M2 EID–Apprentissage de Données Visuelles

B.Matei

LIPN - CNRS UMR 7030

www.lipn.univ-paris13.fr/matei

Course

Organisation

- 15 CM ($10 \times 1,5$)
- 15 TP ($10 \times 1,5$): Python

Contrôle de connaissance

- Examen écrit : 50%
- TP : 50%

Plan du cours

- Introduction.
  1. Motivation : compression des images numériques.
  2. Cadre mathématique.
- Représentations multiéchelles non-linéaire.
  1. Cas 1D (Harten).
  2. Extension au cas 2D.
- Résultats.
  1. Amélioration de la concentration.
  2. Caractérisation de la régularité.
  3. Stabilité des représentations.
- Applications.

Le monde des images numériques

Photographie et vidéo numérique, TV, astronomie, médecine, etc.

Représentation mathématique d'une image :

$I(x, y)$ = Intensité lumineuse au point (x, y)

Noir $= 0 \leq I(x, y) \leq 1 =$ Blanc

Traitement numérique de l'image : ensemble des opérations mathématiques appliquées à $I(x, y)$ par des algorithmes, visant à résoudre des tâches spécifiques concernant l'image :
- Restauration, débruitage : une version dégradée de l'image que l'on cherche à améliorer,
- Compression, codage : réduire la place mémoire occupée par l'image numérique tout en conservant sa qualité.

Choix d'une représentation

Le choix d'une représentation appropriée de l'image est souvent fondamental pour résoudre un problème spécifique. Le point de vue de l'Analyse Harmonique : décomposer une fonction arbitraire en une combinaison de fonctions simples.

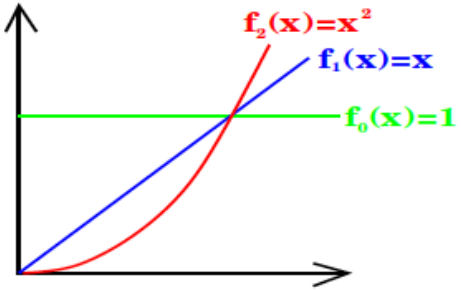
$$f(x) = a_0 f_0(x) + a_1 f_1(x) + a_2 f_2(x) + \cdots + a_n f_n(x) + \cdots$$

$\Rightarrow$ Représentation de f par les coefficients $a_0, a_1, a_2, \cdots$

Exemple 1 : Séries polynomiales

$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + \cdots$

i.e. $f_n(x) = x^n$

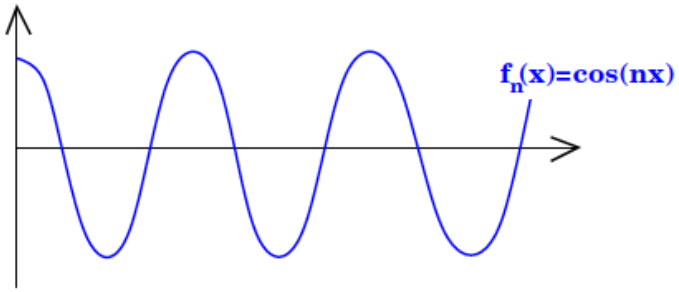


Exemple 2 : Séries de Fourier

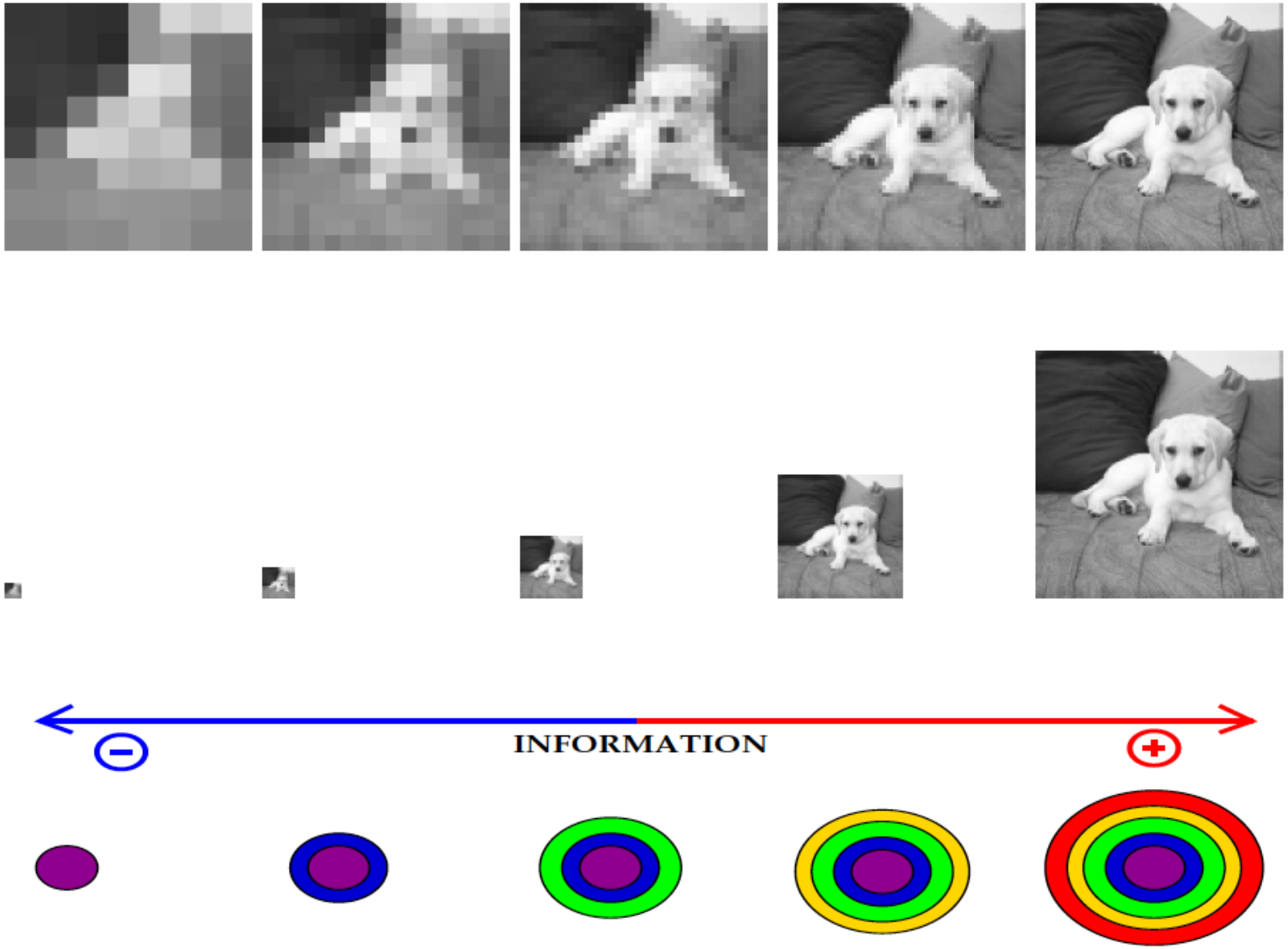
$f(x) = a_0 + a_1 \cos(x) + b_1 \sin(x)$

$+a_2 \cos(2x) + b_2 \sin(2x) + \cdots$

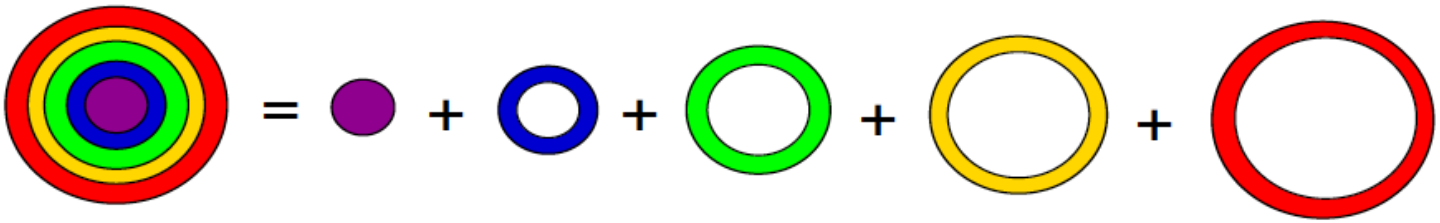
$+a_n \cos(nx) + b_n \sin(nx) + \cdots$



Structure hiérarchique de l'information visuelle



Représentation multiéchelle dans une base d'ondelettes



$$I = I_0 + D_1 + \ldots\ldots + D_{j-1} + D_j$$

où $D_j = I_j - \mathcal{A} I_{j-1}$

- concentration des coefficients numériquement significatifs.
- compression $\sim$ diminuer la taille de D_j.

  $\sim$ améliorer $\mathcal{A} I_{j-1}$ l'approximation de I_j.

Application à la compression des images

Idée de base : on ne code avec précision que les coefficients numériquement significatifs

⇒La résolution s'adapte localement

Exemple : codage de 1 % des coefficients

Standard de compression JPEG 2000 :

- Mêmes principes de base
- Utilise des approximations plus “lisses"
- Bonne qualité avec compression de 1/40

Application au débruitage des images

Représentation multiéchelle d'une image bruitée : les coefficients numériquement significatifs "émergent" au dessus du niveau d'intensité du bruit.

Stratégie naturelle : seuillage

des coefficients d'ondelettes inférieurs

au niveau d'intensité du bruit.

Mesure mathématique de la rareté

Vitesse de décroissance de l'erreur de l'approximation à N termes.

Représentation de l'image $I = \sum a_\lambda \psi_\lambda$.
Approximation à N termes $I_N := \sum_{N \text{ plus grands } |a_\lambda|} a_\lambda \psi_\lambda$, on s'intéresse au plus grand $s > 0$ t.q. : $\|I - I_N\|_{L^2} \leq CN^{-s}$.

Formulation équivalente : $(a_\lambda) \in w\ell^p$ avec 1/p=s+1/2.

Espaces faibles :. $(a_\lambda) \in w\ell^p$ si et seulement si
$\mathrm{Card}\{\lambda \text{ t.q. } |a_\lambda| > \eta\} \leq C\eta^{-p}$.

Intuition

le nombre de coefficients au-dessus du seuil η ne doit pas croître trop vite lorsque $\eta \to 0$

Rappels :

Norme L^2 d'une fonction $I : D \to R$: $\|I\|_{L^2}^2 = \int_D |I|^2$
En discret $\|I\|_{L^2}^2 = \sum_n |I_n|^2$

Modélisation des images réelles

Images : objets complexes = zones homogènes + contours.

Modèle possible

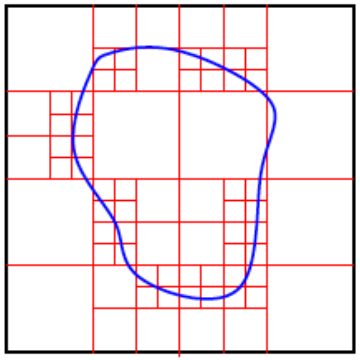
$I \in BV$ ssi $I \in L^1$ et ∇I une mesure finie.

Théorème. (Cohen, DeVore, Petrushev, Dahmen, Daubechies, Xu)

$$I \in BV([0,1]^2) \Rightarrow \|I - I_N\|_{L^2} \leq CN^{-\frac{1}{2}}$$

et pas mieux en général.

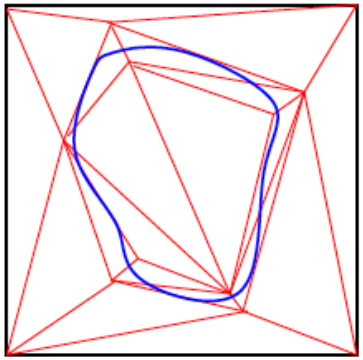
Image : $I = \chi_\Omega$, avec $\partial\Omega$ régulier.



I_N = approximation à N termes

$$\Rightarrow \|I - I_N\|_{L^2} \sim N^{-1/2}$$

Problème : on impose un raffinement isotrope



I_N = fonction constante par morceaux.

$$\Rightarrow \|I - I_N\|_{L^2} \sim N^{-1}$$

Problème : algorithme rapide et hiérarchique ?

Approches récentes pour une représentation des contours

- Adaptation de la base d'ondelettes.
  - Ridgelets/Curvelets. [Donoho et Candes] :

$$\|I - I_N\|_{L^2} \sim \frac{\log N}{N}$$

  - Contourlets. [Do et Vetterli].
  - Bandelettes. [Le Pennec et Mallat] :
    sous hypothèses de régularité et détection

$$\|I - I_N\|_{L^2} \sim N^{-s}, s > 1$$

- Décompositions multiéchelles non-linéaires. [Cohen et BM] :

$$\|I - I_N\|_{L^2} \sim N^{-1}$$

Représentations Multiéchelles Non-linéaires.

But : traitement adapté pour les singularités isolées, par l'utilisation des techniques introduites dans la simulation numérique des ondes de chocs par Harten et Osher.

Basées sur des techniques de reconstruction adaptées à un modèle sous-jacent.

Développées en 1D

Représentations proches des représentations introduites par Baraniuk, Davis et Sweldens en utilisant un lifting scheme non-linéaire..

Analyse des méthodes : jusque-là presque inexistante.

Cadre multiéchelles de Harten .

- $v^j = (v^j(\gamma))_{\gamma \in \Gamma_j}$ données au niveau de résolution 2^{-j}.
  - opérateur de projection : $v^j = P_j^{j+1} v^{j+1}$,
  - opérateur de prédiction : $\hat{v}^{j+1} = P_{j+1}^j v^j$ approximation de v^{j+1}.

Consistance :. $P_j^{j+1} P_{j+1}^j = I_{\Gamma_j}$.

Erreur de prédiction $d^j = v^{j+1} - \hat{v}^{j+1} \in Ker(P_j^{j+1})$

$$v^j \Leftrightarrow (v^{j-1}, d^{j-1}) \Leftrightarrow (v^{j-2}, d^{j-2}, d^{j-1}) \cdots \Leftrightarrow (v^0, d^0, \ldots, d^{j-1})$$

Cadre multiéchelles de Harten : exemple 1 (1D)

Représentation par les valeurs moyennes.

$$u_j = (u_k^j)_{k\in\mathbb{Z}}, \text{ avec } I_{j,k} := 2^{-j}[k, k+1[\text{ et}$$

$$u_k^j = \frac{1}{|I_{j,k}|}\int_{I_{j,k}} u.$$

$\Rightarrow$ fixe l'opérateur de projection

$$u_k^j = \frac{1}{2}\left(u_{2k}^{j+1} + u_{2k+1}^{j+1}\right).$$

Problème :. choix approprié de l'opérateur de prédiction.

Exemple 1 :

On prend

$$\hat{u}_{2k}^{j+1} = \hat{u}_{2k+1}^{j+1} := u_k^j.$$

Equivalente à la décomposition dans la base de Haar.

Problème : ordre peu élevé (prédiction peu précise même si u est régulière)

Exemple 1 Calcul de détails :

On a :

$$e_{2k}^{j+1} = u_{2k}^{j+1} - \hat{u}_{2k}^{j+1} = u_{2k}^{j+1} - u_k^j = u_{2k}^{j+1} - \frac{1}{2}(u_{2k}^{j+1} + u_{2k+1}^{j+1}) = \frac{1}{2}(u_{2k}^{j+1} - u_{2k+1}^{j+1})$$

On a aussi :

$$e_{2k+1}^{j+1} = u_{2k+1}^{j+1} - \hat{u}_{2k+1}^{j+1} = u_{2k+1}^{j+1} - u_k^j = u_{2k+1}^{j+1} - \frac{1}{2}(u_{2k}^{j+1} + u_{2k+1}^{j+1}) = -\frac{1}{2}(u_{2k}^{j+1} - u_{2k+1}^{j+1})$$

Conclusion : $e_{2k}^{j+1} = -e_{2k+1}^{j+1}$, on appelle cette valeur commune détail d'approximation et on la note d_{2k}^{j+1}

On remarque que $u_{2k}^{j+1} = \hat{u}_{2k}^{j+1} + d_{2k}^{j+1}$et que $u_{2k+1}^{j+1} = \hat{u}_{2k+1}^{j+1} - d_{2k}^{j+1}$

Algorithme–Transformée directe

```
v=Haar1d-directe(n,u)
v=copy(u)
w=copy(u)
m=n;
while 1 < m do
    forall k ∈ [0..m/2 − 1] do
        w[k] = (v[2k] + v[2k + 1])/2
        w[k+m/2] = (v[2k] - v[2k + 1])/2
    end
    forall k ∈ [0..m − 1] do
        v[k] =w[k];
    end
    m=m/2;
end
return v
```

Algorithme –Transformée inverse

```
v=Haar1d-inverse(n,u)
v=copy(u)
w=copy(u)
m=1;
while m < n do
    forall k ∈ [0..m/2 − 1] do
        w[2k] = v[k] + v[k + m]
        w[2k + 1] = v[k]- v[k + m]
    end
    forall k ∈ [0..2m − 1] do
        v[k] =w[k];
    end
    m=2m;
end
return v
```

Exemple numérique Haar :

On considère le vecteur $u = [1, 2, 3, 4, 5, 6, 7, 8]$ de dimension $n = 8 = 2^3$. On calcule "à la main" le résultat de la transformée de Haar 1d. directe

- Première étape
  - Calcul de moyennes $u = [1, 2, 3, 4, 5, 6, 7, 8] -> w = \left[\frac{1+2}{2}, \frac{3+4}{2}, \frac{5+6}{2}, \frac{7+8}{2}, 0, 0, 0, 0\right]$
  - Calcul de détails $w -> w = \left[\frac{1+2}{2}, \frac{3+4}{2}, \frac{5+6}{2}, \frac{7+8}{2}, \frac{1-2}{2}, \frac{3-4}{2}, \frac{5-6}{2}, \frac{7-8}{2}\right]$

    $w = \left[\frac{3}{2}, \frac{7}{2}, \frac{11}{2}, \frac{15}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$
- Deuxième étape, en entrée w, on travaille sur la première moitié du vecteur
  - Calcul de moyennes $w = \left[\frac{10}{4}, \frac{26}{4}, 0, 0, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$
  - Calcul de détails $w = \left[\frac{10}{4}, \frac{26}{4}, \frac{3-7}{4}, \frac{11-15}{4}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$

    $w = \left[\frac{10}{4}, \frac{26}{4}, -1, -1, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$
- Troisième étape, en entrée w, on travaille sur la première moitié du vecteur
  - Calcul de moyennes $w = \left[\frac{36}{8}, 0, -1, -1, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$
  - Calcul de détails $w = \left[\frac{36}{8}, \frac{10-26}{8}, -1, -1, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$

    $w = \left[\frac{36}{8}, -2, -1, -1, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$

Cadre multiéchelles de Harten : exemple 2 (1D)

Représentation par les valeurs ponctuelles.

$$u_j = (u_k^j)_{k \in \mathbb{Z}}, \text{ avec } I_{j,k} := 2^{-j}[k, k+1[\text{ et}$$

$$u_k^j = u(2^{-j}k)$$

$\Rightarrow$ fixe l'opérateur de projection

$$u_k^j = u_{2k}^{j+1}$$

Problème : choix approprié de l'opérateur de prédiction.

Exemple 2 :

On prend

$$\hat{u}_{2k+1}^{j+1} = (u_k^j + u_{k+1}^j)/2$$

Equivalente à la décomposition dans une base interpolante.

Problème : ordre peu élevé (prédiction peu précise même si u est régulière), car on approche par une interpolation linéaire la fonction sous-jacente u

Cadre multiéchelles : amélioration de l'ordre d'approximation

Exemple 2 : Reconstruction d'ordre 3.

On définit $p_k(x)$ comme le polynôme cubique t.q. :

$$u_l^j = p_k(2^{-j}l), l = k-1, k, k+1, k+2.$$

On définit

$$\hat{u}_{2k+1}^{j+1} = p_k(k+1/2).$$

On observe que par un changement de variables ce problème devient : on cherche $p_0(x)$ comme le polynôme cubique t.q.

$$u_l^0 = p_k(l), \qquad l = -1, 0, 1, 2.$$

Cela se réduit à la résolution d'un système linéaire d'orde 4

Problème : une singularité isolée affecte l'erreur de prédiction.

Cadre multiéchelles : amélioration de l'ordre d'approximation

Exemple 2 : Reconstruction d'ordre 3.

Idée : remplacer p_k dans la construction précédente par $\tilde{p}_k$ construit comme le moins oscillant des polynômes

$$\{p_{k-1}, p_k, p_{k+1}\}.$$

On définit p_{k-1} en utilisant $k-2, k-1, k, k+1$. On définit p_k en utilisant $k-1, k, k+1, k+2$. On définit p_{k+1} en utilisant $k, k+1, k+2, k+3$.

On améliore au voisinage de l'intervalle $[k, k+1)$, mais pas à l'intérieur de celui-ci.

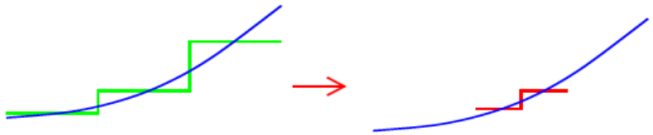
Valeurs moyennes : amélioration de l'ordre d'approximation

On définit $p_k(x)$ comme le polynôme quadratique qui interpole les moyennes t.q.

$$u_l^j = |I_{j,l}|^{-1} \int_{I_{j,l}} p_k, \qquad l = k-1, k, k+1.$$

On définit

$$\hat{u}_{2k}^{j+1} = |I_{j+1,2k}|^{-1} \int_{I_{j+1,2k}} p_k \quad \text{et} \quad \hat{u}_{2k+1}^{j+1} = |I_{j+1,2k+1}|^{-1} \int_{I_{j+1,2k+1}} p_k$$

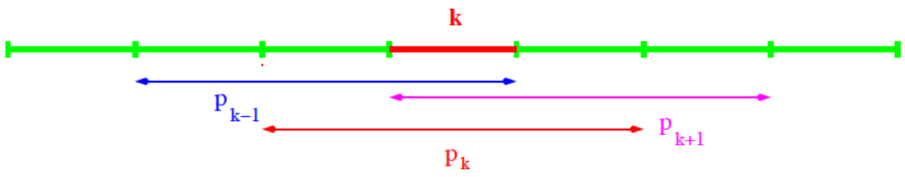


Décomposition multiéchelles obtenue équivalente à la décomposition dans une base d'ondelettes biorthogonales

Problème : une singularité isolée affecte l'erreur de prédiction

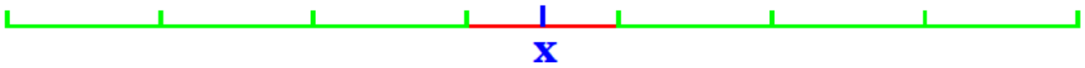
Exemple 3 : Prédiction non-linéaire essentiellement non-oscillante (ENO).

Idée : remplacer p_k dans la construction précédente par $\tilde{p}_k$, construit comme le moins oscillant des polynômes $\{p_{k-1}, p_k, p_{k+1}\}$.



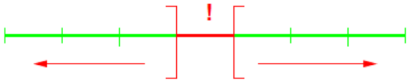
Mesures des oscillations : $|p''|$, $|u^j_{k-1} - 2u^j_k - u^j_{k+1}|$, etc.
Processus **non-linéaire** dépendant des données.

Avantage : on réduit le nombre d'intervalles ayant une mauvaise prédiction.



Exemple 4 : ENO avec la reconstruction intra maille .

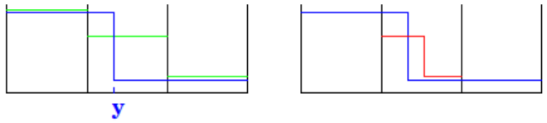
- étape 1 : détection des intervalles contenant une discontinuité.



si pour l'intervalle $k-1$ ENO choisit le polynôme décentré à gauche et pour l'intervalle $k+1$ ENO choisit le polynôme décentré à droite alors l'intervalle k contient une discontinuité

- étape 2 : si $I_{j,k}$ contient une discontinuité, on définit :

$$\tilde{p}_k = a\chi_{]-\infty,y]} + b\chi_{[y,+\infty[}, \text{ où } a = u^j_{k-1}, b = u^j_{k+1} \text{ t.q. } u^j_k = \int_{I_{j,k}} \tilde{p}_k.$$

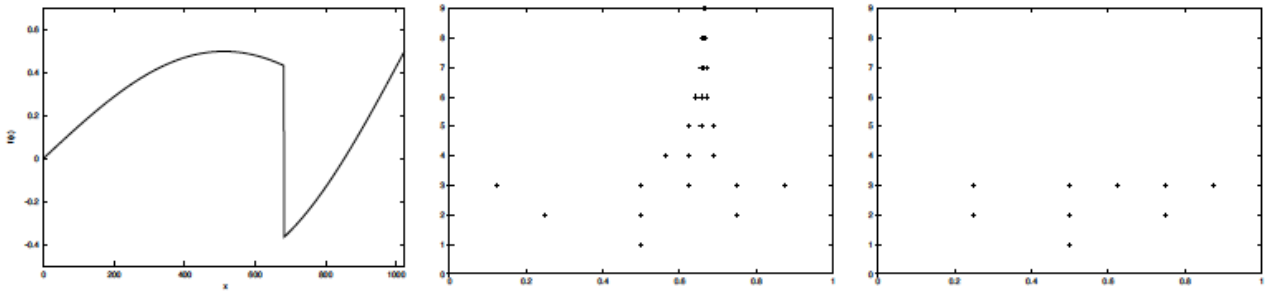


- Pour les autres intervalles (ou lorsque $y \notin I_{j,k}$), on utilise ENO.

Avantage : précision ameliorée au voisinage de la singularité.

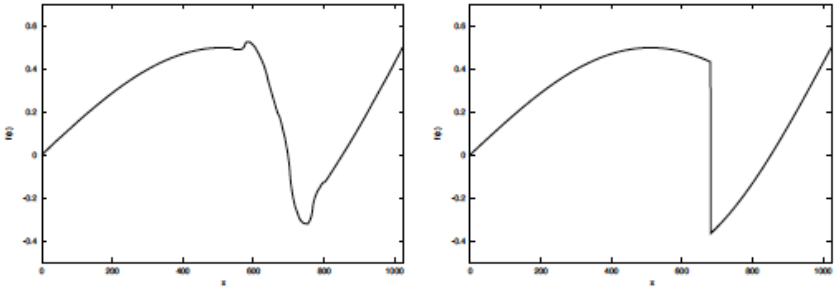
Valeurs moyennes : Exemple numérique

Détails $|d_k^j| > \varepsilon = 10^{-2}$



Prédiction linéaire et ENO-SR

Reconstructions uniquement à partir des moyennes v^3



Prédiction linéaire et ENO-SR

Détails : Reconstruction d'ordre 2 (I) :

On définit $p_k(x)$ comme le polynôme quadratique t.q. :

$$u_l^j = |I_{j,l}|^{-1} \int_{I_{j,l}} p_k, \qquad l = k-1, k, k+1.$$

On définit :

$$\hat{u}_{2k}^{j+1} = |I_{j+1,2k}|^{-1} \int_{I_{j+1,2k}} p_k. \quad \text{et} \quad \hat{u}_{2k+1}^{j+1} = |I_{j+1,2k+1}|^{-1} \int_{I_{j+1,2k+1}} p_k.$$

On observe que par un changement de variables ce problème devient : on cherche $p_0(x)$. comme le polynôme quadratique t.q.

$$u_l^0 = |I_{0,l}|^{-1} \int_{I_{0,l}} p_0 = u_l^0 = \int_{I_{0,l}} p_0, \qquad l = -1, 0, +1.$$

Détails : Reconstruction d'ordre 2 (II) :

Soit

$$p_0(x) = a \cdot x^2 + b \cdot x + c.$$

Les trois conditions $l = -1, 0, +1$ nous donne le système suivant avec les inconnues a, b, c :

$$\begin{cases} \int_{-1,0} x^2 dxa + \int_{-1,0} xdxb + \int_{-1,0} dxc & = u^0_{-1} \\ \int_{0,1} x^2 dxa + \int_{0,1} xdxb + \int_{0,1} dxc & = u^0_0 \\ \int_{1,2} x^2 dxa + \int_{1,2} xdxb + \int_{1,2} dxc & = u^0_{+1} \end{cases}$$

qui devient

$$\begin{cases} \frac{1}{3} a - \frac{1}{2} b + c & = u^0_{-1} \\ \frac{1}{3} a + \frac{1}{2} b + c & = u^0_0 \\ \frac{7}{3} a + \frac{3}{2} b + c & = u^0_{+1} \end{cases}$$

Les solutions du système sont :

$$\begin{cases} a & = \frac{1}{2} u^0_{-1} + u^0_0 + \frac{1}{2} u^0_{+1} \\ b & = -u^0_{-1} + u^0_0 \\ c & = \frac{1}{2} u^0_{-1} + u^0_0 + \frac{1}{2} u^0_{+1} \end{cases}$$

Détails : Reconstruction d'ordre 2 (III) :

On définit :

$$\hat{u}_0^1 = 2\int_{I_{1,0}} p_0 \quad \text{et} \quad \hat{u}_1^1 = 2\int_{I_{1,1}} p_0.$$

avec les a, b, c trouvés précédemment. On obtient ainsi

$$\begin{cases} u_0^1 & = u_0^0 + \frac{1}{8}(u_{-1}^0 - u_{+1}^0) \\ u_1^1 & = u_0^0 - \frac{1}{8}(u_{-1}^0 - u_{+1}^0) \end{cases}$$

Détails : Reconstruction non-linéaire d'ordre 2 (I) :

Idée : remplacer p_k dans la construction précédente par $\tilde{p}_k$ construit comme le moins oscillant des polynômes

$$\{p_{k-1}, p_k, p_{k+1}\}.$$

On définit :

$$\hat{u}_{2k}^{j+1} = |I_{j+1,2k}|^{-1} \int_{I_{j+1,2k}} \tilde{p}_k. \quad \text{et} \quad \hat{u}_{2k+1}^{j+1} = |I_{j+1,2k+1}|^{-1} \int_{I_{j+1,2k+1}} \tilde{p}_k.$$

On observe que par un changement de variables ce problème devient : on cherche $p_{-1}(x), p_0(x), p_1(x)$. comme les polynôme quadratique t.q.

$$\begin{cases} u_l^0 &= \int_{I_{0,l}} p_{-1}, & l = -2, -1, 0 \\ u_l^0 &= \int_{I_{0,l}} p_0, & l = -1, 0, +1 \\ u_l^0 &= \int_{I_{0,l}} p_1, & l = 0, +1, +2. \end{cases}$$

Détails : Reconstruction non-linéaire d'ordre 3 (II) :

Le calcul de p_0 ne change pas. Le calcul de p_{-1} est similaire à celui de p_1. On se concentre sur p_1. Soit

$$p_1(x) = a_1 \cdot x^2 + b_1 \cdot x + c_1.$$

Les trois conditions $l = 0, +1, +2$ nous donne le système suivant avec les inconnues a_1, b_1, c_1 :

$$\begin{cases} \frac{1}{3}a_1 + \frac{1}{2}b_1 + c_1 & = u_0^0 \\ \frac{7}{3}a_1 + \frac{3}{2}b_1 + c_1 & = u_{+1}^0 \\ \frac{19}{3}a_1 + \frac{5}{2}b_1 + c_1 & = u_{+2}^0 \end{cases}$$

Les solutions du système sont :

$$\begin{cases} a_1 & = \frac{1}{2}u_0^0 - u_1^0 + \frac{1}{2}u_{+2}^0 \\ b_1 & = -2u_0^0 + 3u_1^0 - u_2^0 \\ c_1 & = \frac{11}{6}u_0^0 + -76u_1^0 + \frac{1}{3}u_{+2}^0 \end{cases}$$

Détails : Reconstruction non-linéaire d'ordre 3 (III) :

On définit

$$\hat{u}_0^1 = 2\int_{I_{1,0}} p_1 \quad \text{et} \quad \hat{u}_1^1 = 2\int_{I_{1,1}} p_1.$$

avec les a_1, b_1, c_1 trouvés précédemment. On obtient ainsi

$$\begin{cases} u_0^1 & = \frac{11}{8}u_0^0 - \frac{1}{2}u_{+1}^0 + \frac{1}{8}u_{+2}^0 \\ u_1^1 & = \frac{5}{8}u_0^0 + \frac{1}{2}u_{+1}^0 - \frac{1}{8}u_{+2}^0 \end{cases}$$

On obtient de la même manière pour le cas décentré à gauche :

$$\begin{cases} u_0^1 & = \frac{5}{8}u_0^0 + \frac{1}{2}u_{-1}^0 - \frac{1}{8}u_{-2}^0 \\ u_1^1 & = \frac{11}{8}u_0^0 - \frac{1}{2}u_{-1}^0 + \frac{1}{8}u_{-2}^0 \end{cases}$$

Algorithme Quadratique –Transformée directe

```
v=Quad1d-directe(n,u)
v=copy(u)
w=copy(u)
m=n;
while 1 < m do
    forall k = 0, k = m/2 − 1 do
        on calcule les valeurs aux bords
        w[k] = (v[2k] + v[2k + 1])/2
        w[k+m/2] = (v[2k] - v[2k + 1])/2
    end
    forall k ∈ [1..m/2 − 2] do
        on calcule les valeurs moyennes
        w[k] = (v[2k] + v[2k + 1])/2
        on calcule les valeurs prédites v̂[2k]
        v̂[2k] =w[k] -1/8(w[k-1] -w[k+1] )
        on calcule l'erreur
        w[k+m/2] = v[2k] -v̂[2k]
    end
    forall k ∈ [0..m − 1] do
        v[k] =w[k];
    end
    m=m/2;
end
return v
```

Algorithme Quadratique –Transformée inverse

```
v=Quad1d-inverse(n,u)
v=copy(u)
w=copy(u)
m=1;
while m < n do
    forall k = 0, k = m/2 − 1 do
        on calcule les valeurs aux bords
        w[2k] = v[k] + v[k + m]
        w[2k + 1] = v[k]- v[k + m]
    end
    forall k ∈ [1..m/2 − 2] do
        on calcule les valeurs prédites v̂[2k]
        v̂[2k] =v[k] -1/8(v[k-1] -v[k+1] )
        on "ajoute" l'erreur aka le détail pour retrouver les valeurs initiales
        w[2k] = v̂[2k] + v[k + m]
        w[2k + 1] = v̂[2k]- v[k + m]
    end
    forall k ∈ [0..2m − 1] do
        v[k] =w[k];
    end
    m=2m;
end
return v
```