Mallas Poliedrales para Aproximar Soluciones de Ecuaciones Diferenciales en Derivadas Parciales

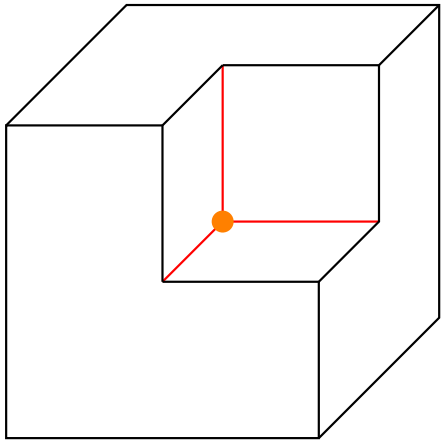
Alexis Jawtuschenko

Departmento de Matemática
Fac. de Cs. Exactas y Naturales
Universidad de Buenos Aires

PyCon Argentina 2018

Dominio No convexo

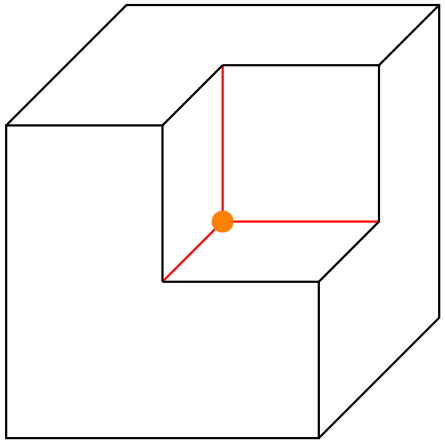
Dados conjuntos $\Omega \subseteq \mathbb{R}^3$ así



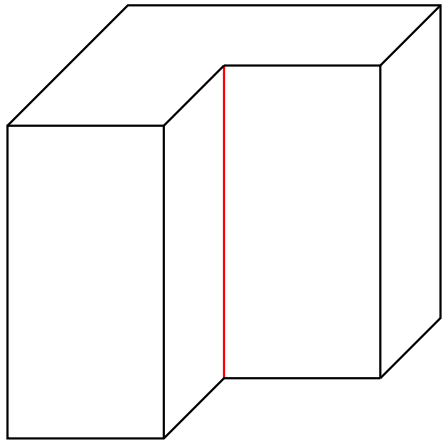
aristas y vértices

Dominio No convexo

Dados conjuntos $\Omega \subseteq \mathbb{R}^3$ así



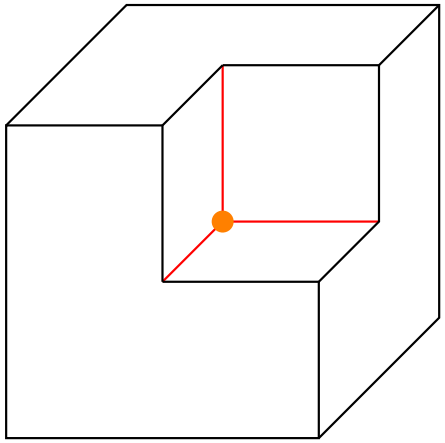
aristas y vértices



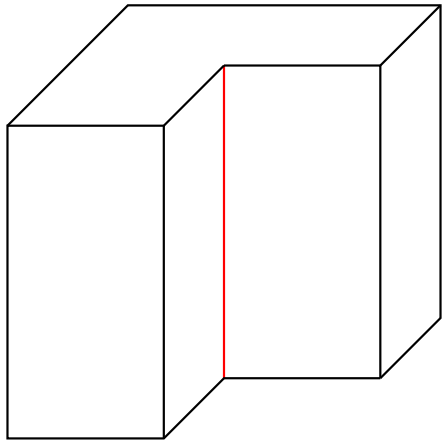
aristas

Dominio No convexo

Dados conjuntos $\Omega \subseteq \mathbb{R}^3$ así



aristas y vértices



aristas

y una función f con media cuadrática finita

Problema 0
Sistema de primer orden mixto de Poisson

- buscamos funciones $\boldsymbol{u}$ y p definidas en Ω que sean sol. de

Problema 0
Sistema de primer orden mixto de Poisson

- buscamos funciones $\boldsymbol{u}$ y p definidas en Ω que sean sol. de

$$\boldsymbol{u} = \nabla p$$

$$-\nabla \cdot \boldsymbol{u} = f$$

$$p|_{\partial\Omega} = 0.$$

Ejemplo de aplicación:

- un objeto con densidad de masa ρ

Ejemplo de aplicación:

- un objeto con densidad de masa ρ
- campo gravitatorio $\boldsymbol{g}$ debido a la atracción del objeto anterior

Ejemplo de aplicación:

- un objeto con densidad de masa ρ
- campo gravitatorio $\boldsymbol{g}$ debido a la atracción del objeto anterior
- aplicamos la Ley de Gauss de la gravedad en forma diferencial

Ejemplo de aplicación:

- un objeto con densidad de masa ρ
- campo gravitatorio $\boldsymbol{g}$ debido a la atracción del objeto anterior
- aplicamos la Ley de Gauss de la gravedad en forma diferencial
  - $-\nabla \cdot \boldsymbol{g} = 4\pi G\rho$

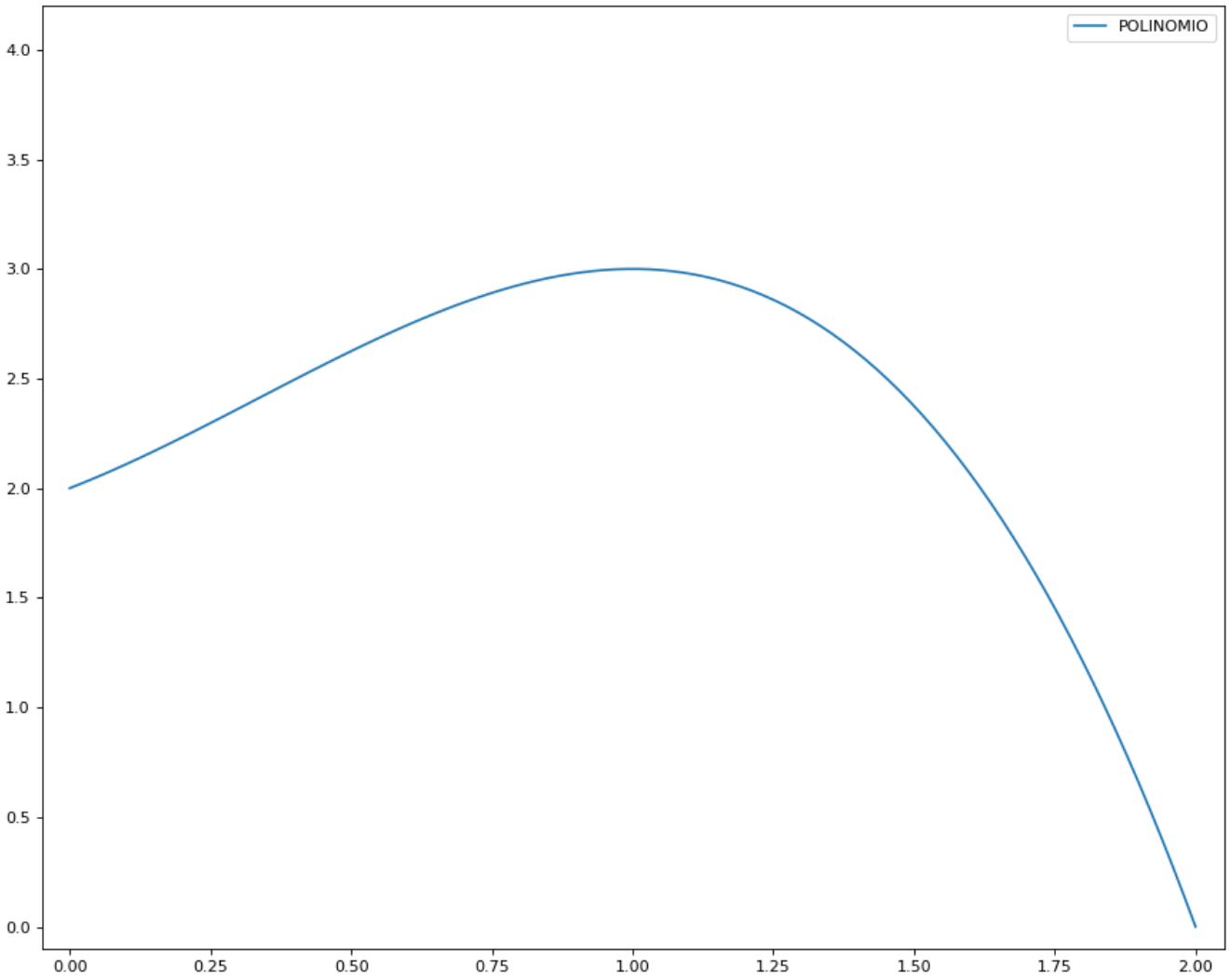
Ejemplo de aplicación:

- un objeto con densidad de masa ρ
- campo gravitatorio $\boldsymbol{g}$ debido a la atracción del objeto anterior
- aplicamos la Ley de Gauss de la gravedad en forma diferencial
  - $-\nabla \cdot \boldsymbol{g} = 4\pi G\rho$
- El campo gravitatorio es conservativo e irrotacional, con lo cual

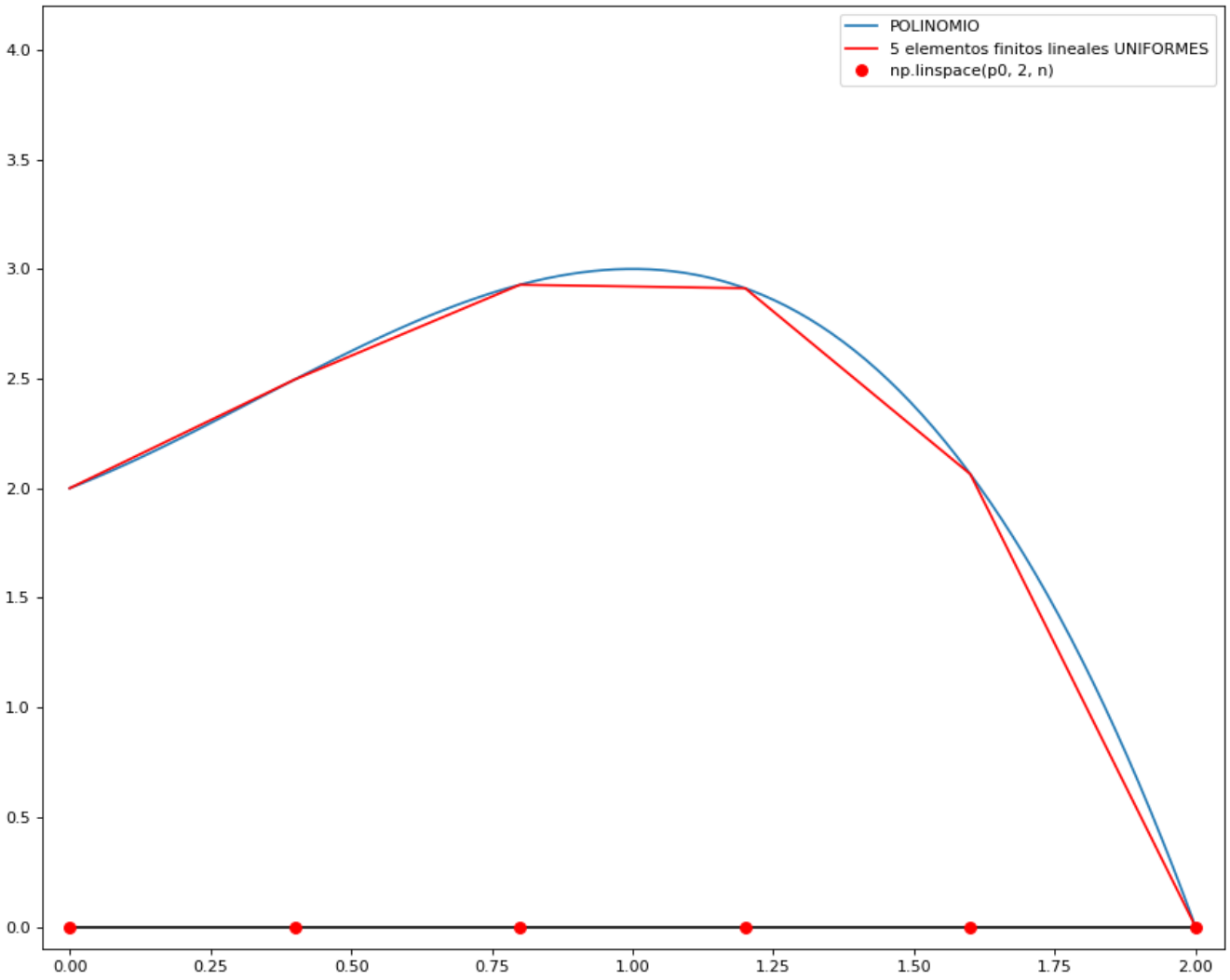
Ejemplo de aplicación:

- un objeto con densidad de masa ρ
- campo gravitatorio $\boldsymbol{g}$ debido a la atracción del objeto anterior
- aplicamos la Ley de Gauss de la gravedad en forma diferencial
  - $-\nabla \cdot \boldsymbol{g} = 4\pi G\rho$
- El campo gravitatorio es conservativo e irrotacional, con lo cual
  - $\boldsymbol{g} = -\nabla\phi$

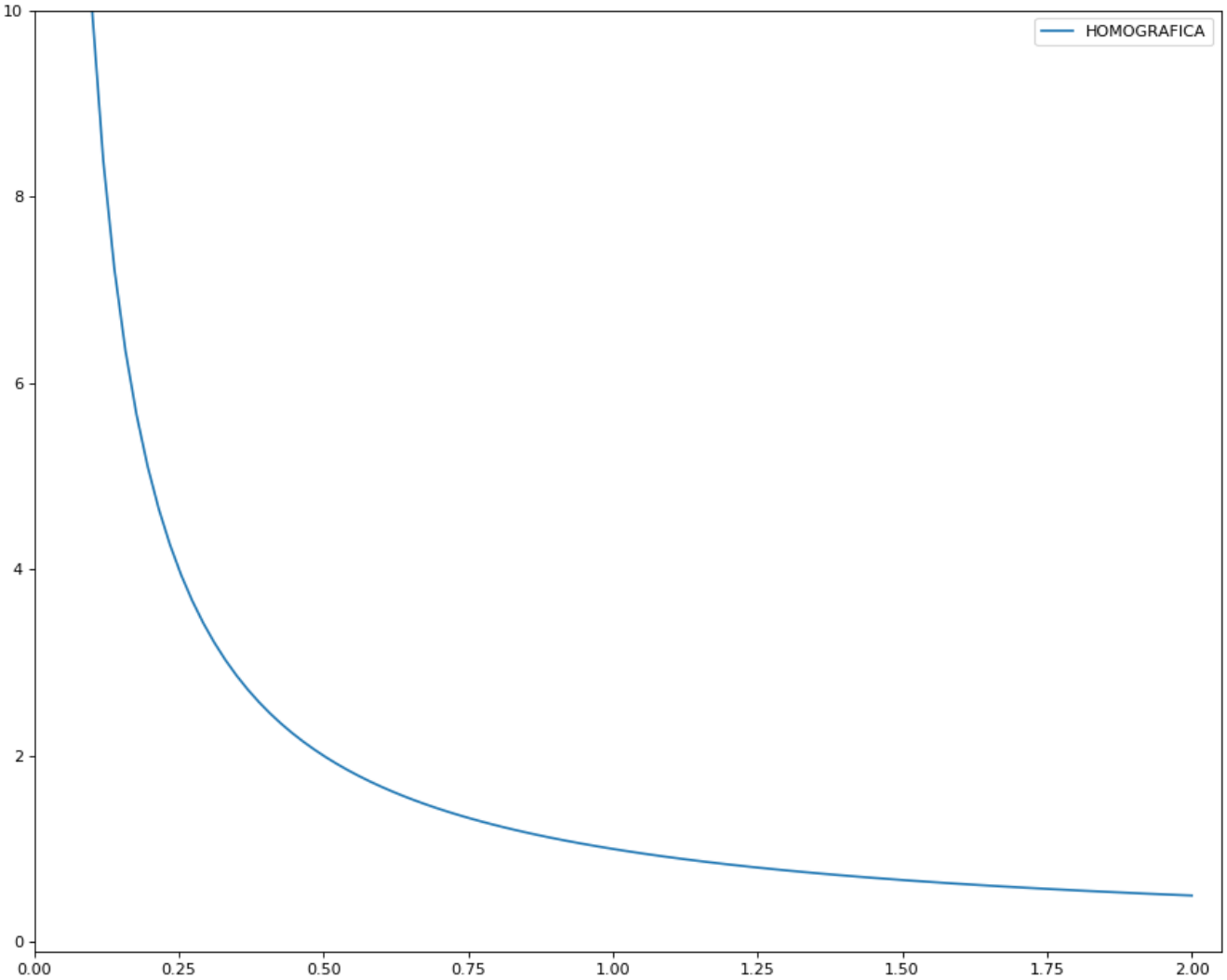
Aproximar por partes
Elementos finitos



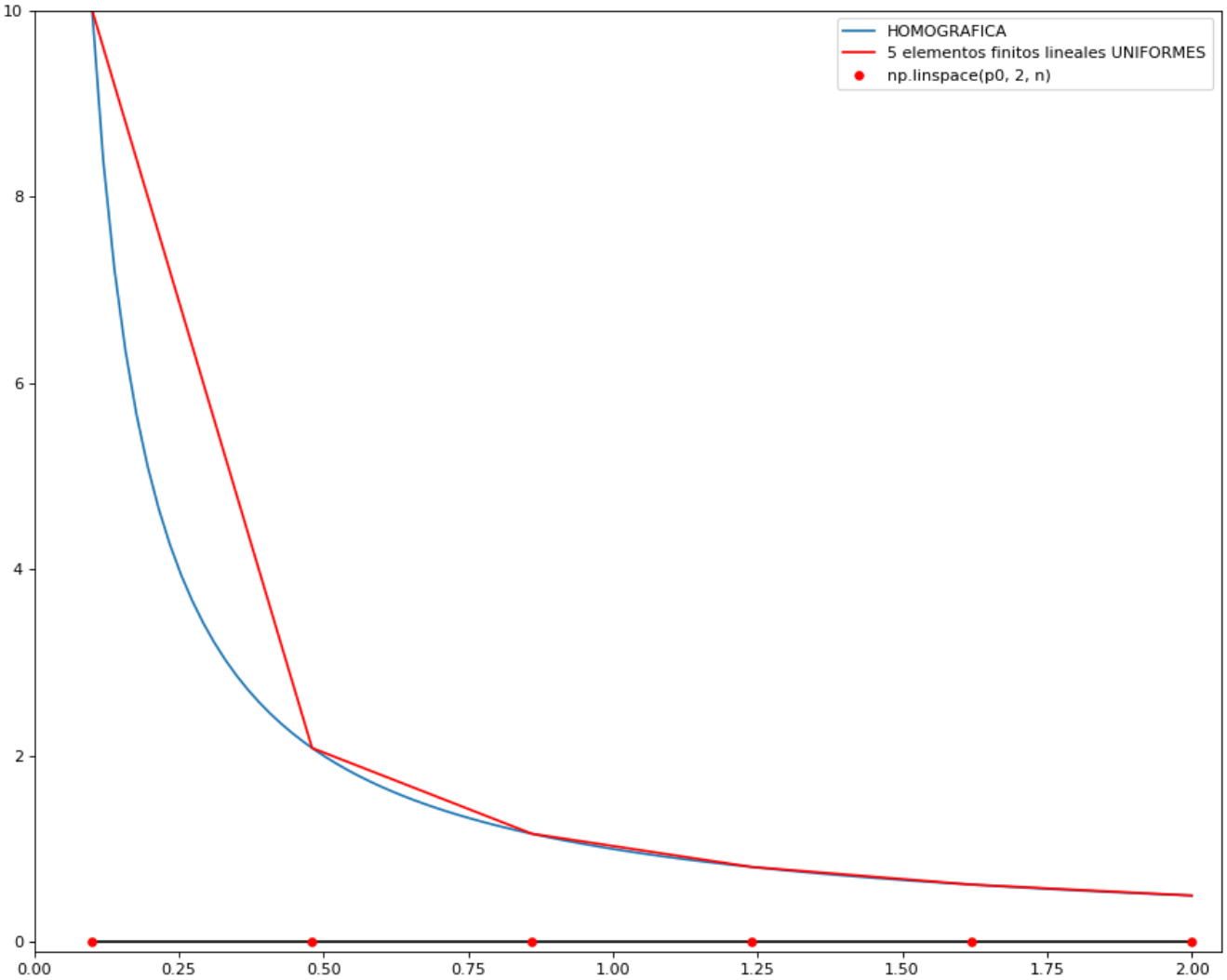
Aproximar por partes
Elementos finitos



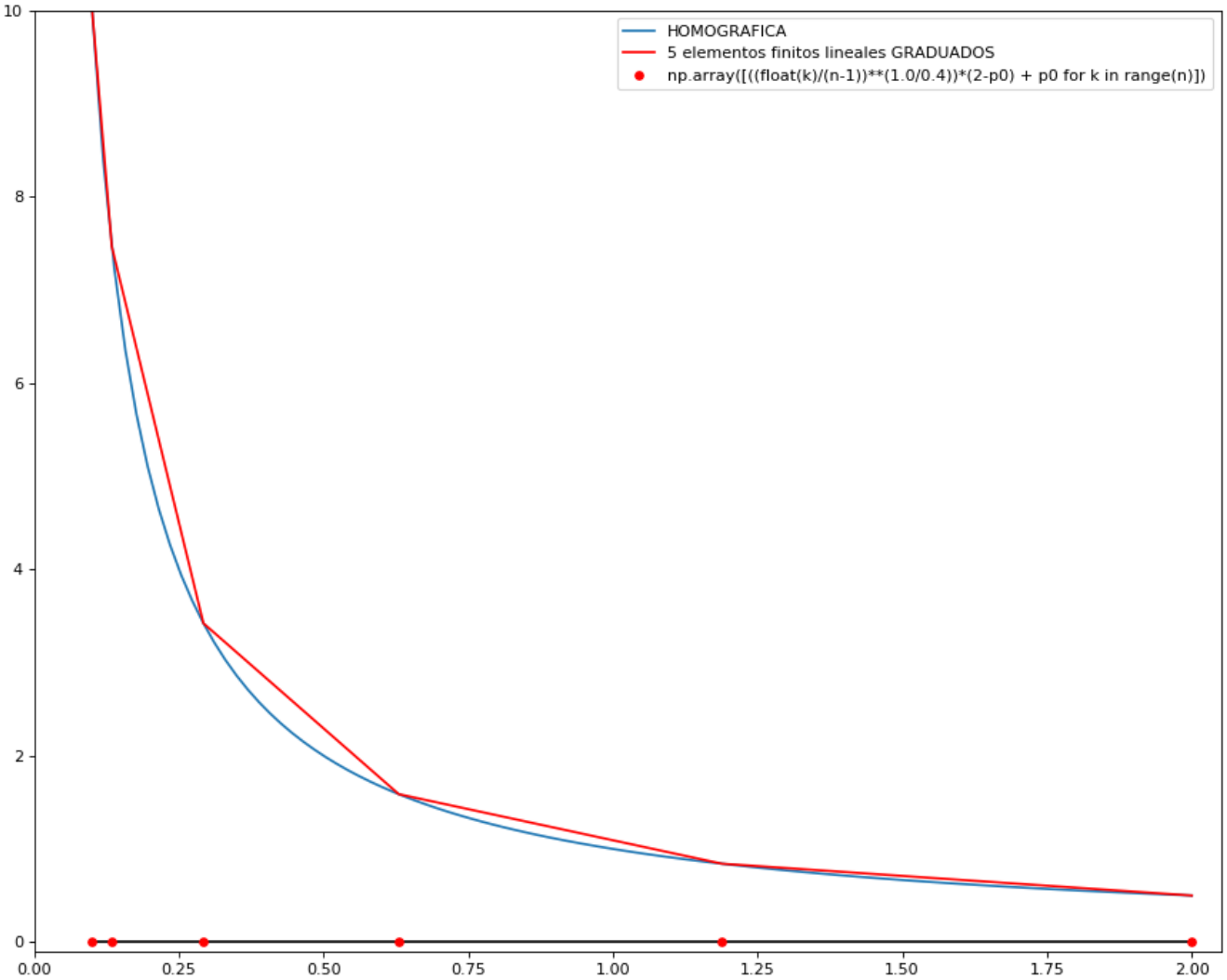
Aproximar por partes
Elementos finitos



Aproximar por partes
Elementos finitos

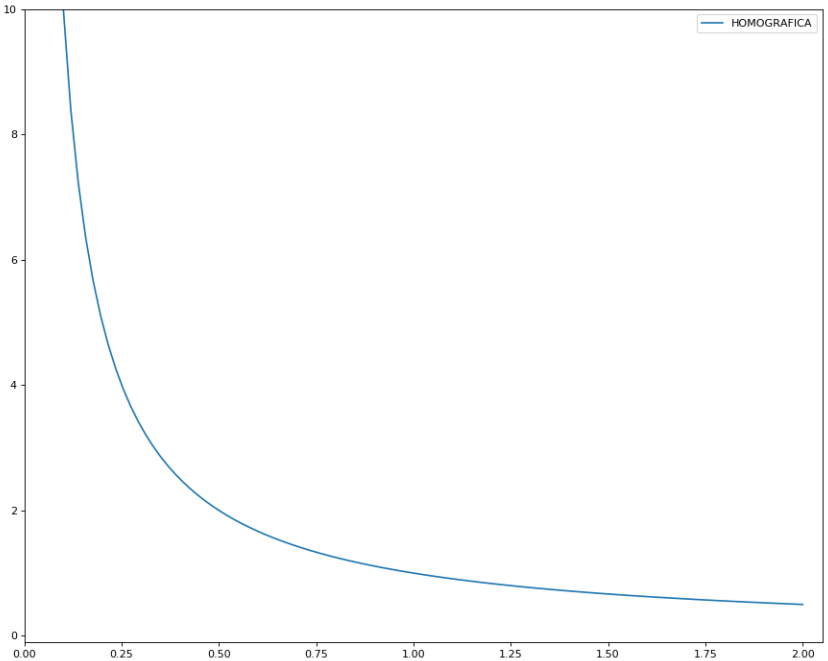


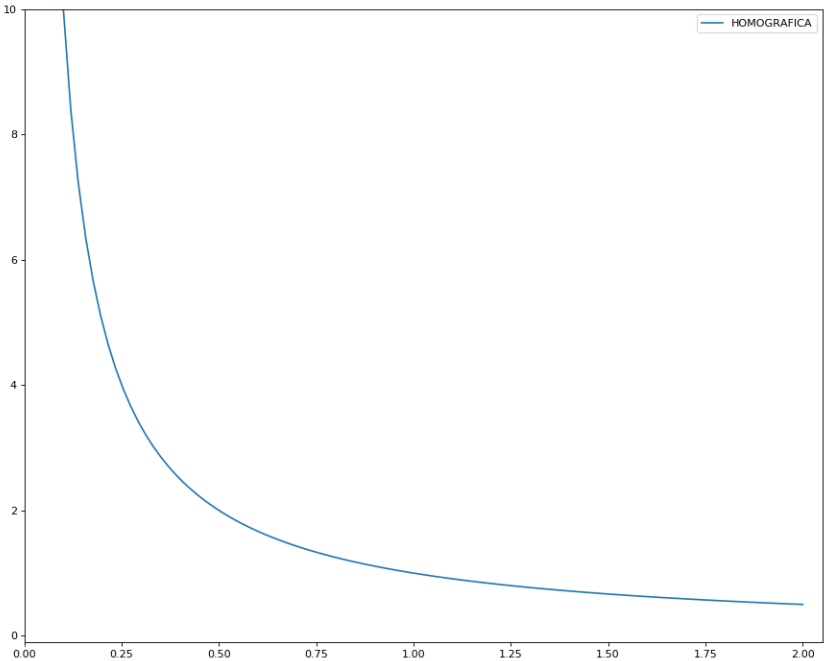
Aproximar por partes
Elementos finitos



Aproximar por partes

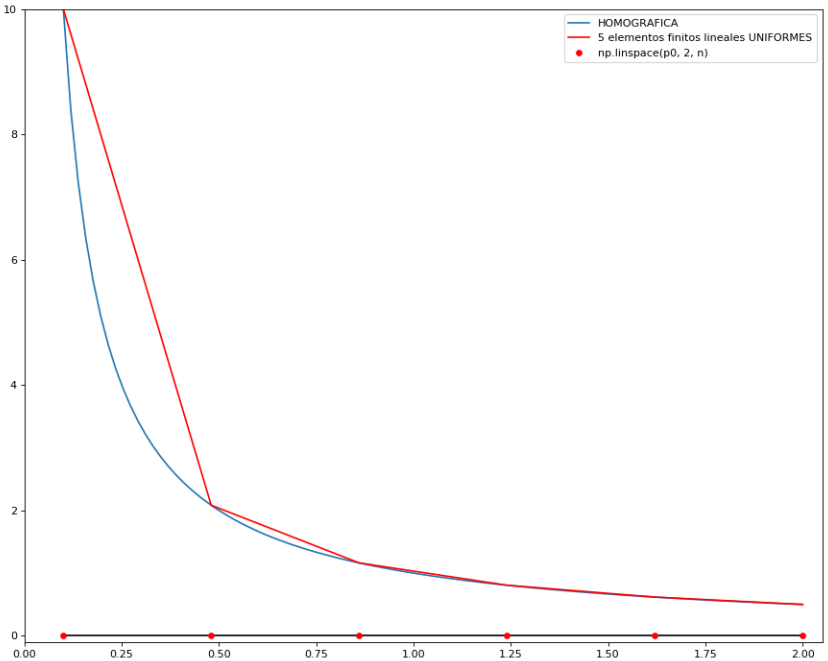
Elementos finitos uniformes vs. graduados *ad hoc*

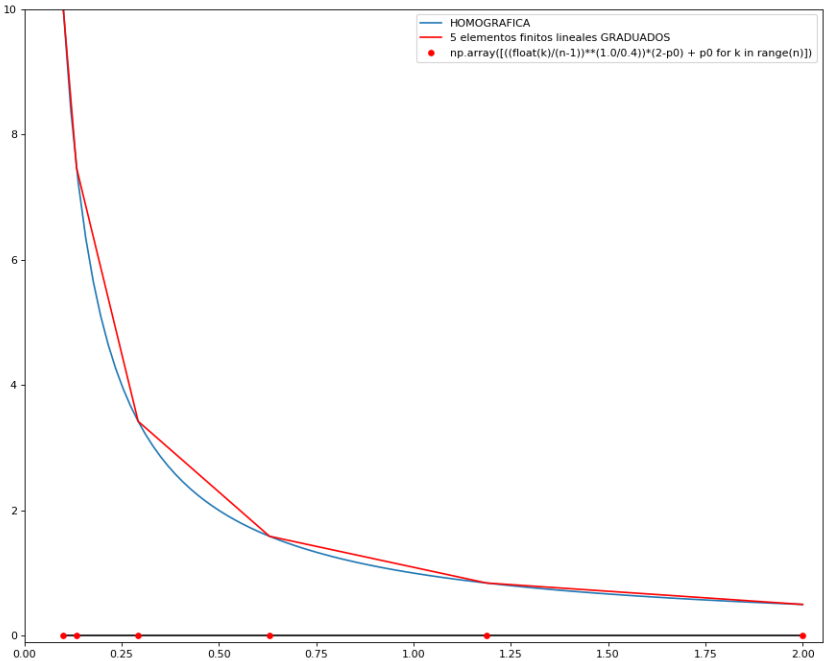




Aproximar por partes

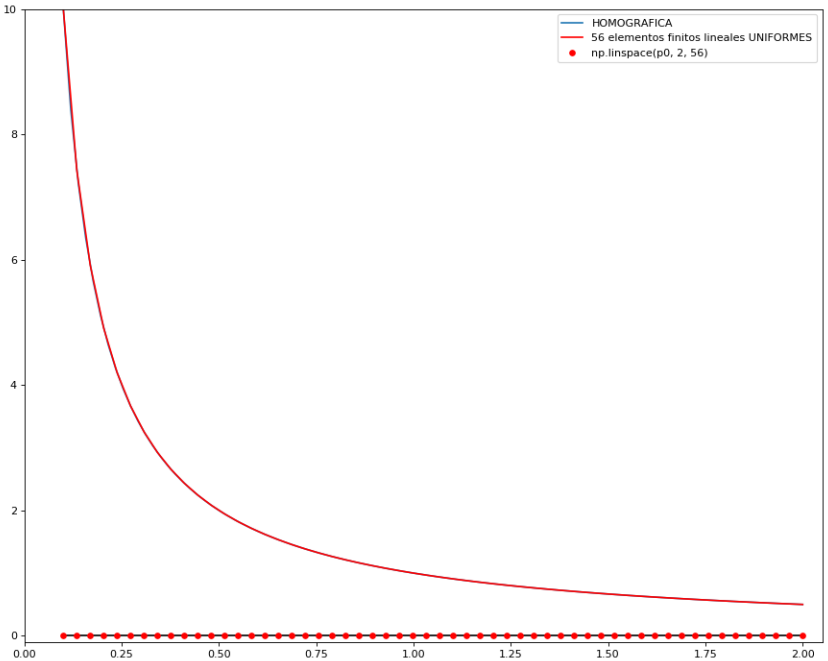
Elementos finitos uniformes vs. graduados *ad hoc*

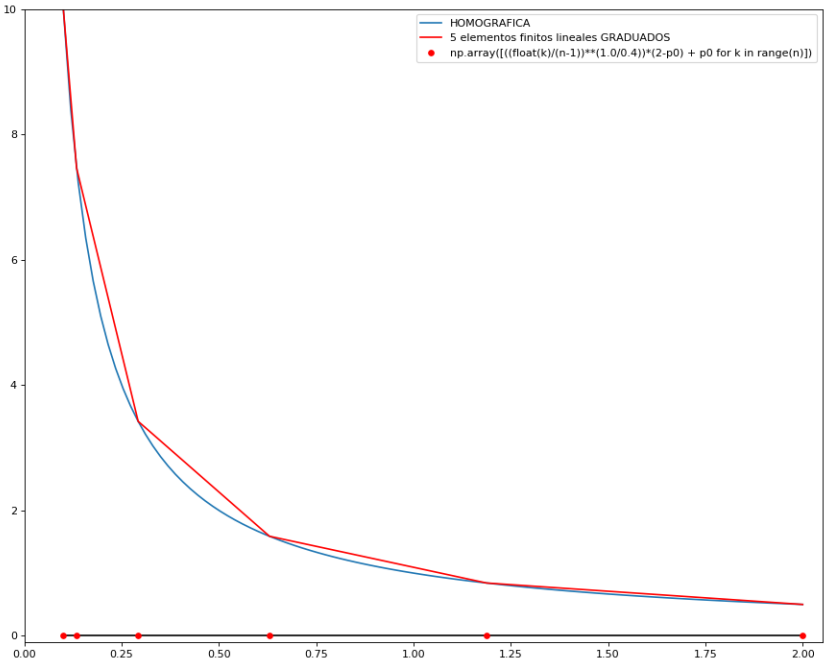




Aproximar por partes

Elementos finitos uniformes vs. graduados *ad hoc*





Nuestro problema está en el dominio Ω

- Ω no convexo.

Nuestro problema está en el dominio Ω

- Ω no convexo.
- Comportamiento singular anisótropo de la solución cerca de aristas y vértices *entrantes*.

Nuestro problema está en el dominio Ω

- Ω no convexo.
- Comportamiento singular anisótropo de la solución cerca de aristas y vértices *entrantes*.
- Consecuencia: orden no óptimo de aproximación por el método de FE *estándar*.

Nuestro problema particular fue

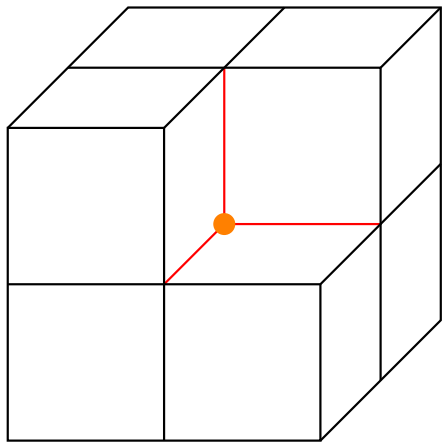
- $f(\boldsymbol{x}) = 1 + R(\boldsymbol{x})^{-3/2} \ln^{-1}(R(\boldsymbol{x})/4)$

Nuestro problema particular fue

- $f(\boldsymbol{x}) = 1 + R(\boldsymbol{x})^{-3/2} \ln^{-1}(R(\boldsymbol{x})/4)$
- $R(\boldsymbol{x}) =$ *distancia al vértice cóncavo*.

Nuestro problema particular fue

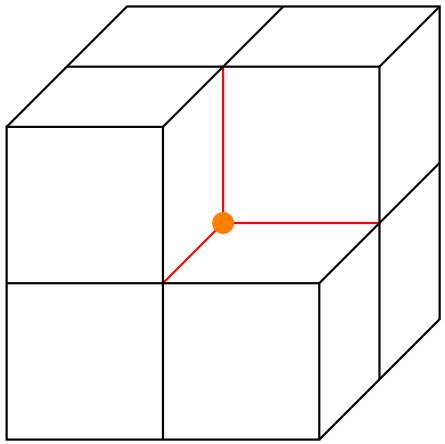
- $f(\boldsymbol{x}) = 1 + R(\boldsymbol{x})^{-3/2} \ln^{-1}(R(\boldsymbol{x})/4)$
- $R(\boldsymbol{x}) =$ *distancia al vértice cóncavo*.



$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}) &= \nabla p(\boldsymbol{x}) \\ -\nabla \cdot \boldsymbol{u}(\boldsymbol{x}) &= 1 + R(\boldsymbol{x})^{-3/2} \ln^{-1}(R(\boldsymbol{x})/4) \\ p|_{\partial\Omega} &= 0. \end{aligned}$$

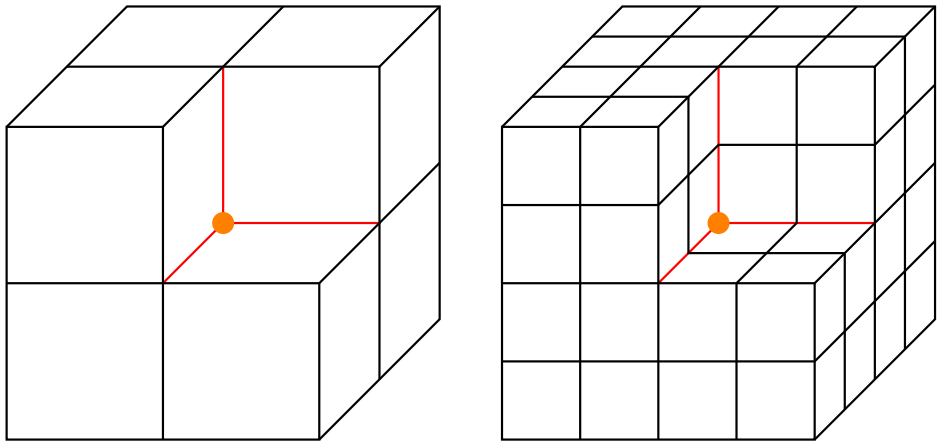
Nuestro problema está en el dominio Ω

Método de FE *estándar*



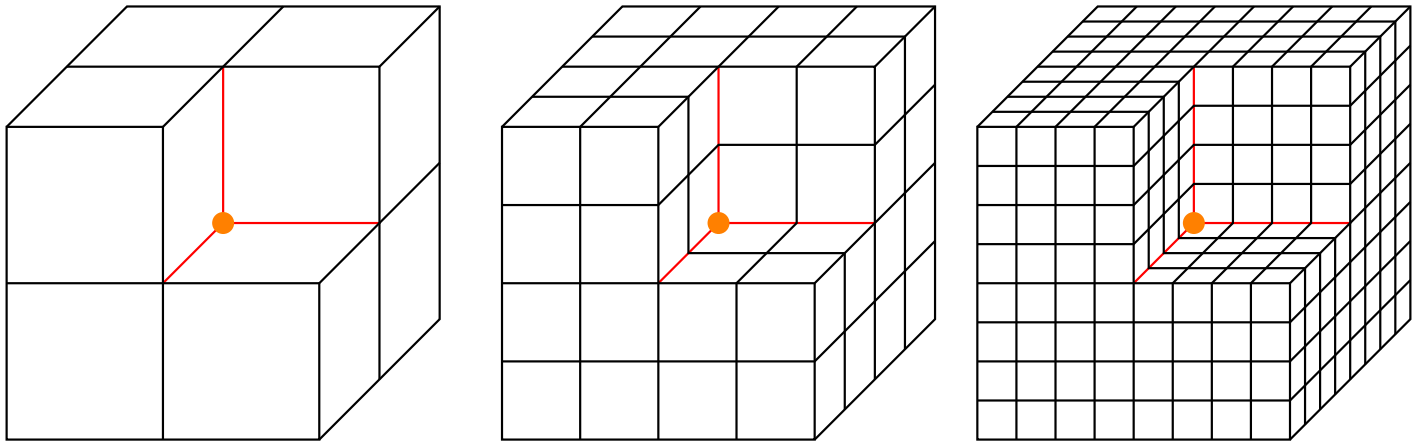
Nuestro problema está en el dominio Ω

Método de FE *estándar*



Nuestro problema está en el dominio Ω

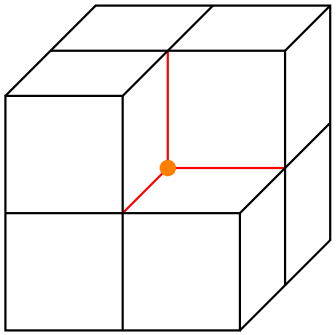
Método de FE *estándar*



(mallas uniformes, cuasi-uniformes).

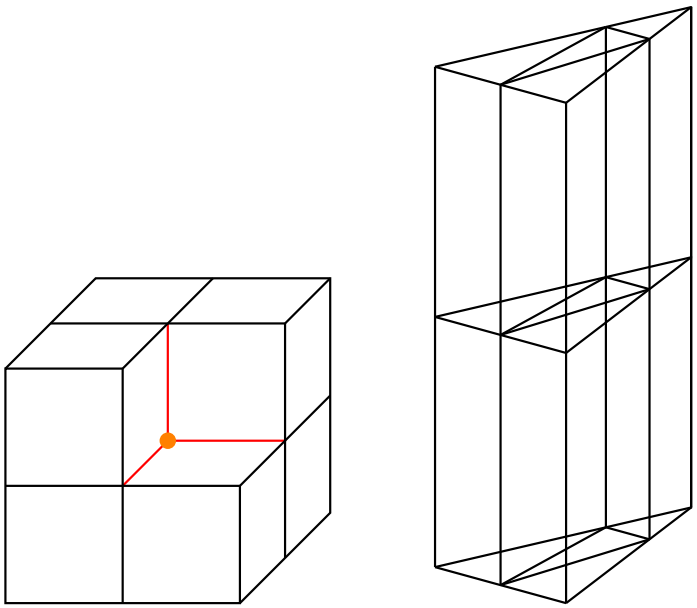
Nuestro problema está en el dominio Ω

- Queremos refinamiento anisótropo cerca de aristas singulares



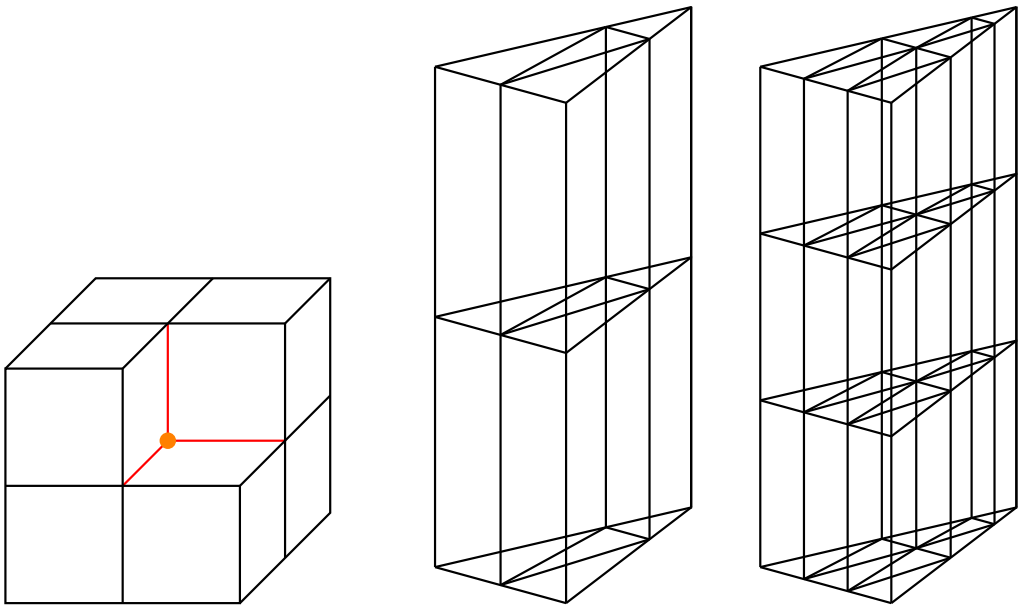
Nuestro problema está en el dominio Ω

- Queremos refinamiento anisótropo cerca de aristas singulares



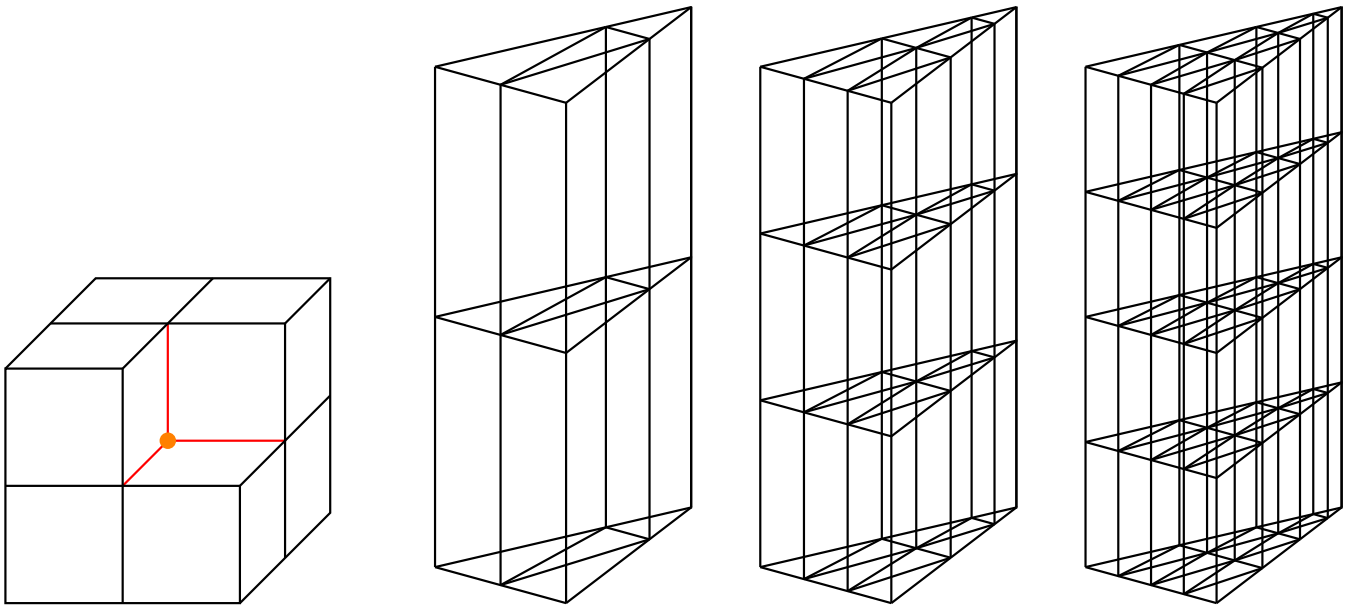
Nuestro problema está en el dominio Ω

- Queremos refinamiento anisótropo cerca de aristas singulares



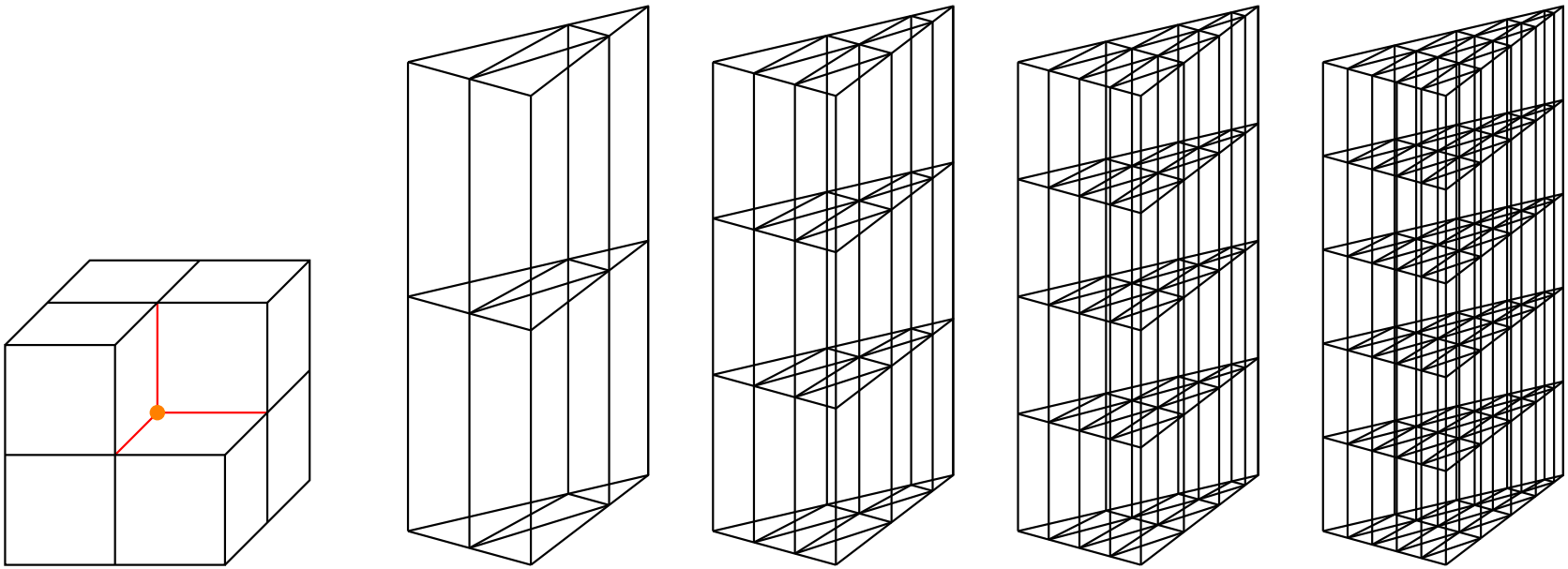
Nuestro problema está en el dominio Ω

- Queremos refinamiento anisótropo cerca de aristas singulares



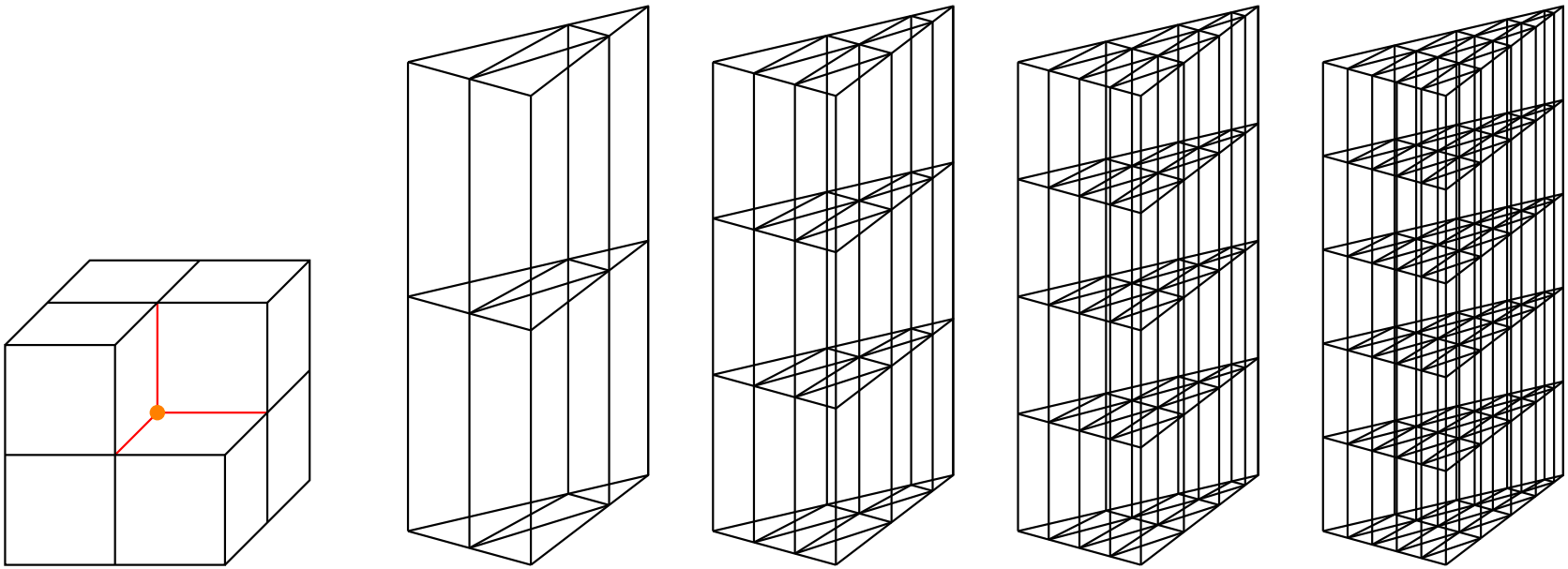
Nuestro problema está en el dominio Ω

- Queremos refinamiento anisótropo cerca de aristas singulares



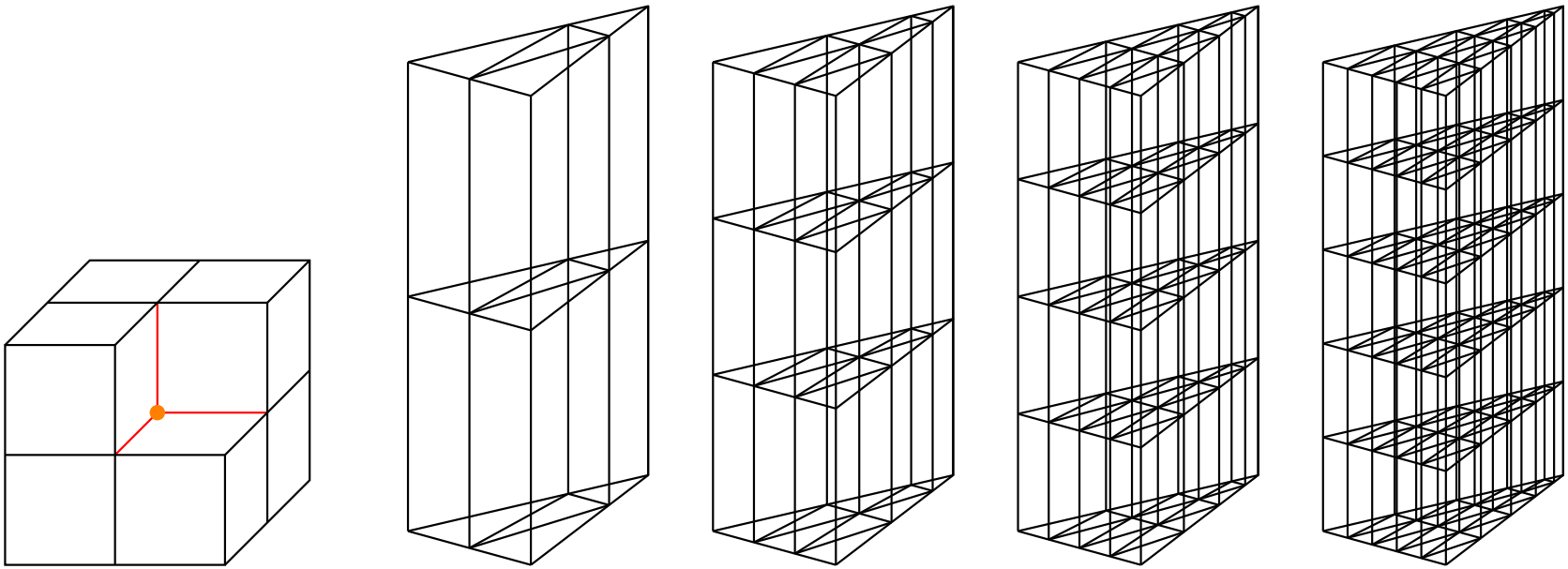
Nuestro problema está en el dominio Ω

- Queremos refinamiento anisótropo cerca de aristas singulares



Nuestro problema está en el dominio Ω

- Queremos refinamiento anisótropo cerca de aristas singulares



$$\left\{ \frac{\text{diam}(E_{n,k})}{\rho(E_{n,k})} \mid n \in \mathbb{N}, 1 \leqslant k \leqslant N_{\mathcal{T}_n} \right\} \text{ no acotado}$$

Nuestro problema está en el dominio Ω

- combinado con

Nuestro problema está en el dominio Ω

- combinado con
  - refinamiento isótropo graduado cerca de vértices singulares

Nuestro problema está en el dominio Ω

- combinado con
  - refinamiento isótropo graduado cerca de vértices singulares
  - submallas cuasi–uniformes en macro–elementos sin singularidades

Nuestro problema está en el dominio Ω

- combinado con
  - refinamiento isótropo graduado cerca de vértices singulares
  - submallas cuasi–uniformes en macro–elementos sin singularidades
- La principal dificultad es describir y construir las mallas en la transición desde zonas de anisotropía hacia zonas de isotropía

Nuestro problema está en el dominio Ω

- combinado con
  - refinamiento isótropo graduado cerca de vértices singulares
  - submallas cuasi–uniformes en macro–elementos sin singularidades
- La principal dificultad es describir y construir las mallas en la transición desde zonas de anisotropía hacia zonas de isotropía
- Dominios poliedrales generales, no necesariamente cilíndricos.

Hacia el método propuesto
un poco de los objetos matemáticos involucrados en la solución

- $\bar{\Omega} = \cup_{\ell=1}^{M} \bar{\Lambda}_\ell$ (los macro–elementos)

Hacia el método propuesto
un poco de los objetos matemáticos involucrados en la solución

- $\bar{\Omega} = \cup_{\ell=1}^{M} \bar{\Lambda}_\ell$ (los macro–elementos)
  - prismáticos o tetraedros.

Hacia el método propuesto
un poco de los objetos matemáticos involucrados en la solución

- $\bar{\Omega} = \cup_{\ell=1}^{M} \bar{\Lambda}_\ell$ (los macro–elementos)
  - prismáticos o tetraedros.
  - una arista o un vértice entrantes en cada uno.

Hacia el método propuesto
un poco de los objetos matemáticos involucrados en la solución

- $\bar{\Omega} = \cup_{\ell=1}^{M} \bar{\Lambda}_\ell$ (los macro–elementos)
  - prismáticos o tetraedros.
  - una arista o un vértice entrantes en cada uno.
- Coordenadas locales en cada Λ_ℓ

Hacia el método propuesto
un poco de los objetos matemáticos involucrados en la solución

- $\bar{\Omega} = \cup_{\ell=1}^{M} \bar{\Lambda}_\ell$ (los macro–elementos)
  - prismáticos o tetraedros.
  - una arista o un vértice entrantes en cada uno.
- Coordenadas locales en cada Λ_ℓ
  - $\boldsymbol{\xi}^{(\ell)} = (\xi_1^{(\ell)}, \xi_2^{(\ell)}, \xi_3^{(\ell)})$

Hacia el método propuesto
un poco de los objetos matemáticos involucrados en la solución

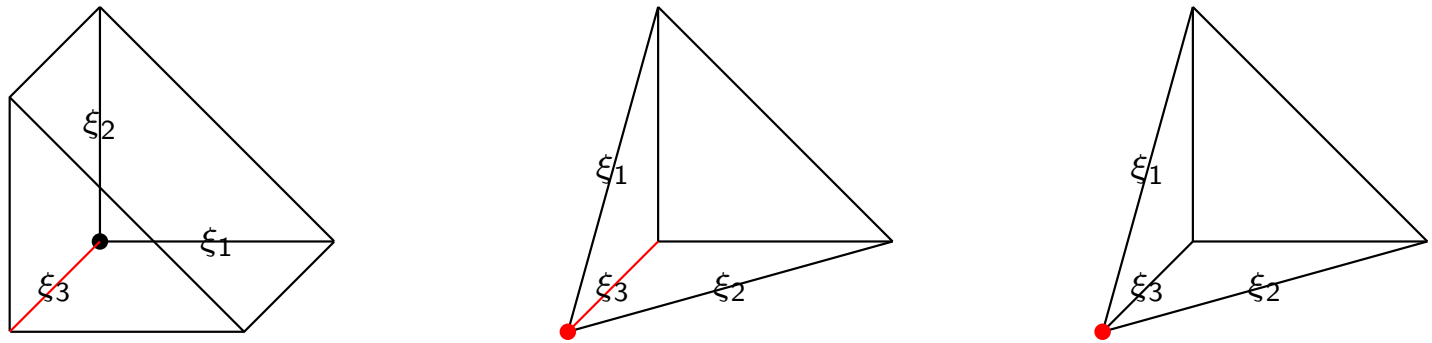
- $\bar{\Omega} = \cup_{\ell=1}^{M} \bar{\Lambda}_\ell$ (los macro–elementos)
  - prismáticos o tetraedros.
  - una arista o un vértice entrantes en cada uno.
- Coordenadas locales en cada Λ_ℓ
  - $\boldsymbol{\xi}^{(\ell)} = (\xi_1^{(\ell)}, \xi_2^{(\ell)}, \xi_3^{(\ell)})$
  - Origen = vértice singular (si existe).

Hacia el método propuesto
un poco de los objetos matemáticos involucrados en la solución

- $\bar{\Omega} = \cup_{\ell=1}^{M} \bar{\Lambda}_\ell$ (los macro–elementos)
  - prismáticos o tetraedros.
  - una arista o un vértice entrantes en cada uno.
- Coordenadas locales en cada Λ_ℓ
  - $\boldsymbol{\xi}^{(\ell)} = (\xi_1^{(\ell)}, \xi_2^{(\ell)}, \xi_3^{(\ell)})$
  - Origen = vértice singular (si existe).
  - Eje $\xi_3^{(\ell)}$: contiene a la arista singular (si existe).

Hacia el método propuesto
un poco de los objetos matemáticos involucrados en la solución

- $\bar{\Omega} = \cup_{\ell=1}^{M} \bar{\Lambda}_\ell$ (los macro–elementos)
  - prismáticos o tetraedros.
  - una arista o un vértice entrantes en cada uno.
- Coordenadas locales en cada Λ_ℓ
  - $\boldsymbol{\xi}^{(\ell)} = (\xi_1^{(\ell)}, \xi_2^{(\ell)}, \xi_3^{(\ell)})$
  - Origen = vértice singular (si existe).
  - Eje $\xi_3^{(\ell)}$: contiene a la arista singular (si existe).

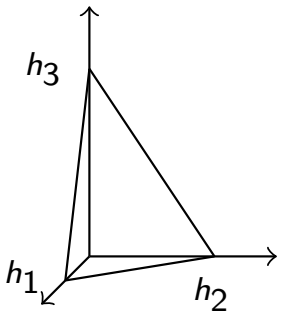


(a) prismático (b) tetr. I (c) tetr. II

Figure: Macro–elementos.

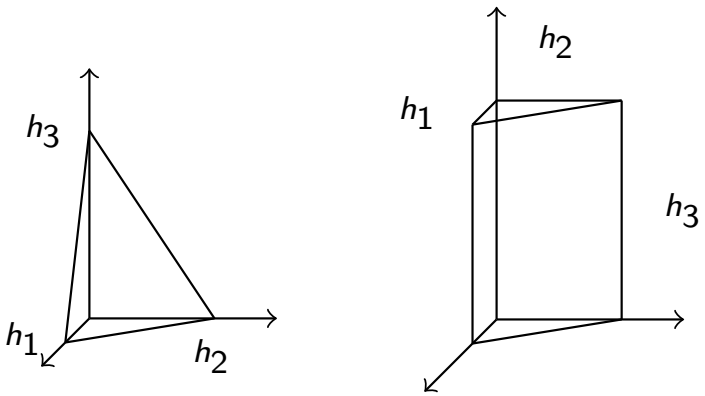
Sub–mallas anidadas

Dentro de cada uno de esos tres habrá:



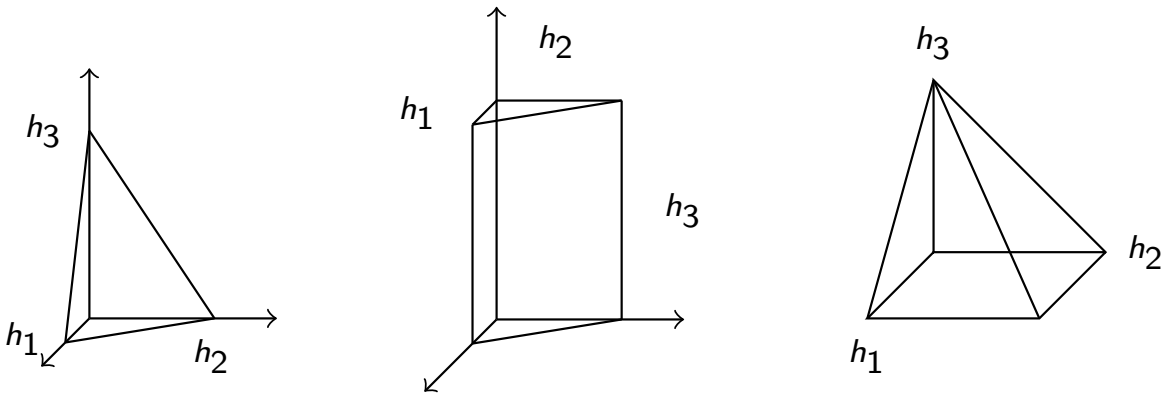
Sub–mallas anidadas

Dentro de cada uno de esos tres habrá:



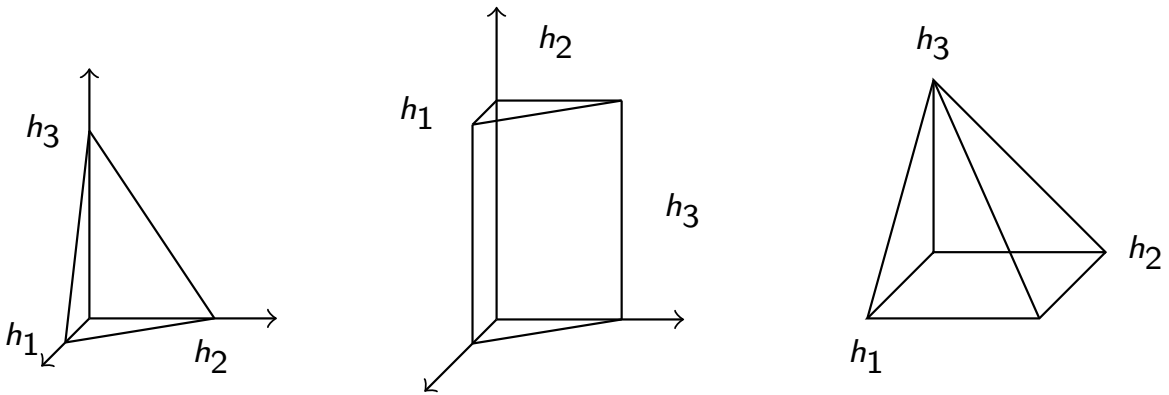
Sub–mallas anidadas

Dentro de cada uno de esos tres habrá:



Sub–mallas anidadas

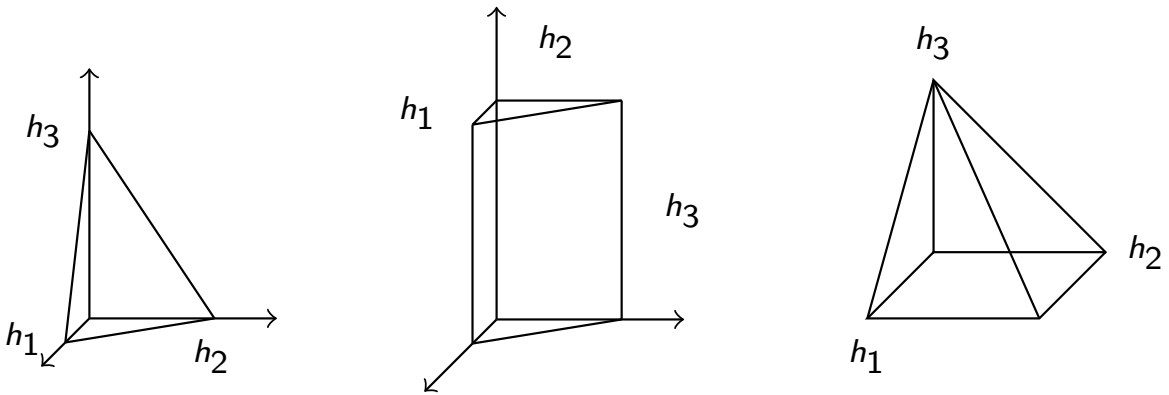
Dentro de cada uno de esos tres habrá:



tetraedros, prismas y pirámdes.

Sub–mallas anidadas

Dentro de cada uno de esos tres habrá:

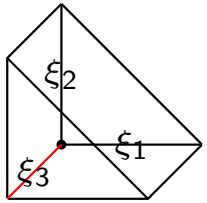


tetraedros, prismas y pirámdes.

Los (h_1, h_2, h_3) son los tamaños seccionales de cada elemento de una malla.

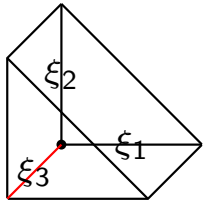
Método propuesto

- prismas angostos cerca de aristas singulares

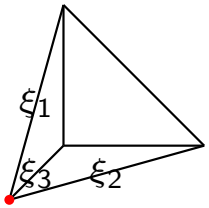


Método propuesto

- prismas angostos cerca de aristas singulares

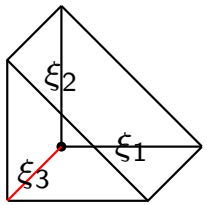


- tetraedros graduados hacia vértices singulares

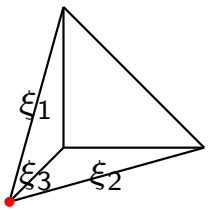


Método propuesto

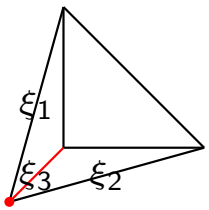
- prismas angostos cerca de aristas singulares



- tetraedros graduados hacia vértices singulares



- pirámides en zonas de transición en donde haya ambas sing.



Restr. para la malla dentro de un macro–elemento

ej. de criterio graduación

En una sucesión de mallas refinadas y anidadas, en la malla número n tenemos:

Restr. para la malla dentro de un macro–elemento

ej. de criterio graduación

En una sucesión de mallas refinadas y anidadas, en la malla número n tenemos:

$$h_1, h_2 \sim \begin{cases} (1/n)^{\frac{1}{\mu}} & \text{if } d(E, A_S) = 0 \\ (1/n)\, d(E, A_S)^{1-\mu} & \text{if } 0 < d(E, A_S) < 1 \end{cases}$$

$$h_3 \sim \begin{cases} (1/n)^{\frac{1}{\nu}} & \text{if } d(E, S) = 0 \\ (1/n)\, d(E, S)^{1-\nu} & \text{if } 0 < d(E, S) < 1 \end{cases}$$

Restr. para la malla dentro de un macro–elemento

ej. de criterio graduación

En una sucesión de mallas refinadas y anidadas, en la malla número n tenemos:

$$h_1, h_2 \sim \begin{cases} (1/n)^{\frac{1}{\mu}} & \text{if } d(E, A_S) = 0 \\ (1/n)\, d(E, A_S)^{1-\mu} & \text{if } 0 < d(E, A_S) < 1 \end{cases}$$

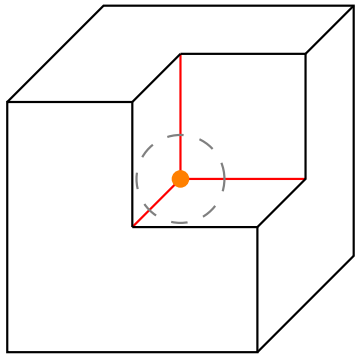
$$h_3 \sim \begin{cases} (1/n)^{\frac{1}{\nu}} & \text{if } d(E, S) = 0 \\ (1/n)\, d(E, S)^{1-\nu} & \text{if } 0 < d(E, S) < 1 \end{cases}$$

μ y ν parámetros dados por la teoría de EDP.

Subdominio solo con tetraedros

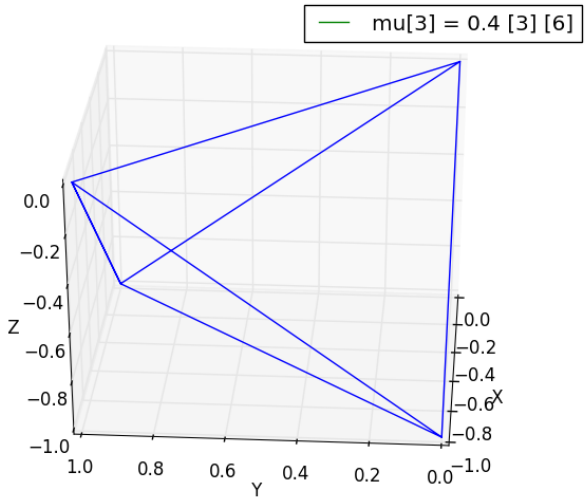
graduado vs. uniforme

Situación:



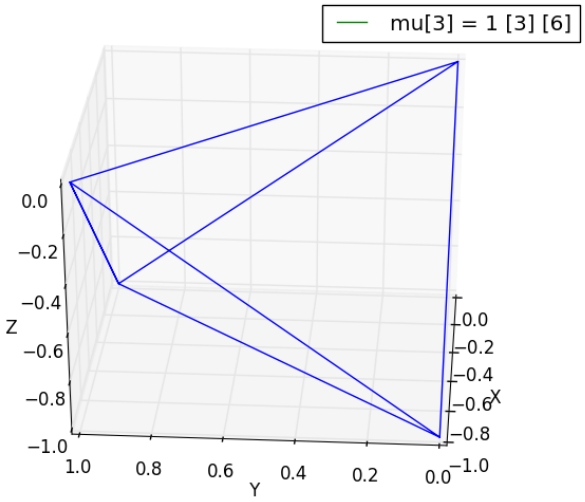
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

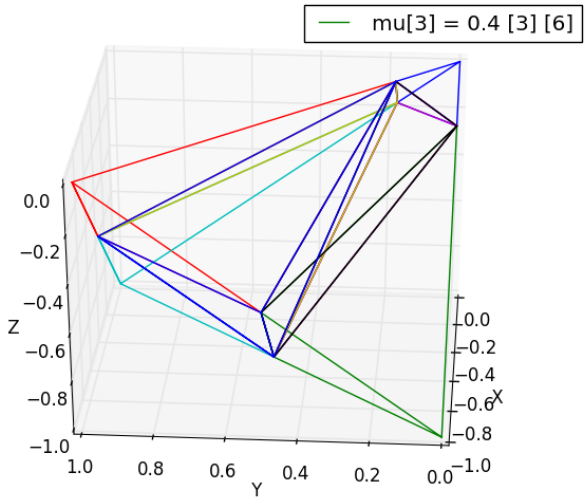
vs.



método estándar

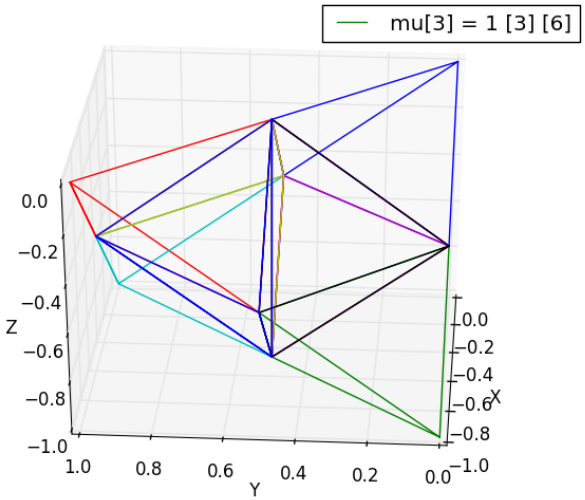
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

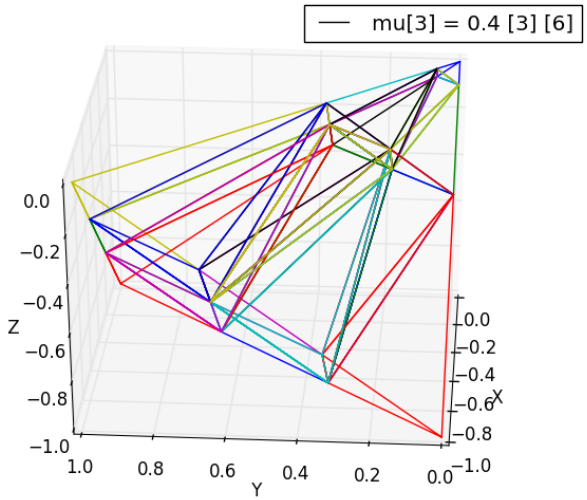
vs.



método estándar

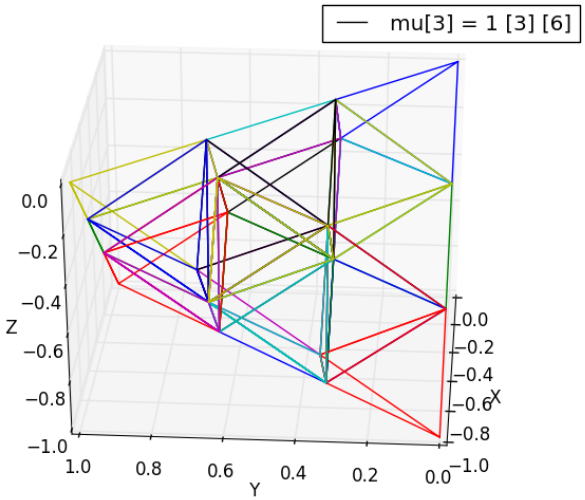
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

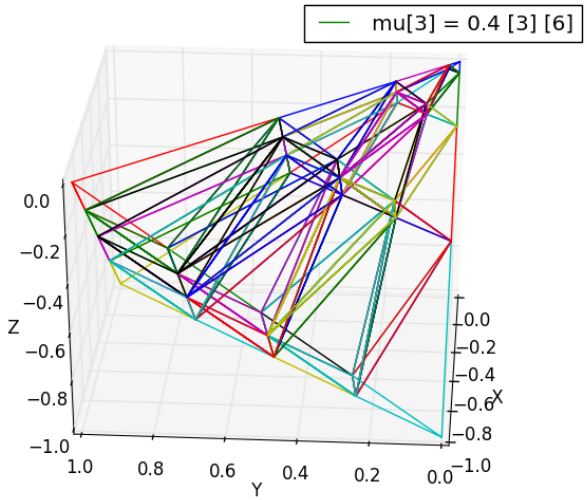
vs.



método estándar

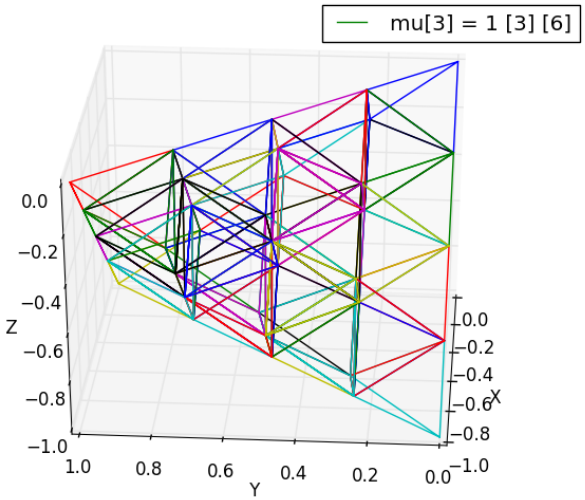
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

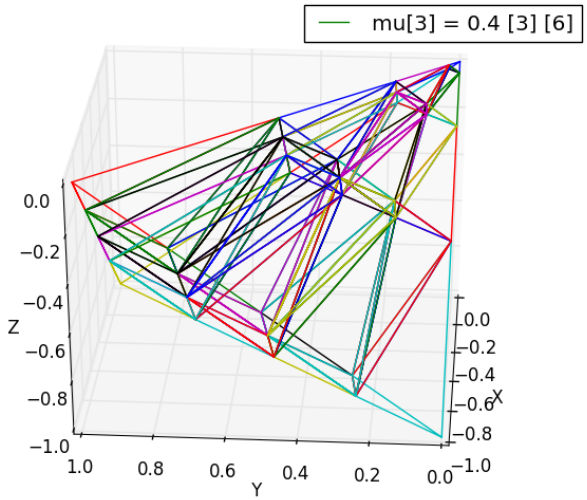
vs.



método estándar

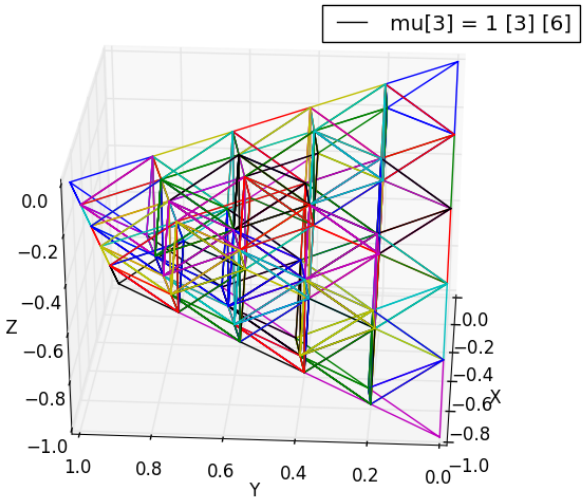
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

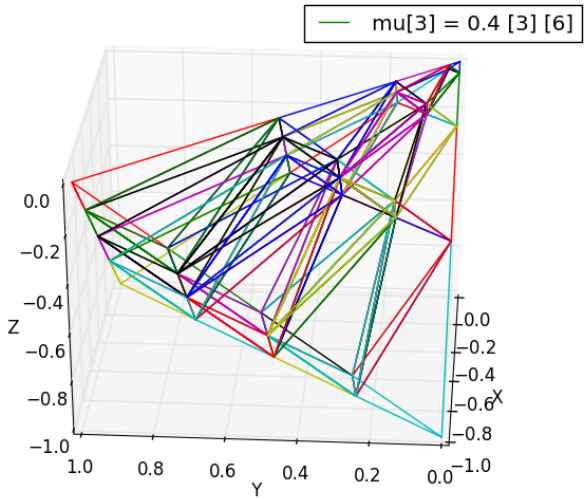
vs.



método estándar

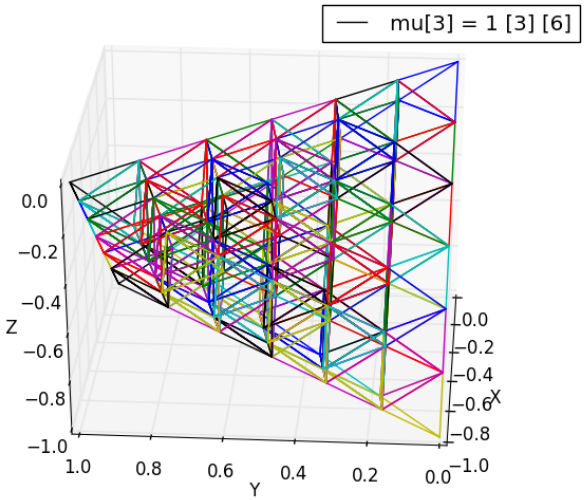
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

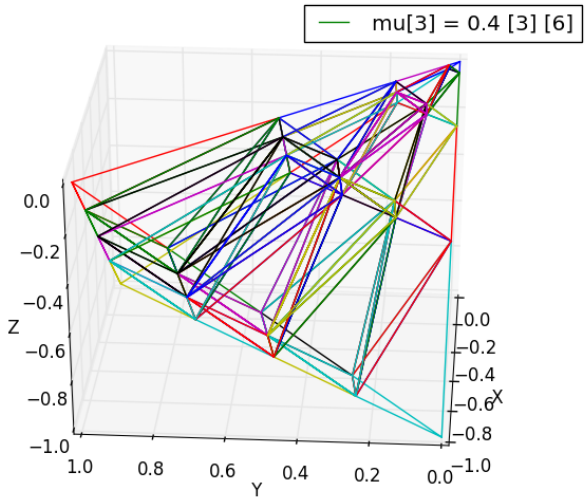
vs.



método estándar

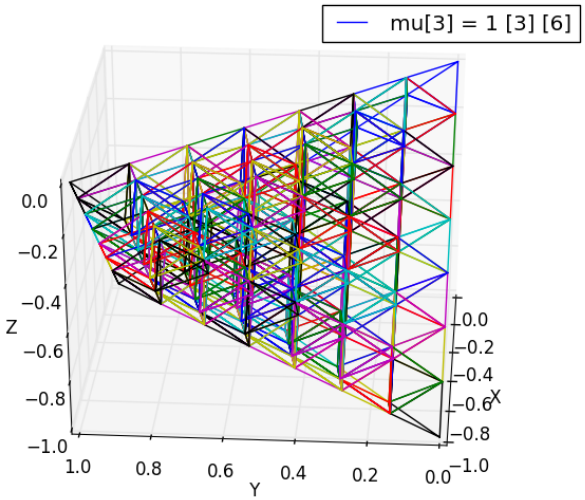
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

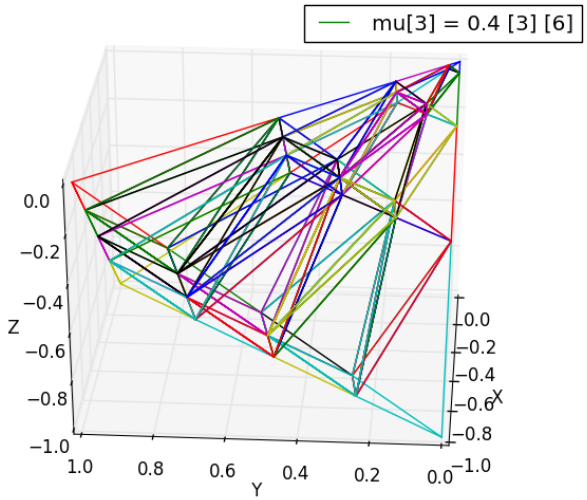
vs.



método estándar

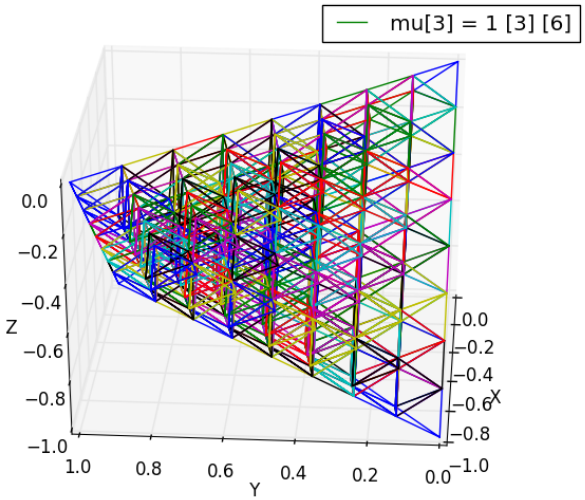
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

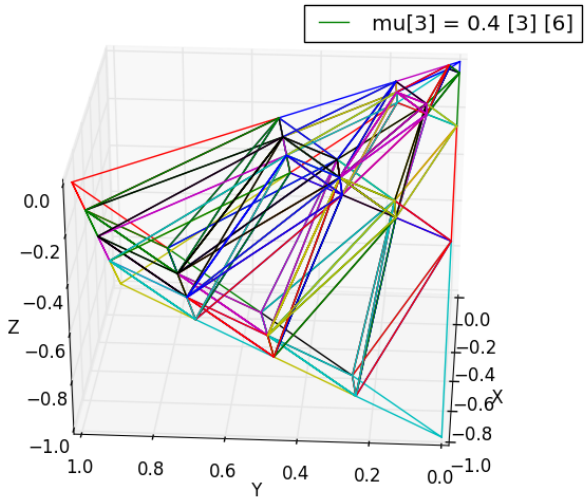
vs.



método estándar

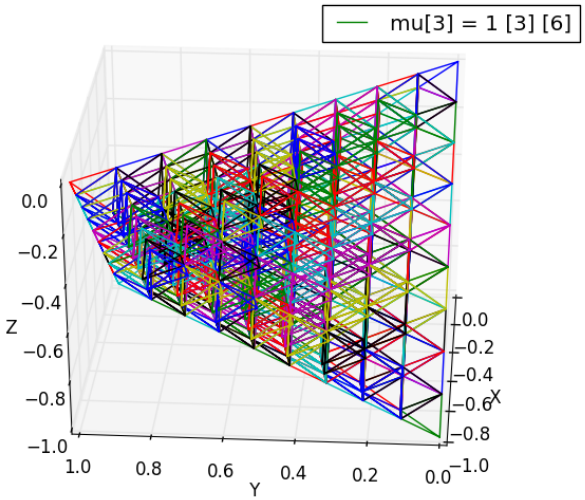
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

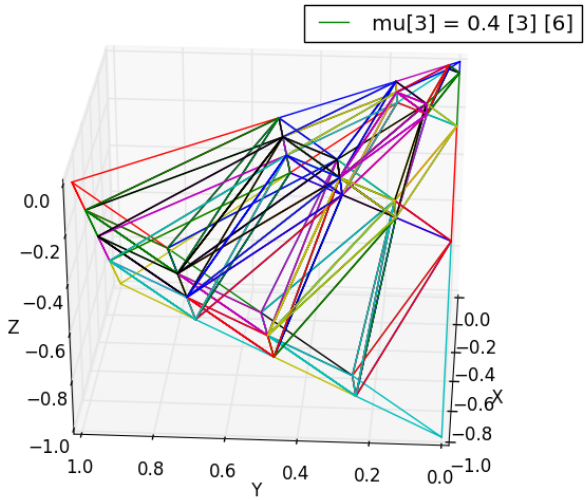
vs.

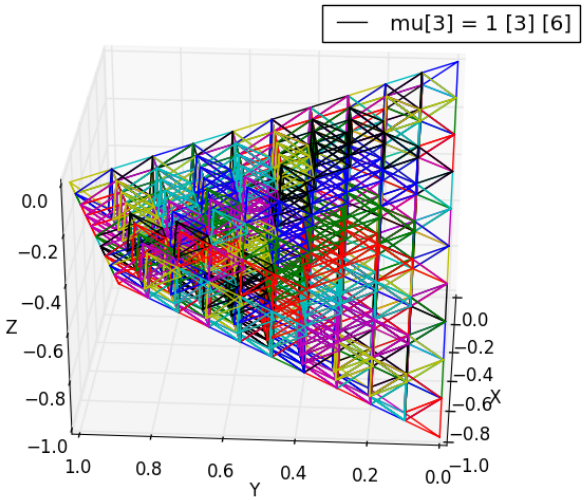


método estándar

Subdominio solo con tetraedros

graduado vs. uniforme

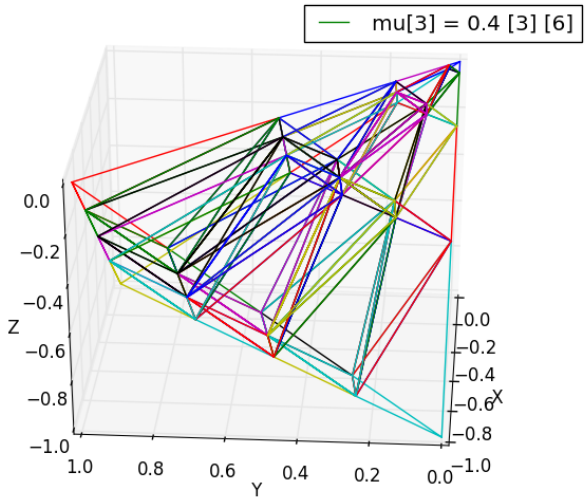




nuestro método vs. método estándar

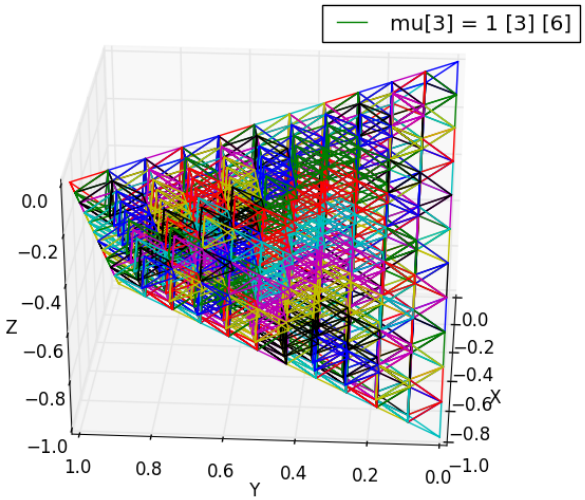
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

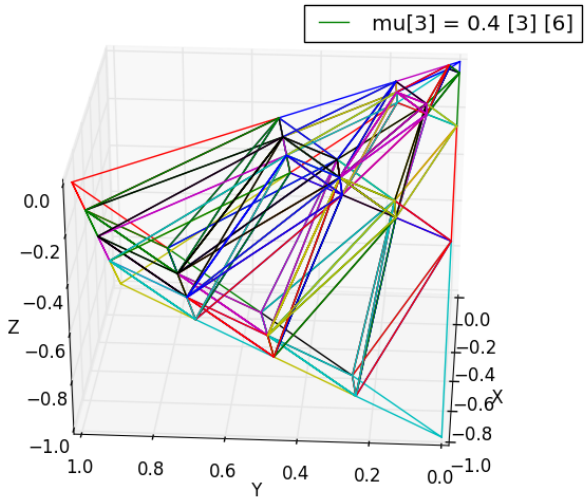
vs.



método estándar

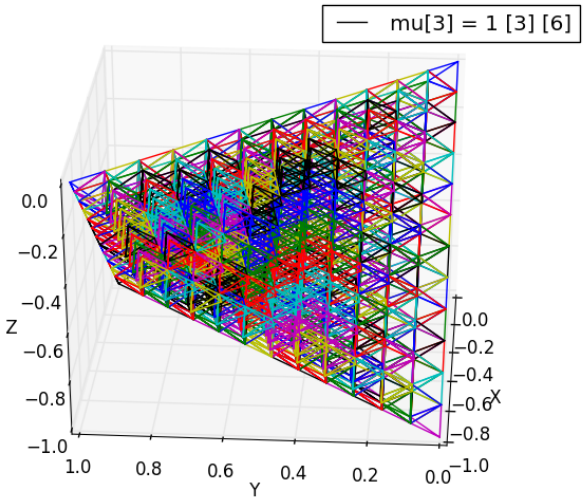
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

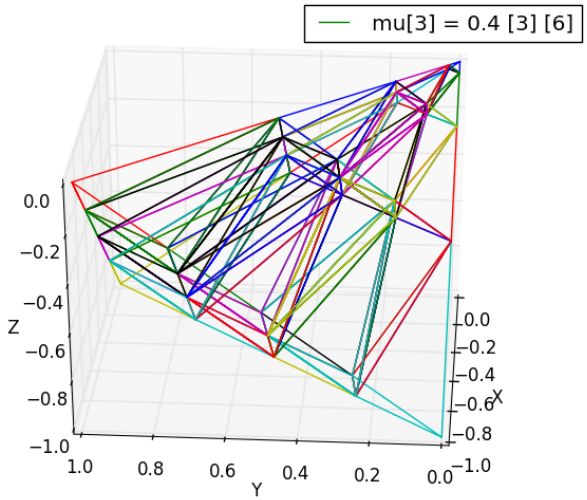
vs.



método estándar

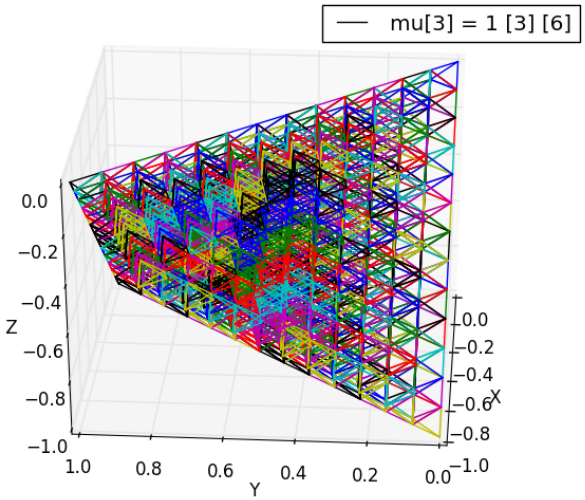
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

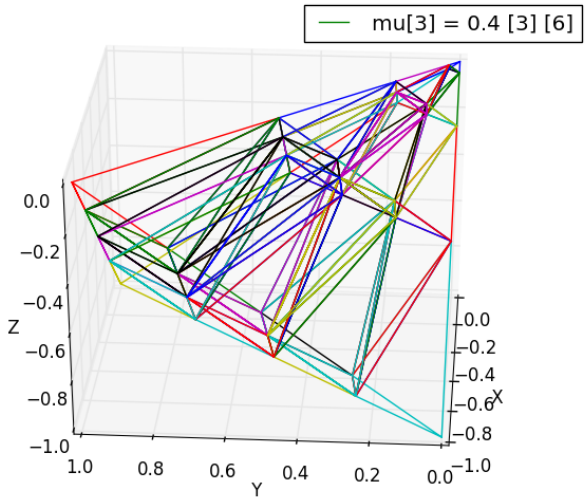
vs.



método estándar

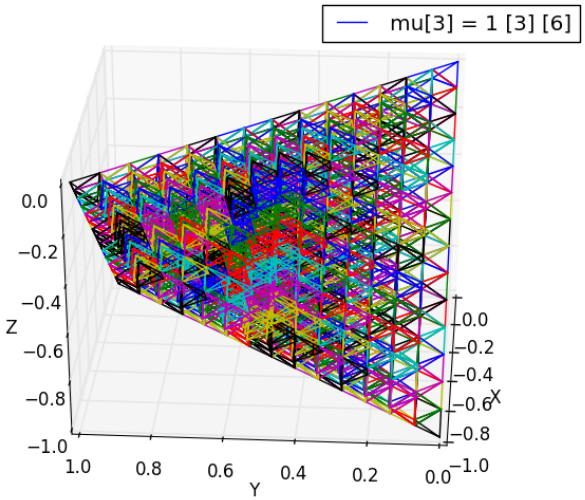
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

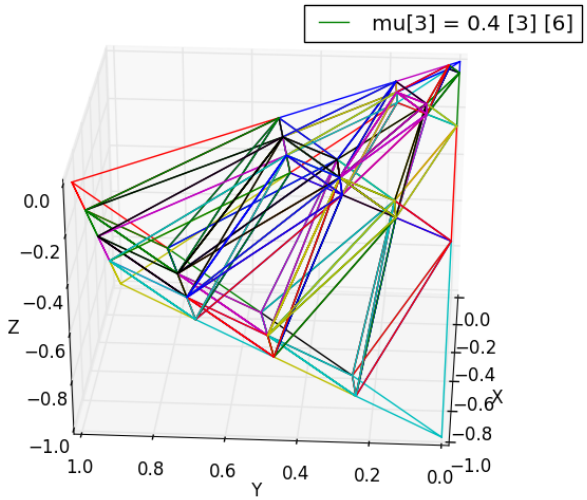
vs.



método estándar

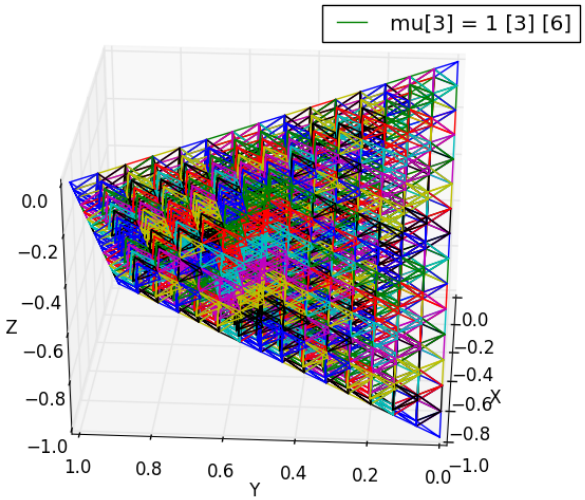
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

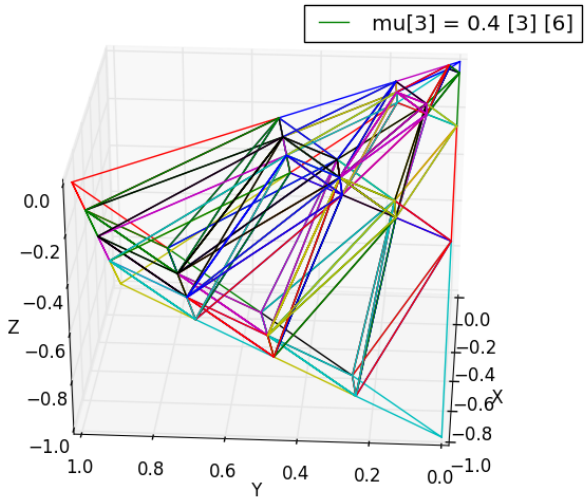
vs.



método estándar

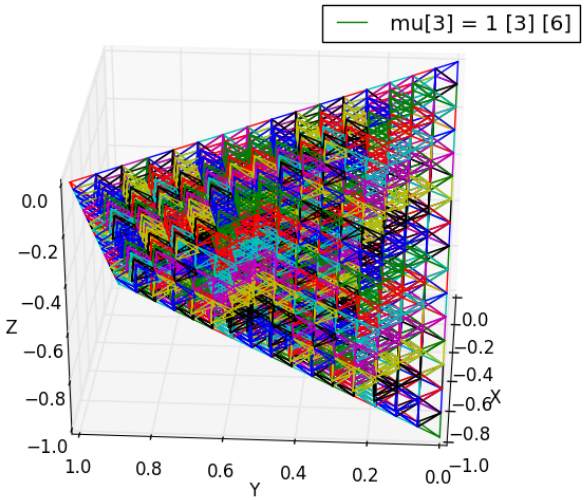
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

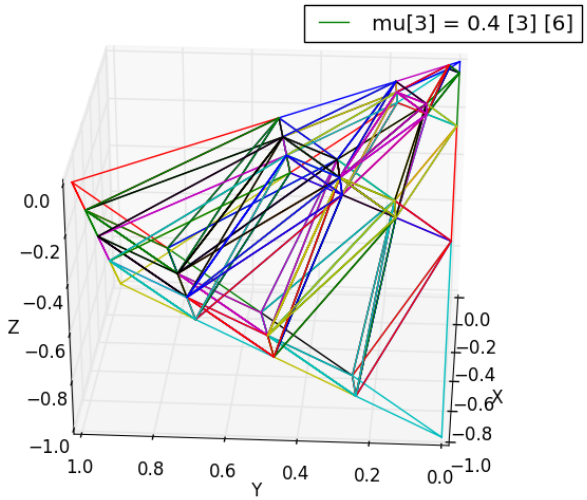
vs.



método estándar

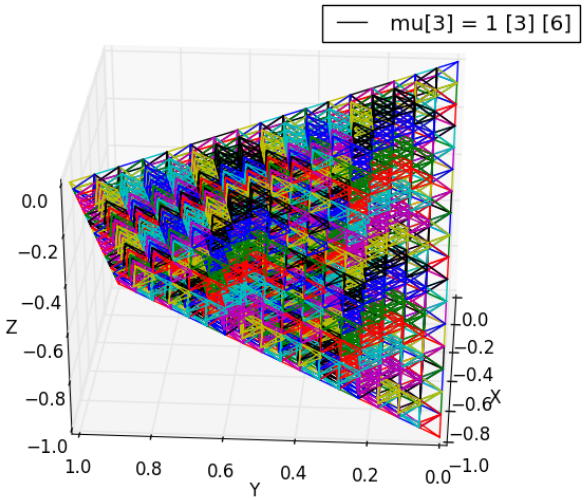
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

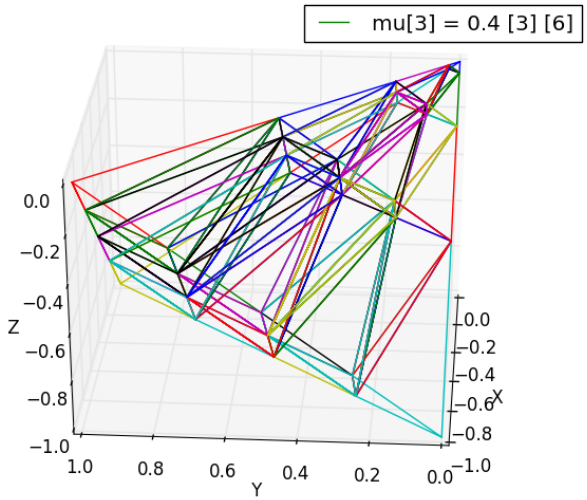
vs.



método estándar

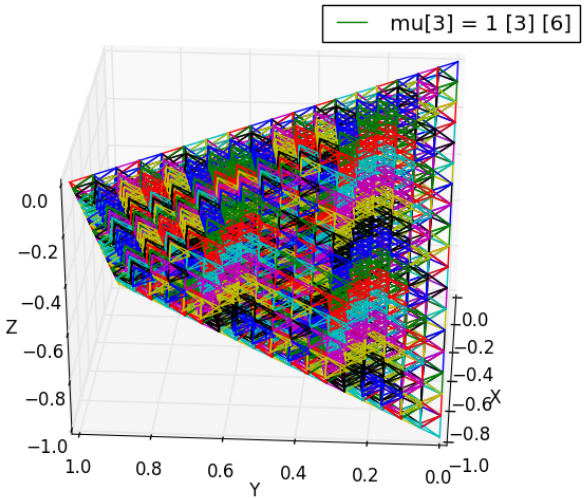
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

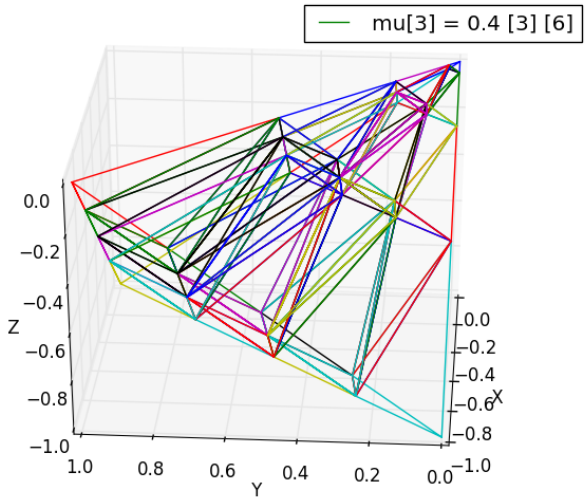
vs.



método estándar

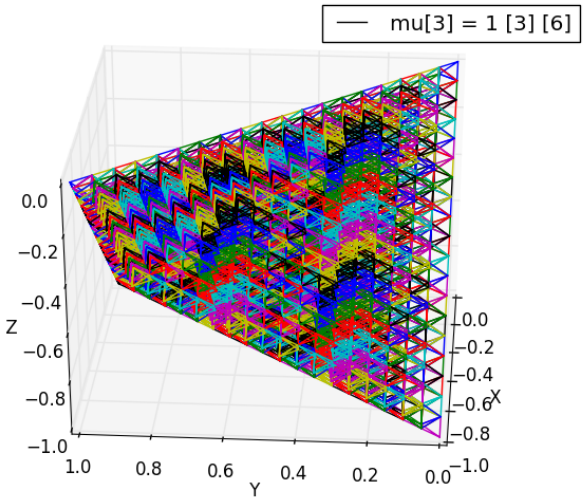
Subdominio solo con tetraedros

graduado vs. uniforme



nuestro método

vs.



método estándar

código

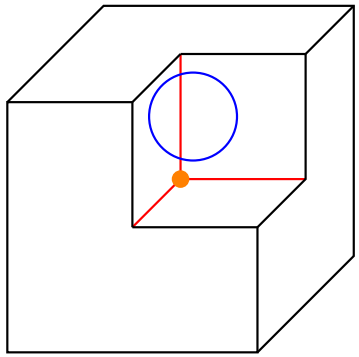
- `mesh.lambda3()`
- `mesh.macroel_sing_vrtx()`

  (coordenadas locales, graduación)

Subdominio solo con prismas

graduado vs. uniforme

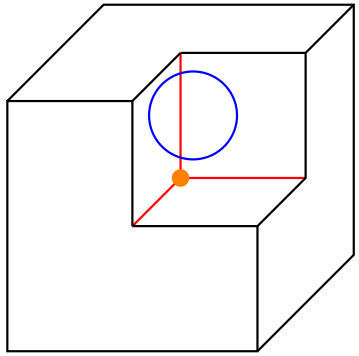
Situación:

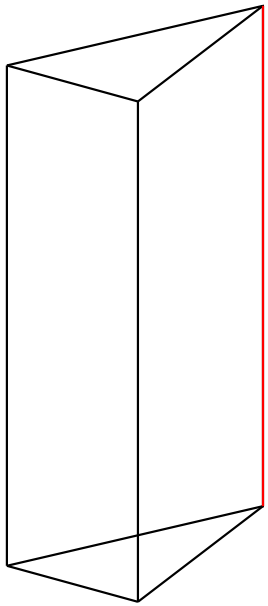


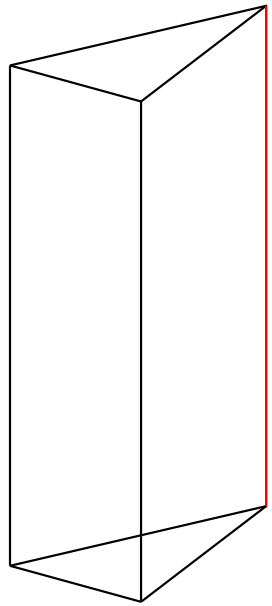
Subdominio solo con prismas

graduado vs. uniforme

Situación:





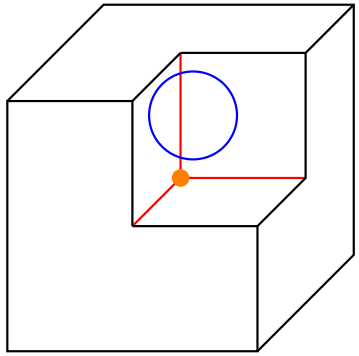


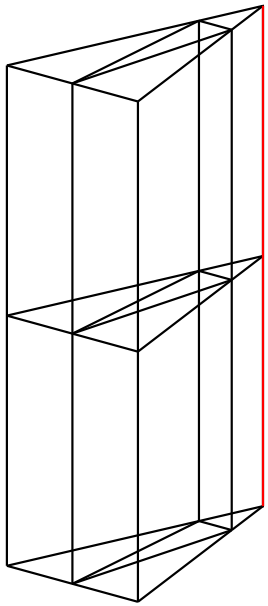
nuestro método vs. método estándar

Subdominio solo con prismas

graduado vs. uniforme

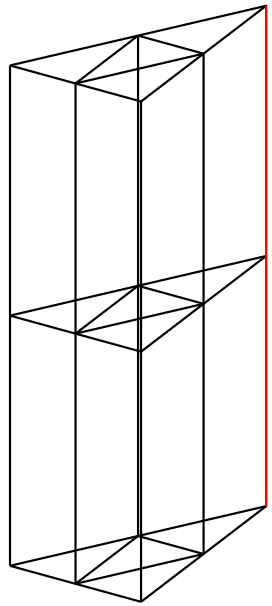
Situación:





nuestro método

vs.

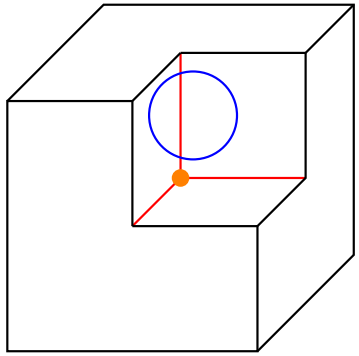


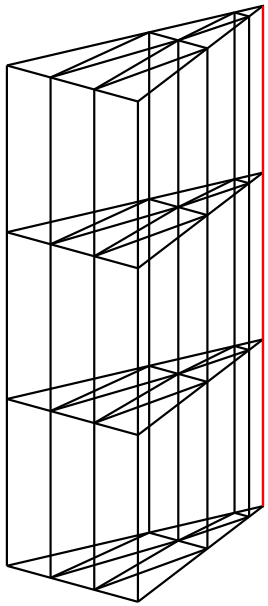
método estándar

Subdominio solo con prismas

graduado vs. uniforme

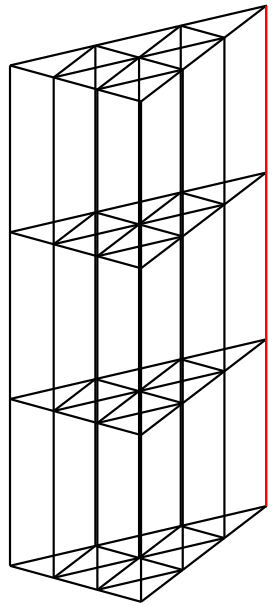
Situación:





nuestro método

vs.

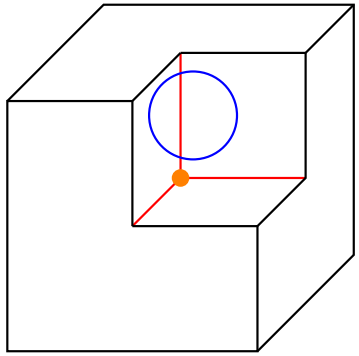


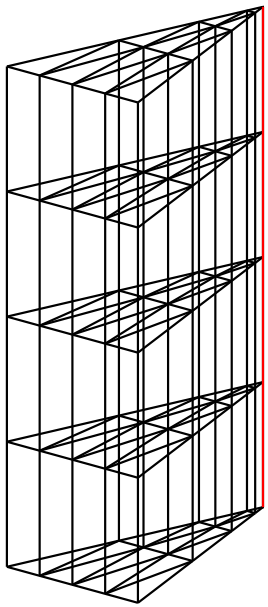
método estándar

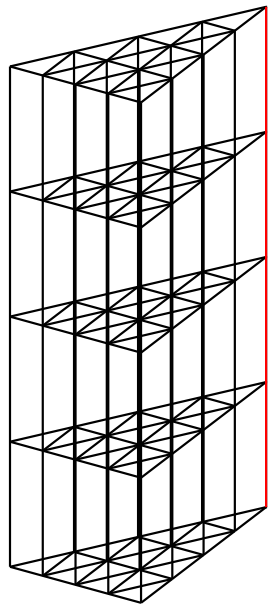
Subdominio solo con prismas

graduado vs. uniforme

Situación:





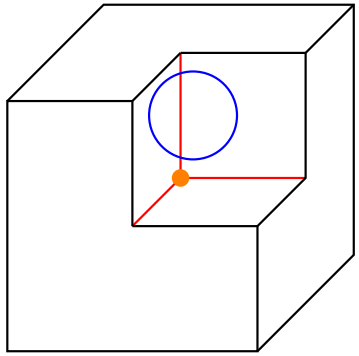


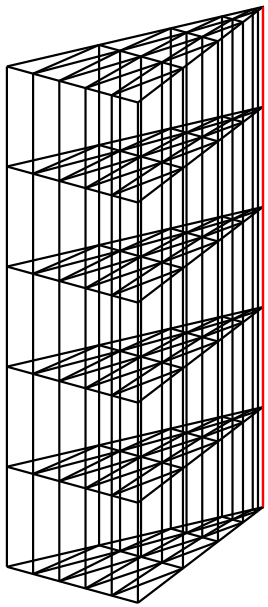
nuestro método vs. método estándar

Subdominio solo con prismas

graduado vs. uniforme

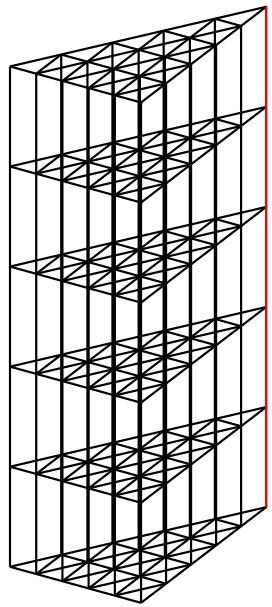
Situación:





nuestro método

vs.

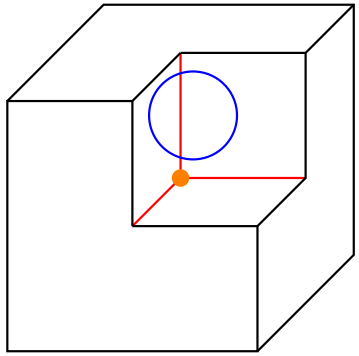


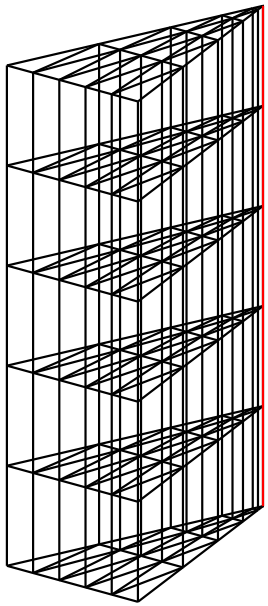
método estándar

Subdominio solo con prismas

graduado vs. uniforme

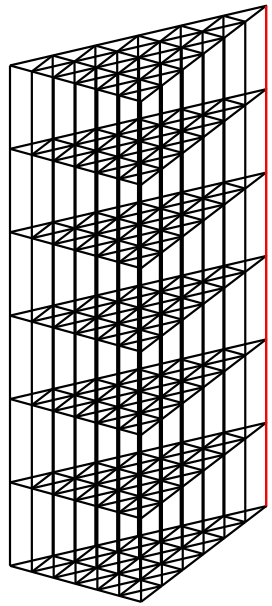
Situación:





nuestro método

vs.

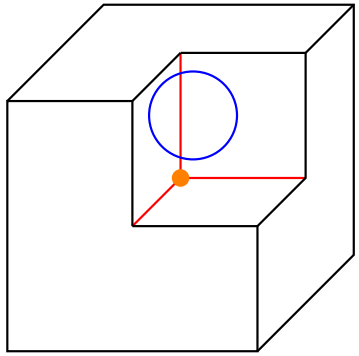


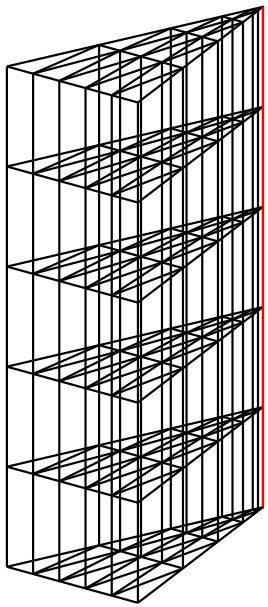
método estándar

Subdominio solo con prismas

graduado vs. uniforme

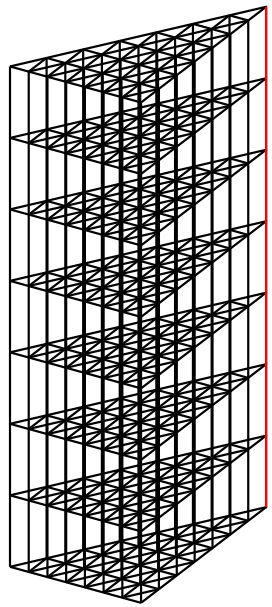
Situación:





nuestro método

vs.

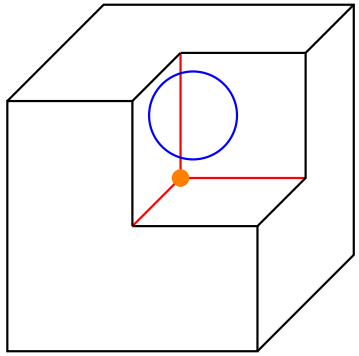


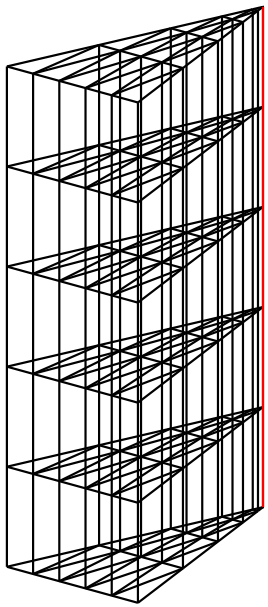
método estándar

Subdominio solo con prismas

graduado vs. uniforme

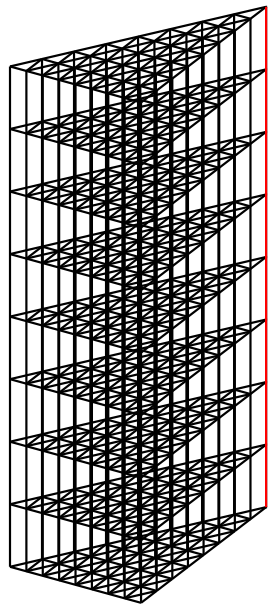
Situación:





nuestro método

vs.

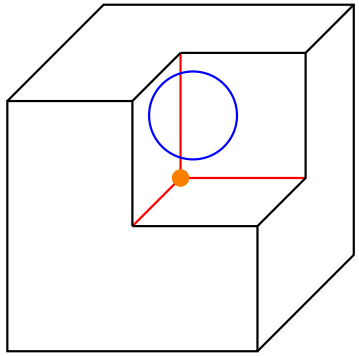


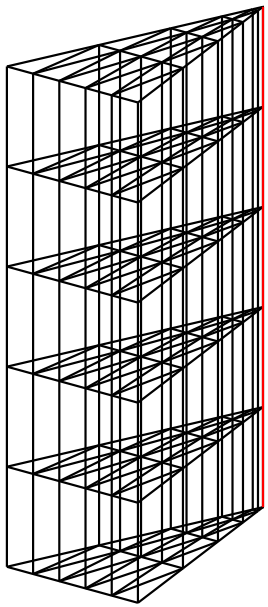
método estándar

Subdominio solo con prismas

graduado vs. uniforme

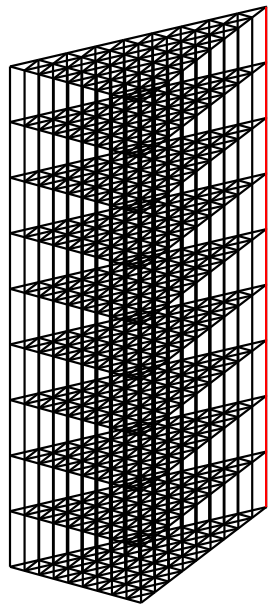
Situación:





nuestro método

vs.

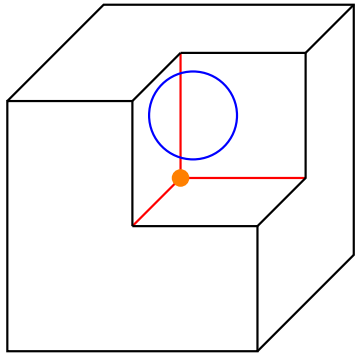


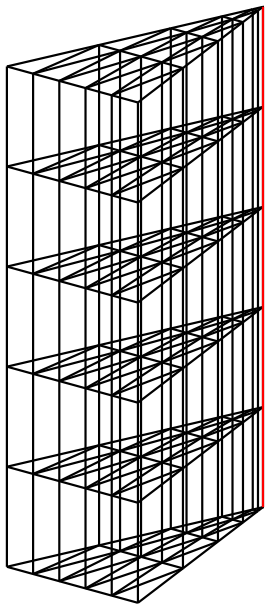
método estándar

Subdominio solo con prismas

graduado vs. uniforme

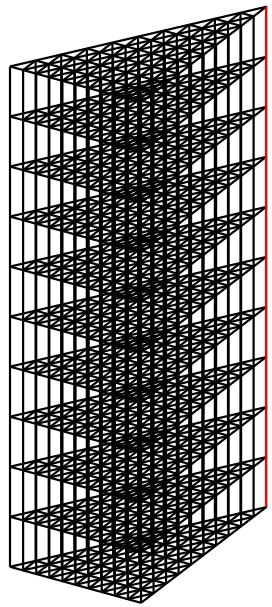
Situación:





nuestro método

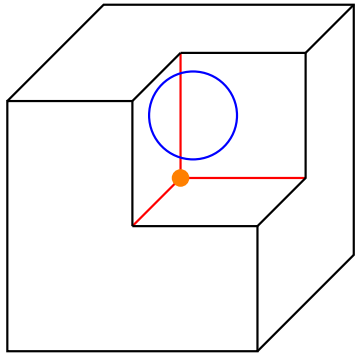
vs.

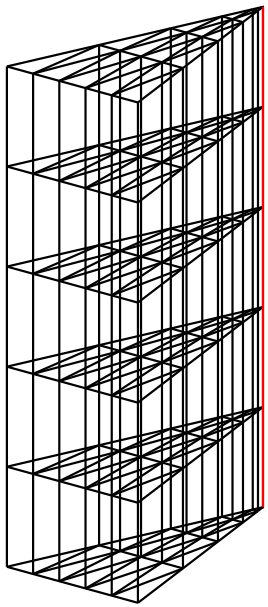


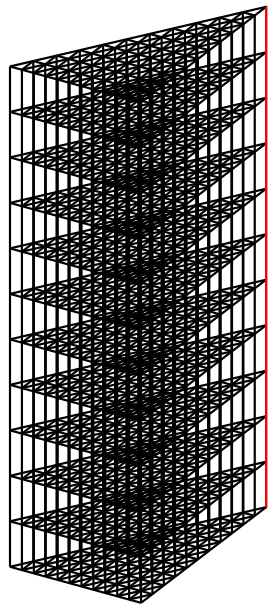
método estándar

Subdominio solo con prismas
graduado vs. uniforme

Situación:





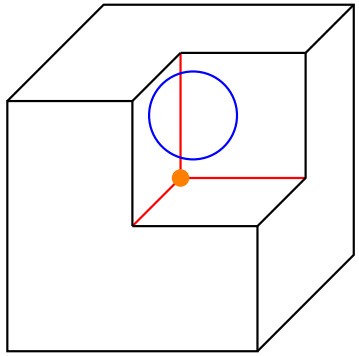


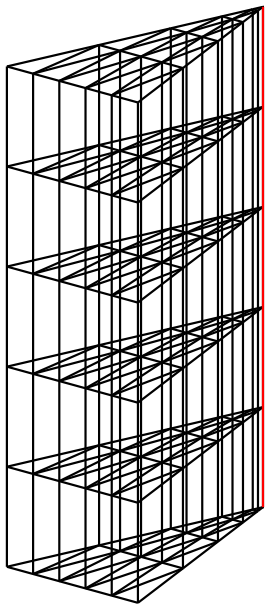
nuestro método vs. método estándar

Subdominio solo con prismas

graduado vs. uniforme

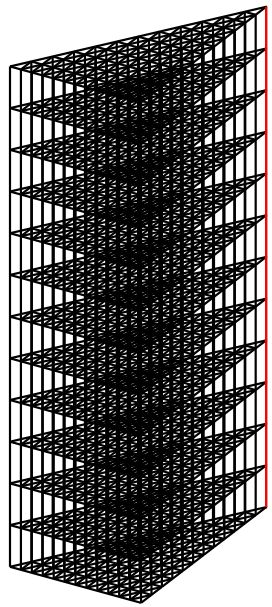
Situación:





nuestro método

vs.

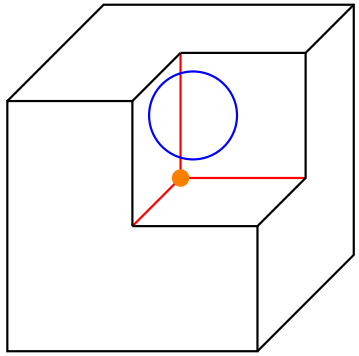


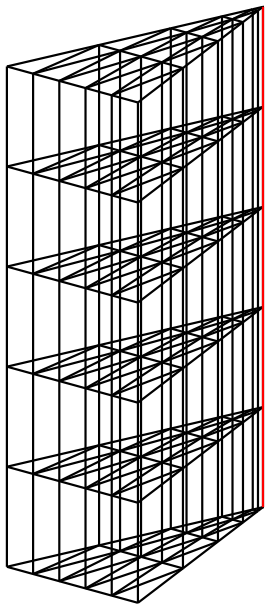
método estándar

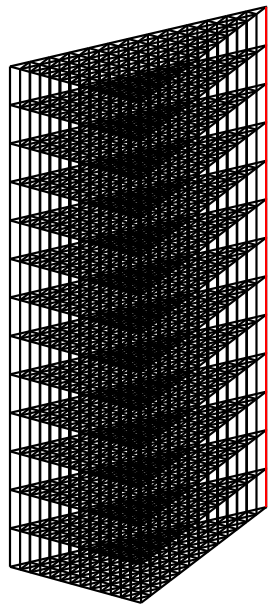
Subdominio solo con prismas

graduado vs. uniforme

Situación:





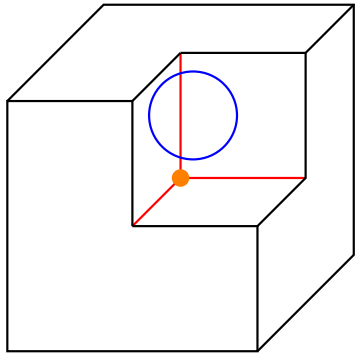


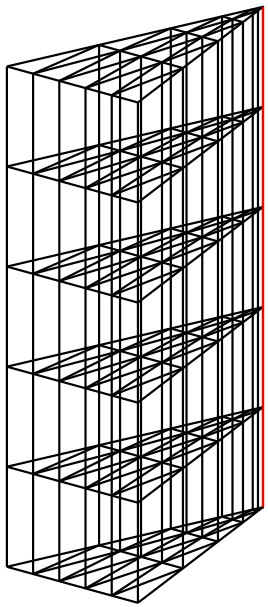
nuestro método vs. método estándar

Subdominio solo con prismas

graduado vs. uniforme

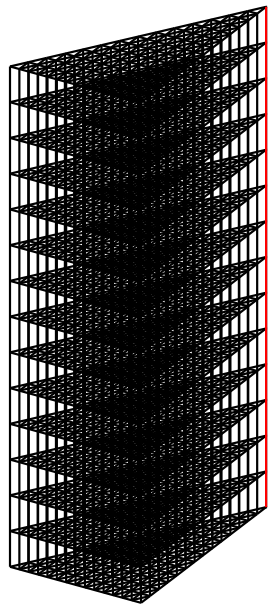
Situación:





nuestro método

vs.

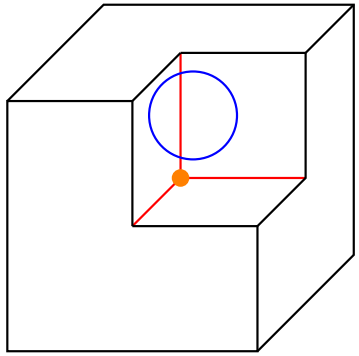


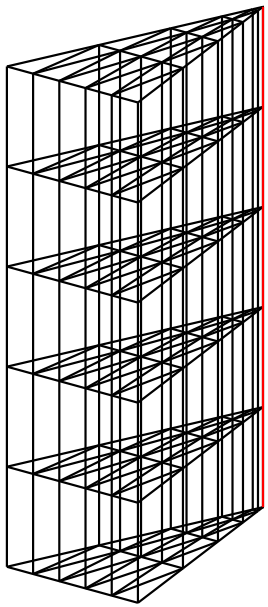
método estándar

Subdominio solo con prismas

graduado vs. uniforme

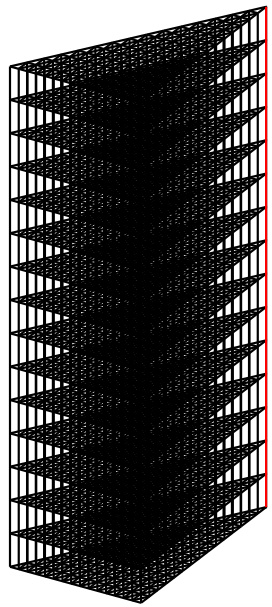
Situación:





nuestro método

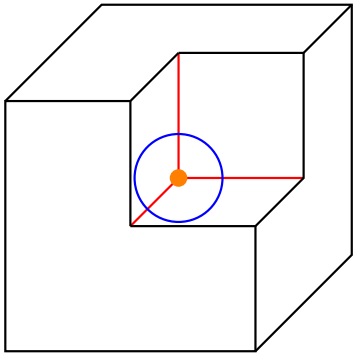
vs.



método estándar

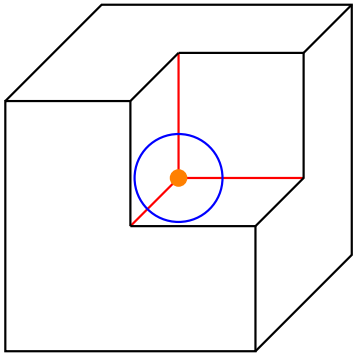
Problema mallas

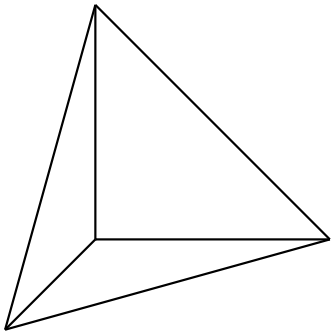
Situación:



Problema mallas

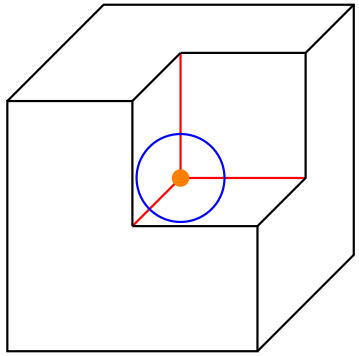
Situación:

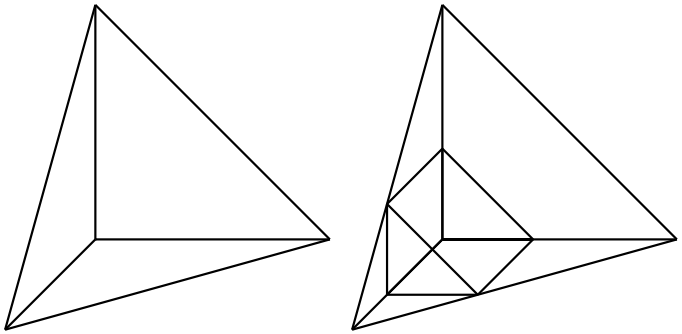




Problema mallas

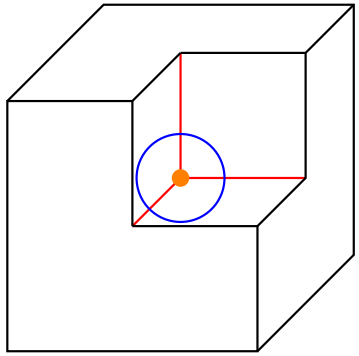
Situación:

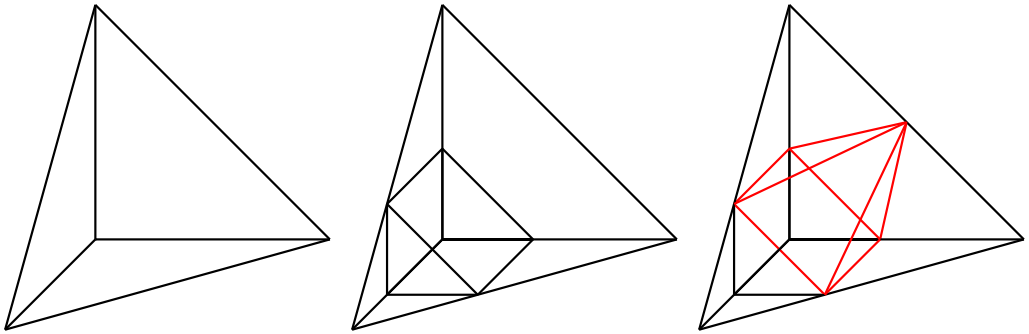




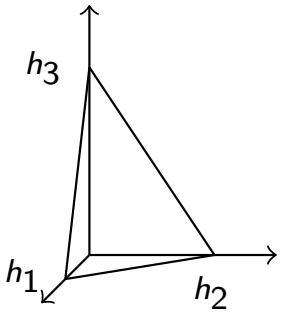
Problema mallas

Situación:

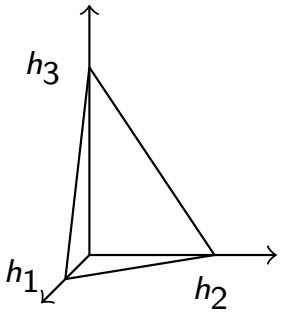


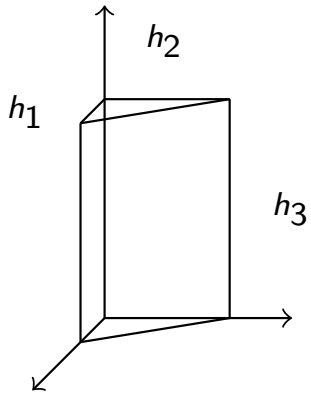


Recuerdo
Tamaños seccionales

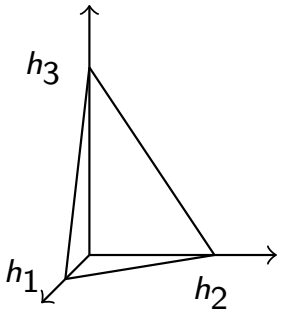


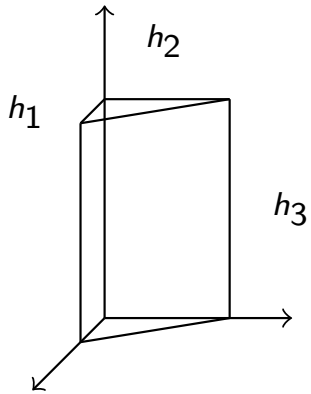
Recuerdo
Tamaños seccionales

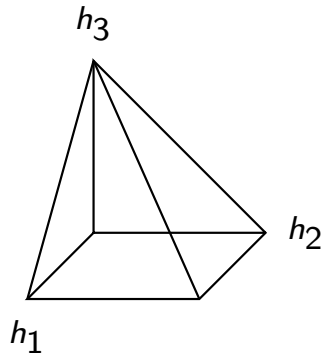




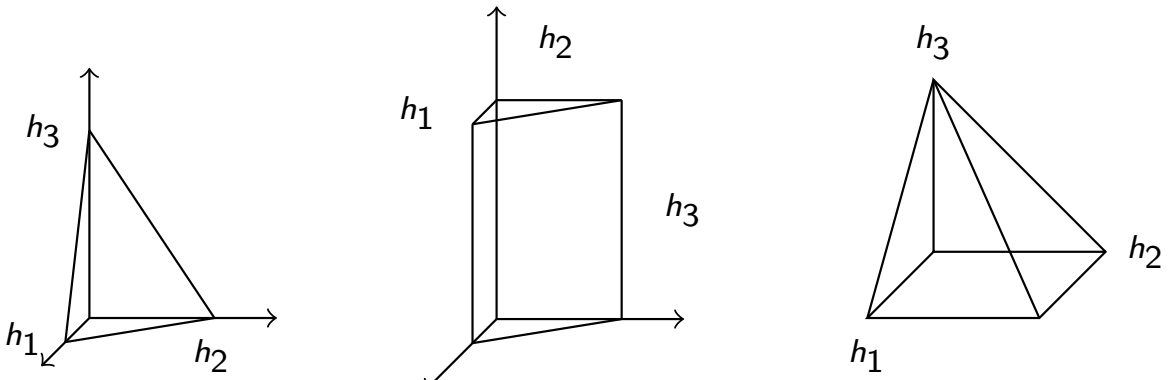
Recuerdo
Tamaños seccionales





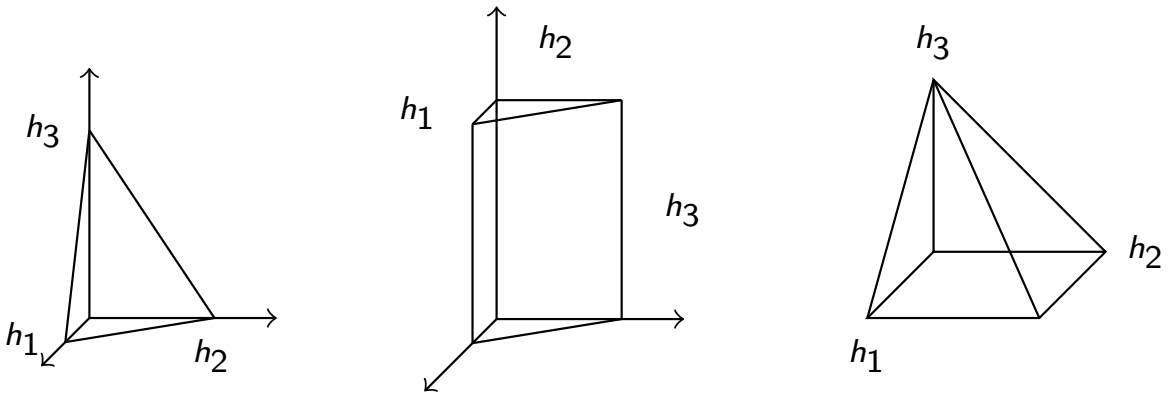


Recuerdo
Tamaños seccionales



tetraedros, prismas y pirámdes.

Recuerdo
Tamaños seccionales



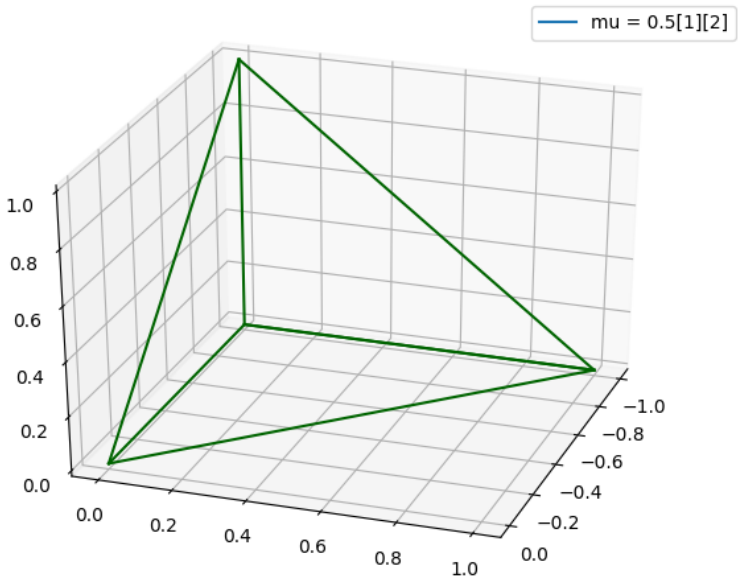
tetraedros, prismas y pirámdes.

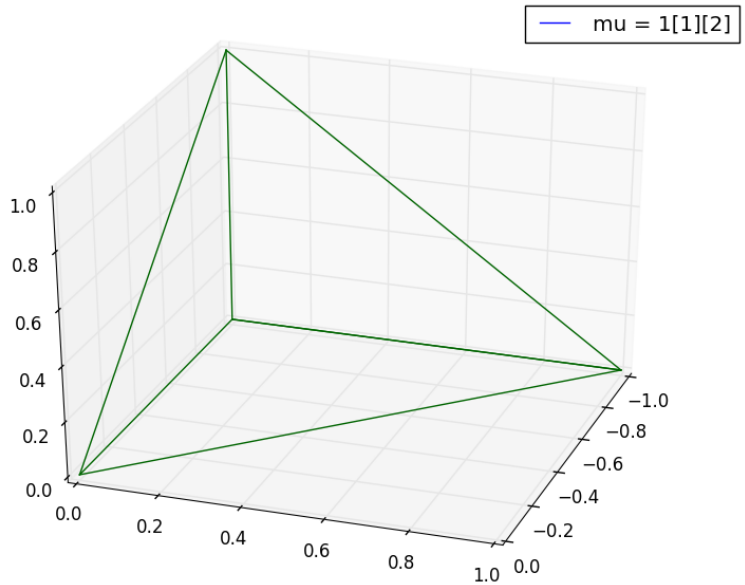
Los (h_1, h_2, h_3) son los tamaños seccionales de cada elemento de una malla.

Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 1 segmentos por arista. (M1) $1+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

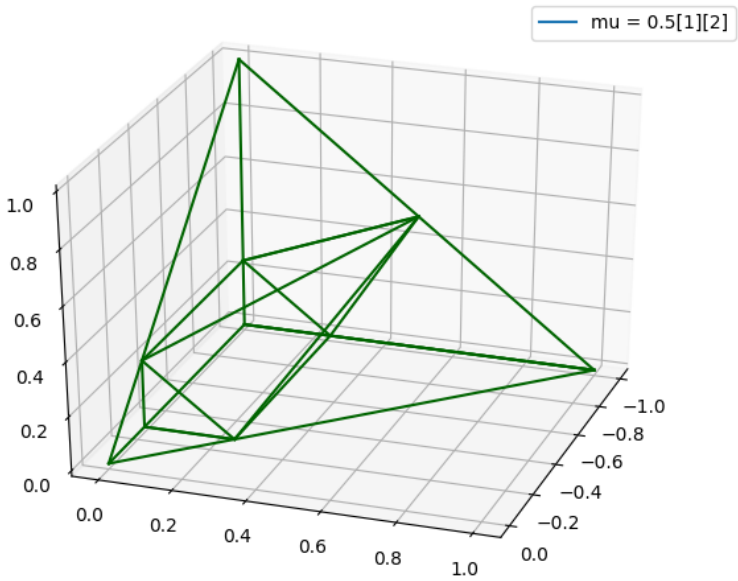


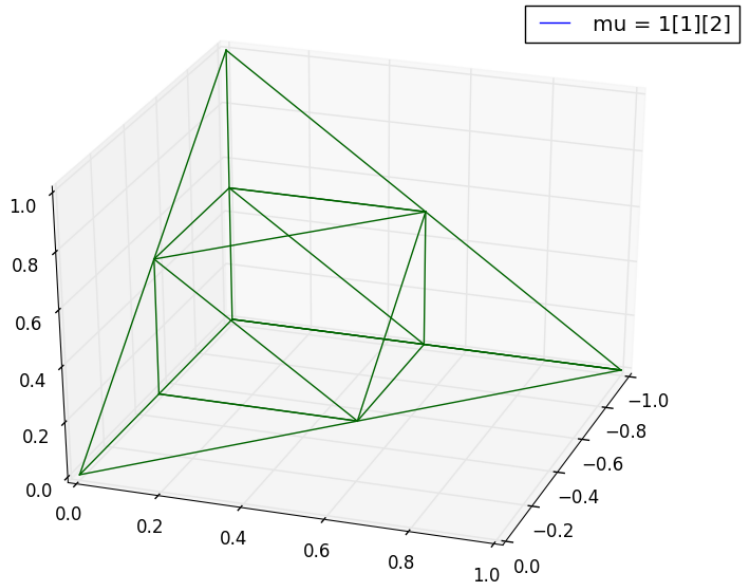


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 2 segmentos por arista. (M1) $2+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

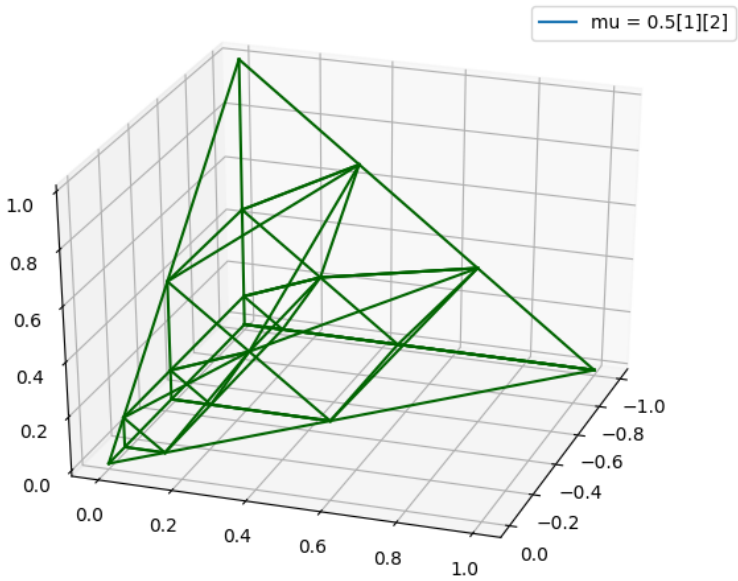


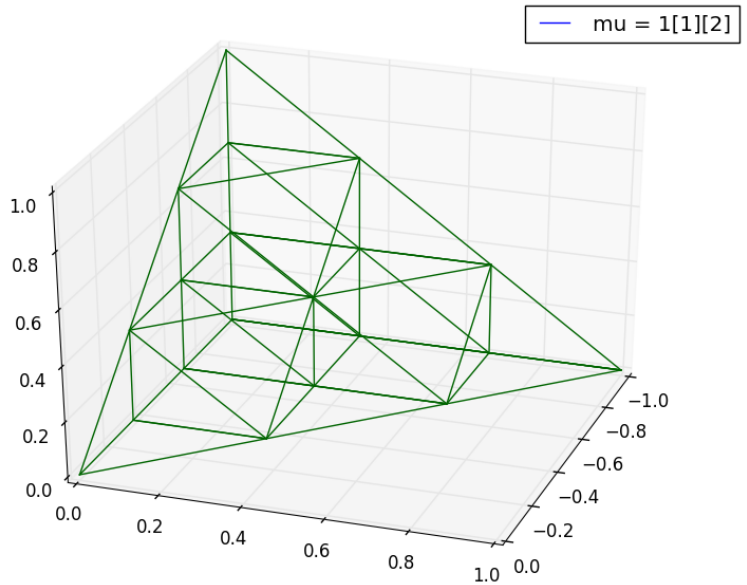


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 3 segmentos por arista. (M1) $3+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

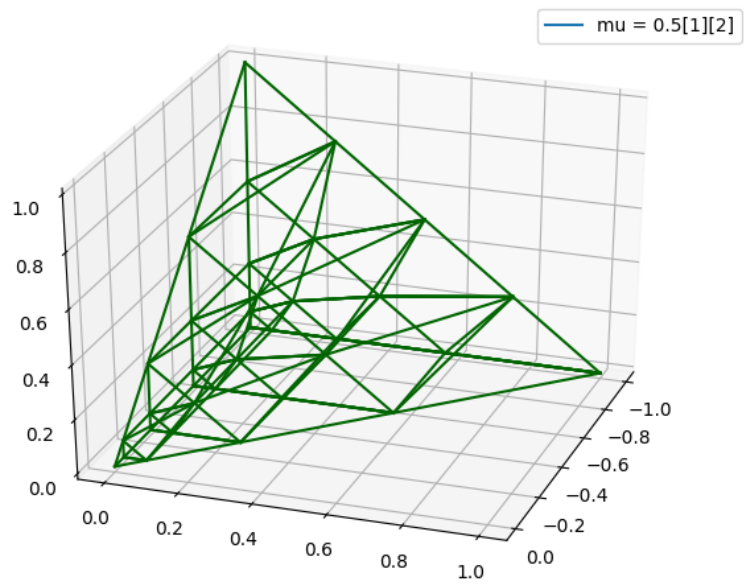


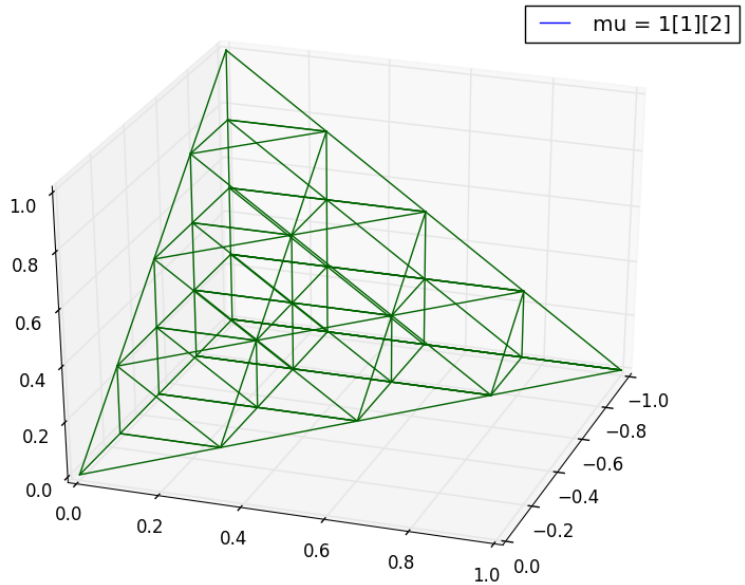


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 4 segmentos por arista. (M1) $4+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

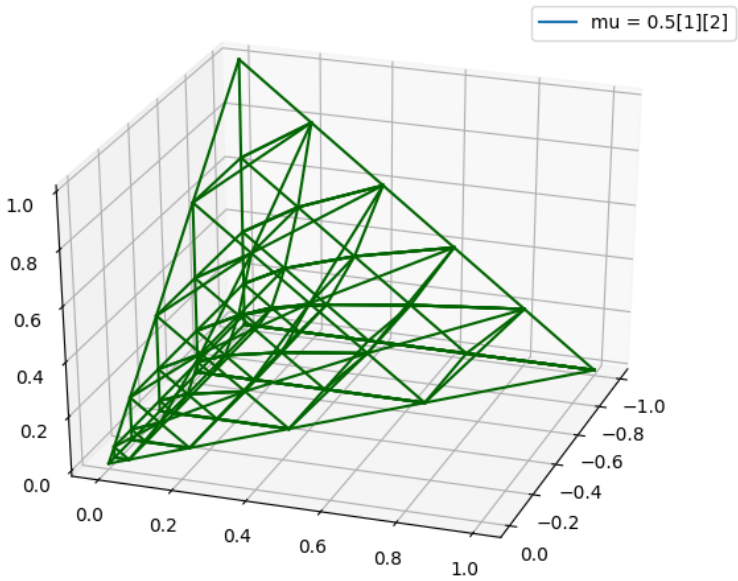


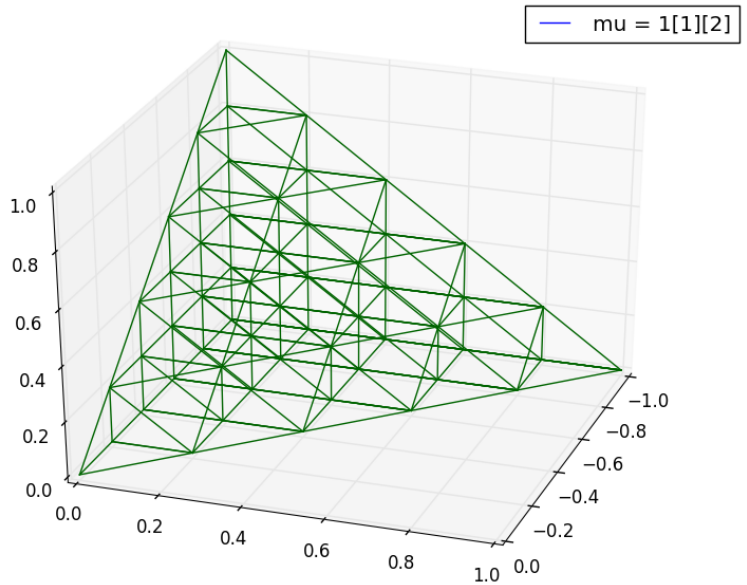


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 5 segmentos por arista. (M1) $5+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

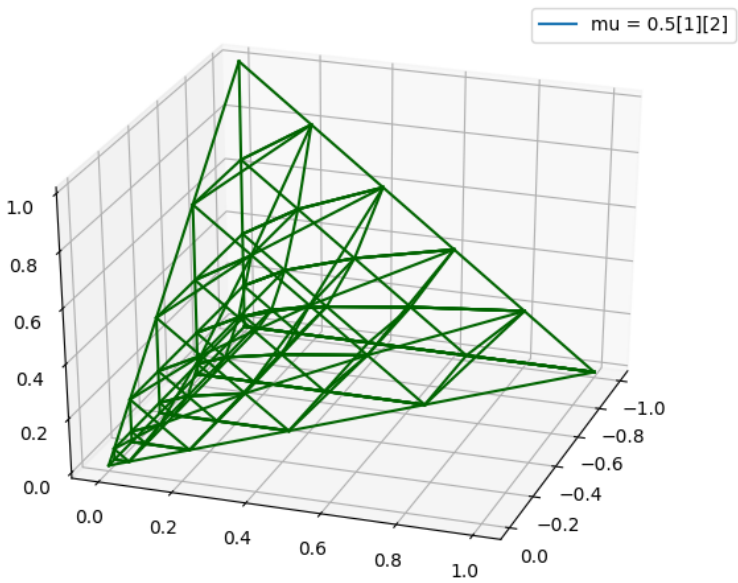


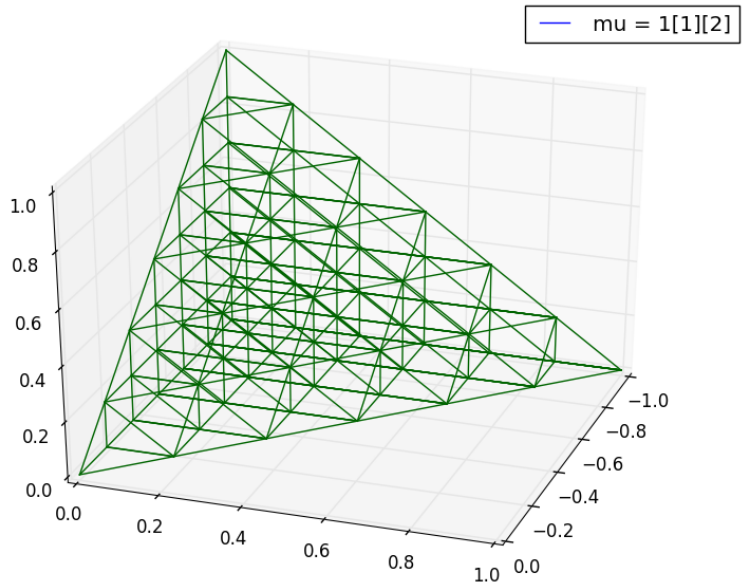


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 6 segmentos por arista. (M1) $6+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

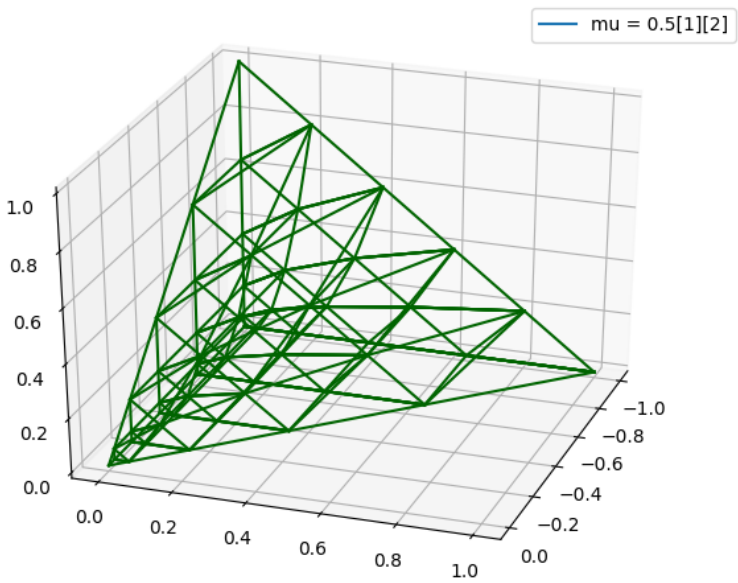


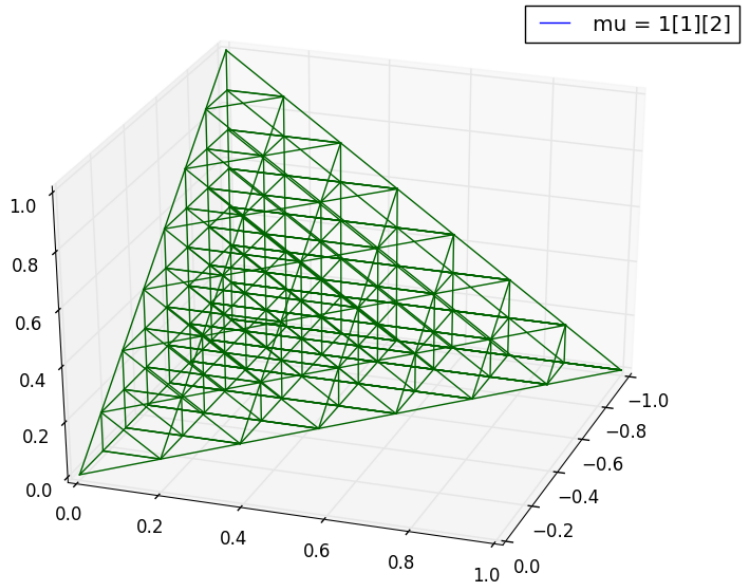


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 7 segmentos por arista. (M1) $7+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

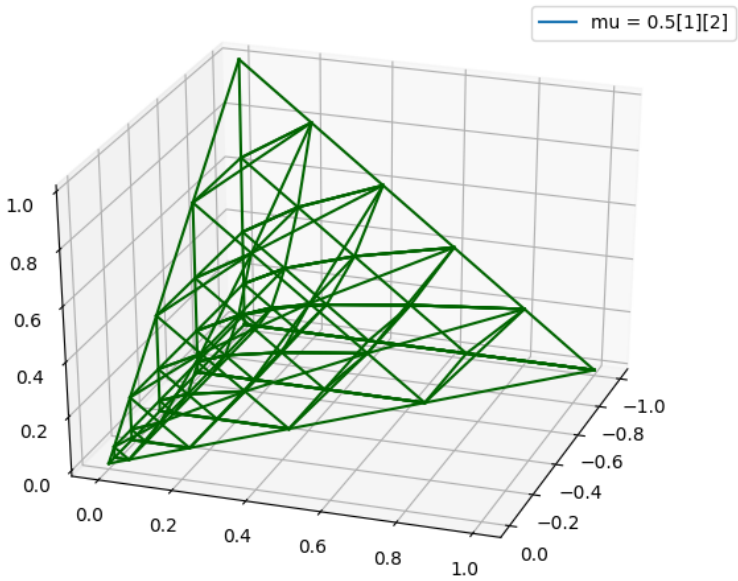


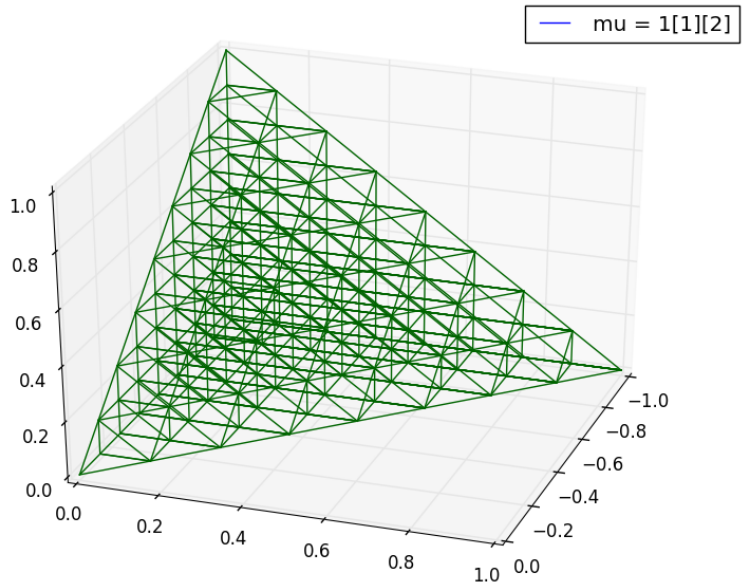


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 8 segmentos por arista. (M1) $8+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

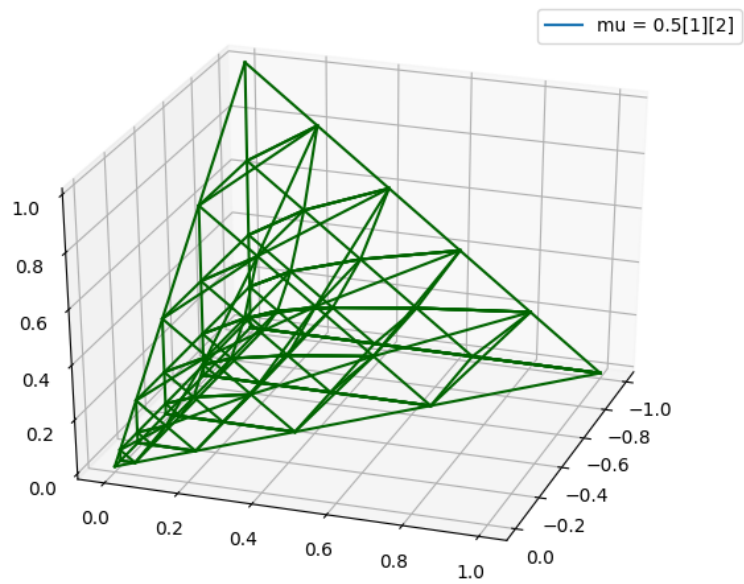


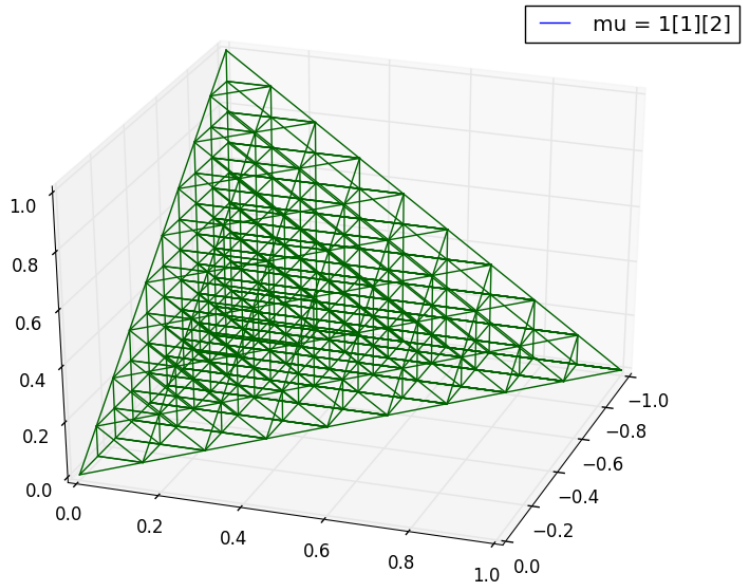


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 9 segmentos por arista. (M1) $9+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

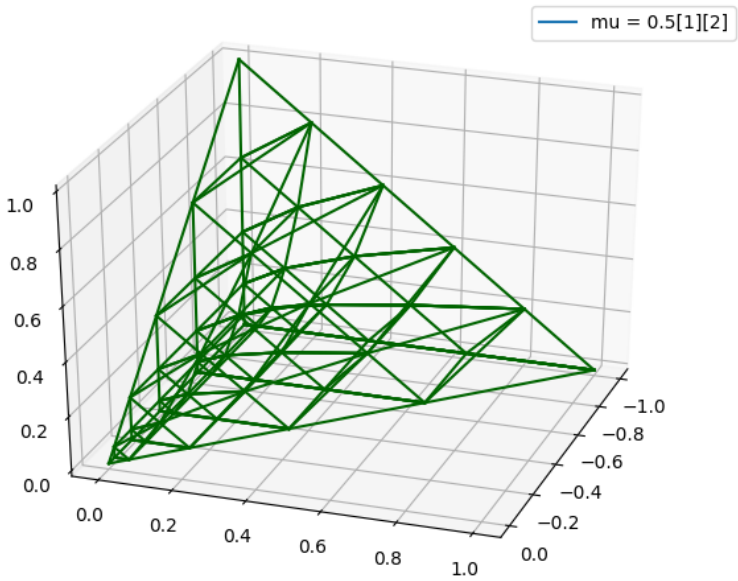


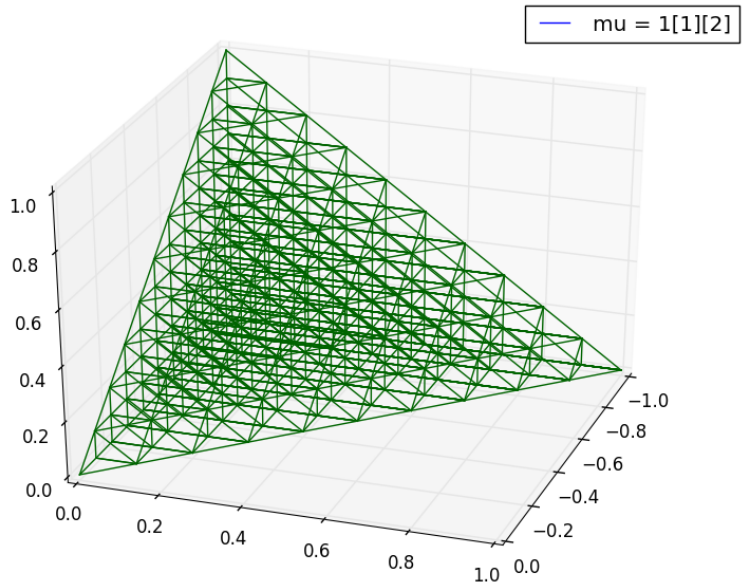


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 10 segmentos por arista. (M1) $10+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

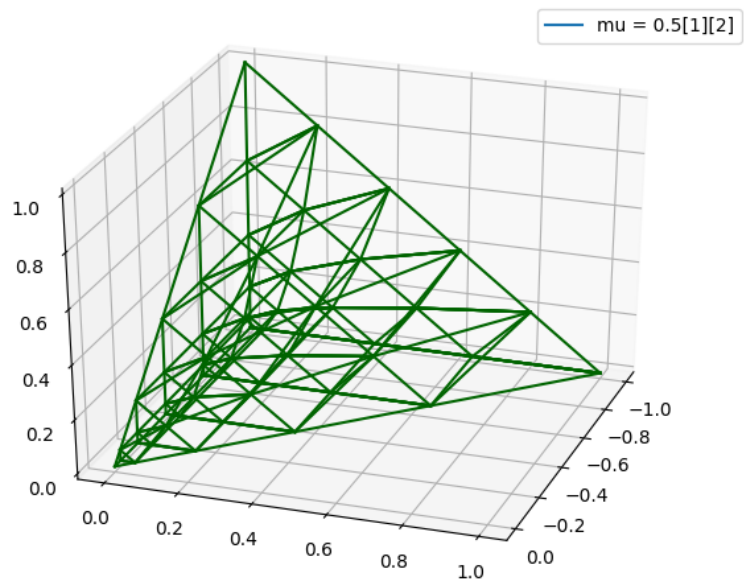


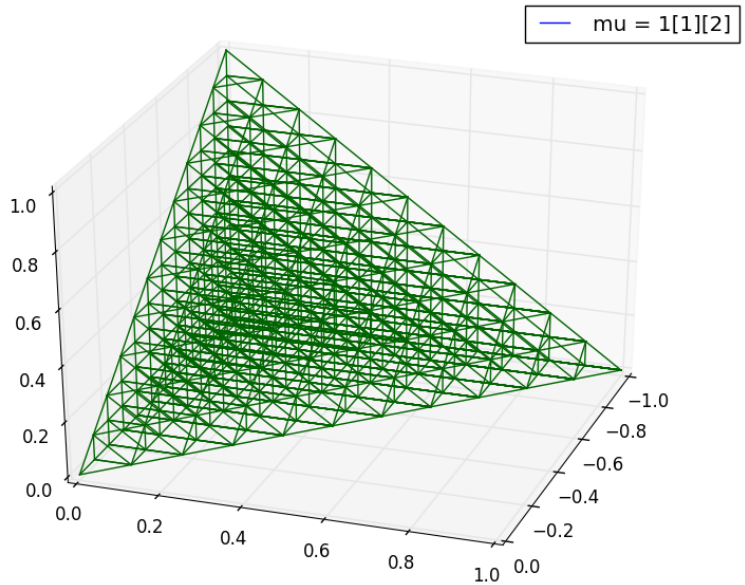


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 11 segmentos por arista. (M1) $11+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

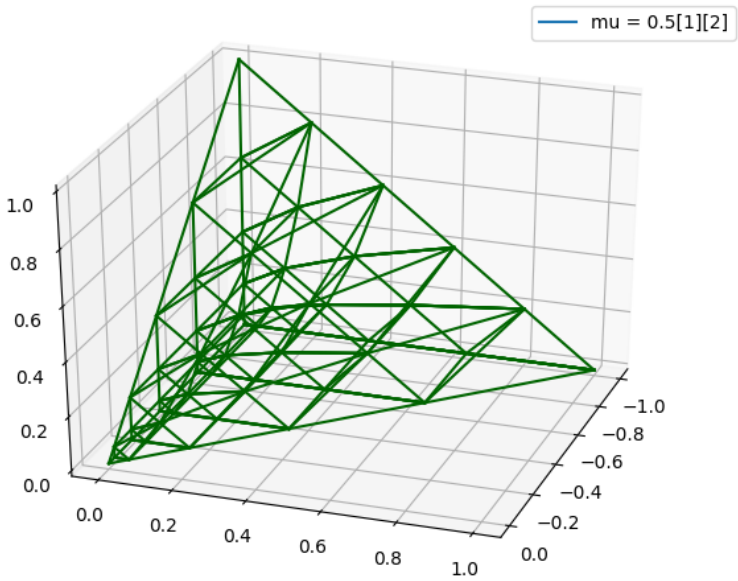


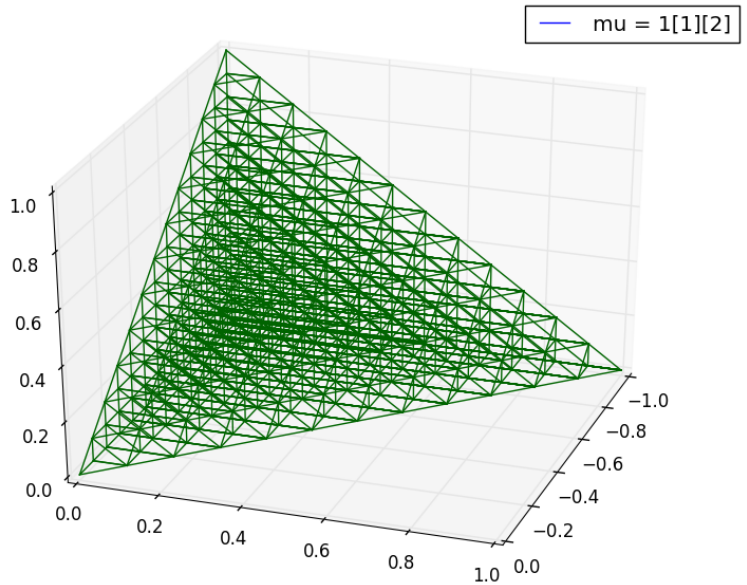


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 12 segmentos por arista. (M1) $12+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

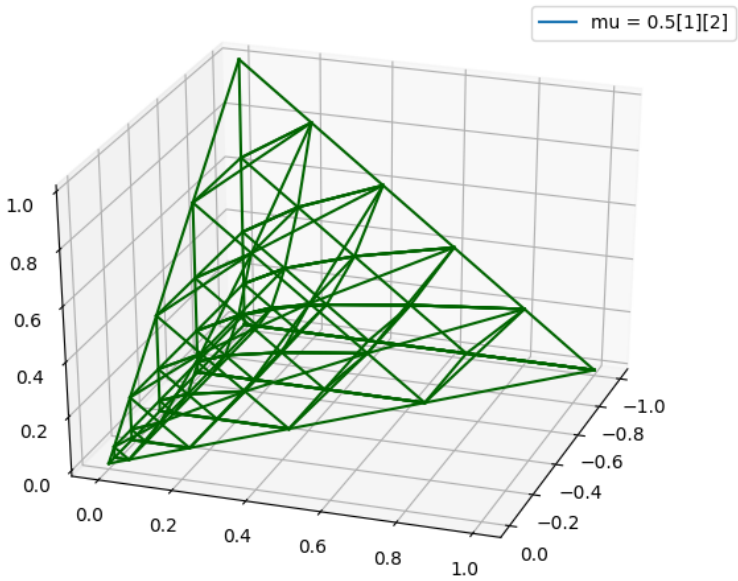


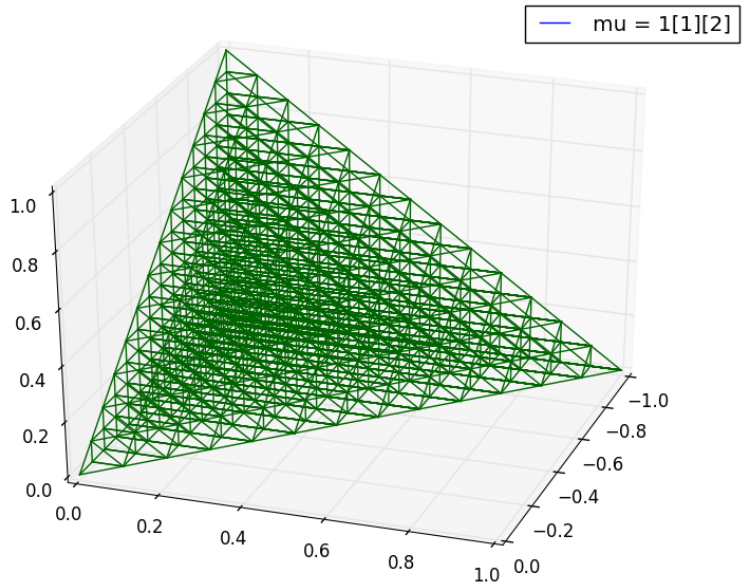


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 13 segmentos por arista. (M1) $13+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

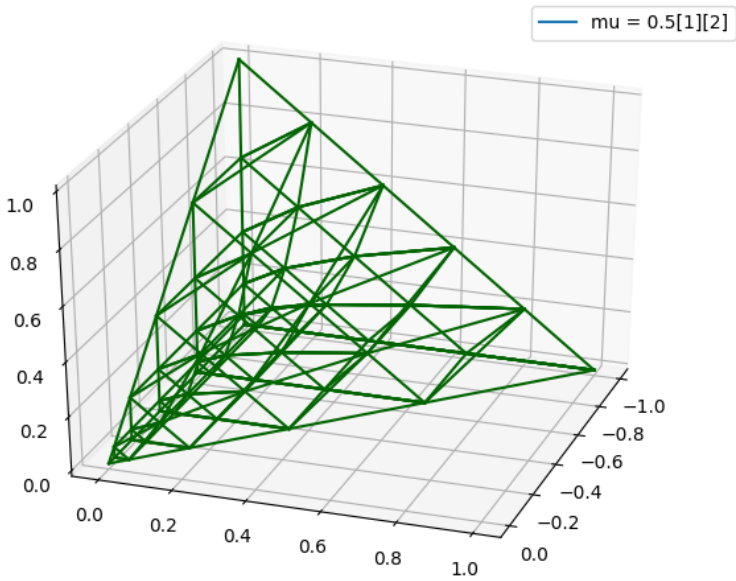


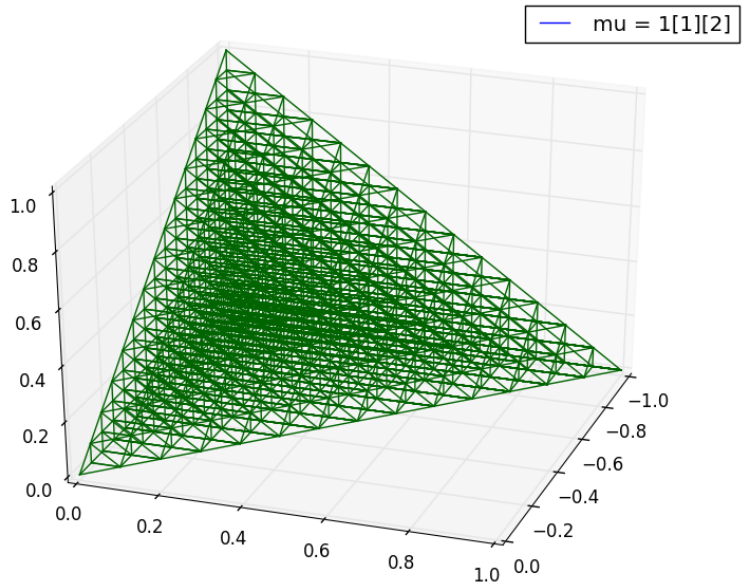


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 14 segmentos por arista. (M1) $14+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

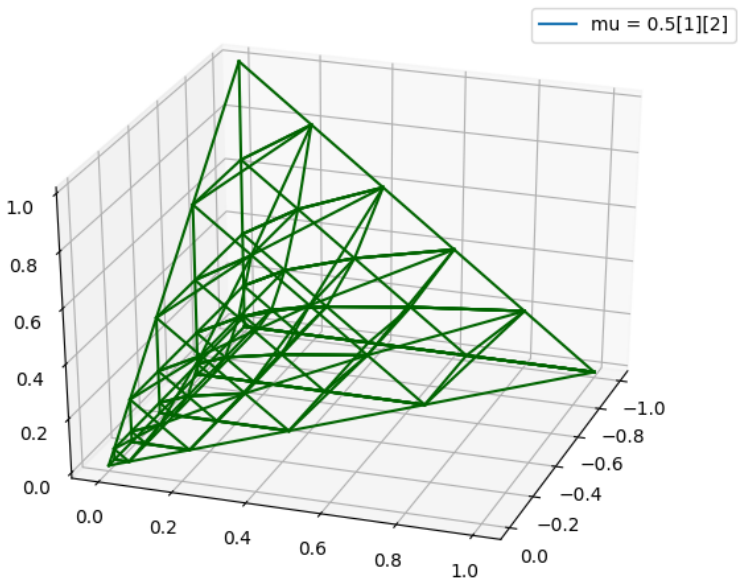


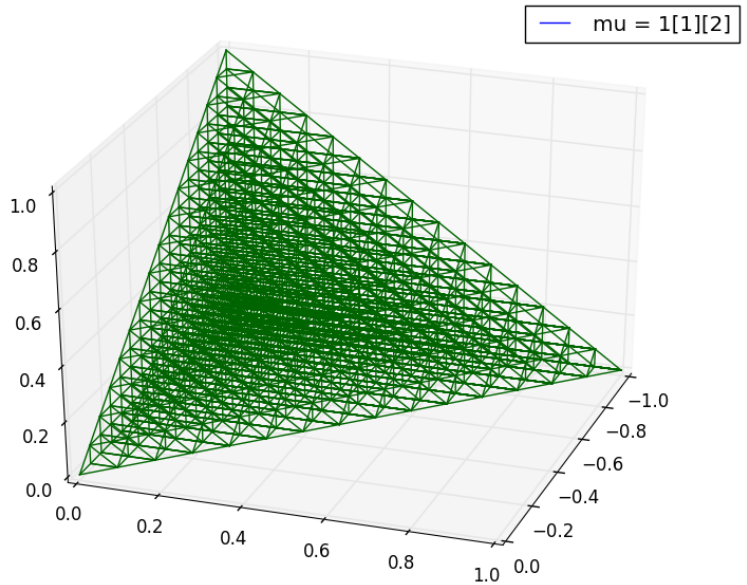


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 15 segmentos por arista. (M1) $15+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

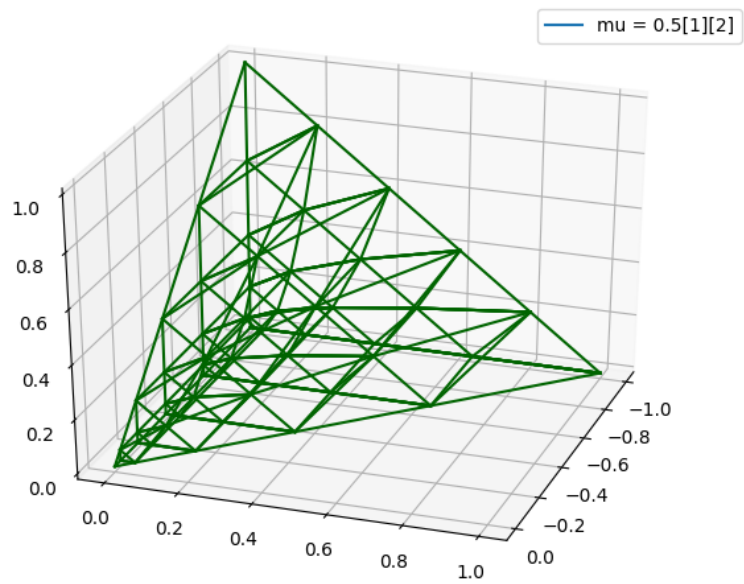


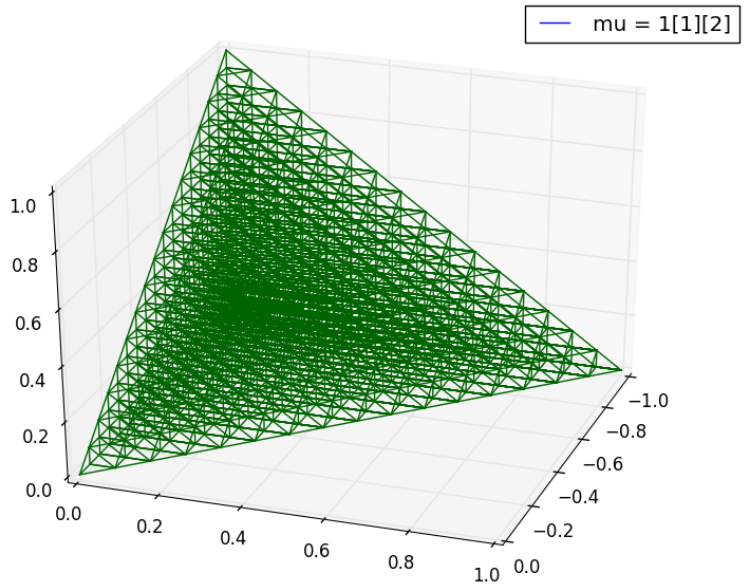


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 16 segmentos por arista. (M1) $16+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

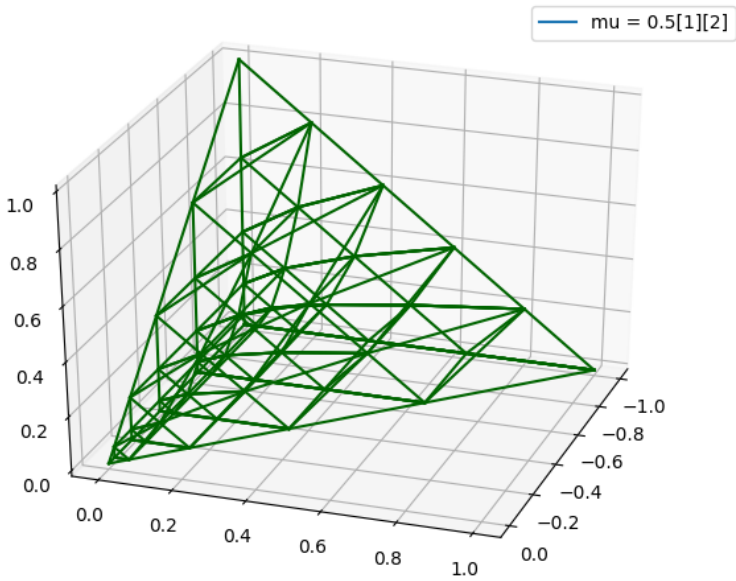


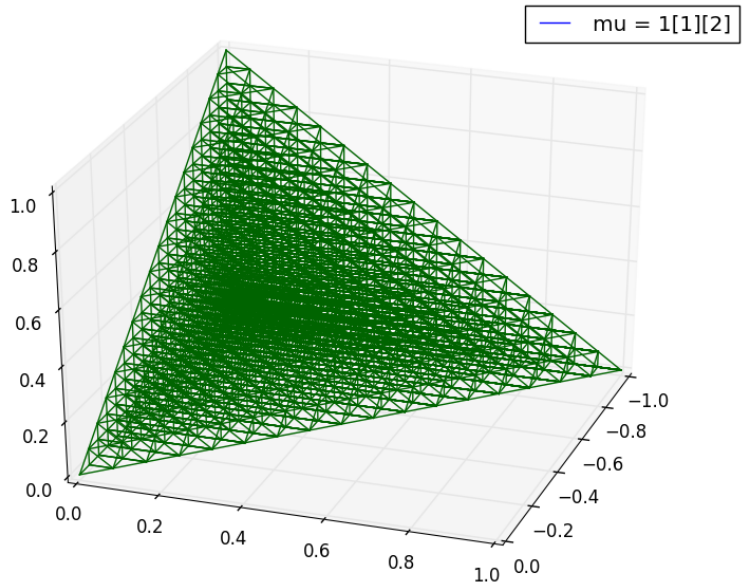


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 17 segmentos por arista. (M1) $17+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

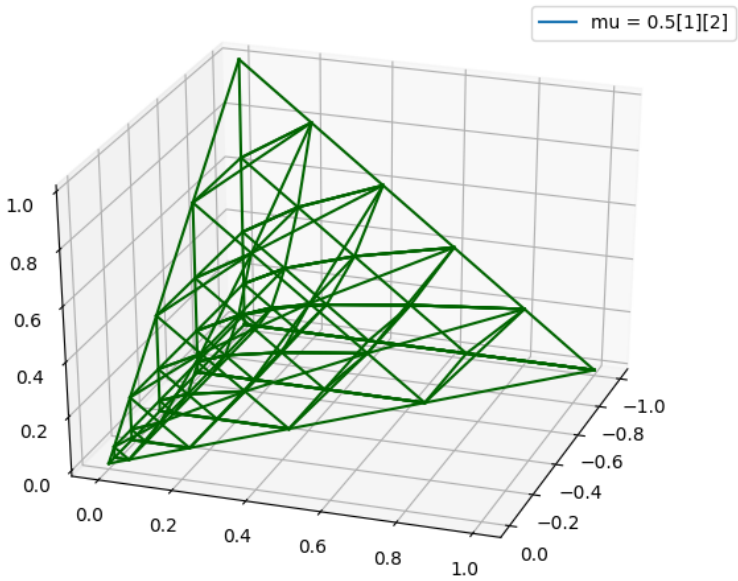


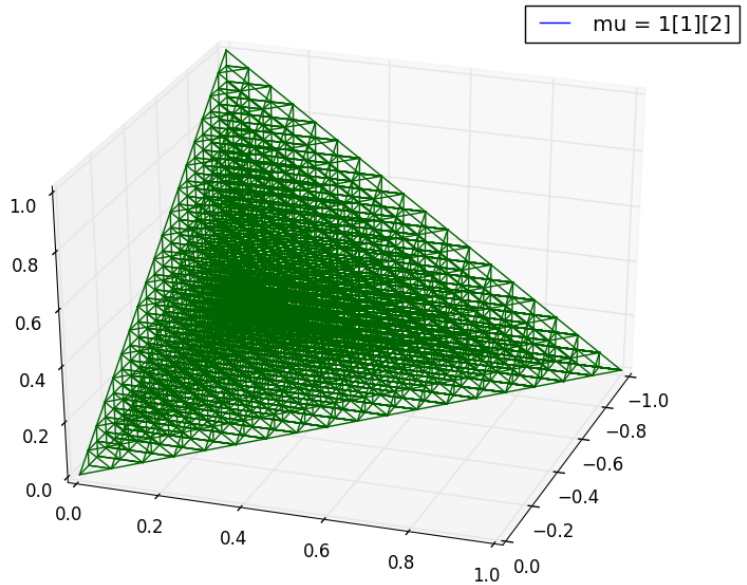


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 18 segmentos por arista. (M1) $18+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

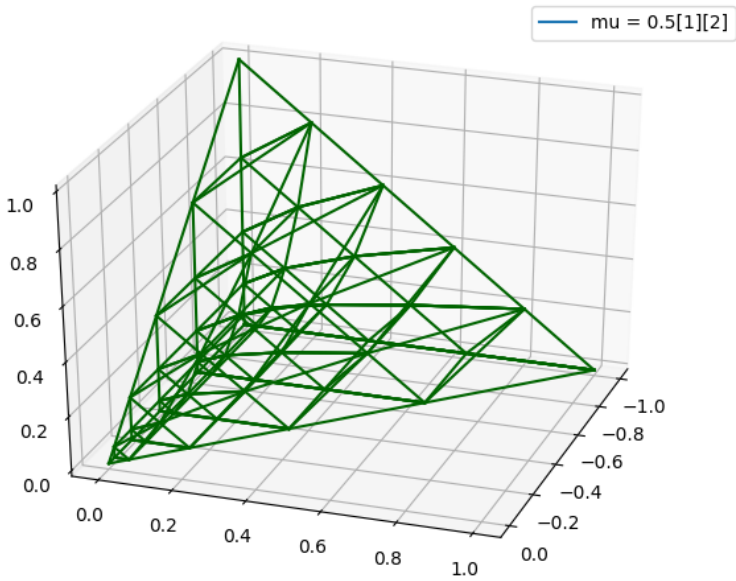


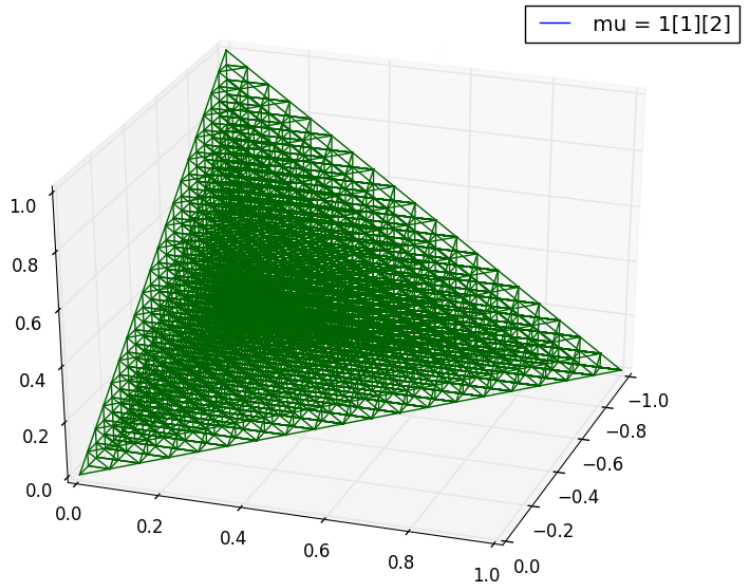


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 19 segmentos por arista. (M1) $19+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

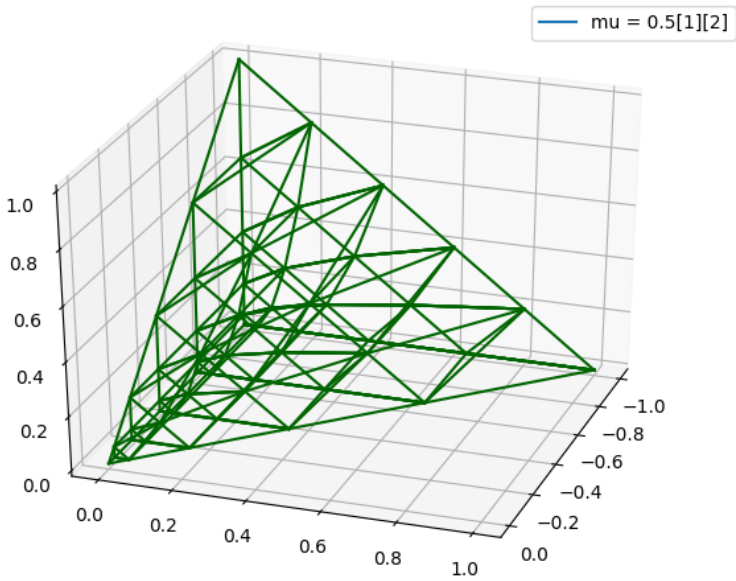


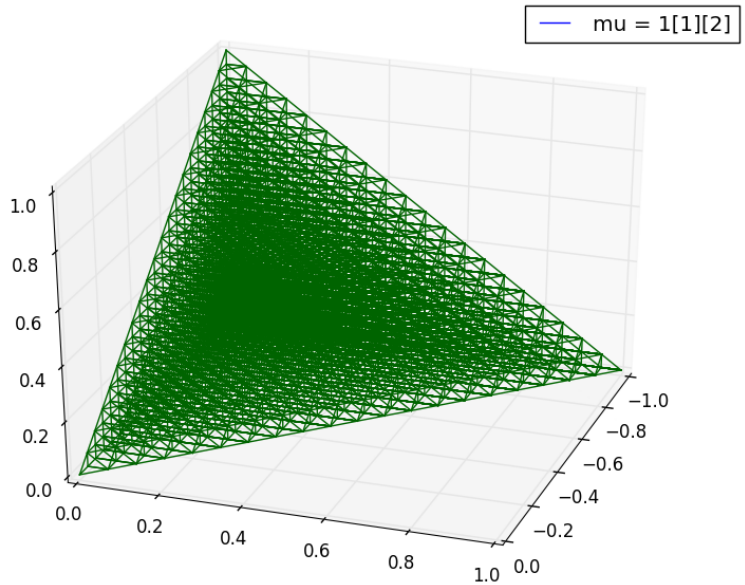


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 20 segmentos por arista. (M1) $20+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

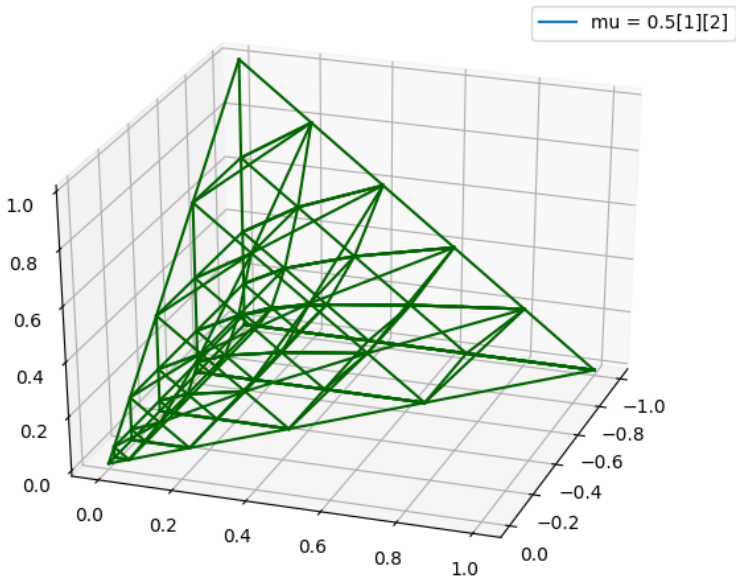


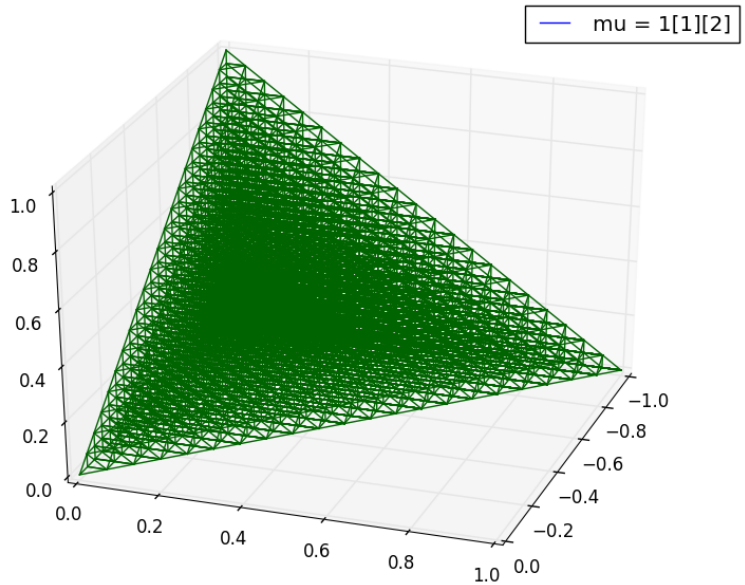


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 21 segmentos por arista. (M1) $21+1$ nodos por arista.

$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$

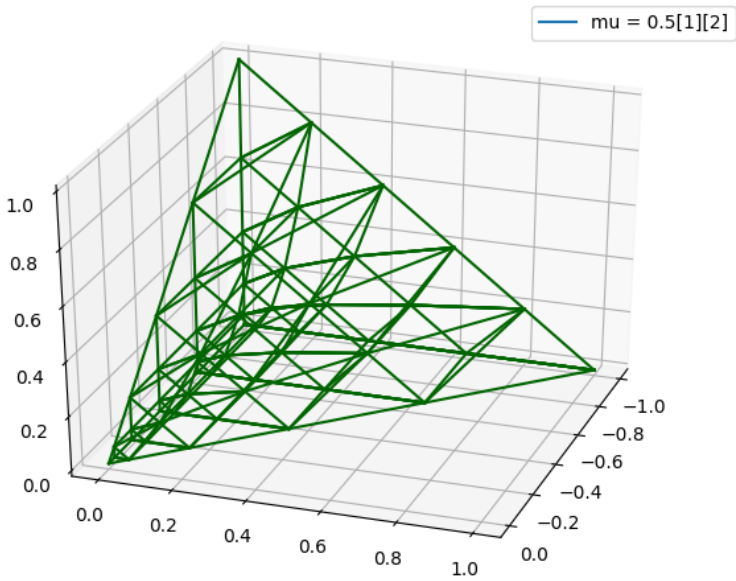


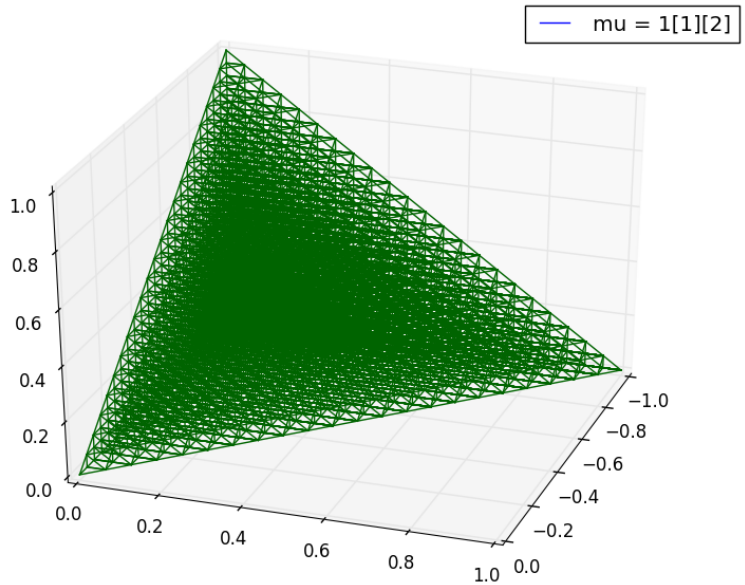


Meshing algorithm
un macro–elemento tetr. I aislado

(M1)' 22 segmentos por arista. (M1) $22+1$ nodos por arista.

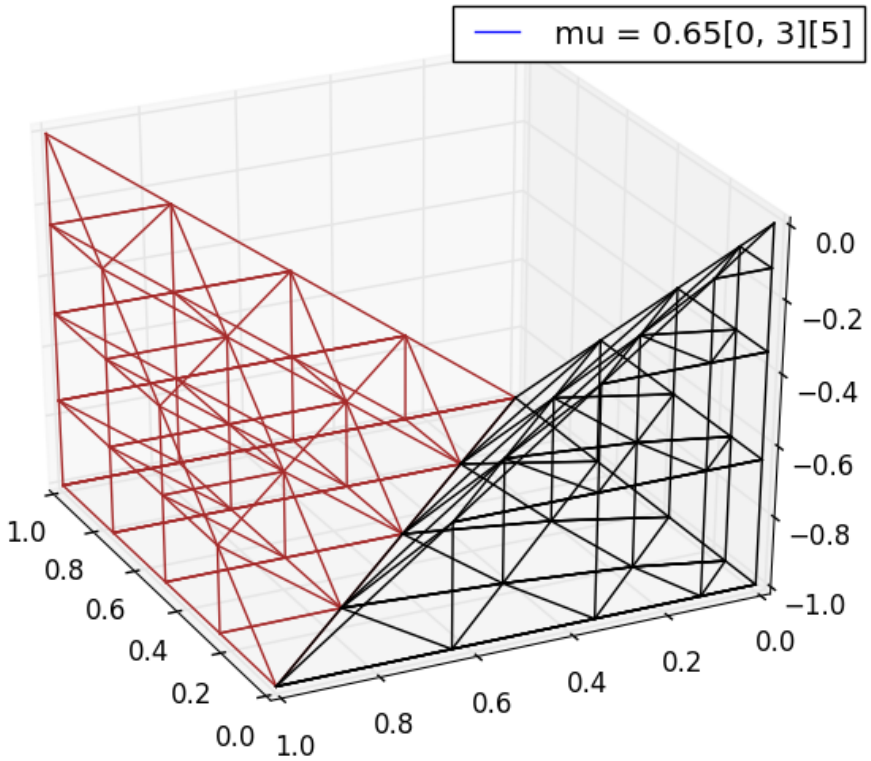
$h_i = h_i(h, \beta, E), \quad 1 \leqslant i \leqslant 3.$





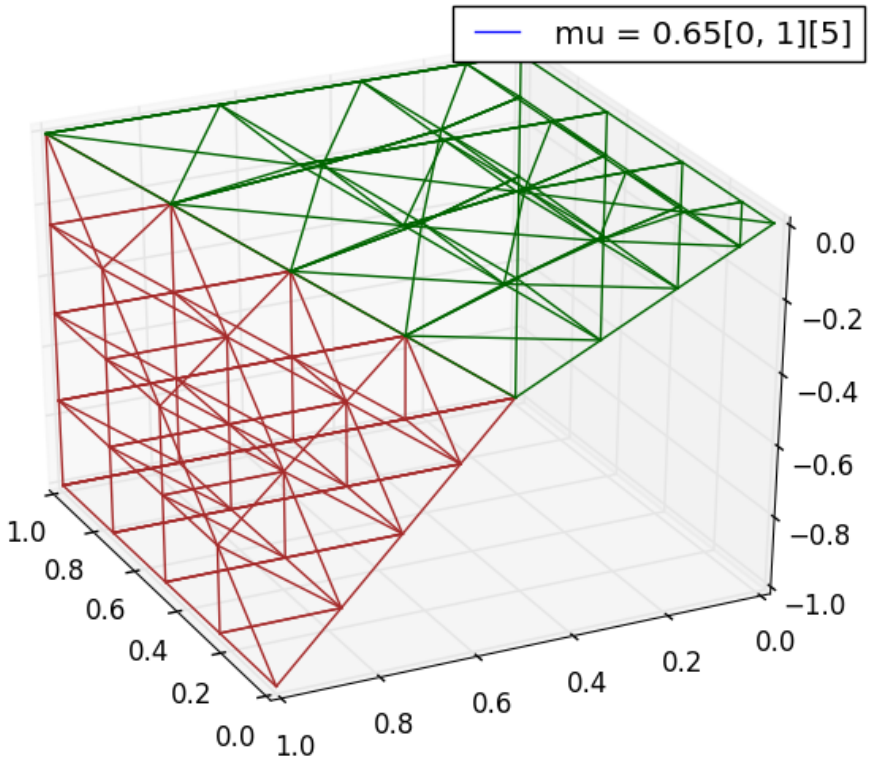
Ver `example1.py` (prismas contra la arista singular).

conformidad
por aristas



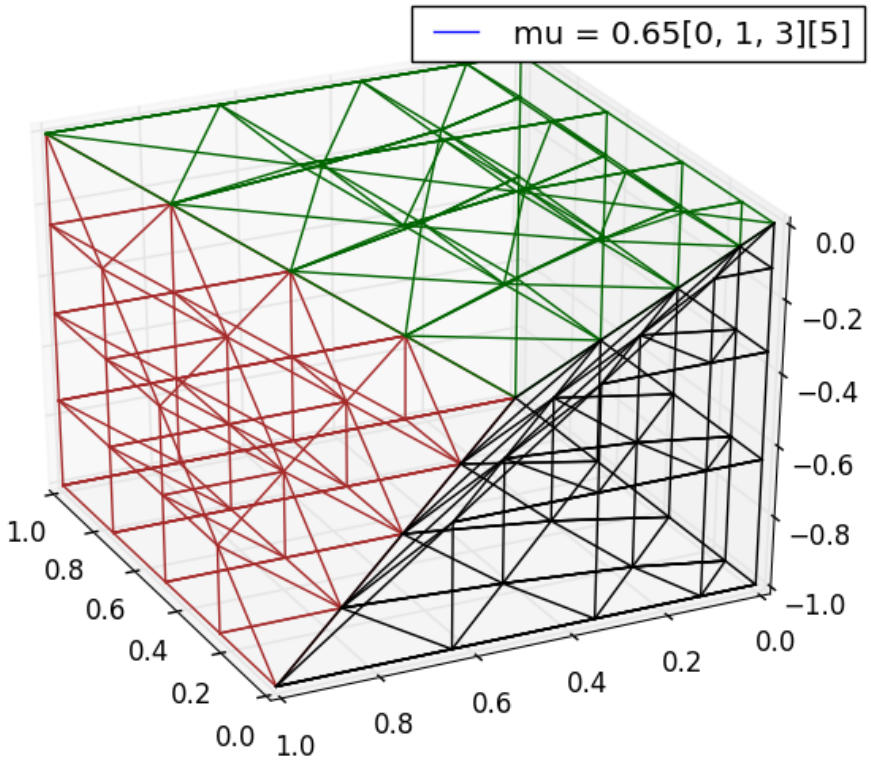
rotación + traslación + graduado indep. para cada macro–elem.

conformidad
por aristas



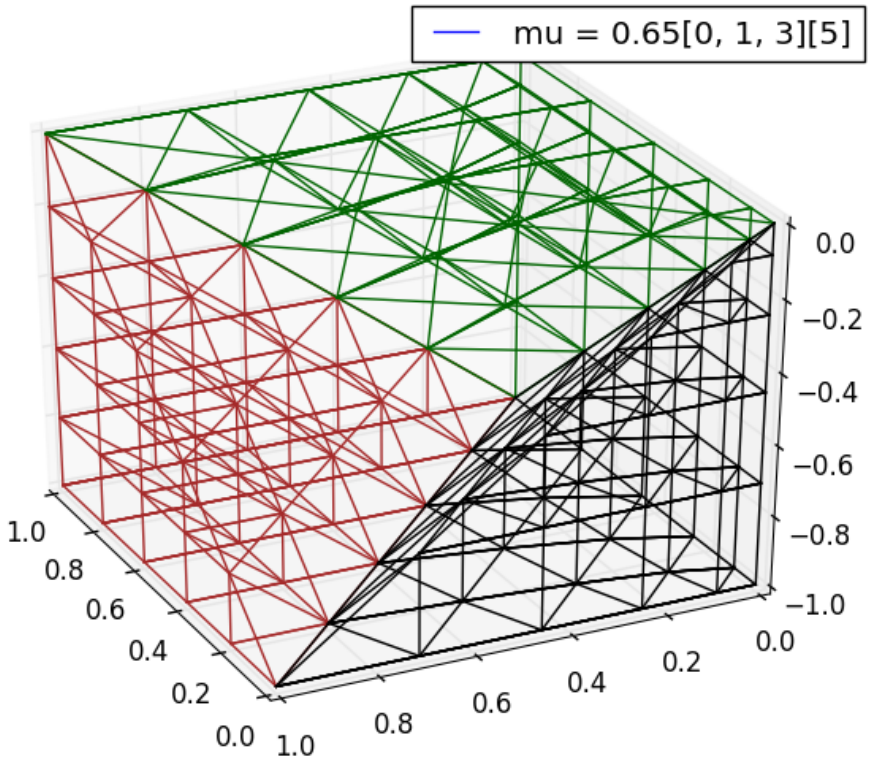
otación + traslación + graduado indep. para cada macro–elem.

conformidad
por aristas



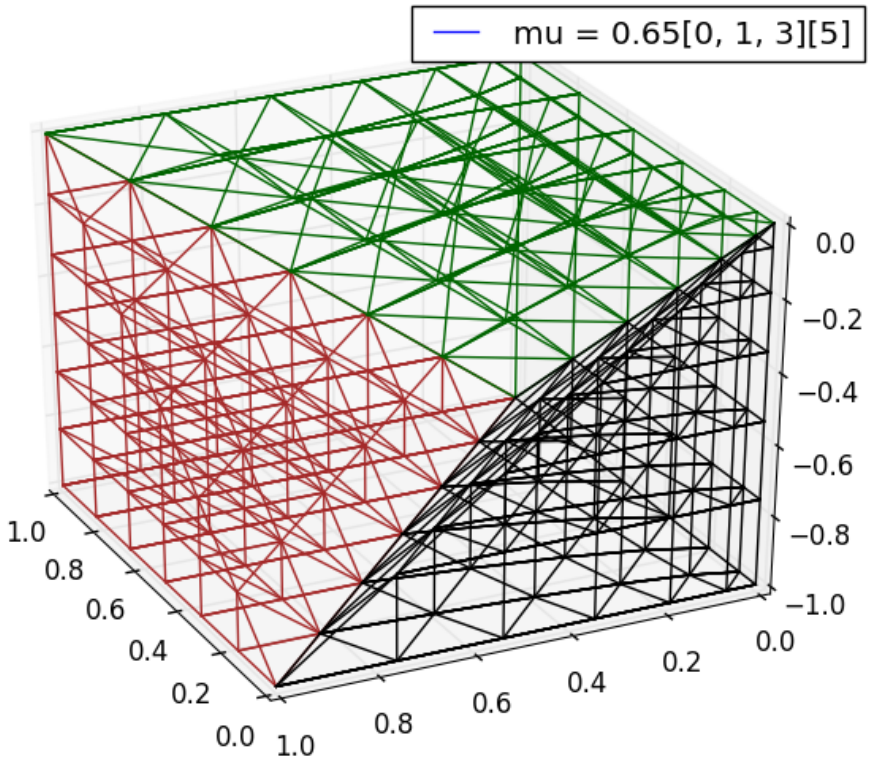
rotación + traslación + graduado indep. para cada macro–elem.

conformidad
por aristas



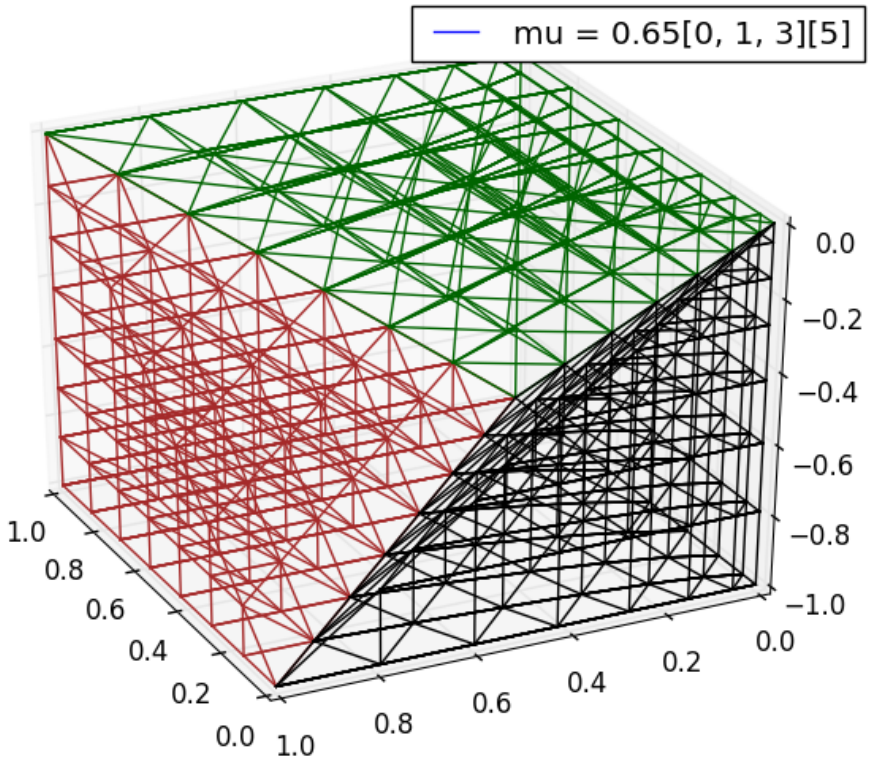
rotación + traslación + graduado indep. para cada macro–elem.

conformidad
por aristas



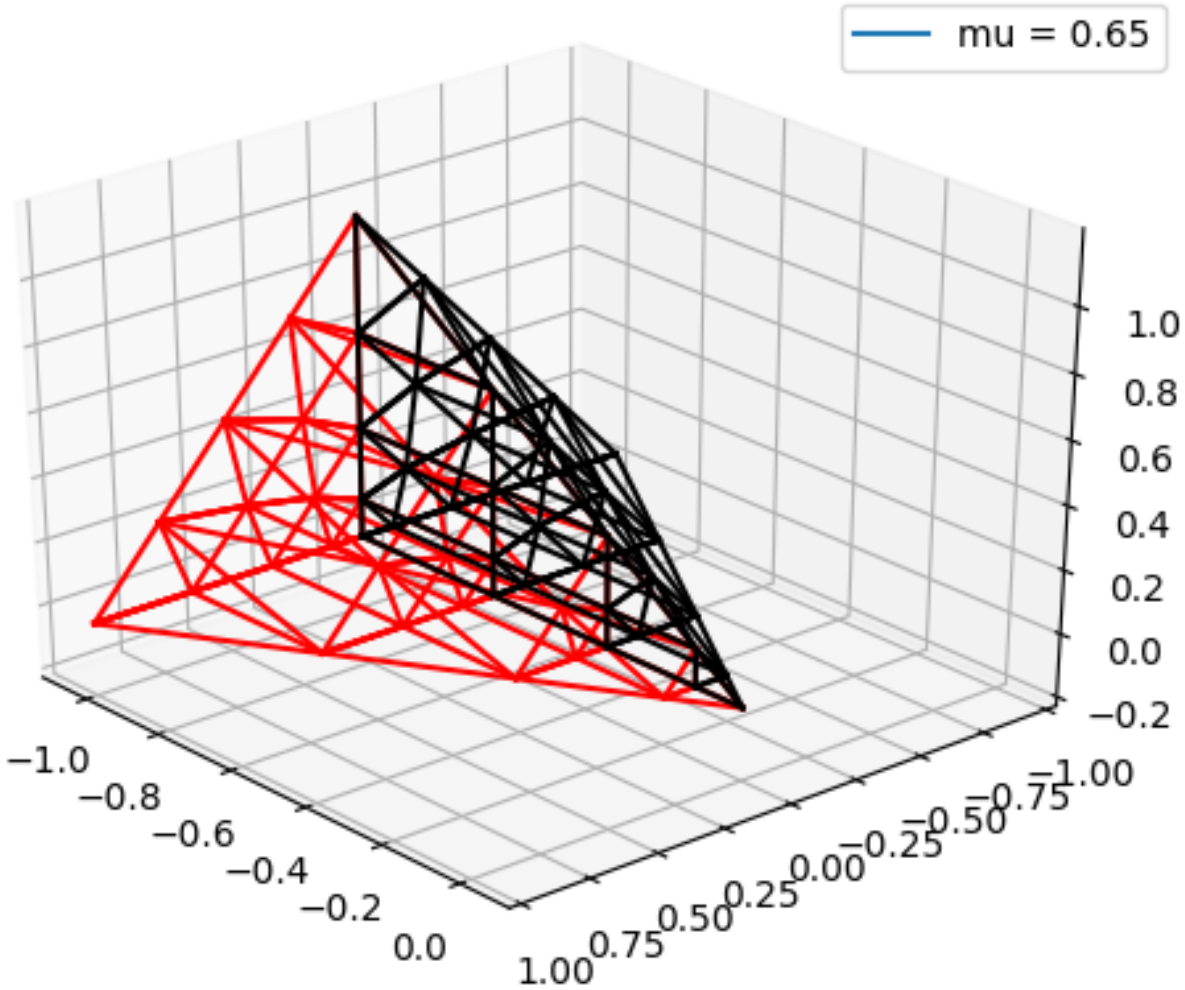
rotación + traslación + graduado indep. para cada macro–elem.

conformidad
por aristas

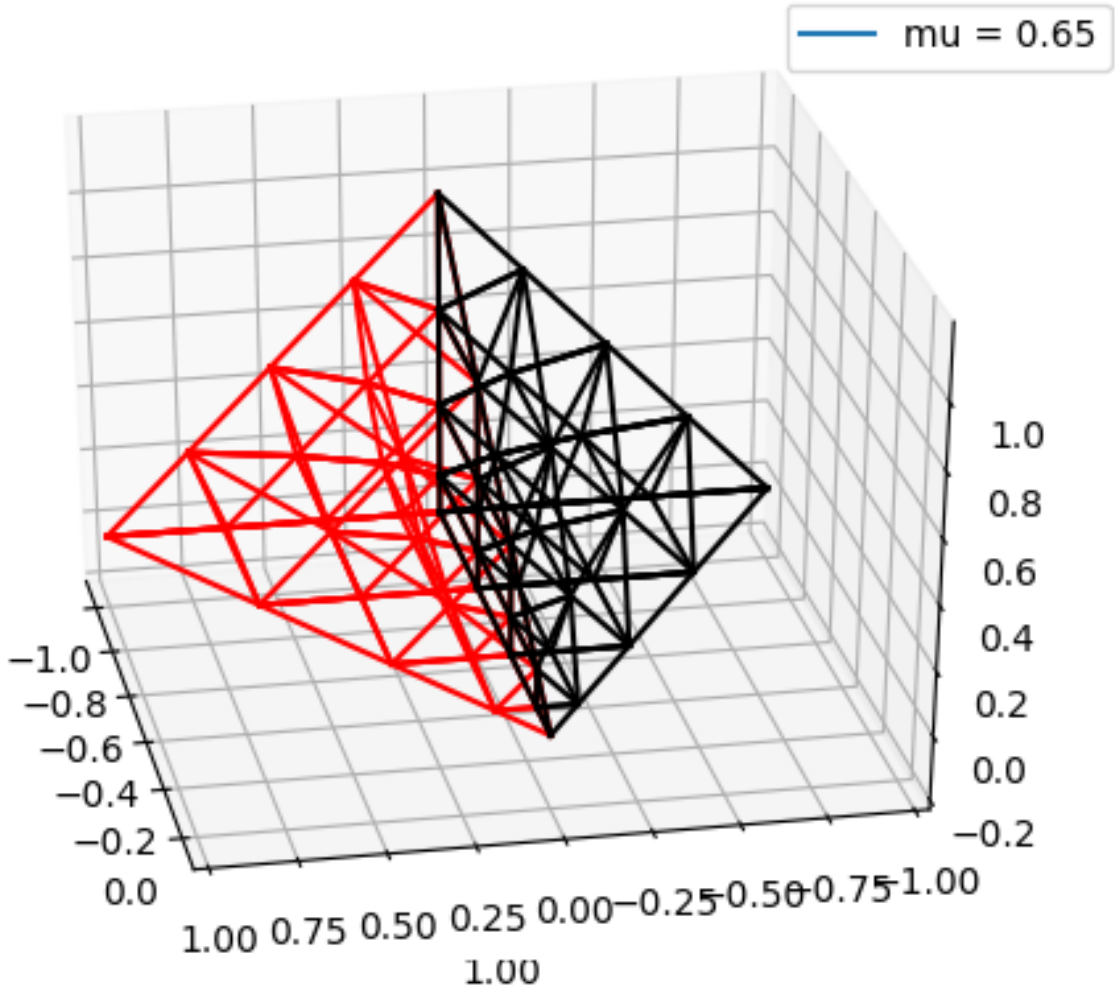


rotación + traslación + graduado indep. para cada macro–elem.

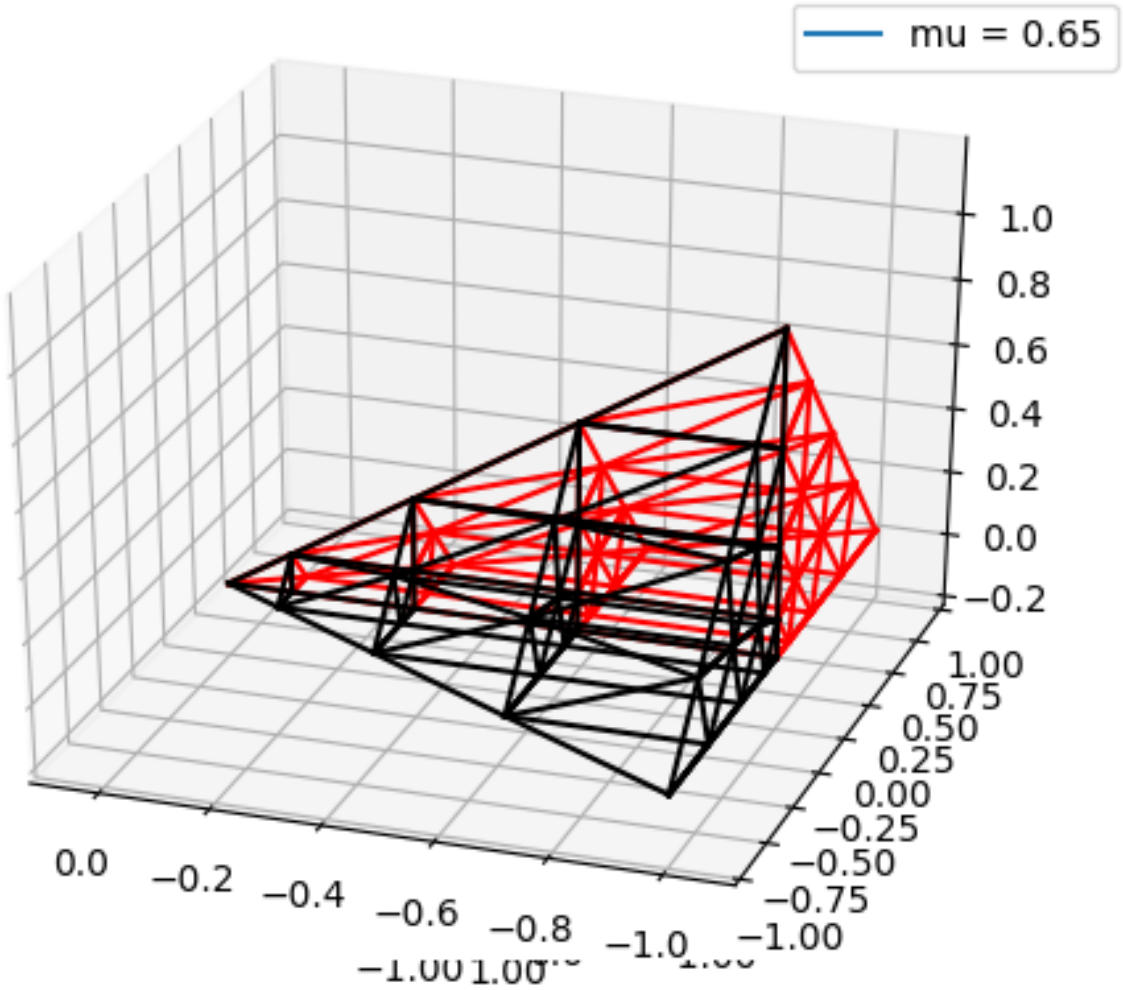
conformidad
por caras



conformidad
por caras



conformidad
por caras



`mesh_conectivity.kill_repeated(vertices_file_name)`

`mesh_conectivity.kill_repeated_faces(vertices_file_name)`

`example3.py` (uncomment lines 46, 47 or 68)

Observaciones

- para trabajar con álgebra lineal usamos NumPy.

Observaciones

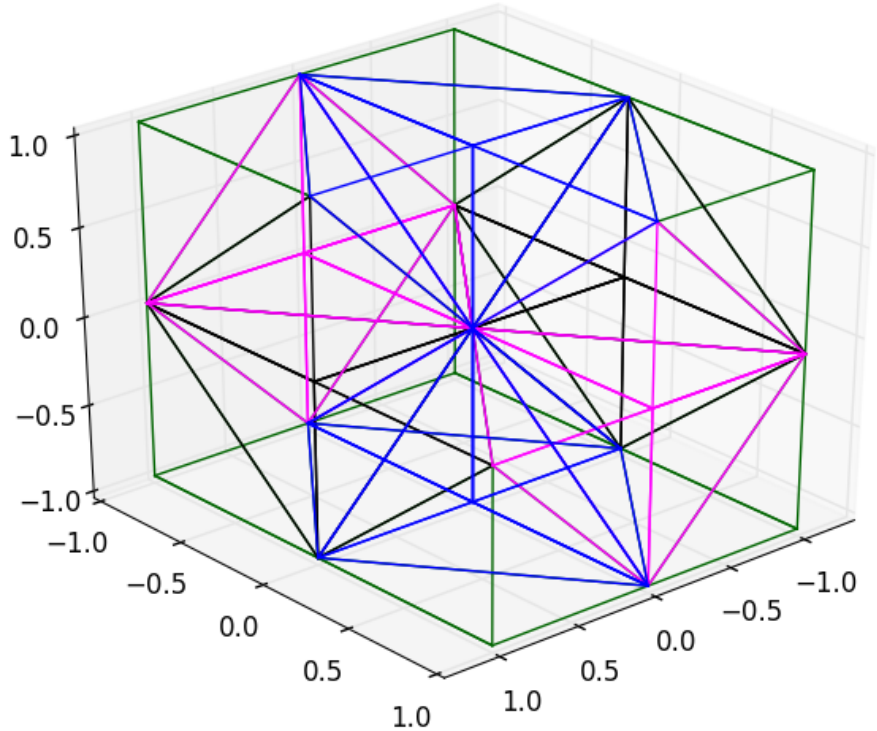
- para trabajar con álgebra lineal usamos NumPy.
- las singularidades pueden ser más fuertes o más débiles en distintas regiones de Ω.

Observaciones

- para trabajar con álgebra lineal usamos NumPy.
- las singularidades pueden ser más fuertes o más débiles en distintas regiones de Ω.
- el proceso de mallado admite parámetros de refinamiento diferentes en diferentes macro–elementos.

Meshing algorithm
Ejemplo $[-1,1]^3 \setminus (0,1)^3$.

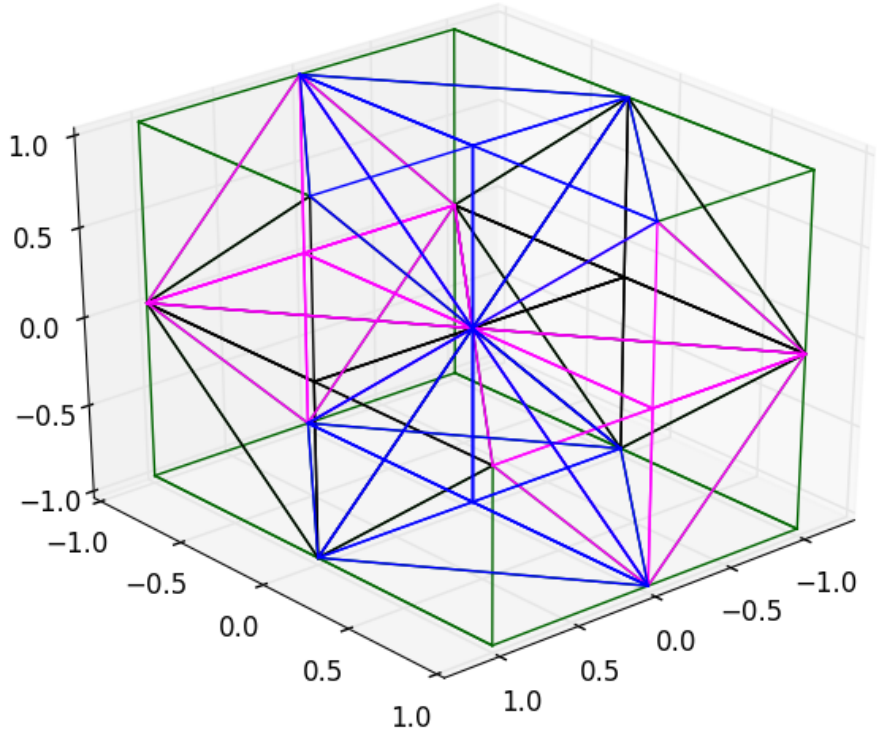
- https://github.com/alexisjawtu/mesher



Meshing algorithm
Ejemplo $[-1,1]^3 \setminus (0,1)^3$.

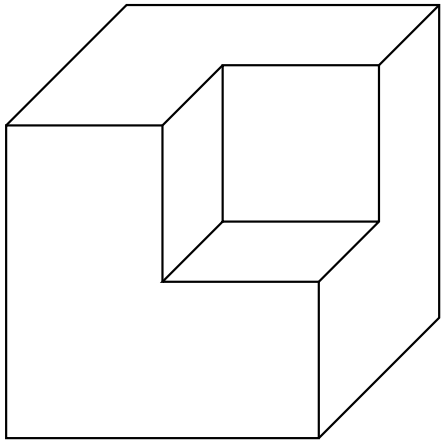
- https://github.com/alexisjawtu/mesher

  $\mathcal{T}_{h_0}$: partición en macro–tetraedros y prismas Λ_ℓ.



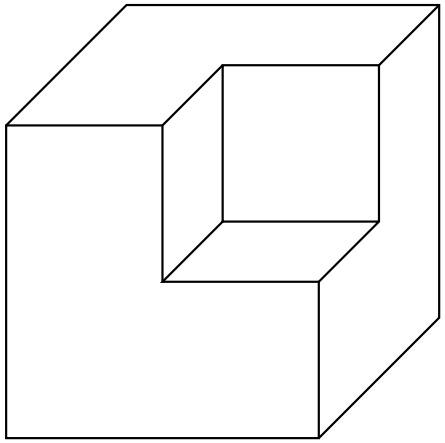
Meshing algorithm
Description/example Fichera Domain

- $[-1,1]^3 - (0,1)^3$



Meshing algorithm
Description/example Fichera Domain

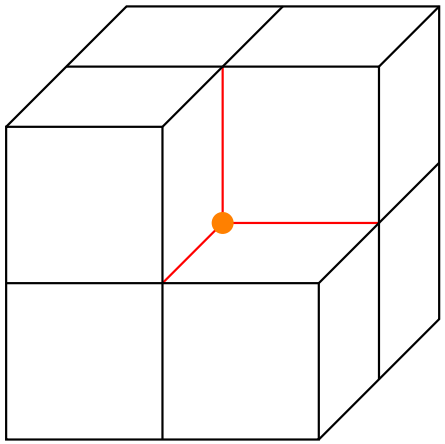
- $[-1,1]^3 - (0,1)^3$



- $\Omega = \cup_{\ell=1}^{N} \overline{\Lambda_\ell}$ (tetrahedra or prisms).

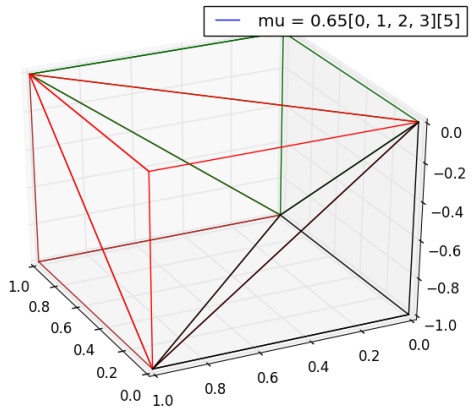
Meshing algorithm
Fichera Domain

- Into 7 cubes:



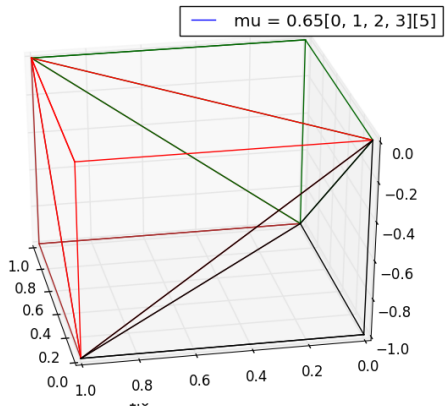
Meshing algorithm
En cada octante

Cubo $C = \cup_{i=1}^{5} T_i$



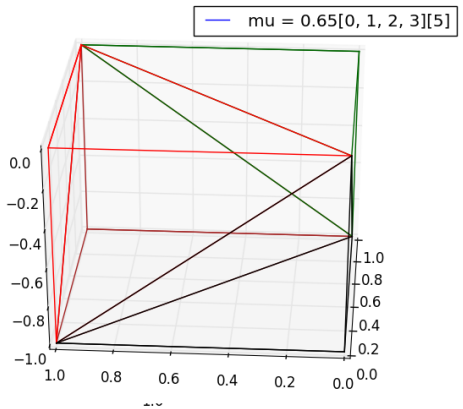
Meshing algorithm
En cada octante

Cubo $C = \cup_{i=1}^{5} T_i$



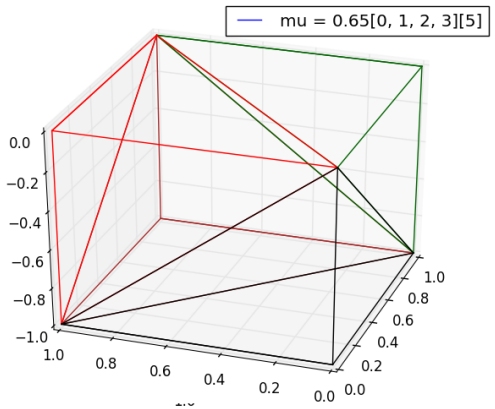
Meshing algorithm
En cada octante

Cubo $C = \cup_{i=1}^{5} T_i$

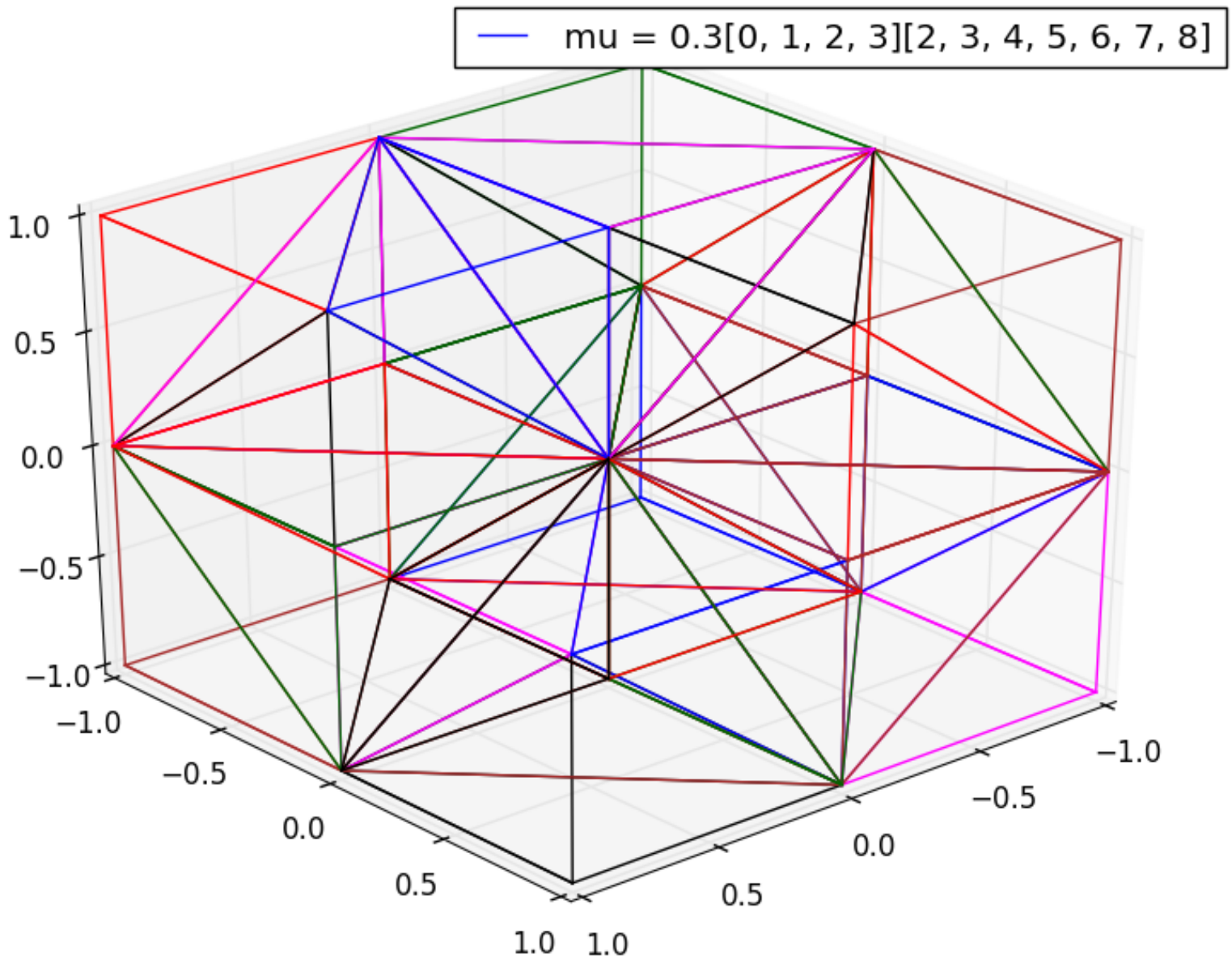


Meshing algorithm
En cada octante

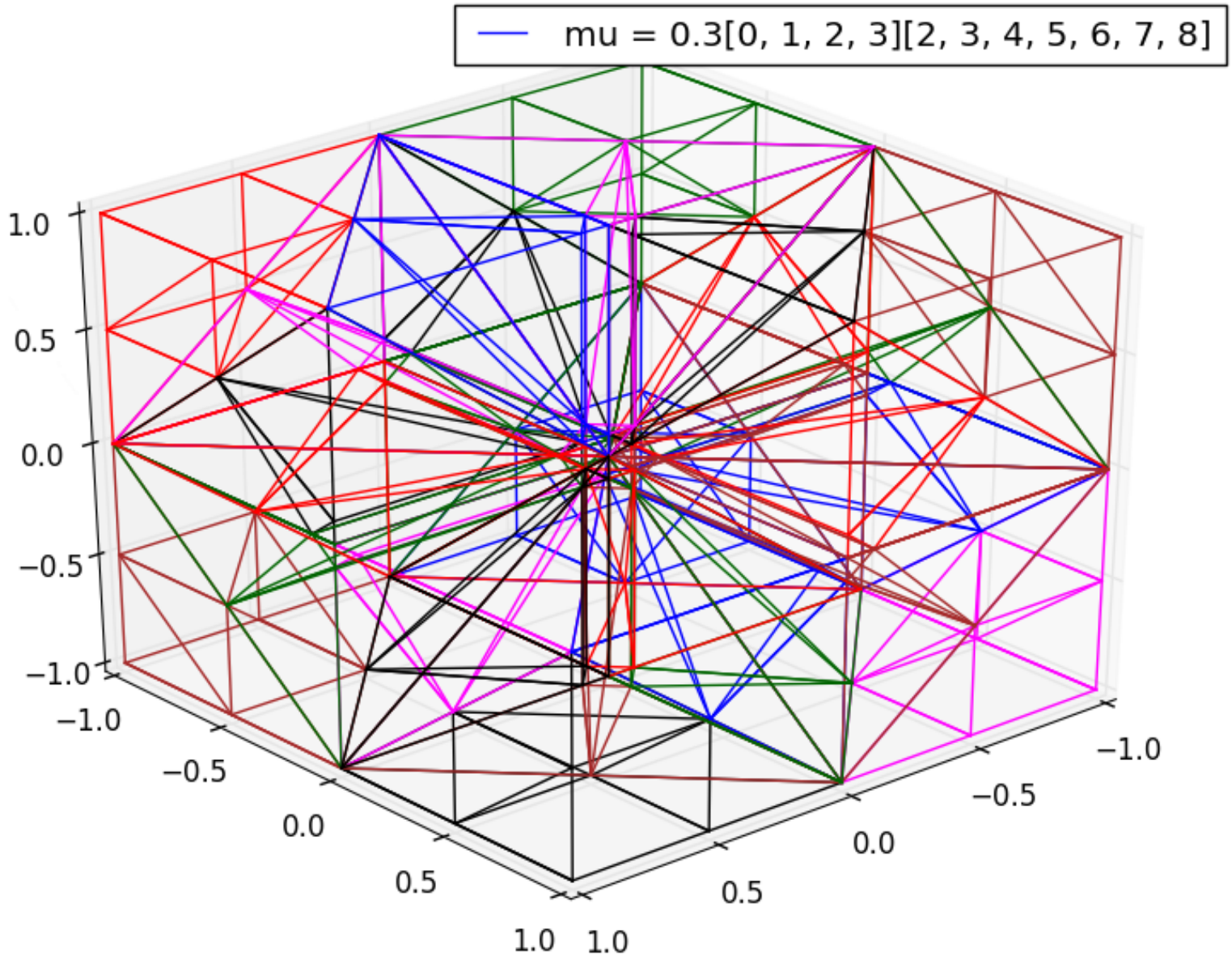
Cubo $C = \cup_{i=1}^{5} T_i$



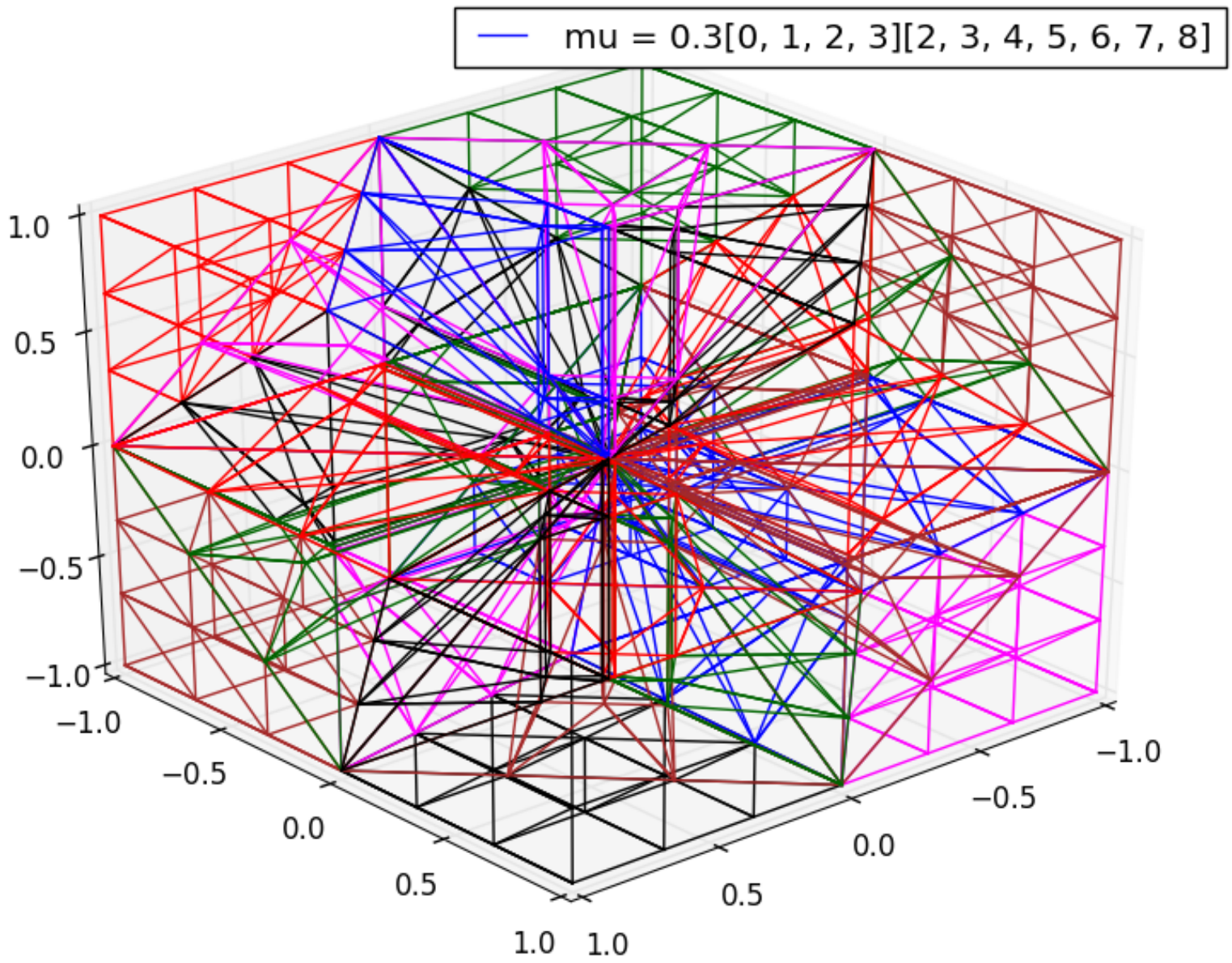
$\Omega \subseteq \mathbb{R}^3$
Dominio de Fichera



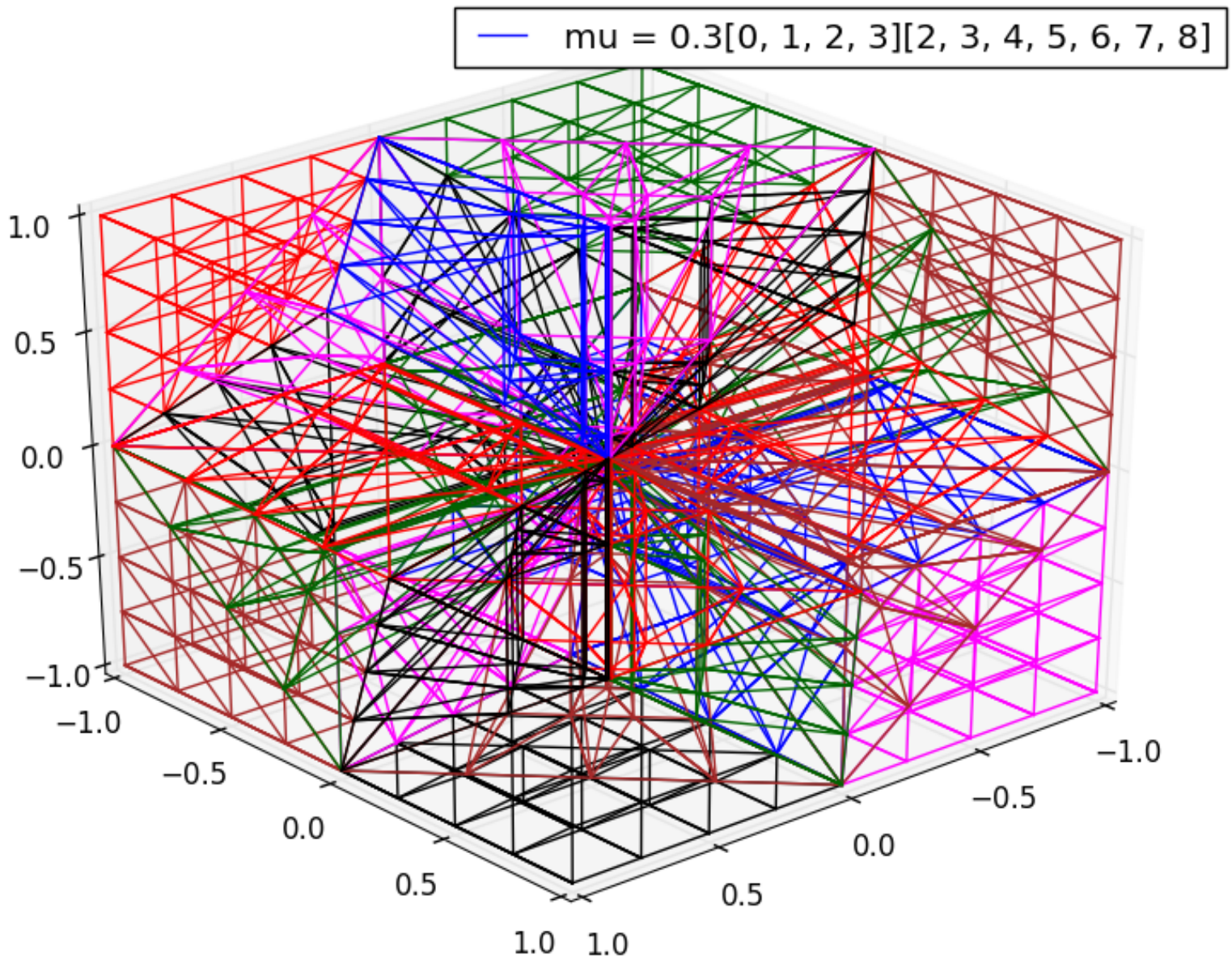
$\Omega \subseteq \mathbb{R}^3$
Dominio de Fichera



$\Omega \subseteq \mathbb{R}^3$
Dominio de Fichera

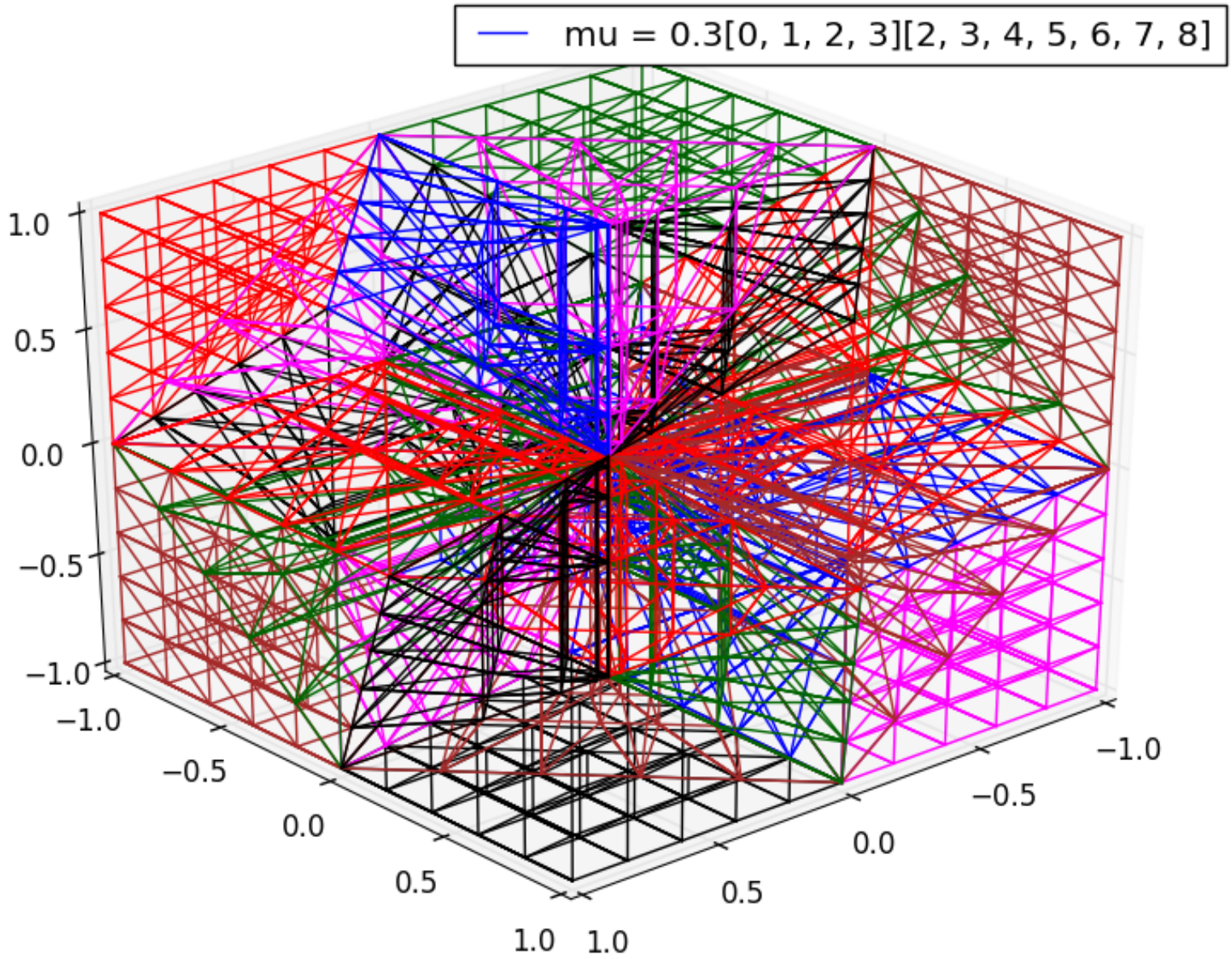


$\Omega \subseteq \mathbb{R}^3$
Dominio de Fichera

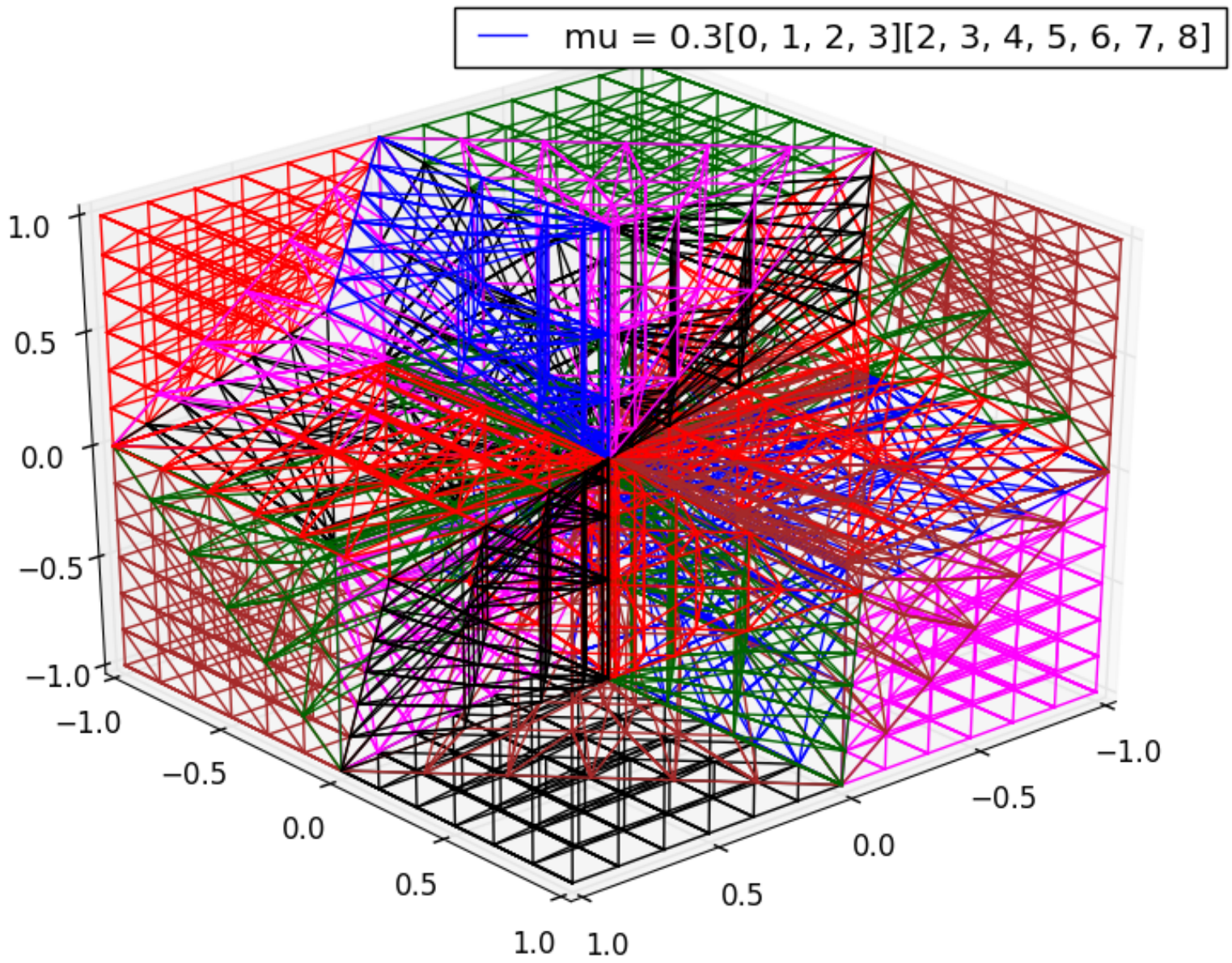


$\Omega \subseteq \mathbb{R}^3$

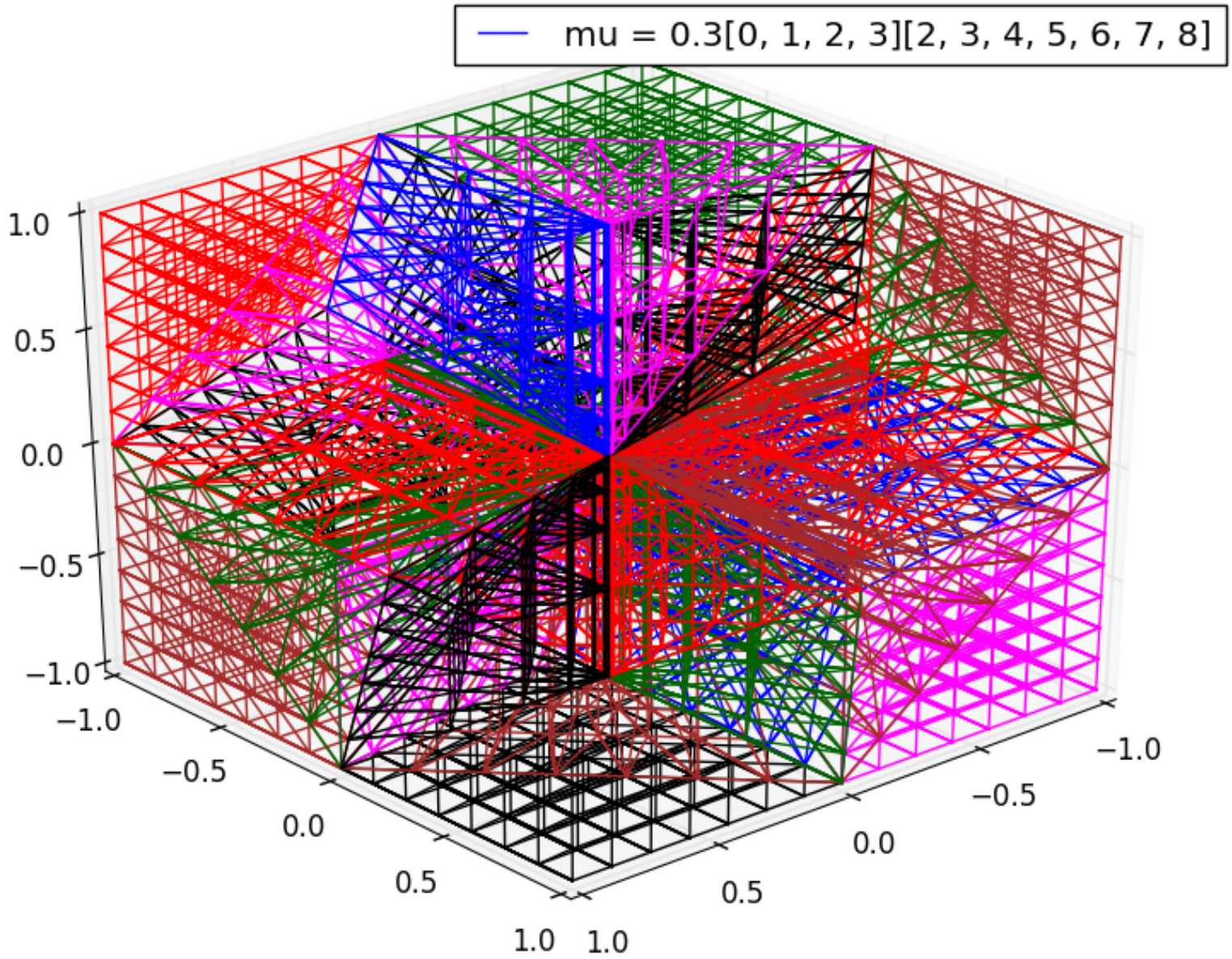
Dominio de Fichera



$\Omega \subseteq \mathbb{R}^3$
Dominio de Fichera



$\Omega \subseteq \mathbb{R}^3$
Dominio de Fichera



Ver

`example2.py`

(parte de la) salida

dos decimales

vertices.txt

```
 1. -1.00  1.00  1.00
 2. -1.00  1.00  0.50
 3. -1.00  1.00  0.00
 4. -1.00  0.50  1.00
 5. -1.00  0.50  0.50
 6. -1.00  0.00  1.00
 7. -0.50  1.00  1.00
 8. -0.50  1.00  0.50
 9. -0.50  0.50  1.00
10.  0.00  1.00  1.00
11. -1.00  0.00  0.00
12. -1.00  0.00  0.34
13. -1.00  0.00  1.00
14. -1.00  0.34  0.00
15. -1.00  0.50  0.50
16. -1.00  1.00  0.00
17. -0.34  0.00  0.00
18. -0.34  0.00  0.34
19. -0.34  0.34  0.00
20.  0.00  0.00  0.00
21.  0.00  0.00  0.00
22. -0.34  0.00  0.34
23. -0.34  0.34  0.00
24.  0.00  0.34  0.34
25. -0.34  0.00  0.34
        ...
```

faces.txt

```
3 1 4 2
3 7 9 8
4 4 1 9 7
4 1 2 7 8
4 2 4 8 9
3 2 5 3
3 2 5 8
3 2 3 8
3 5 3 8
3 4 6 5
3 4 6 9
3 4 5 9
3 6 5 9
3 2 4 5
3 9 8 5
3 7 9 10
3 7 8 10
3 9 8 10
3 11 14 12
3 17 19 18
4 14 11 19 17
4 11 12 17 18
4 12 14 18 19
3 12 5 6
3 12 5 18
   ...
```

elements_by_vertices.txt

```
6 1 4 2 7 9 8
4 2 5 3 8
4 4 6 5 9
5 2 4 9 8 5
4 7 9 8 10
6 11 14 12 17 19 18
4 12 5 6 18
4 14 3 5 19
5 12 14 19 18 5
4 17 19 18 20
4 20 18 19 24
4 18 6 5 9
4 19 5 3 8
4 24 9 8 10
4 18 19 24 9
4 19 24 9 8
4 18 19 5 9
4 19 5 9 8
   ...
```

elements_by_faces.txt (etc...)

```
2 1 2 3 4 5
0 6 7 8 9 0
0 10 11 12 13 0
1 14 7 15 12 5
0 2 16 17 18 0
2 19 20 21 22 23
0 24 25 26 27 0
0 28 29 30 31 0
1 32 25 33 30 23
0 20 34 35 36 0
0 36 37 38 39 0
0 27 40 41 13 0
0 31 42 43 9 0
0 44 45 46 18 0
0 39 47 48 49 0
0 49 50 51 44 0
0 33 47 41 52 0
0 52 42 51 15 0
   ...
```

Ensamblado del sistema

- el problema sobre la malla es un sistema de ecuaciones lineales con matriz en bloques

Ensamblado del sistema

- el problema sobre la malla es un sistema de ecuaciones lineales con matriz en bloques
-

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$

Ensamblado del sistema

- el problema sobre la malla es un sistema de ecuaciones lineales con matriz en bloques
- $$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$
- $(U\ P)^t$ es la incógnita

Ensamblado del sistema

- el problema sobre la malla es un sistema de ecuaciones lineales con matriz en bloques
- $$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$
- $(U\ P)^t$ es la incógnita
- A, B y F hay que calcularlos con un progr. de ensamblado que toma como entrada los archivos de texto de la malla.

Ensamblado del sistema

- https://github.com/alexisjawtu/virtualAssembler

Ensamblado del sistema

- https://github.com/alexisjawtu/virtualAssembler
- en GNU Octave

Ensamblado del sistema

- https://github.com/alexisjawtu/virtualAssembler
- en GNU Octave
- (V.2 en Python usando NumPy)

Ensamblado Global

```
%% Author: Alexis Jawtuschenko.
function [K,F,num_fc,num_el] = assemble()

  vertices              = importdata ('vertices.txt');
  elements              = importdata ('elements_by_vertices.txt');
  elements_by_faces     = importdata ('elements_by_faces.txt');
  faces                 = importdata ('faces.txt');

  ...

  assemble_local    = {};
  assemble_local(4) = @assemble_tetr;
  assemble_local(5) = @assemble_pyram;
  assemble_local(6) = @assemble_prism;

  ...

  n_Faces     = {};
  n_Faces(4)  = 4;
  n_Faces(5)  = 5;
  n_Faces(6)  = 5;
```

Ensamblado Global

```
K = sparse(num_fc + num_el,num_fc + num_el);
F = sparse(num_fc + num_el,1);

for el = 1:num_el

  ...


  [local_matrix, local_F]   = assemble_local{n_VERT}(P,faces_of_E,face_types,
    ...
    face_pts,normalFacesE,measFacesE);
  W                         = ones(1,nFacesE);
  K(faces_of_E,faces_of_E) += local_matrix;
  K(num_fc + el,faces_of_E) = W;
  K(faces_of_E,num_fc + el) = W.';
  F(num_fc + el,1)          = local_F;

end

...

endfunction
```

Ejemplo
Ensamblado local en Prismas

```
%% assemble_prism: Finite Element type local assembly
function [local_matrix, local_F] = assemble_prism(vertices,face_indices,
  face_types,face_pts,normalFacesE,measFacesE)

  ...

  n_vertices    = 6;
  n_vol_pts     = 9;
  dim_Vh        = 5;
  n_vertices    = 6;

  ...

  vol_pts       = zeros(3,n_vol_pts);
  vol_pts(:,1) = mean([vertices(:,1),vertices(:,3)],2);
  vol_pts(:,2) = mean([vertices(:,2),vertices(:,3)],2);
  vol_pts(:,3) = mean([vertices(:,2),vertices(:,1)],2);
  vol_pts(:,7) = mean([vertices(:,4),vertices(:,6)],2);
  vol_pts(:,8) = mean([vertices(:,5),vertices(:,6)],2);
  vol_pts(:,9) = mean([vertices(:,5),vertices(:,4)],2);
  vol_pts(:,4) = mean([vol_pts(:,7),vol_pts(:,1)],2);
  vol_pts(:,5) = mean([vol_pts(:,8),vol_pts(:,2)],2);
  vol_pts(:,6) = mean([vol_pts(:,9),vol_pts(:,3)],2);
```

Ejemplo
Ensamblado local en Prismas

```
%% assemble_prism: Finite Element type local assembly
function [local_matrix, local_F] = assemble_prism(vertices,face_indices,
  face_types,face_pts,normalFacesE,measFacesE)

  ...

  n_vertices    = 6;
  n_vol_pts     = 9;
  dim_Vh        = 5;
  n_vertices    = 6;

  ...

  vol_pts       = zeros(3,n_vol_pts);
  vol_pts(:,1) = mean([vertices(:,1),vertices(:,3)],2);
  vol_pts(:,2) = mean([vertices(:,2),vertices(:,3)],2);
  vol_pts(:,3) = mean([vertices(:,2),vertices(:,1)],2);
  vol_pts(:,7) = mean([vertices(:,4),vertices(:,6)],2);
  vol_pts(:,8) = mean([vertices(:,5),vertices(:,6)],2);
  vol_pts(:,9) = mean([vertices(:,5),vertices(:,4)],2);
  vol_pts(:,4) = mean([vol_pts(:,7),vol_pts(:,1)],2);
  vol_pts(:,5) = mean([vol_pts(:,8),vol_pts(:,2)],2);
  vol_pts(:,6) = mean([vol_pts(:,9),vol_pts(:,3)],2);
```

Ejemplo
Ensamblado local en Prismas

```
  for l = 1:n_vol_pts
    we_basis(:,:,l) = WE_basis(vol_pts(:,l),n_vertices);
  end

  int_E_w_w = zeros(dim_Vh);

  for r = 1:dim_Vh
    for k = 1:dim_Vh
      ...
      vals            = reshape(dot(we_basis(:,r,:),we_basis(:,k,:),1),1,
        n_vol_pts);}
      ...
    end
  end

  ...

  local_matrix = int_E_w_w;
  local_F      = quad_nrmlztn*f(vol_pts)*vol_weights;
endfunction
```

Ejemplo
Ensamblado local en Prismas

```
  for l = 1:n_vol_pts
    we_basis(:,:,l) = WE_basis(vol_pts(:,l),n_vertices);
  end

  int_E_w_w = zeros(dim_Vh);

  for r = 1:dim_Vh
    for k = 1:dim_Vh
      ...
      vals            = reshape(dot(we_basis(:,r,:),we_basis(:,k,:),1),1,
        n_vol_pts);}
      ...
    end
  end

  ...

  local_matrix = int_E_w_w;
  local_F      = quad_nrmlztn*f(vol_pts)*vol_weights;
endfunction
```

Ensamblado local

- en tetraedros es similar.

Ensamblado local

- en tetraedros es similar.
- en pirámides es mucho más largo (los elementos virtuales no permiten conocer explícitamente las funciones a calcular).

```
%% assembly
%% discrete solution
%% with num_fc and num_el we know how to split X
[K,F,num_fc,num_el]     = assemble();
X                       = K\F;
```

```
%% assembly
%% discrete solution
%% with num_fc and num_el we know how to split X
[K,F,num_fc,num_el]     = assemble();
X                       = K\F;
```

Es decir,

$$X = \begin{pmatrix} U \\ P \end{pmatrix} \qquad K = \begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \qquad F = \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$

```
%% assembly
%% discrete solution
%% with num_fc and num_el we know how to split X
[K,F,num_fc,num_el]     = assemble();
X                       = K\F;
```

Es decir,

$$X = \begin{pmatrix} U \\ P \end{pmatrix} \qquad K = \begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \qquad F = \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$

```
%% assembly
%% discrete solution
%% with num_fc and num_el we know how to split X
[K,F,num_fc,num_el]     = assemble();
X                       = K\F;
```

Es decir,

$$X = \begin{pmatrix} U \\ P \end{pmatrix} \qquad K = \begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \qquad F = \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$

$$\begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 \\ -F_I \end{pmatrix}$$

Fin. Muchas gracias.