Computer tutorial N°2

Mesh adaptivity

Poisson equation, conforming finite element method, 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

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

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,

$$u(x, y) = x(x-1)y(y-1), \tag{3}$$

which is a smooth solution.
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

$$u(r, \theta) = r^{\frac{2}{3}} \sin(2\theta/3). \tag{4}$$

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. Compute the "P1" FE approximation and the "RT1" and "P1dc" equilibrated flux on a sequence of uniformly refined meshes for the smooth example (3). Plot the obtained meshes and the elementwise distributions of the actual error and of the a posteriori error estimators. Please use the following parameters:

   `int nds = 4;` // number of mesh points on one unit boundary edge

   `FinalLevel = 3;` // maximal refinement level

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

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

   Remove the comment `/*` and define the data for the smooth example:

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

```
func uEx = x*(x-1)*y*(y-1);
func dxuEx = (2*x-1)*y*(y-1);
...
// define the computational mesh
mesh Th = square(nds,nds); // generate a triangular mesh of a square domain
with nds+1 points per edge
//*/
```

2. Plot the convergence of the errors and estimators against the total number of degrees of freedom (DoFs).

3. Check in the command window what is the convergence rate on the sequence of uniformly refined meshes. Is this convergence rate optimal? Please also comment on the obtained effectivity indices.

4. Compute the "P2" FE approximation and the "RT2" and "P2dc" equilibrated flux. Check the convergence rate on the sequence of uniformly refined meshes.

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

1. One should obtain results as in Figures 1 and 2. The FEM solution is more and more accurate upon refining the meshes.

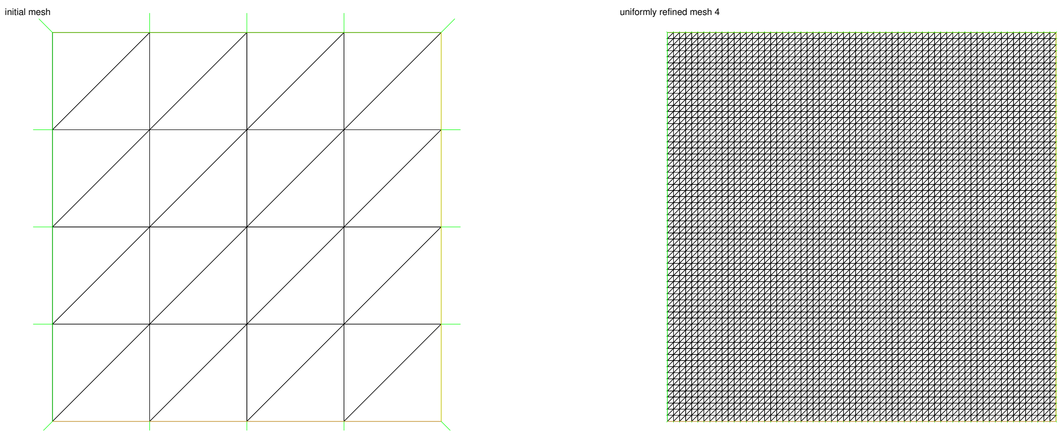


Figure 1: The initial mesh (left) and the finest refined mesh (right) for the smooth example (3)

2. Please go into the directory containing the data generated by the code `TP2.edp` and run the following command:

   `gnuplot ConvergenceRate.plt;` // plot the datum

   (Or just double click on `ConvergenceRate.plt` in Windows after having installed Gnuplot from http://www.gnuplot.info/).

   Then, you should see the a convergence result as in Figure 3, with on the x-axis the total number of degrees of freedom (DoFs) and on the y-axis the energy error and estimator. It is advantageous to use the log-log scale, since then one obtains a straight line.

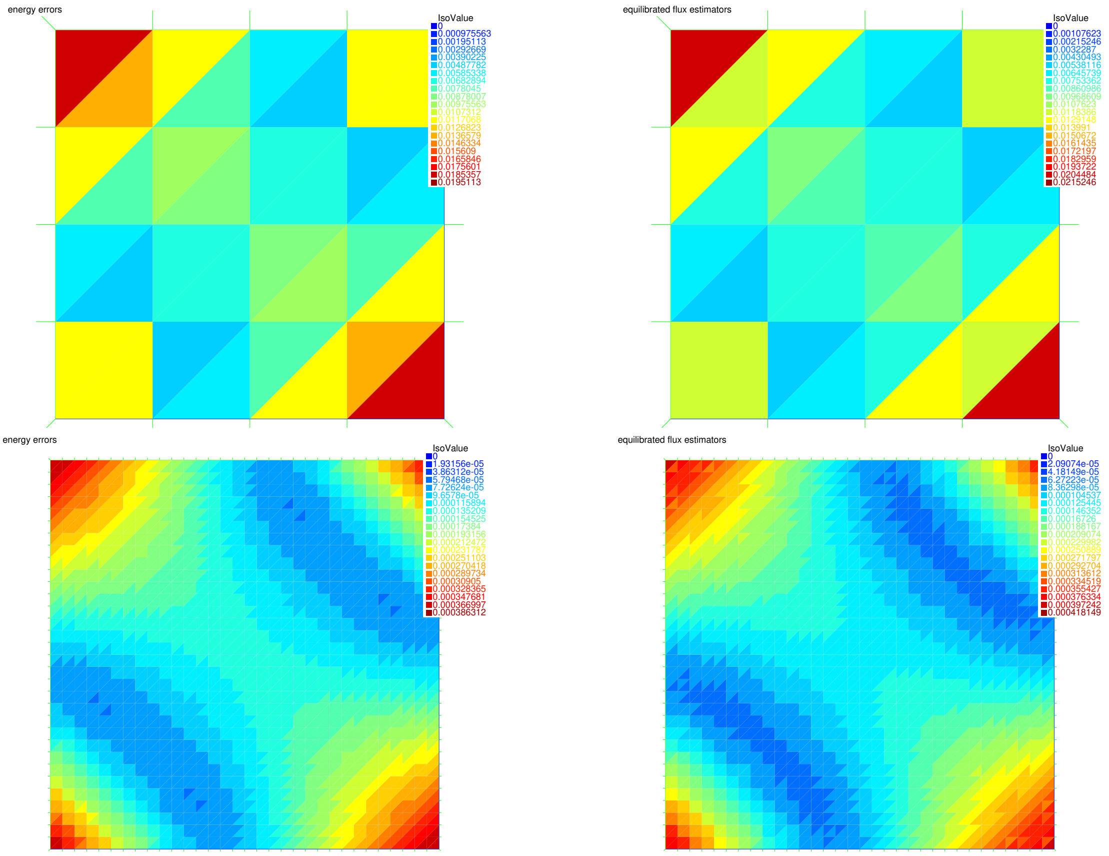


Figure 2: Elementwise errors $\|\nabla(u - u_\ell)\|_K$ (left) and elementwise estimators $\left[\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K + \frac{h_K}{\pi}\|f - \Pi_{p'} f\|_K\right]$ (right) on the initial mesh (top) and on the finest-refined mesh (bottom) for the smooth example (3) and $p = 1$

3. Going back to the command window, one should find the convergence rates and effectivity indices of the numerical solution computed on the sequence of uniformly refined meshes. You should see the following data:

   `Rate of convergence 3`

   `0.5675315829 0.5397301454 0.5212486906`

   `Effectivity indices 4`

   `1.044909373 1.043766139 1.045481868 1.046058539`

   In this case of a smooth solution, the convergence rate is $\mathcal{O}(h^p)$ which becomes $\mathcal{O}(\mathrm{DoFs}^{-\frac{p}{2}})$ in terms of DoFs. So, in the present case with $p = 1$, we need to find $\mathcal{O}(\mathrm{DoFs}^{-\frac{1}{2}})$, which is what we (approximately) observe. In addition, the effectivity index takes very small values close to 1.04.

4. One should set up “P2” FE approximation and the “RT2” and “P2dc” equilibrated flux, and then run the code again. You should see the following data in the command window:

   `Rate of convergence 3`

   `1.073175248     1.040710629     1.021428279`

   `Effectivity indices 4`

   `1.019698545     1.013056183     1.009550153     1.007757069`

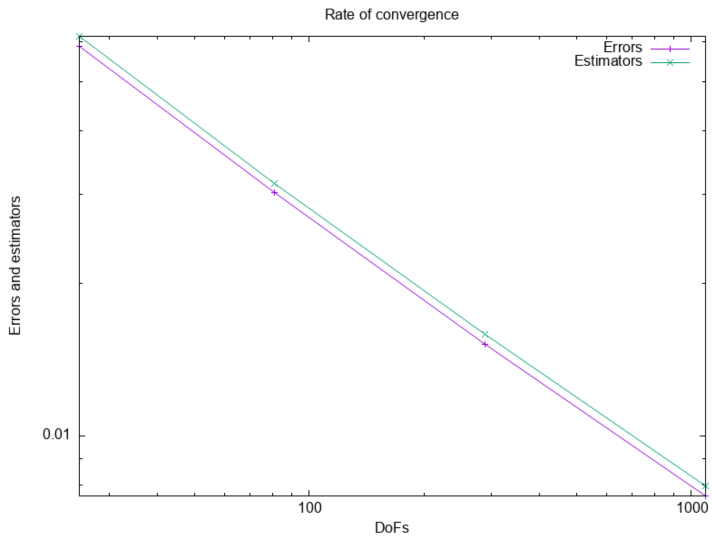


Figure 3: Convergence of the energy errors and of a posteriori error estimators for the smooth example (3) under uniformly refined meshes with $p = 1$

The above data give the convergence rate $\mathcal{O}(\mathrm{DoFs}^{-1})$ of the finite element method with $p = 2$ under the uniform mesh refinement. This is expected from the theoretical result $\mathcal{O}(\mathrm{DoFs}^{-\frac{p}{2}})$. In addition, the effectivity indices take rather stable values close to 1.01. The convergence plot is given in Figure 4.

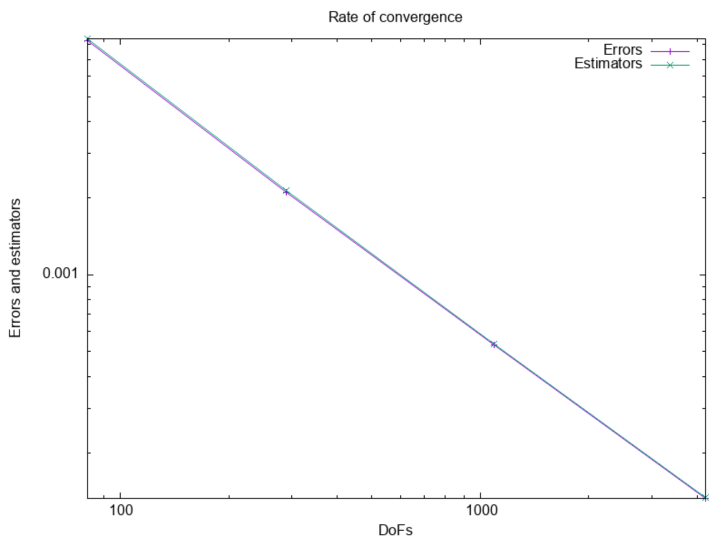


Figure 4: Convergence of the energy errors and of a posteriori error estimators for the smooth example (3) under uniformly refined meshes with $p = 2$

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

1. Compute the "P1" FE approximation and the "RT1" and "P1dc" equilibrated flux on a sequence of uniformly refined meshes for the singular example (4). Plot the obtained meshes and the elementwise distributions of the actual error and of the a posteriori error estimators. Please use the following parameters:

```
int nds = 4; // number of mesh points on one unit boundary edge
FinalLevel = 3; // maximal refinement level
```

```
bool RunSmooth = 0; // 1 means smooth example, 0 means singular example
bool RunAdaptive = 0; // 1 means adaptive mesh refinement, 0 means uniform mesh refinement
```

Remove the comment `/*` and define the data for the singular example:

```
///*
// CASE 2 (vertex singularity)
func theta=atan2(y, x)-2*pi*fmin(sign(y),0);
func r=(x^2+y^2)^(1/2.0);
......
mesh Th = buildmesh(b1(nds) + b2(nds) + b3(2*nds) + b4(2*nds)
+ b5(nds) + b6(nds));
//*/
```

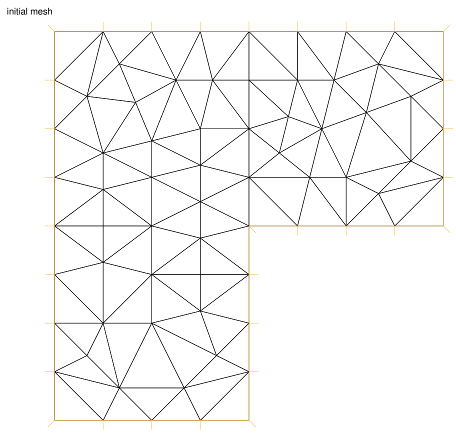
2. Plot the convergence of the errors and estimators against the total number of degrees of freedom (DoFs).

3. Check in the command window what is the convergence rate on the sequence of uniformly refined meshes. Is this convergence rate optimal? Please also comment on the obtained effectivity indices.

4. Compute the "P2" FE approximation and the "RT2" and "P2dc" equilibrated flux. Check the convergence rate on the sequence of uniformly refined meshes.

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

1. One should obtain results as in Figures 5 and 6. The FEM solution is still more and more accurate on uniformly refined meshes. The error and estimator both take large values close to the origin.



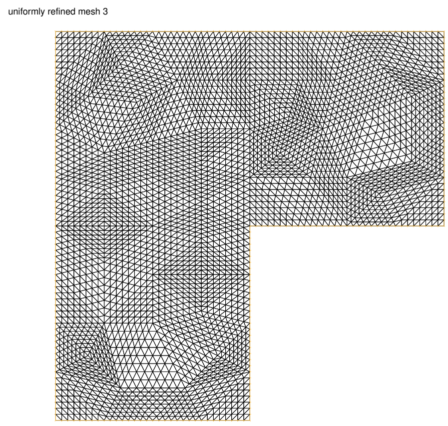


Figure 5: The initial mesh (left) and the finest refined mesh (right) for the singular example (4)

2. One should get into the directory containing the data generated by the code `TP2.edp` and again run the following command:

   `gnuplot ConvergenceRate.plt;` // plot the datum

   Then, you should see the a convergence result as in Figure 7.

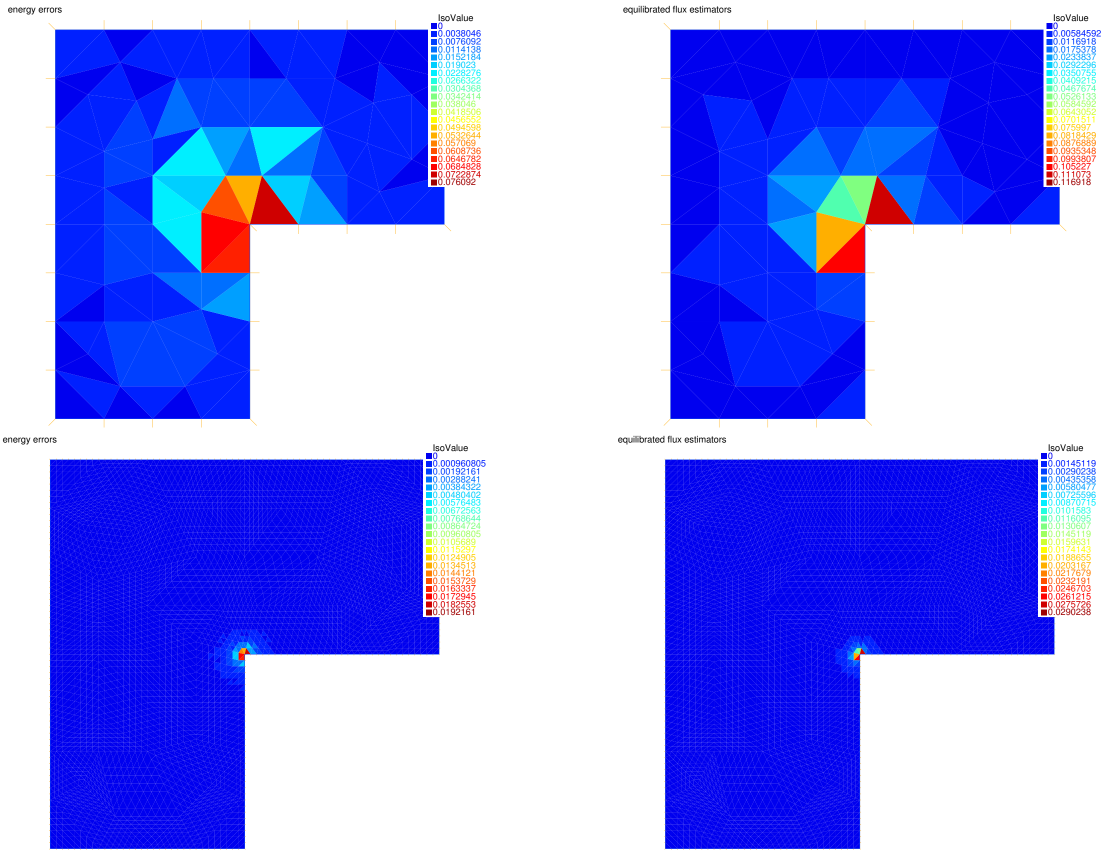


Figure 6: Elementwise errors $\|\nabla(u - u_\ell)\|_K$ (left) and elementwise estimators $\left[\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K + \frac{h_K}{\pi}\|f - \Pi_{p'} f\|_K\right]$ (right) on the initial mesh (top) and on the finest-refined mesh (bottom) for the singular example (4) and $p = 1$

3. Going back to the command window, one should find the convergence rates and effectivity indices of the numerical solution computed on the sequence of uniformly refined meshes. You should see the following data:

   ```
   Rate of convergence 3
   0.3526639324    0.3412467251    0.3360477061
   Effectivity indices 4
   1.241172733    1.226841282    1.219652978    1.216007697
   ```

   The above data give the convergence rate $\mathcal{O}(\text{DoFs}^{-\frac{1}{3}})$ for FEM with $p = 1$ under the uniform mesh refinement. This is expected from the theory since, recall, $u \in H^{\frac{5}{3}-\varepsilon}(\Omega)$ only; $\mathcal{O}(\text{DoFs}^{-\frac{1}{3}})$ is equivalent to $\mathcal{O}(h^{2/3})$ in terms of the mesh size h. This rate is, however, not optimal in terms of DoFs, as we will see in Exercice 3 below. The effectivity index is still stable but takes slightly increased values of around 1.23.

4. One should set up the "P2" FE approximation and the "RT2" and "P2dc" equilibrated flux and then run the code again. You should see the following data in the command window:

   ```
   Rate of convergence 3
    0.3534450138    0.342831769    0.3380040416
   Effectivity indices 4
   1.433017597    1.43406561    1.434120074    1.434114739
   ```

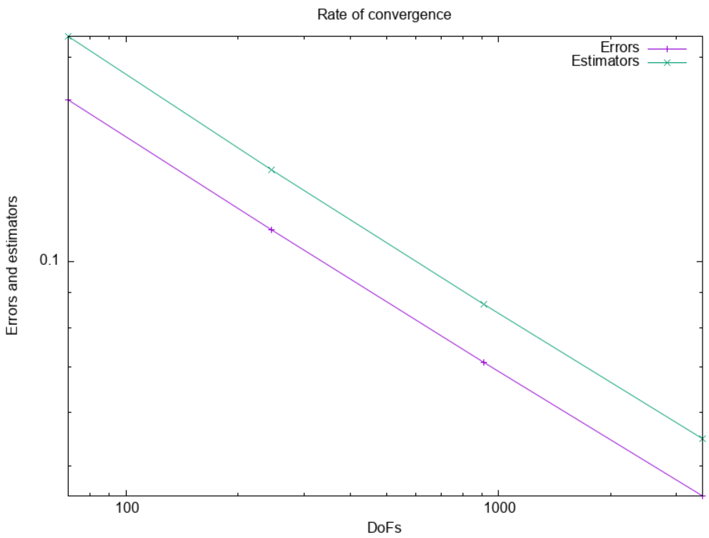


Figure 7: Convergence of the energy errors and of a posteriori error estimators for the singular example (4) under uniformly refined meshes with $p = 1$

The above data show that the convergence rate for FEM with $p = 2$ under the uniform mesh refinement is still $\mathcal{O}(\text{DoFs}^{-\frac{1}{3}})$ (i.e., $\mathcal{O}(h^{2/3})$ in terms of h), which is the same value as for $p = 1$; no improvement of the convergence rate appears here, in contrast to Exercice 1. This convergence rate is illustrated in Figure 8 (note that the values are slightly smaller than in Figure 7). This rate is again not optimal in terms of DoFs but we will see below in Exercice 3 that optimal rates are obtained under adaptive mesh regeneration. The effectivity indices are again stable but slightly increased, with values of around 1.43.

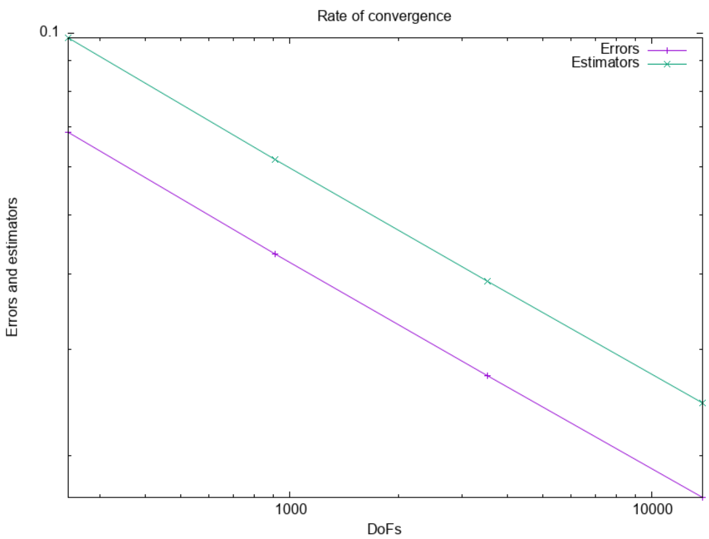


Figure 8: Convergence of the energy errors and of a posteriori error estimators for the singular example (4) under uniformly refined meshes with $p = 2$

The above numerical results confirm the theoretical expectation that the convergence rate of FEM for a singular solution is suboptimal for any order $p \geq 1$ in terms of DoFs under uniform mesh refinement: for uniform mesh refinement, we will always obtain $\mathcal{O}(\text{DoFs}^{-\frac{1}{3}})$ or $\mathcal{O}(h^{2/3})$, independently of the polynomial degree p. We will see that this will crucially change with proper mesh adaptation in Exercice 3.

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

Try to take advantage of the a posteriori error estimators $\eta_K(u_\ell)$ computed in each mesh element $K \in \mathcal{T}_\ell$. The aim is not to refine all mesh elements $K \in \mathcal{T}_\ell$ as previously, but instead only those with a high value $\eta_K(u_\ell)$ of the estimated error. We will more precisely choose a parameter $0 < \theta \leq 1$ and 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 only try to refine the elements in the marked set $\mathcal{M}_\ell$ by a given factor `RefFactor`. The procedure is described in the FreeFem++ script block `if (RunAdaptive)` and is based on the FreeFem++ command

```
Th = adaptmesh(Th, ElSizes, IsMetric=1, keepbackvertices=0, nbvx=1000000);
```

This actually generates a new mesh with the marked elements approximately `RefFactor` smaller than the previous ones.

1. Compute the "P1" FE approximation and the "RT1" and "P1dc" equilibrated flux on a sequence of adaptively generated meshes for the singular example (4). Plot the obtained meshes and the elementwise distributions of the actual error and of the a posteriori error estimators. Please use the following parameters:

   `int nds = 4;` // number of mesh points on one unit boundary edge

   `FinalLevel = 9;` // maximal refinement level

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

   `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

2. Plot the convergence of the errors and estimators against the total number of degrees of freedom (DoFs).

3. Check in the command window what is the convergence rate on the sequence of adaptively generated meshes. Is this convergence rate optimal? Please also comment on the obtained effectivity indices.

4. Compute the "P2" FE approximation and the "RT2" and "P2dc" equilibrated flux. Check the convergence rate on the sequence of adaptively generated meshes.

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

1. One should obtain a sequence of adaptively generated meshes, actual errors, and a posteriori error estimators as in Figures 9 and 10. The errors and estimators both take large values close to the origin, which causes an important grading of the later meshes towards the origin. But even on meshes very fine towards the origin, the elements touching the origin still contain large errors.

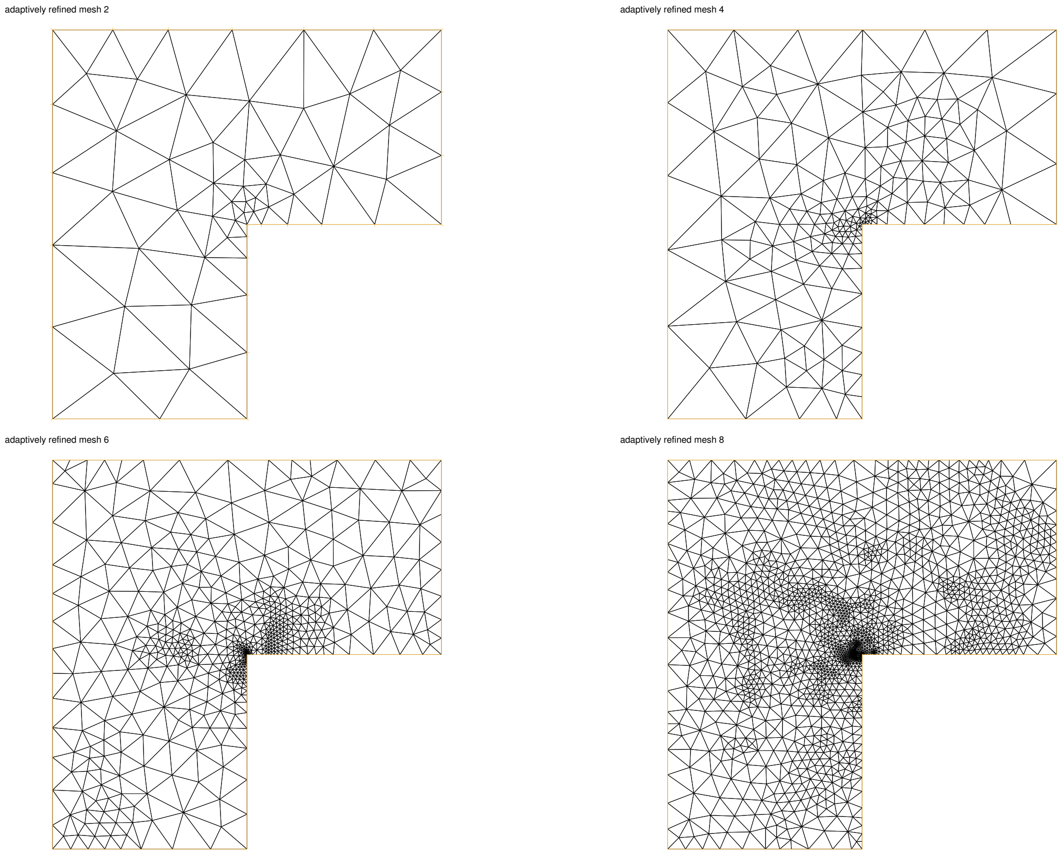


Figure 9: Level 2 adaptively generated mesh (top left), level 4 adaptively generated mesh (top right), level 6 adaptively generated mesh (bottom left), and level 8 adaptively generated mesh (bottom right) for the singular example (4) and $p = 1$

2. One should get into the directory containing the data generated by the code `TP2.edp` and run the following command:

   `gnuplot ConvergenceRate.plt;` // plot the datum

   Then a convergence result as in Figure 11 appears. Actually, FreeFem++ does not refine meshes but rather always generates a new mesh; for this reason, the number of DoFs can decrease at the same time with the error (and estimators). This would not be the case for nested finite element spaces issued from refined meshes.

3. Going back to the command window, one should find the convergence rates and effectivity indices of the numerical solution computed on the sequence of uniformly refined meshes. You should see the following data:

   ```
   Rate of convergence 9
   -0.7669871694   2.939635834  0.3425314836   0.8330618394   0.4913908552
   0.4115164255   0.5529103233   0.5082601328   0.5569987876
   Effectivity indices 10
   1.241172733   1.215444619   1.14398896   1.108194504   1.088578605
   1.073269585   1.066122737   1.063237346   1.062129856   1.062966932
   ```

   The above data show that the convergence rate for FEM with $p = 1$ under the adaptive mesh generation becomes $\mathcal{O}(\mathrm{DoFs}^{-\frac{1}{2}})$ which is substantially better than $\mathcal{O}(\mathrm{DoFs}^{-\frac{1}{3}})$ in Exercice 2 with uniform mesh refinement. Please note that a rate of

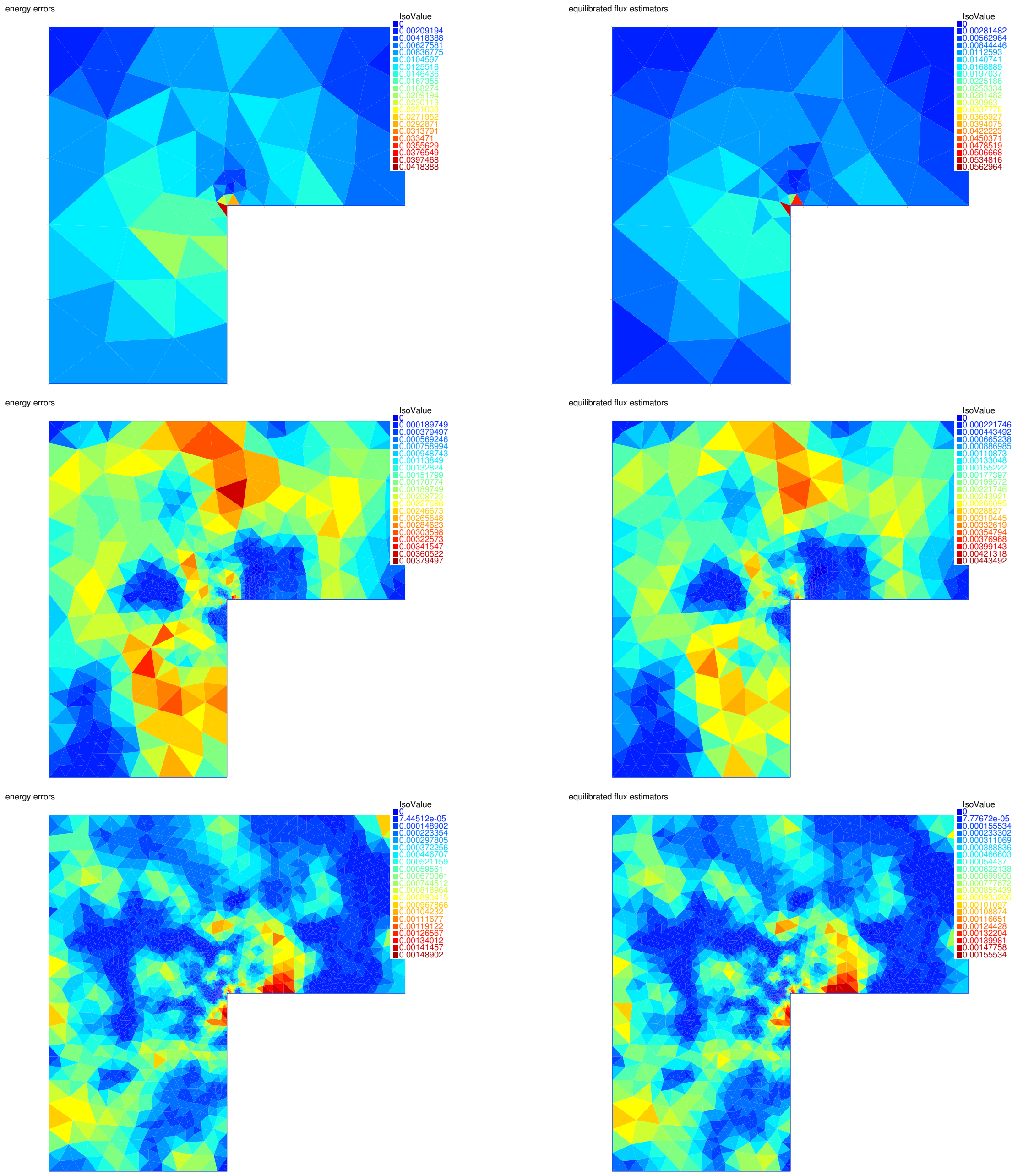


Figure 10: Elementwise errors $\|\nabla(u - u_\ell)\|_K$ (left) and elementwise estimators $\left[\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K + \frac{h_K}{\pi}\|f - \Pi_{p'} f\|_K\right]$ (right) on level 2 mesh (top), level 6 mesh (middle), and level 8 mesh (bottom) for the singular example (4) and $p = 1$

convergence in terms of the mesh size h, of the form $\mathcal{O}(h^\alpha)$, no more has any good meaning. Indeed, the mesh element diameters are nonuniform here and, moreover, the maximal mesh size may not even tend to zero. In conclusion, adaptive mesh generation/refinement improves the convergence rate with respect to uniform mesh refinement and gives a rate that is actually optimal in terms of DoFs: no better rate is possible, neither theoretically, nor practically. The obtained effectivity indices still stay very close to the optimal value of one, though the meshes are highly nontrivial here; they actually improve with respect to Exercice 2, attaining roughly 1.06 for the finest meshes.

4. One should set up the “P2” FE approximation and the “RT2” and “P2dc” equilibrated flux and then run the code again. You should see the following data in the

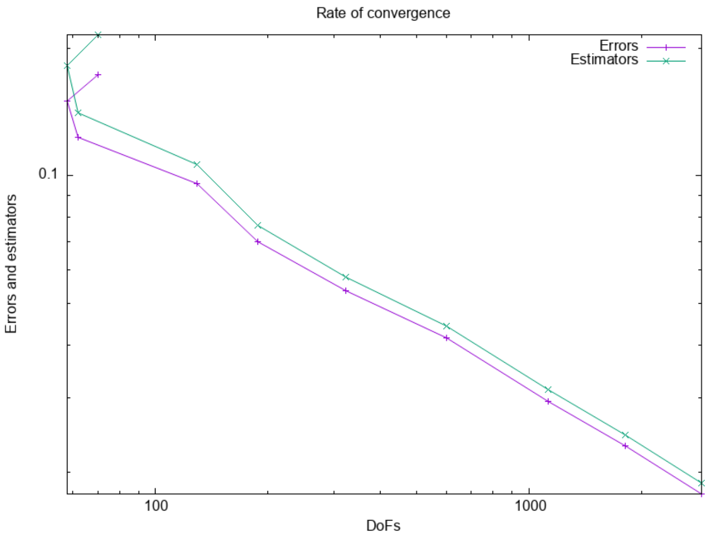


Figure 11: Convergence of the energy errors and of a posteriori error estimators for the singular example (4) under adaptively generated meshes with $p = 1$

command window:

```
Rate of convergence 9
-1.360154142   -6.205630657   15.50040121   2.165995176   1.075495519
0.7647940308   0.973671427   0.8776198796   0.9509795819
Effectivity indices 10
1.433017597   1.308391581   1.280721626   1.250852832   1.219638215
1.060747483   1.038959759   1.034408075   1.023007578   1.022596182
```

The above data show that the convergence rate for FEM with $p = 2$ under the adaptive mesh generation becomes $\mathcal{O}(\mathrm{DoFs}^{-1})$, which is substantially better than $\mathcal{O}(\mathrm{DoFs}^{-\frac{1}{3}})$ in Exercice 2 with uniform mesh refinement, cf. also Figure 14. Even if the solution is singular, one gains from higher-order! But one needs to employ adaptive mesh refinement for this. Again, adaptive mesh generation/refinement improves the convergence rate with respect to uniform mesh refinement and gives a rate that is optimal in terms of DoFs: no better rate is possible, neither theoretically, nor practically. As above for $p = 1$, the obtained effectivity indices stay very close to the optimal value of one and actually improve with respect to Exercice 2, attaining roughly 1.02 for the finest meshes. Figures 12 and 13 then present the sequences of adaptively generated meshes and the actual and predicted error distributions, which are again excellent.

In conclusion, the numerical experiments confirm that the convergence rate of FEM for a singular solution is optimal in terms of DoFs, with the rate $\mathcal{O}(\mathrm{DoFs}^{-\frac{p}{2}})$ with any order p, under the adaptive mesh generation.

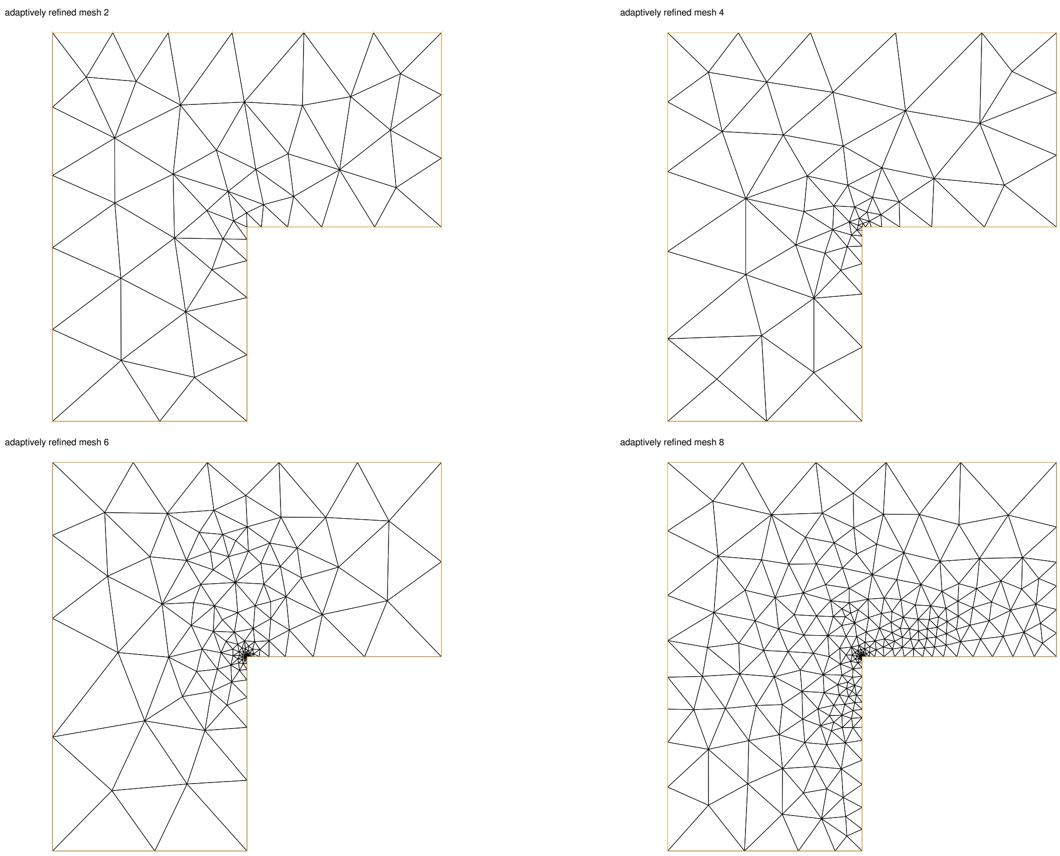


Figure 12: Level 2 adaptively generated mesh (top left), level 4 adaptively generated mesh (top right), level 6 adaptively generated mesh (bottom left), and level 8 adaptively generated mesh (bottom right) for the singular example (4) and $p = 2$

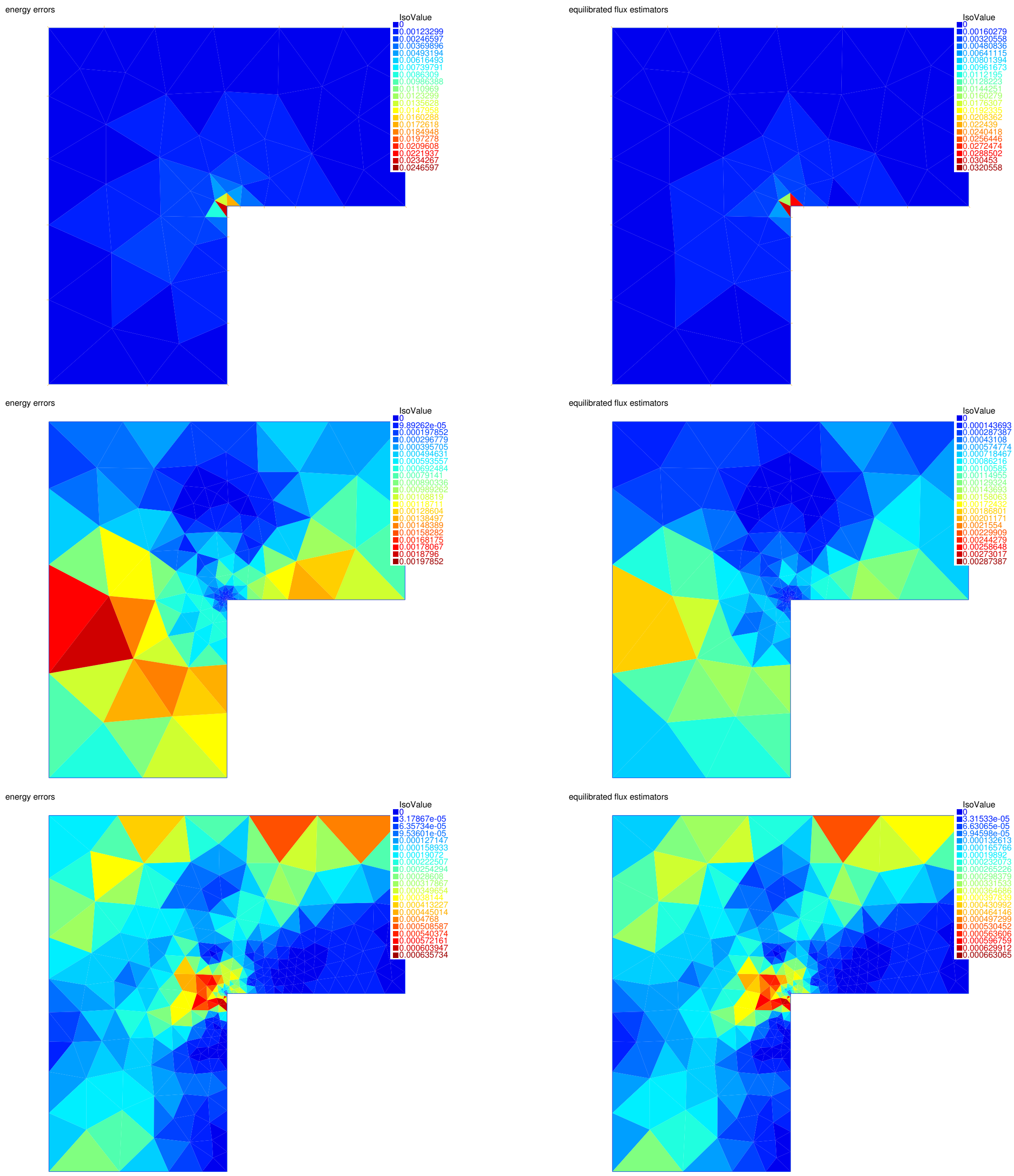


Figure 13: Elementwise errors $\|\nabla(u - u_\ell)\|_K$ (left) and elementwise estimators $\big[\|\nabla u_\ell + \boldsymbol{\sigma}_\ell\|_K + \frac{h_K}{\pi}\|f - \Pi_{p'} f\|_K\big]$ (right) on level 2 mesh (top), level 6 mesh (middle), and level 8 mesh (bottom) for the singular example (4) and $p = 2$

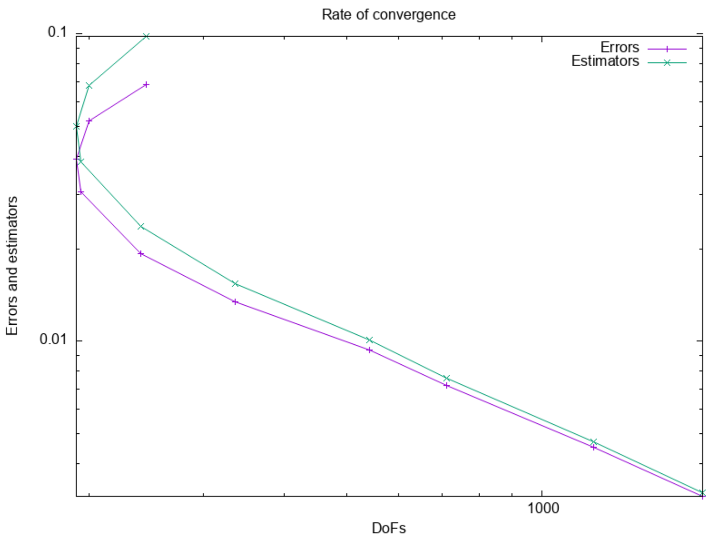


Figure 14: Convergence of the energy errors and of a posteriori error estimators for the singular example (4) under adaptively generated meshes with $p = 2$