

Résolution numérique d'équations

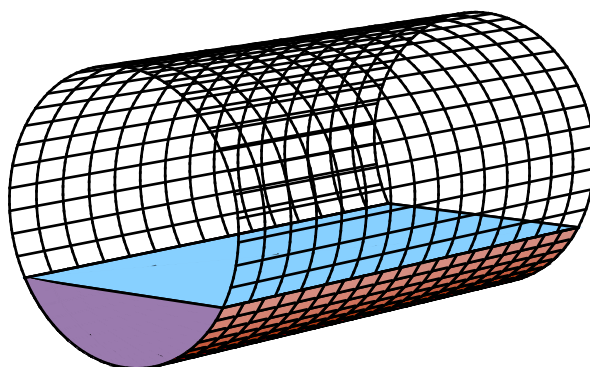


Table des matières

Introduction	2
1 Méthode graphique	5
2 Méthode de la bisection	10
3 Méthode regula falsi	14
4 Méthode du point fixe	18
5 Méthode de Newton	26
6 Résolution avec Mathematica	32
7 Éléments de programmation Mathematica	44
8 Exercices	48
9 Exercices de répétition	60
10 Réponses des exercices	61

Introduction

🌐 Remboursement d'une dette en annuités

Une somme de 10 000 francs est empruntée à une banque. Afin de rembourser cet emprunt, la banque demande de verser 8 annuités¹ de 2000 francs.

Déterminer l'équation que doit satisfaire le taux d'intérêt i qu'a demandé la banque. Pour cela, procéder de la façon suivante :

- (a) Nous notons d_k la dette qu'il reste à rembourser après k années, $k = 0, 1, 2, \dots, 8$. Déterminer d_0, d_1, d_2 et d_3 en fonction du facteur de capitalisation $r := 1 + i$.

.....

.....

.....

.....

.....

.....

.....

.....

- (b) Dédire du point précédent une formule générale dépendant de r pour d_k et la réduire.

.....

.....

.....

.....

.....

.....

1. L'annuité est un montant fixe qui doit être versé chaque année à la banque. Une partie de ce montant est utilisé pour payer l'intérêt annuel sur la dette et le solde est utilisé pour amortir la dette. Comme l'annuité est constante et comme la dette diminue chaque année, la part de l'annuité consacrée au paiement de l'intérêt diminue chaque année tandis que celle consacrée à l'amortissement augmente. Dans notre cas, l'annuité est a été calculée de manière qu'après 8 ans l'entier de la dette soit remboursée.

(c) Mettre le problème en équation.

.....

.....

.....

.....

.....

Expressions analytiques et méthodes numériques

Dans ce chapitre, par équation nous entendrons une équation à une inconnue de la forme

$$f(x) = 0,$$

où f est une fonction définie sur un intervalle et à valeurs réelles. Le fait d'égaliser un membre de l'équation à 0 n'est pas restrictif car il est toujours possible de soustraire tous les termes du membre de droite de telle sorte que celui-ci devient nul.

Les équations étudiées au gymnase ont généralement un ensemble de solutions qui peut s'écrire à l'aide d'expressions analytiques qui représentent des valeurs exactes. Par exemple, l'équation $x^2 - 5 = 0$ a pour ensemble des solutions $\{-\sqrt{5}; \sqrt{5}\}$. Nous disons alors que les solutions peuvent s'exprimer au moyen de fonctions élémentaires (il s'agit de toutes les fonctions usuelles : polynômes, racines, fonctions trigonométriques, exponentielles, logarithmes, etc.). Cependant, toutes les équations ne sont pas de cette sorte. Beaucoup de problèmes conduisent à des équations dont nous ne pouvons décrire l'ensemble des solutions au moyen d'expressions analytiques. En 1826, le mathématicien norvégien Niels Abel (1802-1829) a démontré qu'il n'existe pas de formule générale faisant usage uniquement de fonctions élémentaires pour la résolution des équations polynomiales de degré supérieur ou égal à 5. Le mathématicien français Evarist Galois² (1811-1832) retrouve ce résultat mais avec une méthode différente et permettant une réponse plus complète à la problématique de la résolution des équations polynomiales.

Compte tenu du résultat de Galois et Abel, l'équation obtenue dans l'activité d'introduction ne possède pas de solution analytique. Pour résoudre une telle équation, nous devons donc utiliser une méthode numérique.

Déroulement du chapitre

Nous commencerons par présenter la méthode de résolution graphique car elle est généralement une première étape nécessaire à l'application d'une méthode numérique. Celles-ci

2. Pour un savoir plus sur la vie Galois, consulter la revue *Tangente*, Hors série 82 (2022). Le QR-code donné en marge pointe vers le court-métrage *Évariste Galois* de Alexandre Astruc qui relate les dernières heures de la vie de Galois. Ce film a été primé au Festival de Cannes de 1967.



seront présentées dans les sections suivantes : il s'agira des méthodes de la bisection, regula falsi, du point fixe et finalement de Newton. Pour terminer, nous traiterons la résolution d'équations avec les fonctions offertes par Mathematica.

Remerciements

Le présent document se base essentiellement sur un script de Marcel Déléze, Nicolas Gremaud et Eugène Pasquier : qu'ils soient remerciés pour leur travail.

1 Méthode graphique

La construction d'un graphique ne constitue pas une méthode très précise de résolution des équations cependant celui-ci peut nous apporter des informations utiles dont nous pourrions tirer parti. Nous porterons notre attention sur les questions suivantes :

- Quel graphique faire ?
- Que nous dit le graphique au sujet des solutions de l'équation ?

1.1 Exemple

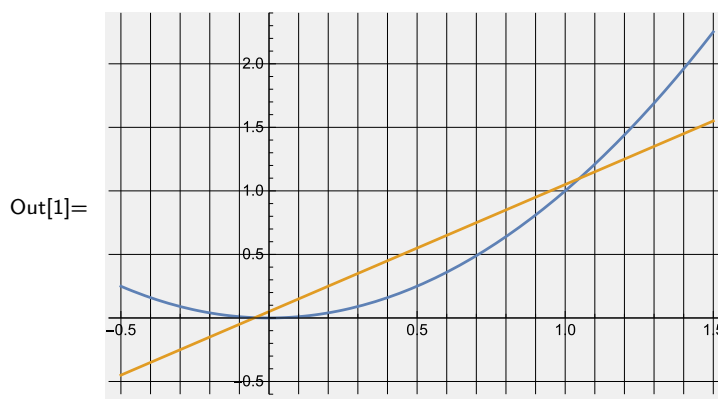
Nous voulons résoudre l'équation

$$0.001x^3 + 0.999x^2 = x + 0.05.$$

Nous allons voir deux points de vue permettant d'approximer graphiquement les solutions cette équation :

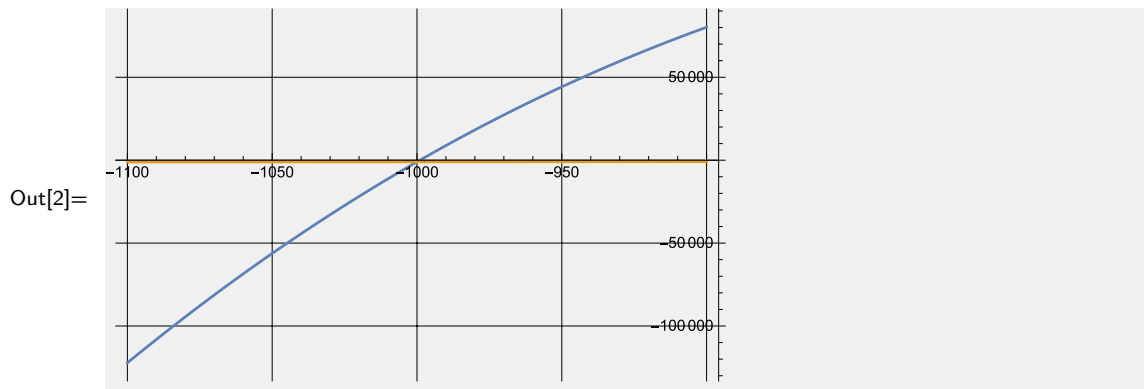
- (a) Nous pouvons considérer qu'il s'agit d'une équation de la forme $g(x) = h(x)$ avec $g(x) = 0.001x^3 + 0.999x^2$ et $h(x) = x + 0.05$. Nous cherchons donc les abscisses des points d'intersection des graphes de g et h :

```
In[1]:= Clear["Global`*"]
f[x_] := 0.001 x^3 + 0.999 x^2
g[x_] := x + 0.05
Plot[
  {f[x], g[x]}, {x, -0.5, 1.5},
  GridLines -> {Range[-0.5, 1.5, 0.1], Automatic},
  GridLinesStyle -> Black
]
```



Comme $\lim_{x \rightarrow -\infty} f(x) = -\infty$, il doit y avoir une troisième solution ! Après plusieurs essais, nous trouvons qu'il y a la troisième solution se trouve entre -1100 et -900 :

```
In[2]:= Plot[
  {f[x], g[x]}, {x, -1100, -900},
  GridLines -> Automatic, GridLinesStyle -> Black
]
```



Comme l'équation à résoudre est du troisième degré, il n'y a pas d'autres solutions et ces solutions (ou racines) se trouvent dans les intervalles

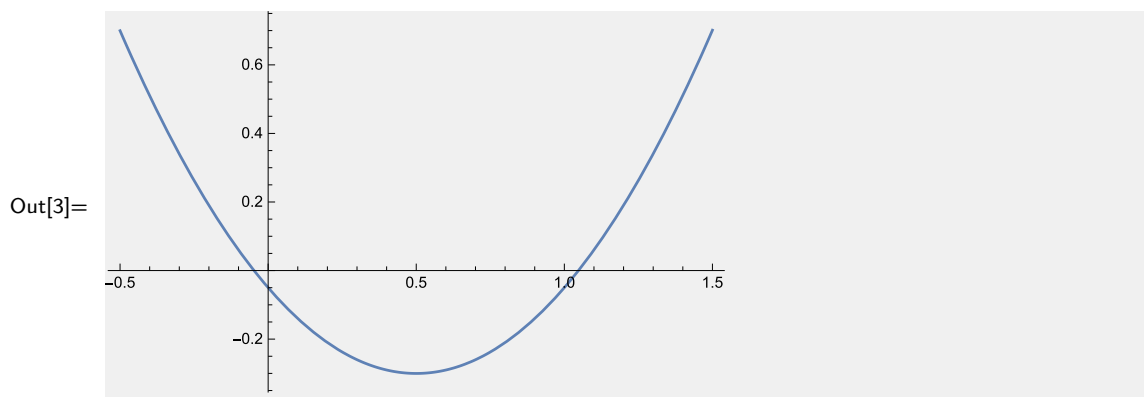
$$]-1010; -990[,]-0.1; 0[\text{ et }]1; 1.1[.$$

(b) Une autre façon de procéder est d'égaliser un des membres de l'équation à zéro :

$$0.001x^3 + 0.999x^2 - x - 0.05 = 0.$$

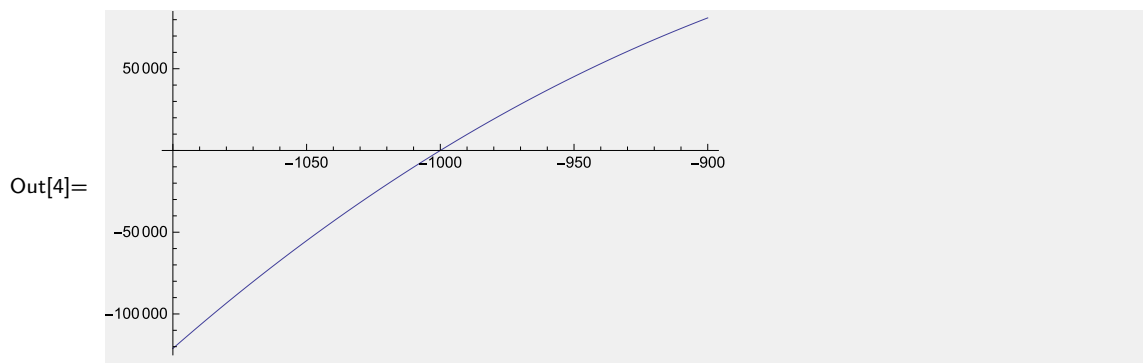
L'équation prend alors la forme $h(x) = 0$ avec $h = f - g$ et nous cherchons les abscisses des points d'intersection du graphe de f avec l'axe des x :

```
In[3]:= h[x_] := f[x] - g[x]
Plot[h[x], {x, -0.5, 1.5}]
```



En traçant le graphe de h entre -1100 et -900 nous trouvons la même solution :

```
In[4]:= Plot[h[x], {x, -1100, -900}]
```



Nous obtenons donc les trois mêmes intervalles contenant les solutions qu'avec le premier point de vue.



Dans l'exemple précédent la résolution graphique nous a permis de trouver des intervalles contenant les racines (ou solutions). Nous disons que nous avons encadré et séparé les racines. Précisons cette terminologie

1.2 Définitions

Considérons une équation à une inconnue réelle.

- (a) **Encadrer une racine** de l'équation signifie donner un intervalle
 - (i) de longueur finie,
 - (ii) inclus dans l'ensemble de définition de l'équation et
 - (iii) contenant au moins une racine.
- (b) **Séparer les racines** signifie encadrer chaque racine de telle sorte que chaque encadrement contienne une unique racine.

1.3 Remarques

Nous avons vu dans l'exemple 1.1 qu'une connaissance des propriétés de l'équation à résoudre était nécessaire pour garantir l'encadrement de *toutes* ses solutions. Cependant, en plus de cela, une lecture attentive des graphiques fournis par Mathematica est nécessaire. Nous verrons dans l'exemple suivant que ceux-ci n'ont pas forcément des axes qui se coupent à l'origine.



1.4 Exemple

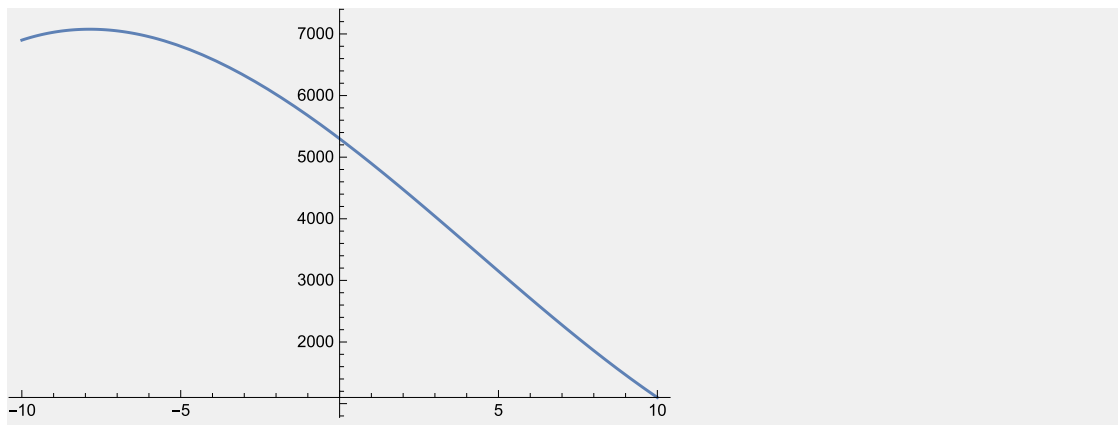
Nous cherchons à encadrer les racines de l'équation

$$5400 - 390x - 13x^2 + x^3 = 0.$$

Pour cela nous posons $f(x) = 5400 - 390x - 13x^2 + x^3$ et nous traçons le graphe de f sur un intervalle choisi arbitrairement :

```
Clear["Global`*"]
In[5]:= f[x_] := 5300 - 390 x - 13 x^2 + x^3
Plot[f[x], {x, -10, 10}]
```

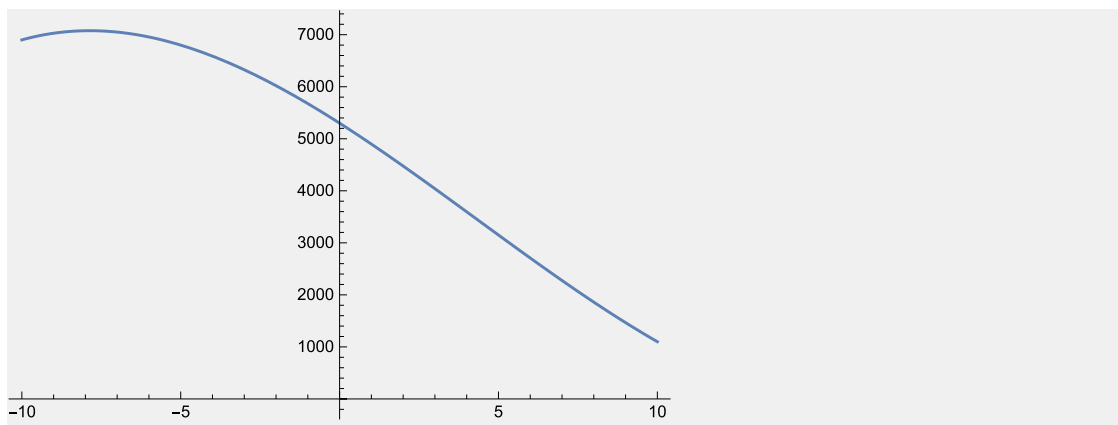
Out[5]=



Nous pourrions être tentés de dire que l'équation possède une solution $x_1 \approx 10$. Nous remarquons cependant que l'axe horizontal n'a pas été dessiné à la hauteur $y = 0$. L'option `AxesOrigin` permet d'éviter cette erreur :

```
In[6]:= Plot[f[x], {x, -10, 10}, AxesOrigin -> {Automatic, 0}]
```

Out[6]=



Il est possible d'éviter d'avoir à insérer cette option dans chaque commande `Plot` grâce à l'instruction `SetOptions`. Cette option devient alors une option par défaut de `Plot` tant que le noyau de Mathematica n'est pas quitté.

```
In[7]:= SetOptions[Plot, AxesOrigin -> {Automatic, 0}, ImageSize -> {500, 300}]
```

```

Out[7]= {AlignmentPoint → Center, AspectRatio →  $\frac{1}{\text{GoldenRatio}}$ , Axes →
True, AxesLabel → None, AxesOrigin → {Automatic, 0},
AxesStyle → {}, Background → None, BaselinePosition →
Automatic, BaseStyle → {}, ClippingStyle → None,
ColorFunction → Automatic, ColorFunctionScaling → True,
ColorOutput → Automatic, ContentSelectable → Automatic
, CoordinatesToolOptions → Automatic , DisplayFunction
→ $DisplayFunction, Epilog → {}, Evaluated → Automatic
, EvaluationMonitor → None, Exclusions → Automatic,
ExclusionsStyle → None, Filling → None, FillingStyle →
Automatic, FormatType → TraditionalForm, Frame → False
, FrameLabel → None, FrameStyle → {}, FrameTicks →
Automatic, FrameTicksStyle → {}, GridLines → None,
GridLinesStyle → {}, ImageMargins → 0., ImagePadding →
All, ImageSize → Automatic, ImageSizeRaw → Automatic,
LabelStyle → {}, MaxRecursion → Automatic, Mesh → None
, MeshFunctions → {#1 &}, MeshShading → None, MeshStyle
→ Automatic, Method → Automatic, PerformanceGoal → $
PerformanceGoal, PlotLabel → None, PlotPoints →
Automatic, PlotRange → {Full , Automatic},
PlotRangeClipping → True, PlotRangePadding → Automatic,
PlotRegion → Automatic, PlotStyle → Automatic,
PreserveImageOptions → Automatic, Prolog → {},
RegionFunction → (True &), RotateLabel → True, Ticks →
Automatic, TicksStyle → {}, WorkingPrecision →
MachinePrecision}

```

Mathematica nous a retourné la liste de toutes les options qui sont choisies par défaut et nous constatons que l'option `AxesOrigin` a été modifiée comme demandée. ☺

1.5 Remarque

Dans le cas où nous n'avons pas à disposition un traceur de graphe, un encadrement et une séparation des racines peuvent s'obtenir en dressant un tableau retournant pour différentes abscisses le signe de la fonction dont on cherche à déterminer les zéros. Pour la fonction traitée de l'exemple 1.4 précédent, un tel tableau a l'allure suivante :

```

TableForm[
In[8]:= Transpose[Table[{x, Sign[f[x]]}, {x, 10, 15}]],
TableHeadings -> {"x", "Sgn_f(x)", None}]

Out[8]=

```

x	10	11	12	13	14	15
$\text{sgn}(f(x))$	1	1	1	1	1	-1

Comme la fonction f est continue, nous pouvons déduire du théorème de Bolzano (théorème de la valeur intermédiaire) et de ce tableau que $[14; 15]$ est un encadrement d'une solution. Pour garantir que cette intervalle ne contient qu'un seul zéro, une possibilité serait de trouver un encadrement des deux autres zéros de f en procédant de façon analogue.

2 Méthode de la bisection

La méthode de la bisection est une méthode facilement faisable avec uniquement une calculatrice. Étymologiquement, « bisection » signifie « couper en deux » : ce qui sera coupé en deux sera l'encadrement de la solution dont une estimation est recherchée. Pour décrire cette méthode, nous allons considérer l'exemple suivant :

2.1 Exemple

Nous voulons déterminer la plus petite racine de l'équation

$$x^3 - 4x^2 - 20x + 1 = 0$$

avec une précision de $1/100$.

- (1) Nous commençons par encadrer et séparer les racines. Pour cela nous considérons le tableau suivant contenant le signe de f pour différentes valeurs :

x	-3	-2	1	7
$f(x)$	$-$	$+$	$-$	$+$

De ce graphe nous déduisons les encadrements suivants des solutions de l'équation considérée :

$$x_1 \in]-3; -2[, \quad x_2 \in]-2; 1[, \quad x_3 \in]1; 7[.$$

L'équation étant de degré 3, nous sommes sûrs que nous avons encadré toutes les solutions et comme nous cherchons la plus petite solution, c'est l'encadrement $] -3; -2[$ qui nous intéresse.

- (2) Comme nous cherchons la plus petite solution, c'est l'encadrement $] -3; -2[$ qui nous intéresse et nous appliquons la méthode de la bisection à cet intervalle. Pour cela, il faut commencer par décider de la précision à laquelle nous voulons parvenir : nous choisissons une précision de ± 0.001 . Rappelons-nous que la fonction f est négative à gauche de l'intervalle et positive à droite :

```
In[1]:= a=-3; b=-2;  
        {Sign[f[-3]], Sign[f[-2]]}
```

```
Out[1]= {-1,1}
```

Nous calculons le signe de la fonction au milieu de l'intervalle :

```
In[2]:= c=(a+b)/2  
        Sign[f[c]]
```

```
Out[2]= -2.5  
        1
```

La fonction change de signe entre -3 et -2.5 . Nous en déduisons un nouvel encadrement : $x_1 \in]-3; -2.5[$. Nous recommençons l'opération et calculons le signe de la fonction au milieu du nouvel intervalle :

```

b=c ;
In[3]:= c=(a+b)/2
Sign[f[c]]

```

```

Out[3]= -2.75
1

```

La fonction change de signe entre -3 et -2.75. Nous en déduisons un nouvel encadrement : $x_1 \in]-3; -2.75[$

En poursuivant de la même façon nous obtenons la suite d'encadrements $]a_n; b_n[$ suivante (i correspondant au nombre de bisections effectuées) :

n	a_n	$\text{sgn}(f(a_n))$	b_n	$\text{sgn}(f(b_n))$	$c_n = \frac{a_n+b_n}{2}$	$\text{sgn}(f(c_n))$
0	-3.	-	-2.	+	-2.5	+
1	-3.	-	-2.5	+	-2.75	+
2	-3.	-	-2.75	+	-2.875	+
3	-3.	-	-2.875	+	-2.9375	-
4	-2.9375	-	-2.875	+	-2.90625	+
5	-2.9375	-	-2.90625	+	-2.921875	+
6	-2.9375	-	-2.921875	+		

Si nous faisons les calculs à la main, nous observons un phénomène désagréable : le nombre de chiffres après la virgule augmente. Étant donné que nous désirons un résultat à la précision d'un centième, nous pouvons à chaque étape arrondir la nouvelle borne à deux chiffres après la virgule et itérer la méthode jusqu'à l'obtention de la précision souhaitée. Dans notre exemple, nous y parvenons en 6 étapes (c_n correspond au centre de l'encadrement arrondi au centième) :

n	a_n	$\text{sgn}(f(a_n))$	b_n	$\text{sgn}(f(b_n))$	$c_n \approx \frac{a_n+b_n}{2}$	$\text{sgn}(f(c_n))$
0	-3.	-	-2.	+	-2.5	+
1	-3.	-	-2.5	+	-2.75	+
2	-3.	-	-2.75	+	-2.88	+
3	-3.	-	-2.88	+	-2.94	-
4	-2.94	-	-2.88	+	-2.91	+
5	-2.94	-	-2.91	+	-2.92	+
6	-2.94	-	-2.92	+		

Nous avons donc $x_1 \in]-2.94; -2.92[$, ce qui peut aussi s'écrire

$$x_1 = -2.93 \pm 0.01.$$

En 6 itérations, la longueur de l'intervalle est passée de 1 à 0.02, c'est-à-dire a été divisée par 50. A chaque itération, la longueur de l'intervalle a été divisée par 2 (ou approximativement divisé par 2 dans le cas où les bornes sont arrondies). En 6 itérations, il a été divisée par $2^6 = 64$



Résumons l'algorithme que nous avons appliqué :



MÉTHODE DE LA BISSECTION

- (1) *Considérer une fonction continue $f : [a; b] \rightarrow \mathbb{R}$ avec $\text{sgn}(f(a)) \cdot \text{sgn}(f(b)) < 0$, ce qui garantit via le théorème de Bolzano que f possède un zéro dans l'intervalle $]a; b[$.*
- (2) *Calculer $c = \frac{a+b}{2}$ le milieu de l'intervalle, c devenant une nouvelle borne de l'encadrement suivant le logigramme de la figure 1.*
- (3) *Pour éviter les longs codes à virgule, possibilité d'arrondir directement la valeur de la nouvelle borne c en fonction de la précision recherchée.*

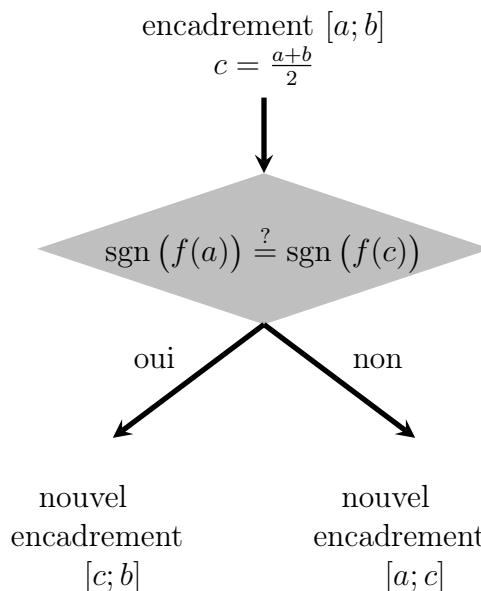


FIGURE 1 – méthode de la bisection

La dernière étape fournie par la méthode de la bisection nous donne l'information $x \in [a; b]$ et nous écrivons alors la réponse finale sous la forme

$$x = \frac{a+b}{2} \pm \frac{b-a}{2},$$

ce qui signifie que la solution x est connue avec une erreur absolue $\Delta x = \frac{b-a}{2}$.

Se pose alors la question du critère d'arrêt. Une possibilité est de se donner au départ une précision Δx et de poursuivre les calculs jusqu'à ce que cette précision soit atteinte. Ceci est le cas lorsque le dernier encadrement $[a; b]$ fourni par la méthode de la bisection satisfait l'inégalité $\frac{b-a}{2} \leq \Delta x$, c'est-à-dire si la largeur de l'intervalle est inférieure au double de la précision désirée

$$(b - a) \leq 2\Delta x$$

nous pouvons arrêter les calculs et écrire la réponse finale, sinon il faut effectuer un ou plusieurs pas supplémentaires.

2.2 Propriétés (de la méthode de la bisection)

La méthode de la bisection présente les avantages suivants :

- (a) *Elle ne nécessite que peu d'hypothèses sur la fonction f . En effet, par le théorème de Bolzano, il suffit que la fonction f soit continue.*
- (b) *La convergence est garantie.*
- (c) *Elle donne une borne de l'erreur : sachant que la solution x appartient à l'intervalle $[a; b]$, alors $\frac{a+b}{2}$ est une approximation de x avec une erreur $\frac{b-a}{2}$.*

La méthode de la bisection présente les inconvénients suivants :

- (a) *La convergence est lente (voir l'exercice [11](#)).*
- (b) *Il n'existe pas de méthode basée sur la bisection qui soit pratiquement utilisable pour résoudre des systèmes d'équations.*

La preuve de la convergence des bornes des intervalles obtenus par la méthode de la bisection a été faite dans le cours de mathématiques de troisième année dans le cadre de la démonstration du théorème de la valeur intermédiaire.

Compte tenu des faiblesses de la méthode de la bisection, il s'avère nécessaire de développer d'autres méthodes.

3 Méthode regula falsi

Soit $f : [a; b] \rightarrow \mathbb{R}$ une fonction continue avec $f(a) \cdot f(b) < 0$ (cette fonction satisfait donc les hypothèses permettant d'appliquer la méthode de la bisection). La méthode de la bisection converge lentement car à chaque étape la longueur de l'intervalle $[a; b]$ encadrant la racine r recherchée est exactement multipliée par $\frac{1}{2}$. Le but de la méthode regula falsi (appelée aussi méthode de la fausse position) est de fournir à chaque étape un nouvel encadrement qui multiplie la longueur de l'encadrement précédemment considéré par un facteur inférieur à $\frac{1}{2}$. Pour cela, nous approximations la fonction f par la droite passant par les points $(a; f(a))$ et $(b; f(b))$ et l'abscisse c du point d'intersection de cette droite avec l'axe des x constitue une des bornes du nouveau encadrement. Le nouveau encadrement considéré ($[a; c]$ ou $[c; b]$) est choisi, comme pour la méthode de la bisection, de façon à ce que les images des bornes soient de signes opposés (ceci est illustré dans la figure 2 où la longueur du nouvel encadrement est approximativement $\frac{1}{4}$ de celle du précédent). Nous répétons l'opération jusqu'à atteindre la précision désirée.

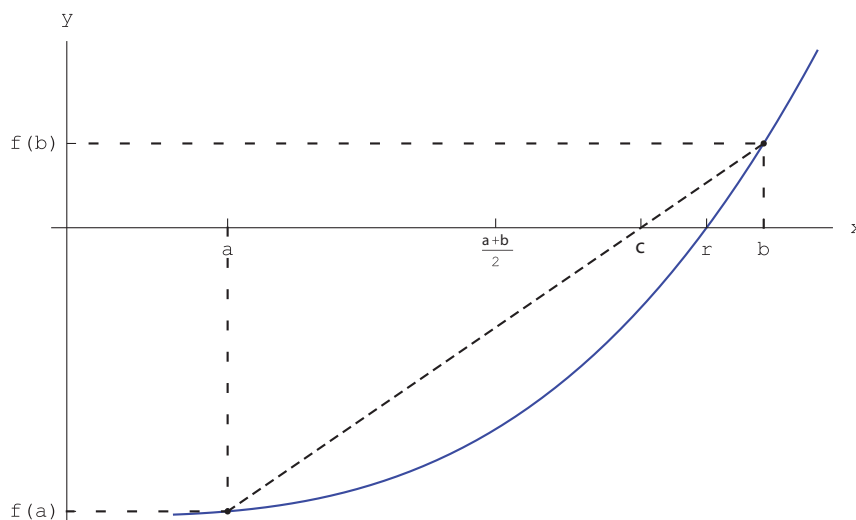


FIGURE 2 – Nouvel encadrement $[c; b]$ obtenu par la méthode regula falsi à partir d'un encadrement initial $[a; b]$.

L'abscisse c de l'intersection de l'axe des x avec la sécante au graphe de f passant par les points $(a; f(a))$ et $(b; f(b))$ et vaut

$$c = \frac{af(b) - bf(a)}{f(b) - f(a)}.$$

Pour une preuve de cette formule, voir l'exercice 15.

Appliquons cette méthode dans un exemple :

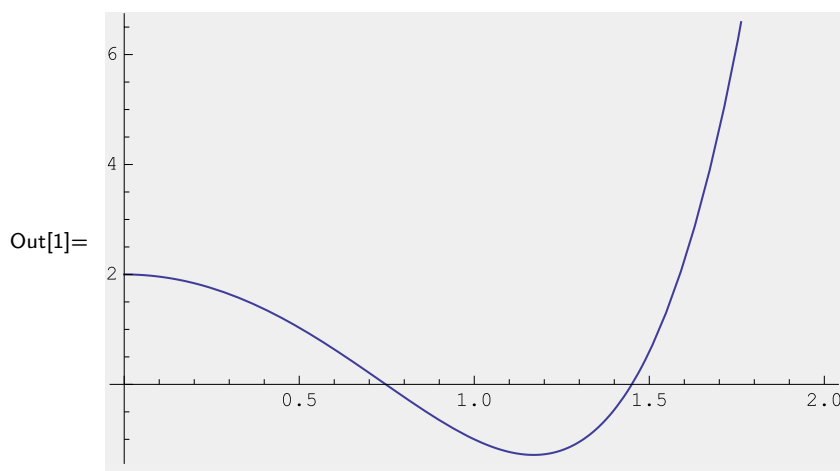
3.1 Exemple

Déterminons à l'aide de la méthode regula falsi le plus petit zéro positif de la fonction

$$f(x) = x^5 - 4x^2 + 2.$$

Nous devons commencer par trouver un encadrement de ce zéro. Pour cela, nous traçons le graphe de f :

```
Clear["Global`*"]
In[1]:= f[x_] := x^5 - 4x^2 + 2
Plot[f[x], {x, 0, 2}]
```



Pour éviter le laborieux calcul de la nouvelle borne, nous définissons une fonction le faisant :

```
In[2]:= rf[{a_, b_}] := (a f[b] - b f[a]) / (f[b] - f[a])
```

Nous prenons comme encadrement initial $[a; b] = [0; 1]$ et calculons la nouvelle borne ainsi que le signe de f à celle-ci :

```
a = 0.; b = 1.;
In[3]:= c = rf[{a, b}] (*nouvelle borne*)
Sign[f[c]]
```

Out[3]= 0.666667

Out[4]= 1

Comme $f(c)$ et $f(a) = f(0)$ sont de même signe, c remplace la borne de gauche $a = 0$ tandis que la borne de droite $b = 1$ reste inchangée. Nous faisons cette affectation et calculons la nouvelle borne ainsi que le signe de f à celle-ci :

```
a = c;
In[5]:= {a, b} (*nouvelle encadrement*)
c = rf[{a, b}] (*nouvelle borne*)
Sign[f[c]]
```

Out[5]= {0.666667,1}

Out[6]= 0.753799

Out[7]= -1

Comme $f(c)$ et $f(b) = f(1)$ sont de même signe, c remplace la borne $b = 1$. Nous faisons cette affectation et calculons la nouvelle borne ainsi que le signe de f à celle-ci :

```
In[8]:= b = c;
{a, b} (*nouvelle encadrement*)
c = rf[{a, b}] (*nouvelle borne*)
Sign[f[c]]
```

Out[8]= {0.666667,0.753799}

Out[9]= 0.7471

Out[10]= 1

Comme $f(c)$ et $f(a)$ sont de même signe, c remplace la borne a . Nous faisons cette affectation et calculons la nouvelle borne ainsi que le signe de f à celle-ci :

```
In[11]:= a = c;
{a, b} (*nouvelle encadrement*)
c = rf[{a, b}] (*nouvelle borne*)
Sign[f[c]]
```

Out[11]= {0.7471,0.753799}

Out[12]= 0.747127

Out[13]= -1

Comme $f(c)$ et $f(b)$ sont de même signe, c remplace la borne b et le nouvel encadrement considéré est donc $[0.7471, 0.747127]$. Nos résultats peuvent se résumer dans le tableau suivant :

n	a_n	$\text{Sgn}(f(a_n))$	b_n	$\text{Sgn}(f(b_n))$	c	$\text{Sgn}(f(c_n))$
0	0.	+	1.	-	0.666 667	+
1	0.666 667	+	1.	-	0.753 799	-
2	0.666 667	+	0.753 799	-	0.7471	+
3	0.7471	+	0.753 799	-	0.747 127	-
4	0.7471	+	0.747 127	-		

Nous constatons qu'après 4 itérations nous avons un intervalle de longueur inférieure à 10^{-4} alors qu'avec le même nombre d'itérations la méthode de la bisection aurait donné un intervalle de longueur $\frac{1}{2^4} = 0.0625$! ☺

3.2 Propriétés (de la méthode regula falsi)

Soit $[a; b]$ un intervalle et $f : [a, b] \rightarrow \mathbb{R}$ une fonction deux fois dérivable avec $f(a)f(b) < 0$.

- (a) Tant que la fonction f change de courbure (convexe puis concave ou l'inverse) sur l'encadrement considéré, la longueur de celui-ci décroît plus vite que pour la méthode de la bisection.
- (b) Si la fonction a la même courbure sur l'encadrement alors une des bornes reste fixe tandis que l'autre converge vers la racine recherchée (pour un exemple, voir l'exercice 16). Cette convergence peut cependant être très lente (pour un exemple, voir l'exercice 18) : la méthode regula falsi n'est donc pas toujours meilleure que la méthode de la bisection !

Sans preuve.

3.3 Remarque

Une méthode qui s'inspire de la méthode regula falsi est la méthode de la sécante. A partir de deux approximations initiales x_0 et x_1 de la racine recherchée nous définissons la suite $x_{n+1} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$. Une variation de la cette méthode est utilisée par Mathematica pour l'implémentation de la méthode `FindRoot[eqn, {x, x0, x1}]` (voir la section 6).

4 Méthode du point fixe

Il n'existe pas de méthode efficace basée sur la bisection pour résoudre un système de plusieurs équations à plusieurs inconnues. C'est un des intérêts de la méthode du point fixe que nous allons voir dans cette section. Pour une première approche commençons pour l'activité suivante :

🌐 Itération d'une fonction

Pour une calculatrice réglée en radian, nous calculons le cosinus de 1, puis le cosinus du résultat, puis encore le cosinus du résultat et ainsi de suite. Nous obtenons alors la suite de nombres $(x_n)_{n \in \mathbb{N}}$ définie de la façon suivante :

$$x_0 = 1, \quad x_{n+1} = \cos(x_n) \text{ pour } n \in \mathbb{N}.$$

(a) Calculer les 4 premiers termes de cette suite.

.....

(b) Estimer la limite de cette suite.

(c) De quelle équation le nombre trouvé au point précédent est-il solution ?

Dans l'activité précédente, nous avons constaté qu'en itérant une fonction, nous pouvons obtenir une solution d'une équation. Nous montrerons que, sous certaines conditions, la valeur vers laquelle converge l'itération est un point fixe de la fonction :

4.1 Définition

Soit $I \subset \mathbb{R}$ et $g : I \rightarrow \mathbb{R}$ une fonction. Un **point fixe** de g est un nombre $r \in I$ tel que

$$g(r) = r.$$

4.2 Exemple

Dans l'activité précédente, $r \approx 0.739\,085\,133\,215\,2$ est un point fixe de la fonction $g(x) = \cos(x)$. ☺

4.3 Définition

Soit $I \subset \mathbb{R}$, $g : I \rightarrow I$ une fonction et soit $x_0 \in I$. La méthode qui consiste à construire la suite $(x_n)_{n \in \mathbb{N}}$ définie par

$$x_{n+1} = g(x_n)$$

pour $n \in \mathbb{N}$, est appelée **méthode du point fixe**. La fonction g est appelée **fonction d'itération**.

En cas de convergence, la méthode du point fixe a la propriété suivante :

4.4 Propriété

Soit I un intervalle ouvert, $g : I \rightarrow I$ une fonction continue sur I et soit $x_0 \in I$. Si la méthode du point fixe appliquée avec g comme fonction d'itération et x_0 comme valeur de départ converge vers un nombre $r \in I$ alors le nombre r est une solution de l'équation

$$x = g(x).$$

Le nombre r est donc un point fixe de g .

DÉMONSTRATION

Soit $(x_n)_{n \in \mathbb{N}}$ la suite obtenue par la méthode du point fixe et r sa limite. Nous avons les égalités suivantes :

$$g(r) = g\left(\lim_{n \rightarrow +\infty} x_n\right) \stackrel{(*)}{=} \lim_{n \rightarrow +\infty} g(x_n) = \lim_{n \rightarrow +\infty} x_{n+1} = r,$$

l'égalité $(*)$ étant vérifiée par la continuité³ de g . □

Afin d'interpréter graphiquement le fonctionnement de la méthode du point fixe, nous allons reprendre l'activité vue en début de section.

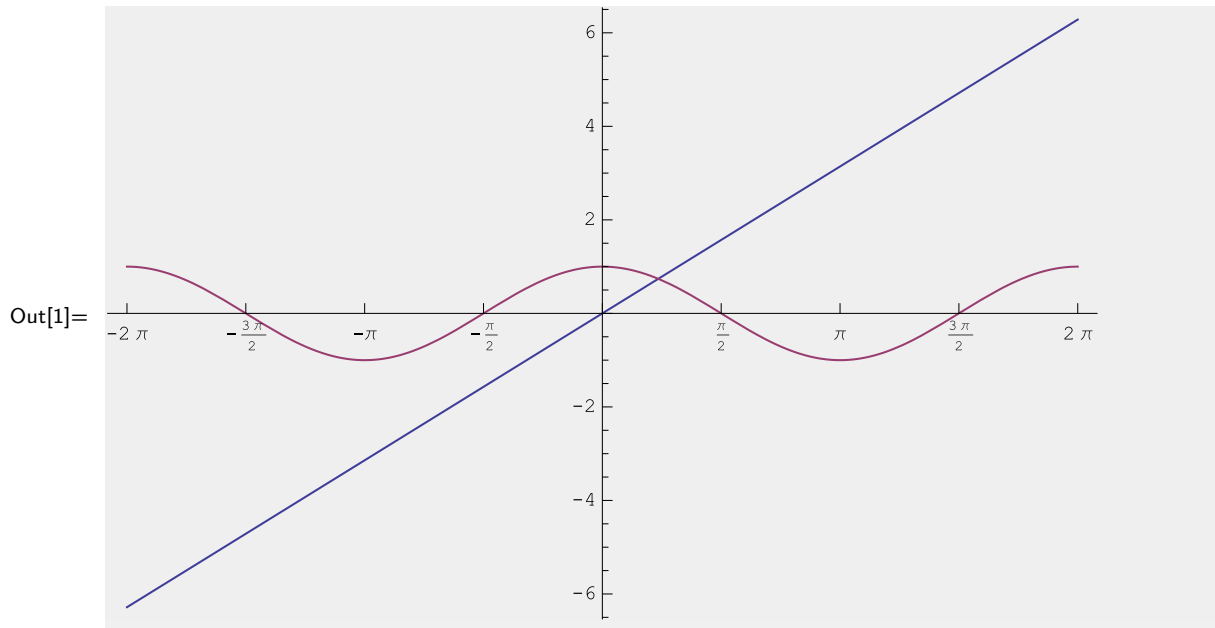
4.5 Exemple

La graphique ci-dessous montre que l'équation $x = \cos(x)$ possède une unique seule solution et celle-ci est située dans l'intervalle $[0; \frac{\pi}{2}]$:

```
Plot[
  {x, Cos[x]}, {x, -2 Pi, 2 Pi},
  Ticks -> {Range[-2 Pi, 2 Pi, Pi/2], Automatic}
]
```

3. Le théorème suivant montre que les suites sont des outils permettant de tester la continuité d'une fonction :

Théorème Soit $I \subset \mathbb{R}$, $g : I \rightarrow \mathbb{R}$ et $a \in I$. La fonction g est continue en a si et seulement si pour toute suite $(x_n)_{n \in \mathbb{N}}$ convergeant vers a la suite $(g(x_n))_{n \in \mathbb{N}}$ converge vers $g(a)$



Pour estimer graphiquement les valeurs des itérations il faut suivre le chemin suivant :

- (1) $(x_0, 0)$ est le point de départ sur l'axe des x ;
- (2) $(x_0, x_1) = (x_0, g(x_0))$ est situé sur la courbe de la fonction $y = g(x)$;
- (3) (x_1, x_1) est situé sur la droite $y = x$;
- (4) $(x_1, x_2) = (x_1, g(x_1))$ est situé sur la courbe $y = g(x)$;
- (5) (x_2, x_2) est un point sur la droite $y = x$;
- (6) ...

Nous parcourons ainsi un chemin qui passe alternativement d'un point sur la courbe à un point sur la droite $y = x$. En reliant ces points et en effectuant un zoom dans le carré $[0.7; 0.8] \times [0.7; 0.8]$, nous obtenons la figure 3. La ligne brisée obtenue s'approche du

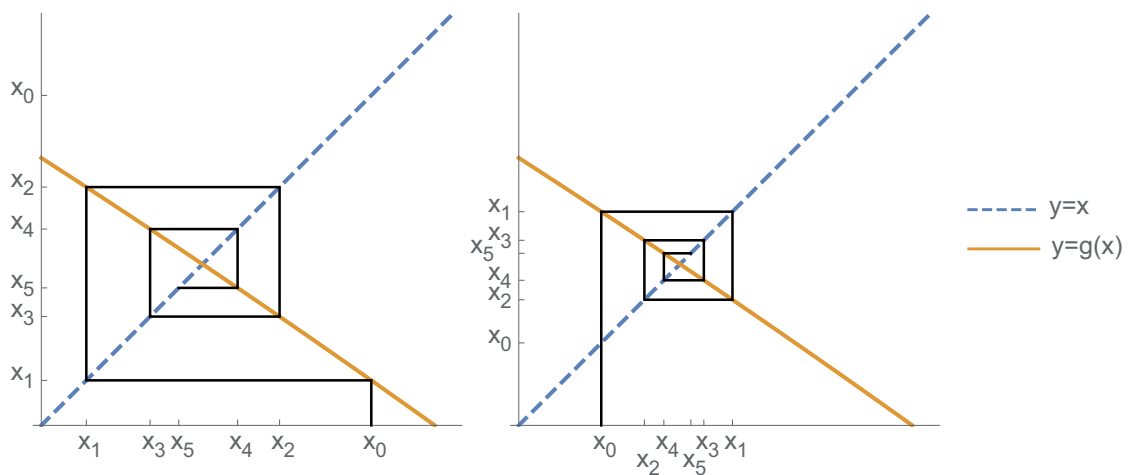


FIGURE 3 – Itérations fournies par la méthode du point fixe pour l'équation $x = \cos(x)$ et les valeurs initiales $x_0 = 0.78$ (à gauche) et $x_0 = 0.72$ (à droite).

point (r, r) qui est situé à l'intersection de la courbe et de la droite. Sur cette même figure, nous pouvons voir que si la méthode démarre avec une autre valeur initiale prise dans la même région, la suite converge vers le même point fixe. ☺

La résolution de n'importe quelle équation peut se ramener à la recherche du point fixe d'une fonction :

4.6 Propriété

Soit $I \subset \mathbb{R}$, $r \in I$ et $f : I \rightarrow \mathbb{R}$ une fonction. Le nombre r est une solution de l'équation $f(x) = 0$ si et seulement si r est un point fixe de la fonction $h(x) = x + f(x)$.

DÉMONSTRATION

Voir l'exercice [20](#). □

L'exemple suivant est une illustration de ce résultat ainsi que du travail avec Mathematica à l'aide des fonctions `NestList`, `FixedPointList` et `FixedPoint` :

4.7 Exemple

Résolvons par la méthode du point fixe l'équation

$$\ln(x) = 4 - x.$$

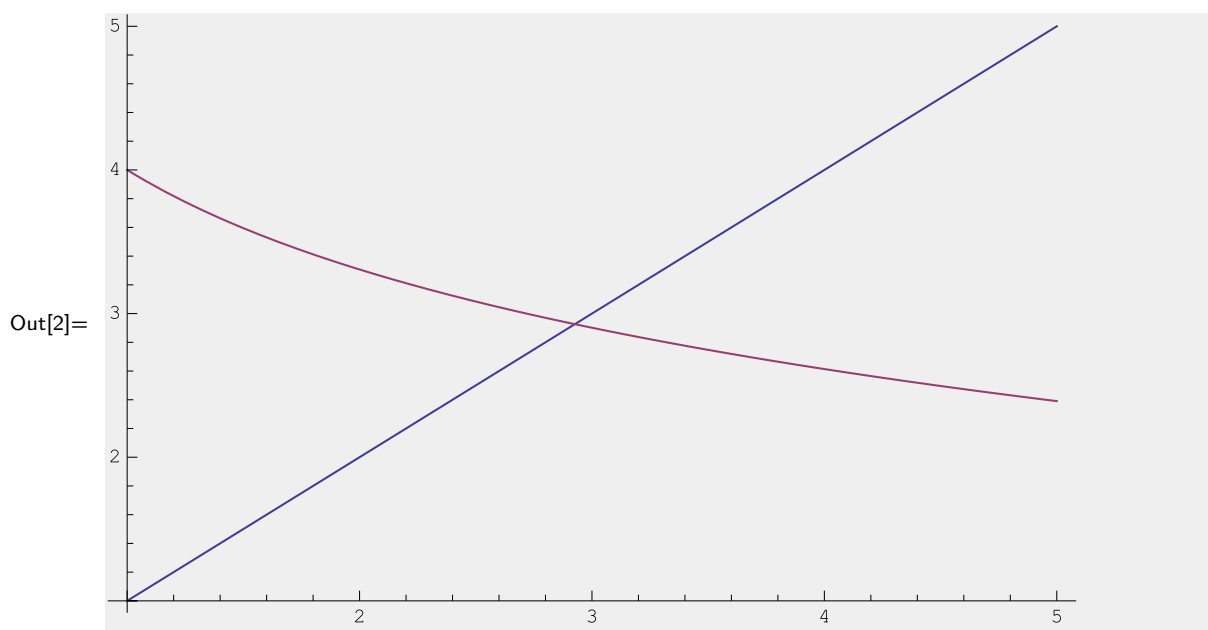
Dans une première étape, nous mettons l'équation sous la forme $x = g(x)$: notre équation à résoudre est équivalente à

$$x = 4 - \ln(x).$$

et elle est de la forme désirée avec $g(x) = 4 - \ln(x)$.

Ensuite, nous déterminons une valeur de départ de l'itération par la méthode graphique :

```
Clear["Global`*"]
In[2]:= g[x_]:=4-Log[x]
Plot[{x,g[x]},{x,1,5}]
```



Nous choisissons alors comme valeur de départ $x_0 = 3$:

```
In[3]:= x0=3;
```

Nous pouvons alors effectuer l'itération :

```
In[4]:= NestList[g, x0, 10]
```

Out[4]= {3., 2.90139, 2.93481, 2.92336, 2.92727, 2.92593, 2.92639, 2.92623, 2.92628, 2.92627, 2.92627}

La situation est représentée graphiquement sur l'intervalle $[2.89; 3.01]$ dans la figure 4. Nous observons que la suite converge vers $r \approx 2.92627$.

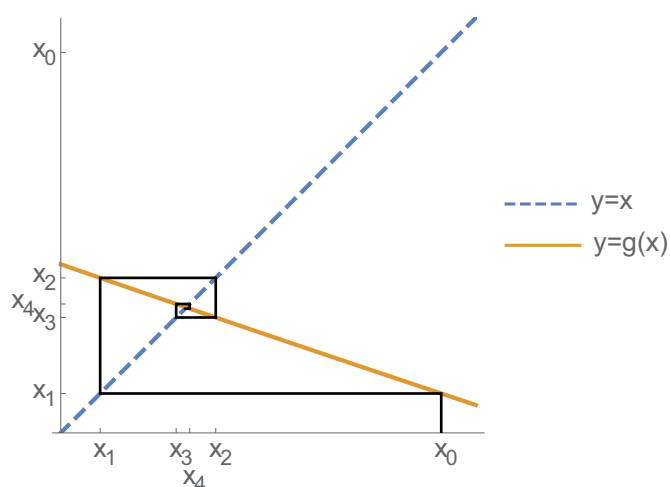


FIGURE 4 – Itérations fournies par la méthode du point fixe pour l'équation $x = 4 - \ln(x)$

Si nous voulons que Mathematica fasse l'itération jusqu'à ce que le point fixe soit « atteint », nous utilisons les fonctions `FixedPointList` et `FixedPoint` :

```
In[5]:= FixedPointList[g, x0]
```

```
Out[5]= {3., 2.90139, 2.93481, 2.92336, 2.92727, 2.92593, 2.92639,
        2.92623, 2.92628, 2.92627, 2.92627, 2.92627, 2.92627, 2.92627,
        2.92627, 2.92627, 2.92627, 2.92627, 2.92627, 2.92627, 2.92627,
        2.92627, 2.92627, 2.92627, 2.92627, 2.92627, 2.92627, 2.92627,
        2.92627, 2.92627, 2.92627, 2.92627, 2.92627}
```

Mathematica continue l'itération jusqu'à avoir 16 chiffres significatifs identiques lors de deux itérations consécutives. Si nous ne voulons pas afficher toutes les itérations amenant au point fixe nous travaillons avec la fonction `FixedPoint`. Nous la combinons ici avec `NumberForm` afin d'afficher les 16 chiffres significatifs calculés par Mathematica :

```
In[6]:= NumberForm[FixedPoint[g, x0], 16]
```

```
Out[6]= 2.926271062443501
```

☺

Malheureusement la méthode ne converge pas toujours (pour des exemples, voir l'exercice 19), cependant, il existe des conditions garantissant sa convergence :

4.8 Théorème

Soit I un intervalle ouvert, $g : I \rightarrow \mathbb{R}$ une fonction continûment dérivable⁴ et $r \in I$ un point fixe de g .

- (a) Si $|g'(r)| < 1$ alors il existe un intervalle ouvert $J \subset I$ contenant r tel que la méthode du point fixe converge vers r quel que soit la valeur de départ $x_0 \in J$ et $|g(x) - r| < |x - r|$ pour tout $x \in J \setminus \{r\}$.
- (b) Si $|g'(r)| > 1$ alors il existe un intervalle ouvert $J \subset I$ contenant r tel que $|g(x) - r| > |x - r|$ pour tout $x \in J \setminus \{r\}$.
- (c) Si $|g'(r)| = 1$, nous pouvons être dans l'une ou l'autre des situations précédentes.

4.9 Remarque

Dans la situation (b) du théorème 4.8, la convergence vers le point fixe r est tout de même possible mais c'est uniquement le cas si l'itération « arrive » sur une préimage de r . Tous les itérés suivants valent alors r .

DÉMONSTRATION (DU THÉORÈME 4.8)

- (a) Comme g' est continue et $|g'(r)| < 1$, il existe un nombre $\delta > 0$ tel que $|g'(x)| < 1$ pour tout $x \in [r - \delta; r + \delta]$. Notons M la valeur maximale de $|g'|$ sur cet intervalle

4. Une fonction réelle f définie sur un intervalle ouvert I est continûment dérivable sur cet intervalle si elle est dérivable en tout point de I et si sa dérivée f' y est continue.

(un théorème qui n'est pas vu au cours de mathématiques garantit l'existence de ce maximum⁵). Par le choix de δ nous avons $M \in [0; 1[$. Posons $J =]r - \delta; r + \delta[$ et montrons que J a la propriété désirée. Pour cela, nous allons démontrer successivement les éléments suivants :

(i) pour tout $x \in J$, nous avons

$$|g(x) - r| \leq M|x - r|; \quad (1)$$

(ii) pour tout $x \in J$ nous avons $g(x) \in J$;

(iii) la méthode du point fixe converge quelle que soit la valeur de départ $x_0 \in J$ (ce qui termine la preuve).

Démontrons ces différentes affirmations :

(i) Par le théorème des accroissements finis, il existe un nombre c entre r et x tel que

$$g(x) - \underbrace{g(r)}_{=r} = g'(c)(x - r). \quad (2)$$

Comme c est entre r et x qui sont dans J et comme J est un intervalle, c est aussi dans J . Il s'en suit que $|g'(c)| \leq M$ et en prenant les valeurs absolues des membres de gauche et droite de l'équation (2), nous obtenons l'inégalité à démontrer :

$$|g(x) - r| = |g'(c)| \cdot |x - r| \leq M|x - r|.$$

(ii) Considérons donc un nombre $x \in J$. Comme $x \in J =]r - \delta; r + \delta[$ nous avons $|x - r| < \delta$. En introduisant cette dernière inégalité dans l'inégalité (1) et en tenant compte du fait que $M < 1$, nous obtenons

$$|g(x) - r| \leq M|x - r| < M\delta < \delta.$$

ce qui montre que $g(x) \in J =]r - \delta; r + \delta[$.

(iii) Considérons une valeur de départ $x_0 \in J$. Il suit de la propriété montrée au point précédent que la méthode du point fixe définit une suite infinie de nombres $(x_n)_{n \in \mathbb{N}}$ et que tous ces nombre sont dans J . Par conséquent, nous déduisons de l'inéquation (1) que nous avons l'inégalité suivante pour tout $n \in \mathbb{N}$:

$$|x_{n+1} - r| = |g(x_n) - r| < M|x_n - r|. \quad (3)$$

De cette inégalité, nous déduisons que

$$|x_{n+1} - r| \leq M^{n+1}|x_0 - r|. \quad (4)$$

Nous avons alors les égalités suivantes :

$$0 \leq \lim_{n \rightarrow \infty} |x_{n+1} - r| \leq \lim_{n \rightarrow \infty} M^n |x_0 - r| = 0 \cdot |x_0 - r| = 0.$$

La suite $(x_n)_{n \in \mathbb{N}}$ converge donc vers r .

5. **Théorème** Soit I un intervalle ouvert et $f : I \rightarrow \mathbb{R}$ une fonction continue. Pour tout intervalle fermé $[a; b] \subset I$, l'image par f de $[a; b]$ est aussi un intervalle borné et fermé. Si nous notons $[m; M]$ ce dernier intervalle, nous avons alors $m \leq f(x) \leq M$ pour tout $x \in [a; b]$, avec m la valeur minimale de f sur $[a; b]$ et M sa valeur maximale.

Dans notre cas, nous avons $f = |g'|$ et $[a; b] = [r - \delta; r + \delta]$.

- (b) Sans preuve.
- (c) Pour montrer cette affirmation, il suffit de donner un exemple de fonction d'itération g avec un point fixe r et $|g'(r)| = 1$ où nous sommes dans la situation (a) et un exemple où nous sommes dans la situation (b) :
- (i) Soit la fonction d'itération $g : \mathbb{R} \rightarrow \mathbb{R}, g(x) = -x^3 + x$. Le nombre $r = 0$ est un point fixe de g et $g'(0) = 1$. Il est possible de montrer que la méthode du point fixe converge vers 0 quelle que soit la valeur de départ prise dans un intervalle J contenant 0 et convenablement choisi (voir l'exercice 21)
- (ii) Soit la fonction d'itération $g : \mathbb{R} \rightarrow \mathbb{R}, g(x) = x^3 + x$. Le nombre $r = 0$ est l'unique point fixe de g et $g'(0) = (3x^2 + 1)|_{x=0} = 1$. Nous pouvons choisir $J = \mathbb{R}$ car pour tout $x \in J \setminus \{r\} = \mathbb{R}^*$ nous avons

$$|g(x) - r| = |x^3 + x - 0| = |x| \cdot |x^2 + 1| > |x| \cdot 1 = |x - 0| = |x - r|,$$

l'inégalité étant vérifiée car x est non-nul.

□

4.10 Propriété

Soit I un intervalle ouvert, $g : I \rightarrow \mathbb{R}$ une fonction continûment dérivable et $r \in I$ un point fixe de g avec $|g'(r)| < 1$. La méthode du fixe converge d'autant plus rapidement que $g'(r)$ est proche de 0.

DÉMONSTRATION

Dans la preuve du théorème du théorème 4.8, nous avons choisi un nombre $\delta > 0$ tel que $|g'(x)| < 1$ pour tout $x \in [r - \delta; r + \delta]$. Nous avons ensuite vu que

$$|x_{n+1} - r| \leq M^{n+1} |x_0 - r|$$

pour tout $x_0 \in]r - \delta; r + \delta[$ où M est le maximum de $|g'|$ sur $[r - \delta; r + \delta]$. La convergence de la méthode du point fixe est donc d'autant plus rapide que M est proche de zéro et M est d'autant plus proche de zéro que $g'(r)$ est proche de zéro. □

5 Méthode de Newton

Dans cette section, nous voulons résoudre une équation de la forme $f(x) = 0$ mais nous adoptons un autre point de vue que ceux présentés jusqu'à présent. L'idée est ici d'approximer la fonction f par une fonction simple dont les zéros sont faciles à calculer. L'approximation choisie pour la méthode de Newton est une tangente au graphe de f .

Nous commençons par préciser géométriquement le fonctionnement de la méthode de Newton. Soit I un intervalle ouvert et $f : I \rightarrow \mathbb{R}$ une fonction dérivable sur I . Nous avons les étapes suivantes pour approximer une solution de l'équation $f(x) = 0$ (ces étapes sont illustrées dans le figure 5) :

- **Etape 0** : Une valeur initiale $x_0 \in I$ est choisie comme approximation initiale de solution de l'équation.
- **Etape 1** : La première approximation fournie par la méthode de Newton est l'abscisse x_1 du point d'intersection de la tangente t_0 au graphe de f en $(x_0; f(x_0))$ avec l'axe des x .
- **Etape 2** : La deuxième approximation est l'abscisse x_2 du point d'intersection de la tangente t_1 au graphe de f en $(x_1; f(x_1))$ avec l'axe des x .
- ...
- **Etape $n + 1$** : La $(n + 1)^{\text{ème}}$ approximation est l'abscisse x_{n+1} du point d'intersection de la tangente t_n au graphe de f en $(x_n; f(x_n))$ avec Ox .

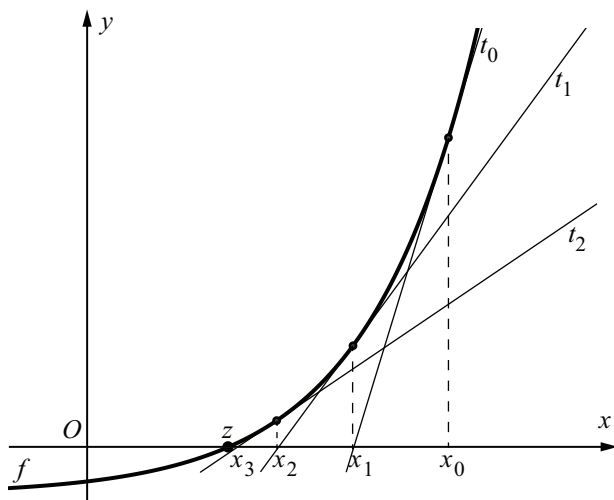


FIGURE 5 – Itérations de la méthode de Newton

Nous déterminons maintenant la formule de récurrence permettant d'obtenir x_{n+1} à partir de x_n . Il a été montré au cours de mathématiques que l'équation de la tangente t_n au graphe de f en $(x_n; f(x_n))$ est

$$y = f'(x_n) \cdot (x - x_n) + f(x_n).$$

Comme x_{n+1} est l'intersection de t_n avec l'axe des x , nous devons avoir

$$0 = f'(x_n) \cdot (x_{n+1} - x_n) + f(x_n).$$

Nous en déduisons alors la formule de récurrence permettant d'obtenir les itérations de la méthode de Newton :

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

En cas de convergence, cette suite a la propriété voulue :

5.1 Théorème

Soit I un intervalle ouvert, $f : I \rightarrow \mathbb{R}$ une fonction continûment dérivable sur I . Si une suite dans I construite par la méthode de Newton converge alors sa limite est un zéro de f .

DÉMONSTRATION

Soit $(x_n)_{n \in \mathbb{N}}$ une suite convergente dans I et construite par la méthode de Newton. Notons z sa limite. Nous devons montrer que $f(z) = 0$:

$$\begin{aligned} f(z) &= f\left(\lim_{n \rightarrow +\infty} x_n\right) \\ &= \lim_{n \rightarrow +\infty} f(x_n) \end{aligned} \tag{5}$$

$$= \lim_{n \rightarrow +\infty} f'(x_n)(x_n - x_{n+1}) \tag{6}$$

$$\begin{aligned} &= \left(\lim_{n \rightarrow +\infty} f'(x_n)\right) \left(\left(\lim_{n \rightarrow +\infty} x_n\right) - \left(\lim_{n \rightarrow +\infty} x_{n+1}\right)\right) \\ &= f'\left(\lim_{n \rightarrow +\infty} x_n\right) (z - z) \\ &= f'(z) \cdot 0 \\ &= 0 \end{aligned} \tag{7}$$

L'égalité (5) est justifiée par la continuité⁶ de f , l'égalité (6) découle de la définition récursive de la suite obtenue par la méthode de Newton et l'égalité (7) est justifiée par la continuité de f' . $\square$

Illustrons le fonctionnement de cette méthode par un premier exemple

5.2 Exemple

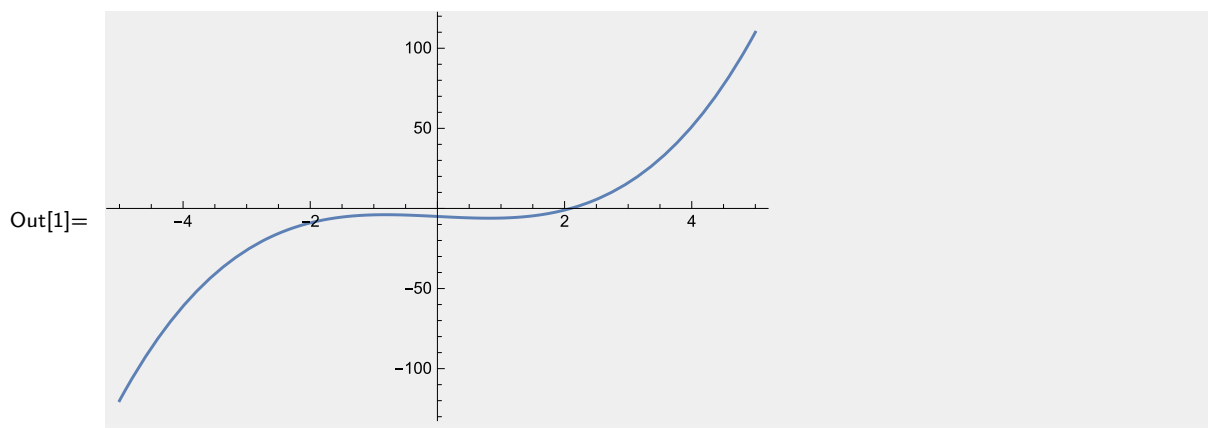
Nous voulons résoudre l'équation

$$x^3 - 2x - 5 = 0.$$

la méthode graphique nous permet une première estimation des solutions.

```
Clear["Global`*"]
In[1]:= f[x_]:=x^3 - 2 x - 5
Plot[f[x], {x, -5, 5}]
```

6. Voir la note de bas de page numéro 3 p.19.



Nous constatons que la fonction f est une fonction cubique avec deux extrema locaux négatifs. Elle possède donc un unique zéro z . Vérifions que l'intervalle $[2; 3]$ est un encadrement de z :

```
In[2]:= Sign[{f[2], f[3]}]
```

```
Out[2]= {-1, 1}
```

Pour mettre en œuvre la méthode de Newton, nous allons prendre $x_0 = 2$ comme valeur de départ. Nous avons $f'(x) = 3x^2 - 2$ et la suite $(x_n)_{n \in \mathbb{N}}$ des approximations de z est donnée par la relation de récurrence

$$x_{n+1} = x_n - \frac{x_n^3 - 2x_n - 5}{3x_n^2 - 2}.$$

Nous obtenons alors facilement à l'aide de la calculatrice les itérations suivantes :

$$x_1 = 2.1, \quad x_3 \approx 2.09457 \quad x_4 \approx 2.09455 \quad \text{et} \quad x_5 \approx 2.09455.$$

Dès lors, nous pouvons prétendre que les cinq décimales de l'approximation $z \approx 2.09455$ sont correctes.

Faisons le travail précédent avec Mathematica :

```
x0 = 2.;
In[3]:= g[x_] := x - f[x]/f'[x]
FixedPointList[g, x0]
```

```
Out[3]= {2., 2.1, 2.09457, 2.09455, 2.09455, 2.09455}
```

```
In[4]:= NumberForm[%[[-1]], 16]
```

```
Out[4]= 2.0945514815423265
```

Nous constatons que seulement 5 itérations ont suffi pour approximer la solution de l'équation avec une précision de 16 chiffres significatifs! ☺

Afin de garantir la convergence, un bon choix de la valeur initial est nécessaire. L'exemple suivant l'illustre.

5.3 Exemple

Pour la fonction et la valeur de départ présentées dans la figure 6, nous remarquons que x_1

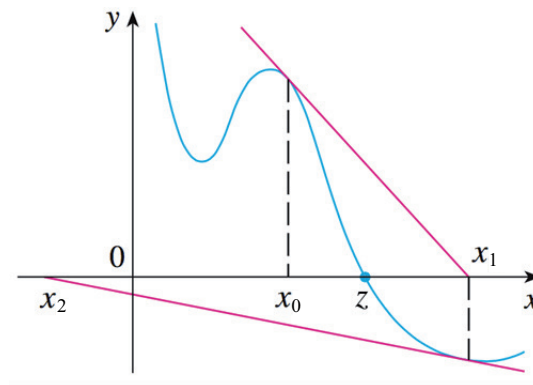


FIGURE 6 – Non-convergence de la méthode de Newton par mauvais choix de la valeur initiale x_0 .

est une moins bonne approximation du zéro recherché que x_0 et la deuxième approximation est même en dehors de l'ensemble de définition de f . ☺

Le théorème suivant présente des hypothèses suffisantes (dites **hypothèses de Fourier**) garantissant la convergence de la méthode de Newton :

5.4 Théorème

Soit $I = [a; b]$ un intervalle et $f : I \rightarrow \mathbb{R}$ une fonction. Si f satisfait toutes les hypothèses suivantes :

- (1) f est deux fois continûment dérivable sur I ,
- (2) $f(a) \cdot f(b) < 0$,
- (3) f' est non nulle,
- (4) f'' est de signe constant sur I ,

alors la suite obtenue par la méthode de Newton est monotone⁷ et converge vers un zéro de f pour toute valeur initiale $x_0 \in I$ telle que $f(x_0)$ ait le signe de f'' .

Sans preuve

5.5 Exemple

Dans la figure 6 relatif à l'exemple 5.3, la non-convergence est due à un choix de la valeur initiale x_0 non-conforme au théorème précédent : $f(x_0)$ et $f''(x_0)$ n'ont pas le même signe. Nous constatons cependant qu'une valeur de départ x_0 légèrement supérieure à z ne satisfera pas les hypothèses de Fourier mais engendrera tout de même une suite convergente. ☺

7. Une suite est monotone si elle est croissante ou décroissante.

Nous avons vu dans l'exemple précédent que la condition imposée par le théorème 5.4 sur le choix de la valeur de départ x_0 n'était pas nécessaire. De plus, l'exercice ?? montre que toutes les hypothèses de Fourier ne sont pas des hypothèses nécessaires à la convergence de la méthode de Newton. Voici une version avec des conditions plus faibles :

5.6 Théorème

Soit I un intervalle ouvert, $f : I \rightarrow \mathbb{R}$ une fonction deux fois continûment dérivable et z un zéro de f . Si $f'(z) \neq 0$ alors il existe un intervalle ouvert $J \subset I$ contenant z tel que la méthode de Newton converge vers z pour toute valeur de départ $x_0 \in J$.

DÉMONSTRATION

La méthode de Newton peut être vue comme la recherche du point fixe z de la fonction

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

La fonction f étant deux fois continûment dérivable, la fonction g est continûment dérivable et satisfait donc les hypothèses du théorème 4.8. Comme

$$g'(x) = \left(x - \frac{f(x)}{f'(x)} \right)' = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}$$

nous avons $g'(z) = \frac{0 \cdot f''(z)}{(f'(z))^2} = 0$ et il suit du théorème du théorème 4.8 qu'il existe un intervalle ouvert $J \subset I$ contenant z tel que la méthode du point fixe appliquée à g (c'est-à-dire la méthode de Newton) converge vers z pour toute valeur de départ $x_0 \in J$. $\square$

5.7 Remarques

(a) La méthode de Newton a les avantages suivants :

- (i) Il est possible de montrer que la méthode de Newton converge avec un ordre 2 vers un zéro de f : cela signifie qu'il existe une constante $M \in \mathbb{R}$ telle que $|x_{n+1} - z| \leq M|x_n - z|^2$ pour tout $n \in \mathbb{N}$. L'équation (3) p.24 nous donnait par contre un ordre 1 pour la méthode du point fixe. Par conséquent, la méthode de Newton converge généralement plus vite que la méthode du point fixe, pour autant que l'estimation de départ soit suffisamment proche de z . D'une certaine façon la méthode de Newton est une « optimisation » de la méthode du point fixe : nous avons vu dans la propriété 4.10 que la convergence de la méthode du point fixe est d'autant plus rapide que la dérivée de la fonction d'itération est proche de 0 au point fixe. Or, nous avons vu dans la preuve du théorème 5.6 que la dérivée de la fonction d'itération de la méthode de Newton vaut 0 à ces points fixes qui sont les zéros de f .
- (ii) La méthode de Newton s'adapte à la résolution de systèmes de n équations non-linéaires avec n inconnues. Des connaissances d'algèbre linéaire et d'analyse à plusieurs variables sont cependant nécessaires pour l'appréhender.

(b) La méthode de Newton a cependant les inconvénients suivants :

- (i) Il est nécessaire de trouver une approximation de départ suffisamment bonne pour garantir la convergence.

- (ii) La convergence n'est pas garantie si le zéro de f qui est recherché est aussi un zéro de f' (pour des exemples voir l'exercice [27](#)).
- (iii) Il faut être capable de calculer f' .
- (iv) La méthode ne fournit pas d'estimation de l'erreur d'approximation.

6 Résolution avec Mathematica

Mathematica dispose de plusieurs méthodes pour résoudre des équations. Dans cette section, nous allons nous intéresser aux méthodes suivantes : **Solve**, **Reduce** pour trouver des solutions exactes, et, **NSolve** et **FindRoot** pour approximer des solutions. Avant d'aborder les spécificités de ces différentes fonctions, nous allons en préliminaire faire un bref rappel sur les symboles d'affectation et d'égalité et nous allons proposer une classification des équations.

Préliminaire

En Mathematica, le symbole `=` désigne une affectation (ou assignation) directe qui consiste à attribuer directement une valeur à une variable. Par contre, le symbole `==` teste une égalité et son évaluation retourne **True** si l'égalité est vérifiée ou **False** sinon. Illustrons cette différence à l'aide d'un exemple :

6.1 Exemple

Nous commençons par l'affectation de la valeur 5 à la variable `x` :

```
In[1]:= x=5
```

```
Out[1]= 5
```

Testons ensuite une égalité :

```
In[2]:= x==1
```

```
Out[2]= False
```

Mathematica retourne **False** car la valeur 5 a été affectée à `x` et $5 \neq 1$.

Testons une autre égalité :

```
In[3]:= x==5
```

```
Out[3]= True
```

Mathematica retourne **True** car la valeur 5 a été affectée à `x`. ☺

C'est le symbole d'égalité `==` qui est utilisé pour poser une équation car dans une équation nous cherchons la(les) valeur(s) à affecter au(`x`) inconnue(s) telle(s) que l'égalité soit vérifiée. Illustrons cela dans un exemple :

6.2 Exemple

Nous considérons l'équation

$$\frac{2}{3} + \frac{x}{2} = 5.$$

```
In[4]:= Clear["Global`*"]
eqn= 1/2 x + 2/3 == 5
```

```
Out[4]=  $\frac{2}{3} + \frac{x}{2} == 5$ 
```

Comme 0 n'est pas une solution de cette équation, si nous remplaçons x par 0 dans `eqn` Mathematica retourne `False` :

```
In[5]:= eqn /. {x -> 0}
```

```
Out[5]= False
```

☺

Nous proposons maintenant une classification des équations :

6.3 Définitions

*Une **équation algébrique** est une équation qui est équivalente (éventuellement sous certaines restrictions) à une équation de la forme $p(x) = 0$, où p est un polynôme. Une équation qui n'est pas algébrique est une **équation transcendante**.*

Illustrons cette classification :

6.4 Exemple

L'équation $\frac{x+1}{x+7} = 2$ est algébrique car elle est équivalente à $(x+1) - 2(x+7) = 0$ sous la condition $x \neq -7$. Par contre, l'équation $\sin(\alpha) = \cos(\alpha)$ est transcendante car elle ne possède pas d'équation polynomiale équivalente. ☺

Afin de comparer le fonctionnement des méthodes de résolution proposées par Mathematica nous allons travailler avec les équations suivantes :

```
In[6]:= Clear["Global`*"]
(* Equations algébriques *)
eqnA1=x^5+x+1==0;
eqnA2=x^5+x^2+1==0;
eqnA3= a x+b==0;
(* Equations transcendantes *)
eqnT1= Cos[x]==1/2;
eqnT2= Cos[x]==Log[x];
```

Calcul des solutions exactes

Les méthodes proposées pour une *résolution exacte* d'équations sont `Solve` et `Reduce`.

Méthode Solve

La méthode `Solve` est spécialement conçue pour résoudre les équations algébriques mais fonctionne aussi pour certaines équations transcendantes. Nous allons voir comment Mathematica traite les cinq équations que nous avons.

```
In[7]:= Solve[eqnA1, x]
```

```
Out[7]= {{x -> -(-1)^(1/3)}, {x -> (-1)^(2/3)}, {x -> -0.755...},
          {x -> 0.877...-0.745... i}, {x -> 0.877...-0.745... i}}
```

En allant avec le curseur sur chacune des boîtes affichées dans la sortie Mathematica, il s'affiche `Root[1-#1^2+#1^3&,1]`, `Root[1-#1^2+#1^3&,2]` et `Root[1-#1^2+#1^3&,3]`. Cela signifie que Mathematica a donné dans chacune de ces boîtes une approximation d'une racine (c'est-à-dire d'un zéro) de la fonction $y = 1 - x^2 + x^3$. Afin d'obtenir chacun de ces nombre sous forme de radicaux (c'est-à-dire d'expression avec des racines), nous utilisons la fonction `ToRadicals` :

```
In[8]:= ToRadicals[%]
```

```
Out[8]= {{x -> -(-1)^(1/3)}, {x -> (-1)^(2/3)},
          {x -> 1/3 (1 - (2/(25-3*sqrt(69)))^(1/3) - (1/2 (25-3*sqrt(69)))^(1/3))},
          {x -> 1/3 + 1/6 (1 + i*sqrt(3)) (1/2 (25-3*sqrt(69)))^(1/3) + (1-i*sqrt(3))/(3*2^(2/3)*(25-3*sqrt(69))^(1/3))},
          {x -> 1/3 + 1/6 (1 - i*sqrt(3)) (1/2 (25-3*sqrt(69)))^(1/3) + (1+i*sqrt(3))/(3*2^(2/3)*(25-3*sqrt(69))^(1/3))}}
```

Nous pouvons préciser que nous ne nous intéressons qu'aux solutions réelles :

```
In[9]:= Solve[eqnA1, x, Reals]
```

```
Out[9]= {{x -> -0.755...}}
```

Nous constatons que Mathematica ne retourne qu'une seule solution réelle ! En effet, pour Mathematica, $(-1)^{1/3}$ correspond à la racine cubique *complexe* de -1 d'argument positif minimal :

```
z = (-1)^(1/3);
In[10]:= ComplexExpand[z]
AbsArg[z]
```

```
Out[10]= 1/2 + i*sqrt(3)/2
```

```
Out[11]= {1, pi/3}
```

-1 est aussi une racine cubique de -1 mais son argument vaut $\pi > \frac{\pi}{3}$ (si nous voulons calculer la racine cubique réelle d'une expression, il faut utiliser la fonction `Surd`, le terme anglais « surd » signifiant « irrationnel »)

Regardons maintenant comment `Solve` traite l'équation algébrique `eqnA2` :

```
In[12]:= Solve[eqnA2]
```

```
Out[12]= {{x -> -1.19...}, {x -> -0.155...-0.828... i}, {x -> -0.155...-0.828... i},
          {x -> 0.752...-0.785... i}, {x -> 0.752...+0.785... i}, }
```

```
In[13]:= ToRadicals [%]
```

```
Out[13]= {{x -> -1.19...}, {x -> -0.155...-0.828... i}, {x -> -0.155...-0.828... i},
          {x -> 0.752...-0.785... i}, {x -> 0.752...+0.785... i}, }
```

Mathematica n'arrive pas à écrire les solutions de cette équation sous forme de radicaux. Ce n'est pas étonnant car nous avons une équation de degré 5 et il n'existe en effet pas de méthode générale pour résoudre les équations de degré 5.

Regardons maintenant comment `Solve` traite l'équation algébrique `eqnA3` :

```
In[14]:= Solve[eqnA3, x]
```

```
Out[14]= {{x -> -b/a}}
```

Mathematica donne la solution générale de l'équation mais ne traite pas le cas particulier $a = 0$.

Regardons maintenant comment `Solve` traite l'équation transcendante `eqnT1` :

```
In[15]:= Solve[eqnT1, x]
```

```
Out[15]= {{x -> -pi/3 + 2pi c1 if c1 in Z}, {x -> pi/3 + 2pi c1 if c1 in Z}, }
```

Bien que l'équation soit transcendante, `Solve` donne toutes les solutions mais ceci est un cas particulièrement simple. Généralement `Solve` ne résout pas les équations transcendantes :

```
In[16]:= Solve[eqnT2, x]
```

```
Solve: This system cannot be solved with the methods available to
Solve.
```

```
Out[16]= Solve[Cos[x]==Log[x], x]
```

Méthode Reduce

La méthode `Reduce` réduit une équation à des équations équivalentes. Regardons comment elle traite l'équation algébrique `eqnA1` :

```
In[17]:= Reduce[eqnA1, x]
```

```
Out[17]= x == -(-1)^(1/3) || x == (-1)^(2/3) || x == -0.755... ||
          x == 0.877...-0.745... i || x == 0.877...-0.745... i
```

Nous pouvons transformer cette équation en une règle de remplacement avec la fonction

ToRules (ceci est utile pour récupérer un résultat sans faire de copier-coller) :

```
In[18]:= {ToRules [%]}
```

```
Out[18]= {{x -> -(-1)^(1/3)}, {x -> (-1)^(2/3)}, , {x -> -0.755...},
          {x -> 0.877...-0.745... i}, {x -> 0.877...-0.745... i}}
```

Remarquons que nous avons dû entourer l'expression d'accolades pour que la sortie soit donnée sous la forme d'une liste

Comme pour **Solve**, nous pouvons préciser à **Reduce** que nous ne sommes intéressés que par les solutions réelles :

```
In[19]:= Reduce [eqnA1, x, Reals]
```

```
Out[19]= x == -0.755...
```

Nous pouvons aussi utiliser la fonction **ToRadicals** :

```
In[20]:= ToRadicals [%]
```

```
Out[20]= x == 1/3 (1 - (2/(25-3*sqrt(69)))^(1/3) - (1/2 (25-3*sqrt(69)))^(1/3))
```

Regardons comment **Reduce** traite l'équation algébrique **eqnA2** :

```
In[21]:= Reduce [eqnA2]
```

```
Out[21]= x == -1.19... || x == -0.155...-0.828... i || x == -0.155...-0.828... i
          || x == 0.752...-0.785... i || x == 0.752... +0.785... i
```

```
In[22]:= ToRadicals [%]
```

```
Out[22]= x == -1.19... || x == -0.155...-0.828... i || x == -0.155...-0.828... i
          || x == 0.752...-0.785... i || x == 0.752... +0.785... i
```

Comme avec **Solve**, Mathematica n'arrive pas à écrire les solutions de cette équation sous forme de radicaux.

Regardons comment **Reduce** traite l'équation algébrique **eqnA3** :

```
In[23]:= Reduce [eqnA3, x]
```

```
Out[23]= (b == 0 && a == 0) || (a != 0 && x == -b/a)
```

Pour comprendre la signification de ce résultat, rappelons que **&&** signifie « et » et que **||** signifie « ou » (ces symboles peuvent être remplacés par $\wedge$ et $\vee$). La réponse fournie par Mathematica donc signifie que nous avons trois cas possibles :

- 1^{er} cas : si $a = 0$ et $b = 0$, alors il n'y a pas de contrainte sur x donc tout $x \in \mathbb{R}$ est

solution ;

- 2^e cas : si $a \neq 0$, alors $x = -\frac{b}{a}$;
- 3^e cas (qui n'apparaît pas explicitement) : si $a = 0$ et $b \neq 0$, l'équation n'a pas de solution.

Regardons comment **Reduce** traite l'équation transcendante **eqnT1** :

```
In[24]:= Reduce [eqnT1 , x]
```

```
Out[24]= c1 ∈ ℤ && (x == -π/3 + 2πc1 || x == π/3 + 2πc1)
```

Bien que l'équation soit transcendante, **Reduce** donne toutes les solutions mais ceci est un cas particulièrement simple. Généralement **Reduce** ne résout pas les équations transcendantes :

Regardons comment **Reduce** traite l'équation transcendante **eqnT2** :

```
In[25]:= Reduce [eqnT2 , x]
```

```
Reduce: This system cannot be solved with the methods available to
Reduce.
```

```
Out[25]= Reduce [Cos [x]==Log [x] , x]
```

Comparaison de Solve et Reduce

Nous constatons que les deux méthodes présentent les mêmes performances pour des équations sans paramètre. **Solve** a l'avantage d'offrir des solutions facilement réutilisables car elles sont fournies sous la forme de règles de remplacement. Par contre **Reduce** est plus performante que **Solve** si l'équation à résoudre possède des paramètres.

6.5 Remarque

Pour que **Solve** et **Reduce** retournent des solutions exactes, il est nécessaire que l'équation soit définie avec des coefficients exactes et non des nombres à virgule flottantes.



Approximation des solutions

Nous allons dans cette sous-section comment *approximer* les solutions des équations que nous n'avons pu résoudre exactement avec **Solve** et **Reduce**. Ceci se fera avec **NSolve** et **FindRoot**.

Méthode NSolve

La méthode **NSolve** est spécialisée dans la résolution des *équations algébriques*. Elle livrera une approximation de *toutes* les solutions.

```
In[26]:= NSolve [eqnA2 , x]
```

```
Out[26]= {{x -> -1.19386}, {x -> -0.15459 - 0.828074i}, {x -> -0.15459 + 0.828074i},
          {x -> 0.751519 - 0.784616i}, {x -> 0.751519 + 0.784616i}}
```

Nous pouvons préciser que nous ne nous intéressons qu'aux solutions réelles :

```
In[27]:= NSolve[eqnA2, x, Reals]
```

```
Out[27]= {x -> -1.19386}
```

```
In[28]:= NSolve[eqnT2, x]
```

```
NSolve: This system cannot be solved with the methods available to
NSolve.
```

```
Out[28]= NSolve[Cos[x]==Log[x], x]
```

Sans surprise, `NSolve` ne résout pas le deuxième exemple d'équation transcendante.

Comparaison Solve NSolve

Au travers d'un exemple, nous allons voir dans cette sous-section que pour une équation algébrique `eqn` à coefficients exacts d'inconnue `inc`, les instructions `Solve[eqn, inc]`, `Solve[N[eqn], inc]` (l'instruction `N[eqn]` transforme les coefficients de l'équation en nombres à virgule flottante) et `NSolve[eqn, inc]` ne sont pas toujours équivalentes.

6.6 Exemple

Nous cherchons les zéros de la fonction

$$f(t) = \left(\frac{t}{100} + 1\right)^{11} - \left(\frac{t+1}{100} + 1\right)^9.$$

Définissons cette fonction et l'équation à résoudre dans Mathematica puis tentons de calculer ses solutions exactes :

```
In[29]:= Clear["Global`*"]
f[t_] := (1 + t/100)^11 - (1 + (t + 1)/100)^9
eqn = f[t] == 0;
Solve[eqn, t]
```

```
Out[29]= {{t -> -195...}, {t -> -102...}, {t -> 4.38...}, {t -> -101... - 0.443... i},
          {t -> -101... + 0.443... i}, {t -> -101... - 0.336... i},
          {t -> -101... + 0.336... i}, {t -> -101... - 0.193... i},
          {t -> -101... + 0.193... i}, {t -> -101... - 0.0625... i},
          {t -> -101... + 0.0625... i}}
```

Un essai permettant de voir que `ToRadicals` n'a aucun effet, nous essayons différentes méthodes permettant d'obtenir des approximations numériques plus précises :

- (a) Nous commençons par demander à Mathematica une approximation numérique de la sortie précédente :

```
In[30]:= solSolve = Solve[eqn, t]//N
```

```
Out[30]= {{t -> -195.367}, {t -> -101.675}, {t -> 4.38386},
          {t -> -101.211 - 0.442553i}, {t -> -101.211 + 0.442553i},
          {t -> -100.912 - 0.335664i}, {t -> -100.912 + 0.335664i},
          {t -> -100.795 - 0.193031i}, {t -> -100.795 + 0.193031i},
          {t -> -100.753 - 0.0625426i}, {t -> -100.753 + 0.0625426i}}
```

Mathematica retourne 11 solutions dont trois sont réelles.

- (b) Examinons ce que retourne NSolve :

```
In[31]:= solNSolve = NSolve[eqn, t]
```

```
Out[31]= {{t -> -195.367}, {t -> -104.695}, {t -> -103.854}, {t -> -102.589},
          {t -> -102.263}, {t -> -101.032}, {t -> -99.724}, {t -> -98.7377},
          {t -> -97.8905}, {t -> -96.9605}, {t -> 4.38386}}
```

Nous constatons que NSolve[...] ne produit pas les mêmes résultats que Solve[...]//N en particulier une des solutions réelles précédentes (-101.675) n'a pas été trouvée ! Évaluons f pour tester si les résultats fournis correspondent effectivement à des solutions :

```
In[32]:= f[t] /. solNSolve
```

```
Out[32]= {3.64897 · 10-12, 1.25996 · 10-13, 1.22736 · 10-14, 6.09925 · 10-17,
          7.37498 · 10-18, -1.41906 · 10-22, -8.96478 · 10-18, -1.55243 · 10-15,
          -2.71794 · 10-14, -2.8637 · 10-13, 4.44089 · 10-16}
```

Les résultats sont proches de 0 mais d'ordres de grandeur très différents ce qui ne permet pas de conclure. Évaluons f un peu avant et un peu après chaque solution et regardons si un changement de signe se produit :

```
In[33]:= test[t_] := {t, Sign[f[0.9999 t]] Sign[f[1.0001 t]]}
test[t] /. solNSolve
```

```
Out[33]= {{-195.367, -1}, {-104.695, 1}, {-103.854, 1}, {-102.589, 1},
          {-102.263, 1}, {-101.032, 1}, {-99.724, 1}, {-98.7377, 1},
          {-97.8905, 1}, {-96.9605, 1}, {4.38386, -1}}
```

Nous constatons qu'il n'y a que deux solutions qui sont correctes (celles-ci avaient été trouvées par Solve). Pour trouver toutes les solutions qui avaient été fournies par Solve, il faut demander à Mathematica de travailler avec plus que les 16 chiffres significatifs usuels :

```
In[34]:= NSolve[eqn, t, WorkingPrecision -> 20]
```

```

Out[34]= {{t -> -195.36723135721455452}, {t -> -101.67514600695975491},
          {t -> -101.210696348455808356 - 0.442552630837044392i},
          {t -> -101.210696348455808356 + 0.442552630837044392i},
          {t -> -100.911954497475663938 - 0.335664276489927917i},
          {t -> -100.911954497475663938 + 0.335664276489927917i},
          {t -> -100.795349995808433264 - 0.193031483861710781i},
          {t -> -100.795349995808433264 + 0.193031483861710781i},
          {t -> -100.752742710771231564 - 0.062542554380698078i},
          {t -> -100.752742710771231564 + 0.062542554380698078i},
          {t -> 4.3838644691965836774}}

```

Dans le cas où nous cherchons uniquement les zéros réels de f , le problème précédent peut être évité en donnant cette restriction à Mathematica :

```
In[35]:= solNSolve = NSolve[eqn, Reals]
```

```
Out[35]= {{t -> -195.367}, {t -> -101.675}, {t -> 4.38386}}
```

Testons les solutions trouvées :

```
In[36]:= test[t] /. %
```

```
Out[36]= {{-195.367, -1}, {-101.675, -1}, {4.38386, -1}}
```

- (c) Examinons ce que retourne `Solve` si l'équation contient des nombres à virgule flottante :

```
In[37]:= Solve[eqn//N, t]
```

```
Out[37]= {{t -> -195.367}, {t -> -104.675}, {t -> -103.82}, {t -> -102.548},
          {t -> -102.276}, {t -> -101.053}, {t -> -99.7264}, {t -> -98.7497},
          {t -> -97.8789}, {t -> -96.9629}, {t -> 4.38386}}
```

Il y a les mêmes erreurs qu'avec `NSolve` mais l'option `WorkingPrecision` n'a aucun effet ici :

```
In[38]:= Solve[eqn//N, t, WorkingPrecision -> 30]
```

```
Out[38]= {{t -> -195.367}, {t -> -104.675}, {t -> -103.82}, {t -> -102.548},
          {t -> -102.276}, {t -> -101.053}, {t -> -99.7264}, {t -> -98.7497},
          {t -> -97.8789}, {t -> -96.9629}, {t -> 4.38386}}
```

Pour que l'option `WorkingPrecision` ait un effet il faut travailler avec `NSolve` :

```
In[39]:= NSolve[eqn // N, t, WorkingPrecision -> 20]
```

```

NSolve::precw: The precision of the argument function
({0. + (1. + 0.01t)11 - 1.(1. + 0.01(1. + t))9}) is less than
WorkingPrecision (20).

```

```
Out[39]= {{t -> -195.36723135721455452}, {t -> -101.67514600695975491},
  {t -> -101.210696348455808356 - 0.442552630837044392i},
  {t -> -101.210696348455808356 + 0.442552630837044392i},
  {t -> -100.911954497475663938 - 0.335664276489927917i},
  {t -> -100.911954497475663938 + 0.335664276489927917i},
  {t -> -100.795349995808433264 - 0.193031483861710781i},
  {t -> -100.795349995808433264 + 0.193031483861710781i},
  {t -> -100.752742710771231564 - 0.062542554380698078i},
  {t -> -100.752742710771231564 + 0.062542554380698078i},
  {t -> 4.3838644691965836774}}
```



Résumons les constations faites dans l'exemple précédent :

RÉSOLUTION NUMÉRIQUES D'UNE ÉQUATION ALGÈBRIQUE

Pour optimiser la résolution, il faut...

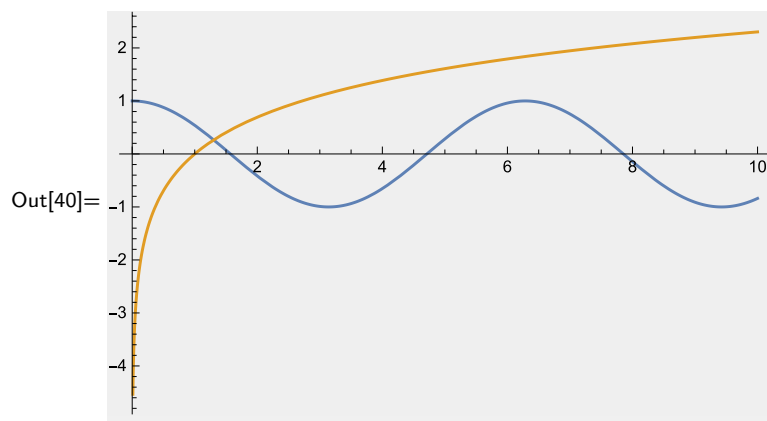
- ... éviter lorsque cela est possible l'utilisation de nombres à virgule flottante,
- ... préférer `Solve[...]` // `N` lorsque les coefficients sont exacts et `NSolve` lorsque ce sont des nombres à virgule flottante.
- ... utiliser le cas échéant l'option `Reals`,
- ... utiliser si nécessaire l'option `WorkingPrecision` pour améliorer les performances de `NSolve`.



Méthode `FindRoot[eqn,x,x0]`

La méthode `FindRoot[eqn,{x,x0}]` est une variation de la méthode de Newton. Pour que Mathematica retourne une approximation d'une solution de l'équation `eqn` dont l'inconnue est `x`, nous devons donner une approximation de départ `x0`. Appliquons cette méthode au deuxième exemple d'équation transcendante. Pour cela, il nous faut trouver une approximation de départ, approximation que nous allons obtenir graphiquement :

```
In[40]:= Plot[Cos[x], Log[x]], {x, 0, 10}]
```



Nous pouvons donc prendre la valeur 1 comme approximation de départ :

```
In[41]:= FindRoot[eqnT2, {x, 1}]
```

```
Out[41]= {x -> 1.30296}
```

Vérifions notre solution :

```
In[42]:= {Cos[x], Log[x]}/.%
```

```
Out[42]= {0.264642, 0.264642}
```

Il est important d'avoir une approximation de notre solution qui soit suffisamment proche de la solution sinon Mathematica n'arrive pas à trouver la solution :

```
In[43]:= FindRoot[eqnT2, {x, 4}]
```

```
FindRoot: The line search decreased the step size to within tolerance
specified by AccuracyGoal and PrecisionGoal but was unable to
find a sufficient decrease in the merit function. You may need
more than MachinePrecision digits of working precision to meet
these tolerances.
```

```
Out[43]= {x -> 6.11902}
```

Mathematica propose une solution mais avant celle-ci il retourne un message indiquant qu'il n'a pas réussi à atteindre la précision attendue. Vérifions-le :

```
In[44]:= {Cos[x], Log[x]}/.%
```

```
Out[44]= {0.986556, 1.8114}
```

Nous sommes effectivement très loin d'avoir une égalité !

Méthode FindRoot[eqn, x, x0, x1]

Pour certaines équations coriaces, il peut être utile de donner deux approximations x_0 et x_1 de la solutions. C'est ce que fait la méthode `FindRoot[eqn, {x, x0, x1}]` qui est une

variation de la méthode de la sécante (voir [3.3](#)). Appliquons cette méthode au deuxième exemple d'équation transcendante.

```
In[45]:= FindRoot[eqnT2,{x,4,5}]
```

```
Out[45]= x -> 1.30296
```

Bien que les deux approximations de départ soient mauvaises, Mathematica arrive tout de même à trouver la solution.

7 Éléments de programmation Mathematica

Le but de cette section est principalement de fournir les éléments de programmation Mathematica permettant d'implémenter la méthode de la bisection.

If et Which

La fonction `If[condition,v,f]` est utilisée pour que Mathematica effectue des instructions différentes suivant la situation proposée. Plus précisément, le paramètre `condition` est une expression booléenne (c'est-à-dire qui prend les valeurs vrai ou faux), `v` est l'instruction que est effectuée si `condition` est vraie tandis que `f` est celle qui est effectuée si `condition` est fausse. Cette fonction peut être utilisée, par exemple, pour définir une fonction en deux morceaux :

7.1 Exemple

Soit la fonction f définie sur $\mathbb{R}$ de la façon suivante :

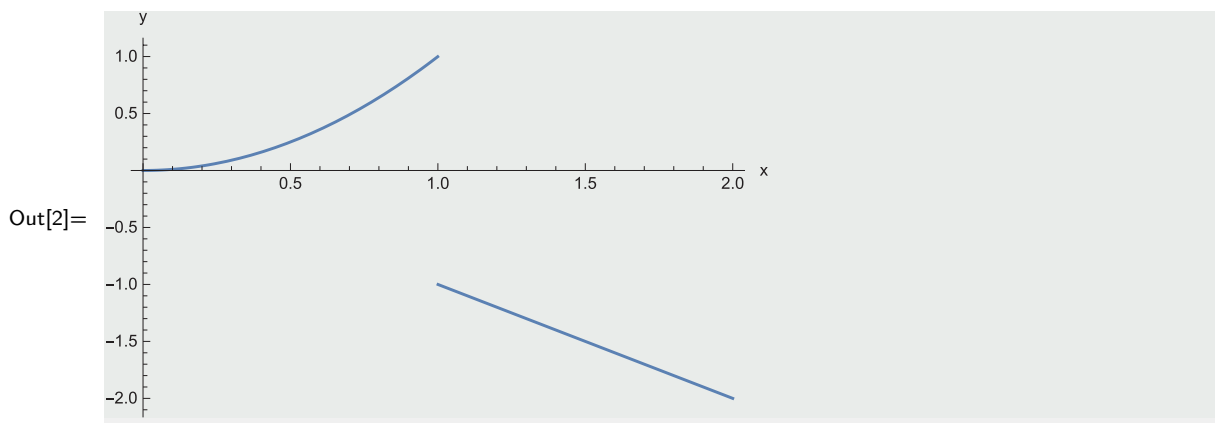
$$f(x) = \begin{cases} x^2 & \text{si } x \leq 1 \\ -x & \text{si } x > 1 \end{cases}$$

Nous définissons dans Mathematica f à l'aide de la fonction `If` : A titre de vérification, traçons le graphe de f :

```
In[1]:= f[x_] := If[x <= 1, x^2, -x]
```

A titre de vérification, traçons le graphe de f :

```
In[2]:= Plot[f[x], {x, 0, 2}, AxesLabel -> {"x", "y"}]
```



☺

Si le nombre d'alternatives à traiter est supérieur à deux, l'instruction `Which[condition1, valeur1, condition2, valeur2, ...]` sera préférée. Celle-ci teste la première condition, si celle-ci est vérifiée c'est la première valeur qui est retournée sinon c'est la seconde valeur qui est testée et ainsi de suite.

7.2 Exemple

Nous voulons implémenter la fonction qui retourne le signe d'un nombre réel :

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{si } x < 0, \\ 0 & \text{si } x = 0, \\ -1 & \text{si } x > 0. \end{cases}$$

```
In[3]:= sgn[x_] := Which[
  x < 0, -1,
  x == 0, 0,
  x > 0, 1
]
Map[sgn, {-10, 0, 7}] (*test*)
```

```
Out[3]= {-1, 0, 1}
```

☺

Nest et NestWhile

Il a été vu au cours de 2^e année que la fonction `NestList[f, expr, n]` retourne la liste des résultats de l'application de `f` à `expr` de 0 à `n` fois. La fonction `Nest[f, expr, n]` permet de ne retourner que le dernier terme de cette liste, c'est-à-dire le résultat de `n` applications de `f` à `expr`.

7.3 Exemple

Soit la suite $(a_n)_{n \in \mathbb{N}}$ définie de la façon suivante :

$$a_0 = 3, \quad a_{n+1} = \frac{2 + a_n^2}{2a_n}.$$

Donnons la liste des termes a_0 jusqu'à a_{10} puis définir la fonction `a[n_]` donnant la valeur de a_n (donner les résultats sous forme de code à virgule) :

```
In[4]:= f[x_] := (2 + x^2)/(2 x)
NestList[f, 3., 10]
```

```
Out[4]= {8., 4.125, 2.30492, 1.58632, 1.42355, 1.41424, 1.41421,
  1.41421, 1.41421, 1.41421, 1.41421}
```

Nous définissons maintenant une fonction permettant de calculer directement le n -ème terme :

```
In[5]:= a[n_] := Nest[f, 3., n]
```

Testons en demandant à Mathematica de calculer a_4 :

```
In[6]:= a[4]
```

```
Out[6]= 1.42355
```

☺

Plutôt que de spécifier le nombre de fois qu'une fonction doit être appliquée à une expression, on peut imposer une condition sur le nombre d'applications. Ceci se fait avec la fonction `NestWhile[f,expr,test]` qui applique la fonction `f` à `expr` tant que ("while" en anglais) la valeur de la fonction booléenne `test` appliquée au dernier résultat est vrai. Mathematica retourne alors ce dernier résultat.

7.4 Exemple

Déterminons la premier terme qui est inférieur à 2 dans la suite $(a_n)_{n \in \mathbb{N}}$ définie de la façon suivante :

$$a_0 = 1000, \quad a_{n+1} = \sqrt{a_n}.$$

```
In[7]:= test[x_] := x >= 2
NestWhile[Sqrt, 1000., test]
```

```
Out[7]= 1.53993
```

Pour vérifier que le résultat est correct, demandons à Mathematica de calculer le terme jusqu'à a_5 de cette suite :

```
In[8]:= NestList[Sqrt, 1000., 5]
```

```
Out[8]= {1000., 31.6228, 5.62341, 2.37137, 1.53993, 1.24094}
```

☺

Module

Lorsqu'on désire que plusieurs instructions soit effectuées successivement par Mathematica, on peut les regrouper dans un module dont la structure est la suivantes : `Module[{var1,var2,...},expr1;expr2;...]`. `var1,var2,...` sont les variables utilisées à l'intérieure du module. Si une de ces variables a déjà été définie à l'extérieur du module, Mathematica en fait abstraction lorsqu'on se trouve à l'intérieur du module et, inversement, Mathematica ne tiendra pas compte des définitions de ces variables faites à l'intérieur du module lorsqu'on se trouve à l'extérieur du module : on dit que ces variables sont des variables *locales*. `expr1,expr2,...` est la suite d'expressions à évaluer et Mathematica n'affichera que le résultat de l'évaluation de la dernière expression (si l'affichage d'un résultat intermédiaire est désiré, il faut insérer une instruction `Print[resultat]`).

7.5 Exemple

Connaissant le nombre de côtés d'un polygone régulier ainsi que le rayon de son cercle circonscrit, nous pouvons définir un module calculant le périmètre de ce polygone.

```

Clear["Global`*"]
perimetrePolygone[r_, n_] := Module[
  {a, c, p},
  a = 2 Pi /n; (*angle au centre*)
  c = 2 r Sin[a/2]; (*côté*)
  p = n c; (*périmètre*)
  Print[
    "Périmètre d'un polygone à ", n,
    " côtés inscrit dans un cercle de rayon ", r, " : "
  ];
  p
]

```

Comme vérification, demandons à Mathematica de calculer le périmètre d'un hexagone inscrit dans un cercle de rayon 10 :

```

In[10]:= perimetrePolygone[10, 6]
         {a, c, p}

```

```

Out[10]= Périmètre d'un polygone à 6 côtés inscrit dans un cercle de
         rayon 10 :

```

```

Out[11]= 60

```

```

Out[12]= {a, c, p}

```

Nous constatons qu'aucune valeur n'a été affectée en dehors du module aux variables **a**, **c** et **p**. ☺

8 Exercices

Mise en équation

Pour tous les exercices de cette sous-section, on demande de mettre en équations chacun des problèmes ci-dessous puis de classer chaque équation obtenue dans l'une des trois catégories suivantes :

- (I) L'équation peut être résolue par l'algèbre élémentaire (par exemple, une équation bicarrée).
- (II) L'équation pourrait être résolue par radicaux, par exemple, une équation polynomiale de degré 3 ou 4. Mais, comme cela réclamerait un effort particulier, il peut être plus commode d'utiliser une méthode numérique.
- (III) La solution de l'équation ne peut pas être exprimée au moyen de fonctions élémentaires (par exemple, $\cos(x) = x$ ou bien une équation polynomiale de degré supérieur ou égal à 5). Une méthode numérique est indispensable.

On résoudra immédiatement les équations de la catégorie (I). Par contre, la résolution des équations des catégories (II) et (III) est reportée après l'étude des sections suivantes.

Exercice 1 (sans Mathematica)

D'un cône droit, on donne sa hauteur h et son aire totale A . Calculer son rayon r .

Exercice 2 (sans Mathematica)

D'un triangle rectangle, on connaît son aire A et la projection orthogonale p d'une cathète sur l'hypoténuse. Calculer les grandeurs des trois côtés a, b, c .

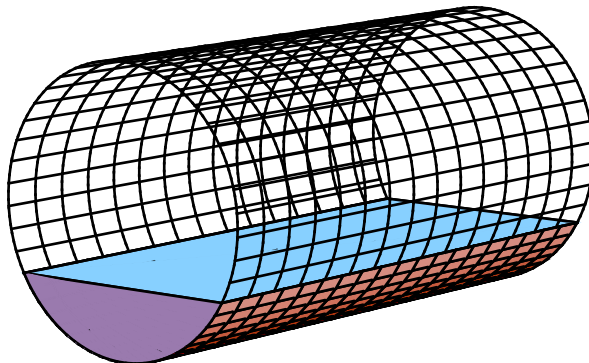
Indication : poser comme quatrième inconnue q , la projection orthogonale de l'autre cathète sur l'hypoténuse puis déterminer l'équation de paramètres A et p que q doit satisfaire et finalement exprimer a, b, c en fonction de A, p et q .

Exercice 3 (sans Mathematica)

Une sphère homogène de rayon r et de masse volumique ρ_1 flotte sur un liquide de masse volumique ρ_0 . Calculer la hauteur h de la partie émergente.

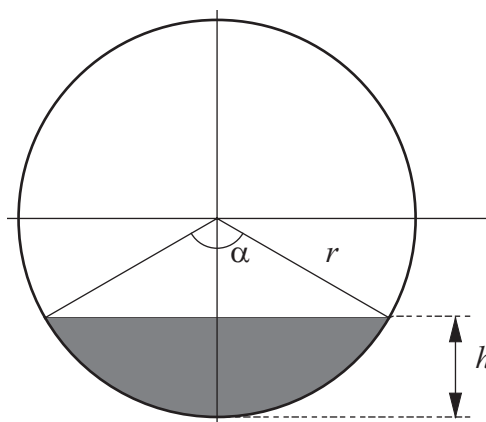
Exercice 4 (sans Mathematica)

On veut graduer une jauge pour une cuve cylindrique horizontale comme celle de la figure suivante :



On connaît les dimensions de la cuve : son rayon r et sa longueur L . Notons C sa capacité, V le volume de liquide que contient effectivement la cuve et $t = \frac{V}{C}$ son taux de remplissage (par exemple $t = 0.3$ signifie que la cuve est remplie à 30%). Sachant que la cuve a un taux de remplissage t donné, calculer la hauteur h du liquide (c'est-à-dire l'endroit où il faut écrire 30% sur la jauge).

Indication : calculer d'abord l'angle au centre α de la figure ci-dessous.

**Exercice 5 (sans Mathematica)**

Un montant c est emprunté à une banque. Afin de rembourser cet emprunt, la banque demande de verser n annuités a . Calculer le taux d'intérêt i qu'a demandé la banque sachant qu'en Suisse le taux est plafonné par la loi à 18 %.

Indication : s'inspirer de ce qui a été fait dans l'activité d'introduction.

Méthode graphique

Exercice 6 (avec Mathematica)

Dans le problème 2, estimer graphiquement la valeur de la projection q pour une aire $A = 1000 \text{ cm}^2$ et une projection $p = 20 \text{ cm}$.

Encadrer chaque solution q entre deux entiers. Séparer les solutions obtenues (si nécessaire).

Expliquer pourquoi il n'y a pas d'autre solution.

Exercice 7 (avec Mathematica)

Dans le problème 3, estimer graphiquement la hauteur h , pour un rayon $r = 0.3 \text{ m}$ et des masses volumiques $\rho_0 = 1000 \frac{\text{kg}}{\text{m}^3}$ et $\rho_1 = 900 \frac{\text{kg}}{\text{m}^3}$.

Encadrer chaque solution entre deux entiers. Séparer les solutions obtenues (si nécessaire).

Expliquer pourquoi il n'y a pas d'autre solution.

Exercice 8 (avec Mathematica)

Dans le problème 4, pour $t = 30\%$ et $r = 1$, estimer graphiquement la valeur de l'angle α .

Encadrer chaque solution α entre deux nombres arrondis au dixième de radian. Séparer les solutions obtenues (si nécessaire).

Expliquer pourquoi il n'y a pas d'autre solution.

Exercice 9 (avec Mathematica)

Dans le problème 5, pour $c = 10000$, $n = 8$, $a = 2000$, estimer graphiquement la valeur du taux i .

Encadrer chaque solution i entre deux nombres entiers de pourcents. Séparer les solutions obtenues (si nécessaire).

Expliquer pourquoi il n'y a pas d'autre solution.

Méthode de la bisection

Exercice 10 (sans Mathematica)

Au moyen de la méthode de la bisection, résoudre l'équation

$$2^x = 10$$

avec une erreur absolue $\Delta x = 0.01$.

Exercice 11 (sans Mathematica)

Soit $f : [a; b] \rightarrow \mathbb{R}$ une fonction continue avec $\text{sgn}(f(a)) \cdot \text{sgn}(f(b)) < 0$ et un unique zéro. Calculer le nombre de bisections à effectuer (sans arrondir les bornes) pour que l'erreur absolue initiale sur l'estimation du zéro de f soit au moins divisée par 10^8 .

Exercice 12 (avec Mathematica)

Avec les données $A = 1000 \text{ cm}^2$, $p = 20 \text{ cm}$, résoudre numériquement le problème 2 (prendre l'encadrement déterminé à l'exercice 6). Pour cela procéder de la façon suivante en utilisant Mathematica comme calculatrice :

- Au moyen de la méthode de la bisection, calculer la valeur numérique de q à la précision de ± 0.02 .
- Calculer ensuite a, b, c .

Exercice 13 (avec Mathematica)

Le but de cet exercice est de programmer une fonction `bisection[f, {a, b}, nbIt]` qui retourne l'intervalle obtenu après avoir fait `nbIt` bisections de l'intervalle $[a; b]$ pour une fonction `f` sachant que `f` et $[a; b]$ vérifient les hypothèses d'utilisation de la méthode de la bisection. Pour cela, suivre la marche à suivre suivante

- Définir la fonction $g(x) = x^3 - 4x^2 - 20x + 1$ qui permettra de tester le bon fonctionnement du programme en comparant les résultats avec ceux de l'exemple 2.1.
- A l'aide des fonctions `If` et `Sign`, définir une fonction `biss[{a_}, {b_}]` qui retourne l'intervalle obtenu après avoir fait une bisection de l'intervalle $[a; b]$ avec la fonction g .
- A l'aide de la fonction `NestList`, donner les intervalles obtenus en faisant 5 bisections des intervalles $[-3, -2]$ et $[0; 1]$ avec la fonction g .
- Définir un module `bisectionList[f, {a, b}, nbIt]` qui retourne la liste des intervalles obtenus après avoir fait `nbIt` bisections de l'intervalle $[a; b]$ avec une fonction f . Vérifier votre module en calculant les intervalles obtenus en faisant 5 bisections de l'intervalle $[0; 1]$ avec la fonction g .
- Modifier le module précédent en utilisant la fonction `Nest` afin d'obtenir la fonction désirée.
- Calculer l'intervalle obtenu après 20 bisections de l'intervalle $[0; 1]$ avec la fonction g .

Méthode regula falsi**Exercice 14 (avec Mathematica)**

Soit la fonction $f : \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = 8x^3 + 16x^2 - 5$.

- Déterminer graphiquement un premier encadrement du plus grand zéro négatif de f . On demande un encadrement dont les bornes sont entières.
- Tracer à l'aide de Mathematica le graphe de f sur cet intervalle, imprimer ce graphe et estimer graphiquement un deuxième et un troisième encadrement à l'aide de la méthode regula falsi.
- Calculer avec Mathematica la valeur des bornes du quatrième encadrement en partant de la valeur de l'encadrement initial.

Exercice 15 (sans Mathematica)

Soit $f : [a; b] \longrightarrow \mathbb{R}$ une fonction continue avec $f(a) \cdot f(b) < 0$.

- Déterminer l'équation de la sécante au graphe de f qui passe par les deux points $(a; f(a))$ et $(b; f(b))$.
- Déduire de l'équation précédente que cette droite coupe l'axe des x en $x_1 = \frac{af(b)-bf(a)}{f(b)-f(a)}$.

Exercice 16 (avec Mathematica)

Au moyen de la méthode de la regula falsi, résoudre numériquement l'exercice 4 avec les données suivantes : $t = 0.3$, $r = 1$ (prendre l'encadrement déterminé à l'exercice 8)

Calculer α avec une erreur absolue $\Delta\alpha = 10^{-5}$ puis calculer h .

Exercice 17 (avec Mathematica)

Programmer un module `regulafalsiListe[f,{a,b},n]` qui retourne l'intervalle de départ ainsi que les n intervalles fournis par la méthode regula falsi lorsqu'elle est appliquée à une fonction $f : [a; b] \longrightarrow \mathbb{R}$.

Tester votre module avec l'exemple 3.1 de la théorie.

Exercice 18 (avec Mathematica)

Soit l'équation $x^{(x^x)} - 2 = 0$.

- Déterminer graphiquement un encadrement de la (des) solution(s) de cette équation. On demande un (des) encadrement(s) dont les bornes sont entières.
- Déterminer l'intervalle fourni par la méthode regula falsi après 10 itérations (utiliser le module défini à l'exercice précédent). Que constatez-vous ?

Méthode du point fixe**Exercice 19 (avec Mathematica)**

- On donne la fonction d'itération g et la valeur de démarrage x_0 suivante :

$$g(x) = \sqrt{1+x}, \quad x_0 = 2.$$

- Faire un graphique contenant les graphes des fonctions $y = x$ et g (afin de faciliter le travail ajouter un quadrillage au graphique avec l'option `GridLines -> Automatic`) puis estimer graphiquement à la main les cinq premières itérations.
 - Calculer les dix premières itérations à l'aide de Mathematica.
 - Si la suite converge vers un point fixe r , de quelle équation r est-il solution ?
- Idem pour $g(x) = x^2$ et $x_0 = \frac{1}{2}$.
 - Idem pour $g(x) = x^2$ et $x_0 = 1.5$.
 - Idem pour $g(x) = \frac{1}{x}$ et $x_0 = \frac{1}{2}$.
 - A partir des différents point de cette exercice énoncer des hypothèses garantissant la convergence de la méthode du point fixe.

Exercice 20 (sans Mathematica)

Soit f une fonction réelle et r un élément de l'ensemble de définition de f . Démontrer que r est une solution de l'équation $f(x) = 0$ si et seulement si r est un point fixe de la fonction $h(x) = x + f(x)$.

Exercice 21 ($\oplus$ sans Mathematica)

Soit la fonction $g : \mathbb{R} \rightarrow \mathbb{R}$, $g(x) = -x^3 + x$ et r son point fixe.

- (a) Calculer r et $g'(r)$
- (b) Déterminer le plus grand intervalle J tel que $|g'(x)| < 1$ pour tout $x \in J \setminus \{r\}$.
- (c) A l'aide du théorème des accroissements finis, montrer que $|g(x) - g(r)| < |x - r|$ pour tout $x \in J \setminus \{r\}$.
- (d) Montrer que la méthode du point fixe converge pour toute valeur initiale $x_0 \in J$.

Exercice 22 (sans/avec Mathematica)

Soit l'équation $1 - \frac{1}{2}x^2 = 0$.

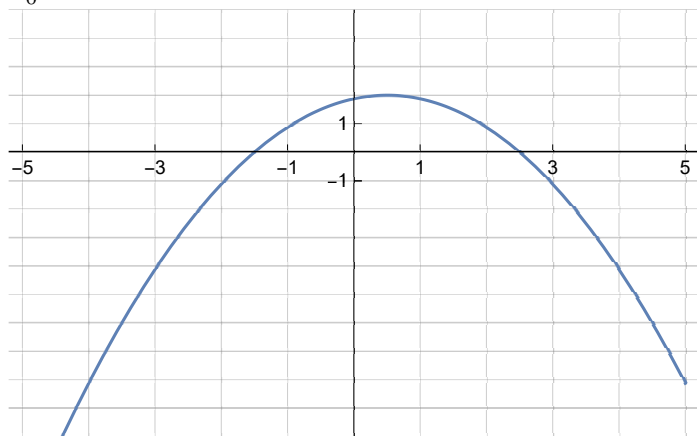
- (a) Approximer au centième, sans Mathematica, la solution positive de cette équation avec la méthode du point fixe.
- (b) Déterminer avec Mathematica la solution positive de cette équation avec la méthode du point fixe. Pour cela, itérer jusqu'à ce que l'évaluation du membre de gauche de l'équation avec l'approximation fournie donne une valeur absolue inférieure à 10^{-10} (utiliser `NestWhile`).
- (c) Discuter avec Mathematica la convergence la méthode du point fixe en fonction de la valeur initiale dans le cas où la fonction d'itération qui a été choisie est $g(x) = -\frac{x^2}{2} + x + 1$.

Méthode de Newton

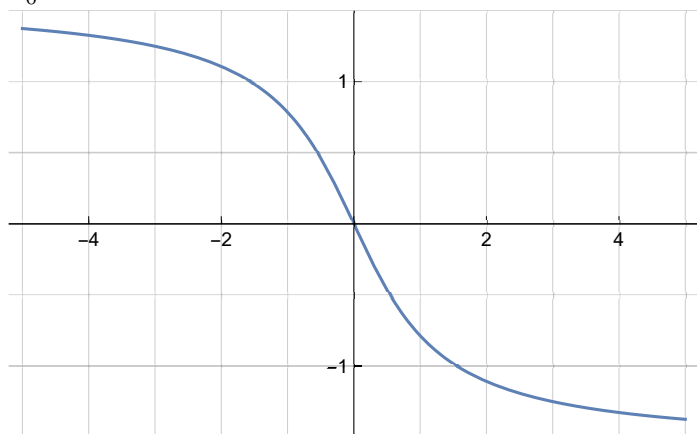
Exercice 23 (sans Mathematica)

Estimer graphiquement les itérations x_1 , x_2 et x_3 générées par la méthode de Newton avec la valeur x_0 donnée.

(a) $x_0 = 1$



(b) $x_0 = -1.5$



Exercice 24 (sans/avec Mathematica)

Soit l'équation

$$x^3 = 3 - x.$$

- Déterminer le nombre de solutions sans Mathematica.
- Déterminer un encadrement à bornes entières de la(des) solution(s) sans Mathematica.
- Au moyen de la méthode de Newton, déterminer une valeur numérique de la(des) solution(s) à 6 chiffres significatifs sans Mathematica puis vérifier avec Mathematica.

Exercice 25 (sans/avec Mathematica)

On considère l'équation

$$3^x = 7 - x.$$

- (a) Sans Mathematica, mettre l'équation sous la forme $f(x) = 0$ puis montrer que l'intervalle $[1; 3]$ contient une *unique* solution.
- (b) A partir de Mathematica, imprimer le graphique de la fonction f sur l'intervalle $[1; 3]$. En partant de la valeur de démarrage $x_0 = 2.5$, construire graphiquement à la main (du point de vue de l'équation $f(x) = 0$) la première approximation x_1 donnée par la méthode de Newton. Construire aussi x_2 .
- (c) Calculer avec Mathematica et la méthode de Newton la solution de cette équation avec une précision de 6 chiffres significatifs
- (d) A partir de Mathematica, imprimer le graphique sur l'intervalle $[1; 3]$ de la fonction d'itération g correspondant à la méthode de Newton. En partant de la valeur de démarrage $x_0 = 2.5$, construire graphiquement à la main (du point de vue de l'équation $g(x) = x$) les approximations x_1 et x_2 données par la méthode de Newton.

Exercice 26 (sans/avec Mathematica)

Soit l'équation

$$\arctan(x) = 0.$$

- (a) Sans Mathematica, montrer qu'il n'existe pas de d'intervalle $[a; b]$, $a < 0 < b$ satisfaisant les hypothèses de Fourier.
- (b) Discuter la convergence de méthode de Newton en faisant à partir de quelques essais faits avec Mathematica.
- (c) Estimer graphiquement l'intervalle J contenant 0 tel que la méthode de Newton converge pour toute valeur initiale $x_0 \in J$.

Exercice 27 (sans/avec Mathematica)

- (a) Soit la fonction $f(x) = x^2$. Sans Mathematica,

- (i) montrer que $f'(0) = 0$,

- (ii) étudier la convergence de la méthode de Newton vers 0.

- (b) Idem pour la fonction suivante mais avec Mathematica :

$$f(x) = \begin{cases} x^3 \sin\left(\frac{1}{x^2}\right) & \text{si } x \neq 0, \\ 0 & \text{si } x = 0. \end{cases}$$

Exercice 28 (sans/avec Mathematica)

Soit la fonction $f(z) = z^3 - 1, z \in \mathbb{C}$.

- (a) Sans Mathematica, calculer les zéros de f .
- (b) La méthode de Newton est aussi applicable à des fonctions $\mathbb{C}$ -différentiables⁸, ce qui est le cas de la fonction f . Pour de telles fonctions les règles de dérivation sont les mêmes que pour les fonctions réelles.

Compte tenu des remarques ci-dessus, dessiner avec Mathematica d'une même couleur les régions de $D = \{x + iy : |x| \leq 1, |y| \leq 1\}$ contenant les valeurs initiales de l'itération de la méthode de Newton convergeant vers un même zéro de f (on parle de **bassin d'attraction**). Pour cela procéder de la façon suivante :

- (i) Définir une primitive graphique⁹ `carre[{x,y},c]` qui permet d'insérer un carré de côté `c` centré en `{x,y}` dans un graphique Mathematica. Par exemple, la commande `Graphics[{Red, carre[{1,2}, 0.5]}]` retourne un carré rouge centré en $(1; 2)$ de côté 0.5.
- (ii) Définir une liste contenant les zéros de f sous la forme de nombres à virgule flottantes ainsi que la fonction d'itération de la méthode de Newton de façon à ce que l'itération se fasse avec des nombres à virgule flottante.
- (iii) Définir un module `couleur[{x,y}]` qui définit une couleur en fonction du zéro de f vers lequel converge la méthode de Newton lorsque $x + iy$ est choisi comme valeur initiale. Pour cela calculer une *approximation numérique* de z_{10} , le 10^e itéré de la méthode de Newton obtenu avec $z_0 = x + iy$ comme valeur de départ, puis déterminer le zéro r la plus proche de z_{10} , c'est-à-dire celui avec $|z_{10} - r|$ minimal. Comme il y a plusieurs cas à distinguer, préférer la fonction `Which` à la fonction `If` et prêter attention au cas particulier $(x; y) = (0; 0)$.

Faire des tests avec des points du plan complexe proches de chacun des zéros de la fonction f .

- (iv) Décomposer D en n^2 carrés de côté $\frac{2}{n}$ auxquels sont attribués des couleurs à l'aide du module `couleur`. Pour tester votre code, commencer avec de très petites valeurs de n (les machines des l'école sont en difficultés pour des valeurs de n supérieur à 200).

8. Une fonction $f : \mathbb{C} \rightarrow \mathbb{C}$ est $\mathbb{C}$ -différentiable (on parle généralement de fonction **holomorphe**) si la limite *complexe* $\lim_{h \rightarrow 0} \frac{f(z+h) - f(z)}{h}$ existe et est finie.

9. Une primitive graphique est une fonction qui permet d'insérer un objet dans un graphique Mathematica. Parmi les primitives graphiques couramment utilisées citons `Point`, `Line`, `Circle` et `Rectangle`. L'affichage de primitives graphiques se fait avec la fonction `Graphics`. A une primitive peut être ajoutée une directive donnant des indications sur l'affichage d'un objet (par exemple sa couleur). L'affichage de plusieurs primitive se fait alors avec une commande de la forme `Graphics[{directive1, primitive1}, {directive2, primitive2}, ...]`.

Résolution avec Mathematica

Exercice 29 (avec Mathematica)

D'un cône, on donne sa hauteur h et son aire totale A .

Calculer son rayon r avec Mathematica (il s'agit de l'exercice 1).

Indications :

- Il s'agit d'un calcul littéral.
- Complétez l'équation par une condition afin de n'obtenir qu'une seule réponse.

Exercice 30 (avec Mathematica)

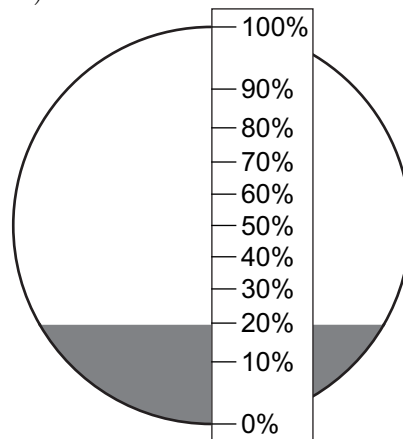
D'un triangle rectangle, on connaît son aire $A = 975 \text{ cm}^2$ et la projection orthogonale $p = 20 \text{ cm}$ d'une cathète sur l'hypoténuse. Calculer les longueurs des trois côtés a , b et c avec Mathematica (il s'agit d'un cas particulier de l'exercice 2).

Exercice 31 (avec Mathematica)

Une sphère homogène de rayon $r = 0.3 \text{ m}$, de masse volumique $\rho_1 = 900 \frac{\text{kg}}{\text{m}^3}$ flotte sur un liquide de masse volumique $\rho_0 = 1000 \frac{\text{kg}}{\text{m}^3}$. Calculer la hauteur h de la partie émergente avec Mathematica (il s'agit d'un cas particulier de l'exercice 3).

Exercice 32 (avec Mathematica)

Une cuve cylindrique de rayons $r = 50$ cm est posée horizontalement sur le sol. Graduer une jauge, c'est-à-dire dessinez une bande de papier comme présentée dans la figure ci-dessous (il s'agit de l'exercice 4).



- (a) Première version : résoudre les équations l'une après l'autre, sans utiliser de listes

Indications

- (i) Il s'agit d'abord de résoudre 8 équations du problème 4 et de former la liste des angles qui correspondent à chaque graduation : $\alpha_0 = 0$; α_1 est la solution de l'équation pour le taux de remplissage $t = 0.1$; α_2 est la solution de l'équation pour le taux de remplissage $t = 0.2$; ... ; $\alpha_{10} = 2\pi$.
- (ii) Il s'agit ensuite de former la liste des hauteurs qui correspondent à chaque angle : $h_0 = 0$; h_1 est la hauteur qui correspond à l'angle α_1 ; h_2 est la hauteur qui correspond à l'angle α_2 ; ... ; $h_{10} = 2r$.
- (iii) Il faut enfin dessiner une liste de traits horizontaux :

```

Graphics [
  {
    Line[{0,h0},{5, h0}],
    ...
    Line[{0,h10},{5, h10}],
    Text["0%",{10,h0}],
    ...
    Text["100%",{10,h10}]
  }
]

```

- (b) Deuxième version : écrire un programme plus compact en utilisant **Table**.

Exercice 33 (avec Mathematica)

Pour rembourser un emprunt de 8200 francs, une banque demande de verser 6 annuités de 2000 francs. Calculer le taux d'intérêt i demandé par la banque (il s'agit d'un cas particulier de l'exercice 5).

Éléments de programmation Mathematica

Exercice 34 (avec Mathematica)

Définir à l'aide d'un `If` une fonction `appreciation[note_]` qui retourne `insuffisant` si `note` est inférieure à 4 et `suffisant` sinon.

Exercice 35 (avec Mathematica)

Soit la suite $(a_n)_{n \in \mathbb{N}}$ définie de la façon suivante :

$$a_0 = 1, \quad a_{n+1} = \frac{2 + 2a_n^3}{3a_n^2}.$$

- A l'aide d'un module et en utilisant la fonction `Nest`, définir la fonction `a[n_]` donnant la valeur exacte de a_n .
- Donner a_{10} avec une précision de 15 chiffres significatifs.

Exercice 36 (sans Mathematica)

Supposons que la liste d'instructions suivante soit évaluée dans Mathematica :

```
Clear["Global`*"]
a = 3;
b = 10;
f[x_] := Module[
  {a, d},
  a = 100;
  c = 1000;
  d = 10000;
  x + a + b
]
```

Répondre aux questions suivantes en s'aidant d'un tableau donnant l'état de la mémoire à l'intérieur et à l'extérieur du module.

- Deviner ce que retourne Mathematica lors de l'évaluation de `{a,b,c,d}`.
- Deviner ce que retourne Mathematica si vous évaluez ensuite `f[1]`.
- Deviner ce que retourne Mathematica si vous évaluez finalement `{a,b,c,d}`.

Exercice 37 (avec Mathematica)

Soit la fonction $f(x) = 2 \sin(x) - x$.

- Définir la fonction `f[x_]` dans un cahier puis tracer son graphe en y ajoutant un point à la position de chacun de ses zéros se trouvant entre $a = -3$ et $b = 3$. Prendre garde à définir vos différentes variables de façon à ce qu'une modification des définitions de a , b ou f modifie automatiquement le résultat lors d'une réévaluation complète du cahier.
- Définir un module `graphZeros[g_,{min_,max_}]` qui trace le graphe d'une fonction g entre $\min$ et $\max$ en y ajoutant un point à la position de ses zéros ou qui écrit qu'il n'y pas de zéro dans l'intervalle demandé. Tester votre module avec différentes fonctions dont les zéros sont connus.

9 Exercices de répétition

Exercice 38 (avec Mathematica)

- (a) Programmer un module `bissectionTol[f,{a,b},tol]` qui retourne le premier intervalle de longueur inférieure à $2 \cdot \text{tol}$ obtenu par bisections successives de l'intervalle $[a; b]$ avec une fonction `f` sachant que `f` et $[a; b]$ vérifient les hypothèses d'utilisation de la méthode de la bisection.
- (b) Tester le module précédent pour calculer avec une précision de 10^{-6} le zéro de $g(x) = x^3 - 4x^2 - 20x + 1$ se trouvant dans l'intervalle $[0; 1]$. En déduire la valeur du zéro ainsi que son erreur (donner la réponse sous la forme $\dots \pm \dots$).

Exercice 39 (sans/avec Mathematica)

Un stère de bois (cube de 1 m de côté) est entassé contre une grande maison. A quelle distance x de la maison faut-il poser le pied d'une échelle de 10 m pour qu'elle s'appuie tant contre la façade que contre l'angle du tas de bois ?

- (a) Partie sans Mathematica
 - (i) Déduire de l'énoncé que les valeurs possibles de x sont les zéros de la fonction $f(x) = (x - 1)\sqrt{100 - x^2} - x$, $x \in [1; 10]$.
 - (ii) Déterminer la plus grande solution avec une erreur absolue inférieure 0.02 à l'aide de la méthode de la bisection en partant d'un intervalle à bornes entières.
 - (iii) Déterminer la plus petite solution avec une précision de trois chiffres après la virgule à l'aide de la méthode *regula falsi*.
- (b) Partie avec Mathematica
 - (i) L'itération de la fonction $g(x) = f(x) + x$ converge-t-elle si on prend comme valeur initiale 1.5 ? Argumenter à l'aide d'un graphique.
 - (ii) Déterminer avec la méthode de Newton les solutions du problème avec une précision de 6 chiffres significatifs.
 - (iii) Déterminer avec une méthode appropriée de Mathematica les solutions exactes et approximatives du problème.
 - (iv) En faisant abstraction du contexte de ce problème, la fonction f est définie sur l'intervalle $[-10; 10]$. Déterminer la réunion d'intervalles sur laquelle la méthode de Newton donne une première itération hors de l'intervalle $[-10; 10]$. L'estimation des bornes des intervalles est demandée avec une précision au centième.

10 Réponses des exercices

- 1** $\pi r(r + \sqrt{h^2 + r^2}) - A = 0$ est une équation de catégorie (I) et a pour solution $r = \frac{A}{\sqrt{\pi(h^2 + 2A)}}$.
- 2** $A = \frac{1}{2}(p + q)\sqrt{pq}$ est une équation d'inconnue $q \in \mathbb{R}^*$ de catégorie (II), $a = p + q$, $b = \sqrt{pq + p^2}$, $c = \sqrt{pq + q^2}$
- 3** $4r^3(\rho_0 - \rho_1) = h^2(3r - h)\rho_0$ est une équation de catégorie (II) d'inconnue $h \in]0; 2r[$.
- 4** $t = \frac{\alpha - \sin(\alpha)}{2\pi}$ est une équation de catégorie (III) d'inconnue $\alpha \in [0; 2\pi]$; $h = r(1 - \cos(\frac{\alpha}{2}))$.
- 5** $cr^n - a\frac{r^n - 1}{r - 1} = 0$ est une équation de catégorie (III) d'inconnue $r \in]1; 1.18]$ avec $i = r - 1$.
- 6** $q \in]45; 46[$
- 7** $h \in]0.10; 0.12[$
- 8** $\alpha \in]2.4; 2.5[$
- 9** $i \in]11\%; 12\%[$
- 10** $x = 3.32 \pm 0.01$
- 11** Il faut effectuer au moins 27 bisections.
- 12** (a) $q = (45.95 \pm 0.02)$ cm
(b) $q \approx 45.95$ cm, $a \approx 65.96$ cm, $b \approx 36.32$ cm, $c \approx 55.06$ cm.
- 13** (c) $\{\{-3, -2\}, \{-3, -2.5\}, \{-3, -2.75\}, \{-3, -2.875\}, \{-2.9375, -2.875\}, \{-2.9375, -2.90625\}, \{0, 1\}, \{0, 0.5\}, \{0, 0.25\}, \{0, 0.125\}, \{0, 0.0625\}, \{0.03125, 0.0625\}\}$
(f) $\{0.0495148, 0.0495157\}$
- 14** (a) $[-1; 0]$
(b) Approximativement $[-1; -0.625]$ puis $[-0.69; -0.625]$
(c) $[-0.690986; -0.625]$
- 15** (a) $y = \frac{x(f(b) - f(a)) + bf(a) - af(b)}{b - a}$
- 16** $\alpha \approx 2.49078$, $h \approx 0.680308$.
- 18** (a) $[1; 2]$
(b) $[1.38; 2]$
- 19** (a) (ii) $\{2, 1.73205, 1.65289, 1.62877, 1.62135, 1.61906, 1.61835\}$
(iii) Le point fixe $r = 1.61804$ est solution de l'équation $x = \sqrt{x + 1}$.
(b) (ii) $\{0.5, 0.25, 0.0625, 0.00390625, 0.0000152588, 2.32831 \cdot 10^{-10}\}$
(iii) Le point fixe $r = 0$ est solution de l'équation $x = x^2$.
(c) (ii) $\{1.5, 2.25, 5.0625, 25.6289, 656.841, 431440\}$
(iii) La suite diverge.
(d) (ii) $\{2., 0.5, 2., 0.5, 2., 0.5\}$
(iii) La suite diverge.
- 21** (a) $r = 0$, $g'(r) = 1$
(b) $J =]-\sqrt{\frac{2}{3}}; \sqrt{\frac{2}{3}}[$

- 22** (a) 1.41
 (b) 1.414 213 562 432 542
 (c) • Si $x_0 \in]-\infty, -\sqrt{2}[\cup]2 + \sqrt{2}, +\infty[$ alors la suite diverge ($\lim_{n \rightarrow \infty} x_n = -\infty$).
 • Si $x_0 \in \{-\sqrt{2}; 2 + \sqrt{2}\}$ alors la suite converge vers $-\sqrt{2}$.
 • Si $x \in]-\sqrt{2}; 2 + \sqrt{2}[$ alors la suite converge vers $\sqrt{2}$.

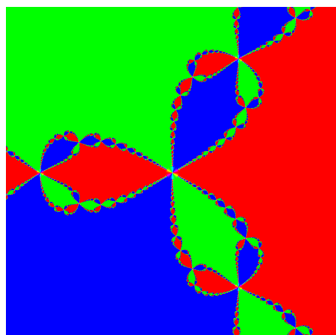
- 23** (a) $x_1 \approx 4.8, x_2 \approx 3.1, x_3 \approx 2.6$.
 (b) $x_1 \approx 1.7, x_2 \approx -2.3, x_3 \approx 5.1$.

- 24** (a) 1
 (b) $[1; 2]$
 (c) 1.213 41

- 25** (a) $3^x - 7 = 0$
 (b) $x_1 \approx 1.9, x_2 \approx 1.6$.
 (c) 1.544 36
 (d) $x_1 \approx 1.9, x_2 \approx 1.6$.

- 26** (b) Pour x_0 suffisamment proche de 0 la méthode de Newton converge vers 0.
 (c) $J \approx]-1.4; 1.4[$

- 28** (a) $\mathcal{Z}_f = \left\{1; e^{\frac{2\pi i}{3}}; e^{\frac{4\pi i}{3}}\right\}$
 (b)



29 $r = \frac{A}{\sqrt{\pi(2A + \pi h^2)}}$

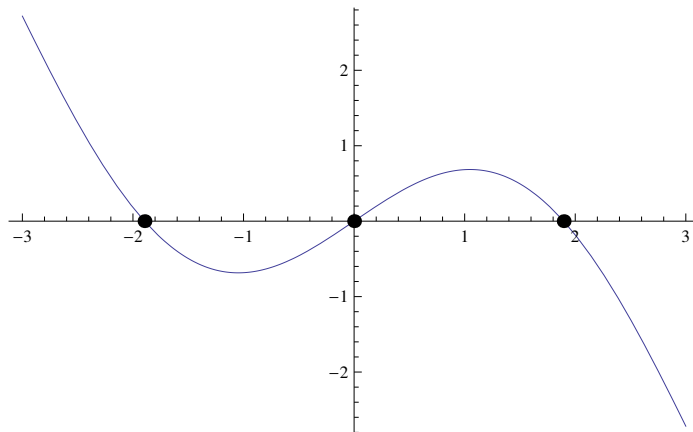
30 $a = 65 \text{ cm}, b \approx 36.055\,512\,754\,639\,89 \text{ cm}, a \approx 54.083\,269\,131\,959\,84 \text{ cm}.$

31 $h \approx 11.748\,006\,339\,545\,503 \text{ cm}$

33 $i \approx 12.098\,182\,925\,159\,01\%$

35 (b) $a_{10} = 1.259\,921\,049\,894\,87 (\approx \sqrt[3]{2})$

- 36** (a) $\{3, 10, c, d\}$
 (b) 111
 (c) $\{3, 10, 1000, d\}$

37 (a)**38** (c) $0.049\,515\,72 \pm 0.000\,000\,95$ (Mathematica retourne $\{0.049\,514\,8, 0.049\,516\,7\}$)**39** (a) (ii) $x_{\max} \approx (9.925 \pm 0.015)$ m (obtenu en arrondissant à chaque étape les bornes au centième)(iii) $x_{\min} \approx 1.112$ m

(b) (i) La méthode diverge.

(ii) La méthode donne $x_{\min} \approx 1.111\,88$ m et $x_{\max} \approx 9.937\,99$ m.(iii) $x_{\min} = \frac{1}{2}(1 + \sqrt{101} - \sqrt{2(49 - \sqrt{101})}) \approx 1.111\,88$ m, $x_{\max} = \frac{1}{2}(1 + \sqrt{101} + \sqrt{2(49 - \sqrt{101})}) \approx 9.937\,99$ m.(iv) $] -9.84; -6.55[\cup] -6.55; -5.47[\cup] 6.42; 7.07[\cup] 7.07; 9.77[$