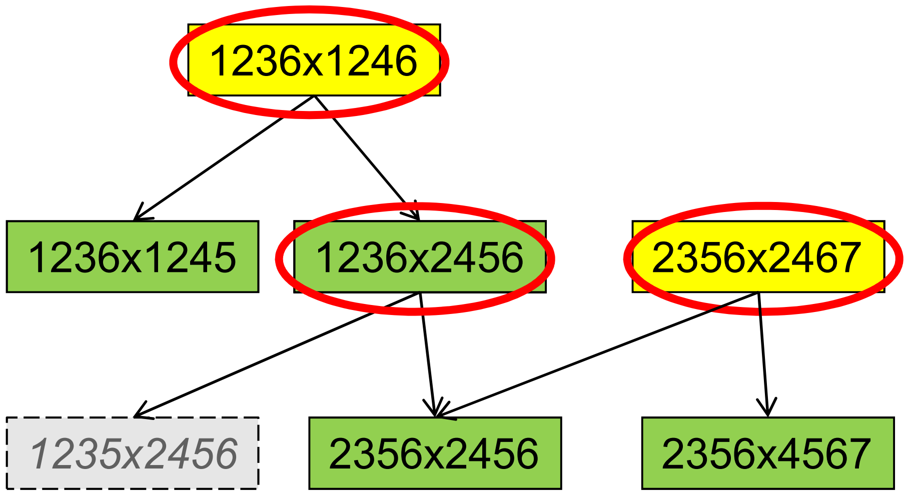
[yellow square] = 1   [green square] = 2 → Linear independecy

Linear dependency example

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

$p_1 = p_2 = 2$





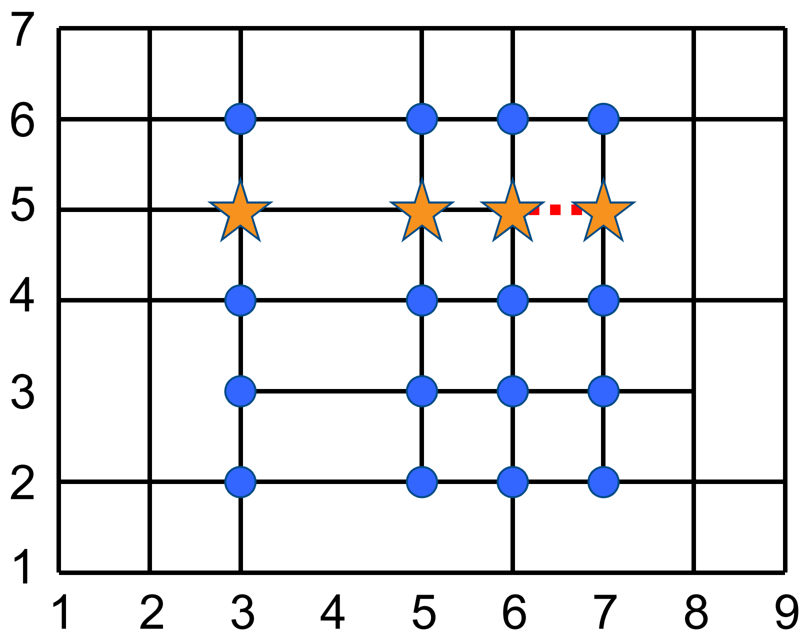
= 2 = 3 → Linear independecy

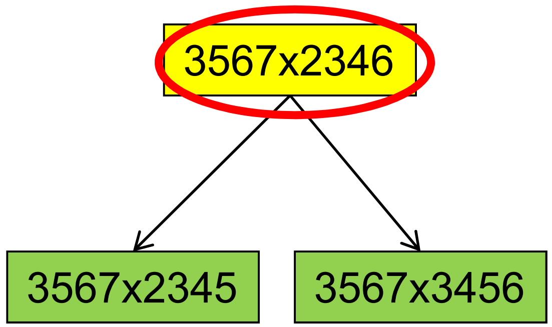
Linear dependency example

(yellow) = lost RMSBF (green) = new RMSBF

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

$p_1 = p_2 = 2$





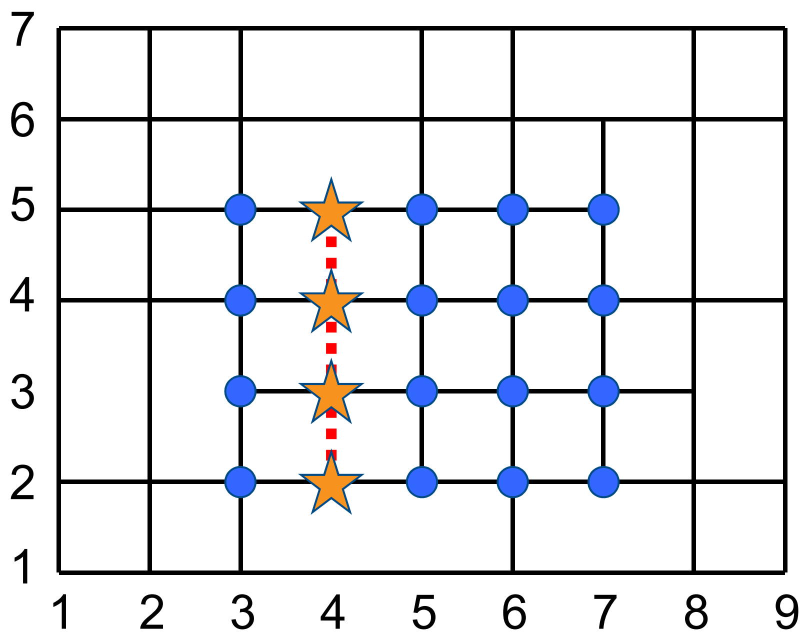
(yellow) = 1 (green) = 2 → Linear independecy

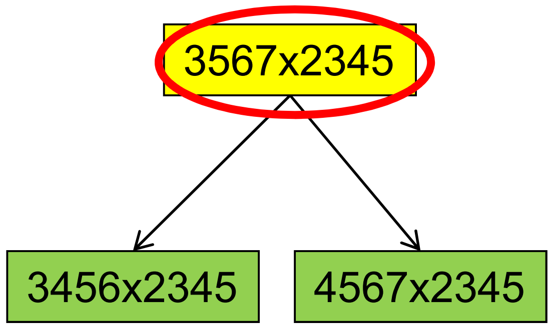
Linear dependency example

= lost RMSBF = new RMSBF

lin. indep. ⇔ (#) = (#) + 1

$p_1 = p_2 = 2$

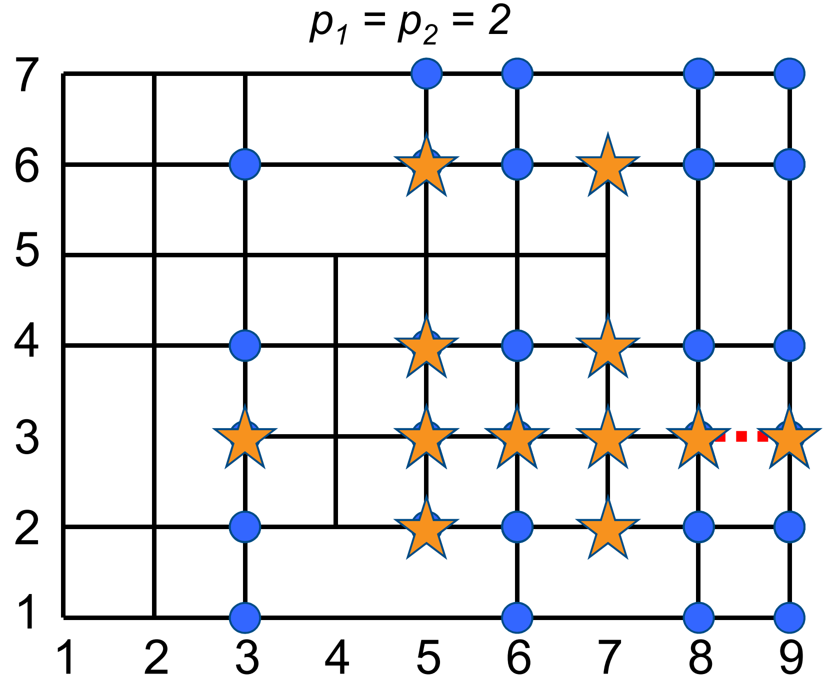


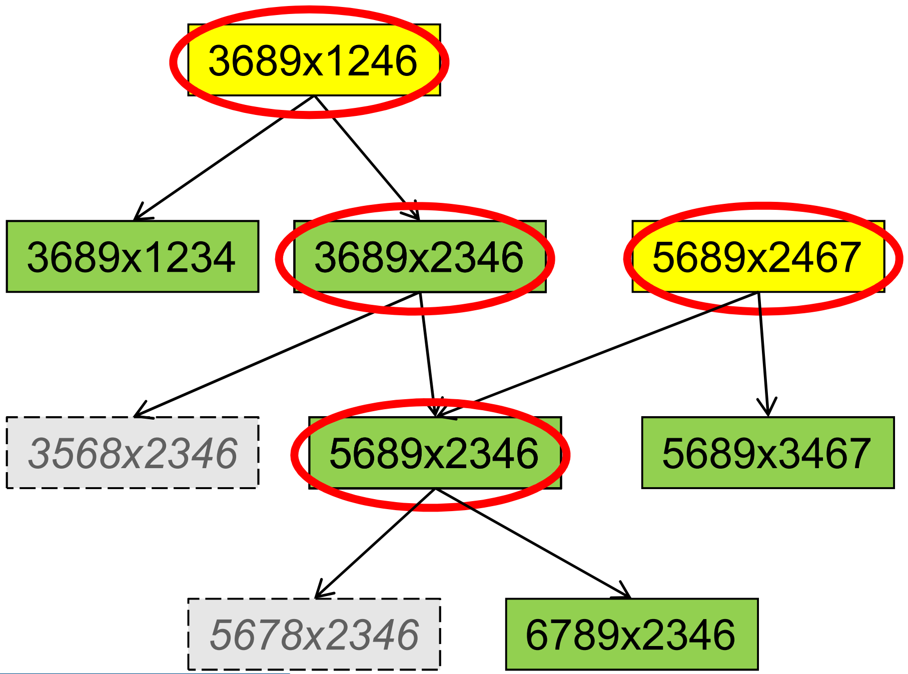


= 1 = 2 → Linear independecy

Linear dependency example

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



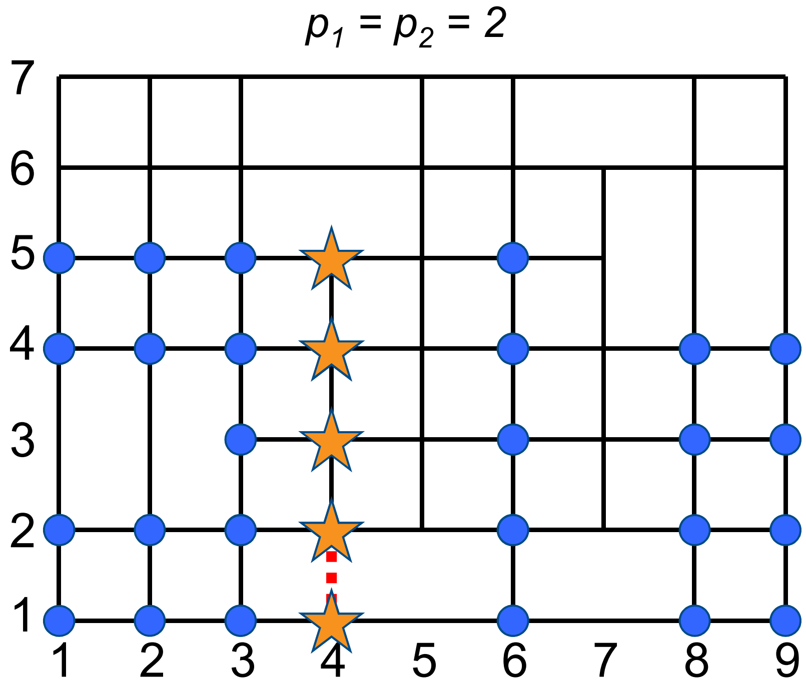


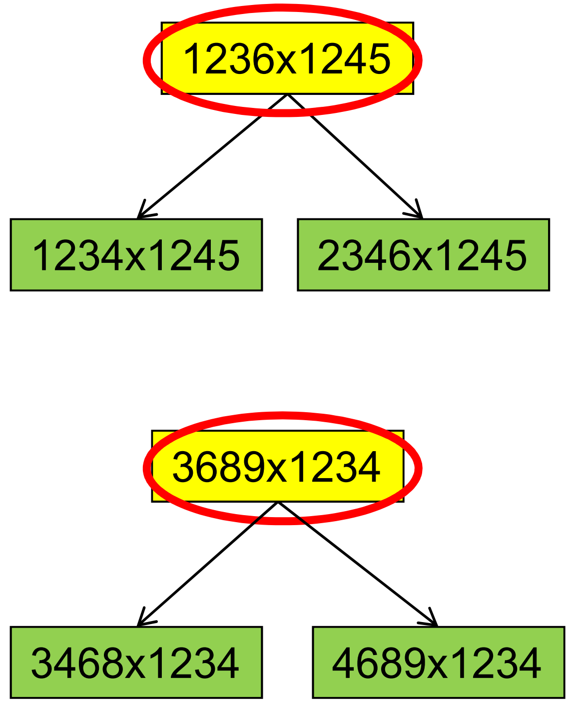
= 2 = 3

Linear independecy

Linear dependency example

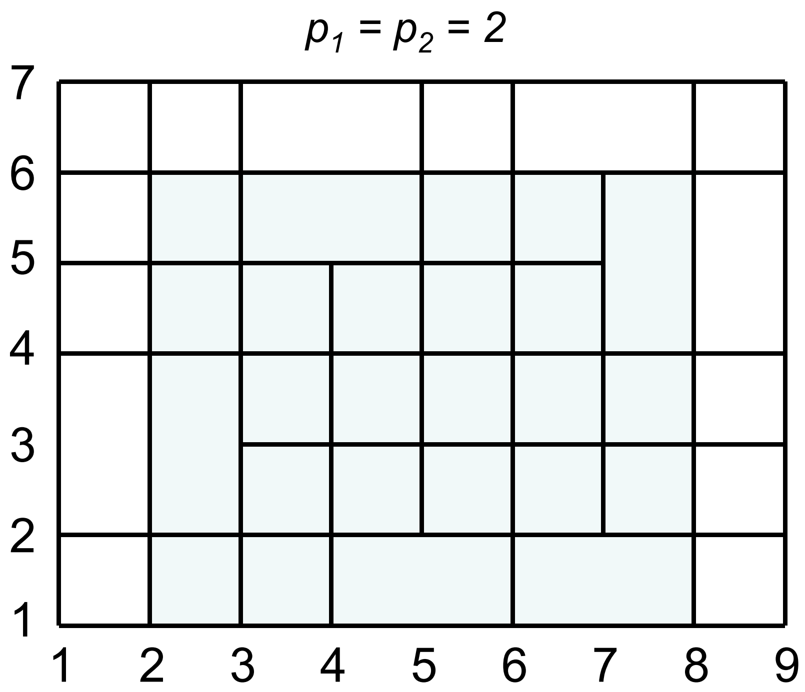
= lost RMSBF = new RMSBF lin. indep. ⇔ (# green) = (# yellow) + 1



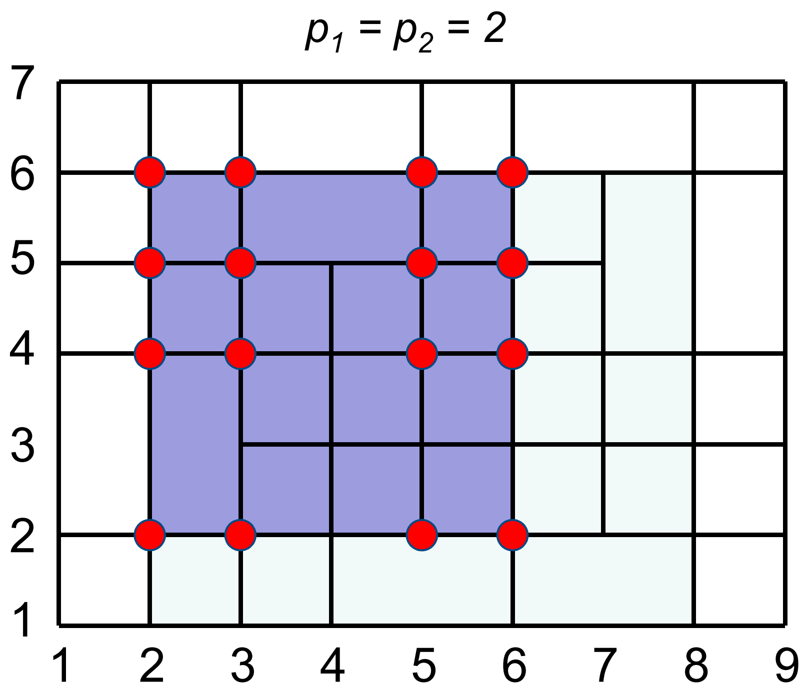


= 2 = 4 **Linear dependecy!!!**

Linear dependency example

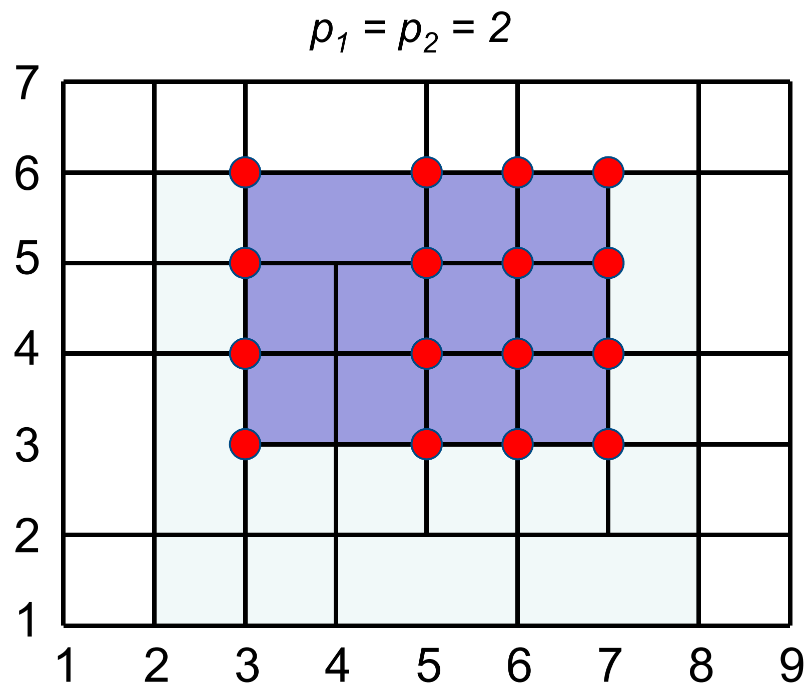


Linear dependency example



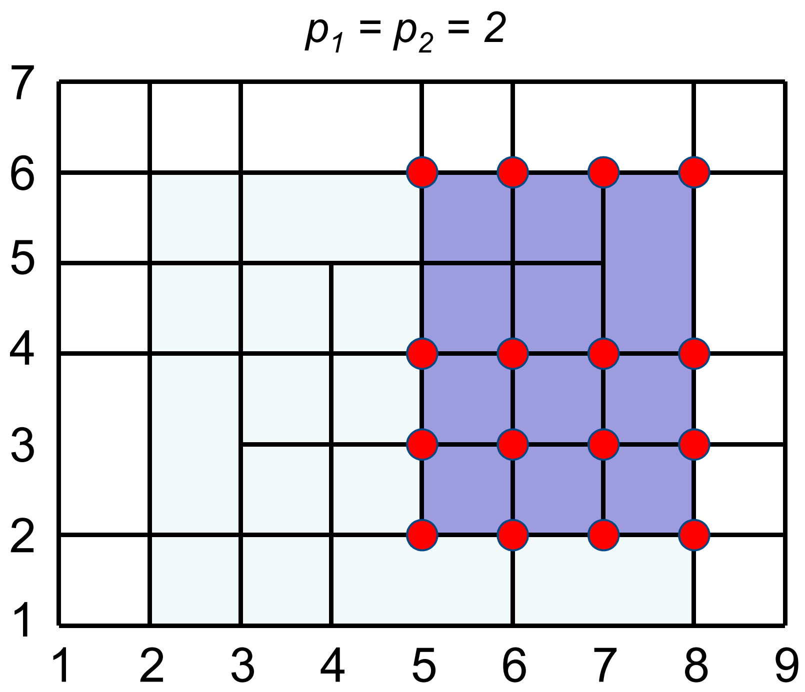
2356x2456

Linear dependency example



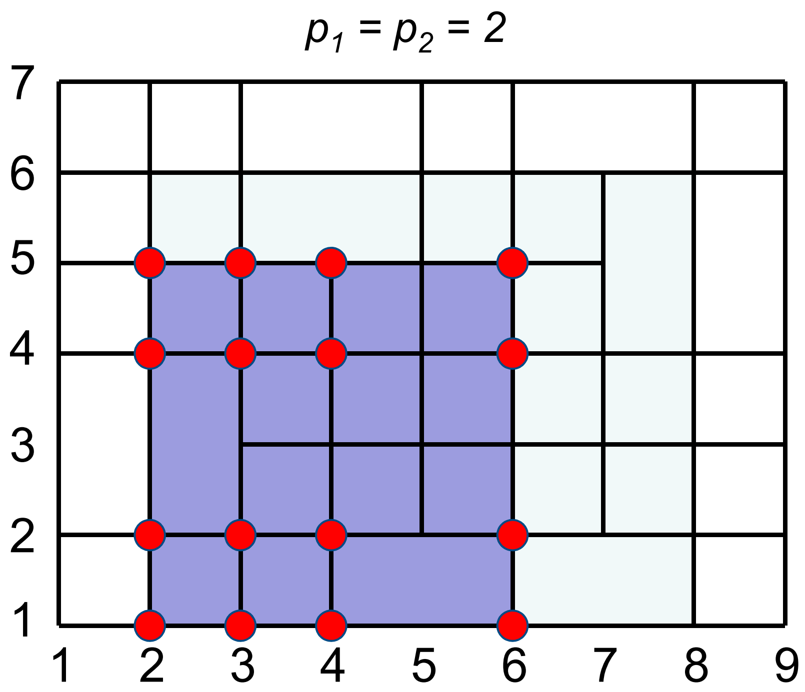
3567x3456

Linear dependency example



5678x2346

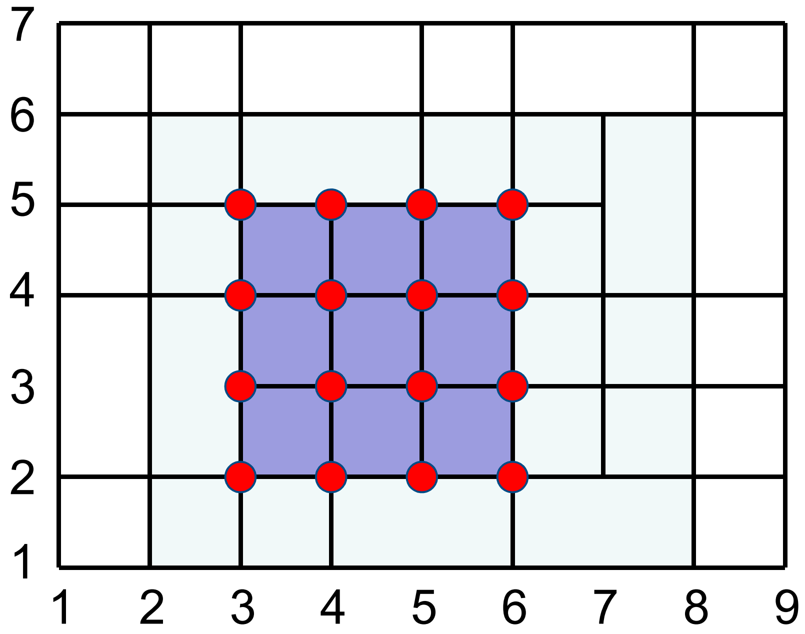
Linear dependency example



2346x1245

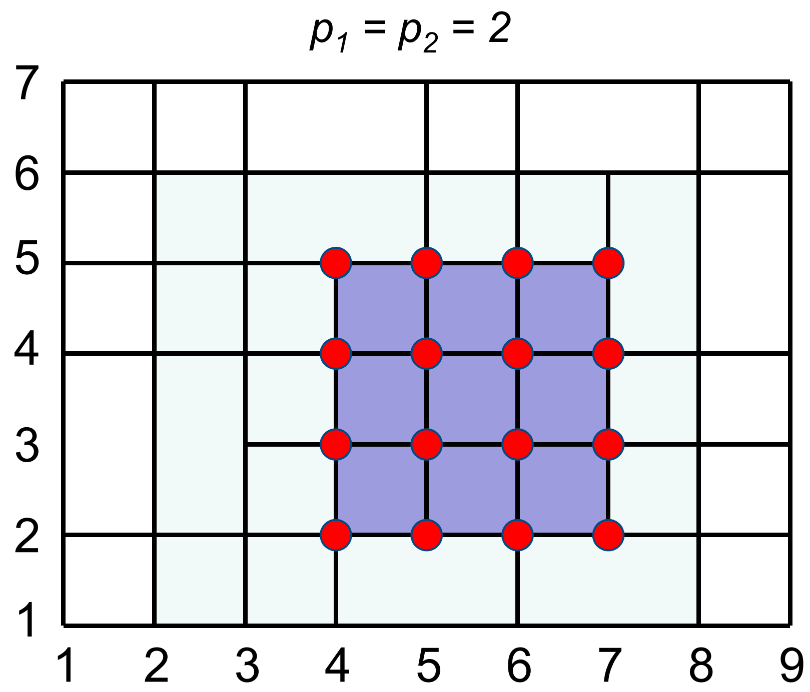
Linear dependency example

$p_1 = p_2 = 2$



3456x2345

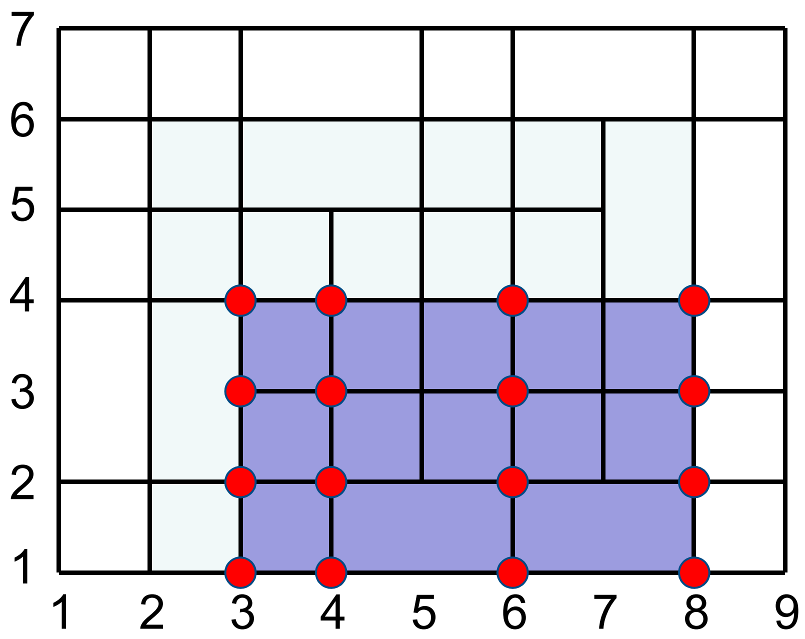
Linear dependency example



4567x2345

Linear dependency example

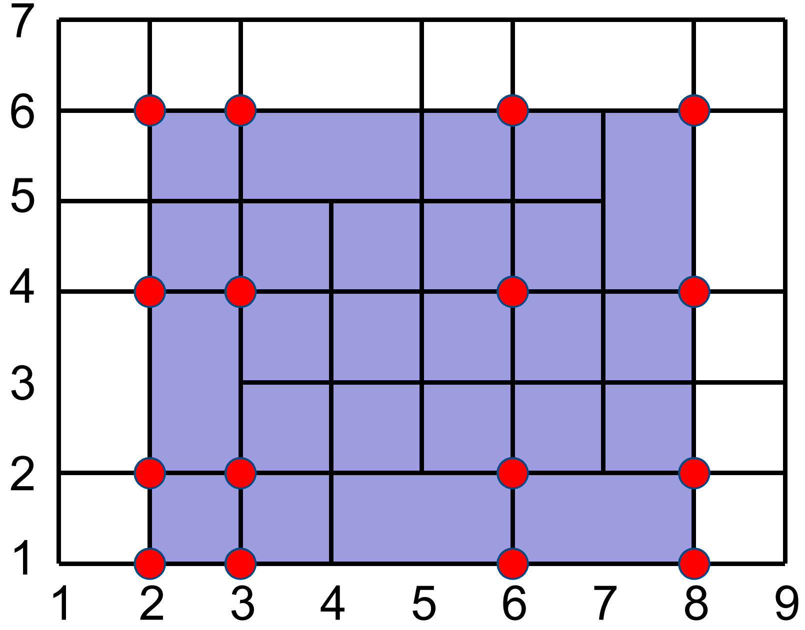
$p_1 = p_2 = 2$



3468x1234

Linear dependency example

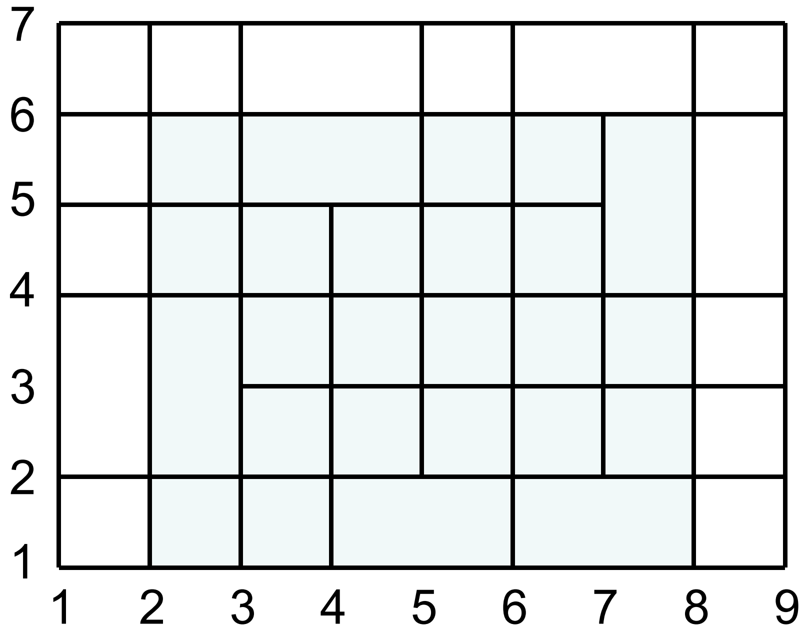
$p_1 = p_2 = 2$



2368x1246

Linear dependency example

$p_1 = p_2 = 2$



Linear relation

(knot value = knot position)

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

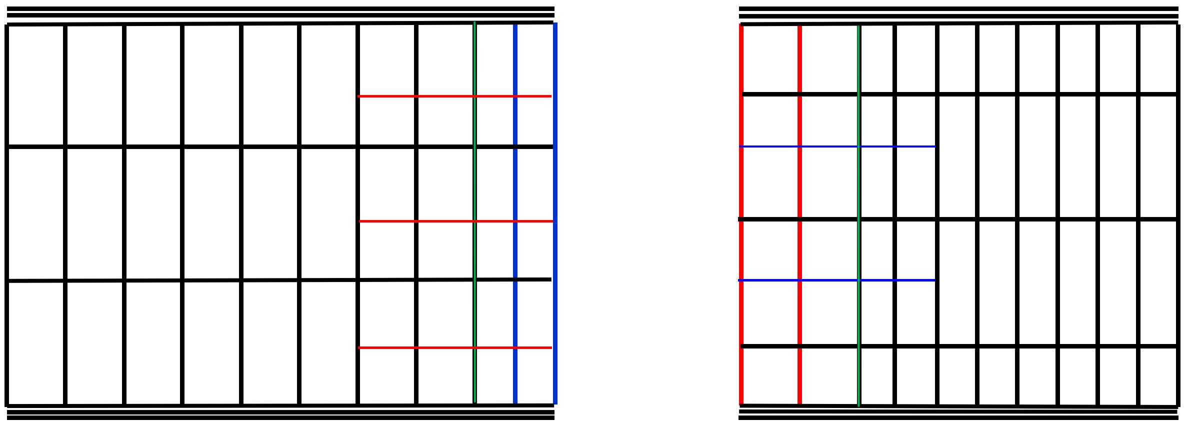
What to do to handle the situation when we produce too many B-splines to have a basis?

- 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

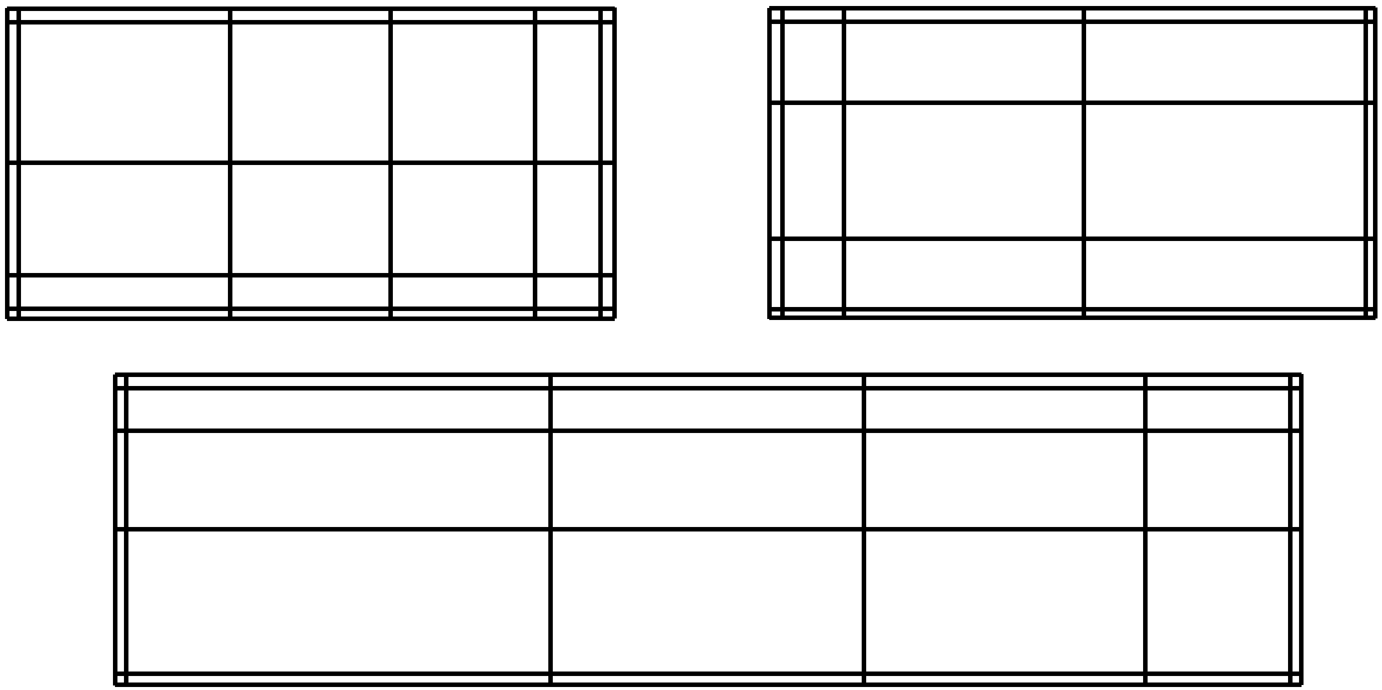
- Stitching of B-spline patches
- Approximation of large data sets

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

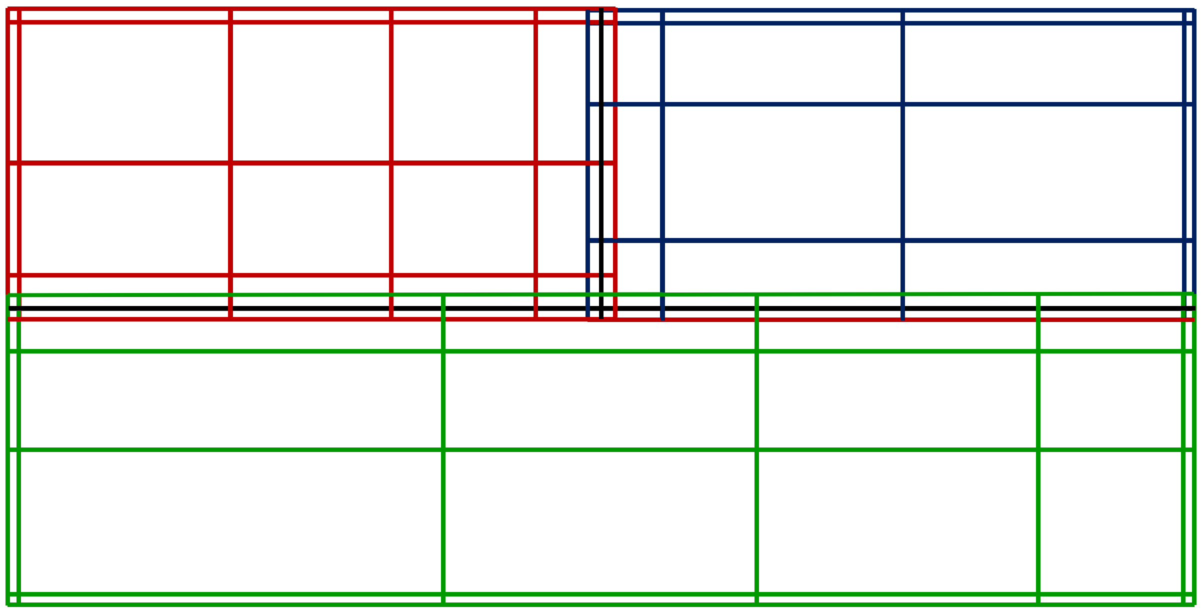


1. Adapt the edge knotlines of A to B
2. Adapt the edge knotlines of B to A
3. Insert horisontal knotline segement from B in A
4. Insert horisontal knotline segement from A in B
5. Merge the parameter domains

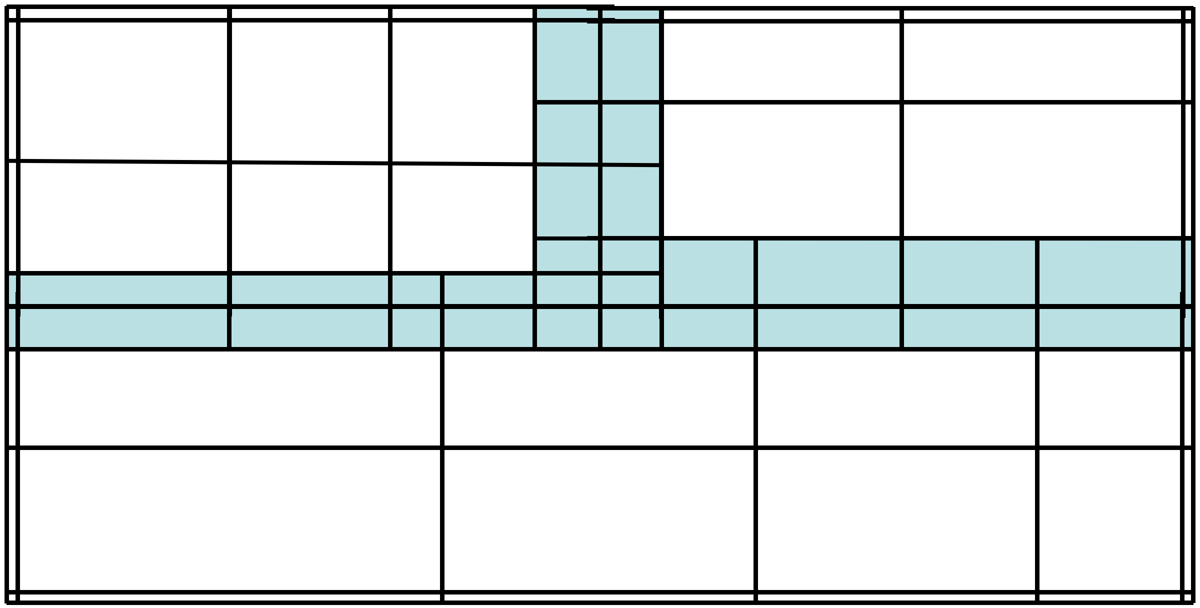
Multi-block T-joints (1) match parametrizaton



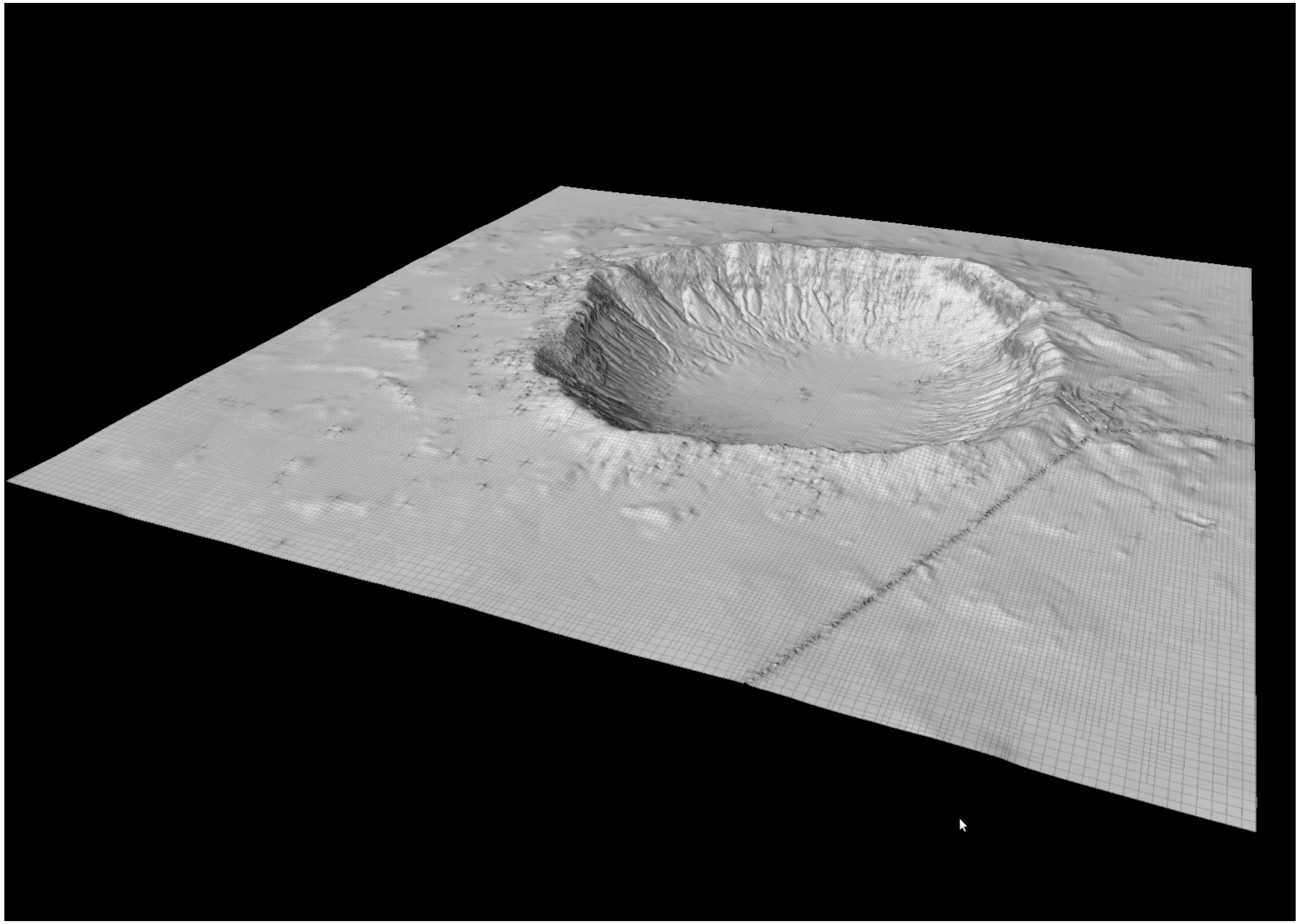
Multi-block T-joints(2) adjust boundary knotlines



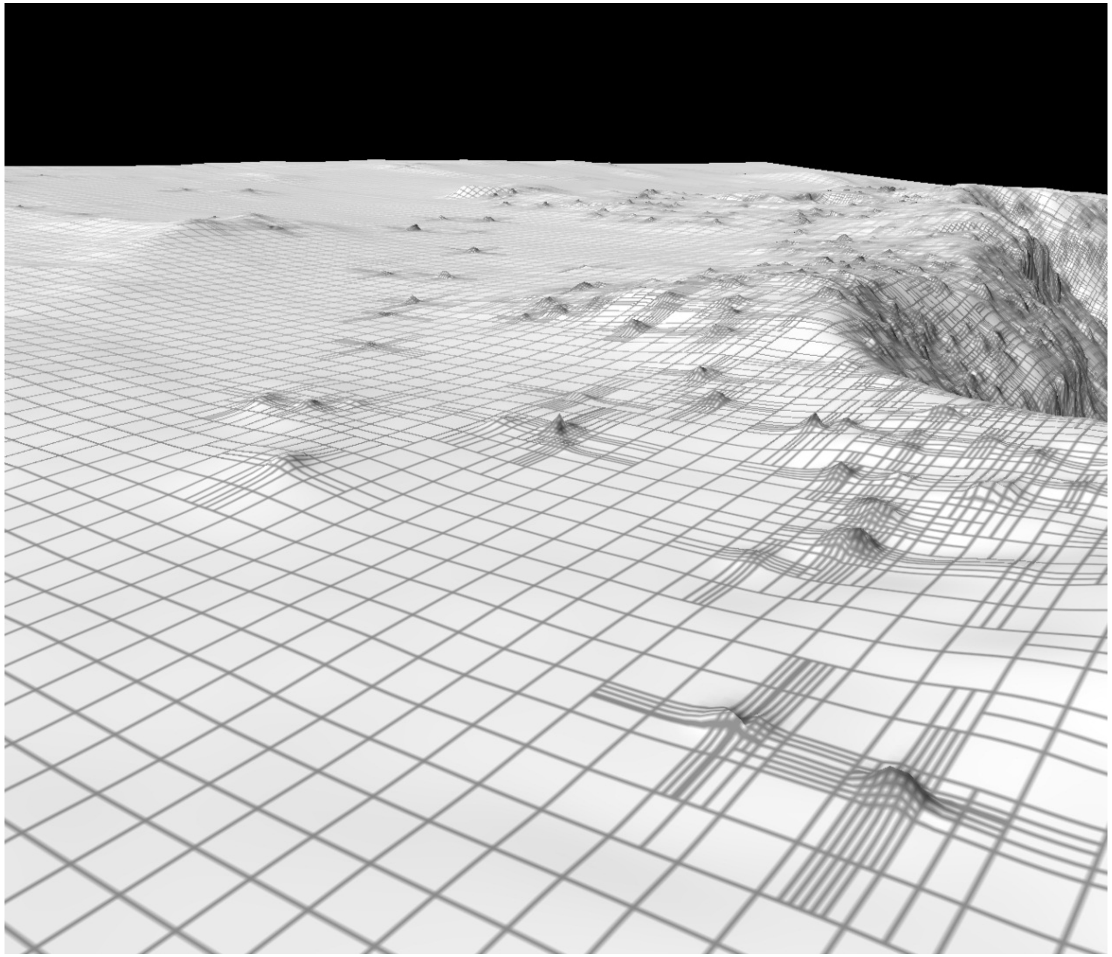
Multi-block T-joints(3)
identify + split transition B-splines



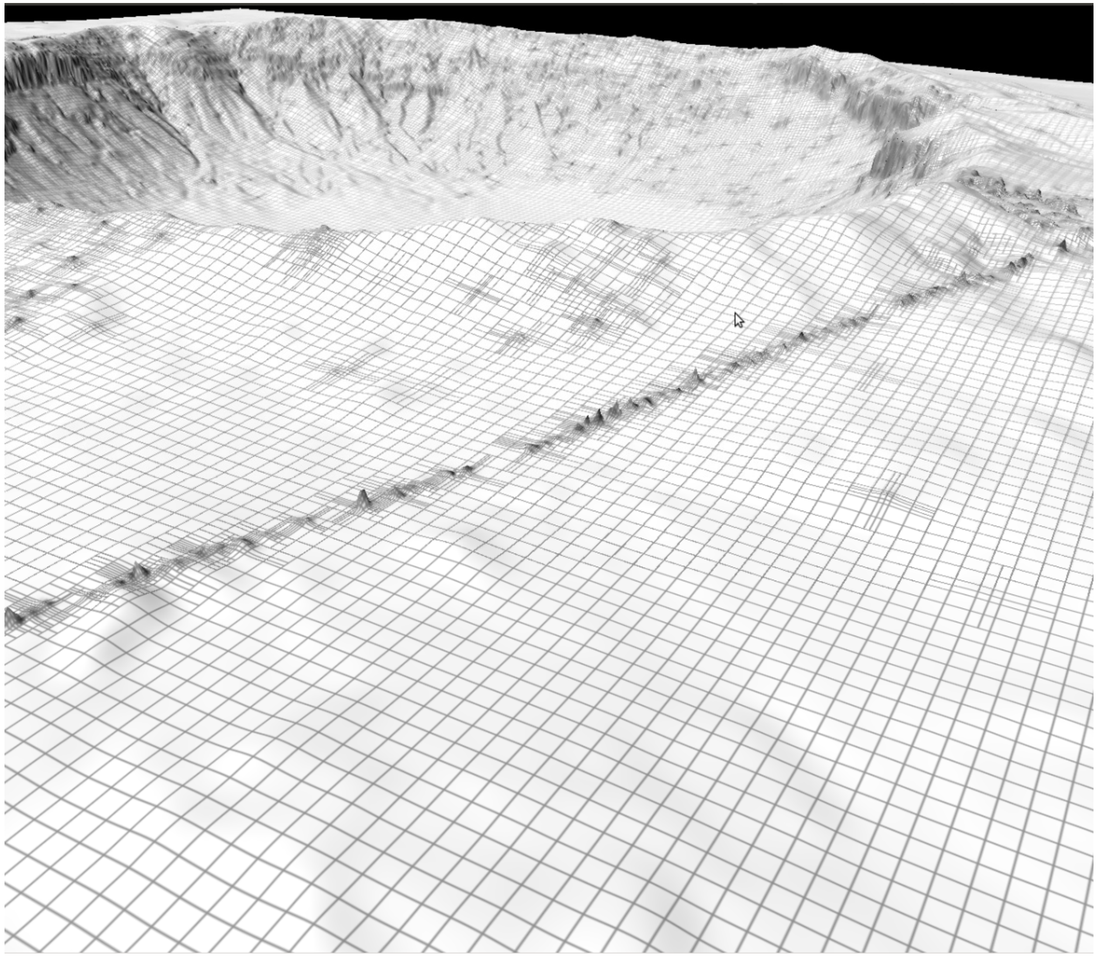
Approximation of large data set
Barringer crater Arizona



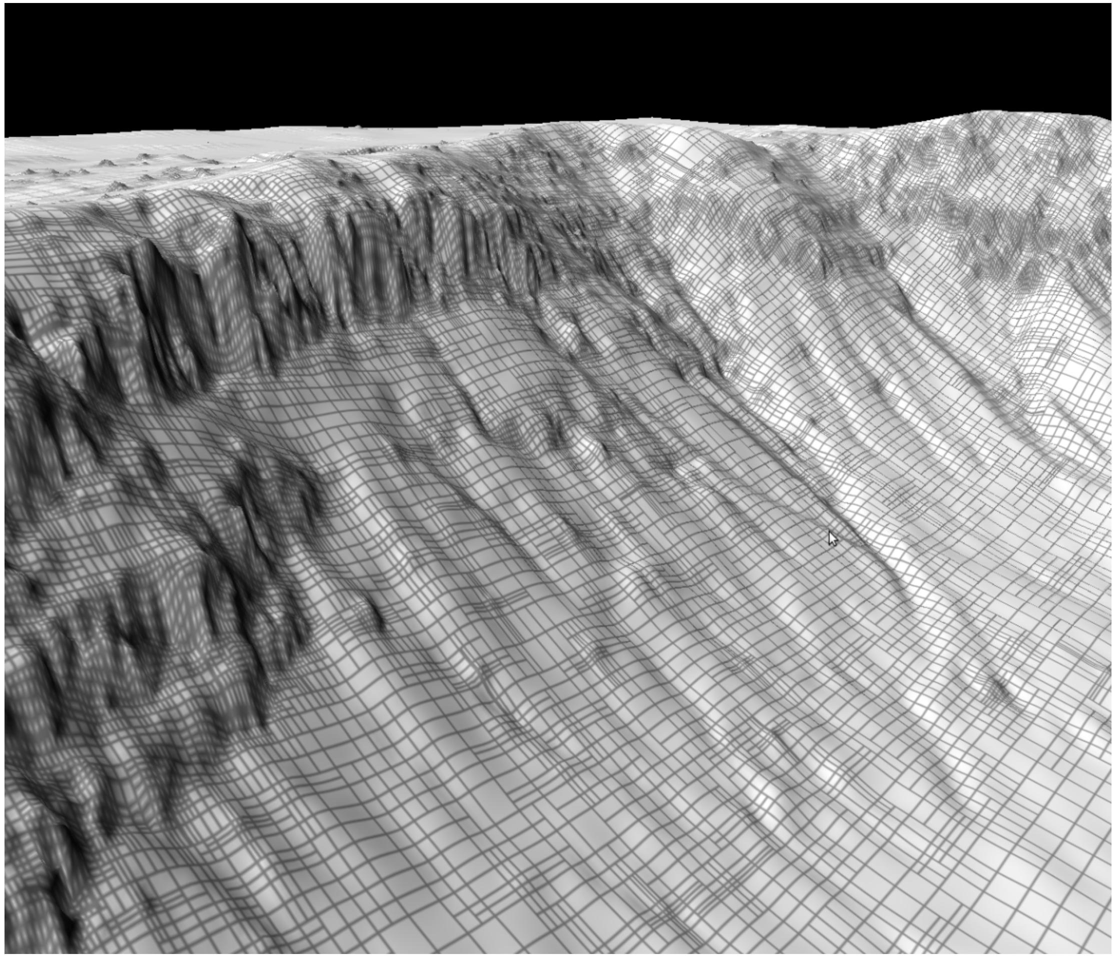
Local refinement to adapt to fine details



Data along powerline? reproduced



Details along inside slope



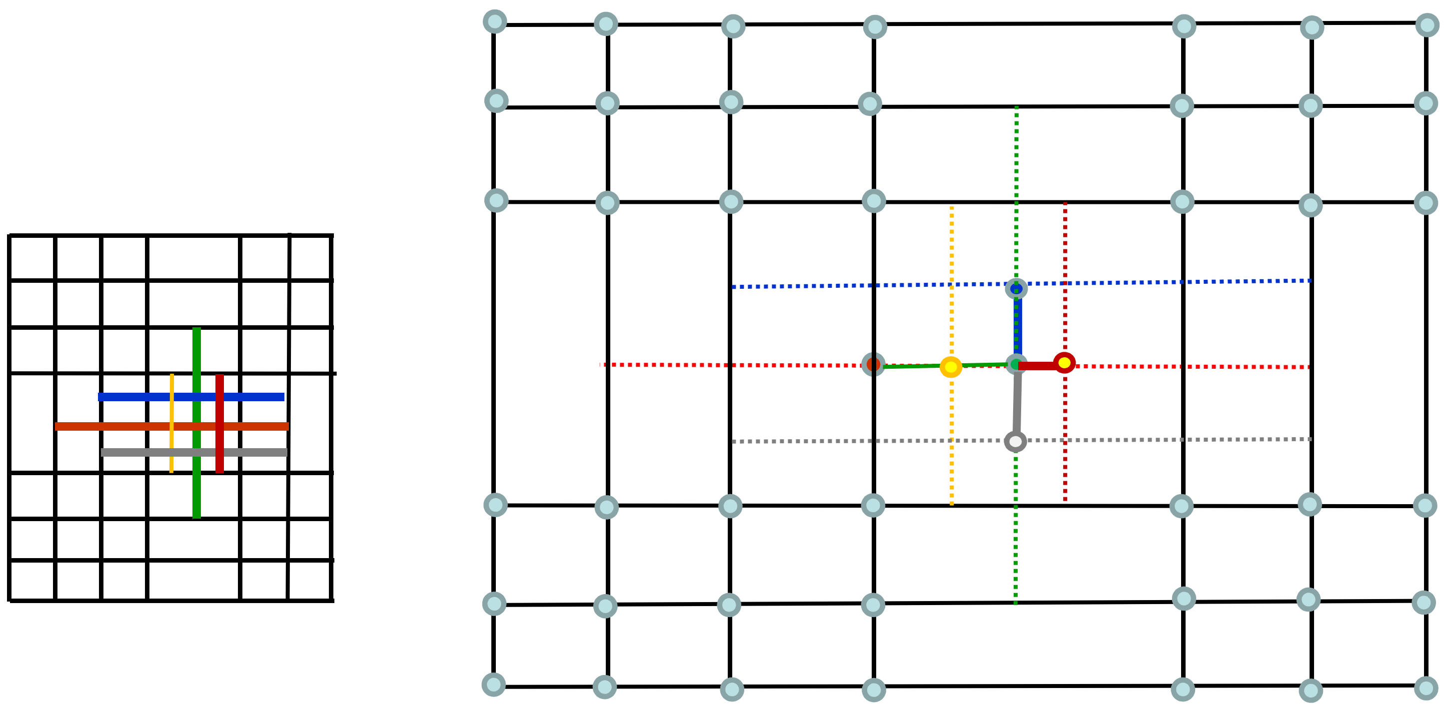
T-spline vertex grid as interface to LR B-splines

- 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



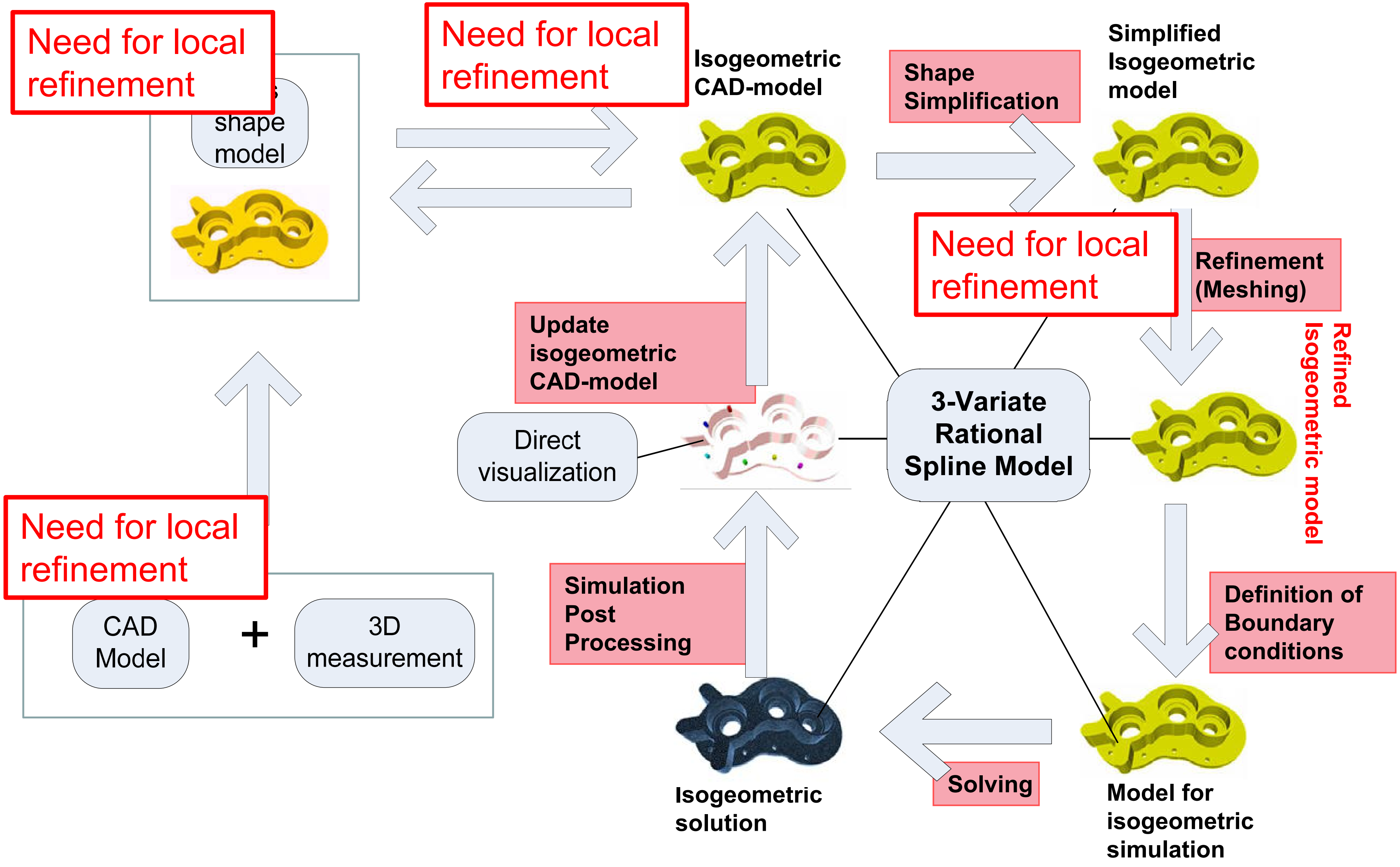
Concluding remarks

- 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