Manipuler les nombres réels (avec solutions)

Laurent Garnier

# À qui s’adresse ce livre

english Mathematics is a language.  
[Josiah Willard Gibbs](https://en.wikipedia.org/wiki/Josiah_Willard_Gibbs)

Ce livre s’adresse initialement aux élèves de seconde du système scolaire français. Mais il sera probablement très utile à un élève de fin de collège qui voudrait assurer son entrée au lycée. Il sera également très utile à un élève de première qui voudrait s’assurer qu’il maîtrise bien les notions de la classe de seconde. Enfin un quatrième public qui pourrait directement être visé est celui des enseignants, profs particuliers ou encore tuteurs (professionnels ou amateurs) qui souhaiteraient consulter des ressources utiles et utilisables pour aider les lycéens.

De façon plus générale, même si je reprends exactement la structure du programme officiel que vous pouvez consulter ici : <https://eduscol.education.fr/document/24553/download>, il me semble que ce livre pourrait être utile à quiconque souhaite approcher les mathématiques de façon honnête, rigoureuse et avec un soucis de compréhension. Tout en respectant le niveau du programme j’essaierai dès que possible de prolonger la perspective.

Vous pouvez d’ores et déjà mettre en pratique de façon interactive toutes les notions abordées grâce aux codes Python accessible sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

Si vous constatez une erreur que ça soit une erreur de calcul, une faute de frappe ou quoique ce soit merci de me contacter à l’adresse indiquée dans le formulaire afin de me la signaler et que je puisse la corriger. Idem si vous avez des suggestions d’amélioration.

Bonne lecture.

# Contenus

english Mathematics has beauty and romance. It’s not a boring place to be, the mathematical world. It’s an extraordinary place; it’s worth spending time there.  
[Marcus du Sautoy](https://en.wikipedia.org/wiki/Marcus_du_Sautoy)

## Contenu 1 : Nombres réels, droite numérique

french Tout l’univers repose sur l’ensemble des entiers naturels.  
[Pythagore de Samos](https://fr.wikipedia.org/wiki/Pythagore)

Cette partie développe la rubrique intitulée : "Ensemble des nombres réels, droite numérique" dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Savez-vous compter sur vos doigts ?

Savez-vous résoudre l’équation ? Je vous laisse réfléchir et je vous donnerai la solution plus tard.

Imaginez que vous êtes en train de jouer à votre jeu vidéo préféré. Il vous reste 3 points de vie, combien pouvez-vous en perdre avant de mourir (dans le jeu) ? Idem, je vous laisse réfléchir.

En général, lorsque vous entrez en seconde vous êtes âgé de 15 ans, du coup combien d’années vous séparent de votre majorité ?

Ces trois questions traitent du même genre de problèmes, la résolution des équations algébriques du premier degré impliquant les nombres les plus simples que vous connaissez à savoir les nombres entiers naturels. Pour la faire simple il s’agit des nombres utiles pour compter le nombre de doigts que vous possédez, le nombre d’enfants dans une famille et toutes les choses qu’on peut dénombrer de zéro à l’infini.

On note l’ensemble des entiers naturels

### Tout est relatif

Mais ces nombres sont insuffisants pour exprimer des températures par exemple dans une ville d’Europe du Nord comme Saint Pétersbourg en Russie ou Oslo en Norvège. En fait, même en France métropolitaine si vous allez en Savoie vous observerez des températures négatives en hiver et positive en été. Le signe de ces températures est *relatif* à la position du zéro.

Les nombres entiers relatifs sont constitués des nombres entiers naturels (les nombres positifs qu’on a vu précédemment) auquels on ajoute leurs symétriques par rapport à zéro.

On les note .

Ce sont également ces nombres qu’on utilise dans un ascenseur pour indiquer si on va au sous-sol (nombre négatif) ou aux étages supérieurs (nombres positifs).

Dans le langage de programmation Python les nombres entiers relatifs sont représentés par le type int pour *integer* (qui veut dire entier relatif en anglais).

### Les ordinateurs s’en contentent

Mais ces nombres sont encore en nombre insuffisant pour exprimer par exemple les unités monétaires. Quand vous achetez une baguette, du carburant, des fournitures scolaires ou quoique ce soit ; la plupart du temps vous utilisez des nombres à virgules qu’on appelle les nombres décimaux.

Contrairement aux deux ensembles précédents, l’ensemble des nombres décimaux ne peut pas être représenté de façon explicite avec des points de suspension, il doit être définit en *compréhension*. Définir un ensemble en compréhension signifie que vous devez expliquer la règle de construction.

On note

l’ensemble des nombres décimaux avec le symbole qui se lit *tel(s) que* et le symbole qui se lit *appartient à*.

Commençons par la fin signifie que et sont tous les deux des entiers relatifs. Ainsi, les décimaux sont les nombres fractionnaires dont le numérateur est un entier relatif et le dénominateur une puissance de 10. Donnons quelques exemples :

Dans la programmation Python on parle du type float pour *floating point number* ou nombre à virgule flottante en bon français (puisque les anglo-saxons utilisent un point comme séparateur décimal). En prenant on obtient tous les entiers relatifs. Mais ce n’est pas suffisant pour décrire le réel (en théorie, parce qu’en pratique les ordinateurs s’en contentent).

### Êtes-vous rationnels ?

Imaginez que vous avez un budget pour une semaine et que vous souhaitiez allouer une part équitable pour chaque jour. Et bien sachez que le nombre n’est pas un décimal, pas plus que le nombre . En effet si vous posez la division ou que vous utilisez une machine (calculatrice, ordinateur, tablette ou téléphone peu importe) vous verrez que et que cette séquence se répètent à l’infini. De même, de façon plus simple, si vous prenez les 6 se répètent à l’infini.

Posons où les 6 se répètent à l’infini. Multiplions par 100 ainsi on obtient . Multiplions par 10 ainsi on obtient . Faisons maintenant la soustraction d’où . Le nombre est bien un nombre qui sort de la définition des décimaux et justifie une nouvelle classe de nombres, les nombres rationnels (ratio en latin veut dire quotient).

Les nombres rationnels sont notés

Ici le numérateur est un entier relatif et le dénominateur est un entier naturel strictement positif parce qu’on ne peut pas diviser par zéro.

Prouvons ce dernier résultat. Imaginez que l’on puisse trouver un nombre entier qui soit divisible par zéro. Alors il existerait un entier tel que . Le problème c’est que zéro est un élément absorbant, tel le trou noir de la multiplication, tout nombre multiplié par zéro donne zéro. La division par zéro n’a donc pas de sens.

Donnons quelques exemples de nombre rationnels :

### Au-delà du réel

Les nombres rationnels sont très intéressants mais ils restent insuffisants pour capturer le réel. Imaginez que l’écran de votre ordinateur portable soit un rectangle de longueur 16 cm pour une largeur de 9 cm (c’est le fameux format 16/9). Alors pour calculer la diagonale nous allons utiliser le théorème de Pythagore :

Tout d’abord le nombre 337 est un nombre premier (prouvez-le, c’est un bon exercice). Et ensuite la racine carrée d’un nombre premier ne peut pas s’écrire sous la forme d’un nombre rationnel.

Pour des raisons de concordance avec le programme de seconde je vous propose de démontrer que n’est pas un nombre rationnel et pour le résultat (plus général) énoncé précédemment, je vous renvoie aux approfondissements (mais vous pouvez déjà commencer à chercher).

Pour démontrer que n’est pas un nombre rationnel on va utiliser plusieurs types de raisonnement. Tout d’abord nous allons utiliser le raisonnement par l’absurde. Le raisonnement par l’absurde repose sur un principe de la logique classique qui dit qu’une affirmation est soit vraie soit fausse. En logique mathématique on ne s’occupe que de ce genre d’affirmation donc nous sommes dans un système **beaucoup** plus simple que la réalité. Du coup, si le contraire de ce que je veux montrer est faux alors ce que je veux montrer est vrai. Prenons un exemple concret un peu simpliste, si ce n’est pas le matin alors c’est l’après-midi. Si vous n’êtes pas majeur alors vous êtes mineur. Ce genre de logique fonctionne bien avec les systèmes binaires.

Revenons à nos moutons, on veut montrer que n’est pas rationnel donc on va supposons le contraire et montrer que ce contraire est faux donc que notre affirmation sera vraie.

Supposons donc par l’absurde que soit rationnel donc d’après la définition de il devrait exister un entier relatif et un entier naturel strictement positif tels que . Comme toute fraction peut se ramener (après simplification) à une fraction irréductible on peut donc considérer que et n’ont pas de diviseurs communs et que la fraction est irréductible. Dire que et n’ont pas de diviseurs communs revient à dire que leur pgcd vaut un c’est-à-dire . Le PGCD est le Plus Grand Commun Diviseur. Bon on devrait l’appeler le PGDC parce qu’en français on dit diviseur commun car l’adjectif vient après le nom mais en anglais ils disent *greatest common divisor* et bien souvent les gens importent les concepts anglophones en oubliant les règles du français c’est pour ça que sur certaines calculatrices et logiciels vous pouvez voir GCD ou gcd.

Faisons un bref récapitulatif :

* Objectif : Montrer que n’est pas rationnel
* Méthode : Raisonnement par l’absurde, on pose et avec

Du coup, on va élever au carré ce qui nous donne et qu’on peut mettre sous la forme .

On peut classer les nombres entiers en deux catégories :

1. Les nombres pairs de la forme où est un entier.
2. Les nombres impairs de la forme où est un entier.

On en déduit que le nombre appartient à la première catégorie avec .

Désormais on va avoir besoin d’un résultat intermédiaire qui va faire intervenir un autre type de raisonnement (pour voir les différents types de raisonnement consulter la playlist éponyme en cliquant [ici](https://www.youtube.com/watch?v=R_L4NEgIxPM&list=PLwWStLtwGECb-3dJmBXOYc8oQw1mRsrep&pp=iAQB)).

Voici le résultat intermédiaire qu’on veut montrer : *si le carré d’un nombre est pair alors le nombre initial est aussi pair*.

On va formaliser cette affirmation :

1. A = "le carré d’un nombre est pair"
2. B = "le nombre initial est pair"
3. P = A B

Le nouveau type de raisonnement qu’on va utiliser s’appelle le raisonnement par contraposée.

Prenons un exemple concret fictif (et simpliste). Imaginez que chaque fois qu’il pleut je prenne mon parapluie (sans **aucune** exception). Si vous me croisez **sans** parapluie alors vous en déduirez qu’il ne pleut pas. C’est une proposition logiquement équivalente à la première formulation.

1. A = "Il pleut"
2. B = "Je prend mon parapluie"
3. non B = "Je n’ai pas de parapluie"
4. non A = "Il ne pleut pas"

Revenons à nos nombres entiers. La contraposée de : "Si le carré d’un nombre est pair alors le nombre initial est aussi pair." est "Si le nombre initial est impair alors son carré aussi."

Décomposons :

1. A = "le carré d’un nombre est pair"
2. B = "le nombre initial est pair"
3. P = A B
4. non(B) = "le nombre initial est impair"
5. non(A) = "le carré est impair"
6. CP(P) = non(B) non(A)

Ce raisonnement nous permet de ramener la proposition à montrer à une proposition plus simple et qu’on peut montrer directement.

Si le nombre initial est impair alors il est de la forme . Il vient que son carré satisfait :

On a bien obtenu un nombre impair. Par équivalence logique la proposition initiale est vraie.

Faisons un bilan d’étape.

1. Objectif final : Montrer que n’est pas rationnel
2. Méthode : Raisonnement par l’absurde, on pose et avec
3. Objectif intermédiaire : est pair donc est pair.
4. Méthode pour l’objectif intermédiaire : raisonnement par contraposée, si est impair alors aussi

À ce stade nous avons bien montré que entraîne que . De cette nouvelle écriture on tire et on peut établir une nouvelle égalité qui donne . Par conséquent, en utilisant notre résultat intermédiaire, on constate que est aussi pair.

Mais si et sont pairs alors ils sont divisibles par deux ce qui contredit . Ce résultat est absurde donc l’hypothèse initiale aussi et finalement n’est pas un rationnel. CQFD

Bravo à vous si vous avez tout compris. Pour les autres, rassurez-vous, c’est normal que ça chauffe dans votre tête. Vous pouvez déjà revoir cette démonstration en vidéo ici : <https://youtu.be/R_L4NEgIxPM>

Le raisonnement par l’absurde consiste à partir d’une hypothèse contraire à celle qu’on souhaite montrer et ensuite à enchaîner des déductions logiques parfaitement valides jusqu’à aboutir à une contradiction. La chaîne de déductions peut parfois s’avérer (très) longue et impliquer des résultats intermédiaires nécessitant à leur tour (éventuellement) d’autres types de raisonnement.

Dans certains ouvrages, certaines personnes préfèrent admettre les résultats intermédiaires. Dans l’absolu il serait impossible de tout démontrer parce qu’on ne pourrait jamais avancer. Néanmoins il est utile de garder à l’esprit qu’en mathématiques le but du jeu est de démontrer les affirmations sans jamais les prendre pour argent comptant.

Finalement, puisque n’est pas un nombre rationnel il doit bien appartenir à un ensemble de nombre et cet ensemble est l’ensemble des nombres réels et on le note .

Nous avons ainsi la chaîne d’inclusions ensemblistes :

Cet ensemble de nombres correspond à l’infinité des points sur une droite et c’est pour cette raison qu’on représentera souvent les nombres réels par une droite graduée qu’on appellera la droite des réels avec 0 au milieu et qui se prolonge vers à droite et à gauche.

### Raisonnement par contraposition

Revenons un petit instant sur la mécanique du raisonnement par contraposée. Pour cela on va décortiquer l’implication logique.

$$\begin{aligned}
A &=\,"x > 2"\\
B &=\,"2x > 4"\\
P &=\,A\Rightarrow B\\
\neg B &=\, "2x \leq 4"\\
\neg A &=\, "x \leq 2"\\
CP(P) &=\,(\neg B)\Rightarrow(\neg A)
\end{aligned}$$

Le symbole est la négation logique.

### Exercice

Reprennez la même mécanique présentée ci-dessus avec cette fois $A
= "x < -1"$ et $B = "-3x > 3"$.

Voir solution [8.1.1](#sec:org7a53d3d)

### Exercice Python

1. Écrire une fonction Python imply qui prend deux arguments en entrée et renvoie la phrase logique "A implique B" avec A et B remplacés par leurs valeurs. Par exemple : imply("x > 1", "x\*\*2 > 1") renverra "’x > 1’implique ’x\*\*2 > 1’"
2. Écrire une autre fonction Python contrapose qui prend deux arguments A et B et qui renvoie la contraposée de "A implique B". Donc la fonction devra renvoyer "non B implique non A". Votre programme écrira littéralement "non" suivie de la valeur de la proposition A (ou B) parce que vous ne pouvez pas (en tout cas je ne sais pas) modifier logiquement le contenu d’une chaîne de caractères avec Python.

Voir solutions [8.1.2](#sec:org4b57ac3)

## Contenu 2 : Intervalles de nombres réels

english One of the big misapprehensions about mathematics that we perpetrate in our classrooms is that the teacher always seems to know the answer to any problem that is discussed.  
[Leon Henkin](https://en.wikipedia.org/wiki/Leon_Henkin)

Cette partie développe la rubrique intitulée : "Intervalles de . Notations et " dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Introduction de la notion d’intervalle

Jusqu’à présent nous avons traité essentiellement des égalités (une équation est une égalité). Mais dans de nombreux cas il est intéressant et utile de traiter des inégalités. Donnons quelques exemples issus du monde réel de la vie de tous les jours :

1. Il faut être âgé de plus de 14 ans pour conduire un véhicule de type mobilette ou scooter.
2. Il faut être âgé entre 12 et 25 ans pour bénéficier du tarif jeune à la SNCF.
3. Les films pornographiques sont interdits aux mineurs donc aux gens qui ont un âge inférieur à 18 ans.
4. En boxe anglaise la catégorie de poids pailles concerne les gens dont le poids est inférieur à 47,128 kg (soit 105 livres, source : <https://fr.wikipedia.org/wiki/Boxe_anglaise> )
5. En France, seuls les gens qui ont moins de 10 777€ par an sont exonérés d’impôt (0%), ensuite de 10 778 à 27 478 c’est 11%, puis 30% de 27 479 à 78 570, puis 41% de 78 571 à 168 994 et enfin 45% pour ceux qui ont plus de 168 994 (source : <https://www.service-public.fr/particuliers/vosdroits/F1419> )
6. La limite autorisée du taux d’alcool dans le sang par la loi en 2023 est de 0,5 g/L soit en équivalent 0,25 mg par litre d’air expiré (source : <https://www.legipermis.com/infractions/alcool-permis-conduire.html> )
7. L’IMC (Indice de Masse Corporelle) permet d’établir des catégories pour mesurer l’obésité avec par exemple le début de la surcharge pondérale (surpoids) à partir de 25 kg/m2 (source : <https://fr.wikipedia.org/wiki/Indice_de_masse_corporelle> )

Traduisons ces exemples concrets en inégalités mathématiques. Dans chaque cas on utilisera la lettre pour désigner la variable inconnue.

Ces inégalités peuvent être représentées graphiquement. On peut les interpréter comme des zones délimitées par des bornes inférieures et supérieures. Lorsqu’il n’y a qu’une seule borne apparente alors ça veut dire que l’autre est du type avec le signe adéquat.

Traduisons ces inégalités en intervalles :

1. on parle d’intervalle fermé.
2. on parle d’intervalle fermé.
3. on parle d’intervalle ouvert.
4. on parle d’intervalle ouvert.
5. on parle d’intervalle fermé.
6. on parle d’intervalle ouvert à droite
7. on parle d’intervalle fermé.

Alors ici il s’agissait de modéliser à partir du monde réel donc il y a des bornes implicites qui sont apparues :

1. Jusqu’à présent personne n’a dépassé le record de longévité de Jeanne Calmant qui a vécu 122 ans. Mais comme on ne peut pas prédire l’avenir alors j’ai décidé de mettre la borne même si c’est une borne théorique. Néanmoins, certains ont prédit [la mort de la mort](https://amzn.to/3OKi2os) (c’est le titre d’un livre de Laurent Alexandre).
2. Ici les bornes étaient claires.
3. Là j’ai considéré que l’âge zéro n’est jamais atteint puisque dès l’instant où l’on naît le temps s’écoule. Mais au regard de l’exemple j’aurais même pu mettre une borne inférieure plus grande.
4. Dans cet exemple c’est la même idée, personne n’a un poids de zéro et encore moins quand on pratique la boxe anglaise.
5. Pour les revenus c’est malheureusement différent, il est hélas possible d’avoir zéro revenu. Et pour information, 10 777 par an ça fait moins de 900€ par mois (environ 898,08).
6. Les gens qui ne boivent pas d’alcool peuvent avoir un taux de zéro gramme dans le sang.
7. Comme pour l’âge, le poids maximal n’est pas figé, d’ailleurs on devrait parler de masse parce que le poids est une force qui dépend du champ gravitationnel (la pesanteur) et donc il peut changer en fonction de l’endroit dans l’espace.

Traduisons ces intervalles en représentations graphiques :

1. on parle d’intervalle fermé. 14 est compris dans l’intervalle.
2. on parle d’intervalle fermé. 12 et 25 sont inclus dans l’intervalle.
3. on parle d’intervalle ouvert. 0 et 18 sont exclus de l’intervalle.
4. on parle d’intervalle ouvert. et sont exclus de l’intervalle.
5. on parle d’intervalle fermé. 0 et 10777 sont inclus dans l’intervalle.
6. on parle d’intervalle ouvert à droite. est inclu mais est exclu de l’intervalle.
7. on parle d’intervalle fermé. 25 est inclu dans l’intervalle.

Ces exemples concrets avaient pour but de vous montrer pourquoi ce principe de représentation d’ensembles de nombres réels ordonnés peut avoir un intérêt et une utilité.

Les différents types d’intervalles de sont les suivants :

1. Les intervalles ouverts à bornes finies .
2. Les intervalles ouverts à gauche à bornes finies (on peut aussi les appeler fermés à droite).
3. Les intervalles ouverts à droite à bornes finies (on peut aussi les appeler fermés à gauche).
4. Les intervalles ouverts à borne infinie à gauche .
5. Les intervalles ouverts à borne infinie à droite .
6. Les intervalles fermés à borne infinie à gauche .
7. Les intervalles fermés à borne infinie à droite .
8. Les intervalles fermés à bornes finies .
9. L’intervalle ouvert **ET** fermé de l’ensemble des réels tout entier

Il est important de rappeler qu’un intervalle est forcément ordonné donc la borne supérieure (à droite) est forcément plus grande que la borne inférieure (à gauche) sinon ça n’a strictement aucun sens.

Donnons quelques exemples :

On peut remarquer que l’ensemble des nombres entiers naturels est inclus dans l’ensemble des nombres réels positifs :

Par contre entre deux entiers naturels il y a une infinité de réels et pour les représenter on utilise un intervalle.

Prenons par exemple les deux plus petits entiers naturels zéro et un. On peut construire un intervalle qui les contient tous les deux. Le centre de cet intervalle est un nombre décimal . Construisons un nouvel interval . Le centre de cet intervalle est un nombre décimal . On pourrait continuer ainsi à l’infini en construisant des intervalles de la forme

Mais si au lieu de prendre le centre on prenait le premier tiers ? Et bien cette fois on aurait une borne supérieure qui serait un nombre rationnel.

Et si au lieu de prendre le tiers on divisait par le nombre ? Et bien cette fois on aurait une borne supérieure qui serait un nombre réel.

Bon dans ce dernier cas j’ai un peu triché parce que est déjà un réel alors que pours les et les on a pu faire la construction uniquement en utilisant des entiers naturels.

Prenons un peu la mesure de ce que nous faisons. L’intervalle a pour longueur de même que l’intervalle . Mais qu’en est-il de l’intervalle ? ? Et de façon générale, comment mesure-t-on la longueur d’un intervalle ?

C’est très simple, un intervalle étant un ensemble ordonné, on mesure sa longueur en soustrayant la borne inférieure à la borne supérieure (que l’intervalle soit ouvert ou fermé).

Il y a donc deux cas de figure :

1. Les bornes et sont finies et dans ce cas c’est juste une soustraction normale .
2. Au moins l’une des bornes est infinie et dans ce cas .

Voici quelques exemples :

La longueur d’un intervalle est aussi appelée son amplitude. Par exemple le nombre appartient à l’intervalle qui a pour amplitude . Dit autrement, le nombre est compris entre et à près.

### Exercice

Faisons un petit exercice. Pour les nombres suivants trouver des intervalles d’amplitude puis et enfin auquels ils appartiennent :

Jouez le jeu en cherchant par vous même vous progresserez alors que si vous regardez la solution directement il vous progression sera nulle.

Voir solutions [8.2.1](#sec:org43d6bc9)

### Exercice

1. À l’aide d’une machine calculez les 5 premières décimales du nombre d’or
2. Considérez la suite de nombres :

* Calculez les 5 termes suivants.

1. Établir la règle de construction de la suite.
2. Calculer les quotients successifs en prenant comme numérateur le suivant et comme dénominateur le prédécesseur ainsi .
3. La suite de nombre de la question 2 est connue sous le nom de [suite de Fibonacci](https://fr.wikipedia.org/wiki/Suite_de_Fibonacci) (mathématicien italien du XIIème siècle qui a appris l’usage des chiffres indo-arabes grâce à son père qui commerçait avec des marchands arabes près d’Alger). Combien de termes de cette suite faut-il calculer pour approcher le nombre d’or à près ? Vous donnerez les intervalles successifs.

Voir solutions [8.2.2](#sec:orgccb4036)

### Petit jeu

Faisons un petit jeu. Voici une série de nombres particuliers :

1. avec une infinité de 1
2. avec une infinité de 12
3. avec une infinité de 123
4. avec une infinité de 1234
5. avec une infinité de 12345

Pour chacun de ces nombres on va donner un intervalle d’amplitude égale à la position de la dernière décimale distincte (donc 0,1 pour , 0,01 pour , 0,001 pour et ainsi de suite) et centré en ce nombre. Puis on va déterminer la fraction qui le définit.

Voir solutions [8.2.3](#sec:orga1e3d2e)

## Contenu 3 : Distances

french Les maths, ça prend le relais dans les situations où l’intelligence habituelle est en panne. Les chaussures sont un instrument pour marcher, les maths sont un instrument pour penser. On peut marcher sans chaussures, mais on va moins loin.  
[Jean-Marie Souriau](https://fr.wikipedia.org/wiki/Jean-Marie_Souriau)

Cette partie développe la rubrique intitulée : "Notation |a|. Distance entre deux nombres réels" dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Introduction de la notion de distance

Quelle est la distance entre 0 et 1 ? Facile, c’est 1 ! Quelle est la distance entre et ? Là déjà c’est plus tout à fait la même histoire. Il va falloir réfléchir un peu à la façon de procéder. Prenons d’abord encore quelques exemples plus simples. Quelle est la distance entre et 0 ? Facile, c’est 1 ! Quelle est la distance entre et ? Vous avez trouvé ? Oui c’est 2. Quelle la distance entre et ? Oui c’est .

Bon, avant de donner la règle générale, ramenons les choses à un problème plus simple. Quelle est la distance d’un nombre par rapport à zéro ? Prenons le nombre par exemple, sa distance à zéro est tout simplement 5. Est-ce le seul nombre à cette distance de zéro ? Ben non puisque le nombre est aussi à une distance 5 par rapport à zéro. En fait, chaque nombre positif est exactement à une distance à zéro égale à lui-même et son opposé est aussi à la même distance mais de l’autre côté. Faisons un schéma :

On notera la distance du nombre par rapport à zéro. On l’appelle aussi la valeur absolue de .

Donnons quelques exemples :

Une façon simple de décrire la chose est de dire que la valeur absolue d’un nombre c’est le nombre sans le signe.

On peut désormais définir proprement la valeur absolue d’un nombre réel :

avec si et si . Autrement dit la fonction attribue le signe d’un nombre. Si un nombre est positif il est égal à sa valeur absolue sinon il est égal à son opposé.

Par exemple et .

Mais revenons à notre problème initial. Et concrétisons-le un peu d’abord. Si vous parcourez 5 km vers l’Est puis que vous parcourez 3 km vers l’Ouest alors vous vous retrouverez 2 km à l’Est de votre position initiale. Mais vous aurez parcouru 8 km (5 + 3). Si on considère votre position initiale comme le zéro et le sens vers l’Est comme le sens positif et le sens vers l’Ouest comme le sens négatif alors le calcul de la position finale devient . Pour fixer les idées on va nommer les différents points. Posons l’origine de cet axe horizontal (Est-Ouest). Soit le point d’abscisse et le point d’abscisse . La distance entre et s’obtient en faisant le calcul . Mais pourquoi devrions-nous commencer par ? Une distance n’a pas d’orientation. Que vous fassiez 100 km à l’Est ou 100 km à l’Ouest dans les deux cas la distance parcourue sera de 100 km. Donc on devrait retrouver le même résultat que le considère le point ou le point comme point de départ. Pourtant … aïe ! On dirait que ça coince… Ben pas tant que ça si on se souvient que la valeur absolue c’est le nombre sans le signe. OK mais pourquoi utiliser la valeur absolue ? Qu’est-ce qui justifie son usage ici ? Pour aller du point au point il y a trois possibilités :

1. Soit le point est avant le point qui lui-même est avant le point et dans ce cas il faut d’abord parcourir la distance de à puis retrancher celle de à . Donc si on schématise on obtient un axe horizontal orienté comme ceci

* Ici on a

1. Soit le point est avant et dans ce cas vous devez d’abord parcourir la distance de à puis celle de à . Donc si on schématise on obtient un axe horizontal orienté dans ce sens

* Ici on a

1. Soit le point est après le point qui lui-même est après le point . Donc si on schématise on obtient un axe horizontal orienté de cette façon

* Ici on a

Dans les 3 cas on aboutit au même calcul final la différence entre le grand moins le petit.

À l’issue de cette concrétisation on pourrait se dire que pour calculer la distance entre deux nombres réels il suffit de calculer la valeur absolue de leur différence. Spoiler alert, c’est exactement ça ! La distance entre les nombres réels et vaut :

Donnons quelques exemples :

Afin de revenir sur ce qu’on a déjà vu on peut aussi voir le distance entre deux réels et comme l’amplitude de l’intervalle entre le maximum et le minimum des deux. Ainsi :

### Exercice

Pour vous entraîner voici un exercice.

1. Calculer la distance entre et . Vous donnerez une approximation à 3 chiffres après la virgule.
2. Calculer la distance et . Vous donnerez une approximation à 3 chiffres près.
3. Calculer la distance et . Vous donnerez une approximation à 6 chiffres.
4. Calculer la distance entre et . Vous donnerez une approximation à 2 chiffres.
5. Calculer la distance entre et . Vous donnerez une approximation à 5 chiffres.

Voir solutions [8.3.1](#sec:org1740f2d)

### Exercice

Faisons un exercice intéressant. Renversons le point de vue. Au lieu de calculer la distance entre deux points, cherchons les points situés à une distance choisie par rapport à un point fixe.

1. Calculer les nombres à une distance 1 de 1.
2. Calculer les nombres à une distance de .
3. Calculer les nombres à une distance de .
4. Calculer les nombres à une distance de 1.
5. Calculer les nombres à une distance de .
6. Calculer les nombres à une distance de 1.
7. Calculer les nombres à une distance de .

Voir solutions [8.3.2](#sec:orga6de2e3)

## Contenu 4 : Disques sur la droite réelle

french Une bonne partie des mathématiques devenues utiles se sont développées sans aucun désir d’être utiles, dans une situation où personne ne pouvait savoir dans quels domaines elles deviendraient utiles. Il n’y avait aucune indication générale qu’elles deviendraient utiles. C’est vrai de toute la science.  
[Neumann János Lajos](https://fr.wikipedia.org/wiki/John_von_Neumann) (connu sous son nom américanisé John Von Neumann)

Cette partie développe la rubrique intitulée : "Représentation de l’intervalle puis caractérisation par la condition " dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Introduction du concept d’intervalle centré

Si vous avez essayé de comprendre ce qui s’est passé dans le dernier exercice de la partie précédente alors vous vous êtes probablement aperçu d’une chose, sur un axe horizontal graduée déterminer les nombres à une distance fixe d’un nombre donné revient à construire les bornes d’un intervalle du type

avec le nombre donné et la distance fixe.

Donnons quelques exemples pour bien comprendre :

1. Prenons alors on obtient l’intervalle en effet et d’où

* Par conséquent, tous les nombres qui sont dans cet intervalle à une distance inférieure ou égale à 1 de 1. En effet, si alors donc si on retranche 1 à chaque membre on obtient . On en déduit que soit et dans ce que soit et dans ce cas et comme alors donc on a bien

1. Prenons alors on obtient l’intervalle . D’où la représentation graphique :

* Pour tout nombre on a alors donc et finalement

1. Prenons alors on obtient l’intervalle . D’où la représentation graphique :

* Pour tout nombre on a alors donc et finalement

Une autre manière de voir les choses consiste à considérer que si

alors par conséquent tout nombre de l’intervalle est nécessairement dans l’une des deux moitiés d’où la majoration par .

Pour que ça rentre je vous propose un petit exercice.

### Exercice

Poursuivre ce que j’ai commencé. À savoir, pour chaque couple calculer l’intervalle , faites sa représentation graphique et démontrez que .

1. pour rappel
2. pour rappel

Voir solutions [8.4.1](#sec:org3e8cc0d)

### Une autre perspective

Une autre façon de voir les choses c’est de considérer que les intervalles du type sont des encadrements de à près. Par exemple si vous construisez un intervalle centré en avec vous obtenez un encadrement à plus ou moins près. Bon, pour être rigoureux ça va être difficile de centrer un intervalle sur un nombre dont on ne connaît pas la valeur exacte. Vous savez que néanmoins mais vous sentez bien venir que plus sera petit plus cela réduira l’erreur d’approximation.

Sinon pour revenir avec rigueur l’intervalle exact vous pouvez visualisez ça comme un *voisinage* du nombre .

### Faisons un peu de programmation Python

Écrions une fonction Python qui prend trois paramètres :

1. le centre de l’intervalle
2. le rayon de l’intervalle (ou la demi-amplitude)
3. le pas de la subdivision

Cette fonction renverra une liste de réels (bon numériquement ça sera des *float* donc des décimaux) contenu dans l’intervalle espacés régulièrement par pas de .

Par exemple si on fournit les arguments :

La sortie sera

def disque(a, r, h):  
 """  
 + a: le centre de l'intervalle  
 + r: le rayon de l'intervalle (ou la demi-amplitude)  
 + h: le pas de la subdivision  
 Cette fonction renverra une liste de réels (bon numériquement ça  
 sera des float donc des décimaux) contenu dans l'intervalle  
 espacés régulièrement par pas de $h$.  
 """  
 borne\_inf, borne\_sup = a - r, a + r  
 x = borne\_inf  
 disk = [x]  
 nb\_digits = round(1/h)  
 while x <= borne\_sup - h:  
 x += h  
 disk.append(round(x, nb\_digits))  
 return disk  
  
# Tests  
a, r, h = 3, 2, 0.4  
print(disque(a, r, h))  
# Sortie  
# [1, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8, 4.2, 4.6, 5.0]  
a, r, h = 3, 2, 0.5  
print(disque(a, r, h))  
# Sortie  
# [1, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0]  
a, r, h = 0, 1, 0.25  
print(disque(a, r, h))  
# Sortie   
# [-1, -0.75, -0.5, -0.25, 0.0, 0.25, 0.5, 0.75, 1.0]

### Exercice

1. Écrire une fonction Python qui prend en entrée un nombre réel (donc un *float*) et un nombre positif . La fonction devra renvoyer un message d’erreur si on lui donne négatif. Le but de cette fonction est de renvoyer les bornes supérieures et inférieures de l’intervalle centré en .

* Vous testerez votre fonction avec pour faire la racine carrée il est plus pratique d’utiliser la puissance .

1. Proposer une amélioration pour contrôler le nombre de décimales à l’affichage. Votre nouvelle fonction doit permettre d’afficher 1, 2, 3 ou autant de décimales qu’on veut mais exactement le nombre souhaité à chaque fois.
2. Écrire une fonction distance qui prend deux nombres réels en entrée et renvoie la distance entre les deux.

Voir solutions [8.4.2](#sec:org409520d)

## Contenu 5 : Nombres décimaux

french Les charmes enchanteurs de cette sublime science ne se décèlent dans toute leur beauté qu’à ceux qui ont le courage de l’approfondir.  
[Carl Friedrich Gauss](https://fr.wikipedia.org/wiki/Carl_Friedrich_Gauss)

Cette partie développe la rubrique intitulée : "Ensemble des nombres décimaux. Encadrement décimal d’un nombre réel à près" dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Introduction sur les nombres décimaux

Pour rappel on a déjà vu dans la première section qu’on note

l’ensemble des nombres décimaux avec le symbole qui se lit *tel(s) que* et le symbole qui se lit *appartient à*.

Et que signifie que et sont tous les deux des entiers relatifs. Ainsi, les décimaux sont les nombres fractionnaires dont le numérateur est un entier relatif et le dénominateur une puissance de 10.

On a déjà vu également que Python (et tous les langages de programmation) utilisent abondamment les nombres décimaux avec le type *float*. Précisément parce que les décimaux ont la particularité d’avoir un nombre fini de chiffres après la virgule. Et les ordinateurs, aussi puissants qu’ils soient, sont tous soumis aux contraintes physiques de leur capacité de stockage qui est nécessairement finie.

De toute façon, nous vivons dans un monde physique limité donc nous sommes bien obligés de manipuler des nombres finis de décimales. Même si en théorie on peut traiter des nombres réels à chaque fois qu’on les utilise concrètement on manipule en fait des décimaux bien finis. On fait des approximations à près.

Encadrer un nombre réel à près revient à construire un intervalle d’amplitude contenant .

Donnons quelques exemples :

### Exercice

Pour les nombres suivants construire 5 encadrements décimaux à pour variant de 1 à 5 (comme montré ci-dessus avec ) :

Vous pouvez le faire sur papier ou en programmant avec Python.

Voir solutions [8.5.1](#sec:orge3130dd)

### Souvenirs souvenirs

Souvenez-vous lorsque vous avez appris comment se structure le système positionnel décimal. Prenons des nombres d’années particuliers : 1212, 1515, 1999, 2002, 2020, 2022.

1. Dans le nombre 1212 il y a deux fois le chiffre 1 et deux fois le chiffre 2. Mais ils n’ont pas la même *valeur* à chaque fois. La *magie* du système positionnel décimal c’est que la *position* d’un chiffre dans un nombre indique sa *puissance* ou plutôt la puissance de 10 par laquelle on va le multiplier. Concrètement . Si on change l’ordre des chiffres on change complètement de nombre.
2. Pour 1515 c’est le même constat. .
3. Avec 1999 il y a carrément trois fois le même chiffre ! . Et si vous ajoutez 1 vous aurez un effet domino qui conduira à 2000.
4. Le nombre 2002 fait partie de la catégorie des palindromes c’est-à-dire les nombres qui peuvent être lus de gauche à droite ou inversement sans changer leur valeur. D’ailleurs profitons-en pour rappeler que lorsqu’on écrit la décomposition en puissances de 10 on peut le faire dans l’ordre des puissances croissantes si on en a envie. .
5. Pour 2020 c’est comme pour 1515 ou 1212. .
6. Enfin pour 2022 on retrouve un triplet mais espacé par un zéro qui permet justement de le différencier de 222 qui est de l’ordre de 10 fois moins. .

Mais qu’en est-il des nombres à virgules ? Et bien c’est la même chose mais avec des puissances d’exposants négatifs. Reprenons les nombres précédents et divisons les par 100 de sorte à obtenir une virgule au milieu.

D’une certaine manière on pourrait interpréter les nombres comme des suites de chiffres (donc les nombres de 0 à 9 inclus). Pour la partie entière il s’agit de sommer le produit de chaque chiffre avec sa puissance de 10 positive associée. Et pour la partie décimale il s’agit de sommer le produit de chaque chiffre avec sa puissance de 10 négative associée.

Prenons par exemple le nombre on pourrait le réécrire :

Où les désignent les chiffres qui composent la partie entière donc , , , , . Et les désignent les chiffres composant la partie décimale (on dit aussi fractionnaire) donc , , , .

On sépare d’une part la partie entière de :

et d’autre part la partie décimale (ou fractionnaire, les deux se disent) :

Avec naturellement :

Vous remarquerez la symétrie, pour la partie entière on lit de droite à gauche dans l’ordre croissant des puissances alors que pour la partie fractionnaire on lit bien de gauche à droite dans l’ordre décroissant des puissances.

Notez qu’on pourrait unifier tout ça en lisant tout de gauche à droite selon les puissances décroissantes.

Cette fois-ci désigne le chiffre associé à peu importe que soit positif ou négatif.

Ici on peut écrire en un seul développement décimal selon les puissances décroissantes de 10:

Cette façon de faire est peu habituelle voire inexistante au lycée. C’est à vous de voir si ça vous aide ou pas. D’un côté ça unifie les notations mais d’un autre ça demande un effort d’abstraction supplémentaire.

Dans tous les cas, comme toujours en mathématiques, ce qui compte c’est qu’il faut toujours chercher à comprendre les différents angles de vue possibles et leurs liaisons.

Ici, dans la rubrique sur l’encadrement à près, ce qui nous intéresse c’est la partie décimale. Plus précisément, le fait que les nombres réels peuvent s’interpréter comme les nombres dont la partie décimale est infinie. Dans se cache l’infini tout comme dans , et une infinité de nombres…

Mais souvenez-vous, dès la première section nous avions remarqué que renfermait déjà sa part d’infini. Alors quelle est la différence entre d’un côté et de l’autre

Les premiers sont rationnels alors que les seconds sont irrationnels.

## Contenu 6 : Nombres rationnels

french La mathématique est l’art de donner le même nom à des choses différentes.  
[Henri Poincaré](https://fr.wikipedia.org/wiki/Henri_Poincar%C3%A9)

Cette partie développe la rubrique intitulée : "Ensemble des nombres rationnels. Nombres irrationnels ; exemples fournis par la géométrie, par exemple et " dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Introduction aux nombres rationnels

Dans la première partie on a déjà vue que les nombres rationnels sont notés

Ici le numérateur est un entier relatif et le dénominateur est un entier naturel strictement positif parce qu’on ne peut pas diviser par zéro.

Nous avions également remarqué que les nombres dont les décimales infinies qui suivaient une structure répétitive (on parle de période) pouvaient être *capturés* sous la forme d’une fraction.

Rien que pour le plaisir du jeu, en voici quelques autres en exercice.

### Exercice

Trouver l’écriture fractionnaire des nombres suivants :

1. avec des à l’infini.
2. avec la période 076923 qui se répète à l’infini.
3. avec la période 0588235294117647 qui se répète à l’infini.

Voir solutions [8.6.1](#sec:org1029e6a)

### Petite astuce

Si vous voulez augmenter votre maîtrise des nombres rationnels alors amusez-vous avec eux. Prenez la suite de chiffres qui vous plaît que ça soit votre date de naissance ou l’année de la première victoire en coupe du monde ou ce que vous voulez et cherchez la fraction associée. Par exemple, prenons la première de victoire la France en coupe du monde. Posons où l’on considère que 1998 correspond à la période qui va se répéter à l’infini. Alors il ne reste plus qu’à décaler de 4 crans ce qui veut dire multiplier par . Ainsi . Maintenant pour se débarasser de la période il suffit de soustraire . Ainsi puisque on obtient et maintenant on a plus qu’à poser la fraction enfin on regarde si on peut simplifier. Déjà on peut simplifier par 9 parce que la somme des chiffres est un multiple de 9. Il vient . Maintenant 222 et 1111 ont l’air de bien s’entendre mais il faut se méfier des apparences. En effet, et sont premiers ente eux ça veut dire qu’ils n’ont pas de diviseurs communs (ou encore . Par conséquent on ne peut pas faire mieux que

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

Sur Google Colab vous trouverez une fonction Python get\_fraction qui prend en entrée la troncature du nombre avant la période et la période et qui renvoie en sortie le numérateur et le dénominateur de la fraction irréductible associée. Par exemple si vous appelez la fonction avec les arguments before\_period = 0.123 et period = 4 vous obtiendrez num = 1111 et denom = 9000. Vous pouvez vérifier que .

### Les nombres irrationnels

Prenez un carré ABCD de côté 1. Alors le triangle ABC est isorectangle en B (isocèle ET rectangle). Par Pythagore :

Donc vous voyez bien que le nombre est bien réel puisqu’il apparaît naturellement dès lors qu’on trace une figure géométrique ultra basique. Qui dit géométrie dit physique. Toute construction physique s’appuie nécessairement sur la géométrie. Donc le nombre a bien une exitence physique. Et pourtant, nous avions vu dans la première section qu’il n’est pas rationnel.

Voyons ensemble deux nouvelles preuves de ce résultat (vues sur [Wikipédia](https://fr.wikipedia.org/wiki/Racine_carr%C3%A9e_de_deux#Preuves_d'irrationalit%C3%A9)).

1. Preuve par parité

* Nous allons procéder avec raisonnement par l’absurde. Considérons le plus petit entier strictement positif tel que soit le double d’un carré, et soit l’entier positif tel que . Tout d’abord on en déduit que puisque entraîne qui entraîne or et sont strictement positif donc nécessairement et donc . D’après le lemme que nous avions utilisé et démontré dans la première section, si alors est pair (nous allons redémontrer ce résultat dans la section démonstration). Ainsi et donc équivaut à par conséquent . Problème, cela signifie que est un entier dont le carré soit le double d’un carré ce qui contredit la minimalité de . CQFD

1. Preuve géométrique

* Démontrer l’irrationalité de revient à démontrer que, pour une unité donnée, il n’existe pas de triangle isocèle rectangle dont les côtés sont chacun de longueur un nombre entier d’unité.
* Si un tel triangle existe, alors il en existe nécessairement un plus petit dont les côtés sont aussi de longueur entière (sa construction est donnée sur le dessin et détaillée ci-dessous). Or si un tel triangle existe, il en existe nécessairement un minimal ayant cette propriété (celui dont le côté de l’angle droit, par exemple, est minimal) d’où une contradiction.
* Soit ABC un triangle isorectangle en B et de côtés entiers. Alors, le cercle de centre A et de rayon AB coupe l’hypoténuse [AC] en B’ tel que B’C soit encore de longueur entière, puisque AC et AB’ le sont. La perpendiculaire à l’hypoténuse [AC] passant par B’ coupe [BC] en A’. Le triangle A’B’C est isocèle rectangle en B’, puisque l’angle en B est droit et l’angle en C est celui du triangle d’origine. Les droites (A’B) et (A’B’) sont les tangentes issues de A’ au cercle de centre A et de rayon AB = AB’, et donc A’B = A’B’, donc A’B = A’B’ = B’C, et A’C est de longueur entière. On peut aussi interpréter la construction comme le pliage du triangle ABC selon la bissectrice de l’angle A.
* [Figure de la preuve géométrique](https://drive.google.com/file/d/1CpRr1jzHiOtFnYS10NL7yp9OSMSFNeU5/view?usp=sharing)

### Exercice

Considérons un cube d’arête 1.

1. Montrer que la diagonale principale (reliant deux sommets opposés en traversant le cube) vaut exactement .
2. Montrer que est irrationnel.

Voir solutions [8.6.2](#sec:orgcc2452d)

### Généralisation

Montrons par l’absurde que si est un nombre premier alors est irrationnel. avec . Il vient or et n’ont pas de diviseur commun donc divise et donc . Ainsi avec . Par conséquent et d’autre part d’où . Puisque et n’ont pas de diviseur commun ne divise pas . Donc c’est forcément d’où la contradiction.

En fait, que l’on choisisse ou comme diviseur de dans les deux cas on aboutit à une contradiction.

De plus, il y a un passage subtil ici, c’est parce que est premier (comme 2 et 3) que divise entraîne divise . Prenez le nombre 4, il divise mais ne divise pas . Ou encore 9 divise mais ne divise pas 6. Donc la primalité de est importante sinon la démonstration est invalide.

### Exercice

1. Trouver 5 contre-exemples de divise mais ne divise pas .
2. Montrer que est irrationnel.

Voir solutions [8.6.3](#sec:org0d068b0)

### Probablement irrationnel

Aussi surprenant que cela puisse paraître il y a infiniment plus de nombres irrationnels que de rationnels. Si vous lanciez un couteau au hasard sur une planche en bois graduée… vous tomberiez probablement sur un nombre irrationnel. Malheureusement les mathématiques nécessaires pour montrer ce résultat dépasse largement le cadre de cet ouvrage et du lycée en général. Il faudrait invoquer des arguments d’analyse topologique à propos de la notion de densité. Tout ça pour vous dire que l’aventure avec les nombres réels (et les nombres en général) continue bien au-delà des limites de ce livres. Et c’est avec regret que je suis contraint de vous laissez sans preuve pour l’irrationalité de . Voyez-vous, les arguments requis font appels aux concepts de dérivée (que vous verrez en première), d’intégrale (que vous verrez en terminale) et d’autres plus sophistiqués qui interviendront dans les mathématiques du supérieur.

Espérons tout de même que la lecture de ce livre vous aura fait découvrir la richesse des nombres réels. Si vous avez la moindre suggestion d’amélioration merci de m’écrire à l’adresse indiquée dans le formulaire avec pour **objet Amazon Manipuler les réels**.

# Capacités attendues

english To me, mathematics, computer science, and the arts are insanely related. They’re all creative expressions.  
[Sebastian Thrun](https://en.wikipedia.org/wiki/Sebastian_Thrun)

## Rappels concernant les capacités attendues

french Il n’y a guère de discipline plus mal connue et plus mal décrite que la mathématique. Combien pourraient répondre à la question : les mathématiques ça parle de quoi ? ça porte sur quoi ? ça propose quel but ?  
[Michel Demazure](https://fr.wikipedia.org/wiki/Michel_Demazure)

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

* Associer à chaque point de la droite graduée un unique nombre réel et réciproquement.
* Représenter un intervalle de la droite numérique. Déterminer si un nombre réel appartient à un intervalle donné.
* Donner un encadrement, d’amplitude donnée, d’un nombre réel par des décimaux.
* Dans le cadre de la résolution de problèmes, arrondir en donnant le nombre de chiffres significatifs adapté à la situation étudiée.

Tout ça se trouve sur Google Colab de façon interactive.

## Capacité attendue numéro 1

french Un mathématicien est une personne capable de trouver des analogies entre les théorèmes ; un meilleur mathématicien peut voir des analogies entre les démonstrations. Les très bons mathématiciens sont ceux capables de déceler des analogies entre des théories. Mais on peut supposer que le mathématicien ultime est celui qui peut voir des analogies entres les analogies.  
[Stephan Banach](https://fr.wikipedia.org/wiki/Stefan_Banach)

Cette partie développe la rubrique intitulée : " : Associer à chaque point de la droite graduée un unique nombre réel et réciproquement." dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Code pour construire les points

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code pour construire les points :

import matplotlib.pyplot as plt  
import numpy as np  
  
c\_1 = "Associer à chaque point de la droite graduée "  
c\_1 += "un unique nombre réel et réciproquement."  
print(c\_1)  
numbers = [-2, -0.5, 0, 1, np.sqrt(2), np.pi]  
labels = ['P\_0', 'P\_1', 'P\_2', 'P\_3', 'P\_4', 'P\_5']  
  
plt.figure(figsize=(10, 1))  
  
# Ploter une ligne horizontale  
plt.plot([min(numbers), max(numbers)], [0, 0], 'k')  
  
# Placer les nombres sur la ligne graduée  
plt.scatter(numbers, np.zeros\_like(numbers), color='red')  
  
# Ajouter des étiquettes aux nombres  
for i, label in enumerate(labels):  
 plt.text(numbers[i], 0.02, label, ha='center')  
  
# Masquer les axes  
plt.axis('off')  
  
plt.show()

### Exercice

Voici le code pour lancer l’exercice :

from math import pi  
  
nombres = {2\*\*0.5, -2, -0.5, 1, pi, 0}  
points = {'P\_0': 0, 'P\_1': 0, 'P\_2': 0, 'P\_3': 0, 'P\_4': 0, 'P\_5': 0}  
print("Voici la liste des points dont tu vas déterminer les abscisses")  
print(points)  
  
for nb in nombres:  
 p = input(f"{round(nb, 4)} correspond à l'abscisse de quel point ? ")  
 points[p] = round(nb, 4)

Voir solution [9.1.1](#sec:org5999ed2)

## Capacité attendue numéro 2

french Le but de la rigueur mathématique est de sanctionner et de légitimiser les conquêtes de l’intuition, et il n’y a jamais eu d’autre raison de la développer.  
[Jacques Hadamard](https://fr.wikipedia.org/wiki/Jacques_Hadamard)

Cette partie développe rubrique intulée : " : Réprésenter un intervalle de la droite numérique. Détrminer si un nombre réel appartient à un intervalle donné." dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Placement des abscisses sur la droite réelle

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code :

import matplotlib.pyplot as plt  
import numpy as np  
  
c\_2 = "Représenter un intervalle de la droite numérique.\n"  
c\_2 += "Déterminer si un nombre réel appartient à un intervalle donné."  
print(c\_2)  
numbers = [1 + i/10 for i in range(11)]  
labels = [str(num) for num in numbers]  
  
plt.figure(figsize=(10, 1))  
  
# Ploter une ligne horizontale  
plt.plot([min(numbers), max(numbers)], [0, 0], 'k')  
  
# Placer les nombres sur la ligne graduée  
plt.scatter(numbers, np.zeros\_like(numbers), color='red')  
  
# Ajouter des étiquettes aux nombres  
for i, label in enumerate(labels):  
 plt.text(numbers[i], 0.02, label, ha='center')  
  
# Masquer les axes  
plt.axis('off')  
  
plt.show()

### Exercice

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code pour générer l’exercice :

num\_to\_square = [1 + i/10 for i in range(5)]  
list\_of\_intervals = []  
for n2s in num\_to\_square:  
 print(f"Dans quel intervalle appartient {n2s}\*\*2 ?")  
 interval = []  
 for borne in ("inf", "sup"):  
 msg = f"Borne {borne}érieure au dixième = "  
 interval.append(float(input(msg)))  
 list\_of\_intervals.append(interval)  
  
print("Voici la liste de tes réponses")  
print(list\_of\_intervals)  
print("Voici tes correspondances")  
for i in range(len(num\_to\_square)):  
 n2s, interval = num\_to\_square[i], list\_of\_intervals[i]  
 msg = f"{n2s}\*\*2 = {round(n2s\*\*2, 2)} appartient à "  
 msg += f"l'intervalle {interval}"  
 print(msg)

Voir solution [9.2.1](#sec:orgf6bdc67)

## Capacité attendue numéro 3

french La physique est écrite dans cet immense livre qui continuellement se tient ouvert devant nos yeux (je veux dire l’univers), mais elle ne peut se comprendre si on ne s’exerce d’abord à comprendre la langue et à connaître les caractères avec lesquels elle est écrite. Elle est écrite en langue mathématique, et les caractères sont les triangles, les cercles et autres figures géométriques, sans lesquelles il est humainement impossible d’en comprendre le moindre mot, sans lesquelles on s’égare vainement dans un labyrinthe obscur.  
[Galilée](https://fr.wikipedia.org/wiki/Galil%C3%A9e_(savant))

Cette partie développe la rubrique intitulée : " : Donner un encadrement, d’amplitude donnée, d’un nombre réel par des décimaux." dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Exercice

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code pour générer l’exercice :

import math  
  
some\_primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 31, 37, 41]  
prime\_roots = [math.sqrt(sp) for sp in some\_primes]  
for i in range(8, len(prime\_roots)):  
 inf, sup = default(prime\_roots[i], i), excess(prime\_roots[i], i)  
 sqrt\_unicode = "\u221A"  
 print(f"Un encadrement de {sqrt\_unicode}{some\_primes[i]} ~", end=" ")  
 print(f"{round(prime\_roots[i], i)} d'amplitude 10\*\*(-{i}) est :", end=" ")  
 print(f"{inf} <= {sqrt\_unicode}{some\_primes[i]} <= {sup}")  
 print(f"Dit autrement, {sqrt\_unicode}{some\_primes[i]} ~", end=" ")  
 print(f"{round(prime\_roots[i], i)}", end=" ")  
 print(f"appartient à l'intervalle [{inf}; {sup}]")  
  
def exo\_amplitude():  
 score, method, born = 0, [default, excess], ["inf", "sup"]  
 sqrt\_unicode = "\u221A"  
 motivation = "À toi de jouer !"  
 print(motivation)  
 instr = "Donne un encadrement à 10\*\*"  
 for i in range(5):  
 print(f"{instr}(-{some\_primes[i]}) près de {sqrt\_unicode}{some\_primes[i]}")  
 for j in range(2):  
 t = method[j](prime\_roots[i], some\_primes[i])  
 r = float(input(f"Borne {born[j]}érieure à {some\_primes[i]} chiffres = "))  
 if r == t:  
 print("Bien joué !")  
 score += 0.2  
 else:  
 print(f"La bonne réponse était {t}")

Voir solutions [9.3.1](#sec:org938bb3b)

### Exercice

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code pour générer l’exercice :

def approx\_nums(some\_numbers, decimals):  
 score, method, born = 0, [default, excess], ["inf", "sup"]  
 motivation = "À toi de jouer !"  
 print(motivation)  
 instr = "Donne un encadrement à 10\*\*"  
 for num in some\_numbers:  
 boundaries = [default(num, decimals), excess(num, decimals)]  
 print(f"{instr}(-{decimals}) près de {num}")  
 for j in range(2):  
 t = boundaries[j]  
 r = float(input(f"Borne {born[j]}érieure à {decimals} chiffres = "))  
 if r == t:  
 print("Bien joué !")  
 score += 1  
 else:  
 print(f"La bonne réponse était {t}")  
 x\_pts, inf, sup, d = num, boundaries[0], boundaries[1], decimals  
 step = 10\*\*(-decimals)  
 zoom(x\_pts, inf, sup, step, decimals, size=(9.75, 1.25))  
 print(f"Score final = {score \* 10}%")  
  
  
phi = (1 + 5\*\*0.5) / 2 # nombre d'or  
phi\_unicode, sqrt\_unicode = "\u03A6", "\u221A"  
print(f"On appelle nombre d'or et on le note {phi\_unicode} = (1 + {sqrt\_unicode}5) / 2 ~ {phi}")  
phi2, tau = phi\*\*2, 2 \* math.pi  
print("Vous pouvez remarquer que {phi\_unicode}\*\*2 = 1 + {phi\_unicode} (vérifiez-le par le calcul !)")  
tau\_unicode, pi\_unicode = "\u03A4", "\u03C0"  
print(f"On appelle {tau\_unicode} = 2{pi\_unicode} ~ {tau}")  
print(f"On ne présente plus {pi\_unicode} ~ {math.pi}")  
print(f"On appelle constante d'Euler e ~ {math.e}")  
some\_numbers = [phi, phi2, math.e, math.pi, tau]  
decimals = random.randint(2, 5)  
approx\_nums(some\_numbers, decimals)

Voir solutions [9.3.2](#sec:org39262d2)

## Capacité attendue numéro 4

french Tout mathématicien digne de ce nom a connu, parfois seulement à de rares intervalles, ces états d’exaltation lucide où les pensées s’enchaînent comme par miracle, et où l’inconscient (quel que soit le sens qu’on attache à ce mot) paraît aussi avoir sa part. […] A la différence du plaisir sexuel, celui-là peut durer plusieurs heures, voire plusieurs jours ; qui l’a connu en désire le renouvellement mais est impuissant à le provoquer, sinon tout au plus par un travail opiniâtre dont il apparaît alors comme la récompense ; il est vrai que le plaisir qu’on en ressent est sans rapport avec la valeur des découvertes auxquelles il s’associe.  
[André Weil](https://fr.wikipedia.org/wiki/Andr%C3%A9_Weil)

Cette partie développe la rubrique intitulée : " : Dans le cadre de la résolution de problèmes, arrondir en donnant le nombre de chiffres significatifs adapté à la situation étudiée." dans le [programme officiel](https://eduscol.education.fr/document/24553/download).

### Exemple

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code pour générer l’exemple :

c = 299\_792\_458  
print(f"La vitesse de la lumière dans le vide est de {c} mètres par seconde.")  
e\_s = "L'écriture scientifique d'un nombre n = a x 10\*\*k"  
e\_s += "\na appartient à l'intervalle [1; 10[ donc a < 10"  
e\_s += "\nk appartient à l'ensemble des entiers relatifs"  
a, k = 2.99792458, 8  
print(f"Son écriture scientifique est c = {a} x 10\*\*{k} m/s")  
print("Pour passer des mètres aux kilomètres il faut diviser par 1 000")  
print("Cela revient à multiplier par 10\*\*(-3)")  
print(f"D'où c = {a} x 10\*\*{k-3} km/s")  
print("Dans 1 heure il y 60 minutes et dans 1 minute il y a 60 secondes")  
print("Donc dans 1 heure il y 60 x 60 = 3600 secondes")  
print("Donc il faut multiplier par 3600")  
print("Mais dans ce cas, autant faire d'une pierre, deux coups")  
print("Multiplier par 3600 et diviser par 1 000 revient à multiplier par 3.6")  
a \*= 3.6  
if a > 10:  
 a /= 10  
 k -= 1  
print(f"D'où c = {a} x 10\*\*{k} km/h")

### Exercice

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code pour générer l’exercice :

print("À toi de jouer !")  
mile = 1609.344  
print(f"Sachant que 1 mile = {mile} mètres")  
print("Calcule la vitesse de la lumière en miles par heure.")  
msg = "Prend le temps de poser les calculs sur un papier avant de répondre"  
print(msg.upper())  
user\_rep = float(input("Vitesse en miles par heure = "))  
print(f"Selon toi c = {user\_rep} mph (miles per hour)")

# Démonstrations

english The pure mathematician, like the musician, is a free creator of his world of ordered beauty.  
[Bertrand Russell](https://amzn.to/3ONEWLI)

## Rappels des éléments du programme officiel

french Une théorie mathématique ne peut être considérée comme complète tant que l’on ne l’a clarifiée au point de pouvoir l’expliquer au premier homme que l’on croise dans la rue.  
[David Hilbert](https://fr.wikipedia.org/wiki/David_Hilbert)

Les deux démonstrations au programme concernent :

* Le nombre rationnel n’est pas décimal.
* Le nombre réel est irrationnel.

Ces deux démonstrations sont traitées sur Google Colab.

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

## Démonstration numéro 1

french Quel que soit le temps passé à faire des mathématiques, ce n’est jamais du temps perdu.  
[Cédric Villani](https://fr.wikipedia.org/wiki/C%C3%A9dric_Villani)

Cette partie développe la rubrique intitulée : "Le nombre rationnel n’est pas décimal."

### À la main

Mais proposons une démonstration utilisant l’astuce des nombres à décimales périodiques vue dans [2.6.3](#sec:orgb56332d).

Posons où 3 se répète à l’infini. Multiplions par 10 puis retranchons . Ainsi donc . CQFD

Pour consulter Google Colab ou les démonstrations précédentes.

En espérant que la lecture de ce livre vous aura satisfait et fait découvrir la richesse des nombres réels. Si vous avez la moindre suggestion d’amélioration merci de m’écrire à l’adresse indiquée dans le formulaire avec pour **objet Amazon Manipuler les réels**.

### Avec Python

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

Voici le code pour générer les 3 démonstrations :

def read\_text(txt):  
 for t in txt:  
 input(f"{t}\nAppuyez sur une touche pour lire la suite...")  
  
  
intro = [  
 "Le nombre 1/3 est-il décimal ?",  
 "Pour que 1/3 soit décimal",  
 "il faudrait trouver a et n tels que 1/3 = a/10\*\*n.",  
 "En faisant un produit en croix cela revient à 10\*\*n = 3\*a.",  
 "Maintenant le problème devient beaucoup plus simple.",  
 "Il suffit de prouver qu'aucune puissance de 10 n'est mulitple de 3."  
 ]  
  
read\_text(txt=intro)  
  
multiple\_of\_3 = [3\*k for k in range(21)]  
print("Voici les multiples de 0 à 60")  
print(multiple\_of\_3)  
rem = input("Que remarquez-vous ? ")  
more = input("Vous en voulez plus ? 0) Non 1) Oui\n")  
if more == "1":  
 for k in range(21, 50):  
 print(3\*k, end=", ")  
 print()  
  
div = input("Quels sont les diviseurs (entiers positifs) de 10 ? ")  
div = div.split(",")  
div = [int(d) for d in div]  
print(f"Selon vous, les diviseurs de 10 sont {div}")  
prime\_proof = [  
 "Les diviseurs de 10 sont 2 et 5",  
 "Donc les diviseurs de n'importe quelle puissance de 10",  
 "sont de la forme 2\*\*p x 5\*\*q",  
 "avec p et q tels que p + q <= n",  
 "Aucune puissance de 2 ne divise 3",  
 "car 2 et 3 sont premiers entre eux (ils n'ont pas de diviseurs communs)",  
 "il en va de même pour 5 et 3",  
 "donc il est impossible d'obtenir 10\*\*n = 3\*a",  
 "une autre façon de faire consiste à poser la division de 10 par 3",  
 "vous constaterez qu'il y aura toujours un reste non nul",  
 "et vous pourriez faire de même à l'infini avec toute puissance de 10"  
]  
  
read\_text(txt=prime\_proof)  
  
div\_proof = [f"10\*\*{n} = 3 x {(10\*\*n)//3} + {(10\*\*n) % 3}" for n in range(10)]  
read\_text(txt=div\_proof)  
  
decimal\_proof = [f"1/3 ~ {round(1/3, i)} à {i} décimale(s)" for i in range(10)]  
read\_text(txt=decimal\_proof)  
  
magic\_proof = [  
 "Supposons qu'il existe un nombre noté x = 0.333333...",  
 "Puisqu'il y a une infinité de 3 après la virgule alors",  
 "10x = 3.333333... avec une infinité de 3 après la virgule",  
 "10x - x = 3 et donc 9x = 3 d'où x = 3/9 = 1/3",  
 "CQFD"  
]  
read\_text(txt=magic\_proof)

# Exemple d’algorithme

english Just because we can’t find a solution, it doesn’t mean there isn’t one.  
[Andrew Wiles](https://en.wikipedia.org/wiki/Andrew_Wiles)

## Rappel des éléments du programme officiel

french Monsieur Fourier avait l’opinion que le but principal des mathématiques était l’utilité publique et l’explication des phénomènes naturels. Un philosophe tel que lui aurait dû savoir que le but unique de la Science, c’est l’honneur de l’esprit humain et que, sous ce titre, une question de nombres vaut bien une question de système du monde.  
[Charles Gustave Jacob Jacobi](https://fr.wikipedia.org/wiki/Charles_Gustave_Jacob_Jacobi)

Le programme officiel propose un seul exemple d’algorithme :

* Déterminer par balayage un encadrement de d’amplitude inférieure ou égale à .

## Petites explications

french Les mathématiques ont des inventions très subtiles et qui peuvent beaucoup servir, tant à contenter les curieux qu’à faciliter tous les arts et à diminuer le travail des hommes.  
[René Descartes](https://fr.wikipedia.org/wiki/Ren%C3%A9_Descartes)

On cherche des réels et tels que avec .

Comme alors on va choisir .

De sorte que .

## Le code

french Si vous touchez aux maths, vous ne devez être ni pressés, ni cupides, fussiez-vous roi ou reine.  
[Euclide](https://amzn.to/3KU1CZE)

Voici le code que vous pouvez retrouver sur Google Colab :

def balayage(n):  
 """  
 Cette fonction renvoie les bornes d'un encadrement par balayage  
 d'amplitude inférieure ou égale à 10\*\*(-n)  
 """  
 a = 1  
 b = a + 10\*\*(-n)  
 while b\*\*2 < 2:  
 a += 10\*\*(-n)  
 b = a + 10\*\*(-n)  
 return round(a, n), round(b, n)  
  
sqrt\_unicode = "\u221A"  
# Tests : attention parce qu'au-delà de n = 7 ça devient très long   
for n in range(1, 7):  
 infimum, supremum = balayage(n)  
 ampli = 10\*\*(-n)  
 print(f"{infimum} < {sqrt\_unicode}2 < {supremum} est un encadrement d'amplitude {ampli}")

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

En espérant que la lecture de ce livre vous aura satisfait et fait découvrir la richesse des nombres réels. Si vous avez la moindre suggestion d’amélioration merci de m’écrire à l’adresse indiquée dans le formulaire avec pour **objet Amazon Manipuler les réels**.

# Approfondissements possibles

english Millions saw the apple fall, but Newton asked why.  
[Bernard Baruch](https://en.wikipedia.org/wiki/Bernard_Baruch)

## Rappel des éléments du programme officiel

french Les mathématiques sont une gymnastique de l’esprit et une préparation à la philosophie.  
[Isocrate](https://fr.wikipedia.org/wiki/Isocrate)

D’apprès le programme officiel les approfondissements possibles sont :

* Développement décimal illimité d’un nombre réel.
* Observation, sur des exemples, de la périodicité du développement décimal de nombres rationnels, du gait qu’un développement décimal périodique correspond à un rationnel.

## Petit bilan

french Il y a une joie réelle à faire des mathématiques, à apprendre de nouvelles méthodes de pensée qui expliquent, organisent et simplifient. On peut ressentir cette joie en découvrant de nouvelles mathématiques, (…) ou en trouvant une nouvelle façon d’expliquer (…) une structure mathématique ancienne.  
[William P. Thurston](https://en.wikipedia.org/wiki/William_Thurston)

Pour le deuxième point on a déjà vu beaucoup d’exemple (cf la petite astuce vue dans [2.6.3](#sec:orgb56332d)).

Mais rien que pour le plaisir en voici encore un. Soit . Ici la période est 2023. Donc on va multiplier par puis retrancher d’où qui entraîne

## Et pour quelques exemples de plus

english Pure mathematics is, in its way, the poetry of logical ideas.  
[Albert Einstein](https://en.wikipedia.org/wiki/Albert_Einstein)

Voici quelques développement décimaux illimités de nombres réels :

En espérant que la lecture de ce livre vous aura satisfait et fait découvrir la richesse des nombres réels. Si vous avez la moindre suggestion d’amélioration merci de m’écrire à l’adresse indiquée dans le formulaire avec pour **objet Amazon Manipuler les réels**.

# Après ce livre

english I’ve always been interested in using mathematics to make the world work better.  
[Alvin E. Roth](https://en.wikipedia.org/wiki/Alvin_E._Roth)

## Conclusion

english Without mathematics, there’s nothing you can do. Everything around you is mathematics. Everything around you is numbers.  
[Shakuntala Devi](https://en.wikipedia.org/wiki/Shakuntala_Devi)

Bravo à vous si vous avez lu jusqu’au bout.

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

Sur Google Colab vous pourrez expérimentez avec des exercices nouveaux à chaque exécution du code. Cela vous permettra de vous entraîner avec des valeurs différentes à chaque fois. C’est beaucoup plus interactifs que tous les exemples de ce livre.

Le but de ce livre est de vous donner un aperçu du monde merveilleux des nombres réels. Mais il y a encore beaucoup de choses à découvrir que je n’ai même pas pu mentionner ici.

Je vous invite à mettre le maximum d’étoiles sur Amazon et à laisser un commentaire pour aider les gens qui cherchent des ressources utiles.

En espérant que la lecture de ce livre vous aura satisfait et fait découvrir la richesse des nombres réels. Si vous avez la moindre suggestion d’amélioration merci de m’écrire à l’adresse indiquée dans le formulaire avec pour **objet Amazon Manipuler les réels**.

# Solutions des exercices de la section : [2](#sec:orgda3f058)

english The only way to learn mathematics is to do mathematics.  
[Paul Richard Halmos](https://en.wikipedia.org/wiki/Paul_Halmos)

## Solutions des exercices de la section : [2.1](#sec:org9ec1b06)

english We will always have STEM with us. Some things will drop out of the public eye and go away, but there will always be science, engineering, and technology. And there will always, always be mathematics.  
[Creola Katherine Coleman](https://en.wikipedia.org/wiki/Katherine_Johnson)

### Solution de l’exercice [2.1.7](#sec:org09bf59f)

On reprend le même schéma que dans [2.1.6](#sec:org94fd512)

$$\begin{aligned}
A &=\,"x < -1"\\
B &=\,"-3x > 3"\\
P &=\,A\Rightarrow B\\
\neg B &=\, "-3x \leq 3"\\
\neg A &=\, "x \geq -1"\\
CP(P) &=\,(\neg B)\Rightarrow(\neg A)
\end{aligned}$$

### Solution Python de l’exercice [2.1.8](#sec:org1d3ea27)

Voir énoncé [2.1.8](#sec:org1d3ea27)

1. Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

* Voici le code de la fonction imply
* def imply(A, B):  
   """  
   Cette fonction prend 2 paramètres A et B en entrée   
   et renvoie la phrase logique A implique B  
   """  
   return f"({A}) implique ({B})"  
    
  # Tests  
  A = ["x > 2", "x < -1"]  
  B = ["2\*x > 4", "-3\*x > 3"]   
  for i in range(len(A)):  
   print(imply(A[i], B[i]))  
    
  # Sorties  
  #(x > 2) implique (2\*x > 4)  
  #(x < -1) implique (-3\*x > 3)

1. 1. Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

* Voici le code de la fonction contrapose
* def contrapose(A, B):  
   """  
   Cette fonction prend 2 paramètres A et B en entrée   
   et renvoie la phrase logique non B implique non A  
   """  
   return imply("non " + B, "non " + A)  
    
  # Tests  
  A = ["x > 2", "x < -1"]  
  B = ["2\*x > 4", "-3\*x > 3"]   
  for i in range(len(A)):  
   print(contrapose(A[i], B[i]))  
    
  # Sorties  
  #(non 2\*x > 4) implique (non x > 2)  
  #(non -3\*x > 3) implique (non x < -1)

## Solutions des exercices de la section : [2.2](#sec:org74265cf)

english Mathematics as an expression of the human mind reflects the active will, the contemplative reason, and the desire for aesthetic perfection. Its basic elements are logic and intuition, analysis and construction, generality and individuality.  
[Richard Courant](https://en.wikipedia.org/wiki/Richard_Courant)

### Solutions de l’exercice [2.2.2](#sec:orgb387ab9)

Voir énoncé [2.2.2](#sec:orgb387ab9)

* qui a pour amplitude
* qui a pour amplitude
* qui a pour amplitude
* On peut classer ces intervalles :
* qui a pour amplitude
* qui a pour amplitude
* qui a pour amplitude
* On peut classer ces intervalles :
* qui pour amplitude
* qui pour amplitude
* qui pour amplitude
* On peut classer ces intervalles :
* qui a pour amplitude
* qui a pour amplitude
* qui a pour amplitude
* On peut classer ces intervalles :
* qui a pour amplitude
* qui a pour amplitude
* qui a pour amplitude
* On peut classer ces intervalles :

### Solutions de l’exercice [2.2.3](#sec:orgc61202f)

Voir énoncé [2.2.3](#sec:orgc61202f)

1. Voici les 16 premiers termes de cette suite de nombres :
2. Pour calculer un nouveau terme on ajoute les deux termes précédents

* Par exemple ou encore .

1. Les 11 premiers termes de la suites de Fibonacci ont permis d’approcher le nombre d’or à 3 chiffres après la virugle. Poursuivons avec ce que nous avons déjà calculer à la question 2.

### Solution du petit jeu [2.2.4](#sec:org5375956)

Voir énoncé [2.2.4](#sec:org5375956)

1. Partie 1

* Voilà pour la première partie de la question
  + en effet
  + en effet
  + en effet
  + en effet
  + en effet

1. Partie 2

* Maintenant occupons-nous de la seconde partie.
  + donc
  + d’où
  + donc
  + d’où
  + donc
  + d’où
  + donc
  + d’où
  + donc
  + d’où

## Solutions des exercices de la section : [2.3](#sec:org112f093)

english Mathematics is the music of reason.  
[James Joseph Sylvester](https://en.wikipedia.org/wiki/James_Joseph_Sylvester)

### Solution de l’exercice [2.3.2](#sec:org8920b07)

Pour voir l’énoncé de l’exercice [2.3.2](#sec:org8920b07)

1. Calcul de la distance entre et à 3 chiffres après la virgule.
2. Calcul de la distance et à 3 chiffres près.
3. Calcul de la distance et à 6 chiffres.
4. Calcul de la distance entre et à 2 chiffres.
5. Calcul de la distance entre et à 5 chiffres.

### Solution de l’exercice [2.3.3](#sec:org275d8c1)

Voir l’énoncé [2.3.3](#sec:org275d8c1)

1. Calcul des nombres à une distance 1 de 1. Là c’est facile on peut le faire de tête c’est et .
2. Calcul des nombres à une distance de . Là on peut poser les calculs : et .
3. Calcul des nombres à une distance de . Mettons les fractions au même dénominateur. et .
4. Calcul des nombres à une distance de 1. et .
5. Calcul des nombres à une distance de . Ici il faut faire appel à une astuce classique pour les racines carrées, la multiplication par la quantité conjuguée (cf l’identité remarquable qui permet d’enlever les racines).
6. Calcul des nombres à une distance de 1.
7. Calcul des nombres à une distance de .

* et .

## Solutions des exercices de la section : [2.4](#sec:org8448569)

english There should be no such thing as boring mathematics.  
[Edsger W. Dijkstra](https://en.wikipedia.org/wiki/Edsger_W._Dijkstra)

### Solution de l’exercice [2.4.2](#sec:orga80e08b)

Voir énoncé [2.4.2](#sec:orga80e08b)

1. Calculs avec
2. Calculs avec
3. Calculs avec
4. Calculs avec

### Solution de l’exercice [2.4.5](#sec:org48657a1)

Voir énoncé [2.4.5](#sec:org48657a1)

1. Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

* def borne\_inf\_sup(a, r):  
   if r < 0: return False  
   else: return a - r, a + r  
    
  # Test  
  a, r = 2, -2\*\*0.5  
  print(borne\_inf\_sup(a, r))  
  # Sortie  
  #False  
  a, r = 2, 2\*\*0.5  
  print(borne\_inf\_sup(a, r))  
  # Sortie  
  #(0.5857864376269049, 3.414213562373095)

1. Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

* def borne\_inf\_sup(a, r, precision):  
   if r < 0: return False  
   else: return round(a - r, precision), round(a + r, precision)  
    
  # Tests  
  a, r = 2, 2\*\*0.5  
  for precision in range(1, 10):  
   print(borne\_inf\_sup(a, r, precision))  
    
  # Sorties  
  #(0.6, 3.4)  
  #(0.59, 3.41)  
  #(0.586, 3.414)  
  #(0.5858, 3.4142)  
  #(0.58579, 3.41421)  
  #(0.585786, 3.414214)  
  #(0.5857864, 3.4142136)  
  #(0.58578644, 3.41421356)  
  #(0.585786438, 3.414213562)

1. Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

* def distance(a, b):  
   if a < b: return b - a  
   else: return a - b  
    
  # Tests  
  a, b = 1, 2  
  print(f"d({a}, {b}) = {distance(a, b)}")  
  a, b = 2\*\*0.5, 1  
  print(f"d({a}, {b}) = {distance(a, b)}")  
  #d(1, 2) = 1  
  #d(1.4142135623730951, 1) = 0.41421356237309515

## Solutions des exercices de la section : [2.5](#sec:org38c43bc)

english ‘Obvious’ is the most dangerous word in mathematics.  
[Eric Temple Bell](https://en.wikipedia.org/wiki/Eric_Temple_Bell)

### Solution avec Python de l’exercice [2.5.2](#sec:org23137bc)

Voir énoncé [2.5.2](#sec:org23137bc)

Si vous l’avez fait sur papier vous avez dû utiliser une machine pour calculer les décimales de , de et des racines carrées. Exactement comme je l’avais montré pour . Ici la façon la plus intelligente de procéder consiste à utiliser Python.

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

from math import ceil, pi, trunc  
  
def decimal\_frame(real, nb\_digits):  
 """  
 Cette fonction prend en entrées :   
 real: un nombre réel à encadrer  
 nb\_digits: le nombre de décimales de l'encadrement  
 Elle renvoie en sortie les bornes inférieures et supérieures   
 de cet encadrement décimal.  
 """  
 # on décale la virgule de nb\_digits vers la droite  
 real\_power\_nb\_digits = real \* 10\*\*nb\_digits   
 # on coupe  
 infimum = round(trunc(real\_power\_nb\_digits) \* 10\*\*(-nb\_digits), nb\_digits)  
 supremum = round(ceil(real\_power\_nb\_digits) \* 10\*\*(-nb\_digits), nb\_digits)  
 return infimum, supremum  
  
# Tests  
tau, phi = 2\*pi, (1+5\*\*0.5)/2  
sqrts = [x\*\*0.5 for x in [2, 3, 5, 7]]  
reals = [tau, phi] + sqrts  
for real in reals:  
 print(f"Encadrement du nombre réel {real} à :")  
 for nb\_digits in range(1, 6):   
 print(f"10\*\*(-{nb\_digits}) près :", end=" ")  
 print(decimal\_frame(real, nb\_digits))

## Solutions des exercices de la section : [2.6](#sec:orgdee9ad7)

english Mathematics are the result of mysterious powers which no one understands, and which the unconscious recognition of beauty must play an important part. Out of an infinity of designs a mathematician chooses one pattern for beauty’s sake and pulls it down to earth.  
[Marston Morse](https://en.wikipedia.org/wiki/Marston_Morse)

### Solution de l’exercice [2.6.2](#sec:orgca24f0a)

Voir énoncé [2.6.2](#sec:orgca24f0a)

Trouvons l’écriture fractionnaire des nombres suivants :

1. avec des à l’infini.

* Vous connaissez maintenant la fraction de James Bond !

1. avec la période 076923 qui se répète à l’infini.

* Le nombre 13 propose une période à 6 chiffres comme le nombre 7.

1. avec la période 0588235294117647 qui se répète à l’infini.

* Le nombre 17 cachait bien son jeu avec sa période à 16 chiffres !

### Solution de l’exercice [2.6.5](#sec:org1191128)

Voir énoncé [2.6.5](#sec:org1191128)

1. Chacune des 6 faces du cube d’arête 1 est un carré de côté 1 donc de diagonale (voir les démonstrations précédentes). Si on considère un triangle formé par une diagonale de la face du bas et d’une arête verticale alors l’hypoténuse sera la diagonale cherchée. Nommons ABC ce triangle avec (l’arête verticale), (la diagonale de la face du bas) et l’hypoténuse. Par Pythagore d’où CQFD

* [Cube en 3D](https://drive.google.com/file/d/1IRA1h0fzjiR9aVBPM4pPzKq3_egsHwVt/view?usp=sharing)

1. Montrons par l’absurde que est irrationnel. Soit et deux entiers tels que et . Alors . Puisque et n’ont pas de diviseur commun il en va de même pour leurs carrés. Donc 3 est un diviseur de puisqu’il divise son carré. Ainsi avec . On peut alors établir que ce qui implique . Comme et n’ont pas de diviseur commun alors ne divise pas donc nécessairement c’est 3. Problème, si 3 divise alors il divise aussi ce qui contredit . CQFD

### Solution de l’exercice [2.6.7](#sec:org0527421)

Voir énoncé [2.6.7](#sec:org0527421)

1. Le nombre 4 divise le carré de 10 mais pas 10. Le nombre 8 divise le carré de 4 mais pas 4. Le nombre 12 divise le carré de 6 mais pas 6. Le nombre 16 divise le carré de 8 mais pas 8. Le nombre 9 divise le carré de 12 mais pas 12.
2. Supposons par l’absurde que le nombre d’or soit rationnel. Alors avec . Or donc mais alors conduit à soit ce qui est absurde car ne divise pas donc pas son carré non plus et d’autre par ne divise pas non plus. CQFD

# Solutions des exercices de la section : [3](#sec:org5737607)

english You don’t have to be a mathematician to have a feel for numbers.  
[John Forbes Nash Jr.](https://en.wikipedia.org/wiki/John_Forbes_Nash_Jr.)

## Solutions des exercices de la section : [3.2](#sec:org5bec0e4)

english It is impossible to be a mathematician without being a poet in soul.  
[Sofya Kovalveskaya](https://en.wikipedia.org/wiki/Sofya_Kovalevskaya)

### Solution de l’exercice [3.2.2](#sec:orgd573e54)

# Solution  
import matplotlib.pyplot as plt  
import numpy as np  
  
numbers = [-2, -0.5, 0, 1, np.sqrt(2), np.pi]  
pi\_unicode, sqrt\_unicode = "\u03C0", "\u221A"  
labels = ['P\_0 (-2)', 'P\_1 (-0.5)', 'P\_2 (0)', 'P\_3 (1)', f'P\_4 ({sqrt\_unicode}2', f'P\_5 ({pi\_unicode})']  
  
plt.figure(figsize=(10, 1))  
  
# Ploter une ligne horizontale  
plt.plot([min(numbers), max(numbers)], [0, 0], 'k')  
  
# Placer les nombres sur la ligne graduée  
plt.scatter(numbers, np.zeros\_like(numbers), color='red')  
  
# Ajouter des étiquettes aux nombres  
for i, label in enumerate(labels):  
 plt.text(numbers[i], 0.02, label, ha='center')  
  
# Masquer les axes  
plt.axis('off')  
  
plt.show()

## Solutions des exercices de la section : [3.3](#sec:orga5f1d95)

english A mathematician who is not also something of a poet will never be a complete mathematician.  
[Karl Weierstrass](https://en.wikipedia.org/wiki/Karl_Weierstrass)

### Solution de l’exercice [3.3.2](#sec:org083f20c)

Voir énoncé [3.3.2](#sec:org083f20c)

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code pour générer les solutions de l’exercice [3.3.2](#sec:org083f20c) :

# Solution  
from math import floor, ceil  
  
  
def default(number, decimals):  
 """  
 Cette fonction prend 2 paramètres en entrées :  
 number: le nombre dont on cherche à faire l'arrondi par défaut  
 decimals: le nombre de chiffres après la virgule  
 Elle renvoie le nombre arrondi par défaut à decimals décimales  
 """  
 num = floor(number \* 10\*\*decimals)  
 den = 10\*\*decimals  
 return num / den  
  
def excess(number, decimals):  
 """  
 Cette fonction prend 2 paramètres en entrées :  
 number: le nombre dont on cherche à faire l'arrondi par excès  
 decimals: le nombre de chiffres après la virgule  
 Elle renvoie le nombre arrondi par excès à decimals décimales  
 """  
 if number == int(number):  
 num = ceil(number \* 10\*\*decimals + 1)  
 else: num = ceil(number \* 10\*\*decimals)  
 den = 10\*\*decimals  
 return num / den  
  
def get\_boundary(nums, method, decimals):  
 """  
 Cette fonction prend 3 paramètres en entrées :  
 nums: la liste de nombres dont on cherche la borne (inf ou sup)  
 method: la fonction utilisée (default ou excess)  
 decimals: le nombre de décimales  
 Elle renvoie la liste des bornes (que des infs ou que des sups)  
 """  
 bounds = [method(num, decimals) for num in nums]  
 return bounds  
  
  
num\_to\_square = [1 + i/10 for i in range(5)]  
squares = [round(n2s\*\*2, 2) for n2s in num\_to\_square]  
infs = get\_boundary(squares, default, 1)  
sups = get\_boundary(squares, excess, 1)  
list\_of\_intervals = [[infs[i], sups[i]] for i in range(len(infs))]  
  
nums = []  
for i in range(len(squares)):  
 nums.extend([infs[i], squares[i], sups[i]])  
 n2s, interval = num\_to\_square[i], list\_of\_intervals[i]  
 msg = f"{n2s}\*\*2 = {squares[i]} appartient à l'intervalle {interval}"  
 print(msg)  
  
plt.figure(figsize=(10, 1))  
  
# Tracer une ligne horizontale  
plt.plot([min(nums), max(nums)], [0, 0], 'k')  
  
# Placer les nombres sur la ligne graduée  
plt.scatter(squares, np.zeros\_like(squares), color='green', marker='x')  
plt.scatter(infs, np.zeros\_like(infs), color='red', marker='$[$')  
plt.scatter(sups, np.zeros\_like(sups), color='red', marker='$]$')  
  
infs\_labels = [str(inf) for inf in infs]  
sqrs\_labels = [str(sqr) for sqr in squares]  
sups\_labels = [str(sup) for sup in sups]  
# Ajouter des étiquettes aux nombres  
for i, inf\_label in enumerate(infs\_labels):  
 plt.text(infs[i], 0.02, inf\_label, ha='center')  
  
for i, sqr\_label in enumerate(sqrs\_labels):  
 plt.text(squares[i], -0.02, sqr\_label, ha='center', va='top')  
  
for i, sup\_label in enumerate(sups\_labels):  
 plt.text(sups[i], 0.02, sup\_label, ha='center')  
  
  
# Masquer les axes  
plt.axis('off')  
  
plt.show()

## Solutions des exercices de la section : [3.4](#sec:org53eb07f)

english In mathematics the art of proposing a question must be held of higher value than solving it.  
[Georg Cantor](https://en.wikipedia.org/wiki/Georg_Cantor)

### Solution de l’exercice [3.4.1](#sec:orgdb5a92b)

Voir énoncé [3.4.1](#sec:orgdb5a92b)

Les solutions sont générées automatiquement après la saisie utilisateur.

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

### Solution de l’exercice [3.4.2](#sec:org03939ae)

Voir énoncé [3.4.2](#sec:org03939ae)

Les solutions sont générées automatiquement après la saisie utilisateur.

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

## Solutions des exercices de la section : [3.5](#sec:orgc5ac9eb)

english It is clear that the chief end of mathematical study must be to make the students think.  
[John Wesley Young](https://en.wikipedia.org/wiki/John_Wesley_Young)

### Solution de l’exercice [3.5.2](#sec:orgf2086dc)

Voir énoncé [3.5.2](#sec:orgf2086dc)

Pour rappels tous les codes sont accessibles sur Google Colab à l’adresse communiquée dans la section concernée lorsque vous aurez rempli ce petit formulaire : <https://forms.gle/pT3GVr8PdNYCeVir7> (vous trouverez également les corrections de tous les exercices du livre).

C’est également sur Google Colab que vous pourrez tester tous les programmes qui vous permettent de pratiquer les capacités attendues de manière interactive.

Voici le code pour générer l’exercice :

# Solution  
print(f"Puisque 1 mile = {mile} alors cela fait {mile/1000} km")  
print(f"Ainsi 1 km = {1000/mile} miles")  
print(f"Il suffit donc de multiplier la vitesse en km/h par {1000/mile}")  
a \*= 1000/mile  
if a > 10:  
 a /= 10  
 k -= 1  
elif a < 1:  
 a \*= 10  
 k += 1  
print(f"D'où c = {a} x 10\*\*{k} mph")