55 Algo python

Algorithme de balayage

a) L'équation $x^3 - 3x + 1 = 0$ admet une unique solution α dans l'intervalle $[-2\,;-1]$ (voir exercice **54**).

Ce programme incomplet se propose d'obtenir par balayage un encadrement d'amplitude 10^{-n} (avec $n \in \mathbb{N}^*$) de α.

```
def f(x):
    y=[        ]
    return y

def Balayage(n):
    s=-2
    while f(s)<=[ ]:
        s=s+10**(-n)
    s=round(s,n)
    t=round([        ],n)
    return t,s
```

Compléter ce programme et le saisir. L'exécuter avec $n = 6$ et interpréter les résultats obtenus.

b) g est la fonction exponentielle.

Adapter le programme précédent pour obtenir un encadrement d'amplitude 10^{-4} de l'unique solution de l'équation $g(x) = 10$ dans l'intervalle $[2\,;3]$.

c) h est la fonction définie sur $\mathbb{R}$ par :

$$h(x) = -x^3 + x + 2.$$

Afficher la courbe de h à l'écran de la calculatrice, puis adapter le programme du **a)** pour obtenir un encadrement d'amplitude 10^{-5} de l'unique solution de l'équation $h(x) = 0$ dans l'intervalle $[1\,;2]$.

RESOLUTION D'EQUATION.

Savoir-faire Algo python

Vidéo JAICOMPRIS.COM — Un autre exercice Algo/python résolu en vidéo

EXERCICE RÉSOLU

16 Utiliser un algorithme de dichotomie

Cours 4. B

L'équation $x^3 - 3x + 1 = 0$ admet une unique solution x_0 dans l'intervalle $[-1\,;1]$ (voir exercice **14**).

L'algorithme ci-contre permet d'obtenir une valeur approchée de x_0 à 10^{-n} près où n est un nombre entier naturel.

a) Exécuter cet algorithme pas à pas en complétant un tableau de valeurs lorsqu'on affecte au début la valeur -1 à la variable a, la valeur 1 à la variable b et la valeur 1 à la variable n. Interpréter la valeur de la variable m à la fin de l'exécution de l'algorithme.

b) Coder cet algorithme en langage Python.

Saisir et exécuter le programme obtenu avec $n = 4$ et interpréter le résultat renvoyé.

```
Tant que b − a > 10^(−n)
    m = (a + b)/2
    Si f(a) × f(m) < 0 alors
        b = m
    sinon
        a = m
    Fin Si
Fin Tant que
```

Solution

a)

a	-1	0	0	0,25	0,25	0,312 5
b	1	1	0,5	0,5	0,375	0,375
$b - a > 10^{-1}$	Vrai	Vrai	Vrai	Vrai	Vrai	Faux
m	0	0,5	0,25	0,375	0,312 5	
$f(a) \times f(m) < 0$	Faux	Vrai	Faux	Vrai	Faux	

Avec la calculatrice, on détermine les signes de $f(a)$ et $f(m)$.

Lorsqu'à la dernière étape, $b - a \leqslant 10^{-1}$ (ici, $0{,}375 - 0{,}312\,5 = 0{,}062\,5$) l'algorithme renvoie la valeur de m obtenue à l'étape précédente. Donc, ici l'algorithme renvoie $m = 0{,}312\,5$.

Cela signifie qu'une valeur approchée de x_0 à 10^{-1} près est 0,312 5.

b)

```
def f(x):
    y=x**3-3*x+1
    return y

def Dichotomie(n):
    a=-1
    b=1
    while b-a>10**(-n):
        m=(a+b)/2
        if f(a)*f(m)<0:
            b=m
        else:
            a=m
    return m
```

Le résultat, renvoyé ci-contre par le programme, fournit une valeur approchée au dix-millième près de x_0.

```
>>> Dichotomie(4)
0.34735107421875
```

Le mot « dichotomie » provient du grec *dikhotomia* qui signifie « division en deux parties ».

L'algorithme de dichotomie consiste à répéter le partage en deux d'un intervalle à l'aide de son centre puis à sélectionner celui des deux « demi-intervalles » dans lequel est localisée la solution x_0.

EXERCICES D'APPLICATION DIRECTE

Sur le modèle de l'exercice résolu 16

17 Adapter, puis exécuter le programme Python de l'exercice **16** pour déterminer une valeur approchée à 10^{-6} près de l'unique solution de l'équation $e^x - x - 2 = 0$ dans l'intervalle $[0\,;2]$.

18 **(E)** est l'équation $3x^3 - x^2 - 3 = 0$.

a) Démontrer que cette équation **(E)** a une unique solution x_0 dans $\mathbb{R}$, puis que $1 < x_0 < 2$.

b) Utiliser un programme de dichotomie pour donner une valeur approchée à 10^{-3} près de x_0.

Exercice 9 : Recherche des points d'intersection entre une droite $\mathscr{D}$ et une parabole $\mathscr{P}$

Cet exercice revient à chercher les points d'intersection éventuels de la droite et de la parabole en résolvant une équation.

Partie A : Premier cas

$\mathscr{D}$ a pour équation $y = -x + 4$ et $\mathscr{P}$ a pour équation $y = \frac{1}{2}x^2 - x - 4$.

1) Construire $\mathscr{D}$ et $\mathscr{P}$ sur la calculatrice et conjecturer les intersections de $\mathscr{D}$ et $\mathscr{P}$.

2) Résoudre $-x + 4 = \frac{1}{2}x^2 - x - 4$, et conclure.

Partie B : Deuxième cas

$\mathscr{D}$ a pour équation $y = x - 4$ et $\mathscr{P}$ a pour équation $y = x^2 - x + 1$.

1) Résoudre $x - 4 = x^2 - x + 1$, et conclure sur les intersections de $\mathscr{D}$ et $\mathscr{P}$.

2) Construire $\mathscr{D}$ et $\mathscr{P}$ sur la calculatrice et vérifier la conclusion de la question précédente .

Partie C : Troisième cas

$\mathscr{D}$ a pour équation $y = x - 2$ et $\mathscr{P}$ a pour équation $y = x^2 - 4x + 2$.

1) Construire $\mathscr{D}$ et $\mathscr{P}$ sur la calculatrice. On constate que $\mathscr{D}$ et $\mathscr{P}$ se coupent en deux points distincts A et B.

2) Résoudre $x - 2 = x^2 - 4x + 2$, puis calculer les coordonnées des points A et B.

Partie D : Généralisation

Si $\mathscr{D}$ a pour équation $y = mx + p$ et $\mathscr{P}$ a pour équation $y = ax^2 + bx + c$ donner une condition nécessaire et suffisante pour que :

1) $\mathscr{D}$ coupe $\mathscr{P}$.

2) $\mathscr{D}$ soit tangente à $\mathscr{P}$.

3) $\mathscr{D}$ ne coupe pas $\mathscr{P}$.

Pour les exercices 81 et 82, f est une fonction continue et strictement monotone sur un intervalle $[a\,;b]$ (avec a et b nombres réels) telle que $f(a) \times f(b) < 0$. Donc, d'après la conséquence énoncée p. 298, l'équation $f(x) = 0$ admet une unique solution α dans l'intervalle $[a\,;b]$.

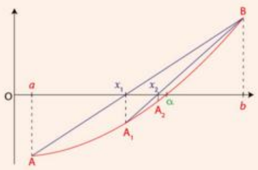
81 MÉTHODE DE LA SÉCANTE

Algo python

Objectif
Présenter un procédé algorithmique, la méthode de la sécante, qui permet d'obtenir des approximations de plus en plus proches de α.

Principe de la méthode

On suppose ici que **$f(a) < 0$, $f(b) > 0$ et f strictement croissante et convexe sur l'intervalle $[a\,;b]$.**
Dans le repère orthonormé ci-contre, la courbe rouge représente la fonction f et $A(a\,;f(a))$, $B(b\,;f(b))$.
1re étape : on trace le segment $[AB]$, il coupe l'axe des abscisses en un point d'abscisse x_1.
2e étape : On trace le segment $[A_1B]$ où $A_1(x_1\,;f(x_1))$, il coupe l'axe des abscisses en un point d'abscisse x_2.
⋮



On note x_n l'abscisse du point A_n. On définit ainsi une suite (x_n) de réels de $[a\,;b]$ et on admet que pour tout entier naturel n, $f(x_n) \leqslant 0$.

1. Relation entre x_{n+1} et x_n

a) Justifier qu'une équation de la droite (A_nB) est $y = (x - x_n)\dfrac{f(b) - f(x_n)}{b - x_n} + f(x_n)$.

b) En déterminant le point d'intersection de (A_nB) avec l'axe des abscisses, justifier que

$$x_{n+1} = x_n - \frac{b - x_n}{f(b) - f(x_n)}f(x_n).$$

2. Étude de la suite (x_n)

(x_n) est la suite définie par $x_0 = a$ et pour tout entier naturel n, $x_{n+1} = x_n - \dfrac{b - x_n}{f(b) - f(x_n)}f(x_n)$.

a) Démontrer que la suite (x_n) est croissante et majorée.

b) En déduire que la suite (x_n) est convergente et déterminer algébriquement sa limite.

3. Un programme en langage Python

Voici un programme incomplet écrit en langage Python qui met en œuvre la suite (x_n) définie ci-dessus.
On choisit $|f(x_n)| \leqslant 10^{-p}$ pour critère d'arrêt où p désigne un entier naturel non nul donné.

```
def f(x):
    y=
    return y

def Secante(a,b,p):
    X=
    n=0
    while abs(f(X))>10**(-p):
        X=
        n=n+1
    return n,X
```

a) Indiquer les expressions cachées par les deux cadres verts.

b) f est la fonction définie sur $\mathbb{R}$ par $f(x) = x^3 + x - 1$.
Vérifier que f est strictement croissante et convexe sur $[0\,;1]$ et que l'équation $f(x) = 0$ admet une unique solution α dans $[0\,;1]$.

c) Saisir le programme pour cette fonction f et l'exécuter avec $p = 6$.

4. Complément

Adapter la suite (x_n) dans le cas où : $f(a) > 0$, $f(b) < 0$, f strictement décroissante et convexe sur $[a\,;b]$.

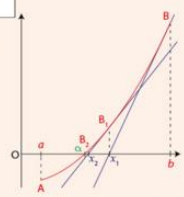
82 MÉTHODE DE NEWTON

Algo python

Objectif
Présenter un procédé algorithmique, la méthode de Newton, qui permet d'obtenir des approximations de plus en plus proches de α.

Principe de la méthode

On suppose ici que **$f(a) < 0$, $f(b) > 0$, f dérivable, strictement croissante et convexe sur l'intervalle $[a\,;b]$ ainsi que $f'(\alpha) \neq 0$.**
Dans le repère orthonormé ci-contre, la courbe rouge représente la fonction f et $A(a\,;f(a))$, $B(b\,;f(b))$.
1re étape : on trace la tangente en B à la courbe, elle coupe l'axe des abscisses en un point d'abscisse x_1.
2e étape : on trace la tangente en $B_1(x_1\,;f(x_1))$ à la courbe, elle coupe l'axe des abscisses en un point d'abscisse x_2.
⋮



On note x_n l'abscisse du point B_n. On définit ainsi une suite (x_n) de réels de $[a\,;b]$ et on admet que pour tout entier naturel n, $f(x_n) \leqslant 0$.

1. Relation entre x_{n+1} et x_n

On note x_n l'abscisse du point B_n.

a) Justifier qu'une équation de la tangente à la courbe en B_n est $y = f'(x_n)(x - x_n) + f(x_n)$.

b) En déterminant le point d'intersection de la tangente en B_n avec l'axe des abscisses, justifier que $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$.

2. Étude de la suite (x_n)

(x_n) est la suite définie par $x_0 = b$ et pour tout entier naturel n, $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$.

a) Démontrer que la suite (x_n) est décroissante et minorée.

b) En déduire que la suite (x_n) est convergente et déterminer algébriquement sa limite.

3. Un programme en langage Python

Voici un programme incomplet écrit en langage Python qui met en œuvre la suite (x_n) définie ci-dessus. On choisit $|f(x_n)| \leqslant 10^{-p}$ pour critère d'arrêt où p désigne un entier naturel non nul donné.

```
def f(x):
    y=
    return y

def df(x):
    y=
    return y

def Newton(a,b,p):
    X=b
    n=0
    while abs(f(X))>10**(-p):
        X=
        n=n+1
    return n,X
```

a) Indiquer l'expression cachée par le cadre vert.

b) f est la fonction définie sur $\mathbb{R}$ par $f(x) = x^3 + x - 1$.
L'équation $f(x) = 0$ a une unique solution α dans l'intervalle $[0\,;1]$ (voir 81 question 3. b)).
Saisir le programme pour cette fonction f et l'exécuter avec $p = 6$. Comparer éventuellement le nombre d'étapes nécessaires avec les méthodes de Newton et de la sécante.
La méthode de Newton converge, en général, plus rapidement que la méthode la sécante.

HISTOIRE DES MATHS

Cette méthode de Newton est présentée en 1736, après sa mort, dans son ouvrage *Method of Fluxions*.

Bien que Joseph Raphson ait déjà utilisé cette méthode en 1690, il semble que Newton ait écrit cet ouvrage en 1671.

Tous deux n'appliquent cette méthode qu'à des polynômes, c'est Thomas Simpson qui la généralise en 1740.

CALCUL D'INTEGRALE.

Savoir-faire **Algo** python

Vidéo JAICOMPRIS.COM Un autre exercice Algo/python résolu en vidéo

EXERCICE RÉSOLU

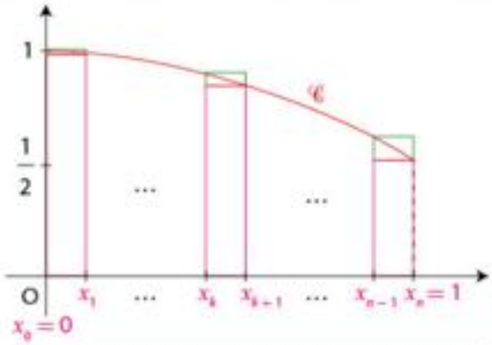
18 Appliquer la méthode des rectangles

f est la fonction définie sur $[0\,;1]$ par $f(x)=\dfrac{1}{1+x^2}$.

$\mathscr{C}$ est la courbe représentative de f dans un repère orthogonal.

On découpe l'intervalle $[0\,;1]$ en n intervalles ($n \in \mathbb{N}^*$) à l'aide des nombres $x_0=0, \ldots, x_k=\dfrac{k}{n}, \ldots, x_n=1$ (avec $0 \leqslant k \leqslant n$).

Sur chaque intervalle $[x_k\,;x_{k+1}]$, on construit les rectangles indiqués sur la figure.



1. a) Interpréter ces sommes S et S′, en termes d'aires.

$$S=\frac{1}{n}(f(x_1)+f(x_2)+\ldots+f(x_n))$$

$$S'=\frac{1}{n}(f(x_0)+f(x_1)+\ldots+f(x_{n-1}))$$

b) Justifier l'encadrement $S \leqslant \int_0^1 f(x)dx \leqslant S'$.

2. Voici ci-contre une fonction **Rectangles** écrite en langage Python.

a) Quelles sont les valeurs des variables Som1 et Som2 renvoyées par cette fonction ?

b) Donner à l'aide de cette fonction un encadrement d'amplitude 0,01 de $\int_0^1 f(x)\mathrm{d}x$.

```python
from math import *

def f(x):
    y=1/(1+x**2)
    return y

def Rectangles(n):
    Som=0
    for k in range(1,n):
        Som=Som+f(k/n)
    Som1=1/n*(Som+f(1))
    Som2=1/n*(Som+f(0))
    return Som1,Som2
```

Solution

1. a) S est la somme des aires des rectangles représentés en rose sur la figure, S′ est celle des aires des rectangles représentés en vert sur la figure.

b) Les rectangles roses se trouvent au-dessous de $\mathscr{C}$ et les rectangles verts au-dessus donc l'aire, en *u.a.*, du domaine sous la courbe $\mathscr{C}$ est comprise entre S et S′, soit $S \leqslant \int_0^1 f(x)\mathrm{d}x \leqslant S'$.

$S=\dfrac{1}{n}\sum_{i=1}^{n} f(x_i)$ et S′ sont des valeurs approchées de $\int_0^1 f(x)\mathrm{d}x$

2. a) À la sortie de la boucle, on obtient $\text{Som}=f(x_1)+f(x_2)+\ldots+f(x_{n-1})$, donc la fonction **Rectangles** renvoie Som1 = S et Som2 = S′.

b) D'après l'affichage ci-contre : $0{,}78 \leqslant \int_0^1 f(x)\mathrm{d}x \leqslant 0{,}79$.

```
>>> Rectangles(100)
(0.7828939967307822, 0.7878939967307822)
```

EXERCICES D'APPLICATION DIRECTE

Sur le modèle de l'exercice résolu 18

19 g est la fonction définie sur $[1\,;2]$ par :

$$g(x)=\frac{1}{1+x^2}.$$

a) Adapter la fonction **Rectangles** de l'exercice **18** à cette nouvelle fonction g.

b) Donner un encadrement d'amplitude 0,01 de $\int_1^2 g(x)\mathrm{d}x$.

20 h est la fonction définie sur $[-1\,;0]$ par :

$$h(x)=\mathrm{e}^{-x^2}.$$

Remarquer que la fonction h est croissante sur l'intervalle $[-1\,;0]$.

a) Adapter la fonction **Rectangles** de l'exercice **18** à cette nouvelle fonction h.

b) Donner un encadrement d'amplitude 0,001 de $\int_{-1}^0 h(x)\mathrm{d}x$.

102 UN ALGORITHME DE BROUNCKER

Algo python

f est la fonction définie sur l'intervalle $[1\,;2]$ par $f(x)=\dfrac{1}{x}$.

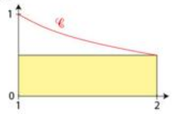
$\mathscr{C}$ est la courbe représentative de la fonction f dans un repère orthogonal.

On se propose d'estimer $\ln(2)$ en utilisant l'aire sous une hyperbole.

1. Premières étapes

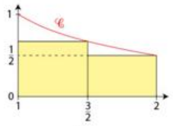
À chacune des étapes i ($i \in \{1, 2, 3, 4\}$) suivantes, S_i désigne la somme des aires, en u.a., des rectangles coloriés sur la figure.

a) Étape 1



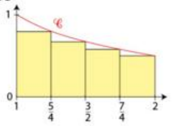
- Justifier que $S_1=\dfrac{1}{1\times 2}$

b) Étape 2



- Établir que $S_2=S_1+\dfrac{1}{3\times 4}$

c) Étape 3



- Établir que $S_3=S_2+\dfrac{1}{5\times 6}+\dfrac{1}{7\times 8}$

d) Étape 4

- Établir que $S_4=S_3+\dfrac{1}{9\times 10}+\dfrac{1}{11\times 12}+\dfrac{1}{13\times 14}+\dfrac{1}{15\times 16}$

2. Une fonction en langage Python

On poursuit la construction précédente, on obtient pour tout entier naturel $n \geqslant 1$,

$$S_n=\frac{1}{1\times 2}+\frac{1}{3\times 4}+\frac{1}{5\times 6}+\ldots+\frac{1}{(2^n-1)\times 2^n}.$$

a) Justifier que la suite (S_n) converge vers $\ln(2)$.

b) Voici une fonction **Br** écrite en langage Python.

Expliquer que, pour une valeur donnée n du paramètre, la fonction **Br** renvoie pour résultat S_n.

c) Saisir et exécuter cette fonction pour les valeurs suivantes du paramètre :

- $n=5$
- $n=10$
- $n=20$

À chaque passage dans la boucle, le compteur k est incrémenté de 2.

```python
def Br(n):
    S=0
    for k in range(2,2**n+1,2):
        S=S+1/((k-1)*k)
    return S
```

HISTOIRE DES MATHS

William Brouncker (1620-1684) est un mathématicien anglais. Ses travaux portent en particulier sur la rectification, c'est-à-dire le calcul des longueurs de certaines courbes.

Il étudie également la mesure des aires délimitées par l'hyperbole.

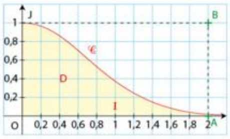
103 LA MÉTHODE DE MONTE-CARLO

Algo python

f est la fonction définie sur l'intervalle $[0\,;2]$ par $f(x)=e^{-x^2}$.

$\mathscr{C}$ est sa courbe représentative dans un repère orthonormé $(O\,;\,\vec{OI},\vec{OJ})$ et D est le domaine situé sous la courbe $\mathscr{C}$.

On note A et B les points de coordonnées respectives $(2\,;0)$ et $(2\,;1)$.



Vidéo
JAICOMPRIS.COM
Cette méthode est expliquée en vidéo

Voici une fonction **M_C** écrite en langage Python.

Elle effectue N choix au hasard d'un point dans le rectangle OABJ et compte le nombre L de points qui sont situés dans le domaine D.

Le rapport $\dfrac{L}{N}$ est alors une approximation du rapport de l'aire du domaine D à l'aire du rectangle OABJ.

```python
1 from math import *
2 from random import *
3
4 def M_C(N):
5     L=0
6     for k in range(1,N+1):
7         x= 2*random()
8         y=random()
9         if y<=exp(-x**2):
10            L=L+1
11    I=2*L/N
12    return I
```

1. Étudier un programme

a) Que représentent les variables x et y dans le contexte de la méthode décrite.

b) Expliquer le rôle de la condition $y \leqslant e^{-x^2}$ du test de la ligne 9 du programme.

c) La fonction **M_C** renvoie pour résultat le contenu de la variable I.

Que représente-t-il ?

2. Obtenir des résultats

a) Saisir ce programme.

b) Exécuter la fonction avec des valeurs du paramètre N de plus en plus grandes.

Comparer les résultats à la valeur de $\int_0^2 e^{-x^2}\,dx$ obtenue ci-contre avec une calculatrice.

$\int_0^2 e^{-x^2}\,dx$

0.8820813908

HISTOIRE DES MATHS

La méthode de Monte-Carlo permet en particulier de déterminer une valeur approchée d'une aire à l'aide d'un programme qui effectue des choix aléatoires de points dans le plan.

Elle est donc basée sur des principes probabilistes et s'applique à d'autres situations : calculs de volumes, physique des particules, calcul du risque en statistique …

Le nom de cette méthode rappelle les jeux de hasard pratiqués au casino de Monte-Carlo ; elle a été inventée en 1947 par le physicien gréco-américain Nicholas Metropolis.