Locally refined splines

Tor Dokken

SINTEF and CMA, University of Oslo.

In cooperation with

Tom Lyche CMA, University of Oslo,

Kjell Fredrik Pettersen and Vibeke Skytt, SINTEF.

Organization of talk

- Challenges of isogeometric representation for CAD
- Why NURBS is not sufficient as a spline representation for isogeometric analysis
- Need for local refinement not satisfied by NURBS
- Local refinement
- **Locally refined splines (LR-splines)**
- Examples of LR-splines

Independent evolution of CAD and FEM

- CAD (NURBS) and Finite Elements evolved in different communities before electronic data exchange
  - FEM 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 (E.g, ,1973: Strang and Fix's *An Analysis of The Finite Element Method)*
  - B-splines: 1972: DeBoor-Cox Calculation, 1980: Oslo Algorithm

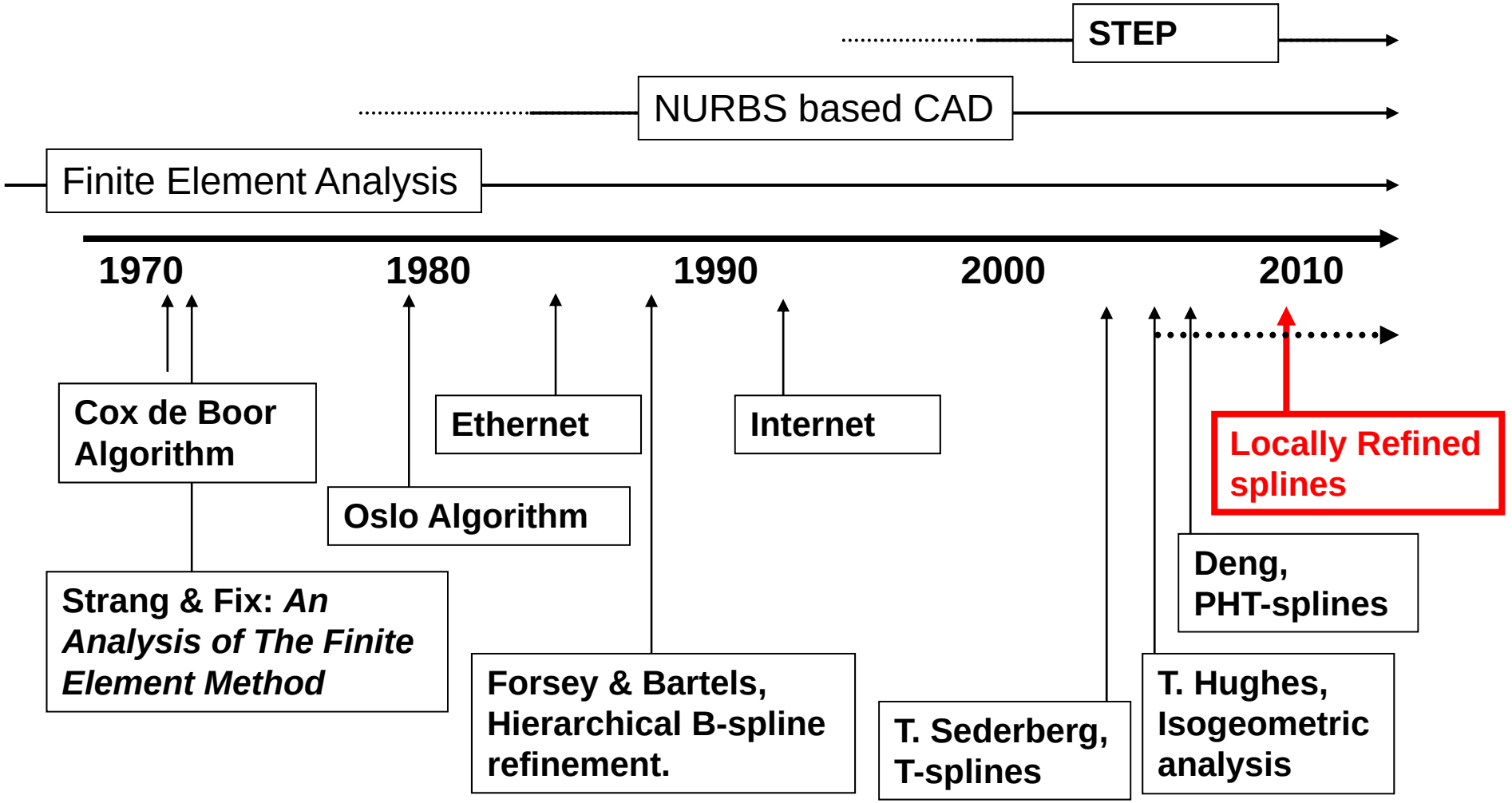
From stand alone computers and systems to integrated information flows

- 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



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 (Oslo Algorithm, 1980, Cohen, Lyche, Riesenfeld)
  - B-spline has a rich set of refinement methods

Why have NURBS not been used in FEA?

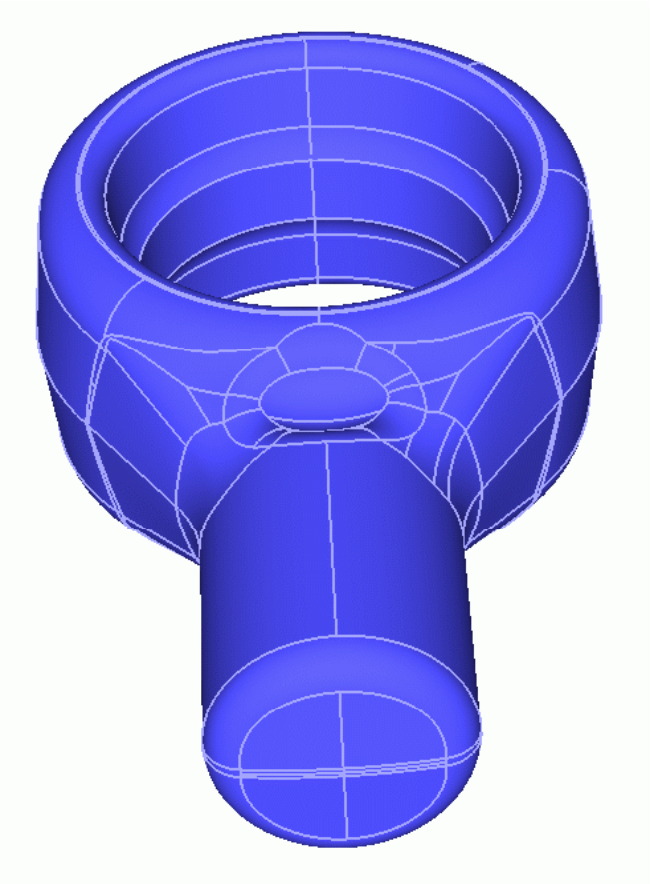
- FEA was developed before the NURBS theory
- NURBS and Finite Elements evolved in different communities before electronic data exchange
  - It was “agreed” that higher order representations in most cases did not contribute to better solutions
- Current computers have extreme performance compared to earlier computers. Allows more generic solutions.
  - Mathematical representation in CAD and FEA chosen based on what was computationally feasible.
- We needed someone with high standing in FEA to promote the idea of splines in analysis
  - Tom Hughes did this in 2005
  - The Computer Aided Design Community has adopted the idea
  - A new drive in spline research after 10 quite years

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 NURBS block structure

- We would like considerably fewer NURBS blocks than the number of surfaces patches
- 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.
- Current CAD is standardized in ISO STEP (ISO 10303)
- 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 STEP should also encompass isogeometric CAD

Two approaches to isogeometric CAD

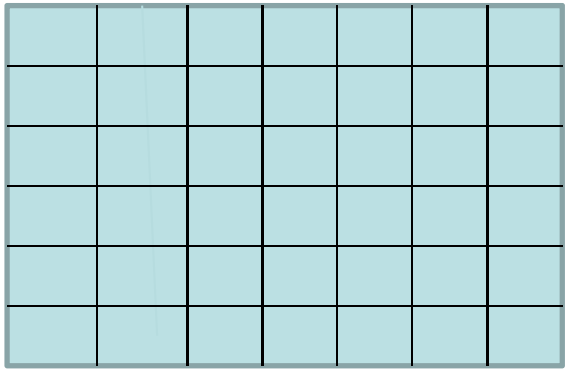
1. Build the block structure one block at the time
   - User responsible for block interfaces and interfaces to outer and inner hulls.
   - Similar to surface modeling without trimming
   - Can be template based
2. Design the trivariate block structure in an already existing ISO STEP type CAD model
   - 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

NURBS lack local refinement

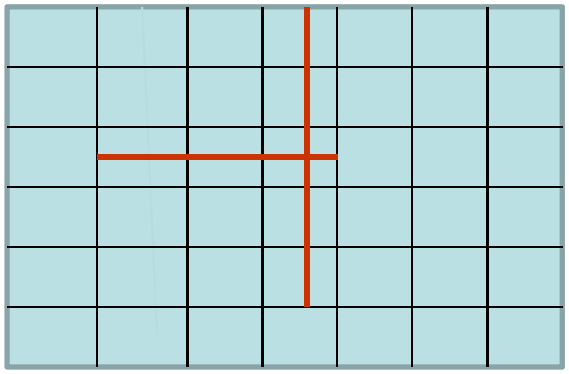
- The regular structure of tensor product NURBS does not allow local refinement
- 1988: Forsey & Bartels: Hierarchical B-spline refinement.
  - $f(s,t) = f_1(s,t) + f_2(s,t) + \ldots + f_n(s,t)$
  - The spline space of f_{i+1} is a refinement of the spline space of f_i
- 1998: Rainer Kraft, Adaptive 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
- 2010: Locally refined splines, addressing local refinement from the viewpoint of CAGD and Analysis

Local refinement 2-variate spline spaces

- Tensor product B-splines is an organization of 2-variate polynomial patches in a grid



- T-splines, PHT-splines and LR-splines address local refinements of these



Two basic approaches to local refinement of B-splines surfaces

Vertex grid refinement

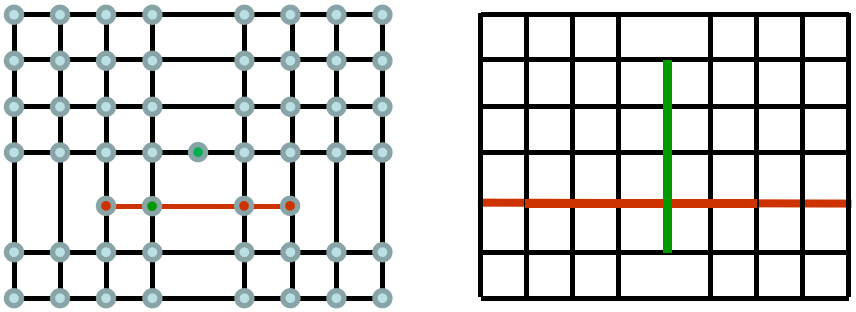
- Insert new vertices in existing vertex grid
  - The insertion has to adhere to a rule set
- Deduct spline space from vertex grid
- Example: T-splines

Spline space refinement

- Refinement by insertion of axis parallel segment of knot line (hyper plane)
  - The segment has to span the width of at least one tensor product B-spline basis function
- Deduct vertex grid from spline space
- Example: LR-splines

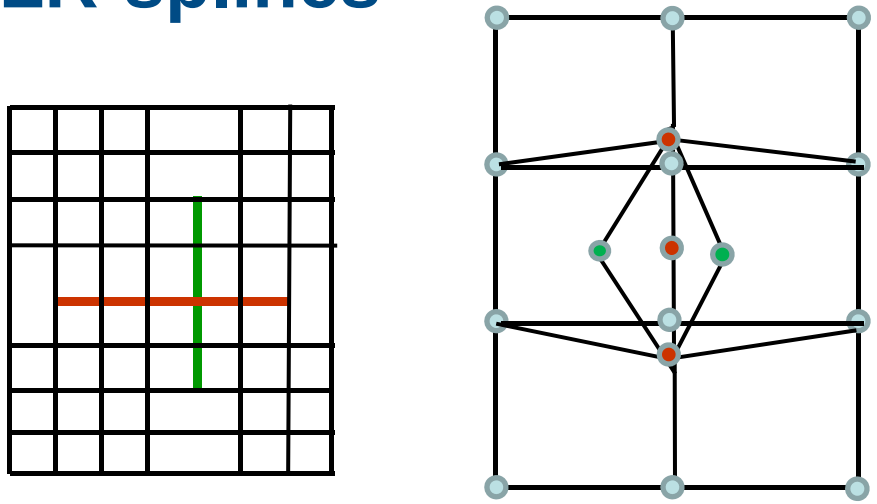
Comparison bicubic example

T-splines



- Adding two close vertices by T-spline refinement creates 11 additional polynomial segments and 5 vertices

LR-splines



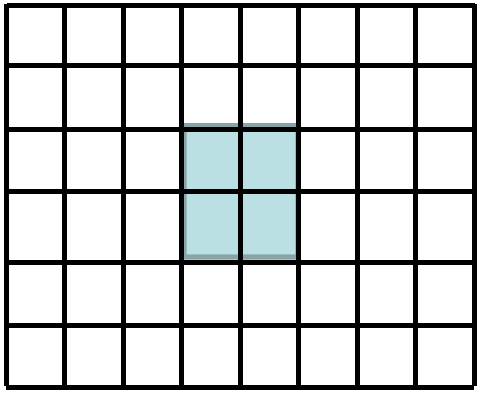
- Adding a minimal “+” structure by LR-splines creates 2 vertices and 8 new polynomial segments
- Position of vertices in parameter domain average of internal knots

The LR-spline method
2-variate example

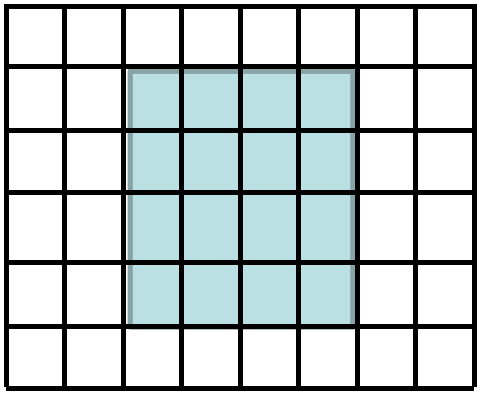
The starting point:

- A 2-variate tensor product B-spline with bi-degree (d_1,d_2)
- The knots defines a knot line grid
- Any B-spline basis function is defined by d_1+2 knots in the first direction and d_2+2 knots in the second direction

- $d_1=d_2=1$

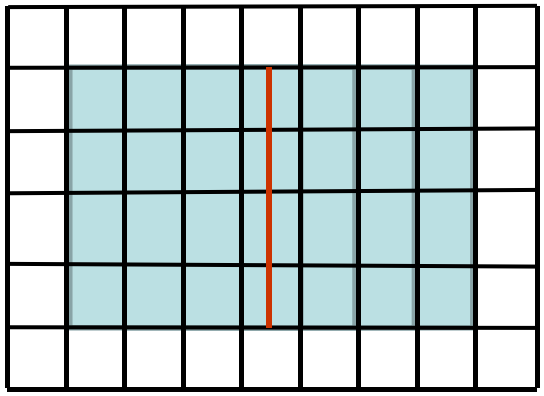


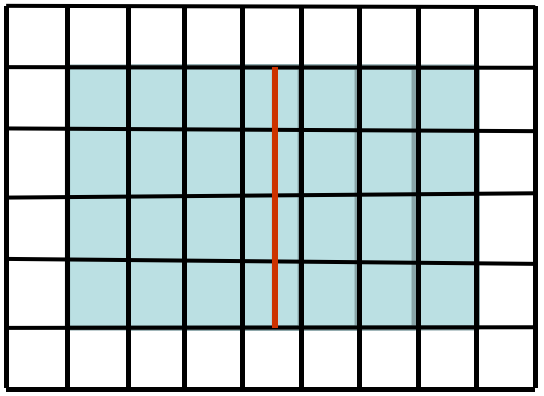
- $d_1=d_2=3$



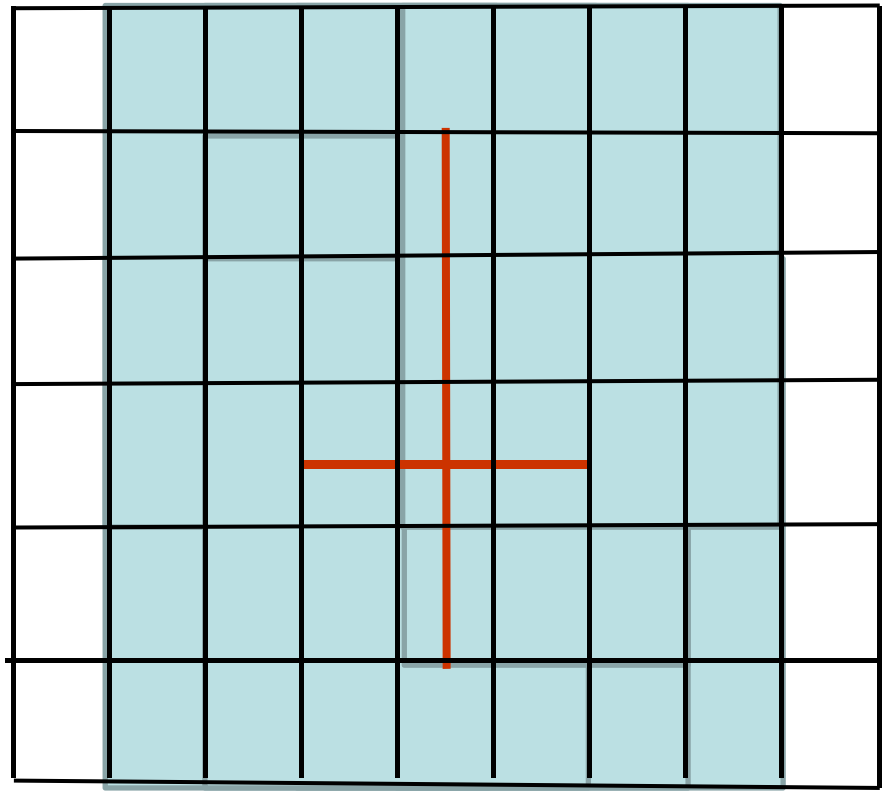
Refinement step

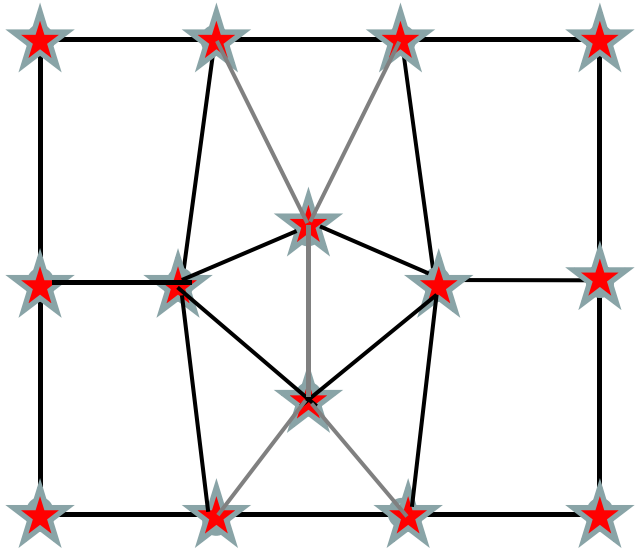
- Insert knot line segments that at least span the width of the basis functions
- Affected basis function
- Refined basis functions





The cubic example
B-spline basis functions and vertices





Handling the knot line segments

- Without loss of generality we index the knot vectors
  - The knot vectors of a B-spline basis function is thus two index sets one in each parameter direction
  - Refinements are described by two tupples from these index sets where either
    - (i_{min}, j) and (i_{max}, j), with $i_{min} < i_{max}$, or
    - (i, j_{min}) and (i, j_{max}), with $j_{min} < j_{max}$
  - The refinement is only valid if the line spans the width of at least on existing basis function.
- This can be generalised to higher dimension by regarding the refinement either as:
  - an axes parallel segment of a hyper plane,
  - or as a degenerate hyper- rectangle , e.g., one of the dimension of the hyper-rectangle is collapsed.

Augmented description for B-spline basis functions

- Traditionally in B-splines all knots are assumed to be used in a consecutive sequence,

$$B_{i,d}(x) = B(x \mid t_i, \ldots, t_{i+d+1}).$$

with d the degree.

- LR-splines need a more flexible notation

$$B_{\mathbf{i},\mathbf{t},d}(x) = B\left(x \mid \mathbf{t}(\mathbf{i}(0)), \ldots, \mathbf{t}(\mathbf{i}(d+1))\right),$$

where $\mathbf{i}(0), \ldots, \mathbf{i}(d+1)$ are pointers into the vector $\mathbf{t}$ containing all knot values in the given parameter direction

Bi-variate LR-splines

$$\mathbf{f}(x,y) = \sum_{(\mathbf{i},\mathbf{j})\in\mathcal{J}} \mathbf{c}_{\mathbf{i},\mathbf{j}}\, \phi_{\mathbf{i}}(x)\psi_{\mathbf{j}}(y), \text{ with } \mathbf{c}_{\mathbf{i},\mathbf{j}} \in \mathbb{P}^l, \mathbf{i},\mathbf{j} \in \mathcal{J}.$$

- Here $\mathcal{J}$ is and index set, and $l > 0$, and

$$\phi_{\mathbf{i}}(x) = B(\boldsymbol{i}, \boldsymbol{t}_1, d_1)(x) = B\left(x \mid \boldsymbol{t}_1\big(\mathbf{i}(0)\big), \ldots, \boldsymbol{t}_1\big(\mathbf{i}(d_1+1)\big)\right),$$

$$\psi_{\mathbf{j}}(y) = B(\boldsymbol{j}, \boldsymbol{t}_2, d_2)(y) = B\left(y \mid \boldsymbol{t}_2\big(\mathbf{j}(0)\big), \ldots, \boldsymbol{t}_2\big(\mathbf{j}(d_2+1)\big)\right).$$

- The provision $c_{\mathbf{i},\mathbf{j}} \in \mathbb{P}^l$ ensures that both rational and polynomial LR-splines are supported.
- Extensions to n-variate LR-splines is straight forward.

The refinement process

- The refinement starts from a source spline space $\mathbb{S}_0$, with a tensor product splines space as a natural choice.
- After a number of n refinements $\mathbb{S}_n$ (the refined spline space) is spanned by the set of tensor product B-spline basis functions $\mathcal{B}_n$.
- We use a refinement specification to select a set of basis functions $\mathcal{R}_n \subset \mathcal{B}_n$ to be refined.
- Then the specified refinement is performed by making $\mathcal{A}_n \supset \mathcal{R}_n$ a set of scaled tensor product B-spline basis functions
- The set of basis functions $\mathcal{B}_{n+1}$ spanning $\mathbb{S}_{n+1}$ can thus be described by

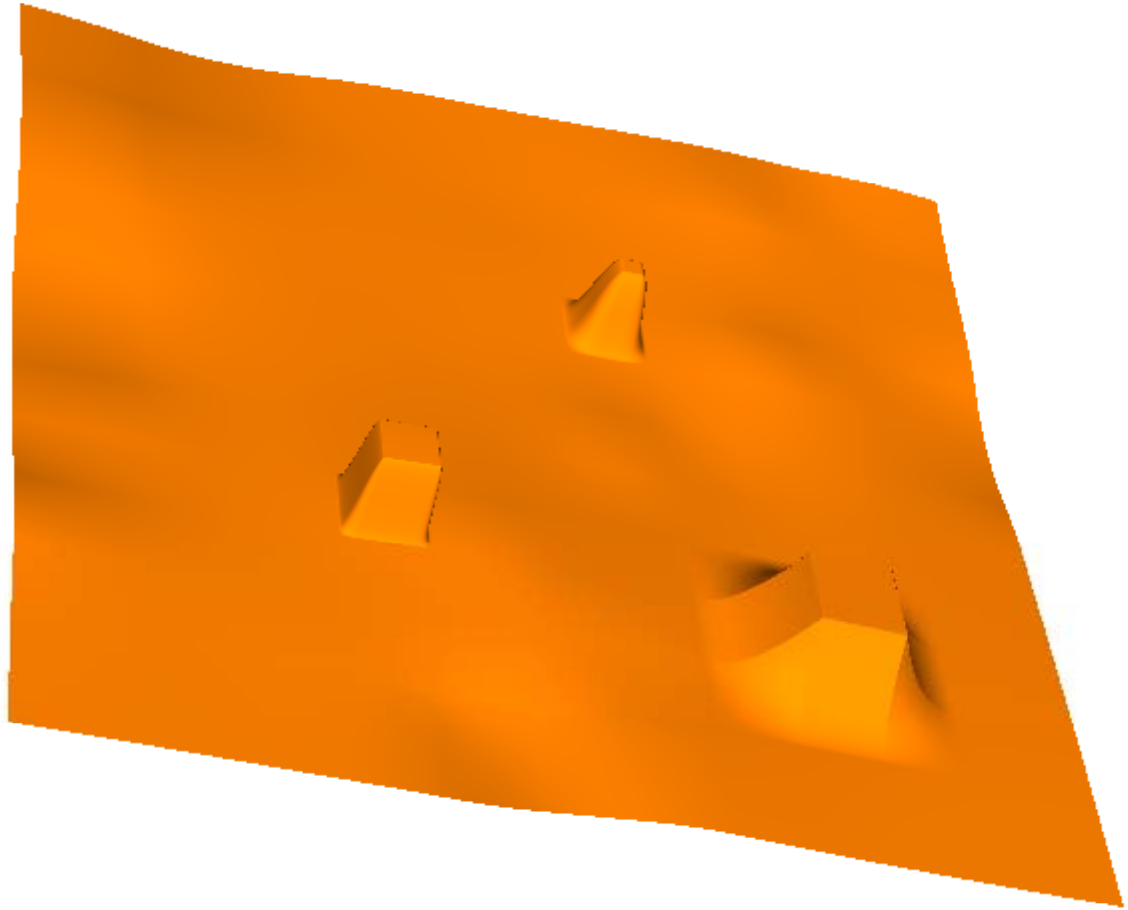
$$\mathcal{B}_{n+1} = \mathcal{B}_n \cup \mathcal{A}_n \backslash \mathcal{R}_n.$$

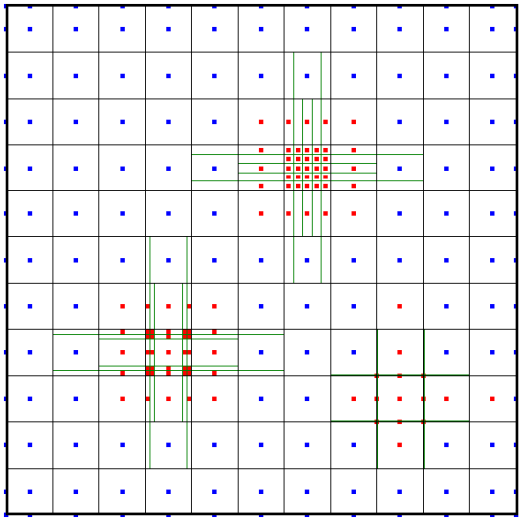
Properties of LR-splines

- Partition of unity, basis function sum to 1 if basis of source space $\mathbb{S}_0$ is partition of unity.
- Linearly independet basis functions - Proved
- Interpolation – no Schoenberg Witney theorem until now telling where to select points to be used in interpolation.
- Marsden's identity
  - Reproduction of polynomials imposed by LR-spline construction
  - Challenging to make closed expression
- Use standard B-spline methods
  - Evaluation, Cox de Boor
  - Refinement, Oslo Algorithm
- Refinement done by inserting knot line segments
  - Symmetric - Independent of refinement sequence

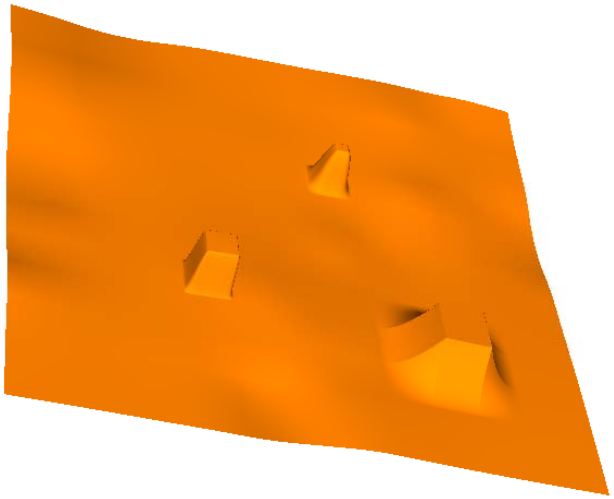
	T-splines	PHT-splines	LR-splines
Degrees	3 (Odd)	3	All
Continuity	C^2 (Degree -1)	C^1	All
Independent of refinement sequence	No	Yes	Yes
Basis functions	Piecewise Rational	To be constructed case by case	Piecewise Polynomial
How local is the refinement	Typically 2x minimal influence	Minimal influence area	Minimal influence area
Linear independent basis	No guarantee	Constraints have to be added	Yes
Control grid for 2 parametric surface in 3D	Yes, logical structure	To some extent	Promising , deducted from spline space
Coding	In 3D control grid	In the spline space	In the spline space
Interrelations	Disjoint to PHT-splines	Disjoint to T-splines	Includes T-splines and PHT-splines
Higher dimensions	Challenging?	Yes	Yes
Elementary shapes	Yes, by rational parametric representation		

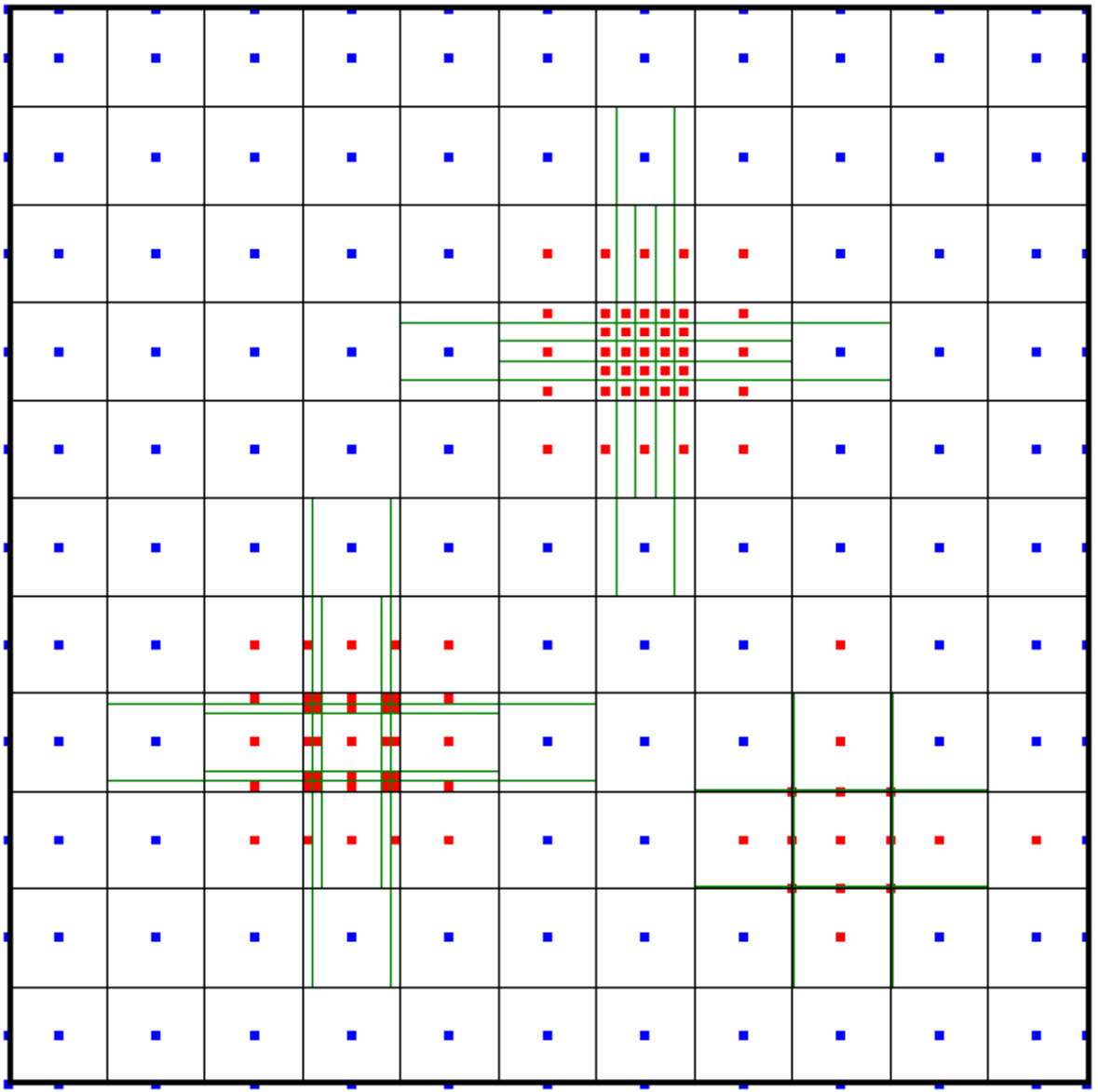
Biquadratic surface with 3 refinements



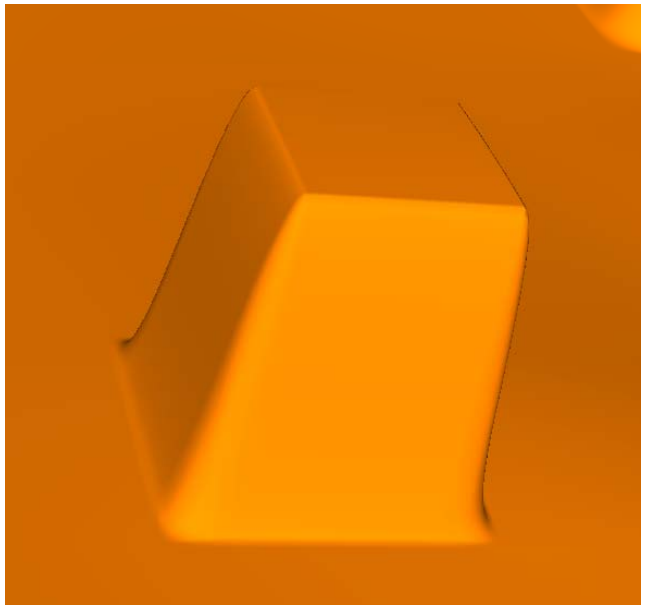


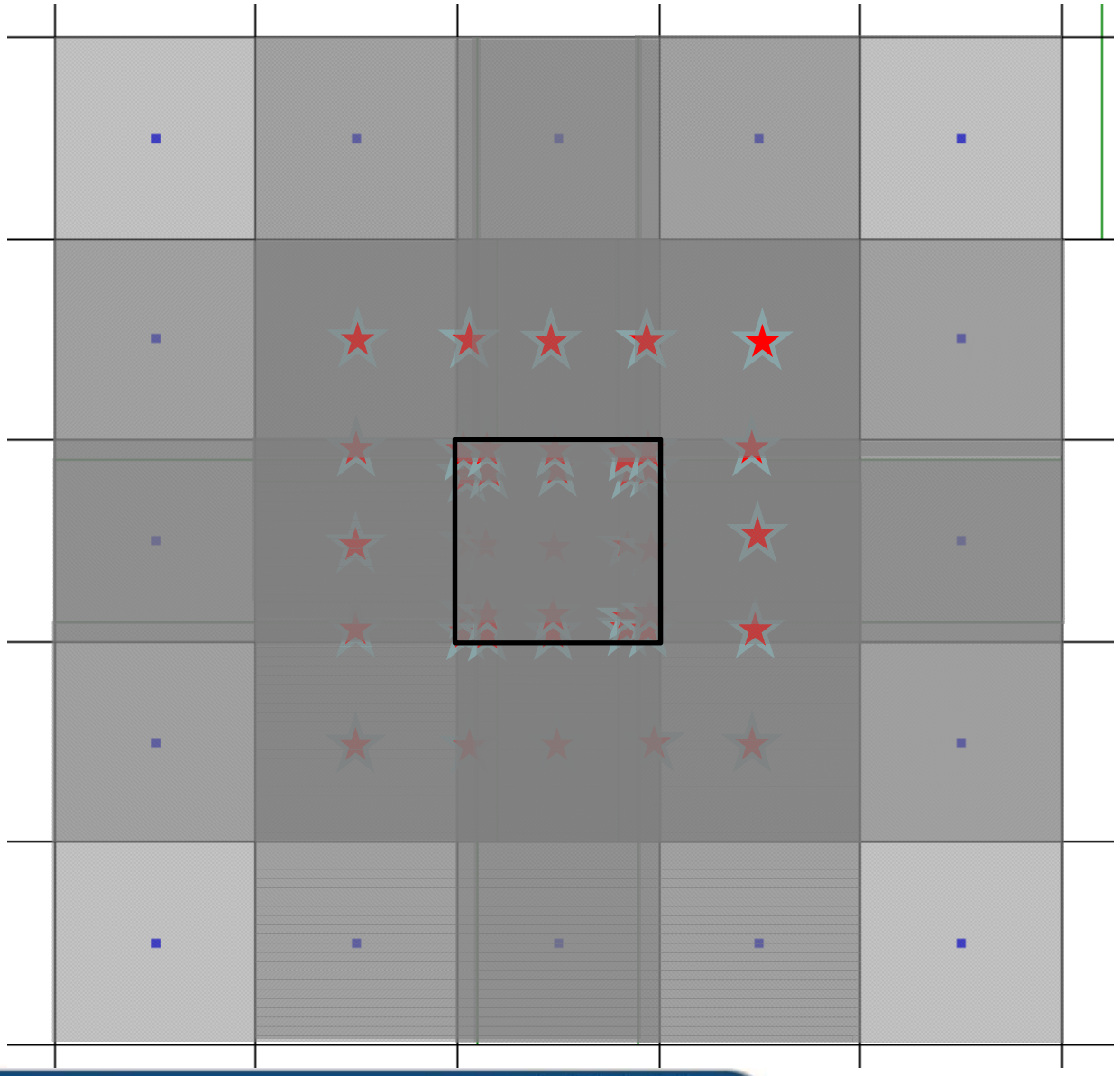
Biquadratic surface with 3 refinements



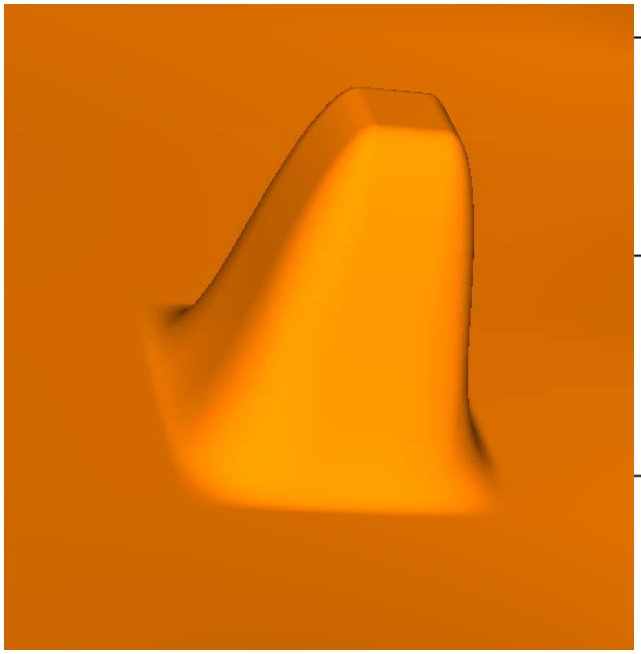


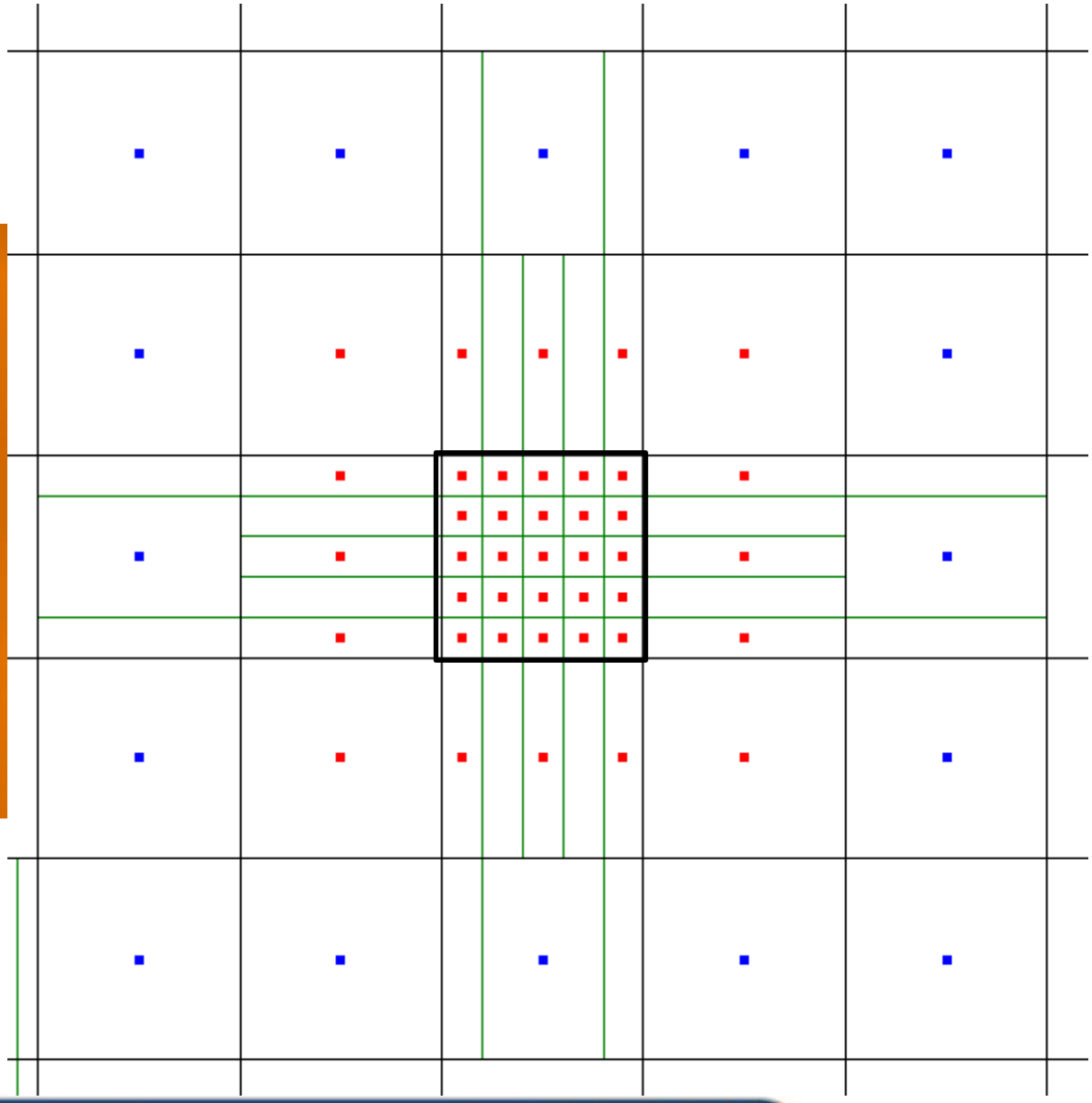
Refinement 1 – C^1



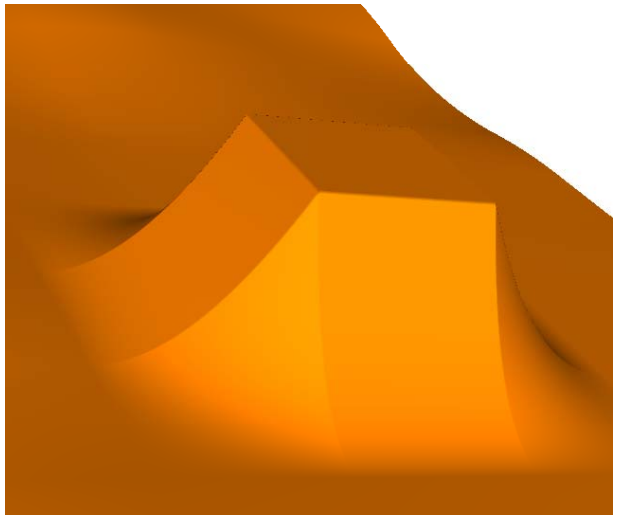


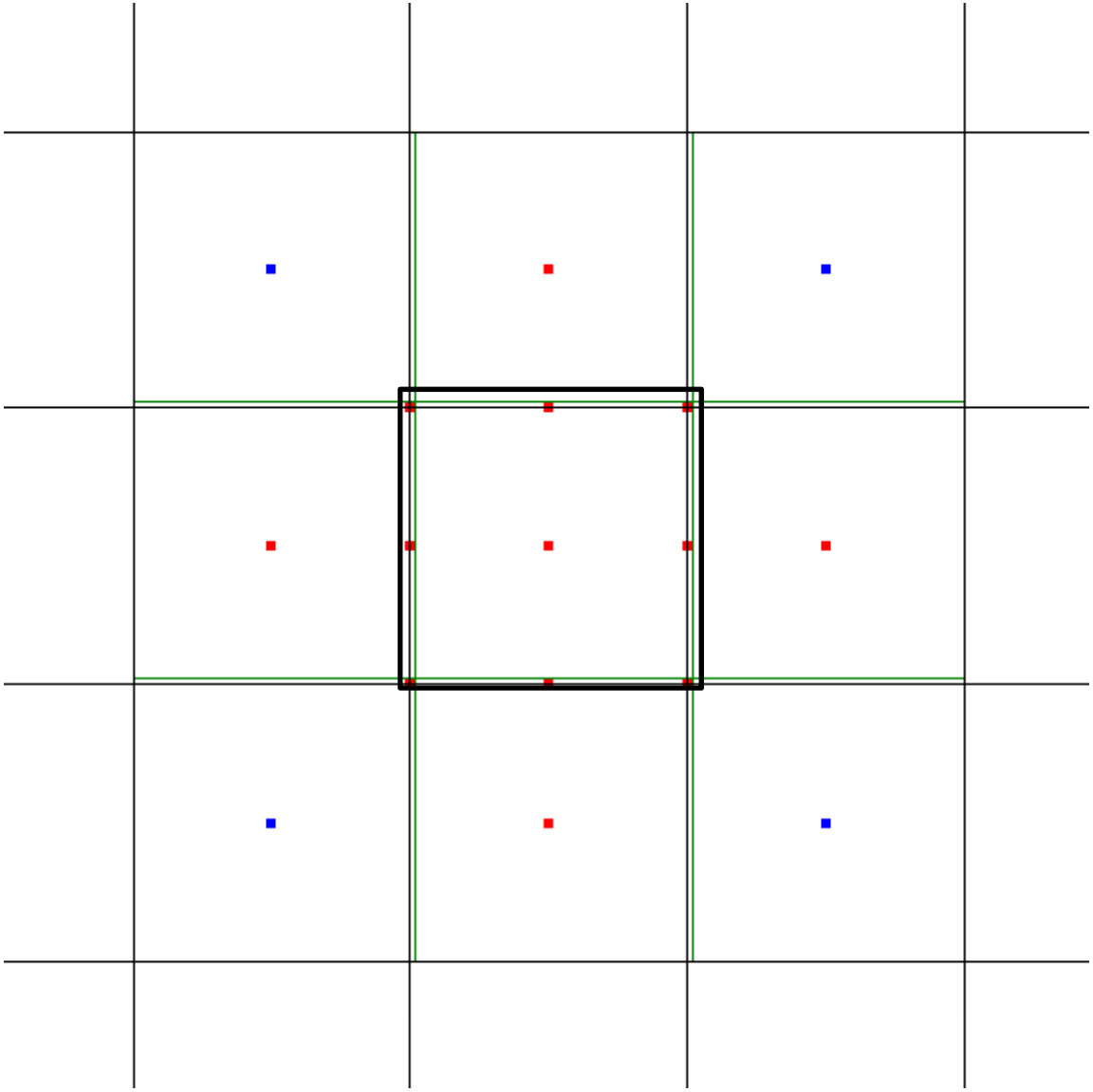
Refinement 2 – C^1



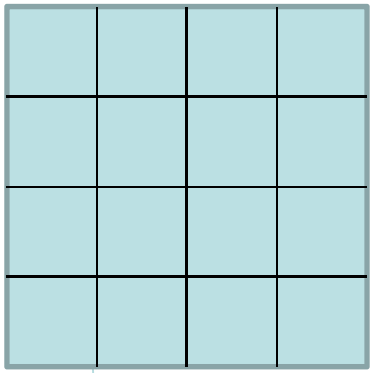


Refinement 3 – C^0

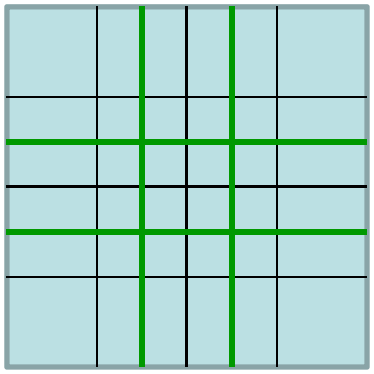


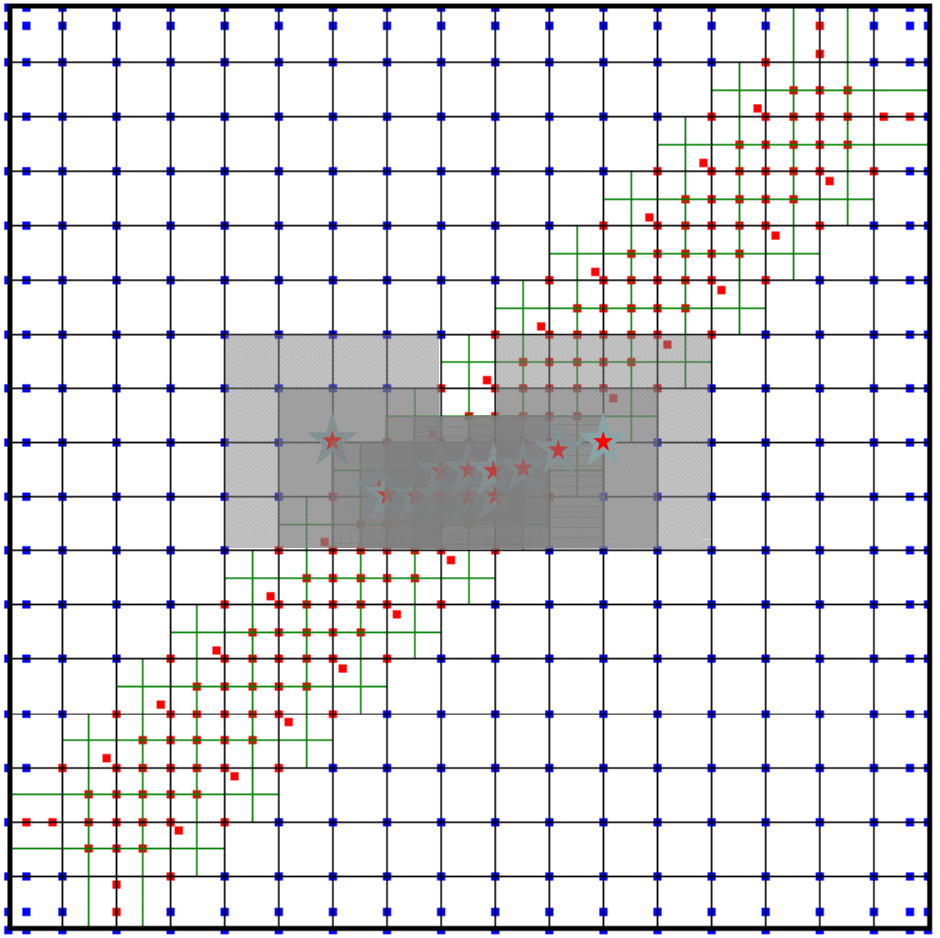


Refinement 1 – split bicubic B-spline basis function into 4



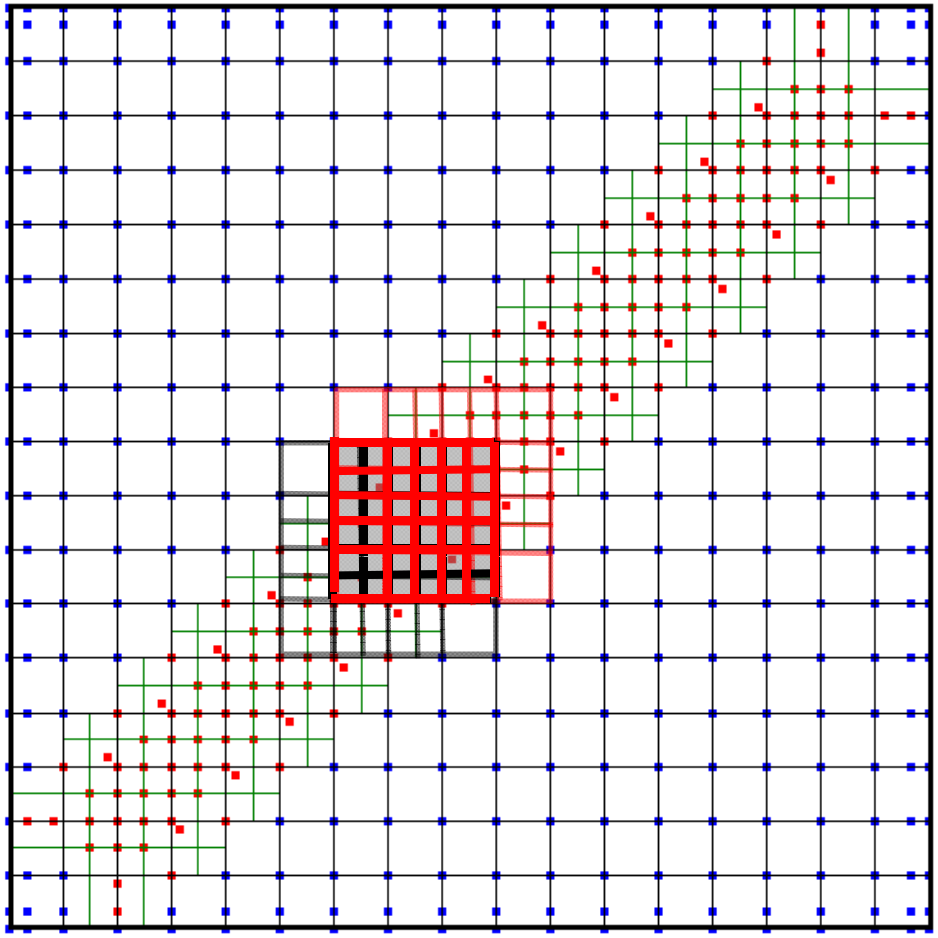
Add two knot segment in both direction



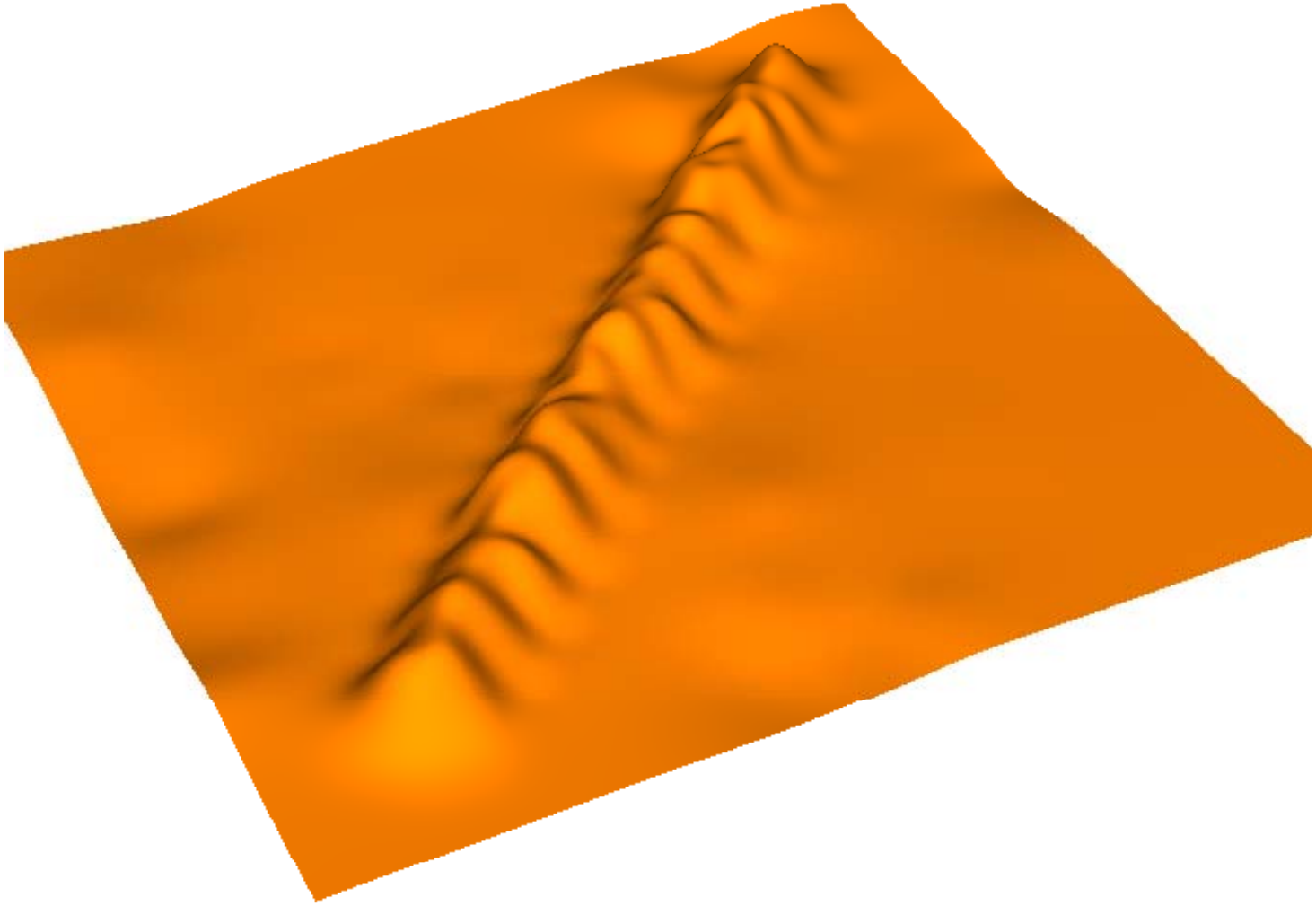


Refinement 1 – split bicubic B-spline basis function into 4

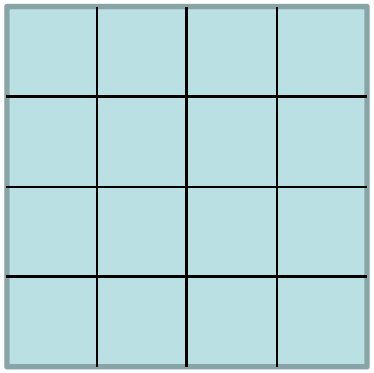
- Dots are the Greville coordinates of a basis function (average of internal knots)
- Blue dots mark basis functions surviving refinement
- Read dots mark basis functions generated by the refinement
- No interference of refinement of neighbouring basis functions giving linear dependencies



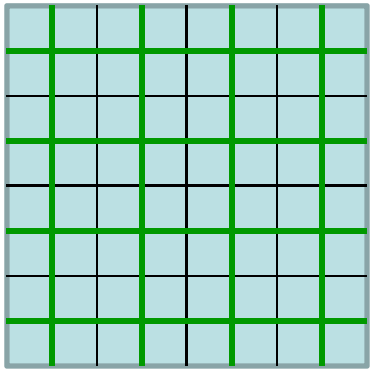
Added random values to coefficients of refined basis functions

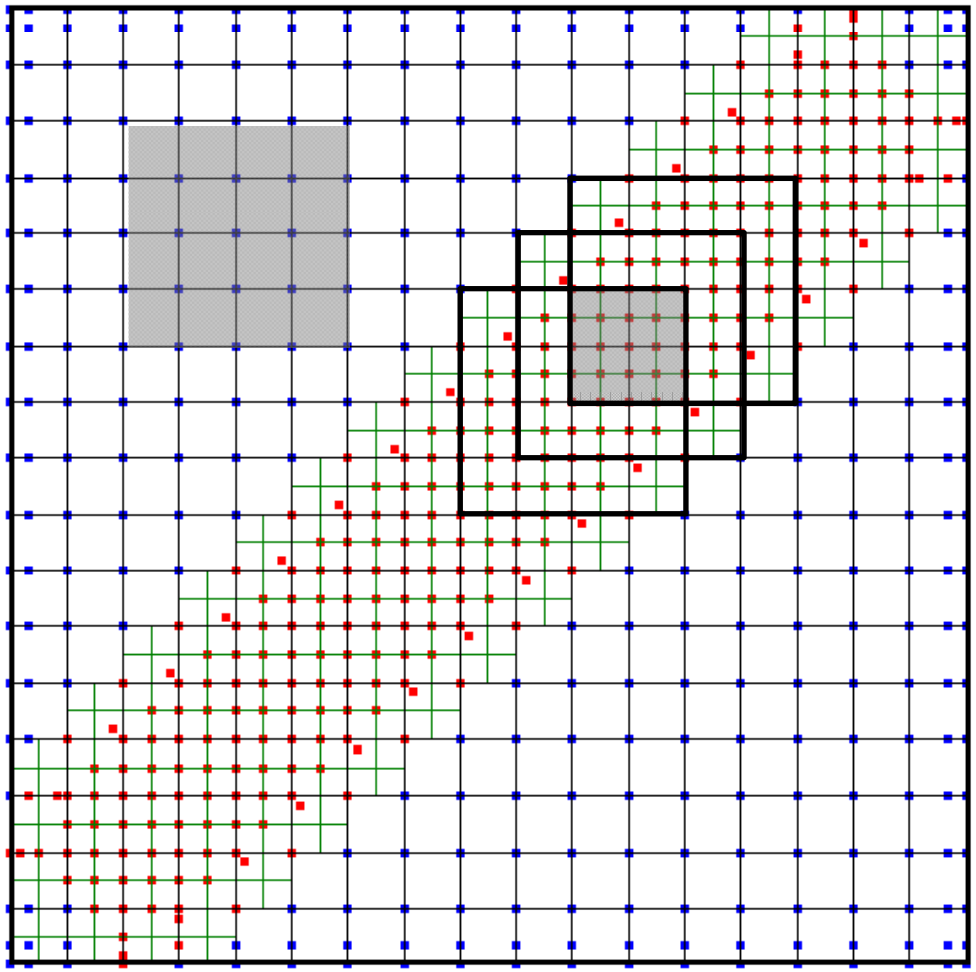


Refinement 2 – split bicubic B-spline basis function into 9

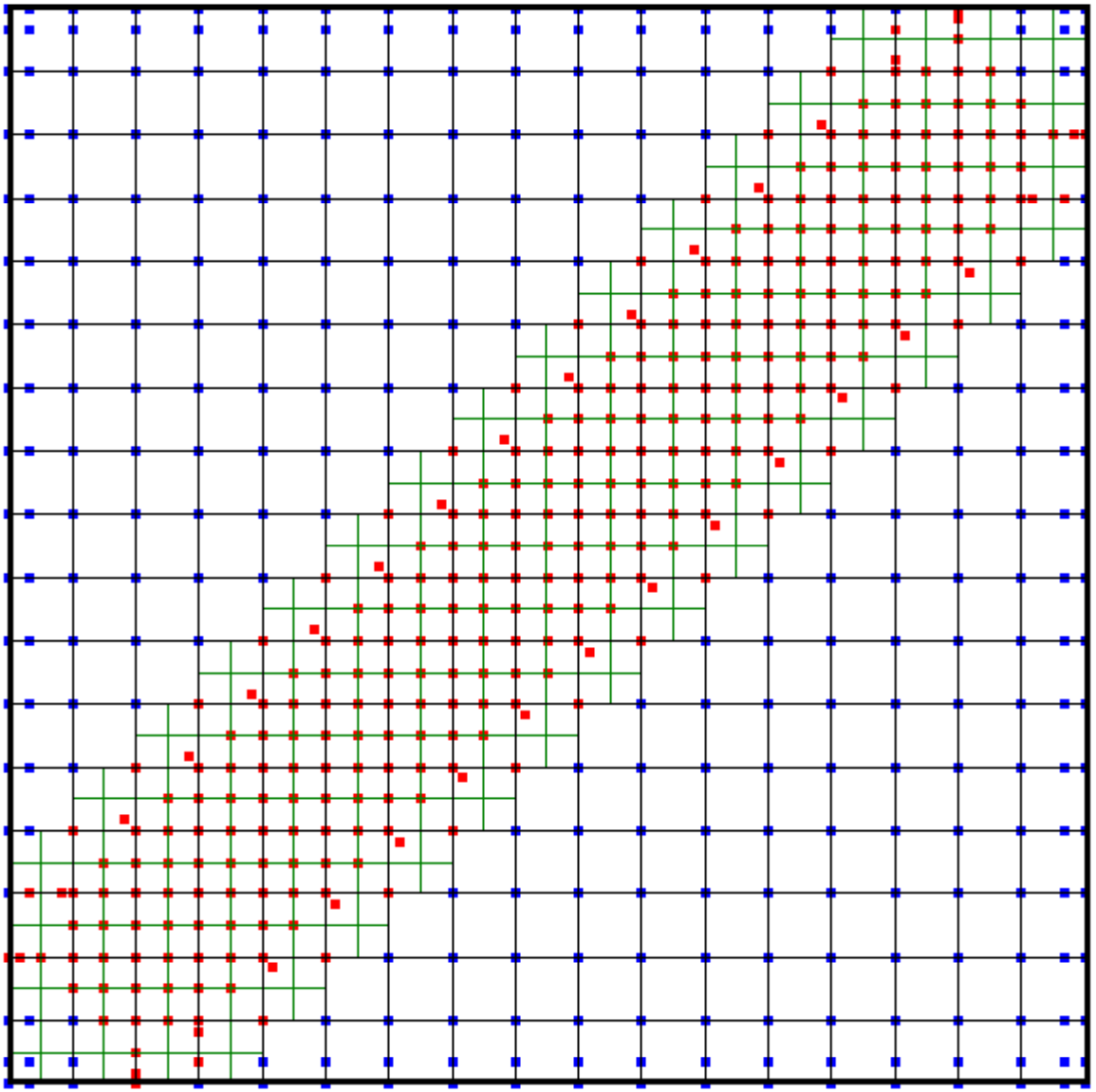


Add two knot segment in both direction

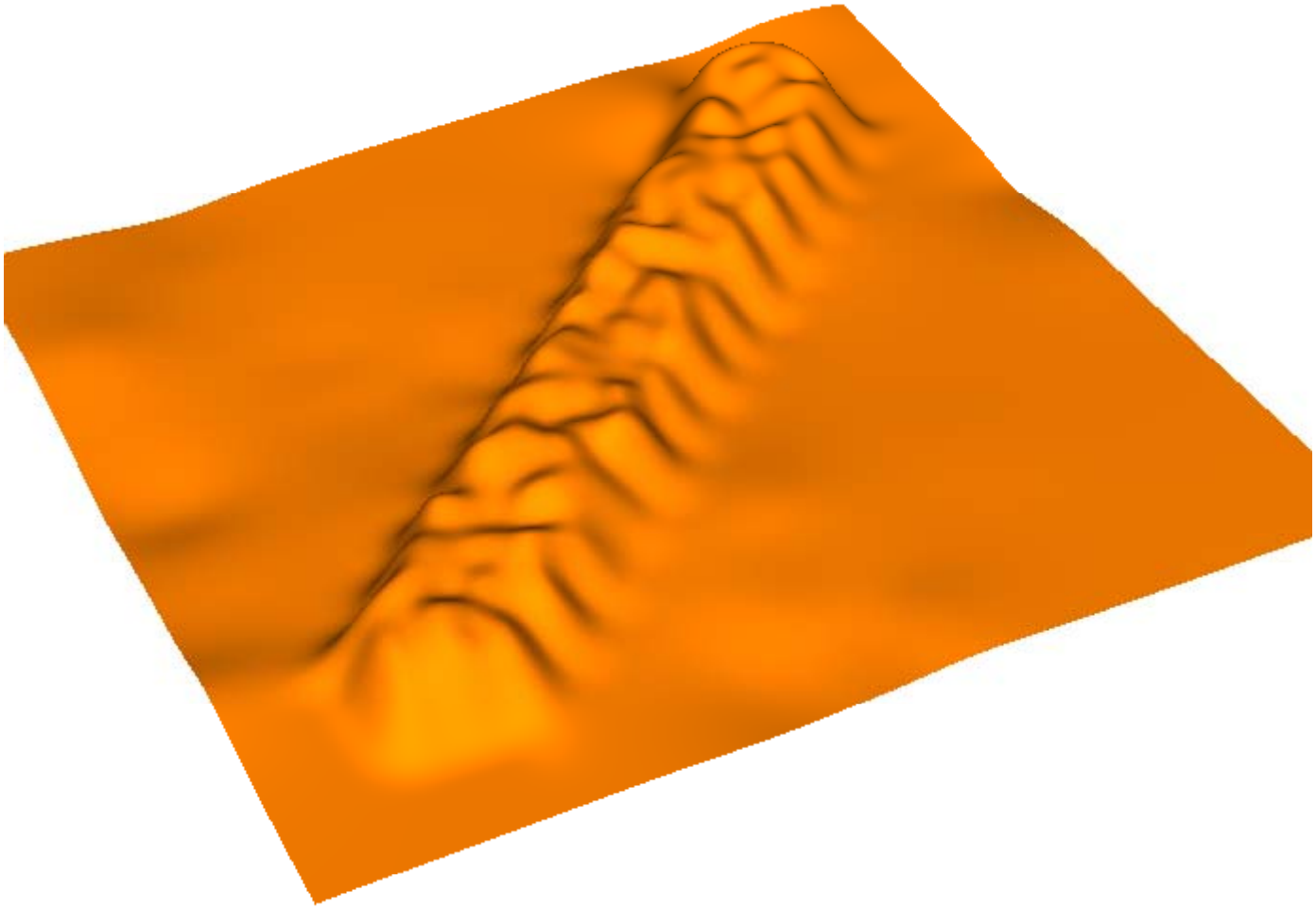




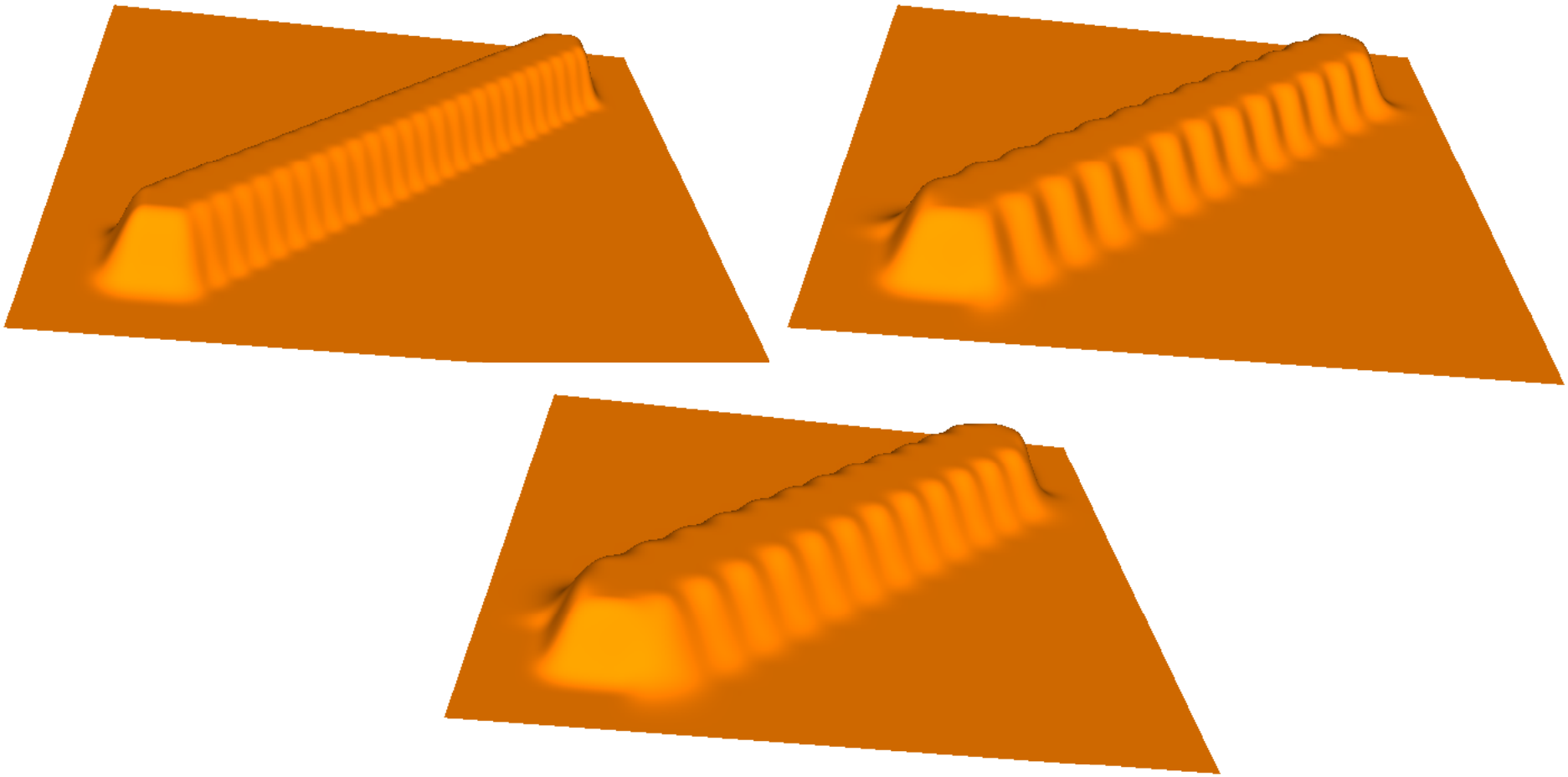
- Dots are the Greville coordinates of a basis function (average of internal knots)
- Blue dots mark basis functions surviving refinement
- Read dots mark basis functions generated by the refinement
- Interference of refinement of neighbouring basis functions sorted out



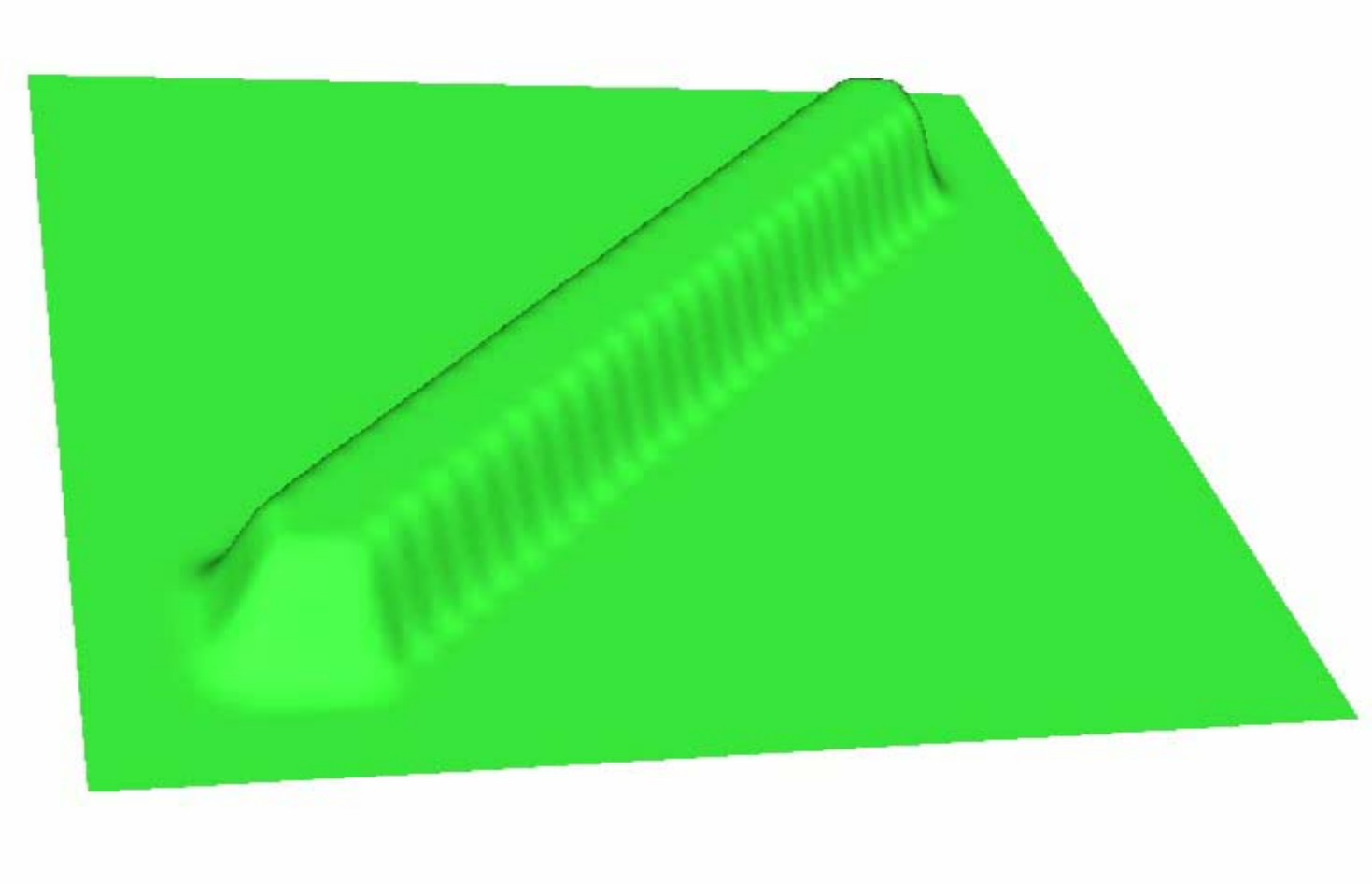
Added random values to coefficients of refinement



Constant value of original and constant values of select refinement



Outline of previous from above



LR-splines is planned to be an extensions to the GNU GPL package GoTools from SINTEF

- GoTools originally supported CAD-type NURBS (one and two variate)
- GoTools recently extended with volumetric NURBS (trivariate) to support isogeometric analysis
- LR-spline refinement of GoTools NURBS surface now being implemented
- LR-spline refinement of GoTools NURBS volumes planned

We have to believe in change to achieve change!

- CAD people have believed that it is impossible to change the interface between CAD and FEA
- FEA people have believed that it is impossible to change the interface between CAD and FEA
- If we act as it is possible to achieve change we can actually achieve change
  - Let us demonstrate the possibilities of isogeometric analysis
  - Let us promote the idea in academia and to industry
  - Let us generate a demand for drastically improved interoperability of CAD and FEA