Devoir - Bilan 1er semestre

* Sauf spécifications dans la/les question(s) étudiée(s), la seule méthode autorisée durant ce DS est la méthode `append(elem)`, permettant d'ajouter un élément `elem` en fin d'une liste.

```
>>> L=[1,2.5,"Coucou"]
>>> L.append([2,3])
>>> L
[1,2.5,"Coucou",[2,3]]
```

* On supposera que la bibliothèque `math` a été importée de la façon suivante :

```
import math as m
```

La bibliothèque `numpy` n'est pas autorisée.

* Ce sujet utilise la syntaxe des annotations pour préciser le type des paramètres et du résultat des fonctions à écrire. Ainsi :

```
def maFonction(n:int, X:[float], c:str, u) -> (int, np.ndarray):
```

signifie que la fonction `maFonction` prend quatre paramètres, le premier (`n`) est un entier, le deuxième (`X`) une liste de nombres à virgule flottante, le troisième (`c`) une chaine de caractères et le type du dernier (`u`) n'est pas précisé. Cette fonction renvoie un couple dont le premier élément est un entier et le deuxième un tableau numpy. **Il n'est pas demandé de recopier les entêtes avec annotations telles qu'elles sont fournies dans ce sujet. Elles sont données dans le sujet de cette façon afin de préciser plus facilement ce qui est demandé.** Ainsi, vous pouvez juste écrire sur votre copie :

```
def maFonction(n,X,c,u) :
```

* Toutes les questions sont indépendantes. Vous pouvez donc répondre à la question n sans avoir répondu à la question $n-1$ en supposant les résultats admis. Toutes les fonctions définies aux questions précédentes sont supposées disponibles et utilisables pour répondre à une nouvelle question.
* Une attention particulière sera portée à la lisibilité, la simplicité et l'efficacité du code proposé. En particulier, l'utilisation d'identifiants (de noms de variables) significatifs, l'emploi **judicieux** de commentaires et la description du principe des programmes seront appréciés.

Attention, la difficulté du sujet est globalement linéaire, mais la dernière partie est intrinsèquement plus simple que d'autres. Gardez donc un peu de temps si besoin pour traiter cette dernière partie.

I - Étude de la dispersion des valeurs de résistance d'un stock de résistors

Objectif

Dans cet exercice, on se place dans le cas hypothétique d'un fabricant de résistors électriques. Celui-ci dispose d'un stock conséquent de ces composants électroniques. Pour un résistor donné, sa résistance peut varier au cours du temps. On souhaite savoir si cette dispersion des valeurs des résistances reste en dessous d'un certain seuil, permettant de s'assurer que 90% du stock reste viable. Pour cela, en se basant sur une densité de probabilité de la valeur de résistance d'un résistor, on pourra, à l'aide d'approches numériques, d'intégration et de dichotomie, déterminer la valeur de dispersion maximale autorisée et la comparer à la dispersion observée sur un échantillon des résistors disponibles dans le stock. On va donc ici utiliser des approches probabilistes et statistiques pour conclure sur la qualité du stock de résistors disponibles.

1 Introduction

Un fabricant de résistors électriques dispose en stock de 100 000 résistors dont la valeur de la résistance théorique est de 100 Ω. Ces résistors sont fabriqués par une méthode dite à « film carbone ». Cependant, bien que le processus de fabrication soit similaire pour chacun de ces résistors, des paramètres aléatoires entrent en jeu et peuvent amener à des valeurs de résistance différentes d'un résistor à l'autre. Cela s'explique par :

- la présence d'impuretés dans le matériau ;
- les défauts cristallins ;
- quelques variations dans le processus de fabrication (température de recuit ou contraintes mécaniques dans le matériau) ;
- les conditions de stockage. Les résistors sont sensibles à l'humidité et aux variations de température.

Ces variations de résistance peuvent aller de $\pm 5\%$ à $\pm 10\%$.

Un potentiel client est prêt à acheter tout le stock, mais à condition qu'au moins 90% des résistors aient une variation de résistance inférieure à $\pm 3\%$. C'est-à-dire que sur les 100 000 résistors, l'acheteur souhaite qu'au moins 90 000 aient une valeur de résistance comprise entre 97 et 103 Ω.

Ne voulant pas tester, un à un, chacun des résistors, le fabricant propose de valider ou d'invalider son stock selon une approche statistique.

Figure 1 – Un résistor

2 Densité de probabilité

D'après le descriptif précédent, la variation de la valeur des résistances peut être vue de manière relativement aléatoire. Afin de modéliser la dispersion de la valeur de la résistance d'un résistor, on utilise alors une loi normale. Cette loi normale est définie par :

$$dens : \begin{array}{ccc} \mathbb{R} & \longrightarrow & \mathbb{R} \\ R & \longmapsto & \dfrac{1}{\sigma \times \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{R-\mu}{\sigma}\right)^2} \end{array}$$

avec R une valeur de résistance, μ : espérance de la loi normale, σ : écart-type de la loi normale.

Dans notre cas d'étude, μ sera considéré comme étant égal à $100\,\Omega$, c'est-à-dire que l'on va supposer que la répartition des valeurs de résistance du stock est centrée autour de $100\,\Omega$. **Le paramètre qu'il va falloir déterminer pour s'assurer que 90% du lot soit bien compris entre 97 et $103\,\Omega$ est donc le paramètre σ. C'est l'objectif de la partie 5 du sujet.**

Cette loi normale est une loi de densité de probabilité. Pour illustrer les différents concepts présentés, on a choisi ici de tracer la courbe de la Figure 2 avec $\mu = 100\,\Omega$ et $\sigma = 10\Omega$ (valeur arbitrairement choisie). Cette densité de probabilité se lit de la façon suivante :

- Pour une valeur de résistance R donnée, la loi de densité permet de connaître la probabilité d'apparition de cette valeur. Par exemple, la probabilité qu'un résistor ait la valeur de $95\,\Omega$ est **exactement** de 0.0352. C'est-à-dire, sur cet exemple, que 3,52% des résistors auront exactement la valeur de $95\,\Omega$ (lorsque $\mu = 100\,\Omega$ et $\sigma = 10\,\Omega$).
- En réalisant l'intégration de cette loi de densité jusqu'à une contrainte R_{lim} donnée, on obtient la valeur $\mathcal{P}_r(R_{lim})$. Cette valeur renseigne alors sur la probabilité que le résistor ait une résistance au plus de R_{lim}. Par exemple, $\mathcal{P}_r(95) \approx 0{,}31$, ce qui signifie qu'il y a 31% de probabilité qu'un résistor ait une valeur de résistance $R \leqslant 95\Omega$ (lorsque $\mu = 100\,\Omega$ et $\sigma = 10\,\Omega$).

On précise bien que la valeur de $\sigma = 10\,\Omega$ est donnée ici pour illustrer les différentes approches faites dans les parties 2, 3 et 4 du sujet.

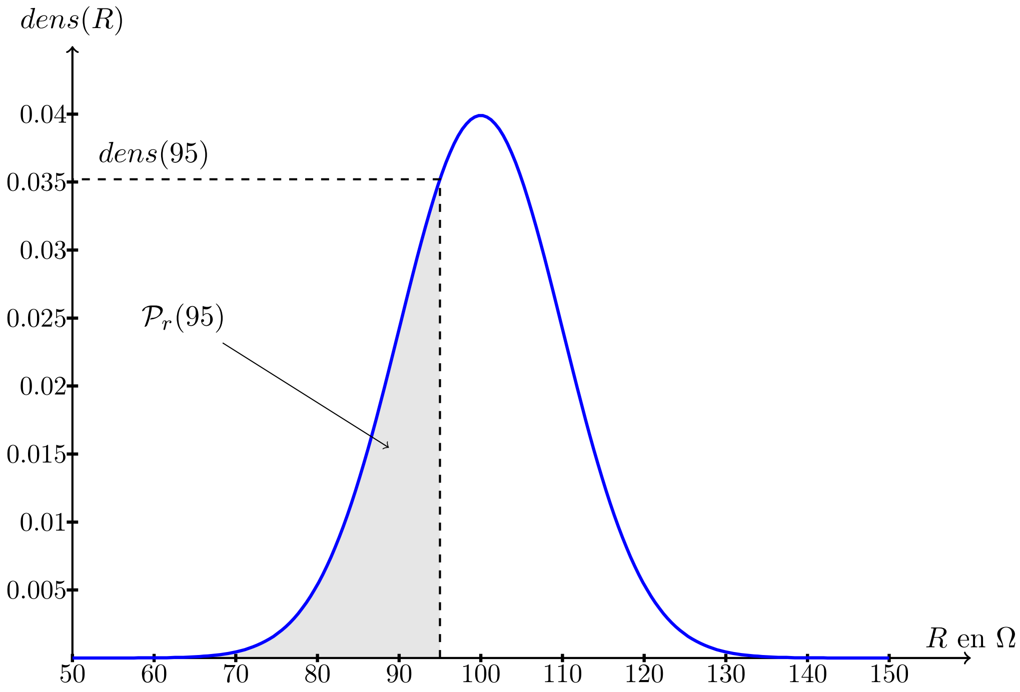


Figure 2 – Densité de probabilité de résistance d'un résistor.

On cherche à tracer la loi de densité présentée en FIGURE 2. **On rappelle que dans tout cet exercice, la bibliothèque NumPy n'est pas autorisée et que la bibliothèque `math` a été importée grâce à la commande `import math as m`.**

Q1. Pour les 4 propositions suivantes, précisez sur votre copie celles qui sont vraies et celles qui sont fausses.

Prop 1 : Dans le cas de la FIGURE 2, il y a quasiment 0% de chance qu'un résistor ait **exactement** une valeur de $150\,\Omega$.

Prop 2 : Dans le cas de la FIGURE 2, il y a 50% de chance qu'un résistor ait une valeur inférieure ou égale à $100\,\Omega$.

Prop 3 : On peut dire que la probabilité qu'un résistor ait une valeur supérieure à $95\,\Omega$ est de $1 - \mathcal{P}_r(95)$.

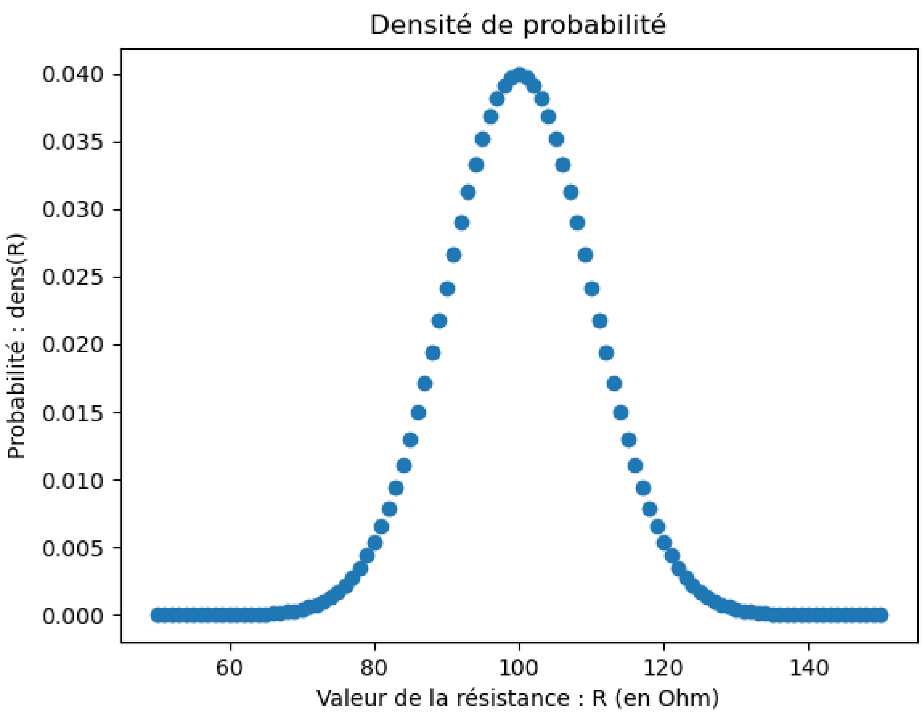
Prop 4 : Dans le cas de la FIGURE 2, la probabilité qu'un résistor ait **exactement** une valeur de $120\,\Omega$ est d'environ 5%.

Q2. Définir une fonction `dens(R:float,mu:float,sigma:float)->float` permettant de calculer $dens(R)$ pour des valeurs d'espérance `mu` et d'écart-type `sigma` données. Ainsi, `dens(95,100,10)` renverra 0,0352.

Q3. Créez une liste de 101 éléments allant de 50 à 150 inclus. Cette liste sera nommée `abscisse` et sera la liste utilisée comme abscisse pour tracer le graphe de la densité de probabilité.

Q4. Créez maintenant la liste `ordonnee` qui va contenir les 101 évaluations de la fonction *dens* de la liste `abscisse`. (On rappelle que `mu`=100 et `sigma`=10, pour créer cette figure.)

Q5. D'après la documentation Matplotlib fournie en annexe, donnez le code permettant de tracer la courbe ci-dessous. Vous serez attentifs à fournir le titre du graphe, ainsi que celui des axes des abscisses et des ordonnées.



3 Méthodes d'intégration numérique d'une fonction

Le document en Annexe 1 présente différentes méthodes d'intégration numérique. On pose n le nombre de rectangles ou de trapèzes utilisés pour réaliser ces intégrations numériques.

Q6. Exprimez pas en fonction des bornes **a** et **b** et du nombre n de rectangles/trapèzes.

Q7. Exprimez **g** et **d** en fonction de i, **a** et de pas. i correspond au numéro du rectangle considéré.

Q8. D'après l'annexe 1, pour la méthode dite des rectangles à gauche, donnez l'expression mathématique de l'aire $\mathcal{A}_i$ du rectangle i en fonction de **g**, pas et f.

Q9. Écrivez une fonction `integration_rectangle_gauche(f:function,a:float,b:float,n:int)->float` permettant de calculer l'intégrale d'une fonction f à valeur $f(x)$ pour $x \in [a, b]$ avec n rectangles et s'appuyant sur la méthode dite des rectangles à gauche.

Q10. Quelle(s) modification(s) faut-il faire pour réaliser l'intégration avec la méthode des rectangles à droite et des trapèzes ? Donner les lignes de code correspondantes pour réaliser ces modifications.

4 Intégration de la fonction de densité de probabilité

La loi de probabilité est définie par

$$\begin{array}{rccc} \mathcal{P}_r : & \mathbb{R}^+ & \longrightarrow & [0,1] \\ & R_{lim} & \longmapsto & \displaystyle\int_{-\infty}^{R_{lim}} dens(R)\mathrm{d}R \end{array} \qquad \text{avec } \mu \text{ et } \sigma \text{ fixés.}$$

Au vu de la courbe de la densité de probabilité en FIGURE 2, on suppose que

$$\int_{-\infty}^{0} dens(R)\mathrm{d}R = 0 \qquad \text{et} \qquad \int_{200}^{+\infty} dens(R)\mathrm{d}R = 0$$

Ce qui revient à définir $\mathcal{P}_r$ par :

$$\begin{array}{rccc} \mathcal{P}_r : &]0,200] & \longrightarrow & [0,1] \\ & R_{lim} & \longmapsto & \displaystyle\int_{0}^{R_{lim}} dens(R)\mathrm{d}R \end{array}$$

Dans la partie précédente, on a créé une fonction `integration_rectangle_gauche` qui fonctionnait pour une fonction f avec un paramètre x. Cependant, nous avons défini en question **Q2**, la fonction `dens` avec 3 paramètres : la valeur de la résistance R, l'espérance μ et l'écart-type σ.

Q11. Réécrivez une fonction `Integrale(dens,a,b,n,mu,sigma)`, permettant de calculer la valeur de l'intégrale de la fonction $dens$ sur l'intervalle $[a, b]$ pour des valeurs de μ et σ données.

Q12. Quelle ligne de commande faut-il alors écrire pour réaliser le calcul de $\mathcal{P}_r(95)$ avec 20 rectangles, pour $\mu = 100\,\Omega$ et $\sigma = 10\,\Omega$?

Q13. Quelle est la valeur de $\displaystyle\int_0^{200} dens(R)\mathrm{d}R$? Justifier votre réponse.

Q14. Écrire un code permettant de déterminer à partir de combien de rectangles le résultat de l'intégrale de la question précédente est atteint avec une précision de 1×10^{-4}. Vous initialiserez votre recherche avec 3 rectangles. (Si vous ne connaissez pas le résultat à la question précédente, on le notera pour cette question comme étant égal à res).

Le code de la question **Q14** permet de déterminer que le nombre nécessaire de rectangles est de 15. On s'est également intéressé à trouver le résultat de $\mathcal{P}_r(95)$ en fonction du nombre de rectangles utilisés pour le calcul d'intégration. Les résultats sont donnés en FIGURE 3.

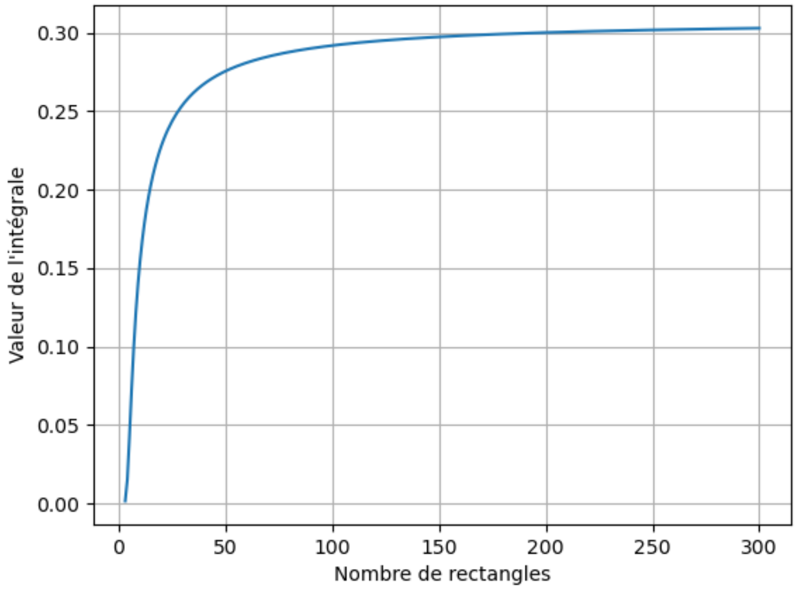


FIGURE 3 – Convergence de la méthode d'intégration pour le calcul de $\mathcal{P}_r(95)$

Q15. On rappelle que $\mathcal{P}_r(95) \approx 0{,}31$. Expliquez pourquoi 15 rectangles ne sont pas suffisants pour obtenir la même précision qu'avec le calcul d'intégrale précédent.

Dans la suite de l'exercice, on utilisera 300 rectangles pour mener le calcul d'intégration.

5 Recherche de l'écart-type maximal possible

On rappelle que le potentiel acheteur prendra le stock de résistors disponibles uniquement si au moins 90% d'entre eux ont une valeur de résistance comprise entre 97 et 103 Ω. L'objectif de cette partie est de déterminer numériquement la valeur maximale possible de σ, écart-type de la loi normale utilisée, pour répondre à cette contrainte. μ sera toujours considéré comme étant égal à 100 Ω.

On pose maintenant $\mathcal{P}_r(R_{lim}, \sigma)$ la probabilité qu'un résistor ait une valeur de résistance $R \leqslant R_{lim}$ pour une valeur d'écart-type σ donnée. (Par exemple, on a vu dans la partie 2 que $\mathcal{P}_r(95, 10) = 0.31$).

Q16. Par rapport à la problématique posée, que représente en terme de probabilité
$\mathcal{P}_r(103, \sigma) - \mathcal{P}_r(97, \sigma)$?

Q17. Complétez sur votre copie l'expression mathématique de $\mathcal{P}_r(103, \sigma) - \mathcal{P}_r(97, \sigma)$ donnée ci-dessous. On précise que σ est ici un paramètre fixé.

$$\mathcal{P}_r(103, \sigma) - \mathcal{P}_r(97, \sigma) = \int_{\dots}^{\dots} dens(R, \sigma)\mathrm{d}R$$

On pose maintenant :

$$\mathcal{DP}_r(\sigma) = \mathcal{P}_r(103, \sigma) - \mathcal{P}_r(97, \sigma)$$

Q18. Définissez une fonction `DPr(sigma:float)->float` permettant de calculer $\mathcal{DP}_r(\sigma)$ pour une valeur de σ donnée. On précise que les calculs d'intégrales nécessaires se feront avec 300 rectangles et que $\mu = 100\,\Omega$.

Afin de déterminer la valeur de σ permettant de valider la condition que 90% des résistors aient une valeur de résistance comprise entre 97 et 103 Ω, on va appliquer une stratégie de dichotomie.

Q19. Exprimez mathématiquement en fonction de $\mathcal{DP}_r$ le problème pour lequel on va chercher le zéro.

La fonction dont on recherche le zéro est notée `f_recherche_0(sigma:float)->float`.

Q20. Mettez en place la fonction `dichotomie(f_recherche_zero, borne_g, borne_d, epsilon)` permettant de trouver la valeur de zéro de la fonction `f_recherche_zero`. Le critère d'arrêt sera atteint lorsque l'intervalle de recherche sera plus petit que `epsilon`. Les bornes de recherche initiale de la dichotomie seront `borne_g` et `borne_d`.

Q21. Pour `epsilon`$=10^{-4}$, combien d'itérations seront nécessaires pour converger avec `borne_g`$=1$ et `borne_d`$=10$? (On ne demande pas d'application numérique).

6 Validation ou invalidation du stock

La réponse à la question **Q20** permet de déterminer que le paramètre σ à ne pas dépasser pour s'assurer qu'au moins 90% des résistors aient une valeur de résistance comprise entre 97 et 103 Ω est de 1.82.

Il s'agit maintenant de vérifier si les résistors présents dans le stock présentent un écart-type sur leur valeur de résistance plus petit ou plus grand que 1.82. Pour cela, le fabricant va tester de manière aléatoire 5% de son stock et faire une estimation de l'écart-type.

Les 5000 mesures des échantillons de résistors choisis aléatoirement sont stockées dans une liste `L`. L'estimation de l'écart-type σ à partir des valeurs contenues dans `L` est donnée par :

$$\sigma = \sqrt{\frac{1}{n-1}\sum_i (\bar{\mathtt{L}} - \mathtt{L}_i)^2}$$

avec $\bar{\mathtt{L}}$ la moyenne des valeurs de la liste `L`, $\mathtt{L}_i$ la $i^{\text{ème}}$ valeur de la liste `L` et n la longueur de la liste `L`.

Remarque : On constate que la formule est légèrement différente de celle pour le calcul de l'écart-type sur une liste. La raison en est qu'ici, on fait une estimation de l'écart-type réel à partir d'un échantillon statistique.

Q22. Codez une fonction `moyenne(L:list)->float`, permettant de calculer $\bar{\mathtt{L}}$.

Q23. Créez maintenant une fonction `estimation_ecart_type(L:list)->float` permettant de calculer l'estimation de σ.

Q24. Quelle est la complexité de votre fonction `estimation_ecart_type` ? Justifiez clairement votre réponse.

Le résultat obtenu est que l'estimation de σ est de 1.5.

Q25. Concluez sur le fait que l'acheteur potentiel peut acheter ou non le stock de résistors.

Q26. À partir de la liste `L`, créez le dictionnaire `dico` ayant pour clés : `'valable'` et `'non valable'`, et pour valeurs respectives le nombre de résistors ayant une résistance comprise entre 97 et 103 Ω et le nombre de résistors en dehors de cet intervalle.

Q27. À partir de `dico`, déterminez le pourcentage de résistors valables.

Annexe 1 - Méthodes d'intégration numérique

I - Méthode dite des rectangles à gauche

La valeur de l'intégrale d'une fonction f continue et intégrable sur l'intervalle $[\mathbf{a}, \mathbf{b}]$ peut être approchée par une méthode des rectangles. C'est-à-dire par une méthode qui évalue l'aire sous la courbe en découpant celle-ci à l'aide de n rectangles. Ici, on illustre en FIGURE 4 la méthode des rectangles dite à gauche.

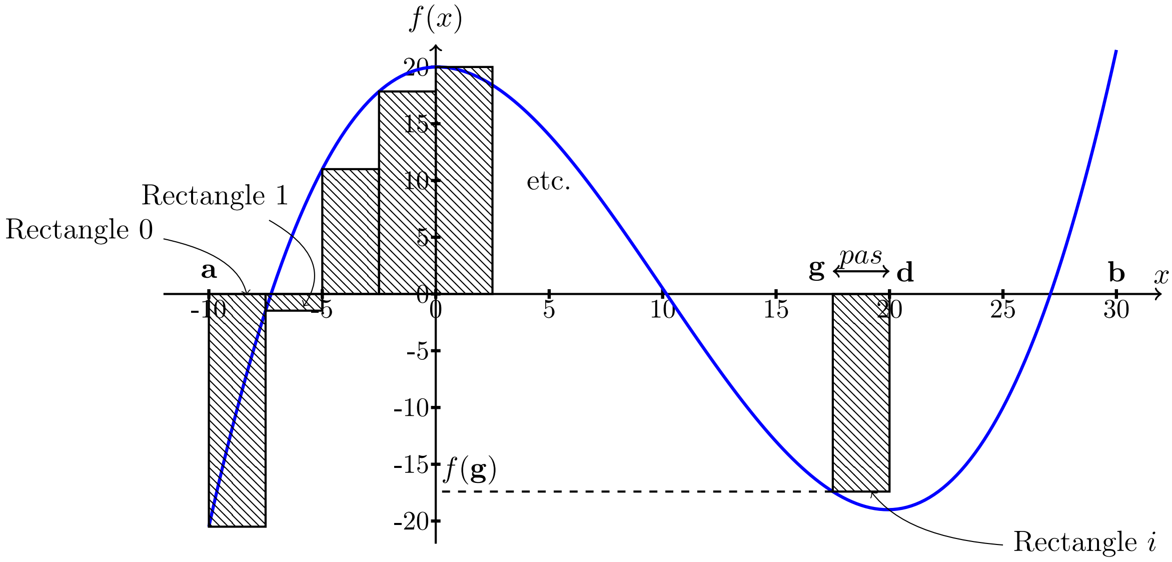


FIGURE 4 – Intégration numérique par la méthode des rectangles à gauche

L'intervalle $[\mathbf{a}, \mathbf{b}]$ est découpé en n subdivisions. On évalue l'aire sous la courbe pour chaque subdivision i, en calculant l'aire $\mathcal{A}_i$ du rectangle i, de largeur *pas* et de hauteur $f(\mathbf{g})$ avec $\mathbf{g}$ la borne gauche de la subdivision considérée. L'intégrale $\displaystyle\int_{\mathbf{a}}^{\mathbf{b}} f(x)\mathrm{d}x$ est alors approchée comme la somme des aires des rectangles.

$$\int_{\mathbf{a}}^{\mathbf{b}} f(x)\mathrm{d}x \approx \sum_{i=0}^{n-1} \mathcal{A}_i$$

On dit que la fonction f est intégrable au sens de Riemann lorsque :

$$\lim_{n\to\infty} \sum_{i=0}^{n-1} \mathcal{A}_i = \int_{\mathbf{a}}^{\mathbf{b}} f(x)\mathrm{d}x$$

II - Méthode dite des rectangles à droite

Cette méthode est totalement similaire à la précédente, excepté que la hauteur du rectangle i considéré est égale à $f(\mathbf{d})$.

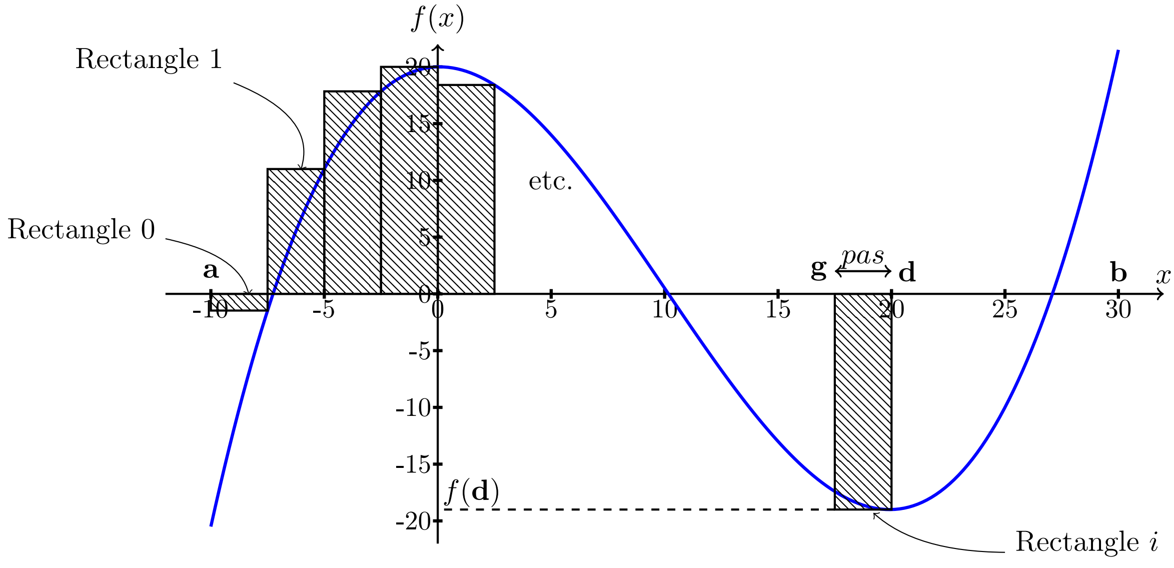


Figure 5 – Intégration numérique par la méthode des rectangles à droite

III - Méthode dite des trapèzes

Cette méthode est totalement similaire aux deux précédentes, excepté que sur la subdivision i, l'aire sous la courbe n'est plus approximée à l'aide d'un rectangle mais d'un trapèze dont la longueur des bases est $f(\mathbf{g})$ et $f(\mathbf{d})$ et la hauteur égale au *pas*.

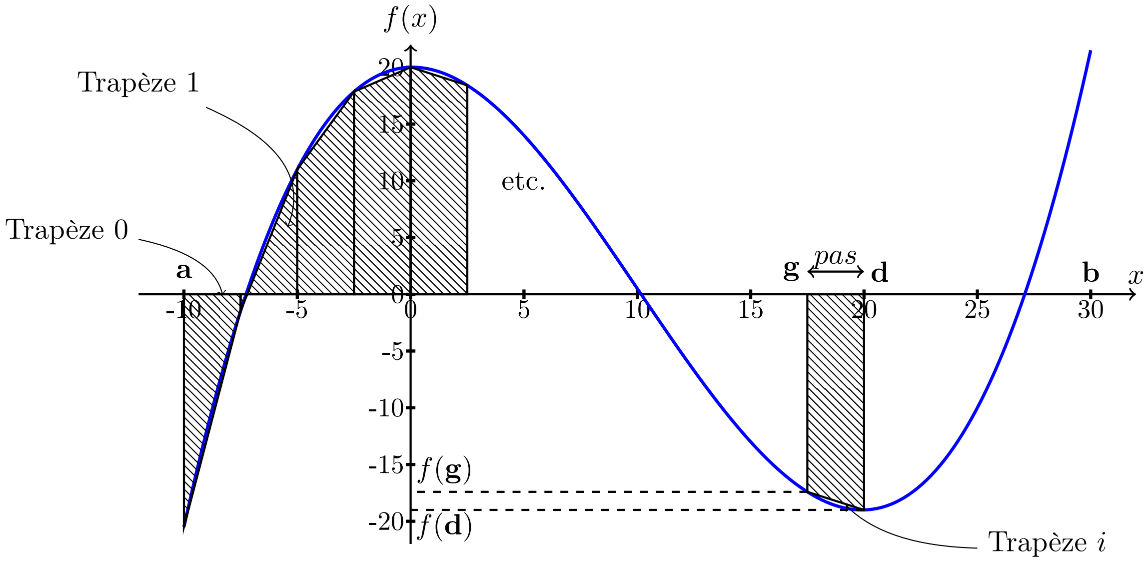


Figure 6 – Intégration numérique par la méthode des trapèzes

Annexe 2 - Commandes Matplotlib

Vous trouverez ci-dessous un extrait de la documentation officielle du module `matplotlib.pyplot`, dédié au tracé et à la mise en forme de graphiques.

Fonction plot

`matplotlib.pyplot.plot(*args,**kwargs)`

Plot lines and/or markers to the Axes. `args` is a variable length argument, allowing for multiple x, y pairs with an optional format string. For example, each of the following is legal :

```
plot(x,y) # plot x and y using default line style and color
plot(x,y,'bo') # plot x and y using blue circle markers
plot(y) # plot y using x as index array 0...N-1
plot(y,'r+') # ditto, but with red plusses
```

An arbitrary number of x, y, fmt groups can be specified, as in :

```
import matplotlib.pyplot as plt
plt.plot(x1,y1,'g^',x2,y2,'g-')
```

character	**description**	**character**	**description**
'-'	solid line style	' s'	square marker
'–'	dashed line style	'p'	pentagon marker
'-.'	dash-dot line style	'*'	star marker
' :'	dotted line style	'h'	hexagon1 marker
'.'	point marker	'H'	hexagon2 marker
','	pixel marker	'+'	plus marker
'o'	circle marker	'x'	x marker
'v'	triangle_down marker	'D'	diamond marker
'^'	triangle_up marker	'd'	thin_diamond marker
'<'	triangle_left marker	'|'	vline marker
'>'	triangle_right marker	'_'	hline marker
'1'	tri_down marker	'3'	tri_left marker
'2'	tri_up marker	'4'	tri_right marker

The following color abbreviations are supported :

character	**description**
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

Fonctions annexes pour la mise en forme

En plus de la fonction `plot`, le module `matplotlib.pyplot` propose diverses fonctions dédiées à la mise en forme des graphiques. En voici quelques-une :

- `xlabel(s)` : écrit le contenu de la chaîne `s` comme étiquette des abscisses.
- `ylabel(s)` : écrit le contenu de la chaîne `s` comme étiquette des ordonnées.
- `title(s)` : écrit le contenu de la chaîne `s` comme titre du graphique.
- `legend(L)` : donne une légende au graphique. `L` doit être une liste de chaînes de caractères : `L[0]` est la légende de la première courbe, `L[1]` de la deuxième, etc.