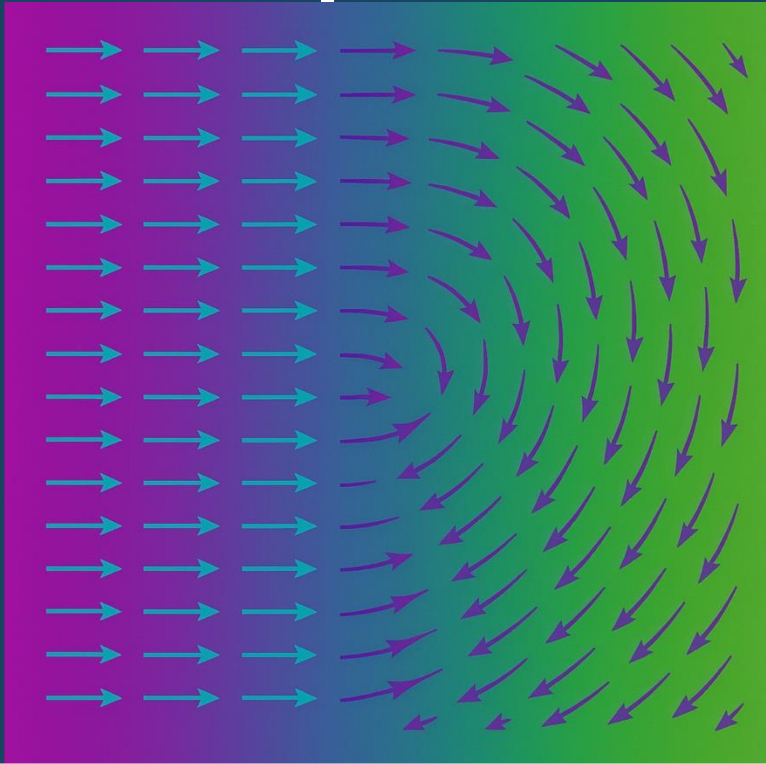
Ferrite now has true vector interpolations - what, why, and how?



Knut Andreas Meyer
Material and Computational Mechanics
Chalmers University of Technology

Outline

- **What** are true vector interpolations?
- **Why** are these useful?
- **How** are these implemented in Ferrite?
- **What** about boundary conditions?
- **Why** are these implemented in Ferrite?
- **How** should you use them?

What are vector interpolations

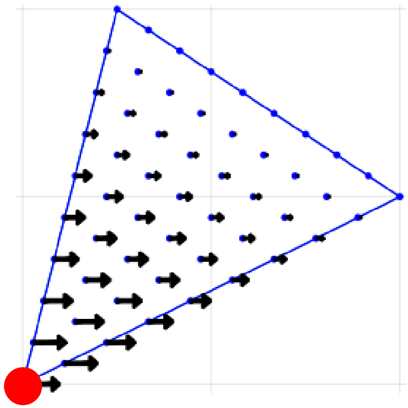
Current vectorized interpolations

- Based on scalar interpolations

$$\boldsymbol{N}_1(\boldsymbol{\xi}) = \begin{bmatrix} M_1(\boldsymbol{\xi}) \\ 0 \end{bmatrix}, \quad \boldsymbol{N}_3(\boldsymbol{\xi}) = \begin{bmatrix} M_2(\boldsymbol{\xi}) \\ 0 \end{bmatrix}, \quad \boldsymbol{N}_5(\boldsymbol{\xi}) = \begin{bmatrix} M_3(\boldsymbol{\xi}) \\ 0 \end{bmatrix}$$

$$\boldsymbol{N}_2(\boldsymbol{\xi}) = \begin{bmatrix} 0 \\ M_1(\boldsymbol{\xi}) \end{bmatrix}, \quad \boldsymbol{N}_4(\boldsymbol{\xi}) = \begin{bmatrix} 0 \\ M_2(\boldsymbol{\xi}) \end{bmatrix}, \quad \boldsymbol{N}_6(\boldsymbol{\xi}) = \begin{bmatrix} 0 \\ M_3(\boldsymbol{\xi}) \end{bmatrix}$$

- Vectors aligned with coordinate system
- Associated with algebraic nodes
- Inherits component-wise the continuity of the scalar interpolation



Lagrange

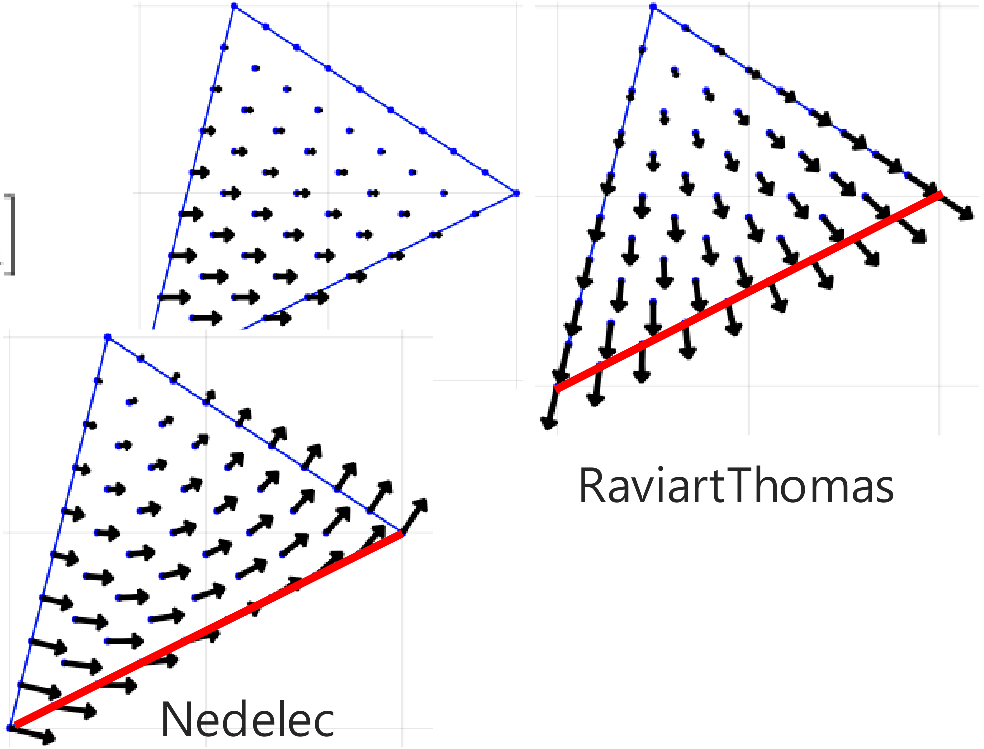
What are vector interpolations

True vector interpolations

- Directly vector shape functions, e.g.
  RaviartThomas{RefTriangle, 1}

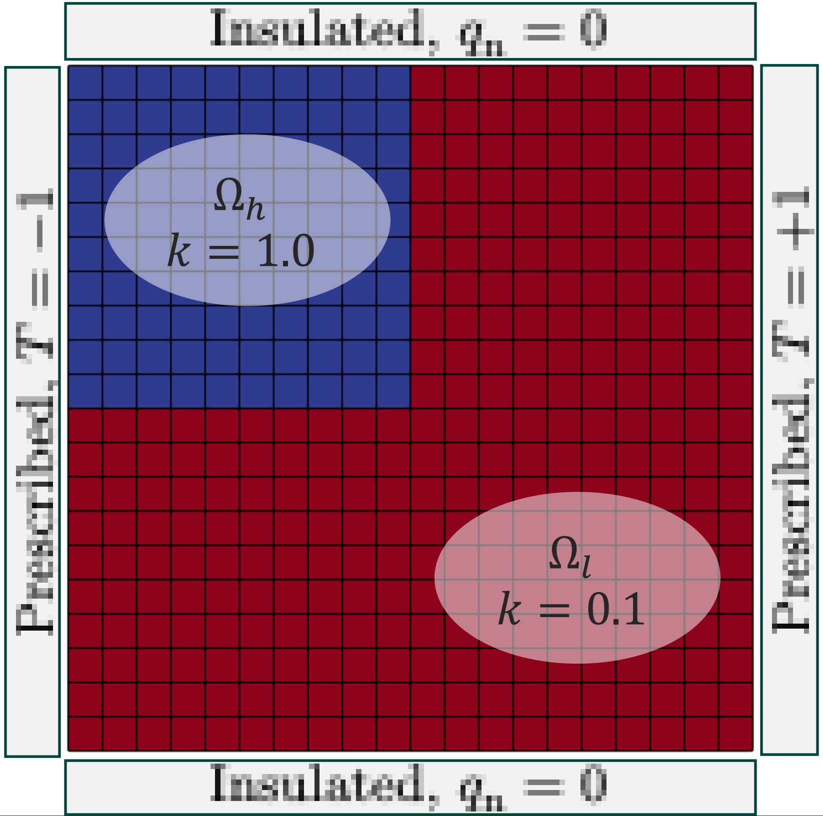
$$\boldsymbol{N}_1(\boldsymbol{\xi}) = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}, \quad \boldsymbol{N}_2(\boldsymbol{\xi}) = \begin{bmatrix} \xi_1 - 1 \\ \xi_2 \end{bmatrix}, \quad \boldsymbol{N}_3(\boldsymbol{\xi}) = \begin{bmatrix} \xi_1 \\ \xi_2 - 1 \end{bmatrix}$$

- Vectors aligned with the cell
- Associated with different entities, e.g. edges and faces
- Allows normal/tangential continuity across element borders



Why? The heat equation

CHALMERS
UNIVERSITY OF TECHNOLOGY

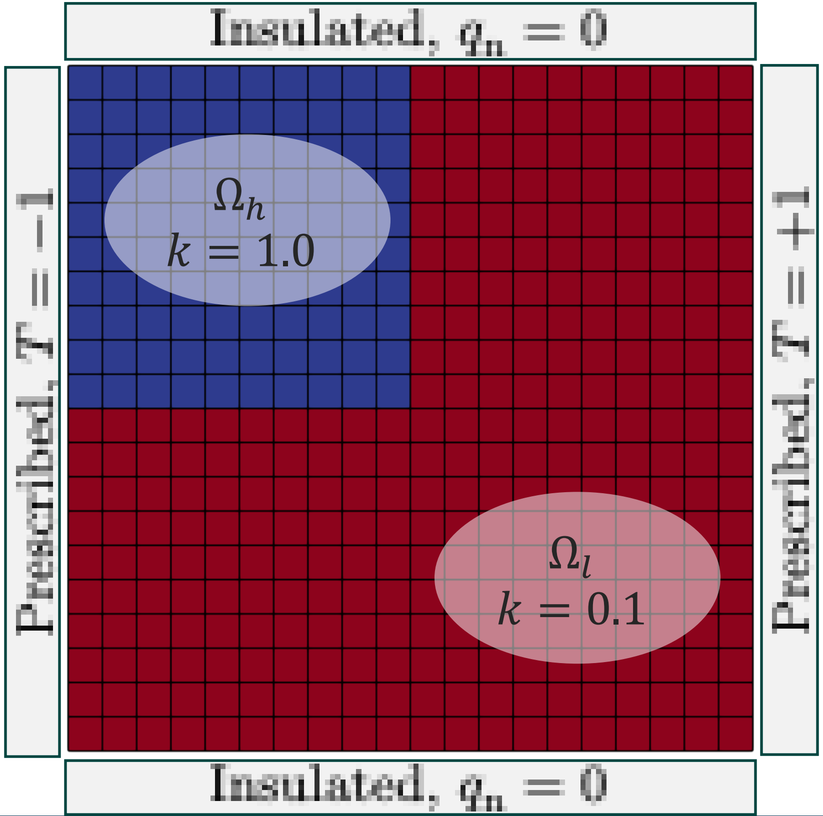


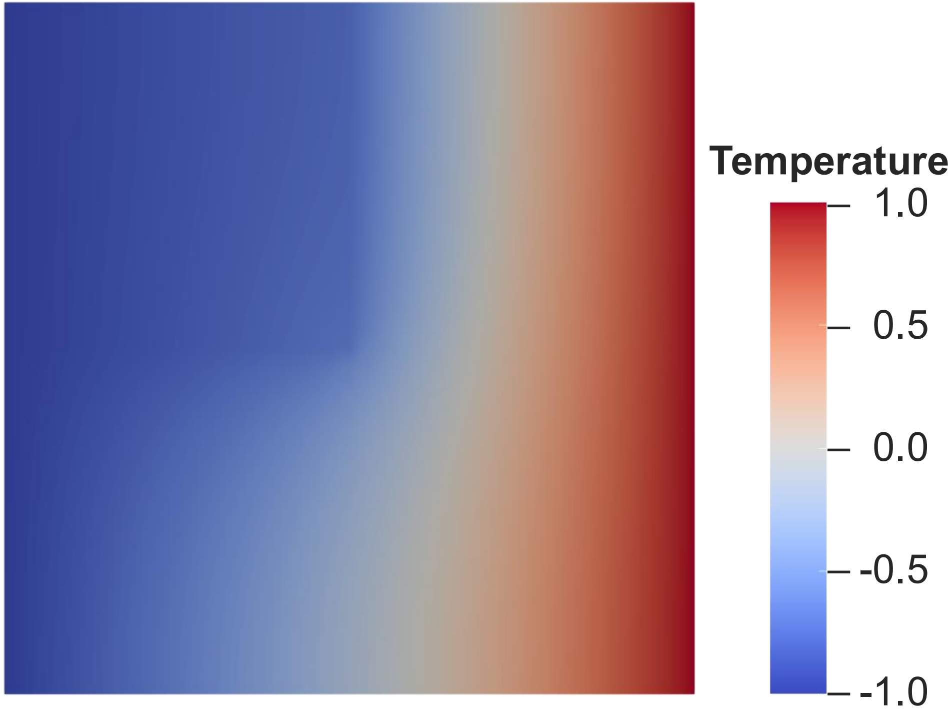
Energy conservation: $\boldsymbol{q}(\boldsymbol{x}) \cdot \boldsymbol{\nabla} = 0$

Fourier's law: $\boldsymbol{q}(\boldsymbol{x}) = -k(\boldsymbol{x})\,[\boldsymbol{\nabla} T(\boldsymbol{x})]$

How will the heat flux field look like?

Why? The heat equation





Why? The heat equation

How can we visualize the heat flux?

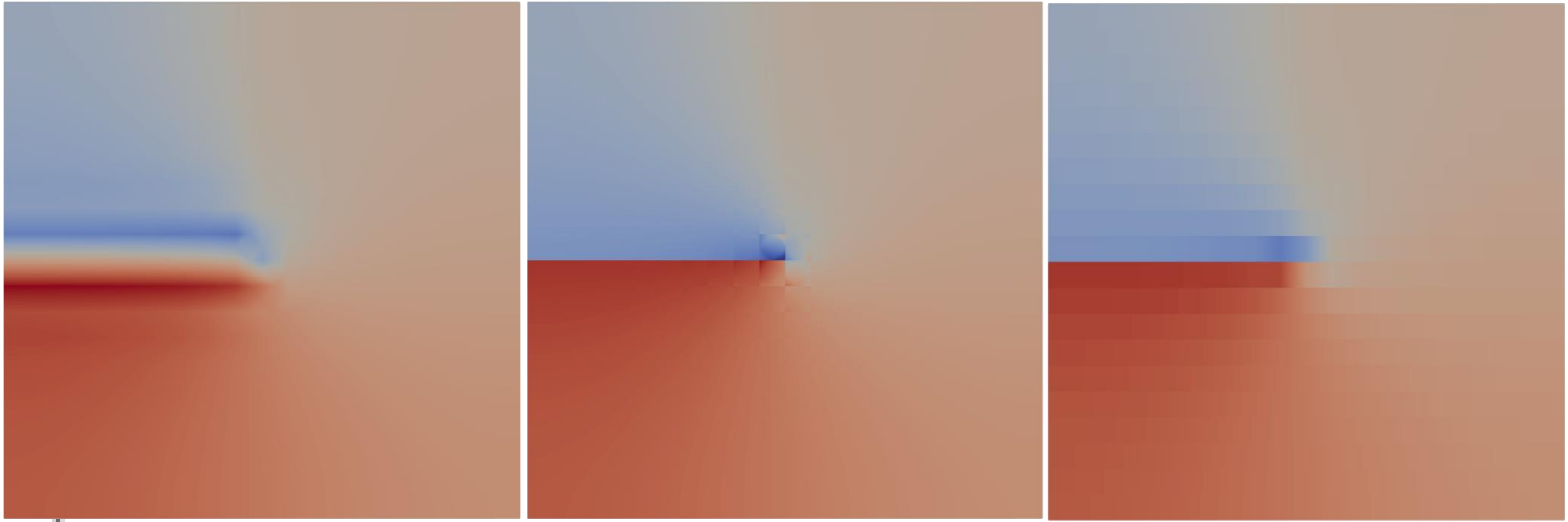
L_2 projection

- Data function, $\boldsymbol{f}(\boldsymbol{x})$
- FE-approximation, $\boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{N_{\text{dofs}}} \boldsymbol{N}_i(\boldsymbol{x}) a_i$
- minimize $e = \int_\Omega ||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x})||^2 \ \mathrm{d}\Omega$

$$\frac{1}{2}\frac{\partial e}{\partial a_i} = \underbrace{\int_\Omega \boldsymbol{N}_i(\boldsymbol{x}) \cdot \boldsymbol{N}_j(\boldsymbol{x}) \ \mathrm{d}\Omega}_{K_{ij}} \ a_j - \underbrace{\int_\Omega \boldsymbol{N}_i(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}) \ \mathrm{d}\Omega}_{f_i} = 0$$

Why? The heat equation

L_2 projection results



H^1 space

Lagrange{
RefQuadrilateral, 1}

L_2 space

DiscontinuousLagrange{
RefQuadrilateral, 1}

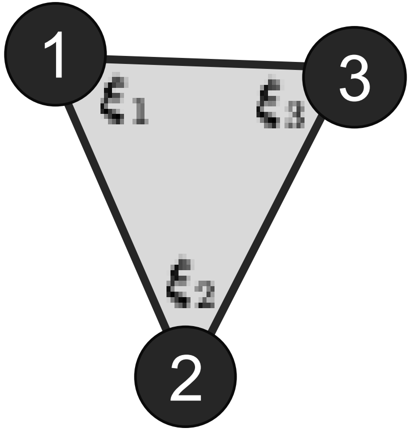
$H(\mathrm{div})$ space

RaviartThomas{
RefQuadrilateral, 1}

How are these implemented?

Nodal interpolations

$$M_i(\boldsymbol{\xi}_j) = \delta_{ij}$$



H(div) interpolations

$$\int_0^1 \boldsymbol{N}_i(\boldsymbol{\xi} = \boldsymbol{g}_j(s))\boldsymbol{f}_j(s) \cdot \boldsymbol{n}_j \, \mathrm{d}s = \delta_{ij} \quad \text{no sum on } j$$

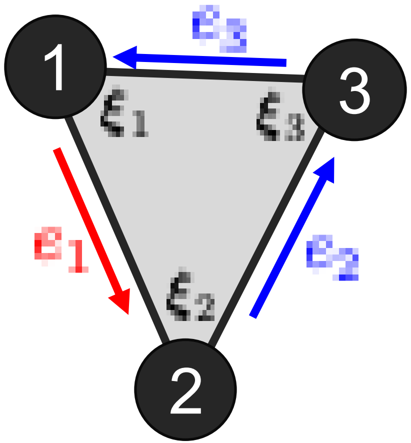
How are these implemented?

Vector interpolations, example: H(div)

RaviartThomas{RefTriangle, 1}

$$\int_0^1 \boldsymbol{N}_i(\boldsymbol{\xi}_1 + \boldsymbol{v}s) \cdot \boldsymbol{n}_1 \, ||\boldsymbol{v}|| \;\mathrm{d}s = \begin{cases} 1 & i = 1 \\ 0 & i \neq 1 \end{cases}$$

$$\boldsymbol{v} = \boldsymbol{\xi}_2 - \boldsymbol{\xi}_1$$



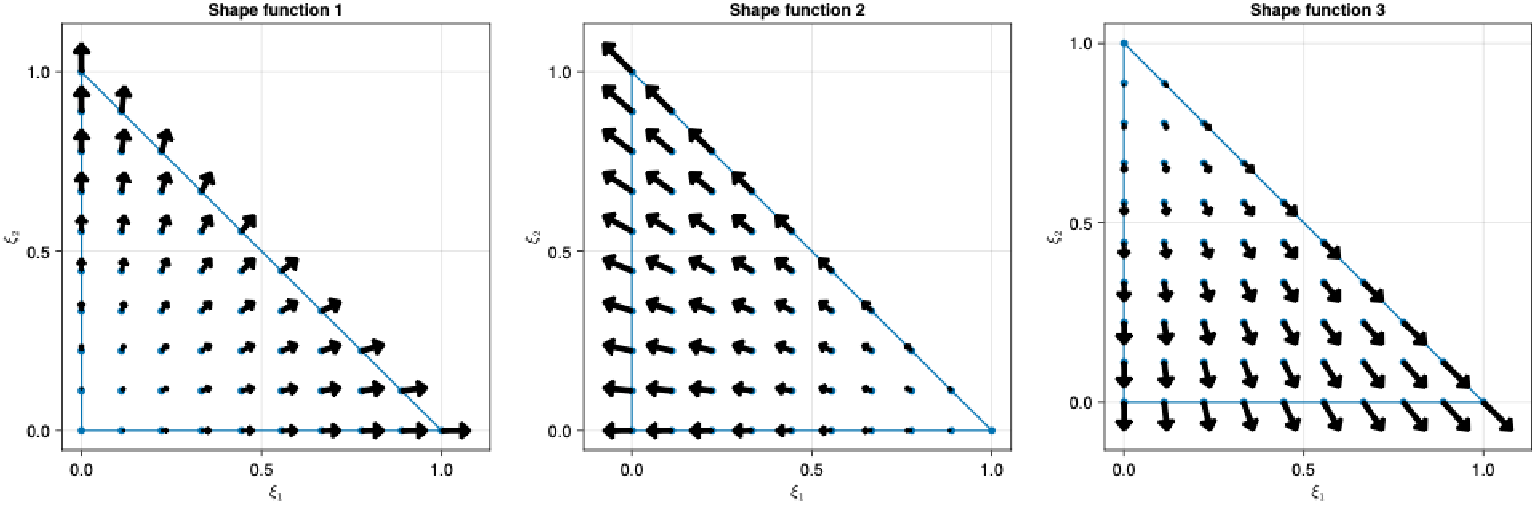
H(div) interpolations

$$\int_0^1 \boldsymbol{N}_i(\boldsymbol{\xi} = \boldsymbol{g}_j(s)) f_j(s) \cdot \boldsymbol{n}_j \;\mathrm{d}s = \delta_{ij} \qquad \text{no sum on } j$$

How are these implemented?

Vector interpolations, example: H(div)

RaviartThomas{RefTriangle, 1}



How are these implemented?

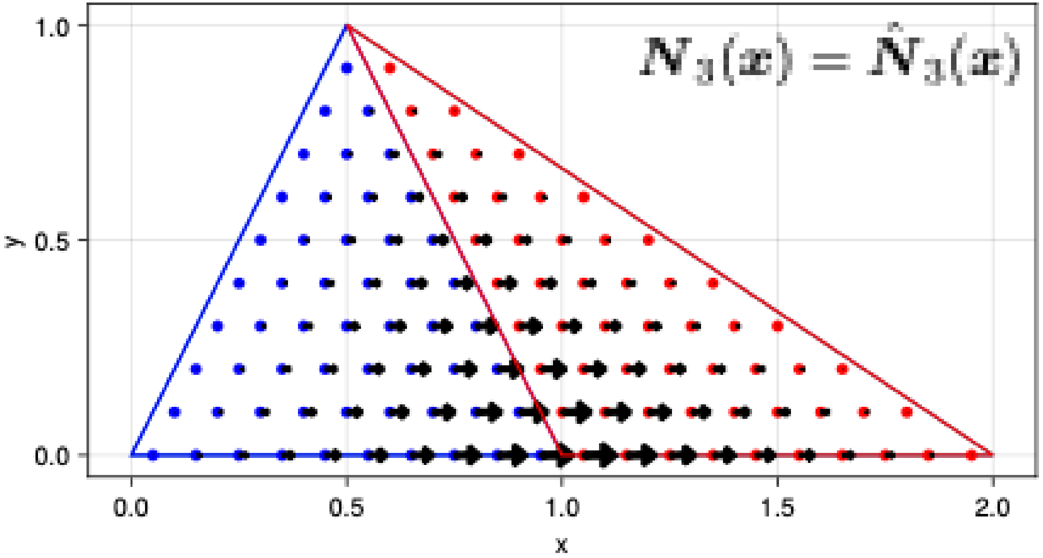
Vectorized interpolations, example:
Lagrange{RefTriangle, 1}()^2

```
Ferrite.IdentityMapping()
```

$$N(x) = \hat{N}(\xi(x))$$

$$\frac{\mathrm{d}N}{\mathrm{d}x} = \frac{\mathrm{d}\hat{N}}{\mathrm{d}\xi} \cdot \frac{\mathrm{d}\xi}{\mathrm{d}x} = \frac{\mathrm{d}\hat{N}}{\mathrm{d}\xi} \cdot J^{-1}$$

$$J := \frac{\mathrm{d}x}{\mathrm{d}\xi}$$



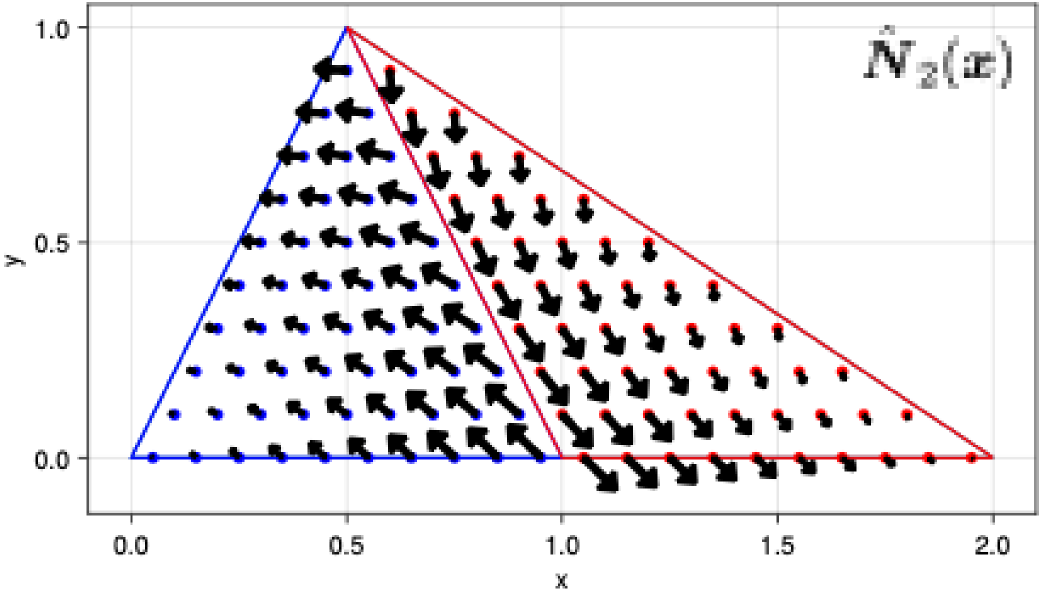

```
N  = shape_value(cellvalues, q_point, shapenr)

N̂ = reference_shape_value(interpolation, local_coord, shapenr)
```

How are these implemented?

Vector interpolations, example: H(div)
RaviartThomas{RefTriangle, 1}

```
Ferrite.IdentityMapping()
```

$$\boldsymbol{N}(\boldsymbol{x}) = \hat{\boldsymbol{N}}(\boldsymbol{\xi}(\boldsymbol{x}))$$




```
N = shape_value(cellvalues, q_point, shapenr)

N̂ = reference_shape_value(interpolation, local_coord, shapenr)
```

How are these implemented?

Vector interpolations, example: H(div)

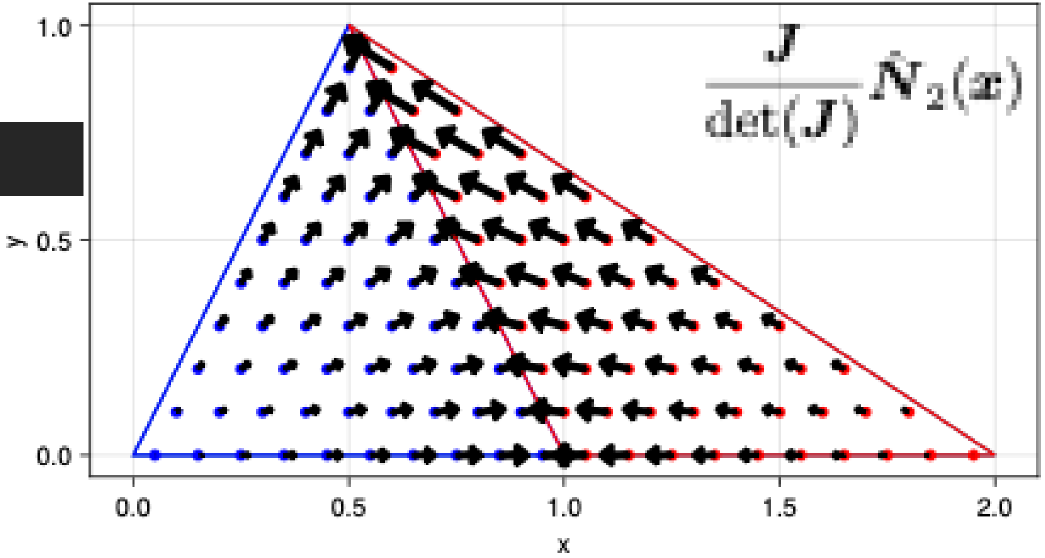
RaviartThomas{RefTriangle, 1}

```
Ferrite.ContravariantPiolaMapping()
```

$$\boldsymbol{N}(\boldsymbol{x}) = \frac{\boldsymbol{J}}{\det(\boldsymbol{J})}\hat{\boldsymbol{N}}(\boldsymbol{\xi}(\boldsymbol{x}))$$

Preserves the **normal** component!

But **sign** is wrong!




```
N = shape_value(cellvalues, q_point, shapenr)
N̂ = reference_shape_value(interpolation, local_coord, shapenr)
```

How are these implemented?

Vector interpolations, example: H(div)

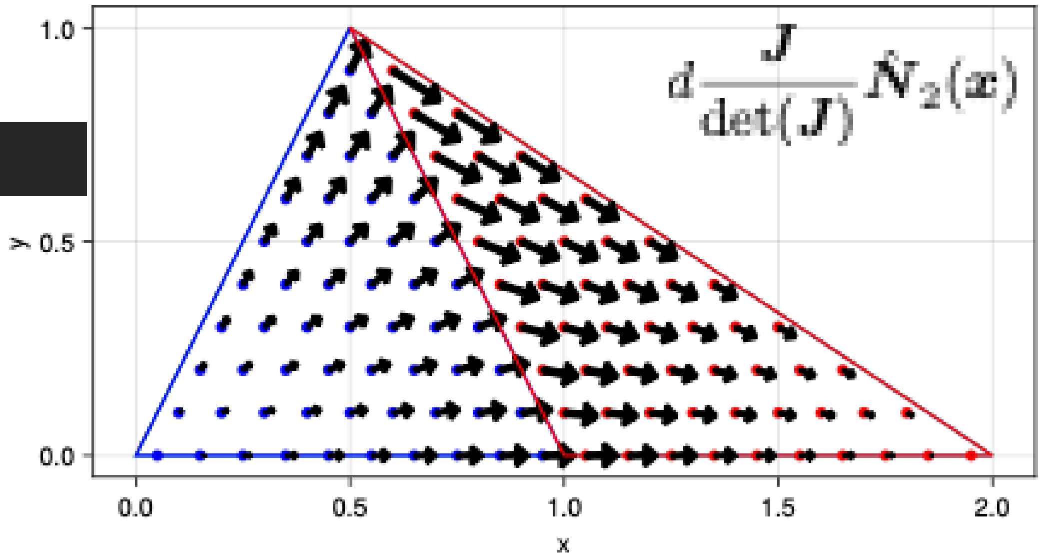
RaviartThomas{RefTriangle, 1}

```
Ferrite.ContravariantPiolaMapping()
```

$$\boldsymbol{N}(\boldsymbol{x}) = d \frac{\boldsymbol{J}}{\det(\boldsymbol{J})} \hat{\boldsymbol{N}}(\boldsymbol{\xi}(\boldsymbol{x}))$$

Introduce the facet direction:

$$d = \begin{cases} +1 & \text{positive facet} \\ -1 & \text{negative facet} \end{cases}$$

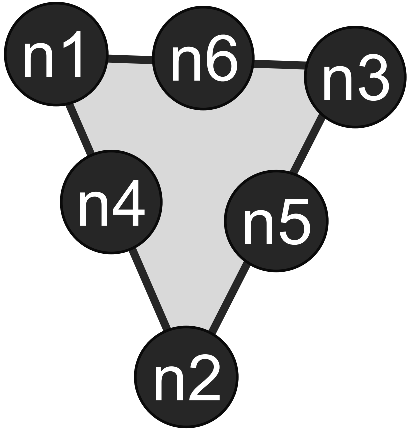



```
N = shape_value(cellvalues, q_point, shapenr)

N̂ = reference_shape_value(interpolation, local_coord, shapenr)
```

How are these implemented?

Facet direction

```
cell = QuadraticTriangle((n1, n2, n3, n4, n5, n6))

julia> facets(cell)
((n1, n2), (n2, n3), (n3, n1))

# Consider facet 2: (n2, n3)

julia> d = n3 > n2 ? +1 : -1
```



This is why the **cell** is required to **reinit!** cellvalues with non-identity mapped interpolations

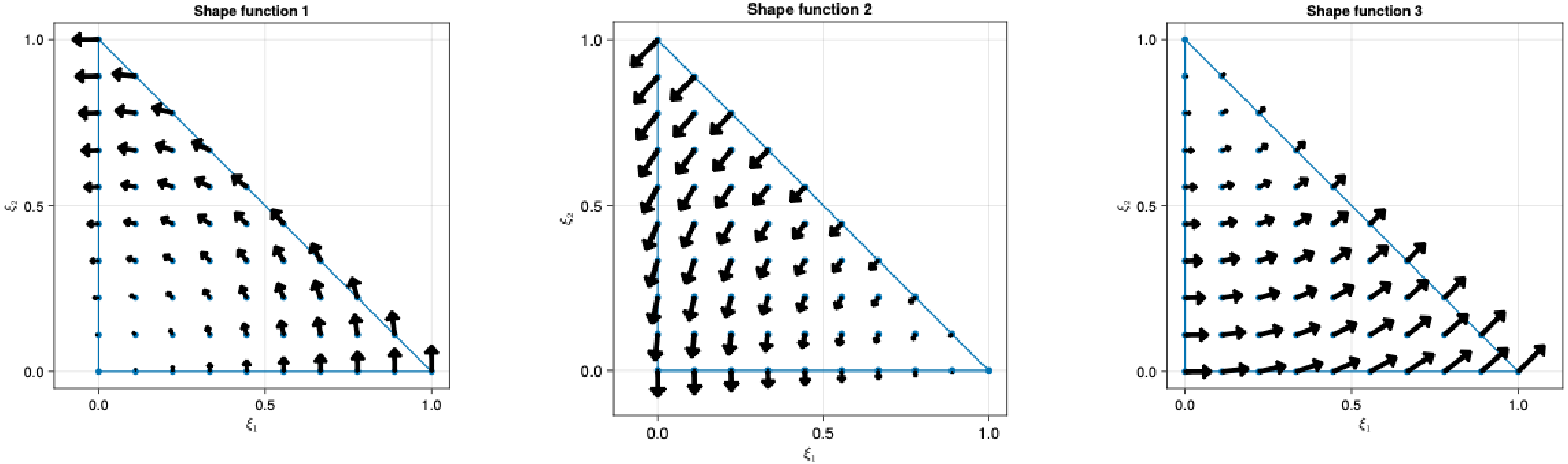
How are these implemented?

Vector interpolations, example: H(curl)

Nedelec{RefTriangle, 1}

```
Ferrite.CovariantPiolaMapping()
```

Preserves the **tangential** component!



What about Boundary Conditions?

$$\boldsymbol{N}_i(\boldsymbol{x})a_i = u_{\text{prescribed}}(\boldsymbol{x}) \text{ on } \Gamma_{\text{D}}$$

$$\boldsymbol{N}_i(\boldsymbol{x}_j) = \delta_{ij} \Rightarrow a_j = u_{\text{prescribed}}(\boldsymbol{x}_j) \text{ for all } \boldsymbol{x}_j \text{ on } \Gamma_{\text{D}}$$

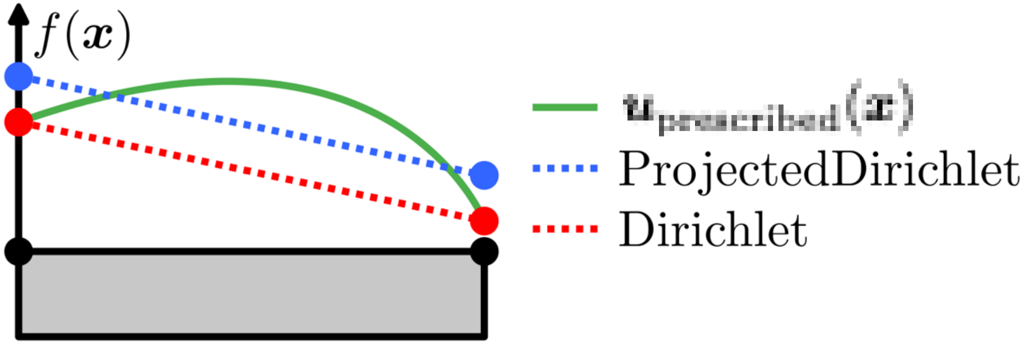
but $\boldsymbol{N}_i(\boldsymbol{x}_j) \neq \delta_{ij}$ for true vector interpolations...

$$\text{minimize} \int_{\Gamma_{\text{D}}} \left[\boldsymbol{N}_i(\boldsymbol{x})a_i - u_{\text{prescribed}}(\boldsymbol{x})\right]^2 \, \mathrm{d}\Gamma$$

```
pdbc = ProjectedDirichlet(
    field_name::Symbol,
    facets::Set{FacetIndex},
    f::Function)
```

```
ch = ConstraintHandler(dh)
add!(ch, pdbc)
close!(ch)
```

What about Boundary Conditions?



$$a_j = u_{\text{prescribed}}(\boldsymbol{x}_j)$$

$$\text{minimize} \int_{\Gamma_D} \left[N_i(\boldsymbol{x})a_i - u_{\text{prescribed}}(\boldsymbol{x})\right]^2 \, d\Gamma$$

```
pdbc = ProjectedDirichlet(
    field_name::Symbol,
    facets::Set{FacetIndex},
    u_prescribed::Function)
```

```
ch = ConstraintHandler(dh)
add!(ch, pdbc)
close!(ch)
```

Why are these implemented in Ferrite?

- Closed **Multiple cellvalues** #674
- Closed **MultiCellValues** #680
- Merged Split values into geometric mapping and function values #764
- Open **CellMultiValues** (new attempt) #872
- Merged Implement H(div) and H(curl) interpolations #1145

```
ip = Lagrange{RefTriangle, 1}()
cv = CellMultiValues(qr, ip, (u=ip^2, p=ip))
```

How should you use them?

- L2Projection (#1161)
  https://ferrite-fem.github.io/Ferrite.jl/previews/PR1161/howto/L2flux/
- Heat equation: Mixed H(div) conforming formulation (#798)
  https://ferrite-fem.github.io/Ferrite.jl/previews/PR798/tutorials/heat_equation_hdiv/
- Maxwell Discretizations: The Good, The Bad, and The Ugly (#798)
  https://ferrite-fem.github.io/Ferrite.jl/previews/PR798/tutorials/maxwell_good_bad_ugly/

Note that links are to previews and will not work in the future

Heat equation:
Mixed H(div) conforming formulation

Theory

We start with the strong form of the heat equation: Find the temperature, $u(\boldsymbol{x})$, and heat flux, $\boldsymbol{q}(x)$, such that

$$\begin{aligned}
\boldsymbol{\nabla} \cdot \boldsymbol{q} &= h(\boldsymbol{x}), \quad \text{in } \Omega \\
\boldsymbol{q}(\boldsymbol{x}) &= -k\ \boldsymbol{\nabla} u(\boldsymbol{x}), \quad \text{in } \Omega \\
\boldsymbol{q}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) &= q_n, \quad \text{on } \Gamma_\mathrm{N} \\
u(\boldsymbol{x}) &= u_\mathrm{D}, \quad \text{on } \Gamma_\mathrm{D}
\end{aligned}$$

From this strong form, we can formulate the weak form as a mixed formulation. Find $u \in \mathbb{U}$ and $\boldsymbol{q} \in \mathbb{Q}$ such that

$$\begin{aligned}
\int_\Omega \delta u [\boldsymbol{\nabla} \cdot \boldsymbol{q}]\ \mathrm{d}\Omega &= \int_\Omega \delta u h\ \mathrm{d}\Omega, \quad \forall\, \delta u \in \delta\mathbb{U} \\
\int_\Omega \boldsymbol{\delta q} \cdot \boldsymbol{q}\ \mathrm{d}\Omega - \int_\Omega [\boldsymbol{\nabla} \cdot \boldsymbol{\delta q}] k u\ \mathrm{d}\Omega &= -\int_\Gamma \boldsymbol{\delta q} \cdot \boldsymbol{n} k\ u\ \mathrm{d}\Gamma, \quad \forall\, \boldsymbol{\delta q} \in \delta\mathbb{Q}
\end{aligned}$$

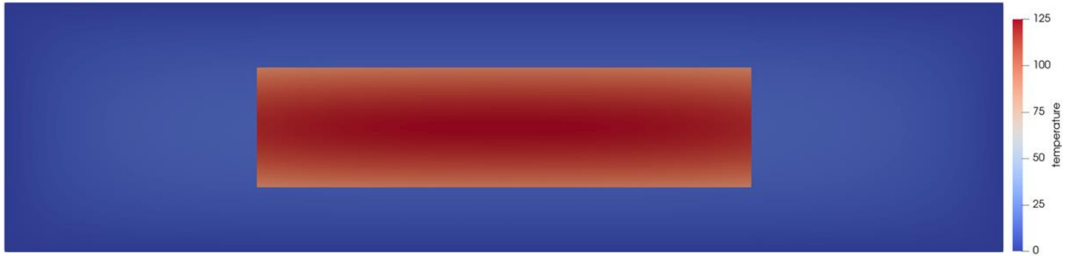
where we have the function spaces,

$$\begin{aligned}
\mathbb{U} &= \delta\mathbb{U} = L^2 \\
\mathbb{Q} &= \{\boldsymbol{q} \in H(\mathrm{div}) \text{ such that } \boldsymbol{q} \cdot \boldsymbol{n} = q_\mathrm{n} \text{ on } \Gamma_\mathrm{D}\} \\
\delta\mathbb{Q} &= \{\boldsymbol{q} \in H(\mathrm{div}) \text{ such that } \boldsymbol{q} \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_\mathrm{D}\}
\end{aligned}$$

A stable choice of finite element spaces for this problem on grid with triangles is using

- `DiscontinuousLagrange{RefTriangle, k-1}` for approximating L^2
- `BrezziDouglasMarini{RefTriangle, k}` for approximating $H(\mathrm{div})$

Heat equation: Mixed H(div) conforming formulation



- Expected since temperature is discontinuous (L2)
- Boundary flux “exact”

Theory

We start with the strong form of the heat equation: Find the temperature, $u(\boldsymbol{x})$, and heat flux, $\boldsymbol{q}(x)$, such that

$$\begin{aligned} \boldsymbol{\nabla} \cdot \boldsymbol{q} &= h(\boldsymbol{x}), \quad \text{in } \Omega \\ \boldsymbol{q}(\boldsymbol{x}) &= -k\ \boldsymbol{\nabla} u(\boldsymbol{x}), \quad \text{in } \Omega \\ \boldsymbol{q}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) &= q_n, \quad \text{on } \Gamma_\mathrm{N} \\ u(\boldsymbol{x}) &= u_\mathrm{D}, \quad \text{on } \Gamma_\mathrm{D} \end{aligned}$$

From this strong form, we can formulate the weak form as a mixed formulation. Find $u \in \mathbb{U}$ and $\boldsymbol{q} \in \mathbb{Q}$ such that

$$\begin{aligned} \int_\Omega \delta u [\boldsymbol{\nabla} \cdot \boldsymbol{q}]\ \mathrm{d}\Omega &= \int_\Omega \delta u h\ \mathrm{d}\Omega, \quad \forall\, \delta u \in \delta\mathbb{U} \\ \int_\Omega \boldsymbol{\delta q} \cdot \boldsymbol{q}\ \mathrm{d}\Omega - \int_\Omega [\boldsymbol{\nabla} \cdot \boldsymbol{\delta q}] k u\ \mathrm{d}\Omega &= -\int_\Gamma \boldsymbol{\delta q} \cdot \boldsymbol{n} k\ u\ \mathrm{d}\Gamma, \quad \forall\, \boldsymbol{\delta q} \in \delta\mathbb{Q} \end{aligned}$$

where we have the function spaces,

$$\begin{aligned} \mathbb{U} &= \delta\mathbb{U} = L^2 \\ \mathbb{Q} &= \{\boldsymbol{q} \in H(\mathrm{div}) \text{ such that } \boldsymbol{q} \cdot \boldsymbol{n} = q_\mathrm{n} \text{ on } \Gamma_\mathrm{D}\} \\ \delta\mathbb{Q} &= \{\boldsymbol{q} \in H(\mathrm{div}) \text{ such that } \boldsymbol{q} \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_\mathrm{D}\} \end{aligned}$$

A stable choice of finite element spaces for this problem on grid with triangles is using

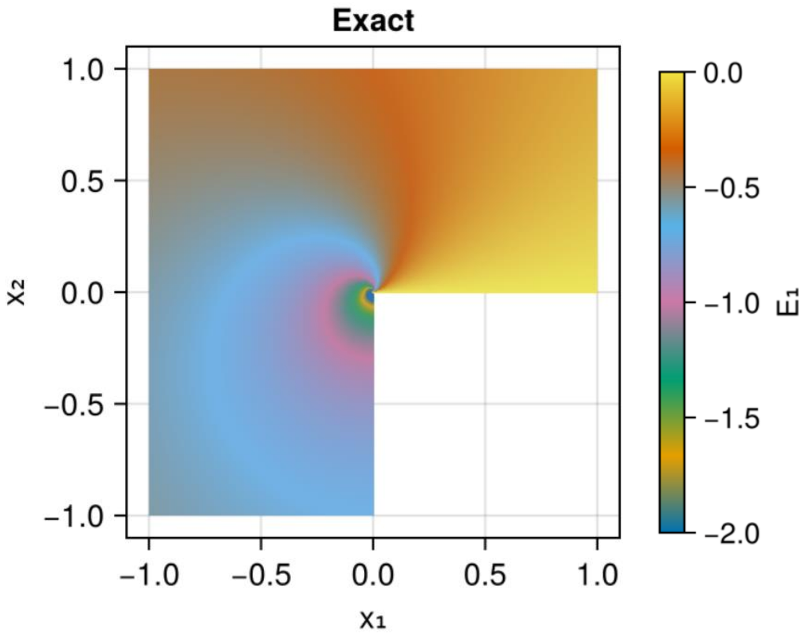
- `DiscontinuousLagrange{RefTriangle, k-1}` for approximating L^2
- `BrezziDouglasMarini{RefTriangle, k}` for approximating $H(\mathrm{div})$

Maxwell Discretizations: The Good, The Bad, and The Ugly

$$\begin{aligned} \mathrm{curl}(\mathrm{curl}(\boldsymbol{E})) &= 0 \quad \text{in } \Omega \\ \mathrm{div}(\boldsymbol{E}) &= 0 \quad \text{in } \Omega \\ \boldsymbol{E} \cdot \boldsymbol{t} &= g \quad \text{on } \Gamma \end{aligned}$$

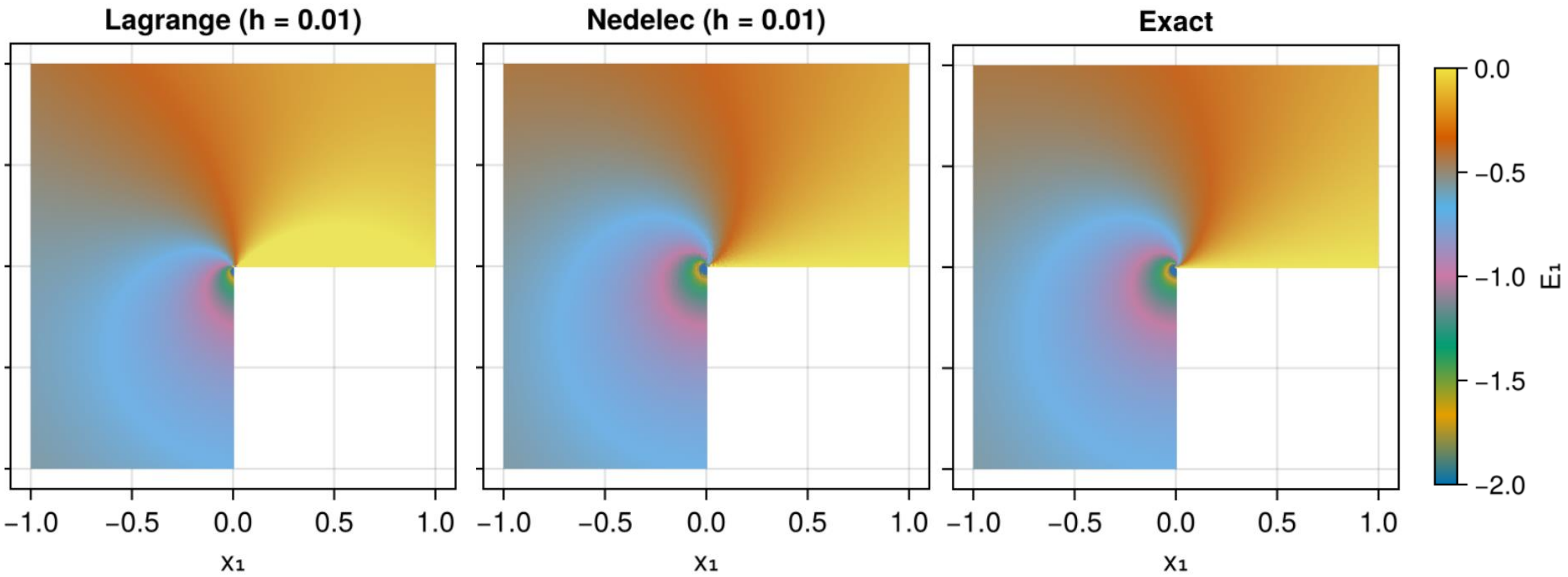
Choose analytical $\boldsymbol{E}$ and calculate g

$$\boldsymbol{E}_{\text{exact}}(\boldsymbol{x}) = \mathrm{grad}(r^{2/3} \sin(2\theta/3))$$



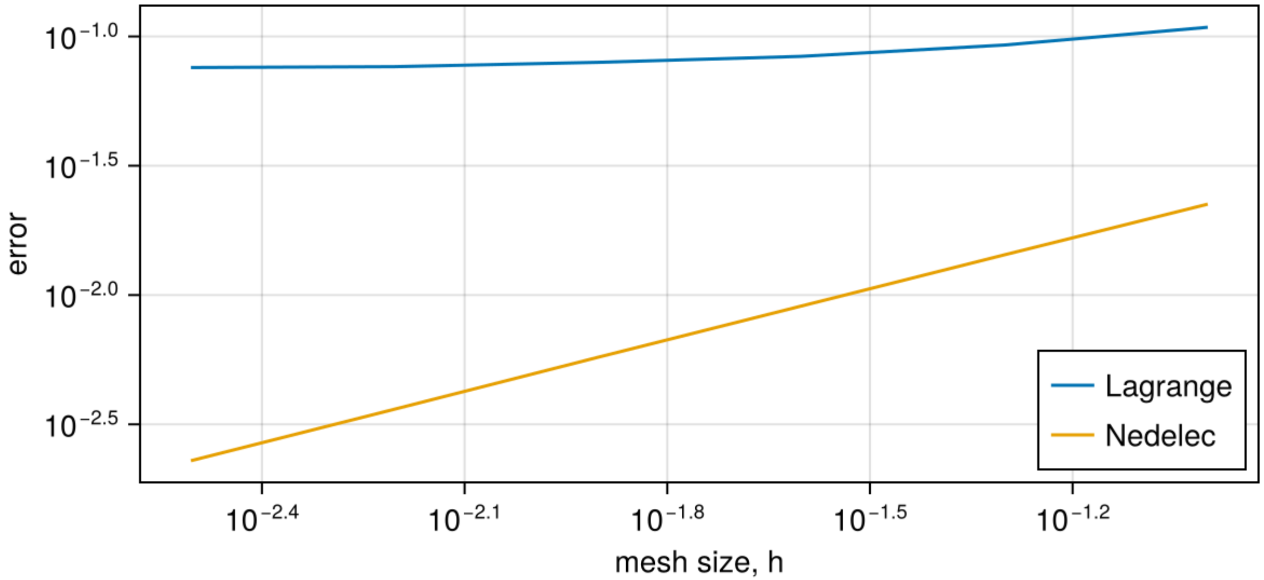
Jay Gopalakrishnan (Portland State University) Maxwell Discretizations: The Good, The Bad & The Ugly
from the graduate course MTH 653: Advanced Numerical Analysis (Spring 2019)

Maxwell Discretizations: The Good, The Bad, and The Ugly



Jay Gopalakrishnan (Portland State University) Maxwell Discretizations: The Good, The Bad & The Ugly from the graduate course MTH 653: Advanced Numerical Analysis (Spring 2019)

Maxwell Discretizations: The Good, The Bad, and The Ugly



Jay Gopalakrishnan (Portland State University) Maxwell Discretizations: The Good, The Bad & The Ugly from the graduate course MTH 653: Advanced Numerical Analysis (Spring 2019)

Ferrite now has true vector interpolations
Conclusions

What to use it for?

- Useful if you work with Maxwell's equations – H(curl)
- Applications for dual/mixed formulations with fluxes
- Can be useful for correct postprocessing

What's next?

- Some missing implementations in 3D for H(curl) (issues/prs are open)
- True tensor-valued interpolations (Double[Contra/Co]variantPiolaMapping)
- Support in IGA.jl (@Basavesh)
- CellMultiValues ☺

AVANCEZ

1829

CHALMERS

UNIVERSITY OF TECHNOLOGY