# To mesh or not to mesh: that is the question

Stefan Kollmannsberger
Computation in Engineering
Technische Universität München

# Computation in Engineering

- **Ordinarius:** Prof. Dr.rer.nat. Ernst Rank
- **Numerical Mechanics:**
  Dr.-Ing. Stefan Kollmannsberger, Tino Bog, Nils Zander, Quanji Cai, Martin Schlaffer, Christian Sorger, Hagen Wille,
- **Efficient Algorithms:**
  *Dr.rer.nat. Ralf Mundani,* Jérôme Frisch, Jovana Knesevic, Dominik Schillinger, Vasco Varduhn, Matthias Flurl
- **Bavarian Graduate School of Science and Engineering**
  Dr.-Ing. Martin Ruess

blue: working on FCM red: Mesh Generation

# Computation in Engineering

**Statement:**

**as stated orally during the talk, all presented work is the result of a common effort of the research group and may not only be attributed to the presentor.**

Statement included for the online version of the presentation on Tuesday, 20th of October 2011

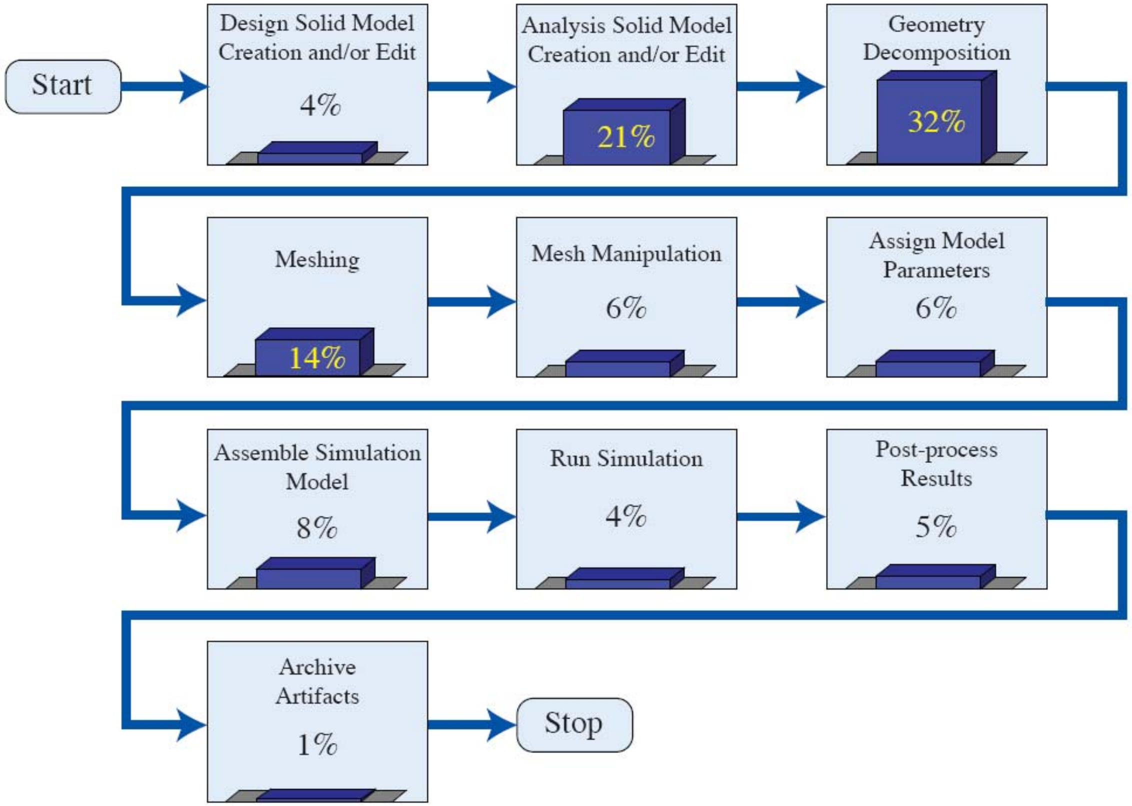
# Outline

Motivation

- the „traditional“ approach incl. mesh generation
- modeling with Isogeometric Analysis
- modeling with Finite Cells

what am I doing here?

# Motivation




**mesh generation**

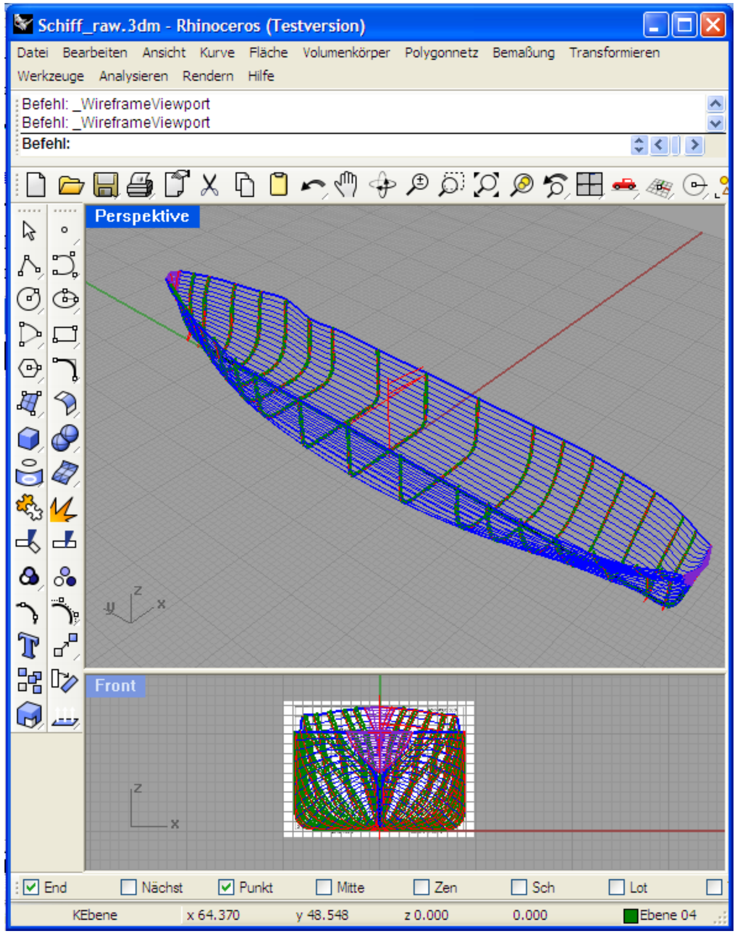
takes more time than

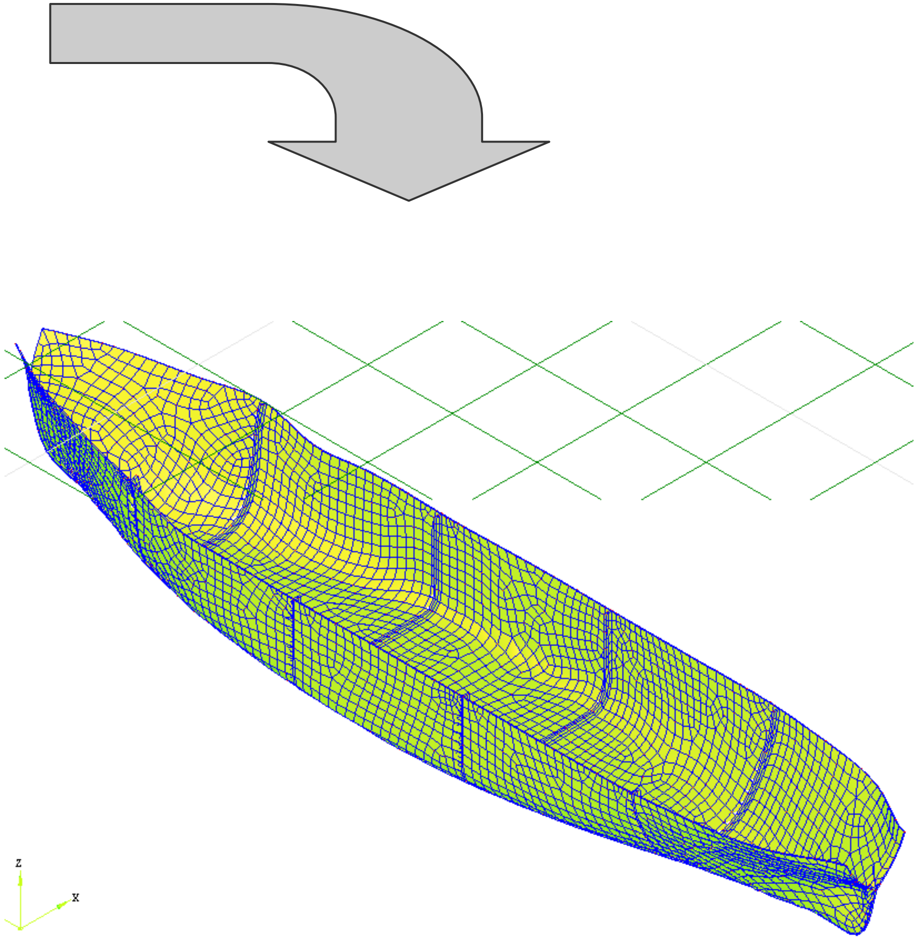
**computation**

Michael Hardwick and Robert Clay, Sandia National Laboratories, as published in:

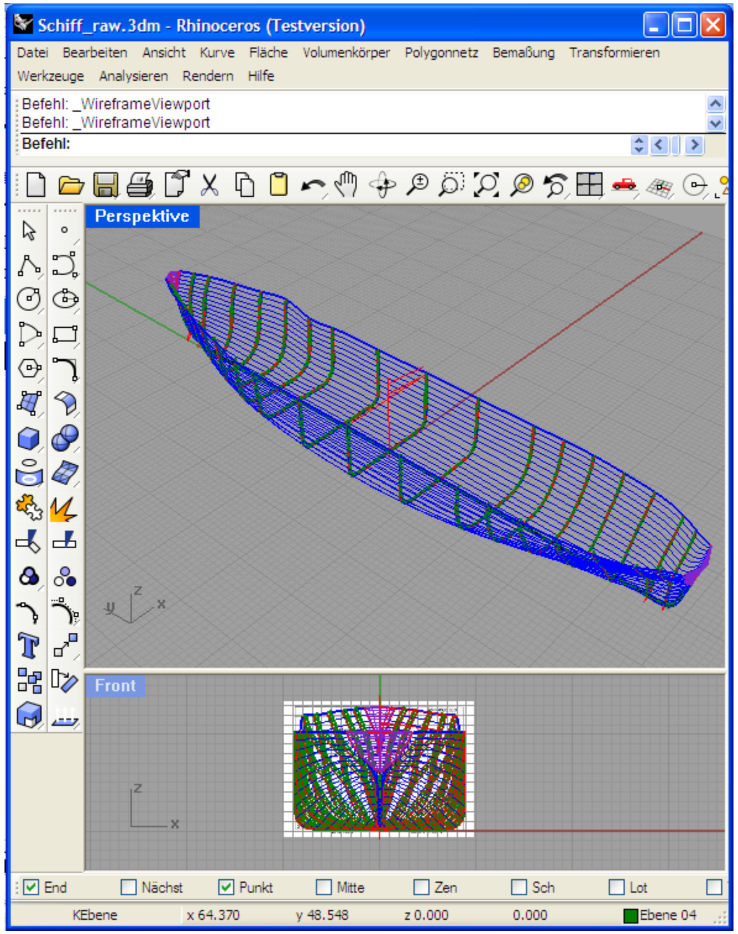
J. Austin Cottrell, Thomas J.R. Hughes, Y. Bazilevs. Isogeometric Analysis. Wiley, 2008, ISBN 978-0-470-74873-2

# In general, the problem is…

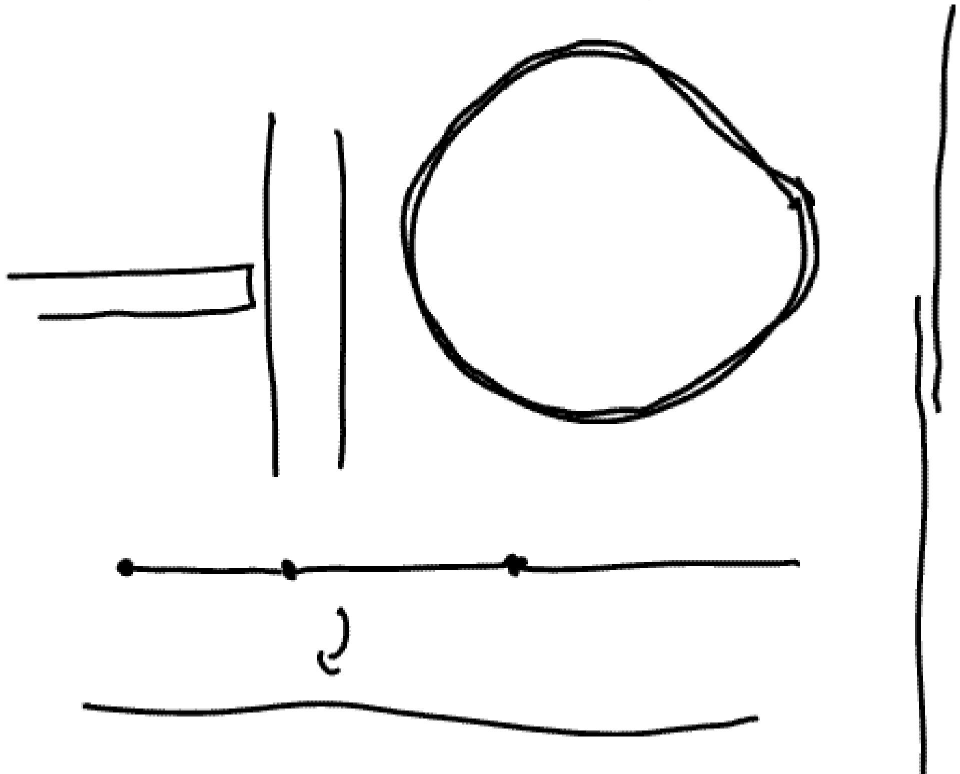


# and the major assumption is:

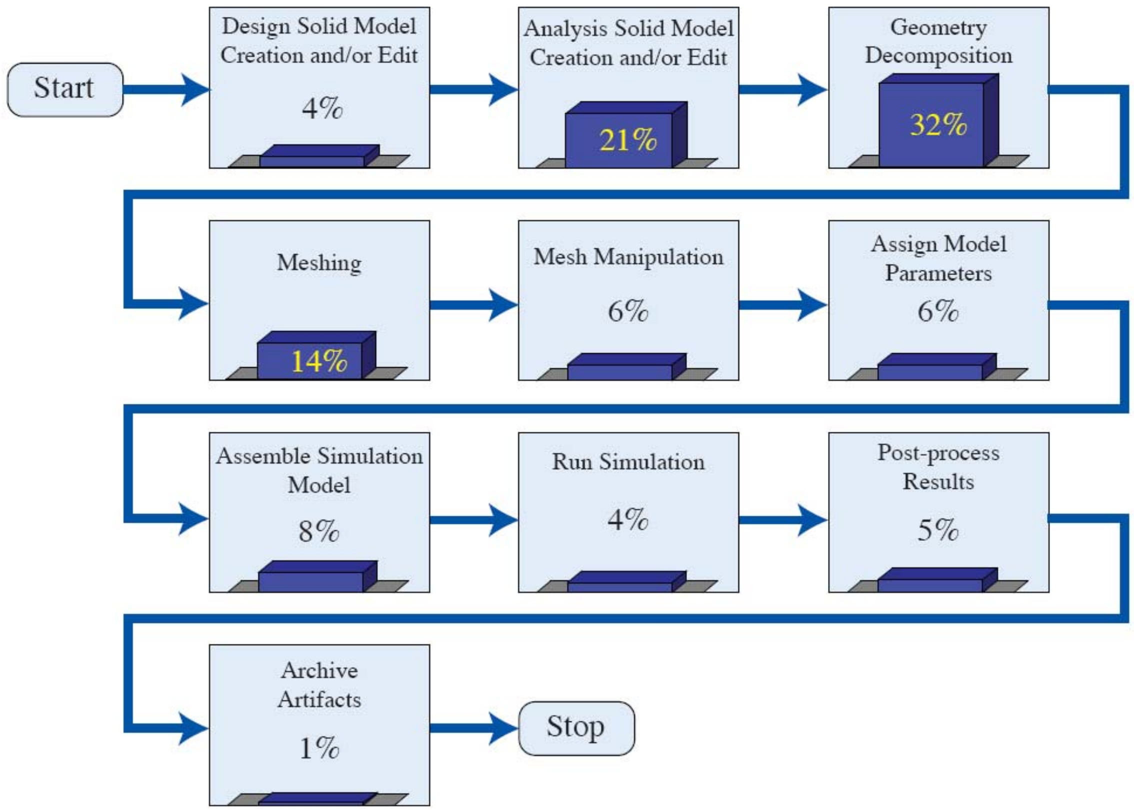



we start with a clean! geometry



be „analysis aware“ or fix it
-> healing is an important step

# Motivation




**mesh generation**

takes more time than

**computation**

Michael Hardwick and Robert Clay, Sandia National Laboratories, as published in:

J. Austin Cottrell, Thomas J.R. Hughes, Y. Bazilevs. Isogeometric Analysis. Wiley, 2008, ISBN 978-0-470-74873-2

# a) better mesh generation (i.e. for high order solid shells)

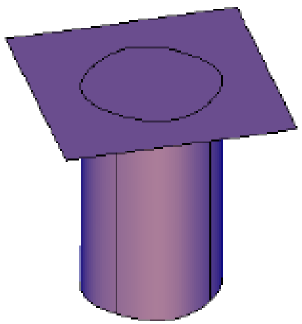
Historic “torch relay” at our chair

Rank, Schweingruber, Halfmann, Scholz,

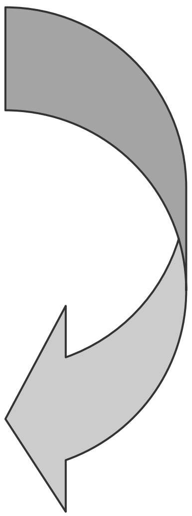
Kollmannsberger, Sorger….

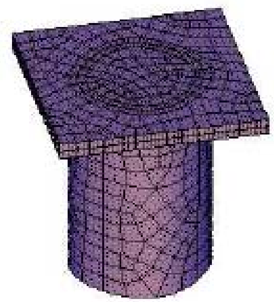
funded by SOFiSTiK

# a) better mesh generation (i.e. for high order solid shells)

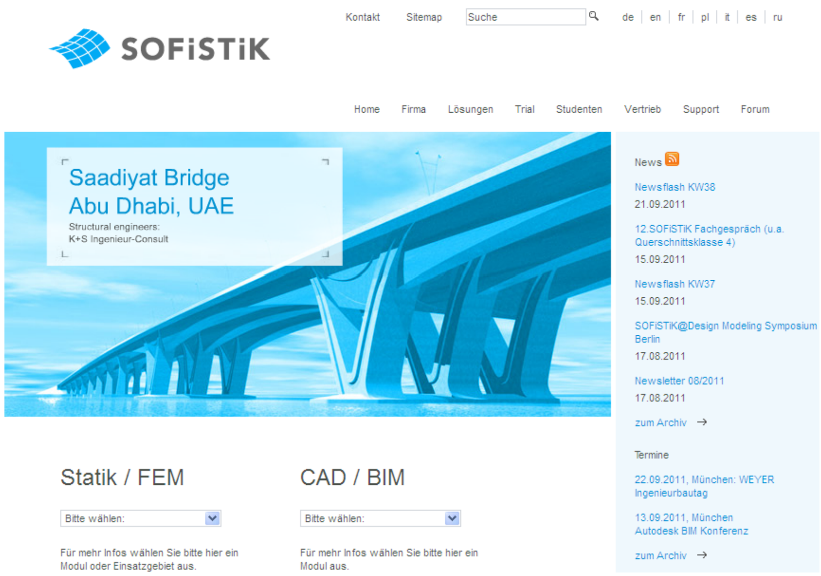


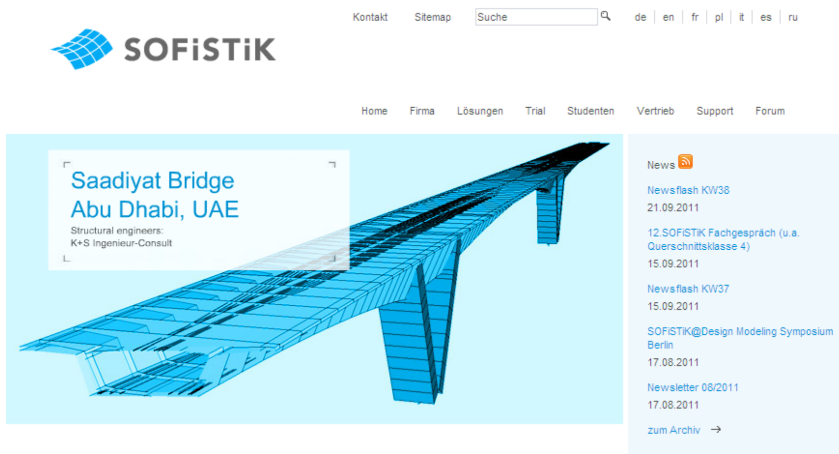
(a)



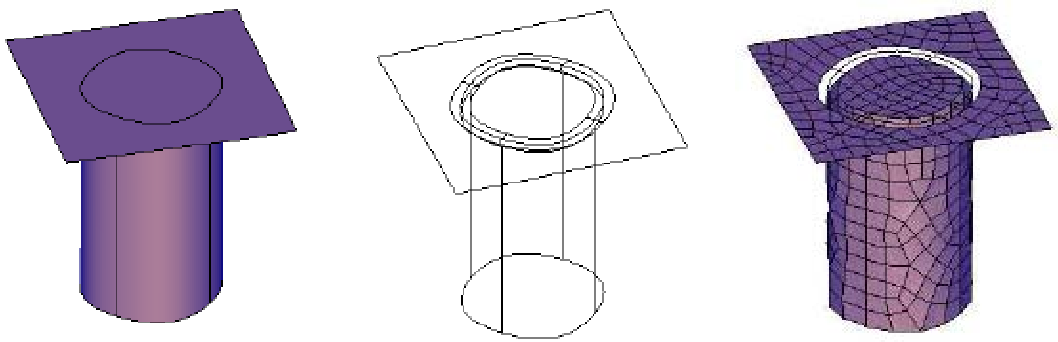


(h)

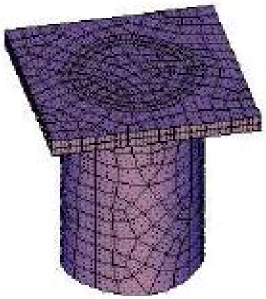


# a) better mesh generation (i.e. for high order solid shells)



(a) (d) (e)

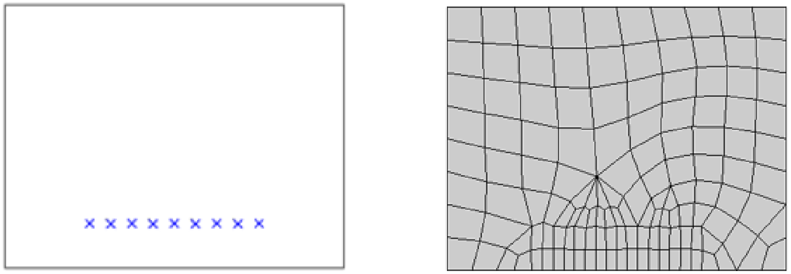


(h)

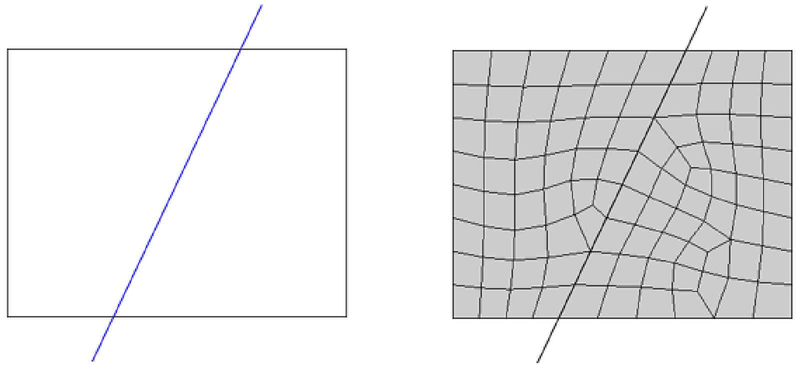
how do we generate a mesh?

Topology
Geometry

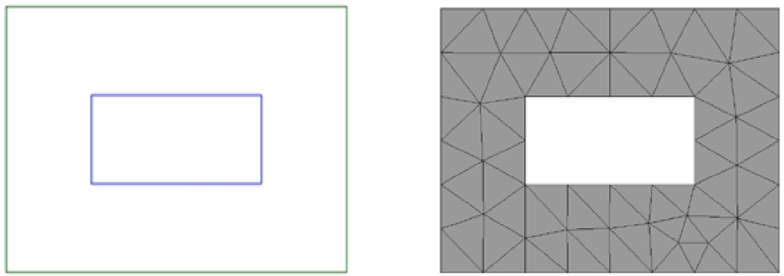
# topological components of mesh generation



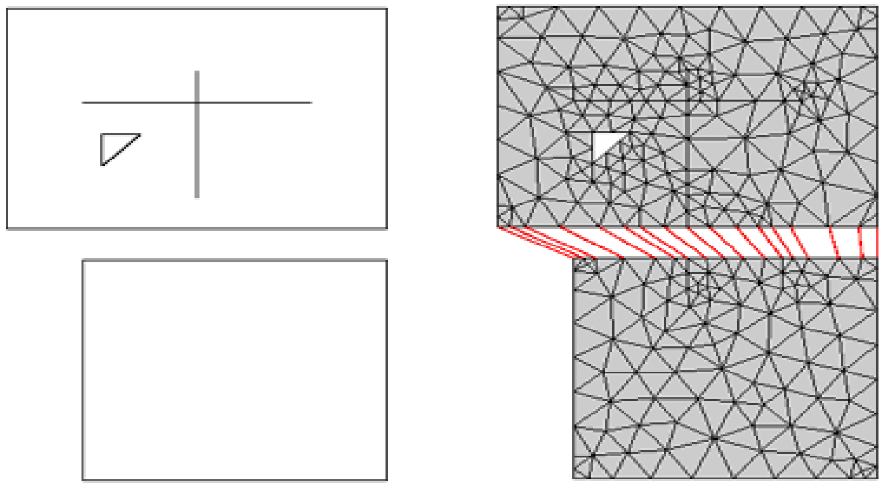
Fixed points



Fixed lines

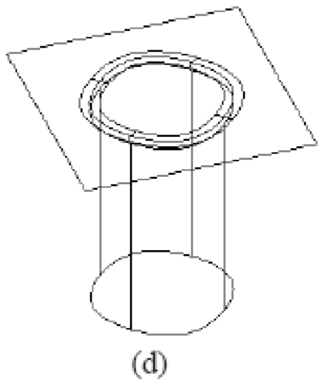


Holes



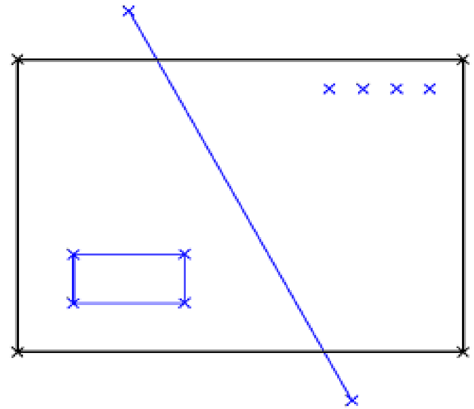
Reference edges

# Topology: Pre-processing

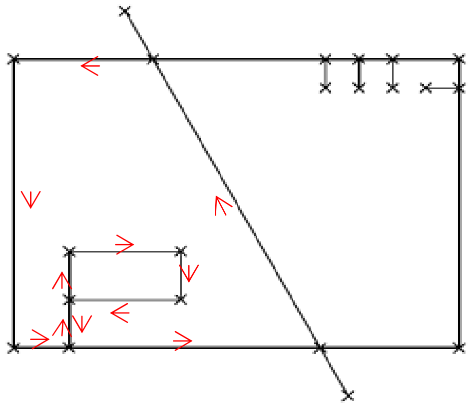



```
Nodes:       Id,   X,       Y,     Z
             ...,  ...,     ...,   ...

Edges:       Id:   NodeA, NodeB, Attribute
             ...,  ...,     ...,   ...

Regions:     Id,   Edges
             ...,  ...

Holes:       Id,   Edges, Reg_nr
             ...,  ...,   ...

Fix-lines:   Id,   Edges, Reg_nr
             ...,  ...,   ...

Fix-points:  Id,   Nodes, Reg_nr
             ...,  ...,   ...
```

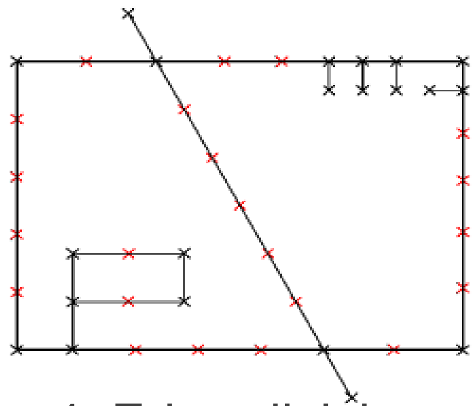
1. Raw data



2. Initial geometry



3. Closed polygons



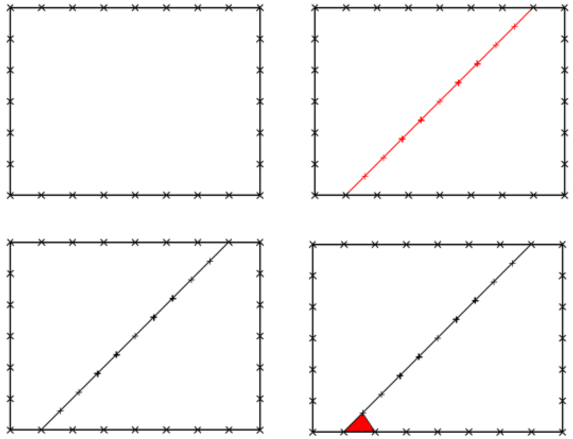
4. Edge division

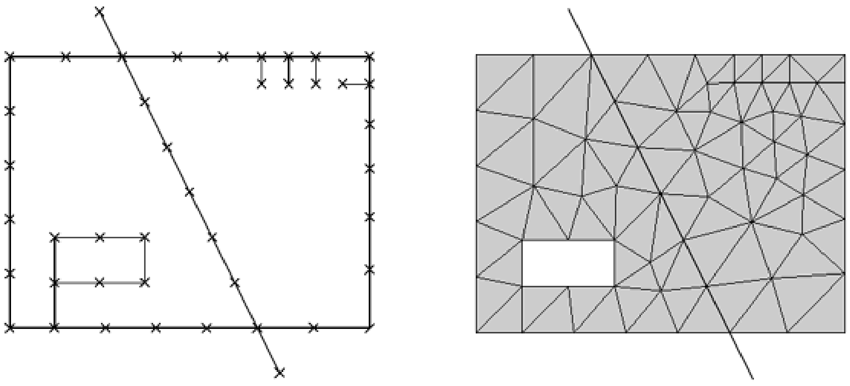
# Topology: recursive subdivision

recursive subdivision:

→ Divide

→ Chop

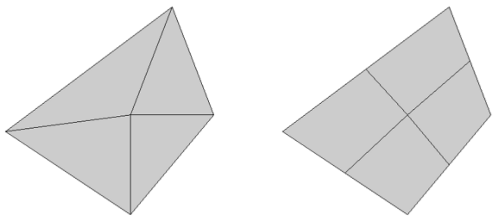




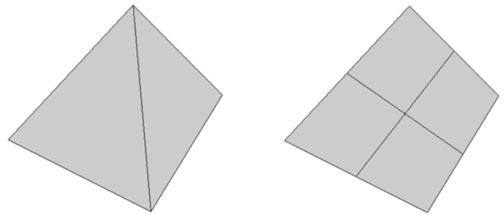
original idea: R. E. Bank, et. al 1983

# Topology: recursive subdivision

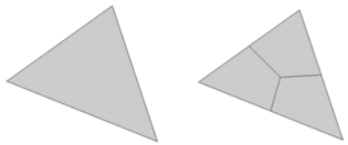
triangle conversion:



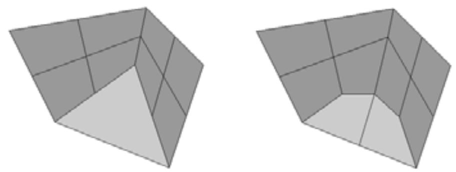
→ **Variant 1:** 4 triangles → 4 quadrilaterals



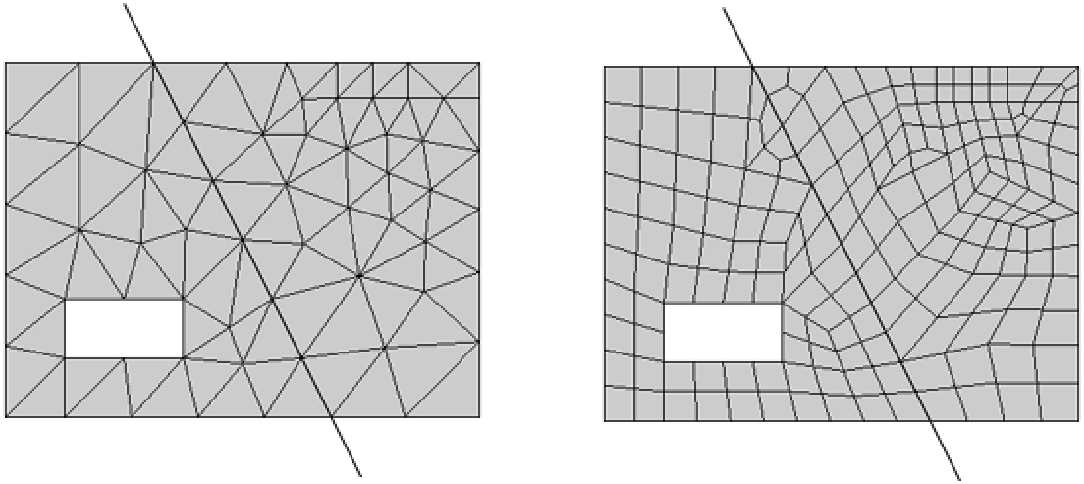
→ **Variant 2:** 2 triangles → 4 quadrilaterals



→ **Variant 3a:** 1 triangles → 3 quadrilaterals

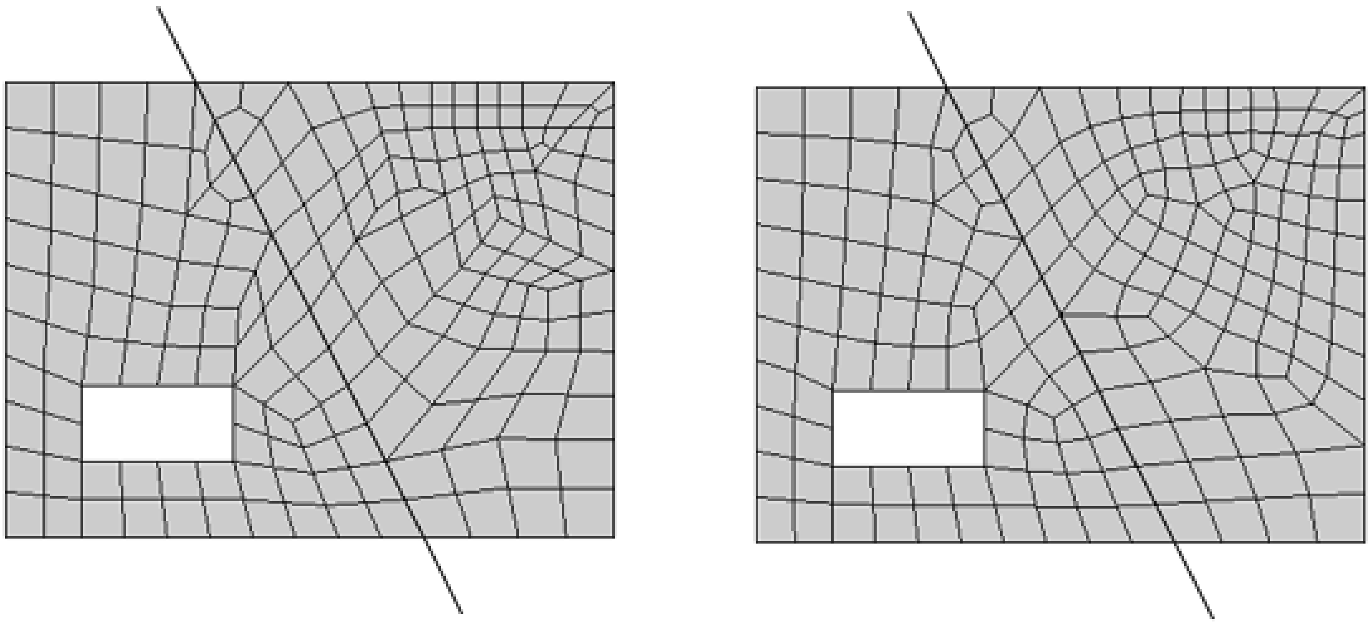


→ **Variant 3b:** 1 triangles → 2 quadrilaterals

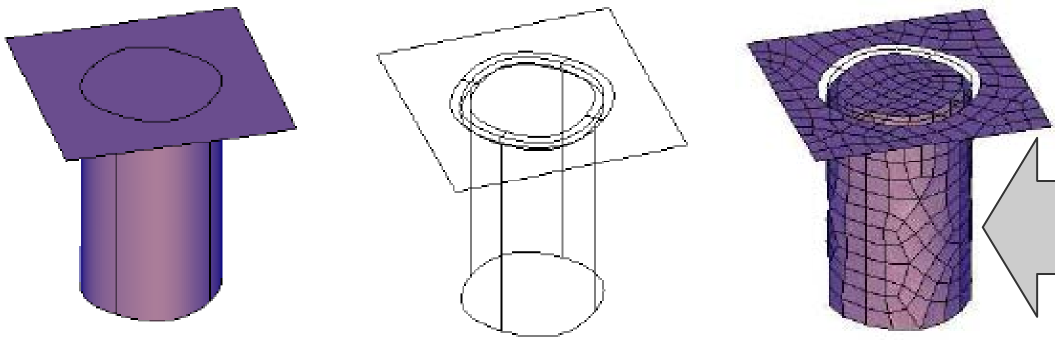


# Topology: recursive subdivision

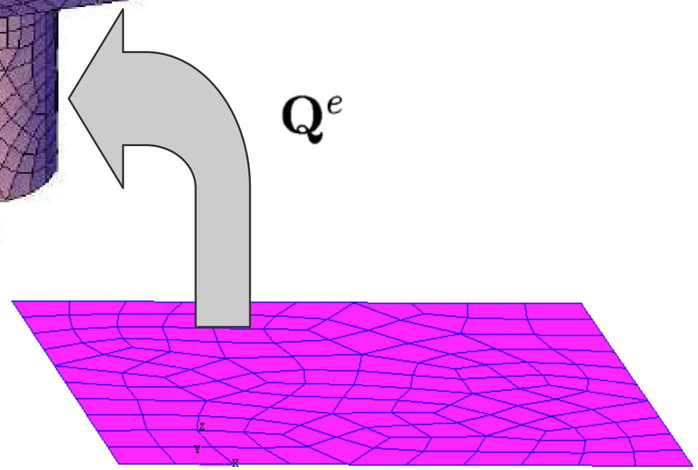
relaxation:



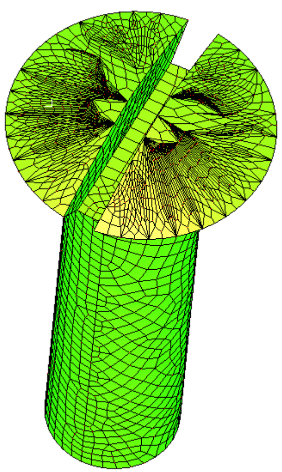
# Geometry: mapping



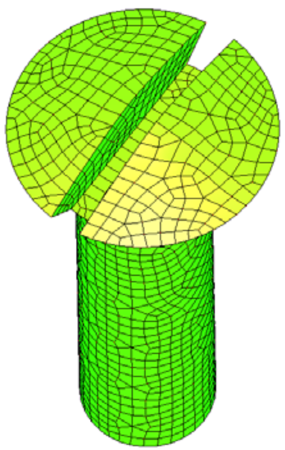
(a) (d) (e)



$\mathbf{Q}^e$



without metric



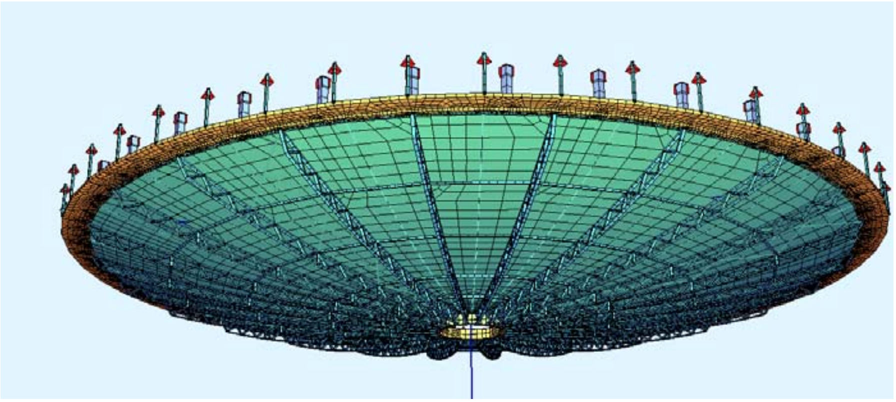
with metric

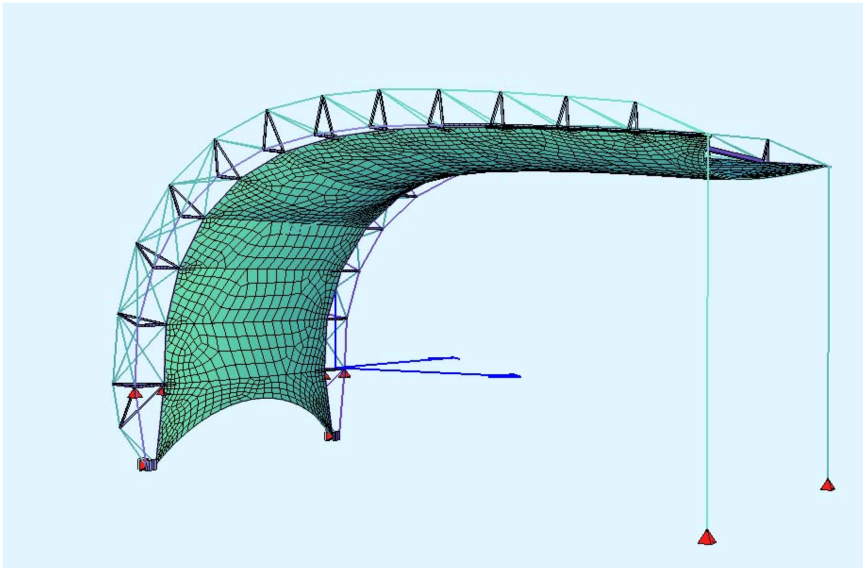
$\mathbf{Q}^e$ is a mapping describing the true geometry i.e. B-splines, NURBS…

do this for n regions…

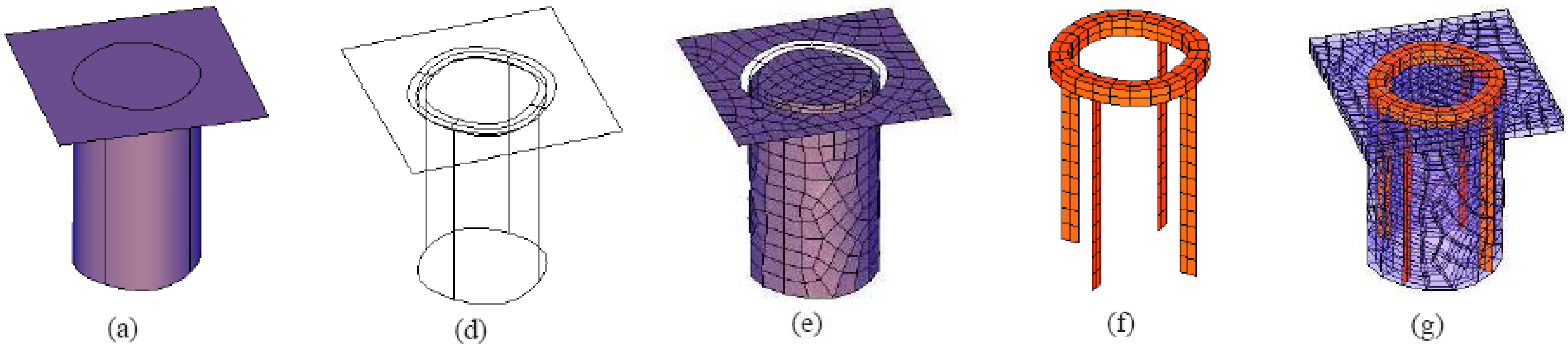
+ adaptivity
+ respect the matrix of the mapping in the meshing

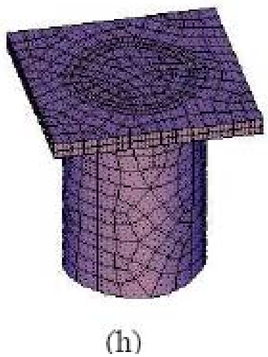
# a) better mesh generation… for the industry…

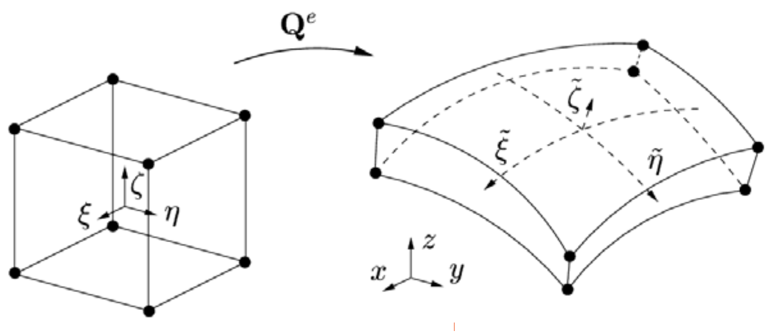


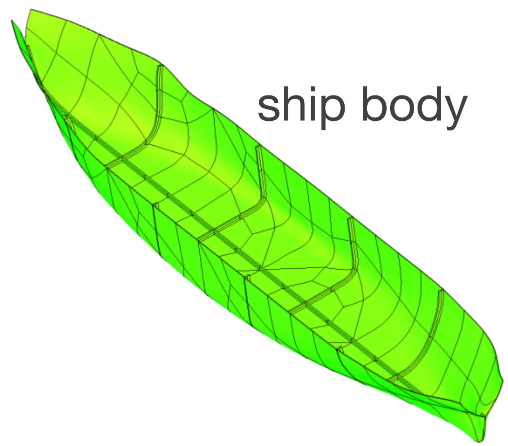


# a) better mesh generation (i.e. for high order solid shells)



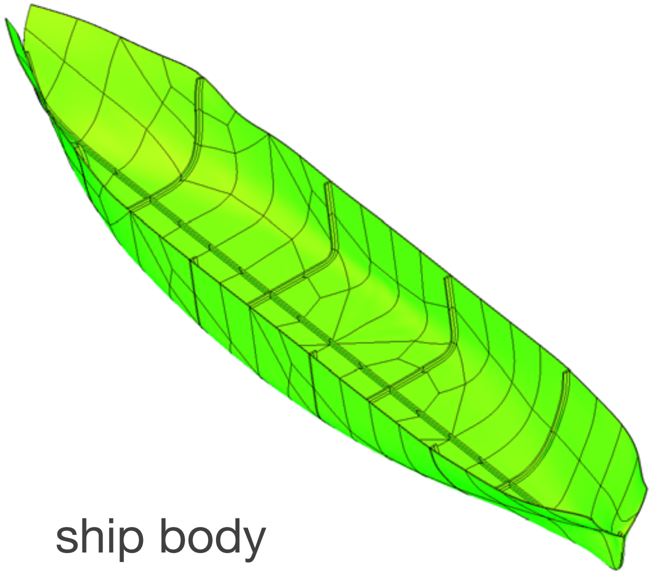



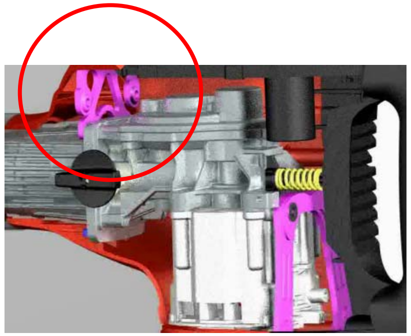




$\mathbf{Q}^e$ is a mapping describing the true geometry i.e. B-splines, NURBS…

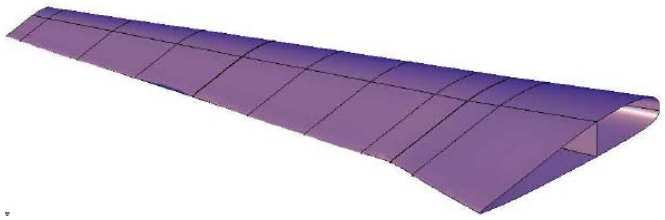
# a) better mesh generation: examples



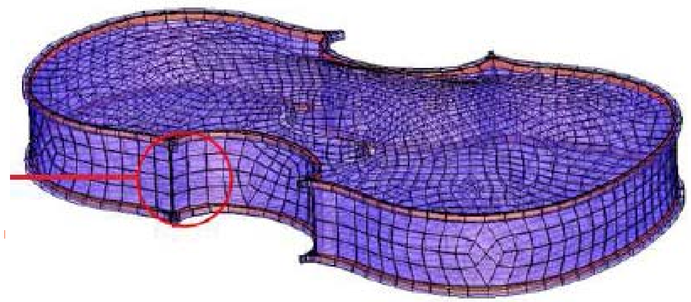
ship body



mechanical spring for a chiseling tool

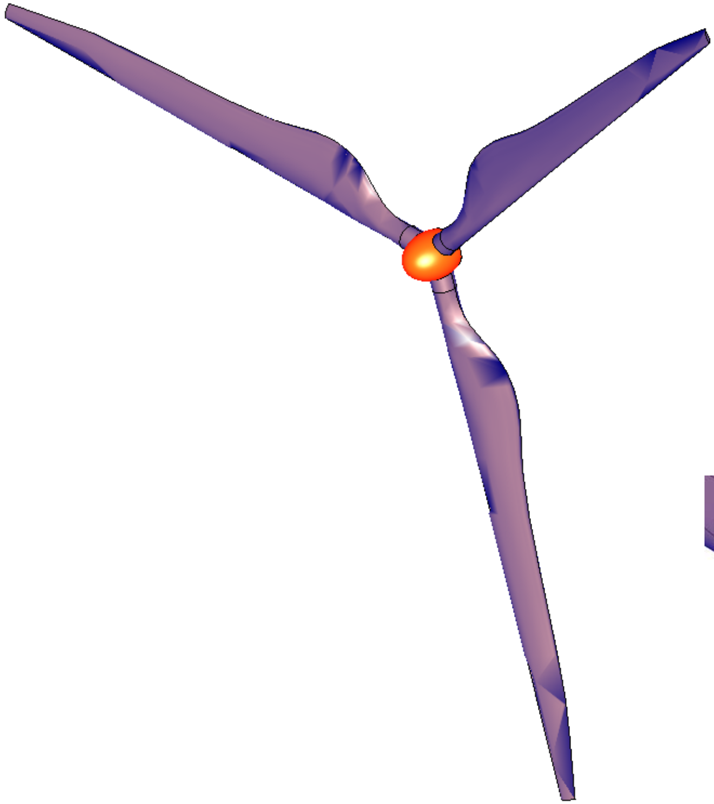


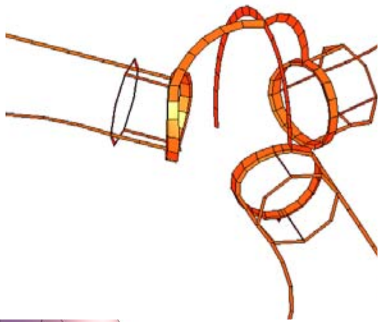
wing

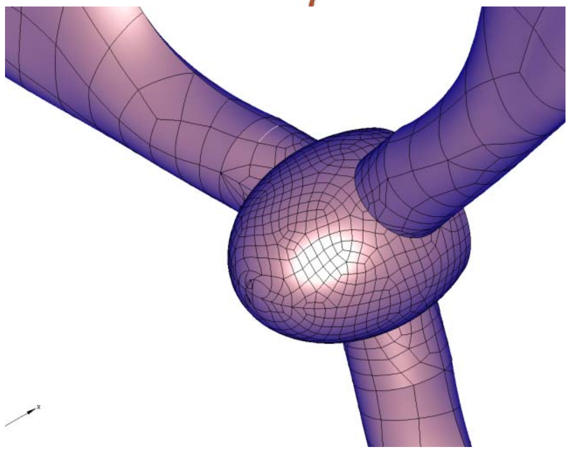


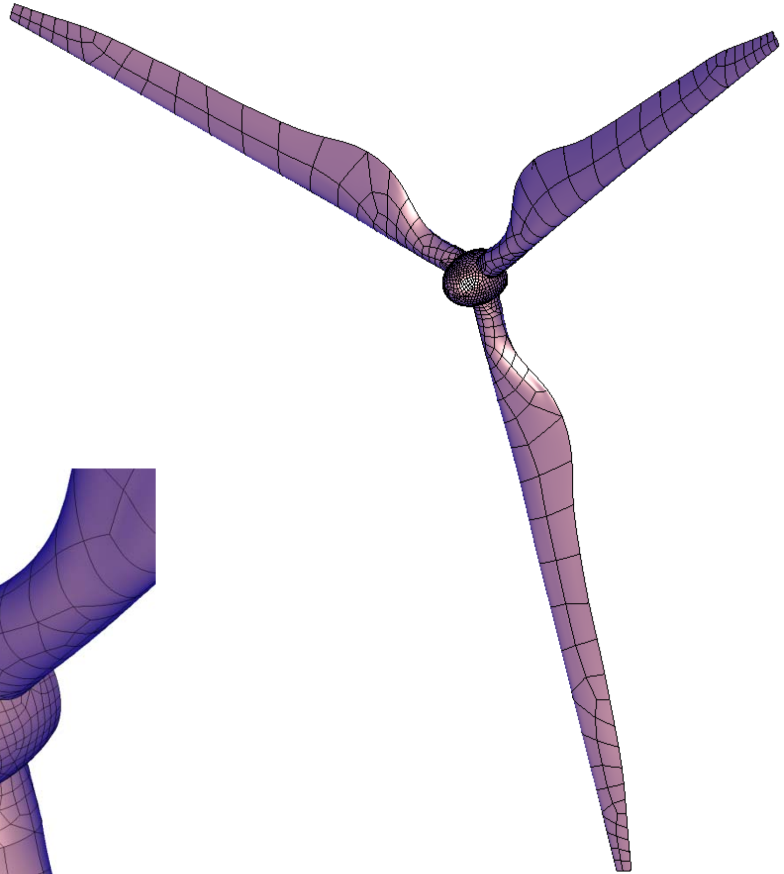
violin

# a) better mesh generation: examples



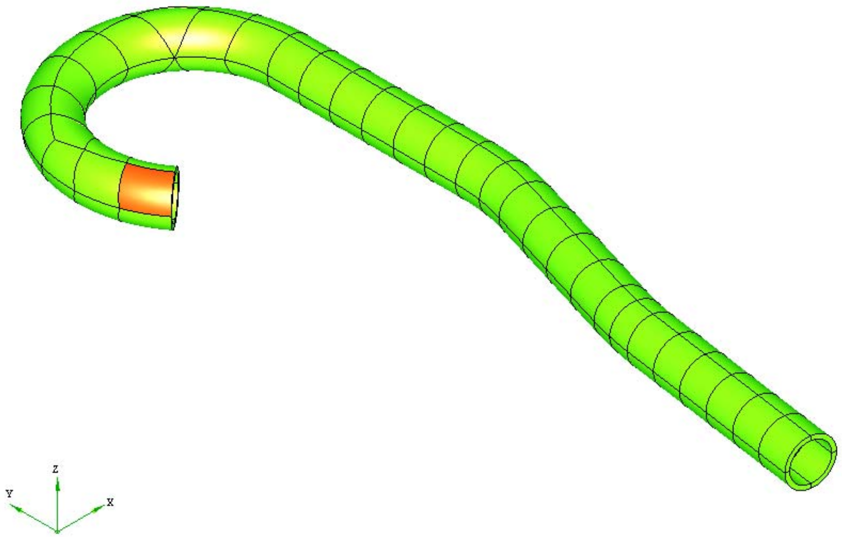


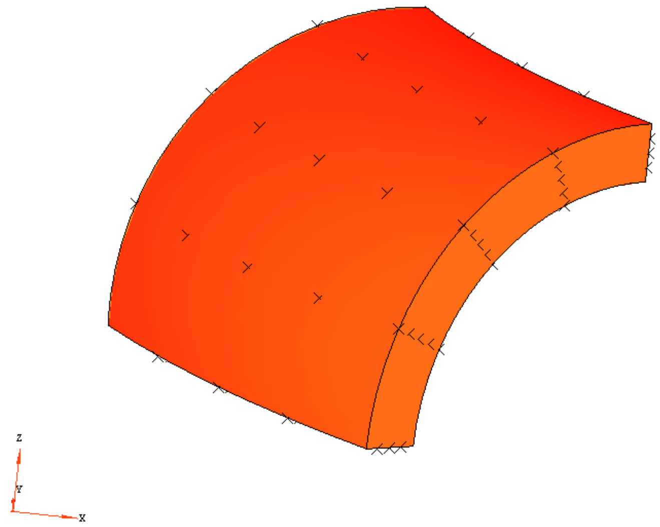




high order mesh of a wind turbine

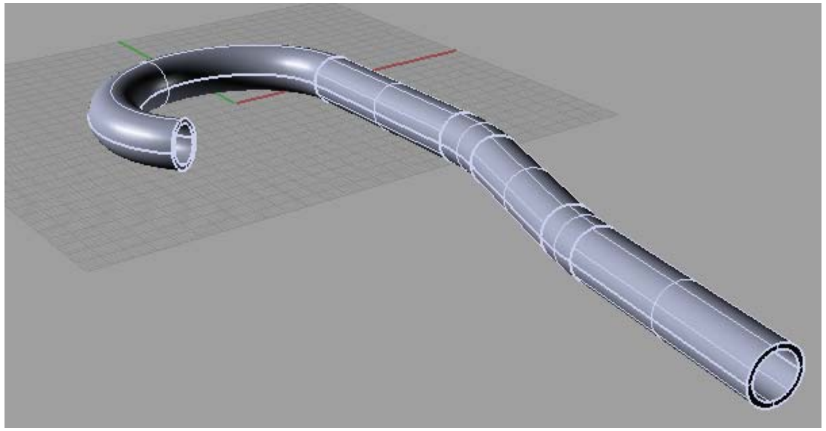
# a) better mesh generation: examples



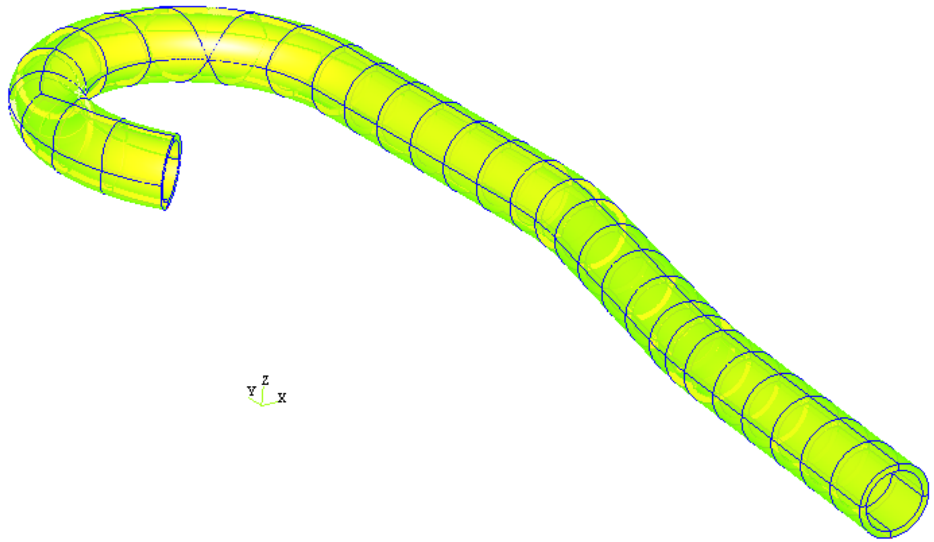


hexahedral mesh of an aorta for “classical” high order FEM

# b) ?



but what if we don’t want to take this (painful) step?



# b) Isogeometric Analysis

Take the geometry from CAD and directly compute on it

but one needs a conforming, hexahedral decomposition

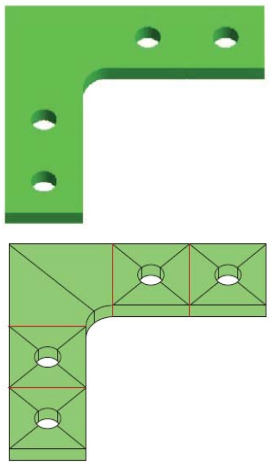


Figure 2.29 The bracket on the top is exactly and concisely represented by five simple NURBS patch (patch boundaries are shown in red, element boundaries in black). The patches match geometrically ar parametrically on the internal faces where they meet.

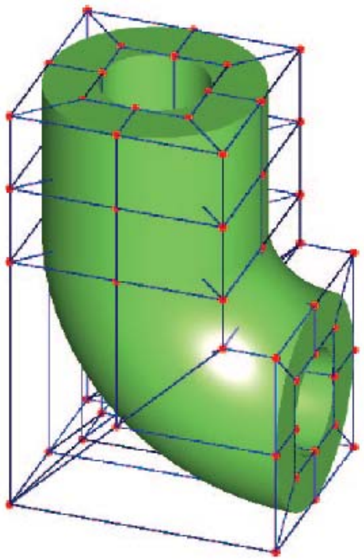
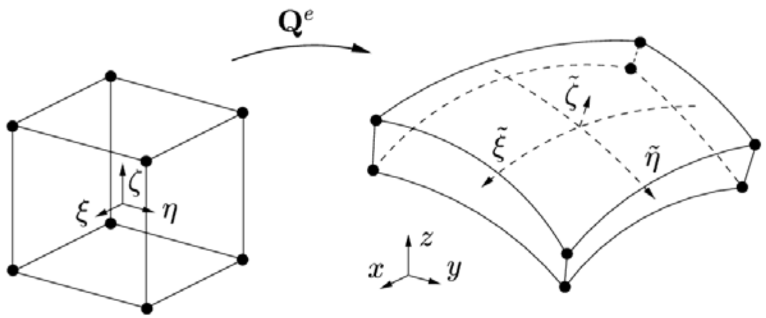


**Figure 2.44** The control lattice for the pipe.

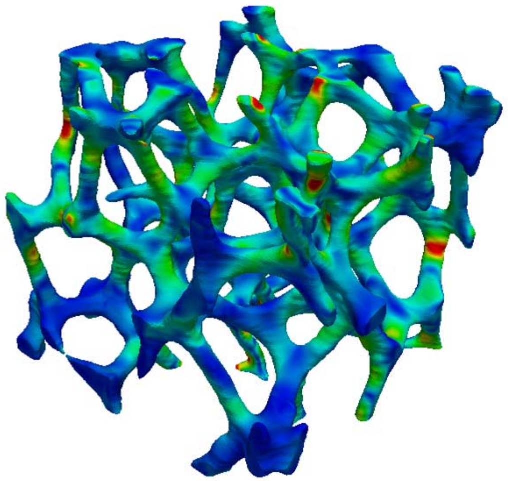



the difference to before?

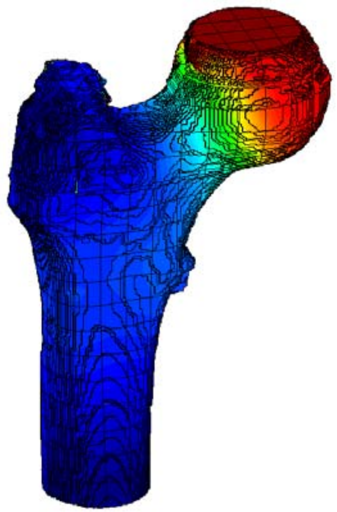
->in order to have to generate a mesh one draws the (coarsest) mesh and refines it

originally published in/by:
J. Austin Cottrell, Thomas J.R. Hughes, Y. Bazilevs. Isogeometric Analysis. Wiley, 2008

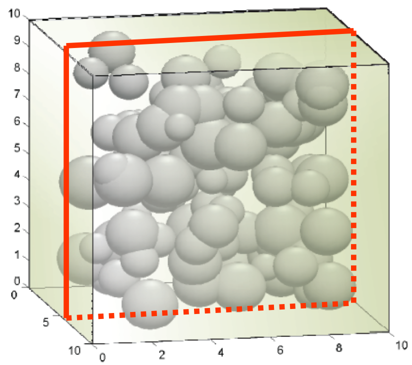
# c) FCM: avoid mesh generation, generate grids instead

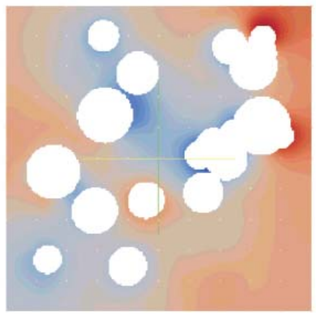


foam



bone



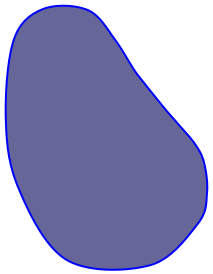


porous media

# Finite Cell Method

(Parvizian, Düster, Rank 2007, Düster, Parvizian, Yang, Rank 2008)

A **fictitious domain method** with **high order polynomial basis functions**

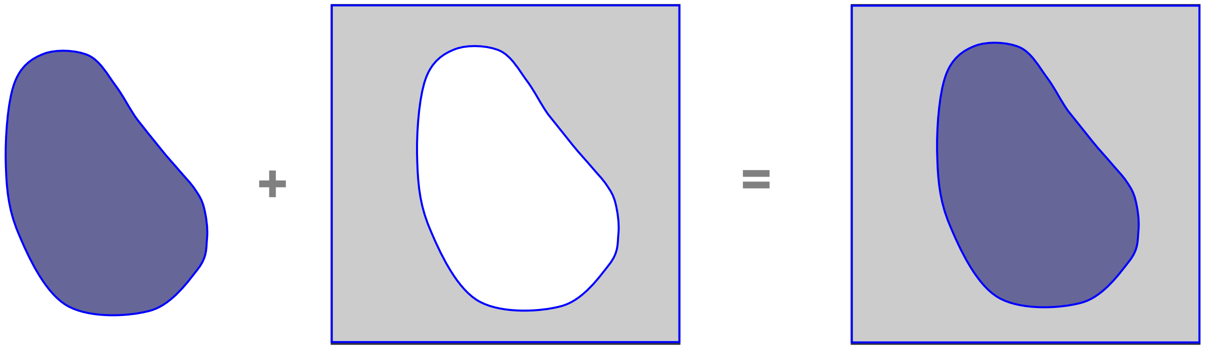


$$\mathcal{B}(\mathbf{u},\mathbf{v}) = \int_{\Omega} [\mathbf{L}\mathbf{v}]^T \mathbf{C}\,[\mathbf{L}\mathbf{u}]\,dxdy$$

# Finite Cell Method

(Parvizian, Düster, Rank 2007, Düster, Parvizian, Yang, Rank 2008)

A **fictitious domain method** with **high order polynomial basis functions**

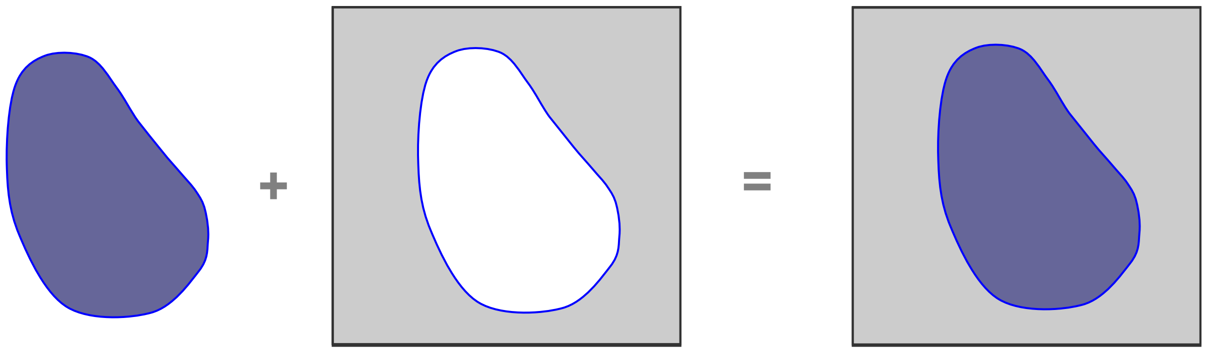


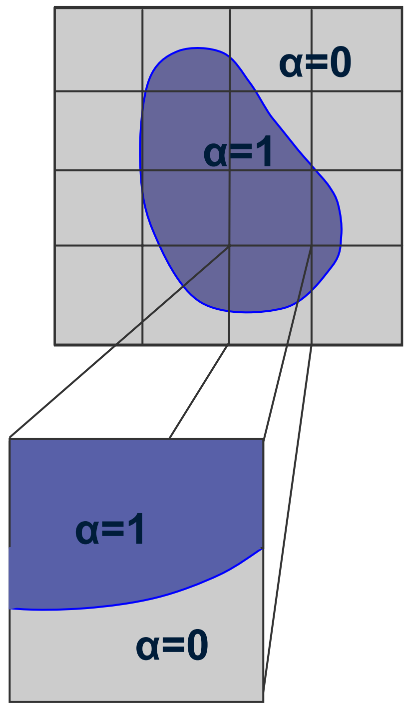
$$\mathcal{B}(\mathbf{u},\mathbf{v}) \;=\; \int_{\Omega} [\mathbf{L}\mathbf{v}]^T \mathbf{C}\,[\mathbf{L}\mathbf{u}]\,dxdy + \int_{\Omega_e-\Omega} [\mathbf{L}\mathbf{v}]^T\,\mathbf{0}\,[\mathbf{L}\mathbf{u}]\,dxdy$$

# Finite Cell Method

(Parvizian, Düster,  Rank 2007, Düster, Parvizian, Yang, Rank 2008)

A **fictitious domain method** with **high order polynomial basis functions**






$$\mathcal{B}(\mathbf{u},\mathbf{v}) = \int_{\Omega} [\mathbf{Lv}]^T \mathbf{C}\,[\mathbf{Lu}]\,dxdy + \int_{\Omega_e - \Omega} [\mathbf{Lv}]^T\,\mathbf{0}\,[\mathbf{Lu}]\,dxdy$$

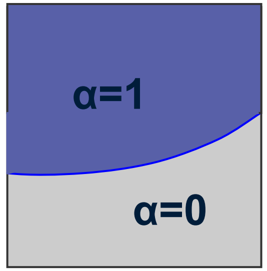
$$= \int_{\Omega_e} [\mathbf{Lv}]^T\,\alpha\,\mathbf{C}\,[\mathbf{Lu}]dxdy$$

# Finite Cell Method: determination of α

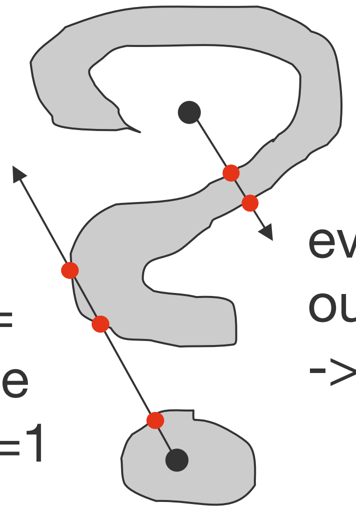
$$\mathcal{B}(\mathbf{u},\mathbf{v}) = \int_{\Omega} [\mathbf{Lv}]^T \mathbf{C}\,[\mathbf{Lu}]\,dxdy + \int_{\Omega_e-\Omega} [\mathbf{Lv}]^T\,\mathbf{0}\,[\mathbf{Lu}]\,dxdy$$

$$= \int_{\Omega_e} [\mathbf{Lv}]^T\,\alpha\,\mathbf{C}\,[\mathbf{Lu}]\,dxdy$$

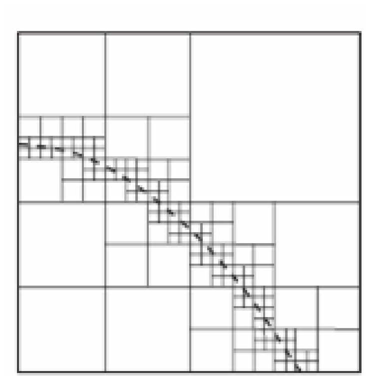
→ discontinous „indicator function“



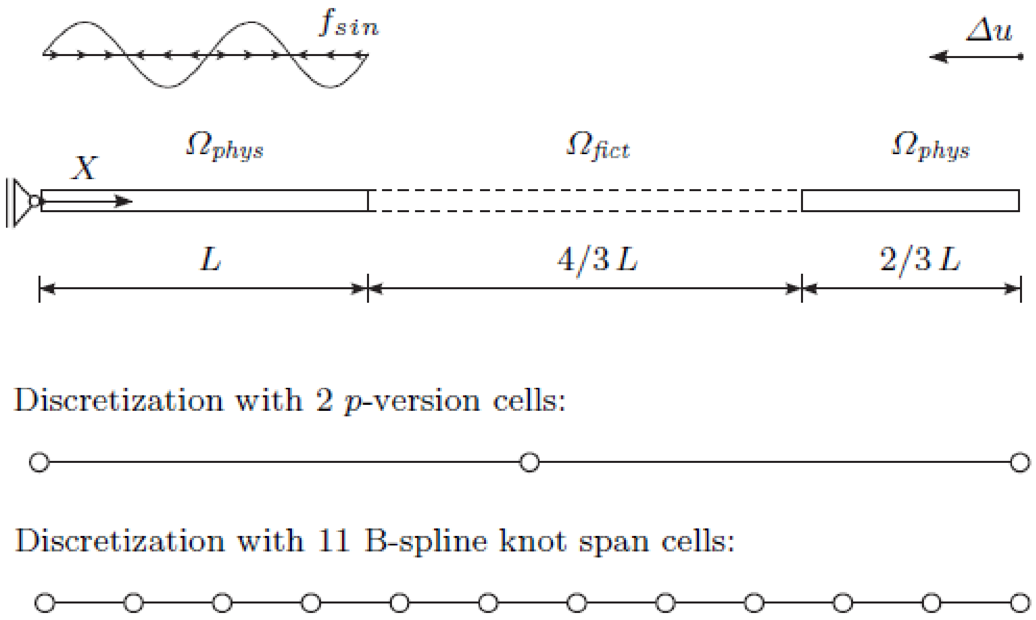

1. Inside or outside?




2. adaptive integration



# FCM: Why does it work?

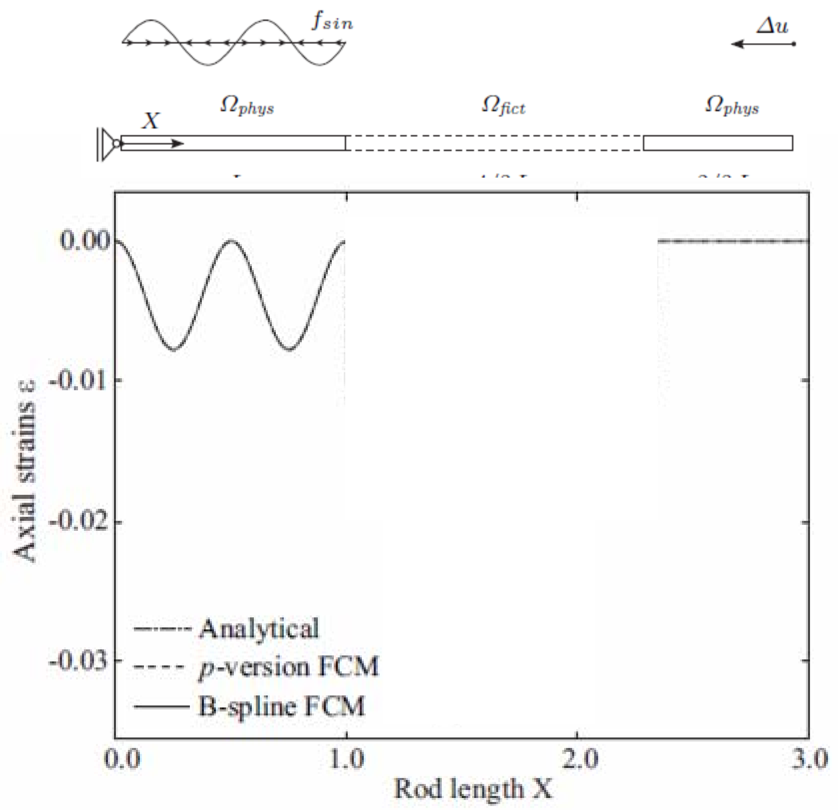


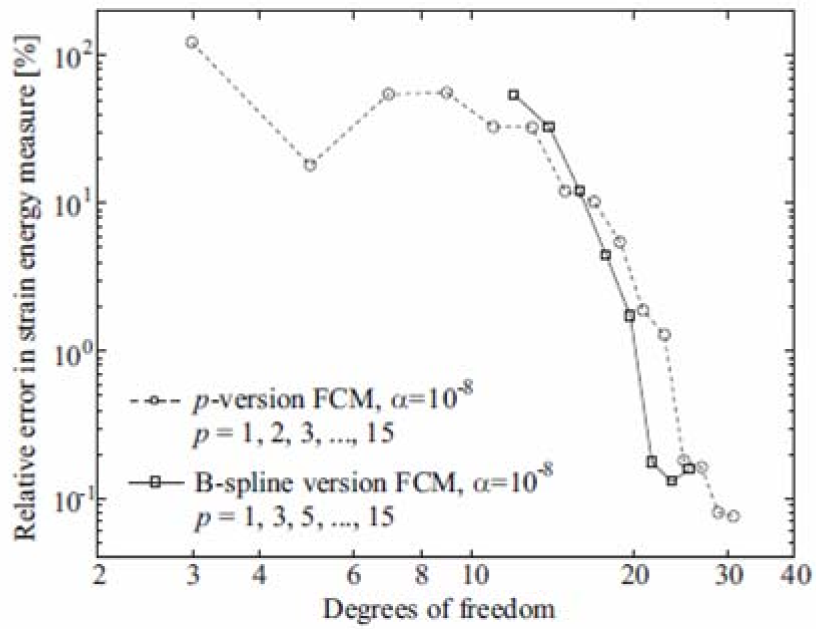

Parameters:

Young's modulus $E$=1.0
Poisson's ratio $\nu = 0.0$
Penalization parameter $\alpha = 10^{-q}$
Area $A$=1.0
Length of each part $L$=1.0
Displacement load $\Delta u$=0.02
Sine load $f_{sin} = 1/20 \sin(4\pi X)$

D. Schillinger, A. Düster, E. Rank: hpd adaptive Finite Cell Method for Geometrically Nonlinear Problems of solid mechanics, submitted to: *IJNME.*

# FCM: Why does it work?



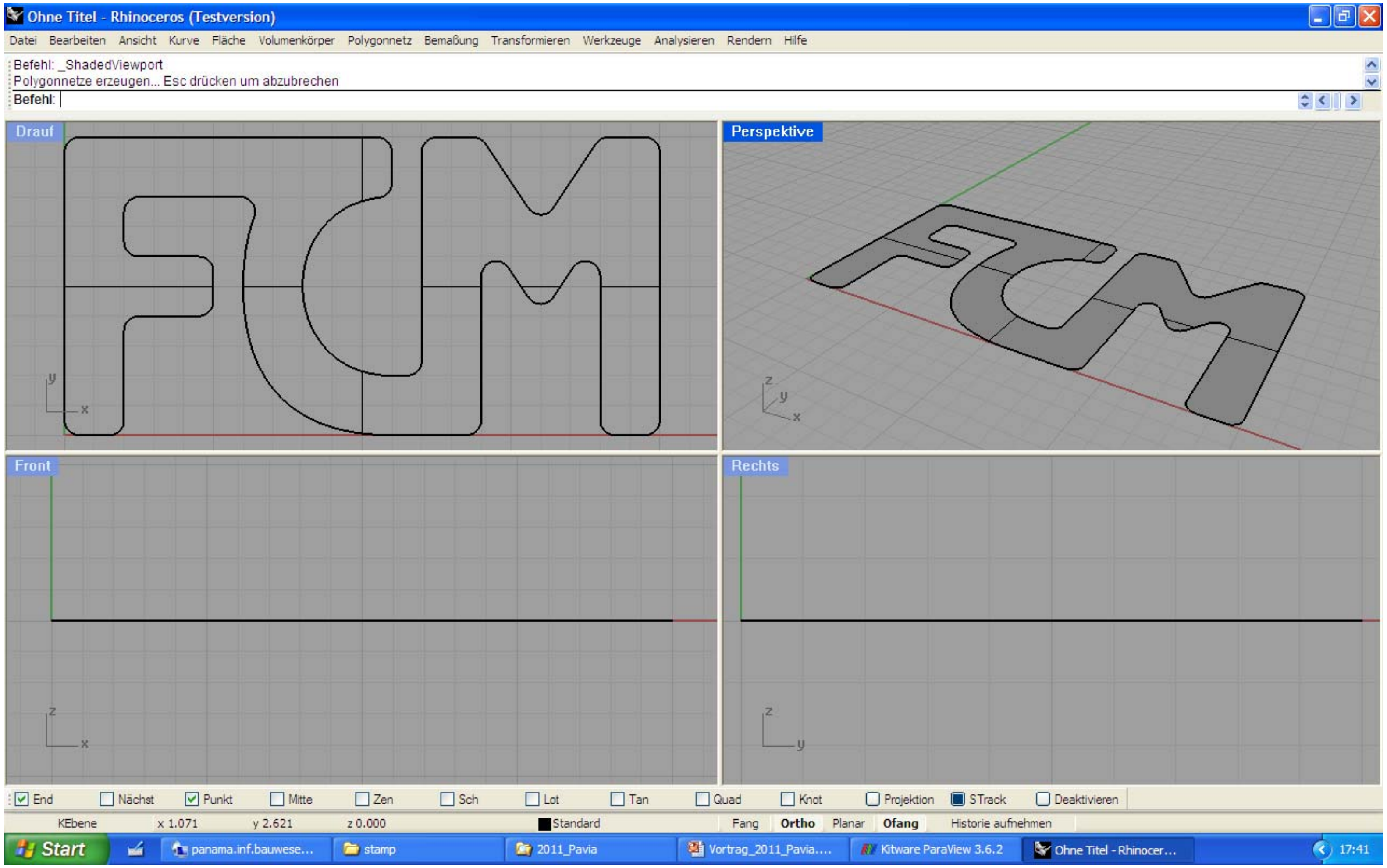

- Smooth extension of solution fields

- Best approximation property to strain energy + penalization of fictitious domain

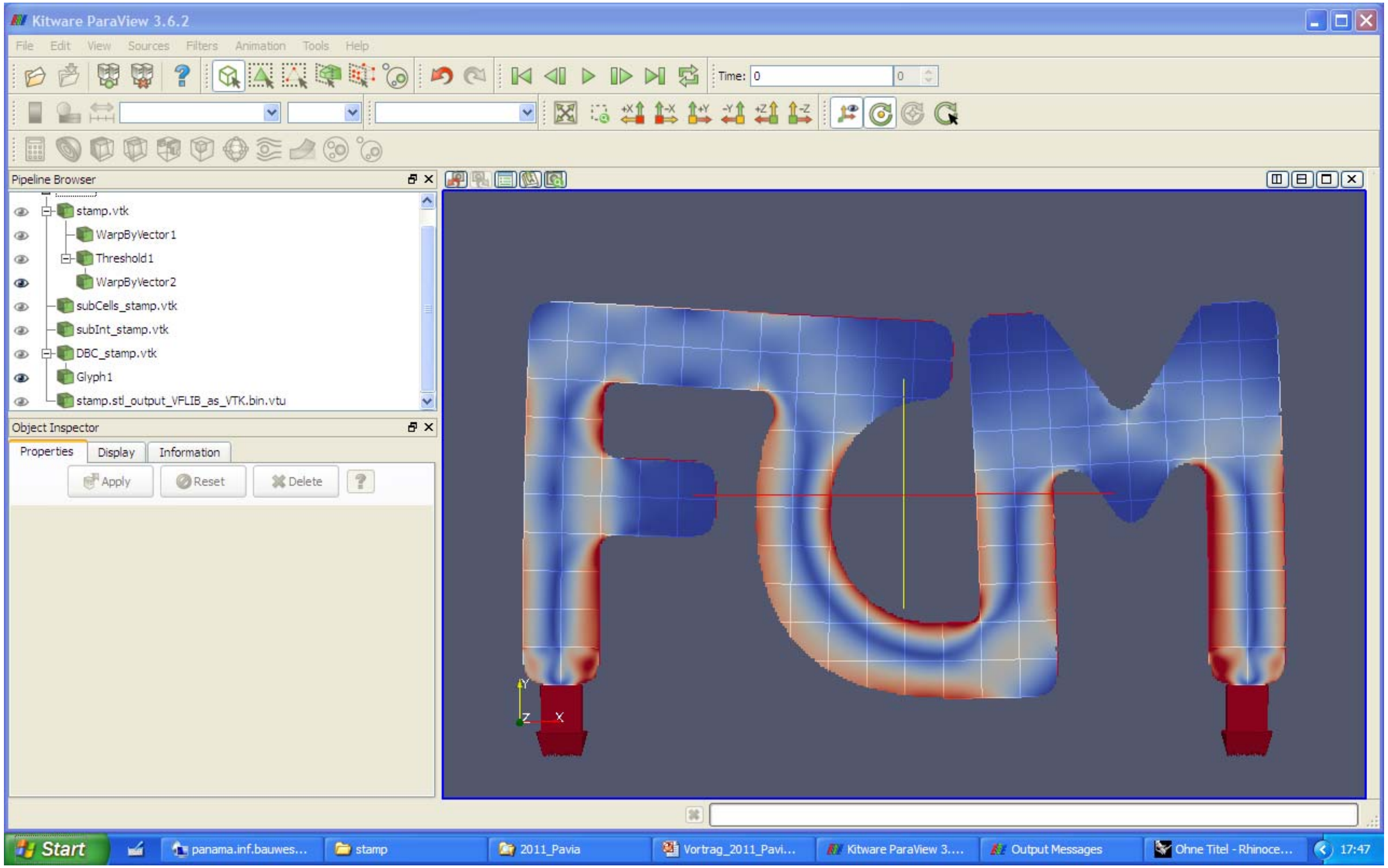
$$U = \int_{\Omega} \Psi \, dV = \frac{1}{2} \int_{\Omega} \boldsymbol{\sigma} : \varepsilon \, dV$$

$$e_r = \sqrt{\frac{|U_{ex} - U_{FCM}|}{U_{ex}}} \times 100\%$$

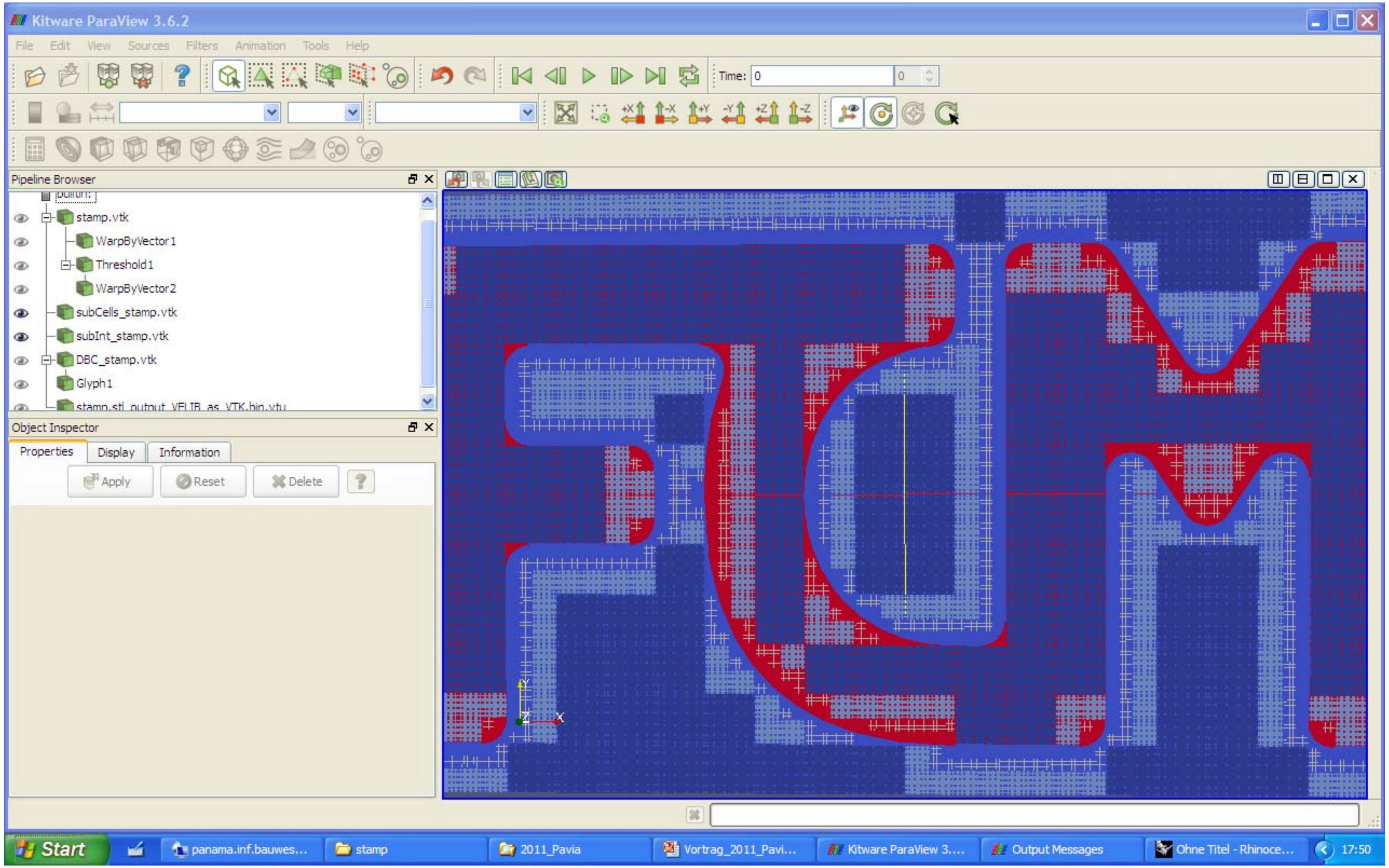
# Finite Cell Method: how does it work…



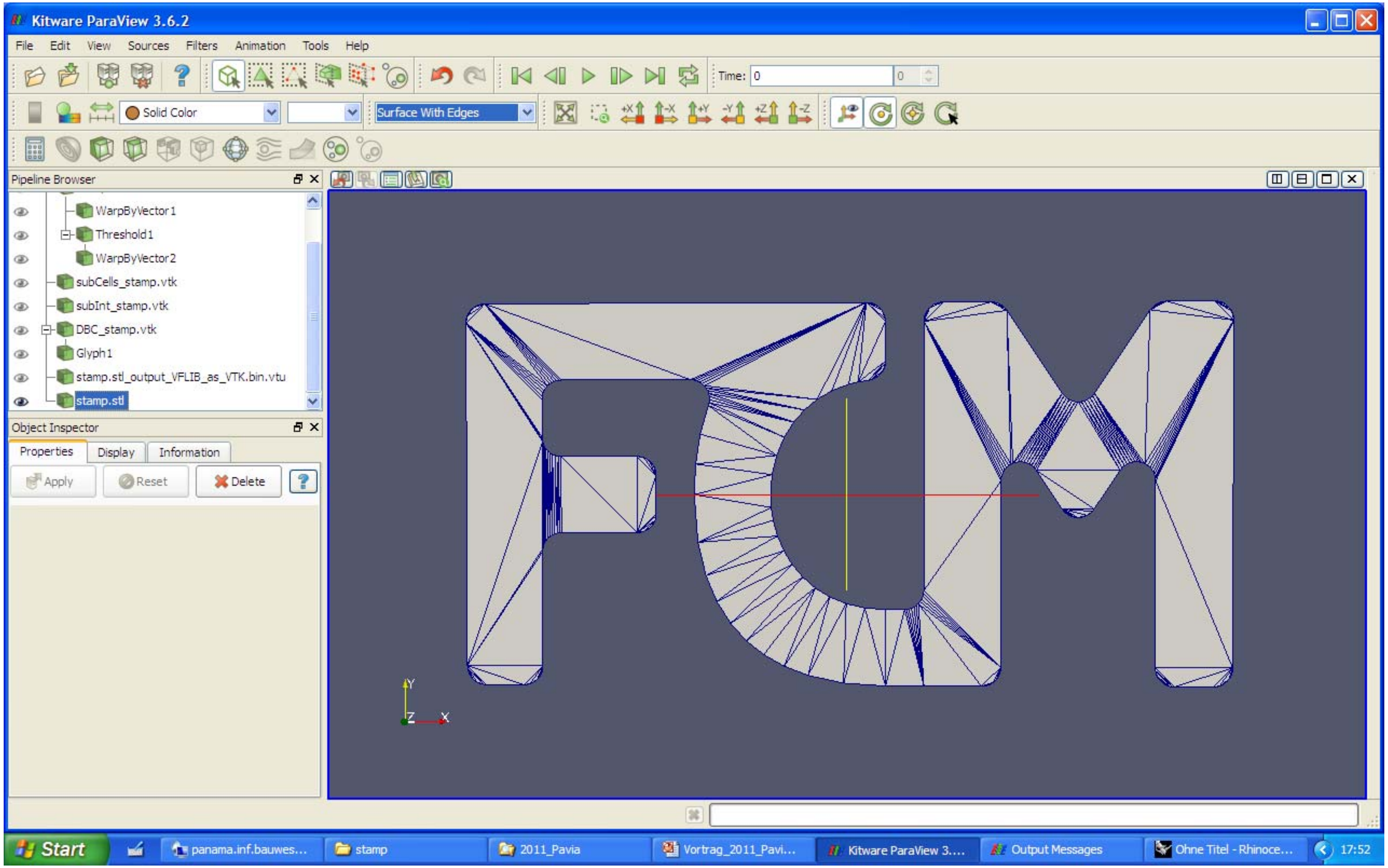
# Finite Cell Method: how does it work…



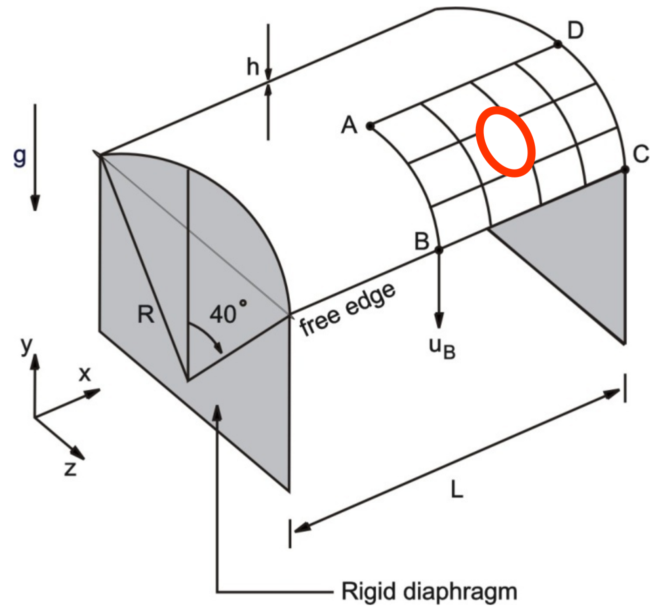
# Finite Cell Method: how does it work…

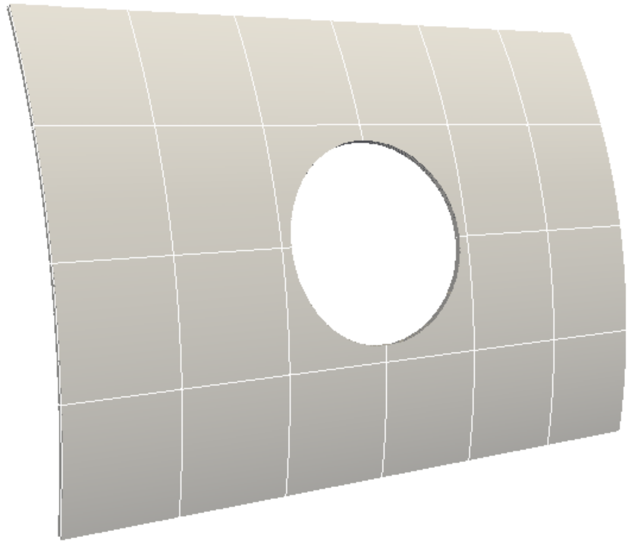


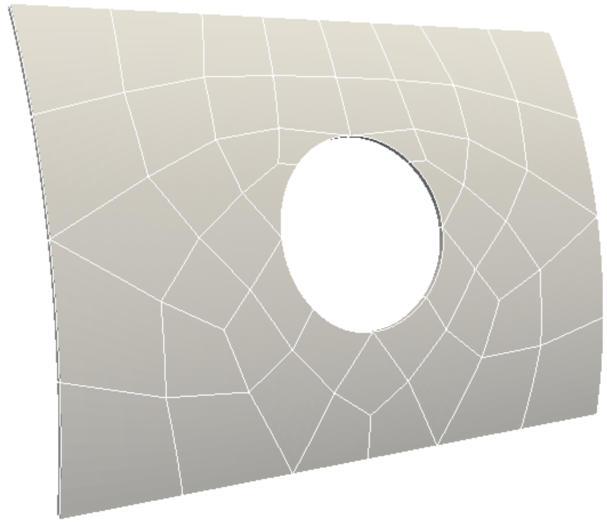
# Finite Cell Method: how does it work…



# Finite Cell Method: thick solid shells





## Geometry:

$R = 200$ mm, $h = 2$ mm, $L = 400$mm

## Boundary conditions:

$u_y = u_z = 0$

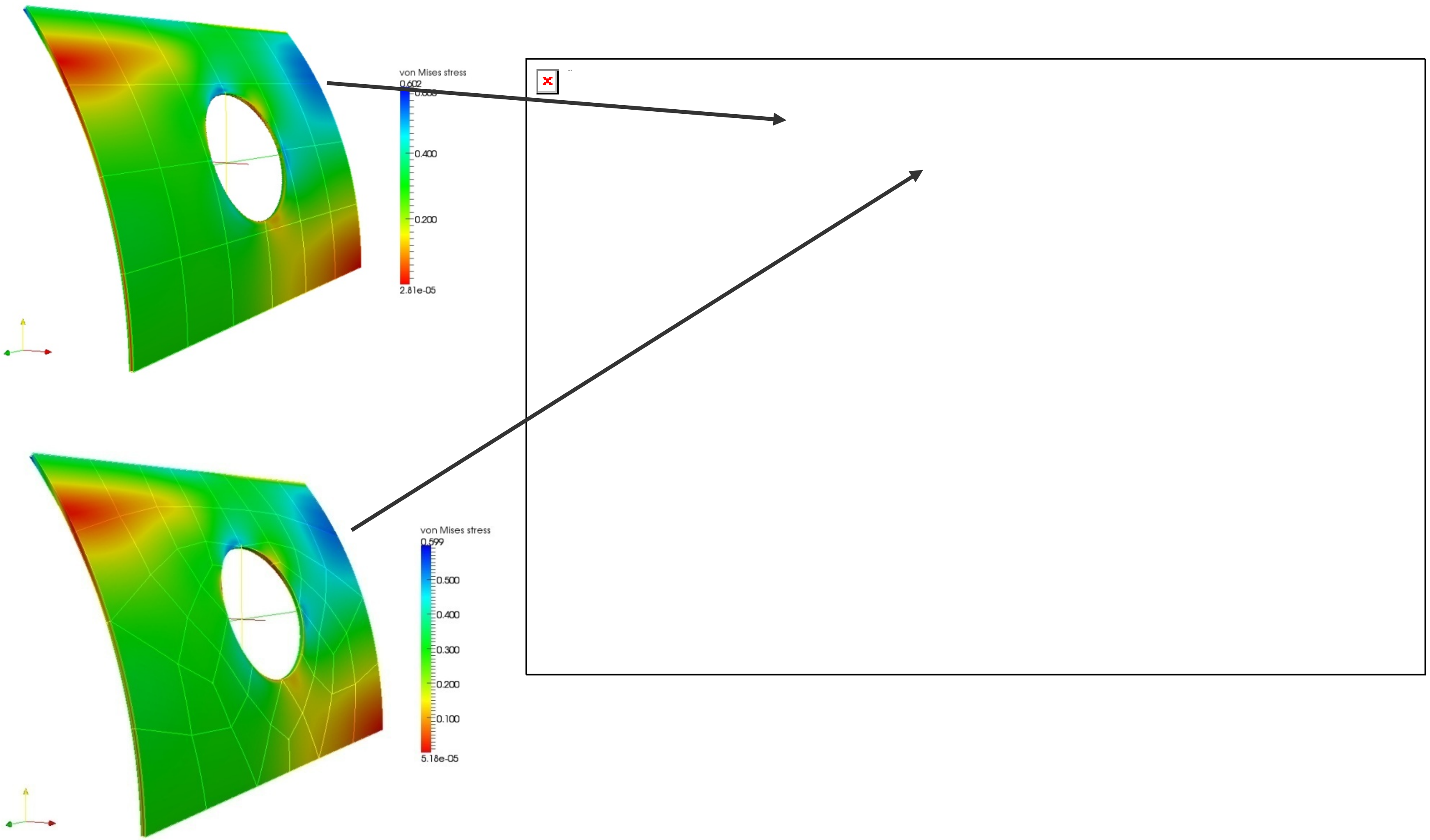
## vertical shell weight:

$\rho = 7850$ kg/m$^3$, $g = 10$ m/s$^2$

## Material:

$E = 2.069 \times 10^5$ MPa, $\nu = 0$

# Finite Cell Method: thick solid shells



# Finite Cells in Biomechanics: Computational Steering

**Joint project with:**

R. Westermann (TUM-IN)

R. Burgkart (TUM Klinikum rechts der Isar)

A.Düster (TUHH)

J. Parvizian (Univ. Isfahan)

Z. Yosibash (Univ. Beer Sheva)

**Scientific staff:**

Ch. Dick, S. Kollmannsberger, M. Ruess, Z. Yang

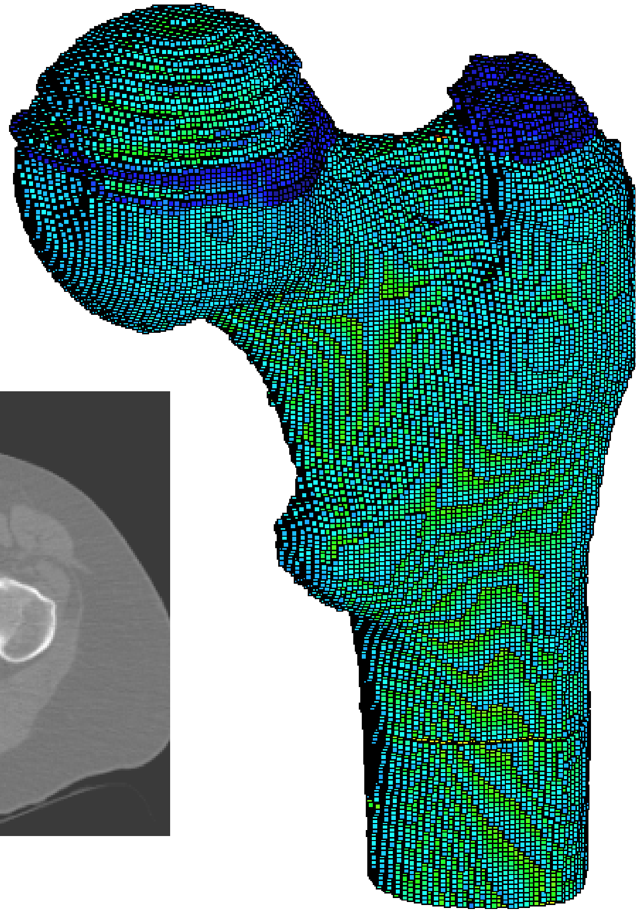
**Funding:**

IGSSE, TUM-IAS, Humboldt-Foundation, SIEMENS

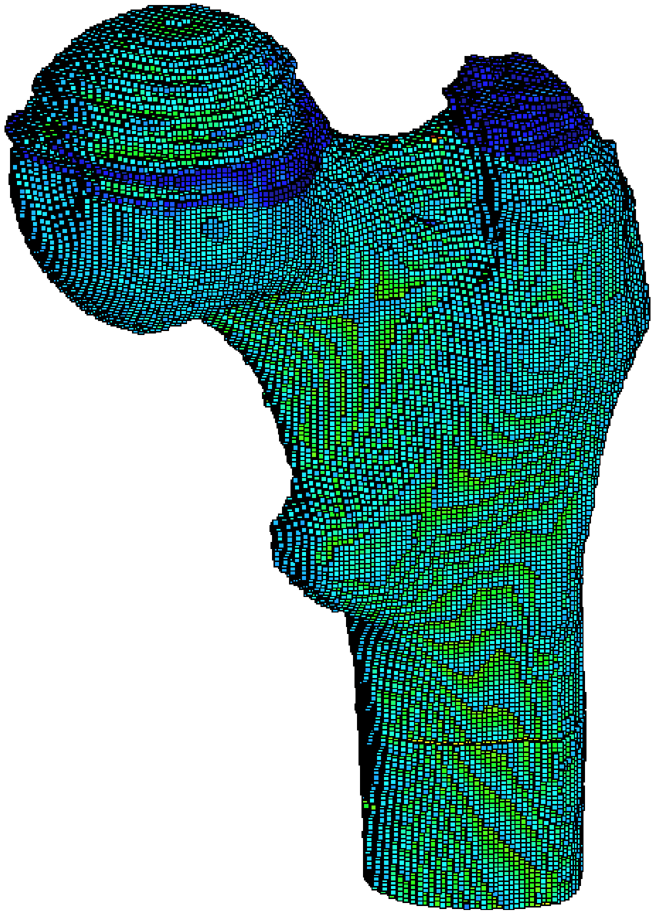
# Finite Cells in Biomechanics

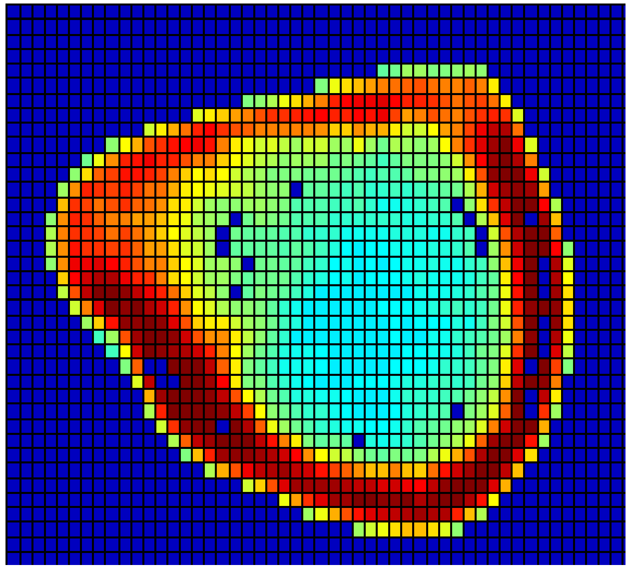
CT scans

CT data (Hounsfield Unit)



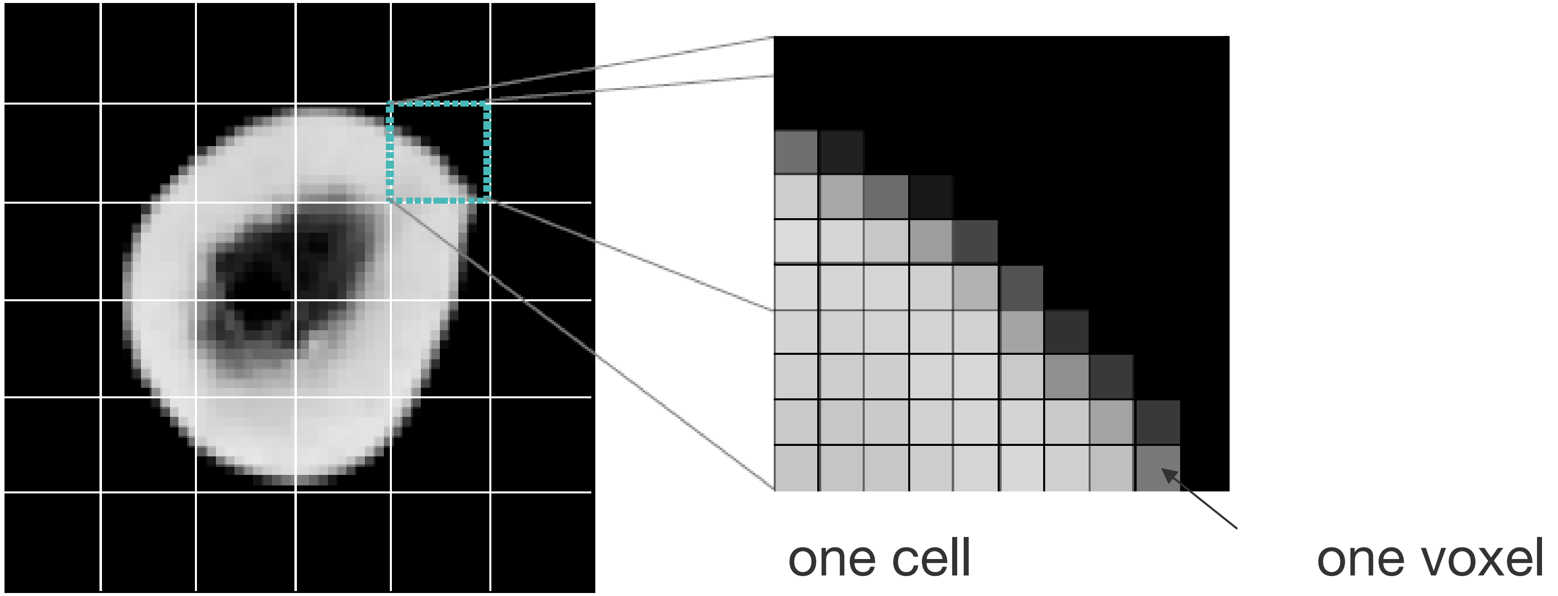
# Finite Cells in Biomechanics





Resolution of CT scan:
$\Delta x = \Delta y = 0.78125$ mm, $\Delta z=0.75$ mm

# Finite Cells in Biomechanics




-> A “cell” is a “finite element”

-> A finite element consists of n voxels

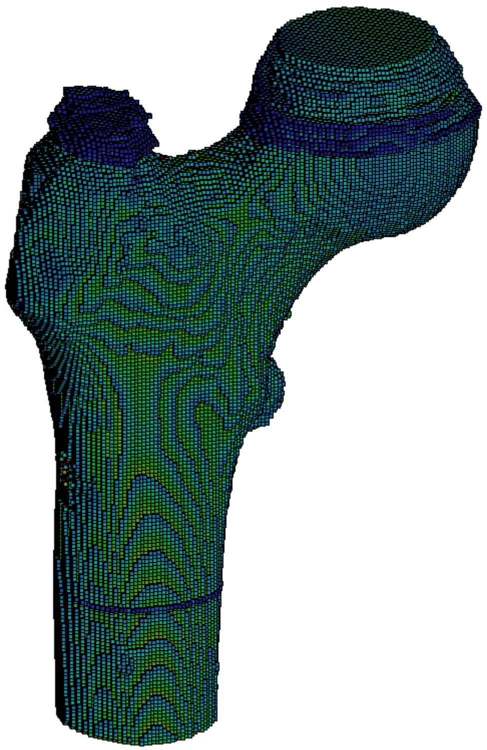
precompute

$$\hat{K}_c = \sum_{i=1}^{n_z} \sum_{j=1}^{n_y} \sum_{k=1}^{n_x} (\lambda_{ijk} \mathbf{K}_{ijk}^{\lambda} + \mu_{ijk} \mathbf{K}_{ijk}^{\mu})$$

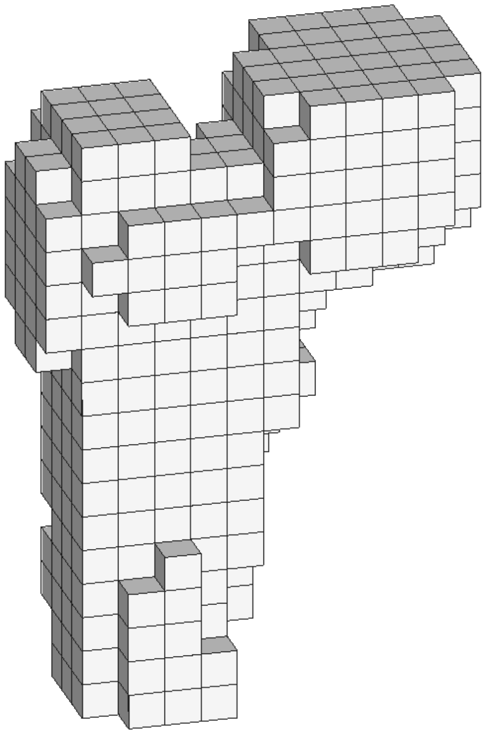
# Finite Cells in Biomechanics

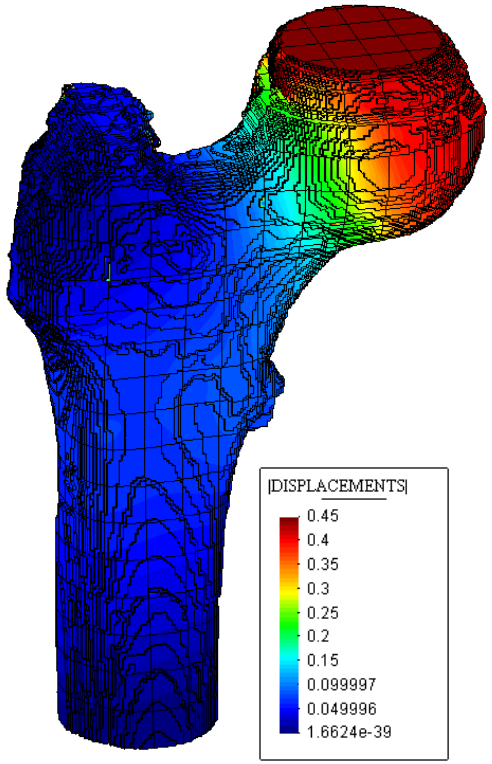
Yosibash et al. 2007

CT data

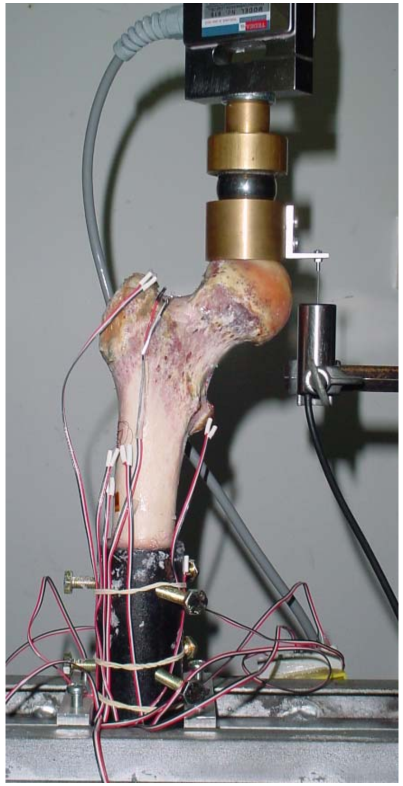


FCM



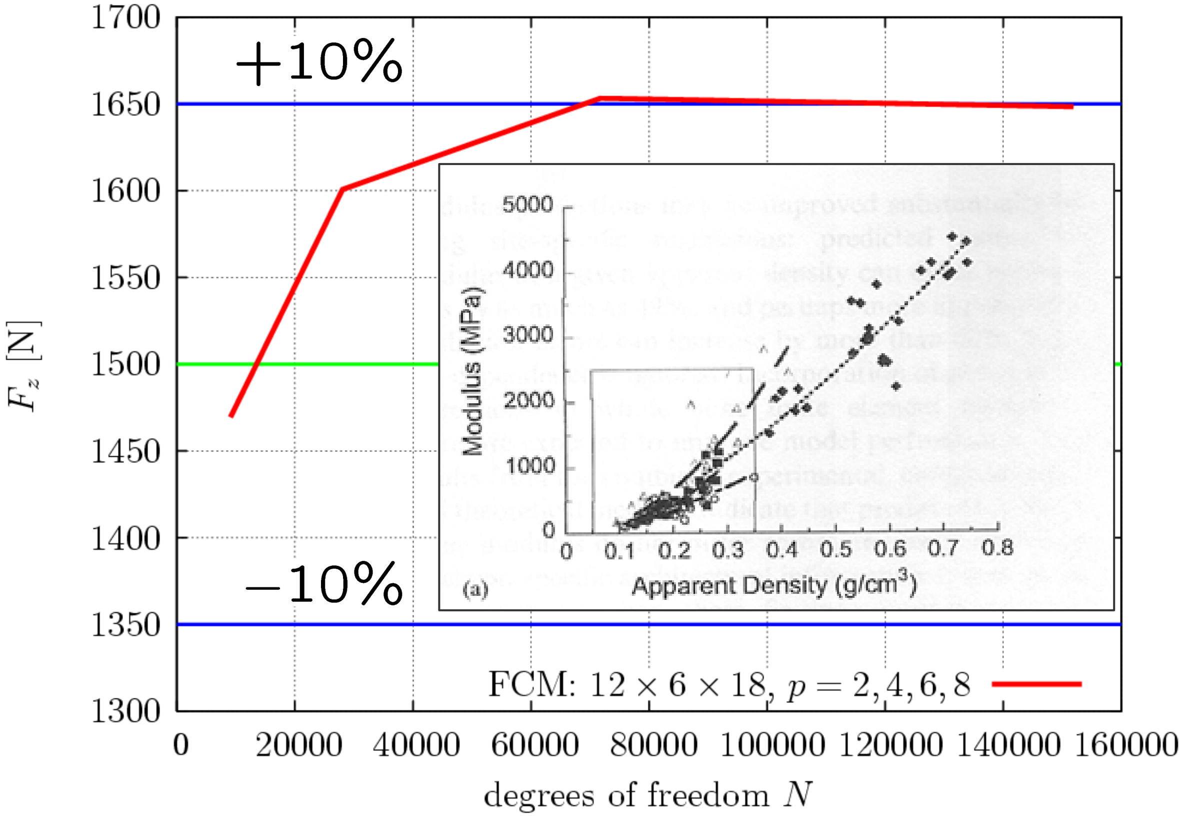



**F**z =1500N,
uz = 0.45mm

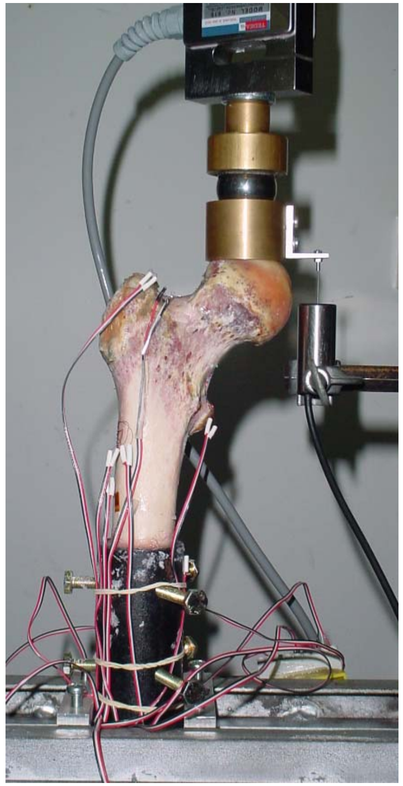


# Finite Cells in Biomechanics

Convergence of $\mathbf{F}_z$

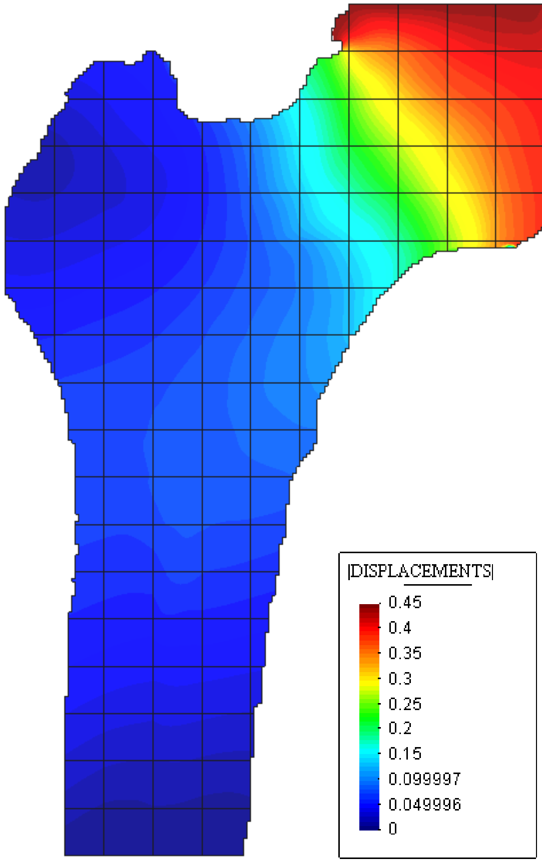



**F**z =1500N,
uz = 0.45mm

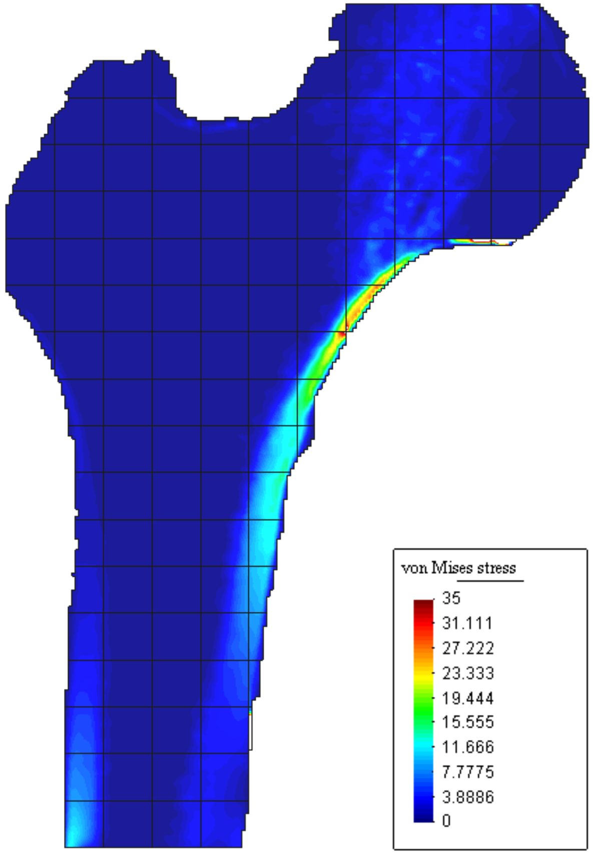


# Finite Cells in Biomechanics

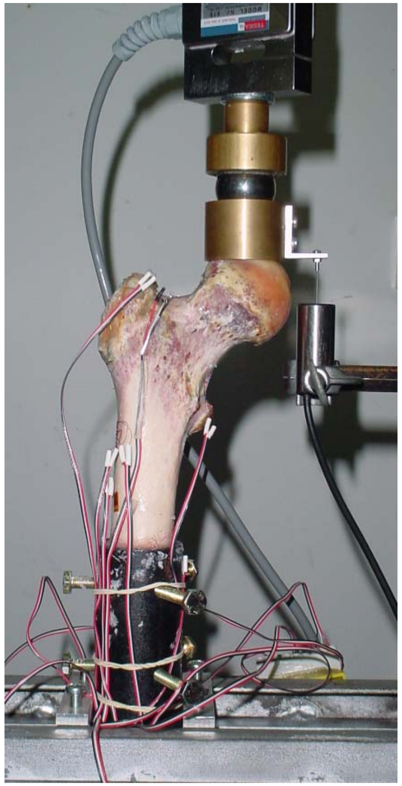
Displacements




von Mises stress




Fz =1500N,
uz = 0.45mm

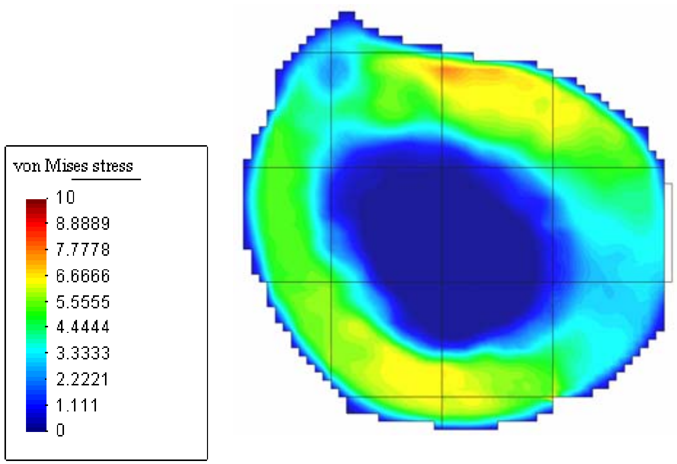


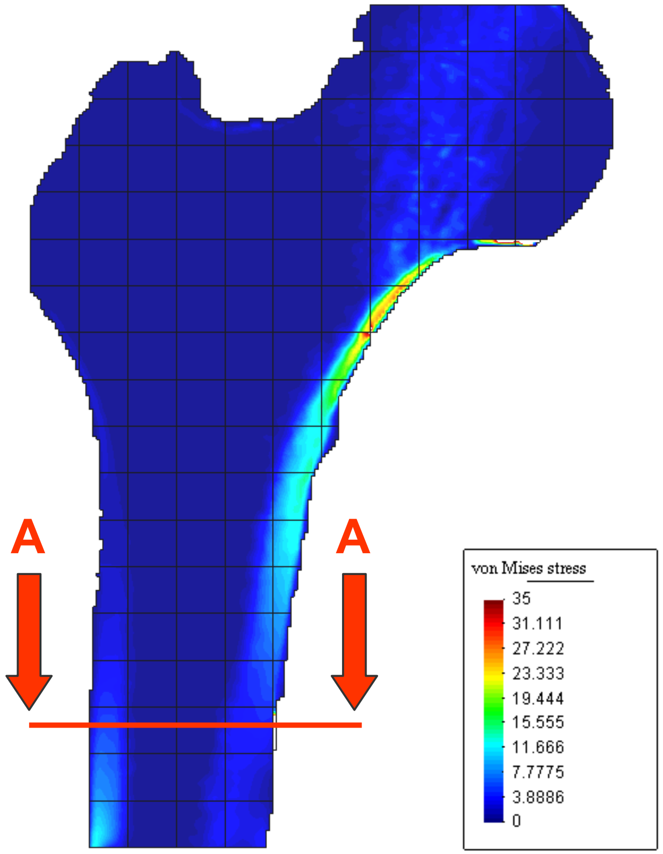
# Finite Cells in Biomechanics

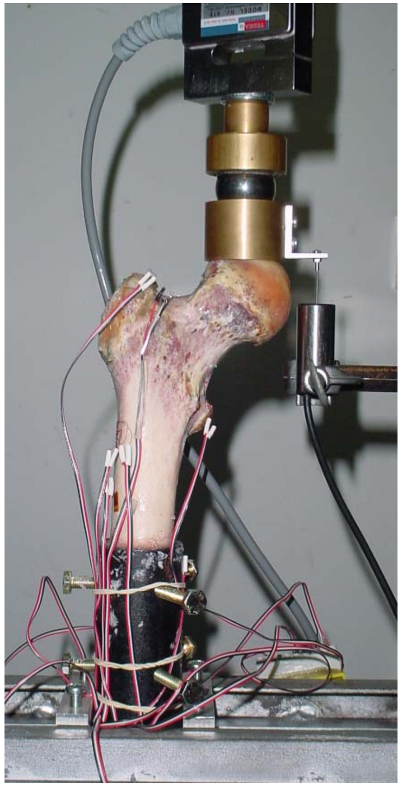
von Mises stress

Fz =1500N,
uz = 0.45mm

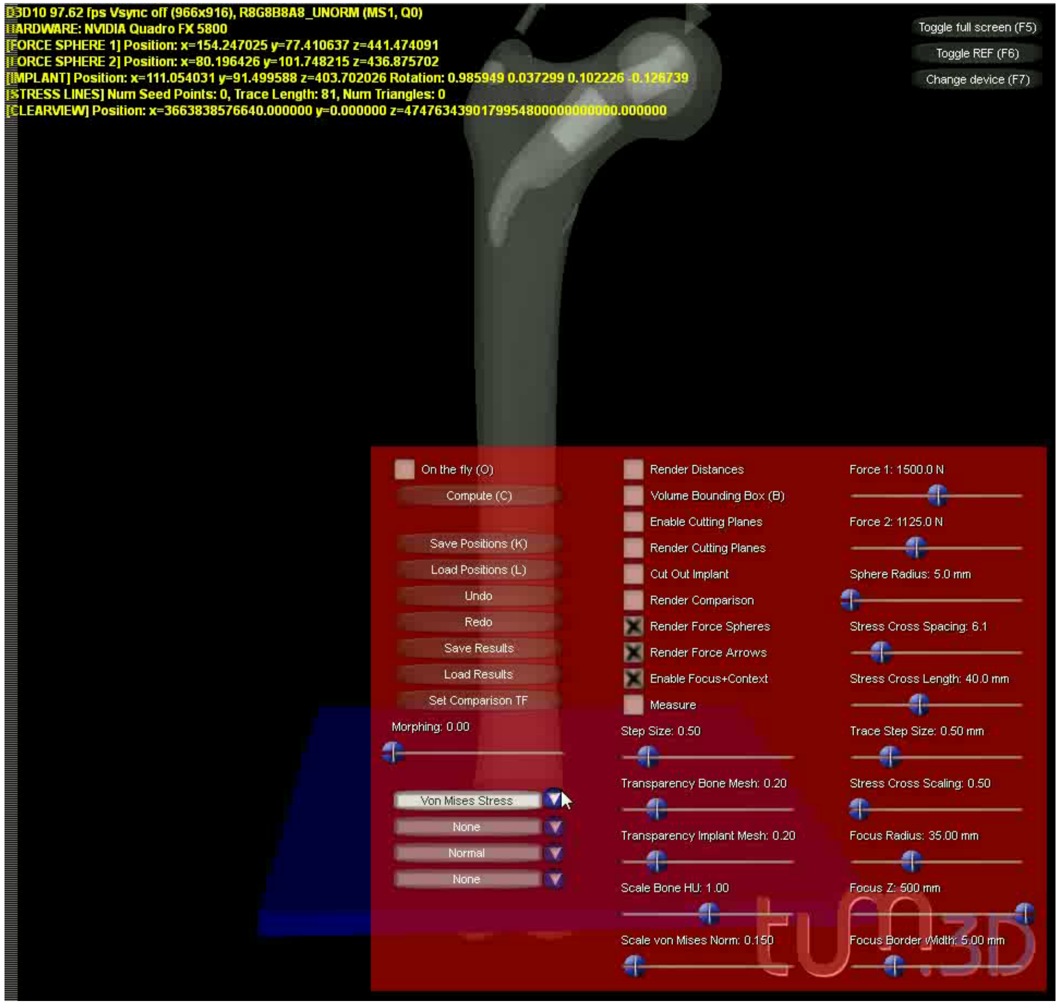
A-A



# Finite Cells in Biomechanics



# back to Isogeometric Analysis

Take the geometry from CAD
and directly compute on it

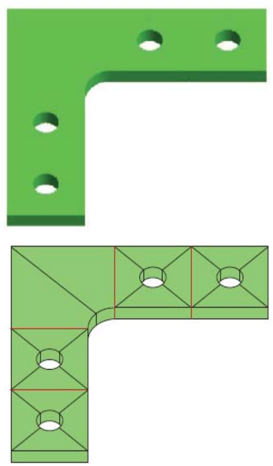


Figure 2.29 The bracket on the top is exactly and concisely represented by five simple NURBS patch (patch boundaries are shown in red, element boundaries in black). The patches match geometrically ar parametrically on the internal faces where they meet.

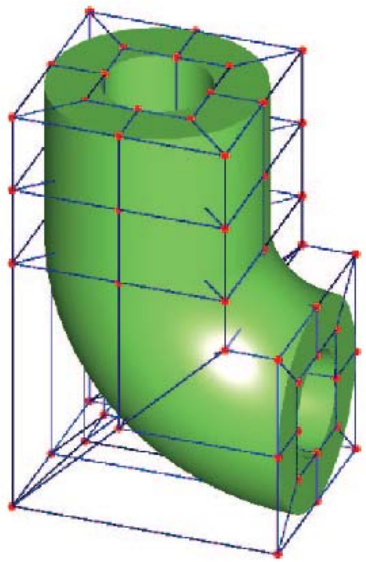
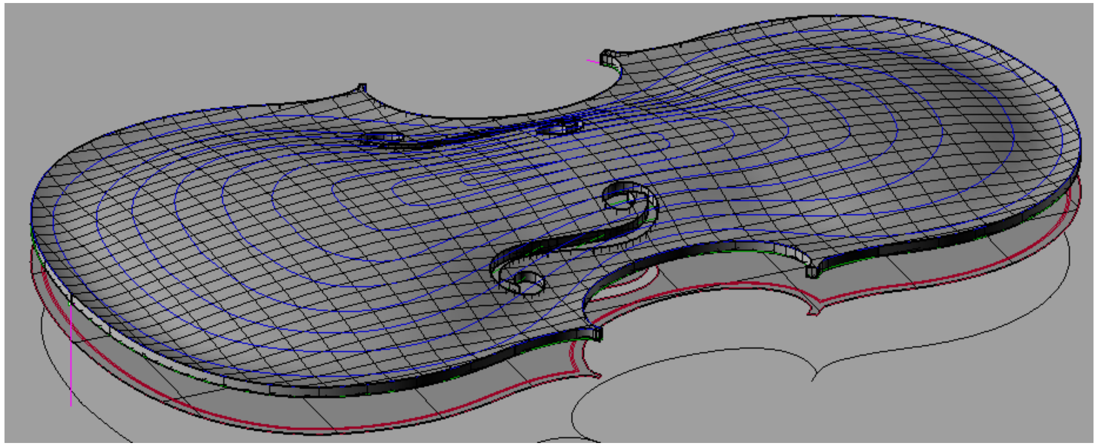


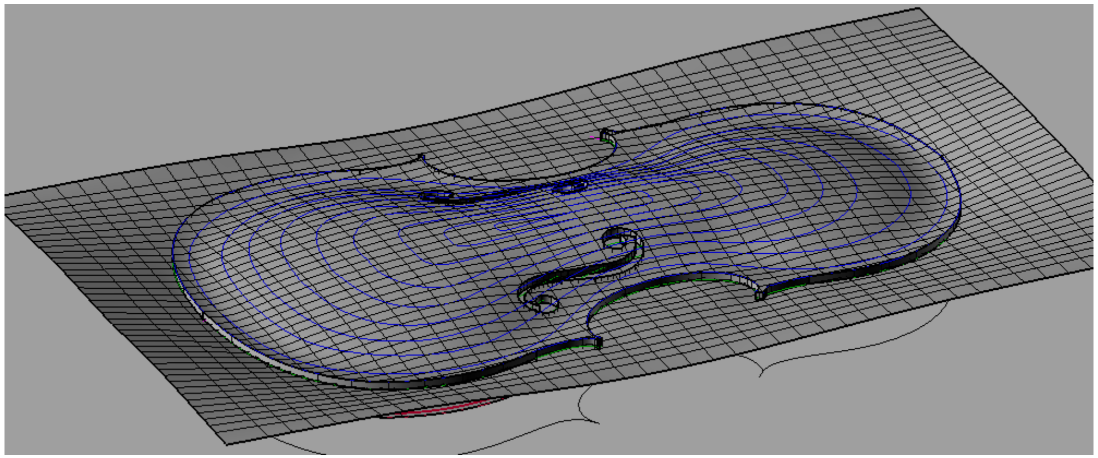
Figure 2.44 The control lattice for the pipe.

originally published in/by:
J. Austin Cottrell, Thomas J.R. Hughes, Y. Bazilevs. Isogeometric Analysis. Wiley, 2008

# reconsider: topology in Isogeometric Analysis

what kind of topology are we provided with?

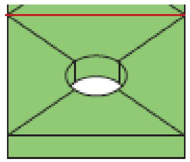




# Remarks on Isogeometric Analysis

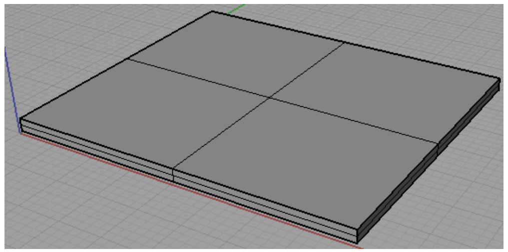
Original idea of IGA: compute with what the CAD modeler provides

A CAD modeler does not think in elements:

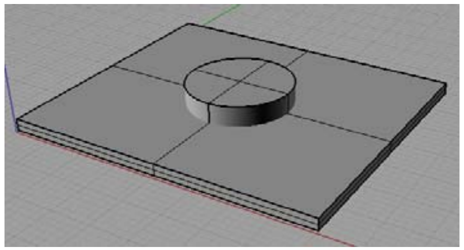


A CAD modeler:

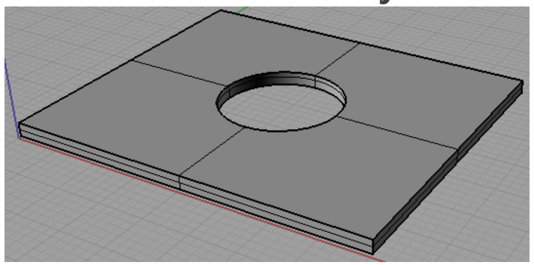
draw volume



draw cylinder



trim volume at cylinder

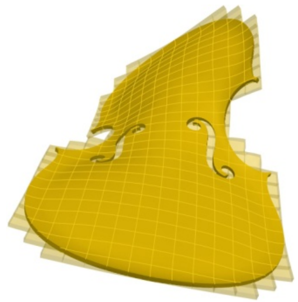


the CAD modeler provides **trimmed** surfaces/volumes

-> We should compute with them!

one possible approach is IGA + FCM

***Isotopological Analysis?***

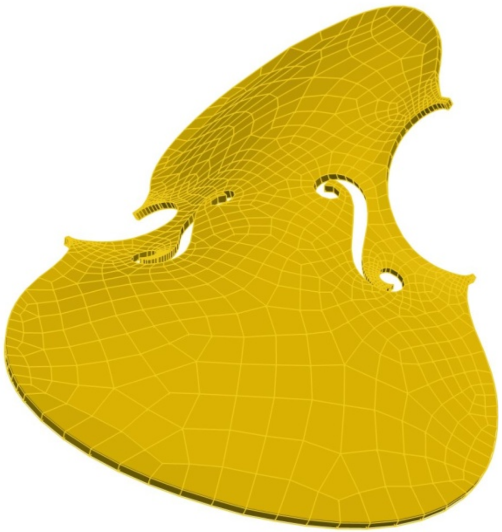


# topology/geometry in Isogeometric Analysis

(Rank, Kollmannsberger, Sorger, Düster, 2011)

NURBS model

a) FEM: generate and compute

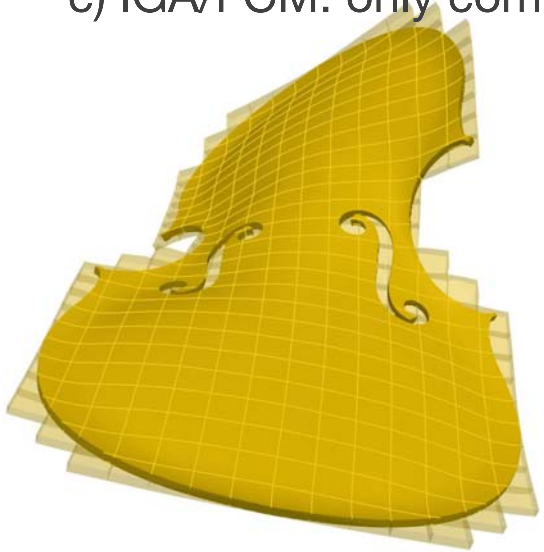


b) IGA:

generate new mesh and compute:

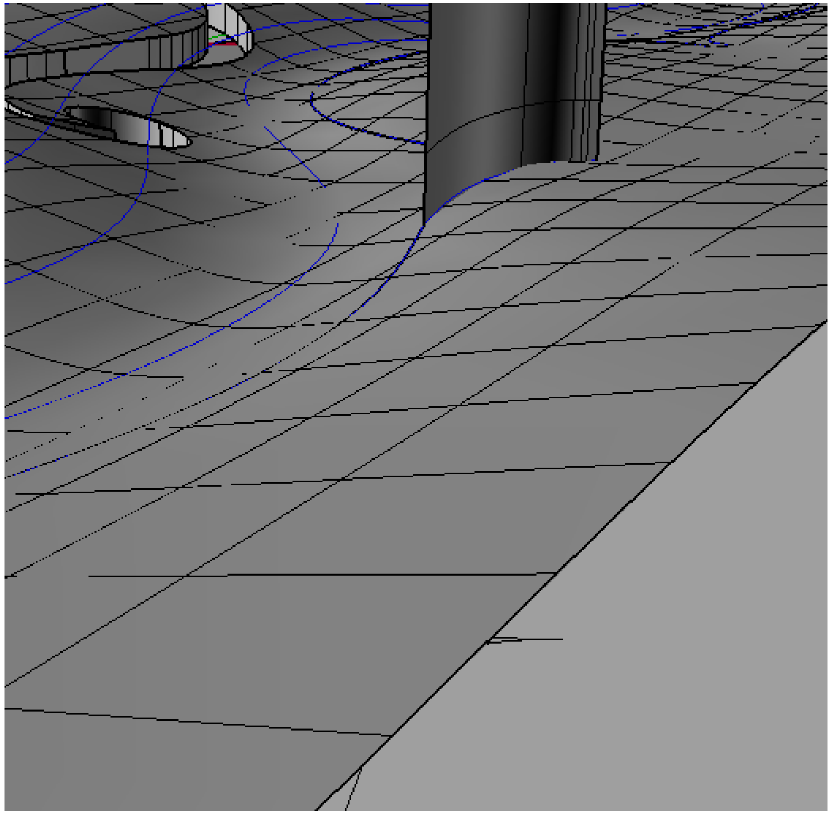
alternatively: convert mesh and compute

c) IGA/FCM: only comput



# Vigoni project proposal

(Kollmannsberger, Reali, Auricchio, Rank 2011)



- compute on trimmed surfaces
- enforce conformity across trimmed patches in a weak sense

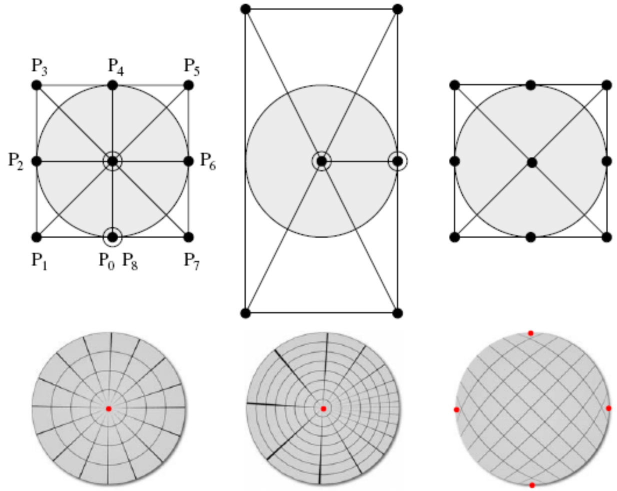
# To mesh or not to mesh: that is the question

Stefan Kollmannsberger
Computation in Engineering
Technische Universität München

# b) remarks to Isogeometric Analysis

-> in order to have to generate a mesh
one draws the (coarsest) mesh and refines it

issue 1: The geometry is the mesh is the discretization
"Analysis aware modeling"




E. Cohen, T. Martin, R.M. Kirby, T. Lyche, R.F. Riesenfeld: Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis, *Comput. Methods Appl. Mechn. Engrg: 199:*2010.