Résolution de systèmes linéaires par développement en série

Mémoire de L3 maths-info

Dorian Lesbre
encadré par Jérémy Berthomieu

12 juin 2019

Plan

1. Résolution dans un anneau général
    - Problème
    - Inversion modulaire
    - Résolution modulaire
2. Résolution dans $\mathbb{Z}$
    - Modèle de calcul
    - Solution modulo p^{2^k}
    - Reconstruction rationnelle
3. Résolution dans $\mathbb{K}[x]$
    - Expansion P-adique
    - Solution en série partielle
    - Composants de haut ordre

Références

John D. Dixon.
Exact solutions of linear equations using *p*-adic expansions.
Numer. Math, (40.1) :137–141, février 1982.

Robert T. Moenck et John H. Carter.
***Symbolic and Algebraic Computation*, chapter Approximate algorithms to derive exact solutions to systems of linear equations, pages 65–73.**
Springer Berlin Heidelberg, 1979.

Arne Storjohann.
High-order lifting.
Proceedings of the 2002 international Symposium on Symbolic and Algebraic computation, pages 246–254, ACM, 2002.

Résolution dans un anneau général

Problème

Soit $(\mathcal{A}, +, \times)$ un anneau intègre et commutatif.

Posons le système

$$Ax = b$$

Avec $A \in \mathcal{M}_{n \times n}(\mathcal{A})$, b et $x \in \mathcal{A}^n$ des vecteurs de taille n.

Complexité en algébrique et binaire

Deux modèles de complexité

Algébrique : considère que les opérations d'anneaux sont élémentaires

Binaire : prend en compte le coût des calculs dans l'anneau.

Complexité en algébrique et binaire

Deux modèles de complexité

Algébrique : considère que les opérations d'anneaux sont élémentaires

Binaire : prend en compte le coût des calculs dans l'anneau.

Exemple : le pivot de Gauss a une complexité algébrique en $O(n^3)$ mais nécessite de grands calculs intermédiaires.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ 0 & a_{2,2} - a_{1,2}a_{1,1}^{-1} & \cdots & a_{2,n} - a_{1,n}a_{1,1}^{-1} \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & 0 & a_{n,n} - \sum_{i=1}^{n-1} \left(a_{i,n} - \sum_{j=1}^{i} a_{j,n} a_{j,j}^{-1} \right) \end{bmatrix}$$

Solutions alternatives

Afin de limiter la taille des calculs intermédiaires, nous pouvons à partir d'une borne M de la solution :

- résoudre modulo $p > M$,
  - complexité binaire de $O(n^3 \log^2 M)$
- résoudre modulo $(p_1, \ldots, p_m)$ avec $\prod p_i > M$ et conclure par reste chinois
  - complexité binaire de $O(n^3 m \log^2 p)$
- résoudre modulo $p^m > M$.
  - complexité binaire de $O(n^3 \log^2 p + n^2 m \log^2 p)$

Inversion matricielle

```
Inversion(A, B, k)
    Pour i = 0 à k − 1 faire
        E ← I − A × B mod p^(2^i)
        B ← B × (I + E)
    renvoyer B
```

Inversion matricielle

```
Inversion(A, B, k)
    Pour i = 0 à k − 1 faire
        E ← I − A × B mod p^(2^i)
        B ← B × (I + E)
    renvoyer B
```

$$
\begin{aligned}
A \times B(I+E) &\equiv A \times B_{2^{i-1}}(I + p^{2^{i-1}}\tilde{E}) \mod \langle p^{2^i} \rangle \\
&\equiv (I - p^{2^{i-1}}\tilde{E})(I + p^{2^{i-1}}\tilde{E}) \mod \langle p^{2^i} \rangle \\
&\equiv I - p^{2^i}\tilde{E}^2 \mod \langle p^{2^i} \rangle \\
&\equiv I \mod \langle p^{2^i} \rangle
\end{aligned}
$$

Inversion matricielle

```
Inversion(A, B, k)
    Pour i = 0 à k − 1 faire
        E ← I − A × B mod p^(2^i)
        B ← B × (I + E)
    renvoyer B
```

Complexité

Complexité algébrique en $O(n^3)$ et binaire en $O(2^k n^3)$

Résolution modulaire dans le cas général

```
ResolutionModulaire(A, b, p, k)
   A' ← A mod p
   b' ← b mod p
   x  ← A'^{-1} × b mod p
   pM ← A' − A
   Pour i = 1 à 2^{k-1} faire
      x ← A'^{-1} × pM × x + A'^{-1} × b
   renvoyer x
```

Transformation de $Ax = b$, en décomposant

$$A = A' - pM$$

Le système devient alors $A'x = pMx + b$.

Correction et complexité

Soit x la solution $Ax = b$ et (x_k) les valeurs calculées.

$$\forall k \geqslant 1,\ A'(x - x_{k+1}) = pM(x - x_k)$$

Il s'agit d'une suite géométrique de raison $pA'^{-1}M$, donc

$$x - x_{2^{k+1}} \equiv 0 \mod \langle p^{k+1} \rangle$$

$x_{2^{k+1}}$ est donc bien solution modulo $\langle p^{2^{k+1}} \rangle$.

Complexité

Complexité algébrique en $O(n^3 + 2^k n^2)$ et binaire en $O(n^3 + 4^k n^2)$

Résolution modulaire optimisée

```
ResolutionModulaire(A, b, p, k)
  C ← inverse(A mod p, p)  // inverse modulo ⟨p⟩
  x ← C × b
  Pour i = 1 à 2^(k−1) faire
    w ← b − A × x mod p^(i+1)
    y ← C × w / p^i mod p
    x ← x + p^i×y mod p^(i+1)
  renvoyer x
```

Calcul des différences successives divisées par p^i.

Résolution modulaire optimisée

```
ResolutionModulaire(A, b, p, k)
  C ← inverse(A mod p, p)  // inverse modulo ⟨p⟩
  x ← C × b
  Pour i = 1 à 2^(k−1) faire
    w ← b − A × x mod p^(i+1)
    y ← C × w / p^i mod p
    x ← x + p^i×y mod p^(i+1)
  renvoyer x
```

L'élément limitant est le produit Ax. Mais comme $x \leftarrow \tilde{x} + p^i y$ il peut se calculer par $A\tilde{x}$ (déjà connu) $+Ay$ (borné).

Résolution dans $\mathbb{Z}$

Calcul dans $\mathbb{Z}$

Les opérations entre entiers ont une complexité proportionnelle à leur logarithme (nombre de bits).

Fixons $\beta = \max\{\|A\|_\infty, \|b\|_\infty\}$ comme unité de taille des entiers.

Bornes de la solution

Les numérateurs et dénominateurs de la solution sont bornés par $\beta^n \sqrt{n^n}$.

Solution modulo p^{2^k}

```
SolutionModulaire(A, b, p, k)
   C ← inverse(A mod p, p)  // inverse modulo p
   x ← tableau(2^k, n)
   x[0] ← C × b mod p
   Pour i = 1 à 2^(k−1) faire
      b ← p^(−1) × (b − A × x[i−1])
      x[i] ← C×b mod p
   renvoyer x[0] + x[1]×p +...+ x[2^(k−1)]×p^(2^(k−1))
```

Solution modulo p^{2^k}

```
SolutionModulaire(A, b, p, k)
  C ← inverse(A mod p, p)  // inverse modulo p
  x ← tableau(2^k, n)
  x[0] ← C × b mod p
  Pour i = 1 à 2^(k−1) faire
    b ← p^(−1) × (b − A × x[i−1])
    x[i] ← C×b mod p
  renvoyer x[0] + x[1]×p +...+ x[2^(k−1)]×p^(2^(k−1))
```

Complexité

Complexité algébrique en $O(n^3 + 2^k n^2)$ et binaire en $O(n^3 + 2^k n^2 \log n)$.

Correction

L'algorithme calcule les suites (b_i) et (x_i)

$$\begin{cases} b_0 = b \text{ et } b_i = p^{-1}(b_{i-1} - Ax_{i-1}) \\ x_i \equiv Cb_i \mod p \end{cases}$$

et renvoie $x = \sum_{i=0}^{2^{k-1}} p^i x_i$

$$Ax = \sum_{i=0}^{2^{k-1}} p^i Ax_i = \sum_{i=0}^{2^{k-1}} p^i (b_i - pb_{i+1}) = b_0 - p^{2^k} b_{2^k} \equiv b \mod p^{2^k}$$

Reconstruction rationnelle

```
SolutionRationnelle(p, k, x_r)
  u1, u2 ← p^(2^k), x_r
  v1, v2 ← 0, 1
  i ← −1
  Tant que u2 < p^(2^(k−1)) faire
    q ← partieEntiere(u1 / u2)
    u1, u2 ← u2, u1 − q × u2
    v1, v2 ← v2, v1 + q × v2
    i ← i+1
  renvoyer (−1)^i × fraction(u1, v1)
```

Complexité

Choix de $2^k = O(n \log n)$ convient et donne $O(n^3 + n^3 \log^2 n)$

Résolution dans $\mathbb{K}[x]$

Idée

Sur les entiers nous avions une complexité en

$$O(\underbrace{n^3}_{\text{inverse}} + \underbrace{n^2 \log n}_{\text{mat}\times\text{vec}} \underbrace{n \log n}_{\text{taille}})$$

Nous allons chercher à la remplacer par

$$O(\underbrace{n^3 \log n}_{\text{inverse}} + \underbrace{n^3 \log n}_{\text{mat}\times\text{mat}} \underbrace{\log n}_{\text{taille}})$$

Où avec des algorithmes optimisés :

$$O(\underbrace{n^\omega \log n}_{\text{inverse}} + \underbrace{n^\omega \log n}_{\text{mat}\times\text{mat}} \underbrace{\log n}_{\text{taille}})$$

avec $2 < \omega < 3$

Expansion P-adique

L'expansion P-adique d'une fraction f dont le dénominateur est premier avec P s'écrit $f = \sum c_i P^i$ où $\deg c_i < \deg P$

Expansion P-adique

L'expansion P-adique d'une fraction f dont le dénominateur est premier avec P s'écrit $f = \sum c_i P^i$ où $\deg c_i < \deg P$

Posons trois fonctions à partir de cette écriture :

- $\mathrm{Gauche}(f, k) = \sum c_{i+k} P^i$ (décalage de k)
- $\mathrm{Tronc}(f, k) = \sum_{i=0}^{k-1} c_i P^i$ (troncature à l'ordre k)
- si $Q \perp P$ alors $\mathrm{Inverse}(Q, k)$ est l'unique R tel que $R = \mathrm{Tronc}(R, k)$ et $\mathrm{Tronc}(RQ, k) = \mathrm{Tronc}(QR, k) = 1$

Série solution partielle

```
SerieSolution[P](A, b, k)
  (E_1, ..., E_k) ← ComposantsHautOrdre[P](A, k)
  B ← [b_0 | b_1 | ... | b_{2^k−1}] // expansion P-adique de b
  Pour i = k − 1 à 1 faire
    B ← premières (2^k − 2^i) colonnes de B
    B ← Gauche(−A×Tronc(Gauche(E_i B, 1), 1), 1)
    B ← [0_{n×m2^i} | B]  // on rajoute m2^i colonnes
    B ← B + B
  B ← Tronc(E_1 B, 2)
  renvoyer B[0] + ... + B[2^k−2]×P^{2^k−2} +
     Tronc(B[2^k−1], 1)P^{2^k−1}
    // avec B[i] la i-ième colonne de B
```

Composants de haut ordre

Notons $Z_i = \mathrm{Inverse}(A, 2^i)$ et $E_i = \mathrm{Gauche}(Z_i, 2^i - 2)$ les composants de haut ordre de A^{-1}. Notons l'expansion P-adique de $A^{-1} = \sum_{i=0}^{\infty} c_i P^i$ alors :

$$Z_1 = \underbrace{c_0 + c_1 P}_{E_1}$$

$$Z_2 = c_0 + c_1 P + \underbrace{c_2 P^2 + c_3 P^3}_{E_2 P^2}$$

$$Z_3 = c_0 + c_1 P + c_2 P^2 + c_3 P^3 + c_4 P^4 + c_5 P^5 + \underbrace{c_6 P^6 + c_7 P^7}_{E_3 P^6}$$

Calcul des composants de haut-ordre

```
ComposantsHautOrdre[P](A, k)
  L ← Inverse(A, 1) // Z_0
  H ← Tronc(L × Gauche(I − A×L,1), 1)
  E_1 ← L + P×H
  Pour i = 2 à k faire
    L ← Tronc(Gauche(E_{i−1}Gauche(−A×L, 1),1),1)
    H ← Tronc(Gauche(E_{i−1}Gauche(−A×H, 1),1),1)
    E_i ← L + P×H
  renvoyer (E_1, ..., E_k)
```

Complexité

Complexité binaire en $O(kn^{\omega})$

Inverse matriciel

Définissons à partir des expansions de A^{-1} et $A^{-1}B$:

$$A^{-1} = \underbrace{* + *P + \ldots + *P^{\ell-1}}_{C} + *\,P^{\ell} + \ldots$$

$$A^{-1}B = \underbrace{* + *P + \ldots + *P^{k-1}}_{D} + \underbrace{*P^{k} + \ldots + *P^{k+\ell-1}}_{EP^{k}} + *P^{k+\ell} + \ldots$$

Théorème

$$E = \mathrm{Tronc}(C \cdot \mathrm{Gauche}(B - AD, k), \ell)$$

Preuve

$$\begin{aligned}
AD &= A(A^{-1}B - EP^k - *P^{k+\ell}) \\
B - AD &= AEP^k - *P^{k+\ell} \\
\mathrm{Gauche}(B - AD, k) &= AE - *P^\ell \\
C \cdot \mathrm{Gauche}(B - AD, k) &= CAE - *P^\ell \\
&= (A^{-1} - *P^\ell)AE - *P^\ell \\
&= E - *P^\ell \\
\mathrm{Tronc}(C \cdot \mathrm{Gauche}(B - AD, k), \ell) &= E
\end{aligned}$$

Schéma de la preuve de correction

Montrer par récurrence

$$L_j = c_{2j-2}$$
$$H_j = c_{2j-1}$$
$$E_j = c_{2j-2} + c_{2j-1}P$$

en posant $H_{i+1} = \mathrm{Tronc}\Big(\underbrace{\mathrm{Gauche}\big(E_i \overbrace{\mathrm{Gauche}(-AH_i, 1)}^{R}, 1\big)}_{S}, 1\Big)$

et montrant que $R = \dfrac{(I - AZ_i)}{P^{2^{i+1}-1}}$ et $S = \mathrm{Gauche}(Z_i R, 1)$

Conclusion

Pour résumer, nous avons les complexités suivantes :

Algorithme	Algébrique	Binaire
Inversion	n^3	$2^k n^3$
Solution modulo $\langle p^{2^k} \rangle$	$n^3 + 2^k n^2$	$n^3 + 4^k n^2$
Solution modulo p^{2^k}	$n^3 + 2^k n^2$	$n^3 + 2^k n^2 \log n.$
Reconstruction rationnelle	$n^3 + n^3 \log n$	$n^3 + n^3 \log^2 n$
Solution modulo P^{2^k}	$kn^{\omega-1}$	$k2^k n^{\omega-1}$