Two approaches to finite element tensor calculus on surfaces

Yakov Berchenko-Kogan, joint with Evan Gawlik

Florida Institute of Technology
Supported by NSF DMS-2411209

May 21, 2026

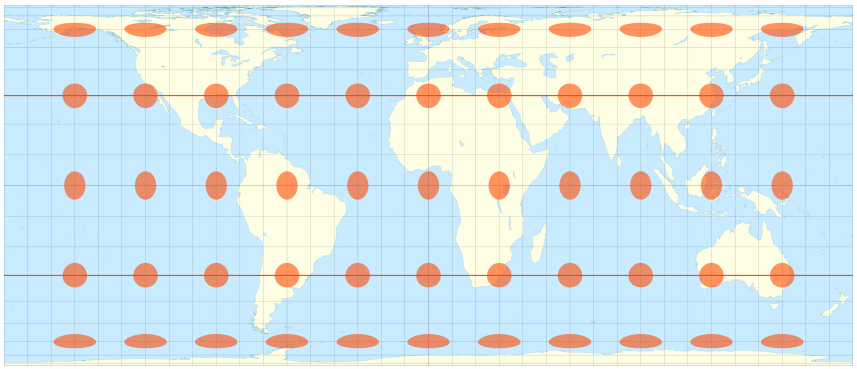
Section 1

Blow-up finite elements

Surfaces: extrinsic vs. intrinsic

Extrinsic

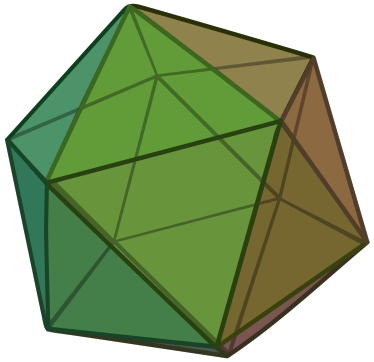
Intrinsic



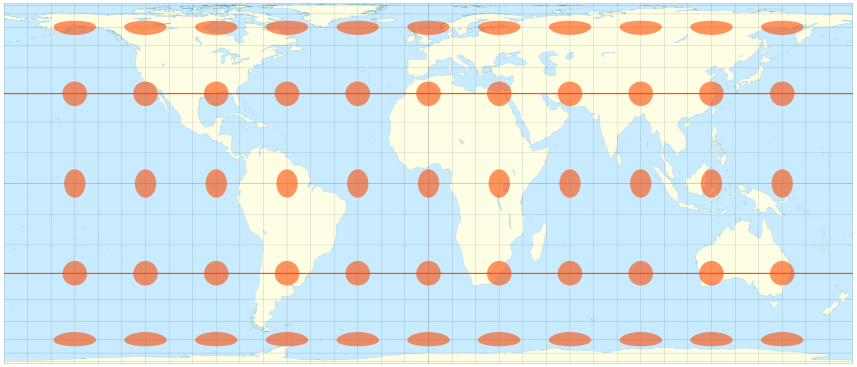
Four images from Wikipedia

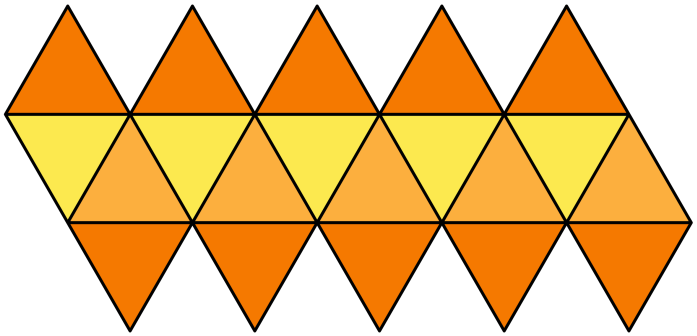
Surfaces: extrinsic vs. intrinsic

Extrinsic



Intrinsic

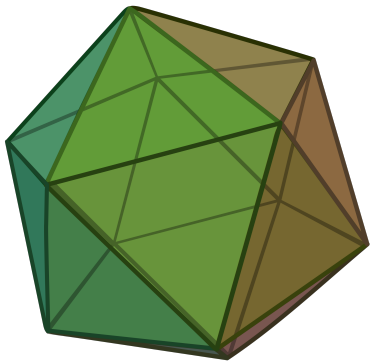




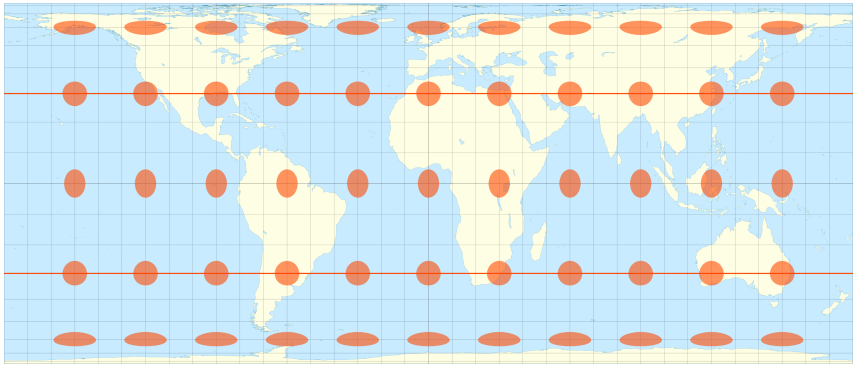
Four images from Wikipedia

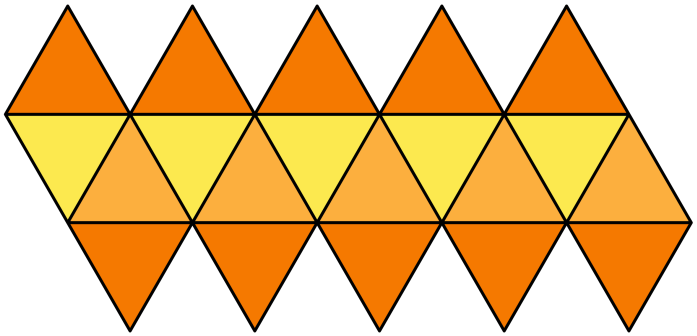
Surfaces: extrinsic vs. intrinsic

Extrinsic



Intrinsic





Why compute intrinsically?

- Intrinsic problems, e.g. numerical relativity, Ricci flow.
- Structure preservation: independence of embedding.

Four images from Wikipedia

Discretizing continuous tangent vector fields on surfaces is "impossible"

Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Discretizing continuous tangent vector fields on surfaces is "impossible"

Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

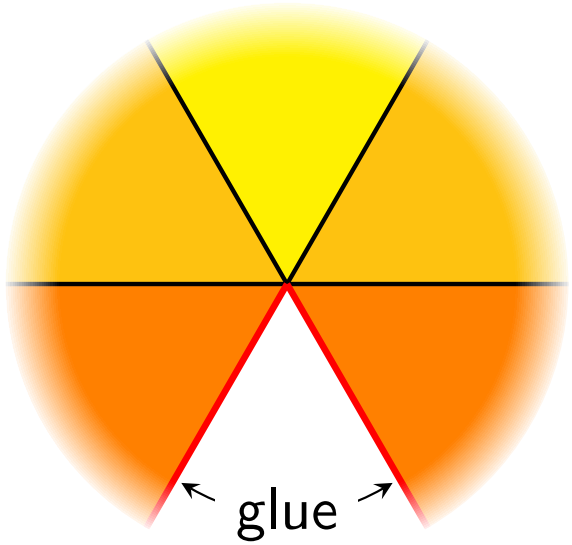
Discretizing continuous tangent vector fields on surfaces is "impossible"

Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?



Discretizing continuous tangent vector fields on surfaces is "impossible"

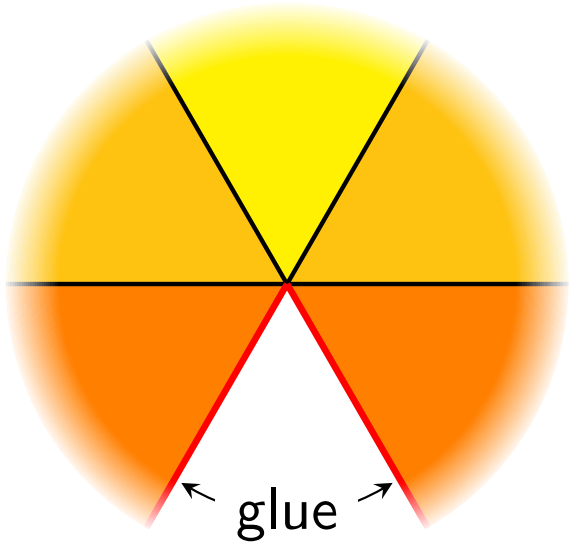
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

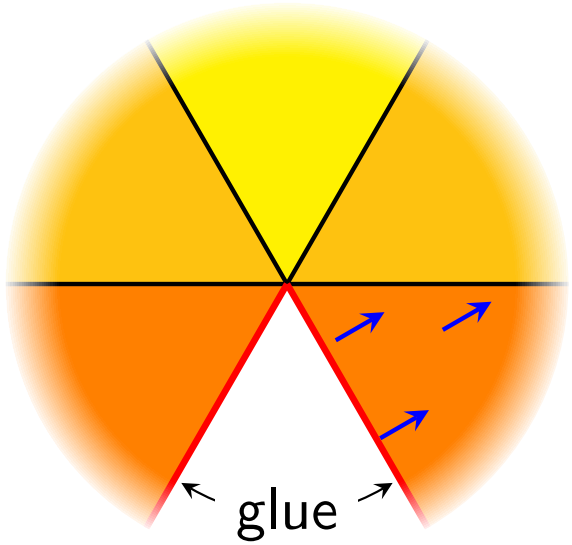
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

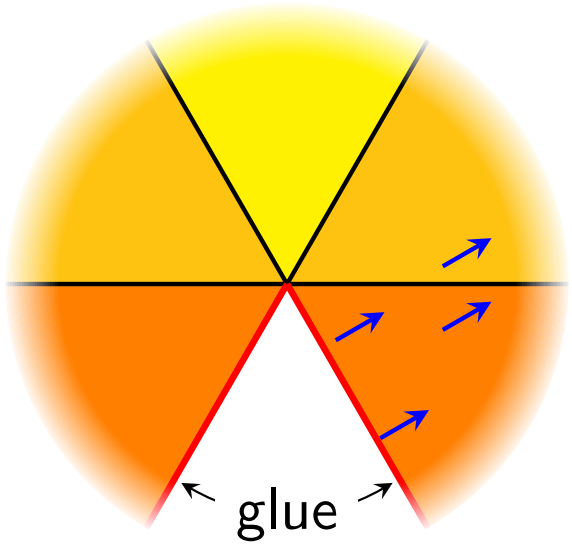
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

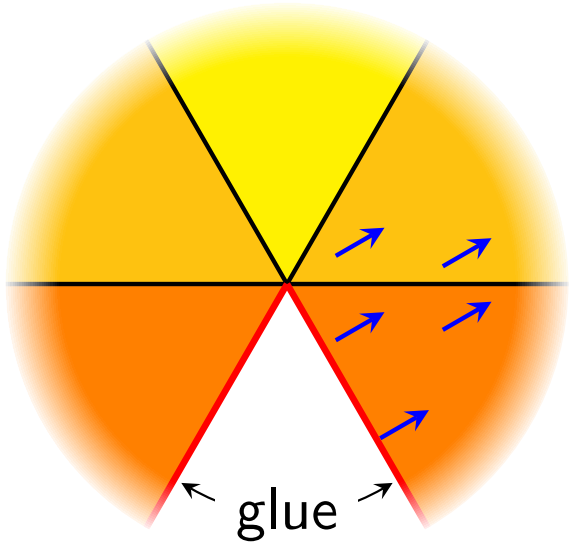
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

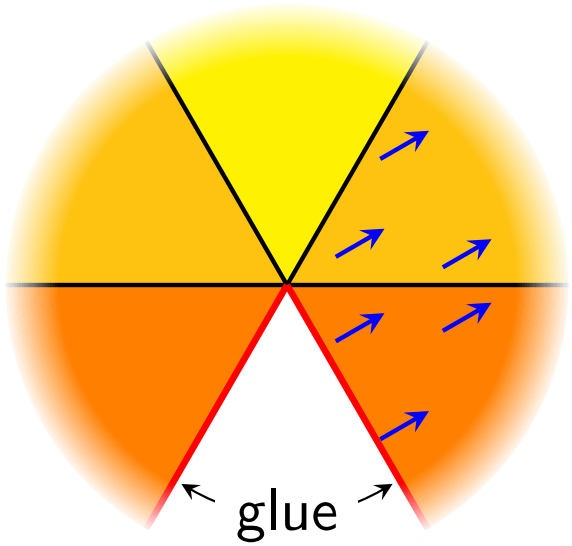
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

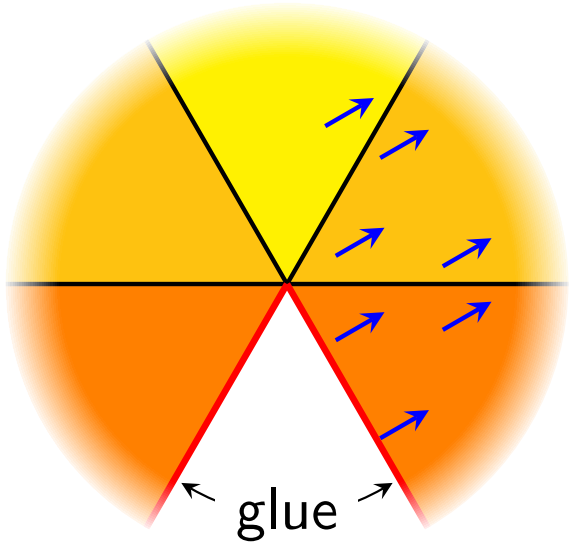
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

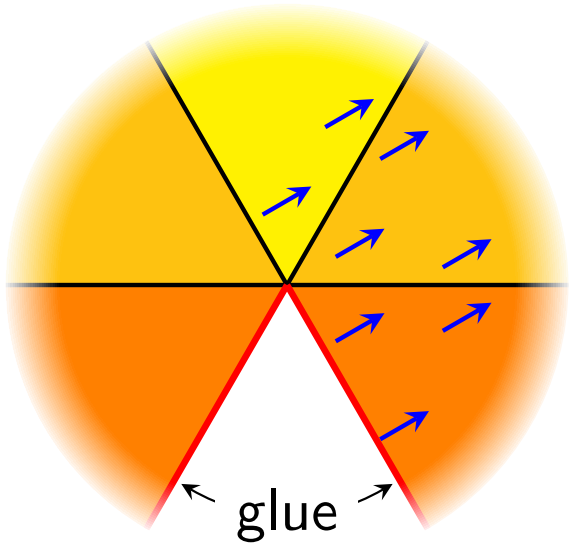
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

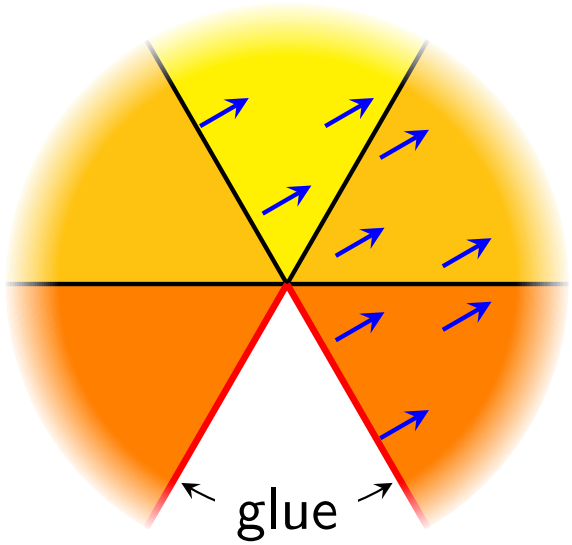
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

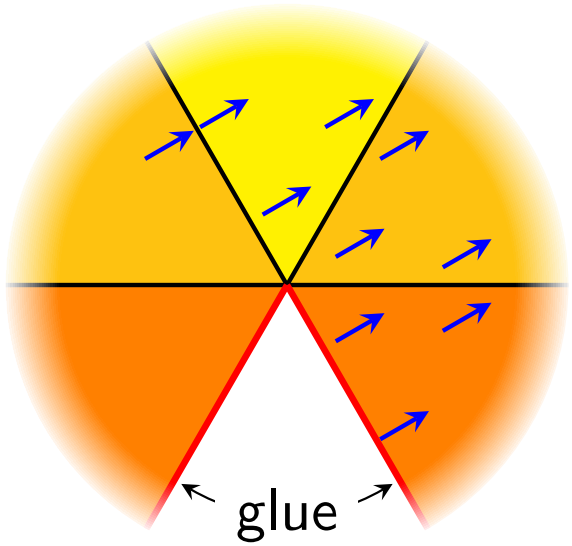
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

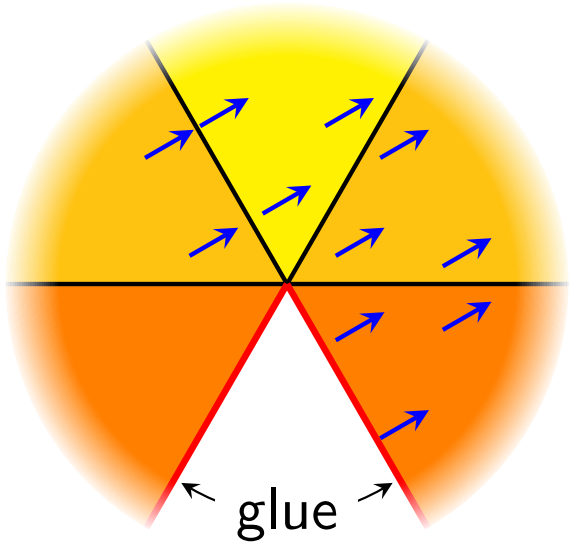
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

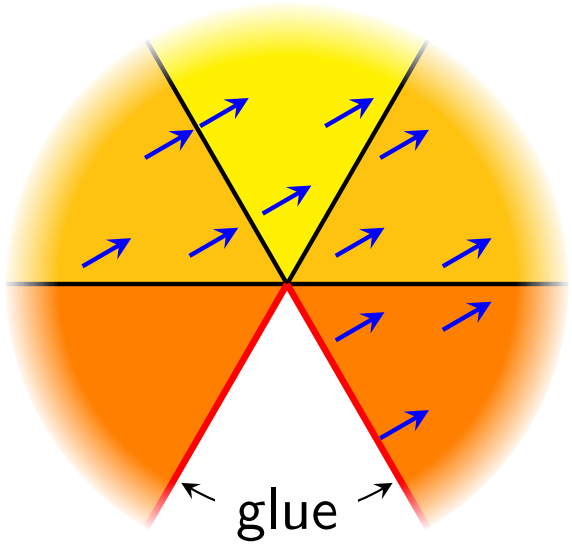
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

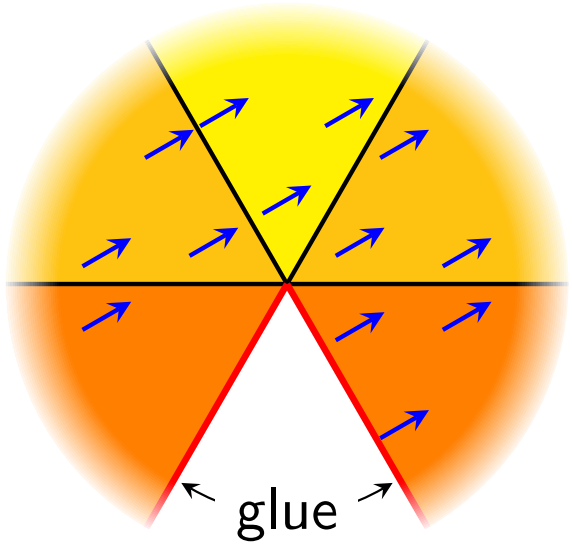
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

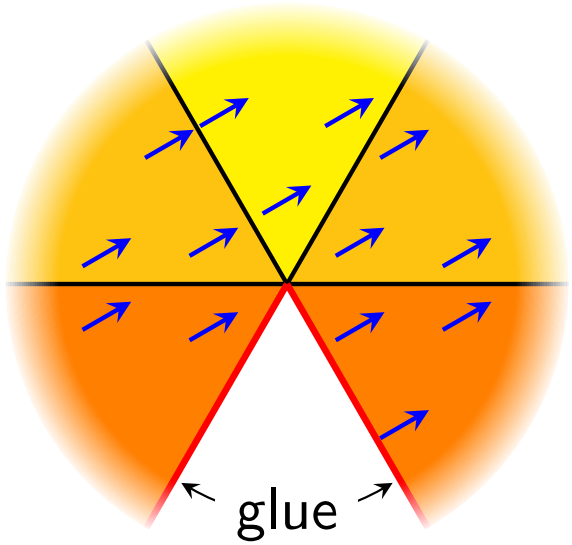
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

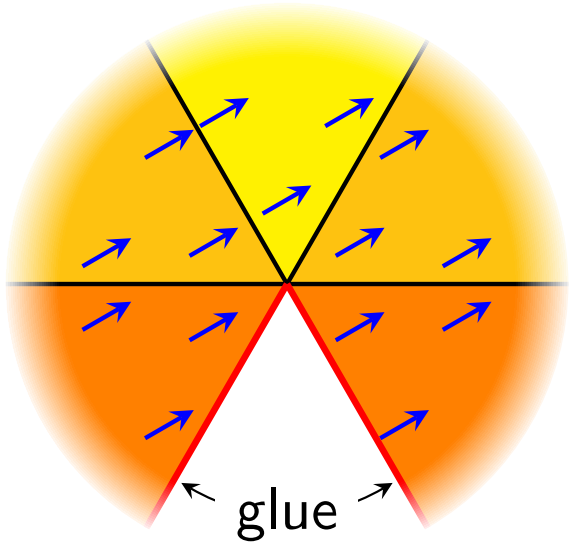
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



Discretizing continuous tangent vector fields on surfaces is "impossible"

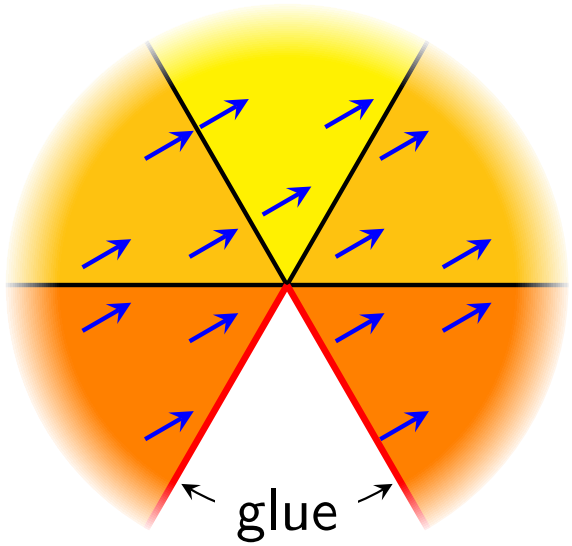
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



discontinuous across red edge

Discretizing continuous tangent vector fields on surfaces is "impossible"

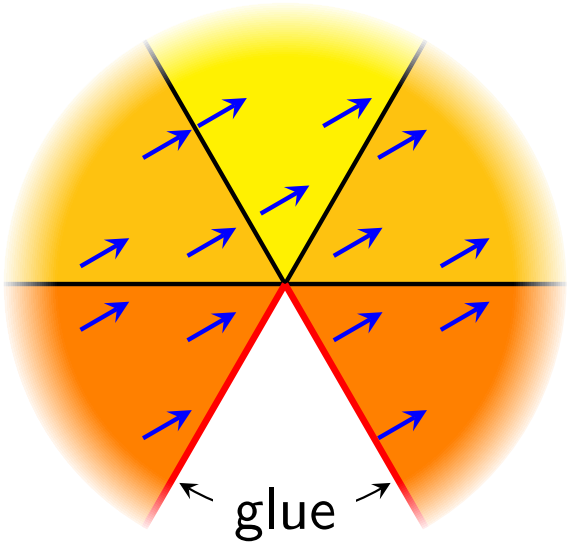
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

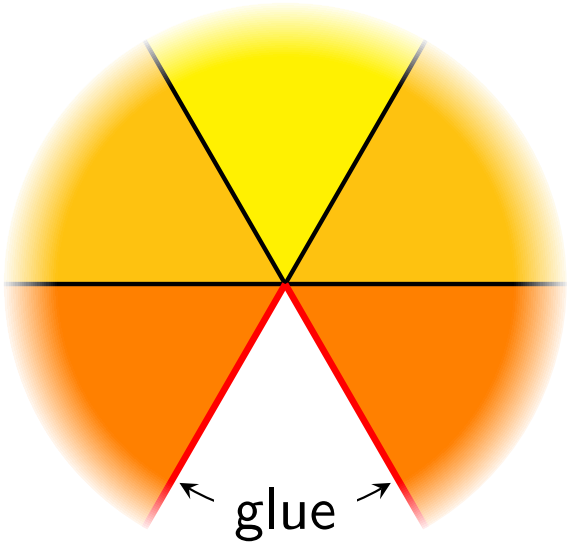
Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



blow-up elements



discontinuous across red edge

Discretizing continuous tangent vector fields on surfaces is "impossible"

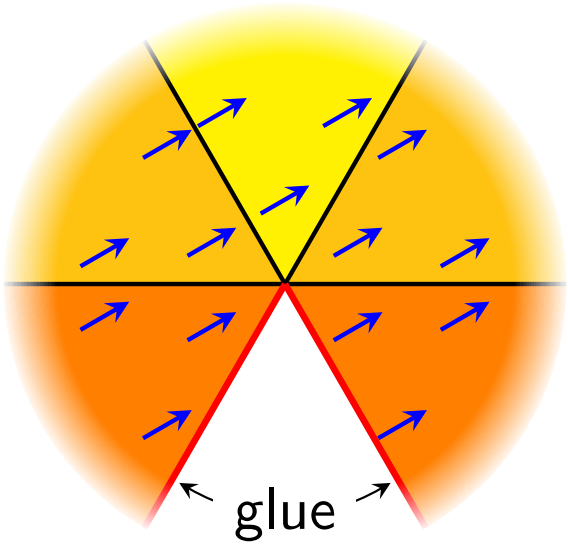
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

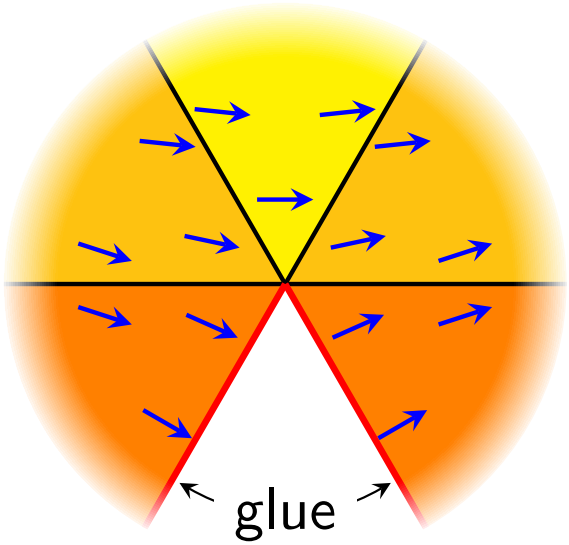
Let's try!

- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



blow-up elements



discontinuous across red edge

Discretizing continuous tangent vector fields on surfaces is "impossible"

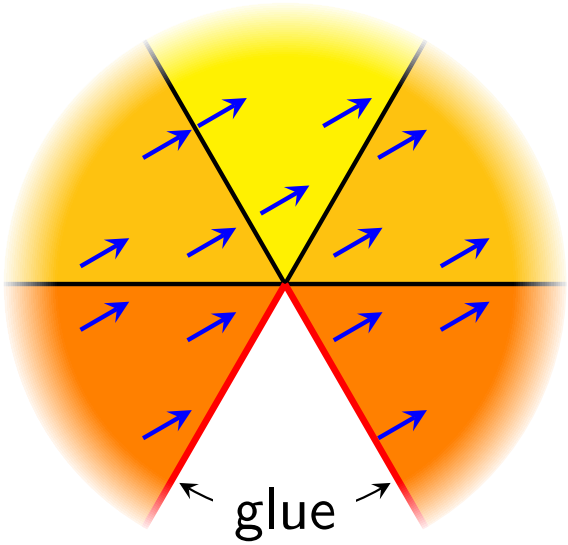
Vector field requirements

1. Tangent to the surface.
2. No jump across edges.
3. Continuous on each triangle.

Let's try!

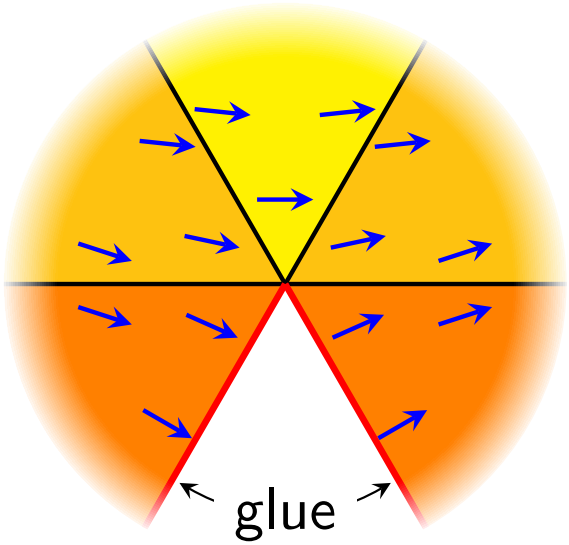
- Zoom in on a vertex of the icosahedron.
- What do we see?

continuous elements



discontinuous across red edge

blow-up elements



continuous across all edges

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).
    - Intrinsic via FEEC (differential forms) formulation.

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).
    - Intrinsic via FEEC (differential forms) formulation.
    - In this context, edge jumps are actually good (full continuity is bad), see (Arnold, Falk, Winther, 2010).

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).
    - Intrinsic via FEEC (differential forms) formulation.
    - In this context, edge jumps are actually good (full continuity is bad), see (Arnold, Falk, Winther, 2010).
  - In contrast, for problems like fluid flow (H^1), edge jumps are bad.

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).
    - Intrinsic via FEEC (differential forms) formulation.
    - In this context, edge jumps are actually good (full continuity is bad), see (Arnold, Falk, Winther, 2010).
  - In contrast, for problems like fluid flow (H^1), edge jumps are bad.
    - One approach: Use FEEC anyways and penalize jump across edges.

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).
    - Intrinsic via FEEC (differential forms) formulation.
    - In this context, edge jumps are actually good (full continuity is bad), see (Arnold, Falk, Winther, 2010).
  - In contrast, for problems like fluid flow (H^1), edge jumps are bad.
    - One approach: Use FEEC anyways and penalize jump across edges.
    - Demlow, Neilan approach: At each vertex, project to a plane (no angle defect), and enforce continuity in that plane.

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).
    - Intrinsic via FEEC (differential forms) formulation.
    - In this context, edge jumps are actually good (full continuity is bad), see (Arnold, Falk, Winther, 2010).
  - In contrast, for problems like fluid flow (H^1), edge jumps are bad.
    - One approach: Use FEEC anyways and penalize jump across edges.
    - Demlow, Neilan approach: At each vertex, project to a plane (no angle defect), and enforce continuity in that plane.
- Break continuity at vertices: Blow-up elements (—, Gawlik).

Approaches to discretizing surface vector fields

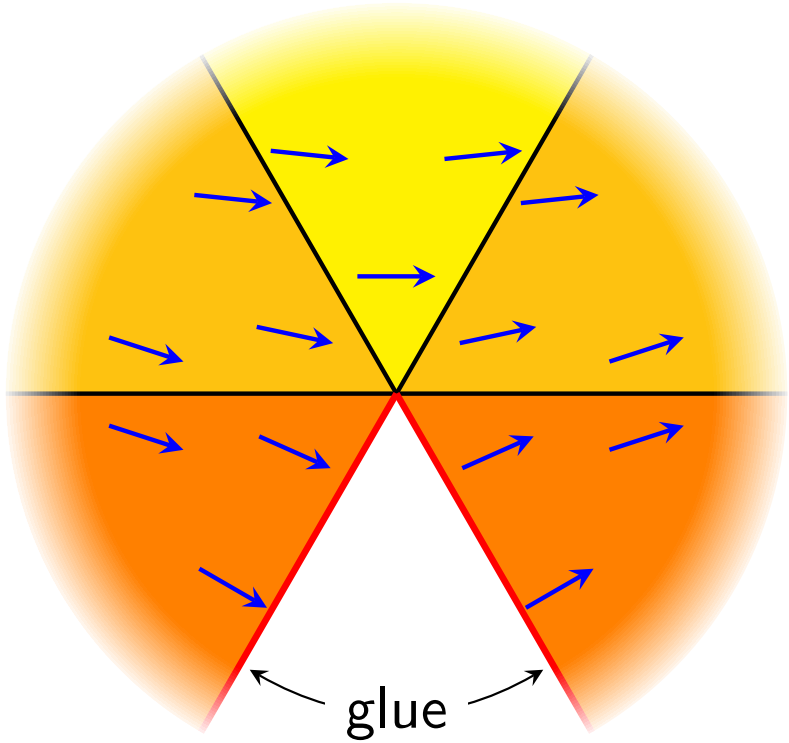
- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).
    - Intrinsic via FEEC (differential forms) formulation.
    - In this context, edge jumps are actually good (full continuity is bad), see (Arnold, Falk, Winther, 2010).
  - In contrast, for problems like fluid flow (H^1), edge jumps are bad.
    - One approach: Use FEEC anyways and penalize jump across edges.
    - Demlow, Neilan approach: At each vertex, project to a plane (no angle defect), and enforce continuity in that plane.
- Break continuity at vertices: Blow-up elements (—, Gawlik).
  - Intrinsic.

Approaches to discretizing surface vector fields

- Break tangentiality to surface: Use vector fields in ambient space, penalize component normal to surface.
  - Unnecessary data (normal component).
  - Extrinsic.
- Break continuity across edges:
  - For problems like electromagnetism ($H(\mathrm{curl})$ (or $H(\mathrm{div})$)), use FEEC (BDM, RT) elements.
    - Jump in component normal to edge but not in component tangent to edge (or vice versa).
    - Intrinsic via FEEC (differential forms) formulation.
    - In this context, edge jumps are actually good (full continuity is bad), see (Arnold, Falk, Winther, 2010).
  - In contrast, for problems like fluid flow (H^1), edge jumps are bad.
    - One approach: Use FEEC anyways and penalize jump across edges.
    - Demlow, Neilan approach: At each vertex, project to a plane (no angle defect), and enforce continuity in that plane.
- Break continuity at vertices: Blow-up elements (—, Gawlik).
  - Intrinsic.
  - Discontinuities are *only* where the problem is (vertices with angle defect).

Break continuity at vertices

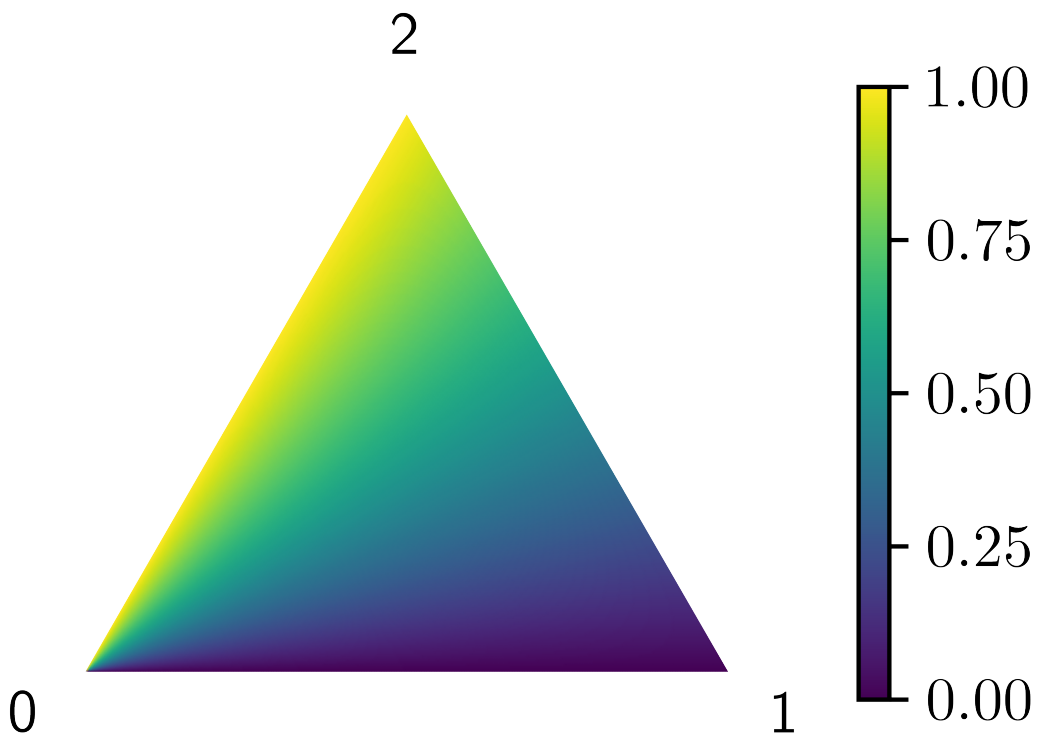
blow-up elements



continuous across all edges

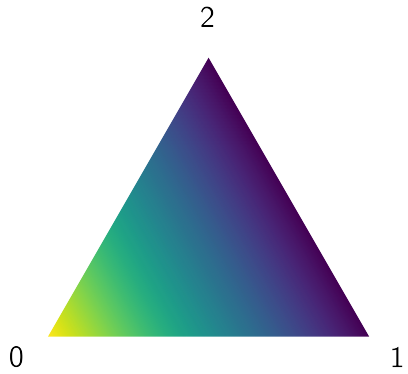
discontinuous at vertices

A simplicial analogue of the angular coordinate

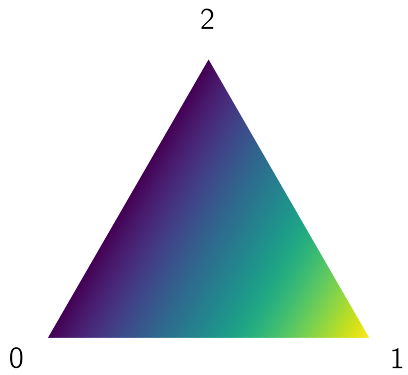


$$\frac{\lambda_2}{\lambda_1 + \lambda_2}$$

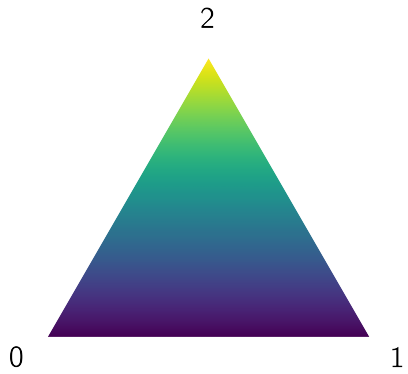
Lagrange $\mathcal{P}_1$ shape functions



λ_0

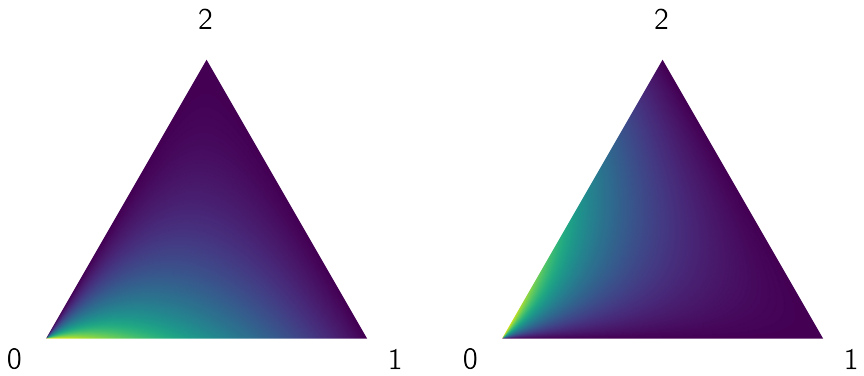


λ_1

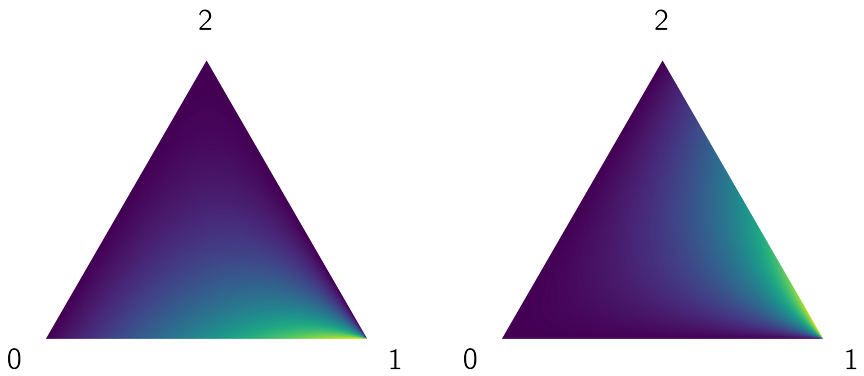


λ_2

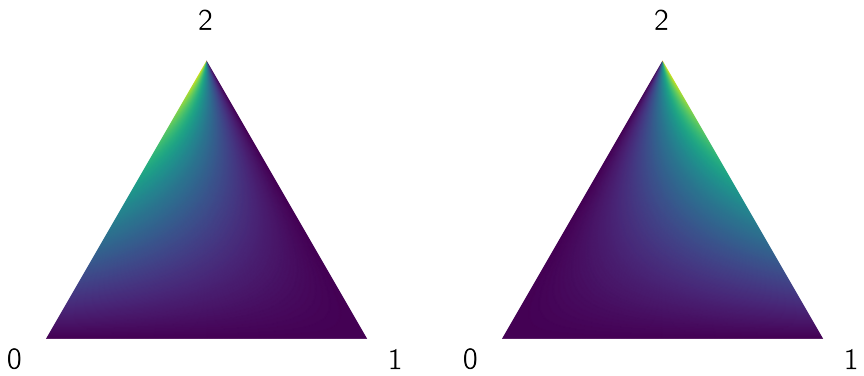
Blow-up $b\mathcal{P}_1$ shape functions



$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2}, \quad \psi_{021} = \frac{\lambda_0\lambda_2}{\lambda_2+\lambda_1},$$



$$\psi_{102} = \frac{\lambda_1\lambda_0}{\lambda_0+\lambda_2}, \quad \psi_{120} = \frac{\lambda_1\lambda_2}{\lambda_2+\lambda_0},$$



$$\psi_{201} = \frac{\lambda_2\lambda_0}{\lambda_0+\lambda_1}, \quad \psi_{210} = \frac{\lambda_2\lambda_1}{\lambda_1+\lambda_0}.$$

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}\cdot\frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{\lambda_2}{\lambda_2}.$$

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}\cdot\frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{\lambda_2}{\lambda_2}.$$

Earlier appearances

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}\cdot\frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{\lambda_2}{\lambda_2}.$$

Earlier appearances

- Geometric invariants (Chen, 1957).

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}\cdot\frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{\lambda_2}{\lambda_2}.$$

Earlier appearances

- Geometric invariants (Chen, 1957).
- Horse betting (Harville, 1973).

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}\cdot\frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{\lambda_2}{\lambda_2}.$$

Earlier appearances

- Geometric invariants (Chen, 1957).
- Horse betting (Harville, 1973).
- Intersection homology (Brasselet, Goresky, MacPherson, 1991; Bendiffalah, 1995).

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}\cdot\frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{\lambda_2}{\lambda_2}.$$

Earlier appearances

- Geometric invariants (Chen, 1957).
- Horse betting (Harville, 1973).
- Intersection homology (Brasselet, Goresky, MacPherson, 1991; Bendiffalah, 1995).

Poisson processes (—, Gawlik)

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}\cdot\frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{\lambda_2}{\lambda_2}.$$

Earlier appearances

- Geometric invariants (Chen, 1957).
- Horse betting (Harville, 1973).
- Intersection homology (Brasselet, Goresky, MacPherson, 1991; Bendiffalah, 1995).

Poisson processes (—, Gawlik)

- Three radiation sources with rates λ_0, λ_1, and λ_2.

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2} \cdot \frac{\lambda_1}{\lambda_1+\lambda_2} \cdot \frac{\lambda_2}{\lambda_2}.$$

Earlier appearances

- Geometric invariants (Chen, 1957).
- Horse betting (Harville, 1973).
- Intersection homology (Brasselet, Goresky, MacPherson, 1991; Bendiffalah, 1995).

Poisson processes (—, Gawlik)

- Three radiation sources with rates λ_0, λ_1, and λ_2.
- Let t_0, t_1, t_2 be the times when the respective radiation sources produce their first particle.

New?

Shape function

$$\psi_{012} = \frac{\lambda_0\lambda_1}{\lambda_1+\lambda_2} = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}\cdot\frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{\lambda_2}{\lambda_2}.$$

Earlier appearances

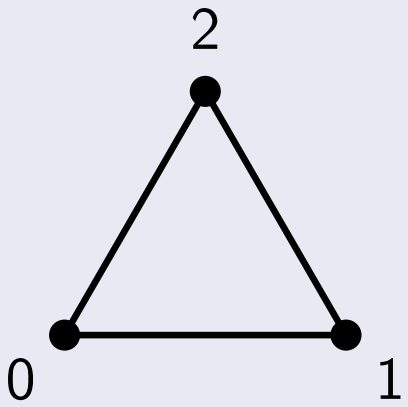
- Geometric invariants (Chen, 1957).
- Horse betting (Harville, 1973).
- Intersection homology (Brasselet, Goresky, MacPherson, 1991; Bendiffalah, 1995).

Poisson processes (—, Gawlik)

- Three radiation sources with rates λ_0, λ_1, and λ_2.
- Let t_0, t_1, t_2 be the times when the respective radiation sources produce their first particle.
- ψ_{012} is the probability that $t_0 \leq t_1 \leq t_2$.

Degrees of freedom

Classical Lagrange $\mathcal{P}_1$

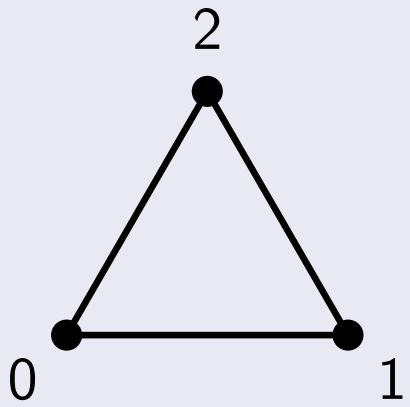


Barycentric coordinates: $\lambda_0 + \lambda_1 + \lambda_2 = 1$.

- $0 : \lambda_0 = 1 \Leftrightarrow \lambda_1 = \lambda_2 = 0$
- $1 : \lambda_1 = 1 \Leftrightarrow \lambda_2 = \lambda_0 = 0$
- $2 : \lambda_2 = 1 \Leftrightarrow \lambda_0 = \lambda_1 = 0$

Degrees of freedom

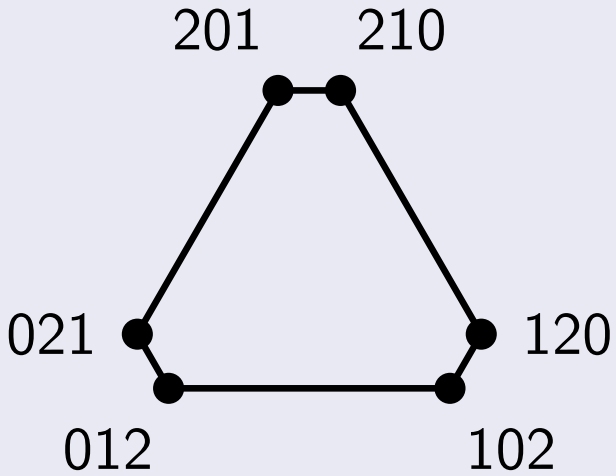
Classical Lagrange $\mathcal{P}_1$



Barycentric coordinates: $\lambda_0 + \lambda_1 + \lambda_2 = 1$.

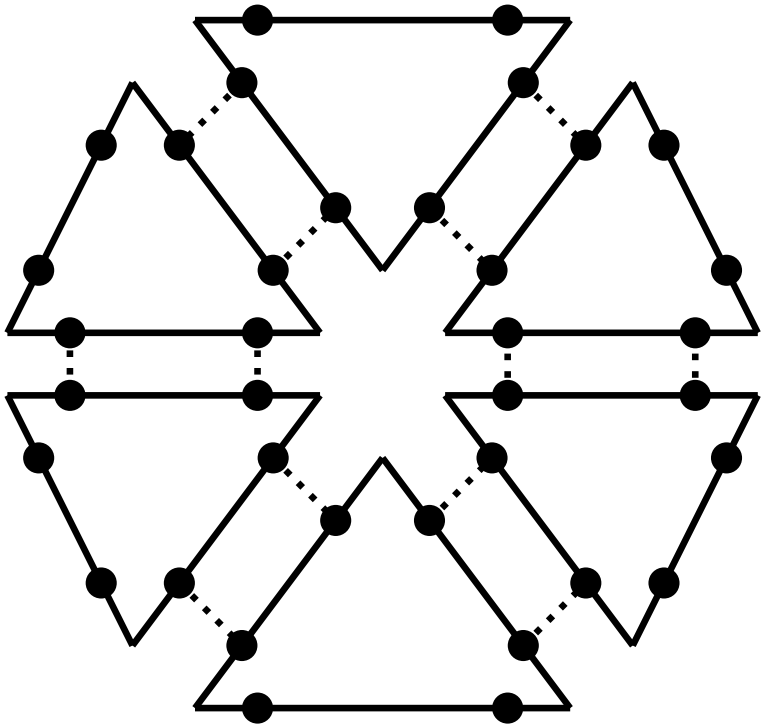
- $0 : \lambda_0 = 1 \Leftrightarrow \lambda_1 = \lambda_2 = 0$
- $1 : \lambda_1 = 1 \Leftrightarrow \lambda_2 = \lambda_0 = 0$
- $2 : \lambda_2 = 1 \Leftrightarrow \lambda_0 = \lambda_1 = 0$

Blow-up $b\mathcal{P}_1$



- $012 : \lim_{\lambda_1 \to 0} \lim_{\lambda_2 \to 0}$
- $120 : \lim_{\lambda_2 \to 0} \lim_{\lambda_0 \to 0}$
- $201 : \lim_{\lambda_0 \to 0} \lim_{\lambda_1 \to 0}$
- $021 : \lim_{\lambda_2 \to 0} \lim_{\lambda_1 \to 0}$
- $102 : \lim_{\lambda_0 \to 0} \lim_{\lambda_2 \to 0}$
- $210 : \lim_{\lambda_1 \to 0} \lim_{\lambda_0 \to 0}$

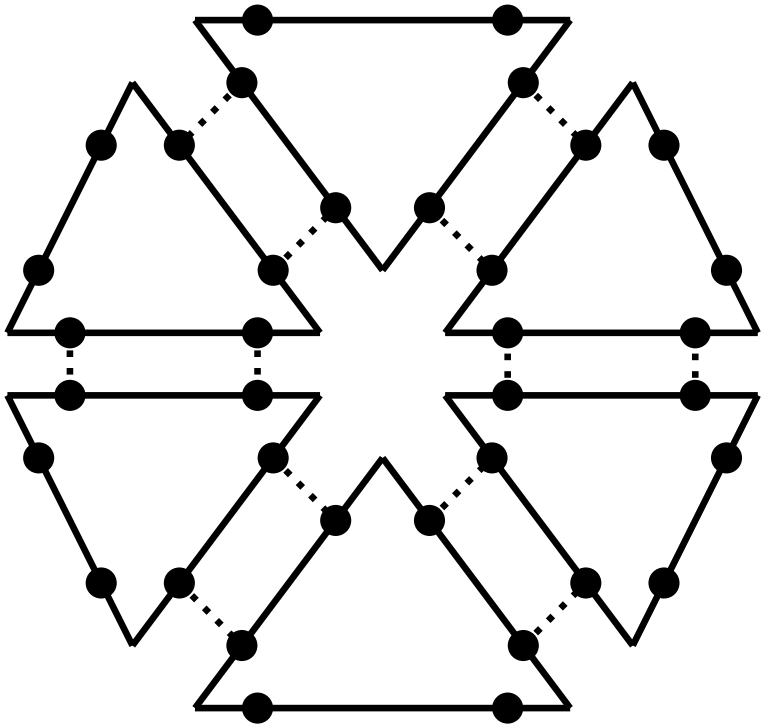
Global spaces



Blow-up finite elements

- Scalar fields: we place a number at each dot.

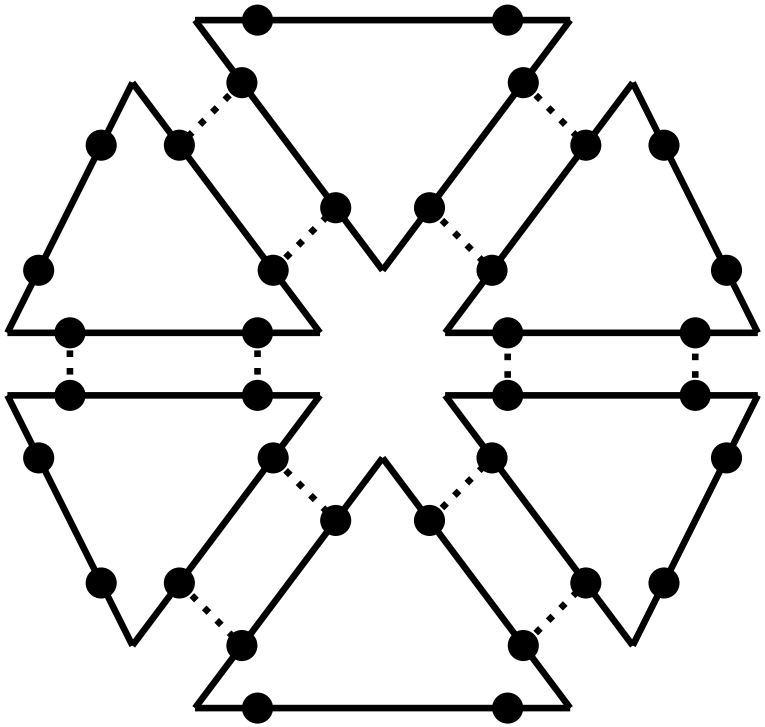
Global spaces



Blow-up finite elements

- Scalar fields: we place a number at each dot.
- Vector fields: we place two numbers at each dot, for the tangential and normal components, respectively.

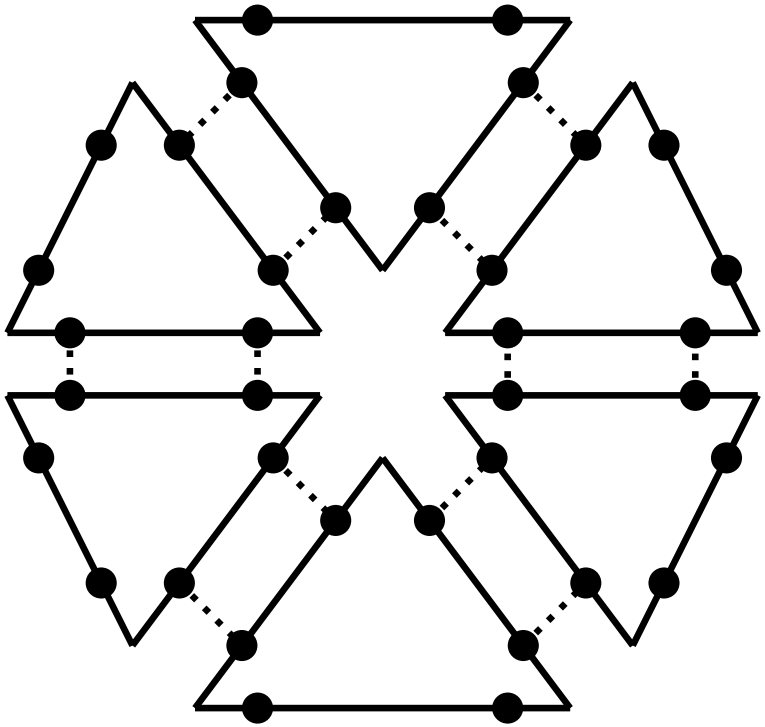
Global spaces



Blow-up finite elements

- Scalar fields: we place a number at each dot.
- Vector fields: we place two numbers at each dot, for the tangential and normal components, respectively.
  - Enforce continuity for both components, yielding full continuity across edges.

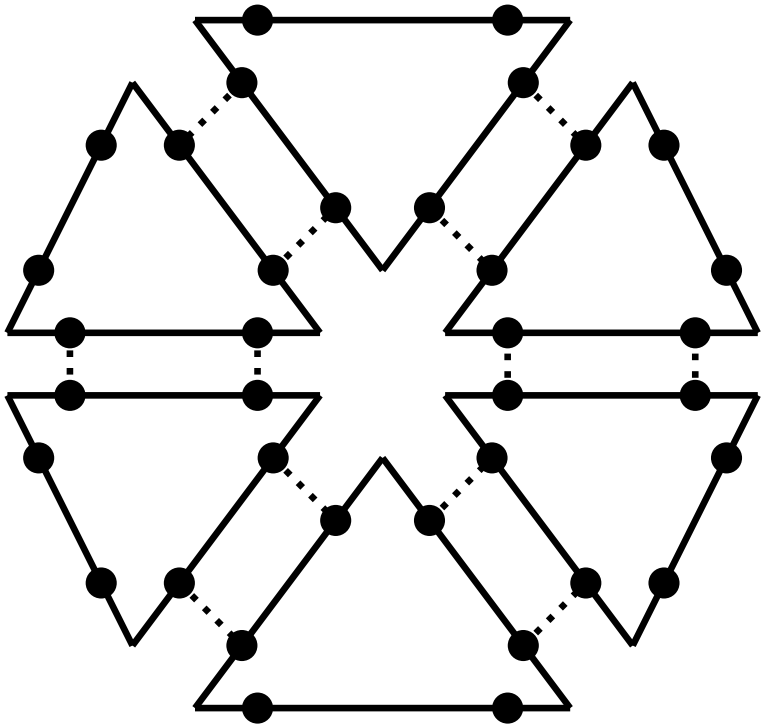
Global spaces



Blow-up finite elements

- Scalar fields: we place a number at each dot.
- Vector fields: we place two numbers at each dot, for the tangential and normal components, respectively.
  - Enforce continuity for both components, yielding full continuity across edges.
- Matrix fields: At each dot, we record the tangential–tangential component, the tangential–normal component, etc.

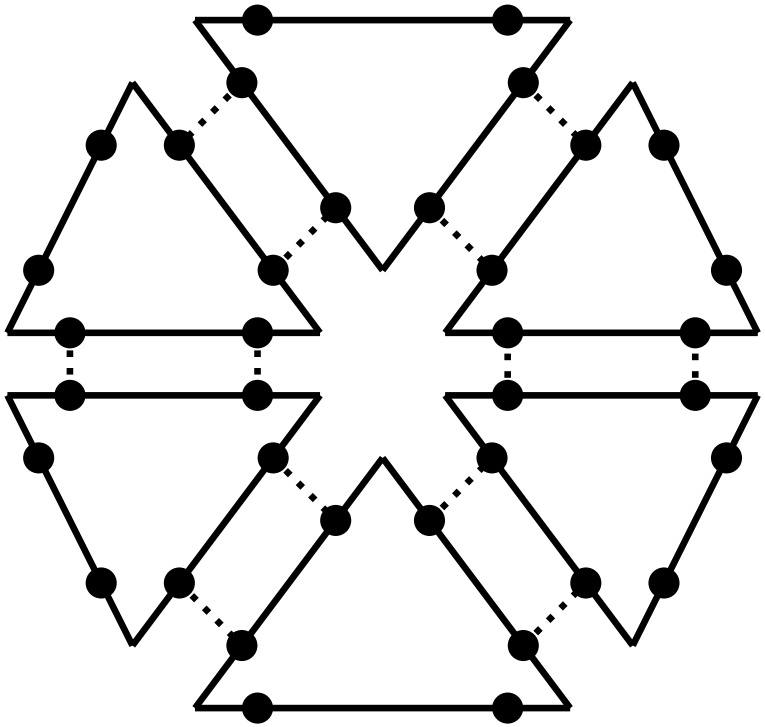
Global spaces



Blow-up finite elements

- Scalar fields: we place a number at each dot.
- Vector fields: we place two numbers at each dot, for the tangential and normal components, respectively.
  - Enforce continuity for both components, yielding full continuity across edges.
- Matrix fields: At each dot, we record the tangential–tangential component, the tangential–normal component, etc.
  - Can impose conditions on the components such as symmetry, trace-free, etc.

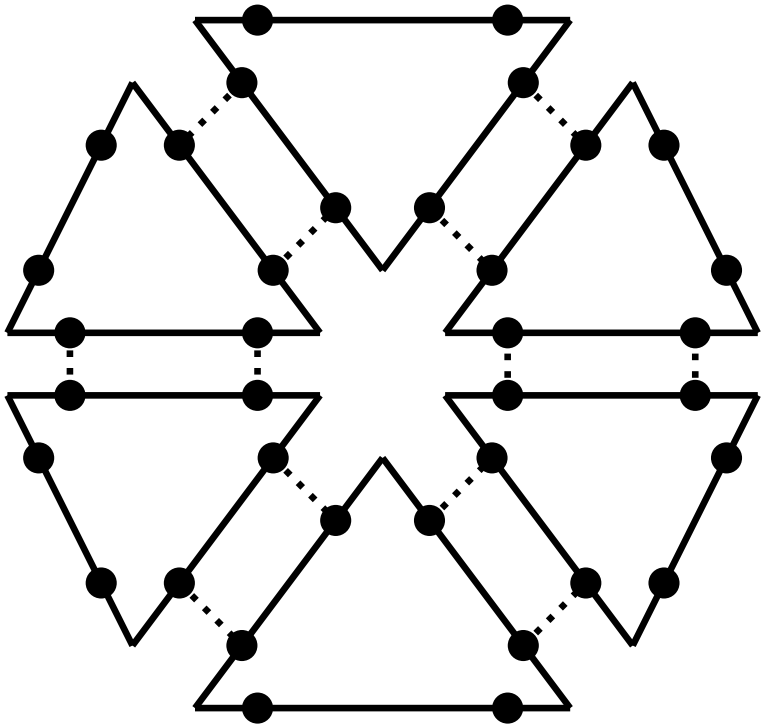
Global spaces



Blow-up finite elements

- Scalar fields: we place a number at each dot.
- Vector fields: we place two numbers at each dot, for the tangential and normal components, respectively.
  - Enforce continuity for both components, yielding full continuity across edges.
- Matrix fields: At each dot, we record the tangential–tangential component, the tangential–normal component, etc.
  - Can impose conditions on the components such as symmetry, trace-free, etc.
  - Can enforce continuity for all components or just some of them.

Global spaces



Blow-up finite elements

- Scalar fields: we place a number at each dot.
- Vector fields: we place two numbers at each dot, for the tangential and normal components, respectively.
  - Enforce continuity for both components, yielding full continuity across edges.
- Matrix fields: At each dot, we record the tangential–tangential component, the tangential–normal component, etc.
  - Can impose conditions on the components such as symmetry, trace-free, etc.
  - Can enforce continuity for all components or just some of them.
- General tensor fields are analogous.

Vector Laplacian eigenvalue problems on surfaces

Hodge Laplacian (e.g. Maxwell)

$$(dd^* + d^*d)v^\flat = \lambda v^\flat.$$

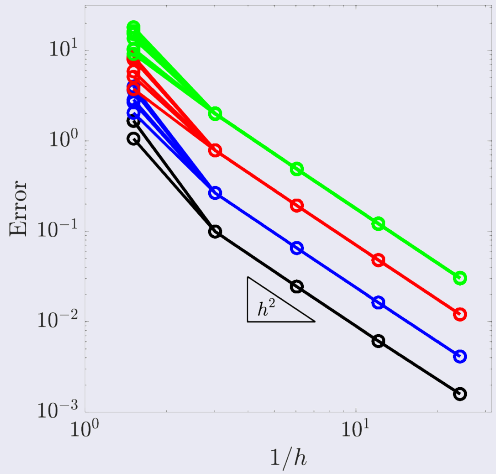
- Tangential continuity across edges suffices.
- Standard FEEC works.
- L^2 pairing suffices.

Bochner Laplacian (e.g. Stokes)

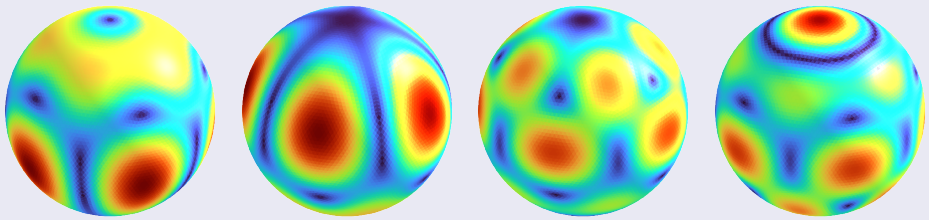
$$\nabla^*\nabla v = \lambda v.$$

- Must have full continuity across edges.
- Can't use standard FEEC.
- Needs Riemannian metric.

Bochner Laplacian on sphere using blow-up elements



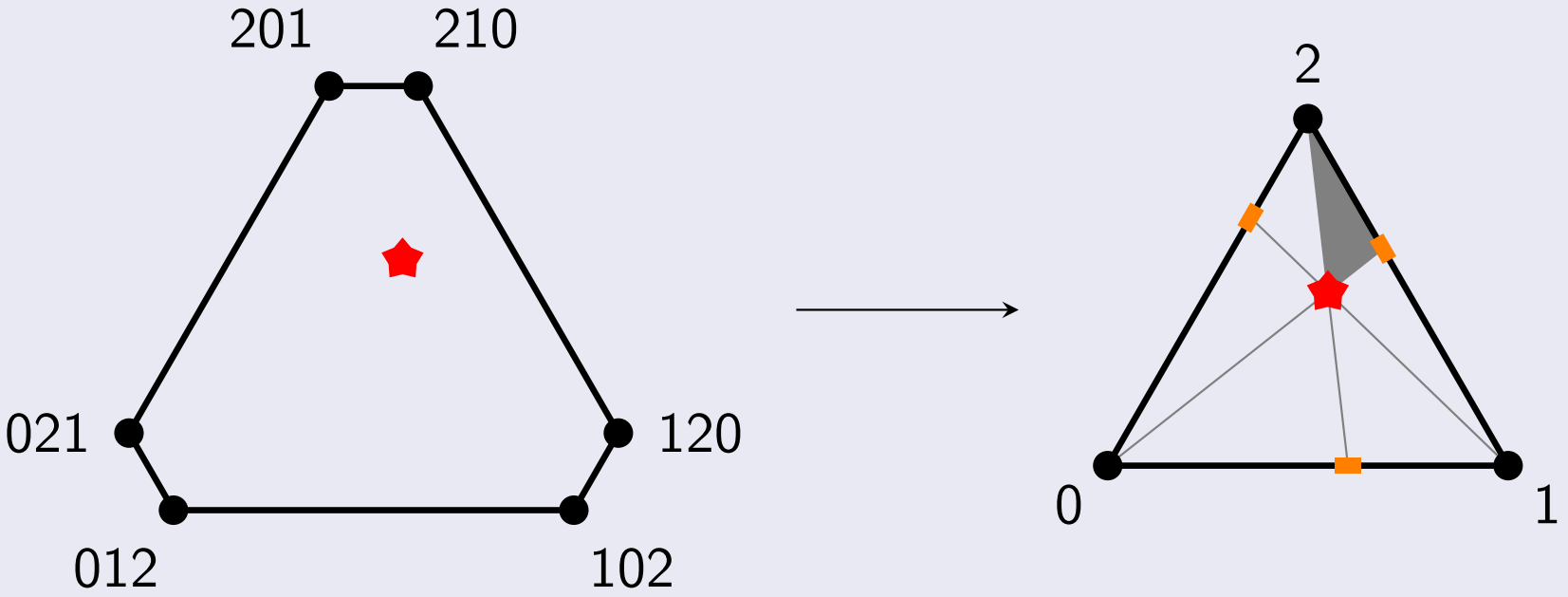
Eigenvalue error



Eigenfield magnitude ($\lambda = 11, 11, 19, 19$)

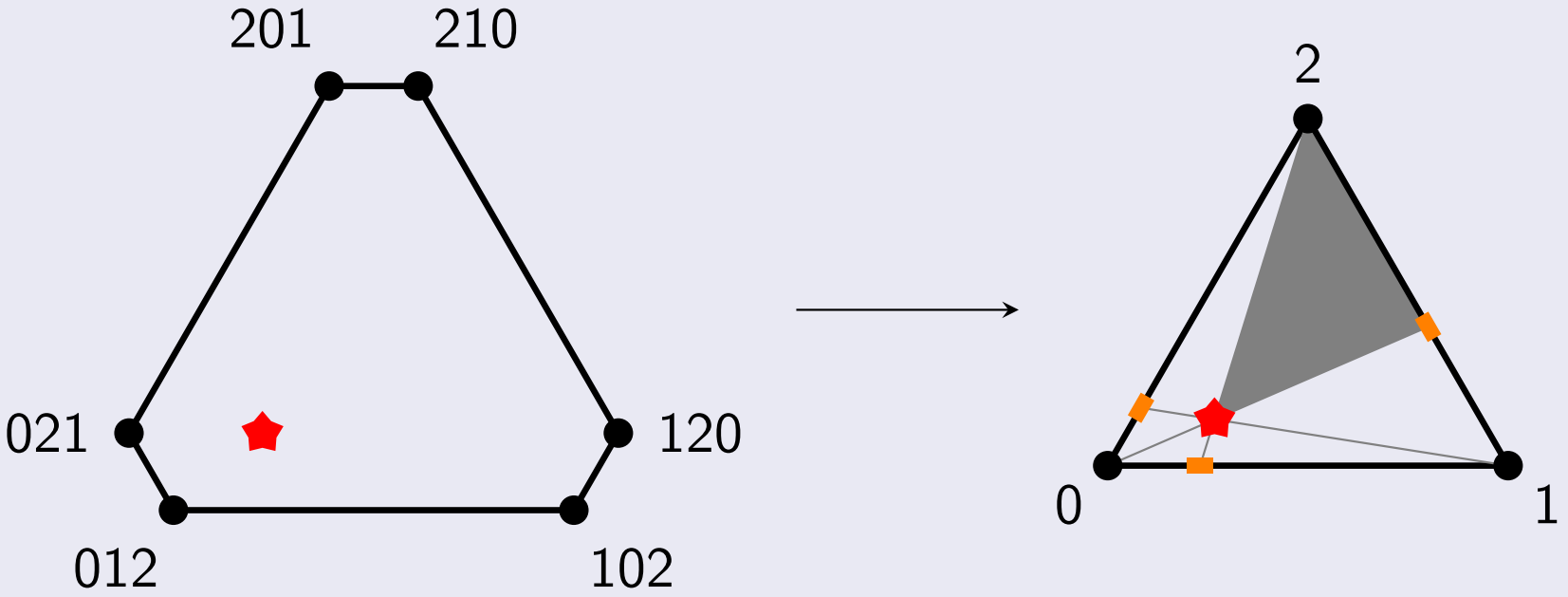
Blowing up the simplex

Configuration space of barycentric subdivisions



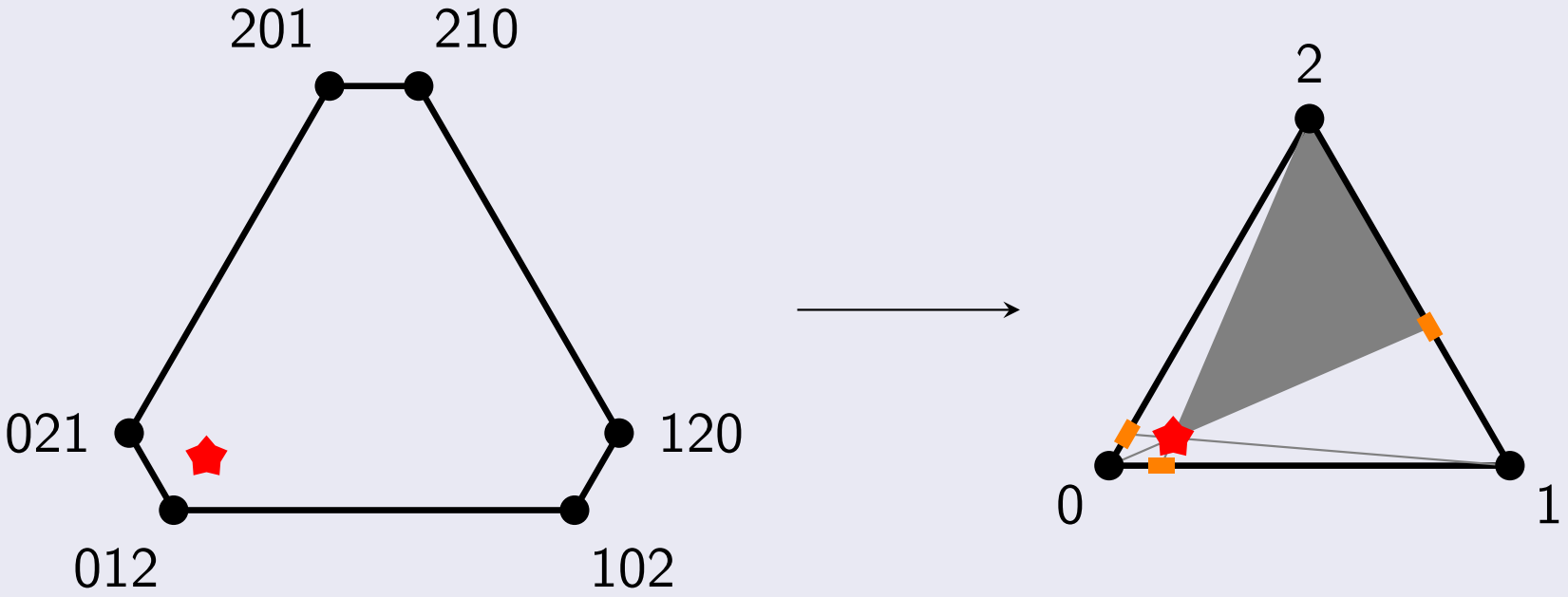
Blowing up the simplex

Configuration space of barycentric subdivisions



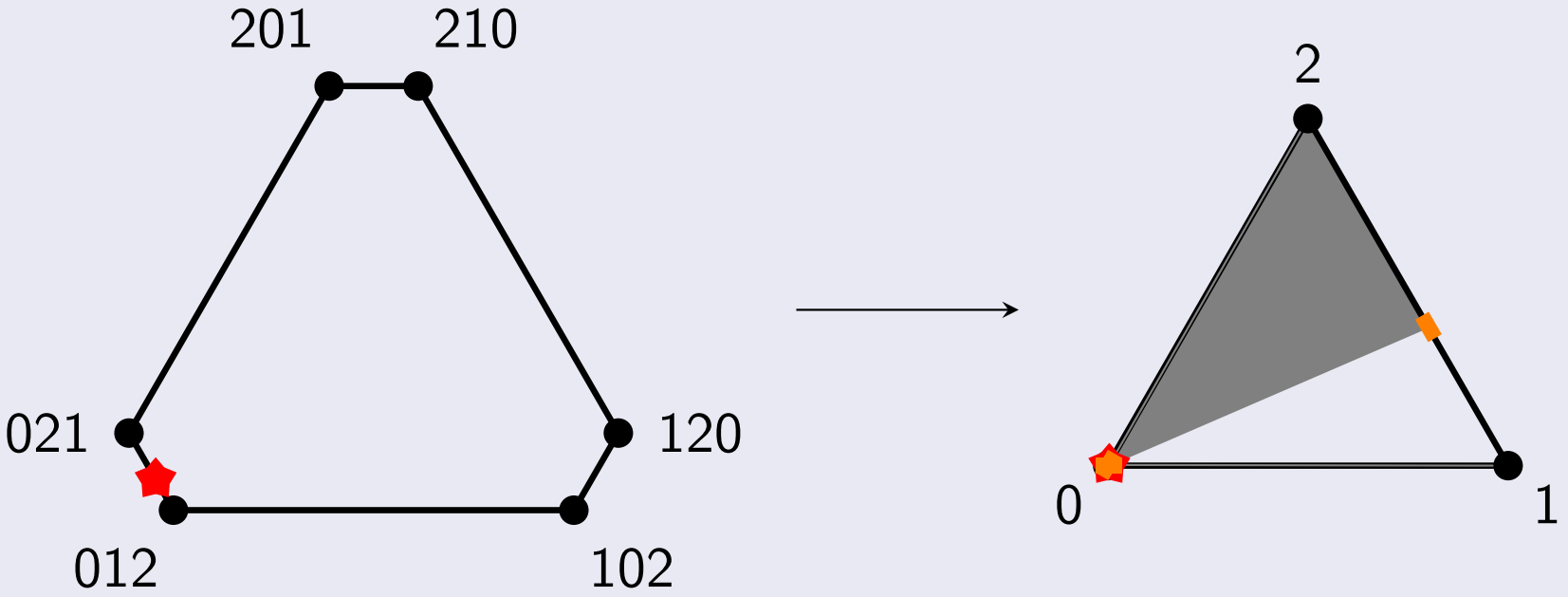
Blowing up the simplex

Configuration space of barycentric subdivisions



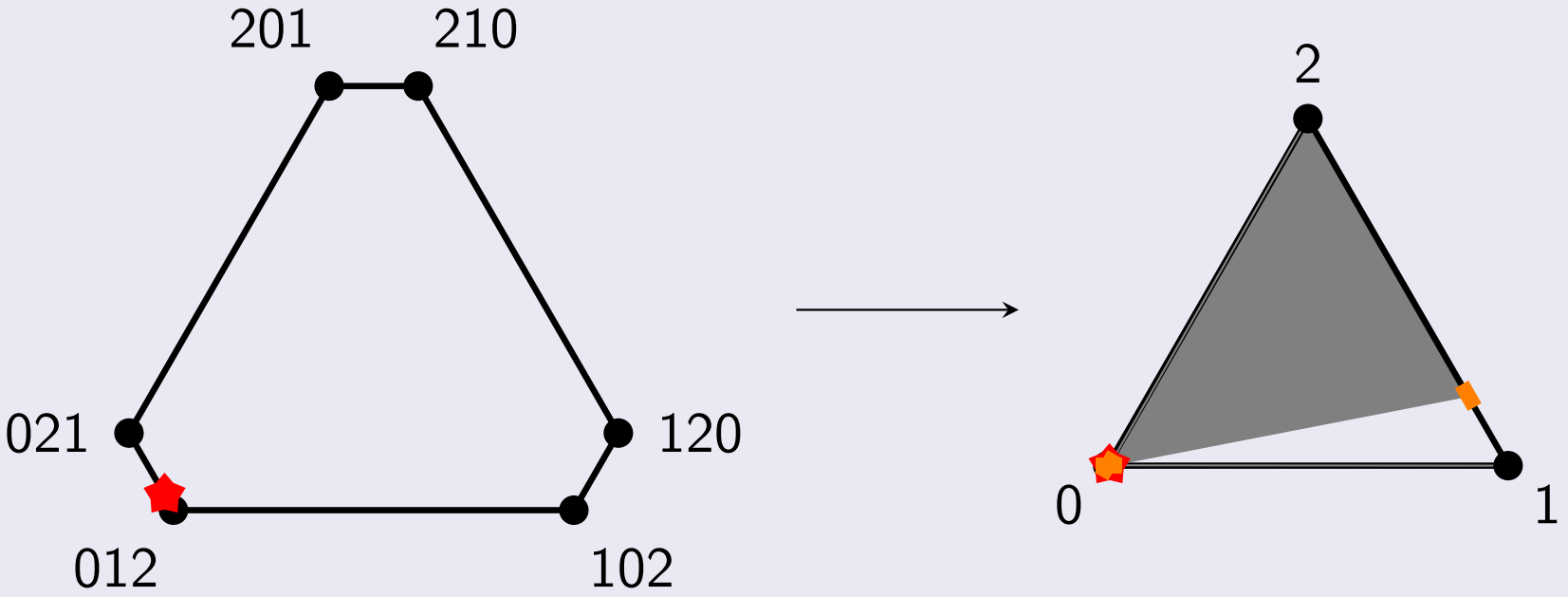
Blowing up the simplex

Configuration space of barycentric subdivisions



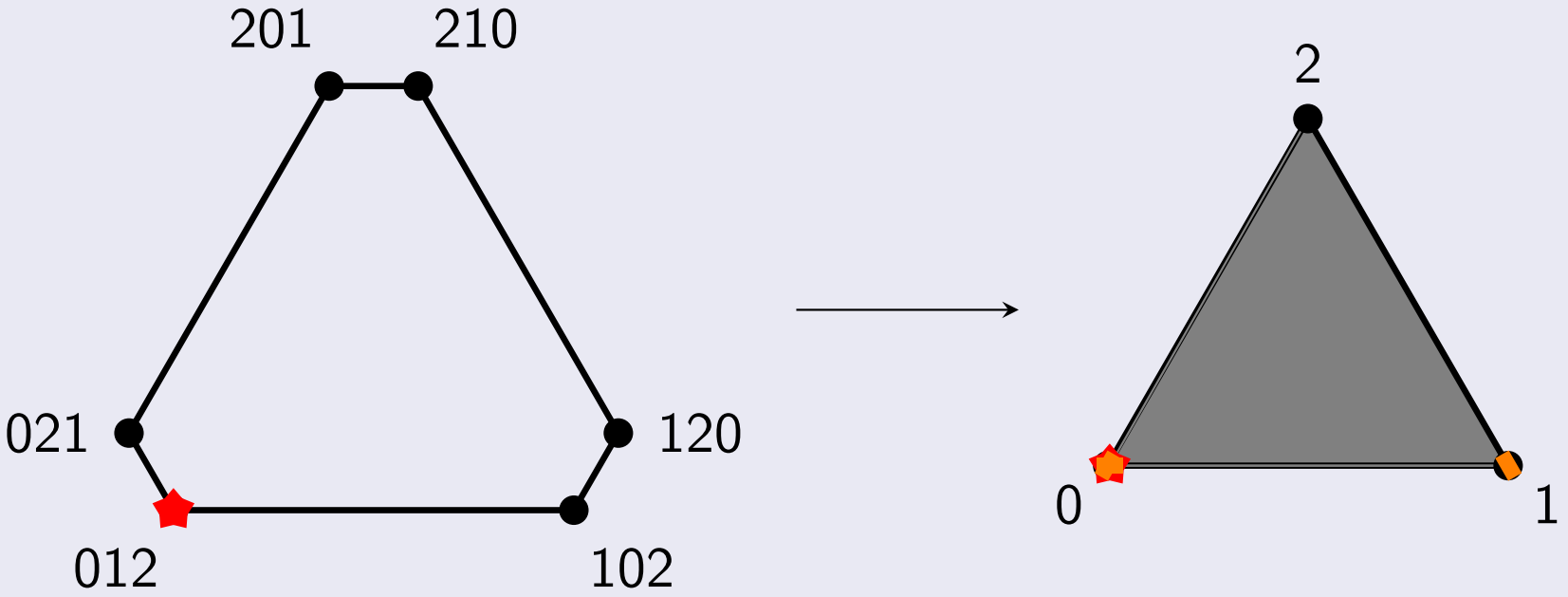
Blowing up the simplex

Configuration space of barycentric subdivisions



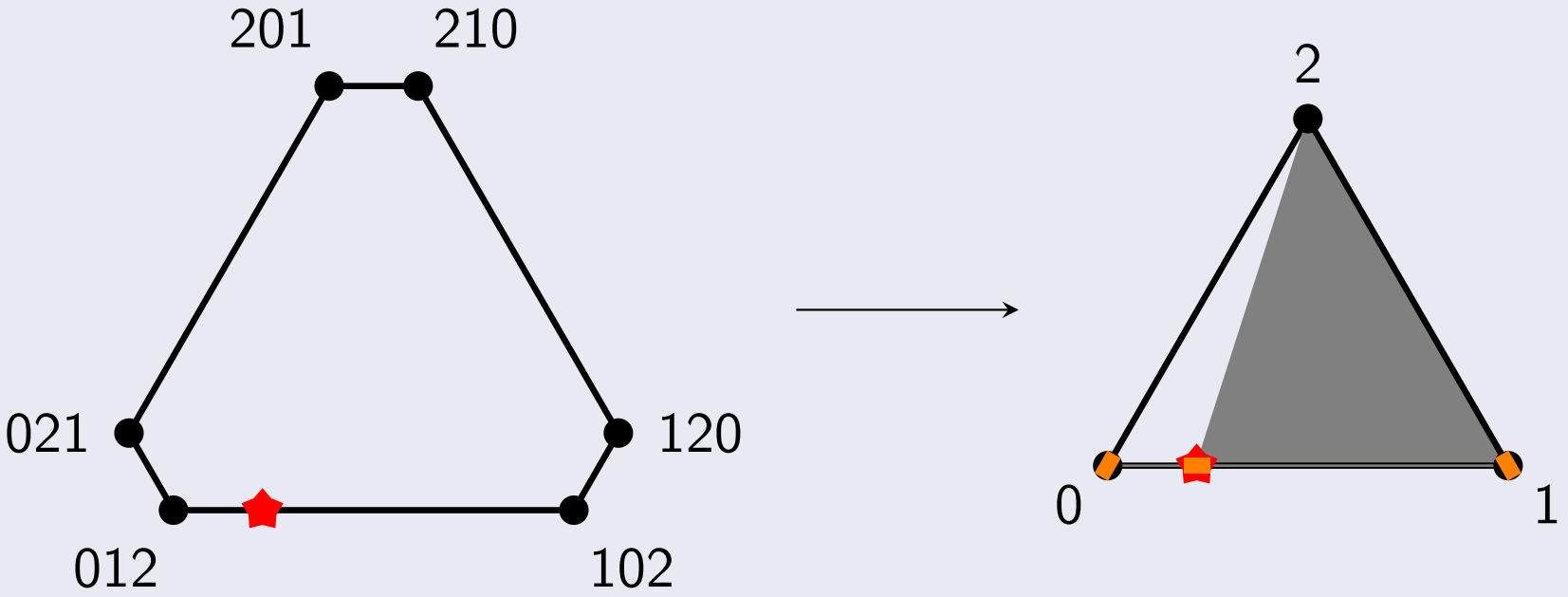
Blowing up the simplex

Configuration space of barycentric subdivisions



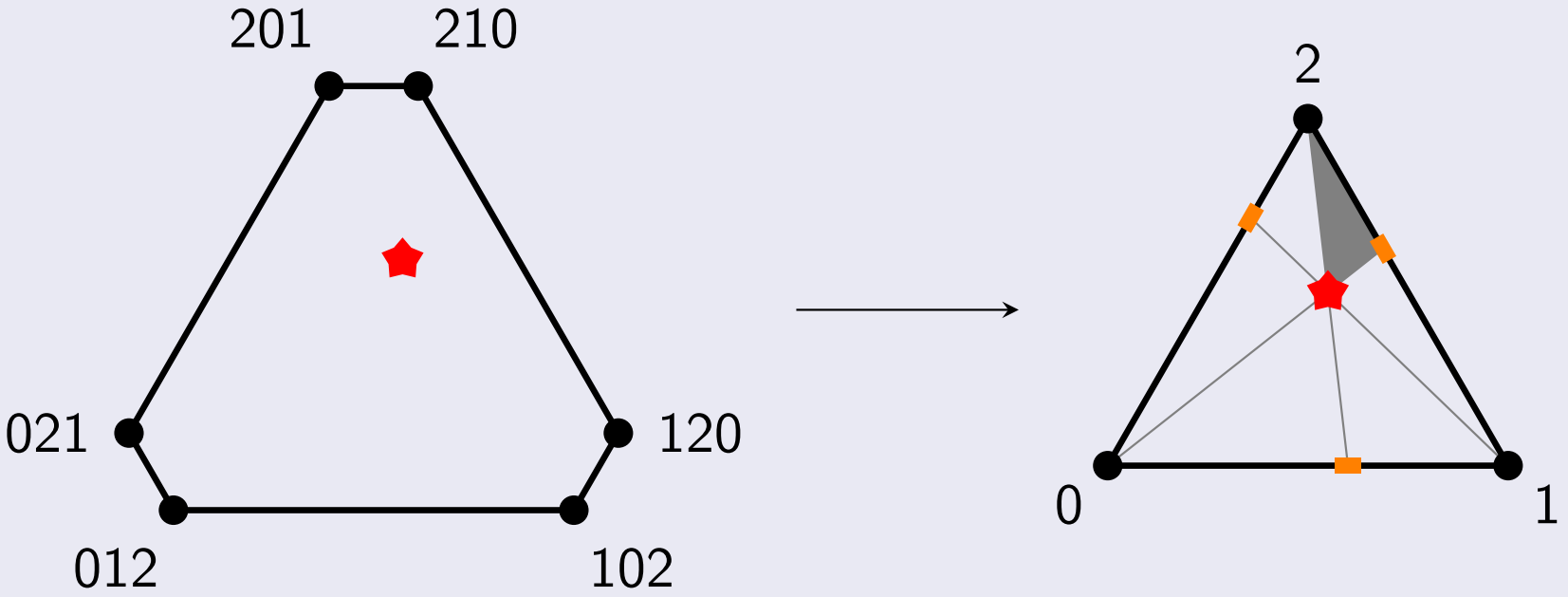
Blowing up the simplex

Configuration space of barycentric subdivisions



Blowing up the simplex

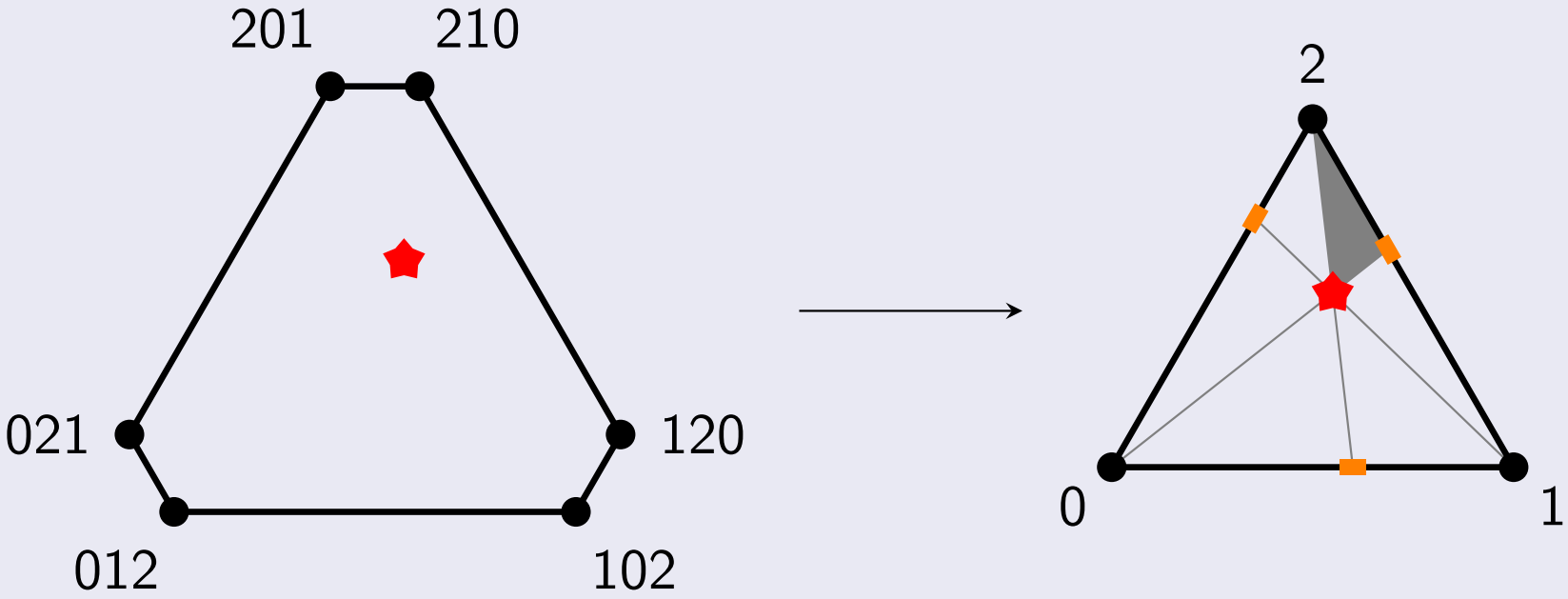
Configuration space of barycentric subdivisions



- Another interpretation of $\psi_{012} = \lambda_0 \frac{\lambda_1}{\lambda_1+\lambda_2}$: the relative area of the shaded region.

Blowing up the simplex

Configuration space of barycentric subdivisions

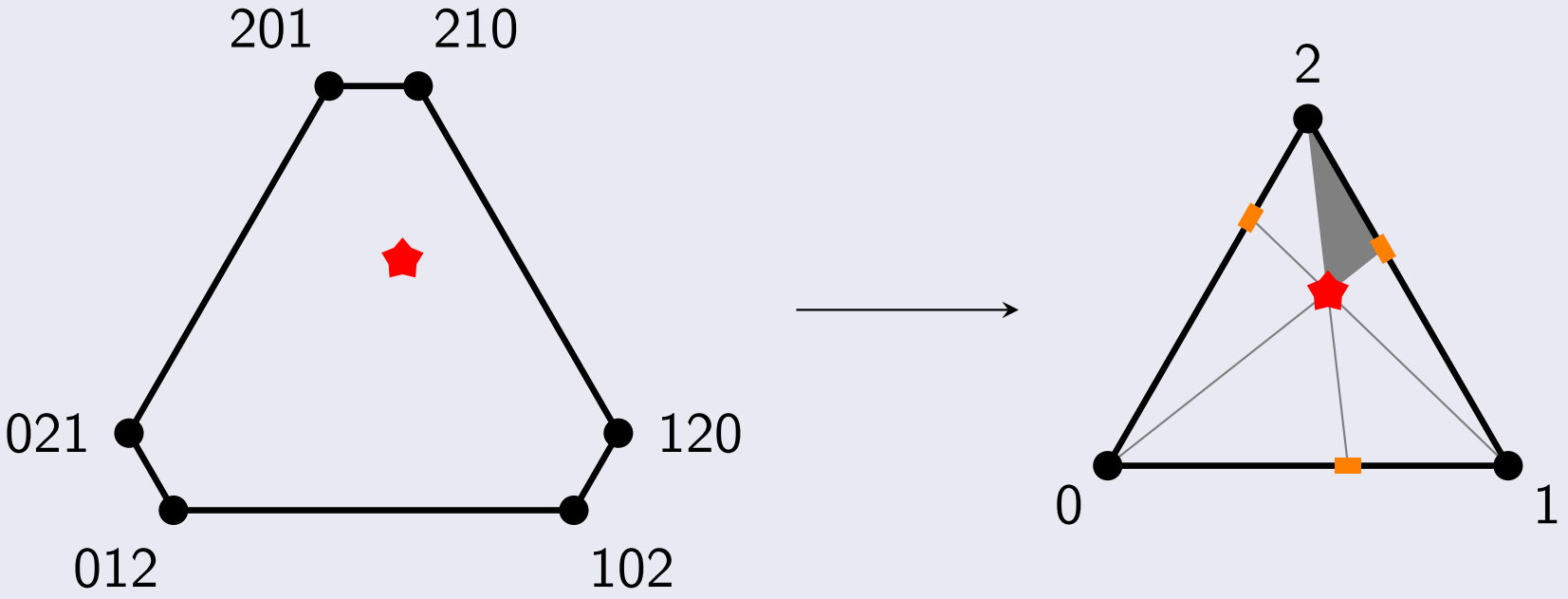


Smooth $\longleftrightarrow$ Smooth "in polar coordinates"

- Another interpretation of $\psi_{012} = \lambda_0 \frac{\lambda_1}{\lambda_1+\lambda_2}$: the relative area of the shaded region.

Blowing up the simplex

Configuration space of barycentric subdivisions



Smooth $\longleftrightarrow$ Smooth "in polar coordinates"

- Another interpretation of $\psi_{012} = \lambda_0 \frac{\lambda_1}{\lambda_1+\lambda_2}$: the relative area of the shaded region.
- For the general theory of blowing up manifolds with corners, see Melrose, 1996.

There's more

So far in this talk

There's more

So far in this talk

- Lowest order blow-up scalar fields in two dimensions, $b\mathcal{P}_1(T^2)$,

There's more

So far in this talk

- Lowest order blow-up scalar fields in two dimensions, $b\mathcal{P}_1(T^2)$,
  - including tensor fields with components in $b\mathcal{P}_1(T^2)$.

There's more

So far in this talk

- Lowest order blow-up scalar fields in two dimensions, $b\mathcal{P}_1(T^2)$,
  - including tensor fields with components in $b\mathcal{P}_1(T^2)$.

Our paper

There's more

So far in this talk

- Lowest order blow-up scalar fields in two dimensions, $b\mathcal{P}_1(T^2)$,
  - including tensor fields with components in $b\mathcal{P}_1(T^2)$.

Our paper

- Differential complex of blow-up Whitney k-forms in any dimension, $b\mathcal{P}_1^-\Lambda^k(T^n)$.

There's more

So far in this talk

- Lowest order blow-up scalar fields in two dimensions, $b\mathcal{P}_1(T^2)$,
  - including tensor fields with components in $b\mathcal{P}_1(T^2)$.

Our paper

- Differential complex of blow-up Whitney k-forms in any dimension, $b\mathcal{P}_1^-\Lambda^k(T^n)$.
  - Shape functions previously studied in (Brasselet, Goresky, MacPherson, 1991), called shadow forms.

There's more

So far in this talk

- Lowest order blow-up scalar fields in two dimensions, $b\mathcal{P}_1(T^2)$,
  - including tensor fields with components in $b\mathcal{P}_1(T^2)$.

Our paper

- Differential complex of blow-up Whitney k-forms in any dimension, $b\mathcal{P}_1^-\Lambda^k(T^n)$.
  - Shape functions previously studied in (Brasselet, Goresky, MacPherson, 1991), called shadow forms.
- ~~Higher-order blow-up scalar fields $b\mathcal{P}_r(T^n)$.~~

There's more

So far in this talk

- Lowest order blow-up scalar fields in two dimensions, $b\mathcal{P}_1(T^2)$,
  - including tensor fields with components in $b\mathcal{P}_1(T^2)$.

Our paper

- Differential complex of blow-up Whitney k-forms in any dimension, $b\mathcal{P}_1^-\Lambda^k(T^n)$.
  - Shape functions previously studied in (Brasselet, Goresky, MacPherson, 1991), called shadow forms.
- ~~Higher-order blow-up scalar fields $b\mathcal{P}_r(T^n)$.~~
- A surprising connection to arrival times of Poisson processes, yielding simpler computations.

There's more

So far in this talk

- Lowest order blow-up scalar fields in two dimensions, $b\mathcal{P}_1(T^2)$,
  - including tensor fields with components in $b\mathcal{P}_1(T^2)$.

Our paper

- Differential complex of blow-up Whitney k-forms in any dimension, $b\mathcal{P}_1^-\Lambda^k(T^n)$.
  - Shape functions previously studied in (Brasselet, Goresky, MacPherson, 1991), called shadow forms.
- ~~Higher-order blow-up scalar fields $b\mathcal{P}_r(T^n)$.~~
- A surprising connection to arrival times of Poisson processes, yielding simpler computations.
- Degrees of freedom of blow-up Whitney k-forms are integration over k-faces of the blow-up simplex.

What's next

Higher order blow-up scalar fields bP_r.

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)
- Implement in NGSolve, joint with undergraduate Aidan Nelappana.

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)
- Implement in NGSolve, joint with undergraduate Aidan Nelappana.

Connections to rational functions in finite element spaces?

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)
- Implement in NGSolve, joint with undergraduate Aidan Nelappana.

Connections to rational functions in finite element spaces?

- Winther's talk on Monday.

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)
- Implement in NGSolve, joint with undergraduate Aidan Nelappana.

Connections to rational functions in finite element spaces?

- Winther's talk on Monday.
- Rational functions for Stokes problem (Guzmán, Neilan, 2014).

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)
- Implement in NGSolve, joint with undergraduate Aidan Nelappana.

Connections to rational functions in finite element spaces?

- Winther's talk on Monday.
- Rational functions for Stokes problem (Guzmán, Neilan, 2014).
- Duffy transform.

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)
- Implement in NGSolve, joint with undergraduate Aidan Nelappana.

Connections to rational functions in finite element spaces?

- Winther's talk on Monday.
- Rational functions for Stokes problem (Guzmán, Neilan, 2014).
- Duffy transform.

Distributional Riemann curvature

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)
- Implement in NGSolve, joint with undergraduate Aidan Nelappana.

Connections to rational functions in finite element spaces?

- Winther's talk on Monday.
- Rational functions for Stokes problem (Guzmán, Neilan, 2014).
- Duffy transform.

Distributional Riemann curvature

- Via blow-ups and frames: (Gawlik, McKee, 2026).

What's next

Higher order blow-up scalar fields bP_r.

- Fluid flow (surface Stokes complex): blow-up generalization of P_2–P_1 velocity–presure pair. (See also Demlow, Neilan on surface Stokes.)
- Implement in NGSolve, joint with undergraduate Aidan Nelappana.

Connections to rational functions in finite element spaces?

- Winther's talk on Monday.
- Rational functions for Stokes problem (Guzmán, Neilan, 2014).
- Duffy transform.

Distributional Riemann curvature

- Via blow-ups and frames: (Gawlik, McKee, 2026).
- Original approach (—, Gawlik) via blow-ups and

$$\int_M \langle R_{\mathrm{dist}} v, w\rangle \wedge \beta := \int_M \langle \nabla v \wedge \nabla w\rangle \wedge \beta + \int_M \langle \nabla v, w\rangle \wedge d\beta$$

Section 2

Double forms

Differential forms corresponding to vector field $\langle M, N, P \rangle$

One-forms Λ^1

- $M\,dx + N\,dy + P\,dz$
- Restricted to the xy-plane $z = 0$:
  - $M\,dx + N\,dy$.
  - Tangential components.

Two-forms Λ^2

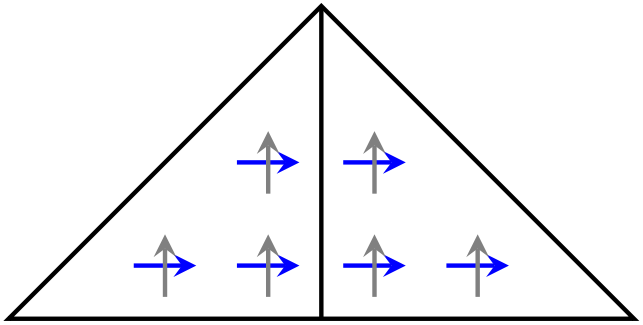
- $M\,dy \wedge dz + N\,dz \wedge dx + P\,dx \wedge dy$.
- Restricted to the xy-plane $z = 0$:
  - $P\,dx \wedge dy$.
  - Normal component.

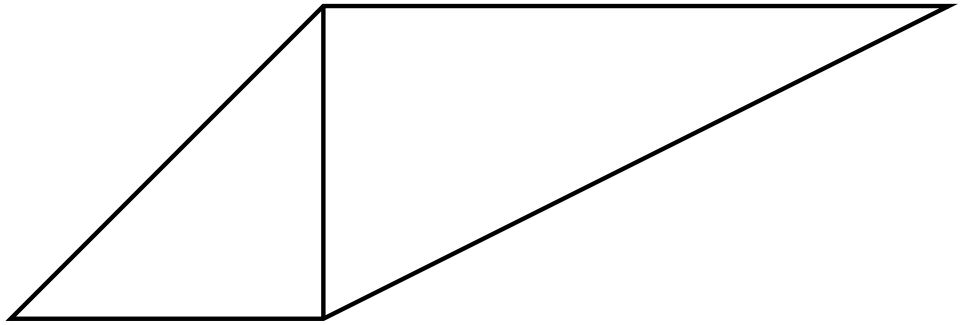
Continuity conditions

- Vector fields with tangential continuity are one-forms.
- Vector fields with normal continuity are $(n-1)$-forms.

How do "vector" fields transform?

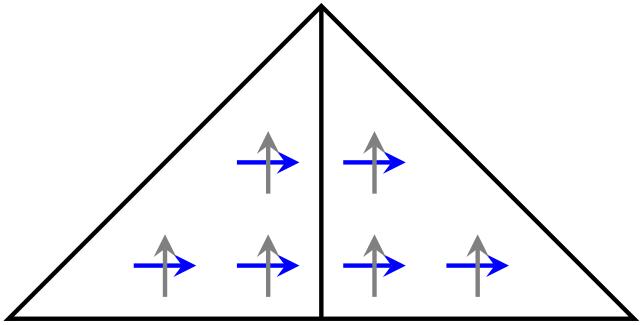
- Transform as velocity (vector field): tangential and normal jumps.

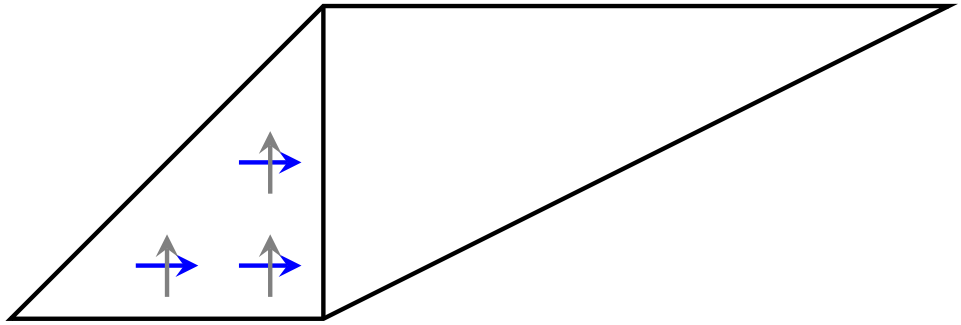




How do "vector" fields transform?

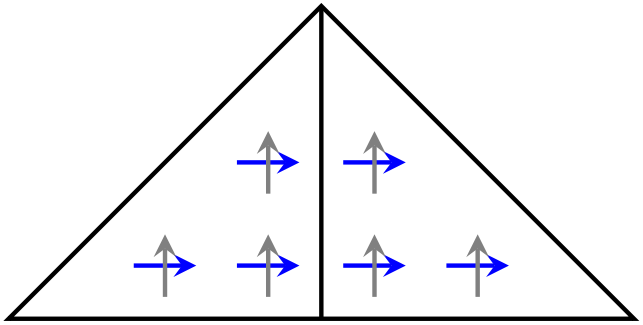
- Transform as velocity (vector field): tangential and normal jumps.

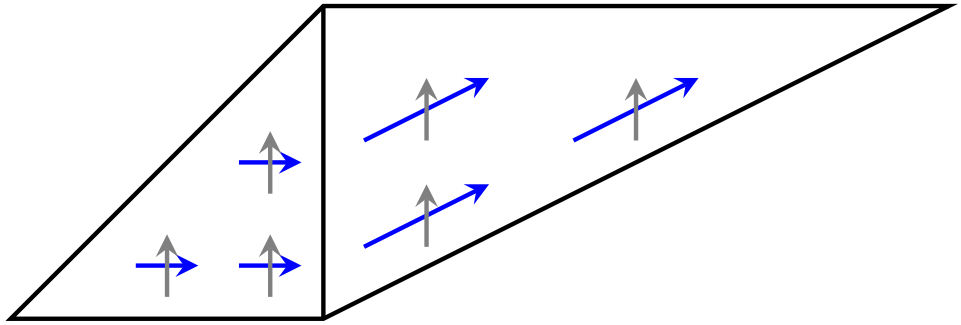




How do "vector" fields transform?

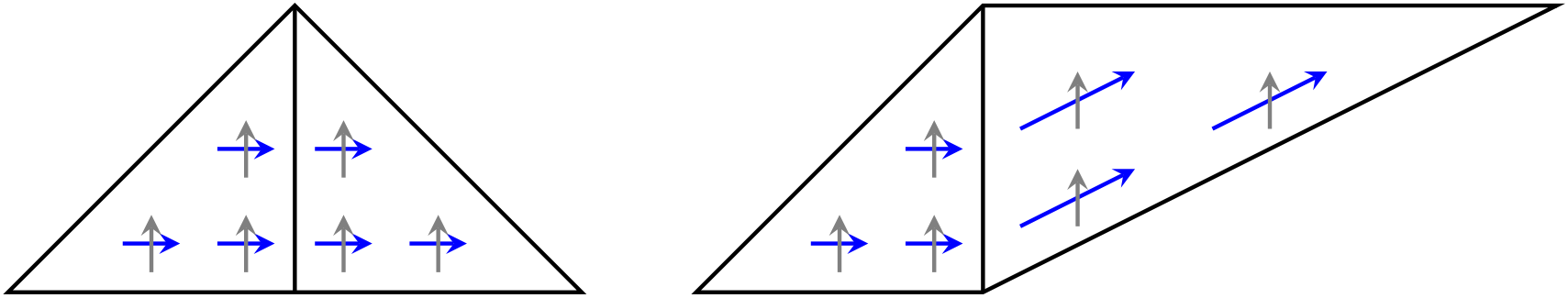
- Transform as velocity (vector field): tangential and normal jumps.



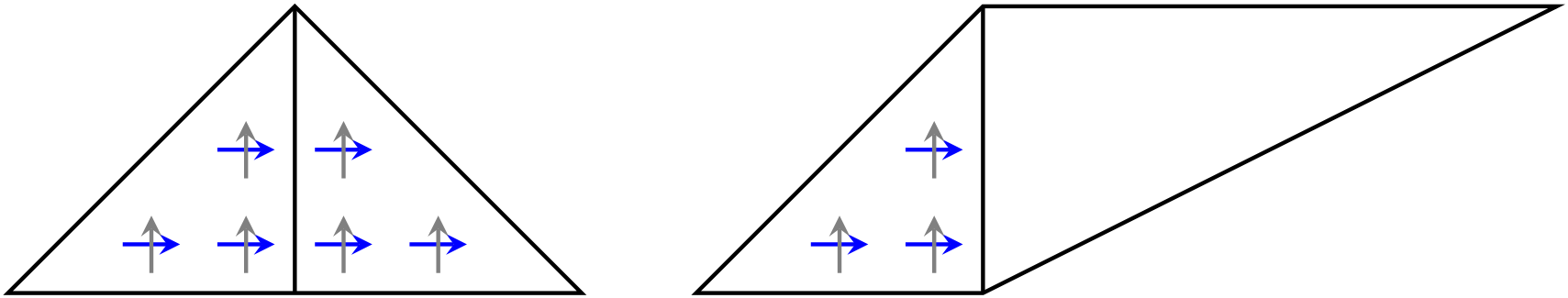


How do "vector" fields transform?

- Transform as velocity (vector field): tangential and normal jumps.

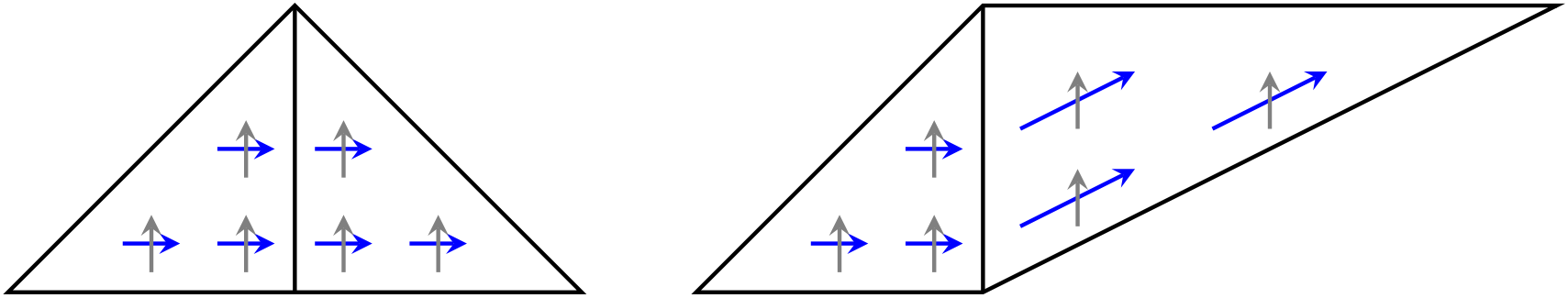


- Transform as gradient (1-form $g(v, \cdot)$): keep tangential continuity.

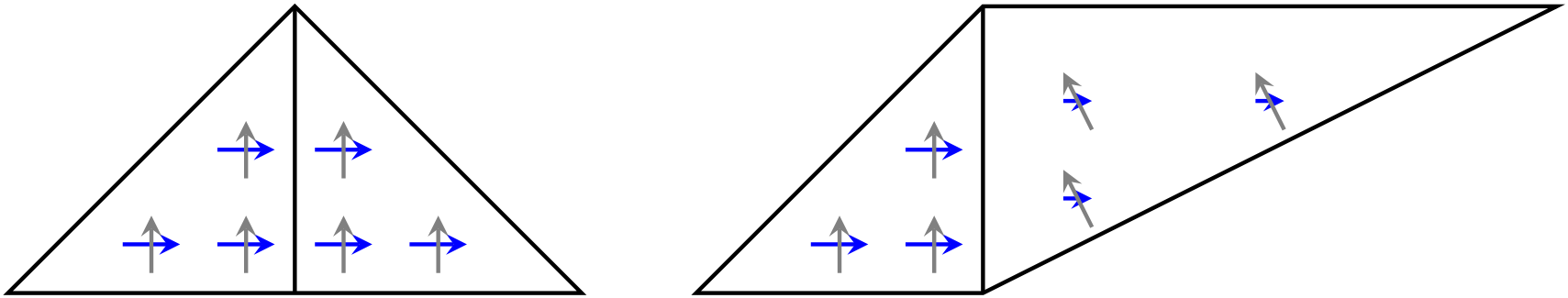


How do "vector" fields transform?

- Transform as velocity (vector field): tangential and normal jumps.

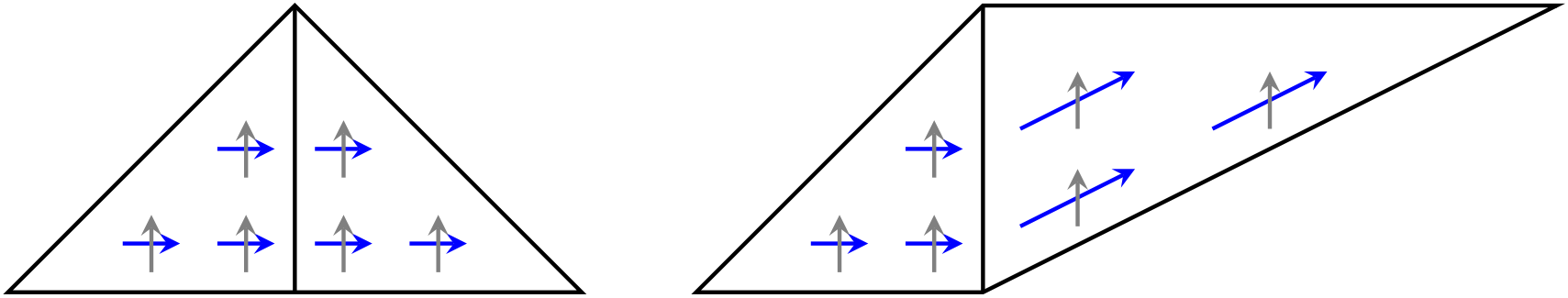


- Transform as gradient (1-form $g(v, \cdot)$): keep tangential continuity.

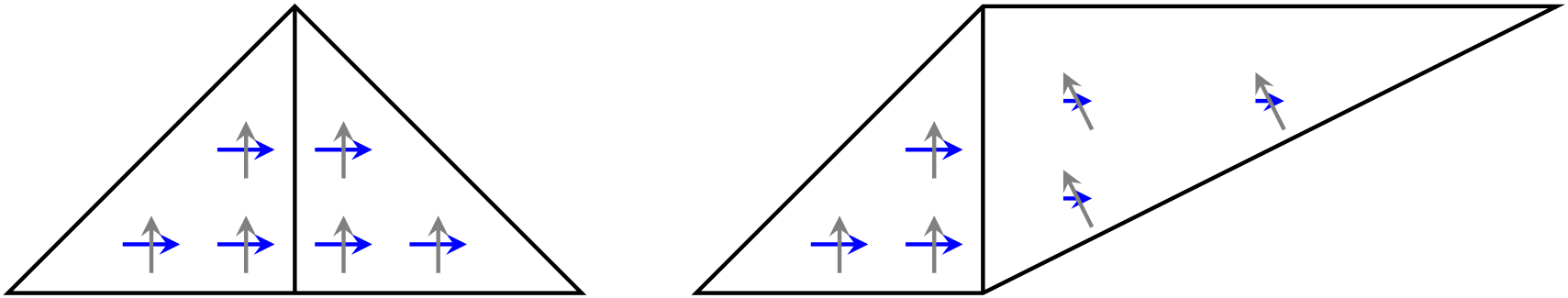


How do "vector" fields transform?

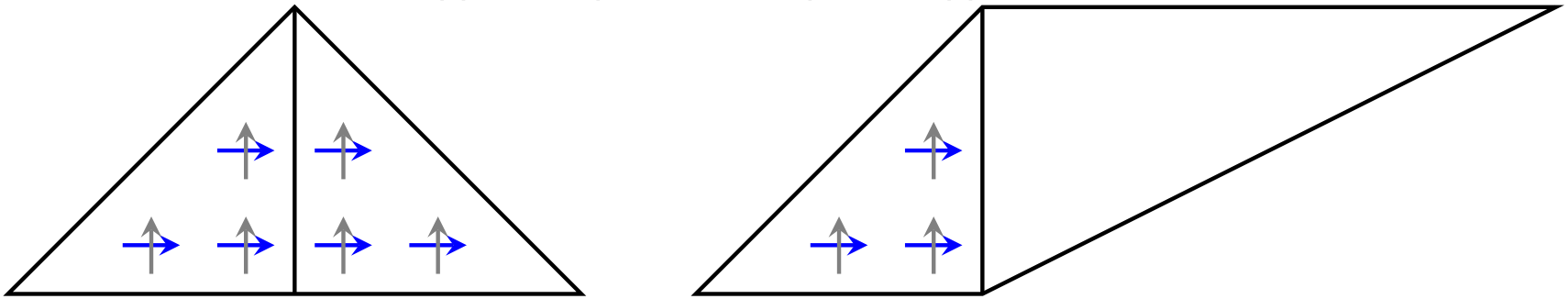
- Transform as velocity (vector field): tangential and normal jumps.



- Transform as gradient (1-form $g(v, \cdot)$): keep tangential continuity.

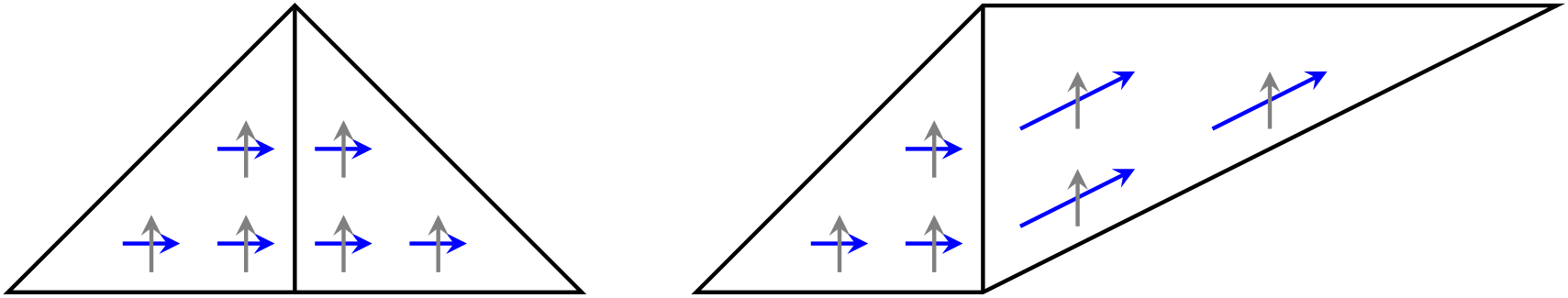


- Transform as flux ($(n-1)$-form $\omega(v, \cdots)$): keep normal continuity.

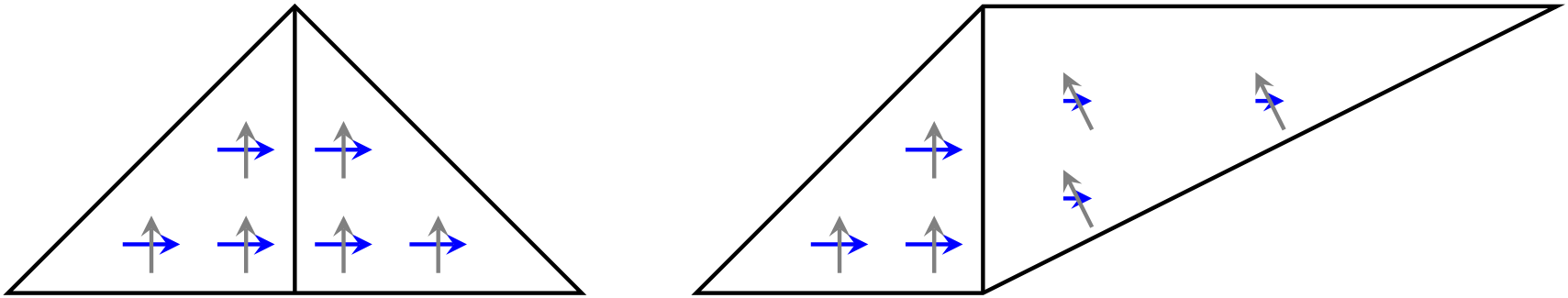


How do "vector" fields transform?

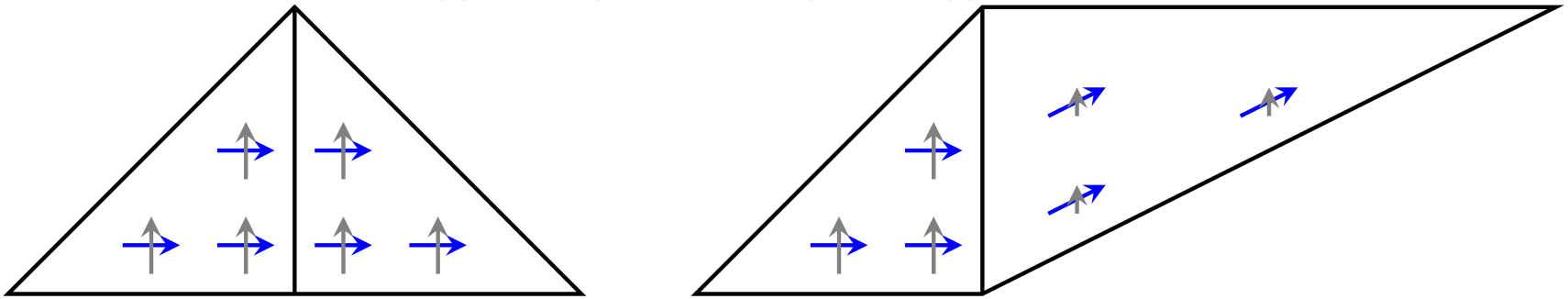
- Transform as velocity (vector field): tangential and normal jumps.



- Transform as gradient (1-form $g(v, \cdot)$): keep tangential continuity.



- Transform as flux ($(n-1)$-form $\omega(v, \cdots)$): keep normal continuity.



Finite Element Exterior Calculus is affine-invariant

- FEEC spaces are invariant under piecewise affine transformations of the mesh.

Finite Element Exterior Calculus is affine-invariant

- FEEC spaces are invariant under piecewise affine transformations of the mesh.
  - Start with a FEEC field on a mesh. Transform the mesh and correspondingly transform the field. The resulting field will be a valid FEEC field on the new mesh.

Finite Element Exterior Calculus is affine-invariant

- FEEC spaces are invariant under piecewise affine transformations of the mesh.
  - Start with a FEEC field on a mesh. Transform the mesh and correspondingly transform the field. The resulting field will be a valid FEEC field on the new mesh.
- We can make surface meshes by applying a piecewise affine transformation to a flat mesh.

Finite Element Exterior Calculus is affine-invariant

- FEEC spaces are invariant under piecewise affine transformations of the mesh.
  - Start with a FEEC field on a mesh. Transform the mesh and correspondingly transform the field. The resulting field will be a valid FEEC field on the new mesh.
- We can make surface meshes by applying a piecewise affine transformation to a flat mesh.
- Consequently:

Finite Element Exterior Calculus is affine-invariant

- FEEC spaces are invariant under piecewise affine transformations of the mesh.
  - Start with a FEEC field on a mesh. Transform the mesh and correspondingly transform the field. The resulting field will be a valid FEEC field on the new mesh.
- We can make surface meshes by applying a piecewise affine transformation to a flat mesh.
- Consequently:
  - FEEC spaces work the same way on surfaces as they do in the plane.

Finite Element Exterior Calculus is affine-invariant

- FEEC spaces are invariant under piecewise affine transformations of the mesh.
  - Start with a FEEC field on a mesh. Transform the mesh and correspondingly transform the field. The resulting field will be a valid FEEC field on the new mesh.
- We can make surface meshes by applying a piecewise affine transformation to a flat mesh.
- Consequently:
  - FEEC spaces work the same way on surfaces as they do in the plane.
  - Spaces that depend on angle defect cannot be affine invariant.

Finite Element Exterior Calculus is affine-invariant

- FEEC spaces are invariant under piecewise affine transformations of the mesh.
  - Start with a FEEC field on a mesh. Transform the mesh and correspondingly transform the field. The resulting field will be a valid FEEC field on the new mesh.
- We can make surface meshes by applying a piecewise affine transformation to a flat mesh.
- Consequently:
  - FEEC spaces work the same way on surfaces as they do in the plane.
  - Spaces that depend on angle defect cannot be affine invariant.
    - No FEEC proxy for vector fields with full continuity.

Finite Element Exterior Calculus is affine-invariant

- FEEC spaces are invariant under piecewise affine transformations of the mesh.
  - Start with a FEEC field on a mesh. Transform the mesh and correspondingly transform the field. The resulting field will be a valid FEEC field on the new mesh.
- We can make surface meshes by applying a piecewise affine transformation to a flat mesh.
- Consequently:
  - FEEC spaces work the same way on surfaces as they do in the plane.
  - Spaces that depend on angle defect cannot be affine invariant.
    - No FEEC proxy for vector fields with full continuity.
- What about Finite Element Tensor Calculus (FETC)? Can we make affine-invariant spaces?

Extending FEEC to matrices and tensors

Vector fields ($\mathbb{R}^3$)

- Vector fields with tangential continuity are one-forms Λ^1.
- Vector fields with normal continuity are two-forms Λ^2.

Matrix fields ($\mathbb{R}^3 \otimes \mathbb{R}^3$)

- Matrix fields with tangential–tangential continuity are $(1,1)$-forms $\Lambda^{1,1} := \Lambda^1 \otimes \Lambda^1$.
- Matrix fields with normal–tangential continuity are $(2,1)$-forms $\Lambda^{2,1} := \Lambda^2 \otimes \Lambda^1$.
- Matrix fields with normal–normal continuity are $(2,2)$-forms $\Lambda^{2,2} := \Lambda^2 \otimes \Lambda^2$.

Applications

Strain/stress tensors

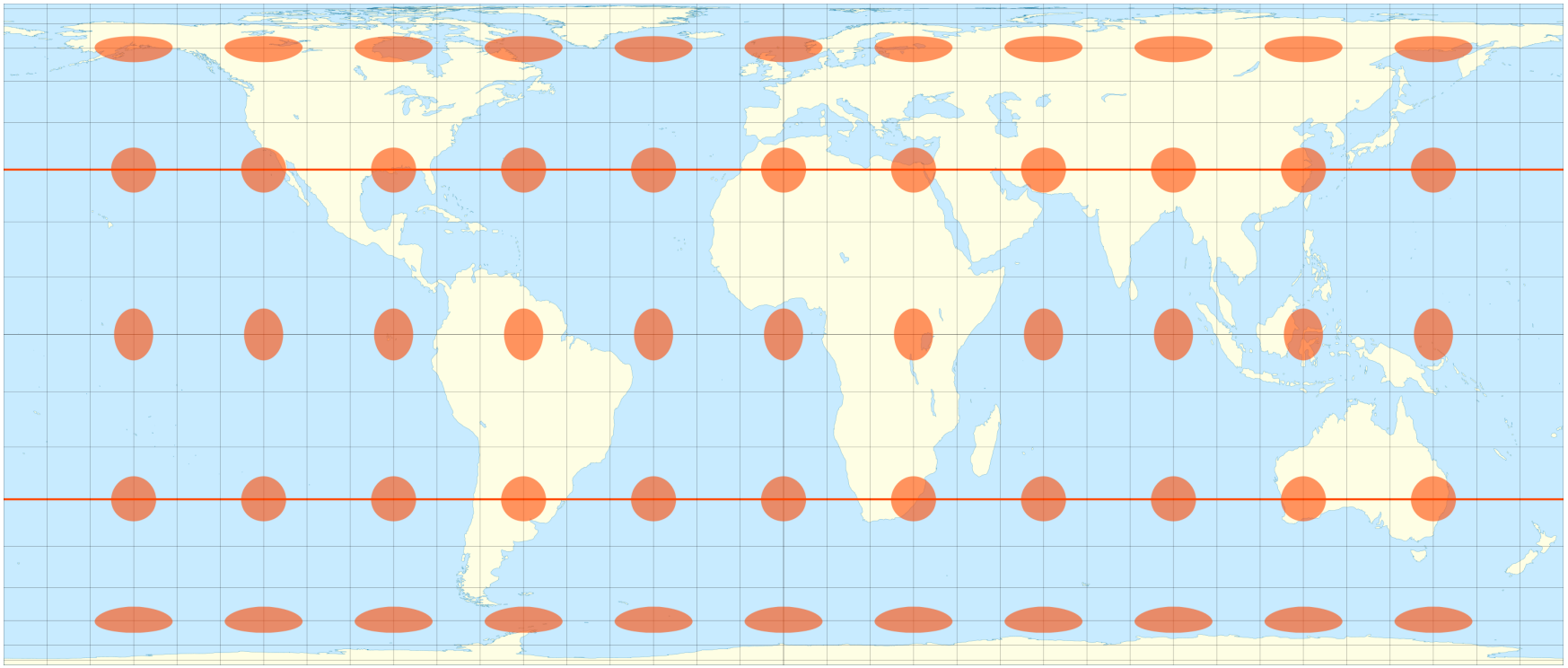
- Elasticity (objects deforming under stress)
- Fluid mechanics (Stokes equations)

Numerical geometry/relativity

- Riemannian/Minkowski metric
- Curvature tensor

Regge metrics $\Lambda_0^{1,1}$

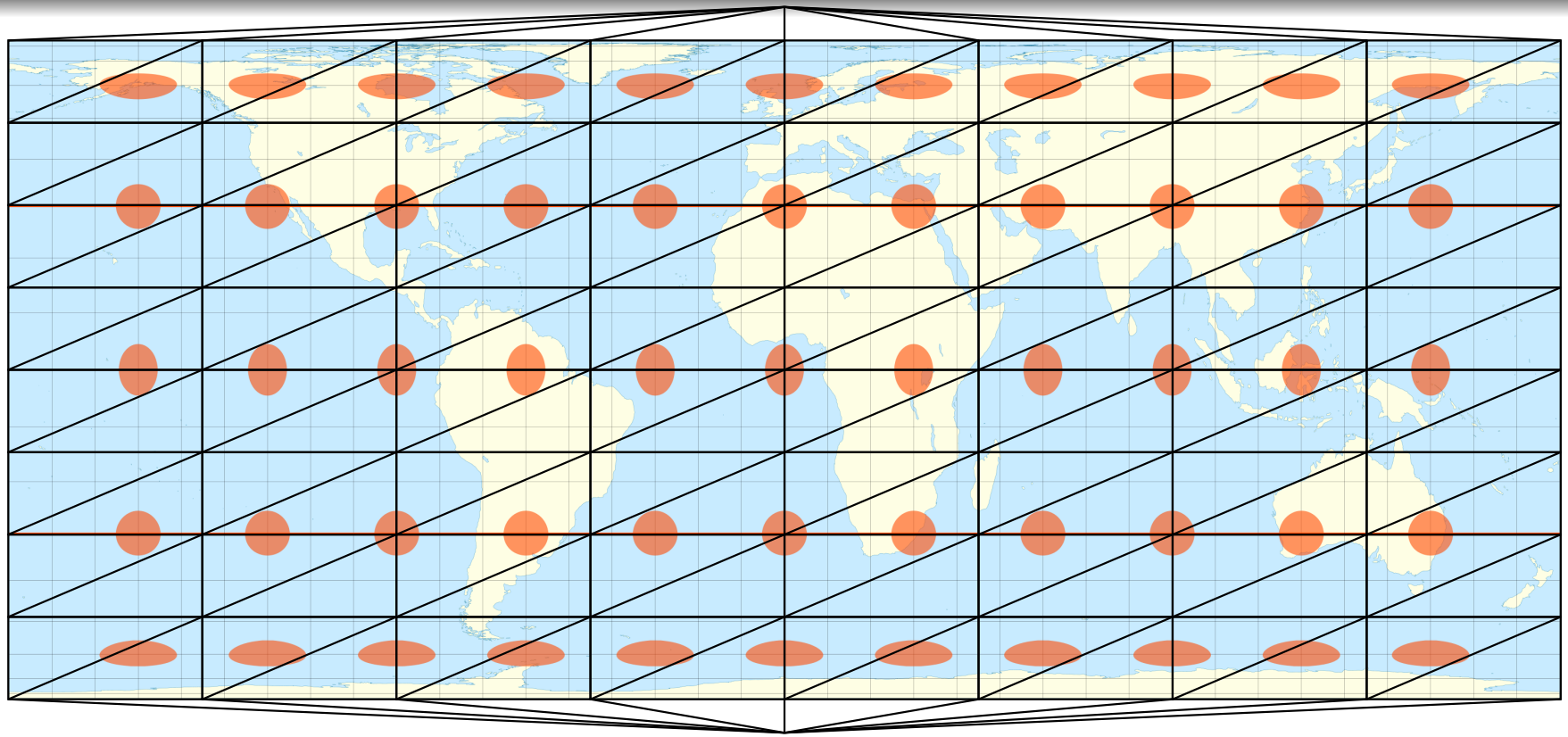
Symmetric matrix fields with tangential–tangential continuity



Map credit: Wikipedia, Gaba

Regge metrics $\Lambda_0^{1,1}$

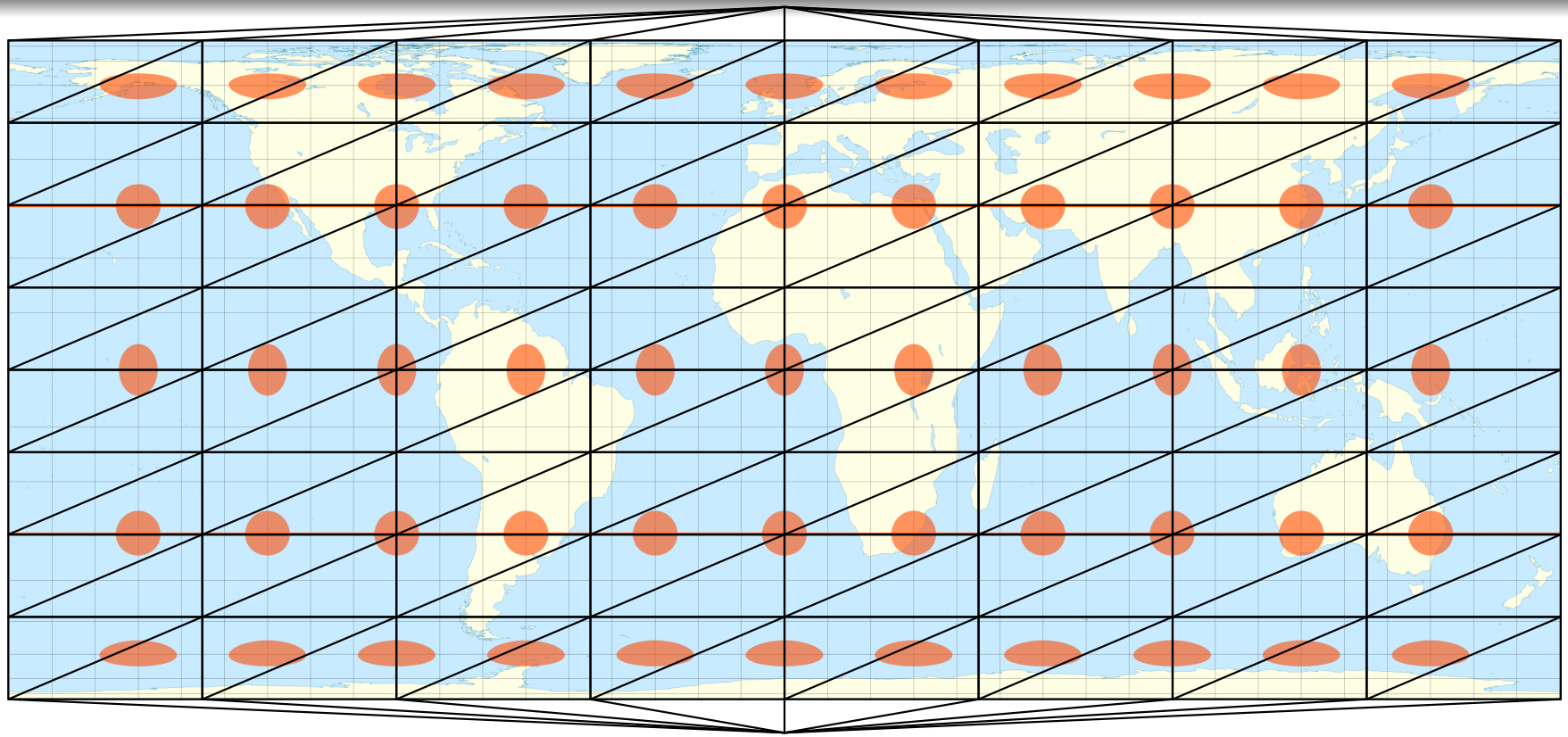
Symmetric matrix fields with tangential–tangential continuity



Map credit: Wikipedia, Gaba

Regge metrics $\Lambda_0^{1,1}$

Symmetric matrix fields with tangential–tangential continuity



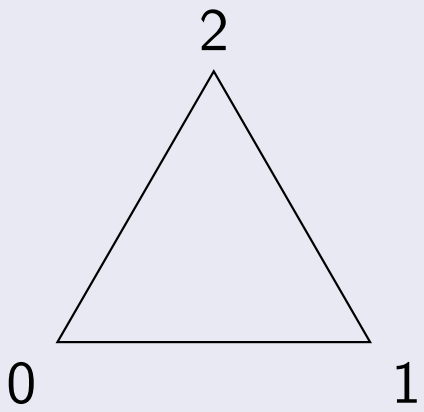
Regge finite elements

- Record the length of each edge.
- For each triangle, use the corresponding Euclidean metric.
- Get piecewise constant metric with tang.–tang. continuity.

Map credit: Wikipedia, Gaba

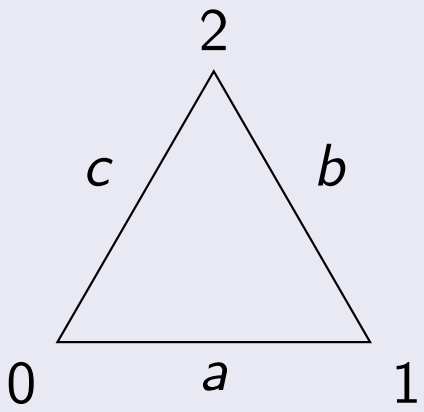
Regge metric on a reference triangle

Barycentric coordinates $\lambda_0 + \lambda_1 + \lambda_2 = 1$



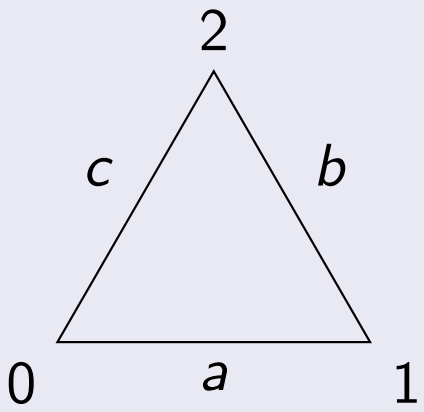
Regge metric on a reference triangle

Barycentric coordinates $\lambda_0 + \lambda_1 + \lambda_2 = 1$



Regge metric on a reference triangle

Barycentric coordinates $\lambda_0 + \lambda_1 + \lambda_2 = 1$

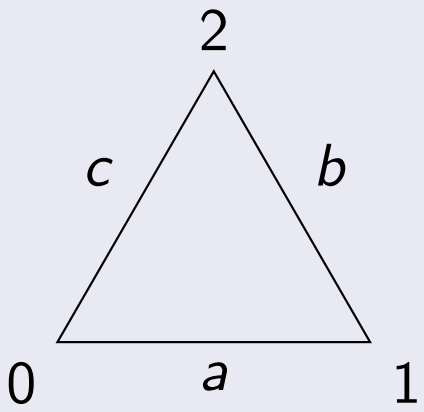


Regge metric:

$$
\begin{aligned}
g = &-\tfrac{1}{2}a^2(d\lambda_0 \otimes d\lambda_1 + d\lambda_1 \otimes d\lambda_0) \\
&-\tfrac{1}{2}b^2(d\lambda_1 \otimes d\lambda_2 + d\lambda_2 \otimes d\lambda_1) \\
&-\tfrac{1}{2}c^2(d\lambda_2 \otimes d\lambda_0 + d\lambda_0 \otimes d\lambda_2)
\end{aligned}
$$

Regge metric on a reference triangle

Barycentric coordinates $\lambda_0 + \lambda_1 + \lambda_2 = 1$



Regge metric:

$$\begin{aligned} g = &-\tfrac{1}{2}a^2(d\lambda_0 \otimes d\lambda_1 + d\lambda_1 \otimes d\lambda_0) \\ &-\tfrac{1}{2}b^2(d\lambda_1 \otimes d\lambda_2 + d\lambda_2 \otimes d\lambda_1) \\ &-\tfrac{1}{2}c^2(d\lambda_2 \otimes d\lambda_0 + d\lambda_0 \otimes d\lambda_2) \end{aligned}$$

Observations

- If $\mathbf{v}$ is the vector from vertex 0 to vertex 1, then

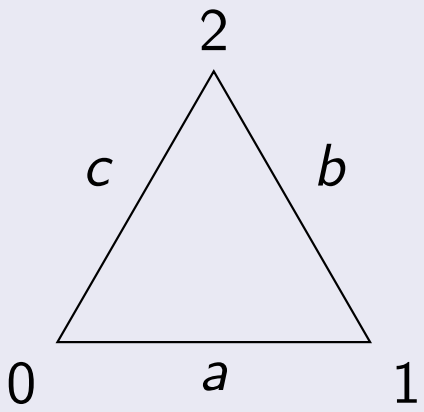
$$d\lambda_0(\mathbf{v}) = -1, \qquad d\lambda_1(\mathbf{v}) = 1, \qquad d\lambda_2(\mathbf{v}) = 0.$$

As desired:

$$g(\mathbf{v},\mathbf{v}) = -\tfrac{1}{2}a^2(-1-1) - \tfrac{1}{2}b^2(0+0) - \tfrac{1}{2}c^2(0+0) = a^2.$$

Regge metric on a reference triangle

Barycentric coordinates $\lambda_0 + \lambda_1 + \lambda_2 = 1$



Regge metric:

$$\begin{aligned} g = &-\tfrac{1}{2}a^2(d\lambda_0 \otimes d\lambda_1 + d\lambda_1 \otimes d\lambda_0) \\ &-\tfrac{1}{2}b^2(d\lambda_1 \otimes d\lambda_2 + d\lambda_2 \otimes d\lambda_1) \\ &-\tfrac{1}{2}c^2(d\lambda_2 \otimes d\lambda_0 + d\lambda_0 \otimes d\lambda_2) \end{aligned}$$

Observations

- If $\mathbf{v}$ is the vector from vertex 0 to vertex 1, then

$$d\lambda_0(\mathbf{v}) = -1, \qquad d\lambda_1(\mathbf{v}) = 1, \qquad d\lambda_2(\mathbf{v}) = 0.$$

  As desired:

$$g(\mathbf{v},\mathbf{v}) = -\tfrac{1}{2}a^2(-1-1) - \tfrac{1}{2}b^2(0+0) - \tfrac{1}{2}c^2(0+0) = a^2.$$

- Crucial: $-\frac{1}{2}a^2(d\lambda_0 \otimes d\lambda_1 + d\lambda_1 \otimes d\lambda_0)$ is zero on other edges.

Constant coefficient finite elements for bilinear forms

Geometrically decomposed bases for finite element spaces

- Each basis element φ must be associated to a face F of the triangulation, such that, for any other face G,

$$\varphi \text{ has nonzero trace on } G \Leftrightarrow G \geq F.$$

Constant coefficient finite elements for bilinear forms

Geometrically decomposed bases for finite element spaces

- Each basis element φ must be associated to a face F of the triangulation, such that, for any other face G,

$$\varphi \text{ has nonzero trace on } G \Leftrightarrow G \geq F.$$

Constant coefficient symmetric bilinear forms $\Lambda^{1,1}_{\text{sym}}$

- Regge's construction works in any dimension. To each edge ij, associate

$$d\lambda_i \odot d\lambda_j := d\lambda_i \otimes d\lambda_j + d\lambda_j \otimes d\lambda_i.$$

Constant coefficient finite elements for bilinear forms

Geometrically decomposed bases for finite element spaces

- Each basis element φ must be associated to a face F of the triangulation, such that, for any other face G,

$$\varphi \text{ has nonzero trace on } G \Leftrightarrow G \geq F.$$

Constant coefficient symmetric bilinear forms $\Lambda^{1,1}_{\text{sym}}$

- Regge's construction works in any dimension. To each edge ij, associate

$$d\lambda_i \odot d\lambda_j := d\lambda_i \otimes d\lambda_j + d\lambda_j \otimes d\lambda_i.$$

Constant coefficient antisymmetric bilinear forms $\Lambda^{1,1}_{\text{asym}}$

- Finite element spaces do not exist in dimension ≥ 3.
- In 3D, antisymmetric bilinear forms $\leftrightarrow$ vector fields with normal continuity.
- A nonzero constant vector field can't be tangent to three faces of a tetrahedron.

Affine-invariant subspaces of double forms

Theorem (Eigendecomposition of s^*s)

$$\Lambda^{p,q} = \bigoplus_m \Lambda^{p,q}_m, \qquad \max\{0, q-p\} \le m \le \min\{q, n-p\}.$$

Example

- $\Lambda^{1,1}_0$: Symmetric bilinear forms, $\varphi(X;Y) = \varphi(Y;X)$.
- $\Lambda^{1,1}_1$: Λ^2, antisymmetric bilinear forms, $\varphi(X;Y) = -\varphi(Y;X)$.

- $\Lambda^{2,1}_0$: spanned by $\alpha \otimes \beta$ such that $\alpha \wedge \beta = 0$.
  - Matrix proxy in 3D: trace-free matrices.
- $\Lambda^{2,1}_1$: Λ^3.
  - Matrix proxy in 3D: multiples of the identity matrix.

- $\Lambda^{2,2}_0$: Symmetric, satisfying the algebraic Bianchi identity.
  - Riemann curvature tensor.
- $\Lambda^{2,2}_1$: Antisymmetric, $\varphi(X,Y;Z,W) = -\varphi(Z,W;X,Y)$.
- $\Lambda^{2,2}_2$: Λ^4.

The decomposition respects the Bianchi sum operator s

$s\colon \Lambda_m^{p,q} \xrightarrow{\sim} \Lambda_{m-1}^{p-1,q-1}$

$$0 \xrightarrow{s} \Lambda^{0,4} \xrightarrow{s} \Lambda^{1,3} \xrightarrow{s} \Lambda^{2,2} \xrightarrow{s} \Lambda^{3,1} \xrightarrow{s} \Lambda^{4,0} \xrightarrow{s} 0$$

The decomposition respects the Bianchi sum operator s

$s\colon \Lambda_m^{p,q} \xrightarrow{\sim} \Lambda_{m-1}^{p-1,q-1}$

$$
\begin{array}{ccccccccccccc}
 & & & & & & \Lambda_0^{2,2} & & & & & & \\
 & & & & & & \oplus & & & & & & \\
 & & & & \Lambda_2^{1,3} & & \Lambda_1^{2,2} & & \Lambda_0^{3,1} & & & & \\
 & & & & \oplus & & \oplus & & \oplus & & & & \\
 & & \Lambda_4^{0,4} & & \Lambda_3^{1,3} & & \Lambda_2^{2,2} & & \Lambda_1^{3,1} & & \Lambda_0^{4,0} & & \\
 & & \wr\| & & \wr\| & & \wr\| & & \wr\| & & \wr\| & & \\
0 & \xrightarrow{s} & \Lambda^{0,4} & \xrightarrow{s} & \Lambda^{1,3} & \xrightarrow{s} & \Lambda^{2,2} & \xrightarrow{s} & \Lambda^{3,1} & \xrightarrow{s} & \Lambda^{4,0} & \xrightarrow{s} & 0
\end{array}
$$

The decomposition respects the Bianchi sum operator s

$s\colon \Lambda_m^{p,q} \xrightarrow{\sim} \Lambda_{m-1}^{p-1,q-1}$

$$
\begin{array}{ccccccccccccc}
 & & & & 0 & \xrightarrow{s} & \Lambda_0^{2,2} & \xrightarrow{s} & 0 & & & & \\
 & & & & & & \oplus & & & & & & \\
 & & 0 & \xrightarrow{s} & \Lambda_2^{1,3} & \xrightarrow[\sim]{s} & \Lambda_1^{2,2} & \xrightarrow[\sim]{s} & \Lambda_0^{3,1} & \xrightarrow{s} & 0 & & \\
 & & & & \oplus & & \oplus & & \oplus & & & & \\
0 & \xrightarrow{s} & \Lambda_4^{0,4} & \xrightarrow[\sim]{s} & \Lambda_3^{1,3} & \xrightarrow[\sim]{s} & \Lambda_2^{2,2} & \xrightarrow[\sim]{s} & \Lambda_1^{3,1} & \xrightarrow[\sim]{s} & \Lambda_0^{4,0} & \xrightarrow{s} & 0 \\
 & & \wr\| & & \wr\| & & \wr\| & & \wr\| & & \wr\| & & \\
0 & \xrightarrow{s} & \Lambda^{0,4} & \xrightarrow{s} & \Lambda^{1,3} & \xrightarrow{s} & \Lambda^{2,2} & \xrightarrow{s} & \Lambda^{3,1} & \xrightarrow{s} & \Lambda^{4,0} & \xrightarrow{s} & 0
\end{array}
$$

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.
- If $p < q$:

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.
- If $p < q$:
  - $s^*s = ss^* + (q-p)\,\mathrm{id}$ is positive definite.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.
- If $p < q$:
  - $s^*s = ss^* + (q-p)\,\mathrm{id}$ is positive definite.
  - So $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ is injective.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.
- If $p < q$:
  - $s^*s = ss^* + (q-p)\,\mathrm{id}$ is positive definite.
  - So $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ is injective.
  - So $s^*\colon \Lambda^{p+1,q-1} \to \Lambda^{p,q}$ is surjective.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.
- If $p < q$:
  - $s^*s = ss^* + (q-p)\,\mathrm{id}$ is positive definite.
  - So $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ is injective.
  - So $s^*\colon \Lambda^{p+1,q-1} \to \Lambda^{p,q}$ is surjective.
- Similarly, if $p > q$:

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.
- If $p < q$:
  - $s^*s = ss^* + (q-p)\,\mathrm{id}$ is positive definite.
  - So $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ is injective.
  - So $s^*\colon \Lambda^{p+1,q-1} \to \Lambda^{p,q}$ is surjective.
- Similarly, if $p > q$:
  - $ss^* = s^*s + (p-q)\,\mathrm{id}$ is positive definite.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.
- If $p < q$:
  - $s^*s = ss^* + (q-p)\,\mathrm{id}$ is positive definite.
  - So $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ is injective.
  - So $s^*\colon \Lambda^{p+1,q-1} \to \Lambda^{p,q}$ is surjective.
- Similarly, if $p > q$:
  - $ss^* = s^*s + (p-q)\,\mathrm{id}$ is positive definite.
  - $s^*\colon \Lambda^{p,q} \to \Lambda^{p-1,q+1}$ is injective.

Short proof of injectivity/surjectivity of the Bianchi sum s

Proposition

The operator $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ *is injective if* $p < q$ *and surjective if* $p+1 > q-1$.

Proof.

- Verify $ss^* - s^*s = (p-q)\,\mathrm{id}$ on $\Lambda^{p,q}$.
- ss^* is positive semidefinite.
- If $p < q$:
  - $s^*s = ss^* + (q-p)\,\mathrm{id}$ is positive definite.
  - So $s\colon \Lambda^{p,q} \to \Lambda^{p+1,q-1}$ is injective.
  - So $s^*\colon \Lambda^{p+1,q-1} \to \Lambda^{p,q}$ is surjective.
- Similarly, if $p > q$:
  - $ss^* = s^*s + (p-q)\,\mathrm{id}$ is positive definite.
  - $s^*\colon \Lambda^{p,q} \to \Lambda^{p-1,q+1}$ is injective.
  - $s\colon \Lambda^{p-1,q+1} \to \Lambda^{p,q}$ is surjective.

Finite element spaces

Theorem (—, Gawlik)

Apart from $\Lambda_q^{p,q} \cong \Lambda^{p+q}$ with constant coefficients, there is a finite element space for every natural space of double forms $\Lambda_m^{p,q}$ with polynomial coefficients of any degree (including zero).

Example (Constant coefficient spaces)

- $\Lambda_0^{1,1}$: symmetric matrices with tangential–tangential continuity (Regge, 1961; Christiansen, 2004).
  - Higher order: (Li, 2018).
- $\Lambda_0^{2,1}$ in 3D: trace-free matrices with normal–tangential continuity (Gopalakrishnan, Lederer, and Schöberl, 2019).
- $\Lambda_0^{2,2}$ in 3D: symmetric matrices with normal–normal continuity (Pechstein and Schöberl, 2011).
- $\Lambda_0^{2,2}$ (or $\Lambda_0^{n-2,n-2}$) in any dimension: finite elements for the Riemann curvature tensor.

Degrees of freedom for constant coefficient spaces

	d						
	0	1	2	3	4	5	6
$\Lambda_0^{1,1}$	0	**1**	0	0	0	0	0
$\Lambda_0^{2,1}$	0	0	**2**	0	0	0	0
$\Lambda_0^{2,2}$	0	0	**1**	**2**	0	0	0
$\Lambda_1^{2,2} \cong \Lambda_0^{3,1}$	0	0	0	**3**	0	0	0
$\Lambda_0^{3,2}$	0	0	0	**3**	**5**	0	0
$\Lambda_1^{3,2} \cong \Lambda_0^{4,1}$	0	0	0	0	**4**	0	0
$\Lambda_0^{3,3}$	0	0	0	**1**	**5**	**5**	0
$\Lambda_1^{3,3} \cong \Lambda_0^{4,2}$	0	0	0	0	**6**	**9**	0
$\Lambda_2^{3,3} \cong \Lambda_1^{4,2} \cong \Lambda_0^{5,1}$	0	0	0	0	0	**5**	0

Table: Number of degrees of freedom for $\Lambda_m^{p,q}$ associated to a face of the triangulation of dimension d is $\frac{p-q+2m+1}{p+m+1}\binom{d+1}{q-m}\binom{q-m-1}{d-p-m}$.

More on algebraic curvature tensors $\Lambda_0^{2,2}$

Joint with Lily DiPaulo

The space $\Lambda_0^{2,2}$

- Symmetric $(2,2)$-forms satisfying the Bianchi identity.
- $\Lambda_0^{2,2}$ is spanned by $\alpha \odot \beta$ where $\alpha, \beta \in \Lambda^2$ and $\alpha \wedge \beta = 0$.

Finite element spaces

- Construct bases for constant coefficient spaces using (—, Gawlik)
- Generalize to higher order similarly to Li's work on Regge finite elements.

Regge finite elements $\mathcal{P}_r\Lambda_0^{1,1}$ (symmetric bilinear forms)

Constant coefficient space $\Lambda_0^{1,1}$

- For i and j distinct vertices, associate $d\lambda_i \odot d\lambda_j$ to edge ij.
- These forms are a basis for the space $\Lambda_0^{1,1}$ of symmetric bilinear forms with constant coefficients.

Higher order spaces $\mathcal{P}_r\Lambda_0^{1,1}$ (Li)

- For a multiindex I, let λ^I be the corresponding monomial, and let $\operatorname{supp} I$ denote the set of vertices whose corresponding exponent is at least one in λ^I.
  - e.g. if $\lambda^I = \lambda_0^5\lambda_3^4$ then $\operatorname{supp} I = \{0, 3\}$.
- Associate $\lambda^I\, d\lambda_i \odot d\lambda_j$ to the face with vertices $\{i, j\} \cup \operatorname{supp} I$.
- These forms are a basis for $\mathcal{P}_r\Lambda_0^{1,1}$ because the monomials are a basis for $\mathcal{P}_r$ and the $d\lambda_i \odot d\lambda_j$ are a basis for $\Lambda_0^{1,1}$.

Constant coefficient space $\Lambda_0^{2,2}$

- Let $d\lambda_{ij} := d\lambda_i \wedge d\lambda_j$.
- To each two-dimensional face ijk, associate

$$\beta_{ijk} := d\lambda_{ij} \odot d\lambda_{jk} + d\lambda_{jk} \odot d\lambda_{ki} + d\lambda_{ki} \odot d\lambda_{ij}$$

- To each three-dimensional face $ijkl$, associate

$$\gamma_{iklj} := d\lambda_{il} \odot d\lambda_{jk} - d\lambda_{ij} \odot d\lambda_{kl},$$
$$\gamma_{iljk} := d\lambda_{ij} \odot d\lambda_{kl} - d\lambda_{ik} \odot d\lambda_{lj}.$$

- These forms are a basis for the space $\Lambda_0^{2,2}$ of algebraic curvature tensors with constant coefficients.
- These formulas can be derived from the representation theory of the symmetric group (Young diagrams), following (—, Gawlik).

Higher order algebraic curvature tensors $\mathcal{P}_r\Lambda_0^{2,2}$

Constant coefficient space $\Lambda_0^{2,2}$

$$\begin{aligned}\beta_{ijk} &:= d\lambda_{ij} \odot d\lambda_{jk} + d\lambda_{jk} \odot d\lambda_{ki} + d\lambda_{ki} \odot d\lambda_{ij},\\ \gamma_{iklj} &:= d\lambda_{il} \odot d\lambda_{jk} - d\lambda_{ij} \odot d\lambda_{kl},\\ \gamma_{iljk} &:= d\lambda_{ij} \odot d\lambda_{kl} - d\lambda_{ik} \odot d\lambda_{lj}.\end{aligned}$$

Higher order space $\mathcal{P}_r\Lambda_0^{2,2}$

- Associate $\lambda^I\beta_{ijk}$ to the face with vertices $\{i,j,k\} \cup \operatorname{supp} I$.
- Associate $\lambda^I\gamma_{iklj}$ and $\lambda^I\gamma_{iljk}$ to the face with vertices $\{i,j,k,l\} \cup \operatorname{supp} I$.
- These forms are a geometrically decomposed basis for $\mathcal{P}_r\Lambda_0^{2,2}$.

What's next?

Explicit bases for arbitrary double forms

What's next?

Explicit bases for arbitrary double forms

- Use representation theory to write down explicit bases for $\Lambda_0^{p,q}$ for general p and q.

What's next?

Explicit bases for arbitrary double forms

- Use representation theory to write down explicit bases for $\Lambda_0^{p,q}$ for general p and q.
- Obtain geometrically decomposed bases for $\mathcal{P}_r\Lambda_0^{p,q}$ as above.

What's next?

Explicit bases for arbitrary double forms

- Use representation theory to write down explicit bases for $\Lambda_0^{p,q}$ for general p and q.
- Obtain geometrically decomposed bases for $\mathcal{P}_r\Lambda_0^{p,q}$ as above.

Arbitrary covariant tensors with arbitrary symmetries

What's next?

Explicit bases for arbitrary double forms

- Use representation theory to write down explicit bases for $\Lambda_0^{p,q}$ for general p and q.
- Obtain geometrically decomposed bases for $\mathcal{P}_r\Lambda_0^{p,q}$ as above.

Arbitrary covariant tensors with arbitrary symmetries

- Given any affine-invariant space of covariant tensors, one can decompose it into irreducible representations.

What's next?

Explicit bases for arbitrary double forms

- Use representation theory to write down explicit bases for $\Lambda_0^{p,q}$ for general p and q.
- Obtain geometrically decomposed bases for $\mathcal{P}_r\Lambda_0^{p,q}$ as above.

Arbitrary covariant tensors with arbitrary symmetries

- Given any affine-invariant space of covariant tensors, one can decompose it into irreducible representations.
- Any irreducible representation is isomorphic to a subspace of double forms or triple forms or quadruple forms etc.

What's next?

Explicit bases for arbitrary double forms

- Use representation theory to write down explicit bases for $\Lambda_0^{p,q}$ for general p and q.
- Obtain geometrically decomposed bases for $\mathcal{P}_r\Lambda_0^{p,q}$ as above.

Arbitrary covariant tensors with arbitrary symmetries

- Given any affine-invariant space of covariant tensors, one can decompose it into irreducible representations.
- Any irreducible representation is isomorphic to a subspace of double forms or triple forms or quadruple forms etc.
- So, constructing finite element spaces for triple forms, quadruple forms, etc., actually gives us all covariant tensors.

What's next?

Explicit bases for arbitrary double forms

- Use representation theory to write down explicit bases for $\Lambda_0^{p,q}$ for general p and q.
- Obtain geometrically decomposed bases for $\mathcal{P}_r\Lambda_0^{p,q}$ as above.

Arbitrary covariant tensors with arbitrary symmetries

- Given any affine-invariant space of covariant tensors, one can decompose it into irreducible representations.
- Any irreducible representation is isomorphic to a subspace of double forms or triple forms or quadruple forms etc.
- So, constructing finite element spaces for triple forms, quadruple forms, etc., actually gives us all covariant tensors.

Distributional Riemann curvature, joint with Gawlik and Neunteufel

What's next?

Explicit bases for arbitrary double forms

- Use representation theory to write down explicit bases for $\Lambda_0^{p,q}$ for general p and q.
- Obtain geometrically decomposed bases for $\mathcal{P}_r\Lambda_0^{p,q}$ as above.

Arbitrary covariant tensors with arbitrary symmetries

- Given any affine-invariant space of covariant tensors, one can decompose it into irreducible representations.
- Any irreducible representation is isomorphic to a subspace of double forms or triple forms or quadruple forms etc.
- So, constructing finite element spaces for triple forms, quadruple forms, etc., actually gives us all covariant tensors.

Distributional Riemann curvature, joint with Gawlik and Neunteufel

- The natural test function space for the distributional Riemann curvature is $\Lambda_0^{n-2,n-2}$.

Thank you

Yakov Berchenko-Kogan and Evan S. Gawlik

Blow-up Whitney forms, shadow forms, and Poisson processes.
Results in Applied Mathematics, special issue on Hilbert complexes, Paper No. 100529, 2025.

J. P. Brasselet, M. Goresky, and R. MacPherson.

Simplicial differential forms with poles.
Amer. J. Math., 113(6):1019–1052, 1991.

Yakov Berchenko-Kogan and Evan S. Gawlik

Finite element spaces of double forms.
https://arxiv.org/abs/2505.17243

Yakov Berchenko-Kogan and Lily DiPaulo.

Finite element spaces of double two-forms with polynomial coefficients.
https://arxiv.org/abs/2511.19297.

Yakov Berchenko-Kogan

Duality in finite element exterior calculus and Hodge duality on the sphere.
Found. Comput. Math. 21(5):1153–1180, 2021.

Evan S. Gawlik and Anil N. Hirani

Sequences from sequences, sans coordinates.
In preparation.

Supported by NSF DMS-2411209.