

Calcul algébrique approché : une introduction

Stef GRAILLAT

ENS de Cachan, antenne de Bretagne
Laboratoire MANO — Université de Perpignan

`grailat@univ-perp.fr`

`http://www.univ-perp.fr/~grailat`

Journée 4^e année, 24 janvier 2003



Présentation du laboratoire

Laboratoire de Modélisation, Analyse Non linéaire et Optimisation (équipe d'accueil)

Présentation du laboratoire

Laboratoire de Modélisation, Analyse Non linéaire et Optimisation (équipe d'accueil)

Effectif :

- 8 enseignant-chercheurs
- 3 doctorants

Présentation du laboratoire

Laboratoire de Modélisation, Analyse Non linéaire et Optimisation (équipe d'accueil)

Effectif :

- 8 enseignant-chercheurs
- 3 doctorants

Thèmes de recherche :

- Analyse convexe, optimisation de problèmes non linéaires
- Topologie, analyse multivoque et théorie minmax
- Analyse numérique pour la mécanique des fluides
⇒ Informatique numérique, qualité numérique de logiciels scientifiques

Nouvelle équipe en Informatique

Domaine de recherche : précision finie et méthodes automatiques

Deux objectifs :

- Comprendre l'influence de la précision finie sur la qualité numérique du logiciel scientifique pour contrôler et limiter ses effets néfastes,
 - résultat imprécis,
 - instabilité numérique.
- Contribuer au développement d'approches automatiques.

Précision finie et méthodes automatiques

Comprendre l'influence de la précision finie sur la qualité numérique du logiciel scientifique pour contrôler et limiter ses effets néfastes :

- Contrôler les effets de la précision finie :
 - (Q1) Comment mesurer la difficulté du problème à résoudre ?
 - (Q2) Comment apprécier la fiabilité de l'algorithme de résolution ?
 - (Q3) Comment estimer la précision de la solution calculée ?
- Limiter les effets de la précision finie :

(Q4) Comment améliorer la précision de la solution ?

Contribuer au développement d'approches automatiques :

- ne pas dissocier mathématique et informatique.

Introduction et motivations

But :

Travailler avec des polynômes ayant des données (coefficients ou racines) connues avec une incertitude : recherche de racines, primalité, calcul de PGCD, etc.

Introduction et motivations

But :

Travailler avec des polynômes ayant des données (coefficients ou racines) connues avec une incertitude : recherche de racines, primalité, calcul de PGCD, etc.

Raisons :

- Résultats provenant d'expériences.
- Représentation des nombres en machine.

Introduction et motivations

But :

Travailler avec des polynômes ayant des données (coefficients ou racines) connues avec une incertitude : recherche de racines, primalité, calcul de PGCD, etc.

Raisons :

- Résultats provenant d'expériences.
- Représentation des nombres en machine.

Applications :

- Traitement du signal et d'images.
- Robotique.
- Biologie moléculaire.

Exemples du PGCD

Exemple 1 :

Soient p et q deux polynômes unitaires et $\deg p > 1$.

On suppose de plus que p divise $q \implies \gcd(p, q) = p$.

Or pour toute constante $\varepsilon > 0$, on a $\gcd(p, q + \varepsilon) = 1$.

Exemples du PGCD

Exemple 1 :

Soient p et q deux polynômes unitaires et $\deg p > 1$.

On suppose de plus que p divise $q \implies \gcd(p, q) = p$.

Or pour toute constante $\varepsilon > 0$, on a $\gcd(p, q + \varepsilon) = 1$.

Exemple 2 :

$$p = z^2 - 3.0001z + 1.9999 \approx (z - 1)(z - 2),$$

$$q = z^2 - 1.9999z + 1.0001 \approx (z - 1)^2.$$

Pour ε une perturbation sur les coefficients, on aimerait dire :

- $\gcd(p, q) = z - 1$ pour $\varepsilon = 0.0001$.

Approche des problèmes

- Problèmes difficiles : on doit rechercher des « singularités ».
- La démarche générale :
 - Donner une **définition** précise de ce que l'on veut calculer.
 - Trouver des **algorithmes** pour ce calcul (souvent des heuristiques).
 - **Certifier** les résultats.
- appliquée à

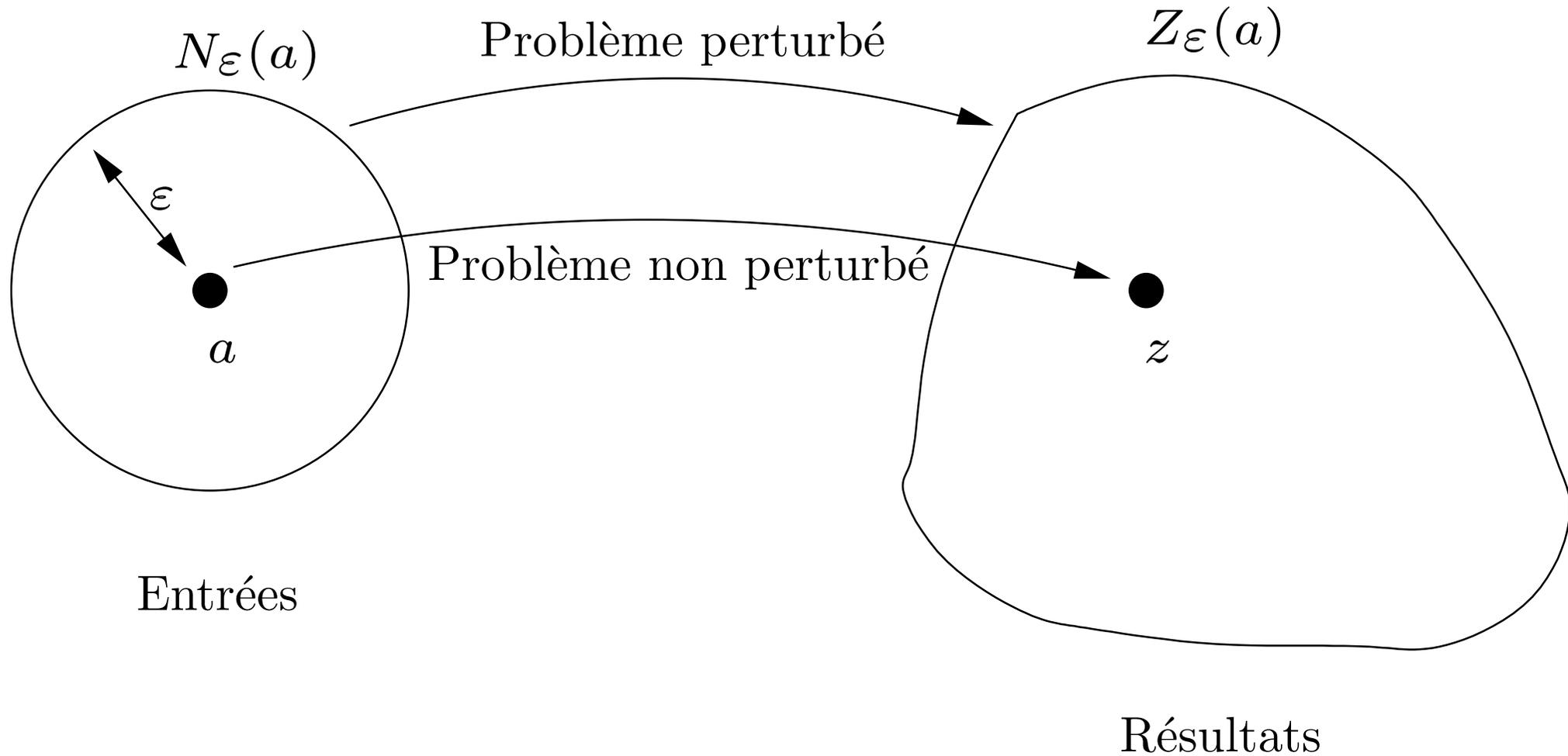
Approche des problèmes

- Problèmes difficiles : on doit rechercher des « singularités ».
- La démarche générale :
 - Donner une **définition** précise de ce que l'on veut calculer.
 - Trouver des **algorithmes** pour ce calcul (souvent des heuristiques).
 - **Certifier** les résultats.
- appliquée à
⇒ **(ε -pseudo)zéros**

Approche des problèmes

- Problèmes difficiles : on doit rechercher des « singularités ».
- La démarche générale :
 - Donner une **définition** précise de ce que l'on veut calculer.
 - Trouver des **algorithmes** pour ce calcul (souvent des heuristiques).
 - **Certifier** les résultats.
- appliquée à
 - ⇒ $(\varepsilon\text{-pseudo})$ zéros
 - ⇒ $(\varepsilon\text{-})$ primalité

Principe des calculs approchés



Un sujet en plein essor

- un sujet récent : 1996
- un package dans MAPLE 8 : 2001

Un sujet à la frontière de :

- l'arithmétique des ordinateurs (précision finie)
(journées Arinews)
- l'analyse numérique (journées LNF)
- le calcul formel

⇒ **Notre idée** : appliquer les outils de la **précision finie** au **calcul formel**.

Un sujet en plein essor

- un sujet récent : 1996
- un package dans MAPLE 8 : 2001

Un sujet à la frontière de :

- l'arithmétique des ordinateurs (précision finie)
(journées Arinews)
- l'analyse numérique (journées LNF)
- le calcul formel

⇒ **Notre idée** : appliquer les outils de la **précision finie** au **calcul formel**.

⇒ **conditionnement, analyse directe, analyse inverse**

Plan de l'exposé

I — Les pseudozéros

- Définition et algorithme de calcul
- Problème du polynôme le plus proche avec une racine donnée

II — Les pseudozéros et la primalité

- Définitions
- Présentation des algorithmes existants
- Apport des pseudozéros

Les pseudozéros : définition, calcul et intérêt

Pseudozéros : définition

Perturbation :

Voisinage du polynôme p

$$N_\varepsilon(p) = \{\hat{p} \in \mathbf{C}_n[z] : \|p - \hat{p}\| \leq \varepsilon\}.$$

Définition de l'ensemble des ε -pseudozéros :

$$Z_\varepsilon(p) = \{z \in \mathbb{C} : \hat{p}(z) = 0 \text{ pour } \hat{p} \in N_\varepsilon(p)\}.$$

On supposera la norme $\|\cdot\|$ *autoconjuguée* : $\|x\| = \|\bar{x}\|$.

Les pseudozéros sont facilement calculables

Théorème :

L'ensemble des ε -pseudozéros vérifie

$$Z_\varepsilon(p) = \left\{ z \in \mathbb{C} : |g(z)| := \frac{|p(z)|}{\|\underline{z}\|_*} \leq \varepsilon \right\},$$

où $\underline{z} = (1, z, \dots, z^n)$ et $\|\cdot\|_*$ est la norme duale de $\|\cdot\|$.

La démonstration nécessite de connaître « le » polynôme le plus proche de p ayant une racine donnée.

Le polynôme le plus proche ayant une racine donnée p_u

Soient p dans $\mathbf{C}_n[z]$ et $u \in \mathbf{C}$.

Énoncé du problème :

Trouver un polynôme $p_u \in \mathbf{C}_n[z]$ vérifiant $p_u(u) = 0$ et tel que s'il existe un polynôme $q \in \mathbf{C}_n[z]$ avec $q(u) = 0$ alors on ait $\|p - p_u\| \leq \|p - q\|$.

On cherche :

- une expression explicite de p_u ;
- un résultat d'unicité.

Calcul de p_u

Notons $\underline{u} := (1, u, u^2, \dots, u^n) \in \mathbf{C}^{n+1}$.

Il existe $d \in \mathbf{C}^{n+1}$ vérifiant $d^* \underline{u} = \|\underline{u}\|_*$ et $\|d\| = 1$.

Définissons les polynômes r et p_u par

$$r(z) = \sum_{k=0}^n r_k z^k \quad \text{avec} \quad r_k = \bar{d}_k,$$

$$p_u(z) = p(z) - \frac{p(u)}{r(u)} r(z).$$

p_u est le polynôme le plus proche de p
ayant u comme racine.

Unicité de p_u

Une condition suffisante d'unicité :

Théorème. *Si la norme $\| \cdot \|$ est strictement convexe, alors p_u est unique.*

C'est le cas, par exemple, pour les normes $\| \cdot \|_p$ pour $1 < p < \infty$.

On n'a pas unicité pour $\| \cdot \|_1$ et $\| \cdot \|_\infty$.

- en $\| \cdot \|_1$: $p(z) = 1 + z$, $u = 1$ alors $p_1^{(1)}(z) = 0$ et $p_1^{(2)}(z) = \frac{1}{3}(1 - z)$ conviennent.
- en $\| \cdot \|_\infty$: $p(z) = 1 + z$, $u = 0$ alors $p_0^{(1)}(z) = z$ et $p_0^{(2)}(z) = \frac{1}{2}z$ conviennent.

Cas polynôme réel, racine réelle

Soient p dans $\mathbf{R}_n[x]$ et $u \in \mathbf{R}$: identique au cas complexe. $\underline{u} := (1, u, u^2, \dots, u^n) \in \mathbf{R}^{n+1}$.

Il existe $d \in \mathbf{R}^{n+1}$ vérifiant $d^* \underline{u} = \|\underline{u}\|_*$ et $\|d\| = 1$.

Définissons les polynômes $r \in \mathbf{R}[x]$ et $p_u \in \mathbf{R}[x]$ par

$$r(z) = \sum_{k=0}^n r_k x^k \quad \text{avec} \quad r_k = d_k,$$

$$p_u(x) = p(x) - \frac{p(u)}{r(u)} r(x).$$

p_u est le polynôme le plus proche de p
ayant u comme racine.

Problème ouvert

Soient p dans $\mathbf{R}_n[x]$ et $u \in \mathbf{C} \setminus \mathbf{R}$.

Énoncé du problème :

Trouver un polynôme $p_u \in \mathbf{R}_n[x]$ annulant u et tel que s'il existe un polynôme $q \in \mathbf{R}_n[x]$ avec $q(u) = 0$ alors on ait $\|p - p_u\| \leq \|p - q\|$.

- Pas encore de formule explicite comme dans le cas complexe.
- Solution par des méthodes d'optimisation.

Algorithme de calcul

Algorithme de tracé de ε -pseudozéros :

1. On maille un carré contenant toutes les racines de p (commande MATLAB : `meshgrid`).
2. On calcule $g(z) := \frac{|p(z)|}{\|z\|_*}$ pour tous les points z de la grille.
3. On affiche la ligne de niveau $|g(z)| = \varepsilon$ (commande MATLAB : `contour`).

Algorithme de calcul

Algorithme de tracé de ε -pseudozéros :

1. On maille un carré contenant toutes les racines de p (commande MATLAB : `meshgrid`).
2. On calcule $g(z) := \frac{|p(z)|}{\|z\|_*}$ pour tous les points z de la grille.
3. On affiche la ligne de niveau $|g(z)| = \varepsilon$ (commande MATLAB : `contour`).

Problèmes :

- Trouver un carré contenant toutes les racines de p et tous les pseudozéros.
- Trouver un pas de grille qui sépare toutes les racines.

Simulation numérique

Ensemble des pseudozéros du polynôme « de Wilkinson »

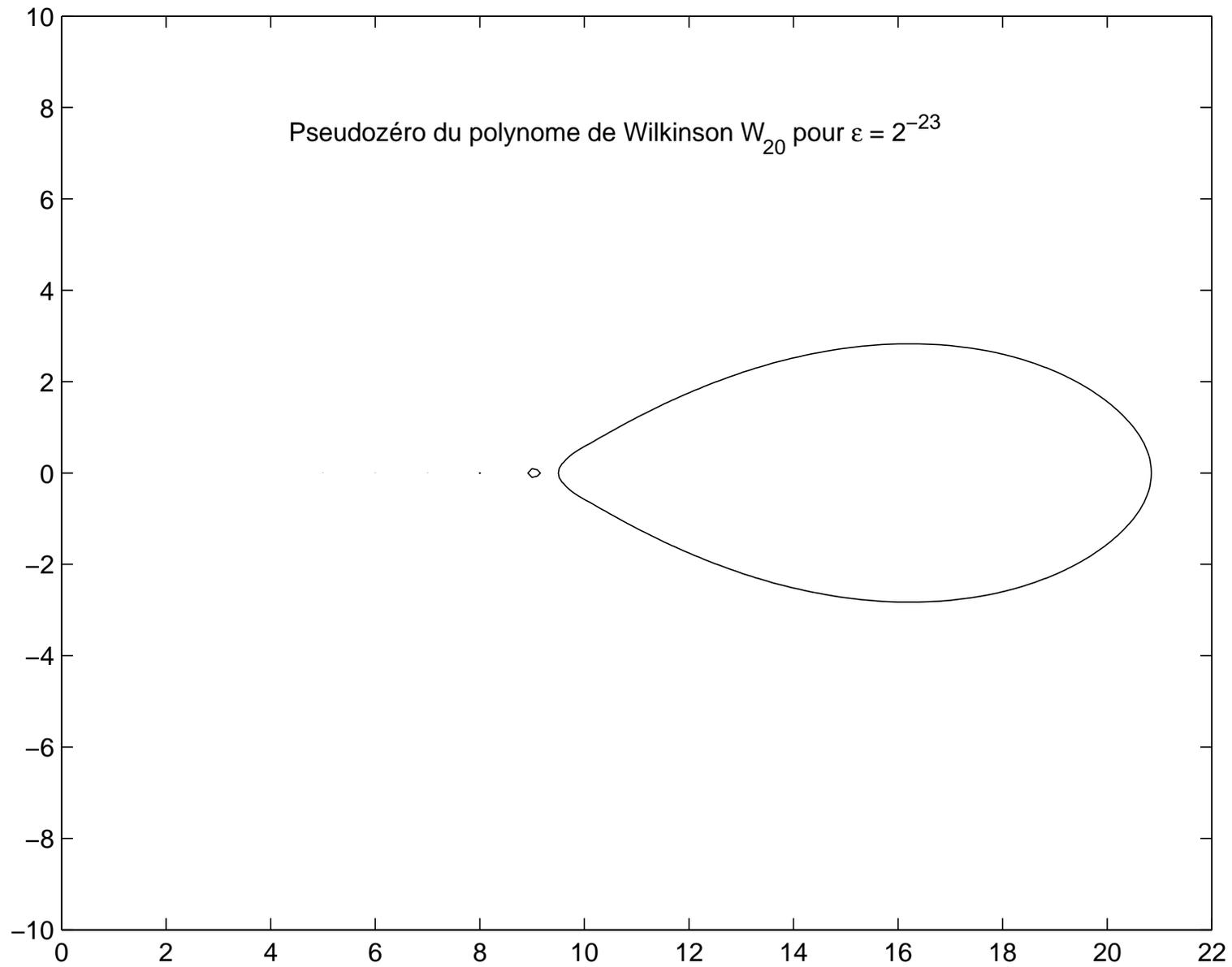
$$\begin{aligned}W_{20} &= (z - 1)(z - 2) \cdots (z - 20), \\ &= z^{20} - 210z^{19} + \cdots + 20!,\end{aligned}$$

en ne perturbant que le coefficient de z^{19} avec une perturbation inférieure à $\varepsilon = 2^{-23}$.

On utilise une norme $\|\cdot\|_\infty$ pondérée :

$$\|p\|_\infty = \max_i \frac{|p_i|}{m_i} \text{ avec } m_i \text{ réels positifs}$$

et la convention $m/0 = \infty$ si $m > 0$ et $0/0 = 0$.



Intérêts des pseudozéros

Intérêts

- une étude qualitative du polynôme
- comprendre les résultats des algorithmes
- utiliser des polynômes avec une incertitude sur leurs coefficients (mesure physique / précision finie)

Inconvénients

- le coût

Application des pseudozéros à la primalité

Définition d'un PGCD approché

Définition classique :

Soient p et q des polynômes de degrés respectifs n et m et soit ε un nombre positif. On appelle :

- **ε -diviseur** (ou diviseur approché) : tout diviseur des polynômes perturbés \hat{p} et \hat{q} vérifiant
 $\deg \hat{p} \leq n, \deg \hat{q} \leq m$ et $\max(\|p - \hat{p}\|, \|q - \hat{q}\|) \leq \varepsilon$.
- **ε -PGCD** (PGCD approché) : un ε -diviseur de degré maximum.

Définition d'un PGCD approché

Définition classique :

Soient p et q des polynômes de degrés respectifs n et m et soit ε un nombre positif. On appelle :

- **ε -diviseur** (ou diviseur approché) : tout diviseur des polynômes perturbés \hat{p} et \hat{q} vérifiant
 $\deg \hat{p} \leq n, \deg \hat{q} \leq m$ et $\max(\|p - \hat{p}\|, \|q - \hat{q}\|) \leq \varepsilon$.
- **ε -PGCD** (PGCD approché) : un ε -diviseur de degré maximum.

Remarques :

- Tolérance sur les coefficients (nombres flottants / mesures).
- Unicité du degré mais non du ε -PGCD.
- Dépendance par rapport au corps de base.

Définition de la ε - primalité

Définition :

Deux polynômes p et q sont ε -premiers entre eux si leur ε -PGCD est 1.

Calcul :

- Optimisation : algorithme de Karmarkar et Lakshman (1995).
- Borne de Sylvester : algorithme COPRIME [Beckermann et Labahn 1998].
- Graphique : les pseudo-zéros.

Algorithme de Karmarkar et Lakshman

Il s'agit d'un algorithme d'optimisation (moindre carré).

- **Entrée** : p et q deux polynômes unitaires de même degré n .
- **Sortie** : \hat{p} , \hat{q} deux polynômes unitaires de degré n et α tels que α soit racine commune de \hat{p} et \hat{q} avec $\|p - \hat{p}\|_2^2 + \|q - \hat{q}\|_2^2$ minimum.
- **Complexité** : polynomiale en n .

[Karmarkar & Lakshman - 98]

Étape de l'algorithme

1. Le minimum de $\|p - \hat{p}\|_2^2 + \|q - \hat{q}\|_2^2$ comme fonction de α sous les contraintes $\hat{p}(\alpha) = \hat{q}(\alpha) = 0$ est

$$\mathcal{N}_M(\alpha) := \frac{p(\alpha)\overline{p(\alpha)} + q(\alpha)\overline{q(\alpha)}}{\sum_{j=0}^{n-1} (\alpha\bar{\alpha})^j}, \quad \alpha = a + ib \in \mathbf{C}.$$

Étape de l'algorithme

1. Le minimum de $\|p - \hat{p}\|_2^2 + \|q - \hat{q}\|_2^2$ comme fonction de α sous les contraintes $\hat{p}(\alpha) = \hat{q}(\alpha) = 0$ est

$$\mathcal{N}_M(\alpha) := \frac{p(\alpha)\overline{p(\alpha)} + q(\alpha)\overline{q(\alpha)}}{\sum_{j=0}^{n-1} (\alpha\bar{\alpha})^j}, \quad \alpha = a + ib \in \mathbf{C}.$$

2. Soit α tel que $\frac{\partial \mathcal{N}_M}{\partial a}(\alpha) = 0$, $\frac{\partial \mathcal{N}_M}{\partial b}(\alpha) = 0$ et $\mathcal{N}_M(\alpha)$ minimum.

Étape de l'algorithme

1. Le minimum de $\|p - \hat{p}\|_2^2 + \|q - \hat{q}\|_2^2$ comme fonction de α sous les contraintes $\hat{p}(\alpha) = \hat{q}(\alpha) = 0$ est

$$\mathcal{N}_M(\alpha) := \frac{p(\alpha)\overline{p(\alpha)} + q(\alpha)\overline{q(\alpha)}}{\sum_{j=0}^{n-1} (\alpha\overline{\alpha})^j}, \quad \alpha = a + ib \in \mathbf{C}.$$

2. Soit α tel que $\frac{\partial \mathcal{N}_M}{\partial a}(\alpha) = 0$, $\frac{\partial \mathcal{N}_M}{\partial b}(\alpha) = 0$ et $\mathcal{N}_M(\alpha)$ minimum.
3. On a

$$\hat{p}_i = p_i - \frac{\overline{\alpha}^i p(\alpha)}{\sum_{k=0}^{n-1} (\overline{\alpha}\alpha)^k}, \quad \hat{q}_i = q_i - \frac{\overline{\alpha}^i q(\alpha)}{\sum_{k=0}^{n-1} (\overline{\alpha}\alpha)^k}.$$

Algorithme COPRIME

$$\|p\| = \sum |p_i|, \quad \|(p, q)\| = \max\{\|p\|, \|q\|\} = \max\{\sum |p_i|, \sum |q_i|\}.$$

Algorithme de Beckermann et Labahn (1998).

- **Entrée** : p et q deux polynômes.
- **Sortie** : borne inférieure de $\epsilon(p, q)$ défini par

$$\epsilon(p, q) = \inf\{\|(p - \hat{p}, q - \hat{q})\| : (\hat{p}, \hat{q}) \text{ ont une racine commune et } \deg \hat{p} \leq n, \deg \hat{q} \leq m\}.$$

- **Complexité** : en $\mathcal{O}((n + m)^2)$.

Matrice de Sylvester

$$S(p, q) = \begin{bmatrix} p_0 & 0 & \dots & 0 & q_0 & 0 & \dots & 0 \\ p_1 & p_0 & \dots & \vdots & q_1 & q_0 & \dots & \vdots \\ \vdots & \dots & \dots & 0 & \vdots & \dots & \dots & 0 \\ p_n & & \dots & p_0 & q_m & & \dots & q_0 \\ 0 & p_n & & p_1 & 0 & q_m & & q_1 \\ \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \vdots \\ 0 & \dots & 0 & p_n & 0 & \dots & 0 & q_m \end{bmatrix} \in \mathcal{M}_{n+m}(\mathbf{C}).$$

Critère de Sylvester : p et q sont premiers entre eux \iff la matrice $S(p, q)$ est régulière.

Présentation de la méthode

$$\epsilon(p, q) \geq \frac{1}{\|S(p, q)^{-1}\|}$$

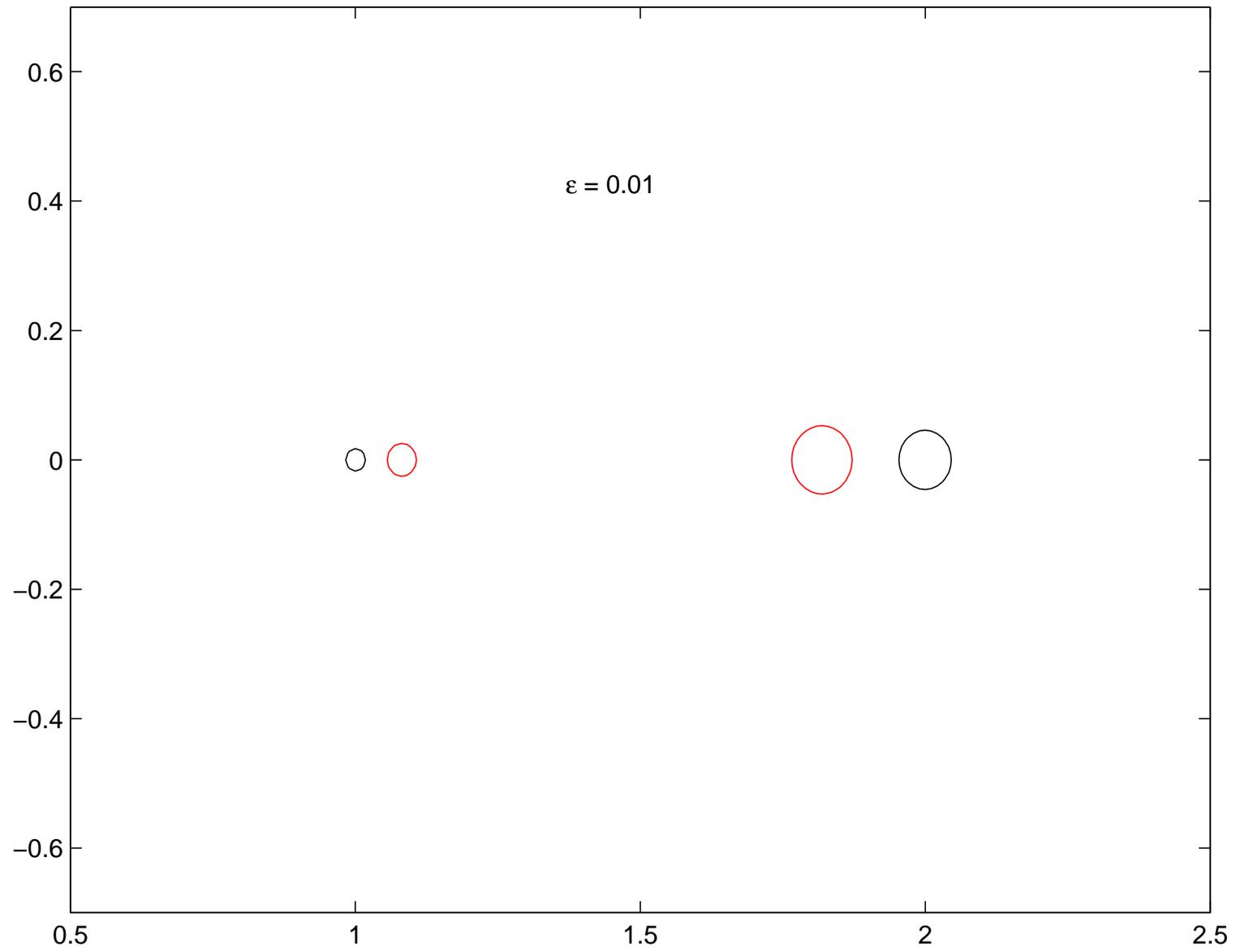
- Estimation de $\|S(p, q)^{-1}\|$ basée sur une SVD trop coûteux.
- On cherche une borne supérieure de $\|S(p, q)^{-1}\|$.

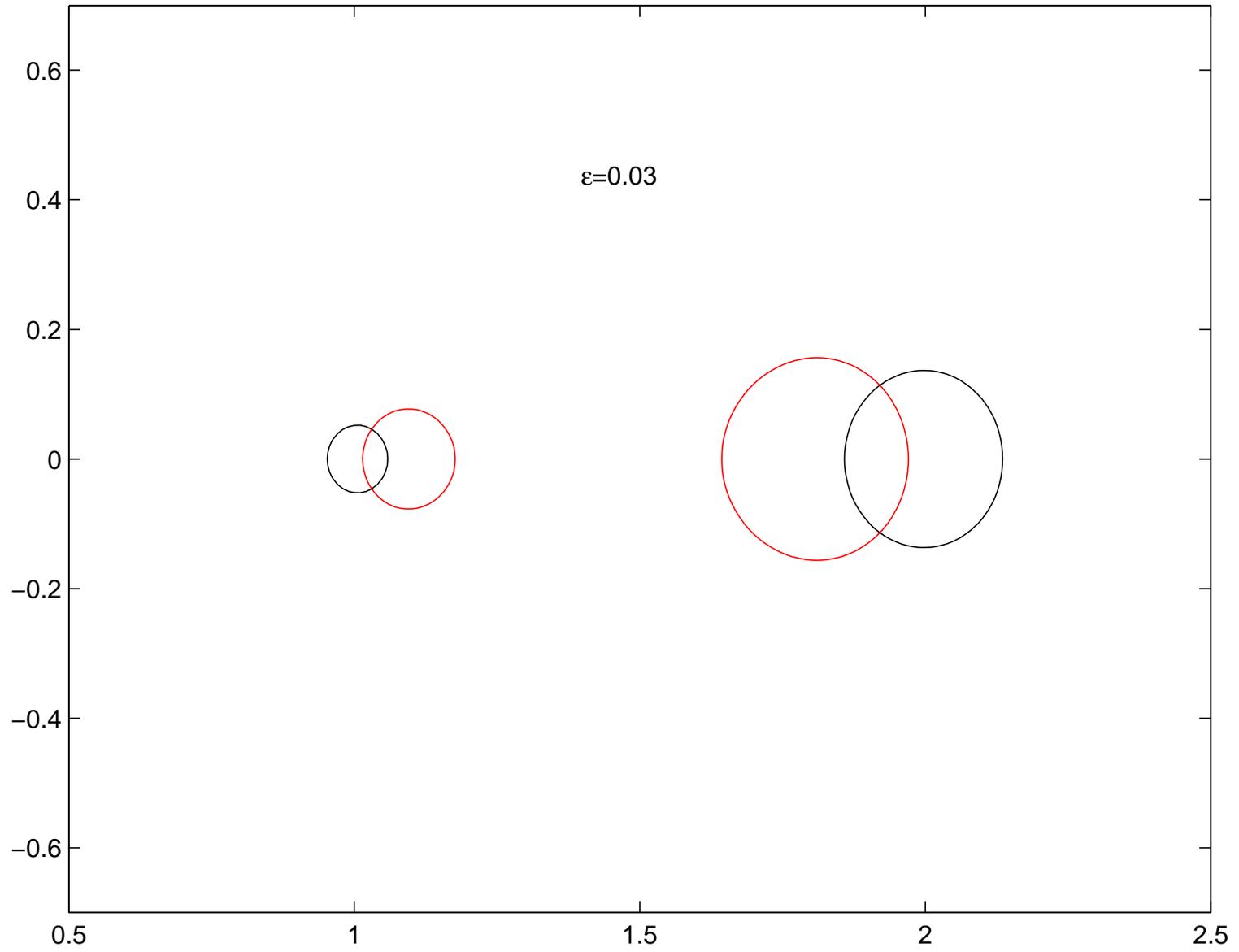
Les pseudozéros

- **Entrée** : p et q deux polynômes.
- **Sortie** : un graphique.
- **Inconvénient** : outil qualitatif.
- **Exemple en $\|\cdot\|_2$** :

$$p = (z - 1)(z - 2) = z^2 - 3z + 2$$

$$q = (z - 1.08)(z - 1.82) = z^2 - 2.9z + 1.9656$$





Perturbations structurées

Théorème. [Gastinel-Kahan] Soit $A \in \mathbf{C}^{n \times n}$ non singulière.

On a
$$\min \{ \|\Delta A\| \mid A + \Delta A \text{ singulière} \} = \frac{1}{\|A^{-1}\|}.$$

Perturbations structurées

Théorème. [Gastinel-Kahan] Soit $A \in \mathbf{C}^{n \times n}$ non singulière.

On a
$$\min \{ \|\Delta A\| \mid A + \Delta A \text{ singulière} \} = \frac{1}{\|A^{-1}\|}.$$

Travaux de S. Rump sur les perturbations structurées.

Soit $A \in \mathbf{C}^{n \times n}$ une matrice Toeplitz non singulière. On a

$$\min \{ \|\Delta A\| \mid \Delta A \text{ Toeplitz}, A + \Delta A \text{ singulière} \} = \frac{1}{\|A^{-1}\|}.$$

Perturbations structurées

Théorème. [Gastinel-Kahan] Soit $A \in \mathbf{C}^{n \times n}$ non singulière.

On a
$$\min \{ \|\Delta A\| \mid A + \Delta A \text{ singulière} \} = \frac{1}{\|A^{-1}\|}.$$

Travaux de S. Rump sur les perturbations structurées.

Soit $A \in \mathbf{C}^{n \times n}$ une matrice Toeplitz non singulière. On a

$$\min \{ \|\Delta A\| \mid \Delta A \text{ Toeplitz}, A + \Delta A \text{ singulière} \} = \frac{1}{\|A^{-1}\|}.$$

\implies Matrice de Sylvester : Toeplitz par bloc

Conclusion et perspectives

Les pseudozéros permettent :

1. une meilleure compréhension de l'effet de la perturbation des coefficients ;
2. un test de la ε -primalité de deux polynômes.

Conclusion et perspectives

Les pseudozéros permettent :

1. une meilleure compréhension de l'effet de la perturbation des coefficients ;
2. un test de la ε -primalité de deux polynômes.

Perspectives :

- Un package convivial de tracé de pseudozéros ;
- Une formule explicite dans le cas réel-complexe du polynôme le plus proche avec une racine donnée ;
- Étude des perturbations structurées de la matrice de Sylvester.

Divers

Trucs utiles

Les pseudozéros

- ▶ Mosier (1986) : étude avec la norme $\| \cdot \|_{\infty}$.
- ▶ Trefethen et Toh (1994) : étude avec la norme $\| \cdot \|_2$.
pseudozéros \approx pseudospectres de la matrice compagnon.
- ▶ Chatelin et Frayssé (1996) : Synthèse des articles précédents dans *Lectures on Finite Precision Computations* (SIAM)
- ▶ Stetter (1999) : *algèbre numérique polynomiale*. Généralise les travaux précédents.
- ▶ Zhang (2001) : Étude en norme $\| \cdot \|_2$ de l'influence de la base (conditionnement de l'évaluation).

Choix de la grille

Soit p un polynôme unitaire de degré n et $\{z_i\}$ l'ensemble de ses n racines. Notons $r = \max_{i=1;\dots;n} |z_i|$. On a

$$r \leq \max\left\{1, \sum_{k=1}^n |p_k|\right\}.$$

Notons $R := \max\left\{1, \sum_{i=1}^n |p_i| + n\varepsilon\right\}$. On montre que (en norme $\|\cdot\|_p$)

$Z_\varepsilon(p) \subset B(0, R)$ boule fermée de centre 0 et de rayon R .

Complexité du tracé

Notons L la longueur du carré et h le pas de discrétisation. L'évaluation de $g(u)$ nécessite

- l'évaluation d'un polynôme, ce qui se fait en $\mathcal{O}(n)$,
- le calcul de la norme d'un vecteur, dont la complexité dépend de la norme.

Notons $\mathcal{O}(\|\cdot\|_*)$ cette complexité. La complexité de l'algorithme précédent est donc en

$$\mathcal{O}\left(\left(\frac{L}{h}\right)^2(n + \|\cdot\|_*)\right).$$

L et h dépendent de n mais aussi des coefficients du polynôme.

Deux théorèmes relatifs aux pseudozéros

Théorème. [Mosier] *Si on suppose que l'ensemble des pseudozéros est borné et si $q \in N_\varepsilon(p)$ alors q et p ont le même nombre de racines (en comptant les multiplicités) dans chaque composante connexe de $Z_\varepsilon(p)$. De plus, il y a au moins une racine du polynôme p dans chaque composante connexe de $Z_\varepsilon(p)$.*

Théorème. [Mosier] *Un voisinage d'une racine de p contient deux racines de p si et seulement si il contient un $u \in \mathbf{C}$ comme racine double de p_u .*

Pseudozéros de polynômes réels

Si $p \in \mathbf{R}_n[x]$, on définit

$$N_\varepsilon(p) := \{q \in \mathbf{R}_n[x] : \|p - q\| \leq \varepsilon\}.$$

Deux cas :

- on cherche les pseudozéros réels : identique au cas complexe ;
- on cherche tous les pseudozéros complexes non réels.

On définit alors l'ensemble des pseudozéros par

$$Z_\varepsilon(p) := \{z \in \mathbf{C} : \hat{p}(z) = 0 \text{ pour } \hat{p} \in N_\varepsilon(p)\}.$$

$Z_\varepsilon(p)$ est symétrique par rapport à l'axe des réels.

Deux théorèmes relatifs au pseudozéros de polynômes réels

Théorème. [Stetter] *Soit Z_μ une composante connexe de $Z_\varepsilon(p)$ de multiplicité 1. Alors ou bien $Z_\mu \subset \mathbf{R}$ ou bien $Z_\mu \cap \mathbf{R} = \emptyset$.*

Théorème. [Stetter] *Une composante connexe Z_μ de $Z_\varepsilon(p)$ vérifiant $\emptyset \neq Z_\mu \cap \mathbf{R} \neq Z_\mu$ est de multiplicité au moins 2.*

Norme et vecteur duals

- Norme duale de $\| \cdot \|$:

$$\|y\|_* = \sup_{\|x\| \neq 0} \frac{|y^*x|}{\|x\|} = \sup_{\|x\|=1} |y^*x|.$$

- Vecteur dual :

Théorème 1. *Pour tout vecteur y , il existe un vecteur z vérifiant*

$$z^*y = \|z\|_* \|y\|.$$

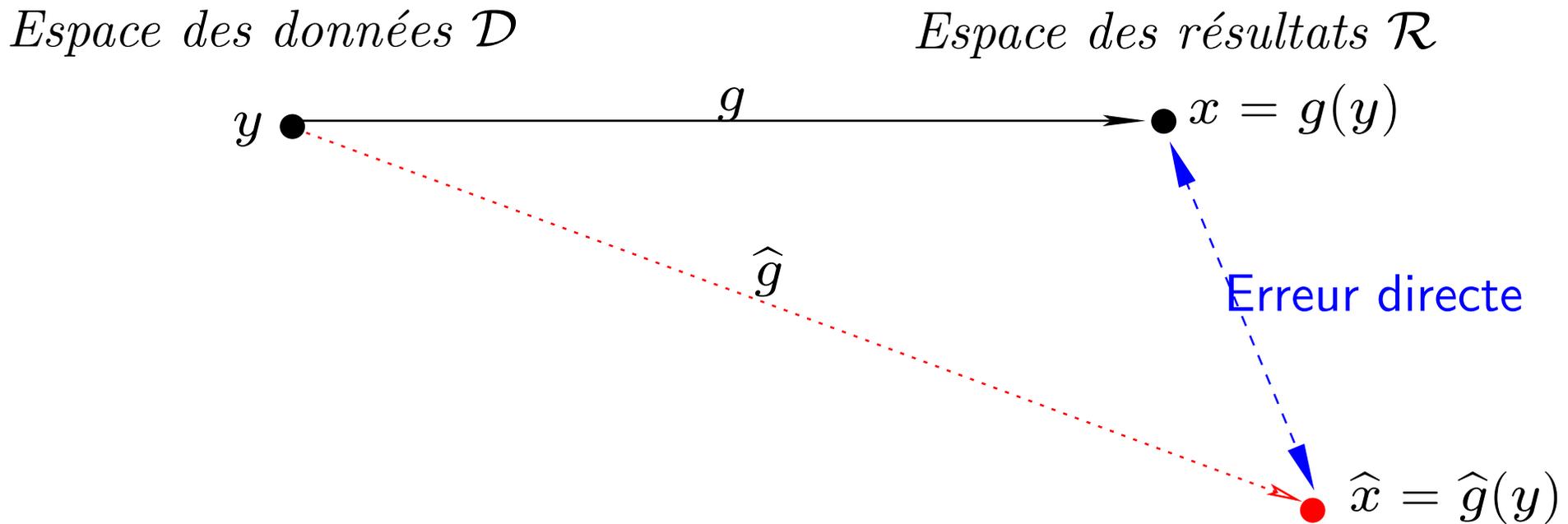
Décomposition en Valeurs Singulières

Théorème 2. Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$ avec $\mathbb{K} = \mathbb{R}$ ou \mathbb{C} de rang k . Alors A peut s'écrire sous la forme

$$A = UDV^*,$$

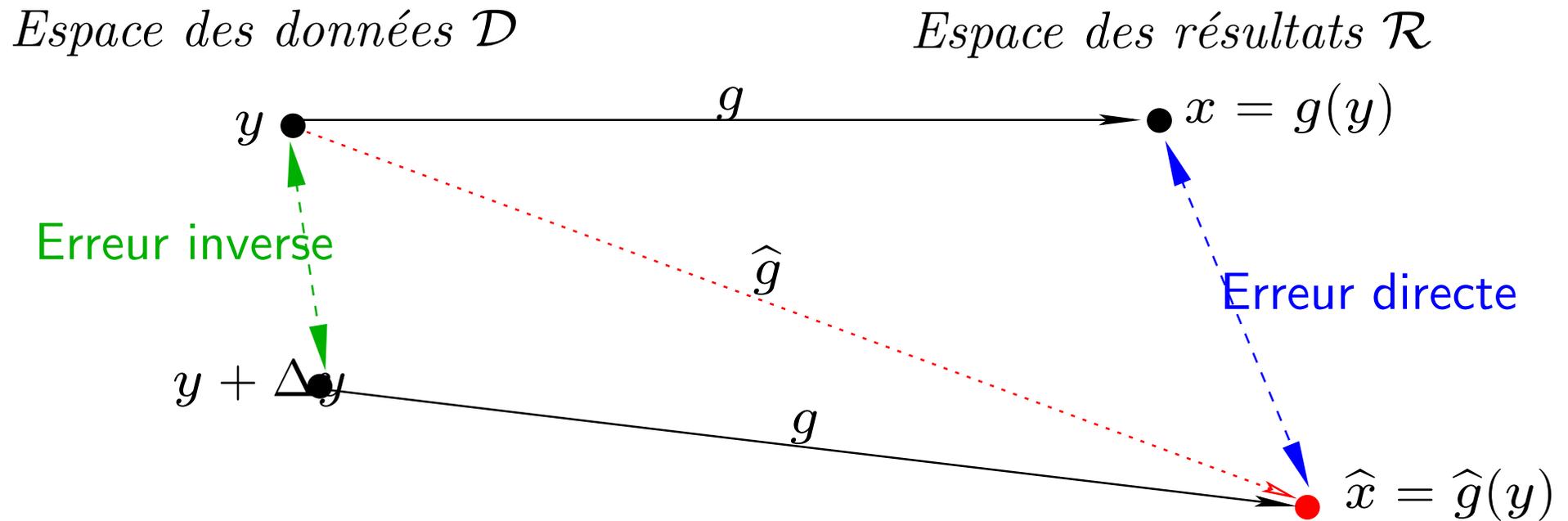
où $U \in \mathcal{M}_m(\mathbb{K})$ et $V \in \mathcal{M}_n(\mathbb{K})$ sont des matrices unitaires (on dit souvent orthogonales dans le cas réel). La matrice $D = (\sigma_{ij}) \in \mathcal{M}_{m,n}(\mathbb{K})$ vérifie $\sigma_{ij} = 0$ pour $i \neq j$ et $\sigma_{11} \geq \sigma_{22} \geq \dots \geq \sigma_{kk} > \sigma_{k+1,k+1} = \dots = \sigma_{qq} = 0$ où $q = \min(m, n)$.

L'effet des approximations



- Analyse directe.
- Analyse inverse :
consiste à identifier \hat{x} à la solution d'un problème perturbé (\hat{P}) : $\hat{x} = g(y + \Delta y)$.

Analyse inverse : identifier \hat{x} à la solution d'un problème perturbé



Intérêt de l'analyse inverse

(Q1) : Comment mesurer la difficulté du problème à résoudre ?

Le conditionnement caractérise la sensibilité de la solution d'un **problème** à des **perturbations de ses données**.

Nombre de conditionnement : $\kappa(P, y) = \limsup_{\delta \rightarrow 0, \Delta y \in \mathcal{P}(\delta)} \left\{ \frac{\|\Delta x\|_{\mathcal{R}}}{\|\Delta y\|_{\mathcal{D}}} \right\}.$

(Q2) : Comment apprécier la fiabilité de l'algorithme de résolution ?

L'erreur inverse associée à \hat{x} est $\eta(\hat{x}) = \min_{\Delta y \in \mathcal{D}} \{ \|\Delta y\|_{\mathcal{D}} : \hat{x} = G(y + \Delta y) \}$.

(Q3) : Comment estimer la précision de la solution calculée ?

Au premier ordre et en général :

précision \lesssim nombre de conditionnement

\times erreur inverse.

- L'algorithme G est **inverse-stable** s'il calcule \hat{x} en précision finie u avec $\eta(\hat{x}) \approx u$.
- inverse-stable $\not\Rightarrow$ solution précise.