Projet 2 - TPs 2 et 3 - MT05

Printemps 2022

1 Présentation du deuxième projet

Les deux prochains TP consistent à travailler sur des méthodes itératives très répandues pour la résolution numérique d'équations. La méthode par dichotomie et la méthode de Newton permettent dans certaines conditions de résoudre une équation de type $f(x) = 0$. Ces méthodes seront implémentées et la vitesse de convergence des suites associées sera discutée.

Ce projet fait l'objet d'un compte rendu. Le rapport présentera les résultats (calculs théoriques ou obtenus à l'aide de Matlab, graphiques pertinents, analyse) et sera de 15 pages maximum. Les codes sources pourront être présentés en annexe du rapport, à condition d'être commentés. Il est tout à fait possible de rédiger les démonstrations et analyses de manière manuscrite. Il est évalué et comptera pour 10% de la note de l'UE. Le rapport est à rendre au plus tard au début du quatrième TP (début du troisième projet).

$\longrightarrow$ Les comptes-rendus électroniques doivent être déposés sur Elearning sous la forme d'un **unique fichier pdf**. Les comptes-rendus manuscrits peuvent-être rendus pendant les TPs ou bien scannés - lisiblement- puis soumis sur Elearning sous la forme d'un **unique fichier pdf**.

2 Rappels Matlab/Octave

Exemple 1 (Vecteurs) On peut définir un vecteur ou une matrice par ses éléments avec les séparateurs lignes ou colonnes ';' et ',' ou par intervalle $'a : b'(a \leq a + k \leq b)$ ou $'a : \delta : b'(a \leq a + \delta k \leq b)$.

```
> x=[1,2];
> x=[1;2];
> x=[0:7];
> x=[0:0.01:7];
```

Exemple 2 (Fonctions) Tous les opérateurs sont matriciels. Pour obtenir une évaluation de x^3 aux points 1 et 2, on ne peut pas écrire $[1, 2]^3$. En revanche, on peut utiliser une multiplication terme à terme : Ce sont les opérateurs $.*, ./, ...$ Les calculs sous Matlab sont rarement utilisés de manière symbolique (définir la fonction $f(x) = x^3$ pour tout x), mais sont pratiquement toujours numériques (définir la fonction $f(x) = x^3$ pour un ensemble de points).

```
> x=[1,2];
> y=[1,2].^3;
```

Exemple 3 (Fonctions prédéfinies) De nombreuses fonctions sont déjà définies sous Matlab : abs, sqrt, sin, cos,...

Exemple 4 (Fonctions) Il est souhaitable de créer une fonction pour chaque algorithme effectué. L'écriture générale est de la forme :

```
function [Output1,Output2] = Exemple1MT05(Input1,Input2)
x=Input1+Input2;
y=Input1*Input2;
Output1=max(x,y);
Output2=min(x,y);
```

Exemple 5 (Boucle For, Instructions conditionnelles) On souhaite faire la somme S des carrés des inverses des entiers impairs de 1 à 2001, puis $\sqrt{8S}$.

```
function y = Exemple2
S=0;
for i=1:2001
 if (mod(i,2)==1)
     S=S+1/i^2;
 end
end
y=(8*S)^0.5;
```

Exemple 6 (Tracé) Pour tracer l'ensemble de points (x, y) on utilise la fonction $plot(x, y)$. On utilise les options hold on/hold off pour afficher plusieurs fonctions sur un même graphique. On souhaite tracer sur une même figure les fonctions $t + sin(t)$ et $\frac{t^2}{10} + 2sin(10t)$ sur l'intervalle [0,10] :

```
> delta=0.01;
> inter=[0:delta:10];
> y= inter+sin(inter);
> z= inter.*inter/10+2*sin(10*inter);
>hold on
>plot(inter,y);
>plot(inter,z);
>hold off
```

Exemple 7 (Nuage de points dans le plan) Une autre représentation du point de coordonnées (x, y) est la valeur complexe $z = x + iy$. Si Z représente un vecteur de complexes, le nuage de points associé peut-être obtenu par $plot(Z)$ (courbe continue passant par les points) ou $plot(Z, '*')$ (nuage de points uniquement).

```
> delta=0.01;
> t=[0:delta:2*pi];
> A=exp(i*t); % cercle
> B=sin(t).*(1+i*cos(t)); % lemniscate
> C=2/3*exp(i*(t/6-2*pi/3)); % arc de cercle
> Z=[A,B,C];
> plot(Z,'*');
```

Exemple 8 (Intégration numérique) On souhaite intégrer de manière numérique la fonction $f(x) = \frac{2sin(x)}{x}$ entre 0 et 201. On crée tout d'abord la fonction f dans un fichier, puis on écrit la commande suivante : $quad(@(x)f(x), 0, 201)$ ou $quadl(@(x)f(x), 0, 201)$.

```
function y=f(x)
 if (x==0)
     y=2;
 else
     y=2.*sin(x)./x;
 end
```

3 Projet 2 : Suites numériques pour la résolution numérique d'équations

On étudie une fonction continue f (à valeurs réelles ou complexes) et on souhaite obtenir les racines de cette fcontion dans un domaine donné. Une solution analytique peut exister mais être trop coûteuse ou bien ne pas exister. Deux méthodes très répandues sont étudiées : la méthode par dichotomie et la méthode de Newton.

3.1 La méthode par dichotomie

3.1.1 Principe

Empr. au gr. διχοτομία "division en deux parties égales" (TLFi).

On suppose que la fonction f est continue sur un intervalle $[a, b]$ et à valeurs réelles. D'autre part, on fait l'hypothèse que $f(a)f(b) < 0$. D'après le théorème des valeurs intermédiaires (MATH01), f s'annule au moins une fois dans $]a, b[$. L'idée est d'évaluer f au milieu de l'intervalle $c = \frac{a+b}{2}$. Si cette valeur est nulle (ou en valeur absolue inférieure à un seuil prédéterminé), la solution est obtenue. Sinon, on étudie le signe de $f(a)f(c)$. S'il est négatif, cela signifie qu'une solution se trouve dans l'intervalle $[a, c]$ et s'il est positif, cela signifie qu'une solution se trouve dans l'intervalle $[c, b]$ (*pourquoi ?*). Cette même technique est employée sur le nouvel intervalle (d'où son nom de méthode itérative) jusqu'à ce que la solution soit obtenue. Le fait que cette méthode converge systématiquement sera étudié plus loin. Voici l'algorithme de dichotomie sur n étapes qui fait appel à plusieurs suites numériques.

Conditions initiales : $a < b$, $f(a)f(b) < 0$, $a_0 = a, b_0 = b$

Étape i ($i = 0..n$) : (à adapter si test d'arrêt)

1. Déterminer le milieu $c_i = \frac{a_i+b_i}{2}$
2. Évaluer $z_i = f(a_i)f(c_i)$.
3. Étudier le signe de z_i :
   (a) nul : on doit logiquement avoir $f(c_i) = 0$, on a identifié une racine de f.
   (b) strictement positif : $a_{i+1} = c_i$ et $b_{i+1} = b_i$.
   (c) strictement négatif : $a_{i+1} = a_i$ et $b_{i+1} = c_i$.

3.1.2 Exemple

On souhaite déterminer les racines du polynôme $f(x) = x^3 - 10x + 2$.

1. Effectuez le tableau de variations de f et vérifiez que f admet trois racines réelles distinctes $t_1 < t_2 < t_3$.
2. Grâce à la méthode de Cardan (hors programme), il est possible de déterminer les racines de f à savoir :

$$t_1 = -((1-i\sqrt{3})(-9+i\sqrt{2919})^{1/3})/(23^{2/3}) - (5(1+i\sqrt{3}))/(3(-9+i\sqrt{2919}))^{1/3}$$

$$t_2 = -((1+i\sqrt{3})(-9+i\sqrt{2919})^{1/3})/(23^{2/3}) - (5(1-i\sqrt{3}))/(3(-9+i\sqrt{2919}))^{1/3}$$

$$t_3 = (-9+i\sqrt{2919})^{1/3}/3^{2/3} + 10/(3(-9+i\sqrt{2919}))^{1/3}$$

Cette écriture n'est pas facilement exploitable et on souhaite approcher ses trois valeurs. Implémentez l'algorithme de dichotomie pour les trois racines. Préciser bien les conditions initiales. On peut commencer par effectuer $n = 20$ itérations (instruction Pour/for) ou préférer une procédure (itérative ou récursive) avec un test d'arrêt (instruction conditionnelle TantQue/while), par exemple lorsque $|f(c_n)| < 10^{-8}$ ou (moins adapté ici) lorsque la distance entre deux valeurs consécutives est inférieure à $\epsilon = 10^{-8}$.

3.1.3 Convergence

D'après la condition initiale $f(a)f(b) < 0$, on sait l'existence d'au moins une racine sur l'intervalle $]a, b[$. Supposons que f soit strictement monotone sur cet intervalle. Il existe donc une unique racine x^* sur cet intervalle. On souhaite vérifier que la suite $(c_n)_{\{n\geq 1\}}$ converge vers x^*.

1. Soit k un entier positif. Déterminer la largeur de l'intervalle $[a_k, b_k]$.
2. Vérifier que grâce à la méthode par dichotomie, x^* appartient à $[a_k, b_k]$.
3. En déduire une borne de $|c_k - x^*|$.
4. Conclure sur la convergence de la suite $(c_n)_{\{n\geq 1\}}$.
5. Plus précisément, on souhaite déterminer le nombre n minimal nécessaire afin d'être certain d'avoir une précision d'au moins ϵ de la solution, i.e. $|c_n - x^*| < \epsilon$. Déterminer n en fonction des données du problème.
6. Déterminer n afin d'avoir une précision de 10^{-10} sur la racine que vous avez choisie dans la précédente section.

3.2 La méthode de Newton

3.2.1 Principe

On considère une fonction f holomorphe ou à valeurs réelles et deux fois continûment dérivable. On souhaite obtenir une racine de f dans un voisinage d'un point x_0. Si l'on considère que f est à valeurs réelles, la méthode de Newton revient à assimiler localement la fonction à sa tangente. En première approximation, on considère que la racine de f se situe au voisinage de x_1, intersection de la tangente avec l'axe des abscisses. Cela revient à négliger les termes d'ordres supérieurs ou égaux à 2 dans le développement de Taylor au voisinage de x_0. Comme la méthode par dichotomie, la méthode de Newton est itérative, puisque l'on réitère là même technique à partir de x_1. Pour que cette intersection existe, il faut que la tangente ne soit pas parallèle à l'axe des abscisses. Dans un premier temps, on fait donc l'hypothèse supplémentaire que f' ne s'annule pas dans le domaine d'étude. Les conditions précises pour avoir la convergence de cette méthode, à savoir la convergence de la suite $(x_n)_{\{n\geq 0\}}$ vers x^* racine de f, seront discutées plus loin. L'algorithme de Newton est le suivant :

$$\begin{cases} x_0 \text{ donné} \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \end{cases}$$

Un exemple d'application de la méthode de Newton (ou Newton-Raphson) est présentée sur la Figure 1, avec $f(x) = x^2 - 2$ et $x_0 = 1$. Dans cette configuration, on obtient $x_{n+1} = \frac{x_n}{2} + \frac{1}{x_n}$, ce qui est un algorithme (méthode de Héron,

algorithme de Babylone) connu depuis plus de 3000 ans pour approcher $\sqrt{2}$. On remarque graphiquement que la convergence est rapide. On peut montrer que la convergence est généralement meilleure que la méthode par dichotomie (vitesse quadratique au lieu d'une vitesse linéaire).

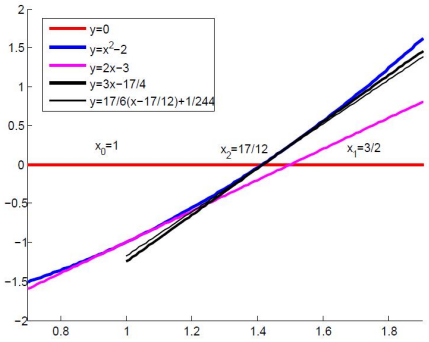

FIGURE 1 – Approche graphique de la méthode de Newton

3.2.2 Exemple

1. Comme dans la précédente section, on souhaite déterminer les racines du polynôme $f(x) = x^3 - 10x + 2$.
   (a) Déterminer l'algorithme de Newton correspondant.
   (b) En prenant un grand nombre de valeurs entre -5 et +5 (noeuds d'une subdivision), vérifiez que l'algorithme converge vers une des racines de f. pour un nombre fini de points (à déterminer). Vérifier numériquement que la convergence en démarrant aux noeuds de la subdivision forme des "bassins d'attraction", à savoir que les convergences se font par intervalles (ou ouverts dans le plan).
   (c) (Sans Matlab/Octave) Déterminer au moins deux valeurs théoriques pour lesquelles l'algorithme ne devrait pas converger. Y-a-t-il plus de deux valeurs concernées ?
2. (Sans Matlab/Octave) Déterminer l'algorithme de Newton pour la fonction $g(x) = \sqrt{|x|}$ et x_0 réel quelconque. Sans implémenter cet algoritme sous Matlab, vérifiez que la méthode de Newton ne peut pas converger. Quelle hypothèse n'est pas vérifiée ici ?
3. La méthode de Newton fonctionne généralement pour obtenir des racines de fonctions holomorphes. On considère le polynome $f(z) = z^3 - 10z + 2$. Vérifier que la convergence a lieu en prenant une maille régulière de votre choix au voisinage de l'origine.
4. (Sans Matlab/Octave) Soit $P(z) = z^7 - 2z^3 + 5$. Justifier mathématiquement que les racines de z ont un module plus petit que 2 (inégalité triangulaire). Justifier mathématiquement que P admet au moins une racine réelle.
5. En effectuant un maillage suffisamment fin du plan complexe, retrouver numériquement l'ensemble des racines de $P(z) = z^7 - 2z^3 + 5$.

La détermination des bassins d'attraction dans le plan n'est pas simple et fait intervenir les fractales de Newton. Les deux figures suivantes montrent ces bassins d'attraction pour le polynôme f. La couleur rouge (verte, jaune) désigne l'ensemble des valeurs de x_0 pour lequel l'algorithme converge vers $t_1(t_2, t_3)$.

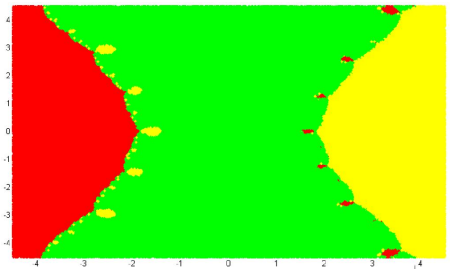

FIGURE 2 – Bassin d'attraction

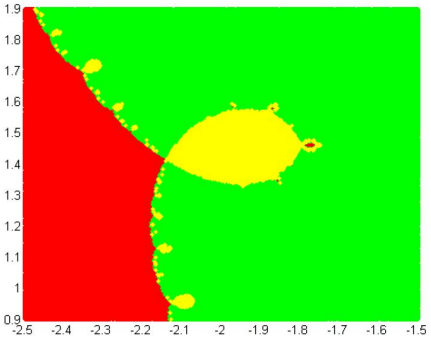

FIGURE 3 – Bassin d'attraction (zoom)

3.2.3 Méthode de la sécante (Non prioritaire)

La méthode de la sécante suit un algorithme comparable à ceux de la dichotomie et de Newton. Par rapport à la méthode de dichotomie, la nouvelle valeur n'est plus la valeur centrale, mais la valeur pour laquelle la corde associée coupe l'axe des abscisses. Par rapport à la méthode de Newton, la méthode de la sécante n'utilise pas la dérivée f' mais le remplace par le taux d'accroissement par rapport aux deux derniers points obtenus.

1. (Sans Matlab/Octave) Déterminer l'algorithme correspondant à la méthode de la sécante.
2. Implémenter la méthode de la sécante pour le polynôme $P(z) = z^3 - 10z + 2$.
3. Comparer empiriquement sa vitesse de convergence avec les deux précédentes méthodes. Cette étude revient par exemple à comparer le nombre d'étapes nécessaire avant que la précision soit atteinte pour chaque méthode et un maillage de points initiaux avec le polynôme précédent.

3.2.4 Convergence de l'algorithme de Newton

La proposition suivante donne une condition de convergence de l'algorithme de Newton : Soit f holomorphe (ou à valeurs réelles et deux fois continûment dérivable) sur un ouvert simplement connexe Ω et $x^* \in \Omega$ vérifiant $f(x^*) = 0$ et $f'(x^*) \neq 0$. Alors il existe un voisinage V de x^* (cercle fermé de rayon non nul centré en x^*) tel que la suite définie par $x_0 \in V$ et $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ converge vers x^*.

1. On pose $g = id - \frac{f}{f'}$ où id est l'opérateur identité. Exprimer x_{n+1} en fonction de x_n et g.
2. Calculer $g(x^*)$.
3. Calculer g' et en déduire que $g'(x^*) = 0$.
4. Soit $\epsilon < 1$. Une fonction holomorphe est en particulier continue. En déduire qu'il existe un cercle V centré en x^* et de rayon α non nul tel $|g'(z)| < \epsilon$ pour tout z de V.
5. Montrer que g vérifie la condition de Lipschitz de rapport ϵ dans V :

$$\forall x \in V, \forall y \in V \quad |g(x) - g(y)| < \epsilon |x - y|$$

6. Soit $z \in V$. En déduire que $|g(z) - x^*| \leq \alpha$. On a montré que l'image par g d'un élément de V reste dans V (stabilité).
7. Soit $x_0 \in V$. En déduire que x_n obtenu par l'algorithme de Newton appartient aussi à V.
8. Montrer par récurrence que $|x_n - x^*| \leq \epsilon^n |x_0 - x^*|$.
9. Conclure.