Advanced Numerical Concepts: Discontinuous Galerkin Method and Adaptive Mesh Refinement

Natasha Sharma

Department of Mathematical Sciences,
University of Texas at El Paso

Computational Methods for PDEs Summer School
Colorado State University, August 3-9, 2019

Outline of this session

1. Discontinuous Galkerin (DG) Method

2. Adaptivity

Outline of this session

1. Discontinuous Galkerin (DG) Method
    1. Overview of Existing Schemes
2. Adaptivity

Outline of this session

1. Discontinuous Galkerin (DG) Method
   1. Overview of Existing Schemes
   2. Why DG?

2. Adaptivity

Outline of this session

1. Discontinuous Galkerin (DG) Method
   1. Overview of Existing Schemes
   2. Why DG?
   3. DG Formulation
2. Adaptivity

Overview of Existing Schemes

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega.$$

The choice of our numerical scheme is motivated by:

Overview of Existing Schemes

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega.$$

The choice of our numerical scheme is motivated by:

1. The way we want to approximate u.

Overview of Existing Schemes

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega.$$

The choice of our numerical scheme is motivated by:

1. The way we want to approximate u.
2. How do want this approximate solution to satisfy the above equation?

Overview of Existing Schemes

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega.$$

The choice of our numerical scheme is motivated by:

1. The way we want to approximate u.
2. How do want this approximate solution to satisfy the above equation?

One natural choice is finite difference method.

Overview of Existing Schemes

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega.$$

The choice of our numerical scheme is motivated by:

1. The way we want to approximate u.
2. How do want this approximate solution to satisfy the above equation?

One natural choice is finite difference method.

1. The local approximation is a polynomial.

Overview of Existing Schemes

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega.$$

The choice of our numerical scheme is motivated by:

1. The way we want to approximate u.
2. How do want this approximate solution to satisfy the above equation?

One natural choice is finite difference method.

1. The local approximation is a polynomial.
2. The PDE is satisfied at grid points x_i.

Overview of Existing Schemes

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega.$$

The choice of our numerical scheme is motivated by:

1. The way we want to approximate u.
2. How do want this approximate solution to satisfy the above equation?

One natural choice is finite difference method.

1. The local approximation is a polynomial.
2. The PDE is satisfied at grid points x_i.

Easy to implement, well established theory.

Overview of Existing Schemes

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega.$$

The choice of our numerical scheme is motivated by:

1. The way we want to approximate u.
2. How do want this approximate solution to satisfy the above equation?

One natural choice is finite difference method.

1. The local approximation is a polynomial.
2. The PDE is satisfied at grid points x_i.

Easy to implement, well established theory. Cannot handle complex geometries!

Finite Volume Method

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega = (0,1)$$

Finite Volume Method

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega = (0, 1)$$

1. The local approximation is a cell average of the form:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} u_i \; dx = h_i \bar{u}_i,$$

Finite Volume Method

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega = (0,1)$$

1. The local approximation is a cell average of the form:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} u_i \; dx = h_i \bar{u}_i,$$

2. The PDE is satisfied on the conservative form.

$$h_i \frac{d\bar{u}_i}{dt} + F_{i+1/2} - F_{i-1/2} = 0$$

where $F_{i+1/2} = F(u_{i+1/2}),\; u_{i+1/2} = \frac{1}{2}(\bar{u}_{i+1} + \bar{u}_i)$.

Finite Volume Method

$$\partial_t u + \partial_x F(u) = 0 \text{ in } \Omega = (0,1)$$

1. The local approximation is a cell average of the form:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} u_i \, dx = h_i \bar{u}_i,$$

2. The PDE is satisfied on the conservative form.

$$h_i \frac{d\bar{u}_i}{dt} + F_{i+1/2} - F_{i-1/2} = 0$$

where $F_{i+1/2} = F(u_{i+1/2}),\ u_{i+1/2} = \frac{1}{2}(\bar{u}_{i+1} + \bar{u}_i)$.
Good on Complex Geometry, local conservative property. But bad for adaptivity.

Finite Element Method

Finite Element Method

1. The approximation is represented in globally:

$$u_h(x) = \sum_{j=1}^{N} u(x_j)\phi_j$$

Finite Element Method

1. The approximation is represented in globally:

$$u_h(x) = \sum_{j=1}^{N} u(x_j)\phi_j$$

2. The PDE is satisfied globally.

$$\int_{\Omega} \Big(\partial_t u + \partial_x F(u)\Big)\phi_j \; dx = 0 \text{ in } \Omega.$$

Finite Element Method

1. The approximation is represented in globally:

$$u_h(x) = \sum_{j=1}^{N} u(x_j)\phi_j$$

2. The PDE is satisfied globally.

$$\int_{\Omega} \Big(\partial_t u + \partial_x F(u)\Big)\phi_j \; dx = 0 \text{ in } \Omega.$$

Good on Complex Geometry, High-order accuracy. Performs poorly for certain pdes.

DG

What DG combines is:

DG

What DG combines is:

1. local high-order/flexible of FEM in the approximation of u.

DG

What DG combines is:

1. local high-order/flexible of FEM in the approximation of u.
2. Local conservative representation for the PDE as seen for FVM.

DG in 1D

We consider a partition of Ω into intervals. Then our approximation locally assumes the form

$$u_h(x,t) = \sum_{i=1}^{N} u_i(x_i,t)\phi_i(x), \quad x \in T.$$

$$\int_T \partial_t u_h \phi_j \; dx + \int_T \partial_x F(u_h)\phi_j \; dx = 0$$

DG in 1D

We consider a partition of Ω into intervals. Then our approximation locally assumes the form

$$u_h(x,t) = \sum_{i=1}^{N} u_i(x_i,t)\phi_i(x), \quad x \in T.$$

$$\int_T \partial_t u_h \phi_j \; dx + \int_T \partial_x F(u_h)\phi_j \; dx = 0$$

Satisfies the two desirable properties but now all the elements are disconnected.

DG in 1D

We consider a partition of Ω into intervals. Then our approximation locally assumes the form

$$u_h(x,t) = \sum_{i=1}^{N} u_i(x_i,t)\phi_i(x), \quad x \in T.$$

$$\int_T \partial_t u_h \phi_j \; dx + \int_T \partial_x F(u_h)\phi_j \; dx = 0$$

Satisfies the two desirable properties but now all the elements are disconnected.

$$\int_T \partial_t u_h \phi_j \; dx - \int_T F(u_h)\partial_x \phi_j \; dx = -F^* \phi_j \Big|_{x_i}^{x_{i+1}}$$

DG in 1D

We consider a partition of Ω into intervals. Then our approximation locally assumes the form

$$u_h(x,t) = \sum_{i=1}^{N} u_i(x_i,t)\phi_i(x), \quad x \in T.$$

$$\int_T \partial_t u_h \phi_j \; dx + \int_T \partial_x F(u_h)\phi_j \; dx = 0$$

Satisfies the two desirable properties but now all the elements are disconnected.

$$\int_T \partial_t u_h \phi_j \; dx - \int_T F(u_h)\partial_x \phi_j \; dx = -F^* \phi_j \Big|_{x_i}^{x_{i+1}}$$

The choice of flux F^* here is crucial to guarantee other desirable properties of the scheme!

Numerical Example: 1D Wave Equation

$$\partial_t u - 2\pi \partial_x u = 0, \quad u(x,0) = \sin\left(kx\right), \ k = \frac{2\pi}{\lambda}, \quad \Omega = [0, 2\pi].$$

$p \backslash K$	2	4	8	16	32	64	Order
1	–	4.0e-1	9.1e-2	2.3e-02	5.7e-03	1.4e-03	2.0
2	2.0e-01	4.3e-02	6.3e-03	8.0e-04	1.0e-04	1.3e-05	3.0
4	3.3e-03	3.1e-04	9.9e-06	3.2e-07	1.0e-08	3.3e-10	5.0
8	2.1e-07	2.5e-09	4.8e-12	2.2e-13	5.0e-13	6.6e-13	$\approx$ 9.0

Table: Global L^2 errors with varying polynomial degree p and using K elements.

The error behaves like

$$\|u - u_h\| \leq Ch^{p+1}$$

Lessons from DG in 1D:

- Stability is enforced by the flux choice

Lessons from DG in 1D:

- Stability is enforced by the flux choice
- No restrictions on local basis it need not even be polynomial and is chosen to provide accuracy.

Lessons from DG in 1D:

- Stability is enforced by the flux choice
- No restrictions on local basis it need not even be polynomial and is chosen to provide accuracy.
- Numerical solution is discontinuous between elements.

Lessons from DG in 1D:

- Stability is enforced by the flux choice
- No restrictions on local basis it need not even be polynomial and is chosen to provide accuracy.
- Numerical solution is discontinuous between elements.
- Boundary conditions and interface conditions are enforced weakly.

Lessons from DG in 1D:

- Stability is enforced by the flux choice
- No restrictions on local basis it need not even be polynomial and is chosen to provide accuracy.
- Numerical solution is discontinuous between elements.
- Boundary conditions and interface conditions are enforced weakly.
- All operators are local.

Lessons from DG in 1D:

- Stability is enforced by the flux choice
- No restrictions on local basis it need not even be polynomial and is chosen to provide accuracy.
- Numerical solution is discontinuous between elements.
- Boundary conditions and interface conditions are enforced weakly.
- All operators are local.

The scheme is very flexible!

What is DG?

1. The standard Galerkin Method is also referred to as Continuous Galkerin Method (CG).
2. The DG Method is a variant of the CG Method by relaxing these continuity requirements.
3. DG methods have more degrees of freedom to solve than CG methods.

What we gain by relaxing continuity...

1. Assembly of stiffness matrix is easier to implement. Easy to parallelize.
2. Adaptivity is more flexible
3. Locally conservative property
4. Thanks to the natural hierarchy, it is suitable for multilevel methods integrated into the solvers.
5. Supports higher order local approximations that can vary nonuniformly over the mesh.

DG for Elliptic Problems

Let $\Omega \subset \mathbb{R}^d$ be a bounded, open polygonal domain with Lipschitz continuous boundary Γ.

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= u_D \quad \text{on } \Gamma_D, \\ \mathbf{n} \cdot \nabla u &= u_N \quad \text{on } \Gamma_N, \end{aligned}$$

where $\Gamma = \Gamma_D \cup \Gamma_N$ and $\mathbf{n}$ denotes the outward normal to Ω.

DG Method

Let $\mathbb{T}_h$: uniform partition of Ω into cells. Introduce our DG space:

$$\mathrm{V}_h := \{\mathrm{v}_h : \bar{\Omega} \to \mathbb{R} \mid \mathrm{v}_h|_T \in \mathcal{P}_k(T), T \in \mathbb{T}_h\}$$

Multiplying by $v \in \mathrm{V}_h$ integrating over each $T \in \mathbb{T}_h$ and summing over all possible cells we arrive at:

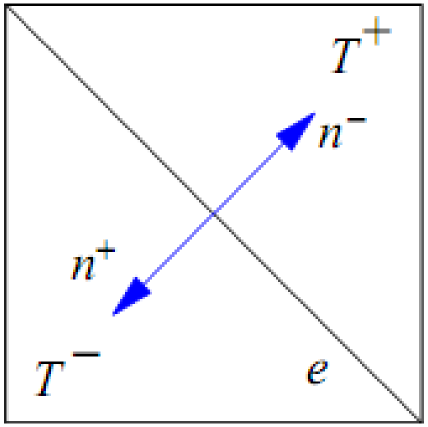
$$-\sum_{T \in \mathbb{T}_h} \int_T \Delta u v \; dx = \sum_{T \in \mathbb{T}_h} \int_T fv \; dx$$

$$\sum_{T \in \mathbb{T}_h} \left(\int_T \nabla u \cdot \nabla v \; dx - \int_{\partial T} \nabla u \cdot \mathbf{n} v \; ds \right) = \sum_{T \in \mathbb{T}_h} \int_T fv \; dx$$

DG Method

$$\int_{\partial T} \nabla u \cdot \mathbf{n} v \; ds = \int_{e} \nabla u^{+} \cdot \mathbf{n}^{+} v + \nabla u^{-} \cdot \mathbf{n}^{-} v \; ds$$

$$= \int_{e} \nabla u^{+} \cdot \mathbf{n}^{+} v + \nabla u^{-} \cdot \mathbf{n}^{-} v \; ds$$



$$\sum_{T\in\mathbb{T}_h}\int_{\partial T}\nabla u\cdot\mathbf{n}v\;ds=\sum_{e\in\mathbf{E}_h^i}\int_e\left(\nabla u^+\cdot\mathbf{n}^+v+\nabla u^-\cdot\mathbf{n}^-v\right)\;ds+$$

$$\sum_{e\in\Gamma_D}\int_e\nabla u\cdot\mathbf{n}v\;ds+\sum_{e\in\Gamma_N}\int_e g_Nv\;ds.$$

Resulting in

$$\sum_{T\in\mathbb{T}_h}\int_T\nabla u\cdot\nabla v\;dx-\sum_{e\in\mathbf{E}_h^i}\int_e\left(\nabla u^+\cdot\mathbf{n}^+v+\nabla u^-\cdot\mathbf{n}^-v\right)\;ds$$

$$+\sum_{e\in\Gamma_D}\int_e\nabla u\cdot\mathbf{n}v\;ds=\sum_{T\in\mathbb{T}_h}\int_T fv\;dx+\sum_{e\in\Gamma_N}\int_e g_Nv\;ds$$

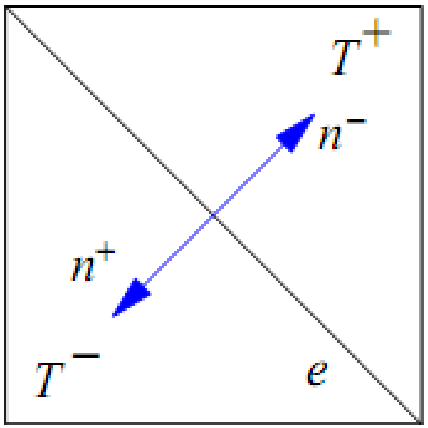
DG Method

$$\nabla u^+ \cdot \mathbf{n}^+ v + \nabla u^- \cdot \mathbf{n}^- v = \{\frac{\partial u}{\partial \mathbf{n}}\}_e [v]_e$$

where,

$$\left[v\right]_e = v^+ - v^- \qquad \text{(jump of v)}$$

$$\{\frac{\partial v}{\partial \mathbf{n}}\}_e = \frac{1}{2}(\nabla v^- \cdot \mathbf{n}^- + \nabla v^+ \cdot \mathbf{n}^+) \qquad \text{(average of v)}$$



Penalty Method

Find $u_h \in V_h$, such that

$$a_h^{IP}(u_h, v_h) = \ell(v_h)\ dx \quad \forall v_h \in V_h$$

where

$$\ell(v_h) = \int_\Omega f v_h\ dx + \int_{\Gamma_N} g_N v\ ds - \int_{\Gamma_D} g_D \frac{\partial v_h}{\partial \mathbf{n}}\ ds + \frac{\alpha}{|e|}\int_{\Gamma_D} g_D v_h\ ds$$

$$\begin{aligned} a_h^{IP}(u_h, v_h) := &\sum_{T \in \mathcal{T}_h} \int_T \nabla u_h \cdot \nabla v_h\ dx - \sum_{e \in \mathcal{E}_h} \int_e \{\frac{\partial u_h}{\partial \mathbf{n}}\}_e \cdot [v_h]_e\ ds \\ &- \sum_{e \in \mathcal{E}_h} \int_e [u_h]_e \cdot \{\frac{\partial v_h}{\partial \mathbf{n}}\}_e\ ds + \sum_{e \in \mathcal{E}_h} \frac{\alpha}{|e|} \int_e [u_h]_e \cdot [v_h]_e\ ds \end{aligned}$$

where $\alpha > 0$ is a suitably chosen penalty parameter.

Well Posedness

Lemma (Coercivity and Continuity)

For penalty parameter $\alpha > \alpha_0 \geq 1$, we have

$$a_h^{IP}(u,v) \leq C_{cont} |||u||| \, |||u|||,$$
$$C_{coer} |||u|||^2 \leq a_h^{IP}(u,u).$$

where

$$|||v|||^2 = \sum_{T \in \mathbb{T}_h} \|\nabla v\|_{0,T}^2 + \sum_{e \in \mathbf{E}_h} \frac{\alpha}{|e|} \|[\mathrm{u}_h]\|_{0,e}^2.$$

A priori Estimates

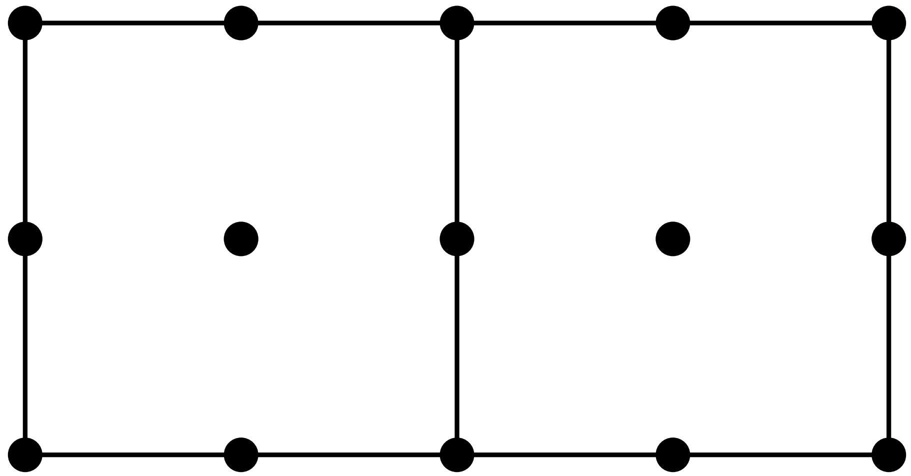
Theorem

For $u \in H^s(\mathbb{T}_h),\ s > 3/2$ and for penalty parameter large enough, we can find a constant $C > 0$ independent of h such that the following optimal a priori estimates hold:

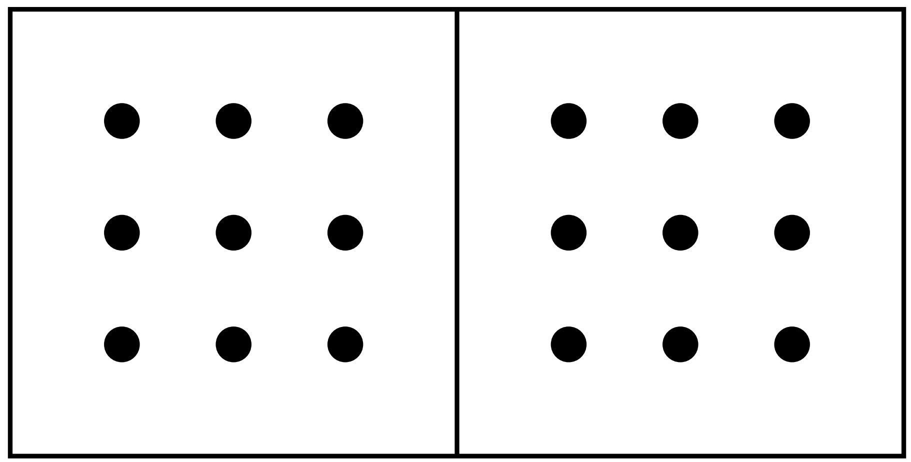
$$|||u - u_h|||^2 \leq Ch^{\min\{k+1,s\}-1}|||v|||^2_{H^s}$$

deal.II Implementation

```
FE_Q<dim> fe_cg;
```



```
FE_DGQ<dim> fe_dg;
```



How does deal.II do it?

For `assemble_system()`, we want to code:

$$a_h^{IP}(u_h, v_h) := \sum_{T\in\mathcal{T}_h} \int_T \nabla u_h \cdot \nabla v_h \; dx - \sum_{e\in\mathcal{E}_h} \int_e \{\frac{\partial u_h}{\partial \mathbf{n}}\}_e [v_h]_e \; ds$$

$$- \sum_{e\in\mathcal{E}_h} \int_e [u_h]_e \{\frac{\partial v_h}{\partial \mathbf{n}}\}_e \; ds + \sum_{e\in\mathcal{E}_h} \frac{\alpha}{|e|} \int_e [u_h]_e [v_h]_e \; ds$$

$$\ell(v_h) = \int_\Omega f v_h \; dx + \int_{\Gamma_N} g_N v \; ds - \int_{\Gamma_D} g_D \frac{\partial v_h}{\partial \mathbf{n}} \; ds + \frac{\alpha}{|e|} \int_{\Gamma_D} g_D v_h \; ds$$

1. `LocalIntegrators::Laplace::cell(dinfo,info)`
2. `LocalIntegrators::Laplace::boundary(dinfo,info)`
3. `LocalIntegrators::Laplace::face(dinfo1,dinfo2,info1,info2)`

`MeshWorker::integration_loop(···)` assembles all the local contributions into the global matrix.
Look into `step-39.cc` for more details!

ADAPTIVE FINITE ELEMENT METHODS

Need for adaptivity

In physical problems, we are faced with two challenges:

1. No knowledge about exact solution of the problem.

Need for adaptivity

In physical problems, we are faced with two challenges:

1. No knowledge about exact solution of the problem.

   Derive computable quantities η_h depending only on "data" of the problem which provide an upper and lower bound for approximation error.

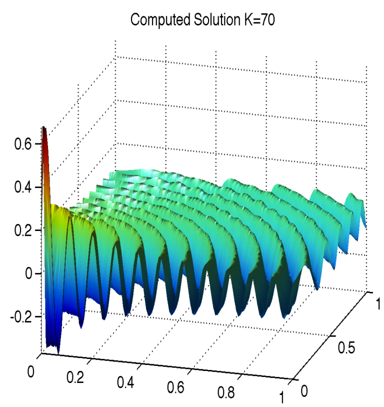
Need for adaptivity

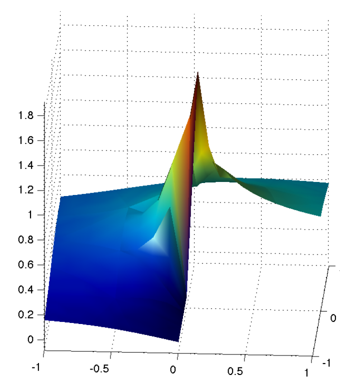
In physical problems, we are faced with two challenges:

1. No knowledge about exact solution of the problem.

   Derive computable quantities η_h depending only on “data” of the problem which provide an upper and lower bound for approximation error.

2. Solutions which are not so smooth.





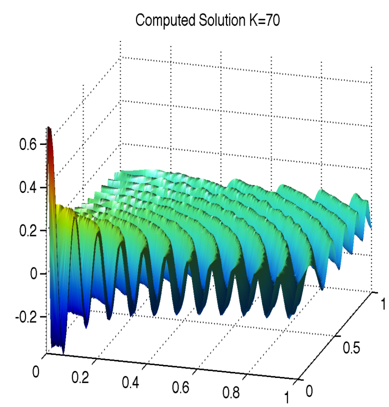
Need for adaptivity

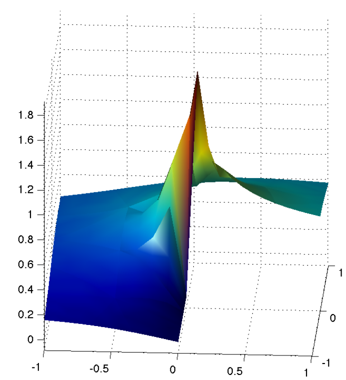
In physical problems, we are faced with two challenges:

1. No knowledge about exact solution of the problem.

   Derive computable quantities η_h depending only on "data" of the problem which provide an upper and lower bound for approximation error.

2. Solutions which are not so smooth.





Devise an algorithm Solve-Estimate-Mark-Refine that efficiently identifies regions where the solution requires better resolution.

Adaptive Algorithm

The implementation of the adaptive algorithm is done according to the cycle:

SOLVE $\Longrightarrow$ ESTIMATE $\Longrightarrow$ MARK $\Longrightarrow$ REFINE

- SOLVE amounts to seeking a solution $u_h \in V_h$ st.

$$a_h(u_h, v_h) = \ell(v_h) \quad v_h \in V_h \quad \text{holds.}$$

- ESTIMATE the computation of η_h based on

$$\mathcal{R}(v) = \ell(v) - a_h(u_h, v)$$

- MARK deals with the selection of the elements and edges for refinement based on $0 < \theta < 1$

$$\theta \; \eta_h \;\lesssim \eta_{\mathcal{M}}$$

- REFINEMENT realizes the subdivision of cells.

Desirable Properties of the estimator

The estimator η_h consists of

$$\eta_h = \sum_{T \in \mathbb{T}_h} \eta_T + \sum_{e \in \mathbf{E}_h} \eta_e$$

E.g: For Poisson Equation, the residual based estimator has cell and edge terms like:

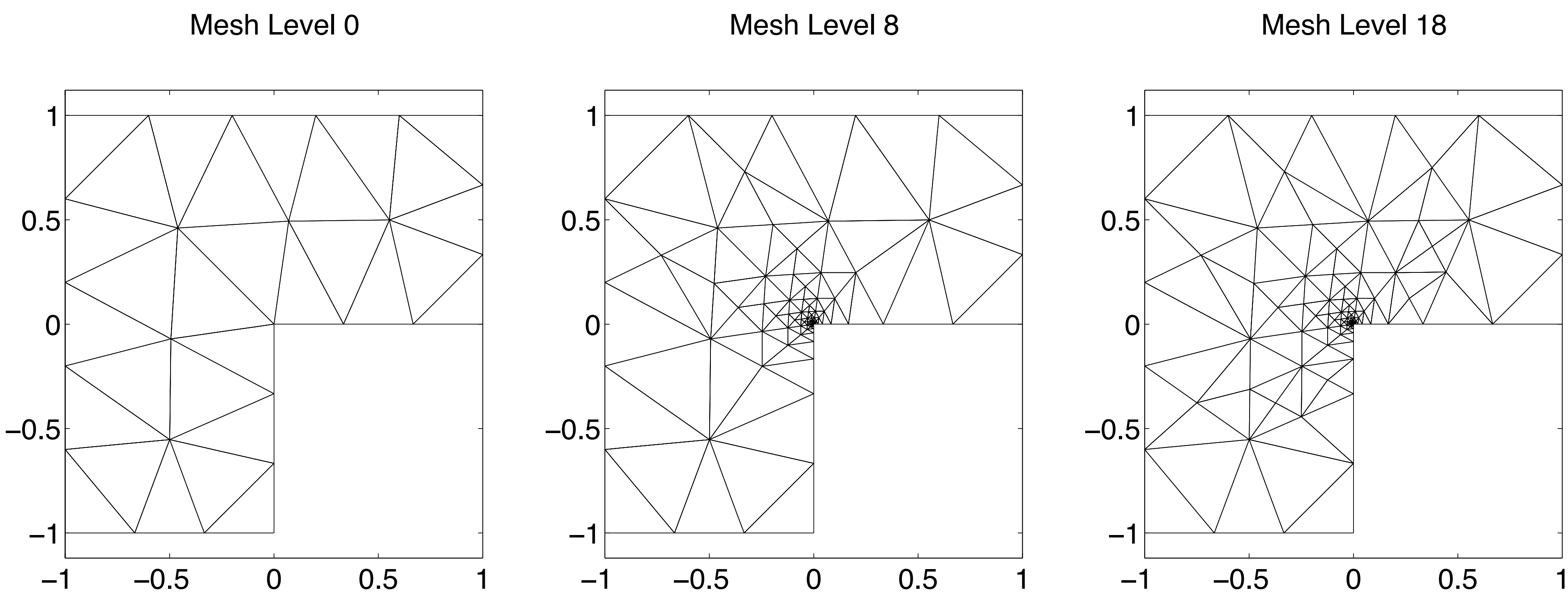
$$\eta_T = h_T ||f - (-\Delta u_h)||_{0,T}, \quad \eta_e = h_e^{1/2} \|[\partial_n u_h]\|_{0,e}$$

- Cheaply computable estimator
- η_h must be reliable, i.e,
$$\|u - u_h\| \leq C_{rel} \eta_h.$$
- η_h must be efficient, i.e,
$$C_{eff} \eta_h \leq \|u - u_h\|.$$

Notorious L-Shaped Domain

$$
\begin{aligned}
-\Delta u &= 0 \quad \text{in } \Omega, \\
u &= 0 \quad \text{on } \Gamma_D := \{0\} \times [-1,0] \cup [0,1] \times \{0\}, \\
\mathbf{n} \cdot \nabla u &= u_N \quad \text{on } \Gamma_N := \Gamma \setminus \Gamma_D,
\end{aligned}
$$

where the data is chosen according to $u(r,\phi) = grad(r^{\frac{2}{3}} \sin(\frac{2}{3}\phi))$.



$\theta = 0.1, k = 4$

Moving Peak

$$\frac{\partial u}{\partial t} - \Delta u = f \quad \text{in} \quad I \times \Omega,$$
$$u = 0 \quad \text{on} \quad I \times \partial\Omega,$$
$$u(0,.) = u_0(.) \quad \text{on} \quad \Omega$$

where $I = [0,1]$ and we choose data according to

$$u(\mathbf{x}, t) = g(t) \exp^{\beta((x_1-0.5)^2+(x_2-0.5)^2)}$$

- $\beta = 25$,
- $g(t) = 1 - \exp^{-\gamma(t-0.5)^2}$, $\gamma = 10$.

Numerical results: Adaptive algorithm

Moving peak

(Loading Video...)