Summer school Hasselt University, Belgium
Adaptive methods and a posteriori error estimation June 24–27, 2024
Web page: link https://freefem.org/

Computer tutorial N°3

Error certification and mesh adaptivity for nonconforming finite elements

Poisson equation, nonconforming finite element method, potential reconstruction by averaging, equilibrated flux reconstruction by prescription, energy error, a posteriori error estimate, effectivity index, mesh adaptivity, rate of convergence with respect to the number of degrees of freedom, optimal (best-possible) error decay

Let $\Omega \subset \mathbb{R}^2$ be a polygon with Lipschitz boundary $\partial\Omega$. We consider the following model problem: for a given source term $f \in L^2(\Omega)$ and a given prescribed data g_D on $\partial\Omega$, find $u : \Omega \to \mathbb{R}$ such that

$$-\Delta u = f \qquad \text{in } \Omega, \tag{1a}$$

$$u = g_\mathrm{D} \qquad \text{on } \partial\Omega. \tag{1b}$$

The weak solution of problem (1) is a function $u \in H^1(\Omega)$ such that $u|_{\partial\Omega} = g_\mathrm{D}$ and

$$(\nabla u, \nabla v) = (f, v) \qquad \forall v \in H^1_0(\Omega). \tag{2}$$

Let $\mathcal{T}_\ell$ be a triangulation of Ω. The mesh index ℓ is not fixed here, as we will consider sequences of refined/adapted meshes indexed by ℓ. Employing the polynomial degree $p = 1$, the space of weakly continuous Crouzeix–Raviart 1st order polynomials is given by

$$V_\ell^{1,\mathrm{nc}} = \{v_\ell \in \mathcal{P}_1(\mathcal{T}_\ell);\ \langle [\![v_\ell]\!], 1\rangle_F = 0 \quad \forall F \in \mathcal{F}_h^{\mathrm{int}}\}. \tag{3}$$

Here, $[\![v_\ell]\!]$ is the jump of the function v_ℓ over the mesh face F: if $F \in \mathcal{F}_h^{\mathrm{int}}$ is an interior face shared by elements K_1 and K_2 (the orientation is not important here), then

$$[\![v_\ell]\!] = (v_\ell|_{K_1})|_F - (v_\ell|_{K_2})|_F.$$

The functions in $V_\ell^{1,\mathrm{nc}}$ have their traces only weakly continuous over all mesh faces in the sense that not the whole jumps but only their mean values vanish over all mesh interior faces. Equivalently, functions from the space $V_\ell^{1,\mathrm{nc}}$ are continuous in the midpoints $\boldsymbol{x}_F$ of all faces $F \in \mathcal{F}_h^{\mathrm{int}}$. An illustration of a basis function is given in Figure 1. The nonconforming finite element method approaches u of (2) by $u_\ell \in V_\ell^{1,\mathrm{nc}}$ such that $u_\ell(\boldsymbol{x}_F) = u_\mathrm{D}(\boldsymbol{x}_F)$ for any face lying on the boundary $\partial\Omega$, $F \in \mathcal{F}_h^{\mathrm{ext}}$, and such that

$$(\nabla_h u_\ell, \nabla_h v_\ell) = (\Pi_0 f, v_\ell) \qquad \forall v_\ell \in V_\ell^{1,\mathrm{nc}} \text{ such that } v_\ell(\boldsymbol{x}_F) = 0 \quad \forall F \in \mathcal{F}_h^{\mathrm{ext}}. \tag{4}$$

Here, ∇_h stands for the elementwise (also called broken) gradient, $(\nabla_h u_\ell)|_K := \nabla(u_\ell|_K)$ for all mesh elements $K \in \mathcal{T}_\ell$. Moreover, Π_0 is the $L^2(\Omega)$-orthogonal projection onto piecewise constants, i.e., $\Pi_0 f$ is given by elementwise mean values of f; employment of $\Pi_0 f$ and not directly of f gives (10b) below.

Exercice 1 below is designed for the case where $\Omega = (0,1)^2$, $g_\mathrm{D} = 0$, and $f = -2(x^2 + y^2) + 2(x + y)$. In this case,

$$u(x, y) = x(x-1)y(y-1), \tag{5}$$

which is a smooth solution.

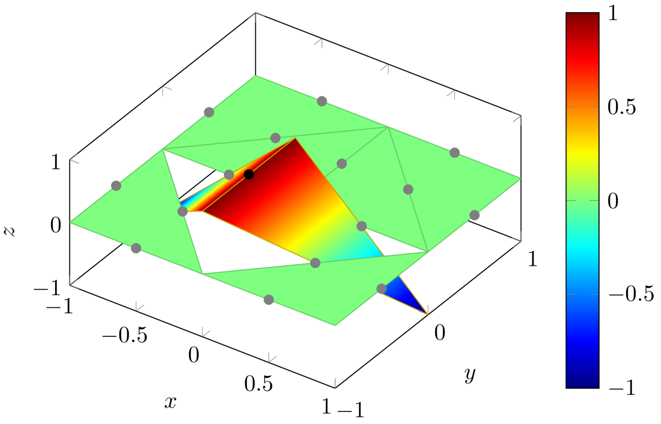

Figure 1: Basis function ψ^F of the space $V_\ell^{1,\mathrm{nc}}$

Exercices 2 and 3 present extensions to the L-shaped domain $\Omega = (-1,1) \times (-1,1) \setminus [0,1] \times [-1,0]$ with the exact solution written, in polar coordinates with $\theta \in (0, 3\pi/2)$, as

$$u(r,\theta) = r^{\frac{2}{3}} \sin(2\theta/3). \tag{6}$$

We remark that the exact solution is singular here, $u \in H^{\frac{5}{3}-\varepsilon}(\Omega)$ for arbitrarily small $\varepsilon > 0$. The corresponding source term $f = 0$, and we take $g_{\mathrm{D}} = u$ on $\partial\Omega$.

Exercice 1. (Errors and estimators on uniformly refined meshes, the smooth example)

1. Set up the example with the following parameters:

   `int nds = 2;` // number of mesh points on one unit boundary edge

   `int FinalLevel = 6;` // maximal refinement level

   `macro Pcontp P2` // H^1-conforming pw polynomials for plotting the exact solution: “P2”

   `macro Pcont P1` // H^1-conforming pw polynomials for the potential reconstruction: “P1” or “P2”

   `macro RT RT0` // H(div)-conforming pw polynomials for the flux reconstruction: “RT0”

   `macro Pdiscp P2dc` // discontinuous pw polynomials for plotting the fluxes and flux reconstruction: “P2dc”

   `int verb = 3;` // verbosity for demonstrations and debugging

   `bool PlotMesh = 1, PlotSolAppr = 1, PlotSolApprFluxes = 1,`

   `PlotPotRec = 1, PlotApprPot = 1, PlotFluxRec = 1,`

   `PlotErr = 1, PlotEst = 1;` // what to plot

   `bool RunSmooth = 1;` // 1 means smooth example, 0 means singular example

   `bool RunAdaptive = 0;` // 1 means adaptive mesh refinement, 0 means uniform mesh refinement

   `bool ImposeBC0 = 1;` // potential reconstruction: impose boundary values 0 where a homogeneous Dirichlet BC is prescribed

   Make sure that the data for the smooth example (5) are chosen:

   ```
   // CASE 1 (smooth polynomial in a unit square)
   ...
   ```

2. Define the nonconforming finite element space $V_\ell^{1,\mathrm{nc}}$ given by (3) in Freefem++: this is simply done by the command:

   `fespace VhNC(Th,P1nc);` // piecewise 1st order polynomials on the mesh $\mathcal{T}_\ell$, continuous in midpoints of the edges

3. Compute the nonconforming finite element approximation of (4) on a sequence of uniformly refined meshes for the smooth example (5). In Freefem++, this is realized by the following commands:

   `VhNC uh,vh;` // NCFE approximation and test function

   ```
   varf a(uh,vh)=int2d(Th)(Grad(uh)'*Grad(vh))
    + int2d(Th)(Pif*vh) + on(1,2,3,4,uh=gd); // weak form
   ```

   `matrix A=a(VhNC,VhNC,solver=SolverGlob);` // construction of the matrix and choice of the linear algebraic solver

   `real[int] b=a(0,VhNC);` // construction of the right-hand side

   `uh[]=A^-1*b;` // algebraic solve

4. Plot the exact solution and the numerical approximation, the exact fluxes and the approximate fluxes, and the exact fluxes and approximate fluxes magnitudes. As in tutorial N°1, Exercice 1, questions 8 and 9, please investigate whether $u_\ell \in H_0^1(\Omega)$ and $-\nabla_h u_\ell \in \boldsymbol{H}(\mathrm{div}, \Omega)$, which is the case for the exact solution, where $u \in H_0^1(\Omega)$ and $-\nabla u \in \boldsymbol{H}(\mathrm{div}, \Omega)$.

5. Implement a potential reconstruction s_ℓ by averaging. Recall the usual conforming finite element space V_ℓ^p of tutorials N°1 and N°2: for a fixed polynomial degree $p \geq 1$,

   $$V_\ell^p := \{v_\ell \in H^1(\Omega), v_\ell|_K \in \mathcal{P}_p(K) \quad \forall K \in \mathcal{T}_\ell\} = \mathcal{P}_p(\mathcal{T}_\ell) \cap H^1(\Omega). \tag{7}$$

   The idea of the potential reconstruction by averaging is to start from u_ℓ, not contained in $H^1(\Omega)$, and to use a simple averaging of the values that u_ℓ takes in all the Lagrangian degrees of freedom of the space V_ℓ^p, plus to impose the homogeneous boundary value on $\partial\Omega$:

   $$s_\ell(\boldsymbol{x}) := \frac{1}{|\mathcal{T}_{\boldsymbol{x}}|} \sum_{K \in \mathcal{T}_{\boldsymbol{x}}} v_\ell|_K(\boldsymbol{x}) \qquad \boldsymbol{x} \text{ is a Lagrange point of } V_\ell^p \text{ included in } \Omega, \tag{8a}$$

   $$s_\ell(\boldsymbol{x}) := 0 \qquad \boldsymbol{x} \text{ is a Lagrange point of } V_\ell^p \text{ included in } \partial\Omega. \tag{8b}$$

   In Freefem++, the space V_ℓ^p is simply obtained by

   `fespace Vh(Th,Pcont);` // H^1-conforming pw polynomials of degree `Pcont` for the potential reconstruction

   and the potential reconstruction s_ℓ will be

   `Vh sh;` // potential reconstruction

   The exact averaging (8a) might take some coding in Freefem++, but a quick workaround is readily available: the command

   `sh=uh;` // automatic interpolation of all possibly different values of $u_\ell \in V_\ell^{1,\mathrm{nc}}$ to prescribe s_ℓ in each Lagrange node of the space V_ℓ^p

   automatically and rapidly interpolates u_ℓ to s_ℓ. It does not impose (8b), but this can be added by the script in the part `if (ImposeBC0)`. Plot the potential reconstruction s_ℓ as well as the difference $u_\ell - s_\ell$.

6. Implement the equilibrated flux reconstruction $\boldsymbol{\sigma}_\ell$ by prescription. In nonconforming finite elements, prescribing

$$\boldsymbol{\sigma}_\ell|_K := -\nabla_h u_\ell|_K + \frac{(\Pi_0 f)|_K}{2}(\boldsymbol{x} - \boldsymbol{x}_K)|_K \qquad \forall K \in \mathcal{T}_\ell \tag{9}$$

gives

$$\boldsymbol{\sigma}_\ell \in \boldsymbol{H}(\mathrm{div}, \Omega), \tag{10a}$$
$$\nabla{\cdot}\boldsymbol{\sigma}_\ell = \Pi_0 f. \tag{10b}$$

In comparison with Exercice 3 of tutorial N°1, we call this a "prescription", since no local (quadratic minimization) problem needs to be solved. Congruently, formula (9) is much cheaper in practice than formula (11) of tutorial N°1. Plot this equilibrated flux reconstruction $\boldsymbol{\sigma}_\ell$ and compare it with the exact flux $-\nabla u$. Plot also the divergence misfit on each element $K \in \mathcal{T}_\ell$,

$$\|\Pi_0 f - \nabla{\cdot}\boldsymbol{\sigma}_\ell\|_K. \tag{11}$$

7. Compute the actual error $\|\nabla_h(u - u_\ell)\|$ as well as its elementwise contributions

$$\|\nabla(u - u_\ell)\|_K \tag{12}$$

for each mesh element $K \in \mathcal{T}_\ell$. Compute also the a posteriori error estimators

$$\|\nabla_h(u-u_\ell)\|^2 \le \eta_\ell^2 := \sum_{K\in\mathcal{T}_\ell} \Big(\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K + \frac{h_K}{\pi}\|f - \nabla{\cdot}\boldsymbol{\sigma}_\ell\|_K\Big)^2 + \sum_{K\in\mathcal{T}_\ell} \|\nabla(u_\ell - s_\ell)\|_K^2 \tag{13}$$

together with the elementwise contributions defined by

$$\eta_K := \|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K + \frac{h_K}{\pi}\|f - \nabla{\cdot}\boldsymbol{\sigma}_\ell\|_K + \|\nabla(u_\ell - s_\ell)\|_K. \tag{14}$$

Plot the elementwise errors $\|\nabla(u - u_\ell)\|_K$ and the elementwise estimators η_K and compare these (predicted) error distributions. What do you observe?

8. Compare the size of the a posteriori error estimator η_ℓ to the size of the error $\|\nabla_h(u - u_\ell)\|$, again using the effectivity index

$$I_{\mathrm{eff},\ell} := \frac{\eta_\ell}{\|\nabla_h(u - u_\ell)\|}. \tag{15}$$

What do you observe?

9. Plot the convergence of the errors and estimators against the total number of degrees of freedom (DoFs). Also check in the command window what is the convergence rate on the sequence of uniformly refined meshes. Is this convergence rate optimal?

Answer 1. (Errors and estimators on uniformly refined meshes, the smooth example)

The answers to items 1–3 are contained in the script `TP3.edp`. We now illustrate answers to questions 4–9. We plot most of the results on the first uniformly refined mesh $\mathcal{T}_1$ and some on the sixth uniformly refined mesh $\mathcal{T}_6$.

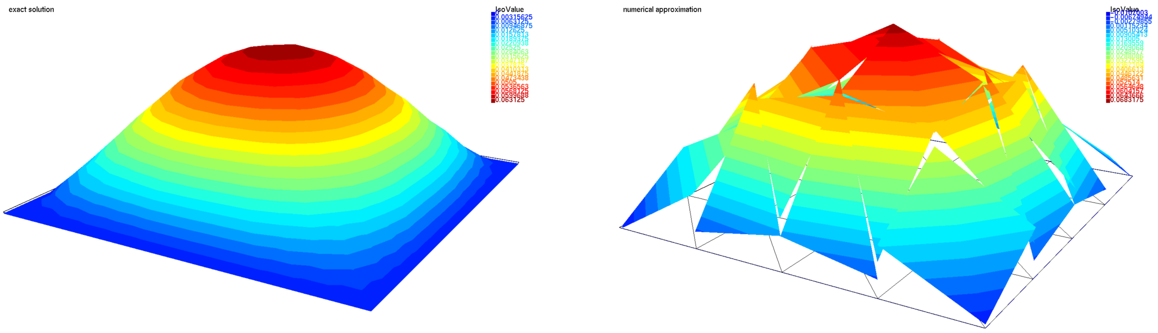

Figure 2: Exact solution u (left) and approximate solution u_ℓ (right), mesh $\mathcal{T}_1$, smooth solution (5)

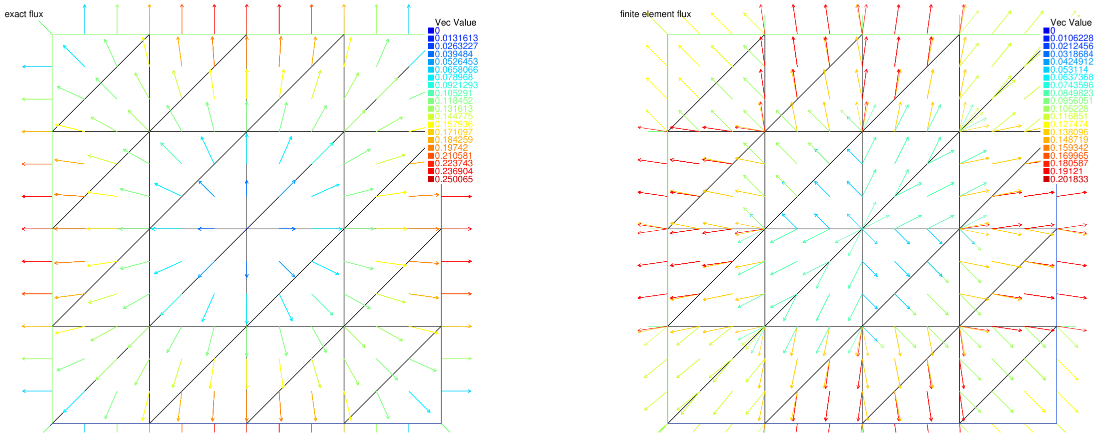

Figure 3: Flux of the exact solution $-\nabla u$ (left) and flux of the approximation $-\nabla_h u_\ell$ (right), mesh $\mathcal{T}_1$, smooth solution (5)

4. The exact solution u is depicted in Figure 2, left. The nonconforming finite element approximation is then illustrated in Figure 2, right; clearly, u_ℓ is not (trace) continuous, $u_\ell \notin H_0^1(\Omega)$. Figure 3 then plots the exact and approximate fluxes; clearly, $-\nabla_h u_\ell$ is not normal-trace continuous, $-\nabla_h u_\ell \notin \boldsymbol{H}(\mathrm{div}, \Omega)$. The magnitude of the fluxes is then displayed in Figures 4 and 5. The approximate fluxes are elementwise constant, with elementwise constant magnitudes, which makes the plots in Figure 4 rather different visually; the visual match is much better in Figure 5.

5. The potential reconstruction s_ℓ of the nonconforming finite element approximation u_ℓ from Figure 2, right, is displayed in Figure 6, left. The difference $u_\ell - s_\ell$ is then plotted in Figure 6, right (note the different the colorbars of the two figures).

6. The equilibrated flux reconstruction $\boldsymbol{\sigma}_\ell$ by prescription of (9) is displayed in Figure 7, left. Just like $-\nabla u$, and unlike $-\nabla_h u_\ell$, $\boldsymbol{\sigma}_\ell$ is normal-trace continuous, $\boldsymbol{\sigma}_\ell \in \boldsymbol{H}(\mathrm{div}, \Omega)$. From (10b), it is also equilibrated, which is confirmed in Figure 7, right.

7. The elementwise errors $\|\nabla(u - u_\ell)\|_K$ from (12) and the elementwise estimators η_K from (14) are plotted side by side in Figure 8 on the mesh $\mathcal{T}_1$ and in Figure 9 on the mesh $\mathcal{T}_6$. We can observe a reasonable match, though less precise than in tutorial N°1, Exercices 4–5. This is a consequence of two facts: 1) the averaging formula (8) is less precise than the patchwise local (quadratic minimization) problems of tutorial N°1, Exercice 3; 2) Freefem++ does not really perform the averaging (8a) but rather some (less precise) automatic interpolation; in particular, the boundary values are

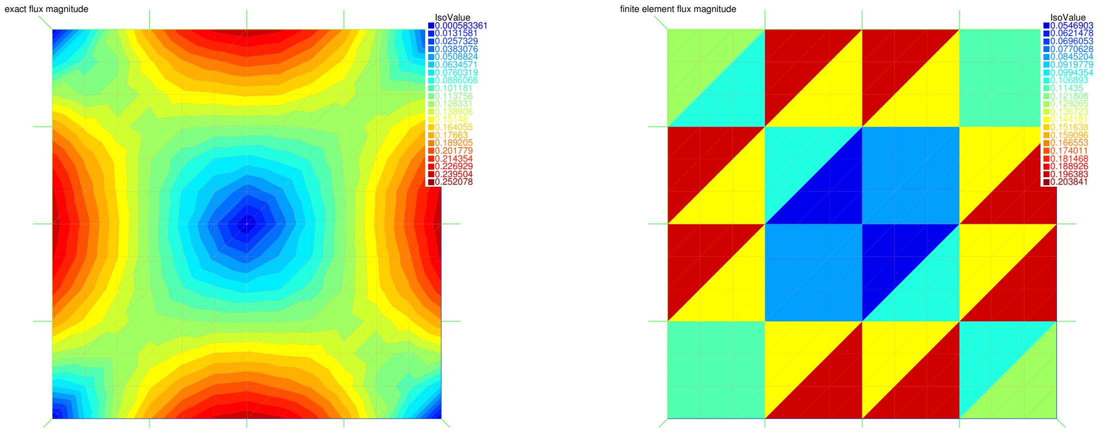

Figure 4: Magnitude of the fluxes, exact solution $|\nabla u|$ (left) and approximation $|\nabla_h u_\ell|$ (right), mesh $\mathcal{T}_1$, smooth solution (5)

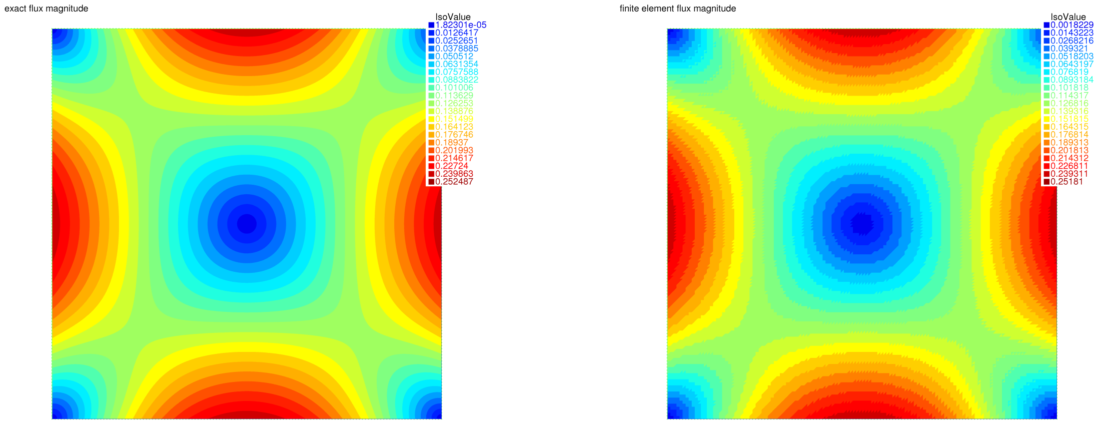

Figure 5: Magnitude of the fluxes, exact solution $|\nabla u|$ (left) and approximation $|\nabla_h u_\ell|$ (right), mesh $\mathcal{T}_6$, smooth solution (5)

affected. Also, upon inspection, it can be noticed that the prescription (9) leads to smaller values of $\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K$ and $\frac{h_K}{\pi}\|f - \Pi_0 f\|_K$ in comparison with $\|\nabla(u_\ell - s_\ell)\|_K$ (the contribution from the potential reconstruction dominates over that from the flux reconstruction).

8. The effectivity indices take the values

```
Effectivity indices 7

1.810485672   1.733789766   1.647820726   1.58845309

1.551808561   1.531134438   1.520113529
```

These are slightly higher than in tutorial N°1, Exercice 5, question 3, but stable, still completely acceptable, and obtained for a (much) smaller computational effort (more details on the computational cost are discussed in Answer 2, question 8).

9. The convergence of the errors and estimators against the total number of degrees of freedom (DoFs) is plotted in Figure 10. The computed convergence rates on the sequence of uniformly refined meshes are precisely

```
Rate of convergence 6

0.4712427639   0.5059034645   0.5088411608   0.5059648252

0.5033706952   0.5017821374
```

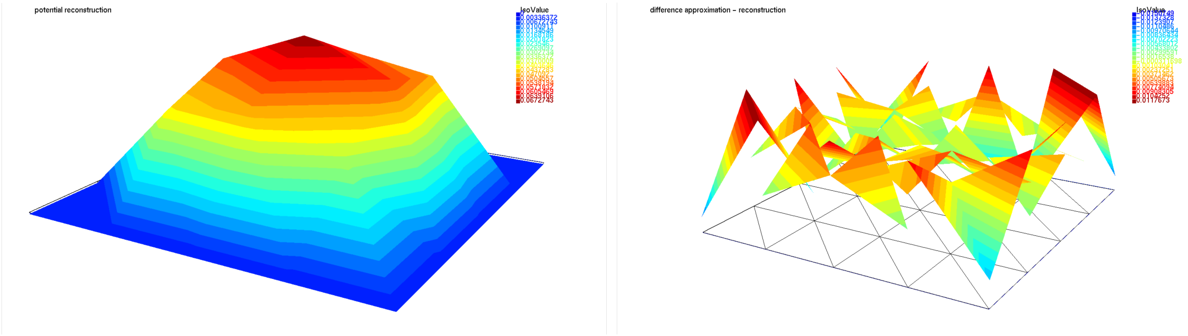

Figure 6: Potential reconstruction s_ℓ (left) and the difference $u_\ell - s_\ell$ (right), mesh $\mathcal{T}_1$, smooth solution (5)

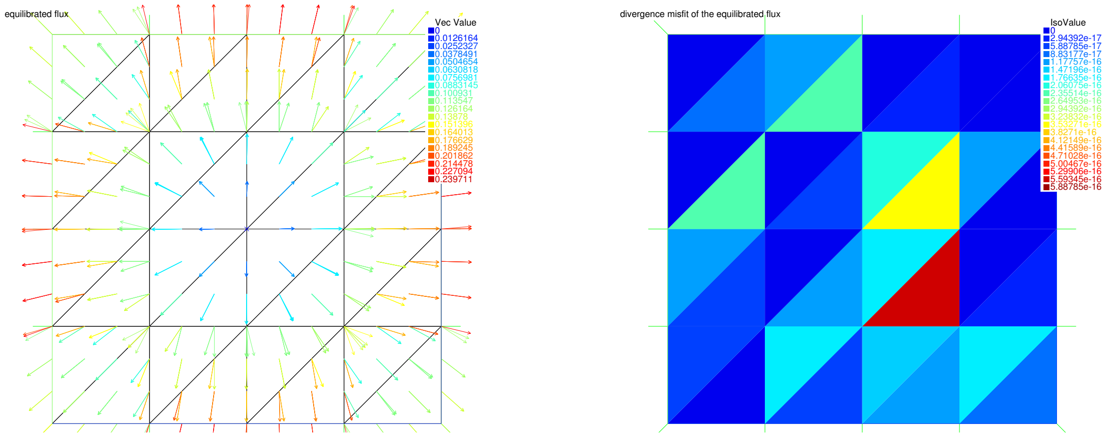

Figure 7: Equilibrated flux $\boldsymbol{\sigma}_\ell$ (left) and its divergence misfit (right), mesh $\mathcal{T}_1$, smooth solution (5)

which is approximately $\mathcal{O}(h)$ in the mesh size h and $\mathcal{O}(\mathrm{DoFs}^{-\frac{1}{2}})$ in terms of DoFs. This is exactly the expected (and optimal) convergence rate for an approximation of a smooth solution on a sequence of uniform meshes.

Exercice 2. (Errors and estimators on uniformly refined meshes, the singular example)

1. Set up the example with the following parameters (only changes with respect to question 1 of Exercice 1 are listed):

   `bool RunSmooth = 0;` // 1 means smooth example, 0 means singular example

   Make sure that the data for the singular example (6) are chosen:

   ```
   // CASE 2 (vertex singularity)
   ...
   ```

2. Define the nonconforming finite element space $V_\ell^{1,\mathrm{nc}}$ given by (3), compute the nonconforming finite element approximation of (4) on a sequence of uniformly refined meshes for the singular example (6), and plot the obtained meshes, the exact solution and the numerical approximation, the exact fluxes and the approximate fluxes, and the exact fluxes and approximate fluxes magnitudes.

3. Plot the potential reconstruction s_ℓ given by (8) as well as the difference $u_\ell - s_\ell$.

4. Plot the equilibrated flux reconstruction $\boldsymbol{\sigma}_\ell$ of (9) and as well as the divergence misfit of (11).

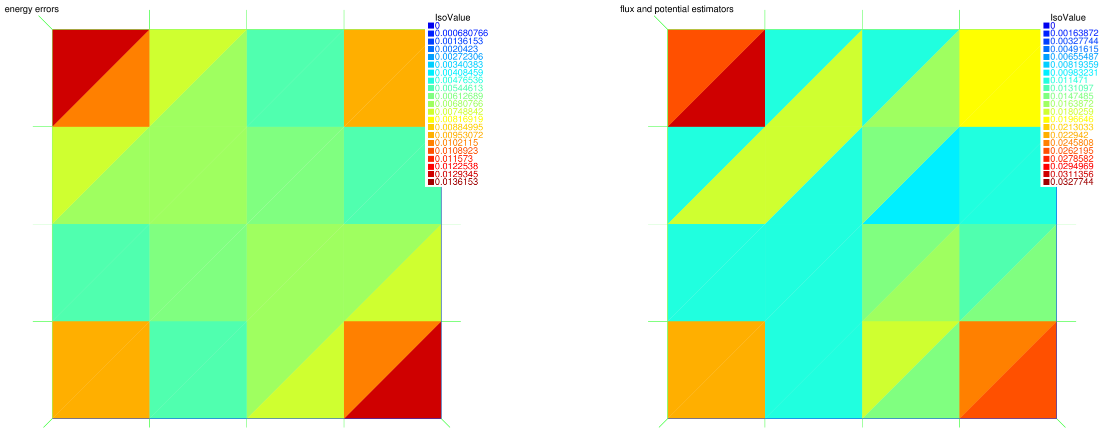

Figure 8: Elementwise errors $\|\nabla(u-u_\ell)\|_K$ (left) and estimators $\big[\|\nabla u_\ell+\boldsymbol{\sigma}_\ell\|_K+\frac{h_K}{\pi}\|f-\Pi_0 f\|_K+\|\nabla(u_\ell-s_\ell)\|_K\big]$ (right), mesh $\mathcal{T}_1$, smooth solution (5)

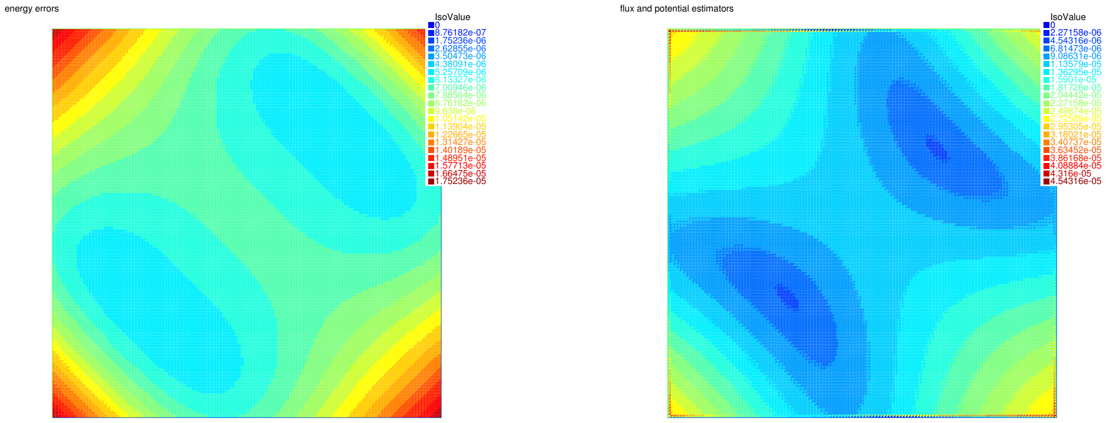

Figure 9: Elementwise errors $\|\nabla(u-u_\ell)\|_K$ (left) and estimators $\big[\|\nabla u_\ell+\boldsymbol{\sigma}_\ell\|_K+\frac{h_K}{\pi}\|f-\Pi_0 f\|_K+\|\nabla(u_\ell-s_\ell)\|_K\big]$ (right), mesh $\mathcal{T}_6$, smooth solution (5)

5. Plot the elementwise errors $\|\nabla(u-u_\ell)\|_K$ of (12) and the elementwise estimators η_K of (14) and compare these (predicted) error distributions. Since $f=0$ here, it follows from (9) that here

$$\Big[\|\nabla u_\ell+\boldsymbol{\sigma}_\ell\|_K+\frac{h_K}{\pi}\|f-\Pi_0 f\|_K+\|\nabla(u_\ell-s_\ell)\|_K\Big]=\|\nabla(u_\ell-s_\ell)\|_K. \tag{16}$$

6. Compute and discuss the effectivity indices (15).

7. Plot the convergence of the errors and estimators against the total number of degrees of freedom (DoFs). Also check in the command window what is the convergence rate on the considered sequence of uniformly refined meshes. Is this convergence rate optimal?

8. Modify the parameters to:

   ```
   int FinalLevel = 8; // maximal refinement level
   ```

   and

   ```
   bool PlotMesh = 0, PlotSolAppr = 0, PlotSolApprFluxes = 0,
   PlotPotRec = 0, PlotApprPot = 0, PlotFluxRec = 0,
   PlotErr = 0, PlotEst = 0; // what to plot
   ```

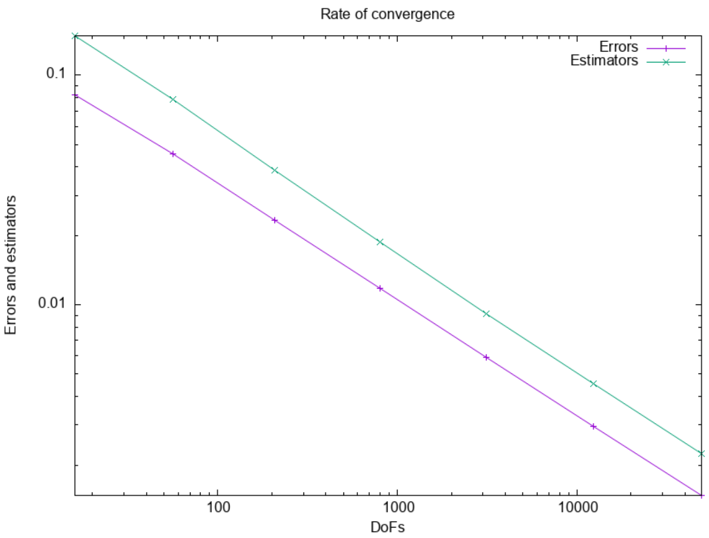

Figure 10: Convergence of the energy errors and of a posteriori error estimators for the smooth example (5) under uniformly refined meshes

Run the script and examine the CPU times of the different parts. What are the most consuming parts and at which rate this changes with mesh refinement?

Answer 2. (Errors and estimators on uniformly refined meshes, the singular example)

The answer to items 1 is contained in the script `TP3.edp`. We now illustrate answers to questions 2–8. We plot the results on the initial mesh $\mathcal{T}_0$ and some on the third uniformly refined mesh $\mathcal{T}_3$.

2. The exact solution u and its nonconforming finite element approximation u_ℓ are depicted in Figure 11. Figure 12 then plots the exact and approximate fluxes; here, actually, clearly, $-\nabla_h u_\ell$ is normal-trace continuous, $-\nabla_h u_\ell \in \boldsymbol{H}(\mathrm{div}, \Omega)$, since $-\nabla_h u_\ell = \boldsymbol{\sigma}_\ell$ in view of (9), since $f = 0$ here. The magnitude of the fluxes is then displayed in Figures 13 and 14.

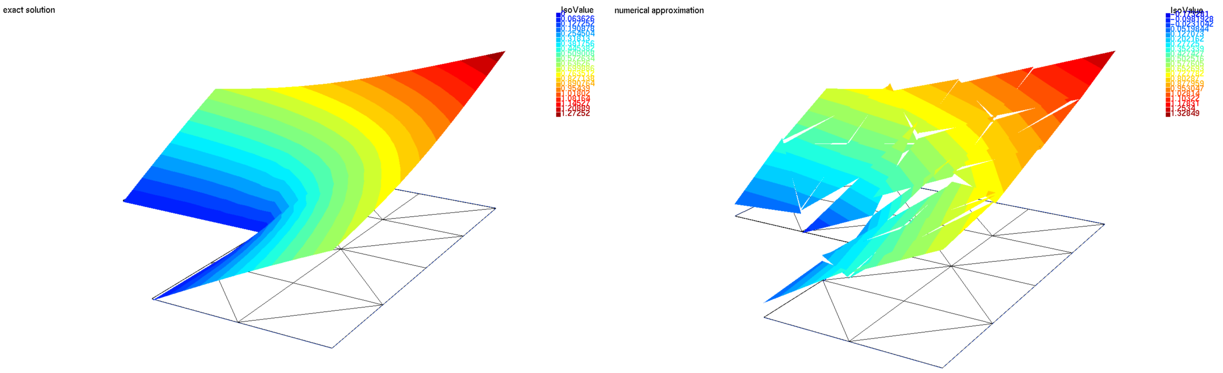

Figure 11: Exact solution u (left) and approximate solution u_ℓ (right), mesh $\mathcal{T}_1$, singular solution (6)

3. The potential reconstruction s_ℓ of the nonconforming finite element approximation u_ℓ from Figure 11, right, is displayed in Figure 15, left. The difference $u_\ell - s_\ell$ is then plotted in Figure 15, right.

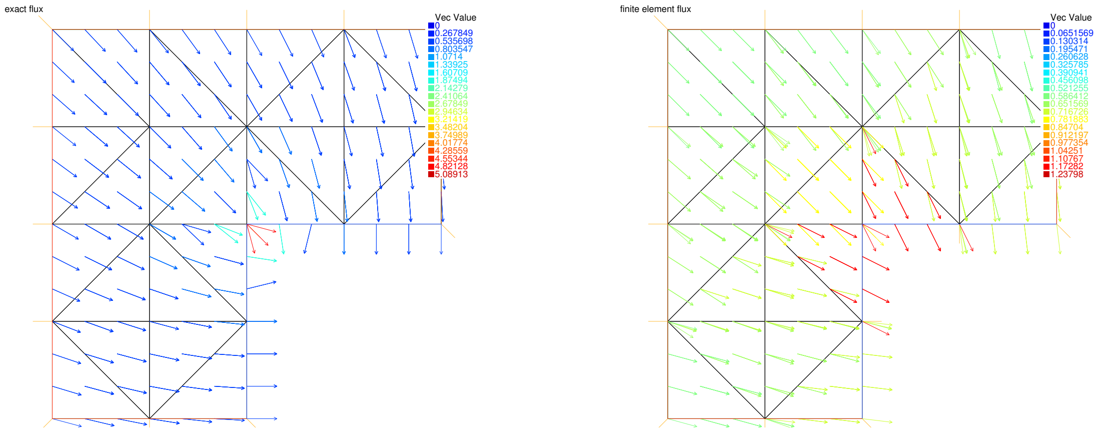

Figure 12: Flux of the exact solution $-\nabla u$ (left) and flux of the approximation $-\nabla_h u_\ell$ (right), mesh $\mathcal{T}_1$, singular solution (6)

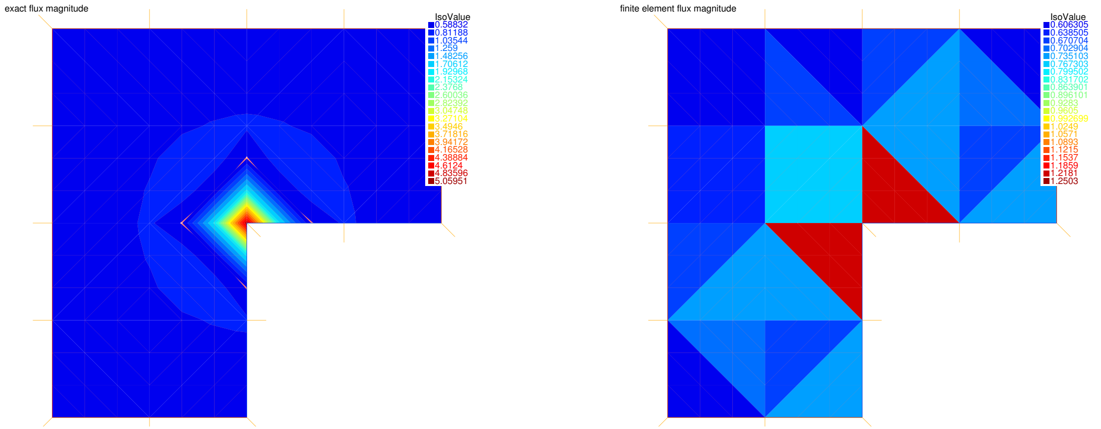

Figure 13: Magnitude of the fluxes, exact solution $|\nabla u|$ (left) and approximation $|\nabla_h u_\ell|$ (right), mesh $\mathcal{T}_1$, singular solution (6)

4. The equilibrated flux reconstruction $\boldsymbol{\sigma}_\ell$ by prescription of (9) is displayed in Figure 16. Recall that $\boldsymbol{\sigma}_\ell = -\nabla_h u_\ell \in \boldsymbol{H}(\mathrm{div}, \Omega)$ in view of (9), since $f = 0$ here.

5. The elementwise errors $\|\nabla(u - u_\ell)\|_K$ from (12) and the elementwise estimators η_K from (14) are plotted side by side in Figure 17 on the mesh $\mathcal{T}_1$ and in Figure 18 on the mesh $\mathcal{T}_3$. We can observe a very good match here, with the error concentrated around the corner singularity.

6. The effectivity indices take the values

   ```
   Effectivity indices 7
   
   1.622031034   1.606218872   1.637342373   1.652335661
   
   1.66018063    1.664770028   1.667545693
   ```

   Again, these are slightly higher than in tutorial N°1, Exercice 5, question 3, but stable, still completely acceptable, and obtained for a (much) smaller computational effort (see question 8 below).

7. The convergence of the errors and estimators against the total number of degrees of freedom (DoFs) is plotted in Figure 19. The computed convergence rates on the sequence of uniformly refined meshes are precisely

   ```
   Rate of convergence 6
   ```

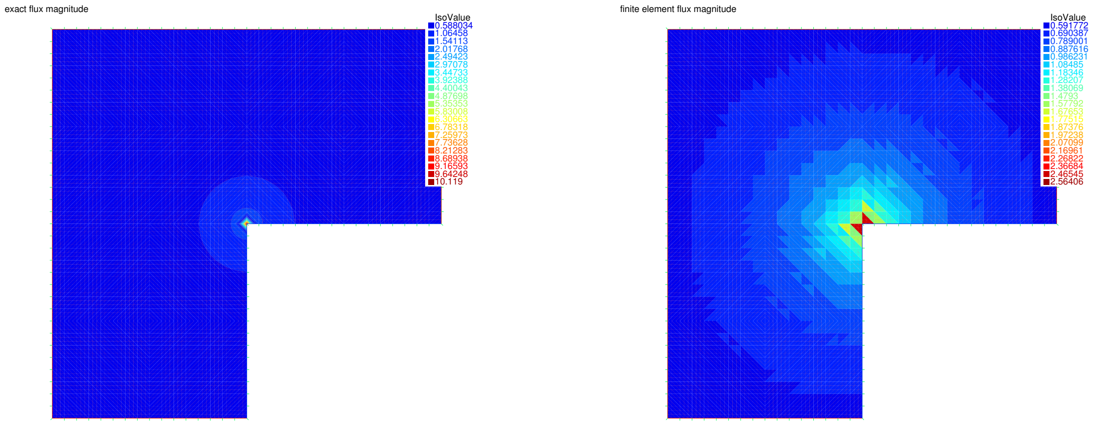

Figure 14: Magnitude of the fluxes, exact solution $|\nabla u|$ (left) and approximation $|\nabla_h u_\ell|$ (right), mesh $\mathcal{T}_3$, singular solution (6)

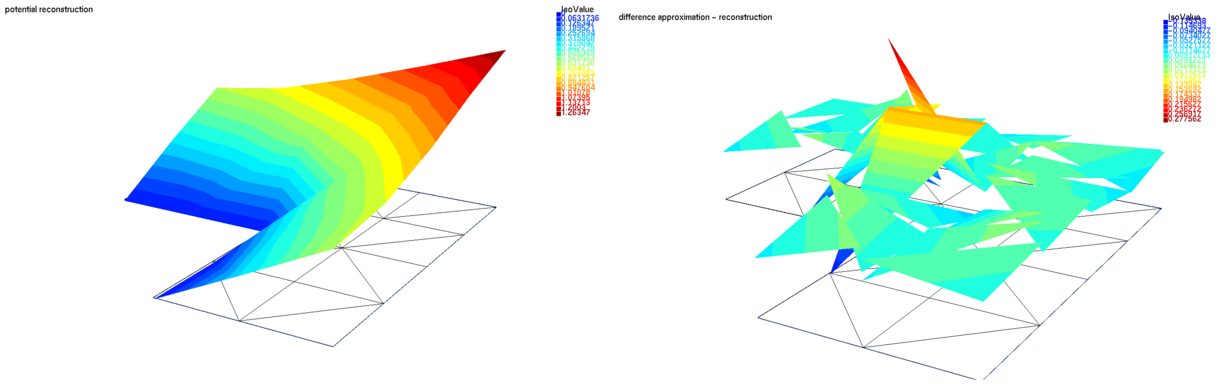

Figure 15: Potential reconstruction s_ℓ (left) and the difference $u_\ell - s_\ell$ (right), mesh $\mathcal{T}_1$, singular solution (6)

```
0.3159583098    0.3244941636    0.3276054793    0.3293865961
0.3306041534    0.3314685946
```

which is approximately $\mathcal{O}(h^{2/3})$ in the mesh size h and $\mathcal{O}(\mathrm{DoFs}^{-\frac{1}{3}})$ in terms of DoFs. This rate is not optimal in terms of DoFs but we will see below in Exercice 3 that optimal rates are obtained under adaptive mesh regeneration.

8. The results below (last two meshes) are obtained on a personal laptop with 6 Intel i7-9850H CPUs at 2.60GHz and 32 GB RAM:

```
mesh level 7
number of degrees of freedom 590848
CPU time of matrix assembly and nonconforming FE solution = 4.488
CPU time of potential reconstruction = 0.153
CPU time of equilibrated flux reconstruction = 1.128
CPU time of error computing = 13.431
CPU time of a posteriori estimators computing = 3.959
CPU time of mesh regeneration = 1.346
```

and

```
mesh level 8
```

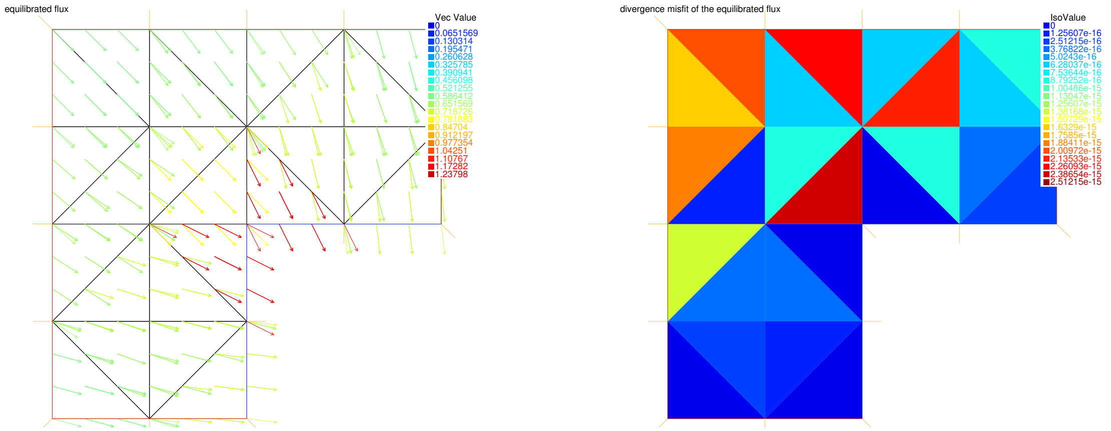

Figure 16: Equilibrated flux $\boldsymbol{\sigma}_\ell$ (left) and its divergence misfit (right), mesh $\mathcal{T}_1$, singular solution (6)

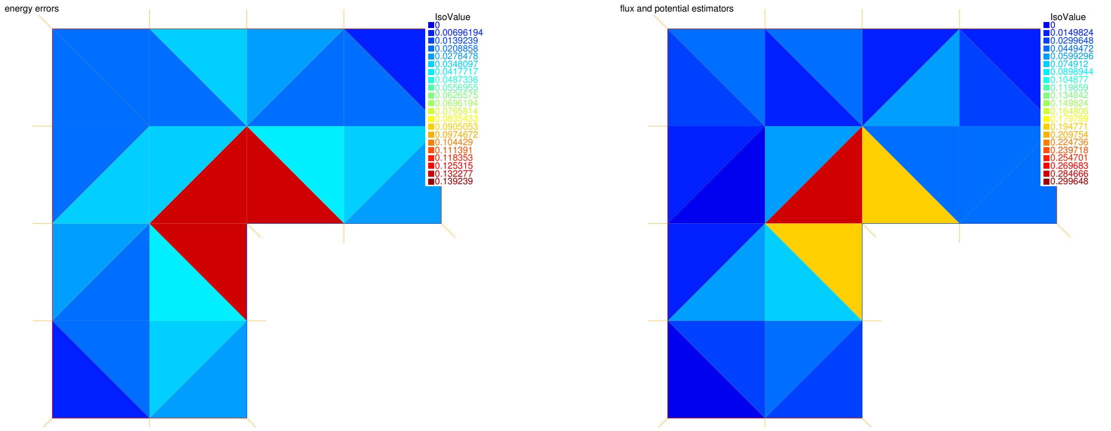

Figure 17: Elementwise errors $\|\nabla(u - u_\ell)\|_K$ (left) and estimators $\left[\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K + \frac{h_K}{\pi}\|f - \Pi_0 f\|_K + \|\nabla(u_\ell - s_\ell)\|_K\right] = \|\nabla(u_\ell - s_\ell)\|_K$ (right), mesh $\mathcal{T}_1$, singular solution (6)

```
number of degrees of freedom 2.36134e+06

CPU time of matrix assembly and nonconforming FE solution = 43.155

CPU time of potential reconstruction = 0.619

CPU time of equilibrated flux reconstruction = 4.427

CPU time of error computing = 51.171

CPU time of a posteriori estimators computing = 15.672
```

Clearly, the CPU times of potential and equilibrated flux reconstructions as well as error and a posteriori estimators computings grow linearly: four times more elements leads to approximately four times longer CPU time. This is not the case with the nonconforming FE solution using the sparse solver of UMFPACK (which is behind `macro SolverGlob sparsesolver`): though it does not dominate on smaller meshes, this changes importantly with mesh refinement. More severely, mesh level 9 seems to be out of reach with this direct solver because of insufficient memory. Using instead the iterative conjugate gradients solver upon setting

```
macro SolverGlob CG
```

one can take

```
int FinalLevel = 9; // maximal refinement level
```

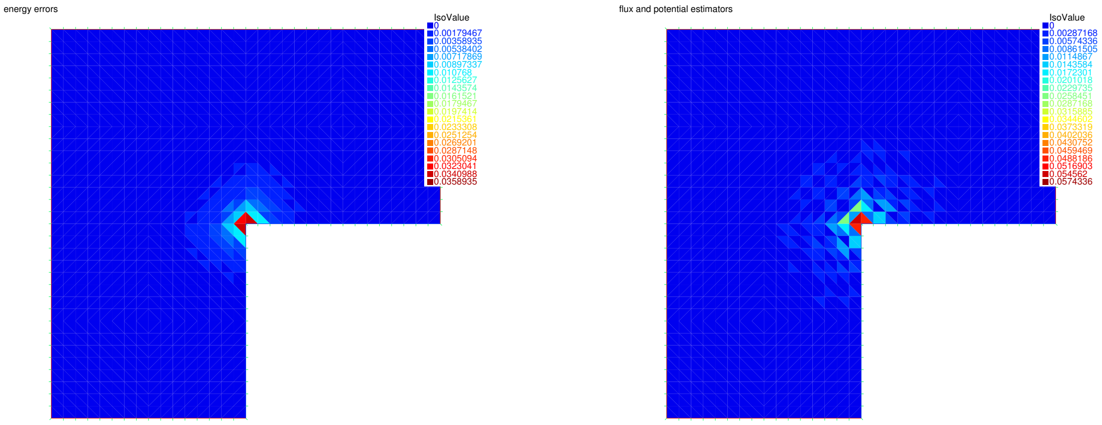

Figure 18: Elementwise errors $\|\nabla(u-u_\ell)\|_K$ (left) and estimators $\left[\|\nabla u_\ell+\boldsymbol{\sigma}_\ell\|_K+\frac{h_K}{\pi}\|f-\Pi_0 f\|_K+\|\nabla(u_\ell-s_\ell)\|_K\right]=\|\nabla(u_\ell-s_\ell)\|_K$ (right), mesh $\mathcal{T}_3$, singular solution (6)

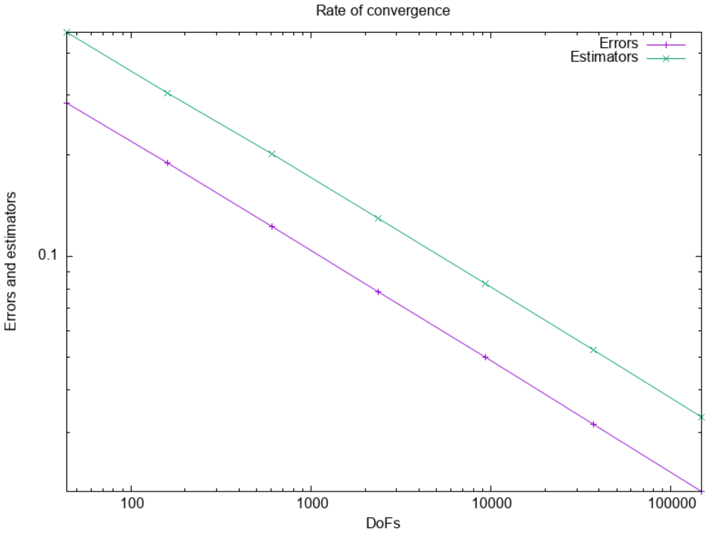

Figure 19: Convergence of the energy errors and of a posteriori error estimators for the singular example (6) under uniformly refined meshes

and obtain for the two last mesh levels:

```
mesh level 8

number of degrees of freedom 2.36134e+06

CPU time of matrix assembly and nonconforming FE solution = 132.457

CPU time of potential reconstruction = 0.611

CPU time of equilibrated flux reconstruction = 3.552

CPU time of error computing = 50.656

CPU time of a posteriori estimators computing = 18.57

CPU time of mesh regeneration = 5.478
```

and

```
mesh level 9

number of degrees of freedom 9.44128e+06

CPU time of matrix assembly and nonconforming FE solution = 1051.25
CPU time of potential reconstruction = 2.434
CPU time of equilibrated flux reconstruction = 14.542
CPU time of error computing = 203.183
CPU time of a posteriori estimators computing = 74.438
```

Recalling that we compute the error only for illustration purposes (and with an over-precise quadrature formula), the dominating part is now the numerical linear algebra solver, whereas less than one tenth of the CPU time on the last mesh is spent on a posteriori error estimates. Typically, only an appropriate multigrid would give CPU time only linearly increasing with the number of DoFs, and then all the parts of the script `TP3.edp` would be of the optimal, linear complexity in terms of the mesh elements/DoFs. On the other hand, in the present case where $f = 0$, we could actually avoid the equilibrated flux reconstruction and the computing of the associated estimators $\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K$ and $\frac{h_K}{\pi}\|f - \Pi_0 f\|_K$ since these are zero here (recall (16)).

Finally, the update of Figure 19 is presented in Figure 20.

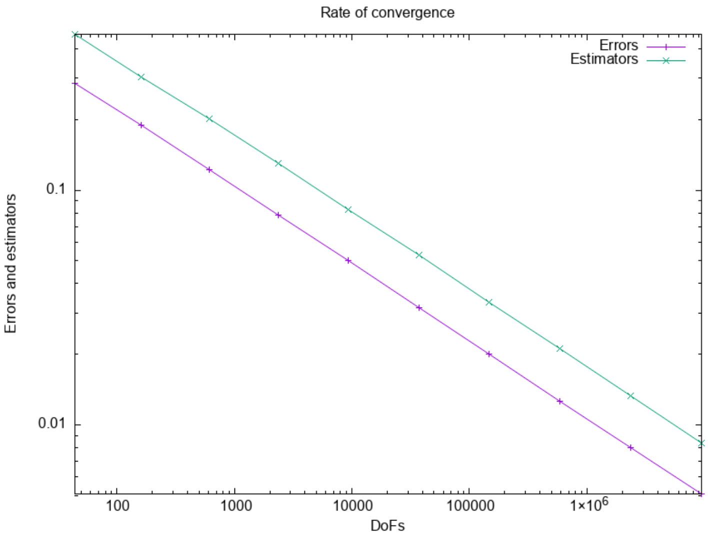

Figure 20: Convergence of the energy errors and of a posteriori error estimators for the singular example (6) under uniformly refined meshes

Exercice 3. (Errors and estimators on adaptively generated meshes, the singular example)

1. Set up the example with the following parameters (only changes/additions with respect to Exercice 2, question 1, are listed):

   ```
   bool RunAdaptive = 1; // 1 means adaptive mesh refinement, 0 means uniform mesh refinement
   real Dtheta=sqrt(0.5); // Dörfler marking parameter
   int RefFactor=4; // factor by which (approximately) the marked elements should become smaller in the adapted mesh
   ```

   Make sure that the data for the singular example (6) are chosen:

   ```
   // CASE 2 (vertex singularity)
   ...
   ```

2. Taking advantage of the a posteriori error estimators $\eta_K(u_\ell)$ computed in each mesh element $K \in \mathcal{T}_\ell$, only "refine" those mesh elements with a high value $\eta_K(u_\ell)$ of the estimated error. More precisely, for the parameter $0 < \theta \leq 1$, identify a subset $\mathcal{M}_\ell$ of all elements of the $\mathcal{T}_\ell$ mesh such that

$$\sum_{K \in \mathcal{M}_\ell} \eta_K(u_\ell)^2 \geq \theta^2 \sum_{K \in \mathcal{T}_\ell} \eta_K(u_\ell)^2$$

and then produce a new mesh with elements in the marked set $\mathcal{M}_\ell$ smaller by the given factor `RefFactor`. Plot some meshes of the adaptively generated mesh sequence.

3. Plot some of the elementwise distributions of the actual error (12) and of the a posteriori error estimators (16).

4. Compute and discuss the effectivity indices (15).

5. Plot the convergence of the errors and estimators against the total number of degrees of freedom (DoFs). Also check in the command window what is the convergence rate on the considered sequence of adaptively generated meshes. Is this convergence rate optimal? What gives a better error/cost ratio? Uniformly refined or adaptively generated meshes?

Answer 3. (Errors and estimators on adaptively generated meshes, the singular example)

1. The answer to item 1 is contained in the script `TP3.edp`.

2. Examples of adaptively generated meshes are plotted in Figure 21. We can see a strong refinement towards the re-entrant corner.

3. Examples of elementwise error and a posteriori estimators distributions are plotted in Figure 22. We can see a decent match in all situations.

4. The effectivity indices take slightly increased but stable values:

```
Effectivity indices 21

1.622031034   2.128426619   1.880197025   1.767225341   1.761006952

1.830774921   1.833058087   1.848562032   1.870945921   1.841472316

1.865356314   1.873385183   1.853134558   1.856375346   1.868329053

1.864346345   1.871147061   1.869364431   1.870792602   1.870664755

1.872694527
```

5. The convergence of the errors and estimators against the total number of degrees of freedom (DoFs) is displayed in Figure 23. In the command window, we find

```
Rate of convergence 20

0.8190981065   1.088483938   0.2784719482   58.9960561   0.3804000868

0.5565713152   0.4644765152   0.5060229063   0.5477635815   0.4823825402

0.5204493483   0.4896863111   0.4897148488   0.4969515192   0.5087273118

0.5037345784   0.4967512957   0.5050554199   0.4946700119   0.5045314553
```

The convergence rate is $\mathcal{O}(\mathrm{DoFs}^{-\frac{1}{2}})$, which is substantially better than $\mathcal{O}(\mathrm{DoFs}^{-\frac{1}{3}})$ in Exercice 2 with uniform mesh refinement. It is actually optimal, the best possible. From comparison of the results, cf. also Figures 23 and 20, we see that $9.44e+06$ DoFs are necessary to reach the error 0.00508628 on the finest (9th) mesh in the uniform refinement case, whereas only 94020 DoFs are sufficient for the comparable precision 0.005366323805 on the 15th adaptively generated meshes (this means roughly 100 times fewer DoFs!). The last, 20th adaptively generated mesh, than only has $1.04e+06$ DoFs and yields to the error 0.001603426738, which is roughly 3 times smaller than the 9th uniformly refined mesh. Thus adaptively generated meshes give a much better error/cost ratio.

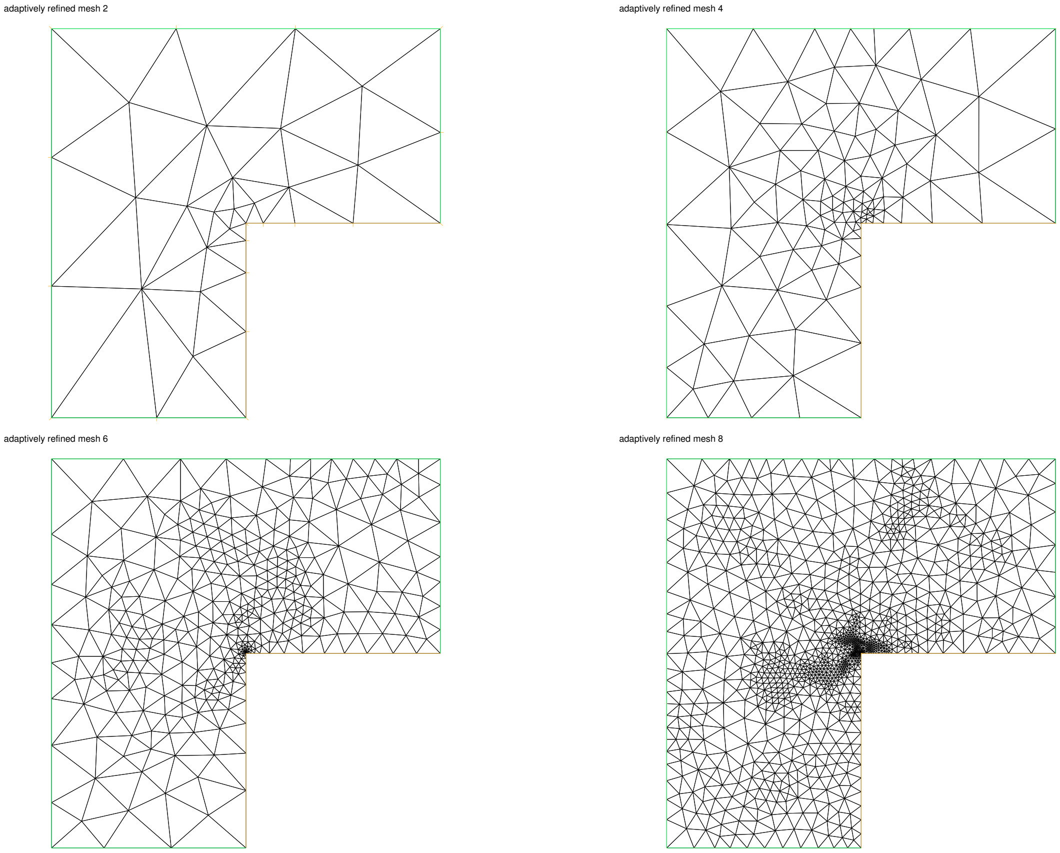
adaptively refined mesh 2
adaptively refined mesh 4
adaptively refined mesh 6
adaptively refined mesh 8

Figure 21: Level 2 adaptively generated mesh (top left), level 4 adaptively generated mesh (top right), level 6 adaptively generated mesh (bottom left), and level 8 adaptively generated mesh (bottom right) for the singular example (6)

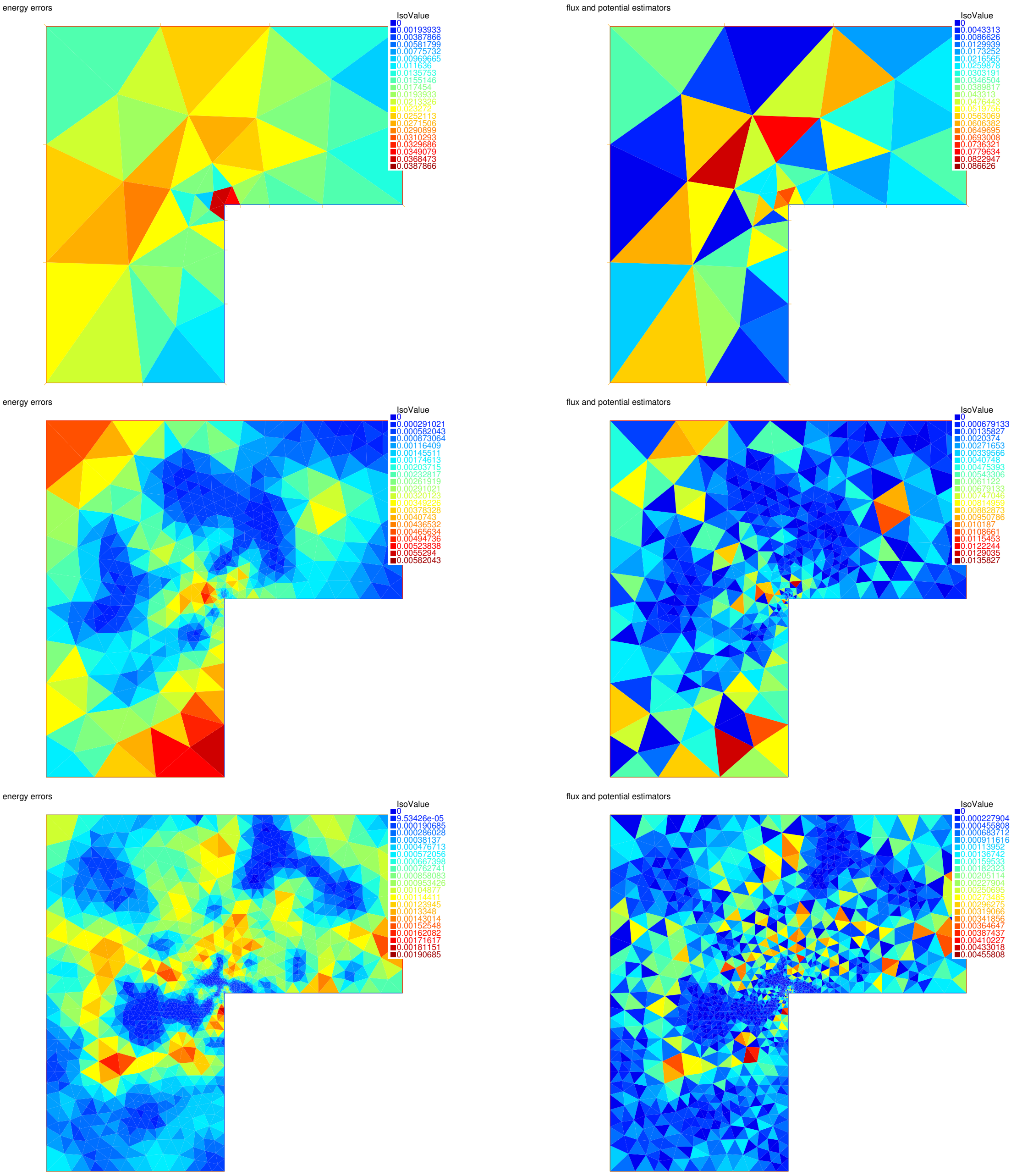

Figure 22: Elementwise errors $\|\nabla(u - u_\ell)\|_K$ (left) and elementwise estimators $\big[\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K + \frac{h_K}{\pi}\|f - \Pi_0 f\|_K + \|\nabla(u_\ell - s_\ell)\|_K\big] = \|\nabla(u_\ell - s_\ell)\|_K$ (right) on level 2 adaptively generated mesh (top), level 6 adaptively generated mesh (middle), and level 8 adaptively generated mesh (bottom) for the singular example (6)

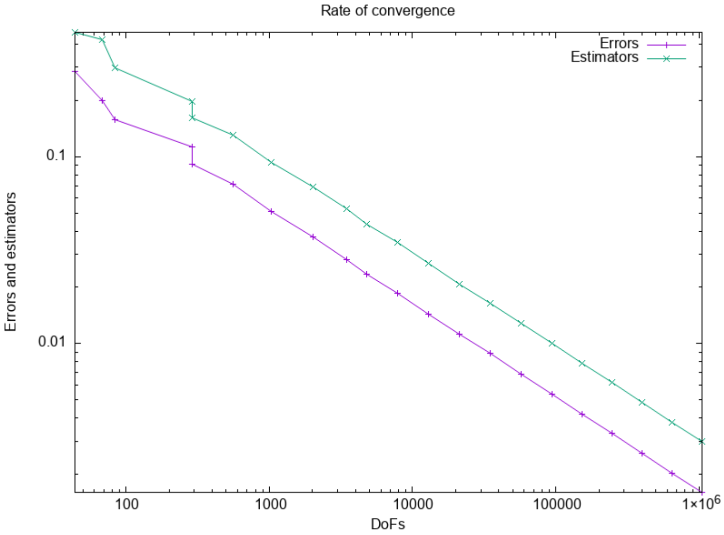

Figure 23: Convergence of the energy errors and of a posteriori error estimators for the singular example (6) on adaptively generated meshes