TP2 – Équations non-linéaires

Théorie. La méthode de Newton (dite souvent Newton-Raphson en dimension $d \geq 1$) s'applique à la résolution d'un système d'équations $F(x) = 0$, où $F : \mathbb{R}^d \to \mathbb{R}^d$ est au moins de classe $\mathcal{C}^1$. L'algorithme consiste en l'itération récurrente suivante, indicée par $n \in \mathbb{N}$ et à valeurs dans $\mathbb{R}^d$:

$$x_{n+1} = x_n - [dF(x_n)]^{-1} F(x_n).$$

La convergence est garantie lorsque l'initialisation $x_0 \in \mathbb{R}^d$ est suffisamment proche de la solution recherchée x^* de $F(x) = 0$, à condition que l'application différentielle $dF(x^*) \in \mathcal{L}(\mathbb{R}^d, \mathbb{R}^d)$ soit inversible, tout ceci par exemple sous l'hypothèse de régularité $F \in \mathcal{C}^2$. La convergence est alors quadratique, i.e.

$$\exists C > 0,\ \forall n \in \mathbb{N}, \quad \|x_{n+1} - x^*\| \leq C \|x_n - x^*\|^2.$$

Quelques itérations suffisent alors pour obtenir une précision redoutable.

But du TP. *Il s'agit de mettre en œuvre la méthode de Newton et d'en estimer numériquement la vitesse de convergence effective sur quelques situations élémentaires, puis de traiter des exemples d'applications plus complexes.*
Une partie de ce TP est repris directement du sujet TP1 précédent. Ce peut être l'occasion de faire évoluer vos programmes.

Consignes d'organisation.

```
import numpy as np
import matplotlib.pyplot as plt
```

Mise en œuvre.

- ☐ Programmer la méthode de Newton dans le *cas monodimensionnel* ($d = 1$) pour une fonction F, de différentielle dF calculable formellement. Idéalement, le critère d'arrêt pourra faire intervenir un nombre maximal d'itération et un test du résidu $|F(x_n)| < \epsilon$.

1. Tester cette méthode d'abord sur l'exemple $F_1(x) = e^x - e^1$ et en vérifier la convergence locale. Modifier le programme de façon à obtenir en sortie tous les itérés calculés. Illustrer graphiquement la vitesse de convergence puis la quantifier : on pourra mettre à l'épreuve l'inégalité de convergence quadratique ci-dessus avec une échelle logarithmique.
Quel est l'ordre de convergence effectif dans le cas de la recherche des zéros de $F_2(x) = (e^x - e)^2$ et $F_3(x) = (e^x - e)^{5/2}$?

2. Application Trouver une approximation des valeurs $\lambda \in [0, 10]$ telles que $\displaystyle\int_0^1 \sin(\lambda e^{x^2})\, \mathrm{d}x = 1/2$.
Indication : on pourra utiliser la commande intégrée `scipy.integrate.quad` ou une stratégie maison pour évaluer les intégrales qui interviennent.

3. Le *cas multidimensionnel* ($d > 1$) requiert l'inversion d'un système linéaire à chaque itération (utiliser `np.linalg.solve`) et, si besoin de nouveau, un stockage des itérés vectoriels. Adapter le programme et l'utiliser pour mettre en œuvre la méthode de Newton afin de résoudre le système polynomial suivant

$$\begin{cases} x^2 + y^2 = 2 \\ x^2 - y^2 = 1 \end{cases}$$

Illustrer de nouveau la convergence une première fois en traçant les itérés dans le plan (x, y), et ensuite en quantifiant l'erreur et l'ordre de convergence. Tester plusieurs initialisations. On tracera sur la même figure le cercle et l'hyperbole qui interviennent dans les équations.

4. Application On souhaite obtenir une approximation de la solution $u \in \mathcal{C}^2([0,1])$ (dont on admettra l'existence) du problème différentiel aux limites :

$$u(0) = 1, \quad u(1) = 2, \quad \frac{d^2}{dx^2}(uv) = u^2, \ x \in [0,1], \quad v(x) = 1 + \cos^2(\pi x).$$

Pour ce faire, on se donne un entier $N \geq 2$ et on pose $V_n = 1 + \cos^2(\pi \frac{n}{N+1})$ pour $0 \leq n \leq N+1$. Le problème continu est approché par une formulation aux différences finies sur $[0,1]$. L'inconnue est un vecteur $U_n \in \mathbb{R}^N$ solution de

$$U_0 = 1, \quad U_{N+1} = 2, \quad (N+1)^2(U_{n+1}V_{n+1} - 2U_nV_n + U_{n-1}V_{n-1}) = U_n^2, \quad 1 \leq n \leq N.$$

Résoudre ce système approché pour N fixé suffisamment grand.

Pour poursuivre...

☐ Dans le cas monodimensionnel, programmer la *méthode de Newton modifiée* définie par l'itération :

$$x_{n+1} = x_n - p[dF(x_n)]^{-1}F(x_n).$$

où p est la « multiplicité » du zéro recherché ($p = 5/2$ pour la fonction F_3).

◇ Tester la résolution liée à F_2 et F_3 avec cet algorithme et estimer l'ordre de convergence effectif. Analyser mathématiquement l'ordre de convergence de cette méthode modifiée.

5. (Extrait du TP1)
On considère le problème aux limites suivant :

$$\begin{cases} -u''(x) + a(x)u(x) = f(x), \\ u(0) = u(1) = 0, \end{cases}$$

où les fonctions $a \geq 0$ et f sont données. Pour résoudre ce problème, on introduit le problème de Cauchy dépendant du paramètre α :

$$\begin{cases} -u''_\alpha(x) + a(x)u_\alpha(x) = f(x), \\ u_\alpha(0) = 0 \text{ et } u'_\alpha(0) = \alpha, \end{cases}$$

◇ Ce problème peut être résolu à l'aide d'une méthode d'intégration des équations différentielles ordinaires. Il suffit ensuite de trouver α tel que $u_\alpha(1) = 0$, ce qui n'est pas difficile puisque l'application $\varphi : \alpha \mapsto u_\alpha(1)$ est affine !

◇ Appliquer la même méthode au problème non-linéaire

$$\begin{cases} -u''(x) + a(x)u^3(x) = f(x), \\ u(0) = u(1) = 0, \end{cases}$$

pour lequel l'application $\varphi : \alpha \mapsto u_\alpha(1)$ est non-linéaire. On pourra alors utiliser la méthode de Newton ou une variante pour résoudre le problème de tir.