# Isogeometric Analysis

*The better alternative to FEM?*

**Delft Institute of Applied Mathematics**

Matthias Möller (m.moller@tudelft.nl)

23 March 2016

# Aim of this lunch lecture

- Give a brief introduction to Isogeometric Analysis
- Outline advantages of IgA over the Finite Element Method
- Address practical transformation of a FEM into an IgA code

# The Finite Element Method

**Strong problem formulation**

Find $u$ such that

$$Lu = f \qquad \text{in } \Omega, \text{ subject to BC's and IC's.}$$

# The Finite Element Method

## Strong problem formulation

Find $u$ such that

$$Lu = f \qquad \text{in } \Omega, \text{ subject to BC's and IC's.}$$

## Variational formulation

Find $u \in V$ such that

$$\int_\Omega wLu\,\mathrm{d}\mathbf{x} = \int_\Omega wf\,\mathrm{d}\mathbf{x} \qquad \text{for all } w \in W$$

subject to BC's and IC's.

# The Finite Element Method

**Strong problem formulation**

Find $u$ such that

$$Lu = f \qquad \text{in } \Omega, \text{ subject to BC's and IC's.}$$

**Discretised variational formulation**

Find $u_h \in V_h \subset V$ such that

$$\int_{\Omega_h} w_h L u_h \mathrm{d}\mathbf{x} = \int_{\Omega_h} w_h f \,\mathrm{d}\mathbf{x} \qquad \text{for all } w_h \in W_h \subset W$$

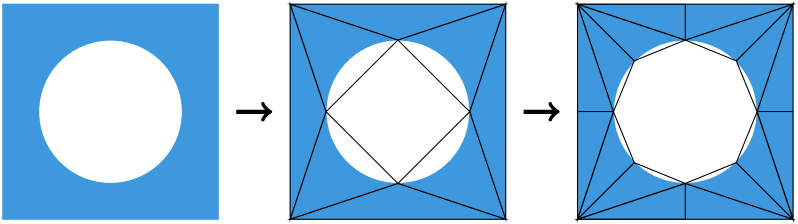
subject to BC's and IC's.

TU Delft

# Is $h = \mathrm{h}$?

# Is $h = \mathrm{h}$?

**No**, since the triangulation $\mathcal{T}_{\mathrm{h}}$ of the geometry $\Omega_{\mathrm{h}}$ is in many cases of lower polynomial order (e.g., pw. linear) than the approximation of the solution $u_h$ (e.g., pw. quadratic or higher)



TU Delft

# Is $h = \mathrm{h}$?

**No**, since the triangulation $\mathcal{T}_\mathrm{h}$ of the geometry $\Omega_\mathrm{h}$ is in many cases of lower polynomial order (e.g., pw. linear) than the approximation of the solution $u_h$ (e.g., pw. quadratic or higher)

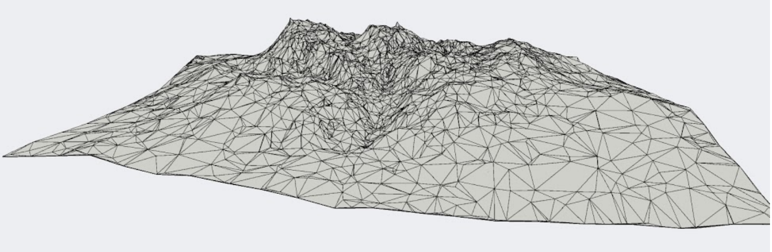
**Theoretical problem**

But we require computational meshes that represent the curved boundary with high accuracy to obtain optimal convergence

$$\|u - u_h\| = \mathcal{O}(h^{p+1})$$

# Is $h = \mathrm{h}$?

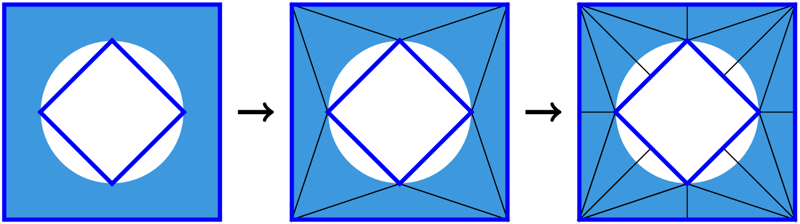
**Even worse**, in many practical simulations the geometry is only given as surface triangulation $\mathcal{S}_{\mathrm{h}}$ from which the volumetric triangulation $\mathcal{T}_{\mathrm{h}}$ of the domain $\Omega_{\mathrm{h}}$ needs to be constructed



TUDelft

# Is $h = \mathrm{h}$?

**Even worse**, in many practical simulations the geometry is only given as surface triangulation $\mathcal{S}_\mathrm{h}$ from which the volumetric triangulation $\mathcal{T}_\mathrm{h}$ of the domain $\Omega_\mathrm{h}$ needs to be constructed



# Common problems with the FEM

1. How to accurately refine, coarsen and/or deform $\Omega_h$ without a **parametric description** of the true geometry $\Omega$?
2. How to generate **high-quality curved computational meshes** for high-order methods in complex geometries?
3. How to define normal vectors along element boundaries?
4. How to construct finite element basis functions with $C^1$ **continuity** (or higher) across element boundaries?
   - Would lead to globally continuous derivative field
   - Would solve many problems with Material Point Method

# Example

**Poisson's problem**

Find $u$ such that

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 = 1\} \\ u &= 0 \qquad \text{on } \Gamma = \partial\Omega \end{aligned}$$

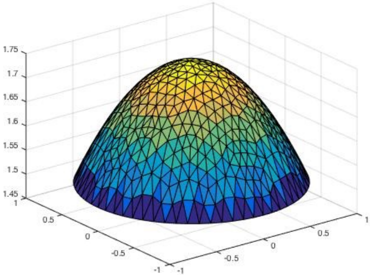
**Discretised variational formulation**

Find $u_h \in V_h = \{u_h \in \mathcal{H}^1(\Omega_\mathrm{h}) : u_h = 0 \text{ on } \Gamma\}$ such that

$$\int_{\Omega_\mathrm{h}} \nabla w_h \cdot \nabla u_h \mathrm{d}\mathbf{x} = \int_{\Omega_\mathrm{h}} w_h f \mathrm{d}\mathbf{x} \qquad \text{for all } w_h \in W_h = V_h$$

# Example

The finite element solution with pw. linear boundary approximation and pw. quadratic basis functions looks like this

# Example

The finite element solution with pw. linear boundary approximation and pw. quadratic basis functions looks like this

# The mission

**Isogeometric Analysis**

Computational analysis framework that ensures $h = \mathrm{h}$

- Make use of a parametric description of the geometry ($\Omega = \Omega_\mathrm{h}$) throughout all computational steps (FE-analysis, refinement/ coarsening, shape deformation, multi-physics coupling, ...)
- Use the same mathematical tools (**B-splines** or NURBS or ...) to represent the geometry $\Omega_\mathrm{h}$ and the FE-solution $u_h$

# Polynomial spaces

**Polynomial space**

The space of polynomials of degree $p$ over the interval $[a, b]$ is

$$\Pi^p([a,b]) := \{q(x) \in \mathcal{C}^\infty([a,b]) \,:\, q(x) = \sum_{i=0}^{p} c_i x^i, c_i \in \mathbb{R}\}$$

Example: $\Pi^2([0,1])$

- Canonical basis

$$\mathcal{B} = \{1, x, x^2\}$$

- Polynomials

$$q(x) = c_0 + c_1 x + c_2 x^2$$

# Spline space

**Polynomial splines**

Let $\mathcal{P} = \{a = x_1 < \cdots < x_{p+1} = b\}$ be a partition of the interval $\Omega_0$ and $\mathcal{M} = \{1 \leq m_i \leq p+1\}$ a set of positive integers. The polynomial spline of degree $p$ is defined as $s : \Omega_0 \mapsto \mathbb{R}$ if

$$s\big|_{[x_i, x_{i+1}]} \in \Pi^p([x_i, x_{i+1}]), \qquad i = 1, \ldots, k$$

$$\frac{d^j}{dx^j} s_{i-1}(x_i) = \frac{d^j}{dx^j} s_i(x_i), \qquad \begin{array}{l} i = 2, \ldots, k, \\ j = 0, \ldots, p - m_i \end{array}$$

Polynomial splines of degree $p$ form the spline space $\mathcal{S}(\Omega_0, p, \mathcal{M}, \mathcal{P})$.

## Open knot vector

An open knot vector is a sequence of non-decreasing coordinates $\xi_i \in [a, b] \subset \mathbb{R}$ in the parameter space $\Omega_0 = [a, b]$

$$\Xi = (\underbrace{\xi_1 = \cdots = \xi_{p+1}}_{p+1 \text{ times}}, \ldots, \underbrace{\xi_i, \ldots, \xi_i}_{m_i \text{ times}}, \ldots, \underbrace{\xi_{n+1} = \cdots = \xi_{n+p+1}}_{p+1 \text{ times}})$$

where

- $p$ is the polynomial order of the B-splines
- $n$ is the number of B-spline functions
- $\xi_i$ is the $i$-th knot with knot index $i$
- $m_i$ is the multiplicity of knot $\xi_i$

# B-spline basis functions

## Cox-de Boor recursion formula

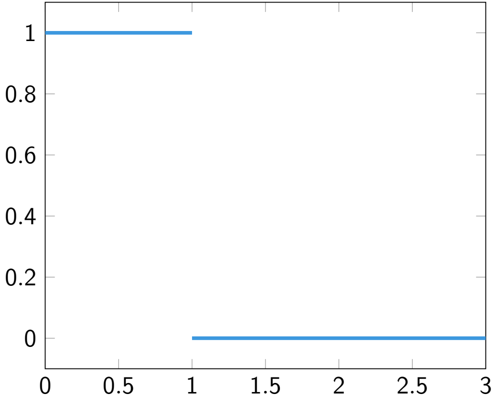
$p = 0$

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$p > 0$

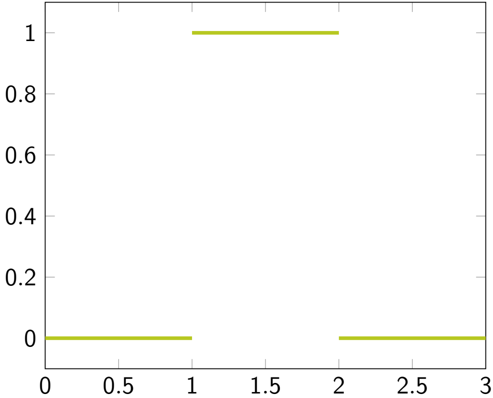
$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi)$$

# B-spline basis functions



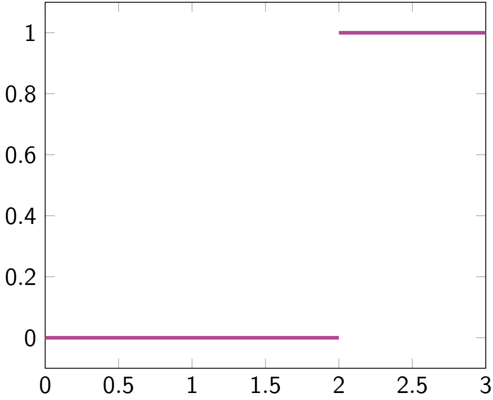

Constant basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions



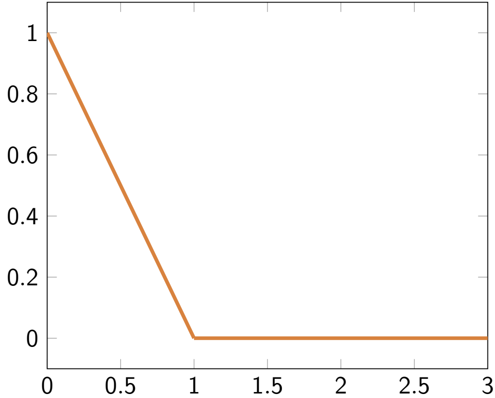

Constant basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions




Constant basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

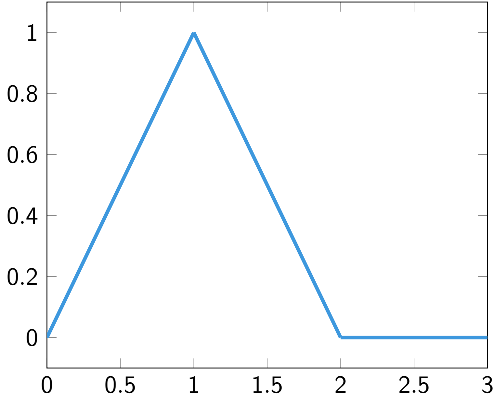
# B-spline basis functions




Linear basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

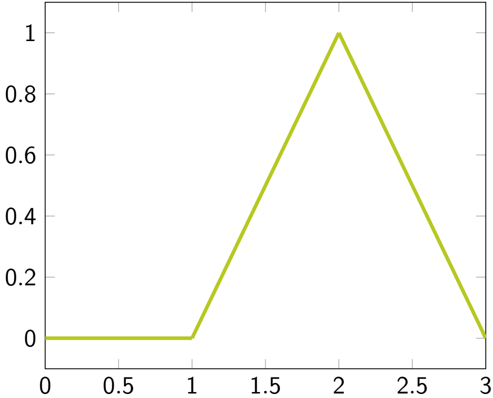
TU Delft

# B-spline basis functions



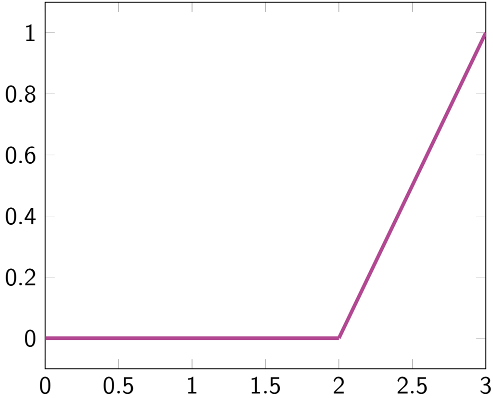

Linear basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions



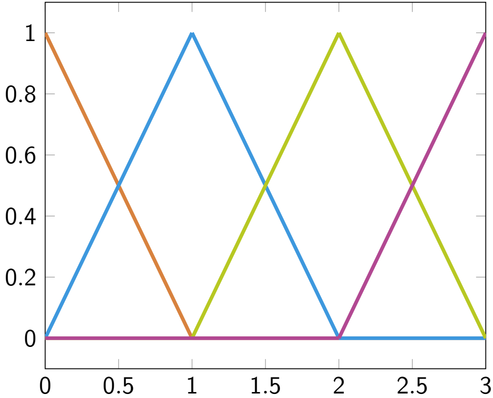

Linear basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions



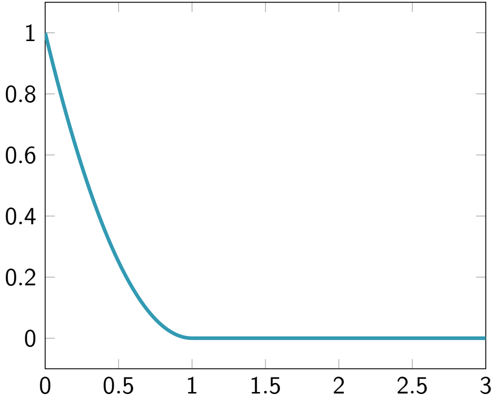

Linear basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions



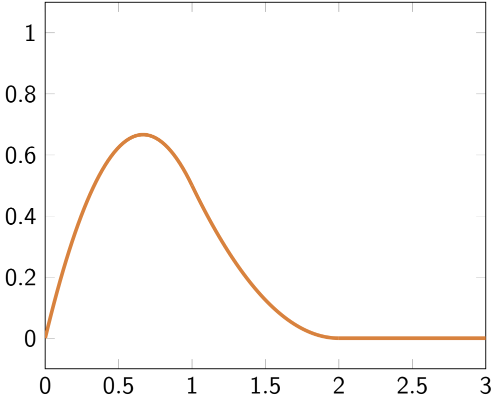

Linear basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions



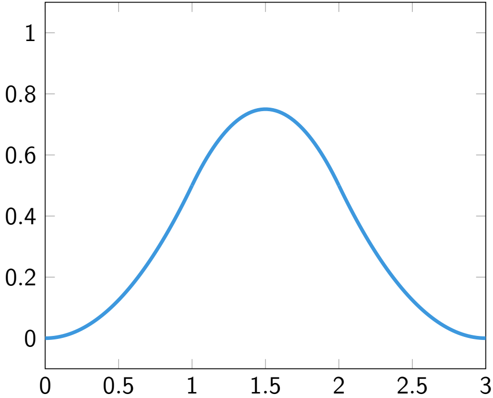

Quadratic basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions



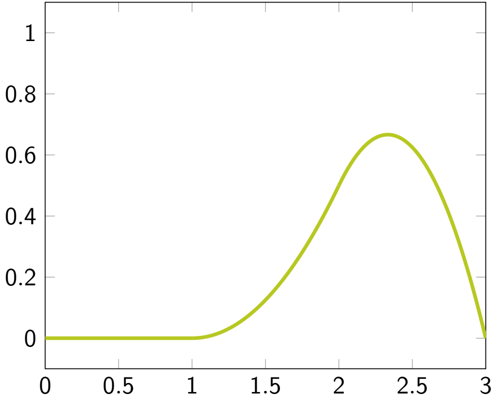

Quadratic basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions



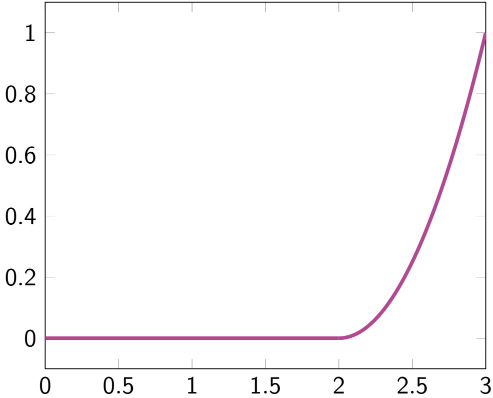

Quadratic basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions



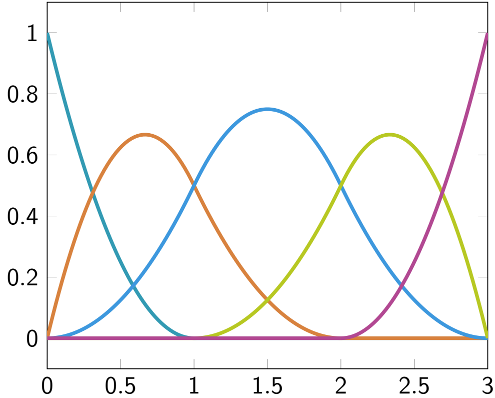

Quadratic basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions




Quadratic basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

# B-spline basis functions




Quadratic basis functions corresponding to $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$

TU Delft

# Properties of B-spline basis functions

**Compact support**

$$\text{supp } N_{i,p}(\xi) = [\xi_i, \xi_{i+p+1}), \quad i = 1, \ldots, n$$

- System matrices are sparse like in the standard FEM
- Support grows with the polynomial order so that system matrices have a slightly broader stencil due to the coupling of degrees of freedom over multiple element layers

# Properties of B-spline basis functions

**Compact support**

$$\text{supp } N_{i,p}(\xi) = [\xi_i, \xi_{i+p+1}), \quad i = 1, \ldots, n$$

**Strict positiveness**

$$N_{i,p}(\xi) > 0 \quad \text{for } \xi \in (\xi_i, \xi_{i+p+1}), \quad i = 1, \ldots, n$$

- Consistent mass matrix has no negative off-diagonal entries
- Lumped mass matrix is not singular (no zero diagonal entries)

# Properties of B-spline basis functions

**Compact support**

$$\text{supp } N_{i,p}(\xi) = [\xi_i, \xi_{i+p+1}), \quad i = 1, \ldots, n$$

**Strict positiveness**

$$N_{i,p}(\xi) > 0 \quad \text{for } \xi \in (\xi_i, \xi_{i+p+1}), \quad i = 1, \ldots, n$$

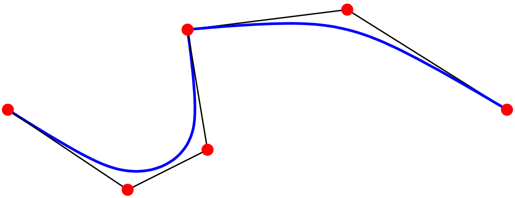
**Partition of unity**

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1 \quad \text{for all } \xi \in [a, b]$$

# Parametric geometry description

**Spline curve**

$$C(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\mathbf{B}_i \qquad \text{set of control points } \mathbf{B}_i \in \mathbb{R}^d, d \geq 1$$

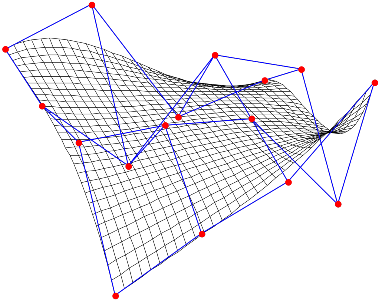


# Parametric geometry description

## Spline surface

$$S(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\xi) N_{j,q}(\eta) \mathbf{B}_{i,j}$$

set of control points
$\mathbf{B}_{i,j} \in \mathbb{R}^d, d \geq 1$



# Marriage of geometry & analysis

## Spline volume

$$V(\xi,\eta,\zeta)=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l}N_{i,p}(\xi)N_{j,q}(\eta)N_{k,r}(\zeta)\mathbf{B}_{i,j,k}$$

## Approximate solution

$$u_h(\xi,\eta,\zeta)=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l}N_{i,p}(\xi)N_{j,q}(\eta)N_{k,r}(\zeta)u_{i,j,k}$$

# Geometry & analysis in practice

**Isogeometric Analysis**

1. Construct **parametric geometry** $\Omega(\xi, \eta, \zeta)$:
   - generate basis $\mathcal{B} = \{N_{i,p} N_{j,q} N_{k,r}\}_{i,j,k}^{n,m,l}$ and
   - choose control points $\{\mathbf{B}_{i,j,k}\}$
2. Construct **computational mesh** $\Omega_{\mathrm{h}}(\xi, \eta, \zeta)$ and **computational basis** $\mathcal{B}_h$ by shape preserving
   - knot insertion ($h$-refinement);
   - order elevation ($p$-refinement);
   - regularity adjustment ($k$-refinement)

# Application: IgA for flow problems

## Flow problems

- Convection-diffusion equation

$$\nabla \cdot (\mathbf{v}u - d\nabla u) = f$$

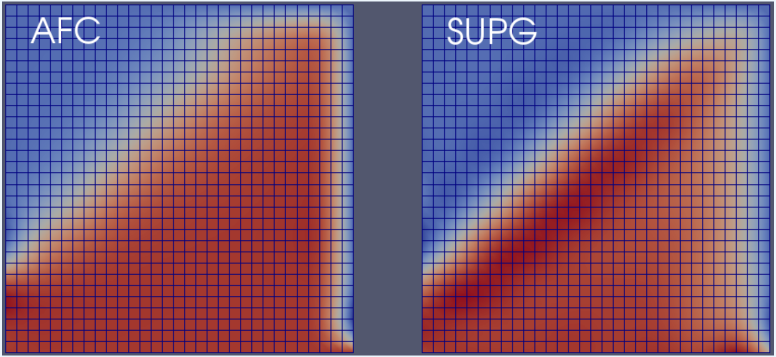
- Compressible Euler equations

$$\partial_t \begin{bmatrix} \rho \\ \rho\mathbf{v} \\ E \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v} \otimes \mathbf{v} + \mathcal{I}p \\ \mathbf{v}(E+p) \end{bmatrix} = 0$$

Collaboration with A. Jaeschke from Technical University Łódź

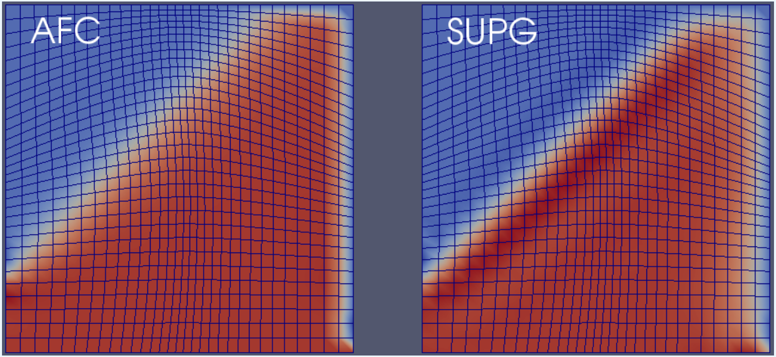
# Application: IgA for flow problems

## Convection skew to the mesh



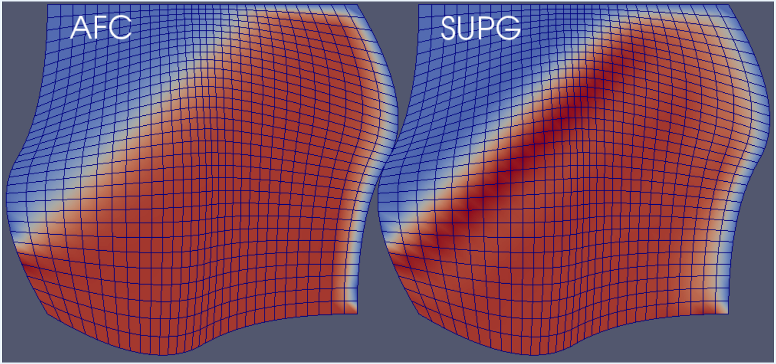
# Application: IgA for flow problems

## Convection skew to the mesh



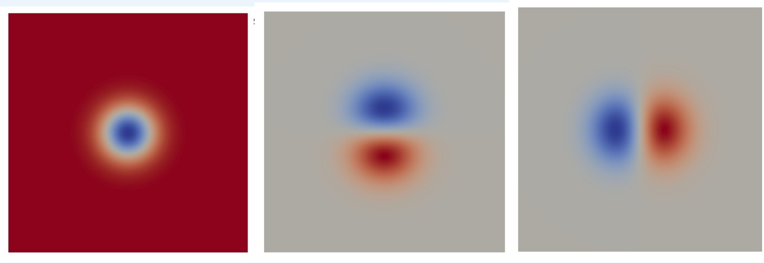
# Application: IgA for flow problems

## Convection skew to the mesh



# Application: IgA for flow problems

**Stationary isentropic vortex**



$\rho$ $\qquad$ $v_x$ $\qquad$ $v_y$

- Animation: Rotation of isentropic vortex ($\rho$-values)

# Application: IgA on evolving manifolds

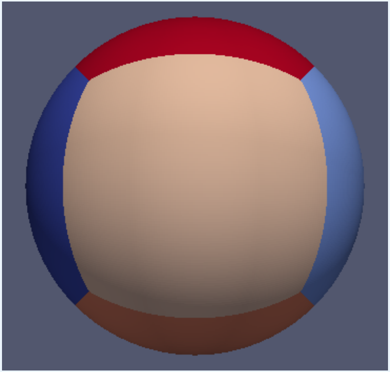
**Gray-Scott reaction-diffusion model**

$$u_t + u(\ln\sqrt{g_t})_t - d_1 \Delta u = F(1-u) - uv^2$$
$$v_t + v(\ln\sqrt{g_t})_t - d_2 \Delta v = -(F+H)v + uv^2$$
$$\mathbf{s} = Kv\mathbf{n}$$

MSc-thesis by J. Hinz from Technical University Delft

# Application: IgA on evolving manifolds

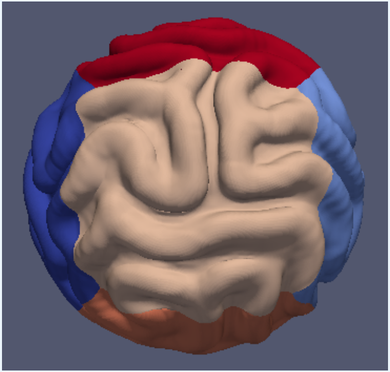
## Brain development



- multi-patch geometry
- periodic basis functions
- $C^{p-1}$ continuity along patch boundaries
- $C^0$ continuity in the vicinity of the triple points

# Application: IgA on evolving manifolds

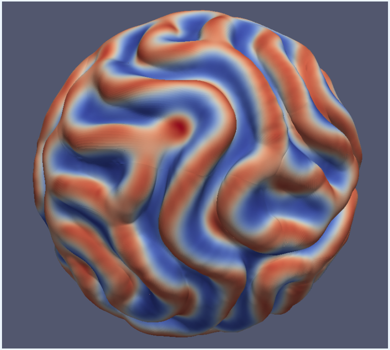
## Brain development



- multi-patch geometry
- periodic basis functions
- $C^{p-1}$ continuity along patch boundaries
- $C^0$ continuity in the vicinity of the triple points

# Application: IgA on evolving manifolds

## Brain development



- multi-patch geometry
- periodic basis functions
- $C^{p-1}$ continuity along patch boundaries
- $C^0$ continuity in the vicinity of the triple points

# Collaboration with Deltares

**Material Point Method**

- Represent properties of continuum (velocity, stresses, etc.) at material points and let particles move in time
- Solve equations of motion on fixed background grid

What people like about it

- Easy treatment of free-surface, multi-phase/-material problems
- Easy treatment of large deformations (no mesh tangling)
- Easy treatment of convection (no spurious wiggles)

# Collaboration with Deltares

**Material Point Method**

- Represent properties of continuum (velocity, stresses, etc.) at material points and let particles move in time
- Solve equations of motion on fixed background grid

What people 'fear' about it

- Occurrence of grid crossing errors/empty cells
- Poor convergence or even lack of convergence
- Accurate data transfer between particles and dof's in FEM
- Singularity of lumped mass matrix in higher-order FEM

# The Material Point Method

# Building blocks of MPM

**Update of particle properties from dof's**

$$\Delta\epsilon_p^{t+\Delta t} = \sum_{i=1}^{N_{\text{dof}}} \nabla\phi_i(x_p^t)\Delta u_i^{t+\Delta t}$$

**Update of dof's from particle properties**

$$\mathbf{F}_i^{\text{int,t}} = \sum_{p=1}^{N_{\text{p}}} \sigma_p^t \nabla\phi_i(x_p^t) V_p^t$$

# IgA the better alternative to FEM?

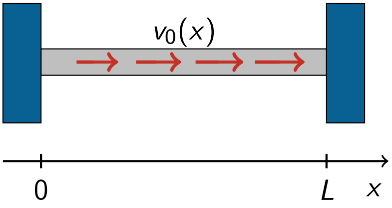
**FEM**

- Lagrange-type basis functions $\phi_i$ are $C^0$ across element boundaries so that the values of $\nabla\phi_i$ can have jumps
- Lumped mass matrix can become singular

**IgA**

- B-spline basis functions $N_{i,p}$ are $C^{p-1}$ across element boundaries so that $\nabla N_{i,p}$ is $C^{p-2}$ (continuous for $p \geq 2$)
- Lumped mass matrix is non-singular

# Vibrating bar




$$\frac{\partial^2 u}{\partial t^2} = \frac{E}{\rho}\frac{\partial^2 u}{\partial x^2}$$

Boundary conditions:

$u(0,t) = 0$

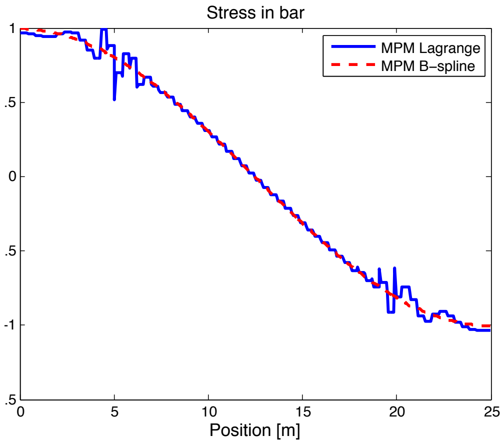
$u(L,t) = 0$

Initial conditions:

$u(x,0) = 0$

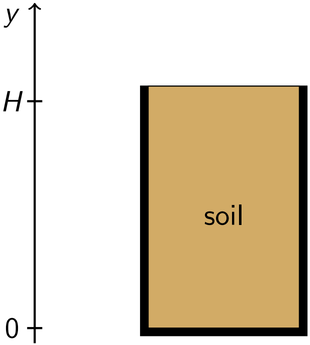
$\frac{\partial u}{\partial t}(x,0) = v_0 \sin\left(\frac{\pi x}{L}\right)$

# Application: Vibrating bar




MSc-project by R. Tielen (jointly supervised with L. Beuth)

# Soil column under self weight




$$\frac{\partial^2 u}{\partial t^2} = \frac{E}{\rho}\frac{\partial^2 u}{\partial y^2} - g$$

Boundary conditions:

$$u(0,t) = 0$$

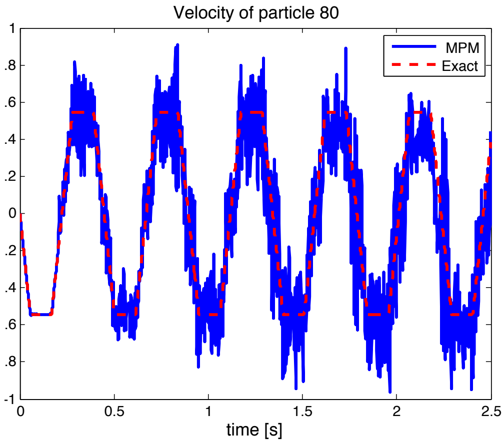
$$\frac{\partial u}{\partial y}(H,t) = 0$$

Initial conditions:

$$u(y,0) = 0$$

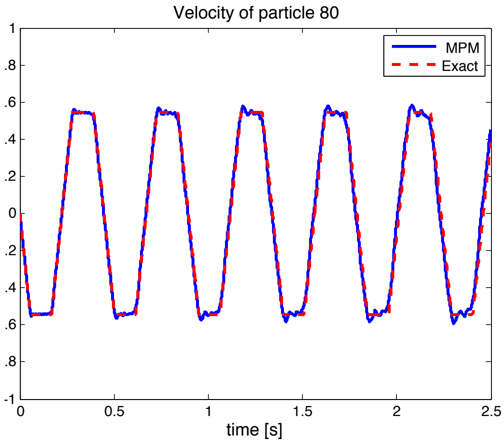
$$\frac{\partial u}{\partial t}(y,0) = 0$$

# Application: Oedometer




MSc-project by R. Tielen (jointly supervised with L. Beuth)

# Application: Oedometer




MSc-project by R. Tielen (jointly supervised with L. Beuth)

# Building blocks of your FEM code

## Finite element loop

$$A = \sum_{e \in \mathcal{T}_h} C_e K_e C_e^\top$$

$$b = \sum_{e \in \mathcal{T}_h} C_e f_e$$

- Element matrix $K_e$ and vector $f_e$
- Connectivity matrix $C_e$ (local-global mapping)

## Numerical quadrature

$$\int_a^b f(x)dx \approx \sum_{c=0}^{N} \omega_c f(x_c)$$

- Quadrature weights $\omega_c$
- Quadrature points $x_c$

TUDelft

# Building blocks of your IgA code

## Loop over elements in index domain

$$A = \sum_{e=1}^{n+p} C_e K_e C_e^\top$$

$$b = \sum_{e=1}^{n+p} C_e f_e$$

- Element matrix $K_e$ and vector $f_e$
- Connectivity matrix $C_e$ (local-global mapping)

## Numerical quadrature

$$\int_a^b f(x)dx \approx \sum_{c=1}^{N} \omega_c f(x_c)$$

- Quadrature weights $\omega_c$
- Quadrature points $x_c$

# Conclusions

- Isogeometric Analysis has several advantages over standard FEM
  - parametric geometry representation
  - no singular lumped mass matrices
  - no grid crossing errors in MPM
- Conversion of FEM code into IgA is straightforward
- Established techniques to reconstruct parametric curves, surfaces, and volumes from non-uniform sampling data
  - multi-variate spline interpolation
  - least-squares spline approximation

# List of IgA software packages

- G+SMO: http://www.gs.jku.at/gs_gismo.shtml
- igatools:
  https://github.com/igatoolsProject/igatools/wiki
- PetIGA: https://bitbucket.org/dalcinl/petiga/
- GeoPDEs: http://rafavzqz.github.io/geopdes/
- igafem: https://sourceforge.net/projects/cmcodes/
- deal.II: https://dealii.org
- LS-DYNA: http://www.lstc.com/products/ls-dyna