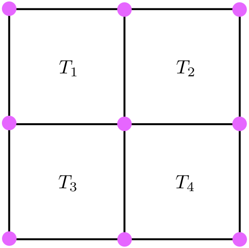
FEniCS Course

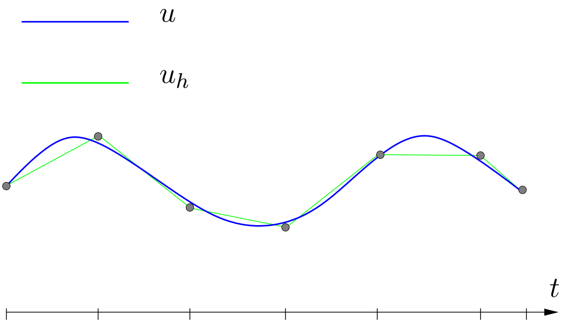
Lecture 10: Discontinuous Galerkin methods for elliptic equations

Contributors
André Massing

FEniCS PROJECT

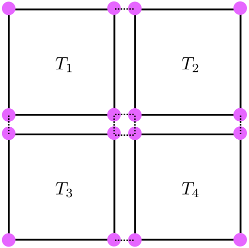
The discontinuous Galerkin (DG) method uses discontinuous basis functions

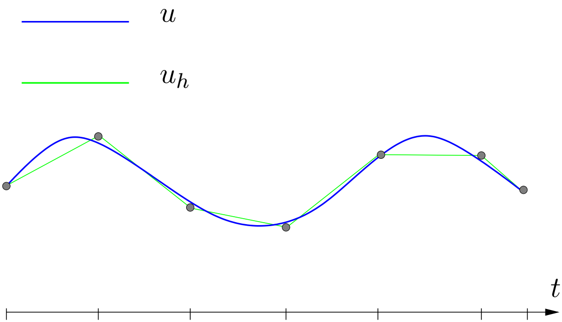




$$V_h = P^k(\mathcal{T}_h) = \{v_h \in L^2(\Omega) : v_h|_T \in P^k(T)\ \forall\, T \in \mathcal{T}_h\}$$

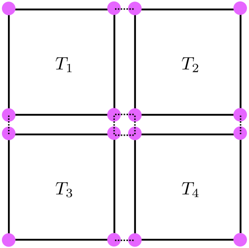
The discontinuous Galerkin (DG) method uses discontinuous basis functions

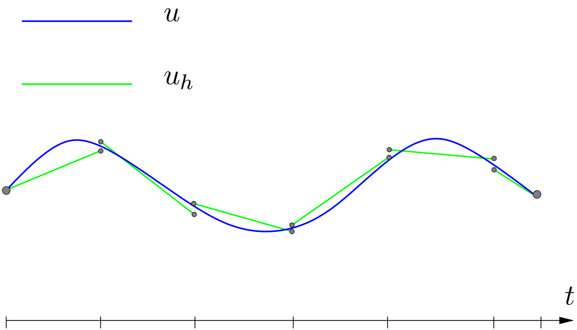




$$V_h = P^k(\mathcal{T}_h) = \{v_h \in L^2(\Omega) : v_h|_T \in P^k(T)\ \forall\, T \in \mathcal{T}_h\}$$

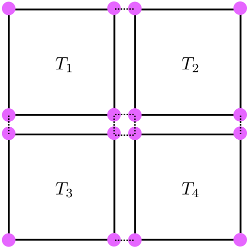
The discontinuous Galerkin (DG) method uses discontinuous basis functions

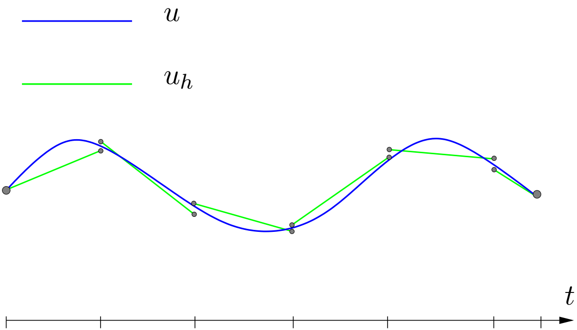




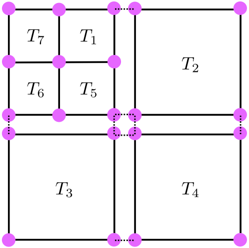
$$V_h = P^k(\mathcal{T}_h) = \{v_h \in L^2(\Omega) : v_h|_T \in P^k(T)\ \forall\, T \in \mathcal{T}_h\}$$

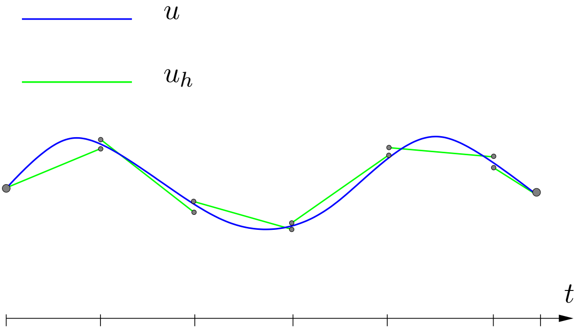
The DG method eases mesh adaptivity



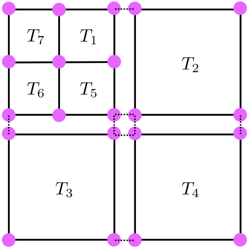


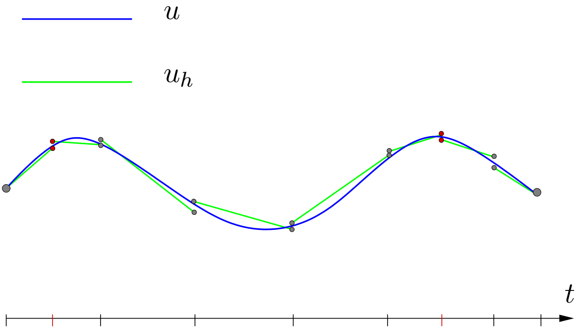
The DG method eases mesh adaptivity



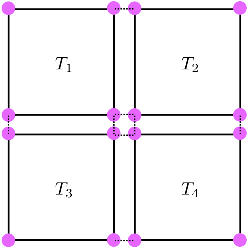


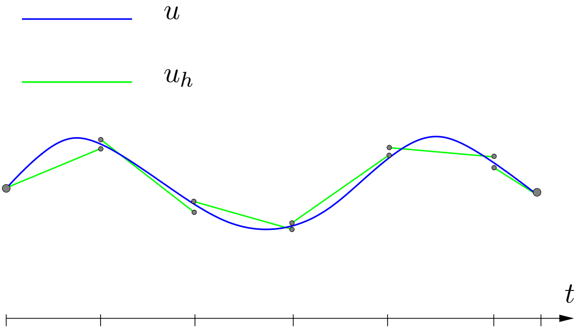
The DG method eases mesh adaptivity



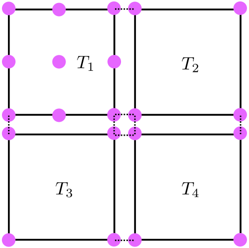


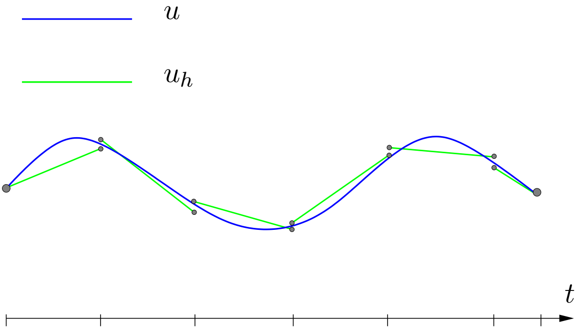
The DG method eases space adaptivity



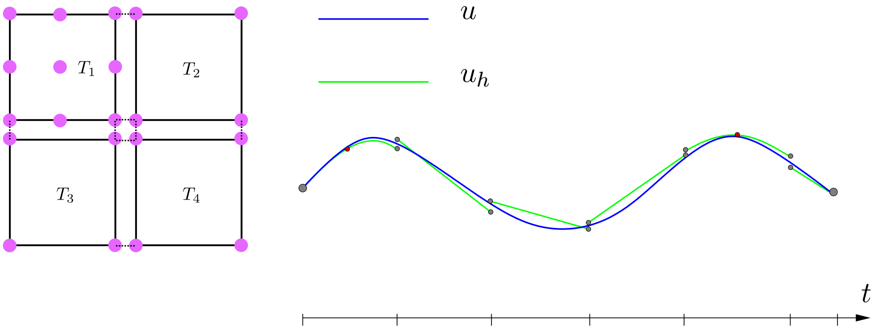


The DG method eases space adaptivity



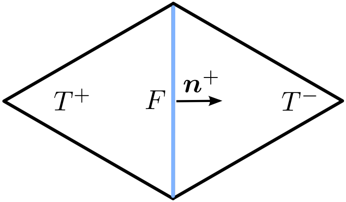


The DG method eases space adaptivity



DG-FEM Notation

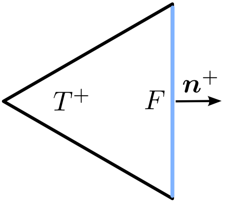
Interface facets



Average $\langle v \rangle = \frac{1}{2}(v^+ + v^-)$

Jump $[v] = (v^+ - v^-)$

Boundary facet



$\langle v \rangle = [v] = v$

Jump identity

$$[(\nabla_h v) w_h] = [\nabla_h v]\langle w_h \rangle + \langle \nabla_h v \rangle [w_h]$$

The symmetric interior penalty method (SIP)

$$a_h(u_h, v_h) = \sum_{T \in \mathcal{T}} \int_T \nabla u_h \cdot \nabla v_h \,\mathrm{d}x - \underbrace{\sum_{F \in \mathcal{F}} \int_F \langle \nabla u_h \rangle \cdot \boldsymbol{n}[v_h] \,\mathrm{d}S}_{\text{Consistency}}$$

$$- \underbrace{\sum_{F \in \mathcal{F}} \int_F \langle \nabla v_h \rangle \cdot \boldsymbol{n}[u_h] \,\mathrm{d}S}_{\text{Symmetry}} + \underbrace{\sum_{F \in \mathcal{F}} \frac{\gamma}{h_F} \int_F [u_h][v_h] \,\mathrm{d}S}_{\text{Penalty}}$$

The symmetric interior penalty method (SIP)

$$a_h(u_h, v_h) = \sum_{T\in\mathcal{T}} \int_T \nabla u_h \cdot \nabla v_h \,\mathrm{d}x - \underbrace{\sum_{F\in\mathcal{F}} \int_F \langle \nabla u_h \rangle \cdot \boldsymbol{n}[v_h] \,\mathrm{d}S}_{\text{Consistency}}$$

$$- \underbrace{\sum_{F\in\mathcal{F}} \int_F \langle \nabla v_h \rangle \cdot \boldsymbol{n}[u_h] \,\mathrm{d}S}_{\text{Symmetry}} + \underbrace{\sum_{F\in\mathcal{F}} \frac{\gamma}{h_F} \int_F [u_h][v_h] \,\mathrm{d}S}_{\text{Penalty}}$$

The symmetric interior penalty method (SIP)

$$a_h(u_h, v_h) = \sum_{T \in \mathcal{T}} \int_T \nabla u_h \cdot \nabla v_h \,\mathrm{d}x - \underbrace{\sum_{F \in \mathcal{F}} \int_F \langle \nabla u_h \rangle \cdot \boldsymbol{n} [v_h] \,\mathrm{d}S}_{\text{Consistency}}$$

$$- \underbrace{\sum_{F \in \mathcal{F}} \int_F \langle \nabla v_h \rangle \cdot \boldsymbol{n} [u_h] \,\mathrm{d}S}_{\text{Symmetry}} + \underbrace{\sum_{F \in \mathcal{F}} \frac{\gamma}{h_F} \int_F [u_h][v_h] \,\mathrm{d}S}_{\text{Penalty}}$$

The symmetric interior penalty method (SIP)

$$a_h(u_h, v_h) = \sum_{T\in\mathcal{T}} \int_T \nabla u_h \cdot \nabla v_h \,\mathrm{d}x - \underbrace{\sum_{F\in\mathcal{F}} \int_F \langle \nabla u_h \rangle \cdot \boldsymbol{n}[v_h]\,\mathrm{d}S}_{\text{Consistency}}$$

$$- \underbrace{\sum_{F\in\mathcal{F}} \int_F \langle \nabla v_h \rangle \cdot \boldsymbol{n}[u_h]\,\mathrm{d}S}_{\text{Symmetry}} + \underbrace{\sum_{F\in\mathcal{F}} \frac{\gamma}{h_F} \int_F [u_h][v_h]\,\mathrm{d}S}_{\text{Penalty}}$$

$$l_h(v_h) = \int_\Omega f v_h \,\mathrm{d}x - \sum_{F\in\mathcal{F}^b} \int_F \langle \nabla v_h \rangle \cdot \boldsymbol{n} g \,\mathrm{d}S + \sum_{F\in\mathcal{F}^b} \frac{\gamma}{h_F} \int_F g v_h \,\mathrm{d}S$$

The symmetric interior penalty method (SIP)

$$a_h(u_h, v_h) = \sum_{T\in\mathcal{T}} \int_T \nabla u_h \cdot \nabla v_h \,\mathrm{d}x - \underbrace{\sum_{F\in\mathcal{F}} \int_F \langle \nabla u_h \rangle \cdot \boldsymbol{n}[v_h] \,\mathrm{d}S}_{\text{Consistency}}$$

$$- \underbrace{\sum_{F\in\mathcal{F}} \int_F \langle \nabla v_h \rangle \cdot \boldsymbol{n}[u_h] \,\mathrm{d}S}_{\text{Symmetry}} + \underbrace{\sum_{F\in\mathcal{F}} \frac{\gamma}{h_F} \int_F [u_h][v_h] \,\mathrm{d}S}_{\text{Penalty}}$$

$$l_h(v_h) = \int_\Omega f v_h \,\mathrm{d}x - \sum_{F\in\mathcal{F}^b} \int_F \langle \nabla v_h \rangle \cdot \boldsymbol{n} g \,\mathrm{d}S + \sum_{F\in\mathcal{F}^b} \frac{\gamma}{h_F} \int_F g v_h \,\mathrm{d}S$$

Split of SIP form into interior and boundary contribution

$$a_h(u_h, v_h) = \sum_{T\in\mathcal{T}} \int_T \nabla u_h \cdot \nabla v_h \,\mathrm{d}x - \underbrace{\sum_{F\in\mathcal{F}^i} \int_F \langle \nabla u_h \rangle \cdot \boldsymbol{n}[v_h] \,\mathrm{d}S}_{\text{Consistency}}$$

$$- \underbrace{\sum_{F\in\mathcal{F}^i} \int_F \langle \nabla v_h \rangle \cdot \boldsymbol{n}[u_h] \,\mathrm{d}S}_{\text{Symmetry}} + \underbrace{\sum_{F\in\mathcal{F}^i} \frac{\gamma}{h_F} \int_F [u_h][v_h] \,\mathrm{d}S}_{\text{Penalty}}$$

$$- \underbrace{\sum_{F\in\mathcal{F}^b} \int_F \nabla u_h \cdot \boldsymbol{n} v_h \,\mathrm{d}s}_{\text{Consistency}} - \underbrace{\sum_{F\in\mathcal{F}^b} \int_F \nabla v_h \cdot \boldsymbol{n} u_h \,\mathrm{d}s}_{\text{Symmetry}}$$

$$+ \underbrace{\sum_{F\in\mathcal{F}^b} \frac{\gamma}{h_F} \int_F u_h v_h \,\mathrm{d}s}_{\text{Penalty}}$$

Useful FEniCS tools (I)

Access facet normals and local mesh size:

```
n = FacetNormal(mesh)
h = CellSize(mesh)
```

Restriction:

```
f = Function(V)
f('+')
grad(f)('+')
```

Useful FEniCS tools (II)

Average and jump:

```
# define it yourself
h_avg = (h('+') + h('-'))/2
# or use built-in expression
avg(h)
jump(v)
jump(v, n)
```

Integration on interior facets:

```
... *dS
alpha/h_avg*dot(jump(v, n), jump(u, n))*dS
```

Exercise

Solve our favorite Poisson problem given

- Domain:

$$\Omega = [0,1] \times [0,1], \qquad \partial\Omega_D = \partial\Omega$$

- Source and boundary values:

$$f(x,y) = 200\cos(10\pi x)\cos(10\pi y)$$

$$g_D(x,y) = \cos(10\pi x)\cos(10\pi y)$$

Mission: Solve this PDE numerically by using the SIP method. Print the errornorm for both the L^2 and the H^1 norm for various mesh sizes. For a `UnitSquareMesh(128,128)` the error should be 0.0009166 and 0.1962, respectively.

Extra mission: Implement the NIP variant, solve the same problem and compare the H^1 and L^2 error for a range of meshes `UnitSquareMesh(N,N)`, $N = 2^j, j = 2, \cdots, 7$. Can you determine the order of convergence?