



Rapport de projet Algorithme Numérique Interpolation, matrice de Vandermonde

Group ilpro212 :

Coordinateur rapporteur:

BOUBEKKI Amirouche

Architecte integrateur

EL AFRIT Mohamed Amine

Développeurs:

LE MAOUT Yves

MERBETH Gaël

SALLIER Mathilde

L^AT_EX

24 mai 2007

Table des matières

1	Introduction	2
1.1	Interpolation polynomiale : matrices de Vandermonde	2
1.2	Propriété de la matrice de vandermonde	2
1.2.1	Propriété	2
1.2.2	Démonstration	2
1.3	Exemple d'interpolation	3
1.4	Résolution du système	3
1.4.1	Génération de la matrice de Vandermonde	3
1.4.2	Élimination de Gauss	4
1.4.3	Résolution à partir d'une matrice triangulaire	5
1.5	Évaluation du polynôme d'interpolation	5
2	Conclusion	6
3	Annexes	7
3.1	Code source	7

Chapitre 1

Introduction

1.1 Interpolation polynomiale : matrices de Vandermonde

On cherche l'unique polynôme de degré n passant par les points (x_i, y_i) pour $i = 0 \dots n$, en supposant les x_i tous distincts.

Le polynôme d'interpolation peut donc s'écrire

$$p_n(x) = \sum_{i=0}^n a_i x^i \quad (1.1)$$

tel que pour tout i , $p_n(x_i) = y_i$, soit en notation matricielle à l'aide de la matrice de Vandermonde :

$$\begin{pmatrix} 1 & x_0^1 & \dots & x_0^n \\ 1 & x_1^1 & \dots & x_1^n \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_n^1 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \quad (1.2)$$

1.2 Propriété de la matrice de vandermonde

1.2.1 Propriété

On considère une matrice V de Vandermonde carrée. Elle est inversible si et seulement si les x_i sont deux à deux distincts.

Cette propriété va nous permettre de savoir au préalable s'il est possible de trouver le polynome nécessaire en se basant sur les points données.

1.2.2 Démonstration

De façon matricielle, elle se présente ainsi :

$$V_{i,j} = x_i^{j-1} \quad \forall i \text{ et } j$$

$$V = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha_m & \alpha_m^2 & \dots & \alpha_m^{n-1} \end{pmatrix}$$

Si deux coefficients α_i sont identiques, la matrice a deux lignes identiques, donc n'est pas inversible. Pour la réciproque, on peut procéder au calcul du déterminant

Le déterminant d'une matrice $n \times n$ de Vandermonde ($m = n$ dans ce cas) peut s'exprimer ainsi : $\det(V) = \prod_{1 \leq i < j \leq n} (\alpha_j - \alpha_i)$

Une preuve d'inversibilité plus rapide est cependant de considérer V comme la matrice du Système d'équations linéaires : système linéaire homogène " $VX=0$ " pour " X " de composantes x_0, x_1, \dots, x_{n-1}

$$\begin{cases} x_0 + \alpha_1 x_1 + \alpha_1^2 x_2 + \dots + \alpha_1^{n-1} x_{n-1} = 0 \\ \dots \\ x_0 + \alpha_n x_1 + \alpha_n^2 x_2 + \dots + \alpha_n^{n-1} x_{n-1} = 0 \end{cases}$$

Mais en introduisant le polynôme $P = \sum_{i=0}^{n-1} x_i X^i$ On voit que si " X " vérifie l'équation " $VX=0$ ", alors " P " admet " n " racines distinctes, soit plus que son degré. Donc " P " est nul, et ainsi " $X=0$ ". Ce qui prouve que " V " est inversible.

1.3 Exemple d'interpolation

On cherche à déterminer le polynôme de degré 3

$$p_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \tag{1.3}$$

passant par les points $(-2, 10), (-1, 4), (1, 6)$ et $(2, 3)$.

Les a_i peuvent être déterminés par la résolution du système linéaire :

$$\begin{pmatrix} 1 & -2 & 4 & -8 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 4 \\ 6 \\ 3 \end{pmatrix} \tag{1.4}$$

La solution de ce système peut être obtenue de la façon suivante :

```
y = [10;4;6;3];
V= [1,-2,4,-8;1,-1,1,-1;1,1,1,1;1,2,4,8];
a = inv(V)*y;
```

1.4 Résolution du système

1.4.1 Génération de la matrice de Vandermonde

La matrice de Vandermonde V peut être déterminée à partir du vecteur x de la façon suivante : (en se basant sur la formule 1.2.2)

```

function V=cons_vand(x)
n=length(x);
V=zeros(n,n);
for j=1:n
    for i=1:n
        V(i,j)=x(i)^(j-1);
    end
end
endfunction;

```

1.4.2 Élimination de Gauss

L'algorithme de Gauss est une méthode permettant l'élimination de certains coefficients non nuls et de rendre la matrice triangulaire. Le problème se situe au niveau du choix du pivot en effet, ceci peut influencer la "qualité" de la solution obtenue après résolution du système linéaire. Nous avons expérimenté trois stratégies concernant le choix du pivot : stratégie du pivot maximum partiel, stratégie du pivot avec seuil, stratégie du pivot du pivot avec maximum total. Le choix du pivot doit permettre de minimiser les erreurs liées aux arrondis.

Stratégie du pivot maximum partiel

Voir sources 3.1

Cette stratégie consiste à sélectionner le plus grand pivot (en valeur absolue) parmi ceux qui sont dans la colonne où l'on travaille. On permute alors ligne avec la ligne où se trouve le pivot sélectionné.

Stratégie du pivot avec seuil

Voir sources 3.1

Cette stratégie cherche un autre pivot que si le pivot actuel est plus petit qu'un certain seuil. Ce seuil est sélectionné en fonction des capacités de calcul (en terme de précision) de la machine, pour un ordinateur classique on peut choisir 10^{-10} . Cette méthode augmente les problèmes liés aux arrondis. Nous avons rencontré des problèmes dans certains cas :

$$\begin{pmatrix} 10^{-10} & 123456783 \\ 15753 & 1 \end{pmatrix} \quad (1.5)$$

La stratégie du pivot avec seuil ne donne pas la bonne solution alors que les deux autres stratégies le font correctement.

Stratégie du pivot avec maximum total

Voir sources 3.1

Cette méthode recherche le plus grand pivot (en valeur absolue) dans toute la matrice. On permute alors les deux lignes et les deux colonnes. La difficulté étant qu'il faut mémoriser ces permutations pour les repercuter sur les résultats. Cette méthode nous permet de récupérer toujours le plus grand pivot mais ceci nécessite beaucoup de calcul.

1.4.3 Résolution à partir d'une matrice triangulaire

Nous avons utilisé la fonction suivante qui permet de remonter la matrice triangulaire pour déterminer les solutions. Cette fonction suppose que la matrice passée en argument est triangulaire supérieure. Elle procède de la façon suivante :

- Tout d'abord, on prend la dernière ligne, qui nous donne X_n en fonction de A_{nn} et B_n :

$$X_n = \frac{B_n}{A_{nn}}$$

- Puis pour chaque ligne on a :

$$\forall i \in [1, n-1], X_i = \frac{B_i - \sum_{k=i+1}^n A_{ik} X^k}{A_{ii}}$$

1.5 Evaluation du polynôme d'interpolation

Indépendamment de la détermination des coefficients du polynôme d'interpolation, on considère dans cette section l'évaluation d'un polynôme $p_n(x) = a_0 + \dots + a_n x^n$ pour $x = z$, en supposant z et les a_i connus.

En utilisant la formule suivante :

$$p_n(x) = (\dots((a_n x + a_{n-1})x + a_{n-2})x + \dots)x + a_0 \quad (1.6)$$

on peut évaluer le polynôme à l'aide de la fonction suivante :

```
function p=evaluation(v)
    s=poly(0,"s");
    n=length(v);
    p=v(n);
    for i=n-1:-1:1
        p= p*s + v(i);
    end
endfunction;
```

Chapitre 2

Conclusion

Le code de notre projet a la forme suivante : les valeurs sont saisies par l'utilisateur, elles sont mises sous forme matrice de Vandermonde et si cette matrice est inversible, alors le système est résolu avec la méthode du pivot de Gauss.

Nous avons essayé de résoudre le problème en utilisant différentes stratégies : la stratégie du pivot maximal partiel, la stratégie du pivot avec seuil et la stratégie du pivot maximal total. Le plus difficile à coder est la stratégie du pivot maximal total car la permutation de colonnes pose problème.

Au cours de ce projet, un des obstacles a été l'apprentissage de la syntaxe de Scilab. Nous avons découvert les mots clé de base, puis quelques fonctions plus évoluées permettant d'implémenter facilement les algorithmes numériques comme celui du pivot de Gauss.

Chapitre 3

Annexes

3.1 Code source