From the Mimetic Finite Difference method to the Virtual Element Method

Gianmarco Manzini

Los Alamos National Laboratory
New Mexico, USA

July 27, 2012 - Columbia University

Outline

1 the Virtual Element Method (VEM) for the Laplace operator:

- the degrees of freedom and the local Virtual Element (VE) space;
- the abstract VE formulation;
- the convergence theorem; consistency, stability;
- the mimetic approximation of the VE bilinear form;
- high-order and high-regular extensions.

2. A numerical experiment.

3. Final remarks, future work.

The linear diffusion problem

- Differential formulation:

$$
\begin{aligned}
-\nabla u &= f \quad \text{in } \Omega, \\
u &= g \quad \text{on } \Gamma,
\end{aligned}
$$

- Variational formulation:

Find $u \in H^1_g(\Omega)$ such that:

$$
\int_\Omega \nabla u \cdot \nabla v \, dV = \int_\Omega fv \, dV \qquad \forall v \in H^1_0(\Omega),
$$

People and References

- People:
  - the "Pavia team": L. Beirão da Veiga, F. Brezzi, A. Cangiani, D. Marini, A. Russo;
  - the "Los Alamos team": K. Lipnikov, D. Svyatskiy, M. Shashkov;
- Papers:
  1. *F. Brezzi, A. Buffa, K. Lipnikov*, M2AN (2009): the low-order node-based MFD;
  2. *L. Beirão da Veiga, K. Lipnikov, G. Manzini*, SINUM (2011): the arbitrary-order node-based MFD;
  3. ***the "six-name paper"***, M3AS (to appear in January 2013): basic principles of VEM; abstract formulation

People and References

- People:
  - the "Pavia team": L. Beirão da Veiga, F. Brezzi, A. Cangiani, D. Marini, A. Russo;
  - the "Los Alamos team": K. Lipnikov, D. Svyatskiy, M. Shashkov;
- Papers:
  1. *F. Brezzi, A. Buffa, K. Lipnikov*, M2AN (2009): the low-order node-based MFD;
  2. *L. Beirão da Veiga, K. Lipnikov, G. Manzini*, SINUM (2011): the arbitrary-order node-based MFD;
  3. ***the "six-name paper"***, M3AS (to appear in January 2013): basic principles of VEM; abstract formulation

People and References

- People:
  - the ”Pavia team”: L. Beirão da Veiga, F. Brezzi, A. Cangiani, D. Marini, A. Russo;
  - the ”Los Alamos team”: K. Lipnikov, D. Svyatskiy, M. Shashkov;
- Papers:
  1. *F. Brezzi, A. Buffa, K. Lipnikov*, M2AN (2009): the low-order node-based MFD;
  2. *L. Beirão da Veiga, K. Lipnikov, G. Manzini*, SINUM (2011): the arbitrary-order node-based MFD;
  3. ***the ”six-name paper”***, M3AS (to appear in January 2013): basic principles of VEM; abstract formulation

People and References

- People:
  - the ”Pavia team”: L. Beirão da Veiga, F. Brezzi, A. Cangiani, D. Marini, A. Russo;
  - the ”Los Alamos team”: K. Lipnikov, D. Svyatskiy, M. Shashkov;
- Papers:
  1. *F. Brezzi, A. Buffa, K. Lipnikov*, M2AN (2009): the low-order node-based MFD;
  2. *L. Beirão da Veiga, K. Lipnikov, G. Manzini*, SINUM (2011): the arbitrary-order node-based MFD;
  3. ***the ”six-name paper”***, M3AS (to appear in January 2013): basic principles of VEM; abstract formulation

People and References

- People:
  - the ”Pavia team”: L. Beirão da Veiga, F. Brezzi, A. Cangiani, D. Marini, A. Russo;
  - the ”Los Alamos team”: K. Lipnikov, D. Svyatskiy, M. Shashkov;
- Papers:
  1. *F. Brezzi, A. Buffa, K. Lipnikov*, M2AN (2009): the low-order node-based MFD;
  2. *L. Beirão da Veiga, K. Lipnikov, G. Manzini*, SINUM (2011): the arbitrary-order node-based MFD;
  3. ***the ”six-name paper”***, M3AS (to appear in January 2013): basic principles of VEM; abstract formulation

The Virtual Element approach

- The ***Virtual Element*** approach for the ***Mimetic Finite Difference*** (MFD) method is based on a ***local finite element space*** $\mathcal{V}_{h,\mathrm{P}}$ on P such that:
  - the degrees of freedom are the vertex values; $\dim \mathcal{V}_{h,\mathrm{P}} = N^{\mathcal{V}}_{\mathrm{P}}$;
  - on triangles $\mathcal{V}_{h,\mathrm{P}}$ must be the linear Galerkin finite element space
    $\Longrightarrow$ $\mathcal{V}_{h,\mathrm{P}}$ must contain the linear polynomials $1, x, y$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ *glue gracefully* to give a conformal global finite element space $\mathcal{V}_h$.

- *Remarks:*
  - we will specify the behavior of the functions of $\mathcal{V}_{h,\mathrm{P}}$ on $\partial\mathrm{P}$, the boundary of P;
  - we will not ask to be able to *compute* the functions of $\mathcal{V}_{h,\mathrm{P}}$!

The Virtual Element approach

- The ***Virtual Element*** approach for the ***Mimetic Finite Difference*** (MFD) method is based on a ***local finite element space*** $\mathcal{V}_{h,\mathrm{P}}$ on P such that:
  - the degrees of freedom are the vertex values; $\mathrm{dim}\mathcal{V}_{h,\mathrm{P}} = N_{\mathrm{P}}^{\mathcal{V}}$;
  - on triangles $\mathcal{V}_{h,\mathrm{P}}$ must be the linear Galerkin finite element space $\Longrightarrow$ $\mathcal{V}_{h,\mathrm{P}}$ must contain the linear polynomials $1, x, y$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ *glue gracefully* to give a conformal global finite element space $\mathcal{V}_h$.
- *Remarks:*
  - we will specify the behavior of the functions of $\mathcal{V}_{h,\mathrm{P}}$ on $\partial\mathrm{P}$, the boundary of P;
  - we will not ask to be able to *compute* the functions of $\mathcal{V}_{h,\mathrm{P}}$!

The Virtual Element approach

- The ***Virtual Element*** approach for the ***Mimetic Finite Difference*** (MFD) method is based on a ***local finite element space*** $\mathcal{V}_{h,\mathrm{P}}$ on P such that:
  - the degrees of freedom are the vertex values; $\dim\mathcal{V}_{h,\mathrm{P}} = N_{\mathrm{P}}^{\mathcal{V}}$;
  - on triangles $\mathcal{V}_{h,\mathrm{P}}$ must be the linear Galerkin finite element space
    $\Rightarrow$ $\mathcal{V}_{h,\mathrm{P}}$ must contain the linear polynomials 1, x, y;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ *glue gracefully* to give a conformal global finite element space $\mathcal{V}_h$.

- *Remarks:*
  - we will specify the behavior of the functions of $\mathcal{V}_{h,\mathrm{P}}$ on $\partial\mathrm{P}$, the boundary of P;
  - we will not ask to be able to *compute* the functions of $\mathcal{V}_{h,\mathrm{P}}$!

The Virtual Element approach

- The ***Virtual Element*** approach for the ***Mimetic Finite Difference*** (MFD) method is based on a ***local finite element space*** $\mathcal{V}_{h,\mathrm{P}}$ on P such that:
  - the degrees of freedom are the vertex values; $\dim \mathcal{V}_{h,\mathrm{P}} = N^{\mathcal{V}}_{\mathrm{P}}$;
  - on triangles $\mathcal{V}_{h,\mathrm{P}}$ must be the linear Galerkin finite element space
    $\Rightarrow$ $\mathcal{V}_{h,\mathrm{P}}$ must contain the linear polynomials 1, x, y;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ *glue gracefully* to give a conformal global finite element space $\mathcal{V}_h$.

- *Remarks:*
  - we will specify the behavior of the functions of $\mathcal{V}_{h,\mathrm{P}}$ on $\partial\mathrm{P}$, the boundary of P;
  - we will not ask to be able to *compute* the functions of $\mathcal{V}_{h,\mathrm{P}}$!

The Virtual Element approach

- The ***Virtual Element*** approach for the ***Mimetic Finite Difference*** (MFD) method is based on a ***local finite element space*** $\mathcal{V}_{h,\mathrm{P}}$ on P such that:
  - the degrees of freedom are the vertex values; $\dim\mathcal{V}_{h,\mathrm{P}} = N^{\mathcal{V}}_{\mathrm{P}}$;
  - on triangles $\mathcal{V}_{h,\mathrm{P}}$ must be the linear Galerkin finite element space
    $\Rightarrow$ $\mathcal{V}_{h,\mathrm{P}}$ must contain the linear polynomials 1, x, y;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ *glue gracefully* to give a conformal global finite element space $\mathcal{V}_h$.
- *Remarks:*
  - we will specify the behavior of the functions of $\mathcal{V}_{h,\mathrm{P}}$ on ∂P, the boundary of P;
  - we will not ask to be able to *compute* the functions of $\mathcal{V}_{h,\mathrm{P}}$!

The Virtual Element approach

- The ***Virtual Element*** approach for the ***Mimetic Finite Difference*** (MFD) method is based on a ***local finite element space*** $\mathcal{V}_{h,\mathrm{P}}$ on P such that:
  - the degrees of freedom are the vertex values; $\mathrm{dim}\mathcal{V}_{h,\mathrm{P}} = N_{\mathrm{P}}^{\mathcal{V}}$;
  - on triangles $\mathcal{V}_{h,\mathrm{P}}$ must be the linear Galerkin finite element space
    $\Rightarrow$ $\mathcal{V}_{h,\mathrm{P}}$ must contain the linear polynomials 1, x, y;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ *glue gracefully* to give a conformal global finite element space $\mathcal{V}_h$.

- *Remarks:*
  - we will specify the behavior of the functions of $\mathcal{V}_{h,\mathrm{P}}$ on $\partial\mathrm{P}$, the boundary of P;
  - we will not ask to be able to *compute* the functions of $\mathcal{V}_{h,\mathrm{P}}$!

The Virtual Element approach

- The ***Virtual Element*** approach for the ***Mimetic Finite Difference*** (MFD) method is based on a ***local finite element space*** $\mathcal{V}_{h,\mathrm{P}}$ on P such that:
  - the degrees of freedom are the vertex values; $\dim\mathcal{V}_{h,\mathrm{P}} = N_{\mathrm{P}}^{\mathcal{V}}$;
  - on triangles $\mathcal{V}_{h,\mathrm{P}}$ must be the linear Galerkin finite element space
    $\Rightarrow$ $\mathcal{V}_{h,\mathrm{P}}$ must contain the linear polynomials 1, x, y;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ *glue gracefully* to give a conformal global finite element space $\mathcal{V}_h$.

- *Remarks:*
  - we will specify the behavior of the functions of $\mathcal{V}_{h,\mathrm{P}}$ on ∂P, the boundary of P;
  - we will not ask to be able to *compute* the functions of $\mathcal{V}_{h,\mathrm{P}}$!

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on $\partial\mathrm{P}$ such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on $\partial\mathrm{P}$ such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on $\partial\mathrm{P}$ such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on ∂P such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on $\partial\mathrm{P}$ such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on $\partial\mathrm{P}$ such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on $\partial\mathrm{P}$ such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on $\partial\mathrm{P}$ such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The local finite element space

We define the local finite element space $\mathcal{V}_{h,\mathrm{P}}$ through a basis.

For each vertex v_i we define a function $\varphi_i \in H^1(P)$:

1. let δ_i be the function defined on $\partial\mathrm{P}$ such that:
   - $\delta_i(\mathrm{v}_j) = 1$ if $i = j$, and 0 otherwise;
   - δ_i is continuous;
   - δ_i is linear on each edge.
2. we set: $\varphi_{i|\partial\mathrm{P}} = \delta_i$;
3. we **formally** extend $\varphi_{i|\partial\mathrm{P}}$ inside P by the **harmonic lifting**:

   $\Rightarrow$ the functions φ_i are uniquely determined by the corresponding δ_i (we can prove the unisolvency!)

Eventually, we set: $\mathcal{V}_{h,\mathrm{P}} := \mathrm{span}\{\varphi_1, \ldots, \phi_{N^{\mathcal{P}}}\}$.

The harmonic lifting

- φ_i is the harmonic function on P having δ_i as boundary value:

$$\begin{cases} -\Delta\varphi_i = 0 & \text{in } \Omega \\ \varphi_i = \delta_i & \text{on } \partial\Omega. \end{cases}$$

  - the functions $\{\varphi_i\}$ are linearly independent;
  - if $w_h \in \mathcal{V}_{h,\mathrm{P}}$, then $w_h = \sum_{i=1}^{N^{\mathcal{P}}} w_h(\mathrm{v}_i)\,\varphi_i$;
  - $1, x, y \in \mathcal{V}_{h,\mathrm{P}}$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ glue together giving a conformal finite element space $\mathcal{V}_h \subset H^1_0(\Omega)$.

- *Remarks:*
  - if P is a triangle, we recover the $\mathbb{P}_1$ Galerkin elements;
  - if P is a parallelogram, we recover the $\mathbb{Q}_1$ bilinear elements.

The harmonic lifting

- φ_i is the harmonic function on P having δ_i as boundary value:

$$\begin{cases} -\Delta\varphi_i = 0 & \text{in } \Omega \\ \varphi_i = \delta_i & \text{on } \partial\Omega. \end{cases}$$

  - the functions $\{\varphi_i\}$ are linearly independent;
  - if $w_h \in \mathcal{V}_{h,\mathrm{P}}$, then $w_h = \sum_{i=1}^{N^{\mathcal{P}}} w_h(\mathrm{v}_i)\,\varphi_i$;
  - $1, x, y \in \mathcal{V}_{h,\mathrm{P}}$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ glue together giving a conformal finite element space $\mathcal{V}_h \subset H^1_0(\Omega)$.

- *Remarks:*
  - if P is a triangle, we recover the $\mathbb{P}_1$ Galerkin elements;
  - if P is a parallelogram, we recover the $\mathbb{Q}_1$ bilinear elements.

The harmonic lifting

- φ_i is the harmonic function on P having δ_i as boundary value:

$$\begin{cases} -\Delta\varphi_i = 0 & \text{in } \Omega \\ \varphi_i = \delta_i & \text{on } \partial\Omega. \end{cases}$$

  - the functions $\{\varphi_i\}$ are linearly independent;
  - if $w_h \in \mathcal{V}_{h,\mathrm{P}}$, then $w_h = \sum_{i=1}^{N^{\mathcal{P}}} w_h(\mathrm{v}_i)\,\varphi_i$;
  - $1, x, y \in \mathcal{V}_{h,\mathrm{P}}$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ glue together giving a conformal finite element space $\mathcal{V}_h \subset H^1_0(\Omega)$.

- *Remarks:*
  - if P is a triangle, we recover the $\mathbb{P}_1$ Galerkin elements;
  - if P is a parallelogram, we recover the $\mathbb{Q}_1$ bilinear elements.

The harmonic lifting

- φ_i is the harmonic function on P having δ_i as boundary value:

$$\begin{cases} -\Delta\varphi_i = 0 & \text{in } \Omega \\ \varphi_i = \delta_i & \text{on } \partial\Omega. \end{cases}$$

  - the functions $\{\varphi_i\}$ are linearly independent;
  - if $w_h \in \mathcal{V}_{h,\mathrm{P}}$, then $w_h = \sum_{i=1}^{N^{\mathcal{P}}} w_h(\mathrm{v}_i)\,\varphi_i$;
  - $1, x, y \in \mathcal{V}_{h,\mathrm{P}}$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ glue together giving a conformal finite element space $\mathcal{V}_h \subset H^1_0(\Omega)$.

- *Remarks:*
  - if P is a triangle, we recover the $\mathbb{P}_1$ Galerkin elements;
  - if P is a parallelogram, we recover the $\mathbb{Q}_1$ bilinear elements.

The harmonic lifting

- φ_i is the harmonic function on P having δ_i as boundary value:

$$\begin{cases} -\Delta\varphi_i = 0 & \text{in } \Omega \\ \varphi_i = \delta_i & \text{on } \partial\Omega. \end{cases}$$

  - the functions $\{\varphi_i\}$ are linearly independent;
  - if $w_h \in \mathcal{V}_{h,\mathrm{P}}$, then $w_h = \sum_{i=1}^{N^{\mathcal{P}}} w_h(\mathrm{v}_i)\,\varphi_i$;
  - $1, x, y \in \mathcal{V}_{h,\mathrm{P}}$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ glue together giving a conformal finite element space $\mathcal{V}_h \subset H^1_0(\Omega)$.

- *Remarks:*
  - if P is a triangle, we recover the $\mathbb{P}_1$ Galerkin elements;
  - if P is a parallelogram, we recover the $\mathbb{Q}_1$ bilinear elements.

The harmonic lifting

- φ_i is the harmonic function on P having δ_i as boundary value:

$$\begin{cases} -\Delta\varphi_i = 0 & \text{in } \Omega \\ \varphi_i = \delta_i & \text{on } \partial\Omega. \end{cases}$$

  - the functions $\{\varphi_i\}$ are linearly independent;
  - if $w_h \in \mathcal{V}_{h,\mathrm{P}}$, then $w_h = \sum_{i=1}^{N^{\mathcal{P}}} w_h(\mathrm{v}_i)\,\varphi_i$;
  - $1, x, y \in \mathcal{V}_{h,\mathrm{P}}$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ glue together giving a conformal finite element space $\mathcal{V}_h \subset H^1_0(\Omega)$.

- *Remarks:*
  - if P is a triangle, we recover the $\mathbb{P}_1$ Galerkin elements;
  - if P is a parallelogram, we recover the $\mathbb{Q}_1$ bilinear elements.

The harmonic lifting

- φ_i is the harmonic function on P having δ_i as boundary value:

$$\begin{cases} -\Delta\varphi_i = 0 & \text{in } \Omega \\ \varphi_i = \delta_i & \text{on } \partial\Omega. \end{cases}$$

  - the functions $\{\varphi_i\}$ are linearly independent;
  - if $w_h \in \mathcal{V}_{h,\mathrm{P}}$, then $w_h = \sum_{i=1}^{N^{\mathcal{P}}} w_h(\mathrm{v}_i)\,\varphi_i$;
  - $1, x, y \in \mathcal{V}_{h,\mathrm{P}}$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ glue together giving a conformal finite element space $\mathcal{V}_h \subset H^1_0(\Omega)$.

- *Remarks:*
  - if P is a triangle, we recover the $\mathbb{P}_1$ Galerkin elements;
  - if P is a parallelogram, we recover the $\mathbb{Q}_1$ bilinear elements.

The harmonic lifting

- φ_i is the harmonic function on P having δ_i as boundary value:

$$\begin{cases} -\Delta\varphi_i = 0 & \text{in } \Omega \\ \varphi_i = \delta_i & \text{on } \partial\Omega. \end{cases}$$

  - the functions $\{\varphi_i\}$ are linearly independent;
  - if $w_h \in \mathcal{V}_{h,\mathrm{P}}$, then $w_h = \sum_{i=1}^{N^{\mathcal{P}}} w_h(\mathrm{v}_i)\,\varphi_i$;
  - $1, x, y \in \mathcal{V}_{h,\mathrm{P}}$;
  - the local spaces $\mathcal{V}_{h,\mathrm{P}}$ glue together giving a conformal finite element space $\mathcal{V}_h \subset H^1_0(\Omega)$.

- *Remarks:*
  - if P is a triangle, we recover the $\mathbb{P}_1$ Galerkin elements;
  - if P is a parallelogram, we recover the $\mathbb{Q}_1$ bilinear elements.

The Harmonic Finite Element Method

The Harmonic Finite Element approximation of our elliptic problems is formally given by:

Find $u_h \in \mathcal{V}_h$ such that

$$\mathcal{A}(u_h, v_h) = F_h(v_h) \quad \text{for all } v_h \in \mathcal{V}_h$$

where (as usual)

$$\mathcal{A}(u_h, v_h) = \int_\Omega \nabla u_h \cdot \nabla v_h$$

and $F_h(v_h)$ is a suitable (and computable!) approximation of $\int_\Omega fv$ (that uses only the vertex values of v_h and f).

The Harmonic Finite Element Method

The Harmonic Finite Element approximation of our elliptic problems is formally given by:

Find $u_h \in \mathcal{V}_h$ such that

$$\mathcal{A}\big(u_h, v_h\big) = F_h(v_h) \quad \text{for all } v_h \in \mathcal{V}_h$$

where (as usual)

$$\mathcal{A}\big(u_h, v_h\big) = \int_\Omega \nabla u_h \cdot \nabla v_h$$

and $F_h(v_h)$ is a suitable (and computable!) approximation of $\displaystyle\int_\Omega fv$ (that uses only the vertex values of v_h and f).

The Harmonic Finite Element Method

The Harmonic Finite Element approximation of our elliptic problems is formally given by:

Find $u_h \in \mathcal{V}_h$ such that

$$\mathcal{A}\big(u_h, v_h\big) = F_h(v_h) \quad \text{for all } v_h \in \mathcal{V}_h$$

where (as usual)

$$\mathcal{A}\big(u_h, v_h\big) = \int_\Omega \nabla u_h \cdot \nabla v_h$$

and $F_h(v_h)$ is a suitable (and computable!) approximation of $\int_\Omega fv$ (that uses only the vertex values of v_h and f).

The Harmonic Finite Element Method

Now, we are very happy, because…

- …under *reasonable assumptions on the mesh*, the harmonic finite element approximation of an elliptic problem using the harmonic space $\mathcal{V}_h$ enjoys the usual convergence properties!

- *Which assumptions?*
  - all geometric objects must scale properly: $|P| \simeq h^2$, $|e| \simeq h$;
  - each polygon is star-shaped (or the union of a *uniformly bounded* number of star-shaped subcells) with respect to an internal ball of points (see Brenner-Scott, etc);
  - …

The Harmonic Finite Element Method

Now, we are very happy, because...

- ...under *reasonable assumptions on the mesh*, the harmonic finite element approximation of an elliptic problem using the harmonic space $\mathcal{V}_h$ enjoys the usual convergence properties!

- *Which assumptions?*
  - all geometric objects must scale properly: $|P| \simeq h^2$, $|e| \simeq h$;
  - each polygon is star-shaped (or the union of a *uniformly bounded* number of star-shaped subcells) with respect to an internal ball of points (see Brenner-Scott, etc);
  - ...

The Harmonic Finite Element Method

Now, we are very happy, because...

- ...under *reasonable assumptions on the mesh*, the harmonic finite element approximation of an elliptic problem using the harmonic space $\mathcal{V}_h$ enjoys the usual convergence properties!

- *Which assumptions?*
  - all geometric objects must scale properly: $|\mathrm{P}| \simeq h^2$, $|\mathrm{e}| \simeq h$;
  - each polygon is star-shaped (or the union of a *uniformly bounded* number of star-shaped subcells) with respect to an internal ball of points (see Brenner-Scott, etc);
  - ...

The Harmonic Finite Element Method

Now, we are very happy, because. . .

- . . . under *reasonable assumptions on the mesh*, the harmonic finite element approximation of an elliptic problem using the harmonic space $\mathcal{V}_h$ enjoys the usual convergence properties!

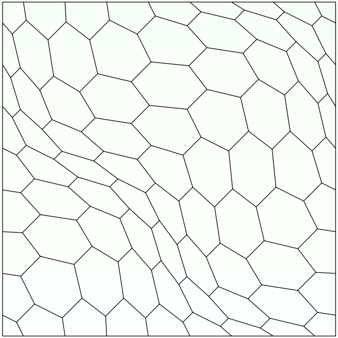
- *Which assumptions?*
  - all geometric objects must scale properly: $|\mathrm{P}| \simeq h^2$, $|\mathrm{e}| \simeq h$;
  - each polygon is star-shaped (or the union of a *uniformly bounded* number of star-shaped subcells) with respect to an internal ball of points (see Brenner-Scott, etc);
  - . . .

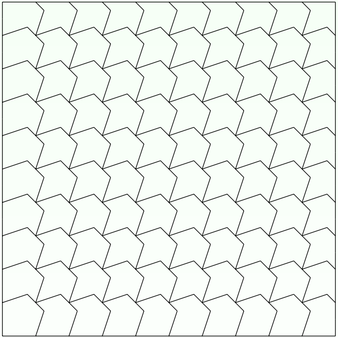
The Harmonic Finite Element Method

Now, we are very happy, because...

- ...under *reasonable assumptions on the mesh*, the harmonic finite element approximation of an elliptic problem using the harmonic space $\mathcal{V}_h$ enjoys the usual convergence properties!
- *Which assumptions?*
  - all geometric objects must scale properly: $|\mathrm{P}| \simeq h^2$, $|\mathrm{e}| \simeq h$;
  - each polygon is star-shaped (or the union of a *uniformly bounded* number of star-shaped subcells) with respect to an internal ball of points (see Brenner-Scott, etc);
  - ...

Polygonal meshes
Examples: convex and non-convex polygonal cells





The Virtual Element Method

- → So, we have a very nice method that works on polygonal meshes with very general shapes (also non-convex cells) and with a solid mathematical foundation (a priori error estimates, etc);
- → we can also extend it to higher order polynomials (considering additional degrees of freedom)...

...BUT...

- → **...if we do not know how to compute explicitly the basis functions...**
- → **...we don't know how to compute the stiffness matrix**

$$\mathcal{A}(\varphi_i, \varphi_j) = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$$

and the right-hand side $F_h(\mathbf{v}_h)$!

- → **Here, the mimetic technology comes into play!**

The Virtual Element Method

- → So, we have a very nice method that works on polygonal meshes with very general shapes (also non-convex cells) and with a solid mathematical foundation (a priori error estimates, etc);
- → we can also extend it to higher order polynomials (considering additional degrees of freedom)...

...BUT...

- → **...if we do not know how to compute explicitly the basis functions...**
- → **...we don't know how to compute the stiffness matrix**

$$\mathcal{A}(\varphi_i, \varphi_j) = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$$

and the right-hand side $F_h(\mathbf{v}_h)$!

- → **Here, the mimetic technology comes into play!**

The Virtual Element Method

- $\longrightarrow$ So, we have a very nice method that works on polygonal meshes with very general shapes (also non-convex cells) and with a solid mathematical foundation (a priori error estimates, etc);
- $\longrightarrow$ we can also extend it to higher order polynomials (considering additional degrees of freedom)...

...BUT...

- $\longrightarrow$ **...if we do not know how to compute explicitly the basis functions...**
- $\longrightarrow$ **...we don't know how to compute the stiffness matrix**

$$\mathcal{A}(\varphi_i, \varphi_j) = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$$

and the right-hand side $F_h(\mathbf{v}_h)$!

- $\longrightarrow$ **Here, the mimetic technology comes into play!**

The Virtual Element Method

→ So, we have a very nice method that works on polygonal meshes with very general shapes (also non-convex cells) and with a solid mathematical foundation (a priori error estimates, etc);

→ we can also extend it to higher order polynomials (considering additional degrees of freedom)...

...BUT...

→ **...if we do not know how to compute explicitly the basis functions...**

→ **...we don't know how to compute the stiffness matrix**

$$\mathcal{A}(\varphi_i, \varphi_j) = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$$

and the right-hand side $F_h(\mathbf{v}_h)$!

→ **Here, the mimetic technology comes into play!**

The Virtual Element Method

→ So, we have a very nice method that works on polygonal meshes with very general shapes (also non-convex cells) and with a solid mathematical foundation (a priori error estimates, etc);

→ we can also extend it to higher order polynomials (considering additional degrees of freedom)...

...BUT...

→ **...if we do not know how to compute explicitly the basis functions...**

→ **...we don't know how to compute the stiffness matrix**

$$\mathcal{A}(\varphi_i, \varphi_j) = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$$

and the right-hand side $F_h(\mathbf{v}_h)$!

→ **Here, the mimetic technology comes into play!**

The Virtual Element Method

→ So, we have a very nice method that works on polygonal meshes with very general shapes (also non-convex cells) and with a solid mathematical foundation (a priori error estimates, etc);

→ we can also extend it to higher order polynomials (considering additional degrees of freedom)...

...BUT...

→ **...if we do not know how to compute explicitly the basis functions...**

→ **...we don't know how to compute the stiffness matrix**

$$\mathcal{A}(\varphi_i, \varphi_j) = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$$

and the right-hand side $F_h(\mathbf{v}_h)$!

→ **Here, the mimetic technology comes into play!**

Mimetic approximation of the bilinear form $\mathcal{A}(\varphi_i, \varphi_j)$

- Let $\mathcal{A}_h$ be such approximation, i.e., $\mathcal{A}_h(\varphi_i, \varphi_j) \approx \mathcal{A}(\varphi_i, \varphi_j)$.
- If $\mathcal{A}_{\mathrm{P}}$ is the restriction of $\mathcal{A}$ to the polygon P

$$\mathcal{A}(v_h, w_h) = \sum_P \mathcal{A}_{\mathrm{P}}(v_{|P}, w_{|P}) = \sum_P \int_P \nabla v \cdot \nabla w$$

it is natural to assume that $\mathcal{A}_h$ can be split in the same way:

$$\mathcal{A}_h(v_h, w_h) = \sum_P \mathcal{A}_{h,\mathrm{P}}(v_{h|P}, w_{h|P}).$$

- Now, we give two conditions on $\mathcal{A}_{h,\mathrm{P}}$ that will guarantee the convergence: **consistency** and **stability**.

Mimetic approximation of the bilinear form $\mathcal{A}(\varphi_i, \varphi_j)$

- Let $\mathcal{A}_h$ be such approximation, i.e., $\mathcal{A}_h(\varphi_i, \varphi_j) \approx \mathcal{A}(\varphi_i, \varphi_j)$.
- If $\mathcal{A}_\mathrm{P}$ is the restriction of $\mathcal{A}$ to the polygon P

$$\mathcal{A}(v_h, w_h) = \sum_P \mathcal{A}_\mathrm{P}(v_{|P}, w_{|P}) = \sum_P \int_P \nabla v \cdot \nabla w$$

it is natural to assume that $\mathcal{A}_h$ can be split in the same way:

$$\mathcal{A}_h(v_h, w_h) = \sum_P \mathcal{A}_{h,\mathrm{P}}(v_{h|P}, w_{h|P}).$$

- Now, we give two conditions on $\mathcal{A}_{h,\mathrm{P}}$ that will guarantee the convergence: **consistency** and **stability**.

Mimetic approximation of the bilinear form $\mathcal{A}(\varphi_i, \varphi_j)$

- Let $\mathcal{A}_h$ be such approximation, i.e., $\mathcal{A}_h(\varphi_i, \varphi_j) \approx \mathcal{A}(\varphi_i, \varphi_j)$.
- If $\mathcal{A}_{\mathrm{P}}$ is the restriction of $\mathcal{A}$ to the polygon P

$$\mathcal{A}(v_h, w_h) = \sum_P \mathcal{A}_{\mathrm{P}}(v_{|P}, w_{|P}) = \sum_P \int_P \nabla v \cdot \nabla w$$

it is natural to assume that $\mathcal{A}_h$ can be split in the same way:

$$\mathcal{A}_h(v_h, w_h) = \sum_P \mathcal{A}_{h,\mathrm{P}}(v_{h|P}, w_{h|P}).$$

- Now, we give two conditions on $\mathcal{A}_{h,\mathrm{P}}$ that will guarantee the convergence: **consistency** and **stability**.

Mimetic approximation of the bilinear form $\mathcal{A}(\varphi_i, \varphi_j)$

- Let $\mathcal{A}_h$ be such approximation, i.e., $\mathcal{A}_h(\varphi_i, \varphi_j) \approx \mathcal{A}(\varphi_i, \varphi_j)$.
- If $\mathcal{A}_\mathrm{P}$ is the restriction of $\mathcal{A}$ to the polygon P

$$\mathcal{A}(v_h, w_h) = \sum_P \mathcal{A}_\mathrm{P}(v_{|P}, w_{|P}) = \sum_P \int_P \nabla v \cdot \nabla w$$

it is natural to assume that $\mathcal{A}_h$ can be split in the same way:

$$\mathcal{A}_h(v_h, w_h) = \sum_P \mathcal{A}_{h,\mathrm{P}}(v_{h|P}, w_{h|P}).$$

- Now, we give two conditions on $\mathcal{A}_{h,\mathrm{P}}$ that will guarantee the convergence: **consistency** and **stability**.

Consistency and Stability $\Rightarrow$ Convergence

Six-name paper: *Basic Principles of Virtual Elements*, M3AS, to appear

Theorem. Assume that for each polygonal cell P the bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ satisfies the following properties:

- ***Consistency:*** for all $q \in \mathbb{P}_1(\mathrm{P})$ and for all $v_h \in \mathcal{V}_{h,\mathrm{P}}$:

$$\mathcal{A}_{h,\mathrm{P}}(v_h, q) = \mathcal{A}(v_h, q)$$

  (an exactness property on linear polynomials).

- ***Stability:*** there exist two positive constants α^* and α_* independent of P, such that

$$\alpha_* \mathcal{A}_{\mathrm{P}}(v_h, v_h) \leq \mathcal{A}_{h,\mathrm{P}}(v_h, v_h) \leq \alpha^* \mathcal{A}_{\mathrm{P}}(v_h, v_h).$$

Consistency and Stability $\Rightarrow$ Convergence

Six-name paper: *Basic Principles of Virtual Elements*, M3AS, to appear

Theorem. Assume that for each polygonal cell P the bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ satisfies the following properties:

- ***Consistency:*** for all $q \in \mathbb{P}_1(\mathrm{P})$ and for all $v_h \in \mathcal{V}_{h,\mathrm{P}}$:

$$\mathcal{A}_{h,\mathrm{P}}(v_h, q) = \mathcal{A}(v_h, q)$$

(an exactness property on linear polynomials).

- ***Stability:*** there exist two positive constants α^* and α_* independent of P, such that

$$\alpha_* \mathcal{A}_{\mathrm{P}}(v_h, v_h) \leq \mathcal{A}_{h,\mathrm{P}}(v_h, v_h) \leq \alpha^* \mathcal{A}_{\mathrm{P}}(v_h, v_h).$$

Consistency and Stability $\Rightarrow$ Convergence

Six-name paper: *Basic Principles of Virtual Elements*, M3AS, to appear

Theorem. Assume that for each polygonal cell P the bilinear form $\mathcal{A}_{h,\mathsf{P}}(\cdot,\cdot)$ satisfies the following properties:

- ***Consistency:*** for all $q \in \mathbb{P}_1(\mathsf{P})$ and for all $v_h \in \mathcal{V}_{h,\mathsf{P}}$:

  $$\mathcal{A}_{h,\mathsf{P}}(v_h, q) = \mathcal{A}(v_h, q)$$

  (an exactness property on linear polynomials).

- ***Stability:*** there exist two positive constants α^* and α_* independent of P, such that

  $$\alpha_* \mathcal{A}_{\mathsf{P}}(v_h, v_h) \leq \mathcal{A}_{h,\mathsf{P}}(v_h, v_h) \leq \alpha^* \mathcal{A}_{\mathsf{P}}(v_h, v_h).$$

Consistency and Stability $\Rightarrow$ Convergence

Six-name paper: *Basic Principles of Virtual Elements*, M3AS, to appear

Theorem. Assume that for each polygonal cell P the bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ satisfies the following properties:

- ***Consistency:*** for all $q \in \mathbb{P}_1(\mathrm{P})$ and for all $v_h \in \mathcal{V}_{h,\mathrm{P}}$:

  $$\mathcal{A}_{h,\mathrm{P}}(v_h, q) = \mathcal{A}(v_h, q)$$

  (an exactness property on linear polynomials).

- ***Stability:*** there exist two positive constants α^* and α_* independent of P, such that

  $$\alpha_* \mathcal{A}_{\mathrm{P}}(v_h, v_h) \leq \mathcal{A}_{h,\mathrm{P}}(v_h, v_h) \leq \alpha^* \mathcal{A}_{\mathrm{P}}(v_h, v_h).$$

Let $u_h \in \mathcal{V}_h$ be such that $\quad \mathcal{A}_h(u_h, v_h) = F_h(v_h) \quad$ *for all* $v_h \in \mathcal{V}_h$.

Consistency and Stability $\Rightarrow$ Convergence

Six-name paper: *Basic Principles of Virtual Elements*, M3AS, to appear

Theorem. Assume that for each polygonal cell P the bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ satisfies the following properties:

- ***Consistency:*** for all $q \in \mathbb{P}_1(\mathrm{P})$ and for all $v_h \in \mathcal{V}_{h,\mathrm{P}}$:

  $$\mathcal{A}_{h,\mathrm{P}}(v_h, q) = \mathcal{A}(v_h, q)$$

  (an exactness property on linear polynomials).

- ***Stability:*** there exist two positive constants α^* and α_* independent of P, such that

  $$\alpha_* \mathcal{A}_{\mathrm{P}}(v_h, v_h) \leq \mathcal{A}_{h,\mathrm{P}}(v_h, v_h) \leq \alpha^* \mathcal{A}_{\mathrm{P}}(v_h, v_h).$$

Then:

$$\boxed{\|u - u_h\|_{H^1(\Omega)} \leq Ch\|u\|_{H^2(\Omega)}}.$$

A crucial remark

- How can we define a local bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ with the properties of consistency and stability? (Remember that we know the functions v_h of $\mathcal{V}_{h,\mathrm{P}}$ only on the boundary of P).

- If $v_h \in \mathcal{V}_{h,\mathrm{P}}$, we can compute the following quantity

$$\overline{\nabla v_h} := \frac{1}{|\mathrm{P}|}\int_{\mathrm{P}} \nabla v_h$$

  using only the vertex values.

- In fact,

- $\overline{\nabla v_h}$ is a constant vector in $\mathbb{R}^2$.

A crucial remark

- How can we define a local bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ with the properties of consistency and stability? (Remember that we know the functions v_h of $\mathcal{V}_{h,\mathrm{P}}$ only on the boundary of P).

- If $v_h \in \mathcal{V}_{h,\mathrm{P}}$, we can compute the following quantity

$$\overline{\nabla v_h} := \frac{1}{|\mathrm{P}|}\int_{\mathrm{P}} \nabla v_h$$

using only the vertex values.

- In fact,

- $\overline{\nabla v_h}$ is a constant vector in $\mathbb{R}^2$.

A crucial remark

- How can we define a local bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ with the properties of consistency and stability? (Remember that we know the functions v_h of $\mathcal{V}_{h,\mathrm{P}}$ only on the boundary of P)
- If $v_h \in \mathcal{V}_{h,\mathrm{P}}$, we can compute the following quantity

$$\overline{\nabla v_h} := \frac{1}{|\mathrm{P}|}\int_{\mathrm{P}} \nabla v_h$$

using only the vertex values.

- In fact,

$$\underbrace{\int_{\mathrm{P}} \nabla v_h = \int_{\partial P} v_h \mathbf{n}_{\mathrm{P}}}_{(Gauss-Green)}$$

- $\overline{\nabla v_h}$ is a constant vector in $\mathbb{R}^2$

A crucial remark

- How can we define a local bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ with the properties of consistency and stability? (Remember that we know the functions v_h of $\mathcal{V}_{h,\mathrm{P}}$ only on the boundary of P)

- If $v_h \in \mathcal{V}_{h,\mathrm{P}}$, we can compute the following quantity

$$\overline{\nabla v_h} := \frac{1}{|\mathrm{P}|} \int_{\mathrm{P}} \nabla v_h$$

  using only the vertex values.

- In fact,

$$\int_{\mathrm{P}} \nabla v_h = \underbrace{\int_{\partial P} v_h \mathbf{n}_{\mathrm{P}} = \sum_{i=1}^{N^{\mathcal{P}}} \left(\int_{\mathrm{e}_i} v_h \right) \mathbf{n}_{\mathrm{P},i}}_{split\ \ the\ \ boundary\ \ integral}$$

A crucial remark

- How can we define a local bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ with the properties of consistency and stability? (Remember that we know the functions v_h of $\mathcal{V}_{h,\mathrm{P}}$ only on the boundary of P)

- If $v_h \in \mathcal{V}_{h,\mathrm{P}}$, we can compute the following quantity

$$\overline{\nabla v_h} := \frac{1}{|\mathrm{P}|}\int_{\mathrm{P}} \nabla v_h$$

using only the vertex values.

- In fact,

$$\int_{\mathrm{P}} \nabla v_h = \int_{\partial P} v_h \mathbf{n}_{\mathrm{P}} = \sum_{i=1}^{N^{\mathcal{P}}} \underbrace{\left(\int_{\mathrm{e}_i} v_h\right)}_{v_{h|\mathrm{e}} \in \mathbb{P}_1(\mathrm{e})} \mathbf{n}_{\mathrm{P},i} = \sum_{i=1}^{N^{\mathcal{P}}} \underbrace{\frac{v_h(\mathrm{v}_i) + v_h(\mathrm{v}_{i+1})}{2} |\mathrm{e}_i|}_{\textit{trapezoidal rule}} \mathbf{n}_{\mathrm{P},i}$$

A crucial remark

- How can we define a local bilinear form $\mathcal{A}_{h,\mathrm{P}}(\cdot,\cdot)$ with the properties of consistency and stability? (Remember that we know the functions v_h of $\mathcal{V}_{h,\mathrm{P}}$ only on the boundary of P)
- If $v_h \in \mathcal{V}_{h,\mathrm{P}}$, we can compute the following quantity

$$\overline{\nabla v_h} := \frac{1}{|\mathrm{P}|}\int_{\mathrm{P}} \nabla v_h$$

  using only the vertex values.
- In fact,

$$\int_{\mathrm{P}} \nabla v_h = \int_{\partial P} v_h \mathbf{n}_{\mathrm{P}} = \sum_{i=1}^{N^{\mathcal{P}}} \left(\int_{\mathrm{e}_i} v_h \right) \mathbf{n}_{\mathrm{P},i} = \sum_{i=1}^{N^{\mathcal{P}}} \frac{v_h(\mathrm{v}_i) + v_h(\mathrm{v}_{i+1})}{2} |\mathrm{e}_i| \mathbf{n}_{\mathrm{P},i}$$

- $\overline{\nabla v_h}$ is a constant vector in $\mathbb{R}^2$.

The local projector $\Pi_{h,P}$

- Now, we are really tempted to say that

$$\int_P \nabla\varphi_i \cdot \nabla\varphi_j \approx \int_P \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j}$$

Why not? If P is a triangle, we get the stiffness matrix of the linear Galerkin FEM!

- Key idea: define a local projection operator for each polygonal cell P

$$\Pi_{h,P} : \mathcal{V}_{h,P} \longrightarrow \mathbb{P}_1(P)$$

that

- approximates the gradients using only the vertex values:

$$\nabla(\Pi_{h,P} v_h) = \overline{\nabla v_h}$$

- and preserves the linear polynomials:

$$\Pi_{h,P}\, q = q \quad \text{for all } q \in \mathbb{P}_1(P).$$

The local projector $\Pi_{h,\mathrm{P}}$

- Now, we are really tempted to say that

$$\mathcal{A}_{\mathrm{P}}(\varphi_i,\varphi_j) := \int_{\mathrm{P}} \nabla\varphi_i \cdot \nabla\varphi_j \approx \int_{\mathrm{P}} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j} =: \mathcal{A}_{h,\mathrm{P}}(\varphi_i,\varphi_j)$$

  Why not? If P is a triangle, we get the stiffness matrix of the linear Galerkin FEM!

- Key idea: define a local projection operator for each polygonal cell P

$$\Pi_{h,\mathrm{P}} : \mathcal{V}_{h,\mathrm{P}} \longrightarrow \mathbb{P}_1(\mathrm{P})$$

  that

  - approximates the gradients using only the vertex values:

$$\nabla\left(\Pi_{h,\mathrm{P}} v_h\right) = \overline{\nabla v_h}$$

  - and preserves the linear polynomials:

$$\Pi_{h,\mathrm{P}}\, q = q \quad \text{for all } q \in \mathbb{P}_1(\mathrm{P}).$$

The local projector $\Pi_{h,\mathsf{P}}$

- Now, we are really tempted to say that

$$\mathcal{A}_{\mathsf{P}}(\varphi_i,\varphi_j) := \int_{\mathsf{P}} \nabla\varphi_i \cdot \nabla\varphi_j \approx \int_{\mathsf{P}} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j} =: \mathcal{A}_{h,\mathsf{P}}(\varphi_i,\varphi_j)$$

  But $\mathcal{A}_{h,\mathsf{P}}(\varphi_i,\varphi_j)$ would have rank 2 for any kind of polygons, thus leading to a singular approximation for $\mathcal{A}_h$!

- Key idea: define a local projection operator for each polygonal cell P

$$\Pi_{h,\mathsf{P}} : \mathcal{V}_{h,\mathsf{P}} \longrightarrow \mathbb{P}_1(\mathsf{P})$$

  that
  - approximates the gradients using only the vertex values:

$$\nabla\left(\Pi_{h,\mathsf{P}} v_h\right) = \overline{\nabla v_h}$$

  - and preserves the linear polynomials:

$$\Pi_{h,\mathsf{P}}\, q = q \quad \text{for all } q \in \mathbb{P}_1(\mathsf{P}).$$

The local projector $\Pi_{h,\mathrm{P}}$

- Now, we are really tempted to say that

$$\mathcal{A}_{\mathrm{P}}(\varphi_i,\varphi_j) := \int_{\mathrm{P}} \nabla\varphi_i \cdot \nabla\varphi_j \approx \int_{\mathrm{P}} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j} =: \mathcal{A}_{h,\mathrm{P}}(\varphi_i,\varphi_j)$$

  But $\mathcal{A}_{h,\mathrm{P}}(\varphi_i,\varphi_j)$ would have rank 2 for any kind of polygons, thus leading to a singular approximation for $\mathcal{A}_h$!

- Key idea: define a local projection operator for each polygonal cell P

$$\Pi_{h,\mathrm{P}} : \mathcal{V}_{h,\mathrm{P}} \longrightarrow \mathbb{P}_1(\mathrm{P})$$

  that
  - approximates the gradients using only the vertex values:

$$\nabla\left(\Pi_{h,\mathrm{P}} v_h\right) = \overline{\nabla v_h}$$

  - and preserves the linear polynomials:

$$\Pi_{h,\mathrm{P}} q = q \quad \text{for all } q \in \mathbb{P}_1(\mathrm{P}).$$

The local projector $\Pi_{h,\mathrm{P}}$

- Now, we are really tempted to say that

$$\mathcal{A}_{\mathrm{P}}(\varphi_i,\varphi_j) := \int_{\mathrm{P}} \nabla\varphi_i \cdot \nabla\varphi_j \approx \int_{\mathrm{P}} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j} =: \mathcal{A}_{h,\mathrm{P}}(\varphi_i,\varphi_j)$$

  But $\mathcal{A}_{h,\mathrm{P}}(\varphi_i,\varphi_j)$ would have rank 2 for any kind of polygons, thus leading to a singular approximation for $\mathcal{A}_h$!

- Key idea: define a local projection operator for each polygonal cell P

$$\Pi_{h,\mathrm{P}} : \mathcal{V}_{h,\mathrm{P}} \longrightarrow \mathbb{P}_1(\mathrm{P})$$

  that
  - approximates the gradients using only the vertex values:

$$\nabla\left(\Pi_{h,\mathrm{P}} v_h\right) = \overline{\nabla v_h}$$

  - and preserves the linear polynomials:

$$\Pi_{h,\mathrm{P}} q = q \quad \text{for all } q \in \mathbb{P}_1(\mathrm{P}).$$

The local projector $\Pi_{h,\mathrm{P}}$

- Now, we are really tempted to say that

$$\mathcal{A}_{\mathrm{P}}(\varphi_i, \varphi_j) := \int_{\mathrm{P}} \nabla\varphi_i \cdot \nabla\varphi_j \approx \int_{\mathrm{P}} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j} =: \mathcal{A}_{h,\mathrm{P}}(\varphi_i, \varphi_j)$$

But $\mathcal{A}_{h,\mathrm{P}}(\varphi_i, \varphi_j)$ would have rank 2 for any kind of polygons, thus leading to a singular approximation for $\mathcal{A}_h$!

- Key idea: define a local projection operator for each polygonal cell P

$$\Pi_{h,\mathrm{P}} : \mathcal{V}_{h,\mathrm{P}} \longrightarrow \mathbb{P}_1(\mathrm{P})$$

that

  - approximates the gradients using only the vertex values:

$$\nabla\left(\Pi_{h,\mathrm{P}} v_h\right) = \overline{\nabla v_h}$$

  - and preserves the linear polynomials:

$$\Pi_{h,\mathrm{P}} q = q \quad \text{for all } q \in \mathbb{P}_1(\mathrm{P}).$$

The mimetic bilinear form $\mathcal{A}_{h,\mathrm{P}}$

We start writing that

$$\mathcal{A}_{h,\mathrm{P}}(u_h, v_h) = \mathcal{A}_{h,\mathrm{P}}(\Pi_{h,\mathrm{P}} u_h, v_h) + \mathcal{A}_{h,\mathrm{P}}(u_h - \Pi_{h,\mathrm{P}} u_h, v_h).$$

With an easy computation it can be shown that

$$\mathcal{A}_{h,\mathrm{P}}(\Pi_{h,\mathrm{P}} u_h, v_h) = \mathcal{A}_{\mathrm{P}}(\Pi_{h,\mathrm{P}} u_h, \Pi_{h,\mathrm{P}} v_h) := \mathcal{A}^0_{h,\mathrm{P}}(u_h, v_h)$$

and

$$\mathcal{A}_{h,\mathrm{P}}((I - \Pi_{h,\mathrm{P}}) u_h, v_h) = \mathcal{A}_{\mathrm{P}}((I - \Pi_{h,\mathrm{P}}) u_h, (I - \Pi_{h,\mathrm{P}}) v_h) \longrightarrow \mathcal{A}^1_{h,\mathrm{P}}(u_h, v_h)$$

We will set:

$$\boxed{\mathcal{A}_{h,\mathrm{P}} = \mathcal{A}^0_{h,\mathrm{P}} + \mathcal{A}^1_{h,\mathrm{P}} = \text{CONSISTENCY} + \text{STABILITY}}$$

The mimetic bilinear form $\mathcal{A}_{h,\mathrm{P}}$

We start writing that

$$\mathcal{A}_{h,\mathrm{P}}\big(u_h, v_h\big) = \mathcal{A}_{h,\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, v_h\big) + \mathcal{A}_{h,\mathrm{P}}\big(u_h - \Pi_{h,\mathrm{P}} u_h, v_h\big).$$

With an easy computation it can be shown that

$$\mathcal{A}_{h,\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, v_h\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, \Pi_{h,\mathrm{P}} v_h\big) := \mathcal{A}^0_{h,\mathrm{P}}\big(u_h, v_h\big)$$

and

$$\mathcal{A}_{h,\mathrm{P}}\big((I - \Pi_{h,\mathrm{P}})u_h, v_h\big) = \mathcal{A}_{\mathrm{P}}\big((I - \Pi_{h,\mathrm{P}})u_h, (I - \Pi_{h,\mathrm{P}})v_h\big) \longrightarrow \mathcal{A}^1_{h,\mathrm{P}}\big(u_h, v_h\big)$$

We will set:

$$\boxed{\mathcal{A}_{h,\mathrm{P}} = \mathcal{A}^0_{h,\mathrm{P}} + \mathcal{A}^1_{h,\mathrm{P}} = \text{CONSISTENCY} + \text{STABILITY}}$$

The mimetic bilinear form $\mathcal{A}_{h,\mathrm{P}}$

We start writing that

$$\mathcal{A}_{h,\mathrm{P}}\big(u_h, v_h\big) = \mathcal{A}_{h,\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, v_h\big) + \mathcal{A}_{h,\mathrm{P}}\big(u_h - \Pi_{h,\mathrm{P}} u_h, v_h\big).$$

With an easy computation it can be shown that

$$\mathcal{A}_{h,\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, v_h\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, \Pi_{h,\mathrm{P}} v_h\big) := \mathcal{A}^0_{h,\mathrm{P}}\big(u_h, v_h\big)$$

and

$$\mathcal{A}_{h,\mathrm{P}}\big((I - \Pi_{h,\mathrm{P}})u_h, v_h\big) = \mathcal{A}_{\mathrm{P}}\big((I - \Pi_{h,\mathrm{P}})u_h, (I - \Pi_{h,\mathrm{P}})v_h\big) \longrightarrow \mathcal{A}^1_{h,\mathrm{P}}\big(u_h, v_h\big)$$

We will set:

$$\boxed{\mathcal{A}_{h,\mathrm{P}} = \mathcal{A}^0_{h,\mathrm{P}} + \mathcal{A}^1_{h,\mathrm{P}} = \text{CONSISTENCY} + \text{STABILITY}}$$

The mimetic bilinear form $\mathcal{A}_{h,\mathrm{P}}$

We start writing that

$$\mathcal{A}_{h,\mathrm{P}}\big(u_h, v_h\big) = \mathcal{A}_{h,\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, v_h\big) + \mathcal{A}_{h,\mathrm{P}}\big(u_h - \Pi_{h,\mathrm{P}} u_h, v_h\big).$$

With an easy computation it can be shown that

$$\mathcal{A}_{h,\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, v_h\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, \Pi_{h,\mathrm{P}} v_h\big) := \mathcal{A}^0_{h,\mathrm{P}}\big(u_h, v_h\big)$$

and

$$\mathcal{A}_{h,\mathrm{P}}\big((I - \Pi_{h,\mathrm{P}})u_h, v_h\big) = \mathcal{A}_{\mathrm{P}}\big((I - \Pi_{h,\mathrm{P}})u_h, (I - \Pi_{h,\mathrm{P}})v_h\big) \longrightarrow \mathcal{A}^1_{h,\mathrm{P}}\big(u_h, v_h\big)$$

We will set:

$$\boxed{\mathcal{A}_{h,\mathrm{P}} = \mathcal{A}^0_{h,\mathrm{P}} + \mathcal{A}^1_{h,\mathrm{P}} = \text{CONSISTENCY} + \text{STABILITY}}$$

The mimetic bilinear form $\mathcal{A}_{h,\mathrm{P}}$

We start writing that

$$\mathcal{A}_{h,\mathrm{P}}\big(u_h, v_h\big) = \mathcal{A}_{h,\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, v_h\big) + \mathcal{A}_{h,\mathrm{P}}\big(u_h - \Pi_{h,\mathrm{P}} u_h, v_h\big).$$

With an easy computation it can be shown that

$$\mathcal{A}_{h,\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, v_h\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}} u_h, \Pi_{h,\mathrm{P}} v_h\big) := \mathcal{A}^0_{h,\mathrm{P}}\big(u_h, v_h\big)$$

and

$$\mathcal{A}_{h,\mathrm{P}}\big((I - \Pi_{h,\mathrm{P}})u_h, v_h\big) = \mathcal{A}_{\mathrm{P}}\big((I - \Pi_{h,\mathrm{P}})u_h, (I - \Pi_{h,\mathrm{P}})v_h\big) \longrightarrow \mathcal{A}^1_{h,\mathrm{P}}\big(u_h, v_h\big)$$

We will set:

$$\boxed{\mathcal{A}_{h,\mathrm{P}} = \mathcal{A}^0_{h,\mathrm{P}} + \mathcal{A}^1_{h,\mathrm{P}} = \text{CONSISTENCY} + \text{STABILITY}}$$

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}(v_h, q) = \mathcal{A}_{\mathrm{P}}(v_h, q)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$
\begin{aligned}
\mathcal{A}^0_{h,\mathrm{P}}(v_h, q) &= \int_{\mathrm{P}} \overline{\nabla v_h} \cdot \overline{\nabla q} = |\mathrm{P}|\, \overline{\nabla v_h} \cdot \overline{\nabla q} = \left(\int_{\mathrm{P}} \nabla v_h\right) \cdot \overline{\nabla q} \\
&= \int_{\mathrm{P}} \nabla v_h \cdot \overline{\nabla q} = \int_{\mathrm{P}} \nabla v_h \cdot \nabla q = \mathcal{A}_{\mathrm{P}}(v_h, q).
\end{aligned}
$$

- the remaining term is zero because $(I - \Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i,\varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i, \Pi_{h,\mathrm{P}}\varphi_j\big)$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h,q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h,q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h,q\big) = \int_{\mathrm{P}} \overline{\nabla v_h}\cdot\overline{\nabla q} = |\mathrm{P}|\,\overline{\nabla v_h}\cdot\overline{\nabla q} = \left(\int_{\mathrm{P}} \nabla v_h\right)\cdot\overline{\nabla q}$$

$$= \int_{\mathrm{P}} \nabla v_h\cdot\overline{\nabla q} = \int_{\mathrm{P}} \nabla v_h\cdot\nabla q = \mathcal{A}_{\mathrm{P}}\big(v_h,q\big).$$

- the remaining term is zero because $(I-\Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i,\varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i,\Pi_{h,\mathrm{P}}\varphi_j\big) = \int_{\mathrm{P}} \nabla\Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla\Pi_{h,\mathrm{P}}\varphi_j$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h,q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h,q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h,q\big) = \int_{\mathrm{P}} \overline{\nabla v_h}\cdot\overline{\nabla q} = |\mathrm{P}|\,\overline{\nabla v_h}\cdot\overline{\nabla q} = \left(\int_{\mathrm{P}} \nabla v_h\right)\cdot\overline{\nabla q}$$

$$= \int_{\mathrm{P}} \nabla v_h\cdot\overline{\nabla q} = \int_{\mathrm{P}} \nabla v_h\cdot\nabla q = \mathcal{A}_{\mathrm{P}}\big(v_h,q\big).$$

- the remaining term is zero because $(I-\Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i,\varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i, \Pi_{h,\mathrm{P}}\varphi_j\big) = \int_{\mathrm{P}} \nabla\Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla\Pi_{h,\mathrm{P}}\varphi_j = \int_{\mathrm{P}} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h,q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h,q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h,q\big) = \int_{\mathrm{P}} \overline{\nabla v_h} \cdot \overline{\nabla q} = |\mathrm{P}|\, \overline{\nabla v_h} \cdot \overline{\nabla q} = \left(\int_{\mathrm{P}} \nabla v_h\right) \cdot \overline{\nabla q}$$
$$= \int_{\mathrm{P}} \nabla v_h \cdot \overline{\nabla q} = \int_{\mathrm{P}} \nabla v_h \cdot \nabla q = \mathcal{A}_{\mathrm{P}}\big(v_h,q\big).$$

- the remaining term is zero because $(I-\Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla \Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the “constant gradient approximation” of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i, \varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i, \Pi_{h,\mathrm{P}}\varphi_j\big) = \int_{\mathrm{P}} \nabla \Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla \Pi_{h,\mathrm{P}}\varphi_j = \int_{\mathrm{P}} \overline{\nabla \varphi_i} \cdot \overline{\nabla \varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h, q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h, q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

- the remaining term is zero because $(I - \Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the “constant gradient approximation” of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i,\varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i,\Pi_{h,\mathrm{P}}\varphi_j\big) = \int_{\mathrm{P}} \nabla\Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla\Pi_{h,\mathrm{P}}\varphi_j = \int_{\mathrm{P}} \overline{\nabla\varphi_i}\cdot\overline{\nabla\varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h,q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h,q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h,q\big) = \int_{\mathrm{P}} \overline{\nabla v_h}\cdot\overline{\nabla q}$$

- the remaining term is zero because $(I-\Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla \Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i, \varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i, \Pi_{h,\mathrm{P}}\varphi_j\big) = \int_{\mathrm{P}} \nabla\Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla\Pi_{h,\mathrm{P}}\varphi_j = \int_{\mathrm{P}} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h, q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h, q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h, q\big) = \int_{\mathrm{P}} \overline{\nabla v_h} \cdot \overline{\nabla q} = |\mathrm{P}|\, \overline{\nabla v_h} \cdot \overline{\nabla q}$$

- the remaining term is zero because $(I - \Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i,\varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i, \Pi_{h,\mathrm{P}}\varphi_j\big) = \int_{\mathrm{P}} \nabla\Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla\Pi_{h,\mathrm{P}}\varphi_j = \int_{\mathrm{P}} \overline{\nabla\varphi_i}\cdot\overline{\nabla\varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h,q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h,q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h,q\big) = \int_{\mathrm{P}} \overline{\nabla v_h}\cdot\overline{\nabla q} = |\mathrm{P}|\,\overline{\nabla v_h}\cdot\overline{\nabla q} = \left(\int_{\mathrm{P}} \nabla v_h\right)\cdot\overline{\nabla q}$$

- the remaining term is zero because $(I-\Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i, \varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i, \Pi_{h,\mathrm{P}}\varphi_j\big) = \int_{\mathrm{P}} \nabla\Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla\Pi_{h,\mathrm{P}}\varphi_j = \int_{\mathrm{P}} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h, q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h, q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h, q\big) = \int_{\mathrm{P}} \overline{\nabla v_h} \cdot \overline{\nabla q} = |\mathrm{P}|\, \overline{\nabla v_h} \cdot \overline{\nabla q} = \left(\int_{\mathrm{P}} \nabla v_h\right) \cdot \overline{\nabla q}$$

$$= \int_{\mathrm{P}} \nabla v_h \cdot \overline{\nabla q}$$

- the remaining term is zero because $(I - \Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla \Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i, \varphi_j\big) = \mathcal{A}_{\mathrm{P}}\big(\Pi_{h,\mathrm{P}}\varphi_i, \Pi_{h,\mathrm{P}}\varphi_j\big) = \int_{\mathrm{P}} \nabla \Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla \Pi_{h,\mathrm{P}}\varphi_j = \int_{\mathrm{P}} \overline{\nabla \varphi_i} \cdot \overline{\nabla \varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h, q\big) = \mathcal{A}_{\mathrm{P}}\big(v_h, q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h, q\big) = \int_{\mathrm{P}} \overline{\nabla v_h} \cdot \overline{\nabla q} = |\mathrm{P}|\, \overline{\nabla v_h} \cdot \overline{\nabla q} = \left(\int_{\mathrm{P}} \nabla v_h\right) \cdot \overline{\nabla q}$$

$$= \int_{\mathrm{P}} \nabla v_h \cdot \overline{\nabla q} = \int_{\mathrm{P}} \nabla v_h \cdot \nabla q$$

- the remaining term is zero because $(I - \Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}} v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}} q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}(\varphi_i,\varphi_j) = \mathcal{A}_\mathrm{P}(\Pi_{h,\mathrm{P}}\varphi_i, \Pi_{h,\mathrm{P}}\varphi_j) = \int_\mathrm{P} \nabla\Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla\Pi_{h,\mathrm{P}}\varphi_j = \int_\mathrm{P} \overline{\nabla\varphi_i} \cdot \overline{\nabla\varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}(v_h, q) = \mathcal{A}_\mathrm{P}(v_h, q)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}(v_h, q) = \int_\mathrm{P} \overline{\nabla v_h} \cdot \overline{\nabla q} = |\mathrm{P}|\, \overline{\nabla v_h} \cdot \overline{\nabla q} = \left(\int_\mathrm{P} \nabla v_h\right) \cdot \overline{\nabla q}$$

$$= \int_\mathrm{P} \nabla v_h \cdot \overline{\nabla q} = \int_\mathrm{P} \nabla v_h \cdot \nabla q = \mathcal{A}_\mathrm{P}(v_h, q).$$

- the second term $\mathcal{A}^1_{h,\mathrm{P}}$ is zero because $(I - \Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The consistency term $\mathcal{A}^0_{h,\mathrm{P}}$

Recall that: $\nabla\Pi_{h,\mathrm{P}}v_h = \overline{\nabla v_h} \quad \forall v_h \in \mathcal{V}_{h,\mathrm{P}}$ and $\Pi_{h,\mathrm{P}}q = q \quad \forall q \in \mathbb{P}_1(\mathrm{P})$.

- $\mathcal{A}^0_{h,\mathrm{P}}$ is the "constant gradient approximation" of the stiffness matrix:

$$\mathcal{A}^0_{h,\mathrm{P}}\big(\varphi_i,\varphi_j\big) = \mathcal{A}_\mathrm{P}\big(\Pi_{h,\mathrm{P}}\varphi_i,\Pi_{h,\mathrm{P}}\varphi_j\big) = \int_\mathrm{P} \nabla\Pi_{h,\mathrm{P}}\varphi_i \cdot \nabla\Pi_{h,\mathrm{P}}\varphi_j = \int_\mathrm{P} \overline{\nabla\varphi_i}\cdot\overline{\nabla\varphi_j}$$

- $\mathcal{A}^0_{h,\mathrm{P}}$ ensures the consistency condition: $\mathcal{A}_{h,\mathrm{P}}\big(v_h,q\big) = \mathcal{A}_\mathrm{P}\big(v_h,q\big)$ for all $q \in \mathbb{P}_1(\mathrm{P})$; in fact,

$$\mathcal{A}^0_{h,\mathrm{P}}\big(v_h,q\big) = \int_\mathrm{P}\overline{\nabla v_h}\cdot\overline{\nabla q} = |\mathrm{P}|\,\overline{\nabla v_h}\cdot\overline{\nabla q} = \left(\int_\mathrm{P}\nabla v_h\right)\cdot\overline{\nabla q}$$
$$= \int_\mathrm{P}\nabla v_h\cdot\overline{\nabla q} = \int_\mathrm{P}\nabla v_h\cdot\nabla q = \mathcal{A}_\mathrm{P}\big(v_h,q\big).$$

- the second term $\mathcal{A}^1_{h,\mathrm{P}}$ is zero because $(I-\Pi_{h,\mathrm{P}})q = 0$ if $q \in \mathbb{P}_1(\mathrm{P})$.

The stability term $\mathcal{A}^1_{h,\mathrm{P}}$

- We need to correct $\mathcal{A}^0_{h,\mathrm{P}}$ in such a way that:
  - consistency is not upset;
  - we get stability;
  - we can compute the correction!

- In the six-name paper we show that we can substitute the (non computable!) term $\mathcal{A}_{\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h,(I-\Pi_{h,\mathrm{P}})v_h\big)$ with

$$\mathcal{A}^1_{h,\mathrm{P}}(u_h,v_h) := \mathcal{S}_{h,\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h,(I-\Pi_{h,\mathrm{P}})v_h\big)$$

where $\mathcal{S}_{h,\mathrm{P}}$ can be **any symmetric and positive definite bilinear form** that behaves (asymptotically) like $\mathcal{A}_{\mathrm{P}}$ on the kernel of $\Pi_{h,\mathrm{P}}$.

- Hence:

$$\mathcal{A}_{h,\mathrm{P}}(u_h,v_h) := \boxed{\mathcal{A}_{\mathrm{P}}(\Pi_{h,\mathrm{P}}u_h,\Pi_{h,\mathrm{P}}v_h)} + \boxed{\mathcal{S}_{h,\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h,(I-\Pi_{h,\mathrm{P}})v_h\big)}$$

CONSISTENCY STABILITY

The stability term $\mathcal{A}^1_{h,\mathrm{P}}$

- We need to correct $\mathcal{A}^0_{h,\mathrm{P}}$ in such a way that:
  - consistency is not upset;
  - we get stability;
  - we can compute the correction!

- In the six-name paper we show that we can substitute the (non computable!) term $\mathcal{A}_{\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h,(I-\Pi_{h,\mathrm{P}})v_h\big)$ with

$$\mathcal{A}^1_{h,\mathrm{P}}(u_h,v_h) := \mathcal{S}_{h,\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h,(I-\Pi_{h,\mathrm{P}})v_h\big)$$

where $\mathcal{S}_{h,\mathrm{P}}$ can be **any symmetric and positive definite bilinear form** that behaves (asymptotically) like $\mathcal{A}_{\mathrm{P}}$ on the kernel of $\Pi_{h,\mathrm{P}}$.

- Hence:

$$\mathcal{A}_{h,\mathrm{P}}(u_h,v_h) := \underbrace{\boxed{\mathcal{A}_{\mathrm{P}}(\Pi_{h,\mathrm{P}}u_h,\Pi_{h,\mathrm{P}}v_h)}}_{\text{CONSISTENCY}} + \underbrace{\boxed{\mathcal{S}_{h,\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h,(I-\Pi_{h,\mathrm{P}})v_h\big)}}_{\text{STABILITY}}$$

The stability term $\mathcal{A}^1_{h,\mathrm{P}}$

- We need to correct $\mathcal{A}^0_{h,\mathrm{P}}$ in such a way that:
  - consistency is not upset;
  - we get stability;
  - we can compute the correction!

- In the six-name paper we show that we can substitute the (non computable!) term $\mathcal{A}_{\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h, (I-\Pi_{h,\mathrm{P}})v_h\big)$ with

$$\mathcal{A}^1_{h,\mathrm{P}}(u_h, v_h) := \mathcal{S}_{h,\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h, (I-\Pi_{h,\mathrm{P}})v_h\big)$$

  where $\mathcal{S}_{h,\mathrm{P}}$ can be **any symmetric and positive definite bilinear form** that behaves (asymptotically) like $\mathcal{A}_{\mathrm{P}}$ on the kernel of $\Pi_{h,\mathrm{P}}$.

- Hence:

$$\mathcal{A}_{h,\mathrm{P}}(u_h, v_h) := \underbrace{\boxed{\mathcal{A}_{\mathrm{P}}(\Pi_{h,\mathrm{P}}u_h, \Pi_{h,\mathrm{P}}v_h)}}_{\text{CONSISTENCY}} + \underbrace{\boxed{\mathcal{S}_{h,\mathrm{P}}\big((I-\Pi_{h,\mathrm{P}})u_h, (I-\Pi_{h,\mathrm{P}})v_h\big)}}_{\text{STABILITY}}$$

Arbitrary-order polynomials

Let us integrate by parts on cell P:

$$\int_{\mathrm{P}} \nabla u \cdot \nabla v = -\int_{\mathrm{P}} \Delta u\, v + \sum_{\mathrm{e}\in\partial\mathrm{e}} \int_{\mathrm{e}} \nabla u \cdot \mathbf{n}_{\mathrm{P},\mathrm{e}}\, v\,.$$

Arbitrary-order polynomials

Let us integrate by parts on cell P:

$$\int_{\mathrm{P}} \nabla u \cdot \nabla v = -\int_{\mathrm{P}} \underbrace{\color{red}{\Delta u}}_{\textbf{not zero!}} v + \sum_{\mathrm{e}\in\partial\mathrm{e}} \int_{\mathrm{e}} \nabla u \cdot \mathbf{n}_{\mathrm{P,e}}\, v\,.$$

If u is a polynomial of degree m on P:

- $\color{red}{\Delta u}$ is a polynomial of degree $m-2$;

Arbitrary-order polynomials

Let us integrate by parts on cell P:

$$\int_{\mathrm{P}} \nabla u \cdot \nabla v = -\int_{\mathrm{P}} \underbrace{\Delta u}_{\textbf{not zero!}} \, v + \sum_{\mathrm{e} \in \partial \mathrm{e}} \int_{\mathrm{e}} \underbrace{\nabla u \cdot \mathbf{n}_{\mathrm{P,e}}}_{\textbf{not constant!}} \, v \,.$$

If u is a polynomial of degree m on P:

- Δu is a polynomial of degree $m - 2$;
- $\nabla u \cdot \mathbf{n}_{\mathrm{P,e}}$ is a polynomial of degree $m - 1$;

Divergence term: internal degrees of freedom

1. We use the **moments of v** to express the integral over P:

if

$$\Delta u = a_0 1 + a_1 x + a_2 y + \ldots \in \mathbb{P}_{m-2}(\mathrm{P})$$

Divergence term: internal degrees of freedom

1. We use the **moments of v** to express the integral over P:

if

$$\Delta u = a_0 1 + a_1 x + a_2 y + \ldots \in \mathbb{P}_{m-2}(\mathrm{P})$$

then

$$\int_{\mathrm{P}} \Delta u\, v = a_0 \underbrace{\int_{\mathrm{P}} 1\, v}_{\hat{v}_{\mathrm{P},0}} + a_1 \underbrace{\int_{\mathrm{P}} x\, v}_{\hat{v}_{\mathrm{P},1,x}} + a_2 \underbrace{\int_{\mathrm{P}} y\, v}_{\hat{v}_{\mathrm{P},1,y}} + \ldots$$

$$= a_0 \hat{\mathbf{v}}_{\mathrm{P},\mathbf{0}} + a_1 \hat{\mathbf{v}}_{\mathrm{P},\mathbf{1},\mathbf{x}} + a_2 \hat{\mathbf{v}}_{\mathrm{P},\mathbf{1},\mathbf{y}} + \ldots$$

Divergence term: internal degrees of freedom

1. We use the **moments of v** to express the integral over P:

if

$$\Delta u = a_0 1 + a_1 x + a_2 y + \ldots \in \mathbb{P}_{m-2}(\mathrm{P})$$

then

$$\int_{\mathrm{P}} \Delta u\, v = a_0 \underbrace{\int_{\mathrm{P}} 1\, v}_{\hat{v}_{\mathrm{P},0}} + a_1 \underbrace{\int_{\mathrm{P}} x\, v}_{\hat{v}_{\mathrm{P},1,x}} + a_2 \underbrace{\int_{\mathrm{P}} y\, v}_{\hat{v}_{\mathrm{P},1,y}} + \ldots$$

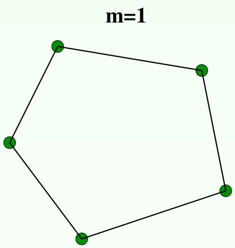
$$= a_0 \hat{\mathbf{v}}_{\mathrm{P},\mathbf{0}} + a_1 \hat{\mathbf{v}}_{\mathrm{P},\mathbf{1},\mathbf{x}} + a_2 \hat{\mathbf{v}}_{\mathrm{P},\mathbf{1},\mathbf{y}} + \ldots$$

This choice suggests us to define

- $m(m-1)/2$ **internal** degrees of freedom $\approx \hat{\mathbf{v}}_{\mathrm{P},\mathbf{0}},\ \hat{\mathbf{v}}_{\mathrm{P},\mathbf{1},\mathbf{x}},\ \hat{\mathbf{v}}_{\mathrm{P},\mathbf{1},\mathbf{y}},\ \ldots$

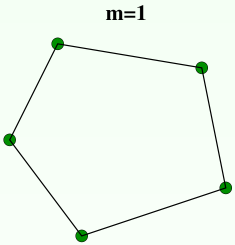
C^0 high-order approximations

- The "$C^0 - \mathbb{P}_1$" approximation requires:
  - one real number per mesh vertex v;

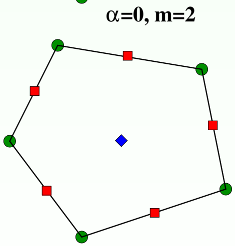


C^0 high-order approximations

- The "$C^0 - \mathbb{P}_1$" approximation requires:
  - one real number per mesh vertex v;

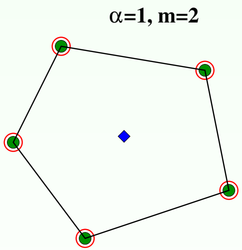


- the "$C^0 - \mathbb{P}_m$" approximations for $m > 1$ require
  - one real number per mesh vertex v;
  - $(m-1)$ real numbers per mesh edge e;
  - $m(m-1)/2$ real numbers per mesh cell P;



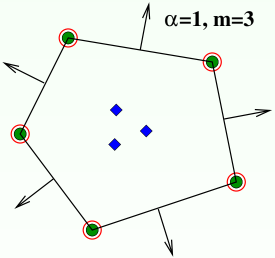
Approximations with high regularity

- The "$C^1 - \mathbb{P}_2$" approximation requires:
  - *vertex dofs* $\rightarrow$ solution and derivatives at each vertex;
  - *cell dofs* $\rightarrow$ solution moments inside the cells;



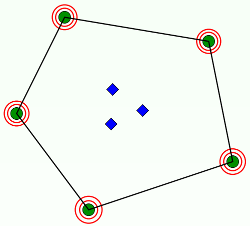
Approximations with high regularity

- The "$C^1 - \mathbb{P}_3$" approximation requires:
  - *vertex dofs* $\rightarrow$ solution and derivatives at each vertex;
  - *cell dofs* $\rightarrow$ solution moments inside the cells;
  - *edge dofs* $\rightarrow$ *solution* and *normal derivatives* along the edges;



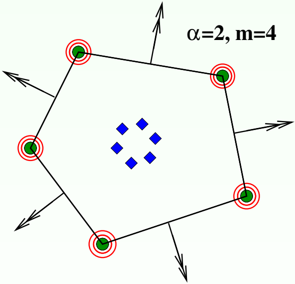
Approximations with high regularity

- The "$C^2 - \mathbb{P}_3$" approximation requires:
  - *vertex dofs* $\rightarrow$ solution and derivatives at each vertex;
  - *cell dofs* $\rightarrow$ solution moments inside the cells;



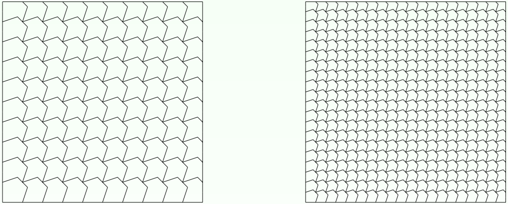
Approximations with high regularity

- The "$C^2 - \mathbb{P}_4$" approximation requires:
  - *vertex dofs* → solution and derivatives at each vertex;
  - *cell dofs* → solution moments inside the cells;
  - *edge dofs* → *solution* and *normal derivatives* along the edges;



Numerical experiments
Meshes with non-convex polygons

- **Meshes:**



- **Exact solution:** $u(x,y) = e^{-2\pi y}\sin(2\pi x)$
- **Diffusion tensor**

$$\mathrm{K}(x,y) = \begin{pmatrix} (x+1)^2 + y^2 & -xy \\ -xy & (x+1)^2 \end{pmatrix}$$

Continuous approximations

$\alpha = 0$, non-convex polygons, $\|\cdot\|_{1,h}$ errors, non-constant K

		$\mathbf{m} = \mathbf{1}$		$\mathbf{m} = \mathbf{2}$	
n	h	Error	Rate	Error	Rate
0	$1.458\,10^{-1}$	3.544	$--$	3.007	$--$
1	$7.289\,10^{-2}$	3.046	0.22	$8.081\,10^{-1}$	1.89
2	$3.644\,10^{-2}$	1.887	0.69	$2.071\,10^{-1}$	1.96
3	$1.822\,10^{-2}$	1.000	0.92	$5.303\,10^{-2}$	1.97
4	$9.111\,10^{-3}$	$5.154\,10^{-1}$	**0.98**	$1.348\,10^{-2}$	**1.98**

High-regular approximations

$\alpha = 1, 2$; non-convex polygons, $\|\cdot\|_{1,h}$ errors, non-constant K

		$\alpha = 1, \mathbf{m} = \mathbf{2}$		$\alpha = 2, \mathbf{m} = \mathbf{3}$	
n	h	Error	Rate	Error	Rate
0	$1.458\,10^{-1}$	$8.901\,10^{-2}$	$--$	$1.054\,10^{-2}$	$--$
1	$7.289\,10^{-2}$	$1.983\,10^{-2}$	2.26	$4.543\,10^{-4}$	4.72
2	$3.644\,10^{-2}$	$4.815\,10^{-3}$	2.08	$4.663\,10^{-5}$	3.36
3	$1.822\,10^{-2}$	$1.198\,10^{-3}$	**2.03**	$5.528\,10^{-6}$	**3.11**

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(P)$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of ∂P and moments inside P;
  - the behavior on ∂P is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on ∂P is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;

- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on $\partial\mathrm{P}$ is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - ”dofs” are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on $\partial\mathrm{P}$ is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on ∂P is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on $\partial\mathrm{P}$ is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on $\partial\mathrm{P}$ is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on $\partial\mathrm{P}$ is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on $\partial\mathrm{P}$ is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on $\partial\mathrm{P}$ is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM is a *family of schemes* on polygonal meshes: new schemes are generated by changing the stabilization term;
- VEM works for any order of accuracy:
  - we can use $\mathbb{P}_k(\mathrm{P})$ polynomials for the local VE space;
  - "dofs" are vertex values, nodal values on the edges of $\partial\mathrm{P}$ and moments inside P;
  - the behavior on $\partial\mathrm{P}$ is given by a polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.
- VEM works for any order of regularity:
  - we use also derivatives as degrees of freedom at vertices and edge nodes
  - the behavior on $\partial\mathrm{P}$ is given by a Hermite-like polynomial interpolation;
  - optimal error estimates in the energy norm are confirmed by experiments.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.
- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - [illegible] freedom (dofs);
  - $C^0 - \mathbb{P}_m$ ($m > 1$) requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.
- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
    - meshes with convex and non-convex elements;
    - meshes with very stretched elements;
    - meshes with hanging nodes;
    - meshes with collapsing nodes.

- VEM can be generalized to 3-D polyhedral mesh (in progress):
    - [illegible] freedom (dofs);
    - $C^0 \cap \mathbb{P}_m$ ($m > 1$) requires vertex values and moments on edges, faces, and inside P;
    - no need of numerical integration. VEM does not use the basis functions explicitly;
    - no need of isoparametric mappings, VEM works in the physical domain.

- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.

- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - [illegible] freedom (dof's);
  - $C^0 - \mathbb{P}_m$ $(m > 1)$ requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.

- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.
- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - $C^0 - \mathbb{P}_1$ works in 3-D just using vertex values as degrees of freedom (dofs);
  - $C^0 - \mathbb{P}_m$ ($m > 1$) requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.
- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.

- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - $C^0-\mathbb{P}_1$ works in 3-D just using vertex values as degrees of freedom (dofs);
  - $C^0-\mathbb{P}_m$ ($m>1$) requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.

- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.
- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - $C^0 - \mathbb{P}_1$ works in 3-D just using vertex values as degrees of freedom (dofs);
  - $C^0 - \mathbb{P}_m$ ($m > 1$) requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.
- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.

- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - $C^0 - \mathbb{P}_1$ works in 3-D just using vertex values as degrees of freedom (dofs);
  - $C^0 - \mathbb{P}_m$ $(m > 1)$ requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.

- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.
- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - $C^0 - \mathbb{P}_1$ works in 3-D just using vertex values as degrees of freedom (dofs);
  - $C^0 - \mathbb{P}_m$ ($m > 1$) requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.
- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.
- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - $C^0 - \mathbb{P}_1$ works in 3-D just using vertex values as degrees of freedom (dofs);
  - $C^0 - \mathbb{P}_m$ $(m > 1)$ requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.
- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Summary

- VEM works on degenerate meshes (experiments):
  - meshes with convex and non-convex elements;
  - meshes with very stretched elements;
  - meshes with hanging nodes;
  - meshes with collapsing nodes.
- VEM can be generalized to 3-D polyhedral mesh (in progress):
  - $C^0 - \mathbb{P}_1$ works in 3-D just using vertex values as degrees of freedom (dofs);
  - $C^0 - \mathbb{P}_m$ $(m > 1)$ requires vertex values and moments on edges, faces, and inside P;
  - no need of numerical integration, VEM does not use the basis functions explicitly;
  - no need of isoparametric mappings, VEM works in the physical domain.
- There is no difference between VEM and the mimetic finite difference method, the two families of schemes coincide.

Current/future developments

- full extension to three dimensional problems;
- other differential equations: elasticity, advection-diffusion, Stokes, etc;
- understand the role of the mimetic stabilization;
- justify the numerical results for degenerate meshes (not covered by the theory);

Thank for your attention.

Current/future developments

- full extension to three dimensional problems;
- other differential equations: elasticity, advection-diffusion, Stokes, etc;
- understand the role of the mimetic stabilization;
- justify the numerical results for degenerate meshes (not covered by the theory);

Thank for your attention.

Current/future developments

- full extension to three dimensional problems;
- other differential equations: elasticity, advection-diffusion, Stokes, etc;
- understand the role of the mimetic stabilization;
- justify the numerical results for degenerate meshes (not covered by the theory);

Thank for your attention.

Current/future developments

- full extension to three dimensional problems;
- other differential equations: elasticity, advection-diffusion, Stokes, etc;
- understand the role of the mimetic stabilization;
- justify the numerical results for degenerate meshes (not covered by the theory);

Thank for your attention.

Current/future developments

- full extension to three dimensional problems;
- other differential equations: elasticity, advection-diffusion, Stokes, etc;
- understand the role of the mimetic stabilization;
- justify the numerical results for degenerate meshes (not covered by the theory);

Thank for your attention.

Current/future developments

- full extension to three dimensional problems;
- other differential equations: elasticity, advection-diffusion, Stokes, etc;
- understand the role of the mimetic stabilization;
- justify the numerical results for degenerate meshes (not covered by the theory);

Thank for your attention.