

<http://irem.univ-reunion.fr/spip.php?article439>



Convergence d'une suite

- Algorithmique et programmation
 - TP d'algorithmique en Première
-

Date de mise en ligne : dimanche 5 septembre 2010

Copyright © IREM de la Réunion - Tous droits réservés

Un robot est chargé de quitter un cercle à partir de son centre en avançant de moins en moins vite. Y arrivera-t-il ?

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH182/robot2-610a4.jpg>]

Le robot peut très bien être une tortue Python ! Dans ce cas il vaut mieux qu'au lieu d'avancer de $\frac{1}{n(n+1)}$ à chaque étape, elle avance de $\frac{100}{n(n+1)}$ et qu'elle essaye de sortir du cercle de rayon 100.

On peut commencer en interactif :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L351xH400/robotpy1-55949.png>]

puis construire un segment à 100 pixels de l'origine (pour voir où la tortue doit aller) puis écrire une boucle :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L198xH400/bernoupy2-52bf0.png>]

Cette méthode permet de cacher les additions dans le caractère cumulatif de la mémoire de la tortue. On obtient ceci après 20 itérations :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L193xH115/bernoupy3-20dc8.png>]

Pour faire un peu plus « algorithmique » avec une variable incrémentée (de moins en moins) on peut utiliser un CaRScript comme celui ci-dessous :

```
<a href="http://irem.univ-reunion.fr/IMG/zirs/bernou1.zirs" title='CarMetal - 6.6 ko' type="application/x-carmetal">
[CarMetal - 6.6 ko]
```

Le script peut ensuite être modifié pour représenter graphiquement la suite (onglet « limite » ci-dessus) et avec une droite horizontale mobile pour conjecturer la valeur de la limite.

On peut aussi (risqué !) transformer l'algorithme en une boucle telle que

```
s=0;
n=1;
while(s<1){
  s+=1/n/(n+1);
  n++;
  Move("R",s,0);
}<div class='code_download' style='text-align: right;'> <a
href='http://irem.univ-reunion.fr/local/cache-code/35434b682674a0d935c81a33fa4c9403.txt' style='font-family:
verdana, arial, sans; font-weight: bold; font-style: normal;'>Télécharger
```

et la question devient : « Est-ce que la boucle s'arrêtera un jour, ou est-elle sans fin ? »

SciLab

Avec SciLab, les suites (finies) sont des objets que le logiciel peut manipuler, sans boucles ni indices (sauf si on veut, la syntaxe des boucles est plutôt simple). Ci-dessous on crée la suite des n par $a=1:20$, puis celle des $n+1$ par $b=a+1$, puis leur produit terme à terme dans c et la suite des $\frac{1}{n(n+1)}$ dans d , à chaque fois suivis par des points-virgules pour ne pas encombrer l'écran par leur affichage. Ensuite *cumsum* donne la suite u_n :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH167/robotsci1-e8c15.png>]

On a le même genre de résultat qu'un tableur, avec le même genre de manip (sans boucle ni affectation de variables numériques). Pour le graphique en rouge (Â« r Â»), en pointillés (Â« : Â») et avec des points ronds (Â« o Â»), on fait

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH260/robotsci2-9513c.png>]

ce qui donne ceci :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH302/robotsci3-0a39e.png>]

Calcul formel

Le même genre de manip sous Xcas donne la valeur exacte de u_{20} qui aide à faire une conjecture, renforcée par le calcul de $u_{100\ 000}$, puis de u_N :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH167/robotxcas-e4845.png>]

Quant à *WxMaxima*, il est écrit en [LISP](#) qui est un langage spécialisé dans les listes, alors on peut avoir les 20 premiers termes de $\frac{1}{n(n+1)}$ et de u_n d'un coup :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH99/robotmax1-2-42842.png>]

Et le paquet *solve_rec* permet de calculer la valeur exacte de u_n en considérant celle-ci comme récurrente (en effet $u_{n+1}=u_n+\frac{1}{(n+1)(n+2)}$) :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH154/robotmaxima2-ed61e.png>]

Bref, il y a bien des manières d'étudier expérimentalement cette suite :

1. le calcul formel donne des résultats immédiatement, de façon assez interactive surtout avec le tableur d'Xcas, non traité ici ;
2. le tableur justement, non traité ici, somme assez facilement les termes d'une suite mais avec un Â« dollar Â» au début, par exemple `B1=SOMME(A$1:A1)` à copier vers le bas si la colonne A contient les fractions à additionner ;
3. SciLab fait des affectations sur des variables qui sont des suites (et non des nombres), ce qui évite l'usage de boucles ;
4. L'algorithmique, avec des sommes calculées par boucles, donc un esprit un peu différent des autres méthodes.

Convergence d'une suite

Pour chacune des démarches, le présent exercice peut être considéré comme une prise en mains de l'outil, en vue d'une utilisation ultérieure de celui-ci dans un contexte où il est plus approprié, voire de plusieurs outils, pour préparer l'élève au choix de l'outil adéquat pour un TP donné.

Sujet

Le sujet du TP est téléchargeable ci-dessous, sous licence CC :

[Creative Commons License](#)

This work is in the [Public Domain](#).

[\[PDF - 116.3 ko\]](http://irem.univ-reunion.fr/IMG/pdf/robot.pdf "PDF - 116.3 ko")

le sujet du TP en pdf