TP3 – Sage

1 Programmation

Comme bien d'autres langages informatiques, Sage permet à l'utilisateur de définir des sous-programmes sur mesure, avec un ou plusieurs arguments : les procédures, qui ne renvoient pas de résultat, et des fonctions, qui en renvoient un.

Une même fonction peut souvent être écrite de façon *itérative* (i.e. en utilisant une boucle et des tests) ou *récursive* (i.e. sans utilisation de boucle, la fonction faisant appel à elle-même pour le calcul de ses valeurs, analogue informatique de la récurrence mathématique).

Exercice 1 - Premiers programmes

1. Sans utiliser la fonction `max`, écrire une fonction `maximum` calculant le maximum entre deux réels a et$b$.
   ↪ *Utiliser une boucle* `if`.
2. Écrire une fonction `prod_first_odds` qui calcule le produit des n premiers entiers impairs.
   ↪ *Utiliser une boucle* `for`.
3. Écrire une fonction `division` qui effectue la division euclidienne d'un entier a par un entier b, et renvoie le quotient q et le reste r de cette division.
   ↪ *Utiliser une boucle* `while`.

Exercice 2 - Factorielle

Sans utiliser la fonction prédéfinie de Sage, écrire deux fonctions calculant $n!$:
- une fonction itérative `factorielle_it`
- une fonction récursive `factorielle_rec`

Exercice 3 - Exponentiation rapide

1. Sans utiliser la fonction `^`, écrire une fonction récursive `puiss(a,n)` calculant a^n.
2. En utilisant le fait que, selon la parité de n, on a $a^{2n} = \left(a^2\right)^n$ ou $a^{2n+1} = a \cdot \left(a^2\right)^n$, écrire une fonction récursive `puiss_rapide(a,n)`.
   ↪ *Un entier n modulo p s'écrit* `n%p`.
3. Utiliser une variable globale `cpt` dans les deux fonctions pour compter le nombre de multiplications effectuées et comparer les deux méthodes.
   ↪ *Définir* `cpt` *comme une variable globale.*

Exercice 4 - Dichotomie

La méthode de dichotomie est basée sur le théorème des valeurs intermédiaires : une fonction $f : [a,b] \to \mathbb{R}$ continue telle que $f(a)$ et $f(b)$ sont de signes contraires admet au moins un zéro dans l'intervalle $]a,b[$.

Programmer la méthode de dichotomie donnant une valeur approchée d'un zéro de la fonction f à la précision ε de deux manières différentes.

- **Fonction récursive** `zero_rec(f,a,b,eps)`

  $m \leftarrow \dfrac{a+b}{2}$
  Si $b - a < \varepsilon$ terminer, et retourner m
  Si $f(a)f(m) \leq 0$ alors $zero\_rec(f,a,m,\varepsilon)$
  Sinon $zero\_rec(f,m,b,\varepsilon)$

- **Fonction itérative** `zero_it(f,a,b,eps)`

  $inf \leftarrow a$, $sup \leftarrow b$, $m \leftarrow \dfrac{a+b}{2}$
  Tant que $sup - inf \geq \varepsilon$, faire :
    Si $f(inf)f(m) \leq 0$ alors $sup = m$
    Sinon $inf = m$
    $m \leftarrow \dfrac{inf + sup}{2}$
  Retourner m

Calculer l'approximation de $\sqrt{3}$ à 10^{-5} avec les deux méthodes.

Exercice 5 - Permutation circulaire

1. Écrire une fonction `permute_1step_right` prenant une liste en entrée, et qui effectue une permutation circulaire des éléments d'une liste à n éléments de un pas vers la droite.
2. Même exercice en effectuant une permutation circulaire de deux pas vers la gauche.

Exercice 6 - C'est à boire, à boire qu'il nous faut...

Calculer le nombre de façons de vider un fût de 7 litres de bière à l'aide d'un demi (quart de litre), d'un sérieux (demi-litre) et d'un formidable (un litre), en tenant compte de l'ordre des opérations. Attention, les contenants doivent être remplis entièrement à chaque étape : un contenant d'un demi litre ne peut être utilisé pour vider un quart de litre !

↪ *Utiliser une fonction récursive.*

---

2 Exercices d'application

Exercice 1 - Simplifications

Vérifier les formules suivantes :

1. $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$
2. $\dfrac{a-b}{\sqrt{a}-\sqrt{b}} = \sqrt{a}+\sqrt{b}$

Exercice 2 - Suites adjacentes

Soit $0 < b < a$. On considère les suites imbriquées définies par

$$\begin{cases} u_0 = a, & v_0 = b \\ u_{n+1} = \dfrac{u_n+v_n}{2} & v_{n+1} = 2\dfrac{u_n v_n}{u_n+v_n} \end{cases}$$

1. Montrer que si (u_n) et (v_n) convergent, elles ont la même limite.
2. Vérifier que
$$u_{n+1} - v_{n+1} = \frac{u_n - v_n}{2} - \frac{(u_n-v_n)v_n}{u_n+v_n}$$
3. En déduire que $0 < u_{n+1} - v_{n+1} < \dfrac{1}{2}(u_n - v_n)$.
4. Montrer que les deux suites sont adjacentes.
5. Pour $a = 2$ et $b = 1$, quelle semble être la valeur de la limite de (u_n) et (v_n) ?
6. Tracer les deux suites pour $n \leq 10$ à l'aide de la commande `point`.

Exercice 3 - Limite de suites

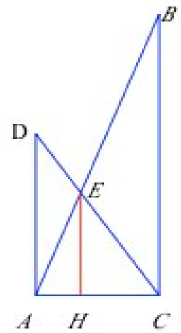
Soient $(u_n)_{n\in\mathbb{N}}$ et $(v_n)_{n\in\mathbb{N}}$ deux suites de nombres réels définies par la donnée de leurs premiers termes (u_0, v_0) et les formules de récurrence suivantes

$$\begin{cases} u_{n+1} = \dfrac{9}{10}u_n + \dfrac{2}{5}v_n \\ v_{n+1} = \dfrac{1}{10}u_n + \dfrac{3}{5}v_n \end{cases}$$

1. Calculer u_n et v_n en fonction de u_0, v_0 et n
2. En déduire $\lim\limits_{n\to+\infty} u_n$ et $\lim\limits_{n\to+\infty} u_n$.

Exercice 4 - Un problème d'échelles

Deux échelles, l'une longue de six mètres, l'autre longue de quatre mètres, sont appuyées sur deux murs verticaux opposés, conformément à la figure ci-contre.



Le point de croisement des deux échelles étant à deux mètres du sol, calculer la distance entre les deux murs et tracer la configuration obtenue à l'aide de la commande `line`.

Exercice 5 - Équations non linéaires

En utilisant la commande `solve` :

1. Résoudre $x^2 - 2 = 0$
2. Résoudre $\begin{cases} x^2 + 3y = -1 \\ -3x + y = 1 \end{cases}$
3. On considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par la relation de récurrence $u_{n-1} = 2\dfrac{u_n(u_n+1)}{u_n^2+4}$. Trouver l'expression de u_n en fonction de u_{n-1}, puis calculer les deux valeurs de u_n possibles pour $u_{n-1} = -1$ en utilisant la commande `.rhs()`.

Exercice 6 - Un problème de milliardaires

Trois milliardaires X, Y et Z décident de se redistribuer leurs fortunes de la manière suivante : X versera la moitié de sa fortune à Y, puis Y versera la moitié de sa fortune à Z, puis Z versera la moitié de sa fortune à X. Ensuite, ils renouvelleront ce cycle de trois opérations jusqu'à ce que leurs fortunes se stabilisent. Lequel des trois milliardaires a proposé cette méthode de partage saugrenue ?
$\hookrightarrow$ *On pourra noter x_n, y_n et z_n les fortunes respectives de X, Y et Z après n cycles de transactions, calculer x_n, y_n et z_n en fonction de x_0, y_0, z_0 et n, puis passer à la limite lorsque n tend vers $+\infty$.*
