Rapport Labo 6 individuel – Résolution d’équations

# Introduction

Pour ce labo individuel j’ai choisi de reprendre et améliorer le labo 2 précédemment réalisé avec l’équipe B2. Ce labo traitait de la résolution numérique d’une équation afin de trouver ses racines. Cette résolution se calcul par la méthode de dichotomie. Les améliorations réalisées permettent à l’utilisateur d’entrer sa propre fonction de *x*. Les racines et une représentation graphique de cette fonction sont alors affichées. Les fonctions de la donnée du labo 2 sont toujours sélectionnables par des boutons radio. Aussi, il est à présent possible de voir une animation expliquant le principe de la bissection.

# Modèle

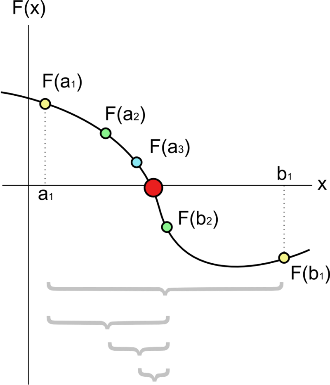
Nous avons utilisé la méthode par dichotomie. L’idée derrière cette méthode provient de l’affirmation suivante (théorème de Bolzano) : « Si, dans un intervalle donné, la fonction (continue) change de signe, alors la fonction a au moins une racine dans cet intervalle ». À partir de cette idée, il suffit de parcourir tout l’intervalle afin de déterminer pour quel(s) *x* la fonction change de signe. Ce(s) *x* sont les racines de la fonction. La détermination de(s) *x* s’obtient de prenant le centre de l’intervalle à chaque itération. Si le changement de signe provient dans le sous-intervalle de gauche, on répète la procédure sur le sous-intervalle de gauche. La procédure est symétrique dans le cas contraire. Avec un nombre d’itération qui tend vers l’infini, la valeur de la racine calculée est exacte.

Figure 1 - Illustration de la méthode de dichotomie (tirée de Wikipédia)

# Résultat

La numérisation ne permettant pas un nombre infini d’itération en un temps fini, il est nécessaire de choisir une condition d’arrêt. Le script va utiliser la bissection jusqu’à atteindre un intervalle minimal (delta = 10-8). Le centre de cet intervalle va donner une approximation de la solution. Plus la taille de l’intervalle minimal est petite, meilleure est l’approximation. Cependant, si cette taille est trop petite, l’arrondi effectué par la machine peut donner la valeur zéro. Dans ce cas, la condition d’arrêt n’étant jamais validée, le script entre dans une boucle infinie.

# Conclusion

L’implémentation de cette méthode a permis de trouver les racines des fonctions de la donnée du labo 2, mais aussi des fonctions entrées par l’utilisateur. Les différents réglages de calculs permettent d’améliorer la précision au détriment du temps de calcul, ou, à l’inverse, permettent de s’apercevoir qu’avec de mauvais réglages, l’algorithme utilisé ne trouve pas toujours les racines de la fonction. De plus, pour certaines fonctions, l’algorithme confond les asymptotes avec de vraies racines, et ce malgré les optimisations apportées afin de détecter ces asymptotes.

# Perspectives

Afin d’améliorer ce projet, il serait possible de sublimer l’animation de la bissection avec, par exemple, des visuels plus poussés, comme des graphes dynamiques plutôt que le tableau HTML. Pour la partie calculs, une idée serait d’implémenter d’autres algorithmes en plus de la dichotomie et moyenner les résultats obtenus par tous les algorithmes afin de donner un résultat plus fiable. Ces algorithmes pourraient être ceux vu au chapitre deux du cours.

# Références

|  |  |
| --- | --- |
| Algorithme | JavaScript / HTML |
| <https://fr.wikipedia.org/wiki/Méthode_de_dichotomie> | <https://www.w3schools.com/> |
| Slides du cours d’algorithmes numériques, Stéphane Gobron, 2017 | <https://jquery.com/> |
| <https://stackoverflow.com/> |