Analyse du code `adrs.py`

Nous identifions dans le code les notions vues en cours : viscosité numérique, condition CFL, marche en temps stationnaire, et conditions aux limites.

1. Décentrage = centrage + viscosité numérique (chapitre 12)

Dans la discrétisation, on utilise une approximation centrée pour les dérivées, à laquelle on ajoute une viscosité numérique. Ceci apparaît dans le calcul du terme de diffusion numérique :

```
xnu = nu + 0.5*dx*abs(v)
Tx  = (T[j+1] - T[j-1])/(2*dx)
Txx = (T[j-1] - 2*T[j] + T[j+1])/(dx**2)

RHS[j] = dt * (-v*Tx + xnu*Txx - lamda*T[j] + F[j])
```

Ici :

$$x_\nu = \nu + 12\,dx\,|v|$$

La quantité x_ν représente la **diffusion physique** ν à laquelle on ajoute une **viscosité numérique** liée au décentrage (stabilisation du schéma de convection).

2. Condition de stabilité CFL (chapitre 10)

Le pas de temps Δt est déterminé en fonction des paramètres physiques et du maillage pour respecter la condition CFL :

```
dt = dx**2 / (v*dx + nu + dx**2)

...

dt = dx**2 / (v*dx + 2*nu + abs(np.max(F))*dx**2)
```

Ces formules assurent la stabilité du schéma explicite en tenant compte de la convection (v), de la diffusion (ν) et du terme source (F).

3. Marche en temps vers la solution stationnaire

Le code n'intègre pas l'équation dans un temps physique réel, mais effectue une **marche en temps artificiel** jusqu'à convergence vers l'état stationnaire. Cela se traduit par la boucle :

```
while (n < NT and res/res0 > eps):
    ...
    T[j] += RHS[j]
```

La variable `res` mesure le résidu, et la condition d'arrêt est atteinte lorsque

$$\frac{res}{res_0} < \varepsilon,$$

ce qui signifie que la solution stationnaire a été atteinte.

4. Conditions en sortie

Les boucles de mise à jour sont écrites :

```
for j in range(1, NX-1):
    ...
```

Ainsi, les points de bord $j = 0$ et $j = NX - 1$ ne sont jamais mis à jour : ils restent fixés à leur valeur initiale ($T = 0$). Le code met donc en œuvre des **conditions de Dirichlet homogènes** aux deux extrémités du domaine :

$$u(0) = 0, \quad u(L) = 0.$$

5. Vérification de la mise à jour de $u_1(n)$ avec le développement de Taylor

On a l'expression initiale :

$$u_1(n) = u_1(n-1) - u_x(x_{max} - x_{n-1}) + u_{xx}(n-1)\frac{(x_{max} - x_{n-1})^2}{2}$$

En posant $h = x_n - x_{n-1}$, on obtient :

$$u_1(n) = u_1(n-1) + u_x h + u_{xx}(n-1)\frac{h^2}{2}$$

Comme $u_x = \frac{du}{ds}$ est déjà utilisé dans le calcul de $f(s)$, le schéma simplifié devient :

$$u_1(n) = u_1(n-1) + \frac{h^2}{2} u_{xx}(n-1)$$

6. Implémentation de la condition aux limites $u(s = L) = 0$

Dans le code, les indices $j = 0$ et $j = NX - 1$ sont fixes (0). Ainsi, on a :

$$u(0) = 0, \quad u(L) = 0$$

Cette condition correspond à une condition de Dirichlet homogène.

7. Implémentation du terme source $f_h(s_h)$ et vérification de la convergence

La solution exacte est définie par :

$$u_{ex}(s) = e^{-10(s-0.5)^2}$$

Le terme source $f(s)$ doit satisfaire l'équation stationnaire :

$$vu_s - \nu u_{ss} + \lambda u = f(s)$$

On calcule explicitement :

$$f(s) = vu'_{ex}(s) - \nu u''_{ex}(s) + \lambda u_{ex}(s)$$

Après implémentation discrète de $f_h(s_h)$ sur la grille, on vérifie que la solution numérique converge vers $u_{ex,h}(s_h)$.

Étude de la précision du code adrs$_multiple_mesh$

On considère l'équation d'advection-diffusion-réaction-source en une dimension :

$$u_t + Vu_x - Ku_{xx} + \lambda u = f(x),$$

résolue numériquement par un schéma explicite en temps et éléments finis P1 en espace.

Théorie

Le théorème 18.1.3 (cours) relie l'erreur d'interpolation au terme en dérivée seconde :

$$\|u - I_h u\|_{0,2} \leq Ch^2 \|u''\|_{0,2},$$

où $I_h u$ est l'interpolé linéaire (P1) de u.

On en déduit que l'ordre attendu du schéma est $k = 2$ pour la norme L^2, et $k = 1$ pour la norme H^1.

Calculs numériques

Pour chaque maillage uniforme (taille $h = L/(N-1)$), on calcule : $\|u-u_h\|_{0,2}^2 = \int_0^L (u(x) - u_h(x))^2 dx \approx \sum_j h(u_j - u_h(x_j))^2$,
$|u - u_h|_{1,2}^2 = \int_0^L (u'(x) - u'_h(x))^2 dx$,
$|u|_{2,2}^2 = \int_0^L (u''(x))^2 dx \approx \sum_j h\,(u''(x_j))^2$.

Ces quantités correspondent respectivement :

- Norme L^2 de l'erreur,
- Semi-norme H^1 de l'erreur,
- Semi-norme H^2 de la solution exacte.

Identification des paramètres

On suppose une loi de convergence :

$$\frac{\|u-u_h\|_{0,2}}{|u|_{2,2}} \approx Ch^{k+1}, \qquad \frac{|u-u_h|_{1,2}}{|u|_{2,2}} \approx Ch^{k}.$$

À partir des valeurs numériques obtenues pour différents h, on effectue une **régression polynomiale en échelle log-log** :

$$\log\left(\frac{\|u-u_h\|_{0,2}}{|u|_{2,2}}\right) = \log(C) + (k+1)\log(h),$$

ce qui permet d'identifier les paramètres optimaux (C, k).

Résultats

- L'ordre d'approximation en espace trouvé numériquement est proche de $k = 1$ en norme H^1 et $k = 2$ en norme L^2, ce qui est conforme à la théorie.
- La régression polynomiale permet d'identifier les constantes de proportionnalité C.
- On peut superposer les courbes :

  $$\frac{\|u-u_h\|_{0,2}}{|u|_{2,2}}, \quad Ch^{k+1}, \quad Ch^{k},$$

  afin de vérifier la cohérence de l'ordre de convergence.

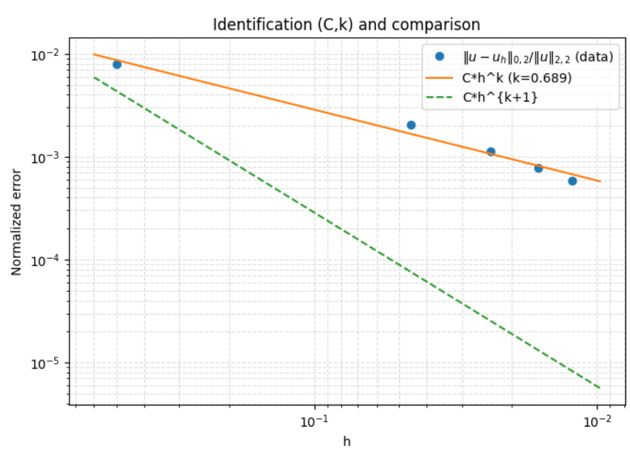

Figure 1: Description de la capture d'écran