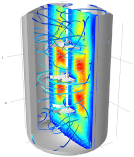
CutFEM 有限元介绍

Luke Chern

March 13, 2025

西安工程大学（机电工程学院）



Overview

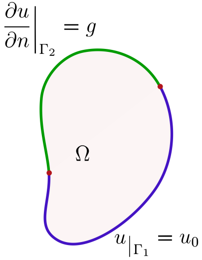
1. Part I: 一般有限元
2. Part II: CutFEM 有限元
3. Part III: 线弹性结构有限元
4. Part IV: 杂项
5. Part V: 项目规划

Part I: 一般有限元

Let Ω be a domain in R^d , d $=$ 2 or 3, with a piecewise smooth boundary $\partial\Omega$ consisting of two disjoint parts

$$\partial\Omega = \Gamma_D \cup \Gamma_N$$

where Γ_D and Γ_N are the Dirichlet and Neumann parts of the boundary, respectively.



一般有限元

Let us consider first the Poisson equation in Ω with Dirichlet boundary conditions on $\Gamma_D \subset \partial\Omega$ and Neumann boundary conditions on $\Gamma_N \subset \partial\Omega/\Gamma_D$.

$$\begin{array}{rcl} -\Delta u & = & f \quad \text{in } \Omega \\ u & = & u_0 \quad \text{on } \Gamma_D \\ \mathbf{n} \cdot \nabla u & = & g_N \quad \text{in } \Gamma_N \end{array}$$

一般有限元

$$
\mathcal{H}_D^1 := \{u \in \mathcal{H}^1(\Omega) \mid u = u_0 \text{ on } \Gamma_D\}
$$

$$
\mathcal{H}_N^1 := \{u \in \mathcal{H}^1(\Omega) \mid u = g_N \text{ on } \Gamma_N\}
$$

The associated weak formulation is the following:
find $u \in \mathcal{H}_D^1$ such that

$$
\boxed{\int_\Omega \nabla u \cdot \nabla v \mathrm{d}\Omega = \int_\Omega v f \mathrm{d}\Omega + \int_{\Gamma_N} v g_N \mathrm{d}s}
$$

for all $v \in \mathcal{H}_N^1$.

Example 1: 二维泊松方程

考虑一个定义在单位圆区域上的特殊问题，在单位元内部

$$u = \frac{1 - x^2 - y^2}{2}$$

在边界 $\partial\Omega$ 上 $u = 0$。

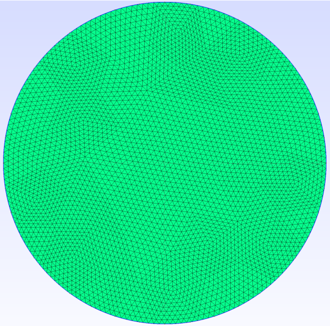


Figure 1: 网格

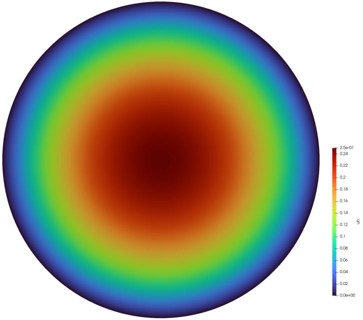


Figure 2: 分析结果

Example 1: 二维泊松方程

Listing 1: 二维泊松方程

```julia
using GridapGmsh
# 读取网格
model = GmshDiscreteModel("./src/circle.msh")

Ω = Triangulation(model)
dΩ = Measure(Ω, 2)

# 有限元空间
reffe = ReferenceFE(lagrangian, Float64, 1)
V = TestFESpace(model, reffe, dirichlet_tags="fixed")

g(x) = 0.0
U = TrialFESpace(V, g)
```

Example 1: 二维泊松方程

Listing 2: 二维泊松方程（续）

```
# 定义弱形式
f(x) = 1.0
a(u,v) = ∫( ∇(v)⋅∇(u) )dΩ
l(v) = ∫( v*f )dΩ

# 求解线性方程组
op = AffineFEOperator(a, l, U, V)
uh = solve(op)

# 输出结果
writevtk(Ω, "results", cellfields=["uh"=>uh])
```

The Nitsche Method

We introduce a method for treating general boundary conditions in the finite element method generalizing an approach, due to Nitsche, for approximating Dirichlet boundary conditions.
Find $u_h \in V_h$ such that

$$\mathcal{A}_h(u_h, v) = l_h(v).$$

for all $v \in V_h$.

The Nitsche Method

$$\mathcal{A}_h(u_h, v) = \int_\Omega \nabla u \cdot \nabla v \mathrm{d}\Omega + \int_{\Gamma_D} (\lambda u v - v(\mathbf{n} \cdot \nabla u) - u(\mathbf{n} \cdot \nabla v))\mathrm{d}s$$

$$l_h(v) = \int_\Omega v f \mathrm{d}\Omega + \int_{\Gamma_D} \left(\lambda v g_D - \frac{\partial v}{\partial \mathbf{n}} g_D\right)\mathrm{d}s$$

Example 2: Nitsche 法求解二维泊松方程

Listing 3: Nitsche 法求解二维泊松方程

```julia
# 读取网格
Ω = Triangulation(model)
dΩ = Measure(Ω, 2)

Γd = BoundaryTriangulation(model,tags="fixed")
dΓd = Measure(Γd, 2)
n_Γd = get_normal_vector(Γd)

# 有限元空间
reffe = ReferenceFE(lagrangian, Float64, 1)
V = TestFESpace(model, reffe, conformity=:H1)

U = TrialFESpace(V)
```

Example 2: Nitsche 法求解二维泊松方程

Listing 4: Nitsche 法求解二维泊松方程（续）

```
γd = 10.0
h = 1.0 / 400

ud(x) = 0.0
f(x) = 1.0

a(u,v) =∫( ∇(v)⋅∇(u) )dΩ +
∫( (γd/h)*v*u  - v*(n_Γd⋅∇(u)) - (n_Γd⋅∇(v))*u )dΓd

l(v) =  ∫( v*f )dΩ  +
∫( (γd/h)*v*ud - (n_Γd⋅∇(v))*ud ) * dΓd
```

分析结果对比:

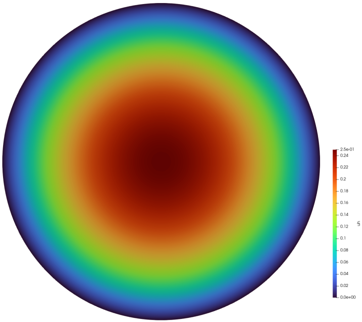


Figure 3: 一般方法

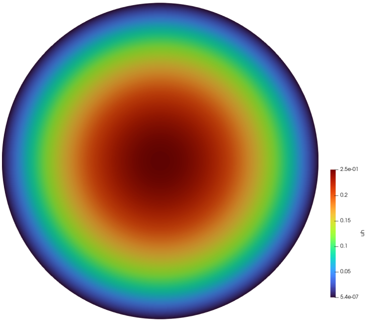
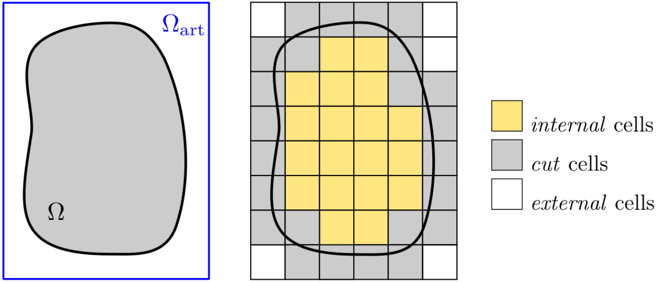


Figure 4: The Nitsche Method

Part II: CutFEM 有限元

CutFEM 有限元

Like in any other *embedded boundary method*, we build the computational mesh by introducing an **artificial domain** Ω_{art} such that it has a simple geometry that is easy to mesh using Cartesian grids and it includes the **physical domain** $\Omega \subset \Omega_{\text{art}}$.

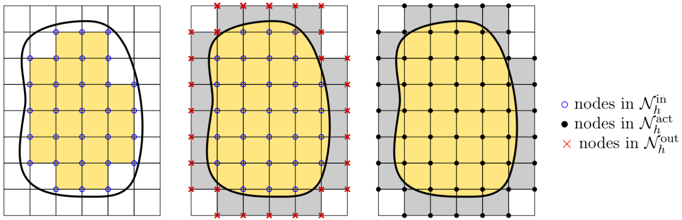


(A) (B)

Figure 5: Embedded boundary setup.

CutFEM 有限元

Let us construct a partition of Ω_{art} into cells, represented by T^h_{art} , with characteristic cell size h. Cells in T^h_{art} can be classified as follows: a cell $K \in T^h_{\text{art}}$ such that $K \subset \Omega$ is an **internal cell**; if $K \cap \Omega = \emptyset$, K is an **external cell**; otherwise, K is a **cut cell** (see Fig. 5). Furthermore, we define the set of **active cells** as $T_h^{\text{act}} = T_h^{\text{in}} \cup T_h^{\text{cut}}$ and its union Ω_{act}.



(A) V_h^{in} (B) V_h^{agg} (C) V_h^{act}

Figure 6: Finite Element spaces.

CutFEM 有限元

We define the FE-wise operators:

$$\mathcal{A}_K(u,v) \doteq \int_{K\cap\Omega} \nabla u \cdot \nabla v \mathrm{d}V$$

$$+ \int_{\partial K\cap\Gamma_D} \left(h^{-1}\lambda uv - v(\mathbf{n}\cdot\nabla u) - u(\mathbf{n}\cdot\nabla v)\right)\mathrm{d}S$$

$$+ \int_{\partial K\cap\Gamma_{\text{ghost}}} h^{-1}\mu[\![\partial_n u]\!][\![\partial_n v]\!]\mathrm{d}S$$

$$l_K(u,v) \doteq \int_{K\cap\Omega} vf\mathrm{d}V + \int_{\partial K\cap\Gamma_D} \left(h^{-1}\lambda v g_D - (\mathbf{n}\cdot\nabla v)g_D)\right)\mathrm{d}S$$

CutFEM 有限元

For Neumann boundary conditions, the consequence is that the stiffness matrix can become arbitrarily ill-conditioned as the cut-size approaches zero.

For a Dirichlet condition, the situation is even worse. For any finite choice of Nitsche constant, λ_D, the bilinear form $a_h(\cdot,\cdot)$ loses coercivity as the size of a cell cut approaches zero.

This makes the above weak formulation essentially useless because we typically can not control how the cells intersect Γ. One way to avoid this problem is to add a so-called ghost penalty term, s_h, to the weak formulation.

CutFEM 有限元

Given a face $F \in \mathcal{F}_h^{\text{ghost}}$ and the two cells K and K' sharing this face, we define the jump operator

$$[\![\partial_n u]\!] \doteq \mathbf{n}_K \cdot \nabla u|_K + \mathbf{n}_{K'} \cdot \nabla u|_{K'},$$

h_F is some average of h_K and $h_{K'}$.
We define the GP stabilisation term:

$$s_h(u,v) = \sum_{F \in \mathcal{F}_h^{\text{ghost}}} \left(\lambda_G h_F [\![\partial_n u]\!], [\![\partial_n v]\!]\right)_F .$$

Example 3: CutFEM 求解

We assume that Ω is described by a level set function $\psi : \mathbb{R}^{\mathrm{dim}} \to \mathbb{R}$ such that

$$\Omega = \{x \in \mathbb{R}^{\mathrm{dim}} : \psi(x) < 0\},$$
$$\Gamma = \{x \in \mathbb{R}^{\mathrm{dim}} : \psi(x) = 0\}.$$

For simplicity, we choose Ω to be a unit disk, so that $\psi(x) = \|x\| - 1$. As can be seen from the figure below, the level set function is negative for points in Ω, zero on the boundary, and positive everywhere else.

Example 3: CutFEM 求解

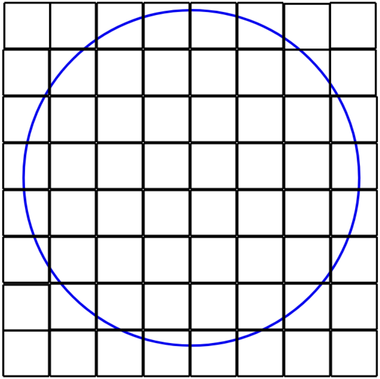


Figure 7: 背景网格 $\mathcal{T}^h$

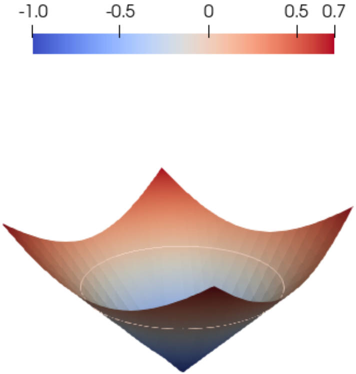


Figure 8: 水平集函数 $\psi(x)$

Example 3: CutFEM 求解

```
# 1. Build background mesh
nn = 40
partition = (nn,nn)
pmin = 1.2*Point(-1,-1)
pmax = 1.2*Point(1,1)
bgmodel = CartesianDiscreteModel(pmin,pmax,partition)

# 2. Build CSG geometry
R = 1.0
geo = disk(R, name="csg")

# 3. Cut the background model
cutgeo = cut(bgmodel,geo)
```

Example 3: CutFEM 求解

To solve this problem, we want to distribute degrees of freedom over the smallest submesh, $\mathcal{T}^h_\Omega$, that completely covers the domain:

$$\mathcal{T}^h_\Omega = \{T \in \mathcal{T}^h : T \cap \Omega \neq \emptyset\}.$$

The finite element space where we want to find our numerical solution, uh, is now

$$V^h_\Omega = \{v \in C(\mathcal{N}^h_\Omega) : v \in Q_p(T),\, T \in \mathcal{T}^h_\Omega\},$$

where $\mathcal{N}^h_\Omega = \bigcup_{T \in \mathcal{T}^h_\Omega} \overline{T}$.

Example 3: CutFEM 求解

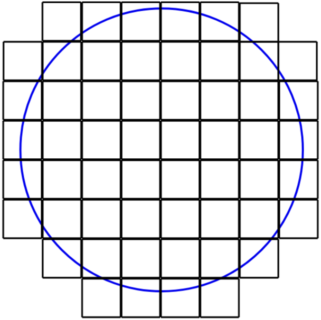
To define the ghost penalty, let $\mathcal{T}_\Gamma^h$ be the set of intersected cells:

$$\mathcal{T}_\Gamma^h = \{T \in \mathcal{T}_\Omega^h : T \cap \Gamma \neq \emptyset\},$$

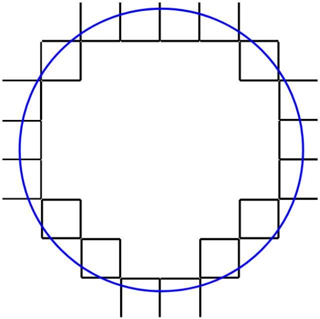
and let $\mathcal{F}_h$ denote the interior faces of the intersected cells in the active mesh:

$$\mathcal{F}_h = \{F = \overline{T}_+ \cap \overline{T}_- : \, T_+ \in \mathcal{T}_\Gamma^h, \, T_- \in \mathcal{T}_\Omega^h\}.$$

Example 3: CutFEM 求解



$\mathcal{T}_{\Omega}^{h}$



$\mathcal{F}_h$

Example 3: CutFEM 求解

```
# 生成计算域
Ω = Triangulation(cutgeo,PHYSICAL,"csg")
Ω_act = Triangulation(cutgeo,ACTIVE,"csg")
```

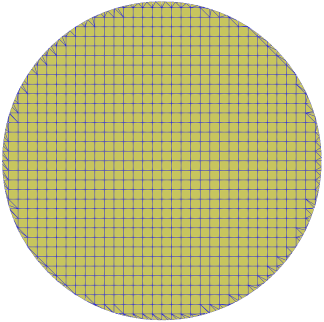


Figure 9: 物理域 Ω

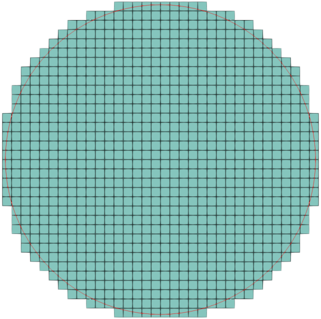


Figure 10: 求解域 Ω_{act}

Example 3: CutFEM 求解

```julia
Γd = EmbeddedBoundary(cutgeo,"csg")
Γg = GhostSkeleton(cutgeo,"csg")
```

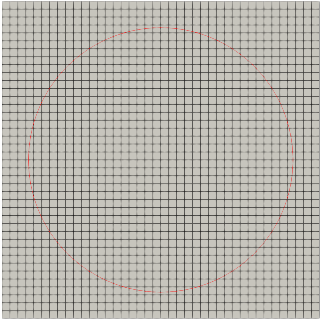


Figure 11: 背景网格 Ω_{bg}

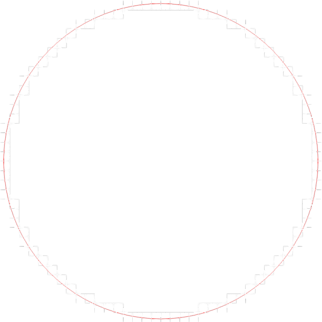


Figure 12: 几何

Example 3: CutFEM 求解

The ghost penalty acts on these faces and reads

$$g_h(u_h, v_h) = \gamma_A \sum_{F \in \mathcal{F}_h} g_F(u_h, v_h),$$

where g_F is the face-wise ghost penalty:

$$g_F(u_h, v_h) = \gamma_A \sum_{k=1}^{p} \left(\frac{h_F^{2k-1}}{k!^2} [\partial_n^k u_h], [\partial_n^k v_h] \right)_F$$

We shall use a continuous space of Q1-elements, so the ghost penalty is reduced to

$$g_h(u_h, v_h) = \gamma_A \sum_{F \in \mathcal{F}_h} (h_F [\partial_n u_h], [\partial_n v_h])_F$$

Example 3: CutFEM 求解

Find $u_h \in V_\Omega^h$ such that

$$a_h(u_h, v_h) = L_h(v_h), \quad \forall v_h \in V_\Omega^h,$$

where

$$\begin{aligned} a_h(u_h, v_h) &= (\nabla u_h, \nabla v_h)_\Omega - (\partial_n u_h, v_h)_\Gamma - (u_h, \partial_n v_h)_\Gamma + \left(\frac{\gamma_D}{h} u_h, v_h\right)_\Gamma \\ L_h(v_h) &= (f, v)_\Omega + \left(u_D, \frac{\gamma_D}{h} v_h - \partial_n v_h\right)_\Gamma . \end{aligned}$$

Example 3: CutFEM 求解

This leads to the stabilized cut finite element method, which reads: Find $u_h \in V_\Omega^h$ such that

$$A_h(u_h, v_h) = L_h(v_h), \quad \forall v_h \in V_\Omega^h,$$

where

$$A_h(u_h, v_h) = a_h(u_h, v_h) + g_h(u_h, v_h).$$

```
# 方程弱形式
a(u,v) = ∫( ∇(v)⋅∇(u) ) dΩ +
∫( (γd/h)*v*u  - v*(n_Γd⋅∇(u)) - (n_Γd⋅∇(v))*u ) dΓd +
∫( (γg*h)*jump(n_Γg⋅∇(v))*jump(n_Γg⋅∇(u)) ) dΓg

l(v) = ∫( v*f ) dΩ +
∫( (γd/h)*v*g - (n_Γd⋅∇(v))*g ) dΓd
```

Example 4: 一个更复杂的例子

```
R = 0.5
geo1 = cylinder(R,v=VectorValue(1,0,0))
geo2 = cylinder(R,v=VectorValue(0,1,0))
geo3 = cylinder(R,v=VectorValue(0,0,1))
geo4 = union(union(geo1,geo2),geo3,name="source")

geo5 = sphere(1)
geo6 = cube(L=1.5)
geo7 = intersect(geo6,geo5)
geo8 = setdiff(geo7,geo4,name="csg")
```

Example 4: 一个更复杂的例子

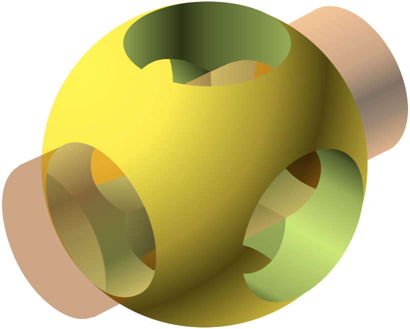


Figure 13: 几何定义

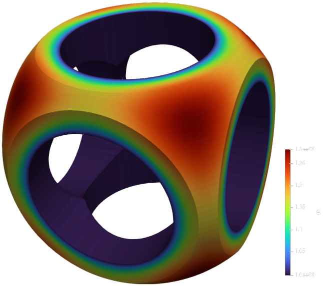


Figure 14: 分析结果

Example 4: 一个更复杂的例子

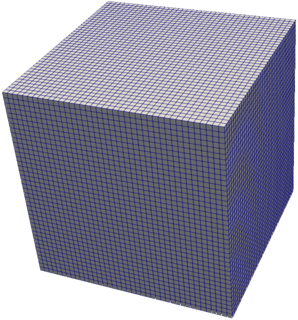


Figure 15: 背景网格

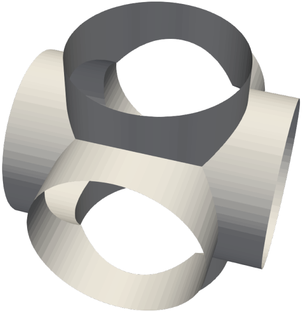


Figure 16: 边界条件

Part III: 线弹性结构有限元

线弹性有限元

Let Ω be a domain in $\mathbb{R}^d$, $d = 2$ or 3, with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, and exterior unit normal $\mathbf{n}$. We consider the following problems:
Find the displacement $\mathbf{u} : \Omega \to \mathbb{R}^d$ such that

$$\begin{aligned} -\nabla \cdot \sigma(\mathbf{u}) &= \mathbf{f} && \text{in } \Omega \\ \sigma(\mathbf{u}) \cdot \mathbf{n} &= \mathbf{T} && \text{on } \Gamma_N \\ \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_D \end{aligned}$$

where the strain and stress tensors are defined by

$$\varepsilon(\mathbf{u}) \doteq \frac{1}{2}\left(\nabla \cdot \mathbf{u} + (\nabla \cdot \mathbf{u})^{\mathrm{T}}\right)$$

$$\sigma(\mathbf{u}) \doteq 2\mu\varepsilon(\mathbf{u}) + \lambda \mathrm{tr}(\varepsilon(\mathbf{u}))$$

with Lamé parameters λ and μ.

线弹性有限元

Let $\mathbf{V}_g = \left\{\mathbf{u} \in \mathbf{H}^1(\Omega) : \mathbf{u} = \mathbf{g} \text{ on } \Gamma_D\right\}$, and define the bilinear form

$$a(\mathbf{u}, \mathbf{v}) = 2\mu \left(\varepsilon(\mathbf{u}), \varepsilon(\mathbf{v})\right)_\Omega + \lambda \left(\mathrm{tr}(\varepsilon(\mathbf{u})), \mathrm{tr}(\varepsilon(\mathbf{v}))\right)_\Omega$$

Find such that

$$a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}_0$$

where the linear form on right hand side is defined by

$$l(\mathbf{v}) = (\mathbf{f}, \mathbf{v})_\Omega + (\mathbf{T}, \mathbf{v})_{\Gamma_N}$$

Example 5: 悬臂梁

Listing 5: 悬臂梁

```julia
using Gridap
using GridapGmsh

# 读取网格文件
model = GmshDiscreteModel("./src/beam.msh")
```

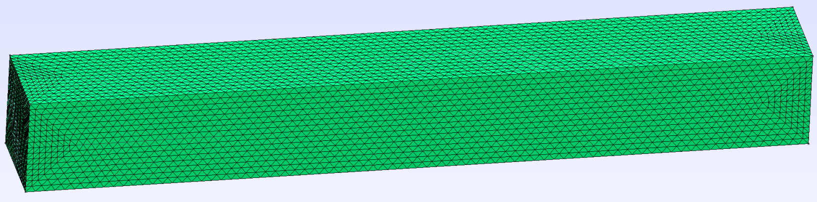


Figure 17: 网格划分

Example 5: 悬臂梁

Listing 6: 悬臂梁

```
# 参考单元
order = 1
reffe = ReferenceFE(lagrangian,VectorValue{3,Float64},order)

V = TestFESpace(model,reffe; conformity=:H1,
dirichlet_tags=["fixed"],
dirichlet_masks=[(true,true,true)])

g(x) = VectorValue(0.0,0.0,0.0)
U = TrialFESpace(V, g)

degree = 2*order
Ω = Triangulation(model)
dΩ = Measure(Ω,degree)
```

Example 5: 悬臂梁

Listing 7: 悬臂梁

```
# 材料参数及本构关系
const E = 210.0
const ν = 0.3
const λ = (E*ν)/((1+ν)*(1-2*ν))
const μ = E/(2*(1+ν))

σ(ε) = λ*tr(ε)*one(ε) + 2*μ*ε
```

Example 5: 悬臂梁

Listing 8: 悬臂梁

```
# 弱形式
a(u,v) = ∫( ε(v) ⊙ (σ $\circ$ ε(u)) )dΩ

f(x) = VectorValue(0.0,-9.8,0.0)
l(v) = ∫(v ⋅ f)dΩ

# 求解及输出结果
op = AffineFEOperator(a,l,U,V)
uh = solve(op)
```

Example 5: 悬臂梁

Figure 18: 一般方法

Define the stabilized Nitsche form

$$\mathcal{A}_h(\mathbf{v},\mathbf{w}) = a_h(\mathbf{v},\mathbf{w}) - (\sigma(\mathbf{v})\cdot\mathbf{n},\mathbf{w})_{\Gamma_D} - (\mathbf{v},\sigma(\mathbf{w})\cdot\mathbf{n})_{\Gamma_D} + \beta h^{-1} b_h(\mathbf{v},\mathbf{w})$$

where $\beta > 0$ is a parameter and

$$b_h(\mathbf{v},\mathbf{w}) = 2\mu\,(\mathbf{v},\mathbf{w})_{\Gamma_D} + \lambda\,(\mathbf{v}\cdot\mathbf{n},\mathbf{w}\cdot\mathbf{n})_{\Gamma_D}$$

线弹性有限元

Find $\mathbf{u}_h \in \mathbf{V}_h$ such that

$$\mathcal{A}_h(\mathbf{u}_h, \mathbf{v}) = \mathcal{L}_h(\mathbf{v}), \quad \forall \mathbf{v}_h \in \mathbf{V}_h$$

where the right hand side is given by

$$\mathcal{L}_h(\mathbf{v}) = (\mathbf{f}, \mathbf{v})_\Omega + (\mathbf{T}, \mathbf{v})_{\Gamma_N} - (\mathbf{g}, \sigma(\mathbf{v}) \cdot \mathbf{n})_{\Gamma_D} + \beta h^{-1} b_h \left(\mathbf{g}, \mathbf{v}\right)_{\Gamma_D}.$$

Example 6: Nitsche 法求解悬臂梁

Listing 9: 悬臂梁

```
Ω = Triangulation(model)
dΩ = Measure(Ω, 2)

Γd = BoundaryTriangulation(model, tags="fixed")
dΓd = Measure(Γd, 2)
n_Γd = get_normal_vector(Γd)

# 参考单元
order = 1
reffe = ReferenceFE(lagrangian, VectorValue{3,Float64}, order)

V = TestFESpace(model, reffe; conformity=:H1)
U = TrialFESpace(V)
```

Example 6: Nitsche 法求解悬臂梁

```
β = 100
h = 1.0 / 400

b(v, w) = 2 * μ * ∫(v ⋅ w)dΓd + λ * ∫((v ⋅ n_Γd) * (w ⋅ n_Γd))dΓd
# 弱形式
a(u, v) = ∫(ε(v) ⊙ (σ ∘ ε(u)))dΩ - ∫(v ⋅ (σ ∘ ε(u)) ⋅ n_Γd)dΓd - ∫(u ⋅ (σ ∘ ε(v)) ⋅ n_Γd)dΓd + β / h * b(u, v)

f(x) = VectorValue(0.0, -9.8, 0.0)
g(x) = VectorValue(0.0, 0.0, 0.0)
l(v) = ∫(v ⋅ f)dΩ - ∫(g ⋅ (σ ∘ ε(v)) ⋅ n_Γd)dΓd + β / h * b(g, v)

# 求解及输出结果
op = AffineFEOperator(a, l, U, V)
uh = solve(op)
```

Example 6: Nitsche 法求解悬臂梁

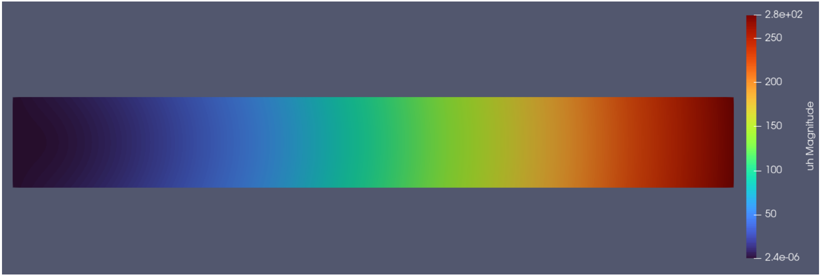


Figure 19: Nitsche 法

线弹性有限元

Let $\mathcal{F}(\partial\Omega)$ in be the set of interior faces that belongs to an element K such that $K \cap \partial\Omega \neq \emptyset$, and define the stabilization form

$$j_h(v, w) = \sum_{F \in \mathcal{F}_h(\partial\Omega)} \sum_{l=1}^{p} h^{2l+1} \left(\left[D^l_{n_F} v \right], \left[D^l_{n_F} w \right] \right)_F$$

where $D^l_{\mathbf{n}_F}$ is the l:th partial derivative in the direction of the normal $\mathbf{n}_F$ to the face $F \in \mathcal{F}(\partial\Omega)$ and $[\cdot]$ denotes the jump in a discontinuous function at F.

Example 7: 悬臂梁

Listing 10: Gmsh 生成网格

```
SetFactory("OpenCASCADE");
Rectangle(1) = {0, 0, 0, 8, 1, 0};

x = 0.0; y = 0.5; r = 0.3;
For i In {1:7}
    x += 1.0;
    Circle(4+i) = {x, y, 0, r, 0, 2*Pi};
    Curve Loop(1+i) = {4+i};
    Plane Surface(1+i) = {1+i};
EndFor
```

Example 7: 悬臂梁

Listing 11: Gmsh 生成网格（续）

```
BooleanDifference(9) = { Surface{1}; Delete; }{ Surface{2}; Delete; };

For i In {1:6}
    BooleanDifference(9+i) = { Surface{8+i}; Delete; }{ Surface{2+i}; Delete; };
EndFor

Physical Surface("domain", 16) = {15};
Physical Curve("fixed", 17) = {2};
Physical Curve("force", 18) = {3};
```

Example 7: 悬臂梁

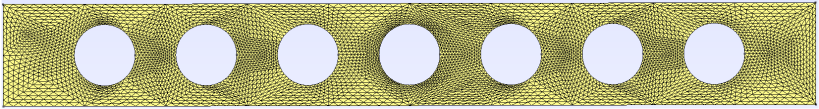


Figure 20: 网格划分

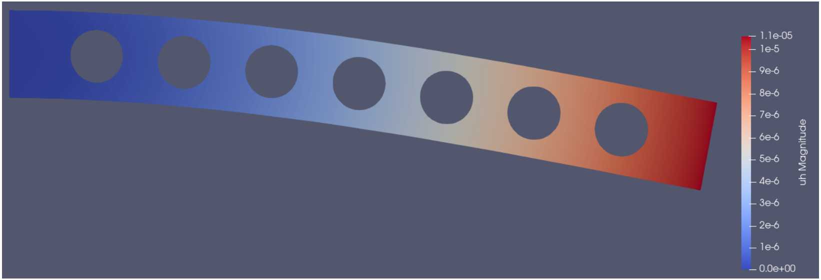


Figure 21: 分析结果

Example 8: CutFEM 求解悬臂梁

Listing 12: 定义几何

```
bgmodel = GmshDiscreteModel("./src/beam2ds.msh")

R = 0.3; x = 0.0; y = 0.5
csg = disk(R,x0=Point(x+1,y))

for dx in range(2,7)
    local geo = disk(R,x0=Point(x+dx,y))

    global csg = union(csg, geo, name="csg")
end

cutgeo = cut(bgmodel, !(csg, name="csg"))

```

Example 8: CutFEM 求解悬臂梁

Listing 13: 定义网格

```
# 1. 物理网格
Ω = Triangulation(cutgeo,PHYSICAL,"csg")
dΩ = Measure(Ω, 2)

# 2. Active网格
Ω_act = Triangulation(cutgeo,ACTIVE,"csg")

# 3. Dirichlet边界条件
Γd = BoundaryTriangulation(bgmodel, tags="fixed")
dΓd = Measure(Γd, 2)

# 4. Neumann边界条件
Γf = BoundaryTriangulation(bgmodel, tags="force")
dΓf = Measure(Γf, 2)
```

Example 8: CutFEM 求解悬臂梁

Listing 14: 定义网格

```
# 5. 嵌入边界
Γ = EmbeddedBoundary(cutgeo,"csg")
dΓ = Measure(Γ, 2)

# 6. Ghost边界
Γg = GhostSkeleton(cutgeo,"csg")
dΓg = Measure(Γg, 2)
n_Γg = get_normal_vector(Γg)
```

Example 8: CutFEM 求解悬臂梁

Listing 15: 有限元空间

```
order = 1
reffe = ReferenceFE(lagrangian,VectorValue{2,Float64},order)
V = TestFESpace(Ω_act, reffe, conformity=:H1, dirichlet_tags=["fixed"])

# Dirichlet values
u0 = VectorValue(0,0)
U = TrialFESpace(V,[u0])
```

Example 8: CutFEM 求解悬臂梁

Listing 16: 定义弱形式

```
# 物理参数
const E = 210.0; const ν = 0.3
const λ = (E*ν)/((1+ν)*(1-2*ν))
const μ = E/(2*(1+ν))

σ(ε) = λ*tr(ε)*one(ε) + 2*μ*ε

# 定义弱形式
γg = 10.0; h = 1.0/400
a(u,v) = ∫( ε(v) ⊙ (σ∘ε(u)) )dΩ + ∫( (γg*h)*jump(∇(v)⋅n_Γg)⋅jump(∇(u)⋅n_Γg) ) * dΓg

T(x) = VectorValue(0.0,-1e3);
l(v) = ∫(v ⋅ T)dΓf
```

Example 8: CutFEM 求解悬臂梁

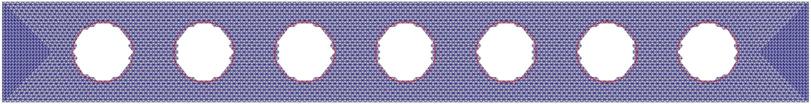


Figure 22: 网格划分

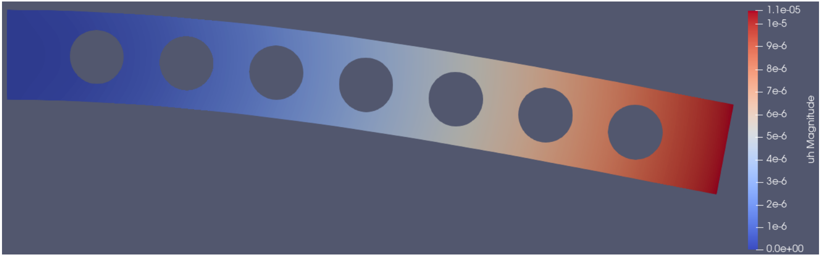


Figure 23: 分析结果

Part IV: 杂项

Part V: 项目规划

结束 Q & A