Méthodes numériques (cours du 15 janv. 2019)

François Cuvelier

Laboratoire d'Analyse Géométrie et Applications
Institut Galilée
Université Paris XIII.

2019/01/15

Divers

- Volume horaire : 8×3h cours/TD, $2 \times 8h$ TP.
- Prérequis des cours :
  - *Cours commun Mathématiques pour l'ingénieur* (L. El Alaoui), ...
  - *Algorithmique*
  - Langage de programmation : *Matlab*/*Octave*
- Note finale : $(P_1 + P_2 + TP)/3$ si possible.
  - P_1 : partiel 1 du 12 février 2019 (1h30/2h00),
  - P_2 : partiel 2 du 26 mars 2019 (1h30/2h00),
  - TP : travaux pratiques en mai et juin.

Objectifs

- Algorithmique numérique.
- Poursuite de l'apprentissage de Matlab/Octave.
- Résolution numériques d'équations différentielles ordinaires (E.D.O.[fr] ou O.D.E.[en]).
- Résolution numériques d'équations aux dérivées partielles (E.D.P.[fr] ou P.D.E.[en]) par des méthodes de différences finies.

Plan du cours

Chapitre I : Algorithmique numérique
Chapitre II : Dérivation numérique
Chapitre III : Résolution numérique des E.D.O.
Chapitre IV : Résolution numérique des E.D.P.

Chapitre I

Algorithmique numérique

Plan

Définition

Definition 1.1: Petit Robert 97

Algorithmique : Enchaînement d'actions nécessaires à l'accomplissement d'une tâche.

Exemple 1 : permutation

Nous voulons permutter deux voitures sur un parking de trois places numérotées de 1 à 3 et ceci sans géner la circulation.
La première voiture, une Renault Zoé, est sur l'emplacement 2, la seconde, une Citroën C1 , est sur l'emplacement 3.
Donner un algorithme permettant de résoudre cette tâche.

Exemple 2 : équation du premier degré

Donner un algorithme permettant de résoudre

$$ax = b$$

Caractéristiques d'un *bon* algorithme

- Il ne souffre d'aucune ambiguité $\Rightarrow$ très clair.
- Combinaison d'opérations (actions) élémentaires.
- Pour toutes les données d'entrée, l'algorithme doit fournir un résultat en un nombre fini d'opérations.

Première approche méthodologique

Etape 1 : Définir clairement le problème.

Première approche méthodologique

Etape 1 : Définir clairement le problème.

Etape 2 : Rechercher une méthode de résolution (formules, ...)

Première approche méthodologique

Etape 1 : Définir clairement le problème.

Etape 2 : Rechercher une méthode de résolution (formules, ...)

Etape 3 : Ecrire l'algorithme (par raffinement successif pour des algorithmes *compliqués*).

Plan

Vocabulaire de base

- constantes,variables,
- opérateurs (arithmétiques, relationnels, logiques),
- expressions,
- instructions (simples et composées),
- fonctions.

Données et constantes

- Donnée $\Rightarrow$ introduite par l'utilisateur
- Constante $\Rightarrow$ symbole, identificateur non modifiable

Variables

Definition 2.1

Une variable est un objet dont la valeur est modifiable, qui possède un nom et un type (entier, caractère, réel, complexe, tableau, matrice, vecteur...).

Opérateurs arithmétiques

Nom	Symbole	Exemple
addition	$+$	$a + b$
soustraction	$-$	$a - b$
opposé	$-$	$-a$
produit	$*$	$a * b$
division	$/$	a/b

Opérateurs relationnels

Nom	Symbole	Exemple
identique	$==$	$a == b$
différent	$\sim=$	$a \sim= b$
inférieur	$<$	$a < b$
supérieur	$>$	$a > b$
inférieur ou égal	$<=$	$a <= b$
supérieur ou égal	$>=$	$a >= b$

Opérateurs logiques

Nom	Symbole	Exemple
négation	$\sim$	$\sim a$
ou	$\vert$	$a \vert b$
et	&	$a \& b$

Opérateur d'affectation

Nom	Symbole	Exemple
affectation	$\leftarrow$	$a \leftarrow b$

Expressions

Definition 2.2

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Expressions

Definition 2.3

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression numérique

$$(b * b - 4 * a * c)/(2 * a)$$

Expressions

Definition 2.4

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression numérique

$$(b * b - 4 * a * c)/(2 * a)$$

Opérandes $\Rightarrow$ identifiants a, b, c, constantes 4 et 2.

Expressions

Definition 2.5

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression numérique

$$(b * b - 4 * a * c)/(2 * a)$$

Opérandes $\Rightarrow$ identifiants a, b, c, constantes 4 et 2.

Opérateurs $\Rightarrow$ symboles $*$, $-$ et $/$

Expressions

Definition 2.6

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression booléenne

$$(x < 3.14)$$

Expressions

Definition 2.7

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression booléenne

$$(x < 3.14)$$

Opérandes $\Rightarrow$ identifiants x et contantes 3.14

Expressions

Definition 2.8

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression booléenne

$$(x < 3.14)$$

Opérandes $\Rightarrow$ identifiants x et contantes 3.14
Opérateurs $\Rightarrow$ symboles $<$

Instructions

Definition 2.9

Une **instruction** est un ordre ou un groupe d'ordres qui déclenche l'exécution de certaines actions par l'ordinateur. Il y a deux types d'instructions : simple et structuré.

Instructions simples

- affectation d'une valeur a une variable.
- appel d'une fonction (procedure, subroutine, ... suivant les langages).

Instructions structurées

1. les instructions composées, groupe de pulsieurs instructions simples,
2. les instructions répétitives, permettant l'exécution répétée d'instructions simples, (i.e. boucles «pour», «tant que»)
3. les instructions conditionnelles, lesquels ne sont exécutées que si une certaine condition est respectée (i.e. «si»)

Exemple : boucle «pour»

Algorithme 1 Exemple boucle «pour»

Données : n, un entier positif

1: $S \leftarrow 0$
2: **Pour** $i \leftarrow 1$ **à** n **faire**
3: $\quad S \leftarrow S + \cos(i\hat{}2)$
4: **Fin Pour**

Listing 1: (Matlab) Exemple boucle for

```
n=input('n=');                1
assert(n>=0)                  2
S=0;                          3
for i=1:n                     4
  S=S+cos(i^2);               5
end                           6
```

Mais que fait-il?

Exemple : boucle «pour»

Algorithme 2 Exemple boucle «pour»

Données : n, un entier positif

1: $S \leftarrow 0$
2: **Pour** $i \leftarrow 1$ **à** n **faire**
3: $\quad S \leftarrow S + \cos(i\hat{}2)$
4: **Fin Pour**

Listing 2: (Matlab) Exemple boucle for

```
n=input('n=');                1
assert(n>=0)                  2
S=0;                          3
for i=1:n                     4
  S=S+cos(i^2);               5
end                           6
```

Mais que fait-il?

Calcul de $S = \sum_{i=1}^{n} \cos(i^2)$

Exemple : boucle «tant que»

Algorithme 3 Exemple boucle «tant que»

1: $i \leftarrow 0,\ x \leftarrow 1$
2: **Tantque** $i < 1000$ **faire**
3: $\quad x \leftarrow x + i * i$
4: $\quad i \leftarrow i + 1$
5: **Fin Tantque**

Listing 3: (Matlab) Exemple boucle `while`

```
i=0;x=1;                    1
while (i<1000)              2
  x=x+i*i;                  3
  i=i+1;                    4
end                         5
```

Mais que fait-il?

Exemple : boucle «tant que»

Algorithme 4 Exemple boucle «tant que»

1: $i \leftarrow 0,\ x \leftarrow 1$
2: **Tantque** $i < 1000$ **faire**
3: $\quad x \leftarrow x + i * i$
4: $\quad i \leftarrow i + 1$
5: **Fin Tantque**

Listing 4: (Matlab) Exemple boucle `while`

```
i=0;x=1;                    1
while (i<1000)              2
  x=x+i*i;                  3
  i=i+1;                    4
end                         5
```

Mais que fait-il?

Calcul de $x = 1 + \sum_{i=0}^{999} i\hat{}2$ et $i = 1000$

Exemple : instructions conditionnelles «si»

Algorithme 5 Exemple instruction «si»

Données : *age*, un réel.

1: **Si** *age* >= 18 **alors**
2: affiche('majeur')
3: **Sinon Si** *age* >= 0 **alors**
4: affiche('mineur')
5: **Sinon**
6: affiche('en devenir')
7: **Fin Si**

Listing 5: (Matlab) Exemple instruction `if`

```
age=input('age=');                  1
if age>=18                          2
  disp('majeur')                    3
elseif age>=0                       4
  disp('mineur')                    5
else                                6
  disp('en␣devenir')                7
end                                 8
```

Plan

Description du problème

- Spécification d'un ensemble de données
  Origine : énoncé, hypothèses, sources externes, ...
- Spécification d'un ensemble de buts à atteindre
  Origine : résultats, opérations à effectuer, ...
- Spécification des contraintes

Recherche d'une méthode de résolution

- Clarifier l'énoncé.
- Simplifier le problème.
- Ne pas chercher à le traiter directement dans sa globalité.
- S'assurer que le problème est soluble (sinon problème d'indécidabilité!)
- Recherche d'une stratégie de construction de l'algorithme
- Décomposer le problème en sous problèmes partiels plus simples.
- Effectuer des raffinements successifs de chaque sous problème. Le niveau de raffinement le plus élémentaire étant celui des instructions.

Réalisation d'un algorithme

- Les types des données et des résultats doivent être précisés.
- L'algorithme doit fournir au moins un résultat (qui peut être graphique).
- L'algorithme doit être exécuté en un nombre fini d'opérations.
- L'algorithme doit être spécifié clairement, sans la moindre ambiguïté.

Exercices

Les deux exercices qui suivent sont intentionnellement mal rédigés!!!

Exercice 1

Ecrire un algorithme permettant de calculer

$$S(x) = \sum_{k=1}^{n} k \sin(2 * k * x)$$

Exercice 2

Ecrire un algorithme permettant de calculer

$$P(z) = \prod_{n=1}^{k} sin(2 * k * z/n)^k$$

Exercices

Exercice 3

Reprendre les deux exercices précédents en utilisant les boucles «tant que».

Plan

Les fonctions

Les fonctions permettent

- d'automatiser certaines tâches répétitives au sein d'un même algorithme,
- d'ajouter à la clarté de la l'algorithme,
- l'utilisation de portion de code dans un autre algorithme,
- ...

Les fonctions prédéfinies

- les fonctions d'affichage et de lecture : **Affiche**, **Lit**

Les fonctions prédéfinies

- les fonctions d'affichage et de lecture : **Affiche**, **Lit**
- les fonctions mathématiques :

$$\sin,\ \cos,\ \exp,\ \cdots$$

Les fonctions prédéfinies

- les fonctions d'affichage et de lecture : **Affiche**, **Lit**
- les fonctions mathématiques :

$$\sin,\ \cos,\ \exp,\ \cdots$$

- les fonctions de gestion de fichiers
- ...

Ecrire ses propres fonctions

Avant d'écrire une fonction, voici les questions à se poser:

1. Que doit-elle calculer/réaliser précisement (but)?
2. Quelles sont ses données (avec leurs limitations)?

Syntaxe

Fonction $[args_1, \ldots, args_n] \leftarrow$ NomFonction($arge_1, \ldots, arge_m$)
  instructions
Fin Fonction

Fonction $args \leftarrow$ NomFonction($arge_1, \ldots, arge_m$)
  instructions
Fin Fonction

Résoudre $ax = b$

Ecrire une fonction permettant de résoudre

$$ax = b$$

- **But** :
- **Données** :
- **Résultats** :

Résoudre $ax = b$

Ecrire une fonction permettant de résoudre

$$ax = b$$

- **But** :
  Enoncé insuffisament précis! On choisi ici le problème:
  $a \in \mathbb{R}^*$ et $b \in \mathbb{R}$ donnés, trouver $x \in \mathbb{R}$ solution de $ax = b$
  On aurait pu prendre a une matrice réelle d'ordre n et $b \in \mathbb{R}^n$...
- **Données** :
- **Résultats** :

Résoudre $ax = b$

Ecrire une fonction permettant de résoudre

$$ax = b$$

- **But** :
  Enoncé insuffisament précis! On choisi ici le problème:
  $a \in \mathbb{R}^*$ et $b \in \mathbb{R}$ donnés, trouver $x \in \mathbb{R}$ solution de $ax = b$
  On aurait pu prendre a une matrice réelle d'ordre n et $b \in \mathbb{R}^n$...
- **Données** :
  $a \in \mathbb{R}^*$ et $b \in \mathbb{R}$.
- **Résultats** :

Résoudre $ax = b$

Ecrire une fonction permettant de résoudre

$$ax = b$$

- **But** :
  Enoncé insuffisament précis! On choisi ici le problème:
  $a \in \mathbb{R}^*$ et $b \in \mathbb{R}$ donnés, trouver $x \in \mathbb{R}$ solution de $ax = b$
  On aurait pu prendre a une matrice réelle d'ordre n et $b \in \mathbb{R}^n$...
- **Données** :
  $a \in \mathbb{R}^*$ et $b \in \mathbb{R}$.
- **Résultats** :
  $x \in \mathbb{R}$.

Résoudre $ax = b$

Algorithme 6 Exemple de fonction : Résolution de l'équation du premier degré $ax = b$.

Données : a : nombre réel différent de 0
b : nombre réel.

Résultat : x : un réel.

1: **Fonction** $x \leftarrow$ REPD(a, b)
2: $x \leftarrow b/a$
3: **Fin Fonction**

Exemple: équation du second degré

On cherche les solutions réelles de l'équation

$$ax^2 + bx + c = 0, \tag{1}$$

Pour celà, on pose $\Delta = b^2 - 4ac$

- si $\Delta < 0$ alors les deux solutions sont complexes,
- si $\Delta = 0$ alors la solution est $x = -\frac{b}{2a}$,
- si $\Delta > 0$ alors les deux solutions sont $x_1 = \frac{-b-\sqrt{\Delta}}{2a}$ et $x_2 = \frac{-b-\sqrt{\Delta}}{2a}$.

Exercice

1. Ecrire la fonction `discriminant` permettant de calculer le discriminant de l'équation (1).
2. Ecrire la fonction `RESD` permettant de résoudre l'équation (1) en utilisant la fonction `discriminant`.
3. Ecrire un programme permettant de valider ces deux fonctions.

Plan

Exercice 4

Soit la série de Fourier

$$x(t) = \frac{4A}{\pi}\left\{\cos\omega t - \frac{1}{3}\cos 3\omega t + \frac{1}{5}\cos 5\omega t - \frac{1}{7}\cos 7\omega t + \cdots\right\}.$$

Ecrire la fonction SFT permettant de calculer $x_n(t)$.

Exercices

Exercice 5

Soient x un réel, m, n, p, q des entiers strictement supérieurs à 1, $\boldsymbol{u} = (u_1, \ldots, u_m)$ un vecteur de $\mathbb{R}^m$, $\boldsymbol{v} = (v_1, \ldots, v_p)$ un vecteur de $\mathbb{R}^p$ et $\boldsymbol{w} = (w_1, \ldots, w_q)$ un vecteur de $\mathbb{R}^q$.
Le réel y est donné par

$$y = \prod_{i=1}^{m} \left((u_i + \cos(x)) \sum_{k=1}^{n} (k + (x - i)^2) \right)$$

Q.1

1. Quelles sont les données nécessaires et suffisantes permettant de calculer y? Préciser les types et les dimensions.
2. Ecrire la fonction **PS** permettant de calculer y. Toutes les données seront passées en paramètre à la fonction.
3. Donner un exemple d'utilisation de cette fonction.

Soit $\boldsymbol{z} = (z_1, \ldots, z_m)$ le vecteur de $\mathbb{R}^m$ défini par

$$z_i = \sum_{k=1}^{p} \left((u_i + \cos(kx)) \prod_{j=1}^{p} (v_k + (x - j)^2) \right), \ \forall i \in [\![1, m]\!].$$

Q.2

1. Quelles sont les données nécessaires et suffisantes permettant de calculer $\boldsymbol{z}$? Préciser les types et les dimensions.
2. Ecrire la fonction **SP** permettant de calculer $\boldsymbol{z}$. Toutes les données seront passées en paramètre à la fonction.
3. Donner un exemple d'utilisation de cette fonction.

Exercices

Exercice 6

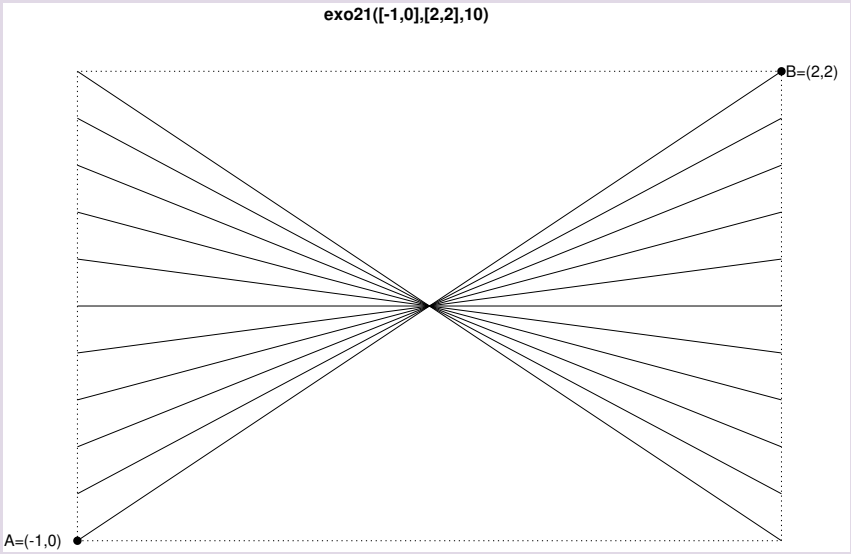
Q.1 Ecrire une fonction `DisReg` permettant de d'obtenir une discrétisation régulière de l'intervalle $[a, b]$ $(a < b)$ en $n + 1$ points.

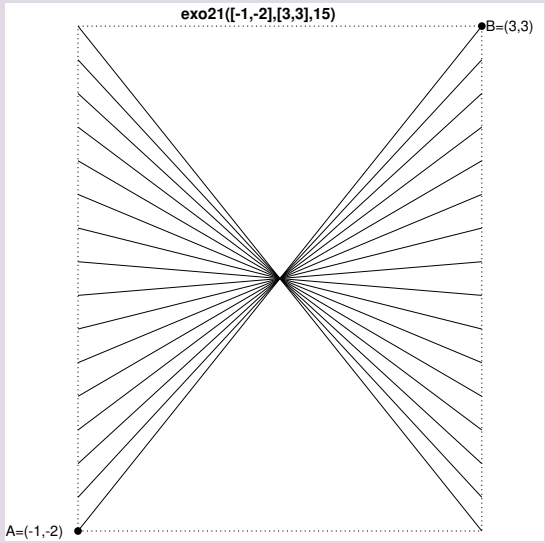
Soient $A = (x_A, y_A)$ et $B = (x_B, y_B)$ deux points du plan tels que $x_A < x_B$ et $y_A < y_B$. Ces deux points permettent de définir le rectangle de sommets A, (x_B, y_A), B et (x_A, y_B).
On suppose que pour tracer un trait entre les points A et B, on dispose de la commande `plot(`$[x_A, x_B]$`,`$[y_A, y_B]$`)`.

Q.2 Ecrire une fonction `exo21` de paramètres A, B et n permettant de

- représenter les bords du rectangle,
- relier les points des bords gauche et droit, dont les ordonnées sont une discrétisation régulière en $n + 1$ points, et passant par le centre de symétrie du rectangle.

Deux exemples d'utilisation de cette fonction sont donnés ci-dessous :

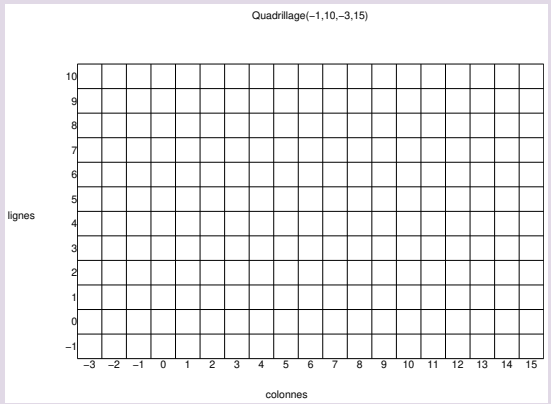




Exercices

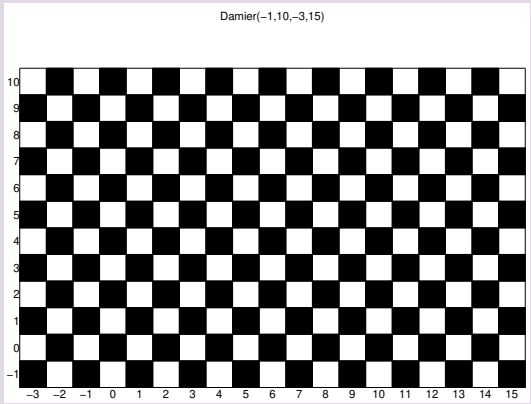
Exercice 7

On dispose d'un quadrillage quelconque généré par la fonction `quadrillage(imin,imax,jmin,jmax)` dont voici un exemple d'utilisation



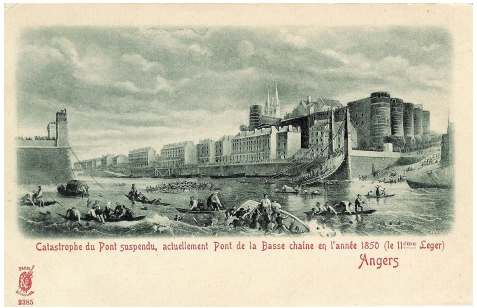
On dispose de plus d'une fonction `black(i,j)` qui dessine un pavé noir en ligne i et colonne j d'un quadrillage.

Q.1 Ecrire une fonction `Damier` permettant de créer un damier quelconque sachant que le pavé en bas à gauche d'un quadrillage doit toujours être noir. Voici une représentation pour le quadrillage précédent :



Plan

Mais avant de pousuivre ...



(a) Pont de la Basse-Chaîne, *Angers* (1850)

(b) Takoma Narrows Bridge, *Washington* (1940)

(c) Millenium Bridge, *London* (2000)

Figure: Une histoire de ponts

Chapitre II

Dérivation numérique

Definition 6.1

Soit $N \in \mathbb{N}^*$. On appelle **discrétisation régulière de** $[a, b]$ **à** N **pas ou** $N+1$ **points** l'ensemble des points $a + nh$, $n \in [\![0, N]\!]$ où le pas h est donné par $h = (b-a)/N$.

On note $t^n = a + nh$, $n \in [\![0, N]\!]$, une **discrétisation régulière de** $[a, b]$ et $(Dy)_n$ une approximation de $y'(t^n)$. On appelle

- **différence finie progressive** l'approximation

$$(Dy)_n^P = \frac{y(t^{n+1}) - y(t^n)}{h}, \ \forall n \in [\![0, N-1]\!] \tag{2}$$

- **différence finie rétrograde** l'approximation

$$(Dy)_n^R = \frac{y(t^n) - y(t^{n-1})}{h}, \ \forall n \in [\![1, N]\!] \tag{3}$$

- **différence finie centrée** l'approximation

$$(Dy)_n^C = \frac{y(t^{n+1}) - y(t^{n-1})}{2h}, \ \forall n \in [\![1, N-1]\!] \tag{4}$$

Proposition 6.2: Développement de Taylor

Soit f une application $f : [a, b] \longrightarrow \mathbb{R}$. Si $f \in \mathcal{C}^{r+1}([a, b])$ alors

- $\forall (x, y) \in [a, b]^2$ il existe un $\xi \in]x, y[$ tel que

$$f(x) = f(y) + \sum_{k=1}^{r} \frac{f^{(k)}(y)}{k!}(x - y)^k + \frac{f^{(r+1)}(\xi)}{(r+1)!}(x - y)^{r+1} \tag{5}$$

- $\forall x \in [a, b]$, $\forall h \in \mathbb{R}^*$ vérifiant $x + h \in [a, b]$, il existe $\xi \in]\min(x, x + h), \max(x, x + h)[$ tel quel

$$f(x + h) = f(x) + \sum_{k=1}^{r} \frac{f^{(k)}(x)}{k!} h^k + \frac{f^{(r+1)}(\xi)}{(r+1)!} h^{r+1} \tag{6}$$

Exercice 8

Q.1 Soit $y \in \mathcal{C}^2([a,b])$.

1. Montrer qu'il existe $\eta_P^n \in]t^n, t^{n+1}[$ et $\eta_R^n \in]t^{n-1}, t^n[$ tels que

$$(Dy)_n^P = y^{(1)}(t^n) + \frac{h}{2} y^{(2)}(\eta_P^n)$$

et

$$(Dy)_n^R = y^{(1)}(t^n) - \frac{h}{2} y^{(2)}(\eta_R^n)$$

2. En déduire que

$$|y^{(1)}(t^n) - (Dy)_n^P| \leqslant C_1 h, \quad \text{avec} \quad C_1 = \frac{1}{2} \max_{t \in [t^n, t^{n+1}]} |y^{(2)}(t)|$$

et

$$|y'(t^n) - (Dy)_n^R| \leqslant C_2 h, \quad \text{avec } C_2 = \frac{1}{2} \max_{t \in [t^{n-1}, t^n]} |y^{(2)}(t)|$$

Q.2 Soit $y \in \mathcal{C}^3([a,b])$.

1. Montrer qu'il existe $\eta_1^n \in]t^n, t^{n+1}[$ et $\eta_2^n \in]t^{n-1}, t^n[$ tels que

$$(Dy)_n^C = y^{(1)}(t^n) - \frac{h^2}{12}(y^{(3)}(\eta_1^n) + y^{(3)}(\eta_2^n))$$

2. En déduire que

$$|y^{(1)}(t^n) - (Dy)_n^C| \leqslant E h^2, \quad \text{avec} \quad E = \frac{1}{6} \max_{t \in [t^{n-1}, t^{n+1}]} |y^{(3)}(t)|$$

Definition 6.3

Soit g une fonction. On dit que g **se comporte comme un grand O de** h^q quand h tend vers 0 si et seulement si il existe $H > 0$ et $C > 0$ tel que

$$|g(h)| \leqslant Ch^q, \quad \forall h \in]-H, H[.$$

On note alors $g(h) = \mathcal{O}(h^q)$.

Definition 6.4

La différence $|y'(t^n) - (Dy)_n|$ est appelée **erreur de troncature au point** t^n. On dira que $|y'(t^n) - (Dy)_n|$ est d'ordre $p > 0$ si il existe une constance $C > 0$ telle que

$$|y'(t^n) - (Dy)_n| \leqslant Ch^p \iff |y'(t^n) - (Dy)_n| = \mathcal{O}(h^p).$$

On a démontré le lemme suivant

Lemme 6.5

Si $y \in \mathcal{C}^2([a,b])$ alors

$$\left| y'(t^n) - \frac{y(t^{n+1}) - y(t^n)}{h} \right| = \mathcal{O}(h), \quad \forall n \in [\![0, N-1]\!], \tag{7}$$

$$\left| y'(t^n) - \frac{y(t^n) - y(t^{n-1})}{h} \right| = \mathcal{O}(h), \quad \forall n \in [\![1, N]\!]. \tag{8}$$

Si $y \in \mathcal{C}^3([a,b])$ alors

$$\left| y'(t^n) - \frac{y(t^{n+1}) - y(t^{n-1})}{2h} \right| = \mathcal{O}(h^2), \quad \forall n \in [\![1, N-1]\!]. \tag{9}$$

Exercice 9

Soit $f \in \mathcal{C}^3([a,b];\mathbb{R})$. On note t^n, $n \in [\![0,N]\!]$, une discrétisation **régulière** de $[a,b]$ de pas h. On note $\boldsymbol{F} \in \mathbb{R}^{N+1}$ le vecteur défini par $F_{n+1} = f(t^n)$, $\forall n \in [\![0,N]\!]$.

Q.1

1. Déterminer en fonction de h et $\boldsymbol{F}$, un vecteur $\boldsymbol{V} \in \mathbb{R}^{N+1}$ vérifiant

$$V_{n+1} = f'(t^n) + \mathcal{O}(h), \quad \forall n \in [\![0,N]\!].$$

2. Ecrire une fonction algorithmique permettant, à partir du vecteur $\boldsymbol{F}$ et de la discrétisation régulière, de calculer le vecteur $\boldsymbol{V}$ précédant.

Q.2

1. Connaissant uniquement le vecteur $\boldsymbol{F}$, déterminer un vecteur $\boldsymbol{W} \in \mathbb{R}^{N+1}$ vérifiant

$$\boldsymbol{W}_n = f'(t^n) + \mathcal{O}(h^2), \quad \forall n \in [\![0,N]\!]$$

2. Ecrire une fonction algorithmique permettant, à partir du vecteur $\boldsymbol{F}$ et de la discrétisation régulière, de calculer le vecteur $\boldsymbol{W}$ précédant.

Application

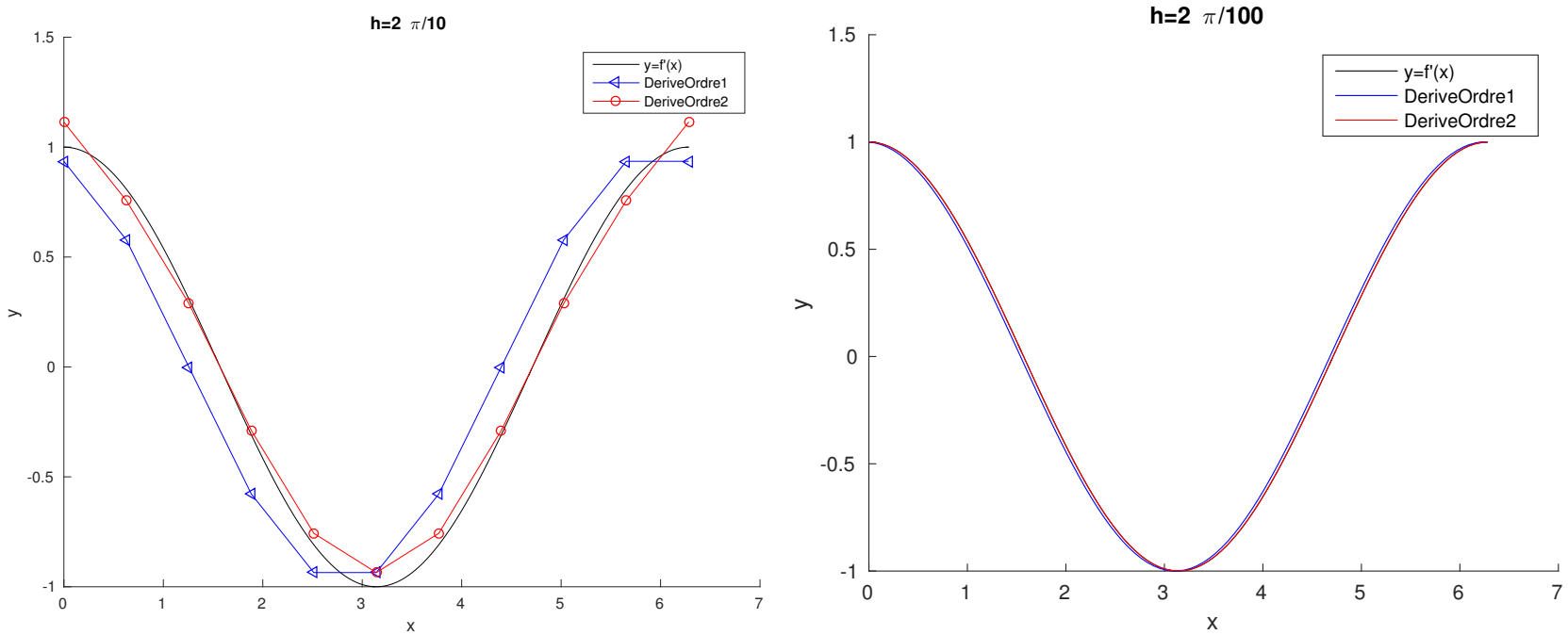


Figure: Représentation des dérivées numériques avec $f(x) := sin(x)$, $a = 0$, $b = 2\pi$. A gauche $h = \frac{2\pi}{10}$, à droite $h = \frac{2\pi}{100}$.

Application

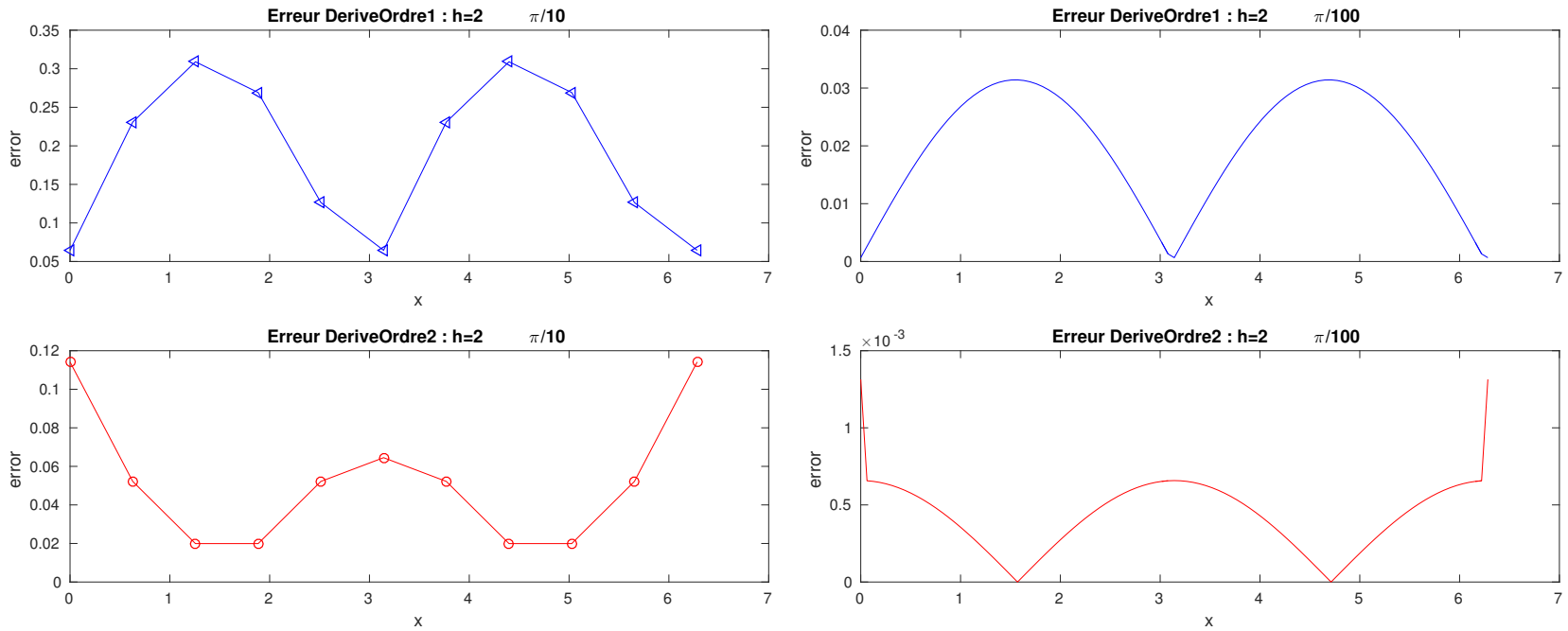


Figure: Erreur des dérivées numériques numériques avec $f(x) := sin(x)$, $a = 0$, $b = 2\pi$. A gauche $h = \frac{2\pi}{10}$, à droite $h = \frac{2\pi}{100}$.

- Dérivée ordre 1 :

  h a été divisé par 10 $\Longrightarrow$ l'erreur, $\mathcal{O}(h)$, divisée par 10.

- Dérivée ordre 2 :

  h a été divisé par 10 $\Longrightarrow$ l'erreur, $\mathcal{O}(h^2)$, divisée par 100.

Application

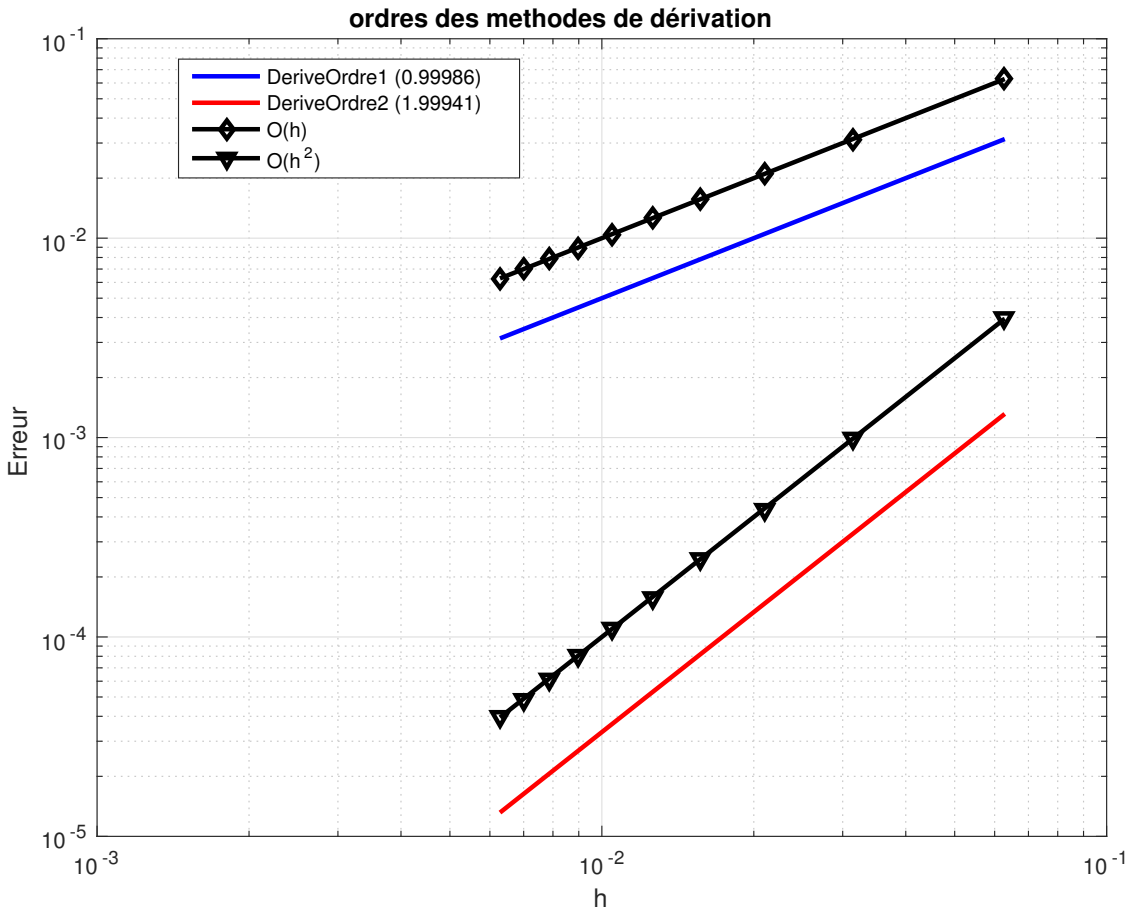


Figure: Dérivation numérique : mise en évidence de l'ordre des méthodes

Figure en échelle logarithmique : intérêt de *monter* en ordre.