On souhaite simuler des phénomènes physiques : obtenir une **solution approchée**, **approximation**, d'une solution exacte le plus souvent inconnue.

Plusieurs sources d'erreurs vont intervenir, à différents niveaux:

- **Erreur de modélisation :** par exemple en négligeant certains termes physiques,
- **Erreur de représentation en machine:** pour un ordinateur il est impossible de représenter de manières exactes des nombres tels que $\frac{1}{3}, \pi, \ldots \rightarrow$ on procédera à l'arrondi, ou troncature,
- **Erreur de troncature:** certaines opérations ($\log$, $\exp$, $\sin$,$\cos$) sont approchées par une **somme finie** d'opérations élémentaires (+,-,x,/). Cependant, le plus souvent, pour être exacte, la somme devrait être infinie...
  Ce dernier type d'erreur interviendra notamment lors de l'étude des **développements de Taylor.**

Quelques catastrophes dues à une mauvaise utilisation de l'outil numérique :
http://ta.twi.tudelft.nl/users/vuik/wi211/disasters.html

Deux définitions pour commencer. Une mesure **quantitative**...

Définition (Erreur absolue)

Soit x un nombre réel, et x^* une approximation de ce nombre. L'erreur absolue est définie par

$$E_a(x^*) = |x - x^*|$$

...ainsi qu'une mesure **qualitative**

Définition (Erreur relative)

Soit x un nombre réel, et x^* une approximation de ce nombre. L'erreur relative est définie par

$$E_r(x^*) = \frac{|x - x^*|}{|x|} = \frac{E_a(x^*)}{|x|}$$

En multipliant E_r par 100, on obtient l'erreur relative en %.

ⓘ En pratique, x n'est pas connu !

On exhibera le plus souvent une **borne supérieure** de l'erreur absolue, notée Δx, *i.e*

$$E_a(x^*) = |x - x^*| \leq \Delta x$$

Par abus de langage, **on confondra la borne Δx avec $E_a(x^*)$**.

Pour l'erreur relative E_r, on a alors

$$E_r(x^*) = \frac{|x - x^*|}{|x|} \leq \frac{\Delta x}{|x|}$$

et si Δx est suffisamment «petit», on dira que

$$E_r(x^*) \approx \frac{\Delta x}{|x^*|}$$

Remarque(s)

$$|x - x^*| \leq \Delta x \iff -\Delta x \leq x - x^* \leq \Delta x \iff x^* - \Delta x \leq x \leq x^* + \Delta x$$

Autrement dit, $x^* = x \pm \Delta x$, Δx représente ainsi l'incertitude.

En conclusion, **sauf si l'on connaît à la fois x et x^***, on utilisera:

- Δx comme valeur pour l'erreur absolue,
- $\frac{\Delta x}{|x^*|}$ comme valeur pour l'erreur relative.

Pour mesurer la précision d'une approximation, on introduit la notion de chiffres significatifs (C.S).

Définition (Chiffres significatifs)

Si l'erreur absolue vérifie $\Delta x \leq 0,5 \times 10^m$, alors le chiffre dans x^* correspondant à la m^e puissance de 10 est dit *significatif* et tous ceux à sa gauche (correspondant aux puissances de 10 supérieures à m) le sont aussi. On arrête le compte au dernier chiffre non nul.

Si le chiffre correspondant à la m^e puissance de 10 dans x^* est nul ainsi que tous ceux à sa gauche, on dit qu'il n'y a aucun chiffre significatif.

Inversement, on estime l'erreur absolue (Δx) à partir du nombre de chiffres significatifs de la manière suivante:

• Si une approximation x^* possède n chiffres significatifs, on commence à compter à partir du premier chiffre non nul à gauche, et l'erreur absolue est inférieure à 0,5 fois la puissance de 10 correspondant au dernier chiffre significatif.

Attention !

Si x^* possède n chiffres significatifs, cela ne veut pas dire que x^* possède n chiffres exacts ! **Mais n chiffres dont on contrôle l'erreur.**

Représentation en virgule flottante: Soit x un nombre réel. La représentation en virgule flottante de x dans **la base b** sera

$$\begin{aligned} x = \pm m \times b^l &= \pm 0, d_1 d_2 \cdots d_n \cdots \times b^l \\ &= \pm(d_1 \times b^{-1} + d_2 \times b^{-2} + \cdots + d_n \times b^{-n} + \cdots) \times b^l \end{aligned}$$

où

- $m = 0, d_1 d_2 \cdots d_n \cdots$ est la **mantisse**, par convention on prendra $m < 1$.
- b est la **base**, $b \in \mathbb{N}$,
- l est l'**exposant**, $l \in \mathbb{Z}$.

De plus, les entiers d_i dans l'écriture de la mantisse vérifient:

- $0 \leq d_i \leq b-1, \; i = 2, 3, \cdots$
- $0 < d_1 \leq (b-1)$, cela permet d'assurer l'unicité de la représentation. On parlera de virgule flottante à **mantisse normalisée**.

Dans ce cas, on peut montrer

$$\frac{1}{b} \leq m < 1$$

à noter que l'écriture de 0 devient une exception (car la mantisse ne s'annule jamais !).

- Le choix de la base b est arbitraire, cependant il y a des choix naturels :
  - b=10, base décimale : utilisée par les humains,
  - b=2, base binaire, ou b=16, base hexadécimale : utilisées par les ordinateurs.

- **Quelle que soit la base choisie**, les nombres irrationnels (π, $\sqrt{2}, \cdots$) auront toujours une mantisse de longueur infinie,
- L'utilisation de mantisse infinie sera bien évidemment impossible sur un ordinateur, où le nombre de bits pour représenter un nombre réel est limité. Ainsi, on recourt à la troncature ou à l'arrondi pour réduire la mantisse à N chiffres. On aura donc

$$m = d_1 \times b^{-1} + d_2 \times b^{-2} + \cdots + d_N \times b^{-N}$$

  La représentation de certains réels sur ordinateur sera donc erronée. De plus les opérations élémentaires utilisant ces nombres seront également erronées.

Troncature et arrondi:

- Troncature à N chiffres: on coupe la mantisse au-delà de la position N.
- Arrondi à N chiffres: On ajoute (5 en décimal, 1 en binaire) à la position $N+1$, puis on fait une troncature à N chiffres.

Définition (Précision machine)

On appelle précision machine (notée ε_m) la plus grande erreur relative commise en représentant un nombre réel sur ordinateur en utilisant la troncature:

$$\varepsilon_m = \max_x \frac{\Delta x}{|x|} = \max_x E_r(x).$$

En utilisant l'arrondi, la plus grande erreur relative est $\frac{\varepsilon_m}{2}$.

Il est possible de montrer que l'on a (p. 12-13)

$$\varepsilon_m \leq b^{1-N}$$

N étant le nombre de chiffres dans la mantisse.

- En pratique, on ne fera pas de distinctions entre ε_m et b^{1-N},
- Norme IEEE-754 (p 14-17): standardisation de la représentation. Garantis l'uniformité du comportement entre les ordinateurs.

  Précision machine [1] :
  - $2^{1-24} = 0,119 \times 10^{-6}$ en simple précision (32 bits)
  - $2^{1-53} = 0,222 \times 10^{-15}$ en double précision (64 bits)

Conclusion: En général les nombres ne seront plus représentés exactement. Il y aura des erreurs dans le résultat d'opérations arithmétiques de base [2] !

[1] voir `script_precision_machine.m` pour le calcul de la précision machine.

[2] voir `script_somme.m`.

Définition (Notation flottante)

Soit $x \in \mathbb{R}$. On note $fl(x)$ sa **représentation en virgule flottante décimal** à n chiffres

$$fl(x) = \pm 0, d_1 d_2 d_3 \cdots d_n \times 10^l$$

le dernier chiffre d_n dépend du procédé retenu pour éliminer les derniers chiffres, à savoir la troncature ou l'arrondi. **On utilisera de préférence l'arrondi**.

On utilise la base décimale, cependant les effets décrits par la suite valent également pour les autres bases !

Principe de fonctionnement des opérations élémentaires: On doit représenter les deux opérandes en notation flottante, effectuer l'opération de la façon habituelle et exprimer le résultat en notation flottante.

Opérations élémentaires:

- $x + y \longrightarrow fl(fl(x) + fl(y))$ • $x - y \longrightarrow fl(fl(x) - fl(y))$
- $x \div y \longrightarrow fl(fl(x) \div fl(y))$ • $x \times y \longrightarrow fl(fl(x) \times fl(y))$

- $+/-$: on devra faire un «décalage » de la mantisse lorsque les exposants ne sont pas les mêmes. Cela impliquera d'ajouter des zéros dans le nombre le plus petit (pour ramener son exposant au niveau du plus grand nombre).
- $\times/\div$: la loi des exposants permet de faire la somme (différence) des exposants et uniquement le produit (la division) des mantisses pour obtenir le résultat.

Attention ! Opérations risquées

- Le décalage de la mantisse peut se révéler **très dangereux si on additionne des nombres dont les ordres de grandeur sont très différents**.
- La troncature ou l'arrondi pouvant se faire sur certains termes intermédiaires, **l'associativité et la distributivité** n'existe plus: **l'ordre des opérations** devient important pour le résultat.
- Il peut être dangereux de soustraire des **nombres très proches**: perte de précision (chiffres significatifs).

Réelle source de problèmes en pratique !

Suivant le problème, il existe certaines méthodes de contournement : additionner les termes en les plaçant par ordre croissant, somme de Kahan (p. 22), . . .

Conclusion :

- Les opérations élémentaires sont sources d’erreurs,
- Pas de solution miracle répondant à tous les problèmes énoncés auparavant !
- On essayera de **minimiser le nombre d’opérations**. À ce titre, le nombre d’opérations élémentaires en virgule flottante sert souvent de point de comparaison entre différents algorithmes.

Voyons l’**algorithme de Horner** permettant de réduire le nombre d’opérations pour l’évaluation d’un polynôme.

Algorithme de Horner : multiplications imbriquées.
Soit le polynôme p de degré n, $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$.

En regroupant judicieusement les termes, on peut écrire:

$$\begin{aligned}
p(x) &= a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n \\
&= a_0 + x(a_1 + a_2 x + a_3 x^2 + \cdots + a_{n-1} x^{n-2} + a_n x^{n-1}) \\
&= a_0 + x(a_1 + x(a_2 + a_3 x + \cdots + a_{n-1} x^{n-3} + a_n x^{n-2})) \\
&= \cdots \\
&= a_0 + x(a_1 + x(a_2 + x(a_3 + \cdots + x(a_{n-1} + a_n x) \cdots)))
\end{aligned}$$

- Approche directe : au plus $\frac{n(n+1)}{2} + n$ opérations élémentaires.
- Horner : au plus $2n$ opérations élémentaires.

Algorithme de Horner, pseudo-code `script_Horner.m`

Données :

- Les coefficients a_i d'un polynôme p de degré n,
- Une abscisse x.

Résultat : la variable r qui contiendra la valeur du polynôme p évaluée en x

$r = a_n$;
pour $i = n, n-1, \cdots, 2, 1$ **faire**
    $r = a_{i-1} + xr$;
fin
$p(x) = r$;

Tout cela concerne l'erreur de représentation. Qu'en est-il de l'erreur de troncature ?

$\rightarrow$ Développement de Taylor

- Quel est le polynôme de degré n, noté $\boldsymbol{P_n(x)}$, qui donne la meilleure approximation d'une **fonction f donnée**, cela **au voisinage d'un point x_0, également donné** ?

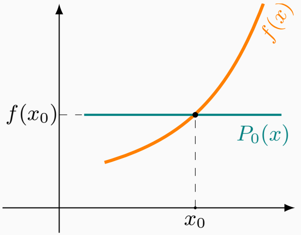
$\rightarrow$ Le polynôme (ou formule) de Taylor nous donne une méthode de construction des polynômes $P_n(x)$, **lorsque la fonction f est suffisamment dérivable.**

Définition (Polynôme de Taylor)

Le polynôme de Taylor de degré n de la fonction $f(x)$, autour du point x_0 est défini par

$$\begin{aligned} P_n(x) = & f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \\ & \frac{f'''(x_0)}{3!}(x-x_0)^3 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n \\ = & \sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!}(x-x_0)^i \end{aligned}$$

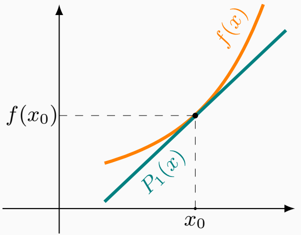
Note: la fonction f doit être n fois dérivable en x_0



- $n = 0$

$P_0(x) = f(x_0)$.

Cela signifie que l'on approche f par une constante, un polynôme de degré 0.

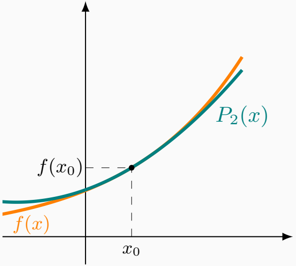


- $n = 1$

$P_1(x) = f(x_0) + f'(x_0)(x - x_0)$.

Cela signifie que l'on approche f par une droite, un polynôme de degré 1.

Note: $P_1(x)$ correspond à l'équation de la tangente à la courbe f en x_0



- $n = 2$

$$P_2(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2.$$

Cela signifie que l'on approche f par une quadratique, un polynôme de degré 2.

Approcher une fonction f par un polynôme de Taylor P_n va induire une erreur[3], donnée par l'expression

$$|f(x) - P_n(x)|$$

et dite **erreur de troncature**. La question est maintenant :

Suis-je en mesure de quantifier cette erreur ?

[3] voir `script_Taylor.m` pour exemples numériques avec Matlab.

Théorème (Développement de Taylor-Lagrange)

Soit f une fonction $\boldsymbol{n+1}$ **fois dérivable** autour de x_0, et $P_n(x)$ le polynôme de Taylor de degré n autour de x_0. On a **l'égalité** suivante

$$f(x) = P_n(x) + R_n(x)$$

R_n est «l'écart» entre f et P_n, défini par

$$R_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x - x_0)^{n+1}$$

où ξ_x est un nombre réel, dépendant de x, compris entre x et x_0, *i.e*

- $x \leq \xi_x \leq x_0$ si $x < x_0$
- $x_0 \leq \xi_x \leq x$ si $x > x_0$

L'erreur de troncature est ainsi définie par

$$|f(x) - P_n(x)| = |R_n(x)| = \left|\frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x - x_0)^{n+1}\right| = \left|\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\right| \left|(x - x_0)^{n+1}\right|$$

Remarque(s)

À n fixé, l'erreur augmente si x s'éloigne de x_0 et diminue si x se rapproche de x_0. Il est donc préférable d'avoir x_0 proche des points où l'on souhaite évaluer la fonction f...

On cherche à évaluer l'erreur de troncature

$$|R_n(x)| = |f(x) - P_n(x)| = \left| \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x - x_0)^{n+1} \right|$$

Cependant $\boldsymbol{\xi_x}$ n'est pas connu !!! On sait seulement qu'il existe et qu'il dépend de n, x, x_0, f.

Afin d'estimer l'erreur, le but du jeu va donc être de chercher à se «débarrasser» de la variable ξ_x.

Pour cela, on va chercher à majorer le terme $\left|f^{(n+1)}(\xi_x)\right|$, *i.e* trouver $M \in \mathbb{R}^+$ tel que

$$\left|f^{(n+1)}(\xi_x)\right| \leq M$$

On aura ainsi $|R_n(x)| = |f(x) - P_n(x)| \leq \frac{M}{(n+1)!}\left|(x - x_0)^{n+1}\right|$. On contrôle ainsi le terme d'erreur.

Formulation en h du développement de Taylor :

On utilise souvent une autre forme du développement de Taylor, obtenu en effectuant le changement de variable $\boldsymbol{h = x - x_0}$. On obtient ainsi

$$f(x) = f(x_0 + h) = P_n(x_0 + h) + R_n(x_0 + h) = \tilde{P}_n(h) + \tilde{R}_n(h)$$

avec

$$\tilde{P}_n(h) = f(x_0) + f'(x_0)h + \frac{f''(x_0)}{2!}h^2 + \frac{f'''(x_0)}{3!}h^3 + \cdots + \frac{f^{(n)}(x_0)}{n!}h^n$$

$$\tilde{R}_n(h) = \frac{f^{(n+1)}(\xi_h)}{(n+1)!}h^{n+1}$$

avec ξ_h compris entre x_0 et $x_0 + h$.

ξ_h compris entre x_0 et $x_0 + h$:

- Si $h > 0 \to \ \xi_h \in [x_0, x_0 + h] \iff x_0 \leq \xi_h \leq x_0 + h$
- Si $h < 0 \to \ \xi_h \in [x_0 + h, x_0] \iff x_0 + h \leq \xi_h \leq x_0$

Avec la formulation en h on cherchera une majoration du terme d'erreur R_n de la forme

$$|R_n(x_0 + h)| = \left| \frac{f^{(n+1)}(\xi_h)}{(n+1)!} \right| |h|^{n+1} \leq \frac{M}{(n+1)!} |h|^{n+1}$$

Encore une fois, on cherchera une majoration de la fonction $|f^{(n+1)}|$ ne faisant plus apparaitre ξ_h.

Au final, on a donc les deux expressions équivalentes suivantes:

$$
\begin{aligned}
f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots \\
+ \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x - x_0)^{n+1}
\end{aligned}
$$

$$
f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{f''(x_0)}{2!}h^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}h^n + \frac{f^{(n+1)}(\xi_h)}{(n+1)!}h^{n+1}
$$

Attention !

Il faut savoir manipuler les deux formulations du développement de Taylor, et ne pas les mélanger !

Définition (Notation de Landau - grand ordre)

Une fonction $e(h)$ est un grand ordre de h^n au voisinage de 0, on notera $e(h) = O(h^n)$ s'il existe une constante C telle que, au voisinage de 0,

$$|e(h)| \leq Ch^n$$

- **Lorsque** h **est assez petit**, la fonction $O(h^n)$ décroît comme Ch^n,
- Plus n est grand, plus la décroissante est rapide,
- Une fonction $O(h^p)$ est obligatoirement $O(h^m)$ pour $m < p$.
- Comment déterminer le grand ordre d'une fonction ? Pour h petit,

$$e\left(\frac{h}{2}\right) \approx C\left(\frac{h}{2}\right)^n = C\frac{h^n}{2^n} \Rightarrow \frac{e(h)}{e\left(\frac{h}{2}\right)} \approx 2^n$$

à titre d'exemple, si $e(h) = O(h^n)$ représente une erreur, en divisant h par 2, l'erreur sera divisée par 2^n.

Définition (Ordre d'une approximation)

Une approximation dont le **terme d'erreur** est un grand ordre de h^n ($O(h^n)$) est dite **approximation d'ordre** n.

• Suivant cette définition, le polynôme de Taylor de degré n sera généralement, mais pas toujours, une approximation d'ordre $n+1$, *i.e*

$$R_n(x) = O((x - x_0)^{n+1}) \text{ ou } R_n(h) = O(h^{n+1}).$$

Un contre exemple est le développement de Taylor d'ordre 5 de $f(x) = \sin(x)$ autour de 0 : il suffit de calculer le polynôme de Taylor de degré 3 pour avoir une approximation d'ordre 5.

• NE PAS CONFONDRE **le degré** du polynôme P_n **avec l'ordre** du développement de Taylor

- Le degré est la puissance de h (ou $(x - x_0)$) la plus grande dans le polynôme P_n,
- L'ordre est la puissance de h (ou $(x - x_0)$) intervenant dans le terme d'erreur R_n, et sert à indiquer la qualité de l'approximation.

Avec la notion de grand ordre, il est donc possible d'écrire les développements de Taylor de la façon suivante ($n = 2$ pour illustrer):

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + O((x - x_0)^3)$$

Formulation en h:

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{f''(x_0)}{2!}h^2 + O(h^3)$$

✓ La formulation en h est pratique pour évaluer facilement la fonction f en un point proche de x_0.

Voyons une application directe du développement de Taylor:
Soit x^* une approximation de x, f une fonction connue. Que peut-on dire de $|f(x) - f(x^*)|$?

$\rightarrow$ On parlera de **propagation d'erreur**, où comment l'incertitude sur x se répercute sur le résultat d'un calcul.

Propagation d'erreur :

Supposons connu une certaine approximation x^* (un paramètre physique par exemple), avec Δx connu. On souhaite évaluer la précision du calcul $f(x^*)$.

L'erreur absolue est définie par

$$\Delta f = |f(x) - f(x^*)| = |f(x^* \pm \Delta x) - f(x^*)|$$

Mais par développement de Taylor de f en x^*, on a

$$f(x^* \pm \Delta x) = f(x^*) \pm f'(x^*)\Delta x + O((\Delta x)^2)$$

On a ainsi

$$\Delta f \approx |f'(x^*)|\Delta x$$

à partir de l'erreur absolue sur x^*, on peut donc estimer l'erreur absolue sur $f(x^*)$.

Le développement de Taylor se généralise pour les fonctions de plusieurs variables.

Soit $f(x, y)$ une fonction de deux variables, à valeurs réelles que l'on suppose suffisamment différentiable. Le développement de Taylor d'ordre 2 de la fonction $f(x, y)$ autour du point (x_0, y_0) est défini par

$$\begin{aligned} f(x_0 + h_1, y_0 + h_2) = f(x_0, y_0) + \frac{\partial f}{\partial x}(x_0, y_0)h_1 + \frac{\partial f}{\partial y}(x_0, y_0)h_2 \\ + O(h_1^2) + O(h_2^2), \end{aligned}$$

Pour une fonction de trois variables $f(x, y, z)$, on aura autour de (x_0, y_0, z_0)

$$\begin{aligned} f(x_0 + h_1, y_0 + h_2, z_0 + h_3) &= f(x_0, y_0, z_0) + \frac{\partial f}{\partial x}(x_0, y_0, z_0)h_1 \\ &+ \frac{\partial f}{\partial y}(x_0, y_0, z_0)h_2 + \frac{\partial f}{\partial z}(x_0, y_0, z_0)h_3 \\ &+ O(h_1^2) + O(h_2^2) + O(h_3^2) \end{aligned}$$

Dans le cadre de ce cours, on n'explicitera pas le terme d'erreur pour les fonctions de plusieurs variables.

Cela nous fournit une formule pour la propagation d'erreur dans le cadre de fonctions à plusieurs variables

$$(x, y, z) = (x^*, y^*, z^*) \pm (\Delta x, \Delta y, \Delta z)$$

$$\begin{aligned}\Delta f = |f(x, y, z) - f(x^*, y^*, z^*)| \simeq \left|\frac{\partial f}{\partial x}(x^*, y^*, z^*)\right| \Delta x \\ + \left|\frac{\partial f}{\partial y}(x^*, y^*, z^*)\right| \Delta y \\ + \left|\frac{\partial f}{\partial z}(x^*, y^*, z^*)\right| \Delta z\end{aligned}$$

Propagation d'erreur absolue pour les opérations élémentaires:

$$\begin{aligned} f(x,y) = x + y \quad &\longrightarrow \quad \Delta f \simeq |\Delta x| + |\Delta y| \\ f(x,y) = x - y \quad &\longrightarrow \quad \Delta f \simeq |\Delta x| + |\Delta y| \\ f(x,y) = x \times y \quad &\longrightarrow \quad \Delta f \simeq |y^*||\Delta x| + |x^*||\Delta y| \\ f(x,y) = x \div y \quad &\longrightarrow \quad \Delta f \simeq \frac{|y^*||\Delta x| + |x^*||\Delta y|}{|y^*|^2} \end{aligned}$$

Je dois être capable de :

- Calculer/estimer l'erreur absolue et relative d'un nombre,
- Établir les chiffres significatifs partant de l'erreur absolue et réciproquement,
- Faire une représentation en virgule flottante avec mantisse à n chiffres par troncature ou par arrondi,
- Normaliser une mantisse et décaler la mantisse le cas échéant,
- Faire des opérations élémentaires en virgule flottante à n chiffres,
- Reconnaître les opérations dangereuses et comprendre l'importance de l'ordre dans les opérations élémentaires,
- Utiliser la méthode de Horner,
- Construire un polynôme de Taylor de degré n ainsi que son terme d'erreur,
- Estimer le reste, approximer l'erreur absolue et obtenir le nombre de chiffres significatifs de mon approximation,
- Comprendre et savoir calculer l'ordre d'une méthode,
- Savoir calculer la propagation d'erreur pour une fonction d'une ou de plusieurs variables et ainsi calculer l'erreur/chiffres significatifs d'une évaluation de fonction.