Construction of polynomial spline spaces over quadtree and octree T-meshes

M. Brovka[(1)], J.I. López[(1)], J.M. Escobar[(1)], J.M. Cascón[(2)] and R. Montenegro[(1)]*

[(1)] University Institute SIANI, University of Las Palmas de Gran Canaria, Spain

[(2)] Department of Economics and History of Economics, University of Salamanca, Spain

23rd International Meshing Roundtable (IMR23)

October 12–15, 2014, London, UK

MINECO y FEDER Project: CGL2011-29396-C03-00

CONACYT-SENER Project, Fondo Sectorial, contract: 163723

http://www.dca.iusiani.ulpgc.es/proyecto2012-2014

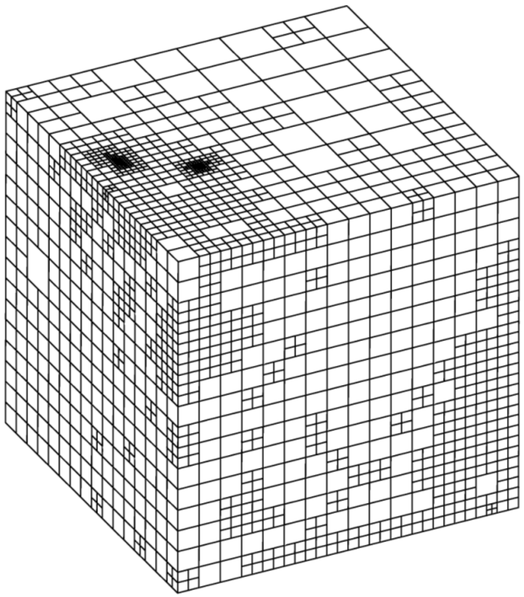
Contents

- Motivation and summary of our previous works
  - T-spline parameterization of 2D and 3D geometries
  - Application to isogeometric modeling and analysis
- Polynomial spline spaces over quadtree and octree T-meshes
  - The main three steps of the new technique in 2D and 3D
  - Straightforward implementation in 2D and 3D
  - Application to isogeometric modeling and analysis
- Comments and future research

T-spline Parameterization

For a given T-mesh (0-balanced quadtree/octree)

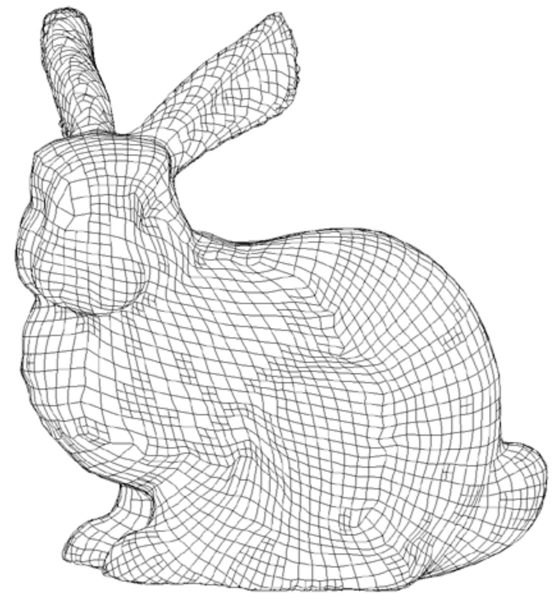
Objective question: Is it possible to define cubic polynomial blending functions?



T-mesh

Parameter space

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\alpha \in A} \mathbf{P}_\alpha \, R_\alpha(\boldsymbol{\xi})$$



T-spline

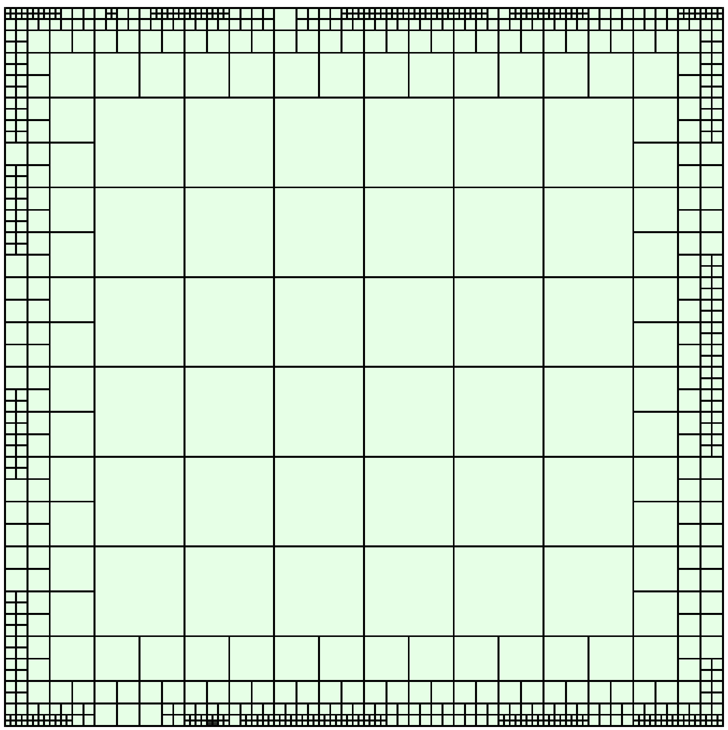
Physical space

The Meccano Method on T-meshes in 2-D

Input data: Boundary representation of the object

Objective: Construction of a high quality T-spline parameterization

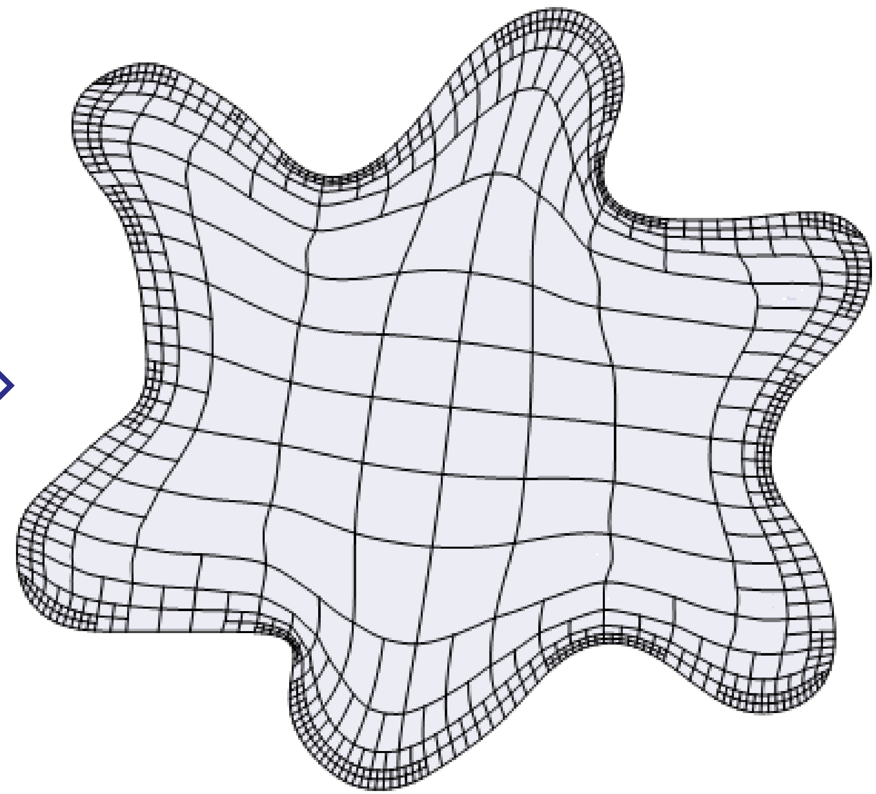
T-mesh



Parameter space

$\mathbf{S}(\boldsymbol{\xi})$

T-spline mesh



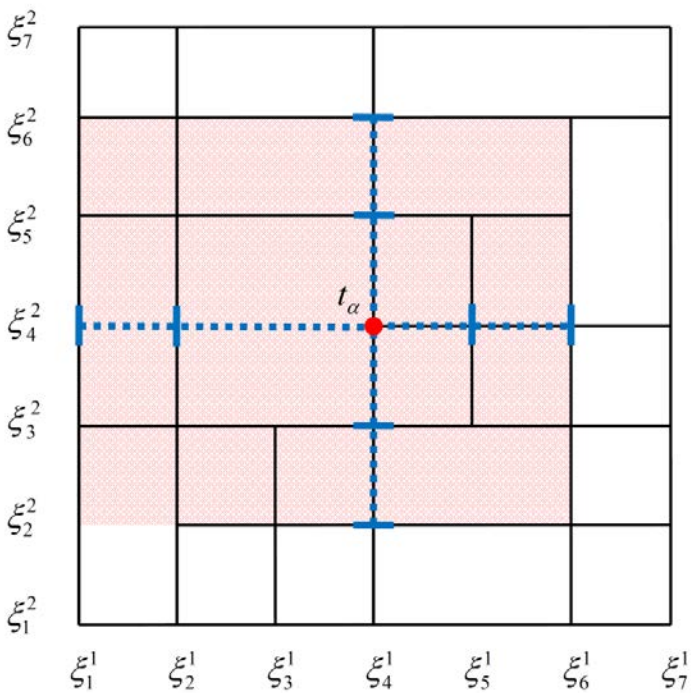
Physical space

T-spline Basis Functions in 2-D

Example of a bivariate cubic basis function on a T-mesh

T.W. Sederberg (rational blending functions with local support)

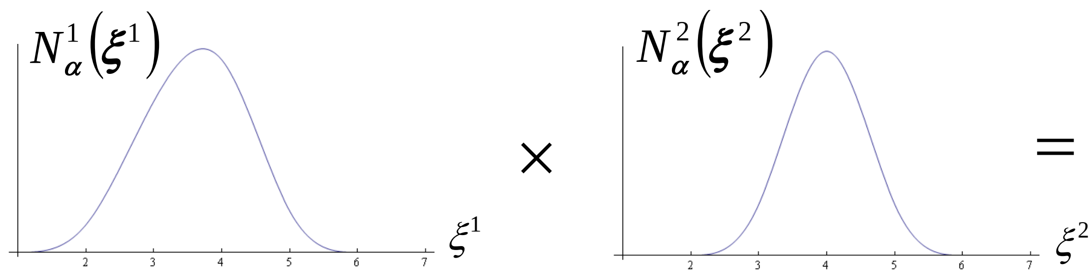
T-mesh and anchor t_a



Knots associated to anchor t_a

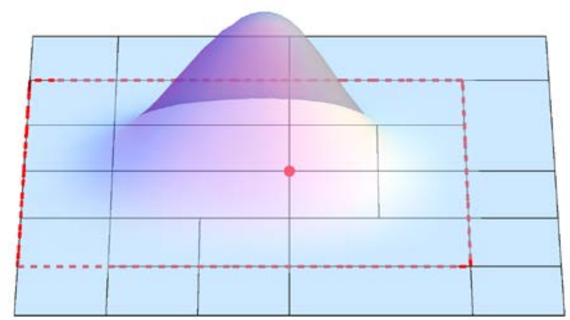
(obtaining by traversing T-mesh edges):

$$\Xi_\alpha^1 = \left\{\xi_1^1, \xi_2^1, \xi_4^1, \xi_5^1, \xi_6^1\right\} \qquad \Xi_\alpha^2 = \left\{\xi_2^2, \xi_3^2, \xi_4^2, \xi_5^2, \xi_6^2\right\}$$



Bivariate Cubic T-spline Basis Function

$$B_\alpha\left(\xi^1, \xi^2\right) = N_\alpha^1\left(\xi^1\right) N_\alpha^2\left(\xi^2\right)$$

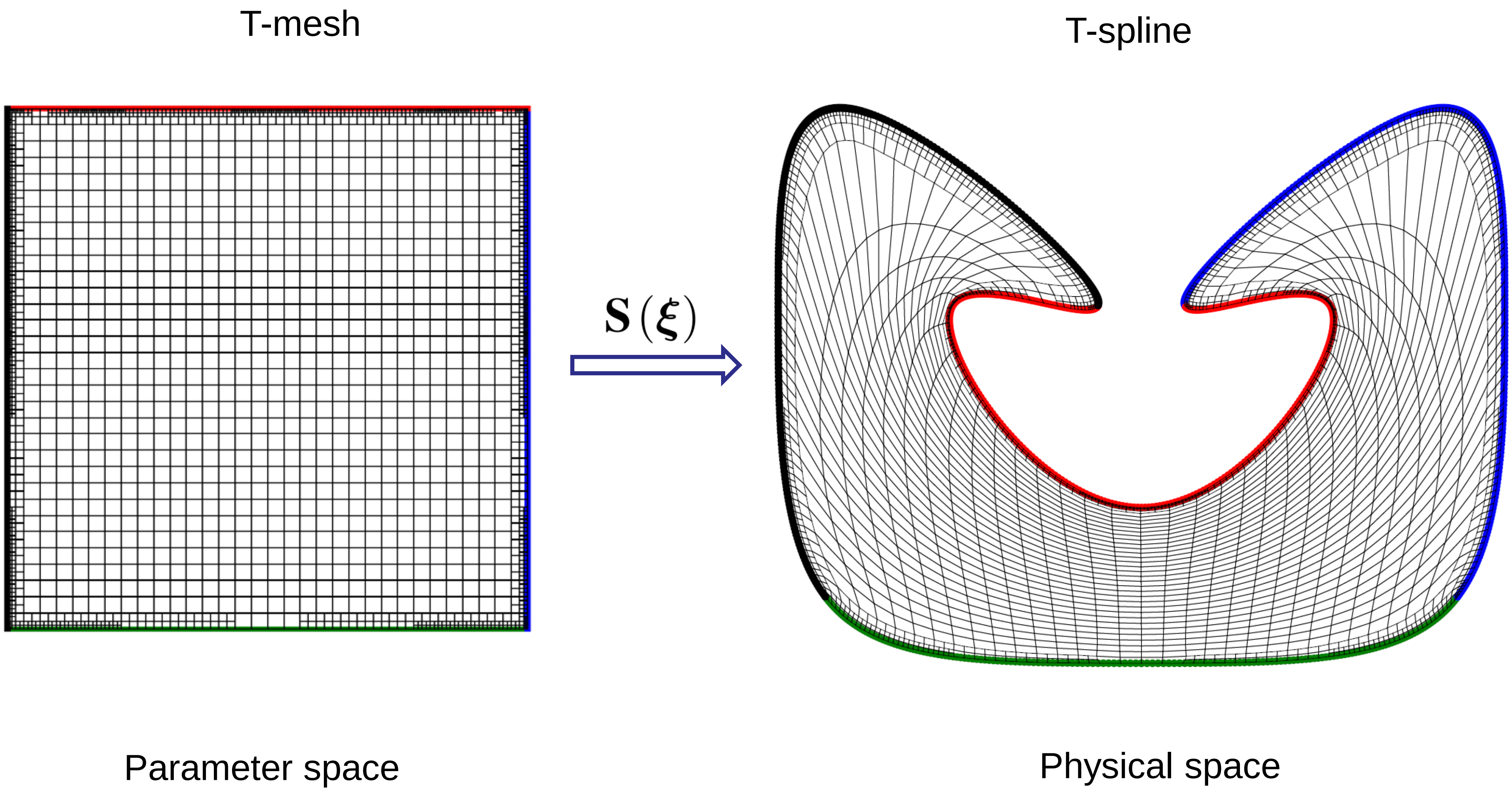


Rational blending function

$$R_\alpha(\xi) = \frac{w_\alpha B_\alpha(\xi)}{\sum_{\beta \in A} w_\beta B_\beta(\xi)}$$

Simultaneous Untangling and Smoothing of T-meshes

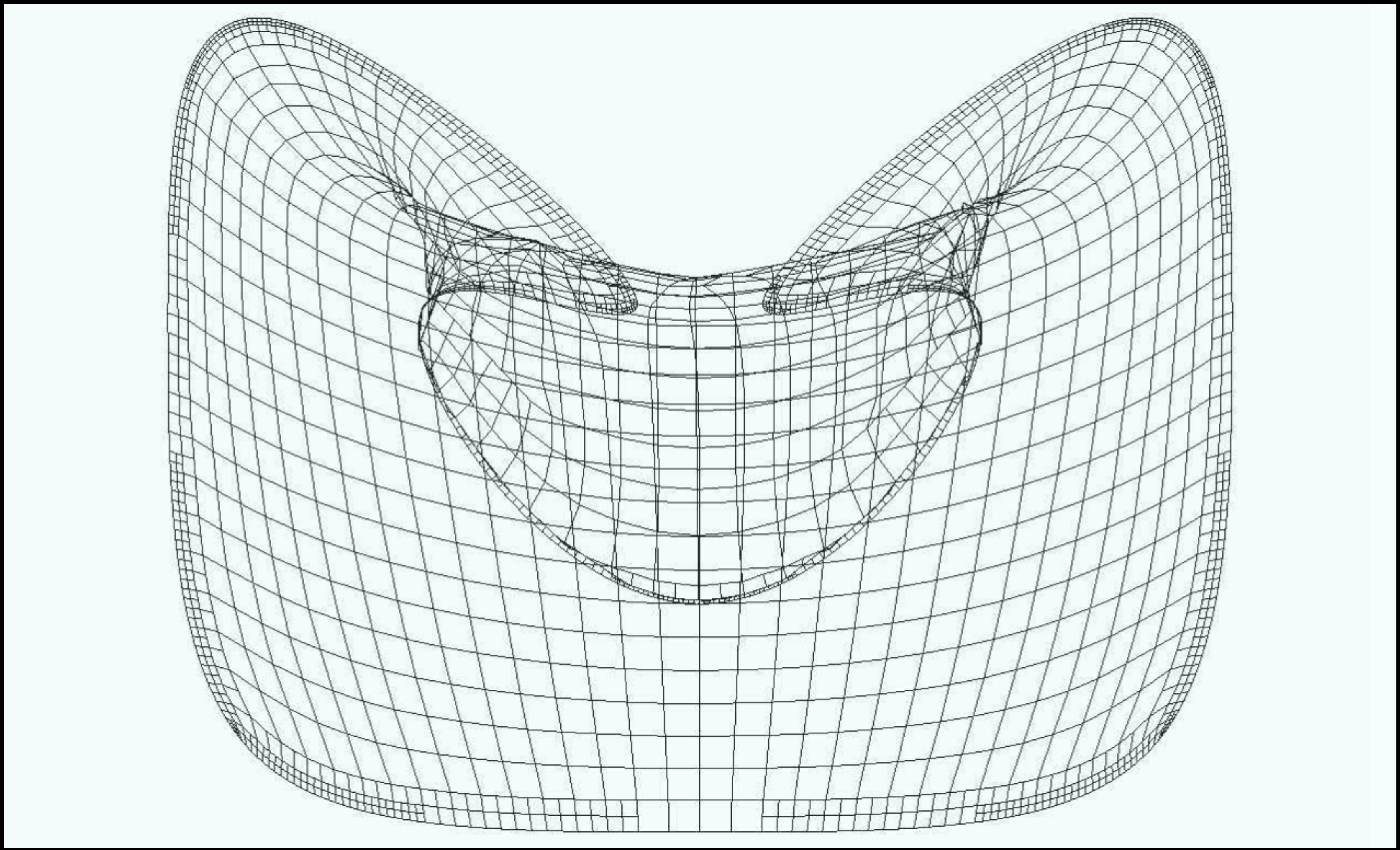
T-mesh transformation along the SUS process: Example



Simultaneous Untangling and Smoothing of T-meshes

T-mesh transformation along the SUS process: Video

SIANI

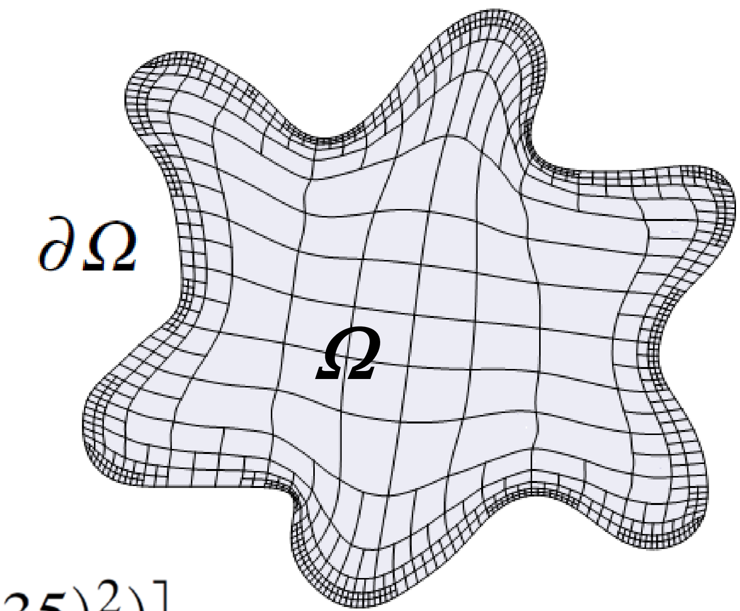


Local Nested Adaptive Refinement

Numerical solution of a Poisson problem

Concentrate source in relation to the initial mesh size

$$
\begin{aligned}
-\triangle u &= f && \text{in } \Omega, \\
u &= g && \text{on } \partial\Omega.
\end{aligned}
$$



Exact solution:

$$u(x,y) = \exp\left[-10^3\left((x-0.6)^2 + (y-0.35)^2\right)\right]$$

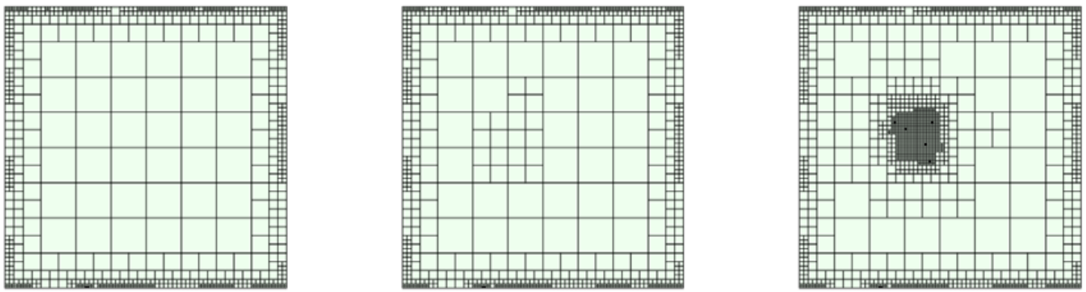
Residual-type error indicator:

$$\eta(\Omega_e)^2 = \|h\,(f+\Delta u_h)\|^2_{0,\Omega_e} = \int_{\Omega_e} h^2\,(f+\Delta u_h)^2\,\mathrm{d}\Omega$$

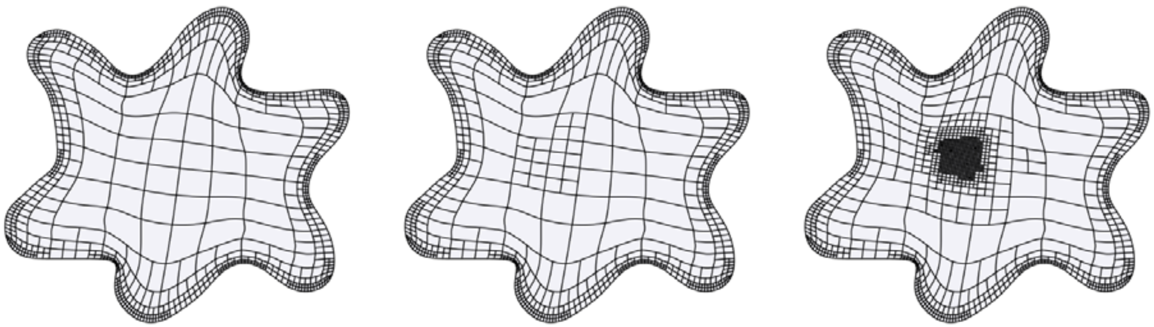
Local Nested Adaptive Refinement

Numerical solution of a Poisson problem

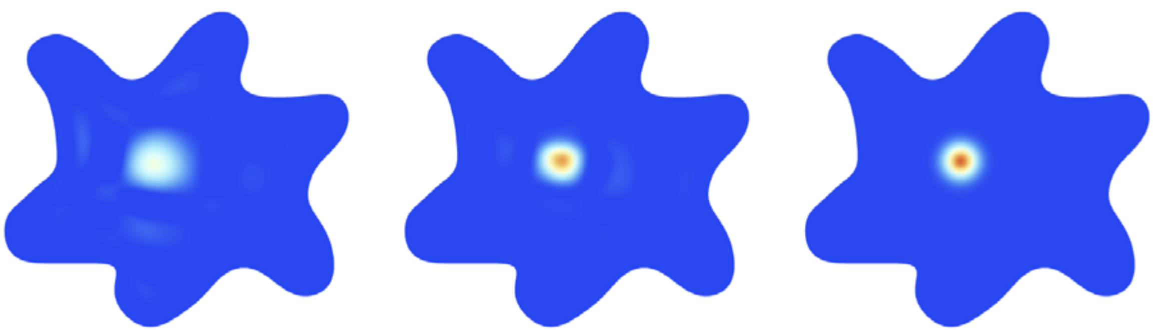
Error indicator : $\eta\,(\Omega_e)^2 = \int_{\Omega_e} h^2\,(f + \Delta u_h)^2\; d\Omega$



(a) 844 cells,1456 DOF (b) 859 cells, 1476 DOF (c) 1552 cells, 2233 DOF

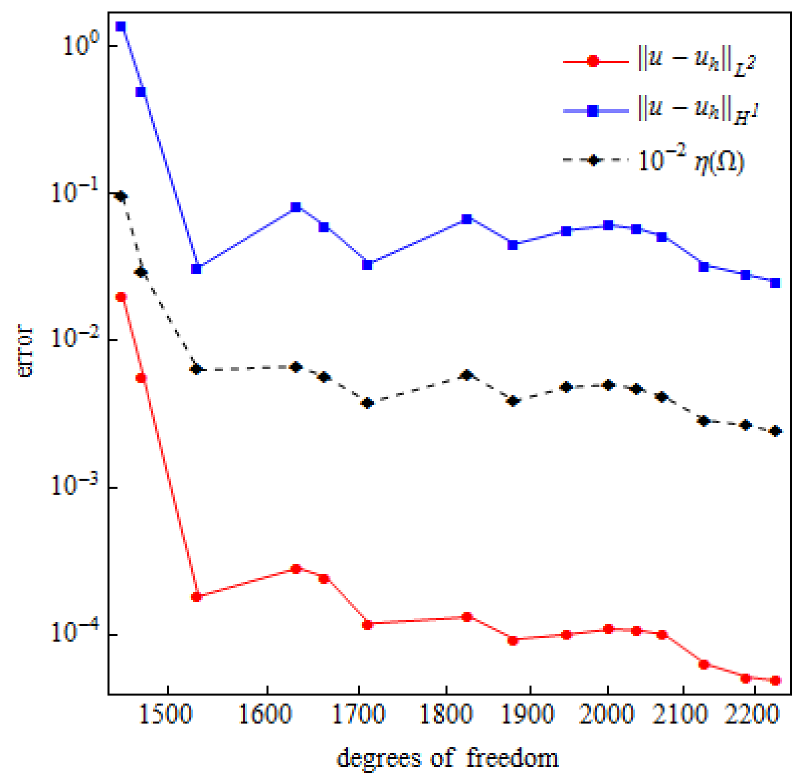


(d) Initial mesh (e) 1-st refinement (f) 14-th refinement



(g) Initial solution (h) 1-st refinement (i) 14-th refinement

Non-monotonous convergence behavior



Polynomial spline spaces over any quadtree and octree T-meshes

SIANI

A new strategy for constructing tensor product cubic spline spaces over arbitrary quadtree (2D) and octree (3D) T-meshes

- For a given T-mesh, it allows to obtain a set of cubic spline functions that span a space with nice properties: C^2 continuous, nested spaces, linear independence
- Simple rules for inferring local knot vectors to define spline blending functions
- Straightforward implementation in 2D and 3D

Three main steps

1. Mesh pretreatment: 0-balancing
2. Inferring local knot vectors
3. Modification of local knot vectors

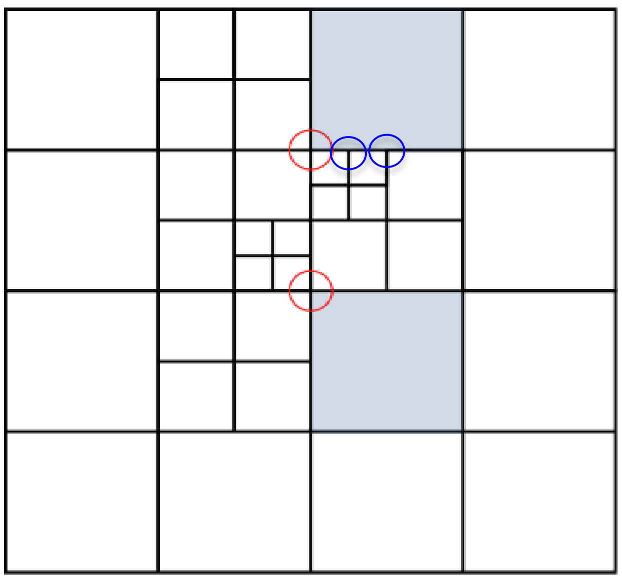
Strategy for defining polynomial spline spaces

Step 1 - Mesh pretreatment

0-balancing

- ❑ Only one hanging node (T-junction) per edge
- ❑ Any node is shared by cells that at most belong to two consecutive levels

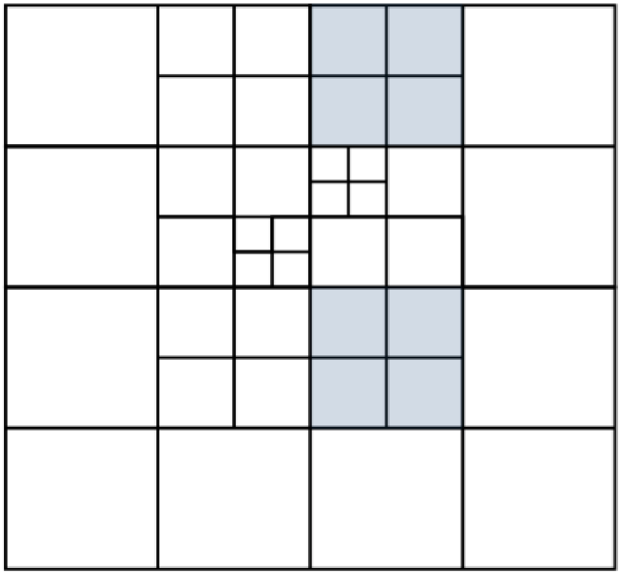
Unbalanced quadtree



Standard procedure

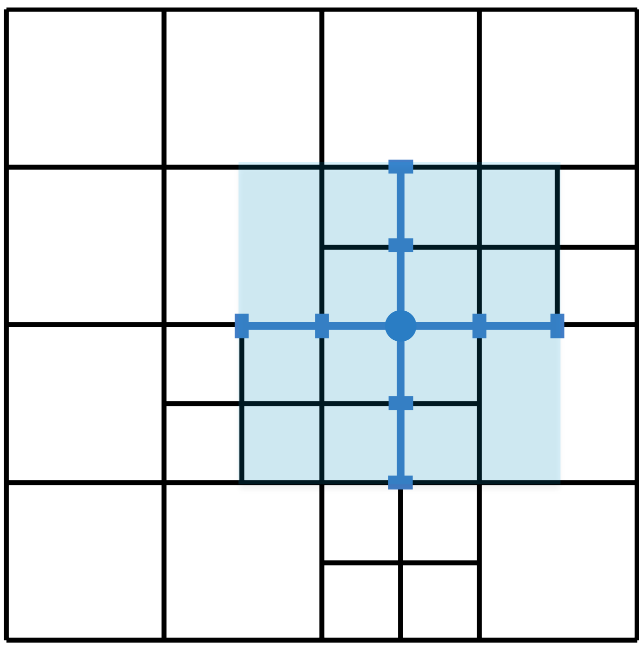
(refinement)

0-balanced quadtree

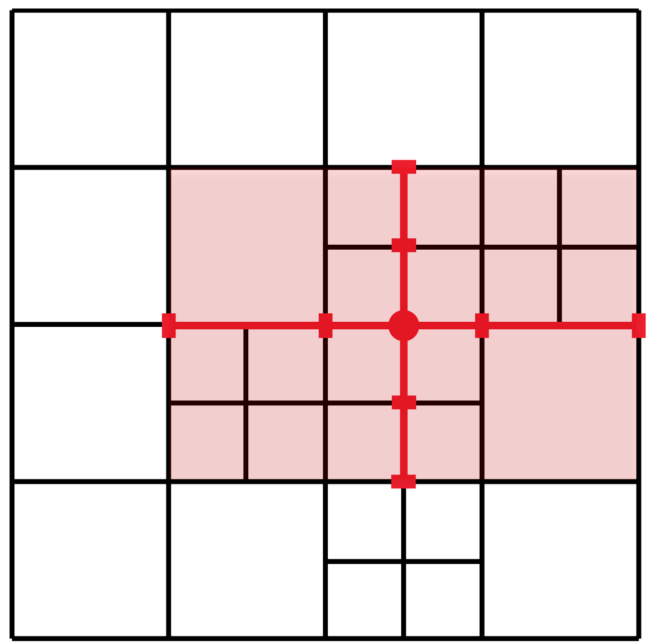


Inferring local knot vectors

- Blending functions are associated **only to regular nodes**
- Local knot vectors are inferred by traversing T-mesh edges
- Hanging nodes are skipped when “walking” along its surrounding edge



Original T-spline (Sederberg)

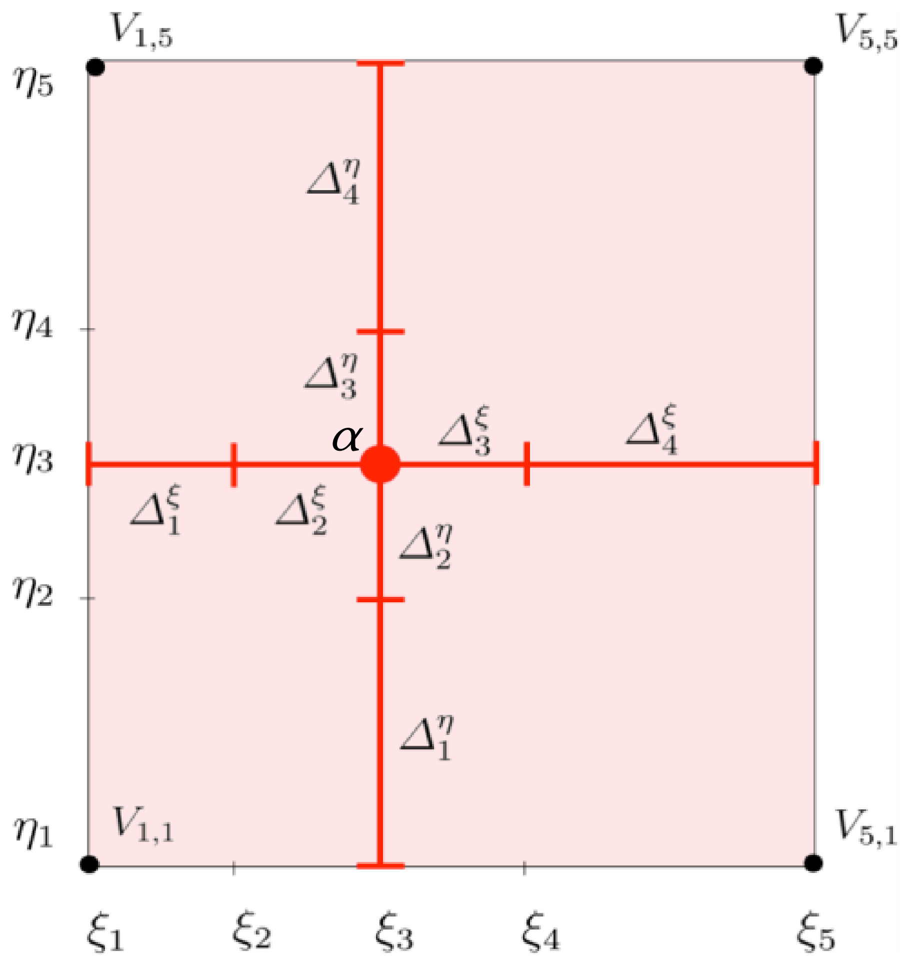


New strategy

Strategy for defining polynomial spline spaces

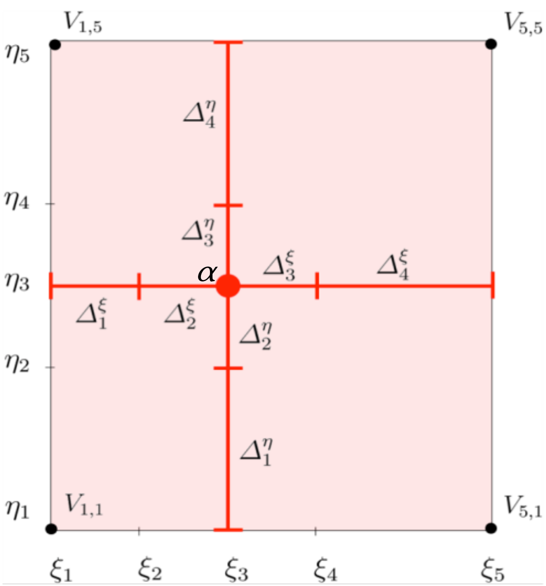
Step 3 - Modification of local knot vectors

Support of the blending function N_α associated to the anchor α



Modification of local knot vectors

Knot vectors of each blending function N_α have to fulfill two conditions:



Condition 1: Local knot vectors of the d-variate function N_α verify:

$$\Delta_1^j \geqslant \Delta_2^j = \Delta_3^j \leqslant \Delta_4^j, \quad j = 1, \ldots, d$$

Condition 2: The frame of the function support should be situated over the mesh skeleton:

$$\operatorname{frm}(\operatorname{supp} N_\alpha) \in \operatorname{skt}(T)$$

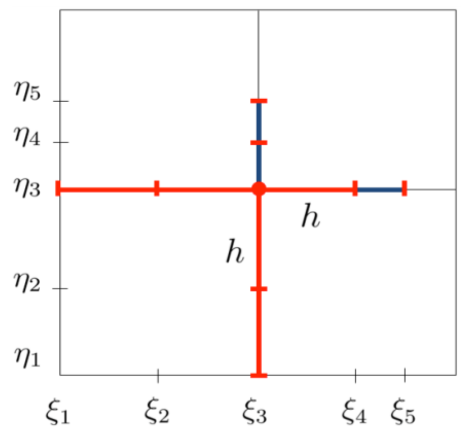
Simple extension rule: If any condition is not verified, then some Δ_i^j are duplicated

Strategy for defining polynomial spline spaces

Step 3 - Modification of local knot vectors (Examples)

Condition 1 is **not satisfied**

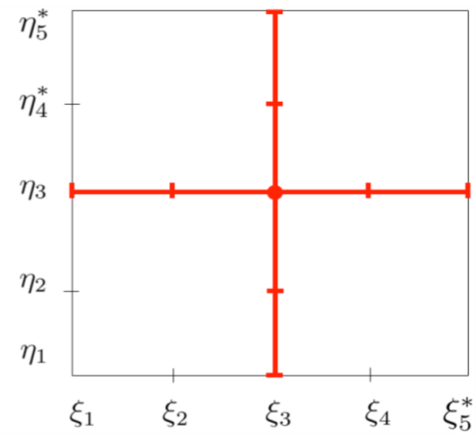
$$\Delta_4^{\xi} < \Delta_3^{\xi} \text{ and } \Delta_3^{\eta} < \Delta_2^{\eta}$$



$$\xi_5^* \leftarrow \xi_3 + 2h$$

$$\eta_4^* \leftarrow \eta_3 + h$$

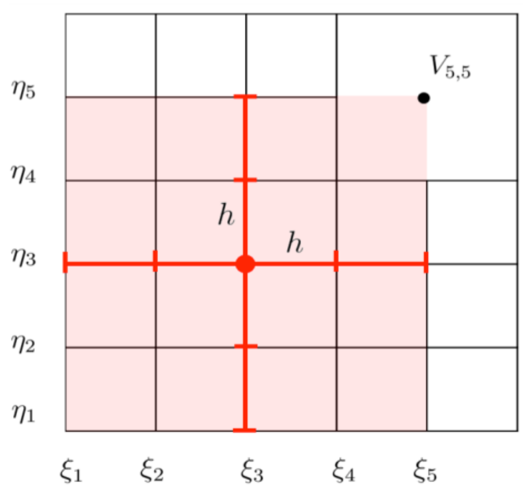
$$\eta_5^* \leftarrow \eta_3 + 2h$$



$$h = \max(\Delta^{\xi}_2 , \Delta^{\xi}_3) = \max(\Delta^{\eta}_2 , \Delta^{\eta}_3)$$

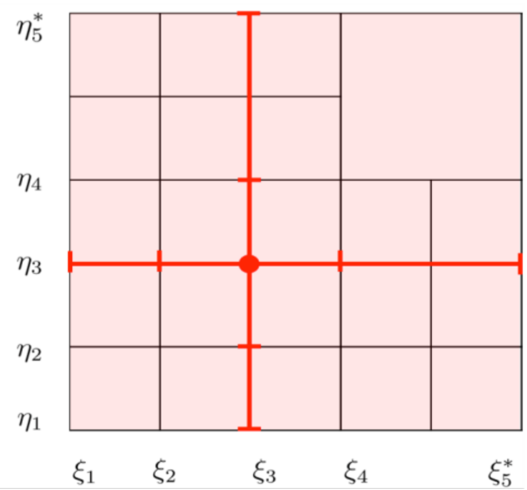
Condition 2 is **not satisfied**

$$V_{5,5} \notin skt(T)$$



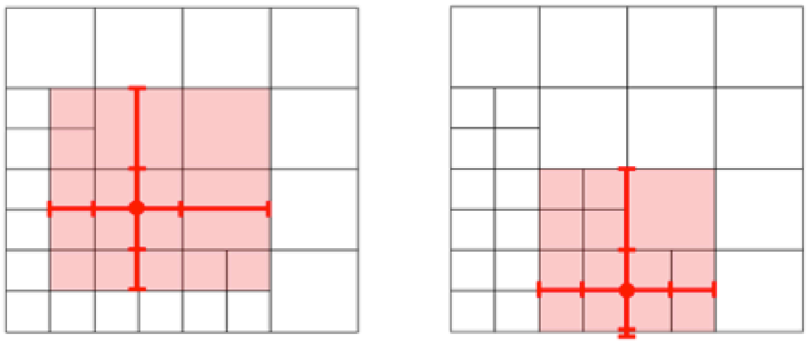
$$\xi_5^* \leftarrow \xi_3 + 3h$$

$$\eta_5^* \leftarrow \eta_3 + 3h$$

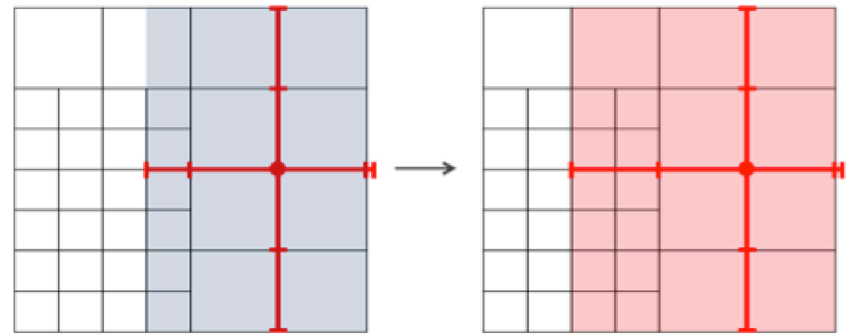


Strategy for defining polynomial spline spaces

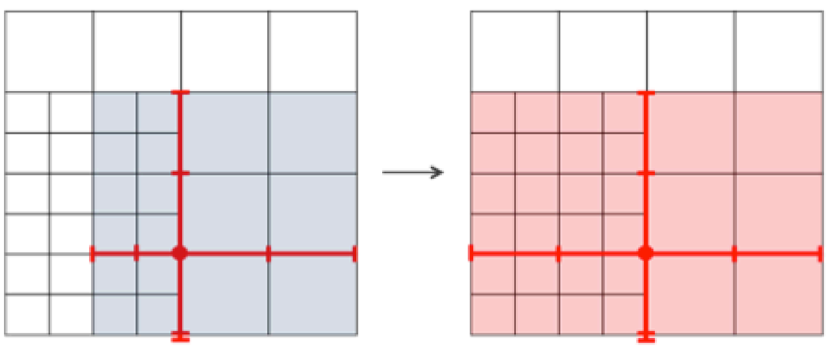
Step 3 - Modification of local knot vectors (more examples)



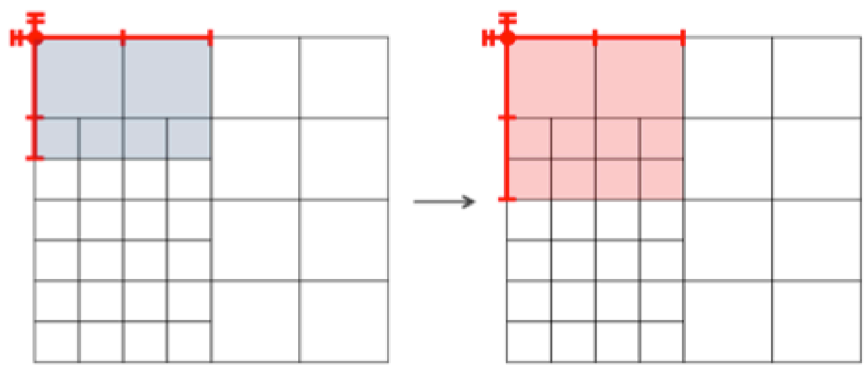
(a) Initial supports that satisfy both conditions and should not be modified.



(b) Condition 1 is not satisfied because $\Delta_1^\xi < \Delta_2^\xi$, so $\xi_1^* \leftarrow \xi_3 - 2h$.



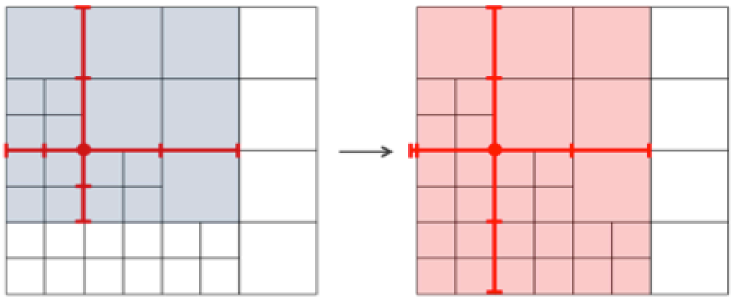
(c) Condition 1 is not satisfied because $\Delta_2^\xi < \Delta_3^\xi$, so $\xi_1^* \leftarrow \xi_3 - 2h$ and $\xi_2^* \leftarrow \xi_3 - h$.



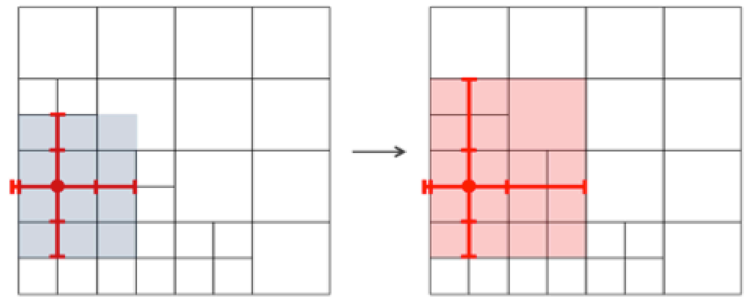
(d) Condition 1 is not satisfied because $\Delta_1^\eta < \Delta_2^\eta$, so $\eta_1^* \leftarrow \eta_3 - 2h$.

Strategy for defining polynomial spline spaces

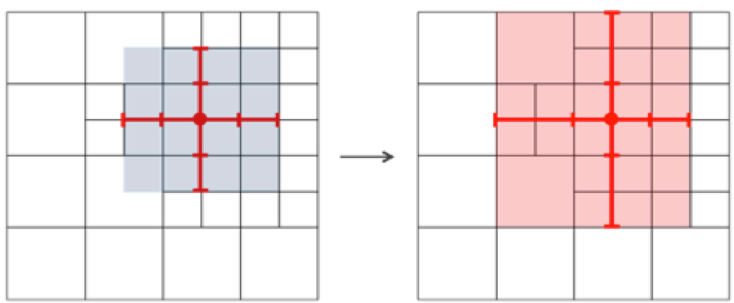
Step 3 - Modification of local knot vectors (more examples)



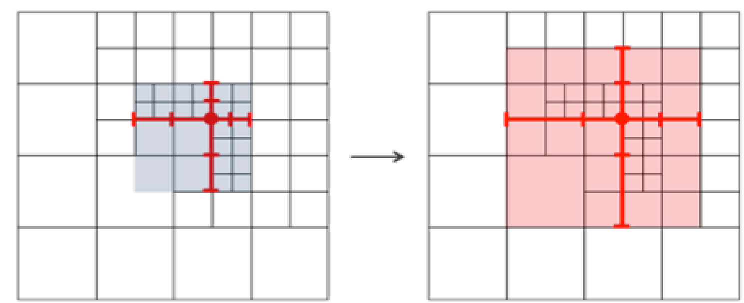
(e) Condition 1 is not satisfied because $\Delta_2^\xi < \Delta_3^\xi$ and $\Delta_2^\eta < \Delta_3^\eta$, so $\xi_2^* \leftarrow \xi_3 - h$, $\eta_2^* \leftarrow \eta_3 - h$ and $\eta_1^* \leftarrow \eta_3 - 2h$.



(f) Condition 2 is not satisfied because $V_{5,5} \notin skt(T)$, so $\xi_5^* \leftarrow \xi_3 + 3h$ and $\eta_5^* \leftarrow \eta_3 + 3h$.



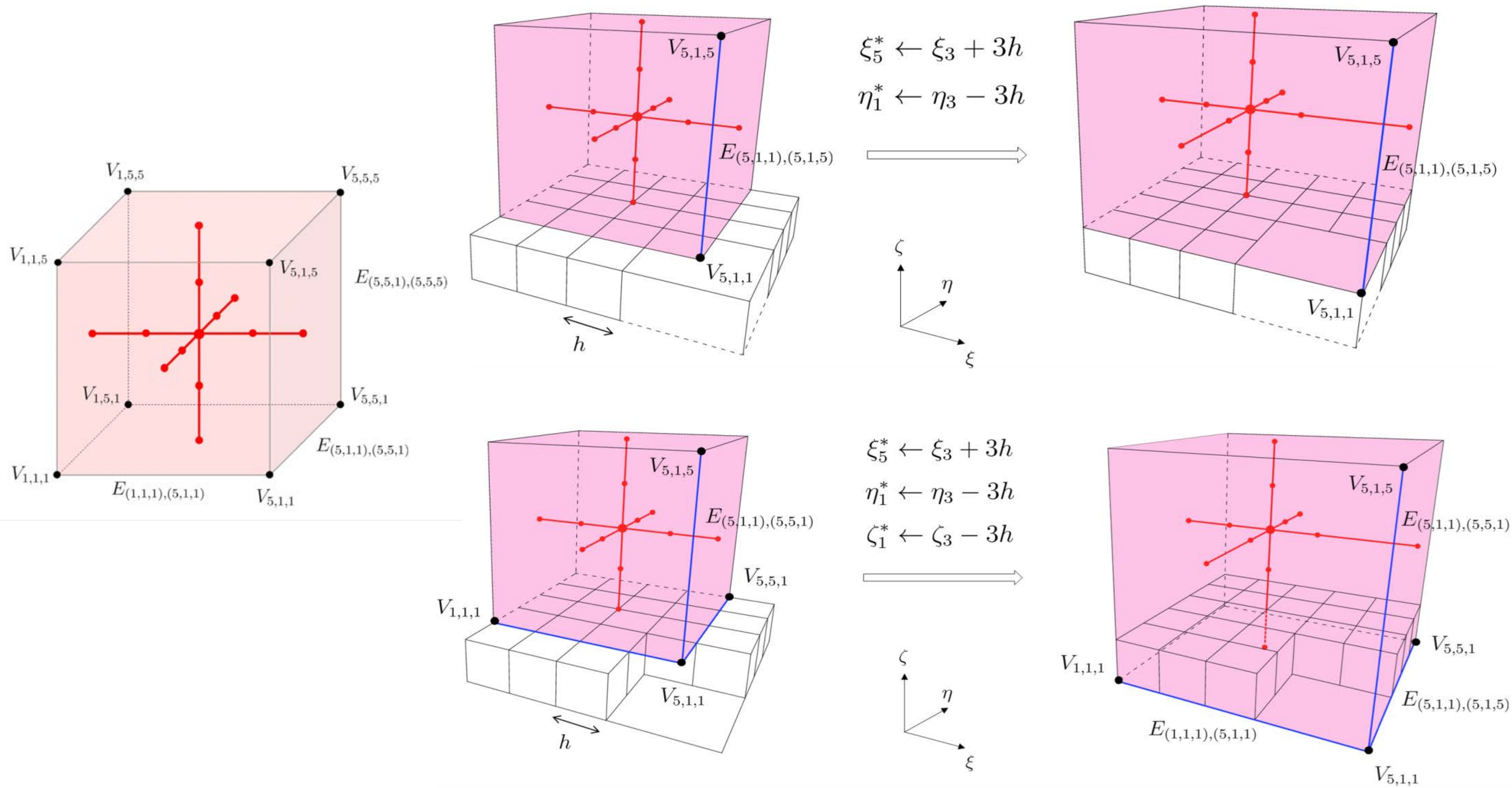
(g) Condition 2 is not satisfied because $V_{1,1}$ and $V_{1,5} \notin skt(T)$, so $\xi_1^* \leftarrow \xi_3 - 3h$, $\eta_1^* \leftarrow \eta_3 - 3h$ and $\eta_5^* \leftarrow \eta_3 + 3h$.



(h) Condition 1 and 2 are not satisfied because $\Delta_2^\xi > \Delta_3^\xi$, $\Delta_2^\eta > \Delta_3^\eta$ and $V_{1,1} \notin skt(T)$, so $\xi_4^* \leftarrow \xi_3 + h$, $\xi_5^* \leftarrow \xi_3 + 2h$, $\eta_4^* \leftarrow \eta_3 + h$, $\eta_5^* \leftarrow \eta_3 + 2h$, $\xi_1^* \leftarrow \xi_3 - 3h$, $\eta_1^* \leftarrow \eta_3 - 3h$.

Strategy for defining polynomial spline spaces

Straightforward implementation in 3D



SIANI

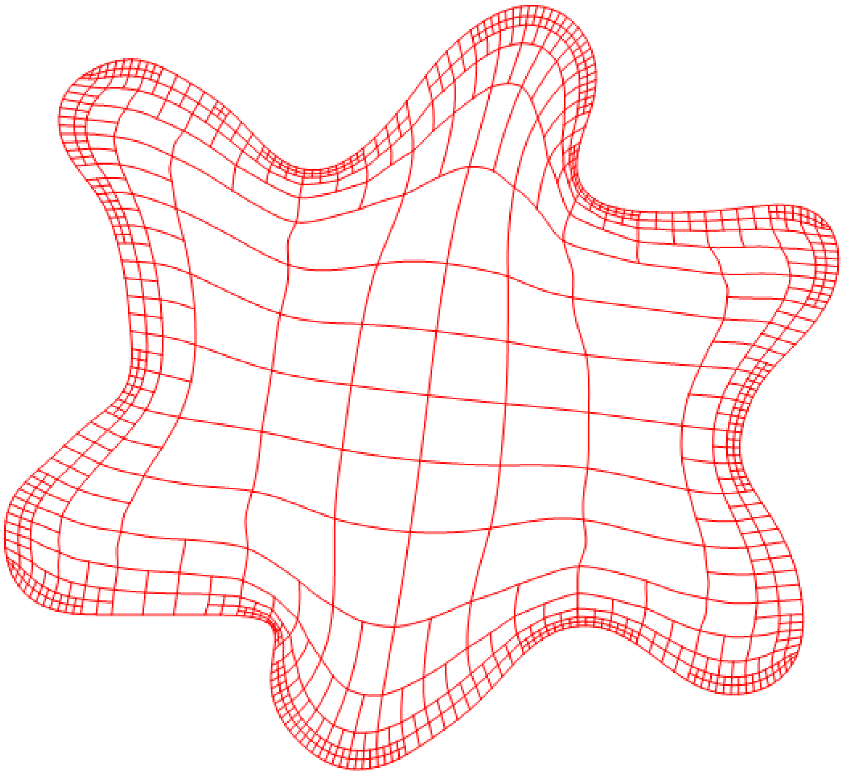
Some remarks

- Function supports can be constructed separately (in parallel)
- For a given T-mesh, the resulting spline space is unique

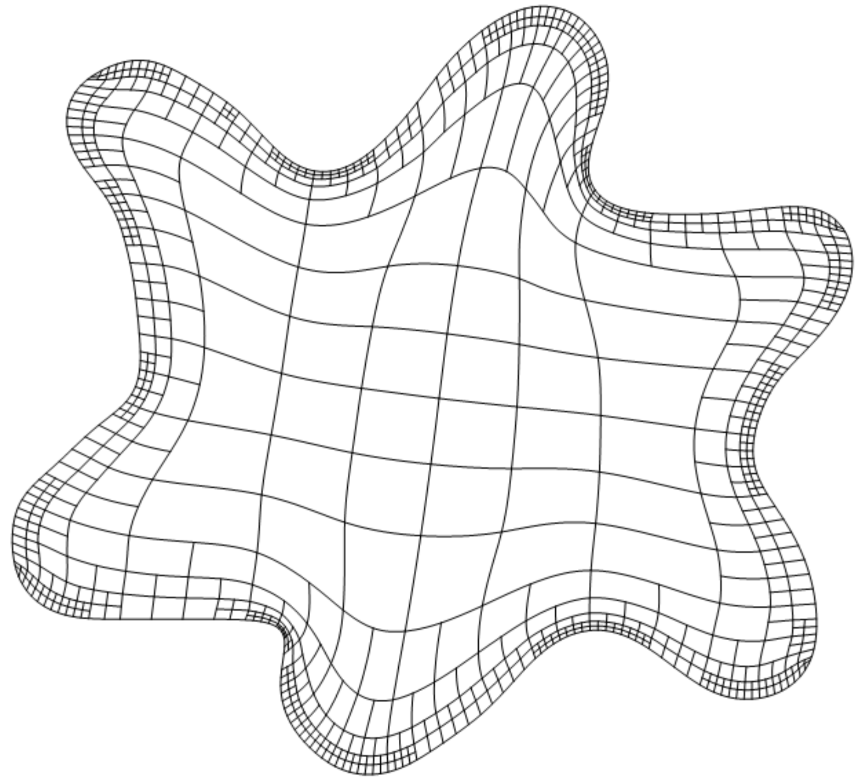
Good quality parameterization for application of IGA

Comparison of rational T-spline parameterization with the new strategy

SIANI



T-spline parameterization, Sederberg
(rational blending functions)



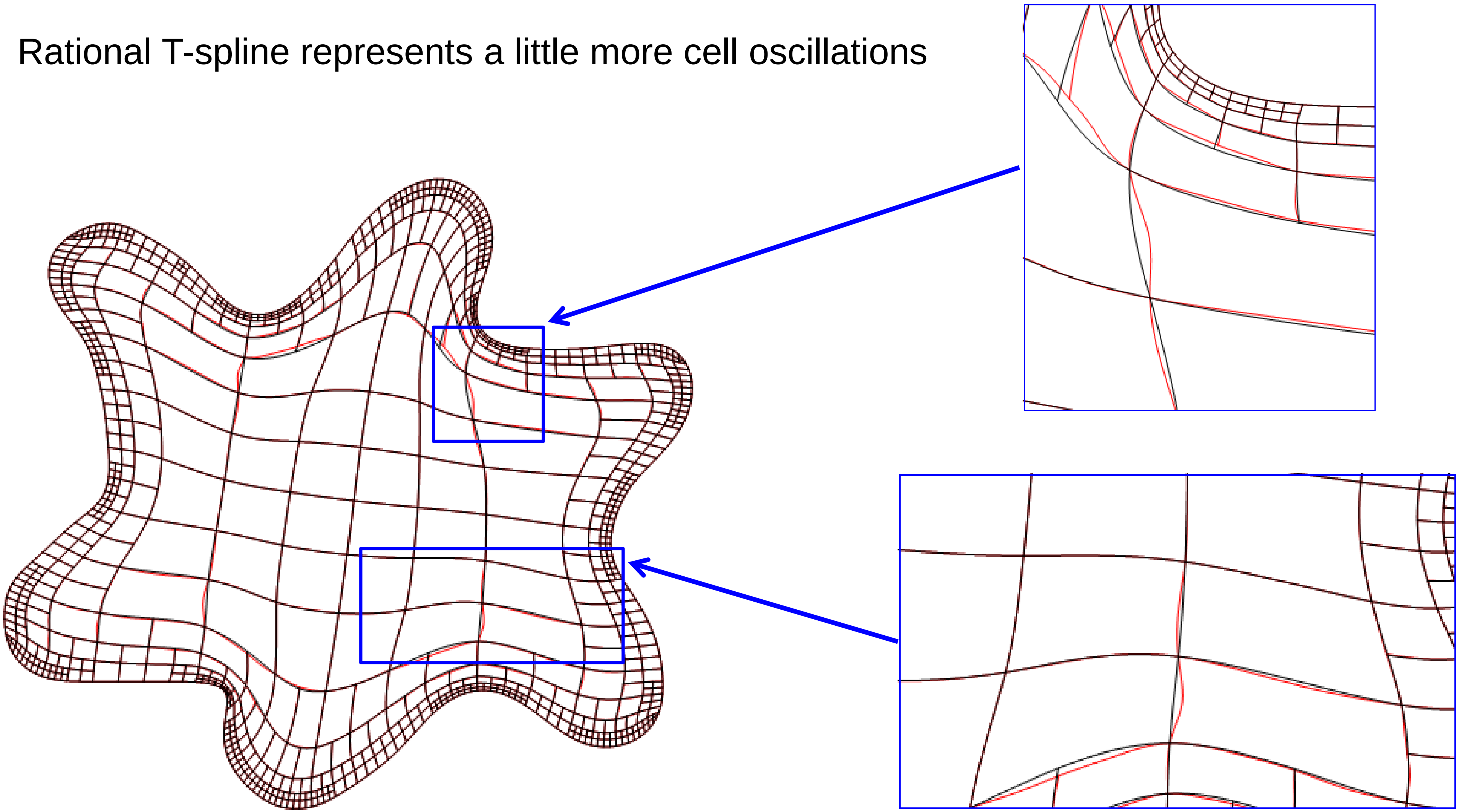
Spline parameterization with the new strategy
(polynomial blending functions)

Good quality parameterization for application of IGA

Comparison of rational T-spline parameterization with the new strategy

SIANI

Rational T-spline represents a little more cell oscillations



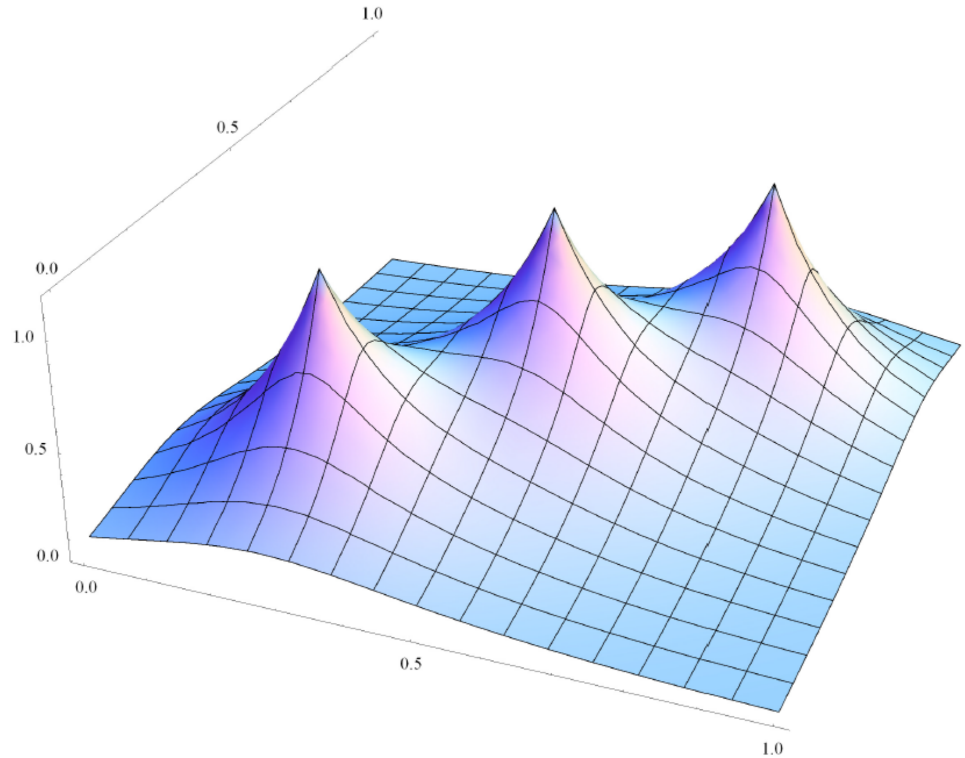
Isogeometric analysis on 2D domain

Poisson problem with Dirichlet boundary condition

$$
\begin{aligned}
-\Delta u &= f \qquad \text{in } \Omega, \\
u &= g \qquad \text{on } \partial\Omega.
\end{aligned}
$$

Analytical solution:

$$
\begin{aligned}
u(x,y) = \exp&\left(-7\sqrt{(x-0.5)^2+(y-0.5)^2}\right) + \\
&+\exp\left(-7\sqrt{(x-0.25)^2+(y-0.25)^2}\right) + \\
&+\exp\left(-7\sqrt{(x-0.75)^2+(y-0.75)^2}\right).
\end{aligned}
$$



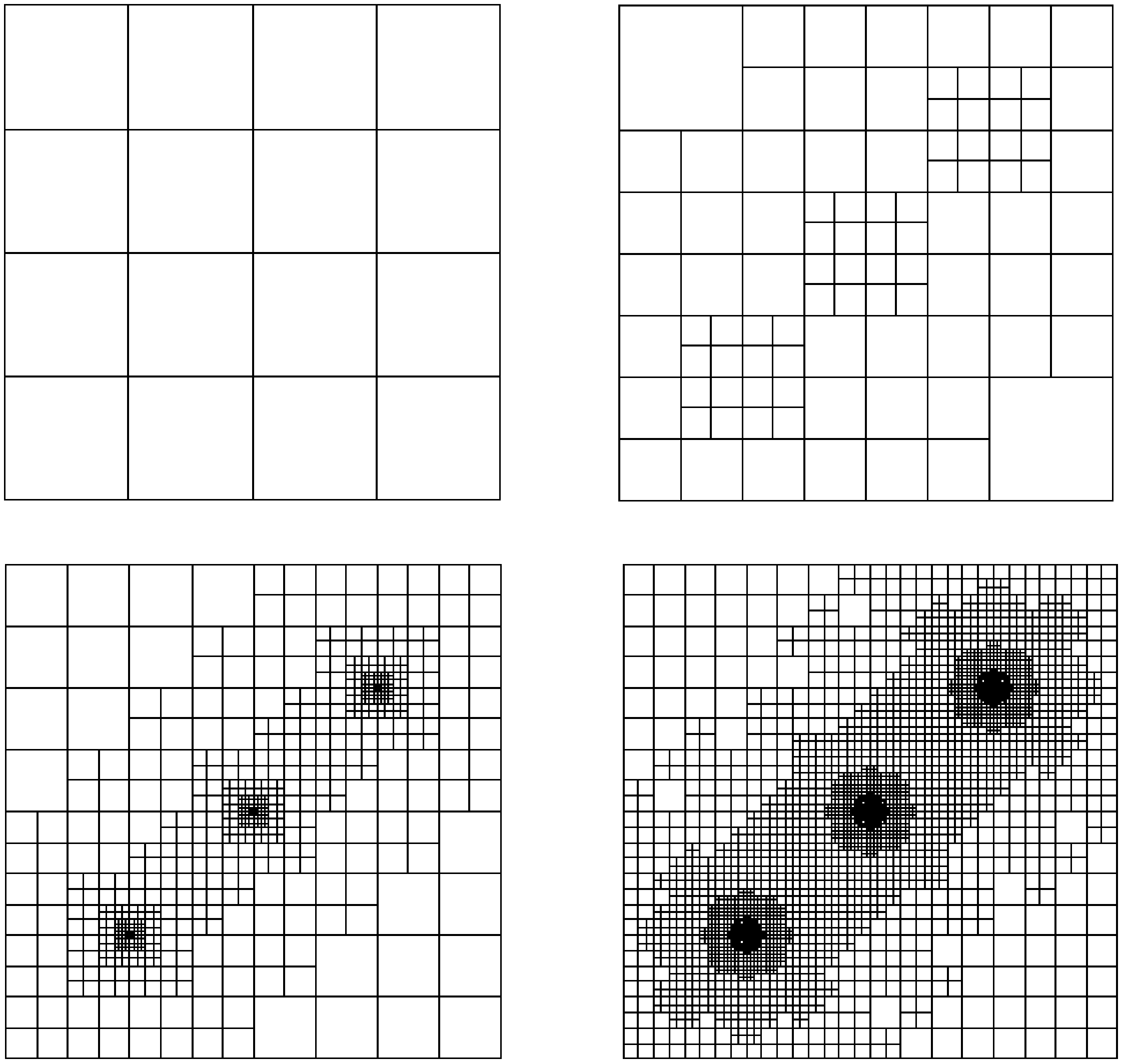
Residual-type error indicator:

$$\eta(\Omega_e)^2 = \int_{\Omega_e} h^2 \left(f + \Delta u_h\right)^2 \, \mathrm{d}\Omega$$

Adaptive refinement for 2D Poisson problem

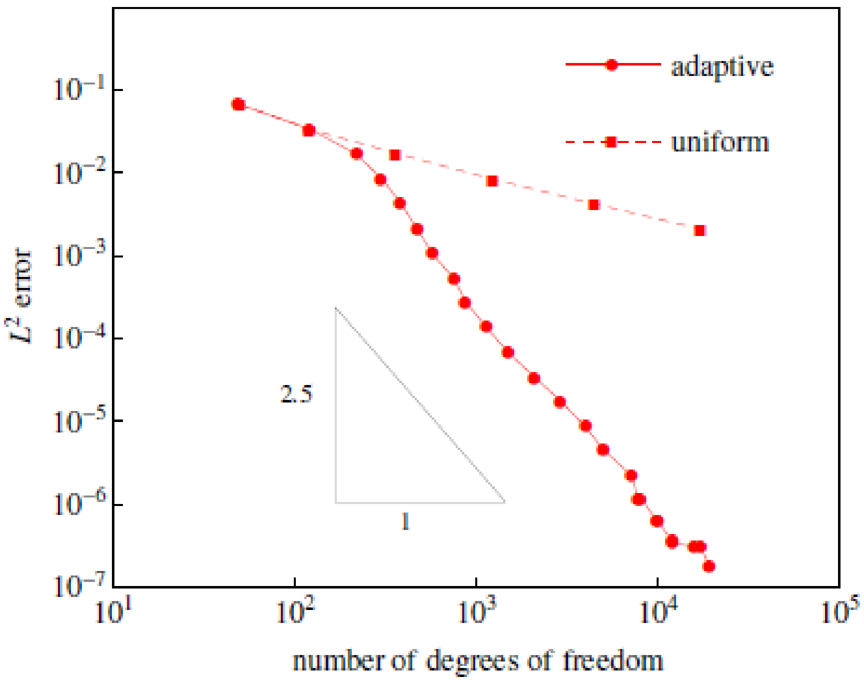
Error indicator : $\eta\left(\Omega_e\right)^2 = \int_{\Omega_e} h^2 \left(f + \Delta u_h\right)^2 \, d\Omega$

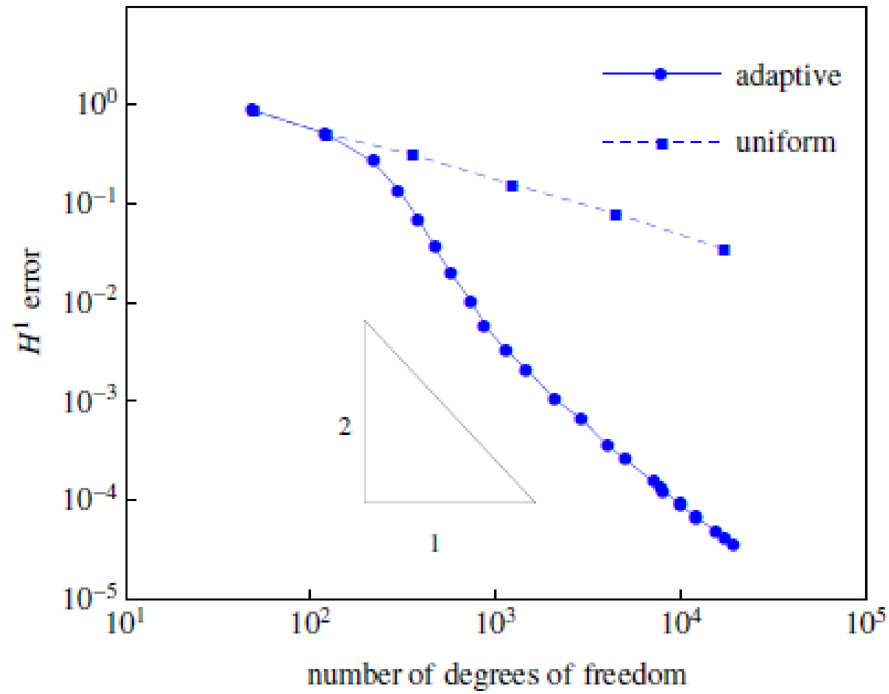
SIANI



Adaptive refinement for 2D Poisson problem

Monotonous convergence: L^2-norm and H^1-seminorm error

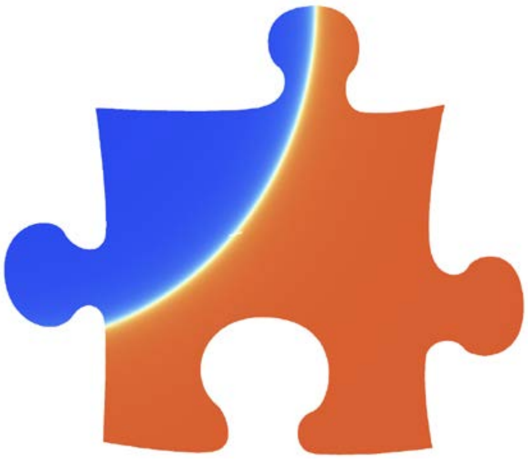




Isogeometric analysis on 2D complex domain

Poisson problem with Dirichlet boundary condition on a puzzle piece

$$-\Delta u = f \qquad \text{in } \Omega,$$
$$u = g \qquad \text{on } \partial\Omega.$$



- Analytical solution: Steep wave front given by

$$u(r) = \arctan(\alpha(r - r_0)), \qquad \text{where } r = \sqrt{(x - x_c)^2 + (y - y_c)^2}\ ,$$
$$(x_c, y_c) = (0, 0),\ \alpha = 200 \text{ and } r_0 = 0.6$$

- Adaptive strategy: Residual-type error indicator:

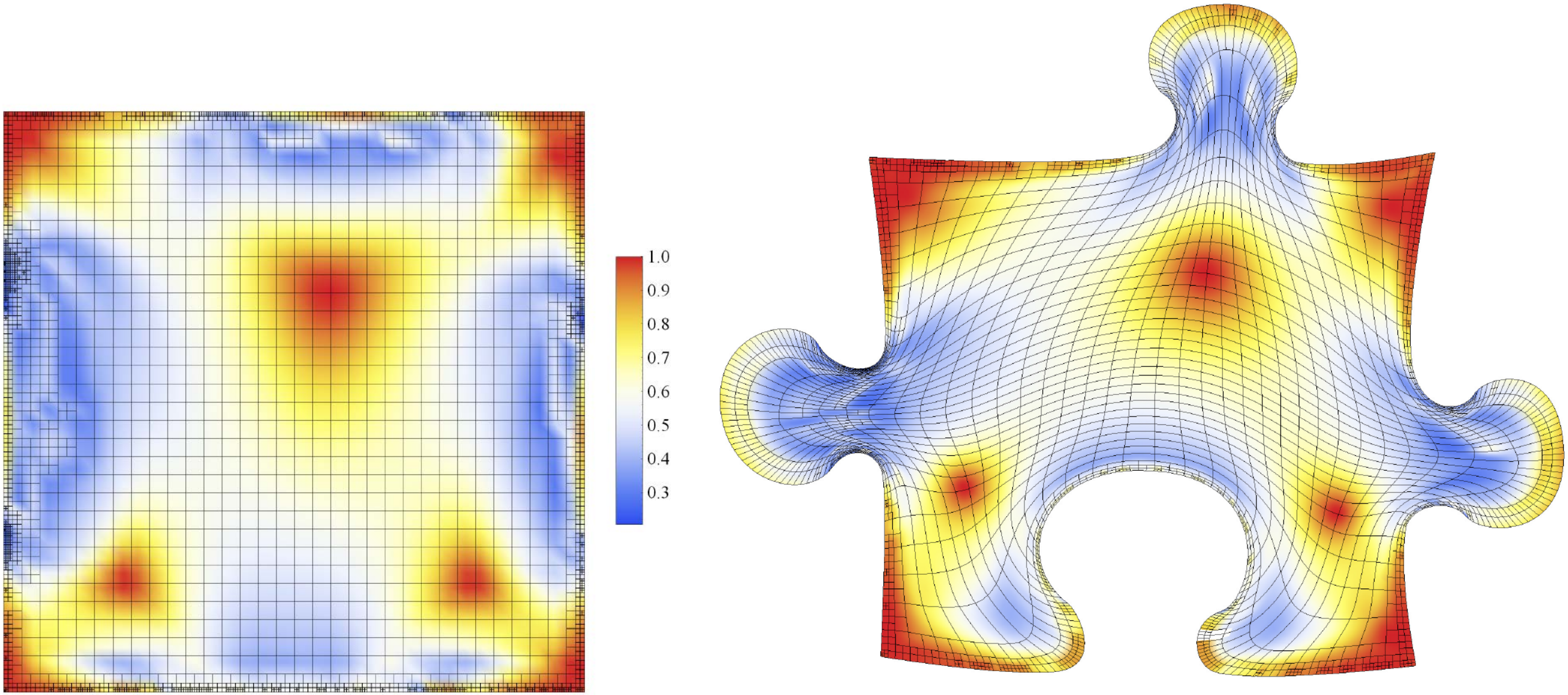
$$\eta(\Omega_e)^2 = \int_{\Omega_e} h^2 \left(f + \Delta u_h\right)^2 \, \mathrm{d}\Omega$$

Isogeometric analysis for 2D Poisson problem

Initial adapted T-mesh and parameterization of the puzzle (Meccano method)

Mean ratio Jacobian as a quality metric

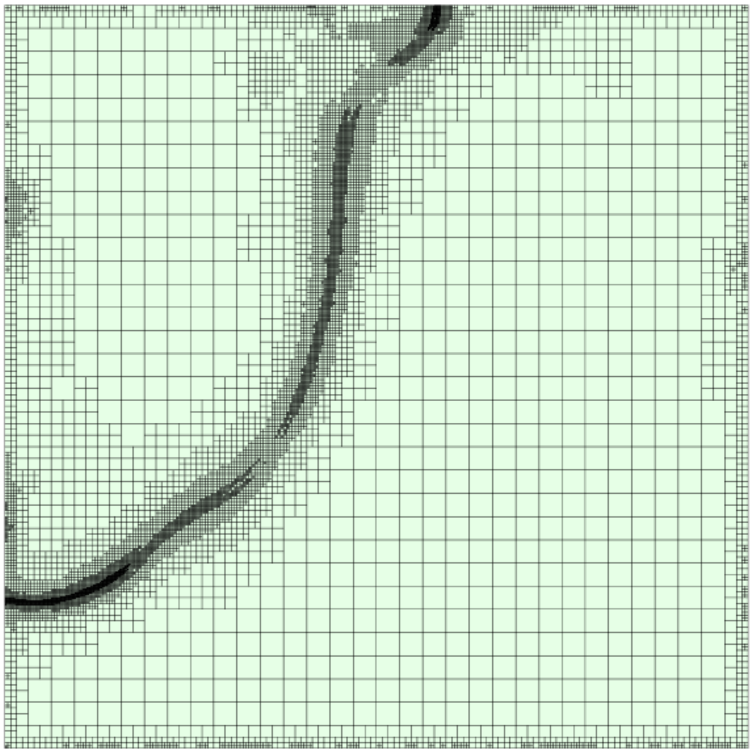
SIANI

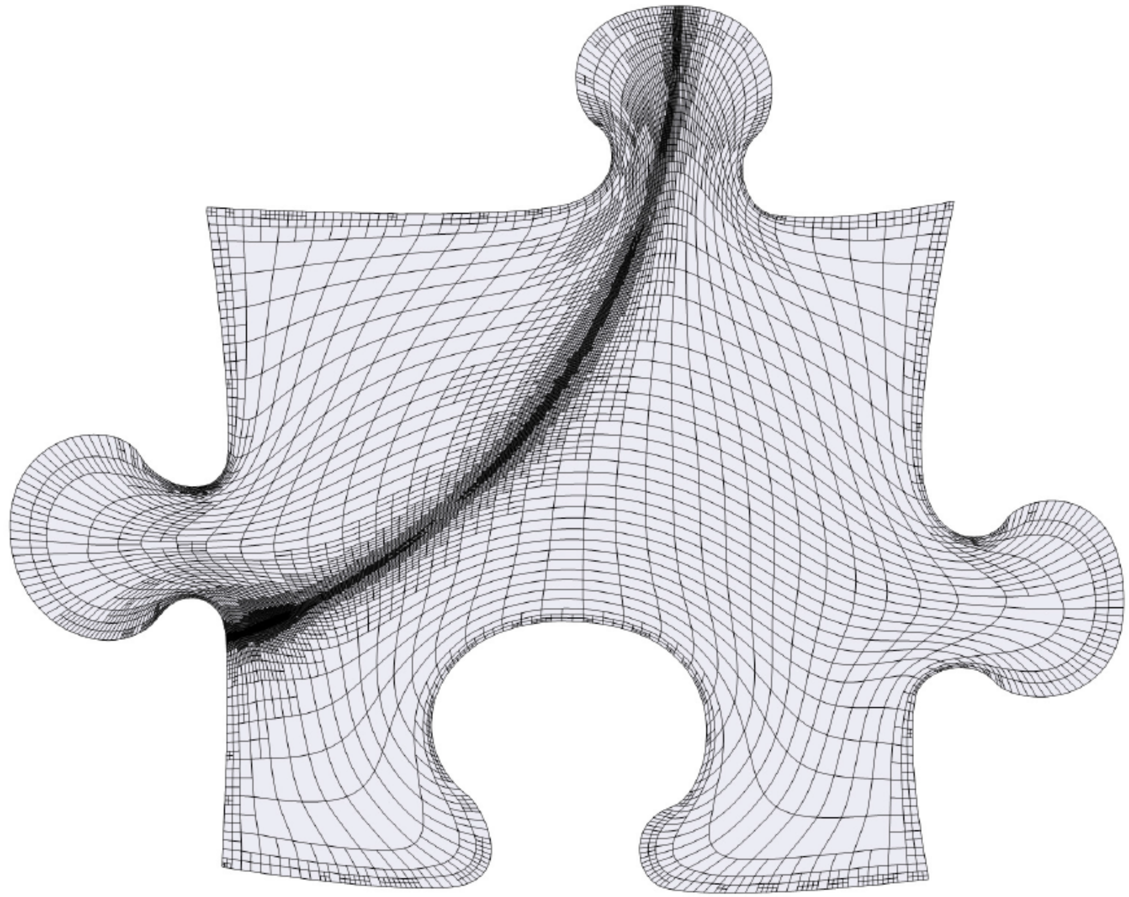


Adaptive refinement for 2D Poisson problem

Adaptive T-mesh after 13 refinement steps

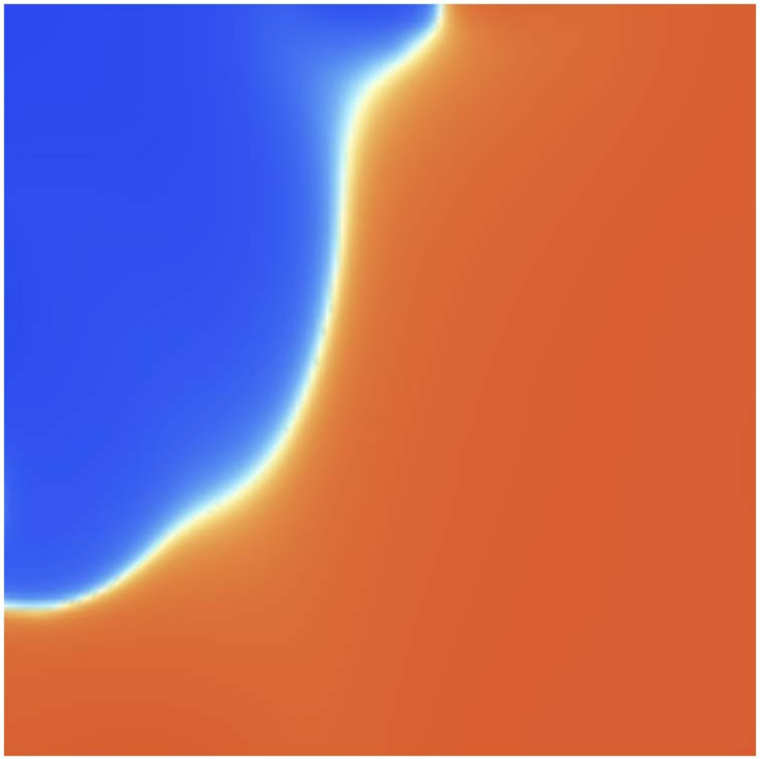
SIANI

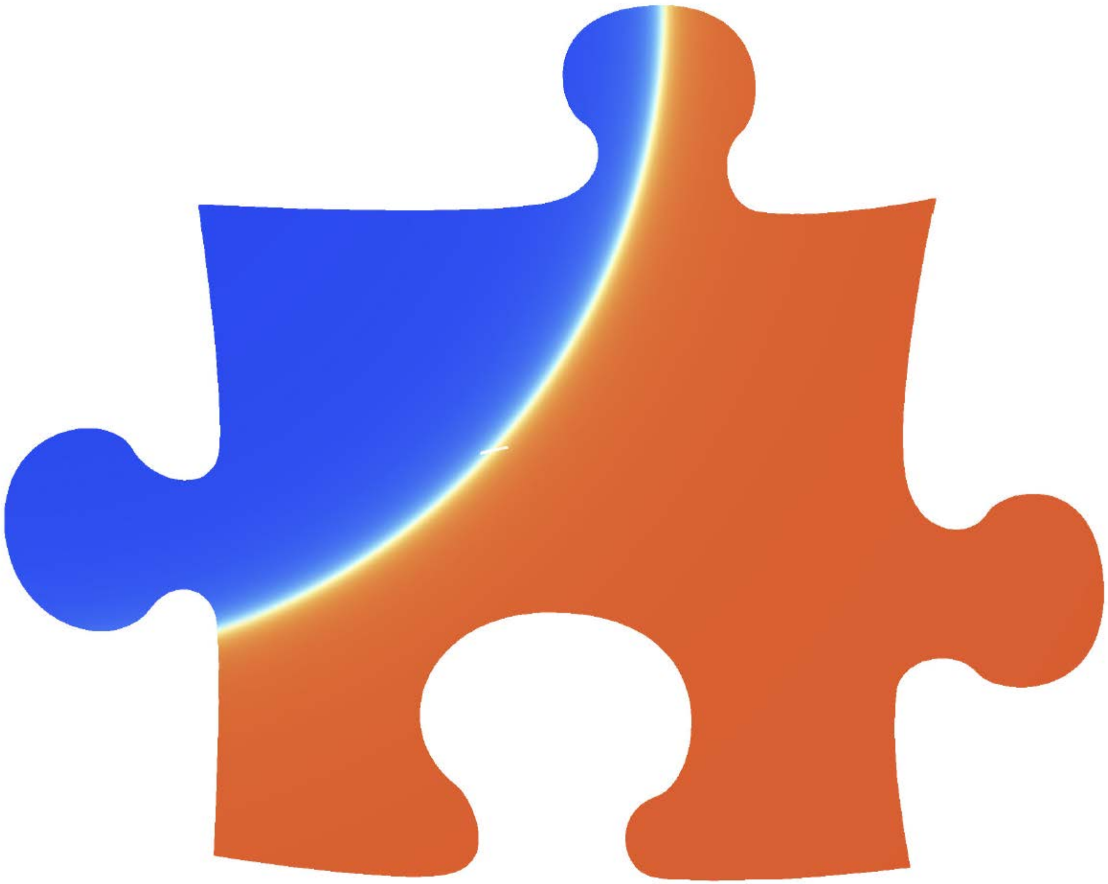




Adaptive refinement for 2D Poisson problem

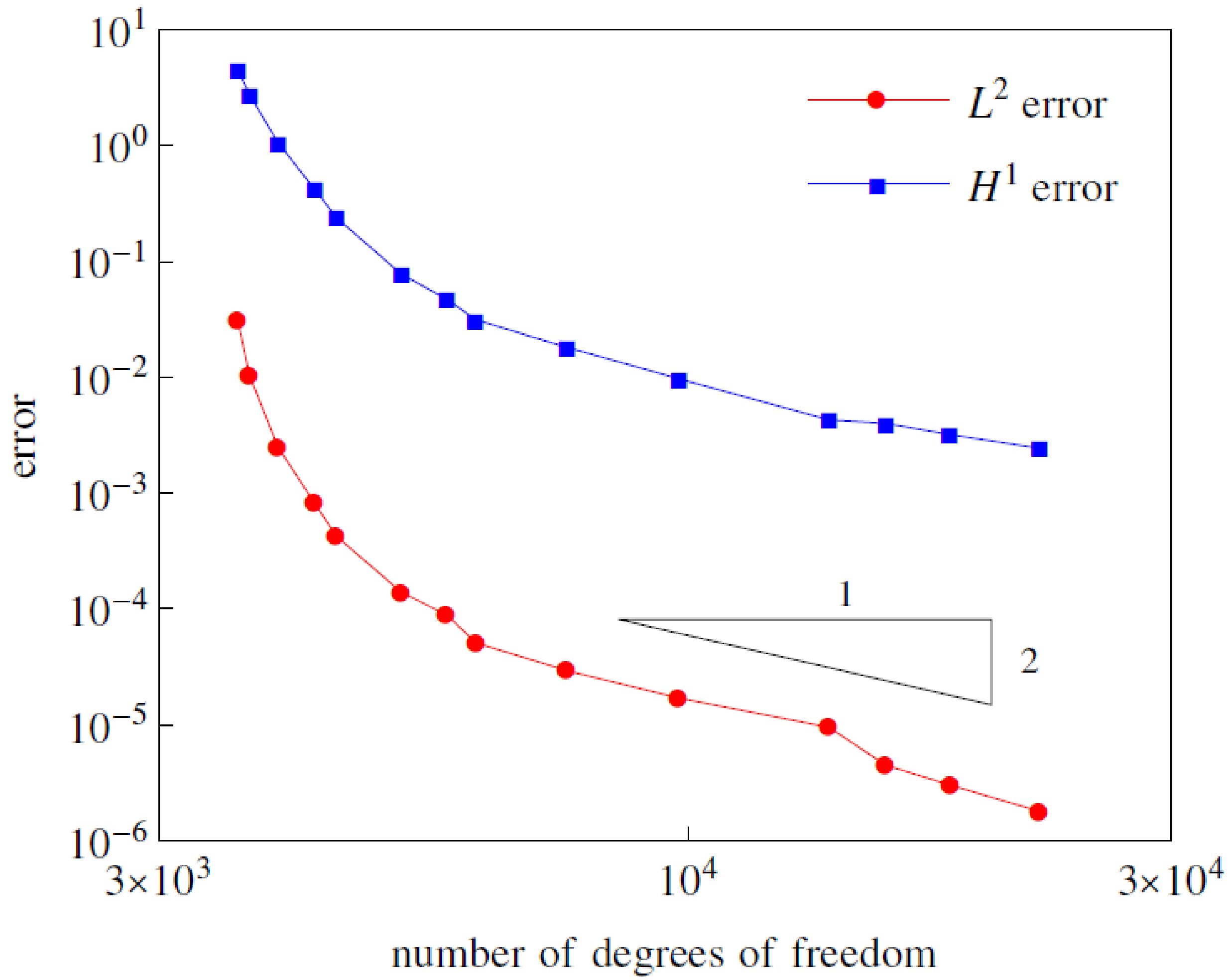
Numerical solution





Adaptive refinement for 2D Poisson problem

Monotonous convergence: L^2-norm and H^1-seminorm error

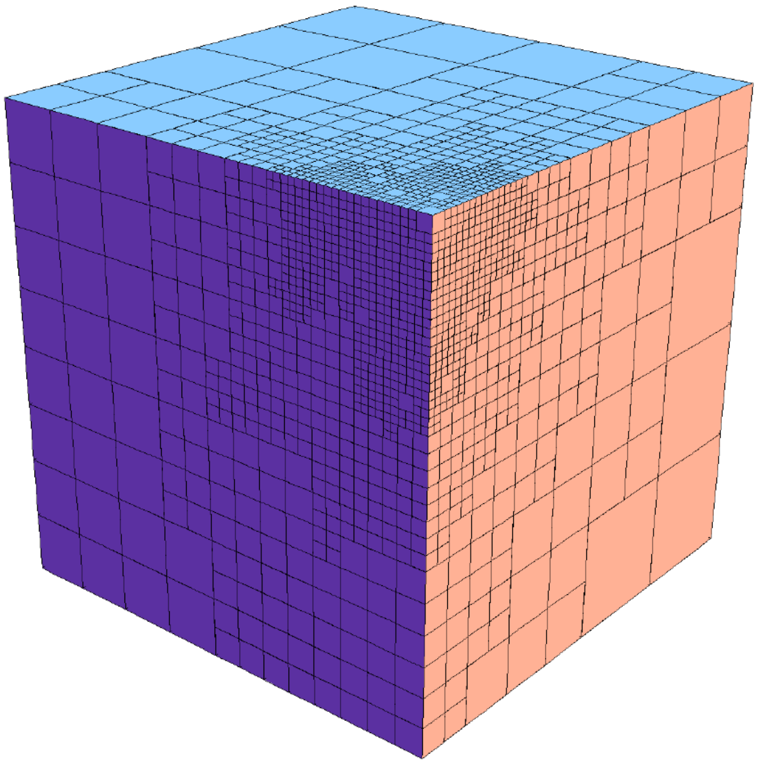


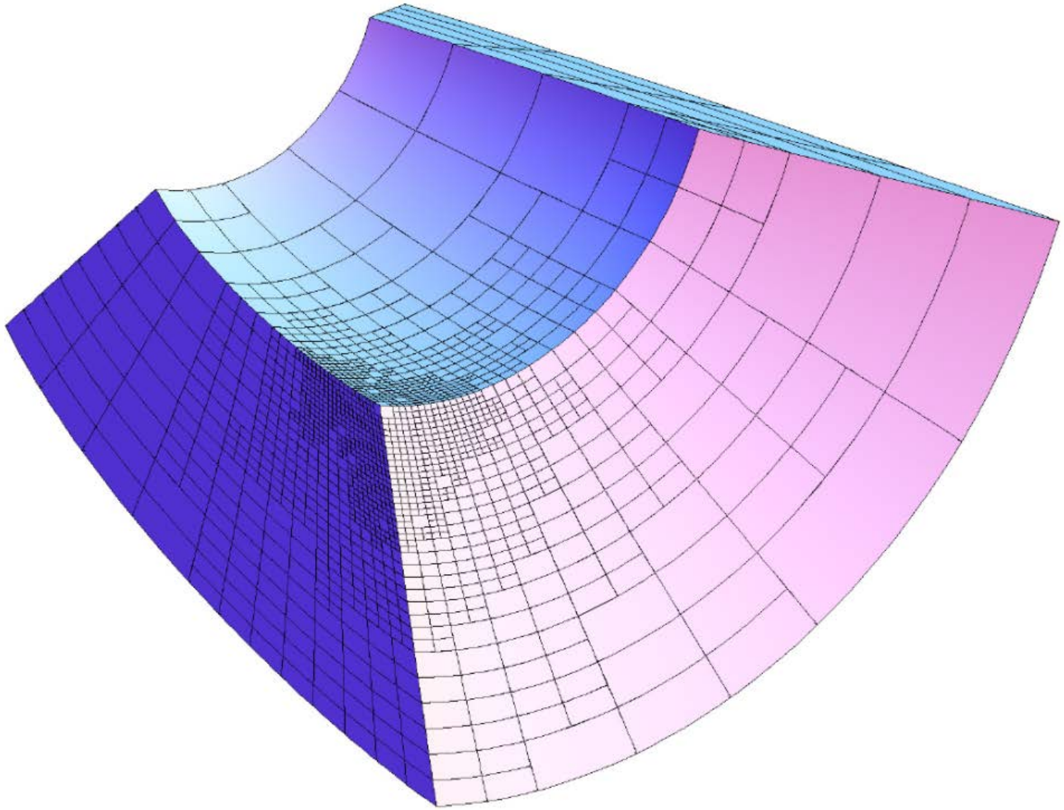
Isogeometric analysis for 3D Poisson problem

Adaptive T-spline parameterization of a sphere portion

Analytical solution:

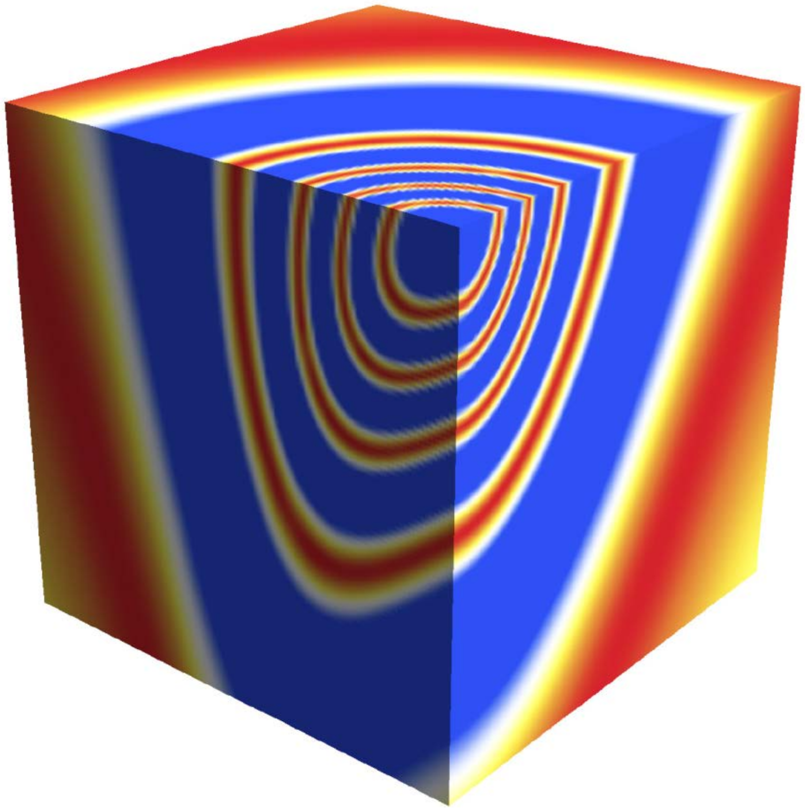
$$u(r) = \sin\left(\frac{1}{\alpha + r}\right), \quad \text{where } r = \sqrt{x^2 + y^2} \quad \text{and oscillation parameter} \quad \alpha = 1/10\pi$$

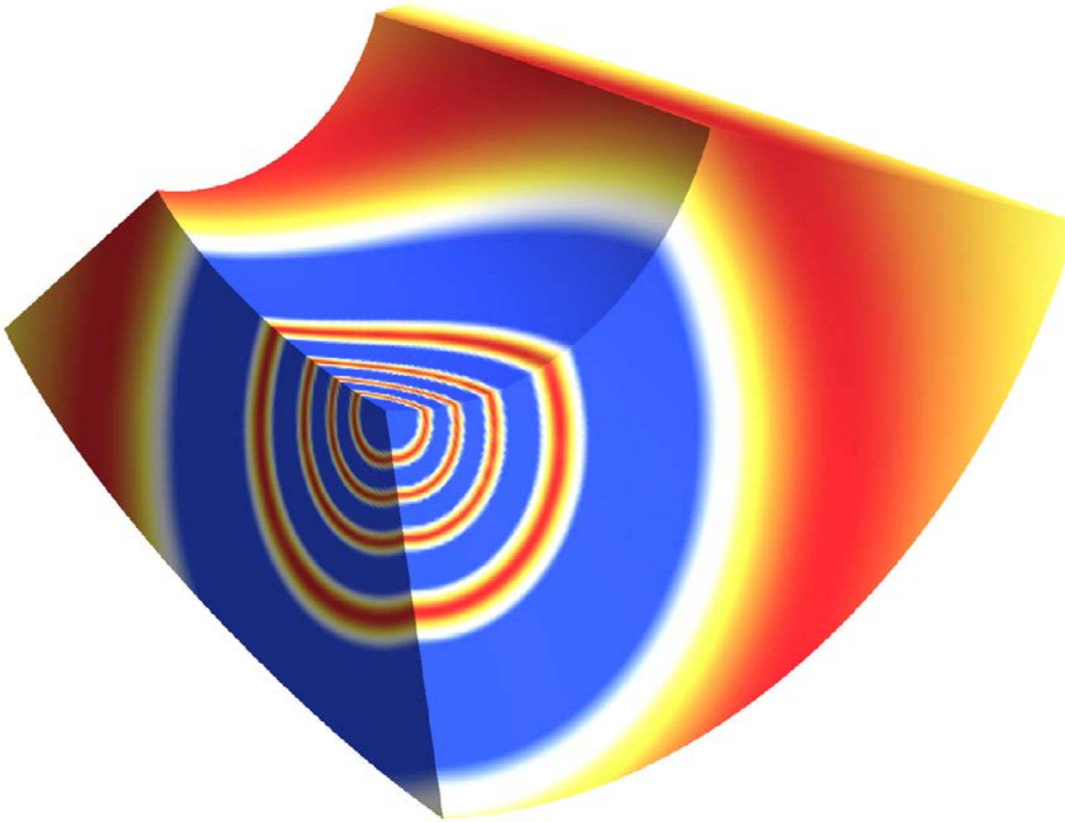




Adaptive refinement for 3D Poisson problem

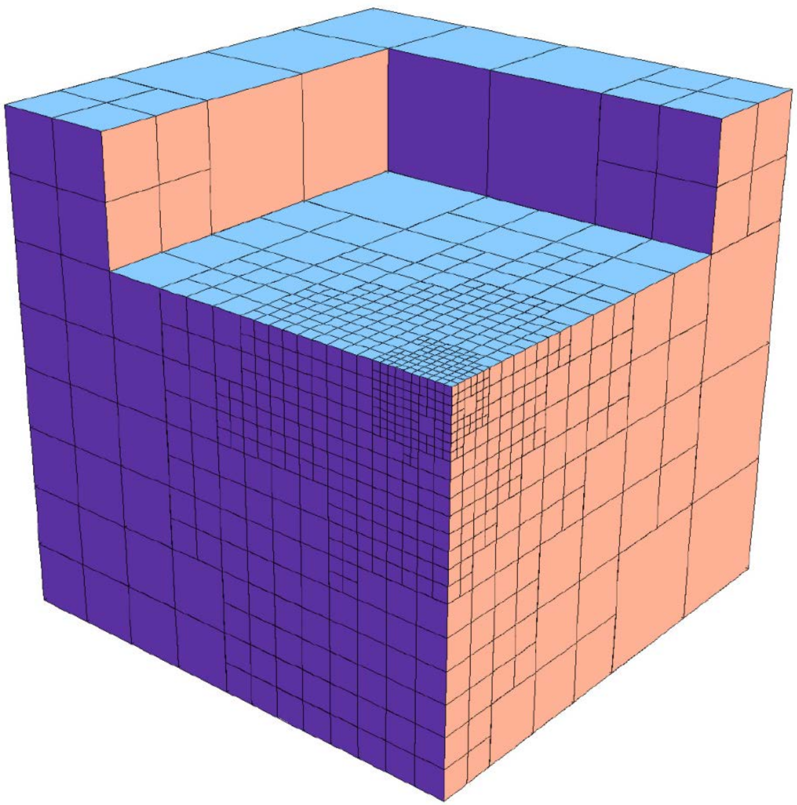
Numerical solution

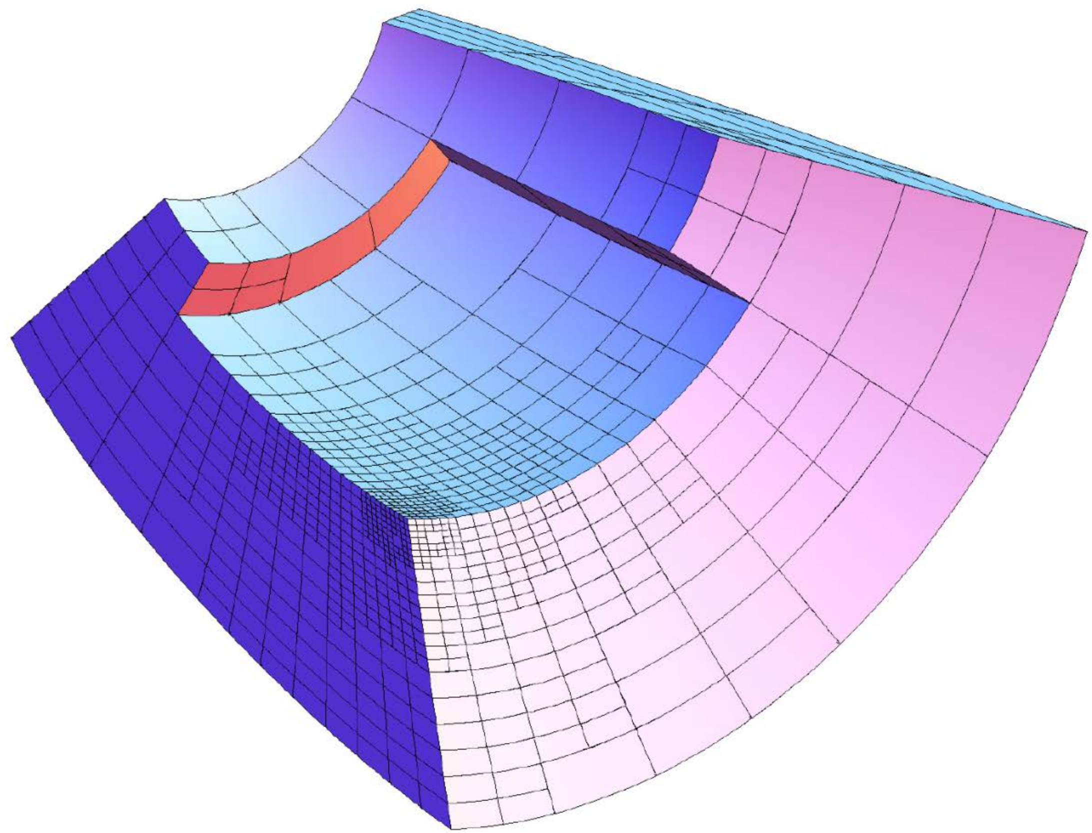




Adaptive refinement for 3D Poisson problem

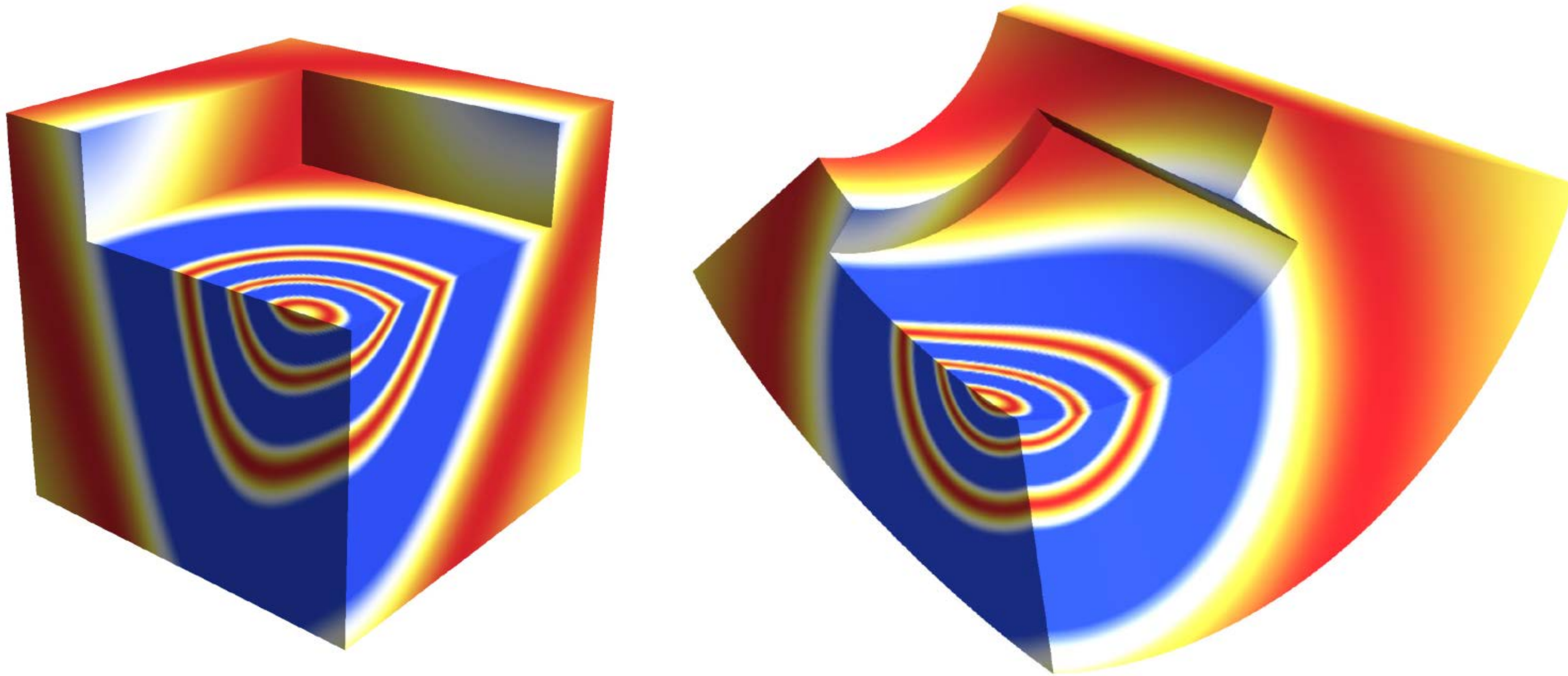
A section of the parametric and physical discretization





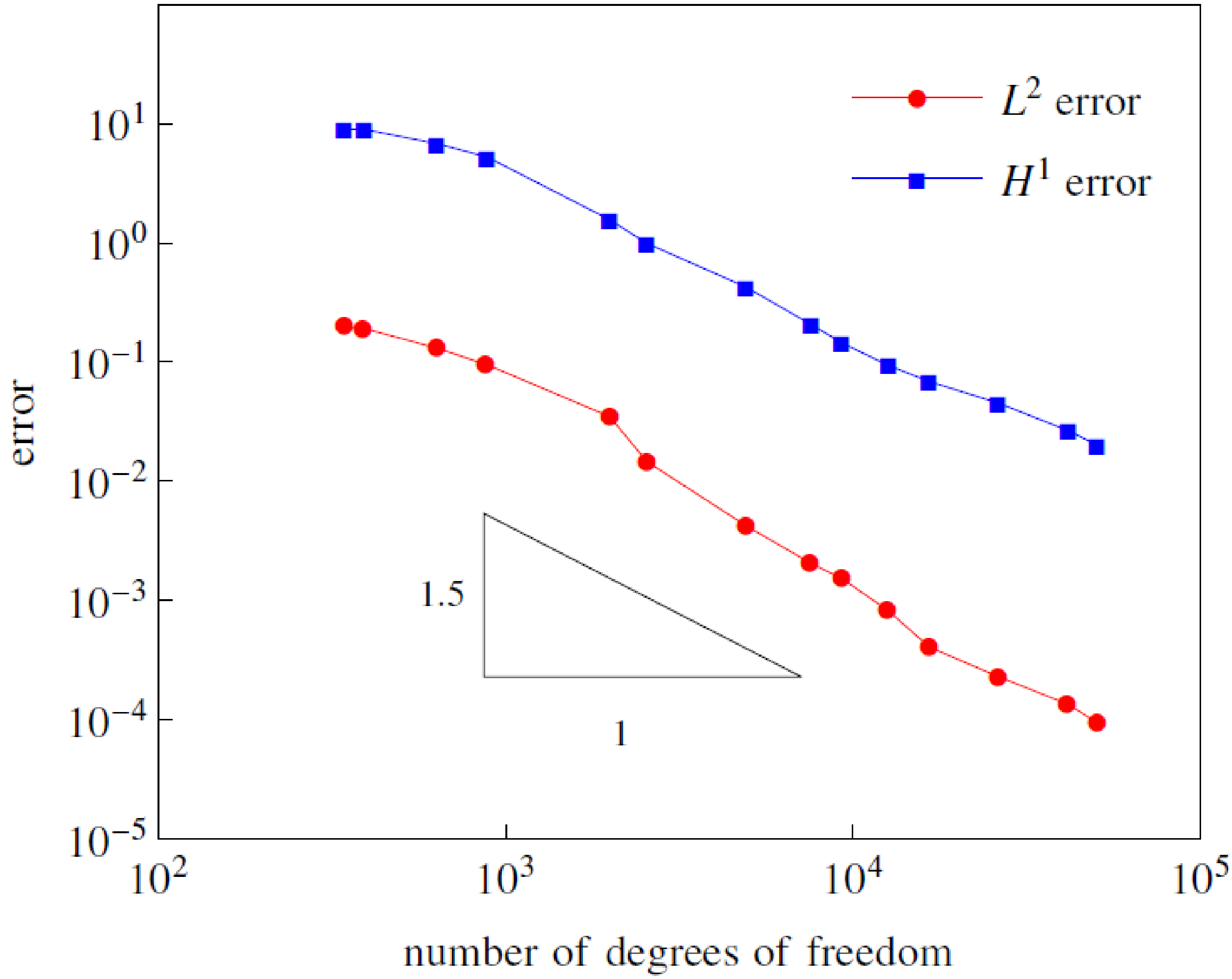
Adaptive refinement for 3D Poisson problem

Numerical solution on a section of the parametric and physical domain



Adaptive refinement for 3D Poisson problem

Monotonous convergence: L^2-norm and H^1-seminorm error



SIANI

Conclusions

- We have developed a new strategy for constructing tensor product cubic spline spaces over arbitrary quadtree (2D) and octree (3D) meshes

- Easy implementation in 2D and 3D

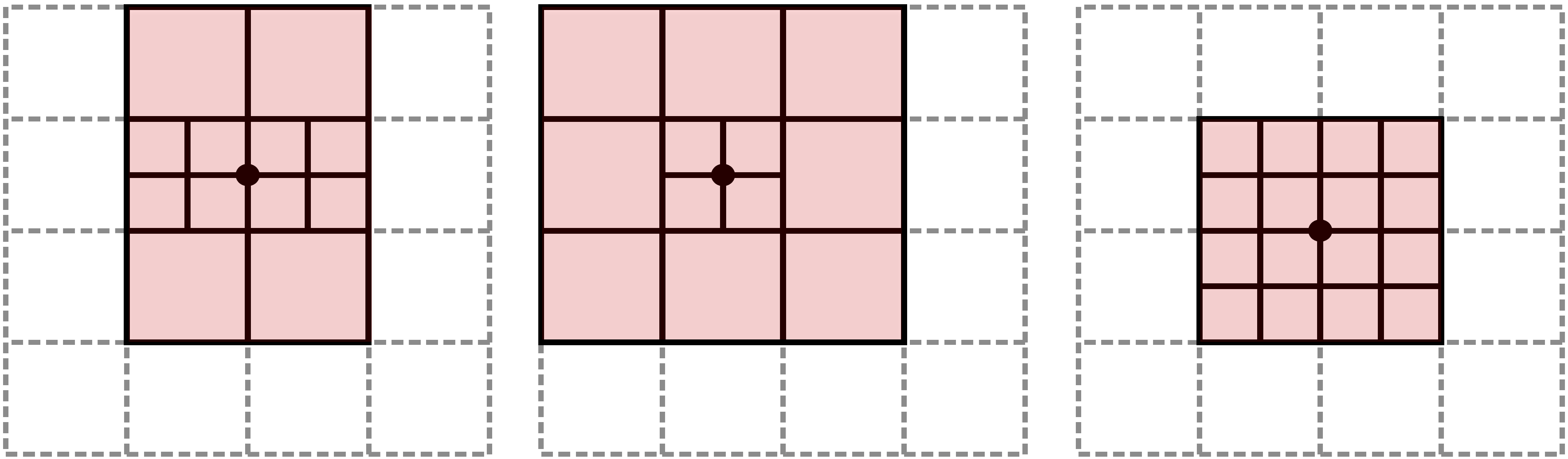
Future Works

- The proof of the nestedness of the spaces and their linear independence is in preparation

Strategy for defining polynomial spline spaces

Comment about the nestedness proof

Three basic types of a blending function of level *K* verifying conditions 1 and 2



Type I **Type II** **Type III**

Function Support of level *K* covers 6 (Type I), 9 (Type II) or 4 (Type III) cells of level *K-1*

SIANI
INSTITUTO UNIVERSITARIO
INGENIERIA COMPUTACIONAL

Construction of polynomial spline spaces over quadtree and octree T-meshes

M. Brovka[(1)], J.I. López[(1)], J.M. Escobar[(1)], J.M. Cascón[(2)] and R. Montenegro[(1)]*

(1) University Institute SIANI, University of Las Palmas de Gran Canaria, Spain

(2) Department of Economics and History of Economics, University of Salamanca, Spain

23rd International Meshing Roundtable (IMR23)

October 12–15, 2014, London, UK

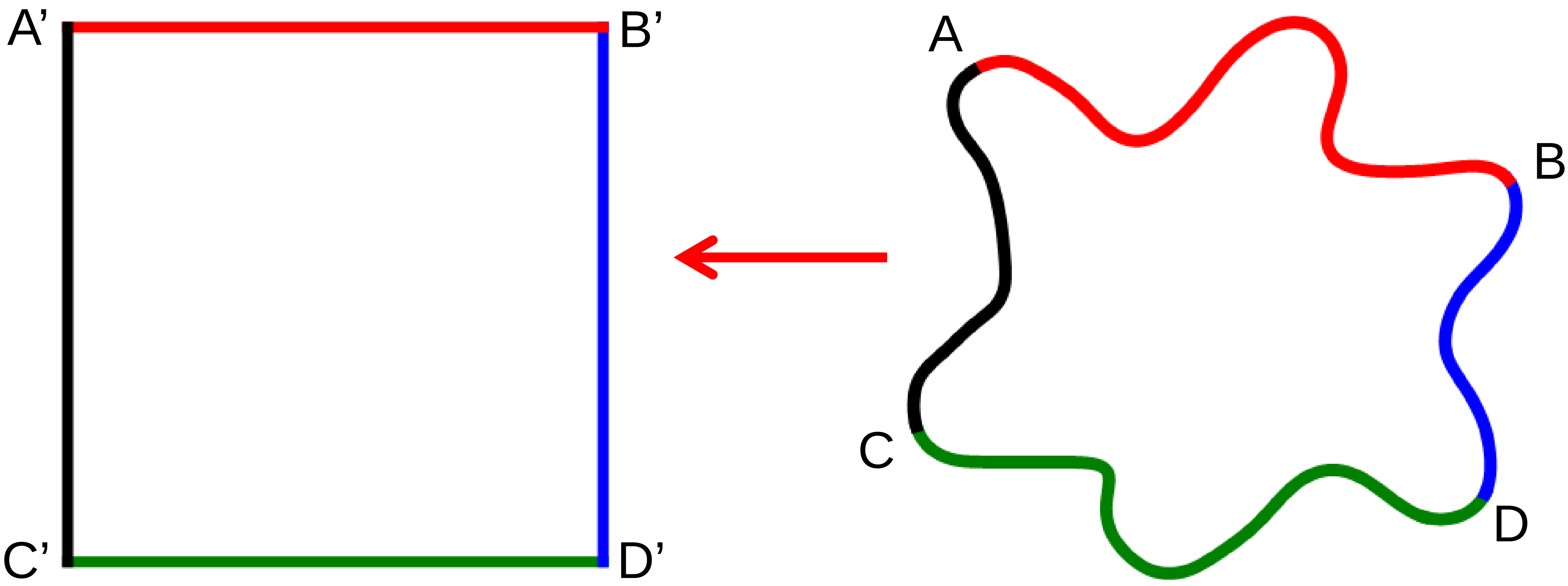
MINECO y FEDER Project: CGL2011-29396-C03-00
CONACYT-SENER Project, Fondo Sectorial, contract: 163723
http://www.dca.iusiani.ulpgc.es/proyecto2012-2014

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

The Meccano Method on T-meshes in 2-D

Step 1: Input boundary (image, polyline, curve, etc.) and boundary mapping

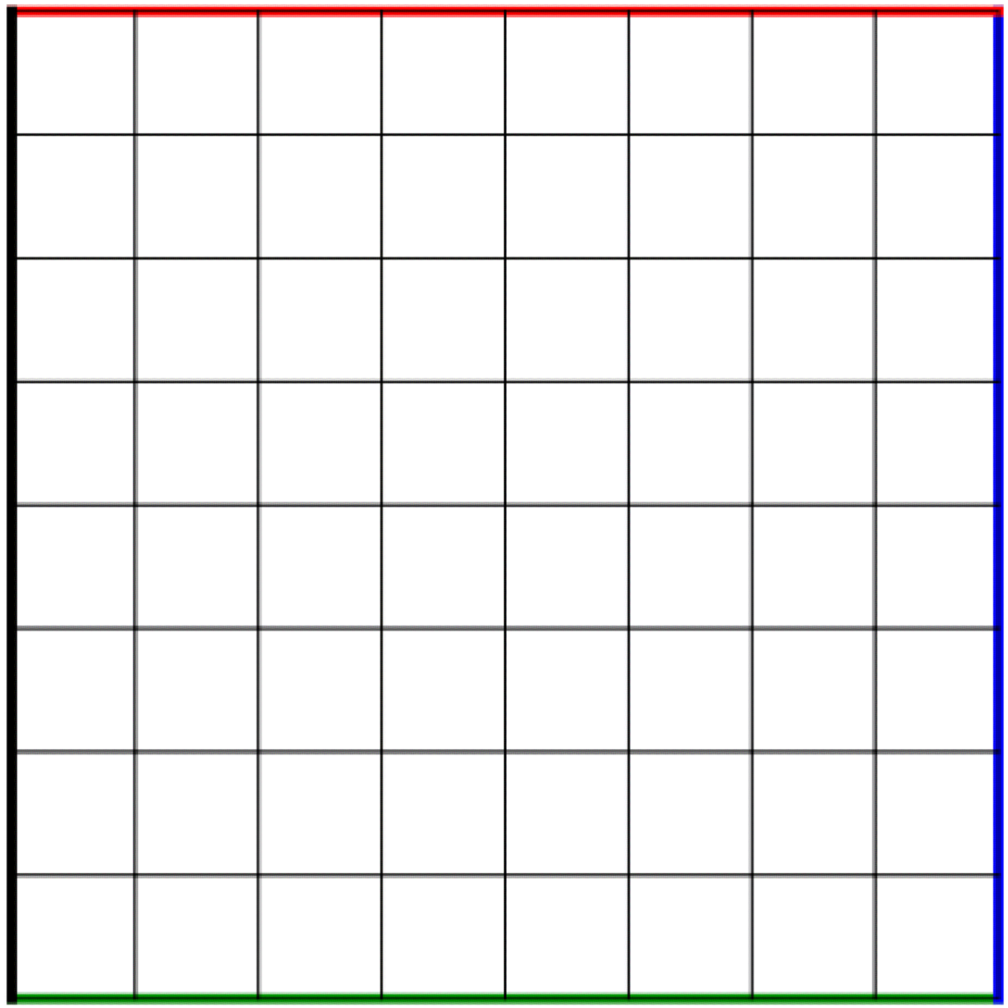
SIANI



- Select four points (A, B, C, D) of the input boundary
- Boundary parameterization via chord-length

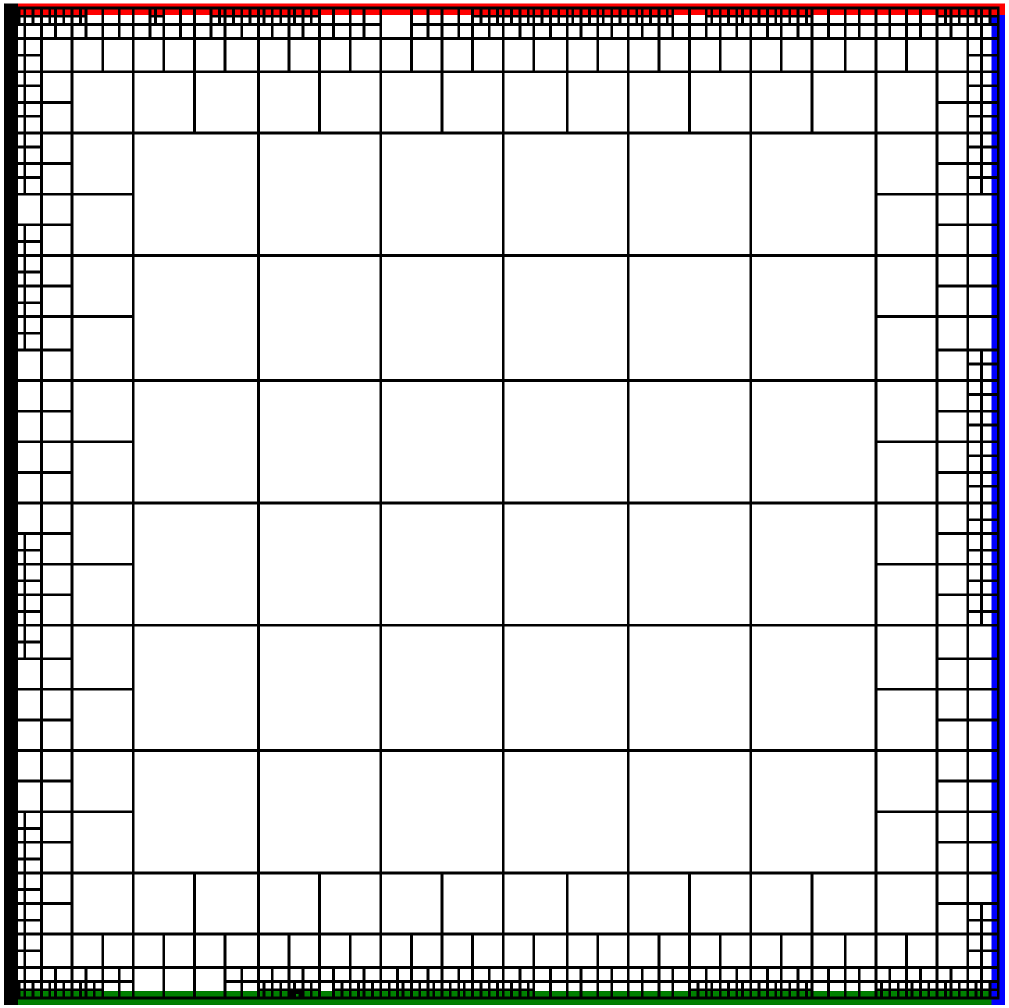
The Meccano Method on T-meshes in 2-D

Step 2: Coarse quadrilateral mesh of the meccano (parameter space)



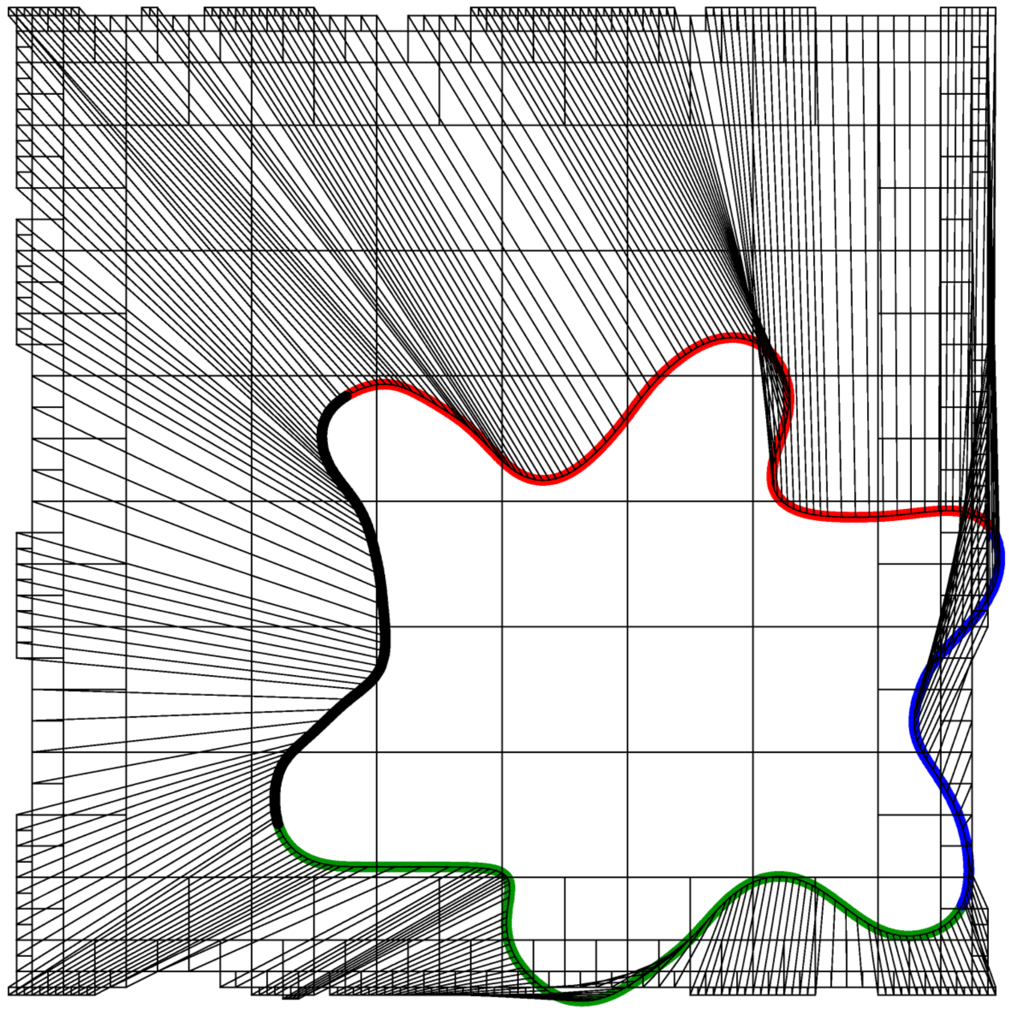
The Meccano Method on T-meshes in 2-D

Step 3: Refine mesh with quadtree subdivisions to approach the boundary



The Meccano Method on T-meshes in 2-D

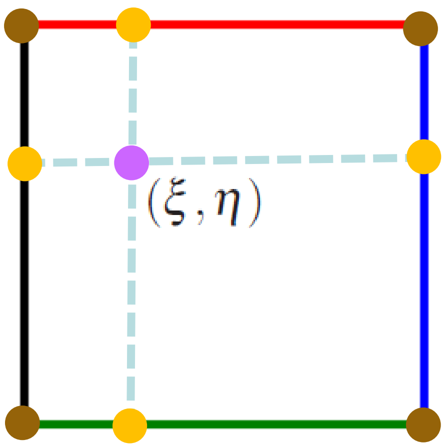
Step 4: Move the meccano boundary nodes to the object boundary

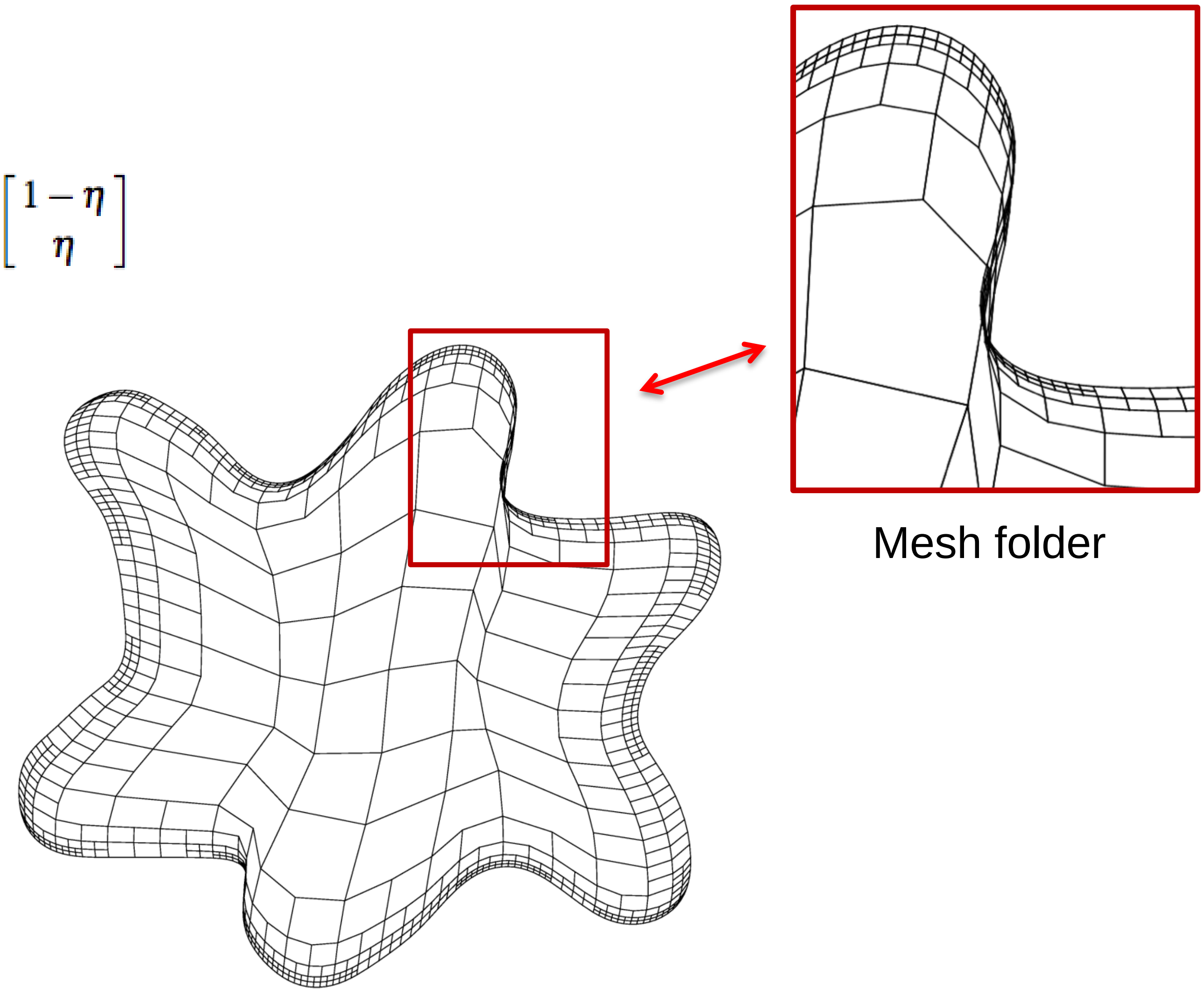


The Meccano Method on T-meshes in 2-D

Step 5: Inner node relocation with Coons patch to facilitate the optimization

$$\mathbf{x}(\xi,\eta) = (1-\xi)\mathbf{x}(0,\eta) + \xi\mathbf{x}(1,\eta) + (1-\eta)\mathbf{x}(\xi,0) + \eta\mathbf{x}(\xi,1) - \begin{bmatrix} 1-\xi & \xi \end{bmatrix} \begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}(0,1) \\ \mathbf{x}(1,0) & \mathbf{x}(1,1) \end{bmatrix} \begin{bmatrix} 1-\eta \\ \eta \end{bmatrix}$$

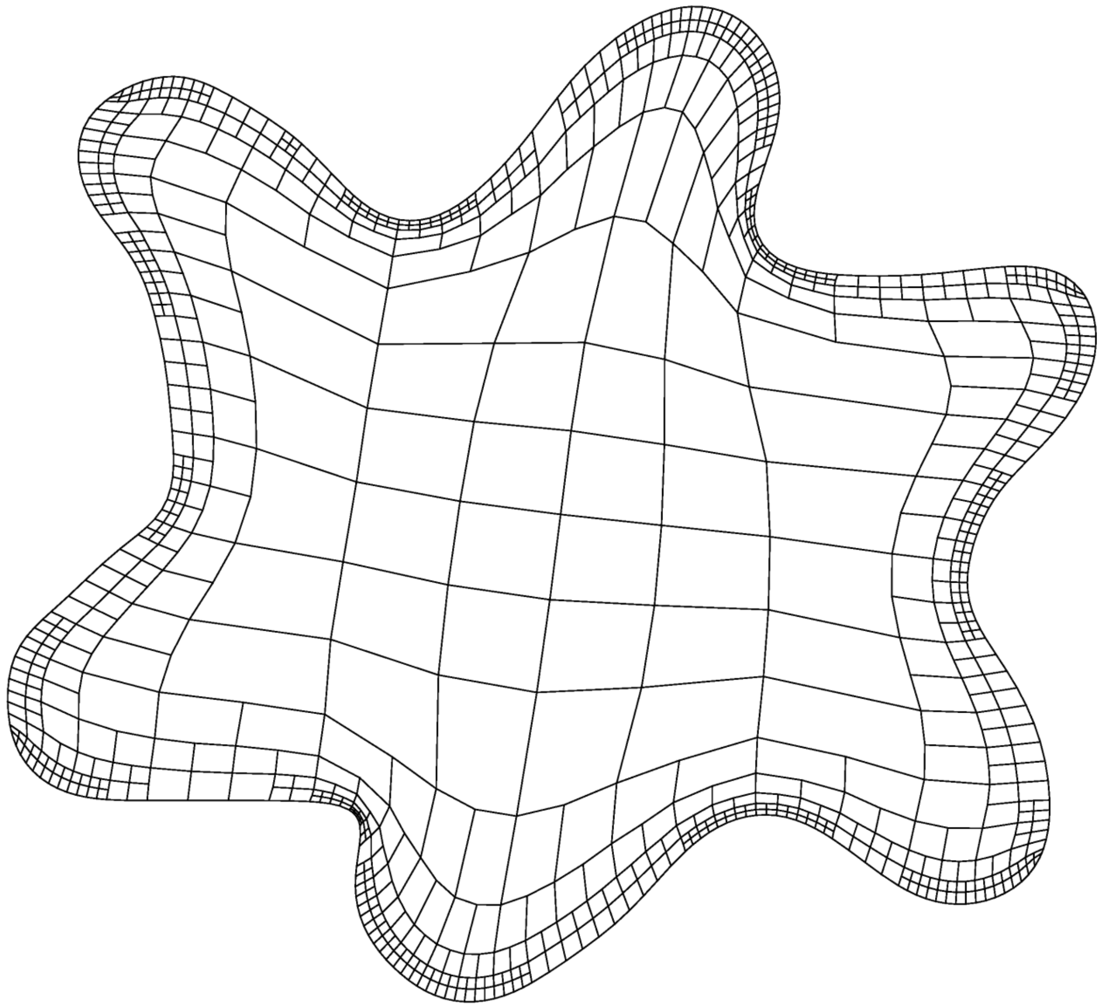




Mesh folder

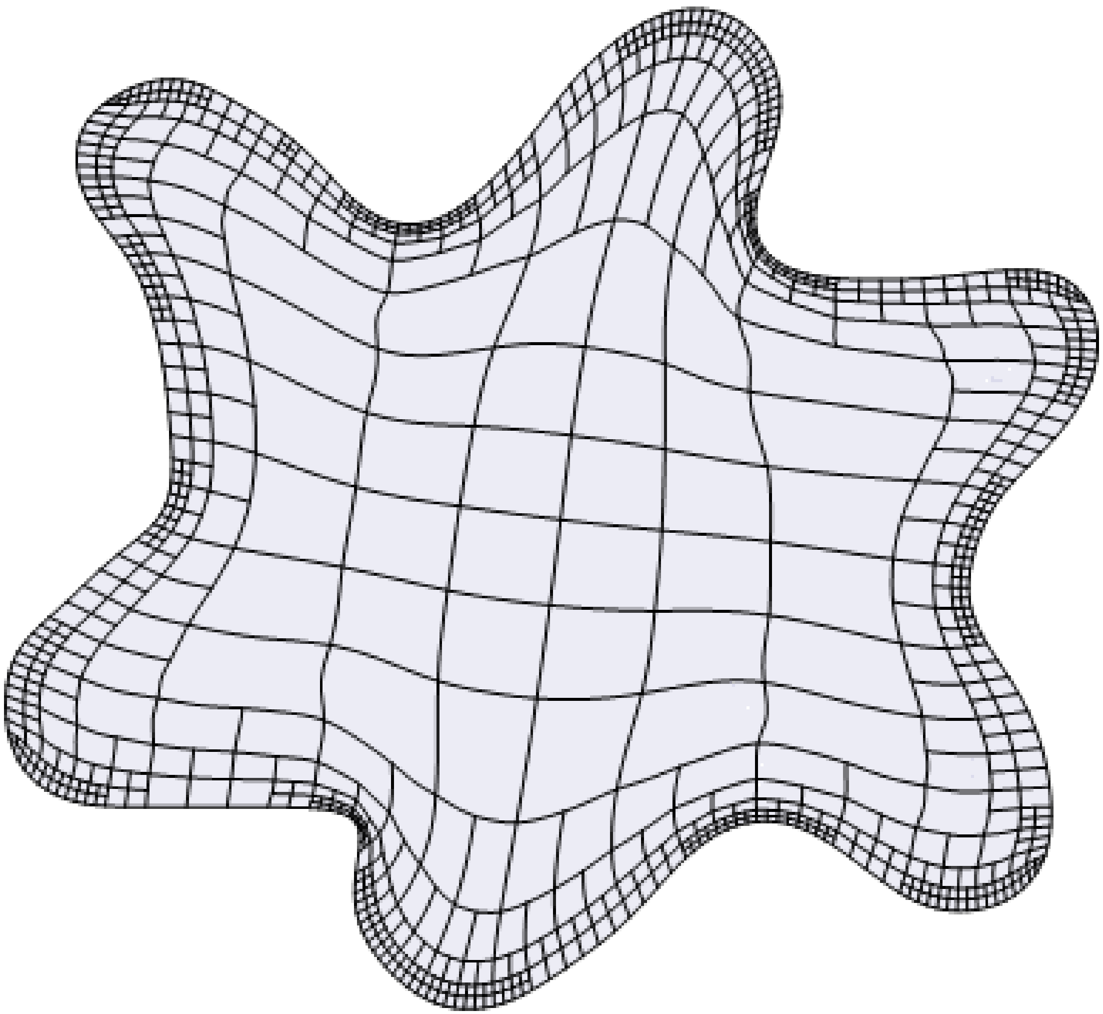
The Meccano Method on T-meshes in 2-D

Step 6: Simultaneous Untangling and Smoothing (SUS) of the T-mesh



The Meccano Method on T-meshes in 2-D

Step 7: T-spline representation of the spot

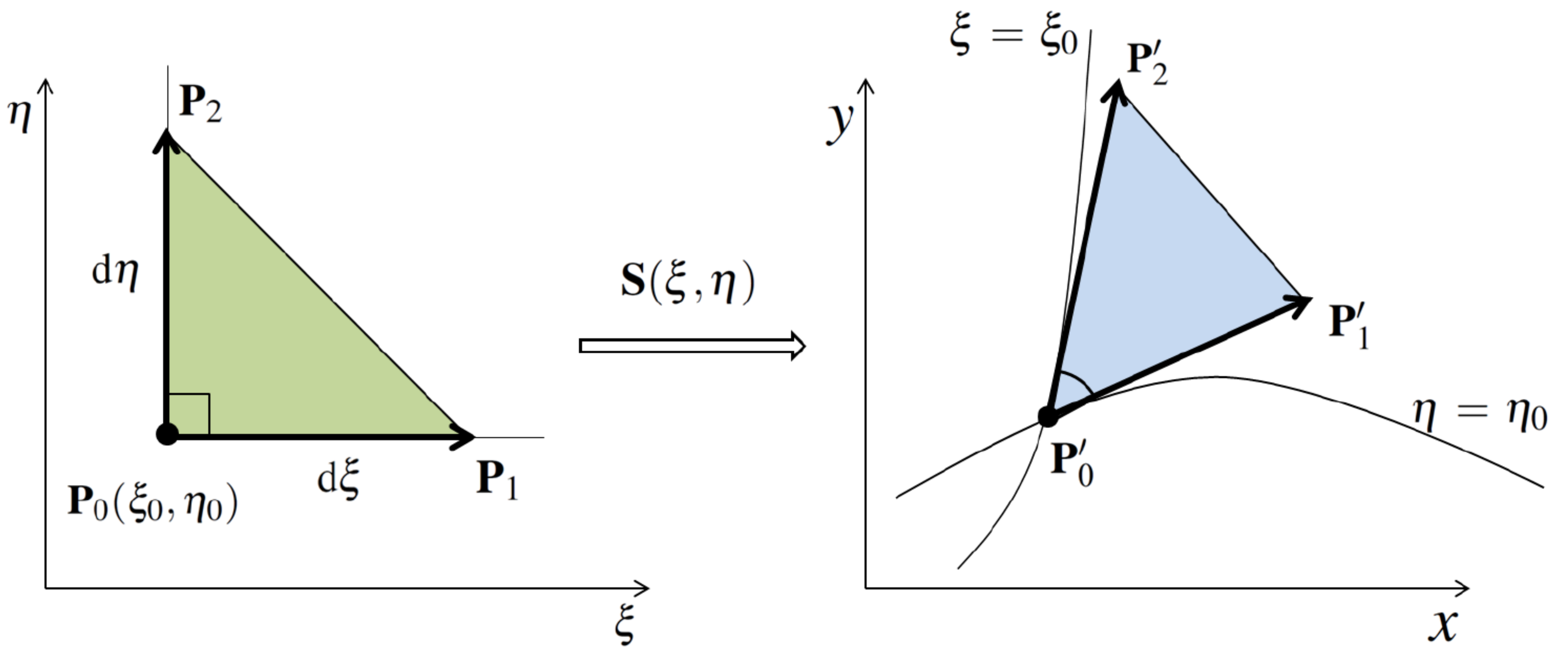


Mean Ratio Jacobian $J_r(\xi)$ of Parametric Transformation

A quality metric of the T-spline mapping at any point P_0

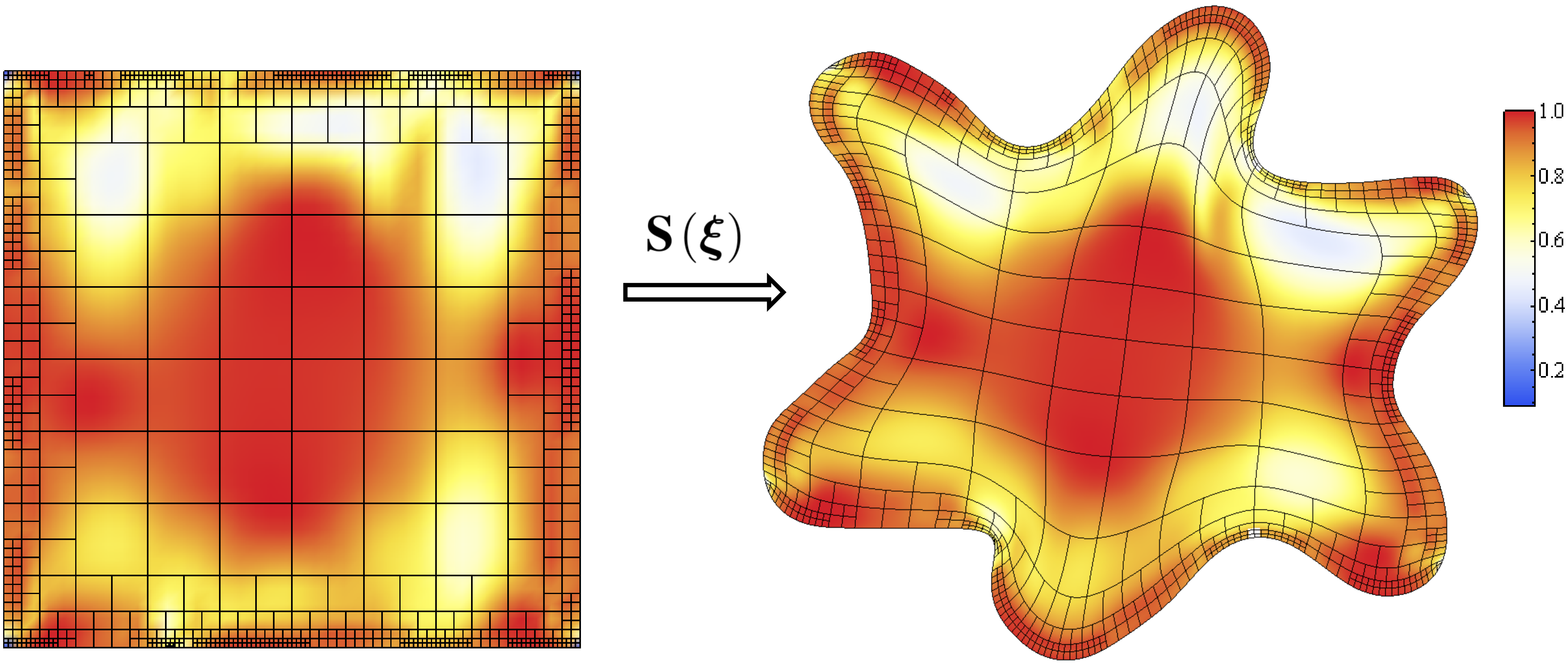
$$-1 \leq J_r(\boldsymbol{\xi}) = \frac{2\det(\boldsymbol{J})}{\|\boldsymbol{J}\|^2} \leq 1$$

where $\boldsymbol{J}$ is the jacobian matrix of the T-spline mapping $\mathbf{S}$



Good quality parameterization for application of IGA

Mean ratio Jacobian as a quality metric

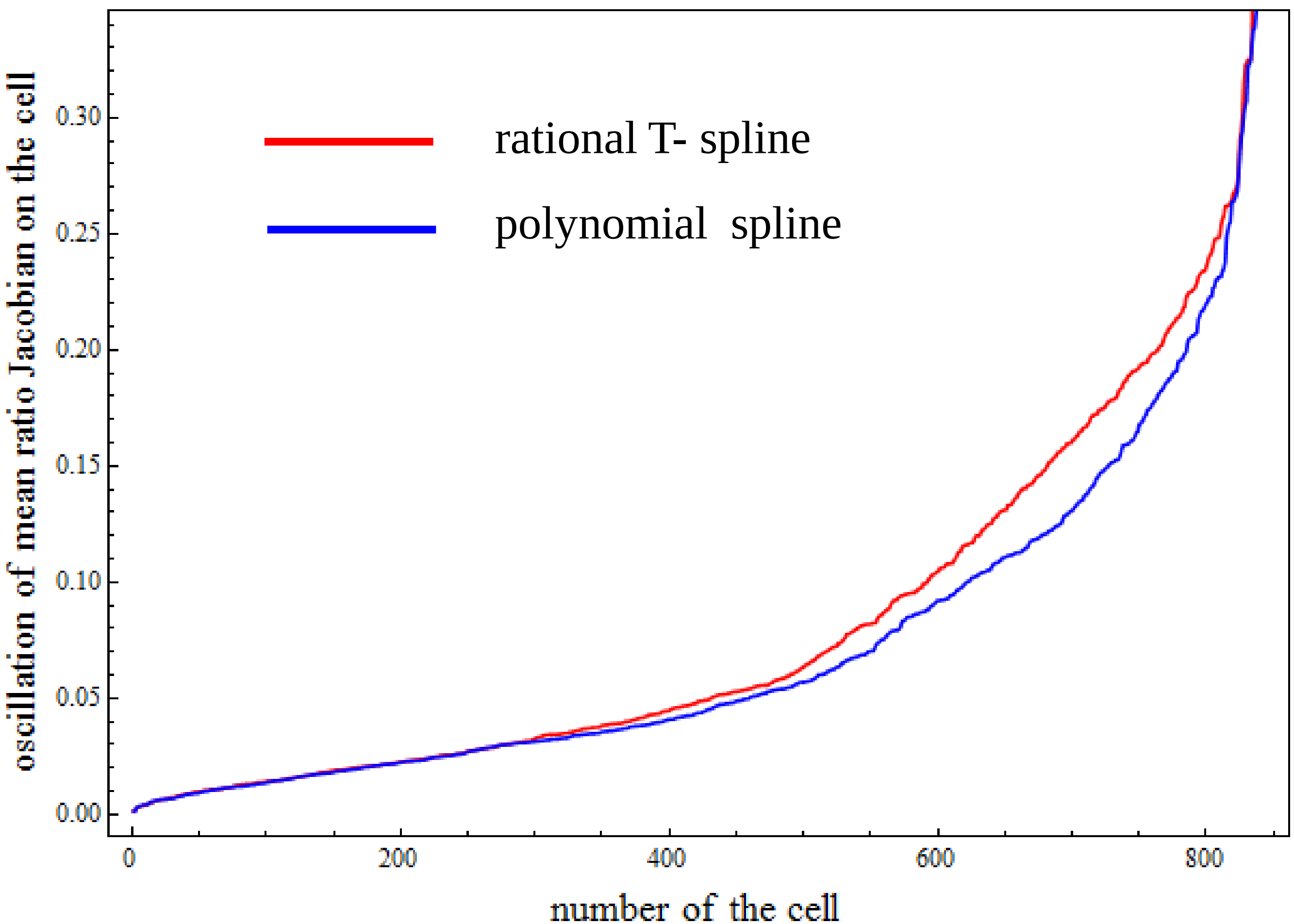


Good quality parameterization for application of IGA

Comparison of rational T-spline parameterization with the new strategy

Oscillation of mean ratio Jacobian on each cell

Representation in increasing order in the 844 cells with 64 quadrature points per cell



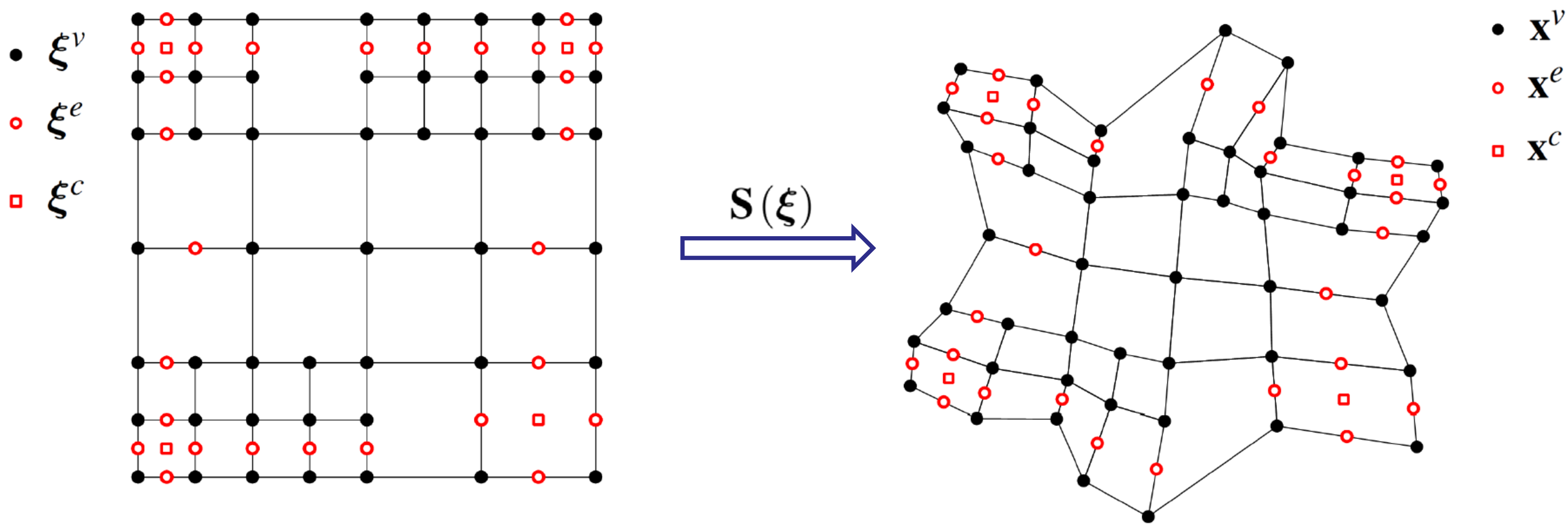
T-spline Parameterization

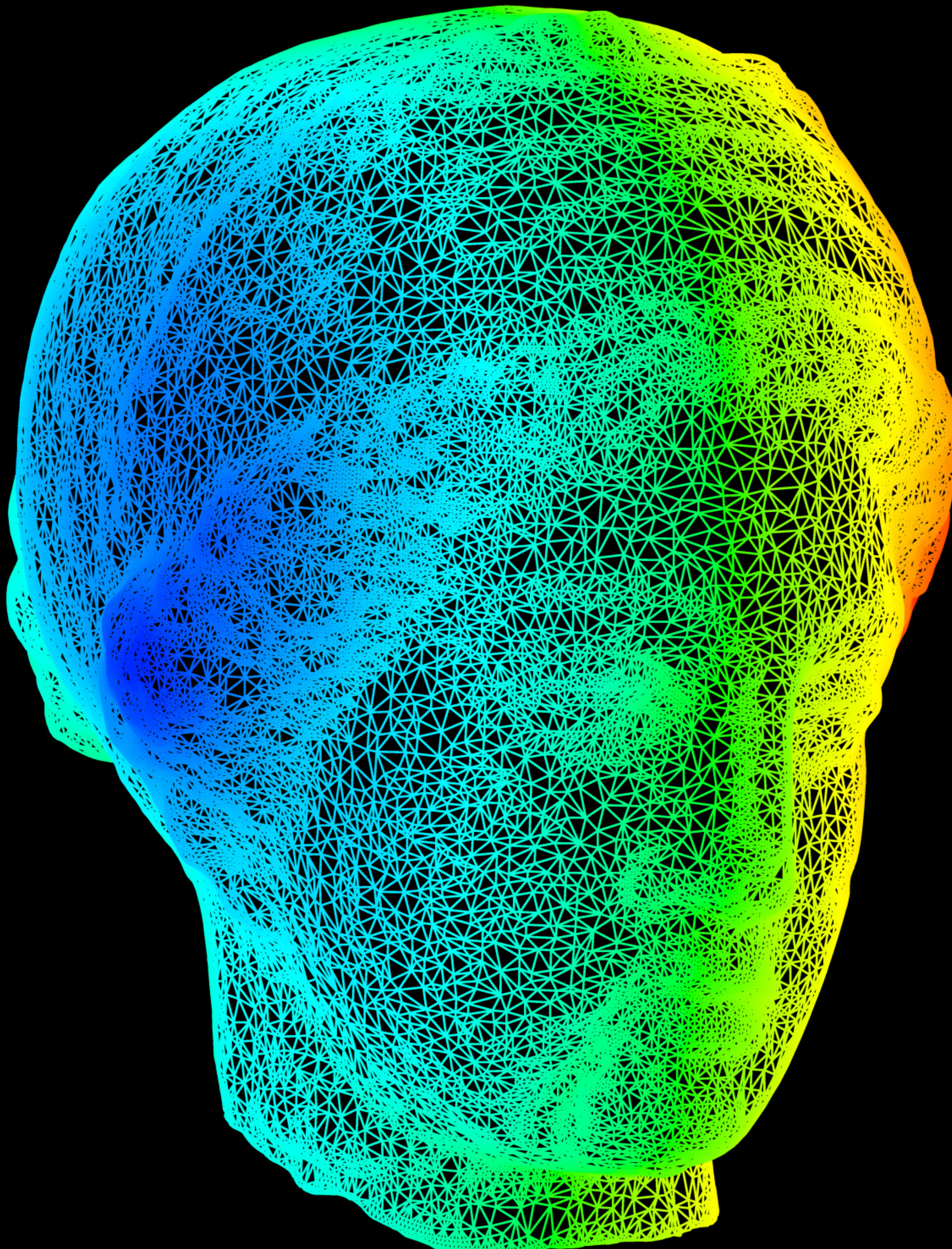
Determination of control points by imposing interpolation conditions

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\alpha \in A} \mathbf{P}_\alpha R_\alpha(\boldsymbol{\xi})$$

where

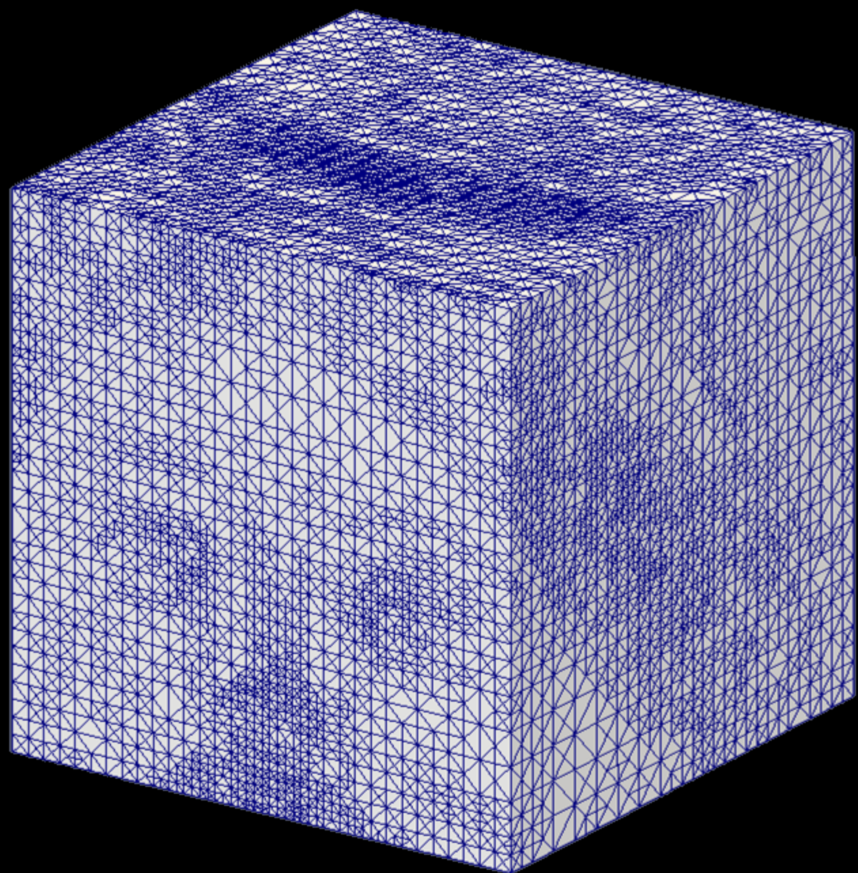
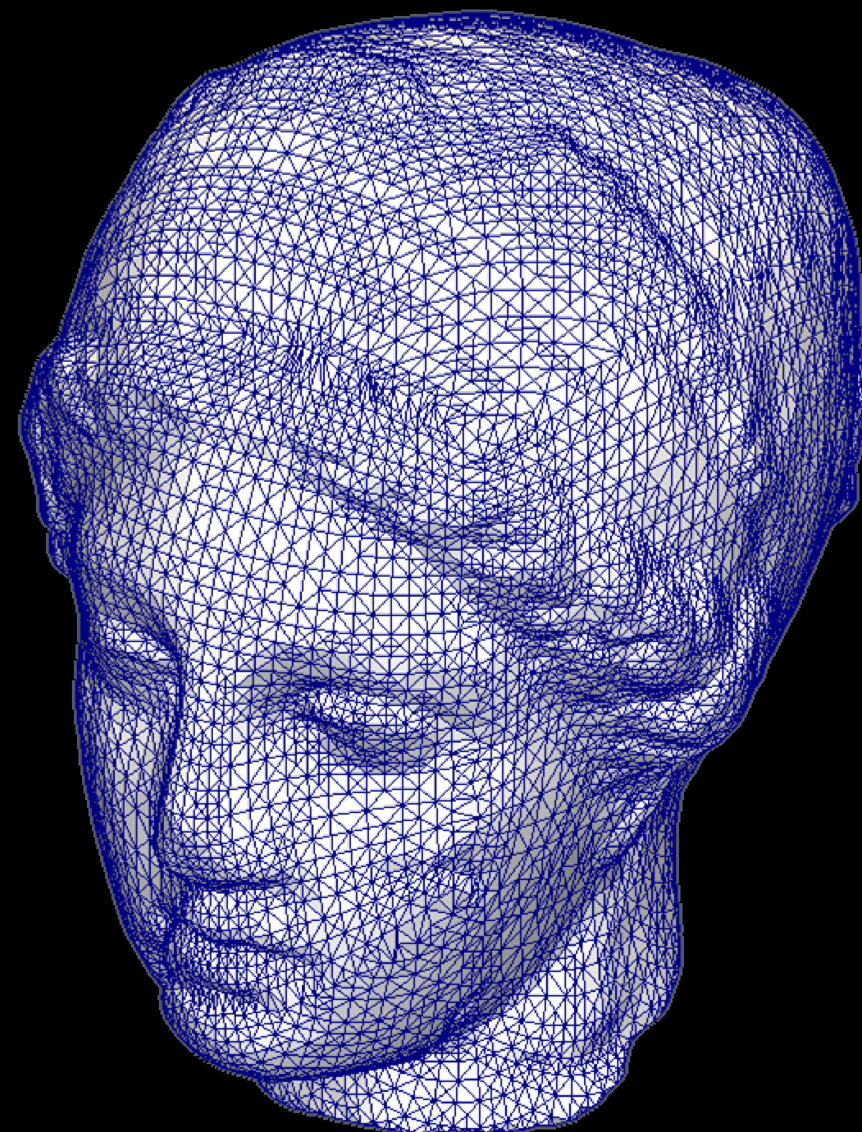
$$R_\alpha(\xi) = \frac{w_\alpha B_\alpha(\xi)}{\sum_{\beta \in A} w_\beta B_\beta(\xi)}$$

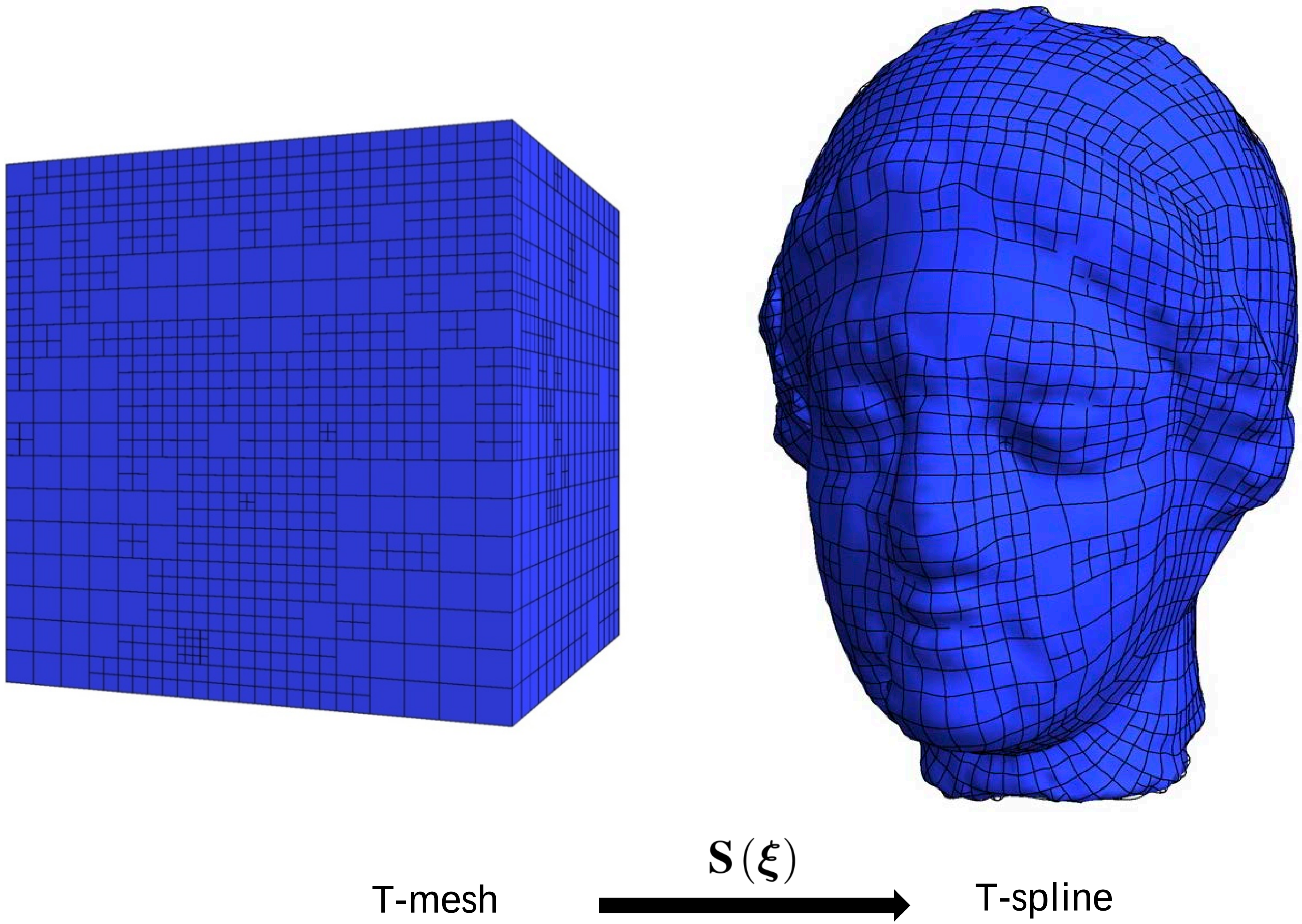




INPUT DATA: Surface Triangulation

http://www.cyberware.com/



$\mathbf{S}(\boldsymbol{\xi})$
T-mesh
T-spline

Adaptive Isogeometric Refinement (EWC 2014)

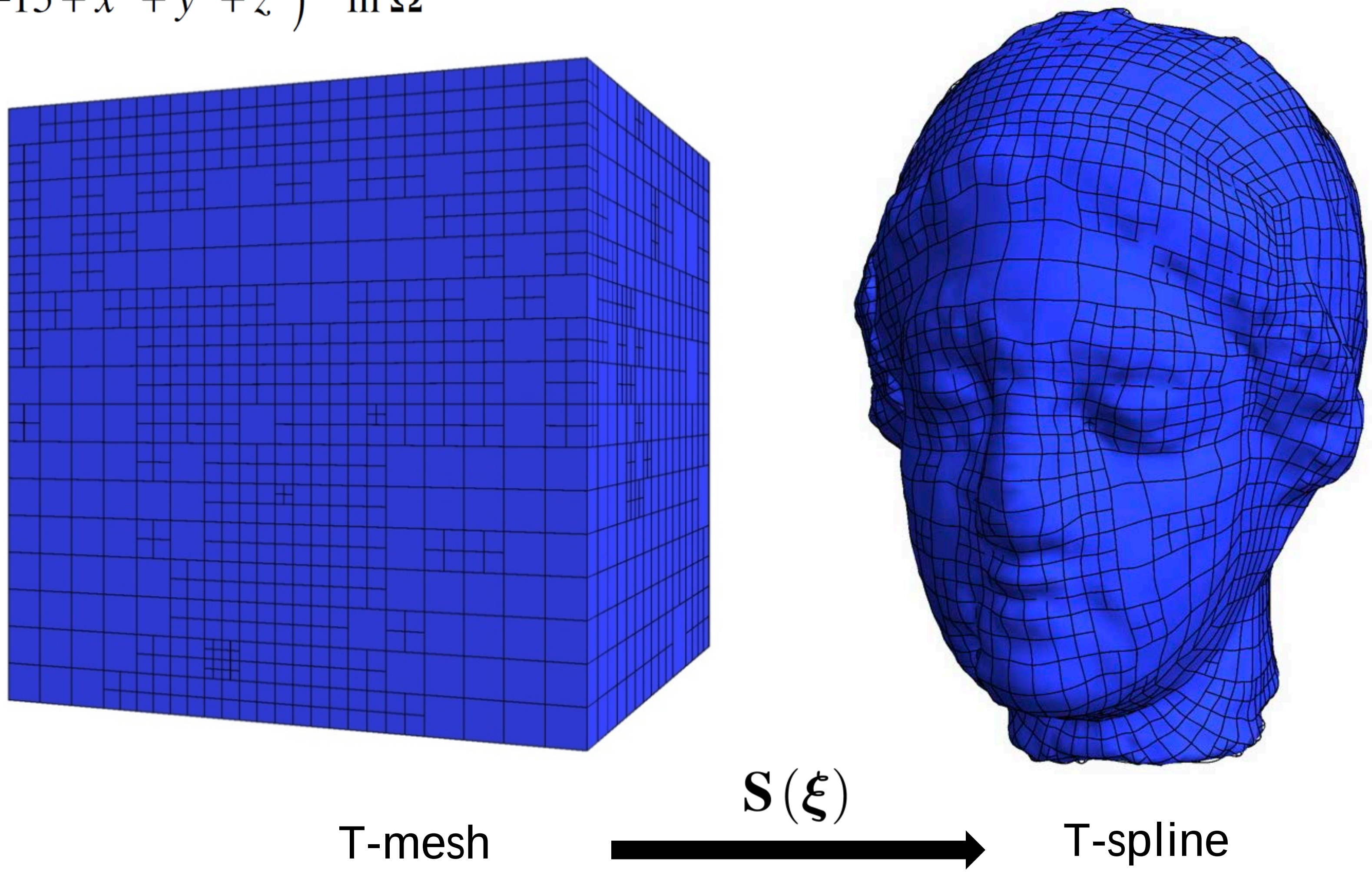
Application in Igea: Poisson problem with a central source

$$\Delta u = \frac{1}{25} e^{-\frac{\left(x^2+y^2+z^2\right)}{10}} \left(-15 + x^2 + y^2 + z^2\right) \quad \text{in } \Omega$$

$$u\big|_{\partial\Omega} = 0$$

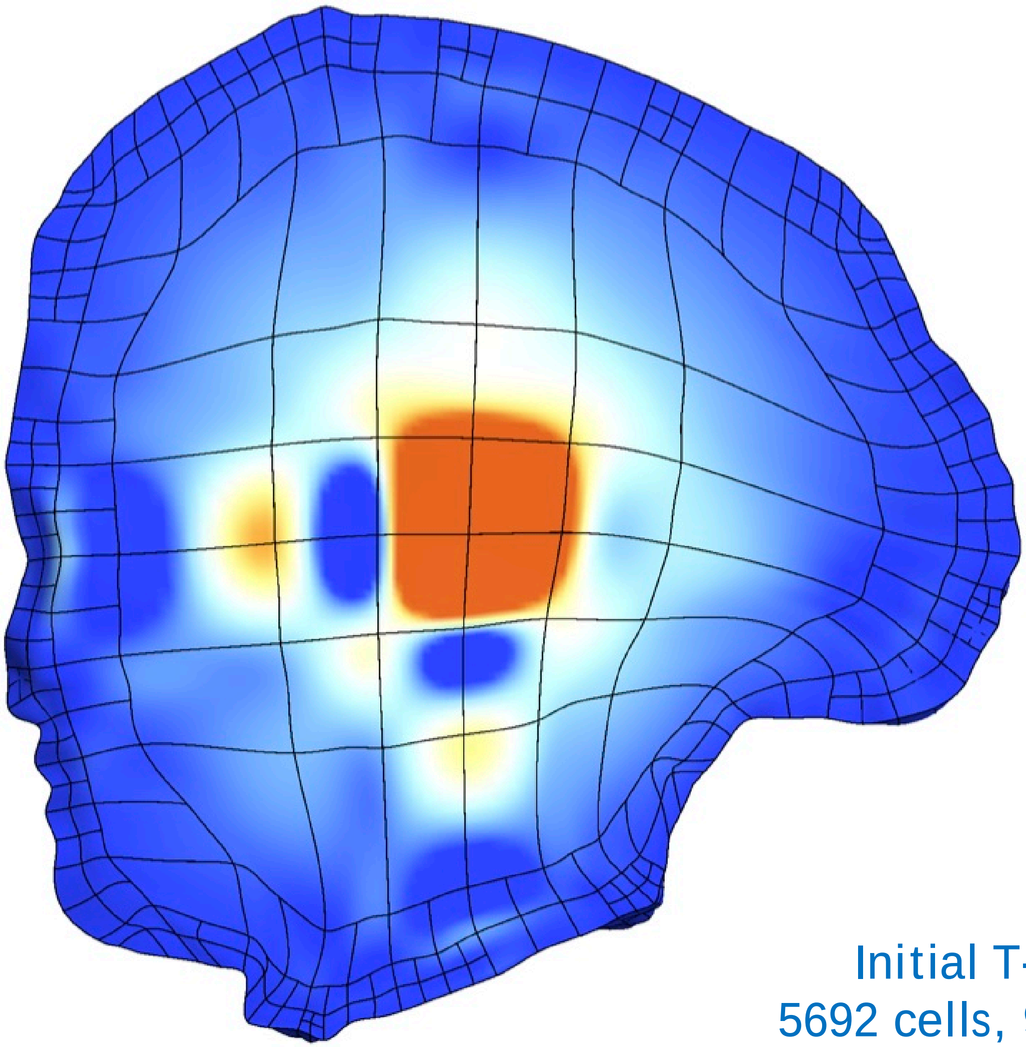
Exact solution:

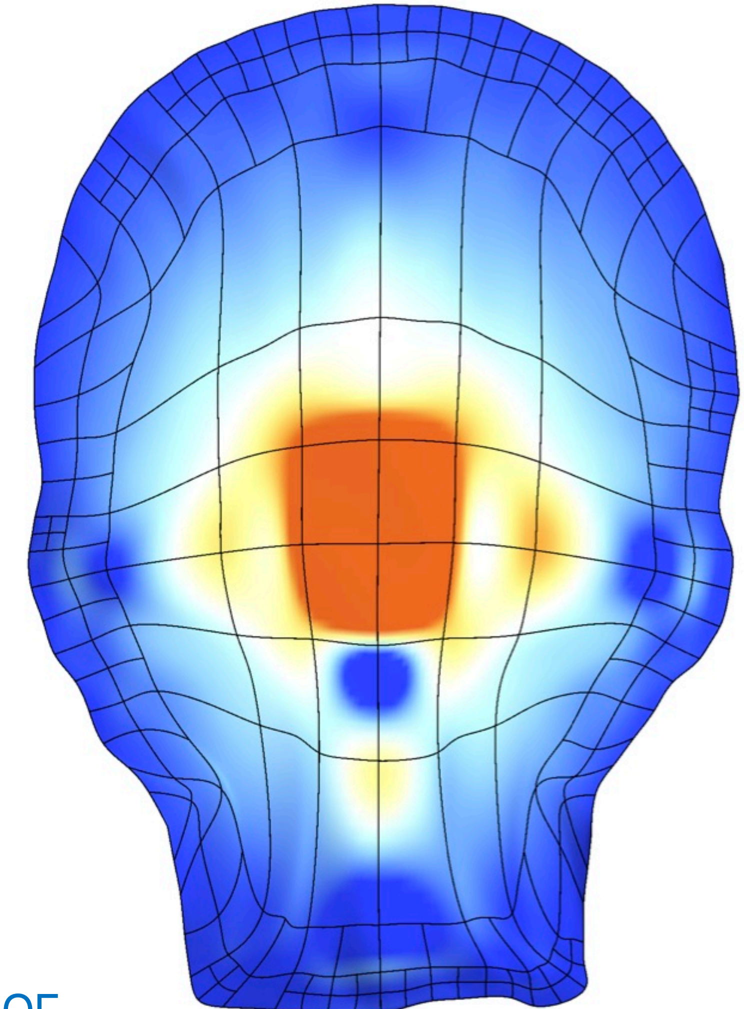
$$u \approx e^{-\frac{\left(x^2+y^2+z^2\right)}{10}}$$



Adaptive Isogeometric Refinement (EWC 2014)

Igea: T-spline of Numerical Solution



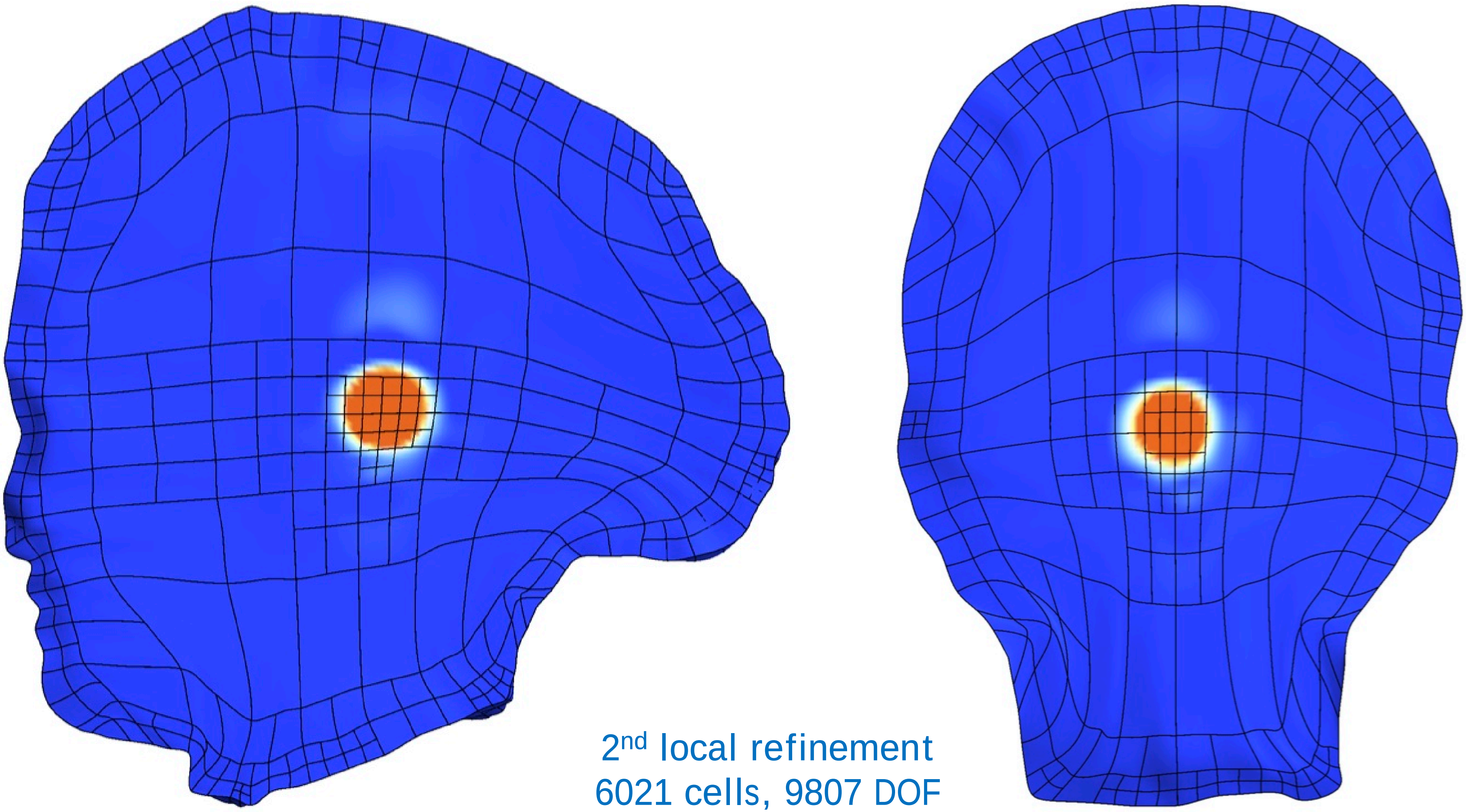


Initial T-mesh
5692 cells, 9304 DOF

Adaptive Isogeometric Refinement (EWC 2014)

Igea: T-spline of Numerical Solution

Error indicator : $\eta\left(\Omega_e\right)^2 = \int_{\Omega_e} h^2 \left(f + \Delta u_h\right)^2 \, d\Omega$



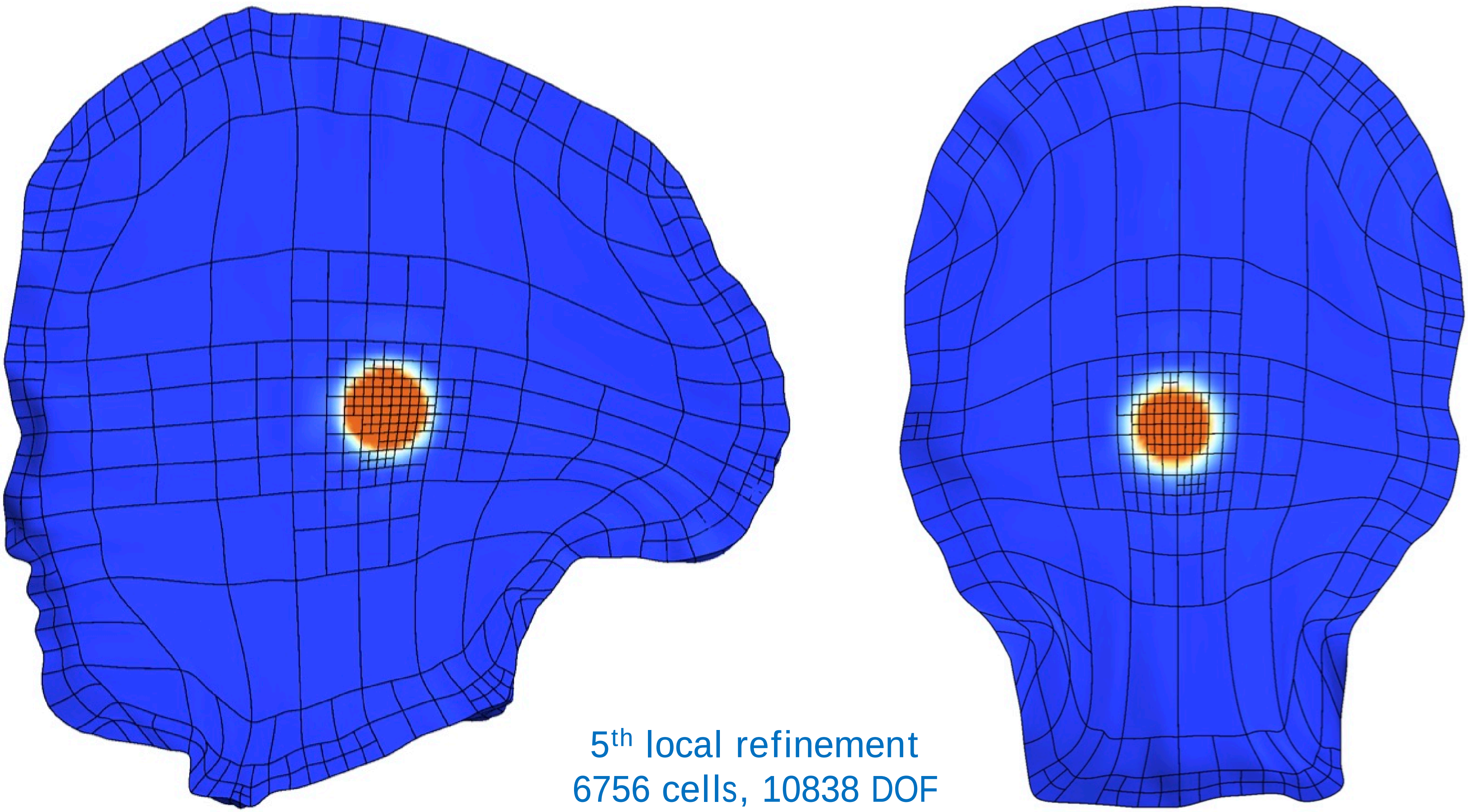
2nd local refinement
6021 cells, 9807 DOF

Adaptive Isogeometric Refinement (EWC 2014)

Igea: T-spline of Numerical Solution

Error indicator : $\eta\left(\Omega_e\right)^2 = \int_{\Omega_e} h^2 \left(f + \Delta u_h\right)^2 \, d\Omega$

SIANI



5th local refinement
6756 cells, 10838 DOF