

Estimations d'erreur *a posteriori* pour les méthodes d'éléments finis

E. Creusé - R. Tittarelli
Journées numériques
Nice, 29-30 Novembre 2016

Travaux Pratiques 2 : Estimateurs résiduels, cas 2D.

Code à télécharger : https://math.univ-lille1.fr/~creuse/Nice/Codes_Nice.html

Correction : https://math.univ-lille1.fr/~creuse/Nice/Correction_TP2_Nice.html

1 Le cas conforme

1.1 Le code de calcul

Soit $\Omega =]0, 1[^2$ de frontière $\Gamma = \partial\Omega$ et $f \in L^2(\Omega)$. Le programme matlab **mainc** qui vous a été transmis résout numériquement le problème aux limites suivant :

$$\begin{cases} -\Delta u(x, y) &= f(x, y) \quad \forall (x, y) \in \Omega, \\ u(x, y)|_{\Gamma} &= u_{ex}(x, y) \text{ donné,} \end{cases} \quad (1)$$

par méthode des éléments finis \mathbb{P}_1 -Lagrange conformes. La structure du code est très similaire à celle utilisée dans le cas 1D durant le premier TP.

- 1) Repérer dans le code les étapes et fonctions caractéristiques du code éléments finis :
 - a) Les fonctions permettant de spécifier les données théoriques : **f**, **uex**, **duexdx**, **duexdy**;
 - b) Le maillage à utiliser : **carre-ri.msh**, avec $1 \leq i \leq 8$ ou, dans le cas de l'utilisation de la pde toolbox, la précision du maillage **hmax**;
 - c) La création des descripteurs et leur signification : **ntriangles**, **nnodes**, **p1**, **t1**, **boundarynodes** ;
 - d) La matrice du système linéaire **A**, le second membre **B**, le vecteur solution **uh**;
 - e) Le calcul de la matrice locale de rigidité et du second membre local (**ComputeLocalStiffnessMatrixandRHS**), la boucle d'assemblage de la matrice globale de rigidité (**Assemblage**);
 - f) L'imposition des conditions aux limites de Dirichlet¹ (**setBC**), utilisant en particulier la solution exacte **uex** sur la frontière;

¹Tel que le code est écrit, le traitement des conditions de Dirichlet **non** homogènes est également possible.

- g) La résolution du système linéaire et la visualisation de la solution (**visu** ou, dans le cas de l'utilisation de la pde toolbox, **pdeplot** et **pdesurf**);
- h) Le calcul des erreurs $L^2(\Omega)$ et $H^1(\Omega)$ commises (**computeerrorL2** et **computeerrorsemiH1**).
- 2) On note $h = \max_{T \in \mathcal{T}} h_T$, avec h_T la longueur de la maille T . En utilisant une série de maillages (**carre-r1.msh** à **carre-r4.msh** ou en spécifiant la valeur de **hmax** et en utilisant la pde toolbox), mettre en évidence les taux de convergence de la méthode pour chacune des deux normes ² :

$$f(x) = 8\pi^2 \sin(2\pi x) \sin(2\pi y).$$

1.2 Mise en place de l'estimateur d'erreur résiduel

- Pour toute maille T de la triangulation, on définit l'estimateur d'erreur résiduel η_T par :

$$\eta_T^2 = h_T^2 \|\pi_0(f)\|_T^2 + \frac{1}{2} \sum_{E \in \partial T \setminus \Gamma} h_E \|[\nabla u_h \cdot n_E]_E\|_E^2.$$

- Pour toute maille T de la triangulation, on définit le terme d'approximation ζ_T par :

$$\zeta_T = h_T \|\pi_0(f) - f\|_T.$$

- On définit respectivement l'estimateur résiduel global et le terme d'approximation global par :

$$\eta^2 = \sum_{T \in \mathcal{T}} \eta_T^2; \quad \zeta^2 = \sum_{T \in \mathcal{T}} \zeta_T^2.$$

- On adopte enfin les définitions suivantes :

$$\eta_{T,V}^2 = h_T^2 \|\pi_0(f)\|_T^2; \quad \eta_{T,B}^2 = \frac{1}{2} \sum_{E \in \partial T \setminus \Gamma} h_E \|[\nabla u_h \cdot n_E]_E\|_E^2;$$

$$\eta_V^2 = \sum_{T \in \mathcal{T}} \eta_{T,V}^2; \quad \eta_B^2 = \sum_{T \in \mathcal{T}} \eta_{T,B}^2.$$

- 1) Ecrire la fonction **eta-V-2** qui estime la valeur de $\eta_{T,V}^2$ pour toute maille T de la triangulation ainsi que la valeur de $\eta_V^2 = \sum_{T \in \mathcal{T}} \eta_{T,V}^2$:

function [etaTV2,valeur-eta-V-2]=eta-V-2(uh)

global ntriangles p1 t1 hT;

etaTV2=zeros(ntriangles,1);

valeur-eta-V-2=0;

²On a clairement dans ce cas $u(x) = \sin(2\pi x) \sin(2\pi y)$.

[A COMPLETER !]

end

- 2) Ecrire la fonction **Compute-gradient-uh** qui calcule la valeur du gradient de **uh** dans chaque maille du maillage :

```
function[Graduh]=Compute-gradient-uh(uh)
```

```
global ntriangles t1 p1;
```

```
Graduh=zeros(2,ntriangles);
```

[A COMPLETER !]

end

- 3) Ecrire la fonction **eta-B-2** qui estime la valeur de $\eta_{T,B}^2$ pour toute maille T de la triangulation ainsi que la valeur de $\eta_B^2 = \sum_{T \in \mathcal{T}} \eta_{T,B}^2$:

```
function[etaB2,etaTB2]=eta-B-2(Graduh)
```

```
global ntriangles nsegments p1 e1 hE;
```

```
etaTB2=zeros(ntriangles, 1);
```

```
etaB2=0;
```

[A COMPLETER !]

end

- 4) Ecrire la fonction **zeta-2** qui estime la valeur de ζ_T^2 pour toute maille T de la triangulation ainsi que la valeur de $\zeta^2 = \sum_{T \in \mathcal{T}} \zeta_T^2$:

```
function[valeur-zeta2,zetaT2]=zeta-2()
```

```
global ntriangles p1 t1 hT;
```

```
zetaT2=zeros(ntriangles,1);
```

```
valeur-zeta2=0;
```

[A COMPLETER !]

end

- 5) En complétant le programme principal et en utilisant les fonctions précédemment développées, étudier *la fiabilité* de l'estimateur, c'est à dire la majoration de l'erreur du type :

$$|u - u_h|_{1,\Omega}^2 \lesssim \eta^2 + \zeta^2,$$

à l'instar de ce qui a été fait dans le premier TP en dimension 1.

- 6) En vous inspirant de la fonction **visu**, donner une représentation graphique de l'estimateur sur le domaine de calcul (on pourra par exemple définir un vecteur de même taille que **uh**, associant à chaque noeud du maillage une valeur moyenne de l'indicateur d'erreur).
- 7) Reprendre l'ensemble de ces questions avec cette fois :

- Ω le "L-shape" représenté sur la Figure 1,
- $u_{ex}(r, \theta) = r^{2/3} \sin\left(\frac{2\theta}{3}\right)$,
- $f \equiv 0$.

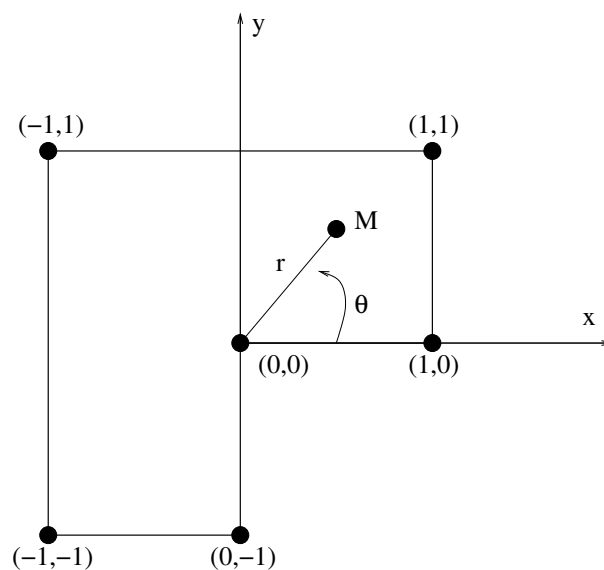


Figure 1: Le "L-Shape".

Conclusion ?

- 8) Utiliser le code de calcul pour visualiser les isovaleurs de la fonction de courant s'installant autour d'un profil d'aile NACA. (Les détails seront expliqués durant la séance). Mettre en évidence le lieu où les estimateurs résiduels demandent de raffiner le maillage.

2 Le cas non conforme

On désire maintenant toujours résoudre le problème (1), mais cette fois en utilisant les éléments finis \mathbb{P}_1 -Lagrange non conformes. La structure du nouveau code qui vous est fourni est très similaire à celle développée pour le cas conforme.

1. Repérer dans le code les étapes et fonctions caractéristiques du code éléments finis. On pourra procéder par comparaison avec les fonctions développées dans le cas conforme en repérant les quelques différences essentielles : valeurs des gradients de fonctions de base, utilisation des descripteurs d'arêtes, modification de la procédure de visualisation.
2. Adapter la fonction **Compute-gradient-uh** développée dans le cas conforme au cas non conforme. On remarquera que **eta-V-2**, **eta-B-2** et **zeta-2** peuvent rester inchangées.
3. Visualiser les zones du domaine pour lesquelles l'estimateur détecte l'erreur maximale, pour le cas test du carré, du "L-Shape" et du profil NACA. Que constatez-vous, en particulier par comparaison avec le cas conforme ?
4. Implémenter la contribution supplémentaire de l'estimateur nécessaire dans le cas non conforme :

$$\eta_{T,Btan}^2 = \frac{1}{2} \sum_{x_i \in \partial T \setminus \Gamma} h_E \|\nabla u_h \cdot t_E\|_E^2,$$

l'estimateur η_T étant maintenant défini par :

$$\eta_T^2 = h_T^2 \|\pi_0(f)\|_T^2 + \frac{1}{2} \sum_{E \in \partial T \setminus \Gamma} h_E (\|\nabla u_h \cdot n_E\|_E^2 + \|\nabla u_h \cdot t_E\|_E^2).$$

Pour cela, développer la fonction **eta-Btan-2** (très similaire à **eta-B-2**, un bon copier/coller et une modification mineure sera la meilleure stratégie !).

```
function[etaBtan2,etaTBtan2]=eta-Btan-2(Graduh)
```

```
global ntriangles nsegments p1 e1 hE;
```

```
etaTBtan2=zeros(ntriangles, 1);
```

```
etaBtan2=0;
```

```
[A COMPLETER !]
```

```
end
```

5. Visualiser à nouveau les éléments demandés à la question 3, mais cette fois avec toutes les contributions de l'estimateur. Conclusion ?