Introduction aux méthodes numériques et projet (PROJ0001): Modélisation de la trajectoire d'une balle de tennis

Premier bachelier en sciences de l'ingénieur
Année académique 2021-2022.Travail réalisé par:

Loic Delbarre (S215072) Rafik Lourmathi (S212097)
Salman Guseynov (S216862)

`r Sys.setlocale("LC_TIME", "French"); gsub(" 0", " ",format(Sys.Date(), "Le %d %B %Y"))`

Contents

1 Question 1 **3**

1.1 Méthode sécante 3

1.2 méthode de la bissection 3

1.3 comparaison des deux methodes 4

1.4 conception de `trajectoirefilethorizontal` 4

2 question4 **4**

2.1 choix de la methode 4

2.2 implementation 4

3 question 5 **4**

1 Question 1

1.1 Méthode sécante

Nous avons donc implémenté la méthode de la sécante et avons étudié les 3 cas qui peuvent se présenter à nous comme annoncé dans l'énoncé. Tout d'abord, envisageons le première possibilité : lorsque la sécante converge, elle renvoie un tuple contenant les coordonnées du zéro approximé. Ensuite, dans le deuxième cas, afin de ne pas diviser par zéro dans l'équation de la sécante(ref), on retourne un statut d'erreur. La troisème plausibilité est que la méthode de la sécante ne nous permet pas d'être assuré de la convergence de la fonction. C'est pour cela que nous implémentons un nombre d'itérations maximum comme gardien de boucle. Si jamais la fonction n'a pas convergé après le nombre maximum d'itérations (`iteration_max`), la fonction en déduit qu'il n y a pas convergence. On a défini le nombre maximal d'itérations à 100. On sait que la trajectoire d'une balle de tennis est parabolique. Donc, en utilisant la méthode de la sécante, on converge rapidement vers un zéro. Dans le cadre de ce cours, nous étudions des trajectoires paraboliques. Une vingtaine d'itérations devrait être suffisante. Mais, au vu de la faible complexité, nous avons fixé le nombre d'itérations à100.

Voici un exemple(ref) d un graphique d une fonction que l on pourrait rencontrer lors de nos calcul de trajectoire.On remarque que avec 10 iteration on atteint assez rapidement une precision satisfaisante.

1.2 méthode de la bissection

La technique de la bissection est plus efficace que la secante car elle assure la convergence lorsque les bornes sont de signes contraires et definie partout entre ses bornes.Si la fonction n est pas convergente on aurra une sortie d erreur.

"'{python diffbis,out.width="300px",echo=FALSE,fig.cap="\label{fig:diffbis}difference bissection et secante"}

import RechercheRacine RechercheRacine.show()

```
# question2
Afin de regrouper les differentes constantes utilise dans le projet. , nous les regroupo
ici la fig \ref{fig:diffbis}
On a profite de la carracteristique dtype des ‘numpy.array‘ pour choisir le nombre de de
D une autre part nous avons implemente l utilisation solve\_ivp de scipy.au vu des valeu
En optant pour la valeur de 0.001 pour
rtol et la valeur 0.000001 pour atol , nous avons remarqué que avec ses valeur par defau
$$\ddot x = \frac{\rho\cdot\pi}{m}\cdot\frac{d^{2}}{8}\cdot\|\dot x\|^{2}(\frac{1}{2+1.9
<!--
#ajouter des equation peut etre utile pour expliquer certaines demarches
#-> numeroter les figures
#-> titre legende
#-> axes
#referencer les figures pour chaques fct comme justification
-->
# question3
‘‘‘{python pyplot2,out.width="300px",echo=FALSE}
import Trajectron
Trajectron.show()
```

1.3 comparaison des deux methodes

En comparant solve_ivp et euler on se rends compte assez rapidement de la difference de temps. solve_ivp est instantane pour une precision de 10e-5 alors que euler prends déja 6secondes.Et pour une precision 10 fois plus petite, Euler augmente drastiquement vers 60 alors que Solve_ivp reste instantane.

1.4 conception de `trajectoirefilethorizontal`

On a decide de prendre un argument non obligatoire a la fonction `trajectoirefilethorizontal` pour gerer le cas ou on a un rebond de facon recursive.En effet , un rebond peut etre considere conmme le point de depart d une nouvelle trajectoire.L avantage de cette technique , c est quelle permet d etre modulaire au niveau du rebond, si l on decide de ne plus prendre en compte le rebond , on ne devra pas changer de fonction.

Les eventment choisi pour l utilisation de solve_ivp on ete choisi de maniere analogique et non discrete car `solve_ivp` est capable d utiliser ces resultat pour converger plus rapidement et plus precisement.

Pour le choix des evenement,pour la prise en compte du filet , notre choix s est porté sur `CubicSpline` et comme `bc_type` nous avons choisit le type natural pour un resultat pour

2 question4

2.1 choix de la methode

Le choix s est porte sur la methode de la secante car celle ci ne necesite pas de borne opose pour pouvoir fonctionner Les points d existence choisi pour la secante sont a determiner au cas par cas en fonction de la fonction donne. En effet les valeurs intiales determinerons l ensemble des valeurs pouvant etre obtenue par la methode.

2.2 implementation

Pour impleter les diverses fonction , au vu de la tache repetitie , nous avons decider de separer la tache en plusieurs fonction (`get cible hauteur`et `getcible rebond`) Pour faire egalement varier les differents elements nous avons implmente `multinorm``multiomega`ainsi que `rotangle` L idee est de faire varier un parametre d entre x a travers une secante ou une bissection. Pour les differentes fonction nous avons du choisir des bornes differentes. Pour la rechercheHauteur il etait specifie de prendre une hauteur de 2 a 3m Pour la rechercheOmega et recherchevitesse il a fallut poser une valeur maximale a la norme , que nous avons fixe a 100m/s et rad/sec

3 question 5

On a pris en compte que la direction de la vitesse angulaire etait donne en entree de la fonction. Nous decidons de calculer la hauteur pour chaque repartition avec une etape de 10e-2 . A chqaue etape on recalcuera la hauteur. A la fin de toutes les repartitions , on essayera de retourner la repartition qui permet de trouver la valeur h la plus elevee.Pour l etape , nous avons choisis 10e-2 .Avec 10e-2 on fait 100 foi s des tours de boucle ce qui est la valeur minimale pour avoir une valeur en pourcentage

Pour savoir la repartition de l energie on se base sur l equation suivante premettant de calculer les diferentes fluctuation. `{python ,derivehauteur,out.width="300px",echo=FALSE,fig.cap="\\labe` `import grafik grafik.show1()` Voici ici **??** un graph de la hauteur finale en fonction de la vitesse.Ce graph demontre clairement que la derivee s annule bien uniquement lorsque la balle est a l appogee de la hauteur maximum