Algorithmes efficaces pour les grands nombres et polynômes : Partie 2

Salem BENFERHAT

Centre de Recherche en Informatique de Lens (CRIL-CNRS)
email : benferhat@cril.fr

Ce que nous avons vu dans la première partie ...

Grâce à la décomposition d'un grand nombre et grâce à une simple reformulation du calcul du produit, nous avons un algorithme en :

$$O(n^{1.584})$$

Peut-on encore mieux faire?

Retour sur les polynômes

Polynômes

Définition

$$p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

où :

- a_i sont des réels (positifs ou négatifs ou nuls)
- a_n est différent de zéro
- n est appelé le degré du polynôme $p(x)$.

Polynômes

Définition

$$p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

où :

- a_i sont des réels (positifs ou négatifs ou nuls)
- a_n est différent de zéro
- n est appelé le degré du polynôme $p(x)$.

Ce que l'on a vu

Evaluation de $p(x)$ lorsque $x = x_0$, avec un algorithme efficace de $O(n)$

Représentatation d'un polynôme

Tableau

Un polynôme :

$$p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n,$$

sera tout simplement représenté par un tableau de réels :

a_0	a_1	a_2	....	a_{n-1}	a_n
0	1	2	....	$n-1$	n

Exercice

- Ecrire une fonction qui calcule la somme de deux polynômes.
- Ecrire une fonction qui calcule le produit de deux polynômes.

Addition de deux polynômes

```c
double *addition(double A[], double B[], int N)
{
  int i;
  double *res=(double *) malloc ((N+1)*sizeof(double));
  for (i=0; i<=N; i++)
  {
    res[i]=A[i]+B[i];
  }
  return res;
}
```

Produit de deux polynômes

```c
double *multiplicaion(double A[], double B[], int N)
{
  int i, j;
  double *res=(double *) calloc ((2*(N+1))*sizeof(double), 0);
  for (i=0; i<=N; i++)
  {
    for (j=0; j<=N; j++)
      res[i+j]=res[i+j]+(A[i]*B[j]);
  }
  return res;
}
```

Complexité sur les opérations de base

Récapitulatifs

- Addition : $O(n)$
- Multiplication : $O(n^2)$
- Evaluation d'une valeur donnée : $O(n)$ (Algorithme de Horner).

Une autre représentation

des polynômes

à base de points

Commençons par le point

Définition d'un type enregistrement qui représente un point :

```
typdef struct
{
float abscisse;
float ordonnée;
} point;
```

La droite

Une droite

Entre deux points de coordonnées : $(x_0, p(x_0))$ et $(x_1, p(x_1))$ distincts on ne peut tracer qu'une et une seule droite qui passe par ces deux points.

La droite

Une droite

Entre deux points de coordonnées : $(x_0, p(x_0))$ et $(x_1, p(x_1))$ distincts on ne peut tracer qu'une et une seule droite qui passe par ces deux points.

Une droite = un polynôme

Une droite est au fait un polynôme de degré 1 de la forme :

$$p(x) = a_0 + a_1.x$$

La droite

Une droite

Entre deux points de coordonnées : $(x_0, p(x_0))$ et $(x_1, p(x_1))$ distincts on ne peut tracer qu'une et une seule droite qui passe par ces deux points.

Une droite = un polynôme

Une droite est au fait un polynôme de degré 1 de la forme :

$$p(x) = a_0 + a_1.x$$

Il est très facile de calculer a_0 et a_1 si on connaît les deux points $(x_0, p(x_0))$ et $(x_1, p(x_1))$.

Avec trois points ...

Un peu plus loin qu'une droite

Avec trois points de coordonnées : $(x_0, p(x_0))$, $(x_1, p(x_1))$ et $(x_2, p(x_2))$ distincts on peut représenter un (et un seul) polynôme de degré 2 de la forme :

$$p(x) = a_0 + a_1.x + a_2.x^2$$

Avec trois points ...

Un peu plus loin qu'une droite

Avec trois points de coordonnées : $(x_0, p(x_0))$, $(x_1, p(x_1))$ et $(x_2, p(x_2))$ distincts on peut représenter un (et un seul) polynôme de degré 2 de la forme :

$$p(x) = a_0 + a_1.x + a_2.x^2$$

Il est très facile de calculer a_0, a_1 et a_2 si on connaît les trois points : nous disposons de trois équations pour trois variables inconnues.

Avec n+1 points ...

Polynôme de degré n

Avec (n+1) points de coordonnées : $(x_0, p(x_0))$, ..., $(x_{n+1}, p(x_{n+1}))$ distincts on peut représenter un (et un seul) polynôme de degré n de la forme :

$$p(x) = a_0 + a_1.x + ... + a_n.x^n$$

Avec n+1 points ...

Polynôme de degré n

Avec (n+1) points de coordonnées : $(x_0, p(x_0))$, ..., $(x_{n+1}, p(x_{n+1}))$ distincts on peut représenter un (et un seul) polynôme de degré n de la forme :

$$p(x) = a_0 + a_1.x + ... + a_n.x^n$$

Il est facile de calculer a_0, ..., a_n si on connaît les $n + 1$ points : nous disposons de $n + 1$ équations pour $n + 1$ variables inconnues.

Première conclusion...

Polynôme de degré n

Un polynôme de degré n peut-être représenté :

- Soit par un vecteur de degrés $(a_0, ..., a_n)$, c'est-à-dire

$$p(x) = a_0 + a_1.x + ... + a_n.x^n$$

Première conclusion...

Polynôme de degré n

Un polynôme de degré n peut-être représenté :

- Soit par un vecteur de degrés (a_0, ..., a_n), c'est-à-dire

$$p(x) = a_0 + a_1.x + ... + a_n.x^n$$

- Soit par (n+1) points de coordonnées distinctes :

$$(x_0, p(x_0)), ..., (x_{n+1}, p(x_{n+1}))$$

Première conclusion...

Polynôme de degré n

Un polynôme de degré n peut-être représenté :

- Soit par un vecteur de degrés (a_0, ..., a_n), c'est-à-dire

$$p(x) = a_0 + a_1.x + ... + a_n.x^n$$

- Soit par (n+1) points de coordonnées distinctes :

$$(x_0, p(x_0)), ..., (x_{n+1}, p(x_{n+1}))$$

- Ces deux représentations sont équivalentes

Deuxième conclusion...

Impact sur la complexité ...

- Comme les deux représentations sont équivalentes, si nous disposons d'algorithmes efficaces pour la représentation à partir de coordonnées distinctes alors nous disposerons également d'algorithmes efficaces pour les polynômes représentés par des vecteurs de degrés

Deuxième conclusion...

Impact sur la complexité ...

- Comme les deux représentations sont équivalentes, si nous disposons d'algorithmes efficaces pour la représentation à partir de coordonnées distinctes alors nous disposerons également d'algorithmes efficaces pour les polynômes représentés par des vecteurs de degrés
- Enfin presque ça ... Car les transformations doivent rester efficaces ...

Addition de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ et $p_2(x)$ deux polynômes représentés par

$$(x_0, p_1(x_0)), ..., (x_{n+1}, p_1(x_{n+1}))$$

Addition de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ et $p_2(x)$ deux polynômes représentés par

$$(x_0, p_1(x_0)), ..., (x_{n+1}, p_1(x_{n+1}))$$

et

Addition de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ et $p_2(x)$ deux polynômes représentés par

$$(x_0, p_1(x_0)), ..., (x_{n+1}, p_1(x_{n+1}))$$

et

$$(x_0, p_2(x_0)), ..., (x_{n+1}, p_2(x_{n+1}))$$

Addition de deux polynômes

Il se fait en $O(n)$. En effet, le polynôme $p_1(x) + p_2(x)$ sera tout simplement représenté par :

Addition de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ et $p_2(x)$ deux polynômes représentés par

$$(x_0, p_1(x_0)), \ldots, (x_{n+1}, p_1(x_{n+1}))$$

et

$$(x_0, p_2(x_0)), \ldots, (x_{n+1}, p_2(x_{n+1}))$$

Addition de deux polynômes

Il se fait en $O(n)$. En effet, le polynôme $p_1(x) + p_2(x)$ sera tout simplement représenté par :

$$(x_0, p_1(x_0) + p_2(x_0)), \ldots, (x_{n+1}, p_1(x_{n+1}) + p_2(x_{n+1}))$$

Produit de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ et $p_2(x)$ deux polynômes représentés par

$$(x_0, p_1(x_0)), ..., (x_{n+1}, p_1(x_{n+1}))$$

et

$$(x_0, p_2(x_0)), ..., (x_{n+1}, p_2(x_{n+1}))$$

Produit de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ et $p_2(x)$ deux polynômes représentés par

$$(x_0, p_1(x_0)), ..., (x_{n+1}, p_1(x_{n+1}))$$

et

$$(x_0, p_2(x_0)), ..., (x_{n+1}, p_2(x_{n+1}))$$

Produit de deux polynômes

Il se fait en $O(n)$. En effet, le polynôme $p_1(x) * p_2(x)$ sera tout simplement représenté par :

$$(x_0, p_1(x_0) * p_2(x_0)), ..., (x_{n+1}, p_1(x_{n+1}) * p_2(x_{n+1}))$$

Evaluationt de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ un polynôme représenté par

$$(x_0, p_1(x_0)), ..., (x_{n+1}, p_1(x_{n+1}))$$

Evaluationt de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ un polynôme représenté par

$$(x_0, p_1(x_0)), ..., (x_{n+1}, p_1(x_{n+1}))$$

Evaluer en x=a

- Aie ... Calculer $p_1(a)$ nécessite d'abord de transformer la représentation à base de points vers une représentation à base de coefficients, puis appliquer l'algorithme de Horner

Evaluationt de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ un polynôme représenté par

$$(x_0, p_1(x_0)), \ldots, (x_{n+1}, p_1(x_{n+1}))$$

Evaluer en x=a

- Aie ... Calculer $p_1(a)$ nécessite d'abord de transformer la représentation à base de points vers une représentation à base de coefficients, puis appliquer l'algorithme de Horner
- La transformation naïve d'une représentation à base de points vers une représentation à base de coefficients se fait en $O(n^2)$

Evaluationt de polynômes à base de points ...

Supposons que les polynômes sont représentés à base de points (coordonnées). Soit $p_1(x)$ un polynôme représenté par

$$(x_0, p_1(x_0)), ..., (x_{n+1}, p_1(x_{n+1}))$$

Evaluer en x=a

- Aie ... Calculer $p_1(a)$ nécessite d'abord de transformer la représentation à base de points vers une représentation à base de coefficients, puis appliquer l'algorithme de Horner
- La transformation naïve d'une représentation à base de points vers une représentation à base de coefficients se fait en $O(n^2)$

Récapitulons

Le tableau suivant donne les complexités des représentations à base de :

Récapitulons

Le tableau suivant donne les complexités des représentations à base de :

	points	coefficients
Addition	$O(n)$	$O(n)$
Multiplication	$O(n)$	$O(n^2)$
Evaluation	$O(n^2)$	$O(n)$

Récapitulons

Le tableau suivant donne les complexités des représentations à base de :

	points	coefficients
Addition	$O(n)$	$O(n)$
Multiplication	$O(n)$	$O(n^2)$
Evaluation	$O(n^2)$	$O(n)$

Il suffit alors de chercher des transformations efficaces pour avoir des complexités inférieurs à $O(n^2)$ pour le produit et l'évaluation des polynômes.

Des coefficients vers des points

Point de départ

Rappelons que la donnée est un polynôme donné sous forme de coefficients, c'est-à-dire :

$$p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n$$

Des coefficients vers des points

Point de départ

Rappelons que la donnée est un polynôme donné sous forme de coefficients, c'est-à-dire :

$$p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n$$

But

Le but est étant donné

$$b_0, \ldots, b_{n+1} \text{ (tous distincts)}$$

de calculer

Des coefficients vers des points

Point de départ

Rappelons que la donnée est un polynôme donné sous forme de coefficients, c'est-à-dire :

$$p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n$$

But

Le but est étant donné

$$b_0, \ldots, b_{n+1} \text{ (tous distincts)}$$

de calculer

$$p(b_0), \ldots, p(b_{n+1})$$

Des coefficients vers des points

Point de départ

Rappelons que la donnée est un polynôme donné sous forme de coefficients, c'est-à-dire :

$$p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n$$

But

Le but est étant donné

$$b_0, \ldots, b_{n+1} \text{ (tous distincts)}$$

de calculer

$$p(b_0), \ldots, p(b_{n+1})$$

Ce qui donnerait une représentation du polynôme $p(x)$ à base des points suivants :

$$(b_0, p(b_0)), \ldots, (b_{n+1}, p(b_{n+1})).$$

Des coefficients vers des points

Intérêt d'une telle transformation

- La multiplication des polynômes à base de coefficients se fait en $O(n^2)$

Des coefficients vers des points

Intérêt d'une telle transformation

- La multiplication des polynômes à base de coefficients se fait en $O(n^2)$
- La multiplication des polynômes à base de points se fait en $O(n)$

Des coefficients vers des points

Intérêt d'une telle transformation

- La multiplication des polynômes à base de coefficients se fait en $O(n^2)$
- La multiplication des polynômes à base de points se fait en $O(n)$
- Toute transformation, en moins de $O(n^2)$, de polynômes à base de coefficients vers des polynômes à base de points, améliorerait le calcul du produit de polynômes à base de coefficients.

Des coefficients vers des points

Un algorithme naïf

Un algorithme naïf consiste simplement à appliquer l'algorithme Horner pour calcul $p(b_i)$ pour chaque chaque $b_i \in \{b_0, ..., b_{n+1}\}$.

Des coefficients vers des points

Un algorithme naïf

Un algorithme naïf consiste simplement à appliquer l'algorithme Horner pour calcul $p(b_i)$ pour chaque chaque $b_i \in \{b_0, ..., b_{n+1}\}$.

Question

Quel est la complexité de cet algorithme?

Des coefficients vers des points

Un algorithme naïf

Un algorithme naïf consiste simplement à appliquer l'algorithme Horner pour calcul $p(b_i)$ pour chaque chaque $b_i \in \{b_0, ..., b_{n+1}\}$.

Des coefficients vers des points

Un algorithme naïf

Un algorithme naïf consiste simplement à appliquer l'algorithme Horner pour calcul $p(b_i)$ pour chaque chaque $b_i \in \{b_0, ..., b_{n+1}\}$.

Réponse

- Comme l'algorithme de Horner est en $O(n)$ pour chaque chaque $b_i \in \{b_0, ..., b_{n+1}\}$, le coût total de l'algorithme naïf de transformation est de $O(n^2)$.

Des coefficients vers des points

Un algorithme naïf

Un algorithme naïf consiste simplement à appliquer l'algorithme Horner pour calcul $p(b_i)$ pour chaque chaque $b_i \in \{b_0, ..., b_{n+1}\}$.

Réponse

- Comme l'algorithme de Horner est en $O(n)$ pour chaque chaque $b_i \in \{b_0, ..., b_{n+1}\}$, le coût total de l'algorithme naïf de transformation est de $O(n^2)$.
- Ce résultat n'est pas très intéressant par rapport à notre objectif (multiplier deux polynômes à base de coefficients en moins de $O(n^2)$).

Des coefficients vers des points

Question

Peut-on faire mieux?

Des coefficients vers des points

Question

Peut-on faire mieux?

Intuitions

- Choisir des b_i particuliers qui permettrait de factoriser un certain nombre de calculs et d'avoir ainsi un algorithme de transformation en moins de $O(n^2)$.

Des coefficients vers des points

Commençons simplement

Supposons que nous avons un polynôme de degré 7 :

$$p(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

Des coefficients vers des points

Commençons simplement

Supposons que nous avons un polynôme de degré 7 :

$$p(x) = a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7$$

Décomposition en deux polynômes

Définissions deux polynômes appelés :

Des coefficients vers des points

Commençons simplement

Supposons que nous avons un polynôme de degré 7 :

$$p(x) = a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7$$

Décomposition en deux polynômes

Définissions deux polynômes appelés :

- Un polynôme, appelé polynôme pair, composé uniquement des coefficients pairs de $p(x)$, c'est-à-dire :

$$p_{\text{pair}}(x) = a_0 + a_2 x + a_4 x^2 + a_6 x^3$$

Des coefficients vers des points

Commençons simplement

Supposons que nous avons un polynôme de degré 7 :

$$p(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

Décomposition en deux polynômes

Définissions deux polynômes appelés :

- Un polynôme, appelé polynôme pair, composé uniquement des coefficients pairs de $p(x)$, c'est-à-dire :

$$p_{\text{pair}}(x) = a_0 + a_2x + a_4x^2 + a_6x^3$$

- Un polynôme, appelé polynôme impair, composé uniquement des coefficients impairs de $p(x)$, c'est-à-dire :

$$p_{\text{impair}}(x) = a_1 + a_3x + a_5x^2 + a_7x^3$$

Des coefficients vers des points

Questions

- Définir $p(x)$ en $p_{\text{pair}}(x)$ et $p_{\text{impair}}(x)$?

Des coefficients vers des points

Questions

- Définir $p(x)$ en $p_{\text{pair}}(x)$ et $p_{\text{impair}}(x)$?
- Ecrire une fonction qui à partir de $p(x)$ calcule $p_{\text{pair}}(x)$ et $p_{\text{impair}}(x)$

Des coefficients vers des points

Questions

- Définir $p(x)$ en $p_{\text{pair}}(x)$ et $p_{\text{impair}}(x)$?
- Ecrire une fonction qui à partir de $p(x)$ calcule $p_{\text{pair}}(x)$ et $p_{\text{impair}}(x)$
- Evaluer sa complexité.

Fonction : extraction du polynôme paire

```c
void recuperer_pair(int N, float p[], float resultat[])
{
  int i, j=0;
  for (i=0; i<N; i=i+2)
  {
    resultat[j]=p[i];
    j++;
  }
}
```

Fonction : extraction du polynôme impaire

```c
void recuperer_impair(int N, float p[], float resultat[])
{
  int i, j=0;
  for (i=1; i<N; i=i+2) {
    resultat[j]=p[i];
    j++;
  }
}
```

Des coefficients vers des points

Questions

Définir $p(x)$ en $p_{\text{pair}}(x)$ et $p_{\text{impair}}(x)$?

Des coefficients vers des points

Questions

Définir $p(x)$ en $p_{\text{pair}}(x)$ et $p_{\text{impair}}(x)$?

Réponse

$$p(x) = p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2).$$

Des coefficients vers des points

Questions

Définir $p(x)$ en $p_{\text{pair}}(x)$ et $p_{\text{impair}}(x)$?

Réponse

$$p(x) = p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2).$$

Vérification de ...

$$p(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

avec

$$p_{\text{pair}}(x) = a_0 + a_2x + a_4x^2 + a_6x^3$$

et

$$p_{\text{impair}}(x) = a_1 + a_3x + a_5x^2 + a_7x^3$$

Des coefficients vers des points

Remarque

- Si on observe : $p_{\text{pair}}(x^2)$ et $p_{\text{impair}}(x^2)$, On remarque que les degrés (non nuls) sont tous de la forme : x^2, x^4,x^6, ...,$x^{2*\frac{n}{2}}$.

Des coefficients vers des points

Remarque

- Si on observe : $p_{\text{pair}}(x^2)$ et $p_{\text{impair}}(x^2)$, On remarque que les degrés (non nuls) sont tous de la forme : x^2, x^4,x^6, ...,$x^{2*\frac{n}{2}}$.
- Ceci est vrai dans les deux polynômes (paire et impair)

Des coefficients vers des points

Remarque

- Si on observe : $p_{\text{pair}}(x^2)$ et $p_{\text{impair}}(x^2)$, On remarque que les degrés (non nuls) sont tous de la forme : x^2, x^4,x^6, ...,$x^{2*\frac{n}{2}}$.
- Ceci est vrai dans les deux polynômes (paire et impair)
- C'est-à-dire que les degrés sont de la forme : $x^{2.j}$ où $j = 1, ..., \frac{n}{2}$.

Des coefficients vers des points

Remarque

- Si on observe : $p_{\text{pair}}(x^2)$ et $p_{\text{impair}}(x^2)$, On remarque que les degrés (non nuls) sont tous de la forme : x^2, x^4,x^6, ...,$x^{2*\frac{n}{2}}$.
- Ceci est vrai dans les deux polynômes (paire et impair)
- C'est-à-dire que les degrés sont de la forme : $x^{2.j}$ où $j = 1, ..., \frac{n}{2}$.

Question

Peut-on exploiter cette propriété?

Des coefficients vers des points

Rappel de notre objectif

- Ne perdons pas de vue notre objectif : Il consiste à générer $n+1$ points distincts depuis $p(x)$ avec une complexité plus petite que n^2.

Des coefficients vers des points

Rappel de notre objectif

- Ne perdons pas de vue notre objectif : Il consiste à générer $n+1$ points distincts depuis $p(x)$ avec une complexité plus petite que n^2.
- Le fait que les degrés sont de la forme : $x^{2.j}$ où $j = 1, \ldots, \frac{n}{2}$. suggère que :

Des coefficients vers des points

Rappel de notre objectif

- Ne perdons pas de vue notre objectif : Il consiste à générer $n+1$ points distincts depuis $p(x)$ avec une complexité plus petite que n^2.
- Le fait que les degrés sont de la forme : $x^{2.j}$ où $j = 1, ..., \frac{n}{2}$. suggère que :
  - Si on évalue $p(b)$ alors $p(-b)$ se calcule efficacement.

Des coefficients vers des points

Rappel de notre objectif

- Ne perdons pas de vue notre objectif : Il consiste à générer $n+1$ points distincts depuis $p(x)$ avec une complexité plus petite que n^2.
- Le fait que les degrés sont de la forme : $x^{2.j}$ où $j = 1, ..., \frac{n}{2}$. suggère que :
  - Si on évalue $p(b)$ alors $p(-b)$ se calcule efficacement.
  - La raison intuitive est que : $b^2 = (-b)^2$!

Des coefficients vers des points

Reprenons notre exemple

$$
\begin{aligned}
p(x) &= a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7 \\
&= p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2) \\
&= (a_0 + a_2 x^2 + a_4 x^4 + a_6 x^6) + x * (a_1 + a_3 x^2 + a_5 x^4 + a_7 x^6)
\end{aligned}
$$

Des coefficients vers des points

Reprenons notre exemple

$$\begin{aligned} p(x) &= a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 \\ &= p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2) \\ &= (a_0 + a_2x^2 + a_4x^4 + a_6x^6) + x * (a_1 + a_3x^2 + a_5x^4 + a_7x^6) \end{aligned}$$

Evaluation de 8 points

Supposons que les points à évaluer sont :

$$b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8$$

Cette décomposition d'un polynôme en deux sous-polynômes nous permet d'évaluer uniquement b_1, b_2, b_3, b_4 si on prend :

Des coefficients vers des points

Reprenons notre exemple

$$\begin{aligned} p(x) &= a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7 \\ &= p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2) \\ &= (a_0 + a_2 x^2 + a_4 x^4 + a_6 x^6) + x * (a_1 + a_3 x^2 + a_5 x^4 + a_7 x^6) \end{aligned}$$

Evaluation de 8 points

Supposons que les points à évaluer sont :

$$b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8$$

Cette décomposition d'un polynôme en deux sous-polynômes nous permet d'évaluer uniquement b_1, b_2, b_3, b_4 si on prend :

$$b_5 = -b_1, \quad b_6 = -b_2, \quad b_7 = -b_3, \quad b_8 = -b_4$$

Des coefficients vers des points

Reprenons notre exemple

$$\begin{aligned} p(x) &= a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 \\ &= p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2) \\ &= (a_0 + a_2x^2 + a_4x^4 + a_6x^6) + x * (a_1 + a_3x^2 + a_5x^4 + a_7x^6) \end{aligned}$$

Des coefficients vers des points

Reprenons notre exemple

$$\begin{aligned} p(x) &= a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7 \\ &= p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2) \\ &= (a_0 + a_2 x^2 + a_4 x^4 + a_6 x^6) + x * (a_1 + a_3 x^2 + a_5 x^4 + a_7 x^6) \end{aligned}$$

En effet :

Il est facile de vérifier que :

$$p_{\text{pair}}(b_1^2) = p_{\text{pair}}((-b_1)^2)$$

Des coefficients vers des points

Reprenons notre exemple

$$\begin{aligned} p(x) &= a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7 \\ &= p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2) \\ &= (a_0 + a_2 x^2 + a_4 x^4 + a_6 x^6) + x * (a_1 + a_3 x^2 + a_5 x^4 + a_7 x^6) \end{aligned}$$

En effet :

Il est facile de vérifier que :

$$p_{\text{pair}}(b_1^2) = p_{\text{pair}}((-b_1)^2)$$

et

$$p_{\text{impair}}(b_1^2) = p_{\text{impair}}((-b_1)^2)$$

Des coefficients vers des points

Reprenons notre exemple

$$\begin{aligned} p(x) &= a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7 \\ &= p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2) \\ &= (a_0 + a_2 x^2 + a_4 x^4 + a_6 x^6) + x * (a_1 + a_3 x^2 + a_5 x^4 + a_7 x^6) \end{aligned}$$

En effet :

Il est facile de vérifier que :

$$p_{\text{pair}}(b_1^2) = p_{\text{pair}}((-b_1)^2)$$

et

$$p_{\text{impair}}(b_1^2) = p_{\text{impair}}((-b_1)^2)$$

Idem pour b_2, b_3, b_4.

Des coefficients vers des points

A ce stade de calcul ...

Un polynôme $p(x)$ de degré n s'écrit comme une combinaison de deux sous polynômes de degré $\frac{n}{2}$:

$$p(x) = p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2)$$

Des coefficients vers des points

A ce stade de calcul ...

Un polynôme $p(x)$ de degré n s'écrit comme une combinaison de deux sous polynômes de degré $\frac{n}{2}$:

$$p(x) = p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2)$$

Pour évaluer $p(x)$ avec les valeurs $\{b_1, ..., b_{\frac{n}{2}}, -b_1, ..., -b_{\frac{n}{2}}\}$, il suffit :

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.

Des coefficients vers des points

A ce stade de calcul ...

Un polynôme $p(x)$ de degré n s'écrit comme une combinaison de deux sous polynômes de degré $\frac{n}{2}$:

$$p(x) = p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2)$$

Pour évaluer $p(x)$ avec les valeurs $\{b_1, ..., b_{\frac{n}{2}}, -b_1, ..., -b_{\frac{n}{2}}\}$, il suffit :

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer :

Des coefficients vers des points

A ce stade de calcul ...

Un polynôme $p(x)$ de degré n s'écrit comme une combinaison de deux sous polynômes de degré $\frac{n}{2}$:

$$p(x) = p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2)$$

Pour évaluer $p(x)$ avec les valeurs $\{b_1, ..., b_{\frac{n}{2}}, -b_1, ..., -b_{\frac{n}{2}}\}$, il suffit :

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer :

$$p(b_i) = p_{\text{pair}}(b_i^2) + b_i * p_{\text{impair}}(b_i^2), \text{ et}$$

$$p(-b_i) = p_{\text{pair}}(b_i^2) - b_i * p_{\text{impair}}(b_i^2)$$

Des coefficients vers des points

A ce stade de calcul ...

Un polynôme $p(x)$ de degré n s'écrit comme une combinaison de deux sous polynômes de degré $\frac{n}{2}$:

$$p(x) = p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2)$$

Pour évaluer $p(x)$ avec les valeurs $\{b_1, ..., b_{\frac{n}{2}}, -b_1, ..., -b_{\frac{n}{2}}\}$, il suffit :

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer :

$$p(b_i) = p_{\text{pair}}(b_i^2) + b_i * p_{\text{impair}}(b_i^2), \text{ et}$$

$$p(-b_i) = p_{\text{pair}}(b_i^2) - b_i * p_{\text{impair}}(b_i^2)$$

Des coefficients vers des points

A ce stade de calcul ...

Un polynôme $p(x)$ de degré n s'écrit comme une combinaison de deux sous polynômes de degré $\frac{n}{2}$:

$$p(x) = p_{\text{pair}}(x^2) + x * p_{\text{impair}}(x^2)$$

Pour évaluer $p(x)$ avec les valeurs $\{b_1, ..., b_{\frac{n}{2}}, -b_1, ..., -b_{\frac{n}{2}}\}$, il suffit :

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer :

$$p(b_i) = p_{\text{pair}}(b_i^2) + b_i * p_{\text{impair}}(b_i^2), \text{ et}$$

$$p(-b_i) = p_{\text{pair}}(b_i^2) - b_i * p_{\text{impair}}(b_i^2)$$

Question

A-t-on gagné en complexité?

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Réponse

- Si on utilise l'algorithme de Horner pour évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$, la première étape coûterait :

$$\frac{n}{2} * 2 * \frac{n}{2} = \frac{n^2}{2}$$

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Réponse

- Si on utilise l'algorithme de Horner pour évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$, la première étape coûterait :

$$\frac{n}{2} * 2 * \frac{n}{2} = \frac{n^2}{2}$$

- la deuxième étape coûte : $\frac{n}{2} + \frac{n}{2} = n$

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Réponse

- Si on utilise l'algorithme de Horner pour évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$, la première étape coûterait :

$$\frac{n}{2} * 2 * \frac{n}{2} = \frac{n^2}{2}$$

- la deuxième étape coûte : $\frac{n}{2} + \frac{n}{2} = n$
- Donc on obtient une complexité de $O(\frac{n^2}{2} + n)$ au lieu $O(n^2)$ (hum...)

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Réponse

- Pour espérer une complexité plus petite, il ne faut pas utiliser l'algorithme de Horner pour évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Réponse

- Pour espérer une complexité plus petite, il ne faut pas utiliser l'algorithme de Horner pour évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
- Il faut ré-itérer le même algorithme, c'est-à-dire :
  - ré-appliquer de manière récursive le processus de décomposition des deux polynômes $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$,

Des coefficients vers des points

A ce stade de calcul ...

1. Pour chaque $i = 1, .., \frac{n}{2}$, d'évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
2. Pour chaque $i = 1, .., \frac{n}{2}$, de calculer : $p(b_i)$ et $p(-b_i)$.

Réponse

- Pour espérer une complexité plus petite, il ne faut pas utiliser l'algorithme de Horner pour évaluer $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$.
- Il faut ré-itérer le même algorithme, c'est-à-dire :
  - ré-appliquer de manière récursive le processus de décomposition des deux polynômes $p_{\text{pair}}(b_i^2)$ et $p_{\text{impair}}(b_i^2)$,
  - jusqu'à atteindre le cas de base (un polynôme de degré 0).

Des coefficients vers des points

Continuons notre exemple

$$p(x) \quad = \quad (a_0 + a_2x^2 + a_4x^4 + a_6x^6) + x * (a_1 + a_3x^2 + a_5x^2 + a_7x^6)$$

Des coefficients vers des points

Continuons notre exemple

$$p(x) = (a_0 + a_2x^2 + a_4x^4 + a_6x^6) + x * (a_1 + a_3x^2 + a_5x^2 + a_7x^6)$$

Décomposition récursive

Décomposons de nouveau :

$$\begin{aligned} p_{\text{pair}}(x^2) &= (a_0 + a_2x^2 + a_4x^4 + a_6x^6) \\ &= (a_0 + a_4x^4) + x^2(a_2 + a_6x^4)) \\ &= Q_{\text{pair}}(x^4) + x^2 Q_{\text{impair}}(x^4) \end{aligned}$$

Continuons notre exemple

Evaluation du sous-polynôme

Notre but est d'évaluer $p_{\text{pair}}(x^2)$ sur les valeurs b_1, b_2, b_3, b_4

$$\begin{aligned}
p_{\text{pair}}(x^2) &= (a_0 + a_2x^2 + a_4x^4 + a_6x^6) \\
&= (a_0 + a_4x^4) + x^2(a_2 + a_6x^4)) \\
&= Q_{\text{pair}}(x^4) + x^2 Q_{\text{impair}}(x^4)
\end{aligned}$$

Continuons notre exemple

Evaluation du sous-polynôme

Notre but est d'évaluer $p_{\text{pair}}(x^2)$ sur les valeurs b_1, b_2, b_3, b_4

$$\begin{aligned} p_{\text{pair}}(x^2) &= (a_0 + a_2x^2 + a_4x^4 + a_6x^6) \\ &= (a_0 + a_4x^4) + x^2(a_2 + a_6x^4)) \\ &= Q_{\text{pair}}(x^4) + x^2 Q_{\text{impair}}(x^4) \end{aligned}$$

Si on reprend le même principe, il faut juste évaluer b_1, b_2 et choisir $b_3 = -b1$ et $b_4 = -b2$, car:

$$(b_1)^4 = (b_3)^4, \qquad (b_2)^4 = (b_4)^4$$

Des coefficients vers des points

Problème

Si on reprend le même principe, il faut juste évaluer b_1, b_2 et choisir $b_3 = -b1$ et $b_4 = -b2$, car:

$$(b_1)^4 = (b_3)^4 \text{ et } (b_2)^4 = (b_4)^4$$

Des coefficients vers des points

Problème

Si on reprend le même principe, il faut juste évaluer b_1, b_2 et choisir $b_3 = -b1$ et $b_4 = -b2$, car:

$$(b_1)^4 = (b_3)^4 \text{ et } (b_2)^4 = (b_4)^4$$

Le problème est que l'on avait déjà posé que :

$$b_5 = -b_1 \text{ et } b_6 = -b_2 \text{ car } b_5^2 = -(b_1)^2 \text{ et } b_6^2 = (-b_2)^2$$

Des coefficients vers des points

Problème

Si on reprend le même principe, il faut juste évaluer b_1, b_2 et choisir $b_3 = -b1$ et $b_4 = -b2$, car:

$$(b_1)^4 = (b_3)^4 \text{ et } (b_2)^4 = (b_4)^4$$

Le problème est que l'on avait déjà posé que :

$$b_5 = -b_1 \text{ et } b_6 = -b_2 \text{ car } b_5^2 = -(b_1)^2 \text{ et } b_6^2 = (-b_2)^2$$

Or le principe de la représentation des polynômes par les points, les valeurs doivent être toutes différentes!!!!!!

Ici :

$$b_5 = b_3 = -b_1$$

Des coefficients vers des points

Comment contourner le problème?

Au fait, si impose : $b_5 = -b_1$ et que b_3 et b_5 soient différents, une solution serait de ne pas travailler avec des réels avec des nombres complexes.

Des coefficients vers des points

Comment contourner le problème?

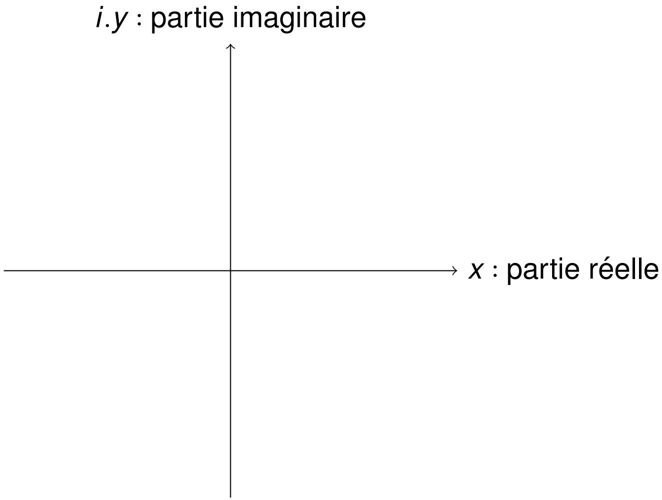
- Avec l'introduction des nombres imaginaires, une équation de la forme :

  $$x^4 = 1$$

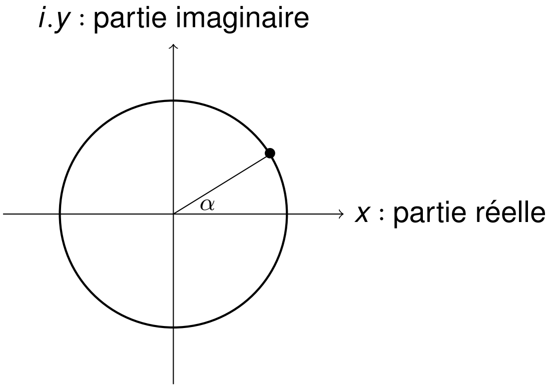
  admet 4 solutions imaginaire :

  - $x = 1$
  - $x = -1$
  - $x = i$
  - $x = -i$

Représentation graphique d'un point imaginaire



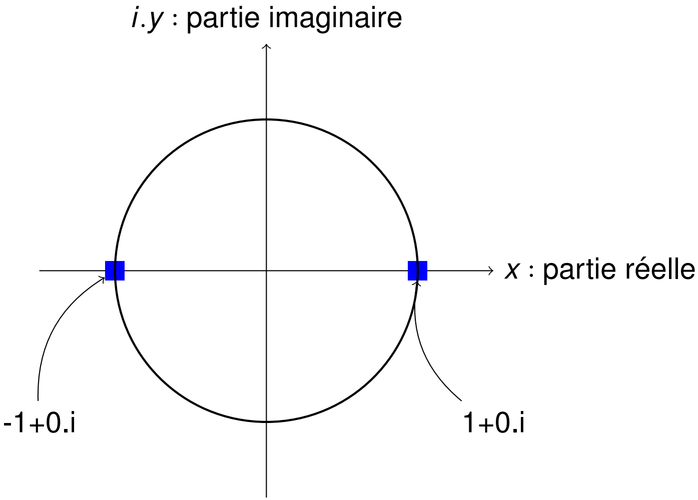
Représentation graphique d'un point imaginaire



Dans la suite, on s'intéresse aux nombre complexes dont les coordonnées se trouvent autour d'un cercle de rayon 1.
Au fait, chaque nombre complexe *A* est de la forme :

$$A = cos(\alpha) + i * sin(\alpha)$$

Exemples : N=2



Des coefficients vers des points

Comment contourner le problème?

- Avec l'introduction, des nombres imaginaires, une équation de la forme :
  $x^4 = 1$
  admet 4 solutions imaginaires :

Des coefficients vers des points

Comment contourner le problème?

- Avec l'introduction, des nombres imaginaires, une équation de la forme :
  $x^4 = 1$
  admet 4 solutions imaginaires :
  - $x = 1$

Des coefficients vers des points

Comment contourner le problème?

- Avec l'introduction, des nombres imaginaires, une équation de la forme :
  $x^4 = 1$
  admet 4 solutions imaginaires :
  - $x = 1$
  - $x = -1$

Des coefficients vers des points

Comment contourner le problème?

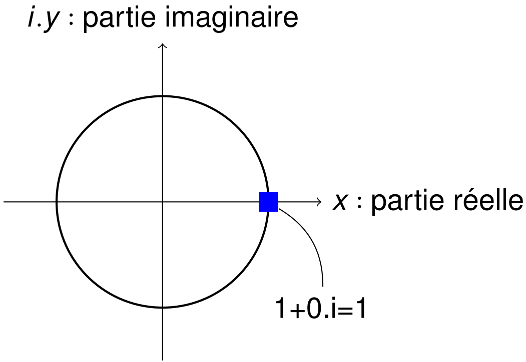
- Avec l'introduction, des nombres imaginaires, une équation de la forme :
  $x^4 = 1$
  admet 4 solutions imaginaires :
  - $x = 1$
  - $x = -1$
  - $x = i$

Des coefficients vers des points

Comment contourner le problème?

- Avec l'introduction, des nombres imaginaires, une équation de la forme :
  $x^4 = 1$
  admet 4 solutions imaginaires :
    - $x = 1$
    - $x = -1$
    - $x = i$
    - $x = -i$

Exemples : N=4

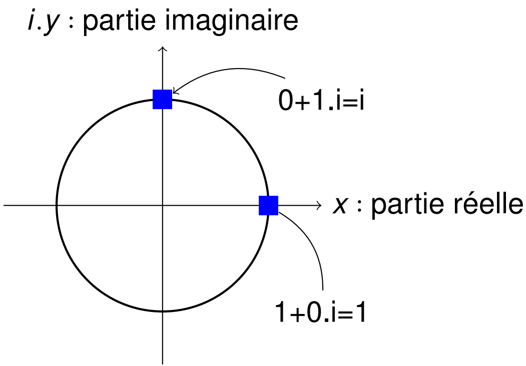


Eléments sur le circle

Les éléments sur le cercle unitaire (rayon = 1) ont des angles multiples de $\left(\frac{2*\pi}{4}\right)$, dans notre cas :

$$\left(\frac{2*\pi}{4}\right), 2*\left(\frac{2*\pi}{4}\right), 3*\left(\frac{2*\pi}{4}\right), 4*\left(\frac{2*\pi}{4}\right)$$

Exemples : N=4

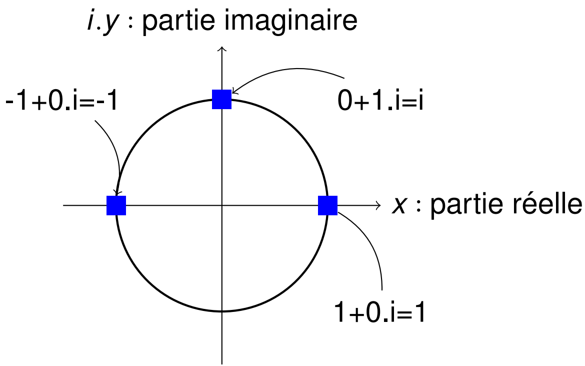


Eléments sur le circle

Les éléments sur le cercle unitaire (rayon = 1) ont des angles multiples de $\left(\frac{2*\pi}{4}\right)$, dans notre cas :

$$\left(\frac{2*\pi}{4}\right), 2*\left(\frac{2*\pi}{4}\right), 3*\left(\frac{2*\pi}{4}\right), 4*\left(\frac{2*\pi}{4}\right)$$

Exemples : N=4

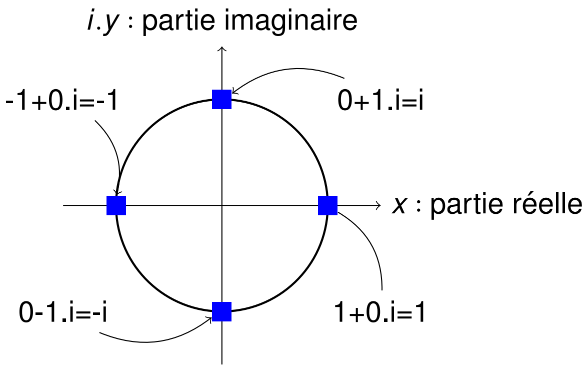


Eléments sur le circle

Les éléments sur le cercle unitaire (rayon = 1) ont des angles multiples de $\left(\frac{2*\pi}{4}\right)$, dans notre cas :

$$\left(\frac{2*\pi}{4}\right), 2*\left(\frac{2*\pi}{4}\right), 3*\left(\frac{2*\pi}{4}\right), 4*\left(\frac{2*\pi}{4}\right)$$

Exemples : N=4



Eléments sur le circle

Les éléments sur le cercle unitaire (rayon = 1) ont des angles multiples de $\left(\frac{2*\pi}{4}\right)$, dans notre cas :

$$\left(\frac{2*\pi}{4}\right), 2*\left(\frac{2*\pi}{4}\right), 3*\left(\frac{2*\pi}{4}\right), 4*\left(\frac{2*\pi}{4}\right)$$

Répartir les points sur un cercle unitaire

Nombres complexes sur un cercle unitaire

- Si on doit évaluer un polynôme de degré (n-1), les nombres complexes choisies pour être évalués sont ceux qui sont répartis de manière uniforme sur le cercle unitaire (de rayon 1)

Répartir les points sur un cercle unitaire

Nombres complexes sur un cercle unitaire

- C'est-à-dire, soit N le nombre de points à évaluer.
  - Soit $\alpha = \frac{2*\pi}{N}$ et $\omega_N = cos(\alpha) + i * sin(\alpha)$

Répartir les points sur un cercle unitaire

Nombres complexes sur un cercle unitaire

- C'est-à-dire, soit N le nombre de points à évaluer.
  - Soit $\alpha = \frac{2 * \pi}{N}$ et $\omega_N = cos(\alpha) + i * sin(\alpha)$
  - Les N points ont des angles : $\alpha, 2 * \alpha, ..., N * \alpha$

Répartir les points sur un cercle unitaire

Nombres complexes sur un cercle unitaire

- C'est-à-dire, soit N le nombre de points à évaluer.
  - Soit $\alpha = \frac{2*\pi}{N}$ et $\omega_N = cos(\alpha) + i * sin(\alpha)$
  - Les N points ont des angles : $\alpha, 2 * \alpha, ..., N * \alpha$
  - Remarque :

$$\begin{aligned}
(\omega_N)^2 &= (cos(\alpha) + i * sin(\alpha))^2 \\
&= \left(cos(\alpha)^2 - sin(\alpha)^2\right) + i * (2 * cos(\alpha) * sin(\alpha)) \\
&= cos(2 * \alpha) + i * cos(2 * \alpha)
\end{aligned}$$

Des coefficients vers des points

Racine n$^{\text{ème}}$ de 1

- Si on doit évaluer un polynôme de degré (n-1), les valeurs choisies pour être évaluées sont les solutions de l'équation :

$$x^{n-1} = 1$$

  les x solutions sont appelées racines n$^{\text{ème}}$ de 1.

Des coefficients vers des points

Racine $n^{\text{ème}}$ de 1

- Si on doit évaluer un polynôme de degré (n-1), les valeurs choisies pour être évaluées sont les solutions de l'équation :

$$x^{n-1} = 1$$

les x solutions sont appelées racines $n^{\text{ème}}$ de 1.

  - Les racines $n^{\text{ème}}$ de 1 ont des propriétés très intéressantes

Des coefficients vers des points

Racine n$^{\text{ème}}$ de 1

- Si on doit évaluer un polynôme de degré (n-1), les valeurs choisies pour être évaluées sont les solutions de l'équation :

$$x^{n-1} = 1$$

les x solutions sont appelées racines n$^{\text{ème}}$ de 1.

  - Les racines n$^{\text{ème}}$ de 1 ont des propriétés très intéressantes
  - Utilisées dans plusieurs domaines comme la cryptographie

Des coefficients vers des points

Racine n$^{\text{ème}}$ de 1

- Si on doit évaluer un polynôme de degré (n-1), les valeurs choisies pour être évaluées sont les solutions de l'équation :

$$x^{n-1} = 1$$

  les *x* solutions sont appelées racines n$^{\text{ème}}$ de 1.
  - Les racines n$^{\text{ème}}$ de 1 ont des propriétés très intéressantes
  - Utilisées dans plusieurs domaines comme la cryptographie
  - Elles permettent surtout de faire la transformation inverse d'une représentation à base de points vers une représentation à base de coefficients.

Des coefficients vers des points

Racine n$^{\text{ème}}$ de 1

- Rappelons :

$$\omega_N = cos\left(\frac{2 * \pi}{N}\right) + i * cos\left(\frac{2 * \pi}{N}\right)$$

avec N un multiple de 2 (ce qui est le cas avec nos polynômes)

Des coefficients vers des points

Racine n^{ème} de 1

- Rappelons :

$$\omega_N = cos\left(\frac{2 * \pi}{N}\right) + i * cos\left(\frac{2 * \pi}{N}\right)$$

  avec N un multiple de 2 (ce qui est le cas avec nos polynômes)
- Alors, les nombres complexes ω_N, ω_N^2, ..., ω_N^{N-1} sont racines $N^{ème}$ de 1, c'est-à-dire :

$$\forall j = 0, .., N - 1, \left(\omega_N^j\right)^N = 1.$$

Fonction : Evaluation des polynômes

La fonction :

```
nbrecomplexe *ftt (int N, float p[])
```

prend en paramètre :

- *N* : un entier qui représente le nombre de points (nombres complexes) à évaluer. Ces nombres complexes sont au fait les racines N$^{\text{èmes}}$ de 1.
- $p[]$: un polynôme de degré $N-1$ représenté sous forme d'un tableau. Chaque case contient un degré.

Fonction : Evaluation des polynômes

La fonction :

```
nbrecomplexe *ftt (int N, float p[])
```

prend en paramètre :

- N : un entier qui représente le nombre de points (nombres complexes) à évaluer. Ces nombres complexes sont au fait les racines N$^{\text{èmes}}$ de 1.
- $p[]$: un polynôme de degré $N - 1$ représenté sous forme d'un tableau. Chaque case contient un degré.

Et retourne :

- Un tableau (ou un pointeur) contenant N. Chaque case contient l'évaluation de p avec une des racines N$^{\text{èmes}}$ de 1.

Fonction : Evaluation des polynômes

Lorsque $N = 1$:

```
nbrecomplexe *ftt (int N, float p[])
{
  if (N==1)
  {
    resultat_evaluation[0].reelle=p[0];
    resultat_evaluation[0].complexe=0;
  }
```

Alors :

- il s'agit d'un polynôme de degré 0 de la forme $p(x) = a_0$ (a_0 sera stocké dans le tableau p d'indice 0).
- Dans ce cas, quelque soit la valeur à évaluer, la fonction retournera toujours a_0
- C'est une condition d'arrêt de la fonction récursive

Fonction : Evaluation des polynômes

Lorsque *N* différent de1 :

```
nbrecomplexe *ftt (int N, float p[])
{
  else  /* N != 1*/
  {
    recuperer_pair (N,p,ppaire);
    recuperer_impair (N,p,pimpaire);
  }
```

la première étape

- consiste à récupérer les deux polynômes : paire et impaire à partir du polynôme initial *p*
- Ces deux polynômes sont stockés dans les tableau : ppaire et pimpaire

Fonction : Evaluation des polynômes

Lorsque N différent de1 :

```
nbrecomplexe *ftt (int N, float p[])
{
  else  /* N != 1*/
  {
    evaluationpaire=ftt(N/2,ppaire);
    evaluationimpaire=ftt(N/2,pimpaire);
  }
```

la deuxième étape

- consiste à ces deux deux polynômes $\frac{N}{2}$ nombres complexese
- Les résultats de l'évaluations sont stockés dans les tableau : evaluationpaire et evaluationimpaire

Fonction : Evaluation des polynômes

Lorsque *N* différent de1 :

```
nbrecomplexe *ftt (int N, float p[])
{
 wn.reelle=cos((2*pi)/N); wn.complexe=sin((2*pi)/N);
 w.reelle=1; w.complexe=0;
 for (i=0; i<N/2; i++)
 {
  resultat_evaluation[i]=addition(evaluationpaire[i],
        produit(evaluationimpaire[i],w));
  resultat_evaluation[i+N/2]=soustraction
      (evaluationpaire[i], produit(evaluationimpaire[i],w)));
  w=produit(w,wn);
 }
 return resultat_evaluation;
}
```

la dernière étape

- Evaluer *p* à partir de deux polynômes evaluationpaire et evaluationimpaire
- On retourne le résultat final

Bonus

Question

Quelle est la complexité de cet algorithme?

Complexité

$$T(n) \quad = \quad 2 * T(n/2) + k * n$$

Complexité

$$
\begin{aligned}
T(n) &= 2 * T(n/2) + k * n \\
&= 2 * \left(2 * T\left(\frac{n}{2^2}\right) + k * \left(\frac{n}{2}\right)\right) + k * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 2 * T(n/2) + k * n \\
&= 2 * \left(2 * T\left(\frac{n}{2^2}\right) + k * \left(\frac{n}{2}\right)\right) + k * n \\
&= 2^2 * T\left(\frac{n}{2^2}\right) + k * n + k * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 2 * T(n/2) + k * n \\
&= 2 * \left(2 * T\left(\frac{n}{2^2}\right) + k * \left(\frac{n}{2}\right)\right) + k * n \\
&= 2^2 * T\left(\frac{n}{2^2}\right) + k * n + k * n \\
&= 2^2 * \left(2 * T\left(\frac{n}{2^3}\right) + k * \left(\frac{n}{2^2}\right)\right) + 2 * k * n
\end{aligned}
$$

Complexité

$$\begin{aligned}
T(n) &= 2 * T(n/2) + k * n \\
&= 2 * \left(2 * T\left(\frac{n}{2^2}\right) + k * \left(\frac{n}{2}\right)\right) + k * n \\
&= 2^2 * T\left(\frac{n}{2^2}\right) + k * n + k * n \\
&= 2^2 * \left(2 * T\left(\frac{n}{2^3}\right) + k * \left(\frac{n}{2^2}\right)\right) + 2 * k * n \\
&= 2^3 * T\left(\frac{n}{2^3}\right) + 3 * k * n
\end{aligned}$$

$$
\begin{aligned}
T(n) &= 2 * T(n/2) + k * n \\
&= 2 * \left(2 * T\left(\frac{n}{2^2}\right) + k * \left(\frac{n}{2}\right)\right) + k * n \\
&= 2^2 * T\left(\frac{n}{2^2}\right) + k * n + k * n \\
&= 2^2 * \left(2 * T\left(\frac{n}{2^3}\right) + k * \left(\frac{n}{2^2}\right)\right) + 2 * k * n \\
&= 2^3 * T\left(\frac{n}{2^3}\right) + 3 * k * n \\
&\;. \\
&\;.
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 2 * T(n/2) + k * n \\
&= 2 * \left(2 * T\left(\frac{n}{2^2}\right) + k * \left(\frac{n}{2}\right)\right) + k * n \\
&= 2^2 * T\left(\frac{n}{2^2}\right) + k * n + k * n \\
&= 2^2 * \left(2 * T\left(\frac{n}{2^3}\right) + k * \left(\frac{n}{2^2}\right)\right) + 2 * k * n \\
&= 2^3 * T\left(\frac{n}{2^3}\right) + 3 * k * n \\
&. \\
&. \\
&= 2^m * T\left(\frac{n}{2^m}\right) + k * m * n
\end{aligned}
$$

Question

Que vaut m?

Réponse

m est tel que : $T\left(\frac{n}{2^m}\right) = T(1)$, c'est à dire :

$$\frac{n}{2^m} = 1.$$

C'est-à-dire : $n = 2^m$.
Appliquons le *log* base (2), ce qui donne:

$$m = log_2(n).$$

Donc : $T(n) \in O(n * log_2(n))$.