

<http://irem.univ-reunion.fr/spip.php?article531>



Résolution approchée d'équations

- Algorithmique et programmation
- Programmation Python en Seconde



Date de mise en ligne : vendredi 18 novembre 2011

Copyright © IREM de la Réunion - Tous droits réservés

Les langages objets permettent la surcharge de méthodes, qui rend presque automatique le changement de cadre numérique : Le même algorithme peut presque sans changement s'appliquer à des réels, des fractions ou des décimaux ayant un grand nombre de chiffres.

L'idée est que l'algorithme de Heron permet avec très peu de modifications

- De calculer une dizaine de décimales d'une racine carrée ;
- D'afficher les fractions obtenues plutôt que les valeurs approchées
- De calculer quelques milliers de décimales en moins d'une seconde

Le secret est que les réels, les fractions, et les décimaux sont des objets qu'on peut choisir à la carte.

Pour calculer algorithmiquement une racine carrée en Seconde, le programme recommande la méthode par dichotomie. Mais celle-ci est complexe à mettre en oeuvre parce qu'elle fait intervenir un test dans une boucle à condition de sortie (comme l'algorithme d'Euclide d'ailleurs). La méthode par balayage est plus simple, puisqu'il n'y a pas de test dans la boucle. Cependant en Python, au moment de la sortie de la boucle, on a effectué un pas de trop, et il faut revenir en arrière de ce pas, avant de passer à la décimale suivante [1]

Le travail de préparation en amont du TP étant long, et les élèves ayant encore trop peu l'habitude de rédiger leurs démonstrations, le moment était idéal pour un DM de narration de recherche, dont le sujet était

Calculer $\sqrt{3}$ à 6 décimales, en n'utilisant que les 4 opérations et l'élevation à une puissance.

Ceci pour éviter de bloquer les élèves qui ne savent pas qu'un carré est un produit ; ceci dit c'est un peu dangereux puisque si un élève fait

```
print(3**(0.5))
```

il a respecté la consigne...

Au final,

- Deux élèves (mais qui ne semblent pas avoir travaillé de façon indépendante) ont utilisé l'algorithme de balayage
- Sept élèves (soit un élève sur cinq) ont accéléré le balayage par une ébauche de dichotomie
- Deux élèves (dont l'une a copié sur l'autre) ont appliqué le même mélange de dichotomie et balayage, mais sur la racine de la racine de trois

Voici quelques extraits de leur travail :

[\[PDF - 1.6 Mo\]](http://irem.univ-reunion.fr/IMG/pdf/NR1.pdf "PDF - 1.6 Mo")

narration de recherche compte-rendu en pdf

Ainsi, la dichotomie a été abandonnée et le balayage, décrit avec tableur lors du corrigé de la narration de recherche, est devenu le hors-d'oeuvre du TP, essentiellement consacré à Heron, et dont voici le sujet :

[\[PDF - 152.8 ko\]](http://irem.univ-reunion.fr/IMG/pdf/TP8.pdf "PDF - 152.8 ko")

le sujet du TP en pdf

Le passage de ce TP a été un peu particulier pour un des deux groupes d'élèves, puisqu'ils ont passé le TP lors du [salon de l'éducation](#), dans une salle numérique qu'ils ne connaissaient pas au préalable. L'opération a été une bonne publicité pour Python, au vu des réactions des spectateurs.

Problèmes d'affichage

Sur l'un des ordinateurs Â« POP Â» (offerts par le Conseil Régional aux élèves de Seconde), l'affichage produit par le premier programme ressemblait à ceci :

1.0000000000000000
1.7000000000000006
1.7300000000000006
1.7320000000000004
1.7320000000000004
1.7320500000000008
1.7320500000000008
1.7320508000000012
1.7320508000000012
1.7320508070000018

Alors que sur les autres, l'affichage est correctement arrondi :

1
1.7
1.73
1.732
1.732
1.73205
1.73205
1.7320508
1.7320508

1.732050807

Ce phénomène est sans doute dû à l'architecture interne de la machine [2].

Chose étrange : L'élève croyait que son ordinateur était plus précis que les autres, puisque plus de chiffres sont affichés [3].

Beaucoup d'élèves ont le réflexe d'ajouter des espaces avant les parenthèses, comme dans *range (10)* au lieu de *range(10)* ; ils ne semblent pas conscients du fait que *range* et *print* sont des fonctions et n'ont donc pas tendance à les noter ainsi. Mais Python est (pour une fois) assez permissif sur ce point.

Deux élèves ont trouvé que les 10 décimales sont atteintes au bout de $\lceil \sqrt{3} \rceil$ itérations !

La plupart des élèves interprètent le tableau de variation comme si la fonction était croissante (ou alors, ils connaissent mal le sens de l'expression « par défaut »). De façon générale, la notion de variations n'est visiblement pas encore acquise à ce stade de l'année. Il faut dire que le cours sur les fonctions « carré » et « inverse » n'a pas encore été fait. Et l'idée que deux nombres positifs sont dans le même ordre que leurs carrés n'est visiblement pas naturelle du tout.

La moyenne obtenue à ce TP a été 12,2 ; les questions auxquelles les élèves ayant le plus rarement répondu étant celles sur les valeurs approchées par défaut et par excès. Ceci dit, le calcul de la moyenne de deux fractions (même lorsqu'elles sont décimales) n'est pas toujours maîtrisé par les élèves.

[1] Pour les nostalgiques du langage Pascal, il faut une boucle *repeat ... until* au lieu d'une *while ... do*. Python et Algobox ne possèdent pas une telle boucle, mais la calculatrice graphique, si (ainsi que JavaScript d'ailleurs...)

[2] À l'attention de « AAA » : Une fois de plus, je n'ai pas eu le temps de photographier l'écran de l'ordinateur en question, on peut à nouveau me soupçonner d'avoir imaginé de toutes mains cette erreur ; sauf qu'elle n'est pas due à l'élève...

[3] Pour beaucoup d'élèves de Seconde, tous les nombres sont décimaux, et les chiffres invisibles sont inexistantes.