

Cours d'analyse numérique

Laurent Seppecher & Grégory Vial

École Centrale de Lyon

Année 2022-2023 – Semestre 5

Informations générales

- ▶ Monitorat (romain.meaux@ecl21.ec-lyon.fr)

Informations générales

- ▶ Monitorat (romain.meaux@ecl21.ec-lyon.fr)
- ▶ Cours : pré-requis à regarder avant.

Informations générales

- ▶ Monitorat (romain.meaux@ecl21.ec-lyon.fr)
- ▶ Cours : pré-requis à regarder avant.
- ▶ TD : à préparer ! Utilisation des ordinateurs portables (Matlab).

Informations générales

- ▶ Monitorat (romain.meaux@ecl21.ec-lyon.fr)
- ▶ Cours : pré-requis à regarder avant.
- ▶ TD : à préparer ! Utilisation des ordinateurs portables (Matlab).
- ▶ Un tutoriel Matlab succinct est disponible (et à apprendre par cœur)

Informations générales

- ▶ Monitorat (romain.meaux@ecl21.ec-lyon.fr)
- ▶ Cours : pré-requis à regarder avant.
- ▶ TD : à préparer ! Utilisation des ordinateurs portables (Matlab).
- ▶ Un tutoriel Matlab succinct est disponible (et à apprendre par cœur)
- ▶ Contrôle des connaissances :

Informations générales

- ▶ Monitorat (romain.meaux@ecl21.ec-lyon.fr)
- ▶ Cours : pré-requis à regarder avant.
- ▶ TD : à préparer ! Utilisation des ordinateurs portables (Matlab).
- ▶ Un tutoriel Matlab succinct est disponible (et à apprendre par cœur)
- ▶ Contrôle des connaissances :
 - ▶ Savoir-faire 25% : mini-test (1H lors du TD4) **à confirmer**
Autorisés : documents cours/TD, programmes matlab.
 - ▶ Savoir 75% : examen terminal (1H30).
Autorisé : un A4 recto-verso manuscrit.



Introduction

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

► Contexte

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**
 - ▶ Pas de solutions explicites en général.



Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**
 - ▶ Pas de solutions explicites en général.
 - ▶ Besoin de calcul approché.

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**
 - ▶ Pas de solutions explicites en général.
 - ▶ Besoin de calcul approché.
- ▶ **Analyse numérique**

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**
 - ▶ Pas de solutions explicites en général.
 - ▶ Besoin de calcul approché.
- ▶ **Analyse numérique**
 - ▶ Mise au point de **méthodes numériques**.

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ Contexte

- ▶ Pas de solutions explicites en général.
- ▶ Besoin de calcul approché.

- ▶ Analyse numérique

- ▶ Mise au point de méthodes numériques.
- ▶ Preuve de convergence. Limitations des méthodes.

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**

- ▶ Pas de solutions explicites en général.
- ▶ Besoin de calcul approché.

- ▶ **Analyse numérique**

- ▶ Mise au point de **méthodes numériques**.
- ▶ Preuve de **convergence**. Limitations des méthodes.
- ▶ Quantification de la **vitesse** de convergence.

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**

- ▶ Pas de solutions explicites en général.
- ▶ Besoin de calcul approché.

- ▶ **Analyse numérique**

- ▶ Mise au point de **méthodes numériques**.
- ▶ Preuve de **convergence**. Limitations des méthodes.
- ▶ Quantification de la **vitesse** de convergence.
- ▶ Tests par **simulations** numériques

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**
 - ▶ Pas de solutions explicites en général.
 - ▶ Besoin de calcul approché.
- ▶ **Analyse numérique**
 - ▶ Mise au point de **méthodes numériques**.
 - ▶ Preuve de **convergence**. Limitations des méthodes.
 - ▶ Quantification de la **vitesse** de convergence.
 - ▶ Tests par **simulations** numériques
- ▶ **Calcul scientifique**

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**

- ▶ Pas de solutions explicites en général.
- ▶ Besoin de calcul approché.

- ▶ **Analyse numérique**

- ▶ Mise au point de **méthodes numériques**.
- ▶ Preuve de **convergence**. Limitations des méthodes.
- ▶ Quantification de la **vitesse** de convergence.
- ▶ Tests par **simulations** numériques

- ▶ **Calcul scientifique**

- ▶ Programmation sur problèmes réels.

Introduction

L'analyse numérique, c'est quoi ? ça sert à quoi ?

- ▶ **Contexte**

- ▶ Pas de solutions explicites en général.
- ▶ Besoin de calcul approché.

- ▶ **Analyse numérique**

- ▶ Mise au point de **méthodes numériques**.
- ▶ Preuve de **convergence**. Limitations des méthodes.
- ▶ Quantification de la **vitesse** de convergence.
- ▶ Tests par **simulations** numériques

- ▶ **Calcul scientifique**

- ▶ Programmation sur problèmes réels.
- ▶ Optimisation, parallélisation, etc.

Introduction

Exemples de vitesses de convergence

■ **Méthode numérique** = suite (u_k) .

► **Convergence**

$$u_k \longrightarrow u.$$

Introduction

Exemples de vitesses de convergence

■ **Méthode numérique** = suite (u_k) .

► **Convergence**

$$u_k \longrightarrow u.$$

► **Convergence polynomiale**

$$\exists C > 0, \exists p > 0, \forall k, \quad |u_k - u| \leq \frac{C}{k^p}$$

Introduction

Exemples de vitesses de convergence

■ **Méthode numérique** = suite (u_k) .

► **Convergence**

$$u_k \longrightarrow u.$$

► **Convergence polynomiale**

$$\exists C > 0, \exists p > 0, \forall k, \quad |u_k - u| \leq \frac{C}{k^p} \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C - p \ln k.$$

Introduction

Exemples de vitesses de convergence

■ **Méthode numérique** = suite (u_k) .

► **Convergence**

$$u_k \longrightarrow u.$$

► **Convergence polynomiale**

$$\exists C > 0, \exists p > 0, \forall k, \quad |u_k - u| \leq \frac{C}{k^p} \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C - p \ln k.$$

► **Convergence géométrique**

$$\exists C > 0, \exists \rho \in [0, 1[, \forall k, \quad |u_k - u| \leq C\rho^k$$



Introduction

Exemples de vitesses de convergence

■ **Méthode numérique** = suite (u_k) .

► **Convergence**

$$u_k \longrightarrow u.$$

► **Convergence polynomiale**

$$\exists C > 0, \exists p > 0, \forall k, \quad |u_k - u| \leq \frac{C}{k^p} \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C - p \ln k.$$

► **Convergence géométrique**

$$\exists C > 0, \exists \rho \in [0, 1[, \forall k, \quad |u_k - u| \leq C \rho^k \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C + k \ln \rho.$$



Introduction

Exemples de vitesses de convergence

■ **Méthode numérique** = suite (u_k) .

► **Convergence**

$$u_k \longrightarrow u.$$

► **Convergence polynomiale**

$$\exists C > 0, \exists p > 0, \forall k, \quad |u_k - u| \leq \frac{C}{k^p} \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C - p \ln k.$$

► **Convergence géométrique**

$$\exists C > 0, \exists \rho \in [0, 1[, \forall k, \quad |u_k - u| \leq C \rho^k \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C + k \ln \rho.$$

► **Convergence quadratique**

$$\exists C > 0, \exists \rho \in [0, 1[, \forall k, \quad |u_k - u| \leq C \rho^{2^k}$$



Introduction

Exemples de vitesses de convergence

■ **Méthode numérique** = suite (u_k) .

► **Convergence**

$$u_k \longrightarrow u.$$

► **Convergence polynomiale**

$$\exists C > 0, \exists p > 0, \forall k, \quad |u_k - u| \leq \frac{C}{k^p} \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C - p \ln k.$$

► **Convergence géométrique**

$$\exists C > 0, \exists \rho \in [0, 1[, \forall k, \quad |u_k - u| \leq C \rho^k \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C + k \ln \rho.$$

► **Convergence quadratique**

$$\exists C > 0, \exists \rho \in [0, 1[, \forall k, \quad |u_k - u| \leq C \rho^{2^k} \quad \text{i.e.} \quad \ln |u_k - u| \leq \ln C + 2^k \ln \rho.$$



Plan du cours

- ▶ **Cours 1.** Systèmes linéaires et calcul de valeurs propres
- ▶ **Cours 2.** Interpolation et intégration numérique
- ▶ **Cours 3.** Optimisation numérique
- ▶ **Cours 4.** Approximation numérique des équations différentielles
- ▶ **Cours 5.** Discrétisation des EDP : l'équation de Laplace
- ▶ **Cours 6.** Discrétisation des EDP : l'équation de transport

Un conseil



Un conseil



« On ne peut pas faire de physique sans un bon niveau en mathématiques. Mais c'est vrai quelle que soit la science. Je vais même être caricatural : si vous voulez trouver du travail facilement, faites des maths. »

COURS 1

Systèmes linéaires

Calcul de valeurs propres

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version production)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_j nécessite a_{ij} biens P_i ,
- ▶ B_i produit une quantité q_i .

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version production)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_j nécessite a_{ij} biens P_i ,
- ▶ B_i produit une quantité q_i .

Bilan pour la branche B_i :

$$q_i - \sum_{j=1}^d a_{ij} q_j = (Mq)_i \quad \text{avec} \quad M = I - A.$$

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version production)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_j nécessite a_{ij} biens P_i ,
- ▶ B_i produit une quantité q_i .

Bilan pour la branche B_i :

$$q_i - \sum_{j=1}^d a_{ij} q_j = (Mq)_i \quad \text{avec} \quad M = I - A.$$

Équilibre offre-demande :

$$Mq = \delta.$$

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version production)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_j nécessite a_{ij} biens P_i ,
- ▶ B_i produit une quantité q_i .

Bilan pour la branche B_i :

$$q_i - \sum_{j=1}^d a_{ij} q_j = (Mq)_i \quad \text{avec} \quad M = I - A.$$

Équilibre offre-demande :

$$Mq = \delta.$$

Question : quelle influence de la demande sur les quantités produites ?

$\Rightarrow \text{système linéaire } Mq = \delta.$

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version bénéfice)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_i nécessite
 - ▶ a_{ji} biens P_j ,
 - ▶ ℓ_i travailleurs au salaire w .
- ▶ B_i vend au prix π_i .
- ▶ salaire fonction linéaire des prix :
$$w = \sum c_j \pi_j.$$

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version bénéfice)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_i nécessite
 - ▶ a_{ji} biens P_j ,
 - ▶ ℓ_i travailleurs au salaire w .
- ▶ B_i vend au prix π_i .
- ▶ salaire fonction linéaire des prix :
 $w = \sum c_j \pi_j$.

Coût de production de la branche B_i :

$$\sum_{j=1}^d a_{ji} \pi_j + \ell_i w = \sum_{j=1}^d (a_{ji} + \ell_i c_j) \pi_j = (N\pi)_i \quad \text{avec} \quad N = A^T + \ell c^T.$$

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version bénéfice)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_i nécessite
 - ▶ a_{ji} biens P_j ,
 - ▶ ℓ_i travailleurs au salaire w .
- ▶ B_i vend au prix π_i .
- ▶ salaire fonction linéaire des prix :
 $w = \sum c_j \pi_j$.

Coût de production de la branche B_i :

$$\sum_{j=1}^d a_{ji} \pi_j + \ell_i w = \sum_{j=1}^d (a_{ji} + \ell_i c_j) \pi_j = (N\pi)_i \quad \text{avec} \quad N = A^T + \ell c^T.$$

Profit relatif de la branche B_i : $\tau_i = \frac{\pi_i - (N\pi)_i}{(N\pi)_i}$.

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version bénéfice)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_i nécessite
 - ▶ a_{ji} biens P_j ,
 - ▶ ℓ_i travailleurs au salaire w .
- ▶ B_i vend au prix π_i .
- ▶ salaire fonction linéaire des prix :
 $w = \sum c_j \pi_j$.

Coût de production de la branche B_i :

$$\sum_{j=1}^d a_{ji} \pi_j + \ell_i w = \sum_{j=1}^d (a_{ji} + \ell_i c_j) \pi_j = (N\pi)_i \quad \text{avec} \quad N = A^T + \ell c^T.$$

Profit relatif de la branche B_i : $\tau_i = \frac{\pi_i - (N\pi)_i}{(N\pi)_i}$.

Question : profit équilibré parmi les branches ?

Analyse numérique matricielle

Le modèle de Léontiev en comptabilité nationale (version bénéfice)



- ▶ d branches : B_i produit P_i .
- ▶ 1 bien P_i nécessite
 - ▶ a_{ji} biens P_j ,
 - ▶ ℓ_i travailleurs au salaire w .
- ▶ B_i vend au prix π_i .
- ▶ salaire fonction linéaire des prix :
 $w = \sum c_j \pi_j$.

Coût de production de la branche B_i :

$$\sum_{j=1}^d a_{ji} \pi_j + \ell_i w = \sum_{j=1}^d (a_{ji} + \ell_i c_j) \pi_j = (N\pi)_i \quad \text{avec} \quad N = A^T + \ell c^T.$$

Profit relatif de la branche B_i : $\tau_i = \frac{\pi_i - (N\pi)_i}{(N\pi)_i}$.

Question : profit équilibré parmi les branches ?

$$\Rightarrow \text{Problème aux valeurs propres } N\pi = \frac{1}{1+\tau} \pi$$

Analyse numérique matricielle

Résolution de systèmes linéaires



Analyse numérique matricielle

Résolution de systèmes linéaires

- **En théorie...** Soit A est inversible. $\exists ! x$ t.q. $Ax = b$. On a $x = A^{-1}b$.

Analyse numérique matricielle

Résolution de systèmes linéaires

► En théorie... Soit A est inversible. $\exists! x$ t.q. $Ax = b$. On a $x = A^{-1}b$.

► En pratique...

► Soit $H \in \mathbb{R}^{d \times d}$ t.q. $H_{ij} = \frac{1}{i+j-1}$.



[*hilbert.m*]

Rq. H est inversible et $H^{-1} \in \mathbb{Z}^{d \times d}$ (*formule explicite*).

Analyse numérique matricielle

Résolution de systèmes linéaires

► En théorie... Soit A est inversible. $\exists ! x$ t.q. $Ax = b$. On a $x = A^{-1}b$.

► En pratique...

► Soit $H \in \mathbb{R}^{d \times d}$ t.q. $H_{ij} = \frac{1}{i+j-1}$.



[hilbert.m]

Rq. H est inversible et $H^{-1} \in \mathbb{Z}^{d \times d}$ (formule explicite).

$\Rightarrow$ comment expliquer le phénomène ?

Analyse numérique matricielle

Résolution de systèmes linéaires

► **En théorie...** Soit A est inversible. $\exists !x$ t.q. $Ax = b$. On a $x = A^{-1}b$.

► **En pratique...**

► Soit $H \in \mathbb{R}^{d \times d}$ t.q. $H_{ij} = \frac{1}{i+j-1}$.  `[hilbert.m]`

Rq. H est inversible et $H^{-1} \in \mathbb{Z}^{d \times d}$ (*formule explicite*).

$\Rightarrow$ **comment expliquer le phénomène ?**

► Formules de Cramer :

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad \text{avec } A_i = (C_1 | \dots | C_{i-1} | b | C_{i+1} | \dots | C_d).$$

Analyse numérique matricielle

Résolution de systèmes linéaires

► **En théorie...** Soit A est inversible. $\exists ! x$ t.q. $Ax = b$. On a $x = A^{-1}b$.

► **En pratique...**

► Soit $H \in \mathbb{R}^{d \times d}$ t.q. $H_{ij} = \frac{1}{i+j-1}$.



[hilbert.m]

Rq. H est inversible et $H^{-1} \in \mathbb{Z}^{d \times d}$ (*formule explicite*).

⇒ **comment expliquer le phénomène ?**

► Formules de Cramer :

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad \text{avec } A_i = (C_1 | \dots | C_{i-1} | b | C_{i+1} | \dots | C_d).$$

Coût si les **det** sont calculés par dvpt p/r lignes : $\simeq (d+1)!$.

Analyse numérique matricielle

Résolution de systèmes linéaires

► **En théorie...** Soit A est inversible. $\exists ! x$ t.q. $Ax = b$. On a $x = A^{-1}b$.

► **En pratique...**

► Soit $H \in \mathbb{R}^{d \times d}$ t.q. $H_{ij} = \frac{1}{i+j-1}$.



[hilbert.m]

Rq. H est inversible et $H^{-1} \in \mathbb{Z}^{d \times d}$ (formule explicite).

⇒ **comment expliquer le phénomène ?**

► Formules de Cramer :

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad \text{avec } A_i = (C_1 | \dots | C_{i-1} | b | C_{i+1} | \dots | C_d).$$

Coût si les $\det$ sont calculés par dvpt p/r lignes : $\simeq (d+1)!$.

⇒ **quels algorithmes ? quelles performances ?**

Analyse numérique matricielle

Conditionnement des systèmes linéaires

Question : comment quantifier la propagation des erreurs ?

Question : comment quantifier la propagation des erreurs ?

Proposition. Soient $A \in \mathbb{R}^{d \times d}$ inversible, $b, \delta b \in \mathbb{R}^d$.

On note $x \in \mathbb{R}^d$ et $x + \delta x \in \mathbb{R}^d$ les solutions de

$$Ax = b, \quad A(x + \delta x) = b + \delta b.$$

Alors

$$\frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|},$$

avec $\text{cond}(A) = \|A\| \times \|A^{-1}\|$.

Question : comment quantifier la propagation des erreurs ?

Proposition. Soient $A \in \mathbb{R}^{d \times d}$ inversible, $b, \delta b \in \mathbb{R}^d$.

On note $x \in \mathbb{R}^d$ et $x + \delta x \in \mathbb{R}^d$ les solutions de

$$Ax = b, \quad A(x + \delta x) = b + \delta b.$$

Alors

$$\frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|},$$

avec $\text{cond}(A) = \|A\| \times \|A^{-1}\|$.

- ▶ $\|A\|$ est la norme subordonnée à $\|x\|$.
- ▶ $\text{cond}(A)$ mesure la propagation des erreurs relatives.
- ▶ Si $\|\cdot\| = \|\cdot\|_p$, on note cond_p .

Question : comment quantifier la propagation des erreurs ?

Proposition. Soient $A \in \mathbb{R}^{d \times d}$ inversible, $b, \delta b \in \mathbb{R}^d$.

On note $x \in \mathbb{R}^d$ et $x + \delta x \in \mathbb{R}^d$ les solutions de

$$Ax = b, \quad A(x + \delta x) = b + \delta b.$$

Alors

$$\frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|},$$

avec $\text{cond}(A) = \|A\| \times \|A^{-1}\|$.

Preuve. Par différence $A(\delta x) = \delta b$ d'où $\|\delta x\| \leq \|A^{-1}\| \times \|\delta b\|$.

Par ailleurs $b = Ax$ d'où $\|b\| \leq \|A\| \times \|x\|$, soit $\frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}$.

D'où le résultat par produit. □

Proposition. Soit $A \in \mathbb{R}^{d \times d}$ inversible.

► $\forall \|\cdot\|, \quad \text{cond}(A) \geq 1.$

► Si A est symétrique,

$$\text{cond}_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

► Si A est orthogonale, $\text{cond}_2(A) = 1.$

Proposition. Soit $A \in \mathbb{R}^{d \times d}$ inversible.

► $\forall \|\cdot\|, \quad \text{cond}(A) \geq 1.$

► Si A est symétrique,

$$\text{cond}_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

► Si A est orthogonale, $\text{cond}_2(A) = 1.$

Rmq. Si $\text{cond}(A)$ est « grand », on dit que A est *mal conditionnée*.

Proposition. Soit $A \in \mathbb{R}^{d \times d}$ inversible.

► $\forall \|\cdot\|, \quad \text{cond}(A) \geq 1.$

► Si A est symétrique,

$$\text{cond}_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

► Si A est orthogonale, $\text{cond}_2(A) = 1.$

Rmq. Si $\text{cond}(A)$ est « grand », on dit que A est *mal conditionnée*.

d	3	5	10	20
$\text{cond}_2(H)$	524	4.8×10^5	1.6×10^{13}	2.5×10^{18}

Table – Conditionnement de la matrice de Hilbert.

Analyse numérique matricielle

L'algorithme du pivot de Gauss

- ▶ Principe.
 - ▶ éliminer les inconnues successivement.
 - ▶ on aboutit à un système triangulaire.

Analyse numérique matricielle

L'algorithme du pivot de Gauss

- ▶ Principe.
 - ▶ éliminer les inconnues successivement.
 - ▶ on aboutit à un système triangulaire.
- ▶ Description.

Analyse numérique matricielle

L'algorithme du pivot de Gauss

- ▶ **Principe.**
 - ▶ éliminer les inconnues successivement.
 - ▶ on aboutit à un système triangulaire.
- ▶ **Description.**

Initialisation

$$\left| \begin{array}{cccc} a_{1,1} & a_{1,2} & \dots & a_{1,d} \\ a_{2,1} & a_{2,2} & \dots & a_{2,d} \\ \vdots & & & \\ a_{d,1} & a_{d,2} & \dots & a_{d,d} \end{array} \right| \left| \begin{array}{c} b_1 \\ b_2 \\ \vdots \\ b_d \end{array} \right|$$

Analyse numérique matricielle

L'algorithme du pivot de Gauss

► Principe.

- éliminer les inconnues successivement.
- on aboutit à un système triangulaire.

► Description.

Initialisation

$$\left| \begin{array}{cccc} a_{1,1}^{[0]} & a_{1,2}^{[0]} & \dots & a_{1,d}^{[0]} \\ a_{2,1}^{[0]} & a_{2,2}^{[0]} & \dots & a_{2,d}^{[0]} \\ \vdots & & & \\ a_{d,1}^{[0]} & a_{d,2}^{[0]} & \dots & a_{d,d}^{[0]} \end{array} \right| \quad \left| \begin{array}{c} b_1^{[0]} \\ b_2^{[0]} \\ \vdots \\ b_d^{[0]} \end{array} \right|$$

Analyse numérique matricielle

L'algorithme du pivot de Gauss

► Principe.

- éliminer les inconnues successivement.
- on aboutit à un système triangulaire.

► Description.

Étape 1.1 : si $a_{1,1}^{[0]} = 0$, on permute $L_1 \leftrightarrow L_k$ t.q. $a_{k,1}^{[0]} \neq 0$.

$$\left| \begin{array}{cccc} a_{1,1}^{[0]} & a_{1,2}^{[0]} & \dots & a_{1,d}^{[0]} \\ a_{2,1}^{[0]} & a_{2,2}^{[0]} & \dots & a_{2,d}^{[0]} \\ \vdots & & & \\ a_{d,1}^{[0]} & a_{d,2}^{[0]} & \dots & a_{d,d}^{[0]} \end{array} \right| \quad \left| \begin{array}{c} b_1^{[0]} \\ b_2^{[0]} \\ \vdots \\ b_d^{[0]} \end{array} \right|$$

Analyse numérique matricielle

L'algorithme du pivot de Gauss

► Principe.

- éliminer les inconnues successivement.
- on aboutit à un système triangulaire.

► Description.

Étape 1.2 : $\tilde{a}_{1,1}^{[0]} \neq 0$; on effectue $\boxed{L_k \leftarrow L_k - r_k L_1}$ avec $r_k = \frac{\tilde{a}_{k,1}^{[0]}}{\tilde{a}_{1,1}^{[0]}}$.

$$\left| \begin{array}{cccc} \tilde{a}_{1,1}^{[0]} & \tilde{a}_{1,2}^{[0]} & \dots & \tilde{a}_{1,d}^{[0]} \\ \tilde{a}_{2,1}^{[0]} & \tilde{a}_{2,2}^{[0]} & \dots & \tilde{a}_{2,d}^{[0]} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{d,1}^{[0]} & \tilde{a}_{d,2}^{[0]} & \dots & \tilde{a}_{d,d}^{[0]} \end{array} \right| \quad \left| \begin{array}{c} \tilde{b}_1^{[0]} \\ \tilde{b}_2^{[0]} \\ \vdots \\ \tilde{b}_d^{[0]} \end{array} \right|$$

Analyse numérique matricielle

L'algorithme du pivot de Gauss

► Principe.

- éliminer les inconnues successivement.
- on aboutit à un système triangulaire.

► Description.

Étape 1.2 : $\tilde{a}_{1,1}^{[0]} \neq 0$; on effectue $\boxed{L_k \leftarrow L_k - r_k L_1}$ avec $r_k = \frac{\tilde{a}_{k,1}^{[0]}}{\tilde{a}_{1,1}^{[0]}}$.

$$\left| \begin{array}{cccc} \tilde{a}_{1,1}^{[0]} & \tilde{a}_{1,2}^{[0]} & \dots & \tilde{a}_{1,d}^{[0]} \\ 0 & a_{2,2}^{[1]} & \dots & a_{2,d}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{d,2}^{[1]} & \dots & a_{d,d}^{[1]} \end{array} \right| \quad \left| \begin{array}{c} \tilde{b}_1^{[0]} \\ b_2^{[1]} \\ \vdots \\ b_d^{[1]} \end{array} \right|$$

Analyse numérique matricielle

L'algorithme du pivot de Gauss

► Principe.

- éliminer les inconnues successivement.
- on aboutit à un système triangulaire.

► Description.

Étape 2 : On recommence avec la sous-matrice !

$$\left| \begin{array}{cccc} \tilde{a}_{1,1}^{[0]} & \tilde{a}_{1,2}^{[0]} & \dots & \tilde{a}_{1,d}^{[0]} \\ 0 & a_{2,2}^{[1]} & \dots & a_{2,d}^{[1]} \\ 0 & \vdots & & \\ \vdots & & & \\ 0 & a_{d,2}^{[1]} & \dots & a_{d,d}^{[1]} \end{array} \right| \quad \left| \begin{array}{c} \tilde{b}_1^{[0]} \\ b_2^{[1]} \\ \vdots \\ b_d^{[1]} \end{array} \right|$$

Analyse numérique matricielle

L'algorithme du pivot de Gauss

- ▶ Principe.

- ▶ éliminer les inconnues successivement.
- ▶ on aboutit à un système triangulaire.

- ▶ Synthèse.

- ▶ Après $d - 1$ étapes, on est ramené à $Uy = \bar{c}$ avec U tri. sup.
- ▶ $Uy = \bar{c}$ est résolu par « remontée ».

Analyse numérique matricielle

L'algorithme du pivot de Gauss

► Principe.

- éliminer les inconnues successivement.
- on aboutit à un système triangulaire.

► Synthèse.

- Après $d - 1$ étapes, on est ramené à $Uy = \bar{c}$ avec U tri. sup.
- $Uy = \bar{c}$ est résolu par « remontée ».

► Coût de calcul.

- L'étape k affecte $(d - k)^2$ coefficients de la matrice.

- Coût total $\simeq \sum_{k=1}^{d-1} (d - k)^2 = \mathcal{O}(d^3)$.

- *Les opérations sur le second membre et la remontée sont négligeables.*

Analyse numérique matricielle

La décomposition LU

► Interprétation matricielle des opérations élémentaires

► Permutation $L_i \leftrightarrow L_k \iff A \mapsto PA$

► Transvection $L_k \leftarrow L_k - rL_i \iff A \mapsto TA$ (pour $k > i$)

$$P = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 0 & & 1 \\ & & & \ddots & \\ & 1 & & & 0 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & -r & & & 1 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}$$

Analyse numérique matricielle

La décomposition LU

► Interprétation matricielle des opérations élémentaires

► Permutation $\boxed{L_i \leftrightarrow L_k} \iff \boxed{A \mapsto PA}$

► Transvection $\boxed{L_k \leftarrow L_k - rL_i} \iff \boxed{A \mapsto TA}$ (pour $k > i$)

$$P = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 0 & & 1 \\ & & & \ddots & \\ & 1 & & & 0 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & -r & & & 1 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}$$

► Écriture matricielle de l'algorithme de Gauss

$$\boxed{A \mapsto M_1 \times M_2 \times \cdots \times M_q A = U} \quad \text{avec les } M_p \text{ de type } P \text{ ou } T.$$

► Interprétation matricielle des opérations élémentaires

► Permutation $\boxed{L_i \leftrightarrow L_k} \iff \boxed{A \mapsto PA}$

► Transvection $\boxed{L_k \leftarrow L_k - rL_i} \iff \boxed{A \mapsto TA}$ (pour $k > i$)

$$P^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 0 & & 1 \\ & & & \ddots & \\ & 1 & & & 0 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & +r & & 1 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}$$

► Écriture matricielle de l'algorithme de Gauss

$$\boxed{A \mapsto M_1 \times M_2 \times \cdots \times M_q A = U} \quad \text{avec les } M_p \text{ de type } P \text{ ou } T.$$

► Interprétation matricielle des opérations élémentaires

► Permutation $\boxed{L_i \leftrightarrow L_k} \iff \boxed{A \mapsto PA}$

► Transvection $\boxed{L_k \leftarrow L_k - rL_i} \iff \boxed{A \mapsto TA}$ (pour $k > i$)

$$P^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 0 & & 1 \\ & & & \ddots & \\ & 1 & & & 0 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & +r & & 1 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}$$

► Écriture matricielle de l'algorithme de Gauss

$$\boxed{A = N_1 \times N_2 \times \cdots \times N_q \times U} \quad \text{avec les } N_p \text{ de type } P \text{ ou } T!$$

Analyse numérique matricielle

La décomposition LU

► Interprétation matricielle des opérations élémentaires

► Permutation $\boxed{L_i \leftrightarrow L_k} \iff \boxed{A \mapsto PA}$

► Transvection $\boxed{L_k \leftarrow L_k - rL_i} \iff \boxed{A \mapsto TA}$ (pour $k > i$)

► Écriture matricielle de l'algorithme de Gauss

$\boxed{A = N_1 \times N_2 \times \cdots \times N_q \times U}$ avec les N_p de type P ou T !

- Si aucune permutation, les N_p sont tri. inf.

Analyse numérique matricielle

La décomposition LU

► Interprétation matricielle des opérations élémentaires

► Permutation $\boxed{L_i \leftrightarrow L_k} \iff \boxed{A \mapsto PA}$

► Transvection $\boxed{L_k \leftarrow L_k - rL_i} \iff \boxed{A \mapsto TA}$ (pour $k > i$)

► Écriture matricielle de l'algorithme de Gauss

$\boxed{A = N_1 \times N_2 \times \cdots \times N_q \times U}$ avec les N_p de type P ou T !

► Si aucune permutation, les N_p sont tri. inf.

► Dans ce cas,

$$A = LU \quad \text{avec } L \text{ tri. inf. et } U \text{ tri. sup.}$$

Analyse numérique matricielle

La décomposition LU

► Interprétation matricielle des opérations élémentaires

► Permutation $\boxed{L_i \leftrightarrow L_k} \iff \boxed{A \mapsto PA}$

► Transvection $\boxed{L_k \leftarrow L_k - rL_i} \iff \boxed{A \mapsto TA}$ (pour $k > i$)

► Écriture matricielle de l'algorithme de Gauss

$\boxed{A = N_1 \times N_2 \times \cdots \times N_q \times U}$ avec les N_p de type P ou T !

► Si aucune permutation, les N_p sont tri. inf.

► Dans ce cas,

$$A = LU \quad \text{avec } L \text{ tri. inf. et } U \text{ tri. sup.}$$

► Comment caractériser ce cas ?

Analyse numérique matricielle

La décomposition LU

Définition. Soit $A \in \mathbb{R}^{d \times d}$. On appelle **mineur fondamental** de taille $k \leq d$ le déterminant

$$\Delta_k = \det \left[(A_{ij})_{1 \leq i, j \leq k} \right].$$

Théorème. Soit $A \in \mathbb{R}^{d \times d}$ telle que

$$\forall k \leq d, \quad \Delta_k \neq 0.$$

Alors $\exists (L, U)$ t.q.

- ▶ L tri. inf. avec diagonale unité.
- ▶ U tri. sup.
- ▶ $A = LU$.

Analyse numérique matricielle

La décomposition LU

Intérêt algorithmique de la décomposition LU .



Analyse numérique matricielle

La décomposition LU

Intérêt algorithmique de la décomposition LU .

- ▶ On doit résoudre $Ax_p = b_p$ pour de nombreux $p = 1, \dots, P$.
 - ▶ 1 fois : $A = LU$.
 - ▶ P fois : $Ly_p = b_p$ et $Ux_p = y_p$.
 - ▶ Coût total : $\mathcal{O}(d^3 + Pd^2)$.

Analyse numérique matricielle

La décomposition LU

Intérêt algorithmique de la décomposition LU .

- ▶ On doit résoudre $Ax_p = b_p$ pour de nombreux $p = 1, \dots, P$.
 - ▶ 1 fois : $A = LU$.
 - ▶ P fois : $Ly_p = b_p$ et $Ux_p = y_p$.
 - ▶ Coût total : $\mathcal{O}(d^3 + Pd^2)$.
- ▶ Mieux que P fois Gauss.

Analyse numérique matricielle

La décomposition LU

Intérêt algorithmique de la décomposition LU .

- ▶ On doit résoudre $Ax_p = b_p$ pour de nombreux $p = 1, \dots, P$.
 - ▶ 1 fois : $A = LU$.
 - ▶ P fois : $Ly_p = b_p$ et $Ux_p = y_p$.
 - ▶ Coût total : $\mathcal{O}(d^3 + Pd^2)$.
- ▶ Mieux que P fois Gauss.
- ▶ Vraiment utile seulement lorsque les b_p sont connus seulement successivement. . .

Analyse numérique matricielle

La décomposition LU

Intérêt algorithmique de la décomposition LU .

- ▶ On doit résoudre $Ax_p = b_p$ pour de nombreux $p = 1, \dots, P$.
 - ▶ 1 fois : $A = LU$.
 - ▶ P fois : $Ly_p = b_p$ et $Ux_p = y_p$.
 - ▶ Coût total : $\mathcal{O}(d^3 + Pd^2)$.
- ▶ Mieux que P fois Gauss.
- ▶ Vraiment utile seulement lorsque les b_p sont connus seulement successivement. . .

Extensions, optimisations

- ▶ Si A est symétrique, définie positive, $A = BB^T$ avec B tri.inf.
- ▶ Si A a une structure particulière, les calculs peuvent être simplifiés.

Analyse numérique matricielle

La décomposition LU

Intérêt algorithmique de la décomposition LU .

- ▶ On doit résoudre $Ax_p = b_p$ pour de nombreux $p = 1, \dots, P$.
 - ▶ 1 fois : $A = LU$.
 - ▶ P fois : $Ly_p = b_p$ et $Ux_p = y_p$.
 - ▶ Coût total : $\mathcal{O}(d^3 + Pd^2)$.
- ▶ Mieux que P fois Gauss.
- ▶ Vraiment utile seulement lorsque les b_p sont connus seulement successivement. . .

Extensions, optimisations

- ▶ Si A est symétrique, définie positive, $A = BB^T$ avec B tri.inf.
- ▶ Si A a une structure particulière, les calculs peuvent être simplifiés.



Méthode utilisée par matlab ? Voir `help \`

Analyse numérique matricielle

La décomposition LU

► Un système linéaire...

$$\begin{cases} \alpha x_1 + x_2 + \dots + x_d = 1 \\ x_1 + \alpha x_2 = 1 \\ x_1 + \alpha x_3 = 1 \\ \vdots \\ x_1 + \alpha x_d = 1 \end{cases}$$

► Un système linéaire...

$$\begin{cases} \alpha x_1 + x_2 + \dots + x_d = 1 \\ x_1 + \alpha x_2 = 1 \\ x_1 + \alpha x_3 = 1 \\ \vdots \quad \dots \quad \vdots \\ x_1 + \alpha x_d = 1 \end{cases}$$

$$\begin{cases} \alpha y_d + y_{d-1} + \dots + y_1 = 1 \\ y_d + \alpha y_{d-1} = 1 \\ y_d + \alpha y_{d-2} = 1 \\ \vdots \quad \dots \quad \vdots \\ y_d + \alpha y_1 = 1 \end{cases}$$

► Un système linéaire...

$$\begin{cases} \alpha x_1 + x_2 + \dots + x_d = 1 \\ x_1 + \alpha x_2 = 1 \\ x_1 + \alpha x_3 = 1 \\ \vdots \quad \dots \quad \vdots \\ x_1 + \alpha x_d = 1 \end{cases}$$

$$\begin{cases} y_1 + y_2 + \dots + \alpha y_d = 1 \\ \alpha y_{d-1} + y_d = 1 \\ \alpha y_{d-2} + y_d = 1 \\ \vdots \quad \dots \quad \vdots \\ \alpha y_1 + y_d = 1 \end{cases}$$

► Un système linéaire...

$$\begin{cases} \alpha x_1 + x_2 + \dots + x_d = 1 \\ x_1 + \alpha x_2 = 1 \\ x_1 + \alpha x_3 = 1 \\ \vdots \quad \dots \quad \vdots \\ x_1 + \alpha x_d = 1 \end{cases}$$

$$\begin{cases} \alpha y_1 + y_d = 1 \\ \vdots \quad \dots \quad \vdots \\ \alpha y_{d-2} + y_d = 1 \\ \alpha y_{d-1} + y_d = 1 \\ y_1 + y_2 + \dots + \alpha y_d = 1 \end{cases}$$

Analyse numérique matricielle

La décomposition LU

► Un système linéaire...

$$\begin{cases} \alpha x_1 + x_2 + \dots + x_d = 1 \\ x_1 + \alpha x_2 = 1 \\ x_1 + \alpha x_3 = 1 \\ \vdots \\ x_1 + \alpha x_d = 1 \end{cases}$$

$$\begin{cases} \alpha y_1 + y_d = 1 \\ \vdots \\ \alpha y_{d-2} + y_d = 1 \\ \alpha y_{d-1} + y_d = 1 \\ y_1 + y_2 + \dots + \alpha y_d = 1 \end{cases}$$

► ... deux matrices...

$$A_1 = \begin{pmatrix} \alpha & 1 & \dots & 1 \\ 1 & \alpha & & \\ \vdots & & \ddots & \\ 1 & & & \alpha \end{pmatrix}$$

$$A_2 = \begin{pmatrix} \alpha & & & 1 \\ & \alpha & & 1 \\ & & \ddots & 1 \\ 1 & \dots & 1 & \alpha \end{pmatrix}$$

Analyse numérique matricielle

La décomposition LU

- Un système linéaire... ($\alpha > d$)

$$\begin{cases} \alpha x_1 + x_2 + \dots + x_d = 1 \\ x_1 + \alpha x_2 = 1 \\ x_1 + \alpha x_3 = 1 \\ \vdots \\ x_1 + \alpha x_d = 1 \end{cases} \quad \begin{cases} \alpha y_1 + y_d = 1 \\ \vdots \\ \alpha y_{d-2} + y_d = 1 \\ \alpha y_{d-1} + y_d = 1 \\ y_1 + y_2 + \dots + \alpha y_d = 1 \end{cases}$$

- ... deux matrices...

$$A_1 = \begin{pmatrix} \alpha & 1 & \dots & 1 \\ 1 & \alpha & & \\ \vdots & & \ddots & \\ 1 & & & \alpha \end{pmatrix}$$

$$A_2 = \begin{pmatrix} \alpha & & & 1 \\ & \alpha & & 1 \\ & & \ddots & 1 \\ 1 & \dots & 1 & \alpha \end{pmatrix}$$

- ... deux coûts si on utilise LU!



[LUfleches.m]

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires



Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

- **But** : construire une suite $(x^{(n)})$ qui converge vers x t.q. $Ax = b$.

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

- **But** : construire une suite $(x^{(n)})$ qui converge vers x t.q. $Ax = b$.
- **Idée** : si $A = M - N$ avec M inversible, alors

$$Ax = b \iff x = \underbrace{M^{-1}(Nx + b)}_{\varphi(x)}.$$

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

► **But** : construire une suite $(x^{(n)})$ qui converge vers x t.q. $Ax = b$.

► **Idée** : si $A = M - N$ avec M inversible, alors

$$Ax = b \iff x = \underbrace{M^{-1}(Nx + b)}_{\varphi(x)}.$$

Problème de point fixe : soit $x^{(0)} \in \mathbb{R}^d$, on définit

$$x^{(n+1)} = \varphi(x^{(n)}).$$

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

► **But** : construire une suite $(x^{(n)})$ qui converge vers x t.q. $Ax = b$.

► **Idée** : si $A = M - N$ avec M inversible, alors

$$Ax = b \iff x = \underbrace{M^{-1}(Nx + b)}_{\varphi(x)}.$$

Problème de point fixe : soit $x^{(0)} \in \mathbb{R}^d$, on définit

$$x^{(n+1)} = \varphi(x^{(n)}).$$

Théorème. La suite $(x^{(n)})$ converge vers la solution de $Ax = b$ pour b quelconque ssi $\rho(M^{-1}N) < 1$.

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

► **But** : construire une suite $(x^{(n)})$ qui converge vers x t.q. $Ax = b$.

► **Idée** : si $A = M - N$ avec M inversible, alors

$$Ax = b \iff x = \underbrace{M^{-1}(Nx + b)}_{\varphi(x)}.$$

Problème de point fixe : soit $x^{(0)} \in \mathbb{R}^d$, on définit

$$x^{(n+1)} = \varphi(x^{(n)}).$$

Théorème. La suite $(x^{(n)})$ converge vers la solution de $Ax = b$ pour b quelconque ssi $\rho(M^{-1}N) < 1$.

Preuve. On pose $e^{(n)} = x - x^{(n)}$: $e^{(n+1)} = M^{-1}N e^{(n)}$ donc

$$e^{(n)} = (M^{-1}N)^n e^{(0)}.$$

Le cas où $M^{-1}N$ est diagonalisable est clair. □

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

► **But** : construire une suite $(x^{(n)})$ qui converge vers x t.q. $Ax = b$.

► **Idée** : si $A = M - N$ avec M inversible, alors

$$Ax = b \iff x = \underbrace{M^{-1}(Nx + b)}_{\varphi(x)}.$$

Problème de point fixe : soit $x^{(0)} \in \mathbb{R}^d$, on définit

$$x^{(n+1)} = \varphi(x^{(n)}).$$

Théorème. La suite $(x^{(n)})$ converge vers la solution de $Ax = b$ pour b quelconque ssi $\rho(M^{-1}N) < 1$.

Remarque. Si $\rho(M^{-1}N) < 1$ alors $\exists ||| \cdot |||$ t.q. $|||M^{-1}N||| < 1$.

Rappel. $\rho(M^{-1}N) \leq |||M^{-1}N|||$ pour toute norme subordonnée.

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

$$x^{(n+1)} = M^{-1}(Nx^{(n)} + b).$$

En pratique.

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

$$x^{(n+1)} = M^{-1}(Nx^{(n)} + b).$$

En pratique.

- Système linéaire $Mx^{(n+1)} = Nx^{(n)} + b$ à chaque itération.

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

$$x^{(n+1)} = M^{-1}(Nx^{(n)} + b).$$

En pratique.

- ▶ Système linéaire $Mx^{(n+1)} = Nx^{(n)} + b$ à chaque itération.
- ▶ M doit être « facile à inverser ».

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

$$x^{(n+1)} = M^{-1}(Nx^{(n)} + b).$$

En pratique.

- ▶ Système linéaire $Mx^{(n+1)} = Nx^{(n)} + b$ à chaque itération.
- ▶ M doit être « facile à inverser ».
 - ▶ Méthode de Jacobi
 $M =$ diagonale de A *si non nulle !*

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

$$x^{(n+1)} = M^{-1}(Nx^{(n)} + b).$$

En pratique.

- ▶ Système linéaire $Mx^{(n+1)} = Nx^{(n)} + b$ à chaque itération.
- ▶ M doit être « facile à inverser ».
 - ▶ Méthode de Jacobi
 $M =$ diagonale de A *si non nulle !*
 - ▶ Méthode de Gauss-Seidel
 $M =$ partie triangulaire inférieure de A *si diagonale non nulle !*

Analyse numérique matricielle

D'autres méthodes de résolution de systèmes linéaires

$$x^{(n+1)} = M^{-1}(Nx^{(n)} + b).$$

En pratique.

- ▶ Système linéaire $Mx^{(n+1)} = Nx^{(n)} + b$ à chaque itération.
- ▶ M doit être « facile à inverser ».
 - ▶ Méthode de Jacobi
 $M =$ diagonale de A *si non nulle !*
 - ▶ Méthode de Gauss-Seidel
 $M =$ partie triangulaire inférieure de A *si diagonale non nulle !*
- ▶ Si $\rho(M^{-1}N) < 1$, convergence en $\mathcal{O}(\rho(M^{-1}N)^n)$.

Analyse numérique matricielle

Recherche d'éléments propres

Problématique : Rechercher quelques vp de A , et les $\vec{vp}$ associés.

Problématique : Rechercher quelques vp de A , et les $\vec{vp}$ associés.

► **Constat.** si $A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ avec $\lambda_1 > \lambda_2 > 0$, et $x^0 = \begin{pmatrix} a \\ b \end{pmatrix}$,

Problématique : Rechercher quelques vp de A , et les $\vec{vp}$ associés.

► **Constat.** si $A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ avec $\lambda_1 > \lambda_2 > 0$, et $x^0 = \begin{pmatrix} a \\ b \end{pmatrix}$,

$$A^n x^0 \sim a \lambda_1^n \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{si } a \neq 0.$$

Problématique : Rechercher quelques vp de A , et les $\vec{vp}$ associés.

► **Constat.** si $A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ avec $\lambda_1 > \lambda_2 > 0$, et $x^0 = \begin{pmatrix} a \\ b \end{pmatrix}$,

$$A^n x^0 \sim a \lambda_1^n \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{si } a \neq 0.$$

Ainsi,

$$\frac{A^n x^0}{\|A^n x^0\|} \sim \mathbf{e}_1 \quad \text{avec} \quad A \mathbf{e}_1 = \lambda_1 \mathbf{e}_1 !$$

Problématique : Rechercher quelques vp de A , et les $\vec{vp}$ associés.

► **Constat.** si $A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ avec $\lambda_1 > \lambda_2 > 0$, et $x^0 = \begin{pmatrix} a \\ b \end{pmatrix}$,

$$A^n x^0 \sim a \lambda_1^n \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{si } a \neq 0.$$

Ainsi,

$$\frac{A^n x^0}{\|A^n x^0\|} \sim \mathbf{e}_1 \quad \text{avec} \quad A \mathbf{e}_1 = \lambda_1 \mathbf{e}_1 !$$

► **Algorithme.** Soit $x^0 \in \mathbb{R}^d$, on construit

$$\boxed{y^{n+1} = Ax^n \quad \text{et} \quad x^{n+1} = \frac{y^{n+1}}{\|y^{n+1}\|}}.$$

Analyse numérique matricielle

La méthode de la puissance

Algorithme. Soit $x^0 \in \mathbb{R}^d$, on construit

$$y^{n+1} = Ax^n \quad \text{et} \quad x^{n+1} = \frac{y^{n+1}}{\|y^{n+1}\|}.$$

Algorithme. Soit $x^0 \in \mathbb{R}^d$, on construit

$$y^{n+1} = Ax^n \quad \text{et} \quad x^{n+1} = \frac{y^{n+1}}{\|y^{n+1}\|}.$$

Théorème. Soit A de vp (complexes) $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_{d-1}| < |\lambda_d|$

Pour presque[†] tout $x^0 \in \mathbb{R}^n$, la suite $(Ax^n | x^n)$ converge vers λ_d . De plus, la suite

$$(\tilde{x}^n) := \left(\frac{|\lambda_d|}{\lambda_d} \right)^n x^n$$

converge vers un vecteur propre associé à λ_d .

Analyse numérique matricielle

La méthode de la puissance

Algorithme. Soit $x^0 \in \mathbb{R}^d$, on construit

$$y^{n+1} = Ax^n \quad \text{et} \quad x^{n+1} = \frac{y^{n+1}}{\|y^{n+1}\|}.$$

Théorème. Soit A de vp (complexes) $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_{d-1}| < |\lambda_d|$

Pour presque[†] tout $x^0 \in \mathbb{R}^n$, la suite $(Ax^n | x^n)$ converge vers λ_d . De plus, la suite

$$(\tilde{x}^n) := \left(\frac{|\lambda_d|}{\lambda_d} \right)^n x^n$$

converge vers un vecteur propre associé à λ_d .

[†] La condition est : la projection de x^0 sur le sous-espace propre associé à λ_1 n'est pas 0.

Attention à l'hypothèse $|\lambda_{d-1}| < |\lambda_d|$ qui est nécessaire.



[puissance1.m]

[puissance2.m]

COURS 2

Intégration numérique

Intégration numérique

Modélisation : calcul d'un temps de parcours



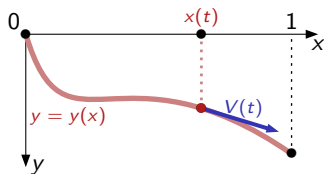
Deux toboggans... deux vitesses...



*Quel temps de descente
pour chaque toboggan ?*

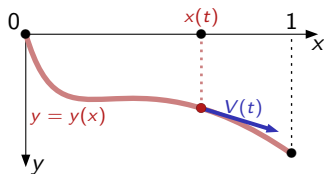
Intégration numérique

Modélisation : calcul d'un temps de parcours



Intégration numérique

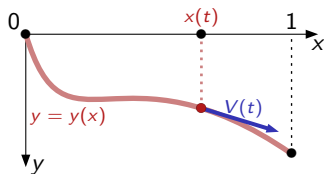
Modélisation : calcul d'un temps de parcours



► $x \mapsto y(x)$: forme du toboggan,

Intégration numérique

Modélisation : calcul d'un temps de parcours

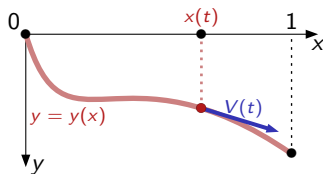


► $x \mapsto y(x)$: forme du toboggan,

► $t \mapsto (x(t), y(x(t)))$: position,

Intégration numérique

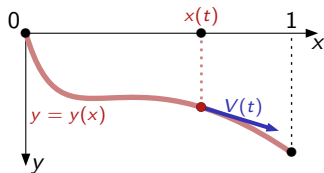
Modélisation : calcul d'un temps de parcours



- ▶ $x \mapsto y(x)$: forme du toboggan,
- ▶ $t \mapsto (x(t), y(x(t)))$: position,
- ▶ T : temps de descente,

Intégration numérique

Modélisation : calcul d'un temps de parcours



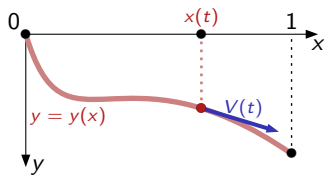
- ▶ $x \mapsto y(x)$: forme du toboggan,
- ▶ $t \mapsto (x(t), y(x(t)))$: position,
- ▶ T : temps de descente,

Vitesse : $V(t) = \begin{bmatrix} x'(t) \\ y'(x(t))x'(t) \end{bmatrix},$

$$|V(t)| = |x'(t)|\sqrt{1 + y'^2(x(t))}$$

Intégration numérique

Modélisation : calcul d'un temps de parcours



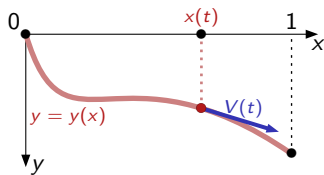
- ▶ $x \mapsto y(x)$: forme du toboggan,
- ▶ $t \mapsto (x(t), y(x(t)))$: position,
- ▶ T : temps de descente,

Vitesse : $V(t) = \begin{bmatrix} x'(t) \\ y'(x(t))x'(t) \end{bmatrix}, \quad |V(t)| = |x'(t)|\sqrt{1 + y'^2(x(t))}$

Énergies : $E_c(t) = \frac{1}{2} m x'(t)^2 [1 + y'(x(t))^2], \quad E_p(t) = -mgy(x(t)).$

Intégration numérique

Modélisation : calcul d'un temps de parcours



- ▶ $x \mapsto y(x)$: forme du toboggan,
- ▶ $t \mapsto (x(t), y(x(t)))$: position,
- ▶ T : temps de descente,

Vitesse : $V(t) = \begin{bmatrix} x'(t) \\ y'(x(t))x'(t) \end{bmatrix}$, $|V(t)| = |x'(t)|\sqrt{1 + y'^2(x(t))}$

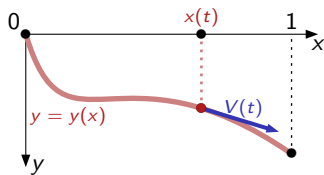
Énergies : $E_c(t) = \frac{1}{2} m x'(t)^2 [1 + y'(x(t))^2]$, $E_p(t) = -mgy(x(t))$.

Bilan entre $t = 0$ et $t = T$: $\frac{1}{2} m (x'(t))^2 [1 + y'(x(t))^2] - mgy(x(t)) = 0$.

$$\Rightarrow x'(t) = \sqrt{2gy(x(t)) / \sqrt{1 + y'(x(t))^2}}.$$

Intégration numérique

Modélisation : calcul d'un temps de parcours



- ▶ $x \mapsto y(x)$: forme du toboggan,
- ▶ $t \mapsto (x(t), y(x(t)))$: position,
- ▶ T : temps de descente,

Vitesse : $V(t) = \begin{bmatrix} x'(t) \\ y'(x(t))x'(t) \end{bmatrix}$, $|V(t)| = |x'(t)|\sqrt{1 + y'^2(x(t))}$

Énergies : $E_c(t) = \frac{1}{2} m x'(t)^2 [1 + y'(x(t))^2]$, $E_p(t) = -mgy(x(t))$.

Bilan entre $t = 0$ et $t = T$: $\frac{1}{2} m (x'(t))^2 [1 + y'(x(t))^2] - mgy(x(t)) = 0$.

$$\Rightarrow x'(t) = \sqrt{2gy(x(t))} / \sqrt{1 + y'(x(t))^2}.$$

Temps de descente : $T = \int_0^T dt = \int_0^1 \frac{\sqrt{1 + y'(x)^2}}{\sqrt{2gy(x)}} dx$.

Problématique. La forme $x \mapsto y(x)$ du toboggan étant donnée, comment calculer le temps de descente T ?

$$T = \int_0^1 \frac{\sqrt{1 + y'(x)^2}}{\sqrt{2gy(x)}} dx.$$

Problématique. La forme $x \mapsto y(x)$ du toboggan étant donnée, comment calculer le temps de descente T ?

$$T = \int_0^1 \frac{\sqrt{1 + y'(x)^2}}{\sqrt{2gy(x)}} dx.$$

► **Problème général** : calcul de $\int_a^b f(x) dx$.

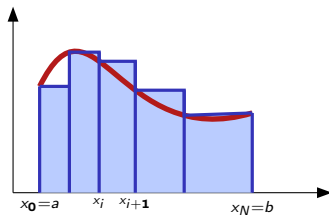
Problématique. La forme $x \mapsto y(x)$ du toboggan étant donnée, comment calculer le temps de descente T ?

$$T = \int_0^1 \frac{\sqrt{1 + y'(x)^2}}{\sqrt{2gy(x)}} dx.$$

- ▶ **Problème général** : calcul de $\int_a^b f(x) dx$.
- ▶ **Enjeux** :
 - ▶ Mise en place d'une méthode numérique,
 - ▶ Étude de la précision,
 - ▶ Analyse du temps de calcul,
 - ▶ Quelle méthode pour quelle fonction f ?

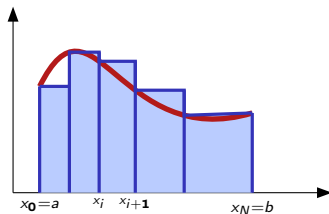
Intégration numérique

Une première idée : la méthode des rectangles



Intégration numérique

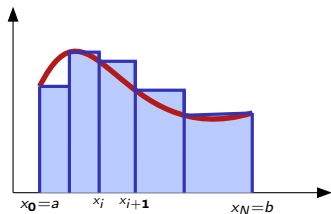
Une première idée : la méthode des rectangles



$$\int_a^b f(x) dx \simeq \sum_{i=0}^{N-1} (x_{i+1} - x_i) f(x_i).$$

Intégration numérique

Une première idée : la méthode des rectangles



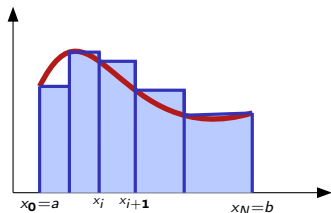
$$\int_a^b f(x) dx \simeq \sum_{i=0}^{N-1} (x_{i+1} - x_i) f(x_i).$$

► Cas équidistant : $x_{i+1} - x_i = h = \frac{b-a}{N}$,

$$\int_a^b f(x) dx \simeq h \sum_{i=0}^{N-1} f(x_i).$$

Intégration numérique

Une première idée : la méthode des rectangles



$$\int_a^b f(x) dx \simeq \sum_{i=0}^{N-1} (x_{i+1} - x_i) f(x_i).$$

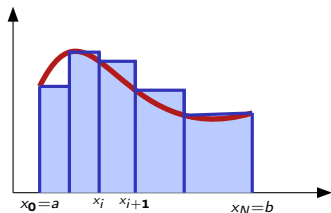
- Cas équidistant : $x_{i+1} - x_i = h = \frac{b-a}{N}$,

$$\int_a^b f(x) dx \simeq h \sum_{i=0}^{N-1} f(x_i).$$

- Est-ce que l'approximation converge lorsque $N \rightarrow +\infty$?

Intégration numérique

Une première idée : la méthode des rectangles



$$\int_a^b f(x) dx \simeq \sum_{i=0}^{N-1} (x_{i+1} - x_i) f(x_i).$$

- Cas équidistant : $x_{i+1} - x_i = h = \frac{b-a}{N}$,

$$\int_a^b f(x) dx \simeq h \sum_{i=0}^{N-1} f(x_i).$$

- Est-ce que l'approximation converge lorsque $N \rightarrow +\infty$?
- Si oui, avec quelle précision ?

Intégration numérique

Convergence de la méthode des rectangles

- Cas équidistant (rappel : $Nh = b - a$)

$$I_N(f) = h \sum_{i=0}^{N-1} f(x_i).$$

Intégration numérique

Convergence de la méthode des rectangles

- Cas équidistant (rappel : $Nh = b - a$)

$$I_N(f) = h \sum_{i=0}^{N-1} f(x_i).$$

- Si f est continue, $I_N(f)$ est une *somme de Riemann*.

$$\implies I_N(f) \xrightarrow[N \rightarrow +\infty]{} \int_a^b f(x) \, dx.$$

Intégration numérique

Convergence de la méthode des rectangles

- Cas équidistant (rappel : $Nh = b - a$)

$$I_N(f) = h \sum_{i=0}^{N-1} f(x_i).$$

- Si f est continue, $I_N(f)$ est une *somme de Riemann*.

$$\implies I_N(f) \xrightarrow[N \rightarrow +\infty]{} \int_a^b f(x) dx.$$

C'est lié à l'approximation uniforme des fonctions continues par des fonctions en escalier / à la continuité uniforme des fonctions continues sur un segment.

Intégration numérique

Convergence de la méthode des rectangles

- Cas équidistant (rappel : $Nh = b - a$)

$$I_N(f) = h \sum_{i=0}^{N-1} f(x_i).$$

- Si f est continue, $I_N(f)$ est une *somme de Riemann*.

$$\implies I_N(f) \xrightarrow[N \rightarrow +\infty]{} \int_a^b f(x) dx.$$

C'est lié à l'approximation uniforme des fonctions continues par des fonctions en escalier / à la continuité uniforme des fonctions continues sur un segment.

- Peut-on quantifier la précision de l'approximation ?

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) \, dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

► Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

- Estimation d'erreur

$$|I - I_N| \leq \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} |f(x) - f(x_i)| dx$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

- Estimation d'erreur

$$|I - I_N| \leq M_1 \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} |x - x_i| dx$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

- Estimation d'erreur

$$|I - I_N| \leq M_1 \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} (x - x_i) dx$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

- Estimation d'erreur

$$|I - I_N| \leq M_1 \sum_{i=0}^{N-1} \left[\frac{(x - x_i)^2}{2} \right]_{x_i}^{x_{i+1}} dx$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

- Estimation d'erreur

$$|I - I_N| \leq M_1 \sum_{i=0}^{N-1} \frac{h^2}{2}$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

- Estimation d'erreur

$$|I - I_N| \leq M_1 N \frac{h^2}{2}$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

- Estimation d'erreur

$$|I - I_N| \leq \frac{M_1(b-a)}{2} h$$

Intégration numérique

Précision de la méthode des rectangles

$$I = \int_a^b f(x) dx, \quad I_N = h \sum_{i=0}^{N-1} f(a + ih).$$

- Erreur :

$$|I - I_N| = \left| \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} [f(x) - f(x_i)] dx \right|.$$

- Inégalité des accroissements finis : si $f \in \mathcal{C}^1([a, b])$,

$$|f(x) - f(x_i)| \leq M_1 |x - x_i| \quad \text{avec} \quad M_1 = \sup_{t \in [a, b]} |f'(t)|.$$

- Estimation d'erreur

$$|I - I_N| \leq \frac{M_1(b-a)^2}{2N}$$

Intégration numérique

Précision de la méthode des rectangles

Si $f \in \mathcal{C}^1([a, b])$, alors, pour la méthode des rectangles,

$$|I - I_N| \leq \frac{(b-a)^2}{2N} \sup_{t \in [a, b]} |f'(t)|$$

Intégration numérique

Précision de la méthode des rectangles

Si $f \in \mathcal{C}^1([a, b])$, alors, pour la méthode des rectangles,

$$|I - I_N| \leq \frac{(b-a)^2}{2N} \sup_{t \in [a, b]} |f'(t)|$$

À retenir :

- ▶ Hypothèse de régularité : $f \in \mathcal{C}^1([a, b])$.
- ▶ Convergence en $\mathcal{O}\left(\frac{1}{N}\right)$.

Intégration numérique

Précision de la méthode des rectangles

Si $f \in \mathcal{C}^1([a, b])$, alors, pour la méthode des rectangles,

$$|I - I_N| \leq \frac{(b-a)^2}{2N} \sup_{t \in [a, b]} |f'(t)|$$

À retenir :

- ▶ Hypothèse de régularité : $f \in \mathcal{C}^1([a, b])$.
- ▶ Convergence en $\mathcal{O}\left(\frac{1}{N}\right)$.

Question : que se passe-t-il si $f \notin \mathcal{C}^1([a, b])$?

- ▶ Simulation pour diverses fonctions.

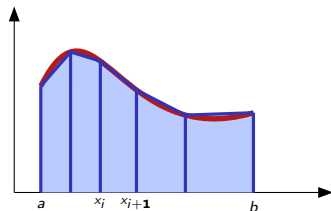


- ▶ Analyse théorique : cf. TD.

Intégration numérique

Une autre méthode

► Méthode des trapèzes

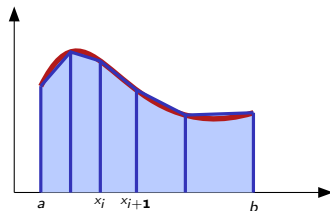


$$\int_a^b f(x) dx \simeq \sum_{i=0}^{N-1} (x_{i+1} - x_i) \frac{f(x_i) + f(x_{i+1})}{2}.$$

Intégration numérique

Une autre méthode

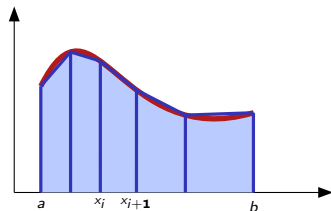
► Méthode des trapèzes



$$\int_a^b f(x) dx \simeq \sum_{i=0}^{N-1} (x_{i+1} - x_i) \frac{f(x_i) + f(x_{i+1})}{2}.$$

► Erreur observée dans le cas général : $\mathcal{O}(N^{-2})$.

► Méthode des trapèzes



$$\int_a^b f(x) dx \simeq \sum_{i=0}^{N-1} (x_{i+1} - x_i) \frac{f(x_i) + f(x_{i+1})}{2}.$$

► Erreur observée dans le cas général : $\mathcal{O}(N^{-2})$.

► Estimation d'erreur théorique ?

↪ *estimation sur $[x_i, x_{i+1}]$ de la différence courbe/corde.*

Intégration numérique

Méthode des trapèzes

- ▶ Estimation d'erreur (subdivision uniforme de pas h)

- **Estimation d'erreur** (subdivision uniforme de pas h)

$$|I - I_N| \leq \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} |f(x) - p(x)| dx,$$

où $p \in \mathbb{P}_1$ satisfait $p(x_i) = f(x_i)$ et $p(x_{i+1}) = f(x_{i+1})$.

- **Estimation d'erreur** (subdivision uniforme de pas h)

$$|I - I_N| \leq \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} |f(x) - p(x)| \, dx,$$

où $p \in \mathbb{P}_1$ satisfait $p(x_i) = f(x_i)$ et $p(x_{i+1}) = f(x_{i+1})$.

- On pose $\psi = f - p$. Th. Rolle : $\exists \xi_i \in [x_i, x_{i+1}]$, $\psi'(\xi_i) = 0$.

$$\forall x \in [x_i, x_{i+1}], \quad |\psi'(x)| \leq |x - \xi_i| \sup_{[x_i, x_{i+1}]} |\psi''| \leq h \sup_{[a, b]} |f''|.$$

- **Estimation d'erreur** (subdivision uniforme de pas h)

$$|I - I_N| \leq \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} |f(x) - p(x)| dx,$$

où $p \in \mathbb{P}_1$ satisfait $p(x_i) = f(x_i)$ et $p(x_{i+1}) = f(x_{i+1})$.

- On pose $\psi = f - p$. Th. Rolle : $\exists \xi_i \in [x_i, x_{i+1}]$, $\psi'(\xi_i) = 0$.

$$\forall x \in [x_i, x_{i+1}], |\psi'(x)| \leq |x - \xi_i| \sup_{[x_i, x_{i+1}]} |\psi''| \leq h \sup_{[a, b]} |f''|.$$

- Donc $\forall x \in [x_i, x_{i+1}]$,

$$|\psi(x)| \leq \int_{x_i}^x |\psi'(x)| \leq h^2 \sup_{[a, b]} |f''|.$$

- **Estimation d'erreur** (subdivision uniforme de pas h)

$$|I - I_N| \leq \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} |f(x) - p(x)| dx,$$

où $p \in \mathbb{P}_1$ satisfait $p(x_i) = f(x_i)$ et $p(x_{i+1}) = f(x_{i+1})$.

- On pose $\psi = f - p$. Th. Rolle : $\exists \xi_i \in [x_i, x_{i+1}]$, $\psi'(\xi_i) = 0$.

$$\forall x \in [x_i, x_{i+1}], |\psi'(x)| \leq |x - \xi_i| \sup_{[x_i, x_{i+1}]} |\psi''| \leq h \sup_{[a, b]} |f''|.$$

- Donc $\forall x \in [x_i, x_{i+1}]$,

$$|\psi(x)| \leq \int_{x_i}^x |\psi'(x)| \leq h^2 \sup_{[a, b]} |f''|.$$

- Conclusion : $|I - I_N| \leq h^2(b-a) \sup_{[a, b]} |f''| = \frac{(b-a)^3}{N^2} M_2.$

Intégration numérique

D'autres méthodes

- **Idée** : remplacer φ par un polynôme π_φ et

$$\int_0^1 \varphi(t) dt \simeq \int_0^1 \pi_\varphi(t) dt.$$

Intégration numérique

D'autres méthodes

- **Idée** : remplacer φ par un polynôme π_φ et

$$\int_0^1 \varphi(t) dt \simeq \int_0^1 \pi_\varphi(t) dt.$$

- **Rappel** : pour $0 \leq t_0 < t_1 < \dots < t_k \leq 1$ et $b_0, b_1, \dots, b_k \in \mathbb{R}$ fixés,

$$\exists ! p \in \mathbb{P}_k, \quad \forall q = 0, 1, \dots, k, \quad p(t_q) = b_q.$$

Dans la base de Lagrange,

$$p(t) = \sum_{q=0}^k b_q L_q(t), \quad \text{avec} \quad L_q(t) = \prod_{\ell \neq q} \frac{t - t_\ell}{t_q - t_\ell}.$$

Intégration numérique

D'autres méthodes

- **Idée** : remplacer φ par un polynôme π_φ et

$$\int_0^1 \varphi(t) dt \simeq \int_0^1 \pi_\varphi(t) dt.$$

- **Rappel** : pour $0 \leq t_0 < t_1 < \dots < t_k \leq 1$ et $b_0, b_1, \dots, b_k \in \mathbb{R}$ fixés,

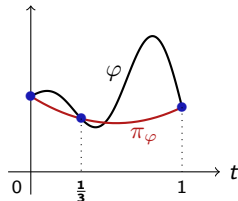
$$\exists ! p \in \mathbb{P}_k, \quad \forall q = 0, 1, \dots, k, \quad p(t_q) = b_q.$$

Dans la base de Lagrange,

$$p(t) = \sum_{q=0}^k b_q L_q(t), \quad \text{avec} \quad L_q(t) = \prod_{\ell \neq q} \frac{t - t_\ell}{t_q - t_\ell}.$$

- **Exemple** :

π_φ = polynôme d'interpolation
de Lagrange de φ
en $0, \frac{1}{3}, 1$.



Intégration numérique

Intermède : interpolation de Lagrange

$$\int_0^1 \varphi(t) dt \simeq \int_0^1 \pi_\varphi(t) dt.$$

Intégration numérique

Intermède : interpolation de Lagrange

$$\int_0^1 \varphi(t) dt \simeq \int_0^1 \pi_\varphi(t) dt.$$

- Il n'est pas nécessaire d'expliciter π_φ pour approcher l'intégrale :

$$\pi_\varphi(t) = \sum_{q=0}^k \varphi(t_q) L_q(t) \quad \Rightarrow \quad \int_0^1 \pi_\varphi(t) dt = \sum_{q=0}^k \underbrace{\left(\int_0^1 L_q(t) dt \right)}_{=w_q} \varphi(t_q)$$

$$\int_0^1 \varphi(t) dt \simeq \int_0^1 \pi_\varphi(t) dt.$$

- Il n'est pas nécessaire d'expliciter π_φ pour approcher l'intégrale :

$$\pi_\varphi(t) = \sum_{q=0}^k \varphi(t_q) L_q(t) \quad \Rightarrow \quad \int_0^1 \pi_\varphi(t) dt = \sum_{q=0}^k \underbrace{\left(\int_0^1 L_q(t) dt \right)}_{=w_q} \varphi(t_q)$$

d'où l'approximation

$$\int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

Intégration numérique

Intermède : interpolation de Lagrange

- ▶ Que se passe-t-il lorsque $k \rightarrow \infty$? (*points (t_q) équidistants*)

Intégration numérique

Intermède : interpolation de Lagrange

- ▶ Que se passe-t-il lorsque $k \rightarrow \infty$? (*points (t_q) équidistants*)

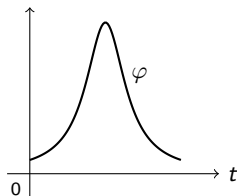
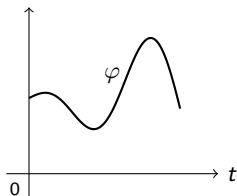
$$\varphi(t) = 1 + \frac{t}{2} \cos(8t)$$

$$\varphi(t) = \frac{2}{1 + 10(2t - 1)^2}$$

Intégration numérique

Intermède : interpolation de Lagrange

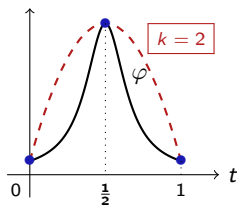
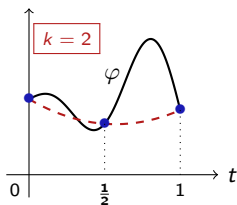
- Que se passe-t-il lorsque $k \rightarrow \infty$? (*points (t_q) équidistants*)



Intégration numérique

Intermède : interpolation de Lagrange

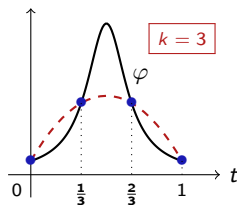
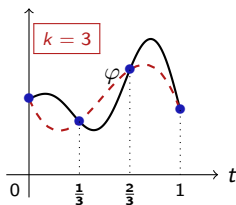
- Que se passe-t-il lorsque $k \rightarrow \infty$? (*points (t_q) équidistants*)



Intégration numérique

Intermède : interpolation de Lagrange

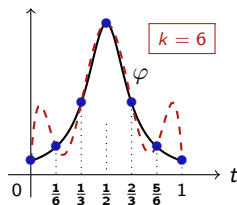
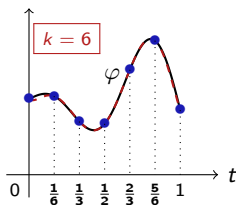
- Que se passe-t-il lorsque $k \rightarrow \infty$? (*points (t_q) équidistants*)



Intégration numérique

Intermède : interpolation de Lagrange

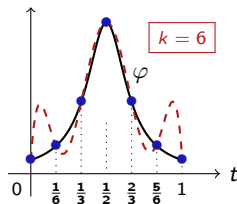
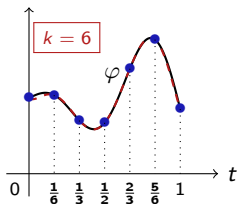
- Que se passe-t-il lorsque $k \rightarrow \infty$? (*points (t_q) équidistants*)



Intégration numérique

Intermède : interpolation de Lagrange

- Que se passe-t-il lorsque $k \rightarrow \infty$? (points (t_q) équidistants)

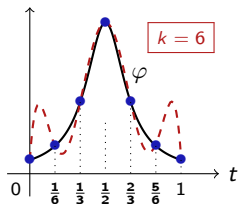
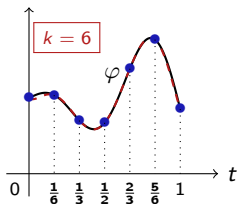


Non convergence lorsque $k \rightarrow +\infty$
(phénomène de Runge)

Intégration numérique

Intermède : interpolation de Lagrange

- Que se passe-t-il lorsque $k \rightarrow \infty$? (*points (t_q) équidistants*)



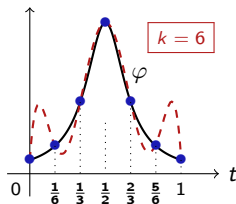
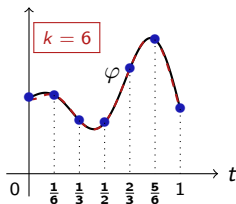
Non convergence lorsque $k \rightarrow +\infty$
(*phénomène de Runge*)

- Pas d'espoir d'obtenir une approximation qui converge lorsque $k \rightarrow +\infty$.

Intégration numérique

Intermède : interpolation de Lagrange

- Que se passe-t-il lorsque $k \rightarrow \infty$? (*points (t_q) équidistants*)



Non convergence lorsque $k \rightarrow +\infty$
(phénomène de Runge)

- Pas d'espoir d'obtenir une approximation qui converge lorsque $k \rightarrow +\infty$.
- On va conserver k fixé et travailler sur une subdivision de l'intervalle.

Intégration numérique

Principe des méthodes composées

$$\int_a^b f(x) dx$$

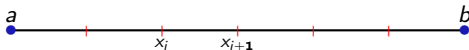


Intégration numérique

Principe des méthodes composées

$$\int_a^b f(x) dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx$$

- On subdivise l'intervalle $[a, b]$ (pas uniforme h).

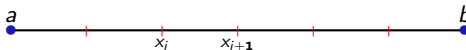


Intégration numérique

Principe des méthodes composées

$$\int_a^b f(x) dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx$$

- ▶ On subdivise l'intervalle $[a, b]$ (pas uniforme h).
- ▶ On choisit un modèle élémentaire sur $[0, 1]$

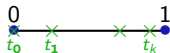
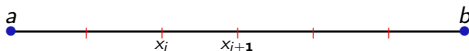


Intégration numérique

Principe des méthodes composées

$$\int_a^b f(x) dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx$$

- ▶ On subdivise l'intervalle $[a, b]$ (pas uniforme h).
- ▶ On choisit un modèle élémentaire sur $[0, 1]$: $\int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q)$

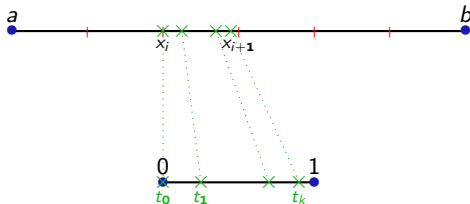


Intégration numérique

Principe des méthodes composées

$$\int_a^b f(x) dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx = \sum_{i=0}^{N-1} h \int_0^1 f(x_i + th) dt \simeq h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

- ▶ On subdivise l'intervalle $[a, b]$ (pas uniforme h).
- ▶ On choisit un modèle élémentaire sur $[0, 1]$: $\int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q)$
- ▶ On transporte sur chaque sous-intervalle.



Intégration numérique

Méthodes composées

Exemple :

► $(k = 1). t_0 = 0, t_1 = 1,$

Intégration numérique

Méthodes composées

Exemple :

► $(k = 1). t_0 = 0, t_1 = 1,$

► $w_0 = \frac{1}{2}, w_1 = \frac{1}{2}.$

Intégration numérique

Méthodes composées

Exemple :

► $(k = 1). t_0 = 0, t_1 = 1,$

► $w_0 = \frac{1}{2}, w_1 = \frac{1}{2}.$

$$\int_a^b f(x) dx \simeq h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

$$= h \sum_{i=0}^{N-1} \frac{f(x_i) + f(x_{i+1})}{2}$$

Intégration numérique

Méthodes composées

Exemple :

► $(k = 1). t_0 = 0, t_1 = 1,$

► $w_0 = \frac{1}{2}, w_1 = \frac{1}{2}.$

$$\int_a^b f(x) dx \simeq h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

$$= h \sum_{i=0}^{N-1} \frac{f(x_i) + f(x_{i+1})}{2}$$

⇒ Méthode des trapèzes !

► Bilan

- Modèle élémentaire sur $[0, 1]$:

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

► Bilan

- Modèle élémentaire sur $[0, 1]$:

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- Méthode composée sur $[a, b]$:

$$(FQ_c) \quad I = \int_a^b f(x) dx \simeq I_N = h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

► Bilan

- Modèle élémentaire sur $[0, 1]$:

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- Méthode composée sur $[a, b]$:

$$(FQ_c) \quad I = \int_a^b f(x) dx \simeq I_N = h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

- *A priori* : $N(k+1)$ évaluations de f .

► Bilan

- Modèle élémentaire sur $[0, 1]$:

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- Méthode composée sur $[a, b]$:

$$(FQ_c) \quad I = \int_a^b f(x) dx \simeq I_N = h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

- *A priori* : $N(k+1)$ évaluations de f .
- Quelle précision pour (FQ_c) ?

► Une remarque

$$I - I_N = h \sum_{i=0}^{N-1} \left(\int_0^1 \underbrace{f(x_i + th)}_{\varphi_i(t)} dt - \int_0^1 \pi_{\varphi_i}(t) dt \right),$$

- $\varphi_i : t \mapsto f(x_i + th)$,
- $\forall \varphi, \pi_\varphi \in \mathbb{P}_k$ satisfait $\pi_\varphi(t_q) = \varphi(t_q)$ ($0 \leq q \leq k$).

► Une remarque

$$I - I_N = h \sum_{i=0}^{N-1} \left(\int_0^1 \underbrace{f(x_i + th)}_{\varphi_i(t)} dt - \int_0^1 \pi_{\varphi_i}(t) dt \right),$$

► $\varphi_i : t \mapsto f(x_i + th),$

► $\forall \varphi, \pi_\varphi \in \mathbb{P}_k$ satisfait $\pi_\varphi(t_q) = \varphi(t_q) \ (0 \leq q \leq k).$

Théorème. pour $\varphi \in \mathcal{C}^{k+1}([0, 1]),$

$$|\varphi(t) - \pi_\varphi(t)| \leq \frac{\sup_{t \in [0,1]} |\varphi^{(k+1)}(t)|}{(k+1)!}.$$

► Une remarque

$$I - I_N = h \sum_{i=0}^{N-1} \left(\int_0^1 \underbrace{f(x_i + th)}_{\varphi_i(t)} dt - \int_0^1 \pi_{\varphi_i}(t) dt \right),$$

► $\varphi_i : t \mapsto f(x_i + th), \quad \boxed{\varphi_i^{(k+1)}(t) = h^{k+1} f^{(k+1)}(x_i + th)}.$

► $\forall \varphi, \pi_\varphi \in \mathbb{P}_k$ satisfait $\pi_\varphi(t_q) = \varphi(t_q) \quad (0 \leq q \leq k).$

Théorème. pour $\varphi \in \mathcal{C}^{k+1}([0, 1]),$

$$|\varphi(t) - \pi_\varphi(t)| \leq \frac{\sup_{t \in [0,1]} |\varphi^{(k+1)}(t)|}{(k+1)!}.$$

► **Conséquence :** si $f \in \mathcal{C}^{k+1}[a, b],$

$$\boxed{|I - I_N| \leq \frac{h^{k+1}(b-a)}{(k+1)!} \sup_{x \in [a,b]} |f^{(k+1)}(x)|.}$$

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

$$(FQ_c) \quad I = \int_a^b f(x) dx \simeq I_N = h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

Théorème. Si (M_e) est exacte pour tout $\varphi \in \mathbb{P}_\ell$ et si $f \in \mathcal{C}^{\ell+1}([a, b])$, alors

$$|I - I_N| \leq \frac{h^{\ell+1}(b-a)}{(\ell+1)!} \sup_{x \in [a, b]} |f^{(\ell+1)}(x)|.$$

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

$$(FQ_c) \quad I = \int_a^b f(x) dx \simeq I_N = h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

Théorème. Si (M_e) est exacte pour tout $\varphi \in \mathbb{P}_\ell$ et si $f \in \mathcal{C}^{\ell+1}([a, b])$, alors

$$|I - I_N| \leq \frac{h^{\ell+1}(b-a)}{(\ell+1)!} \sup_{x \in [a, b]} |f^{(\ell+1)}(x)|.$$

Remarque. La constante n'est pas optimale...

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

$$(FQ_c) \quad I = \int_a^b f(x) dx \simeq I_N = h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

Théorème. Si (M_e) est exacte pour tout $\varphi \in \mathbb{P}_\ell$ et si $f \in \mathcal{C}^{\ell+1}([a, b])$, alors

$$|I - I_N| \leq \frac{h^{\ell+1}(b-a)}{(\ell+1)!} \sup_{x \in [a, b]} |f^{(\ell+1)}(x)|.$$

Remarque. La constante n'est pas optimale...

Question. A priori, $\ell = k$ $N(k+1)$ évaluations de f , précision en $\mathcal{O}\left(\frac{1}{N^{k+1}}\right)$.

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

$$(FQ_c) \quad I = \int_a^b f(x) dx \simeq I_N = h \sum_{i=0}^{N-1} \sum_{q=0}^k w_q f(x_i + t_q h).$$

Théorème. Si (M_e) est exacte pour tout $\varphi \in \mathbb{P}_\ell$ et si $f \in \mathcal{C}^{\ell+1}([a, b])$, alors

$$|I - I_N| \leq \frac{h^{\ell+1}(b-a)}{(\ell+1)!} \sup_{x \in [a, b]} |f^{(\ell+1)}(x)|.$$

Remarque. La constante n'est pas optimale...

Question. A priori, $\ell = k$ $N(k+1)$ évaluations de f , précision en $\mathcal{O}\left(\frac{1}{N^{k+1}}\right)$.

Peut-on faire mieux ?

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

- Points de quadrature fixés : comment calculer les poids ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

- Points de quadrature fixés : comment calculer les poids ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- k et $(t_q)_{q=0,\dots,k}$ donnés.

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

- Points de quadrature fixés : comment calculer les poids ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- k et $(t_q)_{q=0,\dots,k}$ donnés.
- (M_e) exact sur $\mathbb{P}_k \iff (M_e)$ exact pour la base $(1, t, \dots, t^k)$,

$$\iff \forall i = 0, 1, \dots, k, \quad \sum_{q=0}^k w_q t_q^i = \frac{1}{i+1}.$$

(système linéaire de Vandermonde)

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

- Points de quadrature fixés : comment calculer les poids ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- k et $(t_q)_{q=0,\dots,k}$ donnés.
- (M_e) exact sur $\mathbb{P}_k \iff (M_e)$ exact pour la base $(1, t, \dots, t^k)$,

$$\iff \forall i = 0, 1, \dots, k, \quad \sum_{q=0}^k w_q t_q^i = \frac{1}{i+1}.$$

(système linéaire de Vandermonde)

$\implies$ les poids (w_q) sont uniquement déterminés !

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

- Points de quadrature fixés : comment calculer les poids ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- k et $(t_q)_{q=0,\dots,k}$ donnés.
- (M_e) exact sur $\mathbb{P}_k \iff (M_e)$ exact pour la base $(1, t, \dots, t^k)$,

$$\iff \forall i = 0, 1, \dots, k, \quad \sum_{q=0}^k w_q t_q^i = \frac{1}{i+1}.$$

(système linéaire de Vandermonde)

$\implies$ les poids (w_q) sont uniquement déterminés !

■ **Conclusion.** Sauf « miracle », (M_e) est exact sur $\mathbb{P}_k$ seulement...

Intégration numérique

Optimalité des choix des points/poids de quadrature ?



Avec $k + 1$ points de quadrature $\ddagger(\mathbb{R})$: peut-on être exact au delà de $\mathbb{P}_k$?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

Intégration numérique

Optimalité des choix des points/poids de quadrature ?



Avec $k + 1$ points de quadrature $\ddagger(\mathbb{R})$: peut-on être exact au delà de $\mathbb{P}_k$?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

► **Remarque.** pour $p(t) = \prod_{q=0}^k (t - t_q)^2 \in \mathbb{P}_{2k+2}$,

$$\int_0^1 p(t) dt > 0, \quad \sum_{q=0}^k w_q p(t_q) = 0.$$

Intégration numérique

Optimalité des choix des points/poids de quadrature ?



Avec $k + 1$ points de quadrature $\ddagger(\mathbb{R})$: peut-on être exact au delà de $\mathbb{P}_k$?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

► **Remarque.** pour $p(t) = \prod_{q=0}^k (t - t_q)^2 \in \mathbb{P}_{2k+2}$,

$$\int_0^1 p(t) dt > 0, \quad \sum_{q=0}^k w_q p(t_q) = 0.$$

$\implies (M_e)$ ne peut pas être exact $\mathbb{P}_{2k+2}$!

Intégration numérique

Optimalité des choix des points/poids de quadrature ?



Avec $k + 1$ points de quadrature $\ddagger(\mathbb{R})$: peut-on être exact au delà de $\mathbb{P}_k$?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

► **Remarque.** pour $p(t) = \prod_{q=0}^k (t - t_q)^2 \in \mathbb{P}_{2k+2}$,

$$\int_0^1 p(t) dt > 0, \quad \sum_{q=0}^k w_q p(t_q) = 0.$$

$\implies (M_e)$ ne peut pas être exact $\mathbb{P}_{2k+2}$!

► Peut-on espérer être exact $\mathbb{P}_{2k+1}$?



Intégration numérique

Optimalité des choix des points/poids de quadrature ?

$$(M_e) \quad \int_0^1 \varphi(t) \, dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

► (M_e) exact sur $\mathbb{P}_{2k+1}$?

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

► (M_e) exact sur $\mathbb{P}_{2k+1}$? Soit $p \in \mathbb{P}_{2k+1}$ et $\pi(t) = \prod_{q=0}^k (t - t_q) \in \mathbb{P}_{k+1}$.

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- ▶ (M_e) exact sur $\mathbb{P}_{2k+1}$? Soit $p \in \mathbb{P}_{2k+1}$ et $\pi(t) = \prod_{q=0}^k (t - t_q) \in \mathbb{P}_{k+1}$.
- ▶ Division euclidienne : $p = \pi u + r$, avec $d^\circ u \leq k$ et $d^\circ r \leq k$.

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- ▶ (M_e) exact sur $\mathbb{P}_{2k+1}$? Soit $p \in \mathbb{P}_{2k+1}$ et $\pi(t) = \prod_{q=0}^k (t - t_q) \in \mathbb{P}_{k+1}$.
- ▶ Division euclidienne : $p = \pi u + r$, avec $d^\circ u \leq k$ et $d^\circ r \leq k$.

$$\sum_{q=0}^k w_q p(t_q) = \sum_{q=0}^k w_q r(t_q). \quad \text{car } \pi(t_q) = 0.$$

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

► (M_e) exact sur $\mathbb{P}_{2k+1}$? Soit $p \in \mathbb{P}_{2k+1}$ et $\pi(t) = \prod_{q=0}^k (t - t_q) \in \mathbb{P}_{k+1}$.

► Division euclidienne : $p = \pi u + r$, avec $d^\circ u \leq k$ et $d^\circ r \leq k$.

$$\sum_{q=0}^k w_q p(t_q) = \sum_{q=0}^k w_q r(t_q). \quad \text{car } \pi(t_q) = 0.$$

$$\int_0^1 p(t) dt = \int_0^1 \pi(t) u(t) dt + \int_0^1 r(t) dt.$$

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- ▶ (M_e) exact sur $\mathbb{P}_{2k+1}$? Soit $p \in \mathbb{P}_{2k+1}$ et $\pi(t) = \prod_{q=0}^k (t - t_q) \in \mathbb{P}_{k+1}$.
- ▶ Division euclidienne : $p = \pi u + r$, avec $\deg u \leq k$ et $\deg r \leq k$.

$$\sum_{q=0}^k w_q p(t_q) = \sum_{q=0}^k w_q r(t_q). \quad \text{car } \pi(t_q) = 0.$$

$$\int_0^1 p(t) dt = \int_0^1 \pi(t) u(t) dt + \int_0^1 r(t) dt.$$

- ▶ Si (M_e) est exact sur $\mathbb{P}_k$ et $\pi \perp \mathbb{P}_k$, alors (M_e) est exact sur $\mathbb{P}_{2k+1}$!

Intégration numérique

Optimalité des choix des points/poids de quadrature ?

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

- ▶ (M_e) exact sur $\mathbb{P}_{2k+1}$? Soit $p \in \mathbb{P}_{2k+1}$ et $\pi(t) = \prod_{q=0}^k (t - t_q) \in \mathbb{P}_{k+1}$.
- ▶ Division euclidienne : $p = \pi u + r$, avec $d^\circ u \leq k$ et $d^\circ r \leq k$.

$$\sum_{q=0}^k w_q p(t_q) = \sum_{q=0}^k w_q r(t_q). \quad \text{car } \pi(t_q) = 0.$$

$$\int_0^1 p(t) dt = \int_0^1 \pi(t) u(t) dt + \int_0^1 r(t) dt.$$

- ▶ Si (M_e) est exact sur $\mathbb{P}_k$ et $\pi \perp \mathbb{P}_k$, alors (M_e) est exact sur $\mathbb{P}_{2k+1}$!

$\Rightarrow$ $(t_q) = \text{racines du } (k+1)^{\text{e}} \text{ polynôme orthogonal pour } L^2(0,1).$

Théorème. Soit $(P_0, P_1, \dots)$ famille de polynômes orthogonaux (avec $d^\circ P_i = i$) pour le produit scalaire

$$(\varphi, \psi) = \int_0^1 \varphi(t)\psi(t) dt.$$

On note $(t_q)_{q=0, \dots, k}$ les racines de P_{k+1} . On détermine les poids $(w_q)_{q=0, \dots, k}$ t.q. (M_e) soit exact $\mathbb{P}_k$, i.e. solution de

$$\forall i = 0, 1, \dots, k, \quad \sum_{q=0}^k w_q t_q^i = \frac{1}{i+1}.$$

Alors la formule

$$(M_e) \quad \int_0^1 \varphi(t) dt \simeq \sum_{q=0}^k w_q \varphi(t_q).$$

est exacte $\mathbb{P}_{2k+1}$. (Méthode de Gauss-Legendre).

- **Exemple.** Méthode de Gauss-Legendre exacte $\mathbb{P}_5$.

$$\int_0^1 \varphi(t) dt \simeq \frac{5}{18} \varphi\left(\frac{1}{2} - \frac{\sqrt{15}}{10}\right) + \frac{4}{9} \varphi\left(\frac{1}{2}\right) + \frac{5}{18} \varphi\left(\frac{1}{2} + \frac{\sqrt{15}}{10}\right).$$

La méthode composée correspondante est en $\mathcal{O}(N^{-6})$.

Pour k grand, le calcul des points n'est pas explicite.



- **Exemple.** Méthode de Gauss-Legendre exacte $\mathbb{P}_5$.

$$\int_0^1 \varphi(t) dt \simeq \frac{5}{18} \varphi\left(\frac{1}{2} - \frac{\sqrt{15}}{10}\right) + \frac{4}{9} \varphi\left(\frac{1}{2}\right) + \frac{5}{18} \varphi\left(\frac{1}{2} + \frac{\sqrt{15}}{10}\right).$$

La méthode composée correspondante est en $\mathcal{O}(N^{-6})$.

Pour k grand, le calcul des points n'est pas explicite.



- **Remarque.** Si le poids $\omega \geq 0$ est tel que

$$\forall p \in \mathbb{P}, \quad \int_0^1 |p(t)| \omega(t) dt < +\infty,$$

alors on peut faire de même pour le produit scalaire $L^2(\omega(t) dt)$.

Rappel. Si $X \hookrightarrow \mathbb{U}([a, b])$ et $f : [a, b] \rightarrow \mathbb{R}$, alors

$$\mathbb{E}[f(X)] = \frac{1}{b-a} \int_a^b f(x) dx.$$

Si $(x_0, x_1, \dots, x_{N-1})$ est une réalisation d'un échantillon issu de la loi de X , alors

$$\frac{b-a}{N} \sum_{i=0}^{N-1} f(x_i) \quad \text{approche} \quad \int_a^b f(x) dx.$$

Rappel. Si $X \hookrightarrow \mathbb{U}([a, b])$ et $f : [a, b] \rightarrow \mathbb{R}$, alors

$$\mathbb{E}[f(X)] = \frac{1}{b-a} \int_a^b f(x) dx.$$

Si $(x_0, x_1, \dots, x_{N-1})$ est une réalisation d'un échantillon issu de la loi de X , alors

$$\frac{b-a}{N} \sum_{i=0}^{N-1} f(x_i) \quad \text{approche} \quad \int_a^b f(x) dx.$$

► Sous matlab :

```
% Rectangles a gauche  
x=a+(b-a)*(0:N-1)/N;  
Irect=(b-a)/N*sum(f(x));
```

```
% Monte-Carlo  
x=a+(b-a)*rand(N,1)/N;  
Imc=(b-a)/N*sum(f(x));
```

► Méthode en $\mathcal{O}(N^{-\frac{1}{2}})$, surtout utile en grande dimension.



COURS 3

Optimisation numérique

Optimisation numérique

Optimiser : une démarche universelle

► Mécanique :

Optimisation numérique

Optimiser : une démarche universelle

- ▶ Mécanique :

- ▶ Équilibre d'un système minimisant une énergie

Optimisation numérique

Optimiser : une démarche universelle

- ▶ Mécanique :

- ▶ Équilibre d'un système minimisant une énergie
- ▶ Structure qui maximise la résistance

Optimisation numérique

Optimiser : une démarche universelle

- ▶ Mécanique :

- ▶ Équilibre d'un système minimisant une énergie
- ▶ Structure qui maximise la résistance

- ▶ Transport :

Optimisation numérique

Optimiser : une démarche universelle

- ▶ Mécanique :

- ▶ Équilibre d'un système minimisant une énergie
- ▶ Structure qui maximise la résistance

- ▶ Transport :

- ▶ Minimiser la distance, le temps de trajet, le coût,...

Optimisation numérique

Optimiser : une démarche universelle

▶ Mécanique :

- ▶ Équilibre d'un système minimisant une énergie
- ▶ Structure qui maximise la résistance

▶ Transport :

- ▶ Minimiser la distance, le temps de trajet, le coût,...

▶ Economie, gestion :

Optimisation numérique

Optimiser : une démarche universelle

- ▶ **Mécanique :**

- ▶ Équilibre d'un système minimisant une énergie
- ▶ Structure qui maximise la résistance

- ▶ **Transport :**

- ▶ Minimiser la distance, le temps de trajet, le coût,...

- ▶ **Economie, gestion :**

- ▶ Maximiser le profit d'une entreprise

Optimisation numérique

Optimiser : une démarche universelle

▶ Mécanique :

- ▶ Équilibre d'un système minimisant une énergie
- ▶ Structure qui maximise la résistance

▶ Transport :

- ▶ Minimiser la distance, le temps de trajet, le coût,...

▶ Economie, gestion :

- ▶ Maximiser le profit d'une entreprise
- ▶ Minimiser les risques d'un placement boursier

Optimisation numérique

Optimiser : une démarche universelle

► Mécanique :

- Équilibre d'un système minimisant une énergie
- Structure qui maximise la résistance

► Transport :

- Minimiser la distance, le temps de trajet, le coût,...

► Economie, gestion :

- Maximiser le profit d'une entreprise
- Minimiser les risques d'un placement boursier

► Médecine :

Optimisation numérique

Optimiser : une démarche universelle

► Mécanique :

- Équilibre d'un système minimisant une énergie
- Structure qui maximise la résistance

► Transport :

- Minimiser la distance, le temps de trajet, le coût,...

► Economie, gestion :

- Maximiser le profit d'une entreprise
- Minimiser les risques d'un placement boursier

► Médecine :

- Optimiser une thérapie sous contraintes de dosage

Optimisation numérique

Optimiser : une démarche universelle

- ▶ **Mécanique :**
 - ▶ Équilibre d'un système minimisant une énergie
 - ▶ Structure qui maximise la résistance
- ▶ **Transport :**
 - ▶ Minimiser la distance, le temps de trajet, le coût,...
- ▶ **Economie, gestion :**
 - ▶ Maximiser le profit d'une entreprise
 - ▶ Minimiser les risques d'un placement boursier
- ▶ **Médecine :**
 - ▶ Optimiser une thérapie sous contraintes de dosage
- ▶ **Imagerie, problèmes inverses :**

Optimisation numérique

Optimiser : une démarche universelle

► Mécanique :

- Équilibre d'un système minimisant une énergie
- Structure qui maximise la résistance

► Transport :

- Minimiser la distance, le temps de trajet, le coût,...

► Economie, gestion :

- Maximiser le profit d'une entreprise
- Minimiser les risques d'un placement boursier

► Médecine :

- Optimiser une thérapie sous contraintes de dosage

► Imagerie, problèmes inverses :

- Restaurer une image abîmée par minimisation de variation

Optimisation numérique

Optimiser : une démarche universelle

► Mécanique :

- Équilibre d'un système minimisant une énergie
- Structure qui maximise la résistance

► Transport :

- Minimiser la distance, le temps de trajet, le coût,...

► Economie, gestion :

- Maximiser le profit d'une entreprise
- Minimiser les risques d'un placement boursier

► Médecine :

- Optimiser une thérapie sous contraintes de dosage

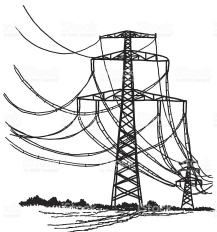
► Imagerie, problèmes inverses :

- Restaurer une image abîmée par minimisation de variation
- Résoudre des équations de forme $\varphi(x) = y$ en minimisant

$$\mathbf{f}(x) = \|\varphi(x) - y\|^2$$

Optimisation numérique

Position d'un câble soumis à la gravité



Quelle est la position exacte, la tension (mécanique), la distance au sol... des lignes HT ?

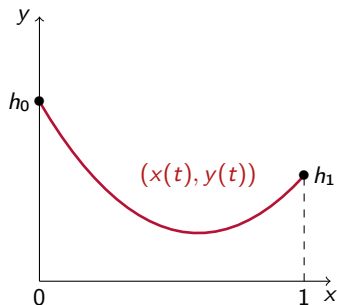


Idem pour les cables d'un téléphérique ?

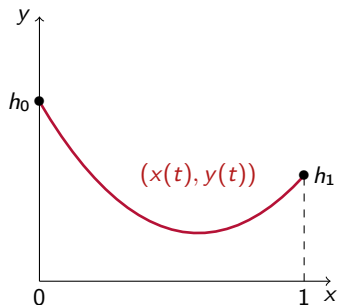
*Les cables **minimisent** leur énergie potentielle*

Optimisation numérique

Modélisation : câbles HT



- ▶ Inconnue : arc paramétré $(x(t), y(t))$
- ▶ Contraintes :
 - ▶ longueur totale L ,
 - ▶ extrémités fixées.



- ▶ Inconnue : arc paramétré $(x(t), y(t))$
- ▶ Contraintes :
 - ▶ longueur totale L ,
 - ▶ extrémités fixées.

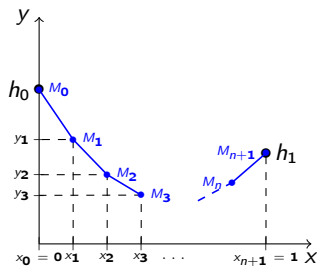
Théorème de l'énergie potentielle.

Le cable est à l'équilibre s'il minimise son énergie potentielle

Optimisation numérique

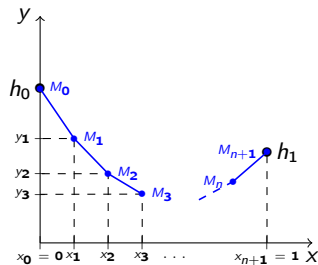
Modélisation du câble HT : discrétisation

Discrétisation de l'espace



Optimisation numérique

Modélisation du câble HT : discrétisation

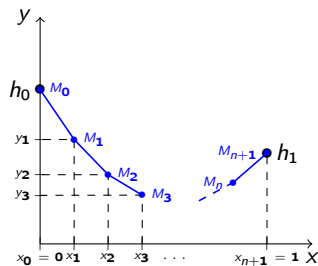


Discrétisation de l'espace

- Câble affine par morceaux

Optimisation numérique

Modélisation du câble HT : discrétisation

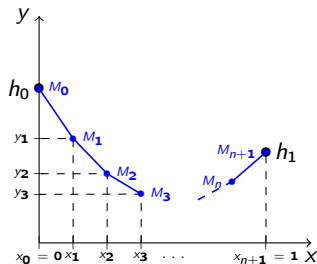


Discrétisation de l'espace

- Câble affine par morceaux
- $M_i = (x_i, y_i)$.

Optimisation numérique

Modélisation du câble HT : discrétisation



Discrétisation de l'espace

► Câble affine par morceaux

► $M_i = (x_i, y_i)$.

► On note

$$\mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_n).$$

$$\text{et } y_0 = h_0, y_{n+1} = h_1.$$

Optimisation numérique

Modélisation du câble HT : discrétisation

Discrétisation de l'espace

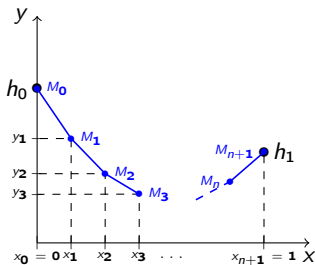
► Câble affine par morceaux

► $M_i = (x_i, y_i)$.

► On note

$$\mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_n).$$

$$\text{et } y_0 = h_0, y_{n+1} = h_1.$$



Problème discret : $\min \{f(\mathbf{x}, \mathbf{y}) ; (\mathbf{x}, \mathbf{y}) \in K\}$

avec

► $f(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n y_i$

► $K = \left\{ (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2n} ; (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = \frac{L^2}{(n+1)^2}, \forall i = 0 \dots n \right\}.$

Problème d'optimisation.

$$\min_{u \in K} f(u)$$

avec $K \subset \mathbb{R}^N$ et $f : K \rightarrow \mathbb{R}$.

Problème d'optimisation.

$$\min_{u \in K} f(u)$$

avec $K \subset \mathbb{R}^N$ et $f : K \rightarrow \mathbb{R}$.

Vocabulaire :

- f : fonction objectif ou énergie.

Problème d'optimisation.

$$\min_{u \in K} f(u)$$

avec $K \subset \mathbb{R}^N$ et $f : K \rightarrow \mathbb{R}$.

Vocabulaire :

- ▶ f : fonction objectif ou énergie.
- ▶ $\mathbb{R}^N$: espace des paramètres.

Problème d'optimisation.

$$\min_{u \in K} f(u)$$

avec $K \subset \mathbb{R}^N$ et $f : K \rightarrow \mathbb{R}$.

Vocabulaire :

- ▶ f : fonction objectif ou énergie.
- ▶ $\mathbb{R}^N$: espace des paramètres.
- ▶ K : ensemble admissible.

Problème d'optimisation.

$$\min_{u \in K} f(u)$$

avec $K \subset \mathbb{R}^N$ et $f : K \rightarrow \mathbb{R}$.

Vocabulaire :

- ▶ f : fonction objectif ou énergie.
- ▶ $\mathbb{R}^N$: espace des paramètres.
- ▶ K : ensemble admissible.
- ▶ Solution : point $u^* \in K$ qui minimise f sur K ; on écrit

$$f(u^*) = \min_{u \in K} f(u) \quad \text{et aussi} \quad u^* = \operatorname{argmin}_{u \in K} f(u)$$

Problème d'optimisation.

$$\min_{u \in K} f(u)$$

avec $K \subset \mathbb{R}^N$ et $f : K \rightarrow \mathbb{R}$.

Vocabulaire :

- ▶ f : fonction objectif ou énergie.
- ▶ $\mathbb{R}^N$: espace des paramètres.
- ▶ K : ensemble admissible.
- ▶ Solution : point $u^* \in K$ qui minimise f sur K ; on écrit

$$f(u^*) = \min_{u \in K} f(u) \quad \text{et aussi} \quad u^* = \operatorname{argmin}_{u \in K} f(u)$$

- ▶ $f(u^*)$: minimum de f sur K .

Problème d'optimisation.

$$\min_{u \in K} f(u)$$

avec $K \subset \mathbb{R}^N$ et $f : K \rightarrow \mathbb{R}$.

Vocabulaire :

- ▶ f : fonction objectif ou énergie.
- ▶ $\mathbb{R}^N$: espace des paramètres.
- ▶ K : ensemble admissible.
- ▶ Solution : point $u^* \in K$ qui minimise f sur K ; on écrit

$$f(u^*) = \min_{u \in K} f(u) \quad \text{et aussi} \quad u^* = \operatorname{argmin}_{u \in K} f(u)$$

- ▶ $f(u^*)$: minimum de f sur K .

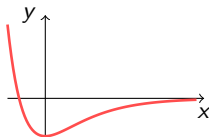
Attention : Il peut y avoir zéro, une ou plusieurs solutions au problème.

Optimisation libre dans $\mathbb{R}$

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **unimodale** s'il existe $x^* \in \mathbb{R}$ tel que f soit strictement décroissante sur $] - \infty, x^* [$ et strictement croissante sur $]x^*, +\infty [$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **unimodale** s'il existe $x^* \in \mathbb{R}$ tel que f soit strictement décroissante sur $] -\infty, x^* [$ et strictement croissante sur $]x^*, +\infty [$.

Exemple : $x \mapsto e^{-2x} - 2e^{-x}$



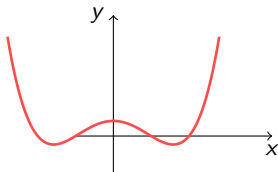
Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **unimodale** s'il existe $x^* \in \mathbb{R}$ tel que f soit strictement décroissante sur $] -\infty, x^* [$ et strictement croissante sur $]x^*, +\infty [$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **coercive** si $\lim_{|x| \rightarrow +\infty} f(x) = +\infty$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **unimodale** s'il existe $x^* \in \mathbb{R}$ tel que f soit strictement décroissante sur $] -\infty, x^* [$ et strictement croissante sur $]x^*, +\infty [$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **coercive** si $\lim_{|x| \rightarrow +\infty} f(x) = +\infty$.

Exemple : $x \mapsto x^4 - 5x^3 + 4$



Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **unimodale** s'il existe $x^* \in \mathbb{R}$ tel que f soit strictement décroissante sur $] -\infty, x^* [$ et strictement croissante sur $]x^*, +\infty [$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **coercive** si $\lim_{|x| \rightarrow +\infty} f(x) = +\infty$.

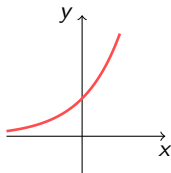
Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ de classe $\mathcal{C}^2$ est **fortement convexe** ssi $\exists \alpha > 0$ tel que pour tout $x \in \mathbb{R}$, $f''(x) \geq \alpha$. (On dit aussi **α -convexe**).

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **unimodale** s'il existe $x^* \in \mathbb{R}$ tel que f soit strictement décroissante sur $] - \infty, x^* [$ et strictement croissante sur $]x^*, +\infty[$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **coercive** si $\lim_{|x| \rightarrow +\infty} f(x) = +\infty$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ de classe $\mathcal{C}^2$ est **fortement convexe** ssi $\exists \alpha > 0$ tel que pour tout $x \in \mathbb{R}$, $f''(x) \geq \alpha$. (On dit aussi **α -convexe**).

Exemple : $x \mapsto e^x$ strictement convexe, mais pas fortement convexe.



Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **unimodale** s'il existe $x^* \in \mathbb{R}$ tel que f soit strictement décroissante sur $] - \infty, x^* [$ et strictement croissante sur $]x^*, +\infty[$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ est **coercive** si $\lim_{|x| \rightarrow +\infty} f(x) = +\infty$.

Définition. Une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ de classe $\mathcal{C}^2$ est **fortement convexe** ssi $\exists \alpha > 0$ tel que pour tout $x \in \mathbb{R}$, $f''(x) \geq \alpha$. (On dit aussi **α -convexe**).

Proposition.

- ▶ fortement convexe $\Rightarrow$ coercive
- ▶ fortement convexe et coercive $\Rightarrow$ unimodale

But : résoudre $F(x) = 0$

- $F : \mathbb{R} \rightarrow \mathbb{R}$ **croissante** et **continue**, s'annule en $x^* \in [a_0, b_0]$.

But : résoudre $F(x) = 0$

- ▶ $F : \mathbb{R} \rightarrow \mathbb{R}$ **croissante** et **continue**, s'annule en $x^* \in [a_0, b_0]$.
- ▶ On construit (a_n) , (b_n) par l'algorithme

On pose $x_n = \frac{a_n + b_n}{2}$.

- ▶ Si $F(x_n) \leq 0$,

$$a_{n+1} = x_n, \quad \text{et} \quad b_{n+1} = b_n.$$

- ▶ Si $F(x_n) > 0$,

$$a_{n+1} = a_n, \quad \text{et} \quad b_{n+1} = x_n.$$

But : résoudre $F(x) = 0$

- ▶ $F : \mathbb{R} \rightarrow \mathbb{R}$ **croissante** et **continue**, s'annule en $x^* \in [a_0, b_0]$.
- ▶ On construit (a_n) , (b_n) par l'algorithme

On pose $x_n = \frac{a_n + b_n}{2}$.

- ▶ Si $F(x_n) \leq 0$,

$$a_{n+1} = x_n, \quad \text{et} \quad b_{n+1} = b_n.$$

- ▶ Si $F(x_n) > 0$,

$$a_{n+1} = a_n, \quad \text{et} \quad b_{n+1} = x_n.$$

Proposition. La méthode converge à vitesse géométrique :

$$\forall n \in \mathbb{N}, \quad |x_n - x^*| \leq \frac{b_0 - a_0}{2^{n+1}}.$$

But : résoudre $\min f(x)$

- $f : \mathbb{R} \rightarrow \mathbb{R}$ **unimodale** et **dérivable**, minimale en $x^* \in [a_0, b_0]$.

But : résoudre $\min f(x)$

- ▶ $f : \mathbb{R} \rightarrow \mathbb{R}$ **unimodale** et **dérivable**, minimale en $x^* \in [a_0, b_0]$.
- ▶ On construit (a_n) , (b_n) par l'algorithme

On pose $x_n = \frac{a_n + b_n}{2}$.

- ▶ Si $f'(x_n) \leq 0$,

$$a_{n+1} = x_n, \quad \text{et} \quad b_{n+1} = b_n.$$

- ▶ Si $f'(x_n) > 0$,

$$a_{n+1} = a_n, \quad \text{et} \quad b_{n+1} = x_n.$$

But : résoudre $\min f(x)$

- ▶ $f : \mathbb{R} \rightarrow \mathbb{R}$ **unimodale** et **dérivable**, minimale en $x^* \in [a_0, b_0]$.
- ▶ On construit (a_n) , (b_n) par l'algorithme

On pose $x_n = \frac{a_n + b_n}{2}$.

- ▶ Si $f'(x_n) \leq 0$,

$$a_{n+1} = x_n, \quad \text{et} \quad b_{n+1} = b_n.$$

- ▶ Si $f'(x_n) > 0$,

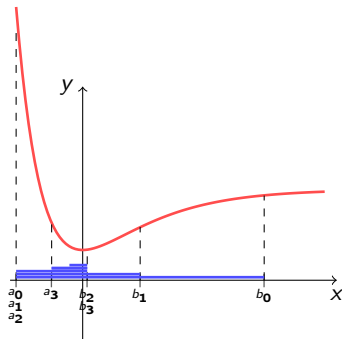
$$a_{n+1} = a_n, \quad \text{et} \quad b_{n+1} = x_n.$$

Proposition. La méthode converge à vitesse géométrique :

$$\forall n \in \mathbb{N}, \quad |x_n - x^*| \leq \frac{b_0 - a_0}{2^{n+1}}.$$

Optimisation numérique

Méthode de dichotomie 2



Avantages :

- ▶ Très simple à mettre en œuvre.
- ▶ Convergence rapide.

Inconvénient :

- ▶ Évaluation de f' .

Coût : À chaque itération, on évalue seulement une fois la dérivée de f .

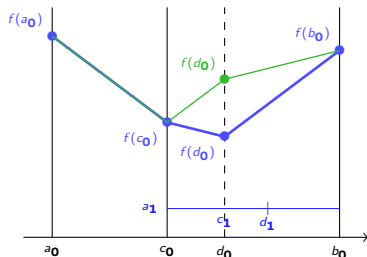


[DichoGold.m]

Optimisation numérique

Méthode du nombre d'or

Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ **unimodale**, minimale en $x^* \in [a_0, b_0]$.



Principe : On évalue f en les 4 points

$$a_0 < c_0 < d_0 < b_0.$$

Deux cas :

- ▶ Si $f(d_0) < f(c_0)$ alors $x^* \in [c_0, b_0]$
- ▶ Si $f(d_0) > f(c_0)$ alors $x^* \in [a_0, d_0]$

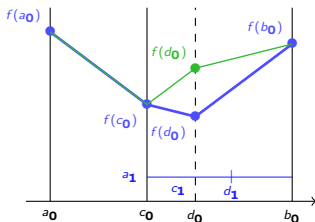
- ▶ La taille de l'intervalle diminue à chaque étape.
- ▶ **Coût :** À chaque itération, une seule nouvelle évaluation de f .



Optimisation numérique

Méthode du nombre d'or

Quel rapport avec le nombre d'or ?



► Par symétrie on impose

$$b_0 - c_0 = d_0 - a_0.$$

► Pour conserver le ratio des intervalles

$$\gamma = \frac{b_0 - a_0}{b_0 - c_0} = \frac{b_1 - a_1}{b_1 - c_1} = \frac{b_0 - c_0}{b_0 - d_0}$$

$$\frac{1}{\gamma} = \frac{b_0 - d_0}{b_0 - c_0} = \gamma - \frac{a_0 - d_0}{b_0 - c_0} = \gamma - 1$$

Donc $\gamma^2 - \gamma - 1 = 0$ et ainsi

$$\gamma = \frac{1 + \sqrt{5}}{2}$$

Optimisation numérique

Méthode du nombre d'or

- ▶ $f : \mathbb{R} \rightarrow \mathbb{R}$ **unimodale**, minimale en $x^* \in [a_0, b_0]$.
- ▶ On construit (a_n) , (b_n) par l'algorithme

▶ Si $f(c_n) \leq f(d_n)$,

$$\begin{array}{lcl} a_{n+1} & = & a_n \\ b_{n+1} & = & d_n \\ c_{n+1} & = & b_{n+1} - \frac{b_{n+1} - a_{n+1}}{\gamma} \\ d_{n+1} & = & c_n \end{array}$$

▶ Si $f(c_n) > f(d_n)$,

$$\begin{array}{lcl} a_{n+1} & = & c_n \\ b_{n+1} & = & b_n \\ c_{n+1} & = & d_n \\ d_{n+1} & = & a_{n+1} + \frac{b_{n+1} - a_{n+1}}{\gamma} \end{array}$$

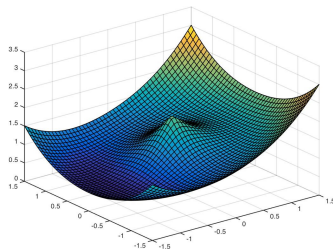
Proposition. La méthode converge à vitesse géométrique :

$$\forall n \in \mathbb{N}, \quad \left| \frac{a_n + b_n}{2} - x^* \right| \leq \frac{1}{2}(b_0 - a_0)\gamma^{-n}.$$

Optimisation libre dans $\mathbb{R}^N$

Exemple :

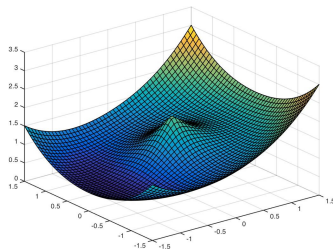
$$f(\mathbf{x}) = \left(\sqrt{x_1^2 + x_2^2} - 1 \right)^2 + \frac{x_1}{2} + 1.$$



Exemple :

$$f(\mathbf{x}) = \left(\sqrt{x_1^2 + x_2^2} - 1 \right)^2 + \frac{x_1}{2} + 1.$$

À partir d'un point initial $\mathbf{x}^0 \in \mathbb{R}^N$,

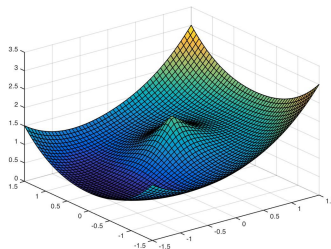


Optimisation numérique

Méthodes de descentes : principe

Exemple :

$$f(\mathbf{x}) = \left(\sqrt{x_1^2 + x_2^2} - 1 \right)^2 + \frac{x_1}{2} + 1.$$



À partir d'un point initial $\mathbf{x}^0 \in \mathbb{R}^N$,

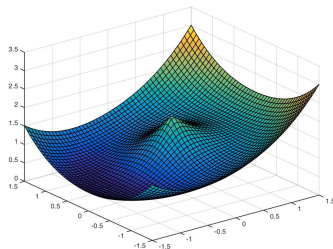
- ▶ on choisit une direction de descente $\mathbf{d}^0 \in \mathbb{R}^N$ et un pas $\rho_0 > 0$,

Optimisation numérique

Méthodes de descentes : principe

Exemple :

$$f(\mathbf{x}) = \left(\sqrt{x_1^2 + x_2^2} - 1 \right)^2 + \frac{x_1}{2} + 1.$$



À partir d'un point initial $\mathbf{x}^0 \in \mathbb{R}^N$,

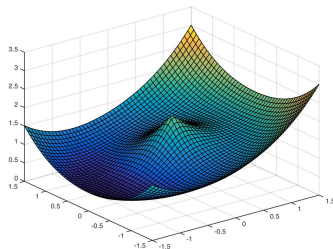
- ▶ on choisit une direction de descente $\mathbf{d}^0 \in \mathbb{R}^N$ et un pas $\rho_0 > 0$,
- ▶ on obtient un nouveau point $\mathbf{x}^1 = \mathbf{x}^0 + \rho_0 \mathbf{d}^0$,

Optimisation numérique

Méthodes de descentes : principe

Exemple :

$$f(\mathbf{x}) = \left(\sqrt{x_1^2 + x_2^2} - 1 \right)^2 + \frac{x_1}{2} + 1.$$



À partir d'un point initial $\mathbf{x}^0 \in \mathbb{R}^N$,

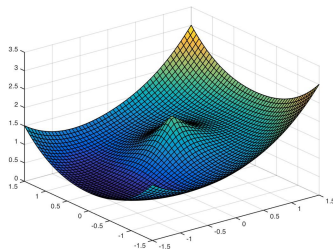
- ▶ on choisit une direction de descente $\mathbf{d}^0 \in \mathbb{R}^N$ et un pas $\rho_0 > 0$,
- ▶ on obtient un nouveau point $\mathbf{x}^1 = \mathbf{x}^0 + \rho_0 \mathbf{d}^0$,
- ▶ on itère le processus : $\mathbf{x}^{n+1} = \mathbf{x}^n + \rho_n \mathbf{d}^n$

Optimisation numérique

Méthodes de descentes : principe

Exemple :

$$f(\mathbf{x}) = \left(\sqrt{x_1^2 + x_2^2} - 1 \right)^2 + \frac{x_1}{2} + 1.$$



À partir d'un point initial $\mathbf{x}^0 \in \mathbb{R}^N$,

- ▶ on choisit une direction de descente $\mathbf{d}^0 \in \mathbb{R}^N$ et un pas $\rho_0 > 0$,
- ▶ on obtient un nouveau point $\mathbf{x}^1 = \mathbf{x}^0 + \rho_0 \mathbf{d}^0$,
- ▶ on itère le processus : $\boxed{\mathbf{x}^{n+1} = \mathbf{x}^n + \rho_n \mathbf{d}^n}$

But. faire en sorte que $f(\mathbf{x}^{n+1}) \leq f(\mathbf{x}^n)$.



Optimisation numérique

Gradient à pas fixe

Comment choisir la direction $\mathbf{d}^n$?

- ▶ On veut $(f(\mathbf{x}^n))_n$ décroissante.

Optimisation numérique

Gradient à pas fixe

Comment choisir la direction $\mathbf{d}^n$?

- ▶ On veut $(f(\mathbf{x}^n))_n$ décroissante.
- ▶ On choisit $\mathbf{d}^n$ telle que $\varphi : t \mapsto f(\mathbf{x}^n + t\mathbf{d}^n)$ soit décroissante au voisinage de 0.

Optimisation numérique

Gradient à pas fixe

Comment choisir la direction $\mathbf{d}^n$?

- ▶ On veut $(f(\mathbf{x}^n))_n$ décroissante.
- ▶ On choisit $\mathbf{d}^n$ telle que $\varphi : t \mapsto f(\mathbf{x}^n + t\mathbf{d}^n)$ soit décroissante au voisinage de 0.
- ▶ Or $\varphi'(0) = (\nabla f(\mathbf{x}^n) \mid \mathbf{d}^n)$.

Optimisation numérique

Gradient à pas fixe

Comment choisir la direction $\mathbf{d}^n$?

- ▶ On veut $(f(\mathbf{x}^n))_n$ décroissante.
- ▶ On choisit $\mathbf{d}^n$ telle que $\varphi : t \mapsto f(\mathbf{x}^n + t\mathbf{d}^n)$ soit décroissante au voisinage de 0.
- ▶ Or $\varphi'(0) = (\nabla f(\mathbf{x}^n) \mid \mathbf{d}^n)$.
- ▶ Un choix simple est

$$\mathbf{d}^n = -\nabla f(\mathbf{x}^n).$$

Comment choisir la direction $\mathbf{d}^n$?

- ▶ On veut $(f(\mathbf{x}^n))_n$ décroissante.
- ▶ On choisit $\mathbf{d}^n$ telle que $\varphi : t \mapsto f(\mathbf{x}^n + t\mathbf{d}^n)$ soit décroissante au voisinage de 0.
- ▶ Or $\varphi'(0) = (\nabla f(\mathbf{x}^n) \mid \mathbf{d}^n)$.
- ▶ Un choix simple est

$$\mathbf{d}^n = -\nabla f(\mathbf{x}^n).$$

Méthode du gradient à pas fixe. Elle est définie par

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - \rho \nabla f(\mathbf{x}^n). \end{cases}$$

avec $\rho > 0$.

Optimisation numérique

Gradient à pas fixe : convergence

Théorème. Soit $f \in \mathcal{C}^1(\mathbb{R}^N, \mathbb{R})$ strictement convexe et coercive telle que

$$(*) \quad \exists M > 0, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, \quad \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq M \|\mathbf{x} - \mathbf{y}\|_2.$$

Si $0 < \rho < \frac{2}{M}$ alors la méthode du gradient à pas fixe converge.

La limite est l'unique point de minimum global de f .

Optimisation numérique

Gradient à pas fixe : convergence

Théorème. Soit $f \in \mathcal{C}^1(\mathbb{R}^N, \mathbb{R})$ strictement convexe et coercive telle que

$$(*) \quad \exists M > 0, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, \quad \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq M \|\mathbf{x} - \mathbf{y}\|_2.$$

Si $0 < \rho < \frac{2}{M}$ alors la méthode du gradient à pas fixe **converge**.

La limite est l'unique point de minimum global de f .

Remarques :

- ▶ Condition $(*)$: $\mathbf{x} \mapsto \nabla f(\mathbf{x})$ est globalement M -lipschitzienne sur $\mathbb{R}^N$.
- ▶ Si on ajoute l'hypothèse (f est fortement convexe)

$$\exists \alpha > 0, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, \quad (Hf(\mathbf{x})\mathbf{y}|\mathbf{y}) \geq \alpha \|\mathbf{y}\|^2.$$

et $\rho < \frac{2\alpha}{M^2}$ alors la convergence est **géométrique**.

Optimisation numérique

Rappel sur la convexité

Pour des fonctions de classe $\mathcal{C}^2$

$f : \mathbb{R} \rightarrow \mathbb{R}$	$f : \mathbb{R}^N \rightarrow \mathbb{R}$	
$f'' \geq 0$	$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, (Hf(\mathbf{x})\mathbf{y} \mathbf{y}) \geq 0$	convexe
$f'' > 0$	$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, \mathbf{y} \neq 0, (Hf(\mathbf{x})\mathbf{y} \mathbf{y}) > 0$	$\implies$ strictement convexe
$f'' \geq \alpha$	$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, (Hf(\mathbf{x})\mathbf{y} \mathbf{y}) \geq \alpha \ \mathbf{y}\ _2^2$	fortement convexe ($\alpha > 0$)

Peut-on faire mieux ?

- ▶ On peut chercher le meilleur pas possible à chaque étape :

$$\rho_n \quad \text{minimise} \quad \rho \mapsto \mathbf{f}(\mathbf{x}^n - \rho \nabla \mathbf{f}(\mathbf{x}^n)).$$

Vocabulaire : cette étape est dite *recherche linéaire*.

Méthode : type dichotomie.

Peut-on faire mieux ?

- ▶ On peut chercher le meilleur pas possible à chaque étape :

$$\rho_n \text{ minimise } \rho \mapsto \mathbf{f}(\mathbf{x}^n - \rho \nabla \mathbf{f}(\mathbf{x}^n)).$$

Vocabulaire : cette étape est dite *recherche linéaire*.

Méthode : type dichotomie.

Méthode du gradient à pas optimal. Elle est définie par

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \rho_n = \arg \min_{\rho \in \mathbb{R}} \mathbf{f}(\mathbf{x}^n - \rho \nabla \mathbf{f}(\mathbf{x}^n)) \\ \mathbf{x}^{n+1} = \mathbf{x}^n - \rho_n \nabla \mathbf{f}(\mathbf{x}^n). \end{cases}$$

Peut-on faire mieux ?

- ▶ On peut chercher le meilleur pas possible à chaque étape :

$$\rho_n \text{ minimise } \rho \mapsto \mathbf{f}(\mathbf{x}^n - \rho \nabla \mathbf{f}(\mathbf{x}^n)).$$

Vocabulaire : cette étape est dite *recherche linéaire*.

Méthode : type dichotomie.

Méthode du gradient à pas optimal. Elle est définie par

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \rho_n = \arg \min_{\rho \in \mathbb{R}} \mathbf{f}(\mathbf{x}^n - \rho \nabla \mathbf{f}(\mathbf{x}^n)) \\ \mathbf{x}^{n+1} = \mathbf{x}^n - \rho_n \nabla \mathbf{f}(\mathbf{x}^n). \end{cases}$$

Remarque. même type de résultats de convergence que pour le pas fixe.

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.



Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.
- ▶ Rappel : $\nabla f(\mathbf{x}) = 2\mathbf{A}^T(\mathbf{Ax} - \mathbf{b})$

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.
- ▶ Rappel : $\nabla f(\mathbf{x}) = 2\mathbf{A}^T(\mathbf{Ax} - \mathbf{b})$

Pour trouver le pas optimal, il faut résoudre

$$\min_{\rho > 0} f(\mathbf{x} - \rho \nabla f(\mathbf{x})).$$

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.
- ▶ Rappel : $\nabla f(\mathbf{x}) = 2A^T(\mathbf{Ax} - \mathbf{b})$

Pour trouver le pas optimal, il faut résoudre

$$\min_{\rho > 0} f(\mathbf{x} - \rho \nabla f(\mathbf{x})).$$

$$f(\mathbf{x} - \rho \nabla f(\mathbf{x})) = \|\mathbf{Ax} - \mathbf{b} - \rho A \nabla f(\mathbf{x})\|_2^2$$

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.
- ▶ Rappel : $\nabla f(\mathbf{x}) = 2A^T(\mathbf{Ax} - \mathbf{b})$

Pour trouver le pas optimal, il faut résoudre

$$\min_{\rho > 0} f(\mathbf{x} - \rho \nabla f(\mathbf{x})).$$

$$f(\mathbf{x} - \rho \nabla f(\mathbf{x})) = \|\mathbf{Ax} - \mathbf{b}\|_2^2 - 2\rho(\mathbf{Ax} - \mathbf{b})^T A \nabla f(\mathbf{x}) + \rho^2 \|A \nabla f(\mathbf{x})\|_2^2$$

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.
- ▶ Rappel : $\nabla f(\mathbf{x}) = 2A^T(\mathbf{Ax} - \mathbf{b})$

Pour trouver le pas optimal, il faut résoudre

$$\min_{\rho > 0} f(\mathbf{x} - \rho \nabla f(\mathbf{x})).$$

$$f(\mathbf{x} - \rho \nabla f(\mathbf{x})) = \|\mathbf{Ax} - \mathbf{b}\|_2^2 - 2\rho(A^T \mathbf{Ax} - A^T \mathbf{b} | \nabla f(\mathbf{x})) + \rho^2 \|A \nabla f(\mathbf{x})\|_2^2$$

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.
- ▶ Rappel : $\nabla f(\mathbf{x}) = 2A^T(\mathbf{Ax} - \mathbf{b})$

Pour trouver le pas optimal, il faut résoudre

$$\min_{\rho > 0} f(\mathbf{x} - \rho \nabla f(\mathbf{x})).$$

$$f(\mathbf{x} - \rho \nabla f(\mathbf{x})) = \rho^2 \|A \nabla f(\mathbf{x})\|_2^2 - \rho \|\nabla f(\mathbf{x})\|_2^2 + \|\mathbf{Ax} - \mathbf{b}\|_2^2$$

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.
- ▶ Rappel : $\nabla f(\mathbf{x}) = 2A^T(\mathbf{Ax} - \mathbf{b})$

Pour trouver le pas optimal, il faut résoudre

$$\min_{\rho > 0} f(\mathbf{x} - \rho \nabla f(\mathbf{x})).$$

$$f(\mathbf{x} - \rho \nabla f(\mathbf{x})) = \rho^2 \|A \nabla f(\mathbf{x})\|_2^2 - \rho \|\nabla f(\mathbf{x})\|_2^2 + \|\mathbf{Ax} - \mathbf{b}\|_2^2$$

C'est un polynôme de degré 2 en ρ !

Optimisation numérique

Gradient à pas optimal pour les moindres carrés

Exemple : Minimisation de $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$.

- ▶ On parle de résolution de $\mathbf{Ax} = \mathbf{b}$ *au sens des moindres carrés*.
- ▶ Éventuellement, A est rectangulaire (injective).
- ▶ Lien avec l'approximation polynomiale.
- ▶ Rappel : $\nabla f(\mathbf{x}) = 2A^T(\mathbf{Ax} - \mathbf{b})$

Pour trouver le pas optimal, il faut résoudre

$$\min_{\rho > 0} f(\mathbf{x} - \rho \nabla f(\mathbf{x})).$$

$$f(\mathbf{x} - \rho \nabla f(\mathbf{x})) = \rho^2 \|A \nabla f(\mathbf{x})\|_2^2 - \rho \|\nabla f(\mathbf{x})\|_2^2 + \|\mathbf{Ax} - \mathbf{b}\|_2^2$$

C'est un polynôme de degré 2 en ρ !

$$\rho^* = \frac{\|\nabla f(\mathbf{x})\|_2^2}{2 \|A \nabla f(\mathbf{x})\|_2^2}.$$

- Résolution d'une équation $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ avec $\mathbf{F} \in \mathcal{C}^1(\mathbb{R}^N, \mathbb{R}^N)$.

Méthode de Newton pour les zeros de $\mathbf{F}$

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - J\mathbf{F}(\mathbf{x}^n)^{-1}\mathbf{F}(\mathbf{x}^n). \end{cases}$$

- Résolution d'une équation $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ avec $\mathbf{F} \in \mathcal{C}^1(\mathbb{R}^N, \mathbb{R}^N)$.

Méthode de Newton pour les zéros de $\mathbf{F}$

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - J\mathbf{F}(\mathbf{x}^n)^{-1}\mathbf{F}(\mathbf{x}^n). \end{cases}$$

Faire un dessin 1D

- Résolution d'une équation $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ avec $\mathbf{F} \in \mathcal{C}^1(\mathbb{R}^N, \mathbb{R}^N)$.

Méthode de Newton pour les zéros de $\mathbf{F}$

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - J\mathbf{F}(\mathbf{x}^n)^{-1}\mathbf{F}(\mathbf{x}^n). \end{cases}$$

- Minimisation de $f \in \mathcal{C}^2(\mathbb{R}^N, \mathbb{R})$. On résout $\nabla f(\mathbf{x}) = \mathbf{0}$.

Méthode de Newton pour minimiser f

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - Hf(\mathbf{x}^n)^{-1}\nabla f(\mathbf{x}^n). \end{cases}$$

- Résolution d'une équation $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ avec $\mathbf{F} \in \mathcal{C}^1(\mathbb{R}^N, \mathbb{R}^N)$.

Méthode de Newton pour les zéros de $\mathbf{F}$

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - J\mathbf{F}(\mathbf{x}^n)^{-1}\mathbf{F}(\mathbf{x}^n). \end{cases}$$

- Minimisation de $f \in \mathcal{C}^2(\mathbb{R}^N, \mathbb{R})$. On résout $\nabla f(\mathbf{x}) = \mathbf{0}$.

Méthode de Newton pour minimiser f

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - Hf(\mathbf{x}^n)^{-1}\nabla f(\mathbf{x}^n). \end{cases}$$

Remarques :

- La méthode est **locale**, mais rapide,
- Ne distingue pas min et max.

Optimisation sous contraintes

Optimisation numérique

Exemple simple

Rappel : Un problème d'optimisation sous contraintes s'écrit

$$\min_{\mathbf{u} \in K} \mathbf{f}(\mathbf{u}), \quad \text{avec } K \subseteq \mathbb{R}^N.$$

Optimisation numérique

Exemple simple

Rappel : Un problème d'optimisation sous contraintes s'écrit

$$\min_{\mathbf{u} \in K} \mathbf{f}(\mathbf{u}), \quad \text{avec } K \subsetneq \mathbb{R}^N.$$

En général, les algorithmes précédents vont converger **en dehors de K** !

Optimisation numérique

Exemple simple

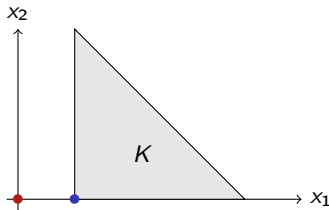
Rappel : Un problème d'optimisation sous contraintes s'écrit

$$\min_{\mathbf{u} \in K} f(\mathbf{u}), \quad \text{avec } K \subsetneq \mathbb{R}^N.$$

En général, les algorithmes précédents vont converger **en dehors de K** !

Exemple : $f(\mathbf{x}) = x_1^2 + x_2^2$.

$$K = \left\{ \mathbf{x} \in \mathbb{R}^2, x_1 \geq \frac{1}{2}, x_2 \geq 0, x_1 + x_2 \leq 1 \right\}$$



Soit $K \subset \mathbb{R}^N$ convexe fermé et non vide.

Définition. Pour tout $\mathbf{x} \in \mathbb{R}^N$, le projeté $\pi_K(\mathbf{x})$ est l'unique point de K qui minimise sa distance à $\mathbf{x}$.

$$\|\pi_K(\mathbf{x}) - \mathbf{x}\| = \min \left\{ \|\mathbf{y} - \mathbf{x}\| ; \mathbf{y} \in K \right\}.$$

Soit $K \subset \mathbb{R}^N$ convexe fermé et non vide.

Définition. Pour tout $\mathbf{x} \in \mathbb{R}^N$, le projeté $\pi_K(\mathbf{x})$ est l'unique point de K qui minimise sa distance à $\mathbf{x}$.

$$\|\pi_K(\mathbf{x}) - \mathbf{x}\| = \min \left\{ \|\mathbf{y} - \mathbf{x}\| ; \mathbf{y} \in K \right\}.$$

Remarques :

- ▶ si $\mathbf{x} \in K$ alors $\pi_K(\mathbf{x}) = \mathbf{x}$,
- ▶ si $\mathbf{x} \notin K$ alors $\pi_K(\mathbf{x}) \in \partial K$,
- ▶ π_K est contractante donc lipschitzienne donc continue.

Méthode du gradient projeté. Elle est définie par

$$\begin{cases} x^0 \in \mathbb{R}^N, \\ x^{n+1} = \pi_K(x^n - \rho \nabla f(x^n)). \end{cases}$$

avec $\rho > 0$.

Méthode du gradient projeté. Elle est définie par

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \pi_K(\mathbf{x}^n - \rho \nabla \mathbf{f}(\mathbf{x}^n)). \end{cases}$$

avec $\rho > 0$.

Théorème. Soient $\mathbf{f} \in \mathcal{C}^1(\mathbb{R}^N, \mathbb{R})$ strictement convexe et coercive, et K convexe fermé non vide, avec

$$\exists M > 0, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, \quad \|\nabla \mathbf{f}(\mathbf{x}) - \nabla \mathbf{f}(\mathbf{y})\|_2 \leq M \|\mathbf{x} - \mathbf{y}\|_2.$$

Si $0 < \rho < \frac{2}{M}$ alors la méthode du gradient projeté **converge** vers l'unique point de minimum global de $\mathbf{f}$ sur K .

Optimisation numérique

Méthode du gradient projeté

Avantage :

- convergence vers la solution du problème **contraint**.

Optimisation numérique

Méthode du gradient projeté

Avantage :

- ▶ convergence vers la solution du problème **contraint**.

Inconvénients :

- ▶ vitesse de convergence non garantie,
- ▶ le projecteur π_K peut être (très) difficile à calculer,

Optimisation numérique

Méthode du gradient projeté

Avantage :

- ▶ convergence vers la solution du problème **contraint**.

Inconvénients :

- ▶ vitesse de convergence non garantie,
- ▶ le projecteur π_K peut être (très) difficile à calculer,

Alternative possible :

- ▶ Méthode de pénalisation.

Définition. On appelle **fonction de pénalisation** de K toute fonction $\beta : \mathbb{R}^N \rightarrow \mathbb{R}$ telle que

- ▶ β est continue
- ▶ $\beta \geq 0$ sur $\mathbb{R}^N$,
- ▶ $\beta(\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} \in K$

Définition. On appelle **fonction de pénalisation** de K toute fonction $\beta : \mathbb{R}^N \rightarrow \mathbb{R}$ telle que

- ▶ β est continue
- ▶ $\beta \geq 0$ sur $\mathbb{R}^N$,
- ▶ $\beta(\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} \in K$

Remarque : Si c'est possible on choisira β convexe. (Il faut K convexe).

Définition. On appelle **fonction de pénalisation** de K toute fonction $\beta : \mathbb{R}^N \rightarrow \mathbb{R}$ telle que

- ▶ β est continue
- ▶ $\beta \geq 0$ sur $\mathbb{R}^N$,
- ▶ $\beta(\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} \in K$

Remarque : Si c'est possible on choisira β convexe. (Il faut K convexe).

Définition. Soit le problème d'optimisation sous contraintes

$$\min_{\mathbf{x} \in K} f(\mathbf{x}),$$

le problème pénalisé associé est

$$\min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + \frac{1}{\varepsilon} \beta(\mathbf{x}).$$

Définition. On appelle **fonction de pénalisation** de K toute fonction $\beta : \mathbb{R}^N \rightarrow \mathbb{R}$ telle que

- ▶ β est continue
- ▶ $\beta \geq 0$ sur $\mathbb{R}^N$,
- ▶ $\beta(\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} \in K$

Remarque : Si c'est possible on choisira β convexe. (Il faut K convexe).

Définition. Soit le problème d'optimisation sous contraintes

$$\min_{\mathbf{x} \in K} f(\mathbf{x}),$$

le problème pénalisé associé est

$$\min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + \frac{1}{\varepsilon} \beta(\mathbf{x}).$$

Pourquoi ça marche ? On fait payer (*pénalise*) le fait de ne pas être dans K .

- Si $K_1 = \{\mathbf{x} \in \mathbb{R}^N, \varphi_i(\mathbf{x}) = 0, i = 1 \dots p\}$, on peut choisir

$$\beta(\mathbf{x}) = \sum_{i=1}^p [\varphi_i(\mathbf{x})]^2.$$



- Si $K_1 = \{\mathbf{x} \in \mathbb{R}^N, \varphi_i(\mathbf{x}) = 0, i = 1 \dots p\}$, on peut choisir

$$\beta(\mathbf{x}) = \sum_{i=1}^p [\varphi_i(\mathbf{x})]^2.$$

- Si $K_2 = \{\mathbf{x} \in \mathbb{R}^N, \psi_j(\mathbf{x}) \leq 0, i = 1 \dots q\}$, on peut choisir

$$\beta(\mathbf{x}) = \sum_{j=1}^q [\psi_j^+(\mathbf{x})]^2.$$

avec $\psi_j^+ = \max(0, \psi_j)$.

- Si $K_1 = \{\mathbf{x} \in \mathbb{R}^N, \varphi_i(\mathbf{x}) = 0, i = 1 \dots p\}$, on peut choisir

$$\beta(\mathbf{x}) = \sum_{i=1}^p [\varphi_i(\mathbf{x})]^2.$$

- Si $K_2 = \{\mathbf{x} \in \mathbb{R}^N, \psi_j(\mathbf{x}) \leq 0, j = 1 \dots q\}$, on peut choisir

$$\beta(\mathbf{x}) = \sum_{j=1}^q [\psi_j^+(\mathbf{x})]^2.$$

avec $\psi_j^+ = \max(0, \psi_j)$.

- Si $K = K_1 \cap K_2$, on somme :

$$\beta(\mathbf{x}) = \sum_{i=1}^p [\varphi_i(\mathbf{x})]^2 + \sum_{j=1}^q [\psi_j^+(\mathbf{x})]^2.$$

Optimisation numérique

Méthode du gradient pénalisé

On résout le problème pénalisé (sans contrainte !)

$$\min_{\mathbf{x} \in \mathbb{R}^N} \mathbf{f}(\mathbf{x}) + \frac{1}{\varepsilon} \beta(\mathbf{x}).$$

par la méthode du gradient à pas fixe.

Optimisation numérique

Méthode du gradient pénalisé

On résout le problème pénalisé (sans contrainte !)

$$\min_{\mathbf{x} \in \mathbb{R}^N} \mathbf{f}(\mathbf{x}) + \frac{1}{\varepsilon} \beta(\mathbf{x}).$$

par la méthode du gradient à pas fixe.

Méthode du gradient pénalisé. Elle est définie par

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - \rho \nabla \mathbf{f}(\mathbf{x}^n) - \frac{\rho}{\varepsilon} \nabla \beta(\mathbf{x}). \end{cases}$$

avec $\rho > 0$.

Optimisation numérique

Méthode du gradient pénalisé

On résout le problème pénalisé (sans contrainte !)

$$\min_{\mathbf{x} \in \mathbb{R}^N} \mathbf{f}(\mathbf{x}) + \frac{1}{\varepsilon} \beta(\mathbf{x}).$$

par la méthode du gradient à pas fixe.

Méthode du gradient pénalisé. Elle est définie par

$$\begin{cases} \mathbf{x}^0 \in \mathbb{R}^N, \\ \mathbf{x}^{n+1} = \mathbf{x}^n - \rho \nabla \mathbf{f}(\mathbf{x}^n) - \frac{\rho}{\varepsilon} \nabla \beta(\mathbf{x}). \end{cases}$$

avec $\rho > 0$.

Remarque :

- ▶ Choix de ε difficile !
- ▶ Conditionne le choix de ρ ...

Difficulté : La méthode du gradient pénalisé va converger vers $\mathbf{x}_\varepsilon$ solution de

$$\min_{\mathbf{u} \in \mathbb{R}^N} \mathbf{f}(\mathbf{x}) + \frac{1}{\varepsilon} \beta(\mathbf{x}).$$

Si ε est petit, on s'attend à ce que $\mathbf{x}_\varepsilon$ soit proche de $\mathbf{x}^*$, solution de

$$\min_{\mathbf{x} \in K} \mathbf{f}(\mathbf{x}).$$

Optimisation numérique

Méthode du gradient pénalisé

Difficulté : La méthode du gradient pénalisé va converger vers $\mathbf{x}_\varepsilon$ solution de

$$\min_{\mathbf{u} \in \mathbb{R}^N} \mathbf{f}(\mathbf{x}) + \frac{1}{\varepsilon} \beta(\mathbf{x}).$$

Si ε est petit, on s'attend à ce que $\mathbf{x}_\varepsilon$ soit proche de $\mathbf{x}^*$, solution de

$$\min_{\mathbf{x} \in K} \mathbf{f}(\mathbf{x}).$$

Théorème. Soient $\mathbf{f} \in \mathcal{C}^1(\mathbb{R}^N, \mathbb{R})$ strictement convexe et coercive, et $K \subset \mathbb{R}^N$ convexe fermé non vide. Soit β une pénalisation de K . Alors pour tout $\varepsilon > 0$, le problème

$$\min_{\mathbf{x} \in \mathbb{R}^N} \mathbf{f}(\mathbf{x}) + \frac{1}{\varepsilon} \beta(\mathbf{x})$$

admet une unique solution $\mathbf{x}_\varepsilon$. Elle vérifie

$$\lim_{\varepsilon \rightarrow 0} \mathbf{x}_\varepsilon = \mathbf{x}^*$$

où $\mathbf{x}^*$ est l'unique solution du problème initial $\min_{\mathbf{x} \in K} \mathbf{f}(\mathbf{x})$.

COURS 4

Approximation numérique des équations différentielles

Approximation des EDO

Quelques usages

- ▶ Mécanique :

Approximation des EDO

Quelques usages

- ▶ **Mécanique :**

- ▶ Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, ...)



Approximation des EDO

Quelques usages

- ▶ **Mécanique :**
 - ▶ Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, . . .)
 - ▶ Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,

Approximation des EDO

Quelques usages

► Mécanique :

- Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, . . .)
- Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- Militaire (balistique),

Approximation des EDO

Quelques usages

- ▶ Mécanique :
 - ▶ Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, . . .)
 - ▶ Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
 - ▶ Militaire (balistique),
- ▶ Biologie, environnement, :

Approximation des EDO

Quelques usages

► Mécanique :

- Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, ...)
- Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- Militaire (balistique),

► Biologie, environnement, :

- Evolution d'un écosystème, d'une population, croissance et extinction,

Approximation des EDO

Quelques usages

▶ Mécanique :

- ▶ Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, ...)
- ▶ Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- ▶ Militaire (balistique),

▶ Biologie, environnement, :

- ▶ Evolution d'un écosystème, d'une population, croissance et extinction,

▶ Chimie, industrie :

Approximation des EDO

Quelques usages

▶ Mécanique :

- ▶ Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, ...)
- ▶ Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- ▶ Militaire (balistique),

▶ Biologie, environnement, :

- ▶ Evolution d'un écosystème, d'une population, croissance et extinction,

▶ Chimie, industrie :

- ▶ Dynamique des processus,

Approximation des EDO

Quelques usages

► Mécanique :

- Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, ...)
- Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- Militaire (balistique),

► Biologie, environnement, :

- Evolution d'un écosystème, d'une population, croissance et extinction,

► Chimie, industrie :

- Dynamique des processus,

► Economie, gestion :

Approximation des EDO

Quelques usages

► Mécanique :

- Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, ...)
- Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- Militaire (balistique),

► Biologie, environnement, :

- Evolution d'un écosystème, d'une population, croissance et extinction,

► Chimie, industrie :

- Dynamique des processus,

► Economie, gestion :

- Modélisation macroéconomique,

Approximation des EDO

Quelques usages

▶ Mécanique :

- ▶ Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, . . .)
- ▶ Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- ▶ Militaire (balistique),

▶ Biologie, environnement, :

- ▶ Evolution d'un écosystème, d'une population, croissance et extinction,

▶ Chimie, industrie :

- ▶ Dynamique des processus,

▶ Economie, gestion :

- ▶ Modélisation macroéconomique,
- ▶ Modèles financiers,

Approximation des EDO

Quelques usages

▶ Mécanique :

- ▶ Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, ...)
- ▶ Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- ▶ Militaire (balistique),

▶ Biologie, environnement, :

- ▶ Evolution d'un écosystème, d'une population, croissance et extinction,

▶ Chimie, industrie :

- ▶ Dynamique des processus,

▶ Economie, gestion :

- ▶ Modélisation macroéconomique,
- ▶ Modèles financiers,

▶ Médecine :

Approximation des EDO

Quelques usages

► Mécanique :

- Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, . . .)
- Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- Militaire (balistique),

► Biologie, environnement, :

- Evolution d'un écosystème, d'un population, croissance et extinction,

► Chimie, industrie :

- Dynamique des processus,

► Economie, gestion :

- Modélisation macroéconomique,
- Modèles financiers,

► Médecine :

- Posologie, absorption et efficacité des traitements, croissance tumorales,

Approximation des EDO

Quelques usages

► Mécanique :

- Prédiction de trajectoire (planètes, comètes, astéroïdes, débris, ...)
- Orbite des satellites, manœuvre spatiales, rentrée atmosphérique,
- Militaire (balistique),

► Biologie, environnement, :

- Evolution d'un écosystème, d'une population, croissance et extinction,

► Chimie, industrie :

- Dynamique des processus,

► Economie, gestion :

- Modélisation macroéconomique,
- Modèles financiers,

► Médecine :

- Posologie, absorption et efficacité des traitements, croissance tumorales,
- Epidémiologie.

Approximation des EDO

Une des premières utilisations célèbre



Mercury-Atlas 6 : comment calculer une trajectoire de rentrée atmosphérique ?



Comment viser le Kazakhstan avec un Soyuz ?

Pourquoi seule l'analyse numérique peut résoudre le problème ?

Quelles méthodes ?

Quelle précision ?

Approximation des EDO

Constat d'échec

Important : en général, on ne sait pas résoudre explicitement une EDO !

Approximation des EDO

Constat d'échec

Important : en général, **on ne sait pas résoudre explicitement une EDO !**

On s'en sort quand...

Approximation des EDO

Constat d'échec

Important : en général, **on ne sait pas résoudre explicitement une EDO !**

On s'en sort quand...

- ▶ l'équation différentielle est **linéaire** à coefficients explicitement intégrables,

Approximation des EDO

Constat d'échec

Important : en général, **on ne sait pas résoudre explicitement une EDO !**

On s'en sort quand...

- ▶ l'équation différentielle est **linéaire** à coefficients explicitement intégrables,
- ▶ l'exercice est fait pour qu'on y arrive,

Approximation des EDO

Constat d'échec

Important : en général, **on ne sait pas résoudre explicitement une EDO !**

On s'en sort quand...

- ▶ l'équation différentielle est **linéaire** à coefficients explicitement intégrables,
- ▶ l'exercice est fait pour qu'on y arrive,
- ▶ on a beaucoup de **chance**.

Approximation des EDO

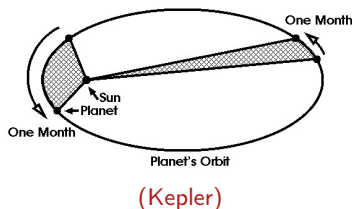
Constat d'échec

Important : en général, **on ne sait pas résoudre explicitement une EDO !**

On s'en sort quand...

- ▶ l'équation différentielle est **linéaire** à coefficients explicitement intégrables,
- ▶ l'exercice est fait pour qu'on y arrive,
- ▶ on a beaucoup de **chance**.

Exemple de coup de chance : les orbites de planètes sont des ellipses.



Équation gravitationnelle :

$$\mathbf{x}'' = -Gm_S \frac{\mathbf{x}}{\|\mathbf{x}\|^3}.$$

Approximation des EDO

Modélisation rentrée atmosphérique



Modélisation du problème de rentrée atmosphérique :

- Loi de Newton :

$m\mathbf{x}'' = \text{force de gravité} + \text{force de frottement}$

$$m\mathbf{x}'' = -\frac{Gmm_T}{\|\mathbf{x}\|^3}\mathbf{x} - \alpha(\mathbf{x})\|\mathbf{x}'\|\mathbf{x}'$$

Approximation des EDO

Modélisation rentrée atmosphérique



Modélisation du problème de rentrée atmosphérique :

- Loi de Newton :

$m\mathbf{x}'' = \text{force de gravité} + \text{force de frottement}$

$$m\mathbf{x}'' = -\frac{Gmm_T}{\|\mathbf{x}\|^3}\mathbf{x} - \alpha(\mathbf{x})\|\mathbf{x}'\|\mathbf{x}'$$

- Inconnue : $\mathbf{x} : [0, T] \rightarrow \mathbb{R}^2 \text{ ou } \mathbb{R}^3$

Approximation des EDO

Modélisation rentrée atmosphérique



Modélisation du problème de rentrée atmosphérique :

- Loi de Newton :

$m\mathbf{x}'' = \text{force de gravité} + \text{force de frottement}$

$$m\mathbf{x}'' = -\frac{Gmm_T}{\|\mathbf{x}\|^3}\mathbf{x} - \alpha(\mathbf{x})\|\mathbf{x}'\|\mathbf{x}'$$

- Inconnue : $\mathbf{x} : [0, T] \rightarrow \mathbb{R}^2 \text{ ou } \mathbb{R}^3$
- Domaine : pour tout $t \in [0, T]$, $\mathbf{x}(t) \in \Omega := \{\mathbf{x} \in \mathbb{R}^d, \|\mathbf{x}\| > R_T\}$.

Approximation des EDO

Modélisation rentrée atmosphérique



Modélisation du problème de rentrée atmosphérique :

- Loi de Newton :

$m\mathbf{x}'' = \text{force de gravité} + \text{force de frottement}$

$$m\mathbf{x}'' = -\frac{Gmm_T}{\|\mathbf{x}\|^3}\mathbf{x} - \alpha(\mathbf{x})\|\mathbf{x}'\|\mathbf{x}'$$

- Inconnue : $\mathbf{x} : [0, T] \rightarrow \mathbb{R}^2 \text{ ou } \mathbb{R}^3$
- Domaine : pour tout $t \in [0, T]$, $\mathbf{x}(t) \in \Omega := \{\mathbf{x} \in \mathbb{R}^d, \|\mathbf{x}\| > R_T\}$.
- Données : G cste de gravité, m masse du vaisseau, m_T masse de la Terre, $\alpha(\mathbf{x})$ coefficient de frottement aérodynamique, R_T rayon de la Terre.

Méthode : on met l'équation sous forme d'un problème de Cauchy.

Problème de Cauchy. C'est un problème différentiel de la forme

$$(C) : \begin{cases} u'(t) = f(t, u(t)) & t \in [0, T], \\ u(0) = u_0. \end{cases}$$

Méthode : on met l'équation sous forme d'un problème de Cauchy.

Problème de Cauchy. C'est un problème différentiel de la forme

$$(C) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

► $\mathbf{u} : [0, T] \rightarrow \Omega \subset \mathbb{R}^d$ est la **fonction inconnue** ,

Méthode : on met l'équation sous forme d'un problème de Cauchy.

Problème de Cauchy. C'est un problème différentiel de la forme

$$(C) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

- ▶ $\mathbf{u} : [0, T] \rightarrow \Omega \subset \mathbb{R}^d$ est la **fonction inconnue** ,
- ▶ le **domaine spatial** Ω est une partie ouverte connexe de l'espace $\mathbb{R}^d$,

Méthode : on met l'équation sous forme d'un problème de Cauchy.

Problème de Cauchy. C'est un problème différentiel de la forme

$$(C) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

- ▶ $\mathbf{u} : [0, T] \rightarrow \Omega \subset \mathbb{R}^d$ est la **fonction inconnue** ,
- ▶ le **domaine spatial** Ω est une partie ouverte connexe de l'espace $\mathbb{R}^d$,
- ▶ $\mathbf{f} : \Omega \times [0, T] \mapsto \mathbb{R}^d$ est la **dynamique** du problème,

Méthode : on met l'équation sous forme d'un problème de Cauchy.

Problème de Cauchy. C'est un problème différentiel de la forme

$$(C) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

- ▶ $\mathbf{u} : [0, T] \rightarrow \Omega \subset \mathbb{R}^d$ est la **fonction inconnue** ,
- ▶ le **domaine spatial** Ω est une partie ouverte connexe de l'espace $\mathbb{R}^d$,
- ▶ $\mathbf{f} : \Omega \times [0, T] \mapsto \mathbb{R}^d$ est la **dynamique** du problème,
- ▶ $\mathbf{u}_0 \in \Omega$ est l'**état initial** où condition initiale.

Approximation des EDO

Problème de Cauchy pour la rentrée atmosphérique

Exemple : rentrée atmosphérique,

$$\mathbf{x}'' = -\frac{Gm_T}{\|\mathbf{x}\|^3} \mathbf{x} - \frac{\alpha(\mathbf{x})}{m} \|\mathbf{x}'\| \mathbf{x}'$$

Approximation des EDO

Problème de Cauchy pour la rentrée atmosphérique

Exemple : rentrée atmosphérique,

$$\mathbf{x}'' = -\frac{Gm_T}{\|\mathbf{x}\|^3} \mathbf{x} - \frac{\alpha(\mathbf{x})}{m} \|\mathbf{x}'\| \mathbf{x}'$$

on pose $\mathbf{u} := \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} := \begin{bmatrix} \mathbf{x} \\ \mathbf{x}' \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x'_1 \\ x'_2 \end{bmatrix} \in \mathbb{R}^4$ et on a

$$\mathbf{f}(\mathbf{u}) = \begin{bmatrix} u_3 \\ u_4 \\ -\frac{Gm_T}{\|(u_1, u_2)\|^3} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \frac{\alpha(u_1, u_2)}{m} \|(u_3, u_4)\| \begin{bmatrix} u_3 \\ u_4 \end{bmatrix} \end{bmatrix}.$$

Approximation des EDO

Problème de Cauchy pour la rentrée atmosphérique

Exemple : rentrée atmosphérique,

$$\mathbf{x}'' = -\frac{Gm_T}{\|\mathbf{x}\|^3} \mathbf{x} - \frac{\alpha(\mathbf{x})}{m} \|\mathbf{x}'\| \mathbf{x}'$$

on pose $\mathbf{u} := \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} := \begin{bmatrix} \mathbf{x} \\ \mathbf{x}' \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_1' \\ x_2' \end{bmatrix} \in \mathbb{R}^4$ et on a

$$\mathbf{f}(\mathbf{u}) = \begin{bmatrix} u_3 \\ u_4 \\ -\frac{Gm_T}{\|(u_1, u_2)\|^3} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \frac{\alpha(u_1, u_2)}{m} \|(u_3, u_4)\| \begin{bmatrix} u_3 \\ u_4 \end{bmatrix} \end{bmatrix}.$$

Le problème de Cauchy s'écrit

$$\begin{cases} \mathbf{u}'(t) = \mathbf{f}(\mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{v}_0 \end{bmatrix}. \end{cases}$$

Approximation des EDO

Comment approcher numériquement la solution du problème de Cauchy ?

Soit le problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

Approximation des EDO

Comment approcher numériquement la solution du problème de Cauchy ?

Soit le problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

Existence et unicité de $u : [0, T^*[\rightarrow \Omega$ dès que $\mathbf{f} \in \mathcal{C}^1$ [Cauchy-Lipschitz].

Approximation des EDO

Comment approcher numériquement la solution du problème de Cauchy ?

Soit le problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

Existence et unicité de $u : [0, T^*[\rightarrow \Omega$ dès que $\mathbf{f} \in \mathcal{C}^1$ [Cauchy-Lipschitz].

Discretisation du temps : soit $h > 0$, et par $t_n = nh$ avec $t_N = N_h h = T$.

Approximation des EDO

Comment approcher numériquement la solution du problème de Cauchy ?

Soit le problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

Existence et unicité de $u : [0, T^*] \rightarrow \Omega$ dès que $\mathbf{f} \in \mathcal{C}^1$ [Cauchy-Lipschitz].

Discrétisation du temps : soit $h > 0$, et par $t_n = nh$ avec $t_N = N_h h = T$.

Cette suite dépend de h , on la note donc aussi $(t_n^h)_{n=0,1,\dots,N}$.

Approximation des EDO

Comment approcher numériquement la solution du problème de Cauchy ?

Soit le problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

Existence et unicité de $u : [0, T^*] \rightarrow \Omega$ dès que $\mathbf{f} \in \mathcal{C}^1$ [Cauchy-Lipschitz].

Discrétisation du temps : soit $h > 0$, et par $t_n = nh$ avec $t_N = N_h h = T$.

Cette suite dépend de h , on la note donc aussi $(t_n^h)_{n=0,1,\dots,N}$.

Principe d'une méthode numérique : approcher les valeurs de la solution exacte $\mathbf{u}(t_n)$ par une suite $(\mathbf{U}_n)_{n=0,1,\dots,N}$ de Ω .

Approximation des EDO

Comment approcher numériquement la solution du problème de Cauchy ?

Soit le problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

Existence et unicité de $u : [0, T^*] \rightarrow \Omega$ dès que $\mathbf{f} \in \mathcal{C}^1$ [Cauchy-Lipschitz].

Discrétisation du temps : soit $h > 0$, et par $t_n = nh$ avec $t_N = N_h h = T$.

Cette suite dépend de h , on la note donc aussi $(t_n^h)_{n=0,1,\dots,N}$.

Principe d'une méthode numérique : approcher les valeurs de la solution exacte $\mathbf{u}(t_n)$ par une suite $(\mathbf{U}_n)_{n=0,1,\dots,N}$ de Ω .

Vocabulaire : Un **schéma numérique** est une relation de récurrence qui permet de définir la suite $(\mathbf{U}_n)$.

Approximation des EDO

Comment approcher numériquement la solution du problème de Cauchy ?

Soit le problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

Existence et unicité de $\mathbf{u} : [0, T^*] \rightarrow \Omega$ dès que $\mathbf{f} \in \mathcal{C}^1$ [Cauchy-Lipschitz].

Discrétisation du temps : soit $h > 0$, et par $t_n = nh$ avec $t_N = N_h h = T$.

Cette suite dépend de h , on la note donc aussi $(t_n^h)_{n=0,1,\dots,N}$.

Principe d'une méthode numérique : approcher les valeurs de la solution exacte $\mathbf{u}(t_n)$ par une suite $(\mathbf{U}_n)_{n=0,1,\dots,N}$ de Ω .

Vocabulaire : Un **schéma numérique** est une relation de récurrence qui permet de définir la suite $(\mathbf{U}_n)$.

Cette suite est dite solution numérique ou discrète.

Approximation des EDO

La méthode d'Euler

Problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

- ▶ On écrit, pour $n = 0, 1, \dots, N_h - 1$,

$$\mathbf{u}(t_{n+1}) = \mathbf{u}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{u}'(s) ds = \mathbf{u}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{f}(s, \mathbf{u}(s)) ds.$$

- ▶ Approximation de l'intégrale par la méthode des rectangles :

$$\int_{t_n}^{t_{n+1}} \mathbf{f}(s, \mathbf{u}(s)) ds \approx h \mathbf{f}(t_n, \mathbf{u}(t_n)).$$

- ▶ D'où la récurrence (explicite mais inexacte) :

$$\mathbf{u}(t_{n+1}) \approx \mathbf{u}(t_n) + h \mathbf{f}(t_n, \mathbf{u}(t_n)).$$

Approximation des EDO

Méthode d'Euler

On a obtenu :

$$\mathbf{u}(t_{n+1}) \approx \mathbf{u}(t_n) + h \mathbf{f}(t_n, \mathbf{u}(t_n)).$$

Méthode d'Euler. C'est le système de récurrence/condition initiale suivant :

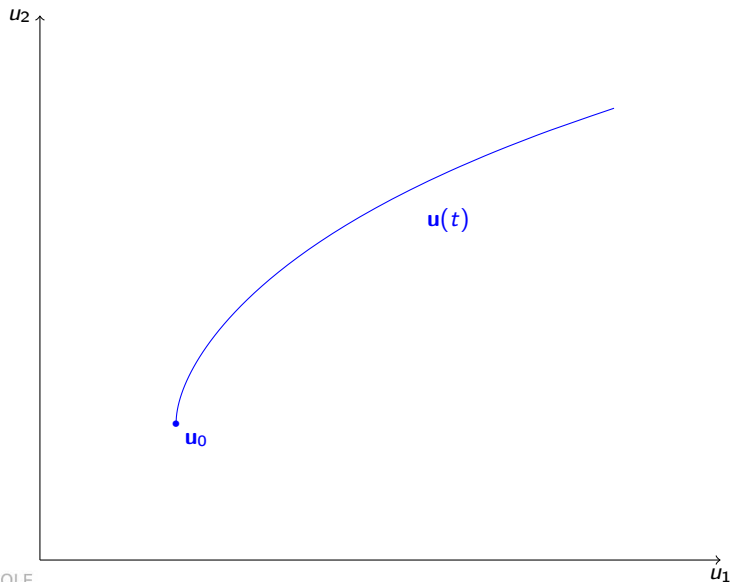
$$(E) : \begin{cases} \mathbf{U}_{n+1} = \mathbf{U}_n + h \mathbf{f}(t_n, \mathbf{U}_n), & n = 0, 1, \dots, N_h - 1 \\ \mathbf{U}_0 = \mathbf{u}_0. \end{cases}$$

Remarques.

- ▶ a priori, on n'a pas $\mathbf{u}(t_n) = U_n$.
- ▶ il n'est pas évident non plus que $\mathbf{u}(t_n)$ approche U_n .

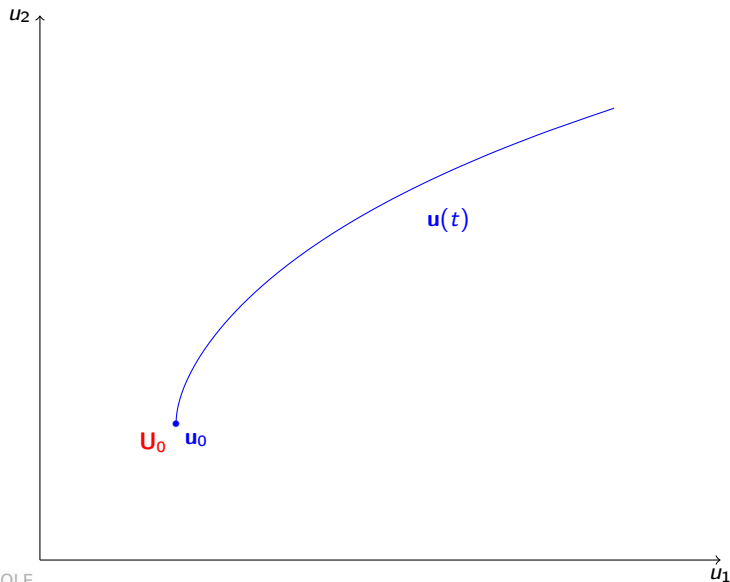
Approximation des EDO

Méthode d'Euler



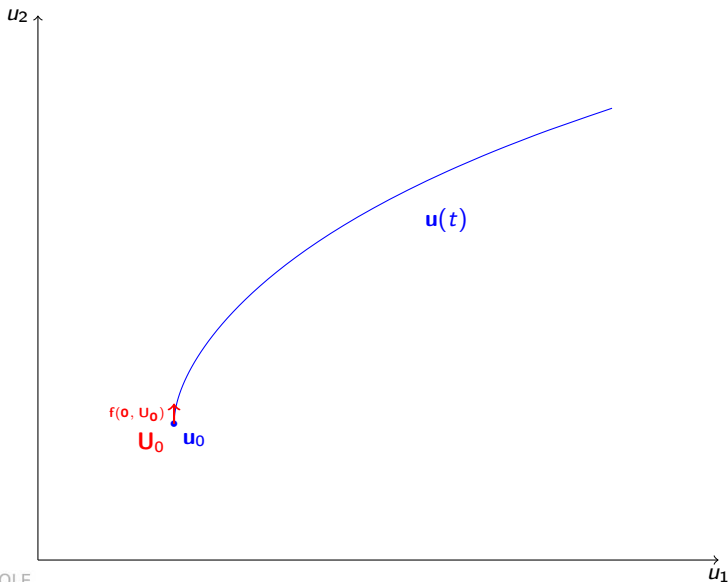
Approximation des EDO

Méthode d'Euler



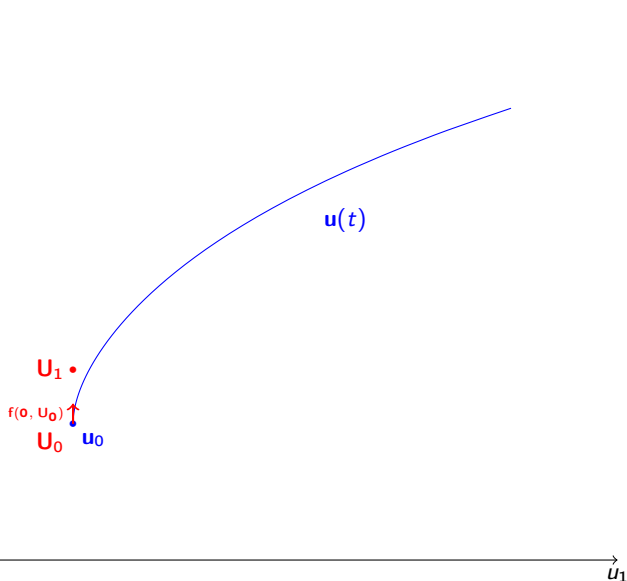
Approximation des EDO

Méthode d'Euler



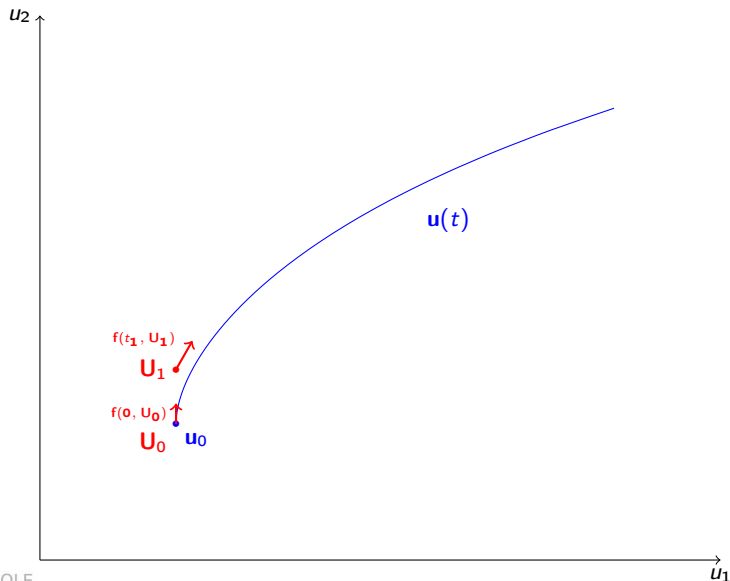
Approximation des EDO

Méthode d'Euler



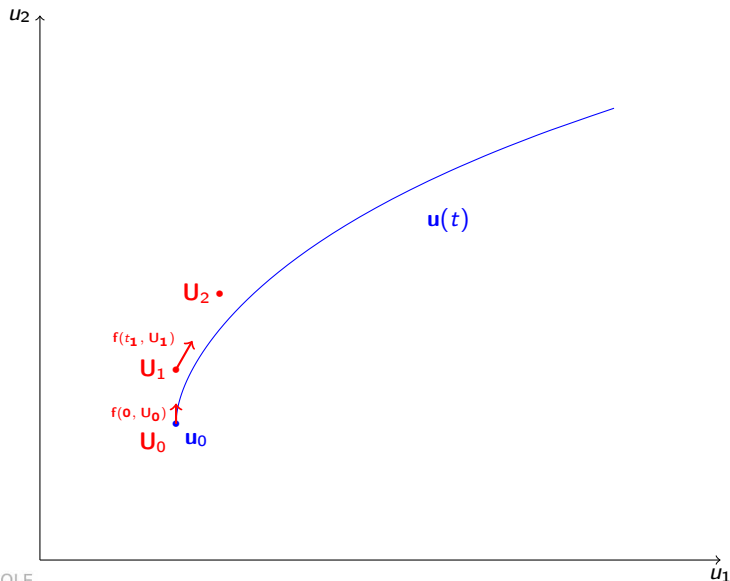
Approximation des EDO

Méthode d'Euler



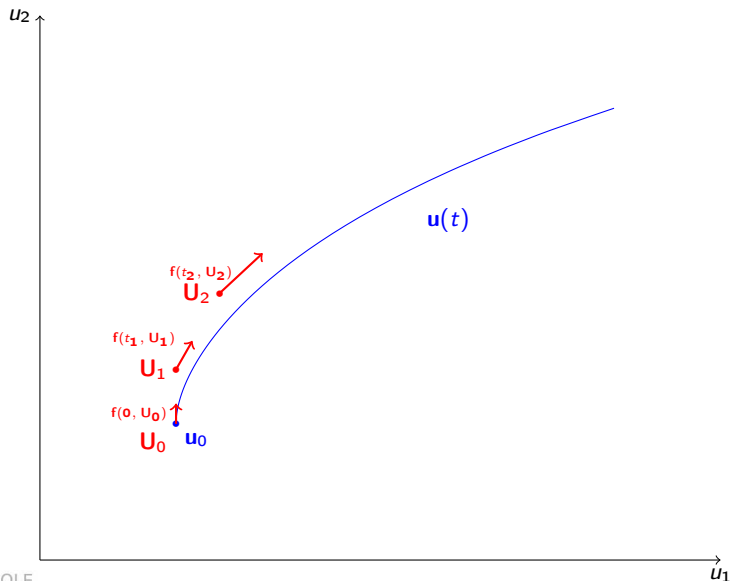
Approximation des EDO

Méthode d'Euler



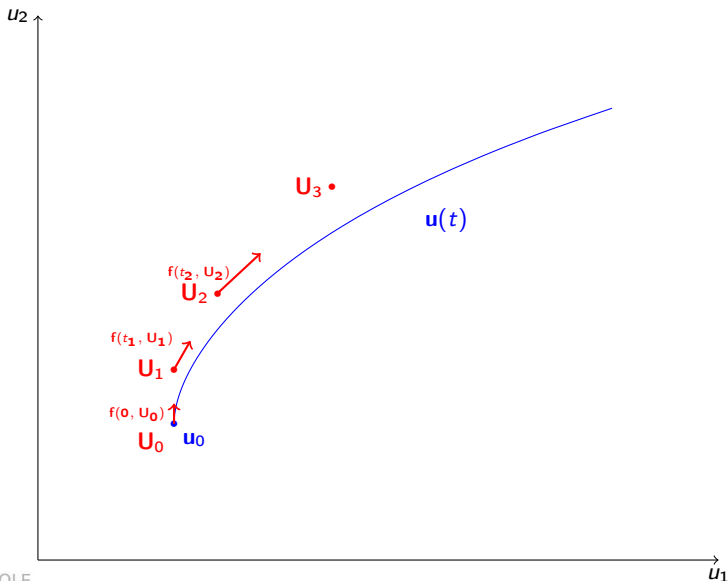
Approximation des EDO

Méthode d'Euler



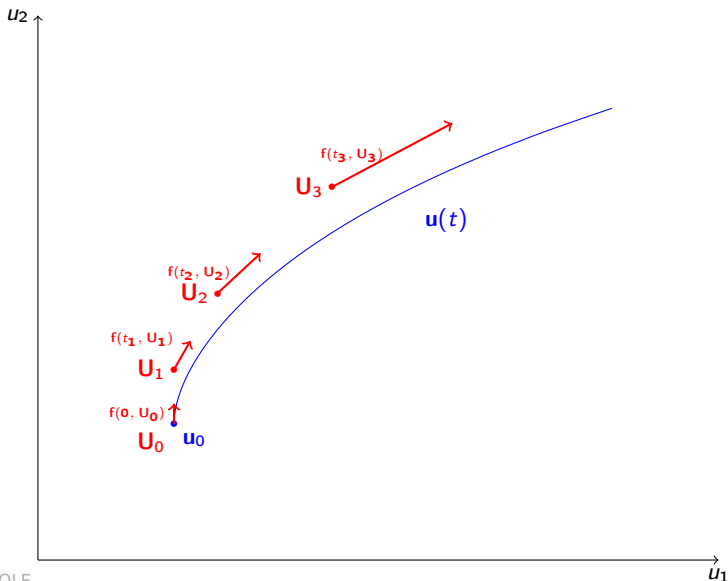
Approximation des EDO

Méthode d'Euler



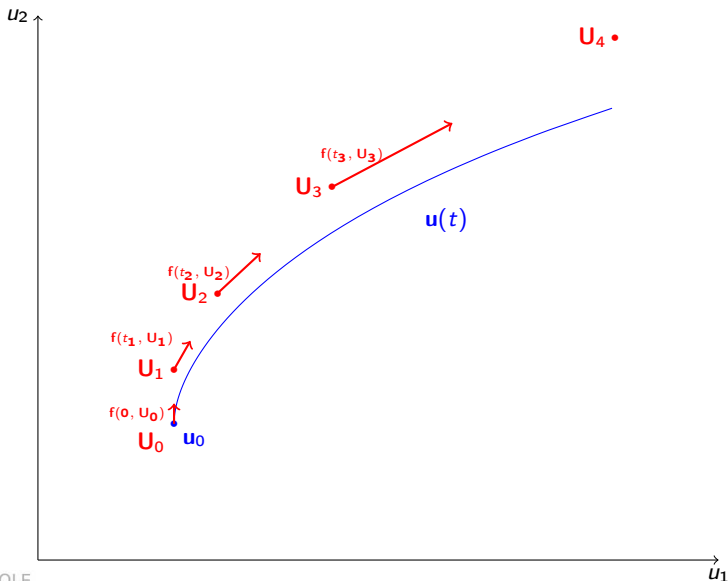
Approximation des EDO

Méthode d'Euler



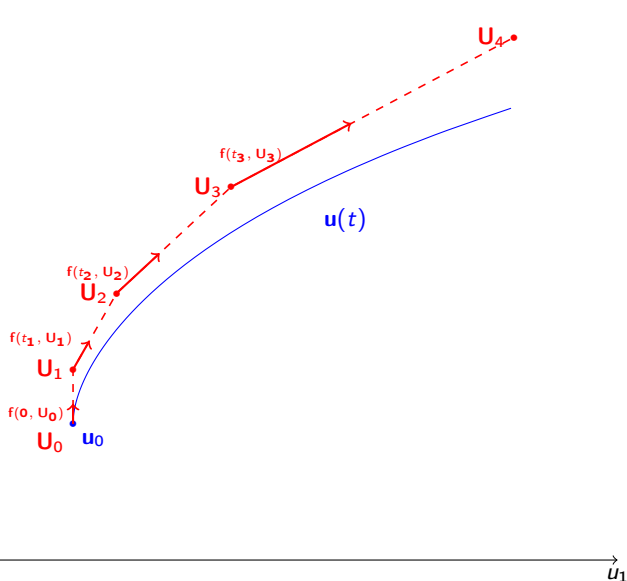
Approximation des EDO

Méthode d'Euler



Approximation des EDO

Méthode d'Euler



Approximation des EDO

Méthode d'Euler : Exemple

Considérons le problème de Cauchy

$$\begin{cases} u'(t) = 2tu(t) & t \in [0, 1], \\ u(0) = 1, \end{cases}$$

dont la solution exacte est $u(t) = e^{t^2}$.

Approximation des EDO

Méthode d'Euler : Exemple

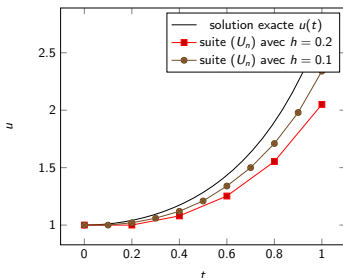
Considérons le problème de Cauchy

$$\begin{cases} u'(t) = 2tu(t) & t \in [0, 1], \\ u(0) = 1, \end{cases}$$

dont la solution exacte est $u(t) = e^{t^2}$.

La méthode d'Euler s'écrit

$$\begin{cases} U_{n+1} = U_n + 2ht_n U_n & t \in [0, 2], \\ U_0 = 1. \end{cases}$$



Approximation des EDO

Méthode d'Euler : Exemple

Considérons le problème de Cauchy

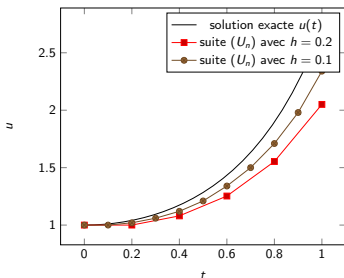
$$\begin{cases} u'(t) = 2tu(t) & t \in [0, 1], \\ u(0) = 1, \end{cases}$$

dont la solution exacte est $u(t) = e^{t^2}$.

La méthode d'Euler s'écrit

$$\begin{cases} U_{n+1} = U_n + 2ht_n U_n & t \in [0, 2], \\ U_0 = 1. \end{cases}$$

- La méthode d'Euler n'est pas exacte,



Approximation des EDO

Méthode d'Euler : Exemple

Considérons le problème de Cauchy

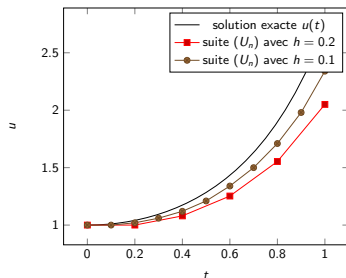
$$\begin{cases} u'(t) = 2tu(t) & t \in [0, 1], \\ u(0) = 1, \end{cases}$$

dont la solution exacte est $u(t) = e^{t^2}$.

La méthode d'Euler s'écrit

$$\begin{cases} U_{n+1} = U_n + 2ht_n U_n & t \in [0, 2], \\ U_0 = 1. \end{cases}$$

- ▶ La méthode d'Euler n'est pas exacte,
- ▶ l'erreur augmente avec le temps,



Approximation des EDO

Méthode d'Euler : Exemple

Considérons le problème de Cauchy

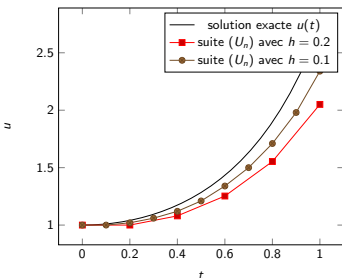
$$\begin{cases} u'(t) = 2tu(t) & t \in [0, 1], \\ u(0) = 1, \end{cases}$$

dont la solution exacte est $u(t) = e^{t^2}$.

La méthode d'Euler s'écrit

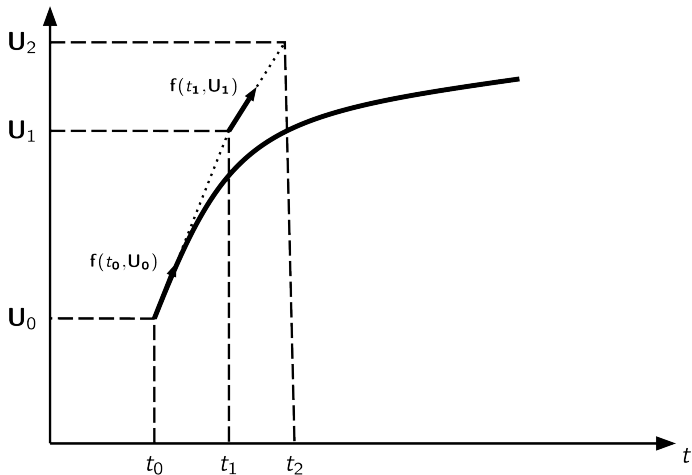
$$\begin{cases} U_{n+1} = U_n + 2ht_n U_n & t \in [0, 2], \\ U_0 = 1. \end{cases}$$

- ▶ La méthode d'Euler n'est pas exacte,
- ▶ l'erreur augmente avec le temps,
- ▶ l'erreur diminue si h diminue.



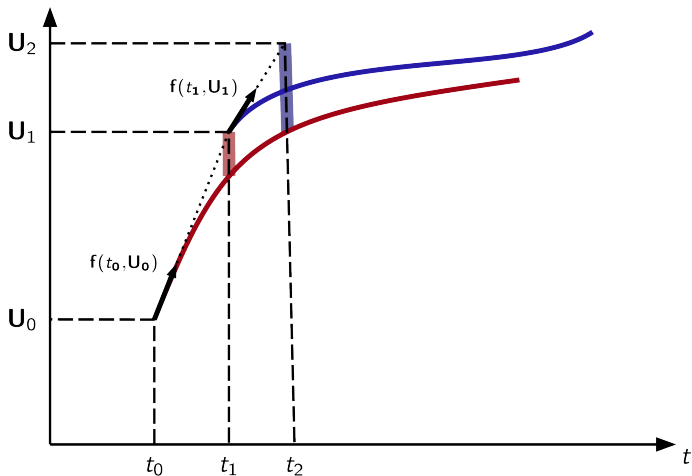
Approximation des EDO

Analyse d'erreur



Approximation des EDO

Analyse d'erreur



Approximation des EDO

Consistance, stabilité, convergence

- **Convergence** : L'erreur globale du schéma tend vers 0

$$\max_{0 \leq n \leq N_h} \|\mathbf{U}_n - \mathbf{u}(t_n)\| \xrightarrow[h \rightarrow 0]{?} 0.$$

Approximation des EDO

Consistance, stabilité, convergence

- **Convergence** : L'erreur globale du schéma tend vers 0

$$\max_{0 \leq n \leq N_h} \|\mathbf{U}_n - \mathbf{u}(t_n)\| \xrightarrow[h \rightarrow 0]{?} 0.$$

- **Analyse en 2 temps** :

- **Consistance** : analyse de l'erreur locale (commise à chaque pas).

Approximation des EDO

Consistance, stabilité, convergence

- **Convergence** : L'erreur globale du schéma tend vers 0

$$\max_{0 \leq n \leq N_h} \|\mathbf{U}_n - \mathbf{u}(t_n)\| \xrightarrow[h \rightarrow 0]{?} 0.$$

- **Analyse en 2 temps** :
 - **Consistance** : analyse de l'erreur locale (commise à chaque pas).
 - **Stabilité** : maîtrise de la propagation des erreurs locales.

Approximation des EDO

Consistance, stabilité, convergence

- **Convergence** : L'erreur globale du schéma tend vers 0

$$\max_{0 \leq n \leq N_h} \|\mathbf{U}_n - \mathbf{u}(t_n)\| \xrightarrow[h \rightarrow 0]{?} 0.$$

- **Analyse en 2 temps** :

- **Consistance** : analyse de l'erreur locale (commise à chaque pas).
- **Stabilité** : maîtrise de la propagation des erreurs locales.

Théorème de Lax.

Consistance et Stabilité $\Rightarrow$ Convergence.

Approximation des EDO

Consistance

On note $\mathbf{u}$ la solution exacte du problème de Cauchy.

Définition. L'erreur de consistance du schéma d'Euler est la suite

$$\varepsilon_n^h := \frac{\mathbf{u}(t_{n+1}) - \mathbf{u}(t_n)}{h} - \mathbf{f}(t_n, \mathbf{u}(t_n)), \quad 0 \leq n \leq N_h - 1.$$

On note $\mathbf{u}$ la solution exacte du problème de Cauchy.

Définition. L'erreur de consistance du schéma d'Euler est la suite

$$\varepsilon_n^h := \frac{\mathbf{u}(t_{n+1}) - \mathbf{u}(t_n)}{h} - \mathbf{f}(t_n, \mathbf{u}(t_n)), \quad 0 \leq n \leq N_h - 1.$$

Remarque : Lien avec l'approximation de l'intégration de $\mathbf{u}'$ sur $[t_n, t_n + h]$ par la méthode des rectangles à gauche :

$$\varepsilon_n^h := \int_0^1 \mathbf{u}'(t_n + hs) ds - \mathbf{u}'(t_n).$$

Rappel : u désigne la solution exacte du problème de Cauchy.

Définition. L'erreur de consistance du schéma d'Euler est la suite

$$\varepsilon_n^h := \frac{u(t_{n+1}) - u(t_n)}{h} - f(t_n, u(t_n)), \quad 0 \leq n \leq N_h - 1.$$

Définition. Le schéma est consistant si l'erreur de consistance converge vers 0 :

$$\max_{0 \leq n \leq N_h - 1} \|\varepsilon_n^h\| \rightarrow 0 \quad \text{quand} \quad h \rightarrow 0.$$

Si de plus il existe $k > 0$ tel que

$$\max_{0 \leq n \leq N_h - 1} \|\varepsilon_n^h\| = \mathcal{O}(h^k),$$

le schéma est dit consistant d'ordre k .

Approximation des EDO

Consistance pour Euler

Proposition. Si la solution exacte u est de classe $\mathcal{C}^2$, alors la méthode d'Euler est consistante d'ordre 1.

Approximation des EDO

Consistance pour Euler

Proposition. Si la solution exacte $\mathbf{u}$ est de classe $\mathcal{C}^2$, alors la méthode d'Euler est consistante d'ordre 1.

Preuve : Par Taylor-Lagrange,

$$\mathbf{u}(t_{n+1}) = \mathbf{u}(t_n + h) = \mathbf{u}(t_n) + h\mathbf{u}'(t_n) + h^2\mathbf{R}_n(h),$$

avec $\mathbf{R}_n(h)$ bornée. Plus précisément, $\|\mathbf{R}_n(h)\| \leq C := \frac{1}{2} \sup_{t \in [0, T]} \|\mathbf{u}''(t)\|$

$$\varepsilon_n^h := \frac{\mathbf{u}(t_{n+1}) - \mathbf{u}(t_n)}{h} - \mathbf{f}(t_n, \mathbf{u}(t_n)),$$

$$\varepsilon_n^h := \mathbf{u}'(t_n) + h\mathbf{R}_n(h) - \mathbf{f}(t_n, \mathbf{u}(t_n)),$$

$$\varepsilon_n^h := h\mathbf{R}_n(h),$$

Ainsi

$$\max_{0 \leq n \leq N_h - 1} \|\varepsilon_n^h\| \leq Ch.$$



Approximation des EDO

Stabilité pour Euler

On considère le schéma d'Euler

$$(E) : \begin{cases} \mathbf{U}_{n+1} = \mathbf{U}_n + h \mathbf{f}(t_n, \mathbf{U}_n), & n \geqslant 0, \\ \mathbf{U}_0 = \mathbf{u}_0. \end{cases}$$

et on le perturbe en ajoutant une suite d'erreurs μ_n

Approximation des EDO

Stabilité pour Euler

On considère le schéma d'Euler

$$(E) : \begin{cases} \mathbf{U}_{n+1} = \mathbf{U}_n + h \mathbf{f}(t_n, \mathbf{U}_n), & n \geqslant 0, \\ \mathbf{U}_0 = \mathbf{u}_0. \end{cases}$$

et on le perturbe en ajoutant une suite d'erreurs μ_n

$$(E_{\text{pert}}) : \begin{cases} \mathbf{V}_{n+1} = \mathbf{V}_n + h \mathbf{f}(t_n, \mathbf{V}_n) + \mu_n, & n \geqslant 0, \\ \mathbf{V}_0 = \mathbf{u}_0. \end{cases}$$

Approximation des EDO

Stabilité pour Euler

On considère le schéma d'Euler

$$(E) : \begin{cases} \mathbf{U}_{n+1} = \mathbf{U}_n + h \mathbf{f}(t_n, \mathbf{U}_n), & n \geqslant 0, \\ \mathbf{U}_0 = \mathbf{u}_0. \end{cases}$$

et on le perturbe en ajoutant une suite d'erreurs μ_n

$$(E_{\text{pert}}) : \begin{cases} \mathbf{V}_{n+1} = \mathbf{V}_n + h \mathbf{f}(t_n, \mathbf{V}_n) + \mu_n, & n \geqslant 0, \\ \mathbf{V}_0 = \mathbf{u}_0. \end{cases}$$

Définition. Le schéma est stable s'il existe une constante $C > 0$ telle que

$$\max_{0 \leqslant n \leqslant N_h} \|\mathbf{V}_n - \mathbf{U}_n\| \leqslant C \sum_{0 \leqslant n \leqslant N_h - 1} \|\mu_n\|.$$

Approximation des EDO

Stabilité pour Euler

Proposition. Si f est L -lipschitzienne, la méthode d'Euler est stable et l'on a

$$\max_{0 \leq n \leq N_h} |V_n - U_n| \leq e^{LT} \sum_{0 \leq n \leq N_h - 1} \|\mu_n\|.$$

Proposition. Si $\mathbf{f}$ est L -lipschitzienne, la méthode d'Euler est stable et l'on a

$$\max_{0 \leq n \leq N_h} |\mathbf{V}_n - \mathbf{U}_n| \leq e^{LT} \sum_{0 \leq n \leq N_h - 1} \|\mu_n\|.$$

Preuve : Par soustraction

$$\begin{aligned}\mathbf{V}_{n+1} - \mathbf{U}_{n+1} &= \mathbf{V}_n - \mathbf{U}_n + h(\mathbf{f}(t_n, \mathbf{V}_n) - \mathbf{f}(t_n, \mathbf{U}_n)) + \mu_n, \\ \|\mathbf{V}_{n+1} - \mathbf{U}_{n+1}\| &\leq \|\mathbf{V}_n - \mathbf{U}_n\| + h \|\mathbf{f}(t_n, \mathbf{V}_n) - \mathbf{f}(t_n, \mathbf{U}_n)\| + \|\mu_n\|, \\ \|\mathbf{V}_{n+1} - \mathbf{U}_{n+1}\| &\leq (1 + hL) \|\mathbf{V}_n - \mathbf{U}_n\| + \|\mu_n\|.\end{aligned}$$

Posons $e_n = \|\mathbf{V}_n - \mathbf{U}_n\|$, et divisons par $(1 + hL)^{n+1}$;

$$\begin{aligned}\frac{e_{n+1}}{(1 + hL)^{n+1}} &\leq \frac{e_n}{(1 + hL)^n} + \frac{\|\mu_n\|}{(1 + hL)^{n+1}}, \\ \frac{e_{n+1}}{(1 + hL)^{n+1}} &\leq \frac{e_n}{(1 + hL)^n} + \|\mu_n\|,\end{aligned}$$

Approximation des EDO

Stabilité pour Euler

Preuve (suite) : On a

$$\frac{e_{n+1}}{(1 + hL)^{n+1}} - \frac{e_n}{(1 + hL)^n} \leq \|\mu_n\| ,$$

En sommant, on obtient

$$\frac{e_n}{(1 + hL)^n} \leq \sum_{k=0}^{n-1} \|\mu_k\| , \quad \forall n \in \mathbb{N},$$

$$e_n \leq (1 + hL)^n \sum_{k=0}^{n-1} \|\mu_k\| , \quad \forall n \in \mathbb{N}.$$

Or pour tout $n \leq N_h$,

$$(1 + hL)^n \leq e^{Ltn} \leq e^{N_h hL} = e^{LT}.$$

Ainsi,

$$\|\mathbf{v}_n - \mathbf{u}_n\| \leq e^{LT} \sum_{k=0}^{n-1} \|\mu_k\| , \quad \forall n = 0, 1, \dots, N_h.$$

Approximation des EDO

Convergence pour Euler

Définition. Le schéma converge si

$$\max_{0 \leq n \leq N_h} |\mathbf{U}_n - \mathbf{u}(t_n)| \longrightarrow 0 \quad \text{quand} \quad h \rightarrow 0.$$

Approximation des EDO

Convergence pour Euler

Définition. Le schéma converge si

$$\max_{0 \leq n \leq N_h} |\mathbf{U}_n - \mathbf{u}(t_n)| \longrightarrow 0 \quad \text{quand} \quad h \rightarrow 0.$$

Théorème de Lax. Un schéma consistant et stable est convergent.

Approximation des EDO

Convergence pour Euler

Définition. Le schéma converge si

$$\max_{0 \leq n \leq N_h} |\mathbf{U}_n - \mathbf{u}(t_n)| \longrightarrow 0 \quad \text{quand} \quad h \rightarrow 0.$$

Théorème de Lax. Un schéma consistant d'ordre k et stable est convergent d'ordre k .

Preuve (Pour Euler) : Par définition,

$$\varepsilon_n^h := \frac{\mathbf{u}(t_{n+1}) - \mathbf{u}(t_n)}{h} - \mathbf{f}(t_n, \mathbf{u}(t_n)).$$

Donc la suite de valeurs exactes est solution du schéma perturbé :

$$\begin{cases} \mathbf{u}(t_{n+1}) = \mathbf{u}(t_n) + h\mathbf{f}(t_n, \mathbf{u}(t_n)) + h\varepsilon_n^h, \\ \mathbf{u}(t_0) = \mathbf{u}_0. \end{cases}$$

Approximation des EDO

Convergence pour Euler

Preuve (suite) : On utilise la stabilité du schéma :

$$\max_{0 \leq n \leq N_h} |\mathbf{U}_n - \mathbf{u}(t_n)| \leq e^{LT} h \sum_{0 \leq n \leq N_h-1} \left\| \varepsilon_n^h \right\|.$$

Or,

$$\sum_{0 \leq n \leq N_h-1} \left\| \varepsilon_n^h \right\| \leq N_h \max_{0 \leq n \leq N_h-1} \left\| \varepsilon_n^h \right\|,$$

Ainsi,

$$\max_{0 \leq n \leq N_h} |\mathbf{U}_n - \mathbf{u}(t_n)| \leq Te^{LT} \max_{0 \leq n \leq N_h-1} \left\| \varepsilon_n^h \right\|.$$

Or, par consistance d'ordre 1,

$$\max_{0 \leq n \leq N_h-1} \left\| \varepsilon_n^h \right\| \leq Ch.$$

Finalement,

$$\max_{0 \leq n \leq N_h} |\mathbf{U}_n - \mathbf{u}(t_n)| \leq CTe^{LT} h.$$

□

Approximation des EDO

Convergence pour Euler

Proposition. Le schéma d'Euler est convergent d'ordre 1 :

$$\max_{0 \leq n \leq N_h} |\mathbf{U}_n - \mathbf{u}(t_n)| \leq CTe^{LT} h,$$

avec $C = \frac{1}{2} \sup_{t \in [0, T]} \|\mathbf{u}''(t)\|$.

Approximation des EDO

Convergence pour Euler

Proposition. Le schéma d'Euler est convergent d'ordre 1 :

$$\max_{0 \leq n \leq N_h} |\mathbf{U}_n - \mathbf{u}(t_n)| \leq C T e^{LT} h,$$

avec $C = \frac{1}{2} \sup_{t \in [0, T]} \|\mathbf{u}''(t)\|$.

Remarques :

- ▶ On a besoin que $\mathbf{u} \in \mathcal{C}^2([0, T])$.
- ▶ La constante $T e^{LT}$ augmente exponentiellement
 - ▶ avec T (problèmes en temps long) ,
 - ▶ avec L (problèmes « raides »).

Définition. Un schéma à un pas (constant) pour approcher la solution du problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

est une relation de récurrence de la forme

$$\begin{cases} \mathbf{U}_{n+1} = \mathbf{U}_n + h\mathbf{G}_h(t_n, \mathbf{U}_n, \mathbf{U}_{n+1}), & n = 0, 1, \dots, N_h - 1, \\ \mathbf{U}_0 = \mathbf{u}_0. \end{cases}$$

où $h > 0$ est le pas et $\mathbf{G}_h$ est une fonction qui définit le schéma.

Approximation des EDO

Schémas classiques

Définition. Un schéma à un pas (constant) pour approcher la solution du problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

est une relation de récurrence de la forme

$$\begin{cases} \mathbf{U}_{n+1} = \mathbf{U}_n + h\mathbf{G}_h(t_n, \mathbf{U}_n, \mathbf{U}_{n+1}), & n = 0, 1, \dots, N_h - 1, \\ \mathbf{U}_0 = \mathbf{u}_0. \end{cases}$$

où $h > 0$ est le pas et $\mathbf{G}_h$ est une fonction qui définit le schéma.

Vocabulaire : Si $\mathbf{G}_h$ dépend effectivement de $\mathbf{U}_{n+1}$ le schéma est dit implicite, sinon il est explicite.

Définition. Un schéma à un pas (constant) pour approcher la solution du problème de Cauchy

$$(\mathcal{C}) : \begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)) & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

est une relation de récurrence de la forme

$$\begin{cases} \mathbf{U}_{n+1} = \mathbf{U}_n + h\mathbf{G}_h(t_n, \mathbf{U}_n, \mathbf{U}_{n+1}), & n = 0, 1, \dots, N_h - 1, \\ \mathbf{U}_0 = \mathbf{u}_0. \end{cases}$$

où $h > 0$ est le pas et $\mathbf{G}_h$ est une fonction qui définit le schéma.

Vocabulaire : Si $\mathbf{G}_h$ dépend effectivement de $\mathbf{U}_{n+1}$ le schéma est dit implicite, sinon il est explicite.

Exemple : Le schéma d'Euler $\mathbf{U}_{n+1} = \mathbf{U}_n + h\mathbf{f}(t_n, \mathbf{U}_n)$ correspond à

$$\mathbf{G}_h(t_n, \mathbf{U}_n, \mathbf{U}_{n+1}) = \mathbf{f}(t_n, \mathbf{U}_n).$$

Rappel : $\mathbf{u}$ désigne la solution exacte du problème de Cauchy.

Définition. L'erreur de consistance d'un schéma général à un pas est

$$\varepsilon_n^h := \frac{\mathbf{u}(t_{n+1}) - \mathbf{u}(t_n)}{h} - G_h(t_n, \mathbf{u}(t_n), \mathbf{u}(t_{n+1})), \quad 0 \leq n \leq N_h.$$

Remarque : Lien avec l'erreur de l'approximation de l'intégration de $\mathbf{u}'$ sur $[t_n, t_n + h]$:

$$\varepsilon_n^h := \int_0^1 \mathbf{u}'(t_n + hs) ds - G_h(t_n, \mathbf{u}(t_n), \mathbf{u}(t_{n+1})).$$

On considère un schéma général

$$(E) : \begin{cases} \mathbf{U}_{n+1} = \mathbf{U}_n + h\mathbf{G}_h(t_n, \mathbf{U}_n, \mathbf{U}_{n+1}), & n = 0, 1, \dots, N_h - 1, \\ \mathbf{U}_0 = \mathbf{u}_0. \end{cases}$$

et on le perturbe en ajoutant une suite d'erreurs μ_n :

$$(E_{\text{pert}}) : \begin{cases} \mathbf{V}_{n+1} = \mathbf{V}_n + h\mathbf{G}_h(t_n, \mathbf{V}_n, \mathbf{V}_{n+1}) + \mu_n, & n = 0, 1, \dots, N_h - 1, \\ \mathbf{V}_0 = \mathbf{u}_0. \end{cases}$$

Définition. Le schéma est stable s'il existe $C > 0$ t.q.

$$\max_{0 \leq n \leq N_h} \|\mathbf{V}_n - \mathbf{U}_n\| \leq C \sum_{0 \leq n \leq N_h - 1} \|\mu_n\|.$$

Approximation des EDO

Euler, Euler implicite

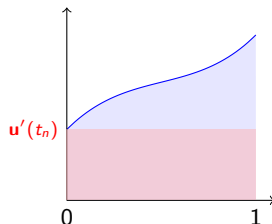
$$\int_{t_n}^{t_{n+1}} u'(t) dt = h \int_0^1 u'(t_n + hs) ds$$

Rectangle à gauche :

$$\begin{aligned} I(h) &= \int_0^1 u'(t_n + hs) ds \\ &\approx u'(t_n) = f(t_n, u(t_n)) \end{aligned}$$

conduit à poser

Euler : $U_{n+1} = U_n + hf(t_n, U_n)$.



Approximation des EDO

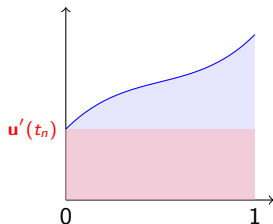
Euler, Euler implicite

Rectangle à gauche :

$$\begin{aligned} I(h) &= \int_0^1 \mathbf{u}'(t_n + hs) ds \\ &\approx \mathbf{u}'(t_n) = \mathbf{f}(t_n, \mathbf{u}(t_n)) \end{aligned}$$

conduit à poser

$$\mathbf{Euler} : \mathbf{U}_{n+1} = \mathbf{U}_n + h\mathbf{f}(t_n, \mathbf{U}_n).$$

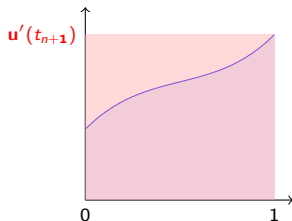


Rectangle à droite :

$$\begin{aligned} I(h) &= \int_0^1 \mathbf{u}'(t_n + hs) ds \\ &\approx \mathbf{u}'(t_{n+1}) = \mathbf{f}(t_{n+1}, \mathbf{u}(t_{n+1})) \end{aligned}$$

conduit à poser

$$\mathbf{Euler} \text{ (implicite)} : \mathbf{U}_{n+1} = \mathbf{U}_n + h\mathbf{f}(t_{n+1}, \mathbf{U}_{n+1})$$



Approximation des EDO

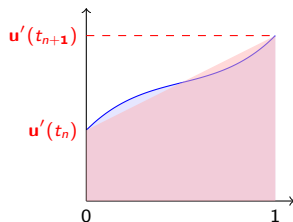
Crank-Nicolson, Heun

Trapèze :

$$I(h) \approx \frac{\mathbf{u}'(t_n) + \mathbf{u}'(t_{n+1})}{2}$$

Crank-Nicolson (implicite)

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \frac{h}{2} (\mathbf{f}(t_n, \mathbf{U}_n) + \mathbf{f}(t_{n+1}, \mathbf{U}_{n+1}))$$



Approximation des EDO

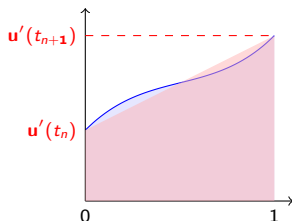
Crank-Nicolson, Heun

Trapèze :

$$I(h) \approx \frac{u'(t_n) + u'(t_{n+1})}{2}$$

Crank-Nicolson (implicite)

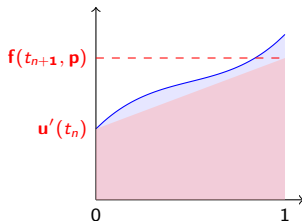
$$U_{n+1} = U_n + \frac{h}{2} (f(t_n, U_n) + f(t_{n+1}, U_{n+1}))$$



Trapèze approché : Le vecteur $\mathbf{p}$ approche $u(t_{n+1})$ par Euler :

Heun (explicite)

$$\begin{cases} \mathbf{p} = \mathbf{U}_n + h\mathbf{f}(t_n, \mathbf{U}_n). \\ \mathbf{U}_{n+1} = \mathbf{U}_n + \frac{h}{2} (\mathbf{f}(t_n, \mathbf{U}_n) + \mathbf{f}(t_{n+1}, \mathbf{p})) \end{cases}$$



Approximation des EDO

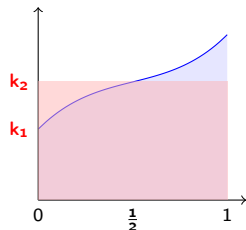
Runge-Kutta

Point milieu approché : **RK-2** (explicite)

$$k_1 = f(t_n, U_n).$$

$$k_2 = f\left(t_n + \frac{h}{2}, U_n + \frac{h}{2}k_1\right).$$

$$U_{n+1} = U_n + h k_2.$$

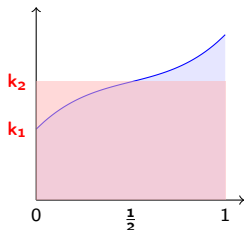


Approximation des EDO

Runge-Kutta

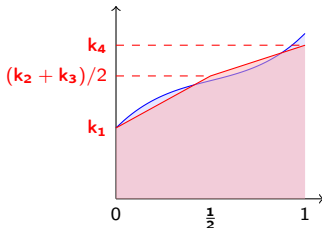
Point milieu approché : **RK-2** (explicite)

$$\left| \begin{array}{l} k_1 = f(t_n, U_n). \\ k_2 = f\left(t_n + \frac{h}{2}, U_n + \frac{h}{2}k_1\right). \\ U_{n+1} = U_n + hk_2. \end{array} \right.$$



Méthode de Simpson approchée : **RK-4** (explicite)

$$\left| \begin{array}{l} k_1 = f(t_n, U_n). \\ k_2 = f\left(t_n + \frac{h}{2}, U_n + \frac{h}{2}k_1\right). \\ k_3 = f\left(t_n + \frac{h}{2}, U_n + \frac{h}{2}k_2\right). \\ k_4 = f(t_n + h, U_n + hk_3). \\ U_{n+1} = U_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{array} \right.$$



Approximation des EDO

Pourquoi les méthodes implicites ?

On considère le problème de Cauchy :

$$\begin{cases} y'(t) = -1000y(t), \\ y(0) = 1. \end{cases}$$

Approximation des EDO

Pourquoi les méthodes implicites ?

On considère le problème de Cauchy :

$$\begin{cases} y'(t) = -1000y(t), \\ y(0) = 1. \end{cases}$$

- ▶ Problème **raide** : $f(y) = -1000y$ est 1000-lipschitzienne.
- ▶ Solution exacte : $y(t) = e^{-1000t}$ (positive, converge vers 0 en $+\infty$).
- ▶ Constante d'erreur pour Euler proportionnelle à Te^{1000T} .

Approximation des EDO

Pourquoi les méthodes implicites ?

On considère le problème de Cauchy :

$$\begin{cases} y'(t) = -1000y(t), \\ y(0) = 1. \end{cases}$$

- ▶ Problème **raide** : $f(y) = -1000y$ est 1000-lipschitzienne.
- ▶ Solution exacte : $y(t) = e^{-1000t}$ (positive, converge vers 0 en $+\infty$).
- ▶ Constante d'erreur pour Euler proportionnelle à Te^{1000T} .

Méthode d'Euler :

$$\begin{cases} Y_{n+1} = Y_n - 1000hY_n, \\ Y_0 = 1, \end{cases}$$

soit $Y_n = (1 - 1000h)^n$.

Approximation des EDO

Pourquoi les méthodes implicites ?

On considère le problème de Cauchy :

$$\begin{cases} y'(t) = -1000y(t), \\ y(0) = 1. \end{cases}$$

- ▶ Problème **raide** : $f(y) = -1000y$ est 1000-lipschitzienne.
- ▶ Solution exacte : $y(t) = e^{-1000t}$ (positive, converge vers 0 en $+\infty$).
- ▶ Constante d'erreur pour Euler proportionnelle à Te^{1000T} .

Méthode d'Euler :

$$\begin{cases} Y_{n+1} = Y_n - 1000hY_n, \\ Y_0 = 1, \end{cases}$$

soit $Y_n = (1 - 1000h)^n$.

La solution numérique reste positive ssi $h \leq 10^{-3}$

Approximation des EDO

Pourquoi les méthodes implicites ?

On considère le problème de Cauchy :

$$\begin{cases} y'(t) = -1000y(t), \\ y(0) = 1. \end{cases}$$

- ▶ Problème **raide** : $f(y) = -1000y$ est 1000-lipschitzienne.
- ▶ Solution exacte : $y(t) = e^{-1000t}$ (positive, converge vers 0 en $+\infty$).
- ▶ Constante d'erreur pour Euler proportionnelle à Te^{1000T} .

Méthode d'Euler :

$$\begin{cases} Y_{n+1} = Y_n - 1000hY_n, \\ Y_0 = 1, \end{cases}$$

soit $Y_n = (1 - 1000h)^n$.

La solution numérique tend vers 0 ssi $h < 2 \cdot 10^{-3}$

Approximation des EDO

Stabilité absolue/asymptotique

Euler implicite :

$$\begin{cases} Y_{n+1} = Y_n - 1000hY_{n+1}, \\ Y_0 = 1, \end{cases}$$

soit $Y_n = (1 + 1000h)^{-n}$.



Approximation des EDO

Stabilité absolue/asymptotique

Euler implicite :

$$\begin{cases} Y_{n+1} = Y_n - 1000hY_{n+1}, \\ Y_0 = 1, \end{cases}$$

soit $Y_n = (1 + 1000h)^{-n}$.

La solution numérique **reste positive** et **converge vers 0** pour tout $h > 0$.

Pas de contrainte sur h .

Approximation des EDO

Stabilité absolue/asymptotique

Euler implicite :

$$\begin{cases} Y_{n+1} = Y_n - 1000hY_{n+1}, \\ Y_0 = 1, \end{cases}$$

soit $Y_n = (1 + 1000h)^{-n}$.

La solution numérique **reste positive** et **converge vers 0** pour tout $h > 0$.

Pas de contrainte sur h .

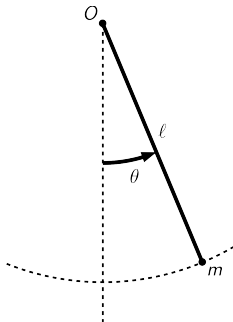
- ▶ Méthode implicite plus coûteuse (pour les problèmes non-linéaires).
- ▶ Mais utile pour les problèmes **raides** en **temps long**.

Approximation des EDO

Conservation d'énergie

Modélisation du pendule sans frottement :

$$\begin{cases} \theta''(t) = -\frac{g}{\ell} \sin(\theta(t)) \\ \theta(0), \theta'(0) \text{ donnés.} \end{cases}$$



Énergie du système :

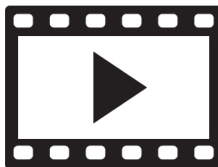
$$E(t) = \frac{1}{2} \theta'(t)^2 - \frac{g}{\ell} \cos(\theta(t)).$$

■ Résultat : $E'(t) = 0$.

Question : est-ce respecté par les schémas ?

Approximation des EDO

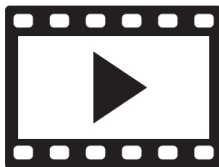
Conservation d'énergie



Méthode d'Euler explicite (ordre 1) pour $h = 0.1$.

Approximation des EDO

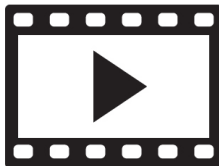
Conservation d'énergie



Méthode d'Euler implicite (ordre 1) pour $h = 0.1$.

Approximation des EDO

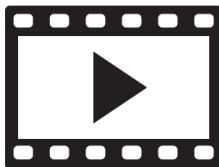
Conservation d'énergie



Méthode RK4 (ordre 4) pour $h = 0.1$.

Approximation des EDO

Conservation d'énergie



Méthode d'Euler symplectique (ordre 1) pour $h = 0.1$.