Agglomeration and refinement of polytopic meshes for DGFEM & VEM

Scott Congreve

Department of Numerical Mathematics,
Faculty of Mathematics & Physics,
Charles University

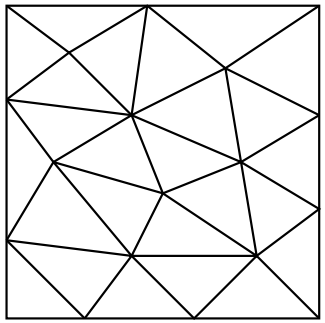
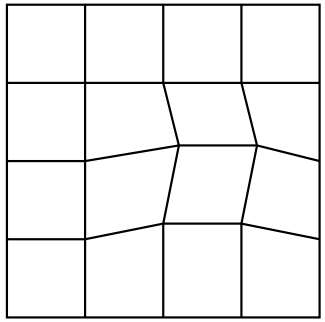
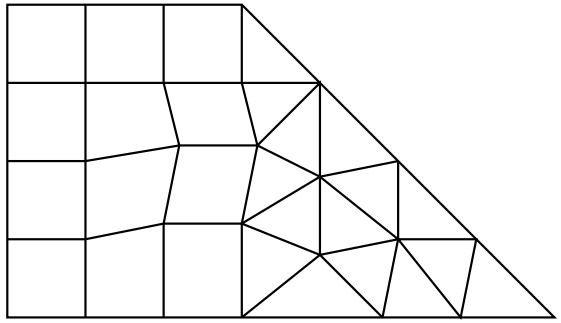
Primus project — PRIMUS/22/SCI/014

https://www2.karlin.mff.cuni.cz/~congreve/primus/

Overview of standard meshes

For finite element (and similar) methods we partition the domain into a mesh consisting of elements.

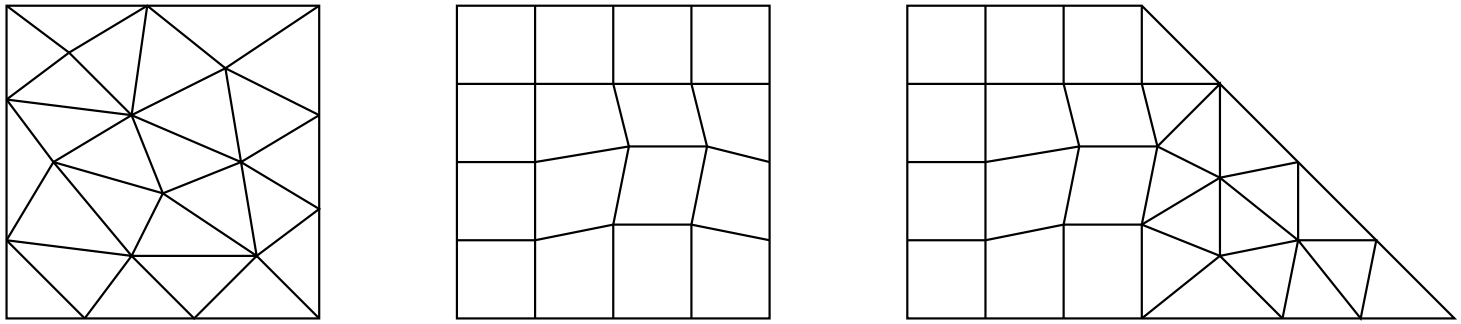
In 2D we use triangles (simplices) and quads:

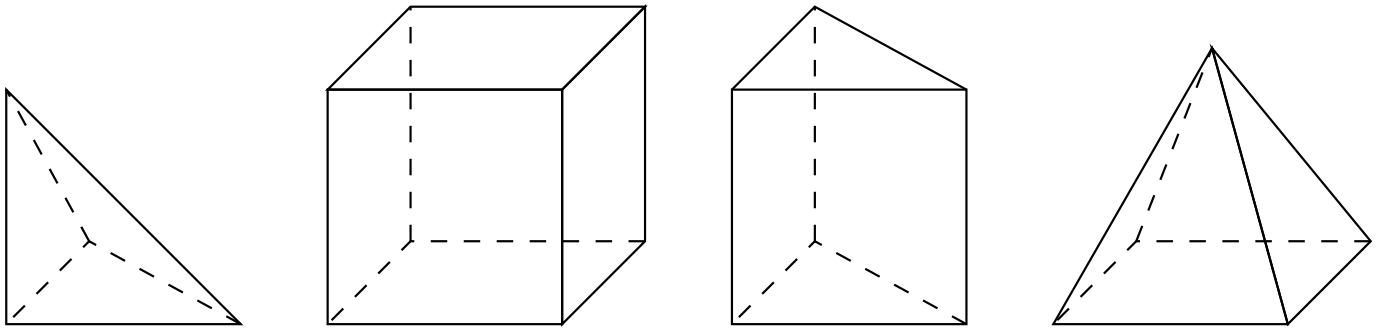
Overview of standard meshes

For finite element (and similar) methods we partition the domain into a mesh consisting of elements.

In 2D we use triangles (simplices) and quads:



In 3D we use tetrahedrons (simplices), hexahedrons, prisms and pyramids.



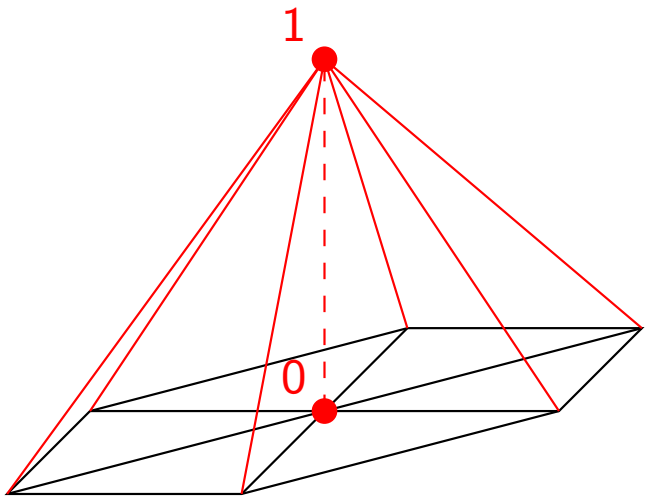
Overview of standard meshes

For classical conforming or non-conforming finite element methods we generally limit meshes to these standard elements, as it is possible to construct basis functions such that the functions are continuous over the whole domain, or for non-conforming methods met certain conditions on the edges (or similar).
The construction these functions is such that we know their definition over the whole element.

Overview of standard meshes

For classical conforming or non-conforming finite element methods we generally limit meshes to these standard elements, as it is possible to construct basis functions such that the functions are continuous over the whole domain, or for non-conforming methods met certain conditions on the edges (or similar).
The construction these functions is such that we know their definition over the whole element.

For example, for triangles we can create (linear) basis functions for a conforming finite element method by using the hat functions of each vertex:



Polytopic meshes

The idea of polytopic elements is instead to construct meshes consisting of

- general simple polygonal elements in 2D, where each polygon has a finite number of straight line segments,
- general polyhedral elements in 3D with of a finite number of simple polygonal faces.

Polytopic meshes

The idea of polytopic elements is instead to construct meshes consisting of

- general simple polygonal elements in 2D, where each polygon has a finite number of straight line segments,
- general polyhedral elements in 3D with of a finite number of simple polygonal faces.

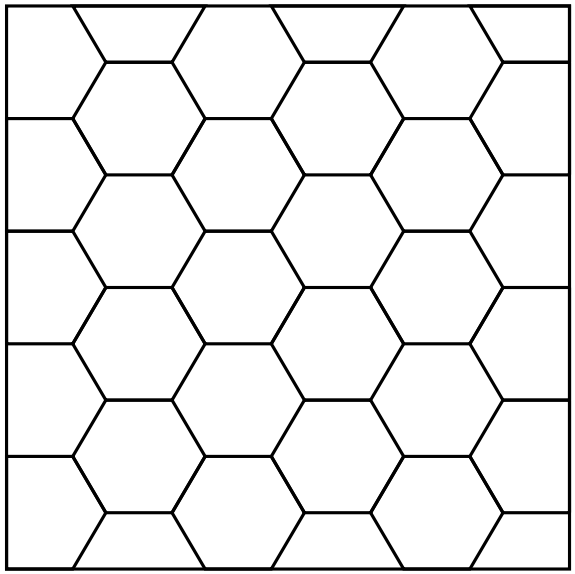
We call these meshes and elements polygonal (2D), polyhedral (3D), or polytopic (in general).

Polytopic meshes

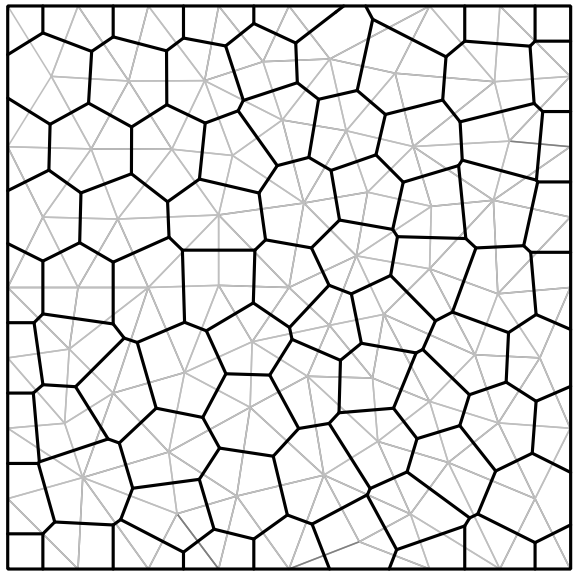
The idea of polytopic elements is instead to construct meshes consisting of

- general simple polygonal elements in 2D, where each polygon has a finite number of straight line segments,
- general polyhedral elements in 3D with of a finite number of simple polygonal faces.

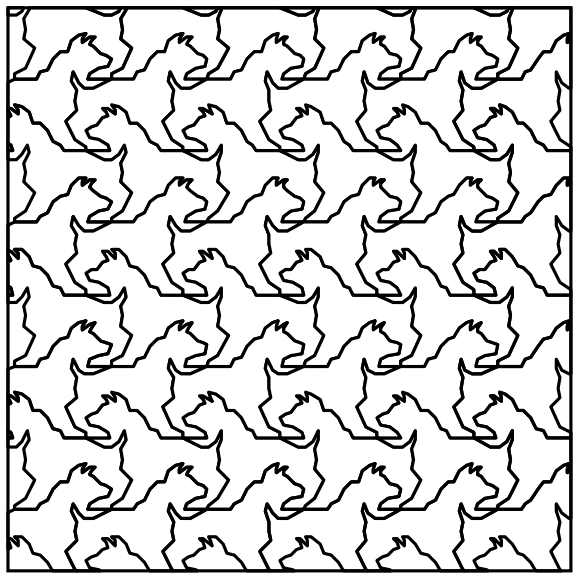
We call these meshes and elements polygonal (2D), polyhedral (3D), or polytopic (in general).



Hexagonal



Voronoi



Escher-like
[Paulino & Gain, 2015]

Methods on polytopic meshes

- Voronoi cell finite element method (VCFEM)
- Conforming polygonal finite element method ((conforming) PFEM)
- *n*-Sided polygonal smoothed finite element method (*n*SFEM)
- Polygonal scaled boundary finite element method (PSBFEM)
- Base forces element method (BFEM)
- Boundary element based FEM (BEM-based FEM)
- Mimetic finite difference (MFD)
- Virtual element method (VEM)
- Virtual node method (VNM)
- Discontinuous Galerkin finite element method (DGFEM)
- Trefftz/Hybrid Trefftz polygonal finite element (T-FEM or HT-FEM)
- Trefftz Discontinuous Galerkin finite element method (TDGFEM)
- Hybrid stress-function (HS-F) polygonal element
- Hybrid discontinuous Galerkin methods (HDG)
- Hybrid higher-order methods (HHO)
- Hybrid finite volume method (HFV)
- Weak Galerkin methods (WG)

Methods on polytopic meshes

- Voronoi cell finite element method (VCFEM)
- Conforming polygonal finite element method ((conforming) PFEM)
- *n*-Sided polygonal smoothed finite element method (*n*SFEM)
- Polygonal scaled boundary finite element method (PSBFEM)
- Base forces element method (BFEM)
- Boundary element based FEM (BEM-based FEM)
- Mimetic finite difference (MFD)
- Virtual element method (VEM)
- Virtual node method (VNM)
- Discontinuous Galerkin finite element method (DGFEM)
- Trefftz/Hybrid Trefftz polygonal finite element (T-FEM or HT-FEM)
- Trefftz Discontinuous Galerkin finite element method (TDGFEM)
- Hybrid stress-function (HS-F) polygonal element
- Hybrid discontinuous Galerkin methods (HDG)
- Hybrid higher-order methods (HHO)
- Hybrid finite volume method (HFV)
- Weak Galerkin methods (WG)

Advantages of polygonal meshes

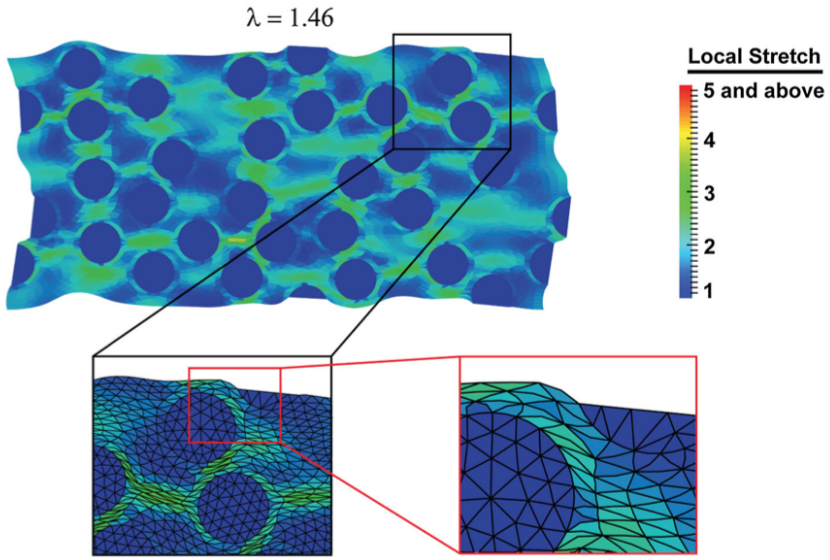
- Easier to capture complex domain geometry, or geometry/material naturally polygonal in nature (organic cells, carbon allotropes, etc.)

Advantages of polygonal meshes

- Easier to capture complex domain geometry, or geometry/material naturally polygonal in nature (organic cells, carbon allotropes, etc.)
- Possible to generate series of hierarchical meshes for unstructured meshes

Advantages of polygonal meshes

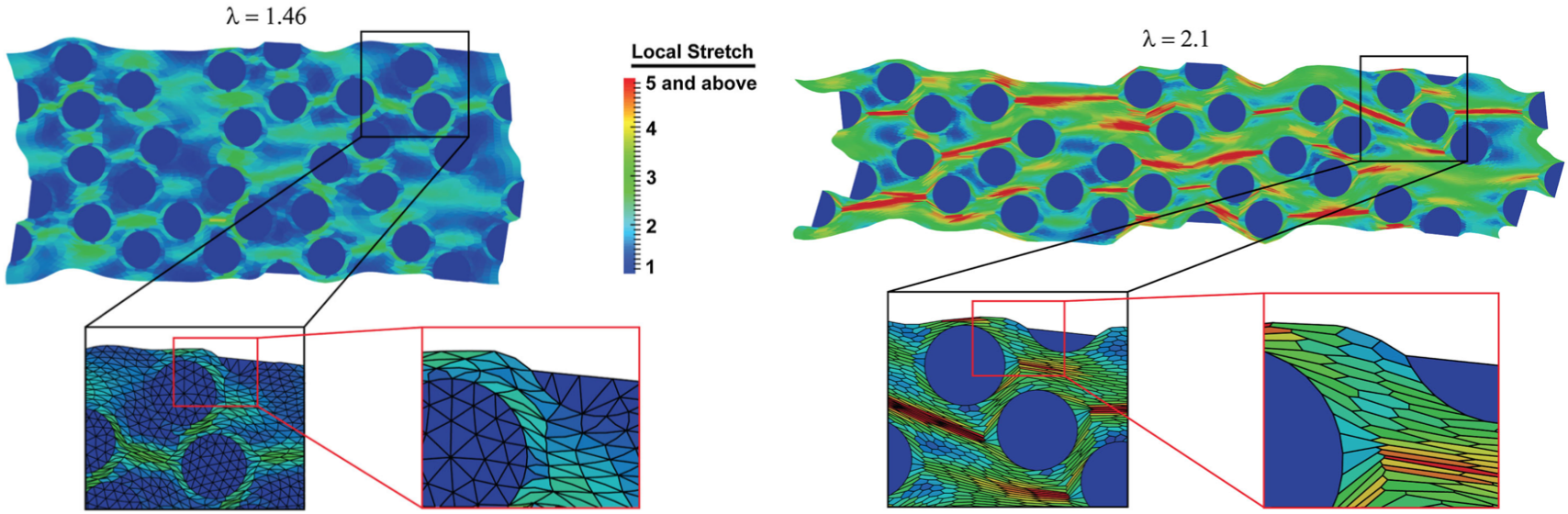
- Easier to capture complex domain geometry, or geometry/material naturally polygonal in nature (organic cells, carbon allotropes, etc.)
- Possible to generate series of hierarchical meshes for unstructured meshes
- FEM on triangular meshes cannot model maximal stretch factors in linear elasticity due to numerical errors from the deformation; whereas, polygonal meshes can allow more realistic stretch factors due to greater admissible shape deformations



[Chi et al., 2015;]

Advantages of polygonal meshes

- Easier to capture complex domain geometry, or geometry/material naturally polygonal in nature (organic cells, carbon allotropes, etc.)
- Possible to generate series of hierarchical meshes for unstructured meshes
- FEM on triangular meshes cannot model maximal stretch factors in linear elasticity due to numerical errors from the deformation; whereas, polygonal meshes can allow more realistic stretch factors due to greater admissible shape deformations



[Chi et al., 2015; Talischi et al., 2015]

- For some methods a polygonal element can be more stable or give better results.

Trefftz DGFEM for Helmholtz

Let $\Omega \subset \mathbb{R}^2$, be a bounded polygonal domain.

$$
\begin{aligned}
-\Delta u - k^2 u &= 0 && \text{in } \Omega, \\
u &= 0 && \text{on } \Gamma_D, && \text{(sound-soft scattering)} \\
\nabla u \cdot \boldsymbol{n} &= 0 && \text{on } \Gamma_N, && \text{(sound-hard scattering)} \\
\nabla u \cdot \boldsymbol{n} + ik\vartheta u &= g_R && \text{on } \Gamma_R.
\end{aligned}
$$

Trefftz DGFEM for Helmholtz

Let $\Omega \subset \mathbb{R}^2$, be a bounded polygonal domain.

$$
\begin{aligned}
-\Delta u - k^2 u &= 0 && \text{in } \Omega, \\
u &= 0 && \text{on } \Gamma_D, && \text{(sound-soft scattering)} \\
\nabla u \cdot \boldsymbol{n} &= 0 && \text{on } \Gamma_N, && \text{(sound-hard scattering)} \\
\nabla u \cdot \boldsymbol{n} + ik\vartheta u &= g_R && \text{on } \Gamma_R.
\end{aligned}
$$

Polynomial DG Finite Element Spaces: DGFEM uses polynomial basis functions defined on a reference element $\widehat{K}$:

$$V_{dg}^q(\mathcal{T}_h) := \{v \in L^2(\Omega) : v|_K \circ F_K \in \mathcal{P}_{q_K}(\widehat{K}), K \in \mathcal{T}_h\}.$$

Trefftz DGFEM for Helmholtz

Let $\Omega \subset \mathbb{R}^2$, be a bounded polygonal domain.

$$\begin{aligned} -\Delta u - k^2 u &= 0 && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma_D, && \text{(sound-soft scattering)} \\ \nabla u \cdot \boldsymbol{n} &= 0 && \text{on } \Gamma_N, && \text{(sound-hard scattering)} \\ \nabla u \cdot \boldsymbol{n} + ik\vartheta u &= g_R && \text{on } \Gamma_R. \end{aligned}$$

Polynomial DG Finite Element Spaces: DGFEM uses polynomial basis functions defined on a reference element $\widehat{K}$:

$$V_{dg}^q(\mathcal{T}_h) := \{v \in L^2(\Omega) : v|_K \circ F_K \in \mathcal{P}_{q_K}(\widehat{K}), K \in \mathcal{T}_h\}.$$

Trefftz Finite Element Space: Use basis functions based on general solutions to the PDE. First define the local and global Trefftz spaces

$$T(K) := \{v|_K : -\Delta u - k^2 u = 0\} \qquad T(\mathcal{T}_h) := \{v \in L^2(\Omega) : v|_K \in T(K), K \in \mathcal{T}_h\}.$$

Trefftz DGFEM for Helmholtz

Let $\Omega \subset \mathbb{R}^2$, be a bounded polygonal domain.

$$\begin{aligned} -\Delta u - k^2 u &= 0 && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma_D, && \text{(sound-soft scattering)} \\ \nabla u \cdot \boldsymbol{n} &= 0 && \text{on } \Gamma_N, && \text{(sound-hard scattering)} \\ \nabla u \cdot \boldsymbol{n} + ik\vartheta u &= g_R && \text{on } \Gamma_R. \end{aligned}$$

Polynomial DG Finite Element Spaces: DGFEM uses polynomial basis functions defined on a reference element $\widehat{K}$:

$$V_{dg}^q(\mathcal{T}_h) := \{v \in L^2(\Omega) : v|_K \circ F_K \in \mathcal{P}_{q_K}(\widehat{K}), K \in \mathcal{T}_h\}.$$

Trefftz Finite Element Space: Use basis functions based on general solutions to the PDE. First define the local and global Trefftz spaces

$$T(K) := \{v|_K : -\Delta u - k^2 u = 0\} \qquad T(\mathcal{T}_h) := \{v \in L^2(\Omega) : v|_K \in T(K), K \in \mathcal{T}_h\}.$$

Let $V_p(K) \subset T(K)$ be a finite dimensional local space; then, the Trefftz FE Space is

$$V_{tdg}^p(\mathcal{T}_h) := \{v \in T(\mathcal{T}_h) : v_K \in V_p(K), K \in \mathcal{T}_h\}.$$

Let $\Omega \subset \mathbb{R}^2$, be a bounded polygonal domain.

$$\begin{aligned} -\Delta u - k^2 u &= 0 && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma_D, && \text{(sound-soft scattering)} \\ \nabla u \cdot \boldsymbol{n} &= 0 && \text{on } \Gamma_N, && \text{(sound-hard scattering)} \\ \nabla u \cdot \boldsymbol{n} + ik\vartheta u &= g_R && \text{on } \Gamma_R. \end{aligned}$$

Polynomial DG Finite Element Spaces: DGFEM uses polynomial basis functions defined on a reference element $\widehat{K}$:

$$V_{dg}^q(\mathcal{T}_h) := \{v \in L^2(\Omega) : v|_K \circ F_K \in \mathcal{P}_{q_K}(\widehat{K}), K \in \mathcal{T}_h\}.$$

Trefftz Finite Element Space: Use basis functions based on general solutions to the PDE. First define the local and global Trefftz spaces

$$T(K) := \{v|_K : -\Delta u - k^2 u = 0\} \qquad T(\mathcal{T}_h) := \{v \in L^2(\Omega) : v|_K \in T(K), K \in \mathcal{T}_h\}.$$

Let $V_p(K) \subset T(K)$ be a finite dimensional local space; then, the Trefftz FE Space is

$$V_{tdg}^p(\mathcal{T}_h) := \{v \in T(\mathcal{T}_h) : v_K \in V_p(K), K \in \mathcal{T}_h\}.$$

We can use plane waves as basis functions $\boldsymbol{x} \mapsto \mathrm{e}^{ik\boldsymbol{d}\cdot\boldsymbol{x}}$, where $\boldsymbol{d}$ is a direction vector. We can generate p basis functions by using p unique directions (usually roughly evenly spaced unit vectors).

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K (-\Delta u - k^2 u)\, \bar{v} \quad \mathrm{d}\boldsymbol{x} \qquad = 0$$

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K (\nabla u \cdot \nabla \bar{v} - k^2 u \bar{v}) \, \mathrm{d}\boldsymbol{x} \qquad - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K u(-\Delta \bar{v} - k^2 \bar{v}) \ \mathrm{d}\boldsymbol{x} + \int_{\partial K} u \nabla \bar{v} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K u(-\Delta \bar{v} - k^2 \bar{v}) \ \mathrm{d}\boldsymbol{x} + \int_{\partial K} u \nabla \bar{v} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K u(-\Delta \bar{v} - k^2 \bar{v}) \ \mathrm{d}\boldsymbol{x} + \int_{\partial K} u \nabla \bar{v} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

- Replace continuous functions by discrete approximations ($u_{hp}, v_{hp} \in V^p_{tdg}(\mathcal{T}_h)$) and traces by numerical fluxes

$$u \to \widehat{u}_{hp}, \qquad \nabla u \to ik\widehat{\boldsymbol{\sigma}}_{hp}.$$

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K u(-\Delta \bar{v} - k^2 \bar{v}) \ \mathrm{d}\boldsymbol{x} + \int_{\partial K} u \nabla \bar{v} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

- Replace continuous functions by discrete approximations ($u_{hp}, v_{hp} \in V^p_{tdg}(\mathcal{T}_h)$) and traces by numerical fluxes

$$u \to \widehat{u}_{hp}, \qquad \nabla u \to ik\widehat{\boldsymbol{\sigma}}_{hp}.$$

- $v_{hp} \in V_p(\mathcal{T}_h) \subset T(\mathcal{T}_h) \quad \Longrightarrow \quad -\Delta \bar{v}_{hp} - k^2 \bar{v}_{hp} = 0$ in K.

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K u(-\Delta \bar{v} - k^2 \bar{v}) \ \mathrm{d}\boldsymbol{x} + \int_{\partial K} u \nabla \bar{v} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

- Replace continuous functions by discrete approximations ($u_{hp}, v_{hp} \in V^p_{tdg}(\mathcal{T}_h)$) and traces by numerical fluxes

$$u \to \widehat{u}_{hp}, \qquad \nabla u \to ik\widehat{\boldsymbol{\sigma}}_{hp}.$$

- $v_{hp} \in V_p(\mathcal{T}_h) \subset T(\mathcal{T}_h) \quad \Longrightarrow \quad -\Delta \bar{v}_{hp} - k^2 \bar{v}_{hp} = 0$ in K.

$$\int_{\partial K} \widehat{u}_{hp} \nabla \bar{v}_{hp} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} ik\widehat{\boldsymbol{\sigma}}_{hp} \cdot \boldsymbol{n}_K \bar{v}_{hp} \, \mathrm{d}s = 0, \qquad \text{for all } K \in \mathcal{T}_h.$$

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K u(-\Delta \bar{v} - k^2 \bar{v}) \ \mathrm{d}\boldsymbol{x} + \int_{\partial K} u \nabla \bar{v} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

- Replace continuous functions by discrete approximations ($u_{hp}, v_{hp} \in V_{tdg}^p(\mathcal{T}_h)$) and traces by numerical fluxes

$$u \to \widehat{u}_{hp}, \qquad \nabla u \to ik\widehat{\boldsymbol{\sigma}}_{hp}.$$

- $v_{hp} \in V_p(\mathcal{T}_h) \subset T(\mathcal{T}_h) \quad \Longrightarrow \quad -\Delta \bar{v}_{hp} - k^2 \bar{v}_{hp} = 0$ in K.

$$\int_{\partial K} \widehat{u}_{hp} \nabla \bar{v}_{hp} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} ik\widehat{\boldsymbol{\sigma}}_{hp} \cdot \boldsymbol{n}_K \bar{v}_{hp} \, \mathrm{d}s = 0, \qquad \text{for all } K \in \mathcal{T}_h.$$

Select the numerical fluxes and summation over all elements in the mesh gives the formulation.

Trefftz DGFEM for Helmholtz

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K u(-\Delta \bar{v} - k^2 \bar{v}) \ \mathrm{d}\boldsymbol{x} + \int_{\partial K} u \nabla \bar{v} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

- Replace continuous functions by discrete approximations ($u_{hp}, v_{hp} \in V_{tdg}^p(\mathcal{T}_h)$) and traces by numerical fluxes

$$u \to \widehat{u}_{hp}, \qquad \nabla u \to ik\widehat{\boldsymbol{\sigma}}_{hp}.$$

- $v_{hp} \in V_p(\mathcal{T}_h) \subset T(\mathcal{T}_h) \quad \Longrightarrow \quad -\Delta \bar{v}_{hp} - k^2 \bar{v}_{hp} = 0$ in K.

$$\int_{\partial K} \widehat{u}_{hp} \nabla \bar{v}_{hp} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} ik\widehat{\boldsymbol{\sigma}}_{hp} \cdot \boldsymbol{n}_K \bar{v}_{hp} \, \mathrm{d}s = 0, \qquad \text{for all } K \in \mathcal{T}_h.$$

Select the numerical fluxes and summation over all elements in the mesh gives the formulation.
The resulting method only has integrals over edges; hence, it can operate on general polygonal meshes.

Given a mesh $\mathcal{T}_h$ on Ω we derive the TDGFEM as follows.

- Multiply by test functions and integrate by parts, element-wise, twice (ultra weak formulation):

$$\int_K u(-\Delta \bar{v} - k^2 \bar{v}) \ \mathrm{d}\boldsymbol{x} + \int_{\partial K} u \nabla \bar{v} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} \nabla u \cdot \boldsymbol{n}_K \bar{v} \, \mathrm{d}s = 0$$

- Replace continuous functions by discrete approximations ($u_{hp}, v_{hp} \in V^p_{tdg}(\mathcal{T}_h)$) and traces by numerical fluxes

$$u \to \widehat{u}_{hp}, \qquad \nabla u \to ik\widehat{\boldsymbol{\sigma}}_{hp}.$$

- $v_{hp} \in V_p(\mathcal{T}_h) \subset T(\mathcal{T}_h) \quad \Longrightarrow \quad -\Delta \bar{v}_{hp} - k^2 \bar{v}_{hp} = 0$ in K.

$$\int_{\partial K} \widehat{u}_{hp} \nabla \bar{v}_{hp} \cdot \boldsymbol{n}_K \, \mathrm{d}s - \int_{\partial K} ik\widehat{\boldsymbol{\sigma}}_{hp} \cdot \boldsymbol{n}_K \bar{v}_{hp} \, \mathrm{d}s = 0, \qquad \text{for all } K \in \mathcal{T}_h.$$

Select the numerical fluxes and summation over all elements in the mesh gives the formulation.

Trefftz DGFEM for Helmholtz

With plane wave basis functions the condition number of the resulting linear system blows up as the number of plane wave basis functions increase.

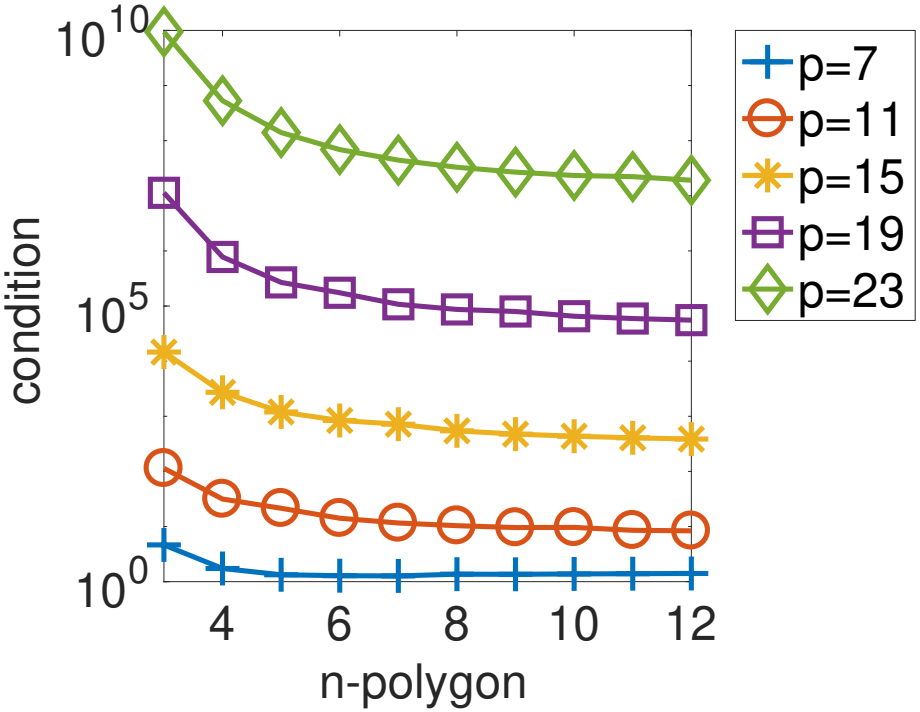
Trefftz DGFEM for Helmholtz

With plane wave basis functions the condition number of the resulting linear system blows up as the number of plane wave basis functions increase.

However, numerical experiments on a single n-polygon, for $n = 3, 4, 5, \ldots$, demonstrate that the condition number is generally better for higher n (in fact, closer to a circle the element becomes).

With plane wave basis functions the condition number of the resulting linear system blows up as the number of plane wave basis functions increase.

However, numerical experiments on a single n-polygon, for $n = 3, 4, 5, \ldots$, demonstrate that the condition number is generally better for higher n (in fact, closer to a circle the element becomes).



[C., Gedicke, & Perugia, 2017]

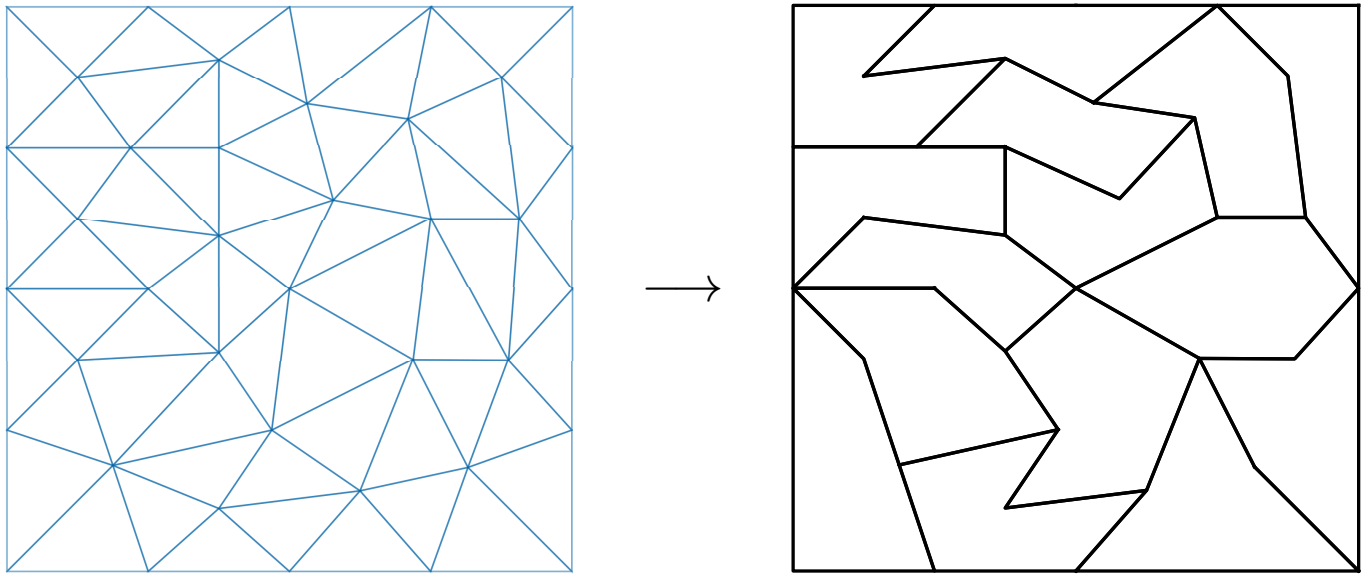
Agglomeration

As well as directly constructing polygonal meshes, it is also possible to generating mesh of polytopal elements by agglomeration.

Agglomeration

As well as directly constructing polygonal meshes, it is also possible to generating mesh of polytopal elements by agglomeration.

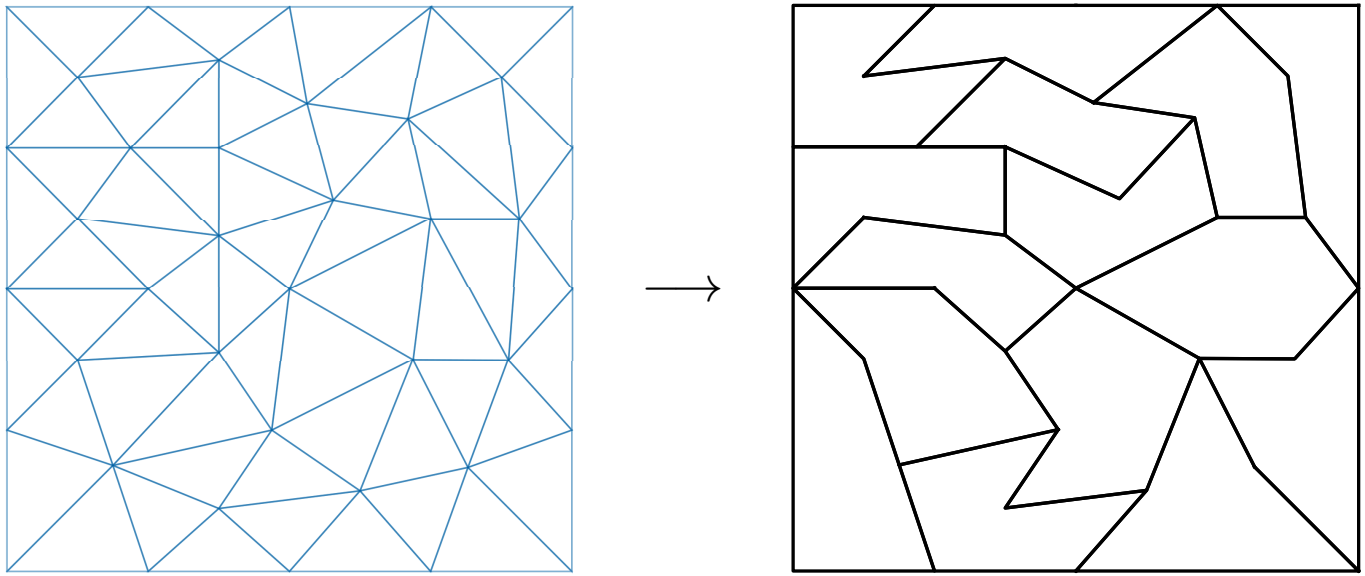
Here, consider an initial, known, mesh of standard (or polytopal) elements, and create a newer coarser mesh by joining (or agglomerating) neighbouring elements into a polytopal element:



Agglomeration

As well as directly constructing polygonal meshes, it is also possible to generating mesh of polytopal elements by agglomeration.

Here, consider an initial, known, mesh of standard (or polytopal) elements, and create a newer coarser mesh by joining (or agglomerating) neighbouring elements into a polytopal element:



This is particularly advantageous if we need multiple, hierarchical, meshes; e.g., for multiscale or multigrid.

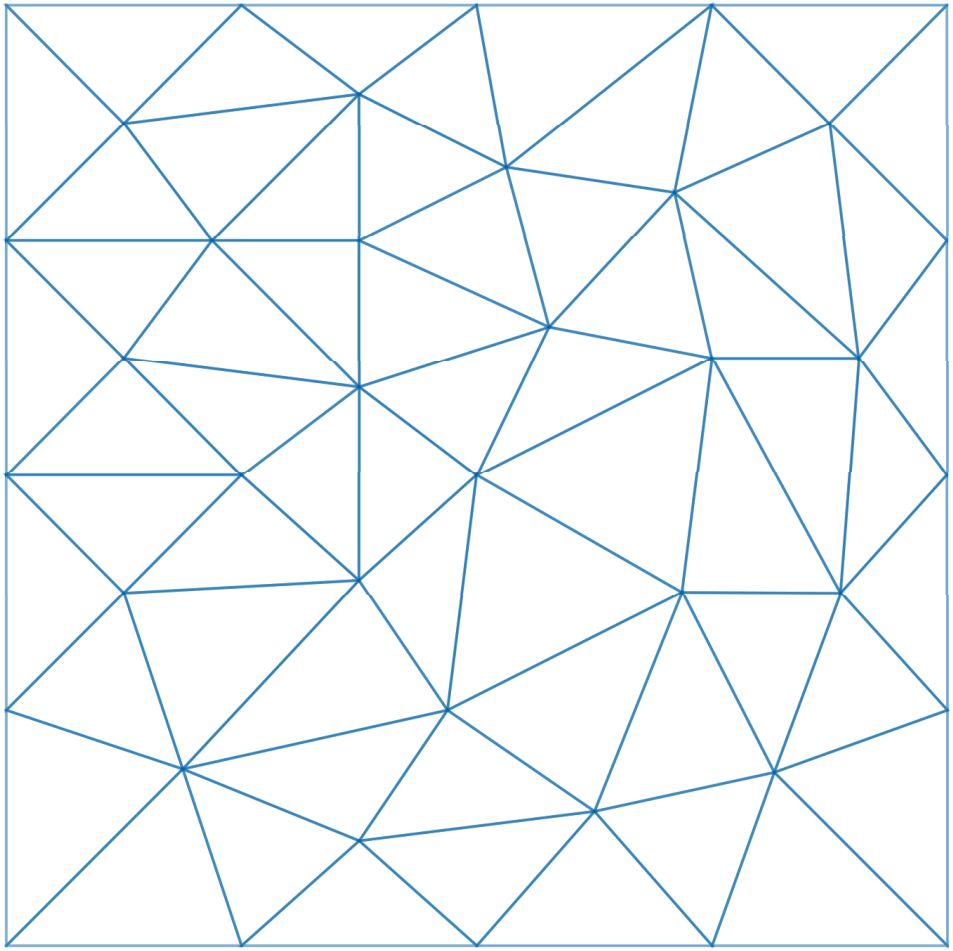
Multiple methods for agglomeration exist. An overview of application of some of these methods for multigrid is available in [Dargaville, Buchan, Smedley-Stevenson, Smith, & Pain, 2021].

Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].

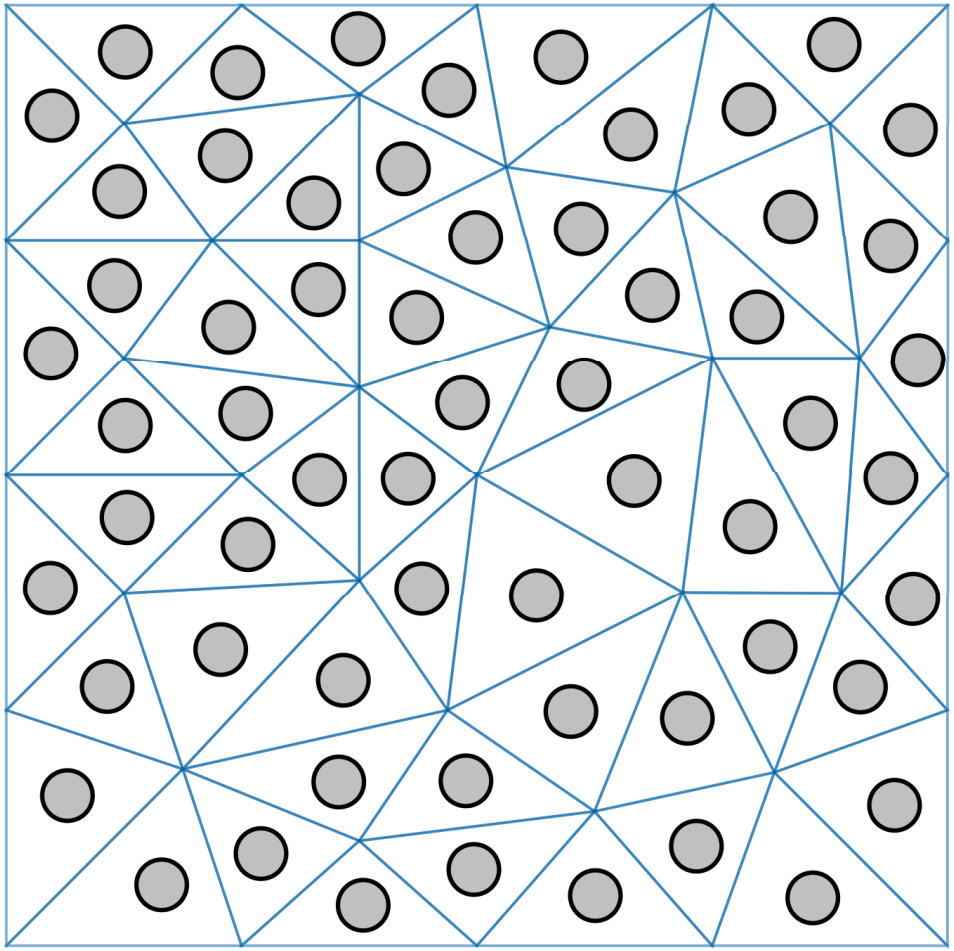
Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].



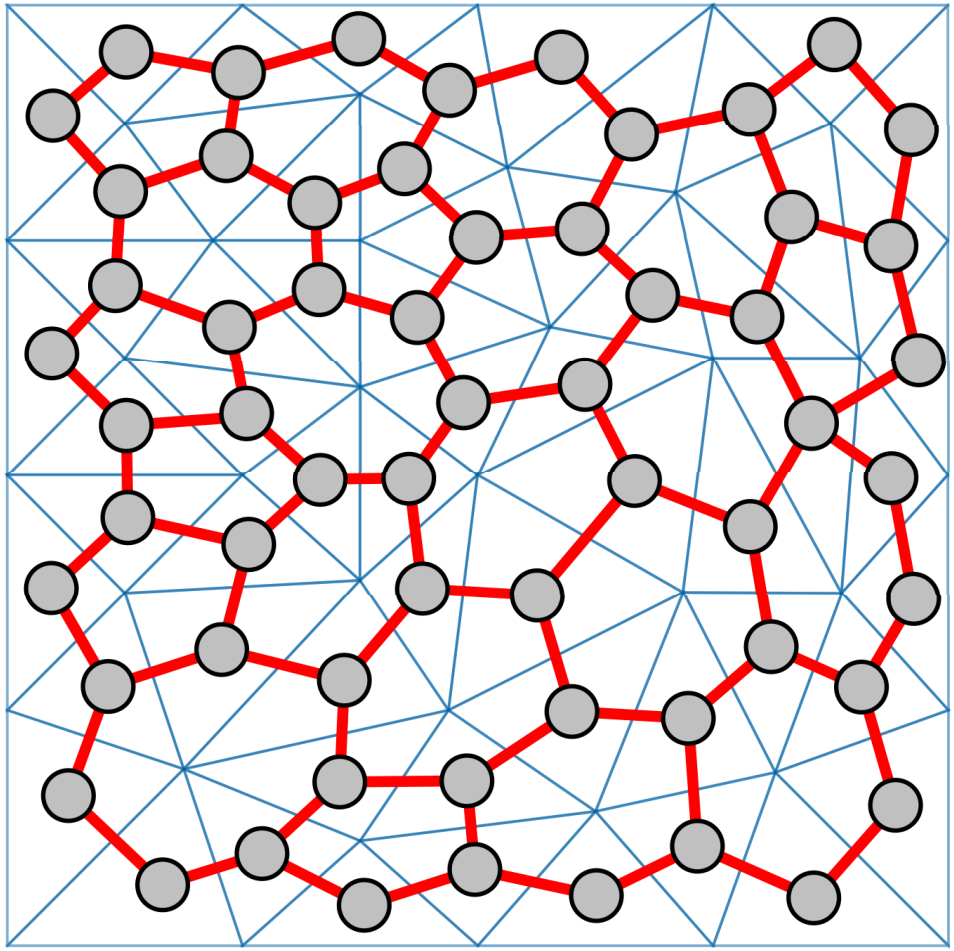
Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].



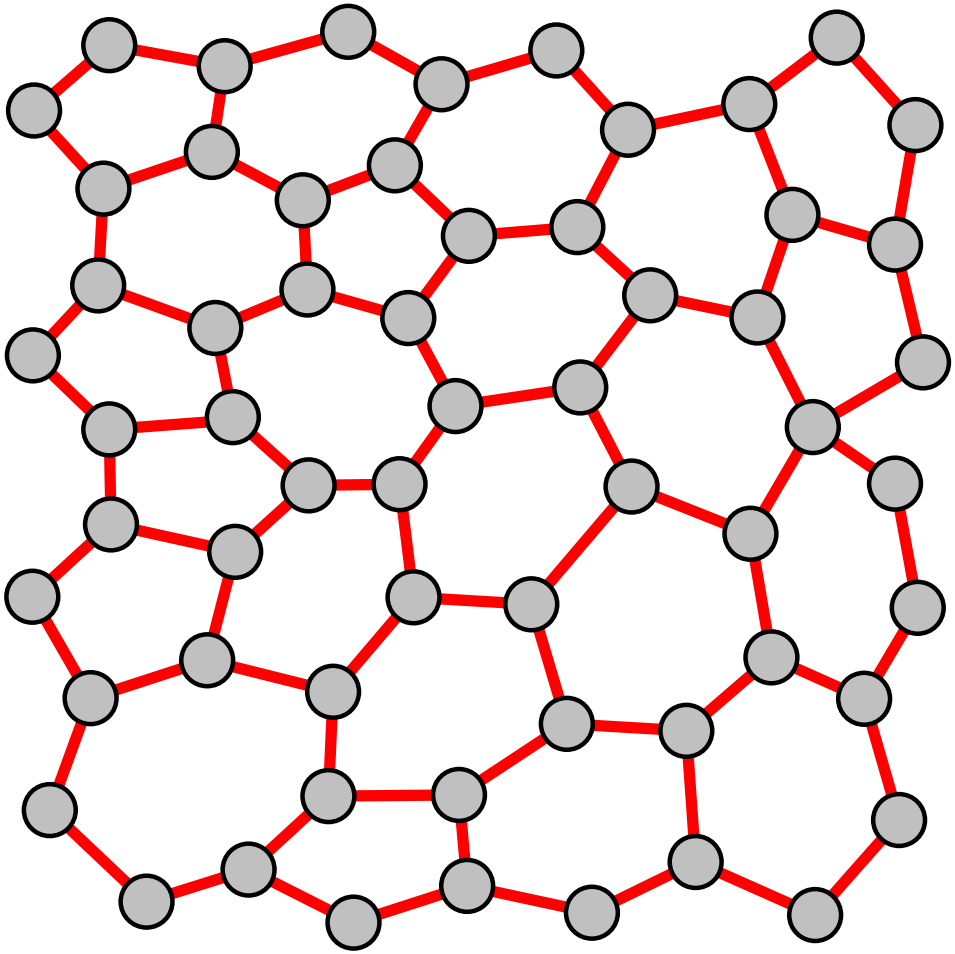
Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].



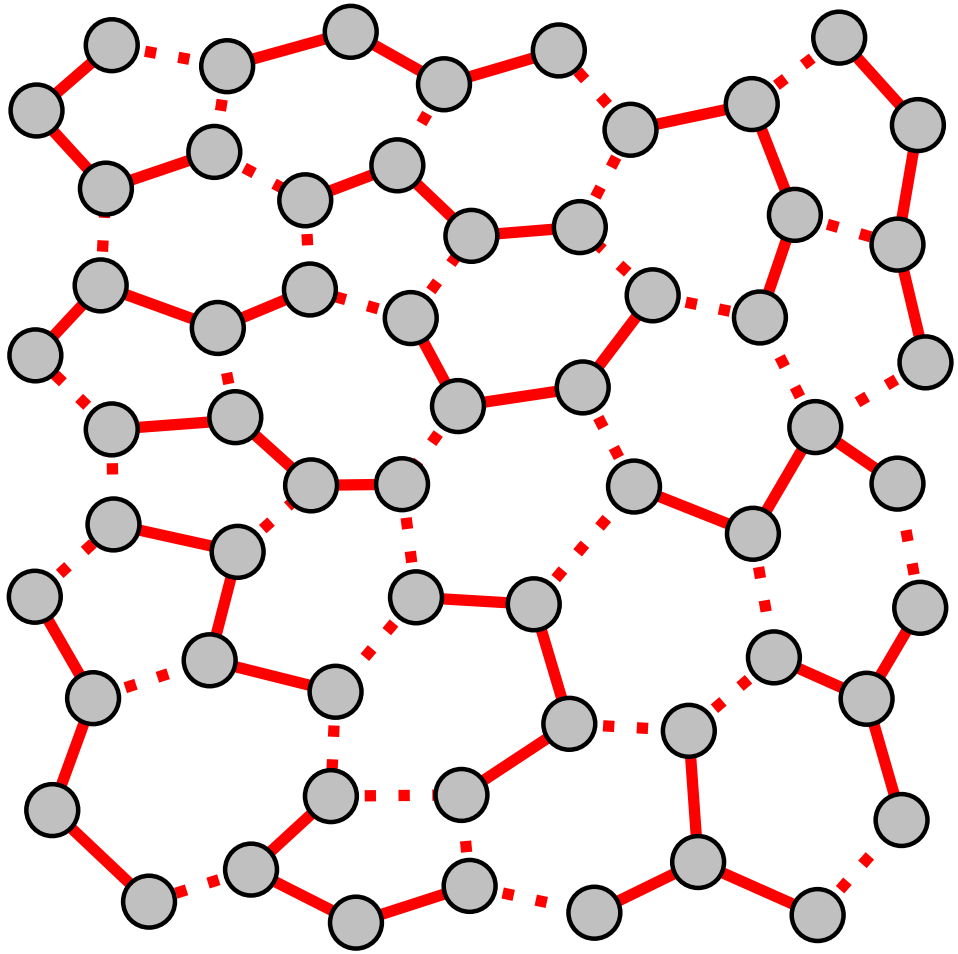
Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].

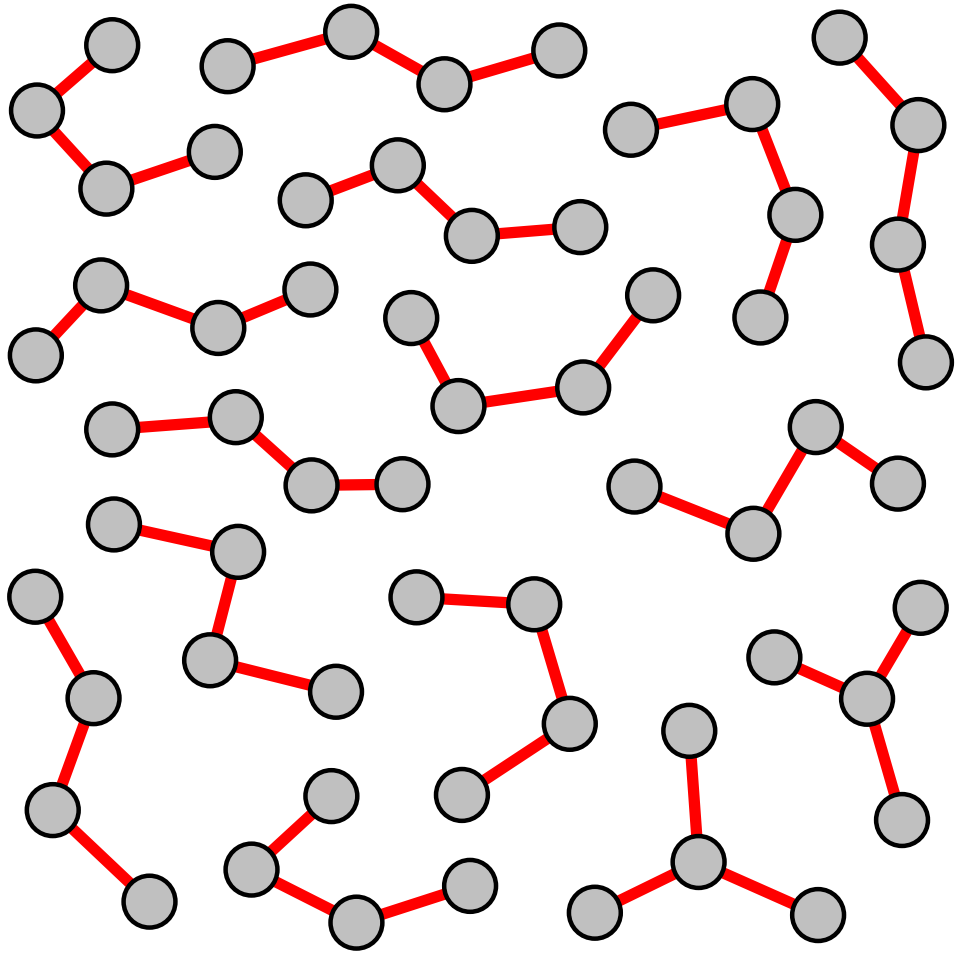


Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].

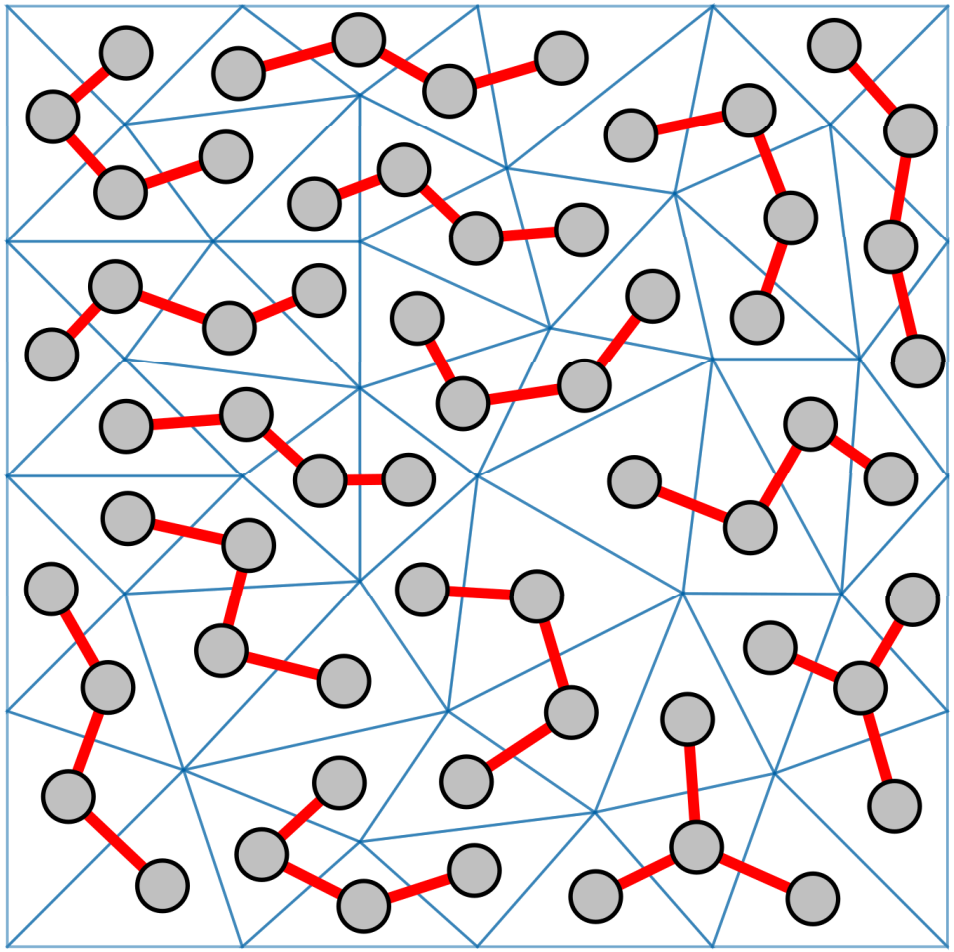


One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].

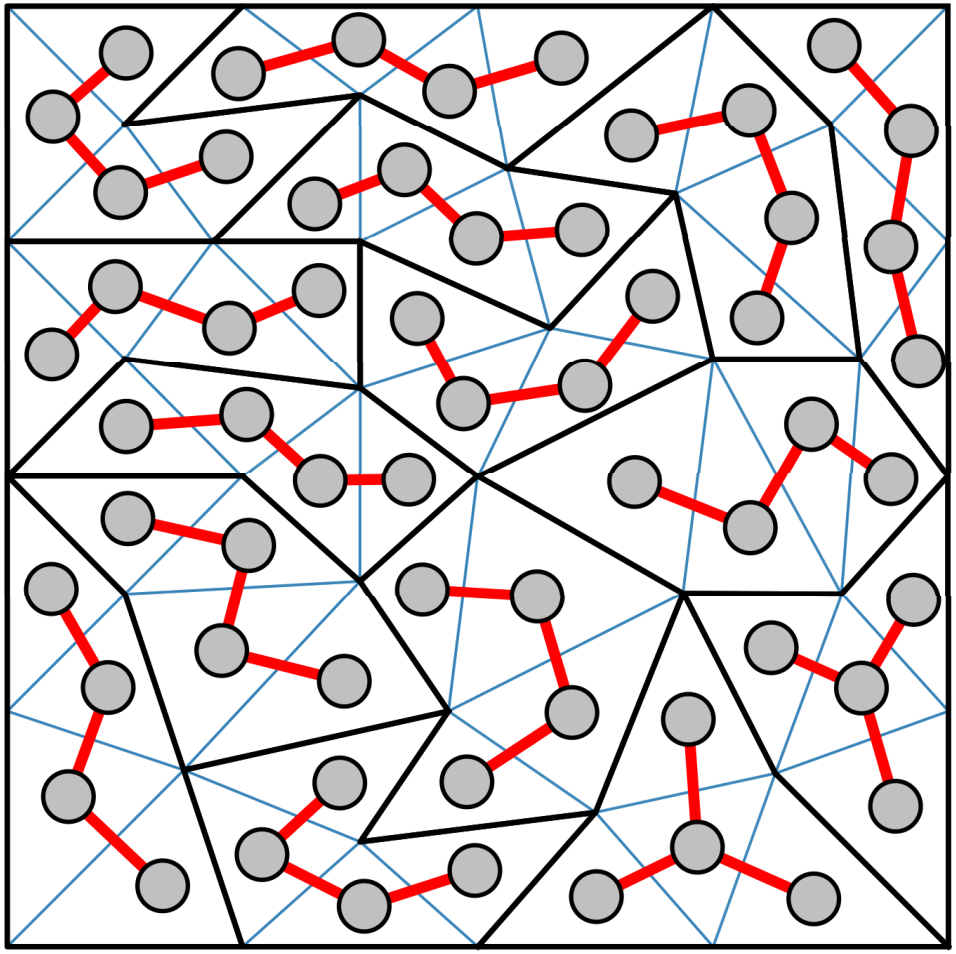


Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].

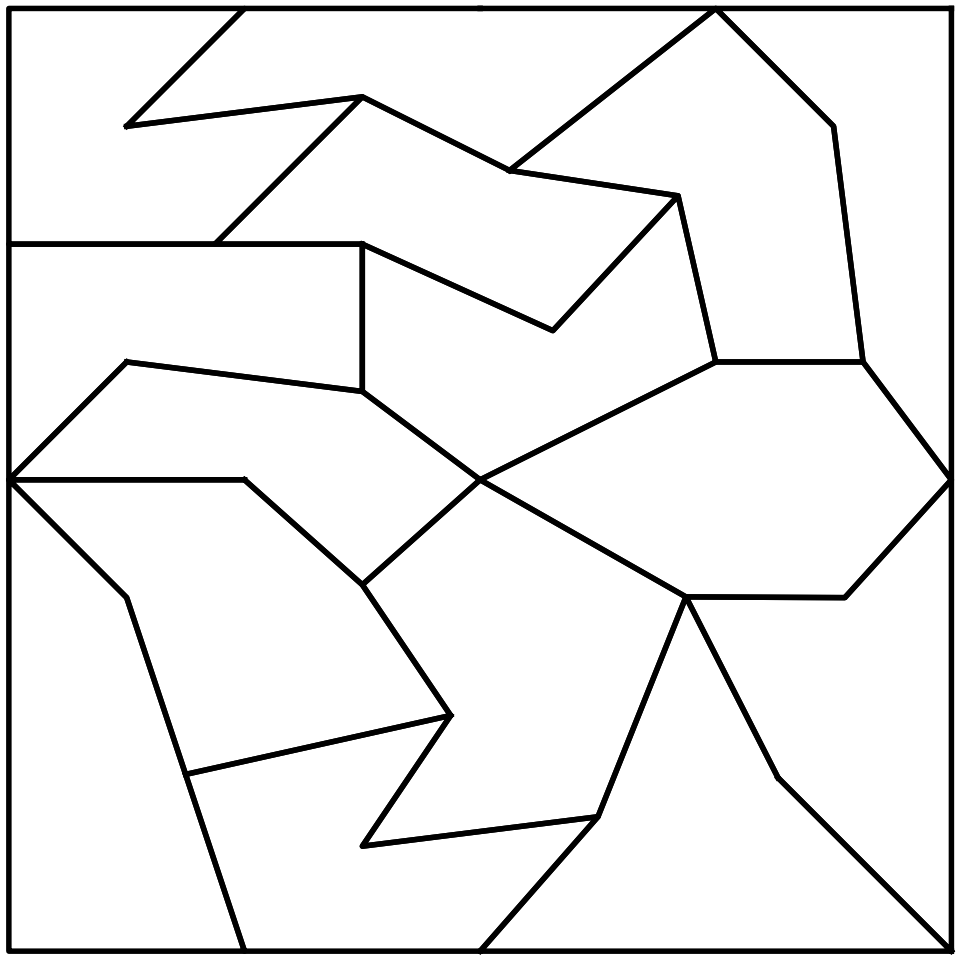


One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].



Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].



Graph partitioning

One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].

As this method minimises the edge cuts of the graph, and each edge cut corresponds to an edge in the agglomerated mesh, this method also minimises the number of edges in the mesh.

Graph partitioning

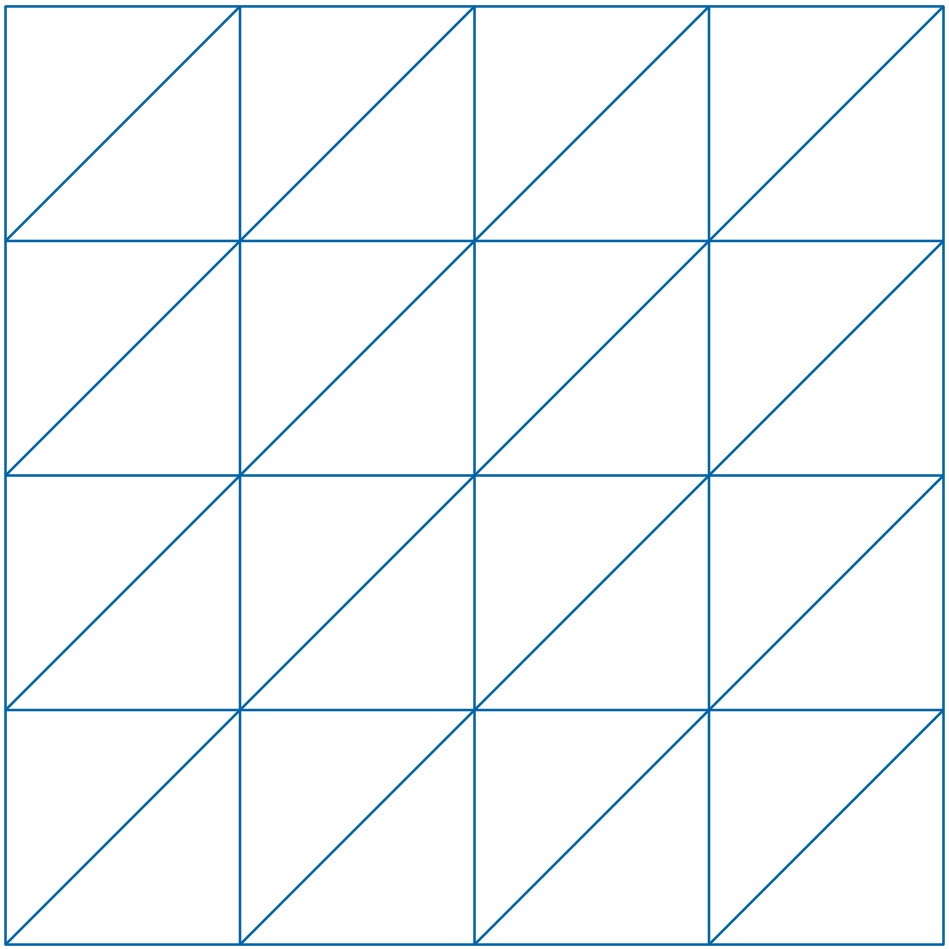
One technique for agglomeration is based on graph partitioning methods; e.g., METIS [Karypis & Kumar, 1999].

As this method minimises the edge cuts of the graph, and each edge cut corresponds to an edge in the agglomerated mesh, this method also minimises the number of edges in the mesh.

However, it operates with no concept of the mesh geometry; hence, there is no optimisation or guarantee on the shape, or properties, of the resulting elements.

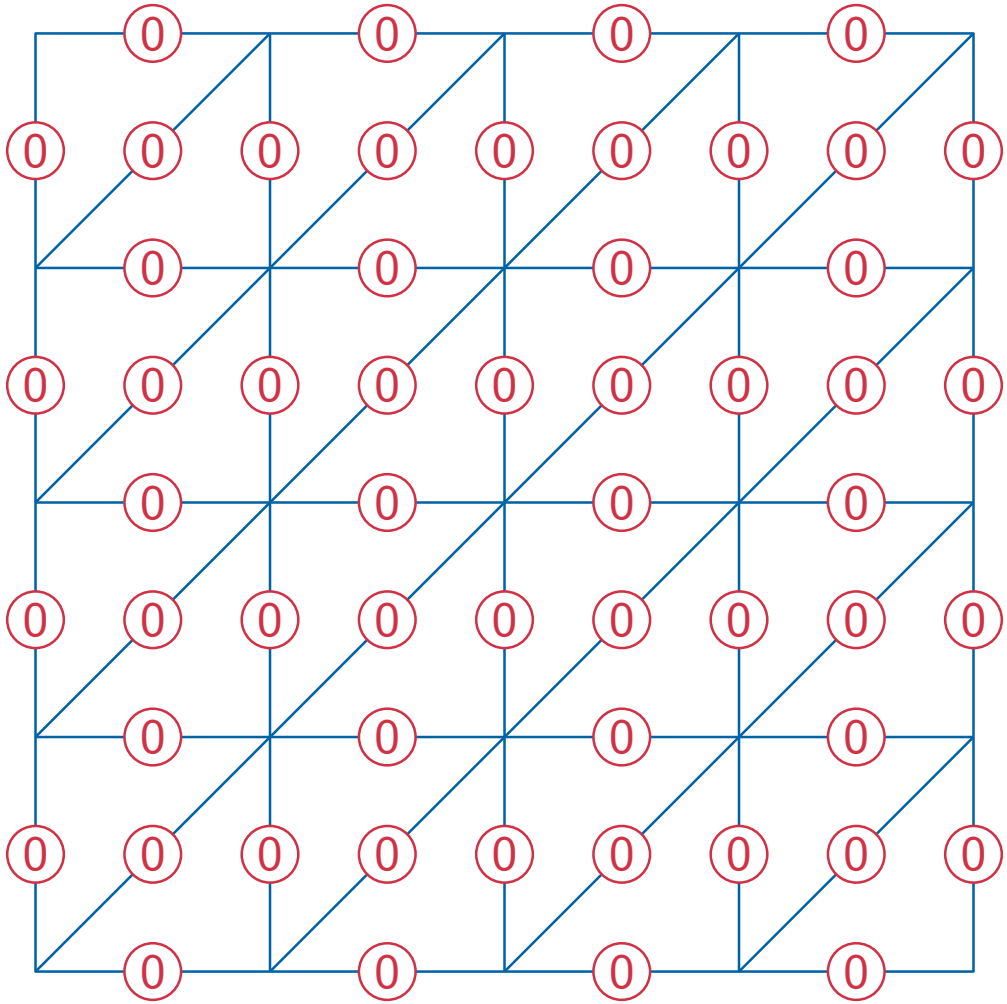
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



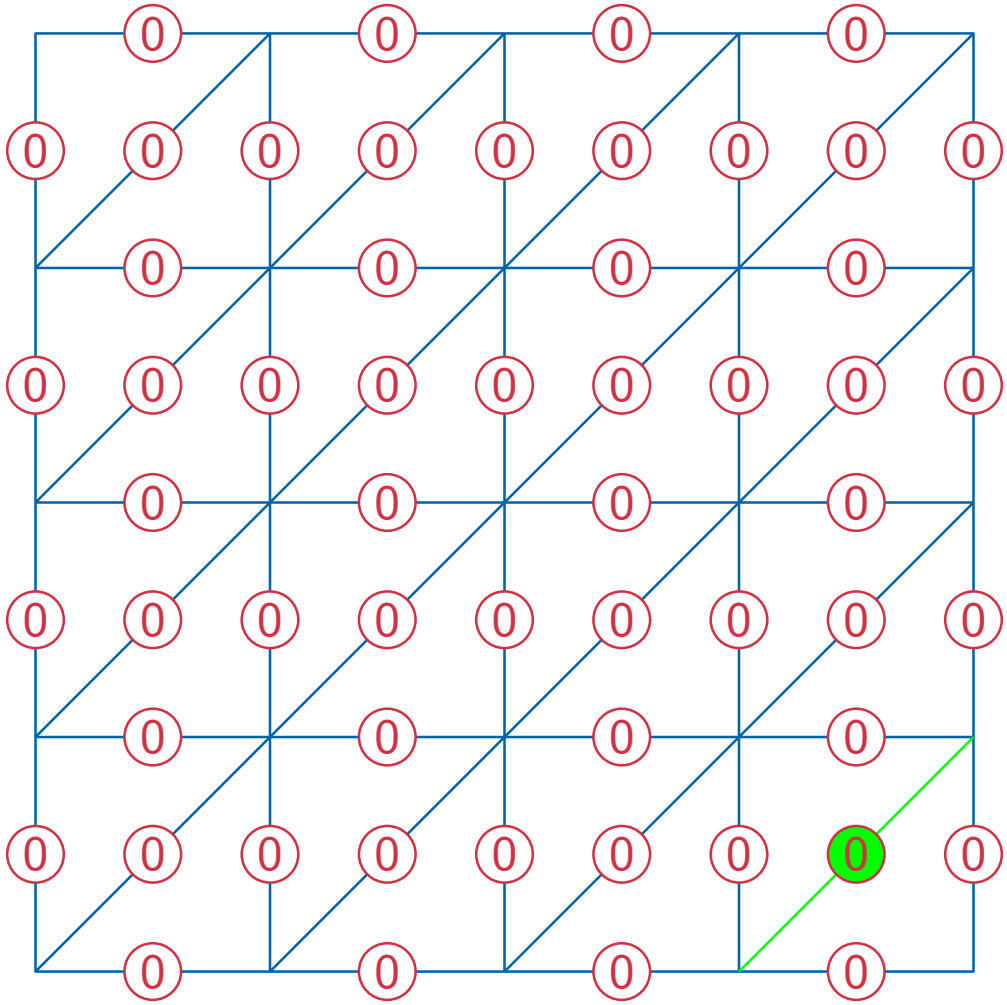
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



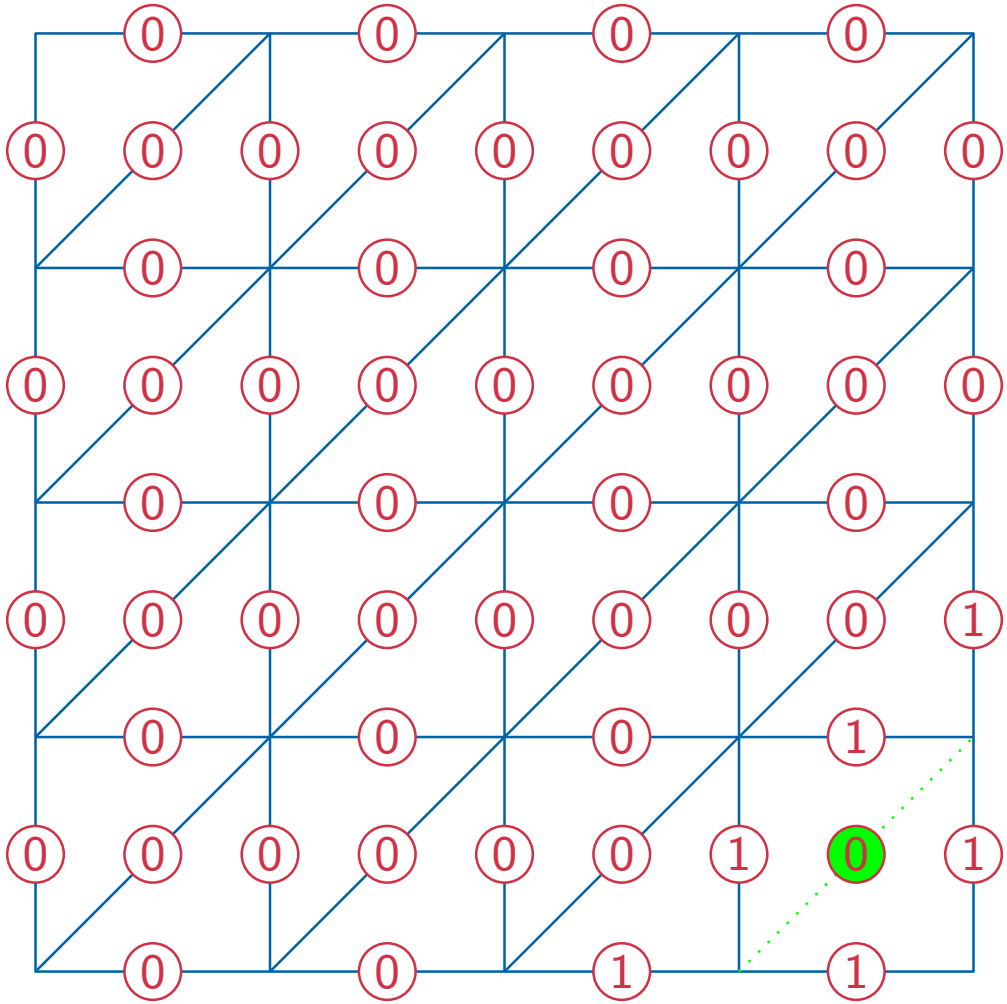
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



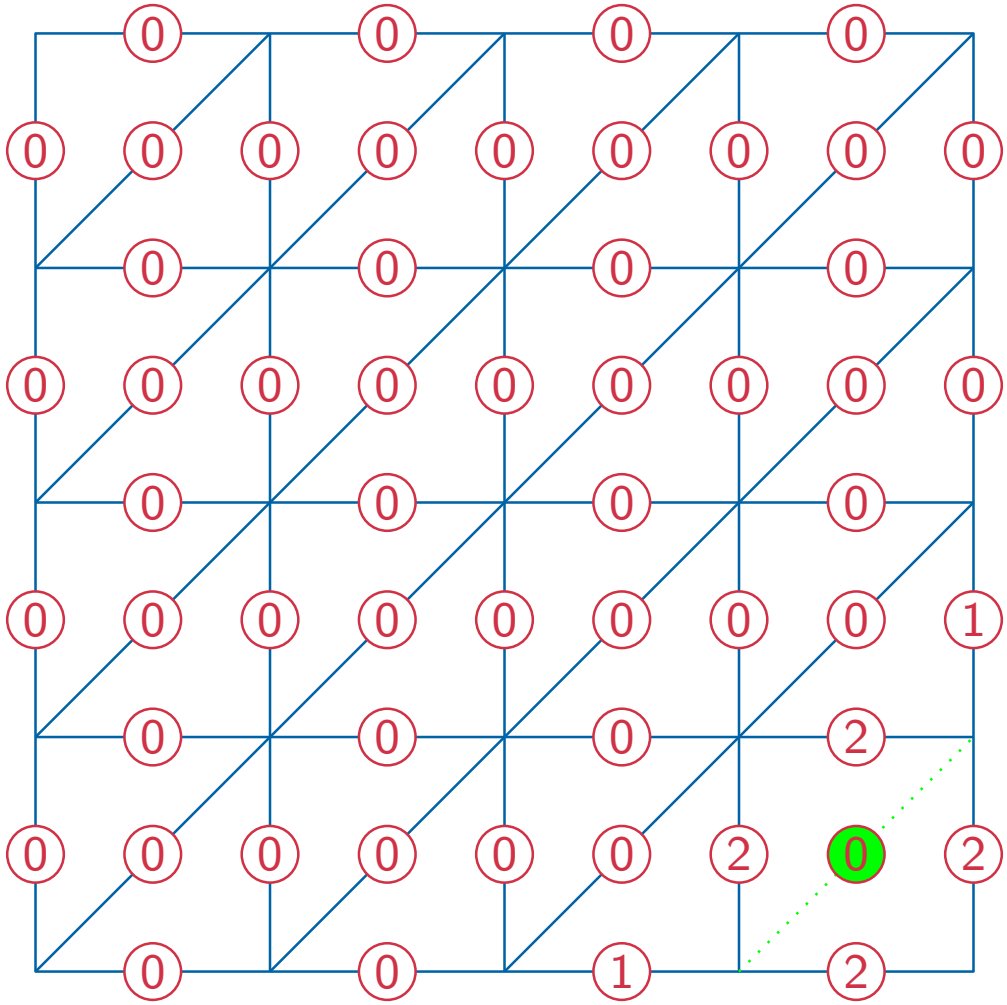
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



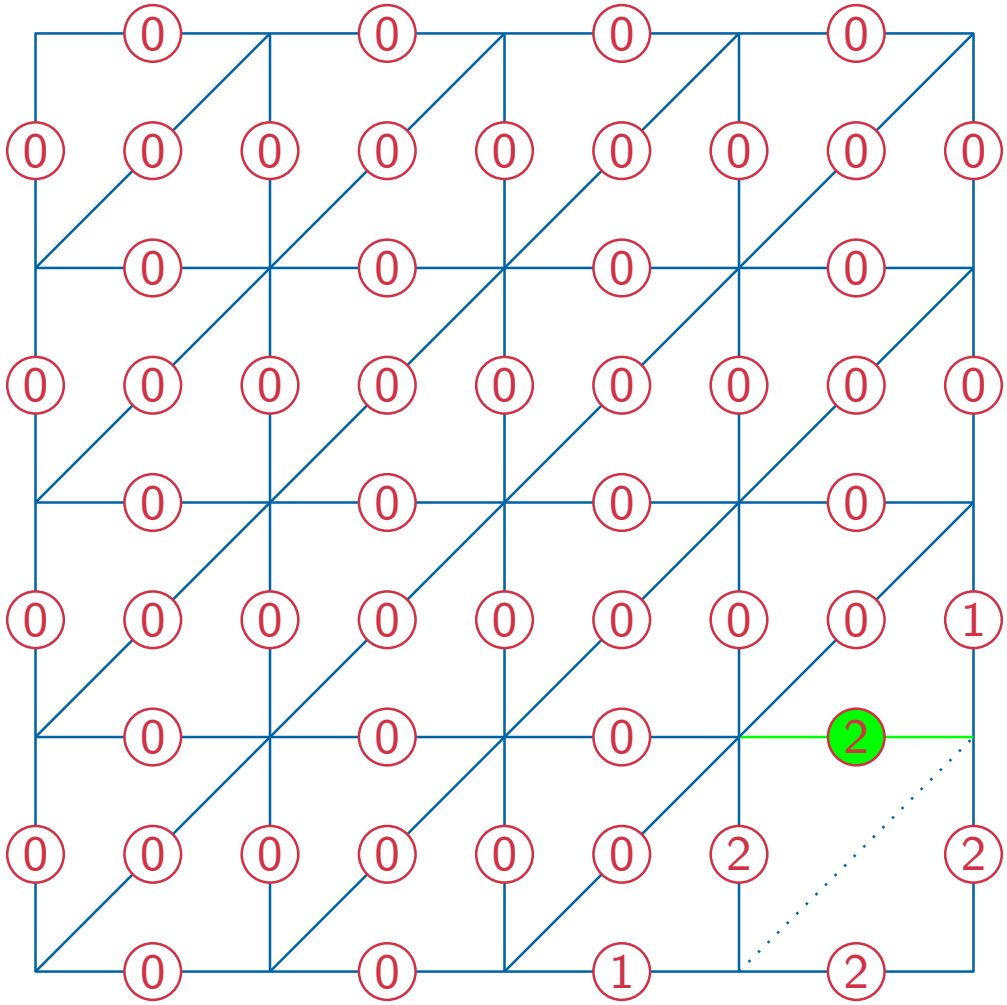
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



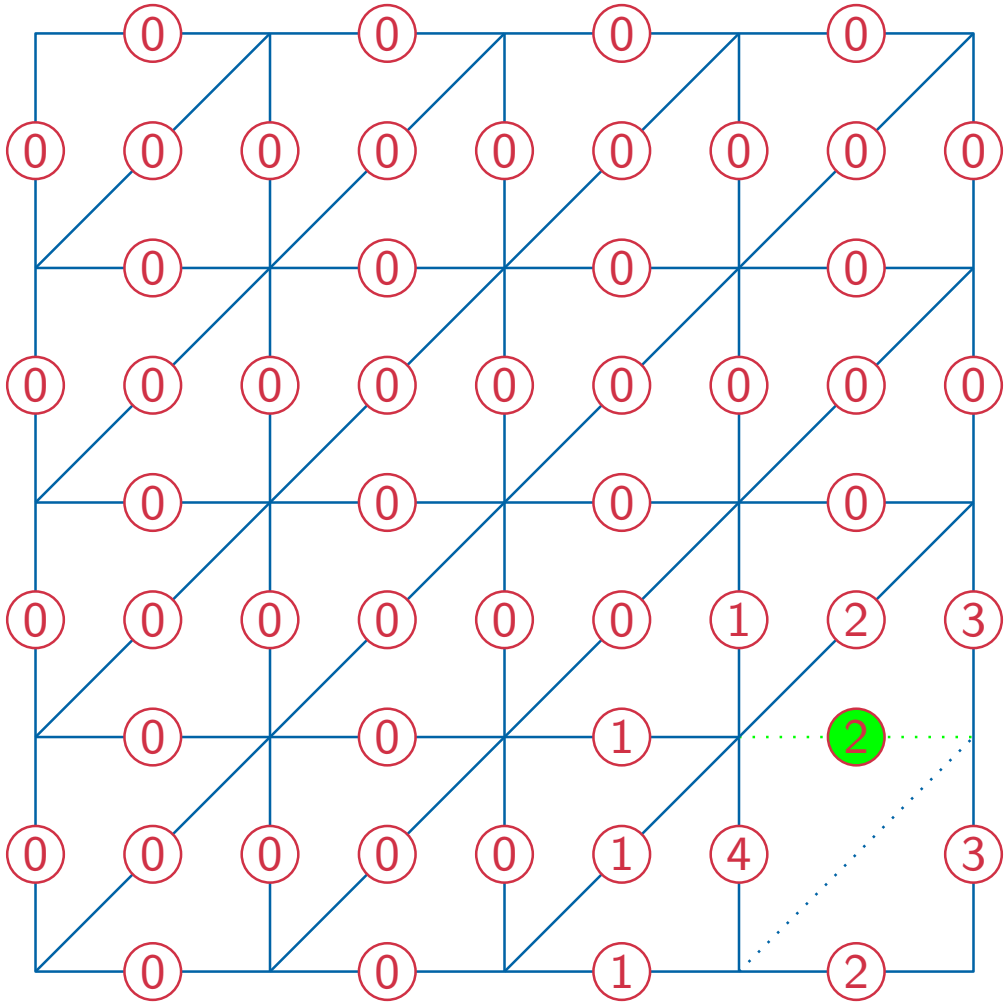
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



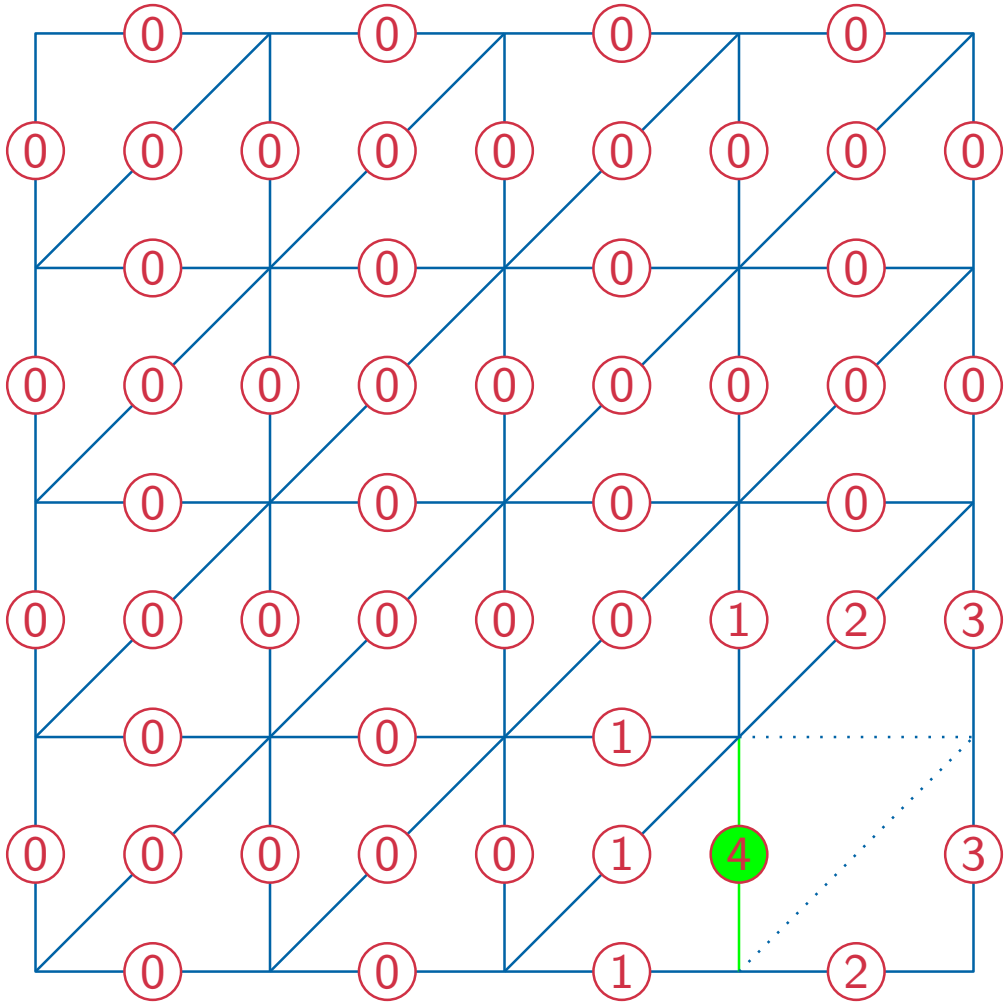
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



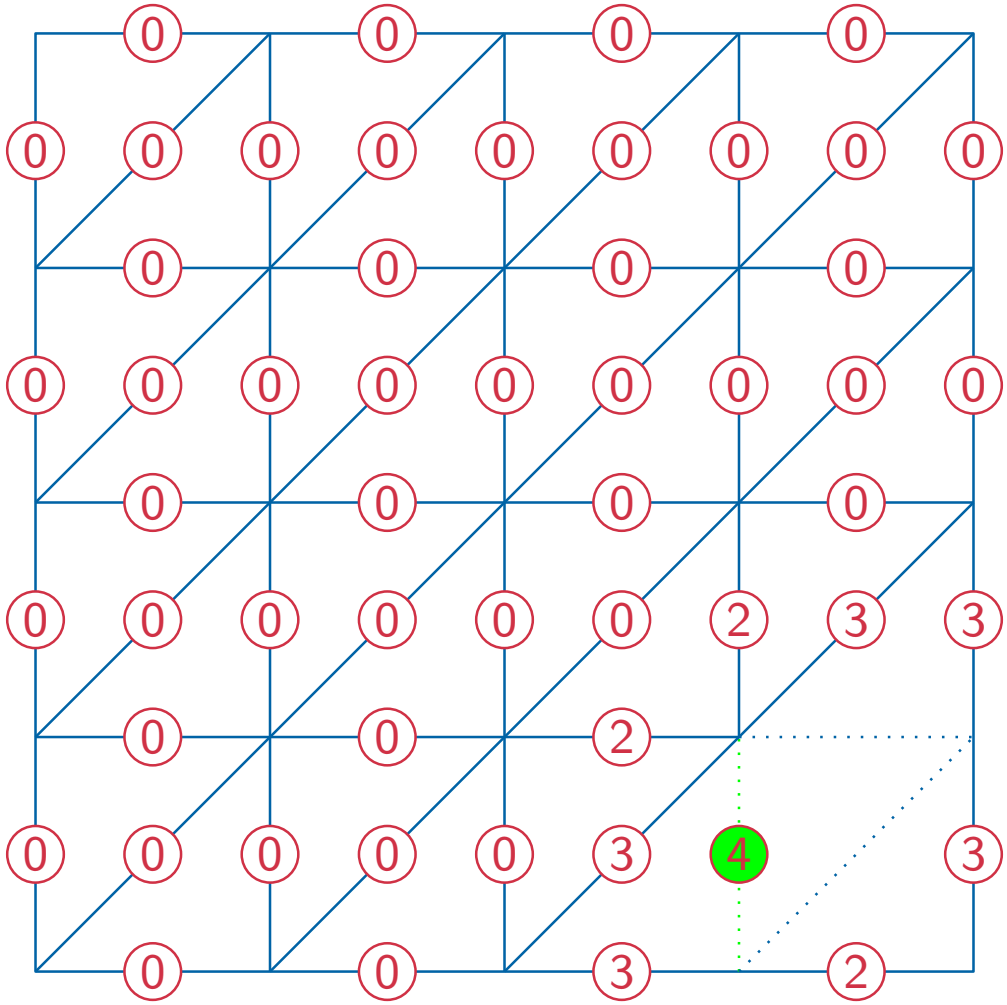
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



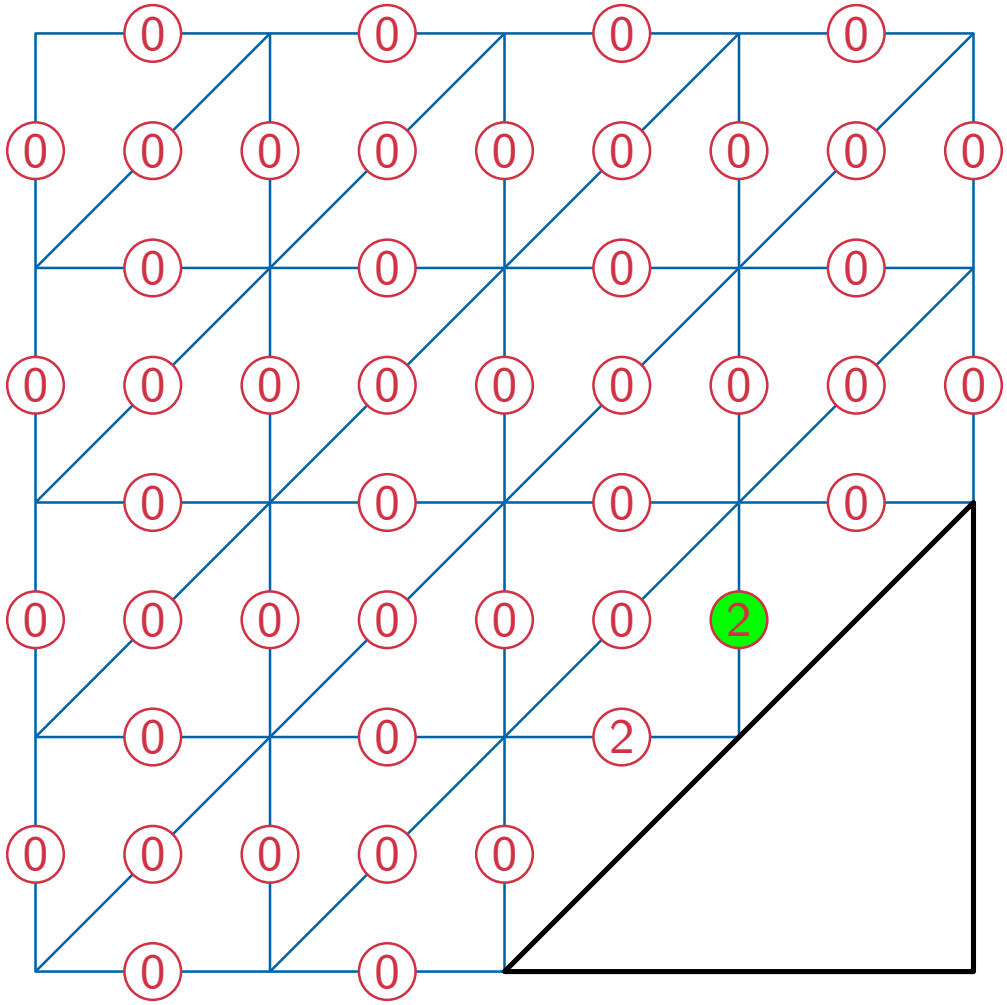
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



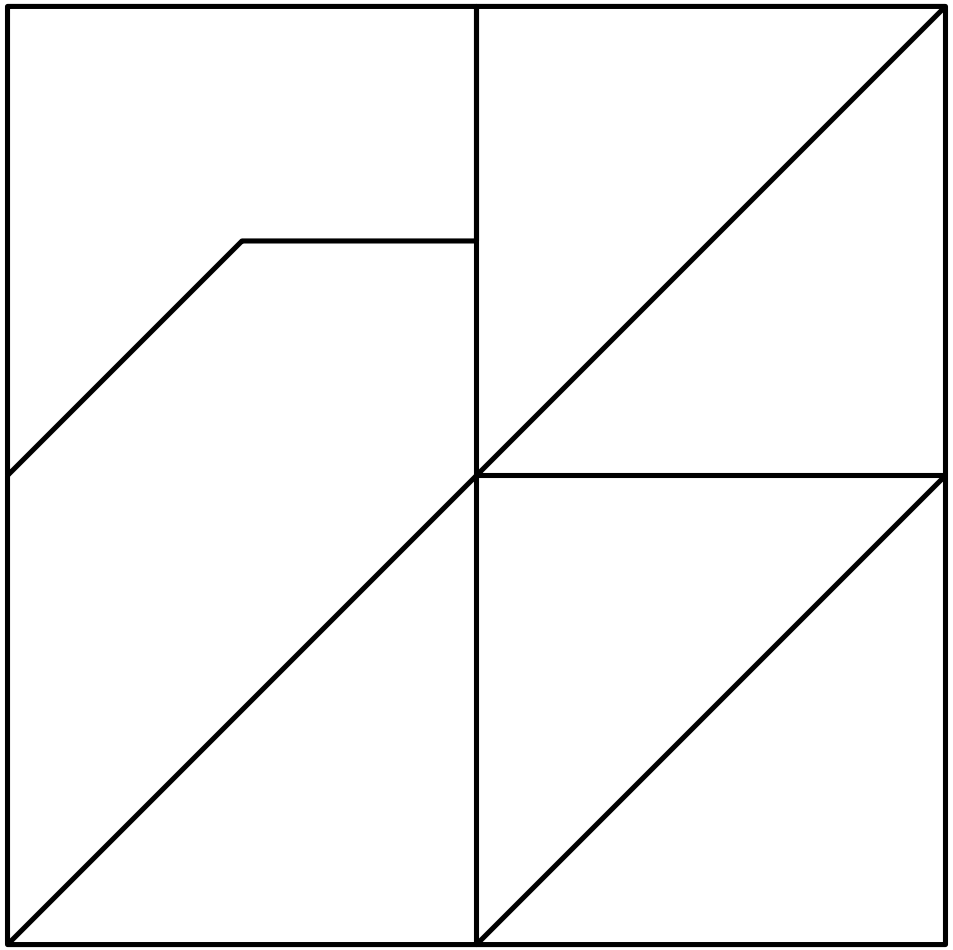
Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].



Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].

It is claimed that this method tends to produce nice agglomerated elements, from a geometric perspective, and for structured meshes produces something close to a standard coarsening.

Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].

It is claimed that this method tends to produce nice agglomerated elements, from a geometric perspective, and for structured meshes produces something close to a standard coarsening.

However, how "nice" the elements are can depend largely on which face is selected in the case that multiple faces have the same weight.

Furthermore, this method can result in elements not being included in an agglomerate, agglomerates which are completely contained within another agglomerate, and potentially agglomerated elements which are not continuous or simple polygons.

Weighted-face agglomeration

Another technique assigns weights to each face, and agglomerates based on the face with the highest weight; cf., [Jones & Vassilevski, 2001].

It is claimed that this method tends to produce nice agglomerated elements, from a geometric perspective, and for structured meshes produces something close to a standard coarsening.

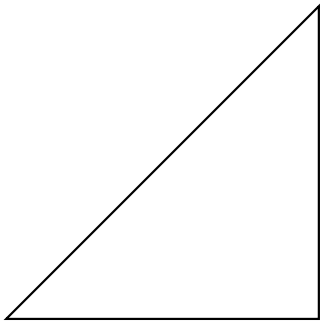
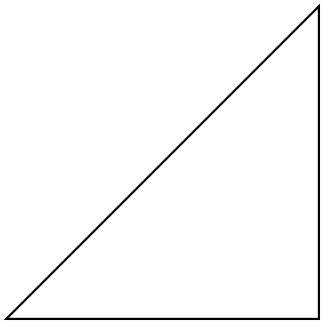
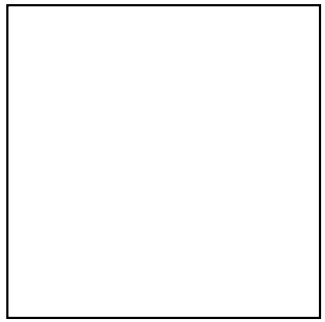
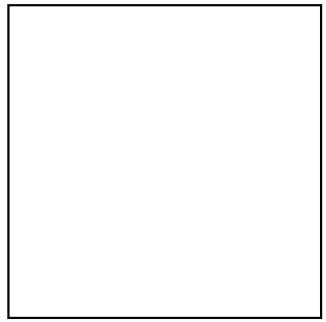
However, how "nice" the elements are can depend largely on which face is selected in the case that multiple faces have the same weight.

Furthermore, this method can result in elements not being included in an agglomerate, agglomerates which are completely contained within another agglomerate, and potentially agglomerated elements which are not continuous or simple polygons.

The number of agglomerates to generate can not be specified, and in order to produce "coarser" meshes it is necessary to apply the algorithm recursively.

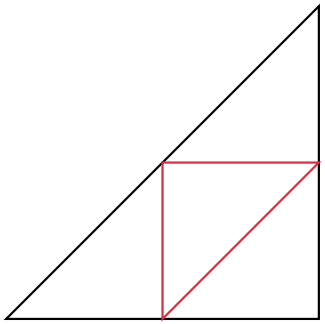
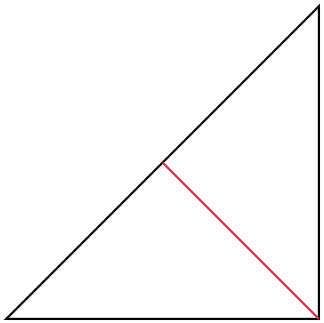
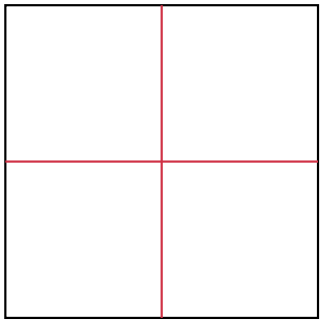
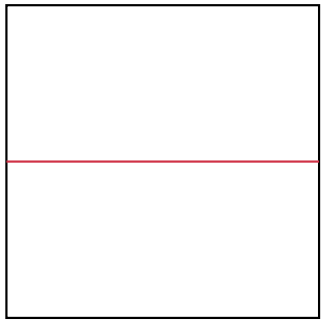
Refinement for agglomerated elements

With the standard elements it is possible to refine, both isotropically and anisotropically, the elements fairly simply as the elements can all be divided into standard elements:

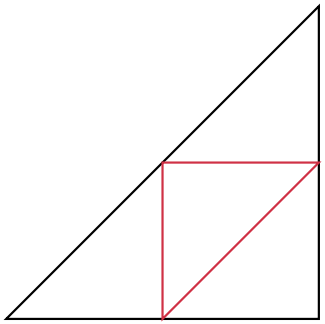
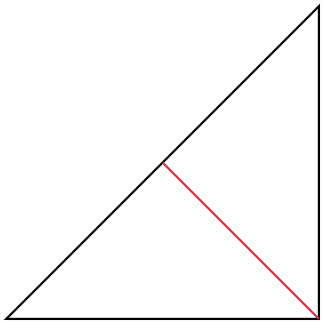
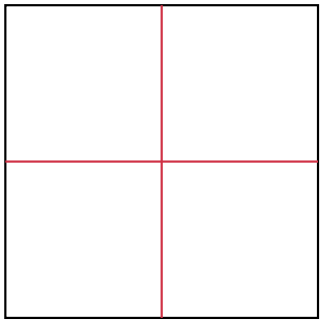
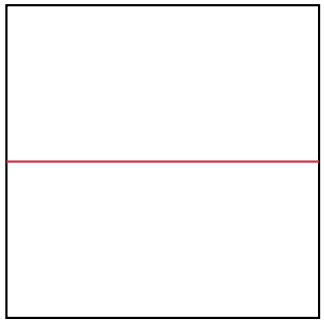
Refinement for agglomerated elements

With the standard elements it is possible to refine, both isotropically and anisotropically, the elements fairly simply as the elements can all be divided into standard elements:

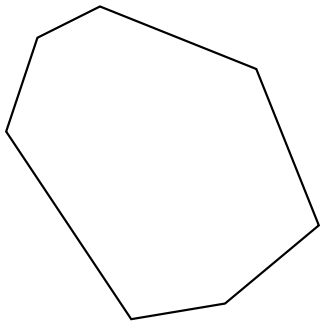





Refinement for agglomerated elements

With the standard elements it is possible to refine, both isotropically and anisotropically, the elements fairly simply as the elements can all be divided into standard elements:

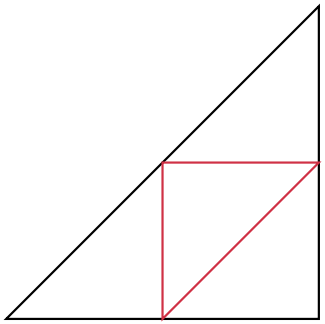
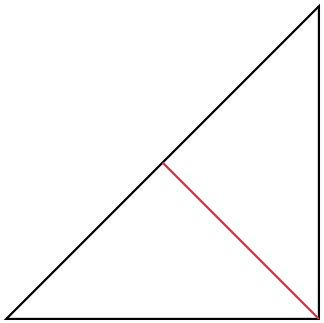
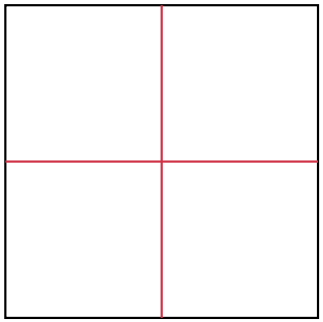
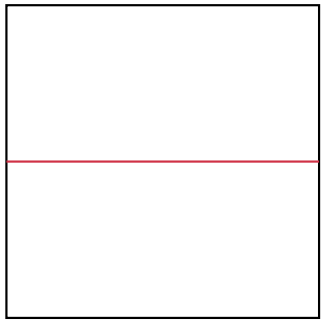





With general polygonal elements a standard refinement does not necessarily exist. One common technique is to join the midpoint of each edge with a point (e.g., barycentre of the element) inside the element. However, this relies on the element being star-shaped.

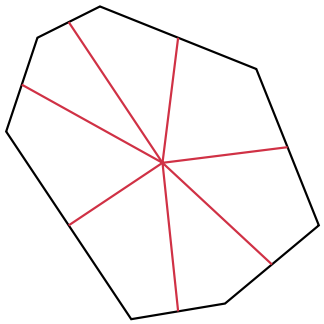


Refinement for agglomerated elements

With the standard elements it is possible to refine, both isotropically and anisotropically, the elements fairly simply as the elements can all be divided into standard elements:

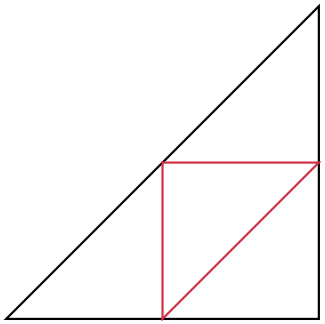
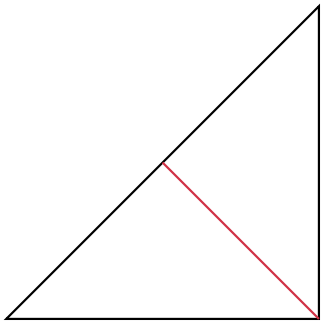
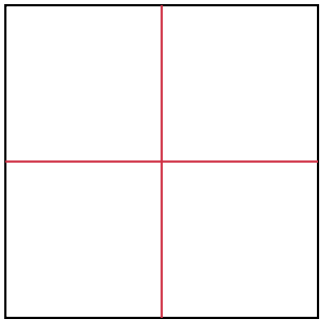
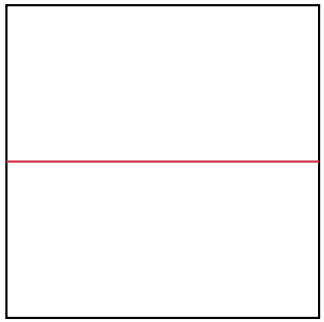





With general polygonal elements a standard refinement does not necessarily exist. One common technique is to join the midpoint of each edge with a point (e.g., barycentre of the element) inside the element. However, this relies on the element being star-shaped.

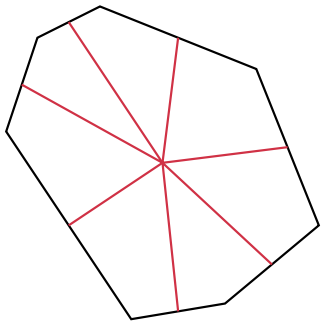


Refinement for agglomerated elements

With the standard elements it is possible to refine, both isotropically and anisotropically, the elements fairly simply as the elements can all be divided into standard elements:

With general polygonal elements a standard refinement does not necessarily exist. One common technique is to join the midpoint of each edge with a point (e.g., barycentre of the element) inside the element. However, this relies on the element being star-shaped.



With agglomerated elements, however, we have another alternative — each agglomerated element consists of a patch of elements from the original mesh, so we can refine the agglomerated element by agglomerating this patch into 2^d elements.

[Collis & Houston, 2016]

Methods on polytopic meshes

In this project we will focus on the discontinuous Galerkin finite element method and virtual element method.

Methods on polytopic meshes

In this project we will focus on the discontinuous Galerkin finite element method and virtual element method.

We present a brief overview of both methods for the following boundary-value problem on an open bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$: find u such that

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where $a(\boldsymbol{x})$ is a symmetric tensor which satisfies, for all $\boldsymbol{x} \in \Omega$,

$$c|\boldsymbol{\xi}|^2 \leq \boldsymbol{\xi} \cdot a(\boldsymbol{x})\boldsymbol{\xi} \leq C|\boldsymbol{\xi}|^2,$$

where c and C are positive constants. For simplicity, we assume a is piecewise constant with respect to the mesh.

Methods on polytopic meshes

In this project we will focus on the discontinuous Galerkin finite element method and virtual element method.

We present a brief overview of both methods for the following boundary-value problem on an open bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$: find u such that

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where $a(\boldsymbol{x})$ is a symmetric tensor which satisfies, for all $\boldsymbol{x} \in \Omega$,

$$c|\boldsymbol{\xi}|^2 \leq \boldsymbol{\xi} \cdot a(\boldsymbol{x})\boldsymbol{\xi} \leq C|\boldsymbol{\xi}|^2,$$

where c and C are positive constants. For simplicity, we assume a is piecewise constant with respect to the mesh.

In general, we can define both methods by a bilinear form and linear functional as: find $u_\star \in V_\star^p$ such that

$$A_\star(u_\star, v_\star) = F_\star(v_\star)$$

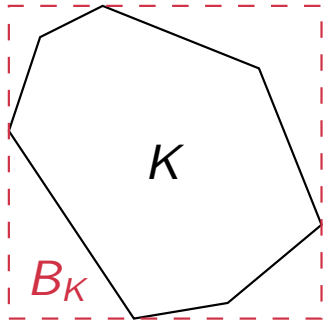
for all $v_\star \in V_\star^p$, where $\star$ is dg or ve depending on the method, on a polygonal mesh $\mathcal{T}_h$, with the sets of interior and boundary faces, $\mathcal{F}_h^I$ and $\mathcal{F}_h^B$, respectively.

Discontinuous Galerkin FEM

Define the DGFEM finite element space

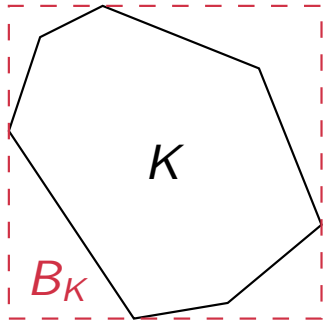
$$V_{dg}^{p}(\mathcal{T}_h) := \left\{ u \in L^2(\Omega) : u|_K \in \mathcal{P}_{p_K}(K), K \in \mathcal{T}_h \right\},$$

where $\mathcal{P}_p(K)$ denotes the space of polynomials of total degree p. Unlike for standard FEM we define the polynomials on the global space, rather than a reference element. For polytopals we simply define basis functions on the bounding box B_K of the element $K \in \mathcal{T}_h$.



Define the DGFEM finite element space

$$V_{dg}^{p}(\mathcal{T}_h) := \left\{ u \in L^2(\Omega) : u|_K \in \mathcal{P}_{p_K}(K), K \in \mathcal{T}_h \right\},$$



where $\mathcal{P}_p(K)$ denotes the space of polynomials of total degree p. Unlike for standard FEM we define the polynomials on the global space, rather than a reference element. For polytopals we simply define basis functions on the bounding box B_K of the element $K \in \mathcal{T}_h$.

We also define the bilinear form and linear functional

$$A_{dg}(u_{dg}, v_{dg}) = \sum_{K \in \mathcal{T}_h} \int_K a \nabla u_{dg} \cdot \nabla v_{dg} \, \mathrm{d}\boldsymbol{x} - \sum_{F \in \mathcal{F}_h^I \cup \mathcal{F}_h^B} \int_F \{\!\{ a \nabla u_{dg} \}\!\} \cdot [\![v_{dg}]\!] \, \mathrm{d}s$$
$$- \sum_{F \in \mathcal{F}_h^I \cup \mathcal{F}_h^B} \int_F \{\!\{ a \nabla v_{dg} \}\!\} \cdot [\![u_{dg}]\!] \, \mathrm{d}s + \sum_{F \in \mathcal{F}_h^I \cup \mathcal{F}_h^B} \int_F \sigma [\![u_{dg}]\!] \cdot [\![v_{dg}]\!] \, \mathrm{d}s,$$

$$F_{dg}(v_{dg}) = \sum_{K \in \mathcal{T}_h} \int_K f v_{dg} \, \mathrm{d}\boldsymbol{x},$$

where, on a face $F = \partial K^+ \cap \partial K^-$, $[\![v]\!] = v|_{K^+} \boldsymbol{n}_{K^+} + v|_{K^-} \boldsymbol{n}_{K^-}$ and $\{\!\{ \boldsymbol{\tau} \}\!\} = \frac{1}{2} (\boldsymbol{\tau}|_{K^+} + \boldsymbol{\tau}|_{K^-})$. σ is a penalisation function to be defined.

Discontinuous Galerkin FEM

To perform the error analysis we need several assumptions and definitions.

To perform the error analysis we need several assumptions and definitions.

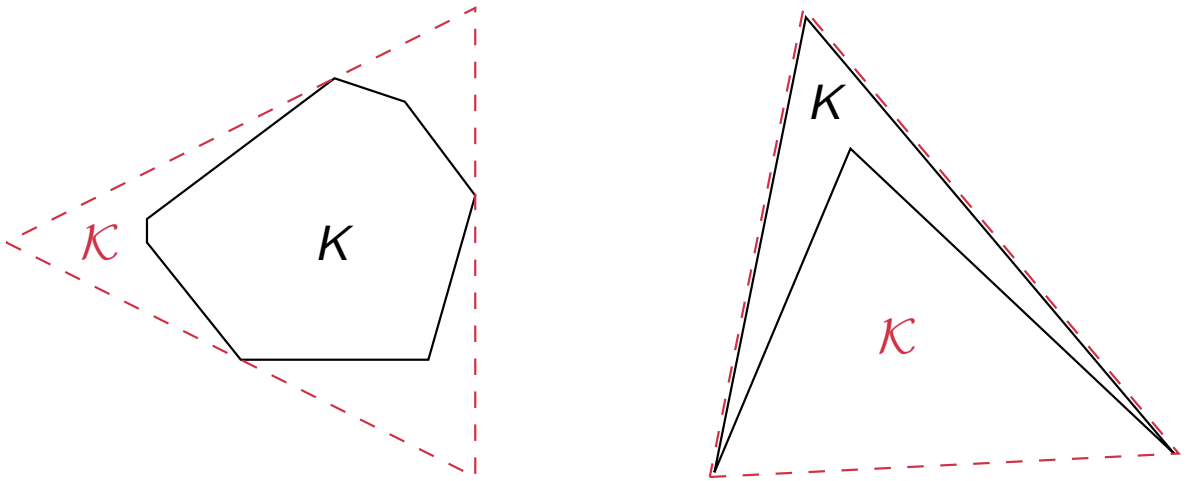
Definition (Covering)

A covering $\mathcal{T}_h^\sharp = \{\mathcal{K}\}$ for a mesh $\mathcal{T}_h$ is the set of open shape-regular d-simplices $\mathcal{K}$ such that, for each $K \in \mathcal{T}_h$, there exists a $\mathcal{K} \in \mathcal{T}_h^\sharp$ such that $K \subset \mathcal{K}$.

To perform the error analysis we need several assumptions and definitions.

Definition (Covering)

A covering $\mathcal{T}_h^\sharp = \{\mathcal{K}\}$ for a mesh $\mathcal{T}_h$ is the set of open shape-regular d-simplices $\mathcal{K}$ such that, for each $K \in \mathcal{T}_h$, there exists a $\mathcal{K} \in \mathcal{T}_h^\sharp$ such that $K \subset \mathcal{K}$.



To perform the error analysis we need several assumptions and definitions.

Definition (Covering)

A covering $\mathcal{T}_h^\sharp = \{\mathcal{K}\}$ for a mesh $\mathcal{T}_h$ is the set of open shape-regular d-simplices $\mathcal{K}$ such that, for each $K \in \mathcal{T}_h$, there exists a $\mathcal{K} \in \mathcal{T}_h^\sharp$ such that $K \subset \mathcal{K}$.

Assumption

Assume there exists a mesh covering $\mathcal{T}_h^\sharp$ of $K \in \mathcal{T}_h$ and a positive constant $\mathcal{O}_h$, independent of the mesh parameters, such that

$$\max_{K \in \mathcal{T}_h} \text{card} \left\{ K' \in \mathcal{T}_h : K' \cap \mathcal{K} \neq \emptyset, \mathcal{K} \in \mathcal{T}_h^\sharp \text{ such that } K \subset \mathcal{K} \right\} \leq \mathcal{O}_h,$$

and

$$h_{\mathcal{K}} := \text{diam}(\mathcal{K}) \leq C_{\text{diam}} h_K$$

for each pair $K \in \mathcal{T}_h$ and $\mathcal{K} \in \mathcal{T}_h^\sharp$, with $K \subset \mathcal{K}$, for a constant C_{diam}, uniformly with respect to the mesh size.

[Cangiani, Dong, Georgoulis, & Houston, 2017]

Discontinuous Galerkin FEM

We need one of the following assumptions ([Cangiani, Dong, Georgoulis, & Houston, 2017]).

Discontinuous Galerkin FEM

We need one of the following assumptions ([Cangiani, Dong, Georgoulis, & Houston, 2017]).

Assumption (Limited number of faces)

For each element $K \in \mathcal{T}_h$ we define $C_K := \text{card}\left\{F \in \mathcal{F}_h^I \cup \mathcal{F}_h^B : F \subset \partial K\right\}$. We assume that there exists a positive constant C_F, independent of the mesh parameters, such that

$$\max_{K \in \mathcal{T}_h} C_K \leq C_F.$$

We need one of the following assumptions ([Cangiani, Dong, Georgoulis, & Houston, 2017]).

Assumption (Limited number of faces)

For each element $K \in \mathcal{T}_h$ we define $C_K := \text{card}\left\{F \in \mathcal{F}_h^I \cup \mathcal{F}_h^B : F \subset \partial K\right\}$. We assume that there exists a positive constant C_F, independent of the mesh parameters, such that

$$\max_{K \in \mathcal{T}_h} C_K \leq C_F.$$

Definition

For each element $K \in \mathcal{T}_h$ we define a family $\mathcal{F}_b^K$ of all possible d-simplices contained within K sharing at last one face with K. Moreover, K_b^F denotes a simplex belonging to $\mathcal{F}_b^K$ which shares the face $F \subset \partial K$ with K.

Assumption (Arbitrary number of faces)

There exists a positive constant C_S such that for each $K \in \mathcal{T}_h$ there exists a set of non-overlapping d-simplices $\{K_b^K\}_{F \subset \partial K} \subset \mathcal{F}_b^K$ contained in K, such that for all $F \subset \partial K$,

$$h_K \leq C_s d |K_b^F| |F|^{-1}.$$

The previous two assumptions require a different penalisation function:

$$\sigma(\boldsymbol{x}) = \begin{cases} C_\sigma \max\limits_{K \in \{K^+, K^-\}} \left(C_{INV}(p_K, K, F) \dfrac{\overline{a}_K p_K^2 |F|}{|K|} \right) & \text{if limited number of faces met,} \\ C_\sigma \max\limits_{K \in \{K^+, K^-\}} \left(C_{inv,1} \dfrac{\overline{a}_K p_K^2}{h_K} \right) & \text{if arbitrary number of faces met.} \end{cases}$$

where C_σ is a sufficiently large constant, and C_{INV}, $C_{inv,1}$ are from inverse estimates.

The previous two assumptions require a different penalisation function:

$$\sigma(\boldsymbol{x}) = \begin{cases} C_\sigma \max\limits_{K\in\{K^+,K^-\}} \left(C_{INV}(p_K, K, F) \dfrac{\overline{a}_K p_K^2 |F|}{|K|} \right) & \text{if limited number of faces met,} \\ C_\sigma \max\limits_{K\in\{K^+,K^-\}} \left(C_{inv,1} \dfrac{\overline{a}_K p_K^2}{h_K} \right) & \text{if arbitrary number of faces met.} \end{cases}$$

where C_σ is a sufficiently large constant, and C_{INV}, $C_{inv,1}$ are from inverse estimates.

Theorem (*a priori* error bound [Cangiani, Dong, Georgoulis, & Houston, 2017])

For $u|_K \in H^{\ell_K}(K)$, $s_K = \min(p_K + 1, \ell_K)$

$$\|u - u_{dg}\|_{dg}^2 \leq C \sum_{K\in\mathcal{T}_h} \frac{h_K^{2(s_k-1)}}{p_K^{2(\ell_K-1)}} (\overline{a}_K + \mathcal{G}_K) \|\mathfrak{E}u\|_{H^{\ell_K}(\mathcal{K})}^2.$$

where

$$\mathcal{G}_K = \begin{cases} \overline{a}_K^2 \frac{p_K}{h_K^d} \sum\limits_{F\subset\partial K} C_m \sigma |F| + \overline{a}_K^2 \dfrac{p_K^2}{|K|} \sum\limits_{F\subset\partial K} C_{INV} \sigma^{-1} |F| + \dfrac{h_K^{2-d}}{p_K} \sum\limits_{F\subset\partial K} C_m \sigma |F| & \textit{if LNF,} \\ \overline{a}_K^2 p_K h_K^{-1} \max\limits_{F\subset\partial K} \sigma|_F^{-1} + \overline{a}_K^2 p_K^2 h_K^{-1} \max\limits_{F\subset\partial K} \sigma|_F^{-1} + p_K^{-1} h_K \max\limits_{F\subset\partial K} \sigma|_F & \textit{if ANF,} \end{cases}$$

$C_{INV} = C_{INV}(p_K, K, F)$, *and* $C_m = C_m(p_K, K, F)$.

Virtual element method (VEM)

For simplicity we consider only $d = 2$. Split the boundary of an element $K \in \mathcal{T}_h$ into N straight edges, with vertex endpoints.

Virtual element method (VEM)

For simplicity we consider only $d = 2$. Split the boundary of an element $K \in \mathcal{T}_h$ into N straight edges, with vertex endpoints.

Define for each element $E \in \mathcal{T}_h$ the local virtual space

$$V_K^p = \{v \in H^1(K) \cap C^0(K) : -\Delta v \in \mathcal{P}_{p-2}(K), v|_e \in \mathcal{P}_p(e) \, \forall e \in \partial K\},$$

where $\mathcal{P}_{-1}(K) = \{0\}$.

Virtual element method (VEM)

For simplicity we consider only $d = 2$. Split the boundary of an element $K \in \mathcal{T}_h$ into N straight edges, with vertex endpoints.

Define for each element $E \in \mathcal{T}_h$ the local virtual space

$$V_K^p = \{v \in H^1(K) \cap C^0(K) : -\Delta v \in \mathcal{P}_{p-2}(K), v|_e \in \mathcal{P}_p(e) \, \forall e \in \partial K\},$$

where $\mathcal{P}_{-1}(K) = \{0\}$.

We can define the local degrees of freedom for this space as

- $\mathcal{V}_K^p$: values at the vertices of K,
- $\mathcal{E}_K^p$: $(k-1)$ distinct point values (e.g. at Gauss-Lobatto nodes or evenly spaced) on each edge $e \in \partial K$
- $\mathcal{I}_K^p$: the moments

$$\frac{1}{|K|} \int_K m(\boldsymbol{x}) v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \qquad \text{for all } m \in \mathcal{M}_{p-2}(K),$$

  where $\mathcal{M}_{p-2}(K)$ is the set of $(p^2-p)/2$ scaled monomials forming a basis of $\mathcal{P}_{p-2}(K)$.

For simplicity we consider only $d = 2$. Split the boundary of an element $K \in \mathcal{T}_h$ into N straight edges, with vertex endpoints.

Define for each element $E \in \mathcal{T}_h$ the local virtual space

$$V_K^p = \{v \in H^1(K) \cap C^0(K) : -\Delta v \in \mathcal{P}_{p-2}(K), v|_e \in \mathcal{P}_p(e) \, \forall e \in \partial K\},$$

where $\mathcal{P}_{-1}(K) = \{0\}$.

We can define the local degrees of freedom for this space as

- $\mathcal{V}_K^p$: values at the vertices of K,
- $\mathcal{E}_K^p$: $(k-1)$ distinct point values (e.g. at Gauss-Lobatto nodes or evenly spaced) on each edge $e \in \partial K$
- $\mathcal{I}_K^p$: the moments

$$\frac{1}{|K|} \int_K m(\boldsymbol{x}) v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \qquad \text{for all } m \in \mathcal{M}_{p-2}(K),$$

  where $\mathcal{M}_{p-2}(K)$ is the set of $(p^2-p)/2$ scaled monomials forming a basis of $\mathcal{P}_{p-2}(K)$.

Construct the global space $V_{ve}^p \subset H_0^1(\Omega)$ by glueing these local spaces together (by sharing edge and vertex degrees of freedom) such that $V_{ve}^p|_K = V_K^p$.

[Beirão da Veiga et al., 2013; Beirão da Veiga, Lovadina & Russo, 2017]

Virtual element method (VEM)

Defining the local bilinear form, for all $u, v \in H_0^1(\Omega)$

$$A^K(u, v) = \int_K a \nabla u \cdot \nabla v \, \mathrm{d}\boldsymbol{x},$$

we can define a projection operator $\Pi_K^p : V_K^p \to \mathcal{P}_k(K)$, for all $v_{ve} \in V_K^p$, as

$$\begin{aligned} A^K(v_{ve} - \Pi_K^p v_{ve}, w) &= 0 \qquad && \forall w \in \mathcal{P}_p(K) \\ \mathcal{R}(v_{ve} - \Pi_K^p v_{ve}) &= 0, \end{aligned}$$

where $\mathcal{R}$ is a projection operator onto $\mathcal{P}_0(K)$; e.g., $\mathcal{R} v_{ve} = |\partial K|^{-1} \int_{\partial K} v_{ve}$.

Defining the local bilinear form, for all $u, v \in H^1_0(\Omega)$

$$A^K(u, v) = \int_K a \nabla u \cdot \nabla v \, \mathrm{d}\boldsymbol{x},$$

we can define a projection operator $\Pi^p_K : V^p_K \to \mathcal{P}_k(K)$, for all $v_{ve} \in V^p_K$, as

$$\begin{aligned} A^K(v_{ve} - \Pi^p_K v_{ve}, w) &= 0 \qquad && \forall w \in \mathcal{P}_p(K) \\ \mathcal{R}(v_{ve} - \Pi^p_K v_{ve}) &= 0, \end{aligned}$$

where $\mathcal{R}$ is a projection operator onto $\mathcal{P}_0(K)$; e.g., $\mathcal{R} v_{ve} = |\partial K|^{-1} \int_{\partial K} v_{ve}$.
Define a local discrete bilinear form for all $u_{ve}, v_{ve} \in V^p_K$)

$$A^K_{ve}(u_{ve}, v_{ve}) = A^K(\Pi^p_K u_{ve}, \Pi^p_K v_{ve}) + S_K(u_{ve} - \Pi^p_K u_{ve}, v_{ve} - \Pi^p_K v_{ve}),$$

where $S_K(\cdot, \cdot)$, called the stabilisation, is any symmetric positive definite bilinear form such that

$$c_0 A^K(v, v) \leq S_K(v, v) \leq c_1 A^K(v, v) \qquad \forall v \in V^p_K \text{ with } \Pi^p_K v = 0,$$

for some positive constants c_0 and c_1.

[Beirão da Veiga et al., 2013; Beirão da Veiga, Lovadina & Russo, 2017]

Virtual element method (VEM)

Split the stabilisation into interior and edge terms

$$S_K(u, v) = S_K^I(u, v) + S_K^E(u, v),$$

where the interior term is

$$S_K^I(u, v) = \sum_{\chi \in \mathcal{I}_K^p} \chi(u)\chi(v),$$

and the edge terms is one of the following:

$$S_K^E(u, v) = \sum_{\chi \in \mathcal{V}_K^p} \chi(u)\chi(v) + \sum_{\chi \in \mathcal{E}_K^p} \chi(u)\chi(v), \tag{1}$$

$$S_K^E(u, v) = h_K \int_{\partial K} \partial_s u \partial_s v \, \mathrm{d}s. \tag{2}$$

Split the stabilisation into interior and edge terms

$$S_K(u,v) = S_K^I(u,v) + S_K^E(u,v),$$

where the interior term is

$$S_K^I(u,v) = \sum_{\chi \in \mathcal{I}_K^p} \chi(u)\chi(v),$$

and the edge terms is one of the following:

$$S_K^E(u,v) = \sum_{\chi \in \mathcal{V}_K^p} \chi(u)\chi(v) + \sum_{\chi \in \mathcal{E}_K^p} \chi(u)\chi(v), \tag{1}$$

$$S_K^E(u,v) = h_K \int_{\partial K} \partial_s u \partial_s v \, \mathrm{d}s. \tag{2}$$

Finally, construct the global bilinear form and linear functional

$$A_{ve}(u_{ve}, v_{ve}) = \sum_{K \in \mathcal{T}_h} A_{ve}^K(u_{ve}, v_{ve}),$$

$$F_{ve}(v_{ve}) = \sum_{K \in \mathcal{T}_h} \int_K f_{ve} v_{ve} \, \mathrm{d}\boldsymbol{x},$$

where f_{ve} is the element-wise $L^2(K)$-projection of f onto $P_{p-2}(K)$.

[Beirão da Veiga et al., 2013; Beirão da Veiga, Lovadina & Russo, 2017]

Virtual element method (VEM)

We require the following assumptions on the mesh. [Beirão da Veiga, Lovadina & Russo, 2017].

Assumption

There exists a $\gamma > 0$ *such that all elements* $K \in \mathcal{T}_h$ *are star-shaped with respect to a ball* B_K *of radius* $\rho_K \geq \gamma h_K$ *and centre* $\boldsymbol{x}_K$.

Virtual element method (VEM)

We require the following assumptions on the mesh. [Beirão da Veiga, Lovadina & Russo, 2017].

Assumption

There exists a $\gamma > 0$ such that all elements $K \in \mathcal{T}_h$ are star-shaped with respect to a ball B_K of radius $\rho_K \geq \gamma h_K$ and centre $\boldsymbol{x}_K$.

For the first stabilisation it is also required that one of the following assumptions is met:

Assumption

There exists a $C \in \mathbb{N}$ such that $N \leq C$ for all $K \in \mathcal{T}_h$; i.e., the number of edges of K is bounded.

Assumption

There exists a $\eta > 0$ such that or all $K \in \mathcal{T}_h$ and edges $e \in \partial K$ it holds that $h_e \geq \eta h_K$; i.e., edges are not too small with respect to the element.

We require the following assumptions on the mesh. [Beirão da Veiga, Lovadina & Russo, 2017].

Assumption

There exists a $\gamma > 0$ such that all elements $K \in \mathcal{T}_h$ are star-shaped with respect to a ball B_K of radius $\rho_K \geq \gamma h_K$ and centre $\boldsymbol{x}_K$.

For the first stabilisation it is also required that one of the following assumptions is met:

Assumption

There exists a $C \in \mathbb{N}$ such that $N \leq C$ for all $K \in \mathcal{T}_h$; i.e., the number of edges of K is bounded.

Assumption

There exists a $\eta > 0$ such that or all $K \in \mathcal{T}_h$ and edges $e \in \partial K$ it holds that $h_e \geq \eta h_K$; i.e., edges are not too small with respect to the element.

The first and third assumption implies the second assumption; however, the second assumption is a weaker condition than the third.

Theorem (*a priori* error bound [Beirão da Veiga, Lovadina & Russo, 2017])

As long as the first assumption, and either the second or third assumption is met, the VEM with the first stabilisation has the following error bound, for $1 < s \leq p + 1$,

$$\|u - u_{ve}\|_{H^1(\Omega)} \leq \mathcal{C}(h) h^{s-1} |u|_{H^s(\Omega)},$$

where

$$\mathcal{C}(h) = \begin{cases} C \max\limits_{K \in \mathcal{T}_h} \left(\log \left(1 + h_K h_{\min,K}^{-1} \right) \right) & \text{if the second assumption is met,} \\ C & \text{if the third assumption is met,} \end{cases}$$

$h_{\min,K}$ *is the length of the smallest edge of* K.

Theorem (*a priori* error bound [Beirão da Veiga, Lovadina & Russo, 2017])

As long as the first assumption, and either the second or third assumption is met, the VEM with the first stabilisation has the following error bound, for $1 < s \leq p+1$,

$$\|u - u_{ve}\|_{H^1(\Omega)} \leq \mathcal{C}(h) h^{s-1} |u|_{H^s(\Omega)},$$

where

$$\mathcal{C}(h) = \begin{cases} C \max\limits_{K \in \mathcal{T}_h} \left(\log \left(1 + h_K h_{\min,K}^{-1} \right) \right) & \text{if the second assumption is met,} \\ C & \text{if the third assumption is met,} \end{cases}$$

$h_{\min,K}$ *is the length of the smallest edge of* K.

Theorem (*a priori* error bound [Beirão da Veiga, Lovadina & Russo, 2017])

As long as the first assumption the VEM with the second stabilisation has the following error bound, for $3/2 < s \leq p+1$,

$$\|u - u_{ve}\|_{H^1(\Omega)} \leq C h^{s-1} |u|_{H^s(\Omega)}.$$

"Uniform" refinement

Solve, using the symmetric interior penalty DGFEM, Poisson's equation

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega = (0,1)^2, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

with forcing function f selected such that

$$u(x,y) = \sin(\pi x)\sin(\pi y),$$

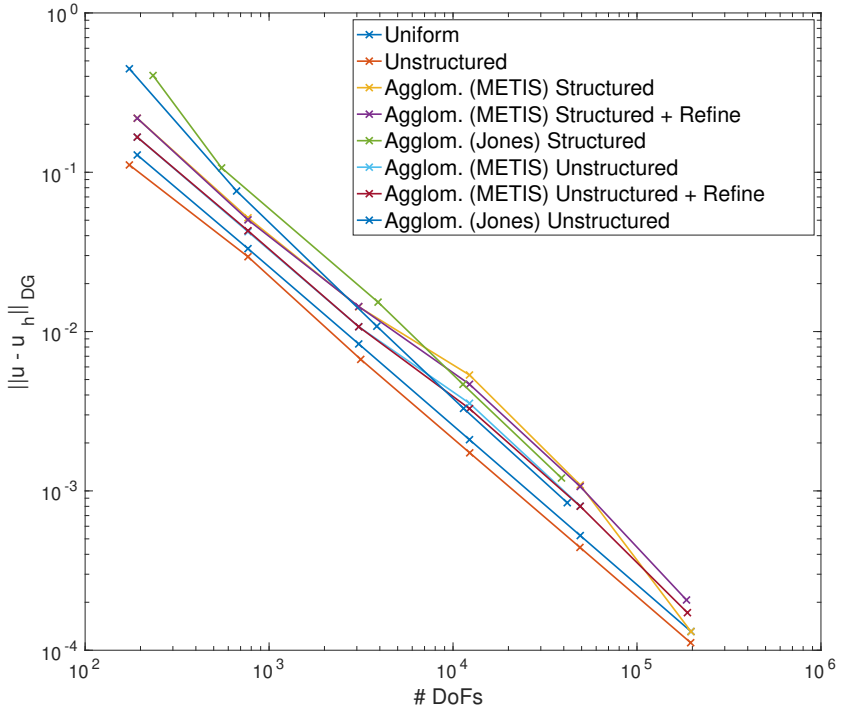
We consider, for $N = 4, 16, 32, 64, 128$,

1. sequence of uniform $N \times N$ triangular mesh
2. sequence of unstructured triangulations with roughly $2N^2$ elements,
3. sequence of meshes agglomerating (both methods) the 128×128 uniform mesh into roughly $2N^2$ elements,
4. sequence of meshes agglomerating (both methods) the finest unstructured mesh into roughly $2N^2$ elements,
5. sequence of meshes by refining agglomerated (METIS) mesh of 32 elements of the finest unstructured or 128×128 uniform mesh.

"Uniform" refinement

Solve, using the symmetric interior penalty DGFEM, Poisson's equation

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega = (0,1)^2, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

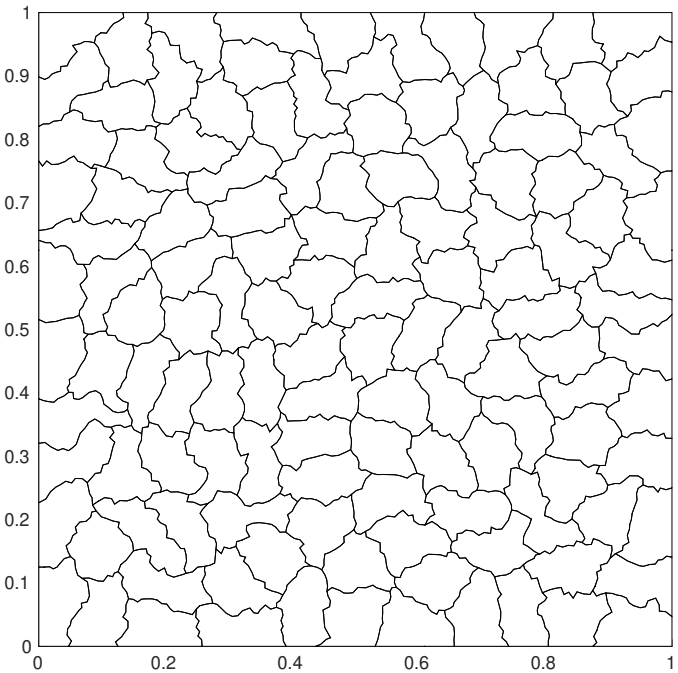


"Uniform" refinement

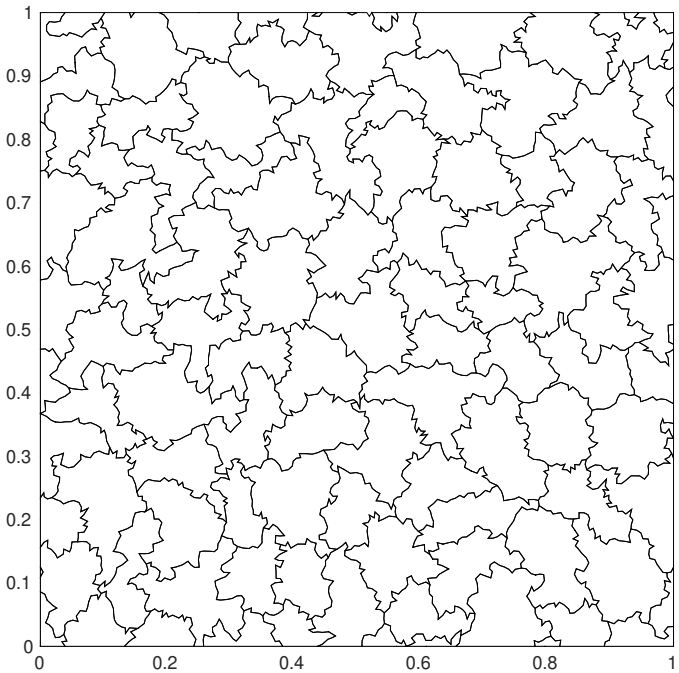
Solve, using the symmetric interior penalty DGFEM, Poisson's equation

$$-\Delta u = f \quad \text{in } \Omega = (0,1)^2,$$
$$u = 0 \quad \text{on } \partial\Omega,$$

METIS



Jones & Vassilevski

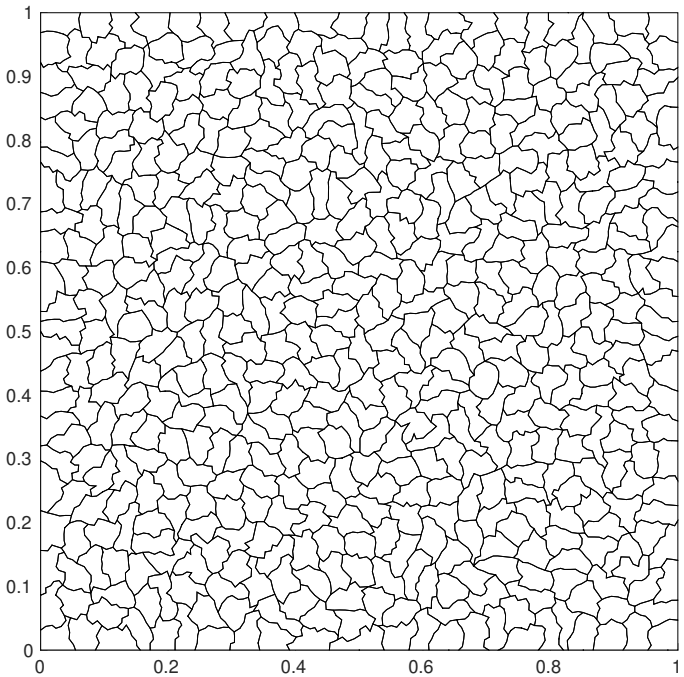


"Uniform" refinement

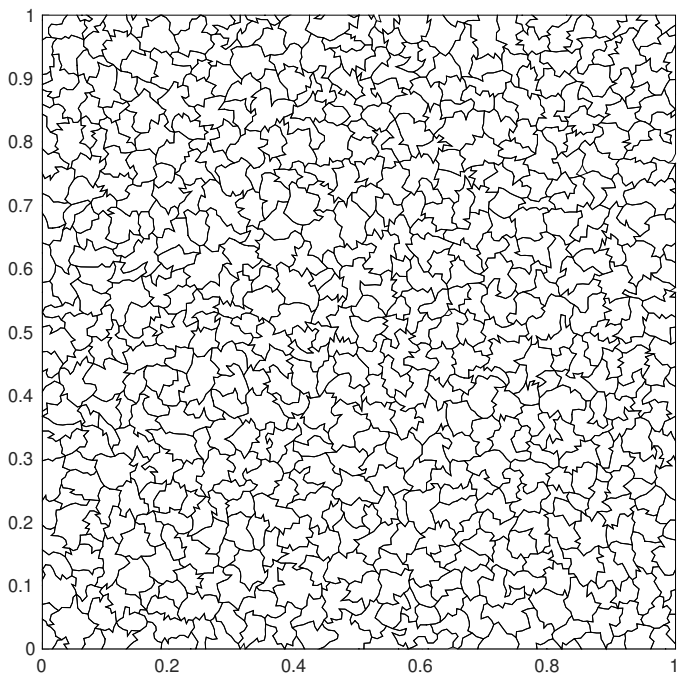
Solve, using the symmetric interior penalty DGFEM, Poisson's equation

$$-\Delta u = f \quad \text{in } \Omega = (0,1)^2,$$
$$u = 0 \quad \text{on } \partial\Omega,$$

METIS



Jones & Vassilevski

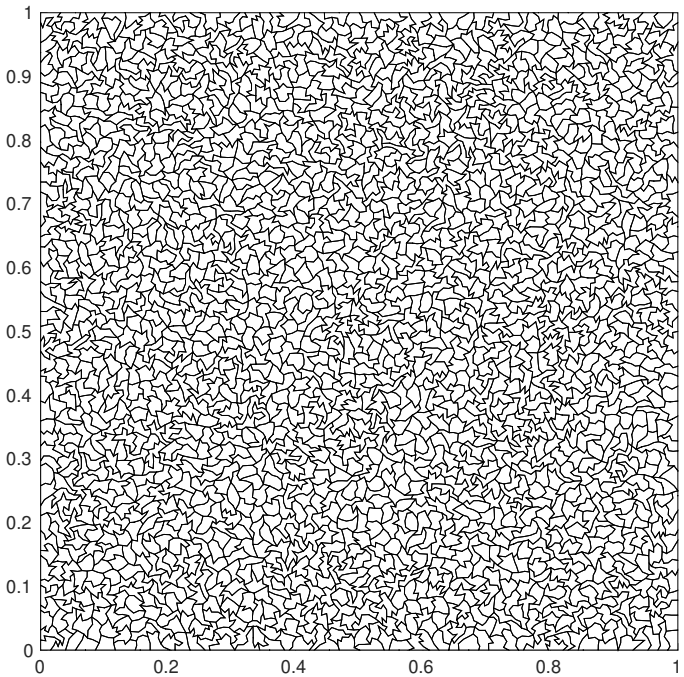


"Uniform" refinement

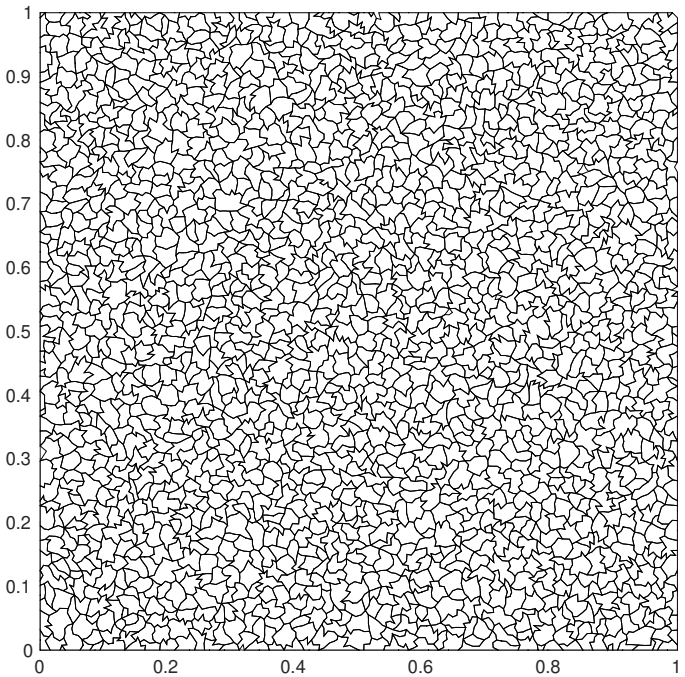
Solve, using the symmetric interior penalty DGFEM, Poisson's equation

$$-\Delta u = f \quad \text{in } \Omega = (0,1)^2,$$
$$u = 0 \quad \text{on } \partial\Omega,$$

METIS



Jones & Vassilevski

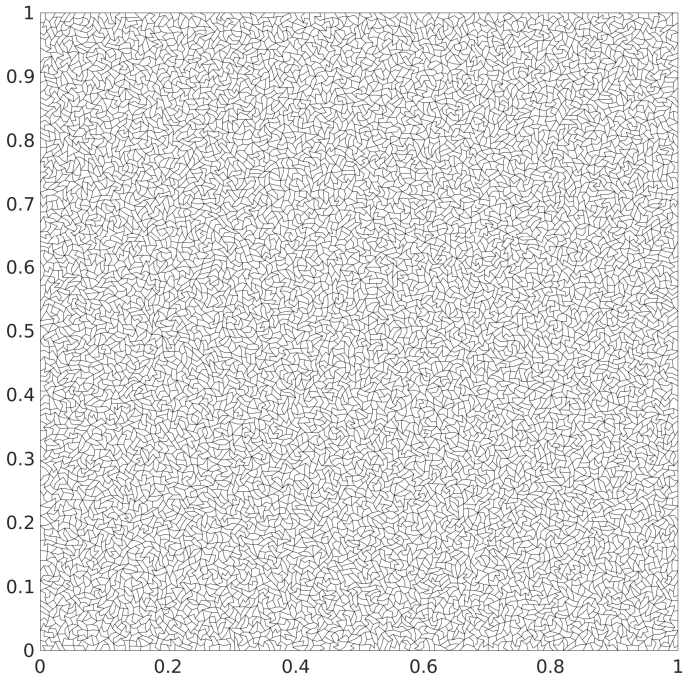


"Uniform" refinement

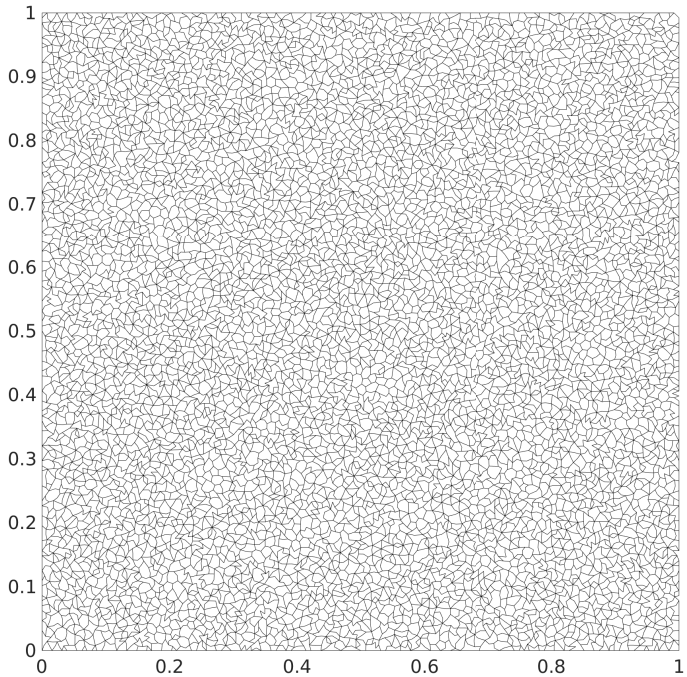
Solve, using the symmetric interior penalty DGFEM, Poisson's equation

$$-\Delta u = f \quad \text{in } \Omega = (0,1)^2,$$
$$u = 0 \quad \text{on } \partial\Omega,$$

METIS



Jones & Vassilevski



Two-grid adaptive refinement

To demonstrate some possibilities for refinement we consider a case where we need to refine two hierarchical meshes. The fine mesh $\mathcal{T}_h$ is a standard (unstructured) triangulation, and the coarse mesh is an agglomeration of the fine mesh. We will then consider the so-called two-grid method for solving the following nonlinear PDE:

Two-grid adaptive refinement

To demonstrate some possibilities for refinement we consider a case where we need to refine two hierarchical meshes. The fine mesh $\mathcal{T}_h$ is a standard (unstructured) triangulation, and the coarse mesh is an agglomeration of the fine mesh. We will then consider the so-called two-grid method for solving the following nonlinear PDE: Given $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ and $f \in L^2(\Omega)$, find u such that

$$\begin{aligned} -\nabla \cdot \{\mu(\boldsymbol{x}, |\nabla u|)\nabla u\} &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma. \end{aligned}$$

Assumption

1. $\mu \in C(\bar{\Omega} \times [0, \infty))$ *and*
2. *there exists positive constants* m_μ *and* M_μ *such that*

$$M_\mu(t - s) \leq \mu(\boldsymbol{x}, t)t - \mu(\boldsymbol{x}, s)s \leq M_\mu(t - s), \quad t \geq s \geq 0, \quad \boldsymbol{x} \in \bar{\Omega}.$$

Two-Grid Approximation

1. Construct coarse and fine FE spaces $V_{dg}^P(\mathcal{T}_H)$ and $V_{dg}^p(\mathcal{T}_h)$.
2. Compute the coarse grid approximation $u_H \in V_{dg}^P(\mathcal{T}_H)$ such that

$$A_H(u_H; u_H, v_H) = F_H(v_H)$$

   for all $v_H \in V_{dg}^P(\mathcal{T}_H)$.
3. Determine the fine grid approximation $u_{tg} \in V_{dg}^p(\mathcal{T}_h)$ such that

$$A_h(u_H; u_{tg}, v_h) = F_h(v_H)$$

   for all $v_h \in V_{dg}^p(\mathcal{T}_h)$.

[C., Houston, & Wihler 2013]

Here, $A_h(\cdot;\cdot,\cdot)$ and $A_H(\cdot;\cdot,\cdot)$ are nonlinear in the first term, linear in the last two terms, and are defined similarly on the fine and coarse meshes, respectively,

Given an *a posteriori* error bound which can be broken into local, elementwise on the fine mesh, error indicators η_K and ξ_K representing the fine mesh error, and the error caused by the linearisation, we can create an algorithm for adaptively refining both meshes.

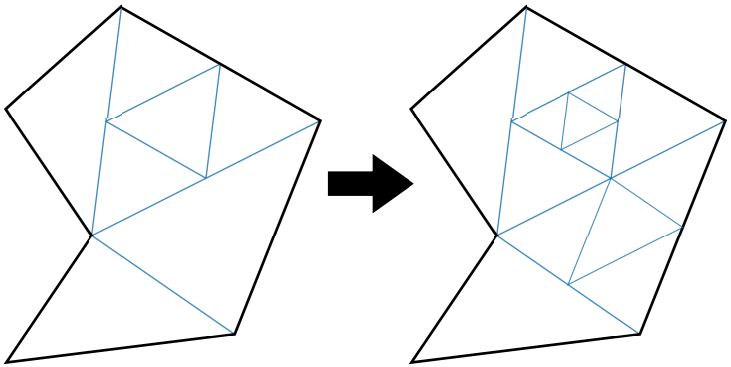
Two-Grid Adaptivity ( )

1. Construct initial coarse and fine FE spaces, with coarse mesh created by agglomerating the fine mesh.
2. Compute the coarse grid approximation and two-grid solution.
3. Select elements for refinement based on η_K and ξ_K:
   - 3.1 Use $\sqrt{\eta_K^2 + \xi_K^2}$ to determine set $\mathfrak{R}() \subseteq \langle$ of elements to refine.
   - 3.2 Choose fine or coarse mesh refinement. For all $K \in \mathfrak{R}(\langle)$
     - if $\lambda_F \xi_K \leq \eta_K$ refine the fine element K, and
     - if $\lambda_C \eta_K \leq \xi_K$ refine the coarse element $K_H \in \langle$, where $K \in \langle(K_H)$.
4. Perform *h*-/*hp*-mesh refinement of the fine space.
5. Select *h*- or *p*-refinement for each coarse element to refine.
6. Perform *h*-/*hp*-refinement of the coarse space.
7. Goto 2.

The constants λ_F and λ_C are steering parameters.

Two-grid adaptive refinement

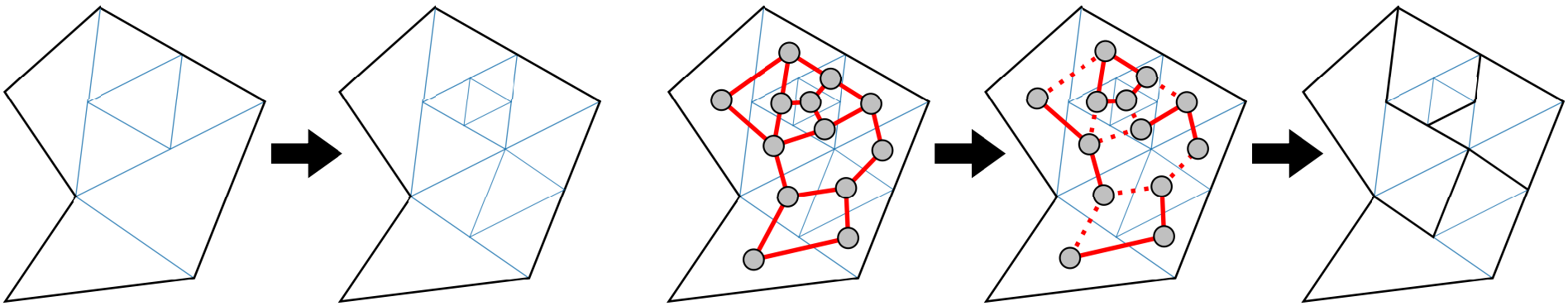
Fine Element Refine



Two-grid adaptive refinement

Fine Element Refine

Coarse Element Refine



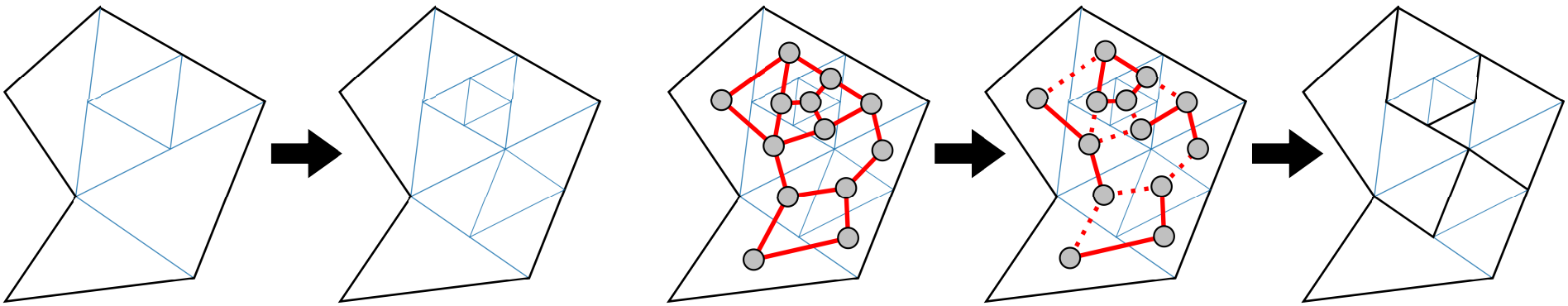
Refining coarse elements by agglomerating using METIS will attempt to create agglomerated elements with the same number of *child* fine elements.

However, we have information about the error for each fine element — can we distribute the agglomeration using this information?

Two-grid adaptive refinement

Fine Element Refine

Coarse Element Refine



Refining coarse elements by agglomerating using METIS will attempt to create agglomerated elements with the same number of *child* fine elements.

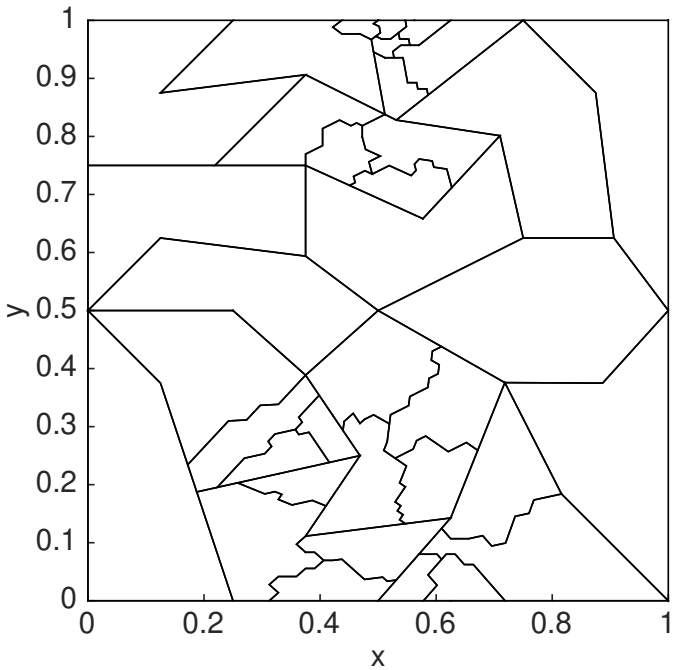
However, we have information about the error for each fine element — can we distribute the agglomeration using this information?

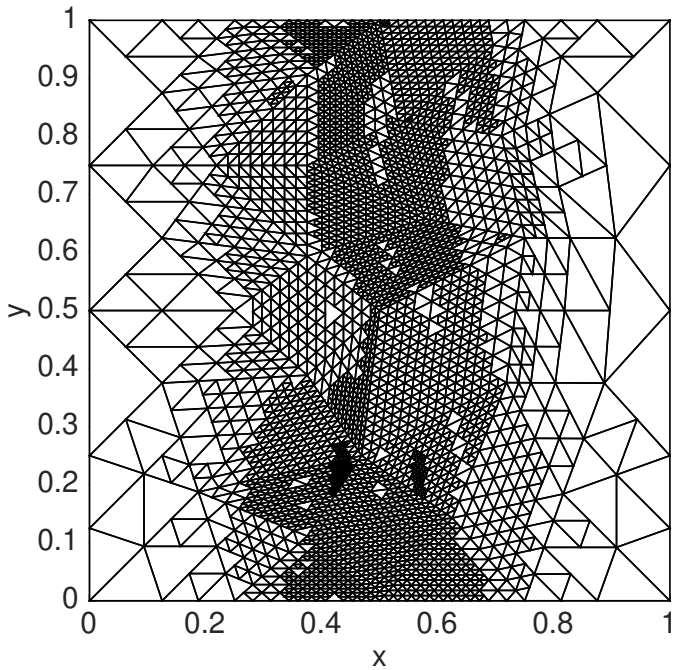
Possible to assign *weights* to each vertex and use a graph partitioning algorithm that balances these weights, rather than the number of elements. [Karypis & Kumar 1998]

We set the weight to the total local error indicator: $\eta_K^2 + \xi_K^2$

We let $\Omega = (0,1)^2, \mu(\boldsymbol{x}, |\nabla u|) = 2 + \frac{1}{1+|\nabla u|^2}$ and select f so that

$$u(x,y) = x(1-x)y(1-y)(1-2y)\mathrm{e}^{-20(2x-1)^2}.$$





8 h-refinement (Weighted Coarse Refinement)

Aim of project

Existing agglomeration techniques have several drawbacks:

- no real optimisation of the geometry of the elements
- no guarantee on the number of edges in the mesh,
- no guarantee that the elements will be star-shaped,
- no guarantee that about the size of the edges in respect to the size of the elements,
- possibilities in some methods for polygons that are not simply connected, or are completely contained inside other elements,
- for some methods the size of elements can have large variation when agglomerating from a fairly uniform initial mesh,
- the methods have no concept of the domain geometry, or any *a priori* knowledge about the domain/method

Aim of project

Existing agglomeration techniques have several drawbacks:

- no real optimisation of the geometry of the elements
- no guarantee on the number of edges in the mesh,
- no guarantee that the elements will be star-shaped,
- no guarantee that about the size of the edges in respect to the size of the elements,
- possibilities in some methods for polygons that are not simply connected, or are completely contained inside other elements,
- for some methods the size of elements can have large variation when agglomerating from a fairly uniform initial mesh,
- the methods have no concept of the domain geometry, or any *a priori* knowledge about the domain/method

The aim of this project is to develop an agglomeration technique which address at least some of these problems. Most notable, the idea will be develop an agglomeration technique which accounts for the assumptions in the numerical methods used, and attempts to optimise these assumptions (minimises the constants in the assumptions).

Aim of project

Existing agglomeration techniques have several drawbacks:

- no real optimisation of the geometry of the elements
- no guarantee on the number of edges in the mesh,
- no guarantee that the elements will be star-shaped,
- no guarantee that about the size of the edges in respect to the size of the elements,
- possibilities in some methods for polygons that are not simply connected, or are completely contained inside other elements,
- for some methods the size of elements can have large variation when agglomerating from a fairly uniform initial mesh,
- the methods have no concept of the domain geometry, or any *a priori* knowledge about the domain/method

The aim of this project is to develop an agglomeration technique which address at least some of these problems. Most notable, the idea will be develop an agglomeration technique which accounts for the assumptions in the numerical methods used, and attempts to optimise these assumptions (minimises the constants in the assumptions). The weighted graph partitioning in METIS allows for weighting refinement, and handling domain geometry details, if weights are selected correctly. However, the performance of the partitioning is greatly effected by the weights. Developing the agglomeration techniques to allow for some form of weighting to the agglomeration is also desirable.

Time-dependent problems

We can agglomerate neighbouring elements, or un-agglomerate elements, based on error estimates for time dependent problems, to make a tracking mesh.

[Cangiani et al., 2018; Cangiani, Georgoulis, & Sutton, 2021]

Time-dependent problems

We can agglomerate neighbouring elements, or un-agglomerate elements, based on error estimates for time dependent problems, to make a tracking mesh.

[Cangiani et al., 2018; Cangiani, Georgoulis, & Sutton, 2021]

▸ https://sites.google.com/view/oliversutton/home

Time-dependent problems

We can agglomerate neighbouring elements, or un-agglomerate elements, based on error estimates for time dependent problems, to make a tracking mesh.

[Cangiani et al., 2018; Cangiani, Georgoulis, & Sutton, 2021]

▸ https://sites.google.com/view/oliversutton/home

One of the aims of the project will be see if the agglomeration techniques developed can be extended to the time dependent cases without affecting the optimisations.