

IMN430

Chapitre 5

Visualisation de champs vectoriels et autres sujets palpitants

Olivier Godin

Université de Sherbrooke

20 mars 2017

Plan de la présentation

- 1 Visualisation de vecteurs
- 2 Visualisation de tenseurs
- 3 Références

Visualisation de vecteurs

- 1 Visualisation de vecteurs
 - Déformation et déplacement
 - Représentation de trajectoires
 - Vecteurs et textures

2 Visualisation de tenseurs

3 Références

Déformation et déplacement

- 1 Visualisation de vecteurs
 - Déformation et déplacement
 - Représentation de trajectoires
 - Vecteurs et textures

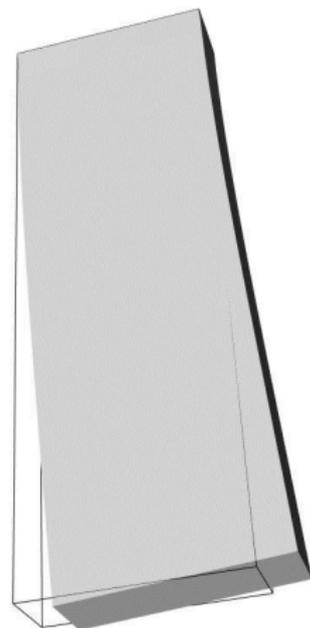
2 Visualisation de tenseurs

3 Références

Vecteurs et mouvement

Les notions de vecteur et de champ vectoriel sont souvent associées à celles de **mouvement** et de **déplacement**.

Ainsi, une technique efficace pour représenter des données vectorielles est d'**illustrer la déformation d'une forme géométrique** selon les forces présentes dans le champ vectoriel.

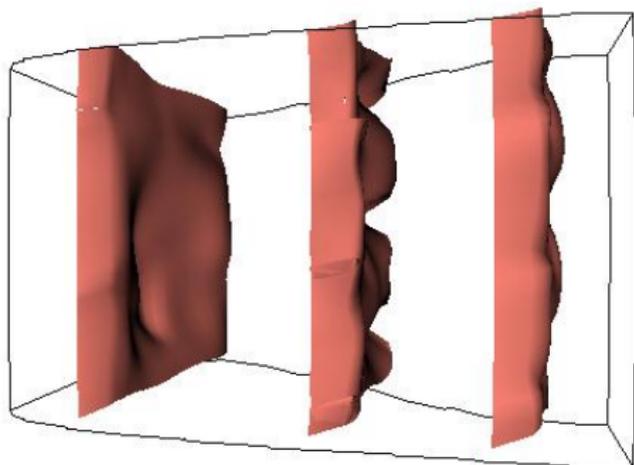


[3]

Vecteurs et mouvement

Dans l'exemple précédent, on illustre **la déformation d'une poutre** selon un certain champ vectoriel. On estime l'amplitude de celui-ci en **comparant avec la forme originale**.

Dans la figure suivante, **on déforme des plans** selon la direction dans un champ vectoriel 3D.



Vecteurs et mouvement

Typiquement, il sera utile de faire une mise à l'échelle du champ vectoriel afin de **contrôler la distorsion géométrique** :

- une distorsion trop petite sera difficile à percevoir ;
- une distorsion trop grande pourrait amener l'objet à avoir des intersections avec lui-même.

Dans un tel cas, il sera plus difficile de bien **saisir le comportement du champ vectoriel**.

Graphiques de déplacement

Cette idée de déformation est à la base de la technique des **graphiques de déplacement** qui propose de représenter un champ vectoriel autrement que par un ensemble discret de trajectoires, comme le suggère la technique des glyphes vue au chapitre 3.

En effet, un glyphe ayant son origine à un certain point p peut être interprétée comme **la trajectoire de p dans un court intervalle de temps Δt** .

Graphiques de déplacement

Tandis que les glyphes illustrent à la fois **le point de départ et le point d'arrivée** d'une trajectoire (c'est-à-dire p et $p + \mathbf{v}(p)\Delta t$, respectivement), les graphiques de déplacement proposent de **n'afficher que les points d'arrivée** de ces trajectoires.

Soit une surface $S \in D$, où D est le domaine d'un champ vectoriel 3D. S est alors discrétisé en un ensemble de points p_i .

Graphiques de déplacement

On définit alors le **graphique de déplacement** de S comme étant la nouvelle surface S' donnée par l'ensemble de points

$$p'_i = p_i + k\mathbf{v}'(p_i).$$

Ici, \mathbf{v}' est le champ vectoriel qui **contrôle le déplacement de S** et k est un **facteur d'échelle**.

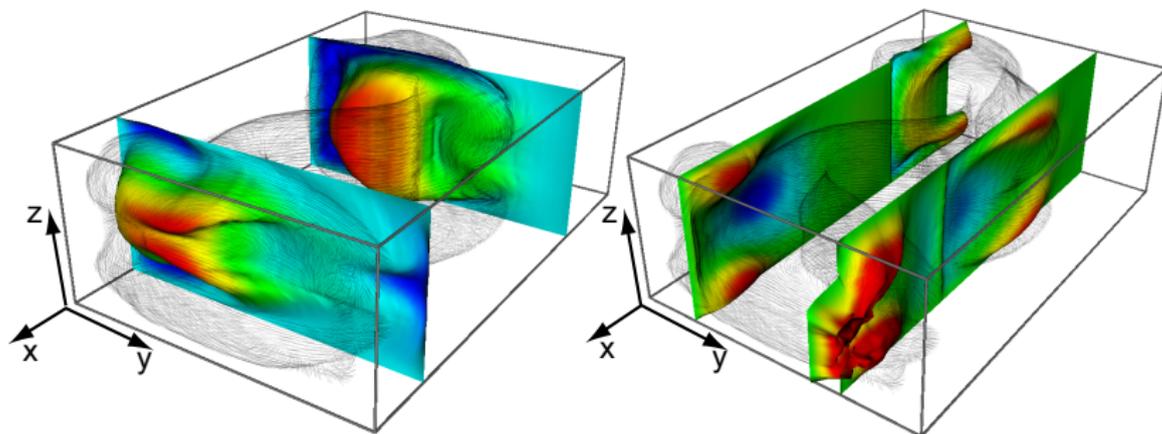
Dans le cas le plus simple, on prendra $\mathbf{v}' = \mathbf{v}$ et la surface S sera déplacée dans la direction exacte du champ vectoriel.

Graphiques de déplacement

Dans l'image suivante, on applique la technique des graphiques de déplacement à un champ vectoriel 3D. Dans l'image de gauche, on déforme **deux plans perpendiculaires à l'axe des x** , tandis que dans l'image de droite, on applique le même traitement à **deux plans perpendiculaires à l'axe des y** .

Les deux résultats utilisent aussi le facteur d'échelle $k = 20$.

Graphiques de déplacement



[6]

Graphiques de déplacement

Dans cet exemple, on a combiné les techniques des graphiques de déplacement et du placage de couleurs : en plus d'être déformées, les surfaces sont **colorées selon l'amplitude** de la composante par rapport à laquelle la surface est perpendiculaire (v_x et v_y , respectivement).

En bleu, on a les points présentant un déplacement minimal et en rouge ceux présentant un déplacement maximal.

Graphiques de déplacement

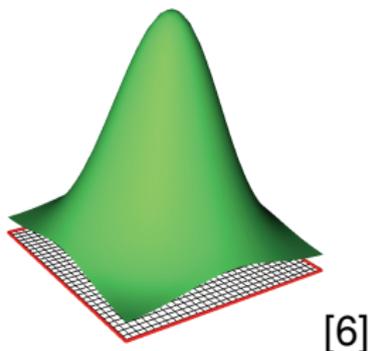
Les graphiques de déplacement ont l'avantage, par rapport aux techniques basées sur les glyphes, d'**illustrer de manière continue un champ vectoriel 3D**.

Cependant, il peut être moins intuitif et évident d'interpréter la signification d'un graphique de déplacement pour en **déduire les directions** présentes dans un champ vectoriel.

Graphiques de déplacement

Il est aussi tentant d'associer cette technique avec celle des **cartes de hauteurs** présentée au chapitre 3. Leur fonctionnement est toutefois bien différent.

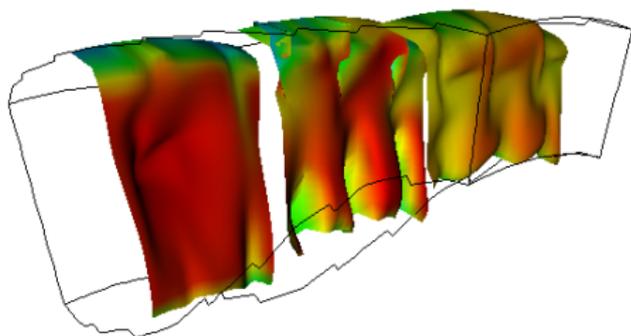
Les cartes de hauteurs permettent de représenter visuellement des données scalaires 2D en déformant une surface selon les valeurs scalaires associées à des positions sur une grille 2D.



Graphiques de déplacement

La surface d'une carte de hauteurs est donc **perpendiculaire à la normale** définie par l'ensemble de données scalaires.

À l'opposé, les graphiques de déplacement permettent d'obtenir une représentation graphique pour des données vectorielles 3D, et la surface sera **perpendiculaire à l'ensemble de données**.



Paramètres de contrôle

Plusieurs facteurs ont un impact sur la **qualité de la représentation** offerte par les graphiques de déplacement. À titre d'exemple, nous avons déjà mentionné l'impact que peut avoir le **facteur d'échelle**.

On s'attarde maintenant à la **forme** et la **position** de la surface à déformer. En variant ces deux paramètres, on modifie **l'ensemble de points d'intérêt** (ceux dont le mouvement sera visualisé) dans le champ vectoriel.

Paramètres de contrôle

Les surfaces planaires sont des choix très populaires pour les graphiques de déplacement. Comme elles sont plates, il est plus facile de **distinguer les régions perturbées** de même que la direction du mouvement.

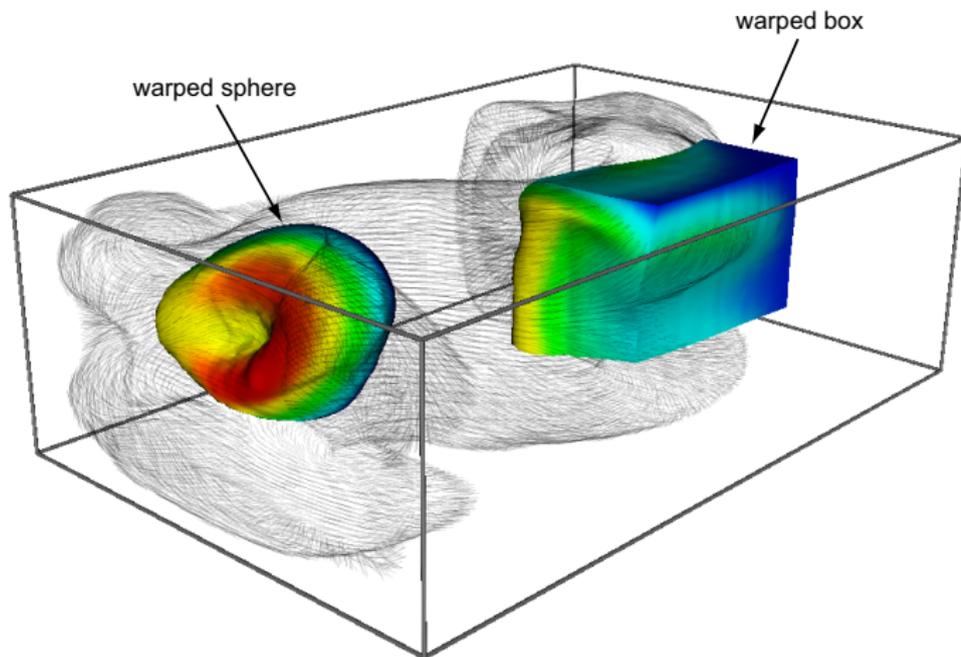
Si on les utilise, il faut porter une attention particulière au positionnement de ces plans. En effet, si une surface se retrouve à être **tangente au champ vectoriel**, il n'y aura aucun déplacement et la visualisation n'apportera aucune information.

Paramètres de contrôle

Cette approche n'est toutefois pas limitée aux surfaces planaires. L'exemple suivant présente des graphiques de déplacement créés **à partir d'une sphère et d'une boîte**.

La différence visuelle entre la **forme originale anticipée** et la **forme réelle perçue** donne alors l'information sur l'amplitude du champ vectoriel.

Paramètres de contrôle



[6]

Paramètres de contrôle

Dans un monde idéal, le choix de l'objet à déformer devrait être relié à la **signification réelle du champ vectoriel** et à son comportement anticipé.

À titre d'exemple, si une force est appliquée sur une certaine pièce mécanique, il serait pertinent d'**utiliser un modèle 3D de cette pièce** comme surface de base pour la déformation. On aurait alors une signification encore plus naturelle et évidente pour la représentation visuelle.

Représentation de trajectoires

- 1 **Visualisation de vecteurs**
 - Déformation et déplacement
 - **Représentation de trajectoires**
 - Vecteurs et textures
- 2 Visualisation de tenseurs
- 3 Références

Position en fonction du temps

Certaines techniques présentées jusqu'à maintenant peuvent être interprétées comme une façon de **représenter le mouvement d'un point ou d'un objet** sur un bref intervalle de temps.

Le meilleur exemple est un des segments dans un diagramme en hérisson. On peut aisément associer celui-ci au **déplacement du point d'origine dans la direction donnée** durant une période de temps déterminée par le facteur d'échelle.

Position en fonction du temps

En d'autres mots, si on considère le vecteur \mathbf{v} comme étant une vitesse, c'est-à-dire que

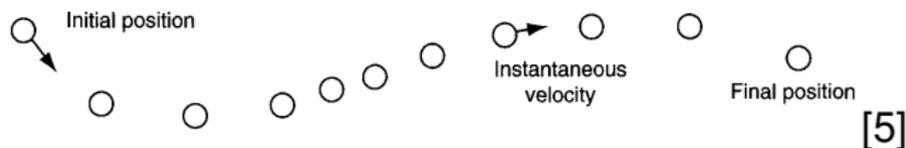
$$\mathbf{v} = \frac{dx}{dt},$$

alors **le déplacement d'un point** est donné par $dx = \mathbf{v} dt$.

Cette idée nous amène à étendre le concept du graphique en hérisson afin de permettre le suivi d'un point donné en fonction du temps.

Position en fonction du temps

Le fonctionnement est simple : on indique par un point la position initiale d'un objet, puis en suivant les directions données par le champ vectoriel, **on affiche les différentes positions prises par l'objet.**



Notre cerveau se charge du reste et permet de déduire **l'amplitude et l'orientation du champ vectoriel** à ces positions.

Position en fonction du temps

Il faut toutefois faire preuve de prudence avec cette technique. En effet, dans un champ vectoriel, **la vitesse en un point est une mesure instantanée**. Or, on considère ici qu'elle est constante pour toute la durée d'un intervalle de temps fixé par le facteur d'échelle.

En utilisant un intervalle trop long, on risque de rater des changements dans le champ vectoriel, et **les positions obtenues ne seront plus valides**.

Position en fonction du temps

Ce faisant, le choix du **pas d'incrément de temps** est une étape cruciale dans le calcul des positions à partir d'un champ vectoriel. Pour évaluer le déplacement d'un point, on fait appel au calcul intégral :

$$\mathbf{x}(t) = \int_t^{t+\Delta t} \mathbf{v} dt.$$

Or, comme il est **impossible de résoudre analytiquement** cette équation avec des données réelles, on utilise l'intégration numérique pour estimer le résultat.

Position en fonction du temps

La forme la plus simple pour l'intégration numérique est la **méthode d'Euler** qui donne l'équation

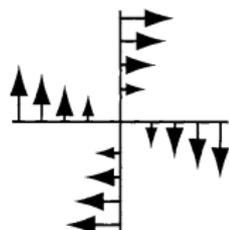
$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_i \Delta t,$$

où la position au temps $i + 1$ est la somme de la position précédente \mathbf{x}_i et de la vitesse instantanée \mathbf{v}_i affectée par le pas d'incrément Δt .

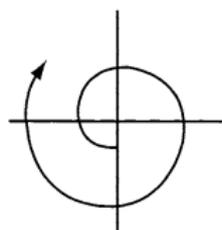
L'erreur attendue par cette technique est de l'ordre $O(\Delta t^2)$, ce qui peut ne pas être suffisamment précis selon le domaine d'application.

Position en fonction du temps

La figure suivante illustre une **conséquence de ce manque de précision**. Le champ vectoriel décrit une rotation parfaite autour d'un point central, mais en utilisant l'estimation proposée par la méthode d'Euler, on trouvera toujours une trajectoire divergente.



(a) Rotational vector field



(b) Euler's method

[5]

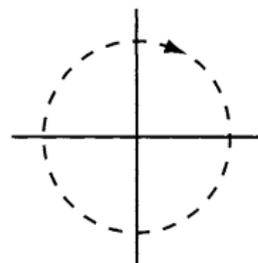
Position en fonction du temps

Une technique d'intégration numérique plus efficace (mais plus complexe) est celle de **Runge-Kutta** où la position au temps $i + 1$ est donnée par l'équation

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{\Delta t}{2} (\mathbf{v}_i + \mathbf{v}_{i+1}),$$

où la vélocité \mathbf{v}_{i+1} est quant à elle évaluée à l'aide de la méthode d'Euler.

Cette technique **gère bien la situation problématique précédente** avec le champ vectoriel circulaire.



(c) Runge-Kutta

[5]

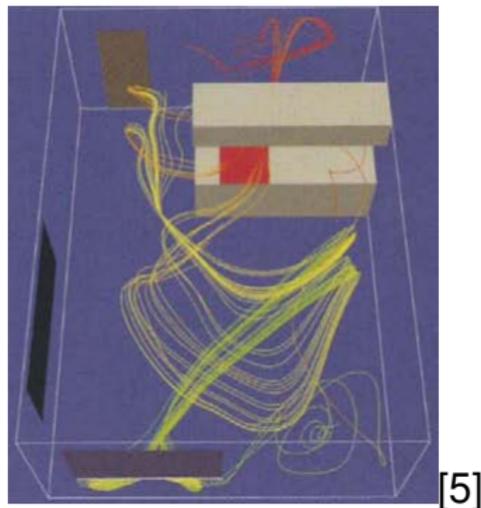
Position en fonction du temps

De manière générale, on trouve **les positions successives d'un point** dans un ensemble de données vectorielles en effectuant les étapes suivantes :

- 1 **Déterminer dans quelle cellule** (voxel ou pixel) se trouve le point
- 2 **Évaluer la vitesse** en ce point par interpolation des cellules voisines.
- 3 **Trouver les positions du point** de manière incrémentale avec la méthode de Runge-Kutta jusqu'à ce que le point sorte de l'ensemble de données.

Lignes de courant 2D

Une suite naturelle à la représentation précédente est de **relier les positions trouvées par un segment de droite**. Le résultat est une approximation numérique du parcours suivi par ce point.



Lignes de courant 2D

Tant les techniques basées sur les glyphes que les graphiques de déplacement cherchent à **illustrer la relation entre un point p et sa position déplacée $p + \mathbf{v}\Delta t$** selon un champ vectoriel.

Les glyphes illustrent une **trajectoire linéaire** sur un court intervalle de temps Δt , tandis que les graphiques de déplacement donnent les **positions finales** d'un ensemble dense de points sur une surface après un bref délai.

Lignes de courant 2D

Une question naturelle est donc de savoir si cet intervalle Δt peut être allongé afin de mieux suivre l'évolution des positions des particules qui seraient lâchées dans un champ vectoriel.

Les **objets de courant** (*stream objects*) proposent un ensemble de solutions à cette question en fournissant une gamme d'outils rendant possible cette forme de visualisation.

Lignes de courant 2D

On débute notre étude de cette famille de techniques de visualisation avec sa forme la plus simple et la plus répandue : la **ligne de courant**. Dans un champ vectoriel, une ligne de courant S correspond à la **trajectoire d'une particule** imaginaire lancée en un point p_0 appelé la source, sur une durée T .

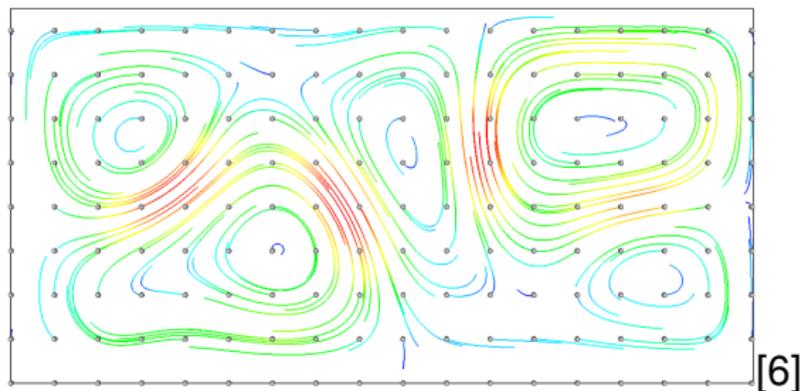
On l'obtient en prenant **l'intégrale du champ vectoriel** sur un intervalle de temps T à partir d'un point p_0 :

$$S = \{p(\tau) | \tau \in [0, T]\} \quad \text{avec} \quad p(\tau) = \int_{t=0}^{\tau} \mathbf{v}(p) dt,$$

et avec $p(0) = p_0$.

Lignes de courant 2D

L'intégrale précédente est alors **calculée avec une des techniques d'intégration numérique** vues précédemment. La figure suivante illustre les lignes de courant obtenues avec 18×9 sources réparties uniformément.



Lignes de courant 2D

Les lignes sont colorées **en fonction de l'amplitude du champ vectoriel** : bleu pour les basses vitesses et rouge pour les vitesses élevées.

Même si **les lignes sont de longueurs différentes**, l'intervalle de temps qu'elles représentent est le même. Les variations dans les longueurs sont le résultat des variations de la vitesse dans le champ vectoriel.

Lignes de courant 2D

Un des problèmes des lignes de courant est que la technique repose sur **une succession d'intégrations numériques**. Or, celles-ci ont la fâcheuse propriété d'accumuler les erreurs à mesure que le temps d'intégration τ augmente.

Ce faisant, plus on s'approche de la fin d'une ligne, **plus on s'éloigne de la trajectoire réelle de la particule**.

Lignes de courant 2D

Un autre paramètre crucial dans la visualisation par lignes de courant est **le choix du nombre de la position des sources**.

Une ligne de courant illustre le comportement d'un champ vectoriel pour les points situés à proximité de sa trajectoire. Toutefois, en choisissant une source, **on ne sait pas d'avance par quelles cellules du champ vectoriel passera la ligne de courant**.

Choisir la position d'une source permet uniquement de connaître l'ensemble des points sur **une trajectoire ayant la source comme point de départ**.

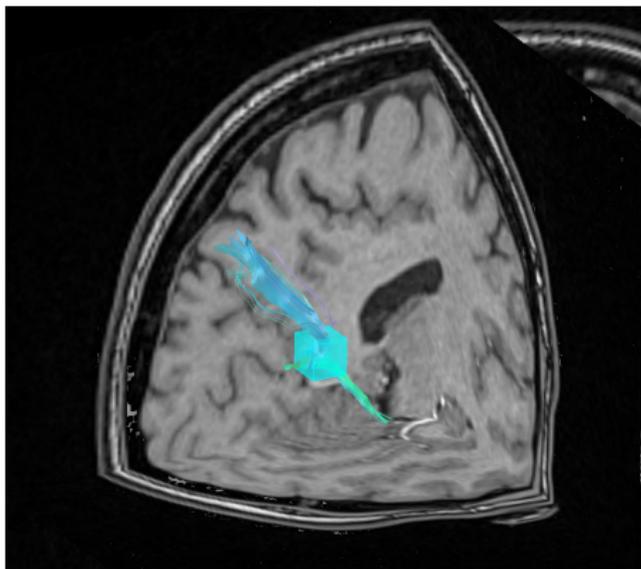
Lignes de courant 2D

On peut aussi facilement obtenir l'ensemble des points sur une trajectoire se terminant en un point donné. Il suffit pour cela de **prendre des valeurs négatives pour Δt** .

En traçant une ligne de courant à partir d'une source pour des valeurs positives et négatives de Δt , on se retrouve avec **l'ensemble des points sur la trajectoire passant par le point source**.

Lignes de courant 2D

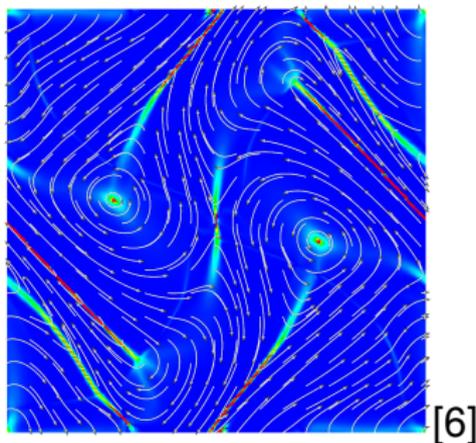
Ces notions amènent à une utilisation pratique des lignes de courant : en plaçant plusieurs sources dans une région d'intérêt, on s'assure de **suivre l'évolution des points passant par cette région.**



Lignes de courant 2D

Cependant, même en augmentant le nombre de sources utilisées dans cette région, **rien ne garantit que la totalité du champ vectoriel sera traversé** par des lignes de courant.

Selon les directions présentes dans le champ vectoriel, il se peut que de grandes portions de celui-ci n'aient **aucune information visuelle**.



Lignes de courant 2D

Au lieu de se contenter d'une région d'intérêt, on peut **répartir uniformément un grand nombre de sources** sur la totalité du champ vectoriel. L'objectif devient alors d'illustrer **l'ensemble des trajectoires possibles**.

Le résultat visuel devrait alors respecter certaines propriétés :

- **Couverture** : tous les points dans l'ensemble de données doivent être à proximité d'au moins une ligne de courant de manière à avoir une idée du comportement du champ vectoriel en ce point.

Lignes de courant 2D

- **Uniformité** : la densité des lignes de courant doit être contenue entre des valeurs minimale et maximale. On assure ainsi que toutes les régions de l'ensemble de données soient traversées par des lignes, tout en limitant autant que possible le croisement entre celles-ci.
- **Continuité** : on préférera les lignes de courant ayant une certaine longueur afin de favoriser l'interprétation sous la forme de trajectoire.

Lignes de courant 2D

Comme c'était le cas avec les glyphes, **le recouvrement est aussi un grave problème** avec les lignes de courant. Avec les glyphes, la seule solution envisageable était de réduire la densité de la représentation.

Bien que cette solution soit toujours disponible avec les lignes de courant, elle n'est plus la seule. Une meilleure solution est de **poursuivre le traçage d'une ligne tant qu'elle reste à une certaine distance des lignes déjà tracées**. Si elle atteint ce seuil de distance, on arrête tout simplement de la dessiner.

Lignes de courant 2D

Avec une telle pratique, on règle le problème de l'**uniformité**. Celui de la **couverture** peut être facilement réglé en forçant la présence de sources supplémentaires dans les régions où la densité des lignes n'est pas assez grande.

Quant au problème de la **continuité**, on peut simplement introduire un seuil minimal pour la longueur des lignes. En deçà de celui-ci, les lignes ne seront tout simplement pas affichées.

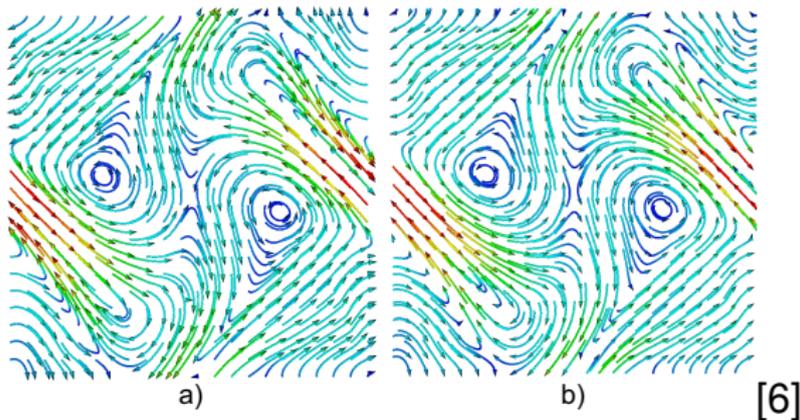
Tubes de courant 2D

Comme on l'a mentionné précédemment, cette forme de visualisation n'est **nullement limitée aux lignes**. Par exemple, en déplaçant un cercle le long d'une ligne de courant, on obtient un **tube de courant**.

Plus formellement, on positionne un cercle de rayon r de façon à ce qu'il soit **perpendiculaire à la tangente de la ligne** en tout point de celle-ci.

Tubes de courant 2D

L'espace supplémentaire occupé par les tubes rend aussi possible l'ajout d'un glyphe à une extrémité afin d'indiquer **la direction de la trajectoire**.



Tubes de courant 2D

Les deux figures précédentes présentent le même nombre de trajectoires pour un échantillon uniforme de l'espace. La différence réside dans **l'utilisation qui est faite de cet échantillon**.

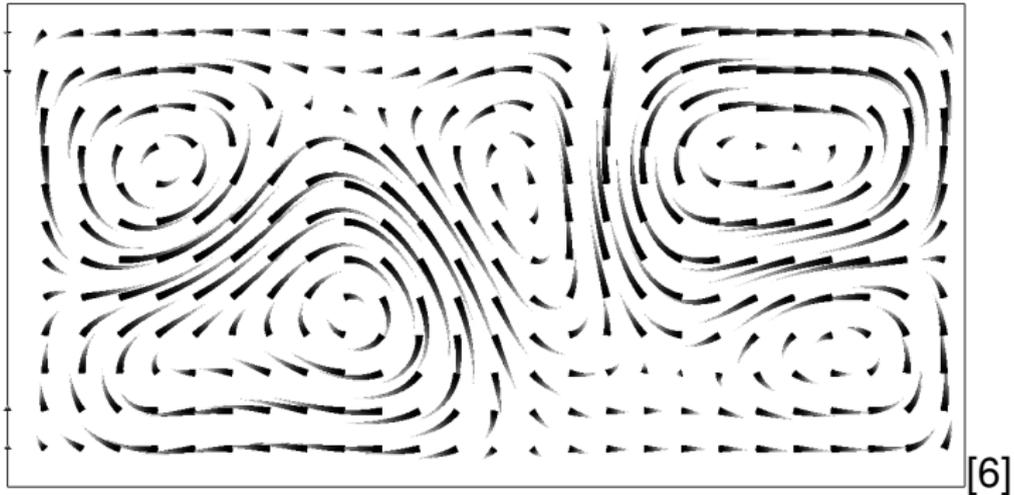
Dans la figure de gauche, les positions échantillonnées sont utilisées comme **points de départ** des trajectoires (Δt positif), tandis que dans l'image de droite, ce sont les **positions d'arrivée** qui sont uniformément distribuées sur une grille (Δt négatif).

Tubes de courant 2D

Le rayon r d'un tube n'a pas forcément à être fixe. On peut **l'utiliser comme un paramètre** supplémentaire pour transmettre de l'information. Par exemple, il est possible de **moduler le rayon en fonction de la vorticit ** du champ vectoriel.

On doit toutefois **utiliser ce nouveau degr  de libert  avec prudence**. D'une part, il sera difficile de distinguer les variations quand le rayon sera petit et d'autre part, un rayon trop grand occupera davantage d'espace   l' cran et pourra causer de l'occlusion.

Tubes de courant 2D



Lignes et tubes en 3D

Bien que le passage du 2D au 3D se fasse de façon plutôt naturelle, il apporte son lot de défis associés au **choix d'un bon échantillonnage** afin de respecter les propriétés de **couverture**, **densité** et **continuité**.

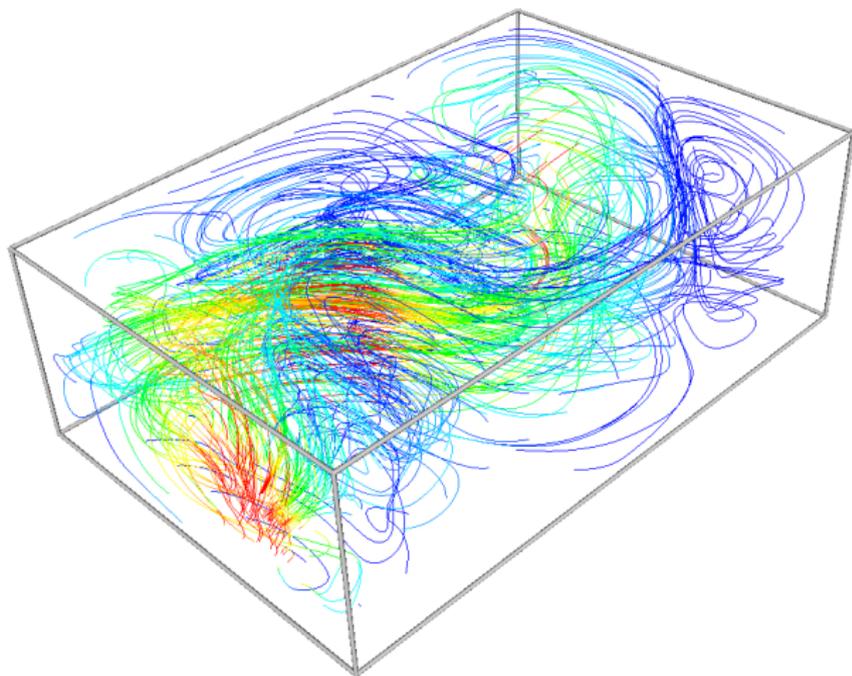
Les figures suivantes illustrent différentes stratégies d'échantillonnage pour la construction de lignes de courant dans un champ vectoriel 3D.

Lignes et tubes en 3D

Dans tous les cas, on utilise un ensemble de **sources uniformément distribuées par sous-échantillonnage** dans un champ vectoriel de dimension $128 \times 85 \times 42$. Les lignes sont aussi **colorées selon l'amplitude** du champ vectoriel en un point donné.

Dans le premier exemple, on a échantillonné à un taux de $10 \times 10 \times 10$ et on a fixé une longueur maximale de 100 pour les lignes. En considérant si peu de sources, **on évite le recouvrement**, mais on obtient une représentation clairsemée qui n'apporte pas beaucoup d'information.

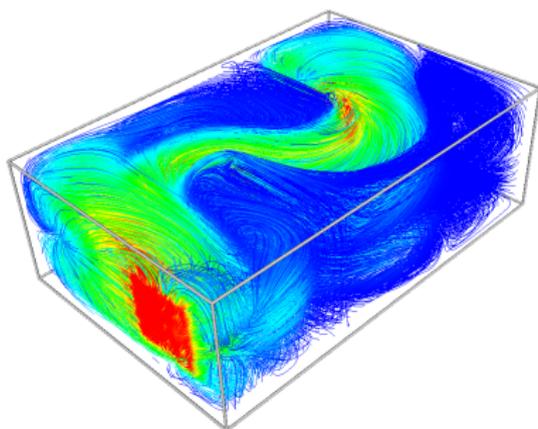
Lignes et tubes en 3D



[6]

Lignes et tubes en 3D

On peut **améliorer la couverture** en augmentant l'échantillonnage à un taux de $3 \times 3 \times 3$. Il est alors plus facile de suivre la progression du champ vectoriel.

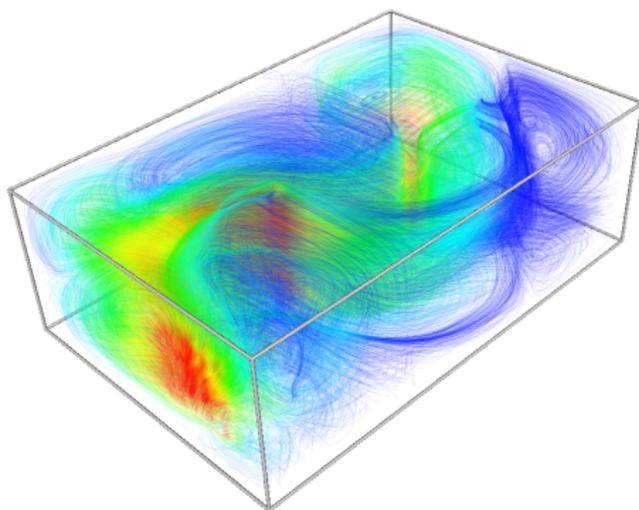


[6]

Avec cet échantillonnage plus dense, **l'occlusion devient cependant un problème.**

Lignes et tubes en 3D

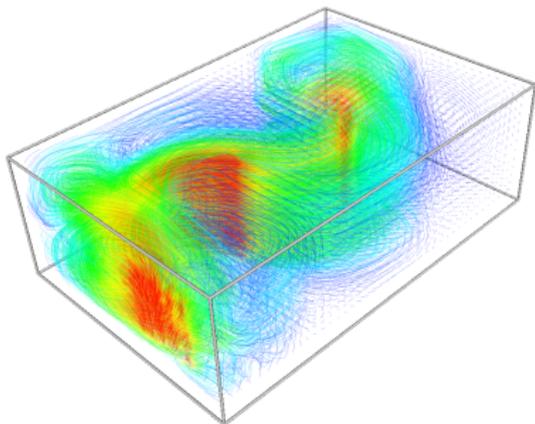
La solution est alors la même que pour les glyphes : **introduire un certain facteur de translucidité** dans le traçage des lignes.



[6]

Lignes et tubes en 3D

Au lieu de fixer une longueur maximale pour les lignes, on peut **fixer l'intervalle de temps** sur lequel les lignes sont calculées. On obtient ainsi **des lignes de longueurs différentes**.

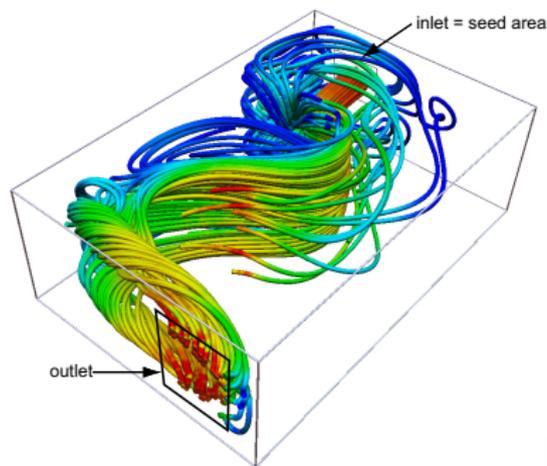


[6]

On peut voir cette technique comme une **généralisation directe des diagrammes en hérisson**, mais où les glyphes peuvent être des lignes courbes.

Lignes et tubes en 3D

Les **tubes de courant** prennent quant à eux tout leur sens dans un environnement 3D. En plus de **fournir la même information que les lignes** pour les trajectoires, les tubes ont l'avantage de **permettre l'utilisation du shading** puisqu'ils définissent des surfaces dans \mathbb{R}^3 .



[6]

Rubans et autres surfaces

Un autre membre de cette famille de techniques de visualisation est le **ruban de courant**. On crée celui-ci en démarrant deux lignes de courant à partir de deux sources situées à proximité.

La surface créée en affichant **l'ensemble des segments de plus courte distance** entre les deux lignes de courant forme le ruban.

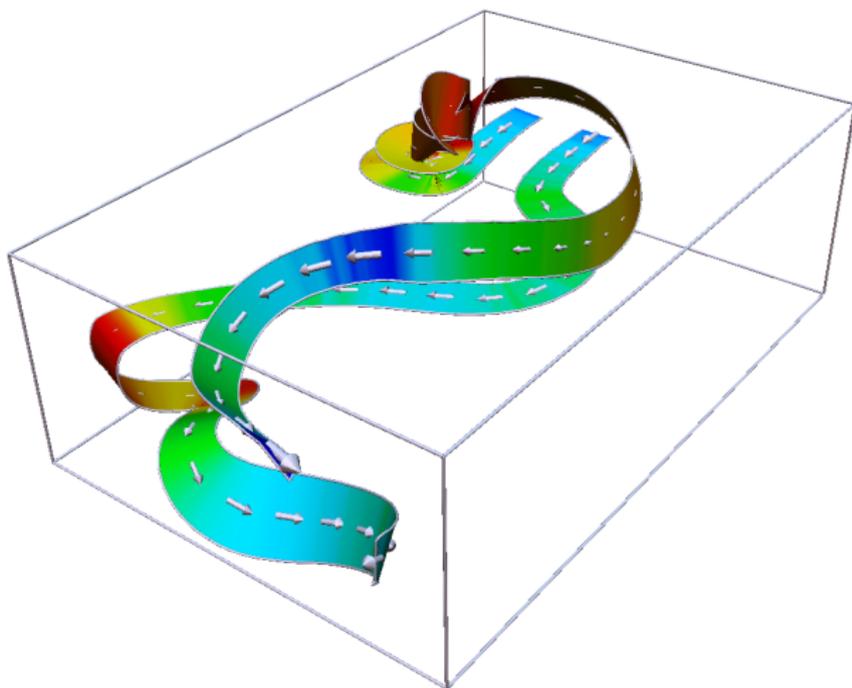
Cette façon de faire permet de plus facilement **visualiser la vorticit ** d'un champ vectoriel.

Rubans et autres surfaces

La figure suivante illustre simultanément deux rubans. La largeur de ceux-ci est **directement reliée à la distance entre les sources**, puis celle entre les lignes de courant qui leur sont associées.

Plus les sources sont éloignées, plus les lignes de courant risquent d'avoir des trajectoires différentes, ce qui générera un ruban très large et n'apportant aucune information pertinente.

Rubans et autres surfaces



[6]

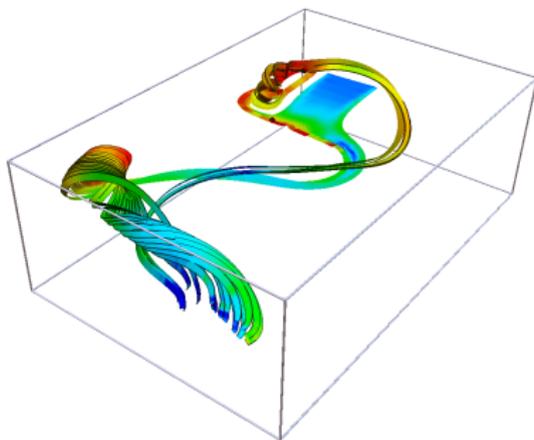
Rubans et autres surfaces

Dans cette figure, les rubans sont colorés du bleu au rouge **selon la vorticité du champ vectoriel**. On voit d'ailleurs que le ruban de gauche entre rapidement dans une région présentant une forte vorticité en raison de l'obstacle présent à cet endroit. À l'opposé, le ruban de droite présente une trajectoire plus directe vers la sortie.

On constate donc que dans cette forme de représentation, l'emphase est mise sur **l'identification rapide des régions de grande vorticité**.

Rubans et autres surfaces

Dans la figure suivante, on utilise la région délimitée par les deux rubans précédents, et **on en construit plutôt 20 ayant une largeur inférieure** afin de suivre le comportement du champ vectoriel avec plus de précision.



[6]

Vecteurs et textures

- 1 **Visualisation de vecteurs**
 - Déformation et déplacement
 - Représentation de trajectoires
 - **Vecteurs et textures**

2 Visualisation de tenseurs

3 Références

Principe de fonctionnement

Jusqu'à maintenant, on a vu deux approches pour la visualisation de champs vectoriels :

- les techniques reposant sur des **objets discrets** (glyphes, lignes de courant, etc.)
- les techniques proposant une **représentation dense** (code de couleurs, graphiques de déplacement)

Principe de fonctionnement

Avec la première approche, on obtenait un résultat visuel intéressant à partir duquel on pouvait aisément interpréter le comportement du champ vectoriel. Il était cependant nécessaire de **porter une attention particulière à l'échantillonnage des données** afin d'éviter des problèmes de recouvrement.

Avec le deuxième type de représentation, **on n'avait pas à se soucier de l'échantillonnage** ou du recouvrement, mais les résultats visuels étaient aussi plus difficiles à interpréter.

Principe de fonctionnement

Ne serait-il pas merveilleux d'avoir une technique qui combine la clarté des représentations discrètes, mais sans avoir à se soucier de l'échantillonnage ? C'est ce que propose de faire la **visualisation de champs vectoriels basée sur les textures**.

L'idée est donc de créer une texture qui contiendra l'information sur la **direction** et l'**amplitude** du champ vectoriel.

Principe de fonctionnement

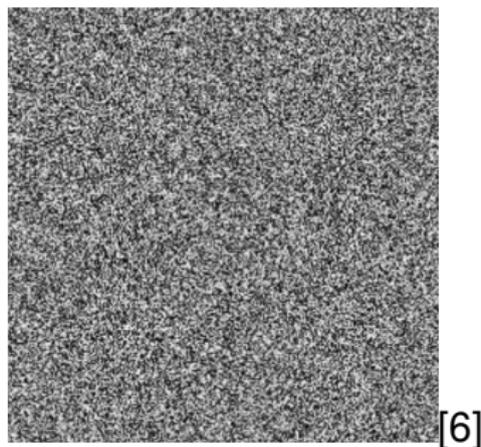
Pour encoder cette information, on dispose de **paramètres visuels** comme

- la luminance ;
- la couleur ;
- la granulation ;
- les motifs.

Le défi principal sera de **quantifier ces paramètres** en fonction de l'information contenue dans le champ vectoriel.

Principe de fonctionnement

Soit un champ vectoriel 2D de dimension $W \times H$. On définit une texture à niveaux de gris N de mêmes dimensions et dont la valeur des pixels sera déterminée par **une fonction de bruit aléatoire**.



Principe de fonctionnement

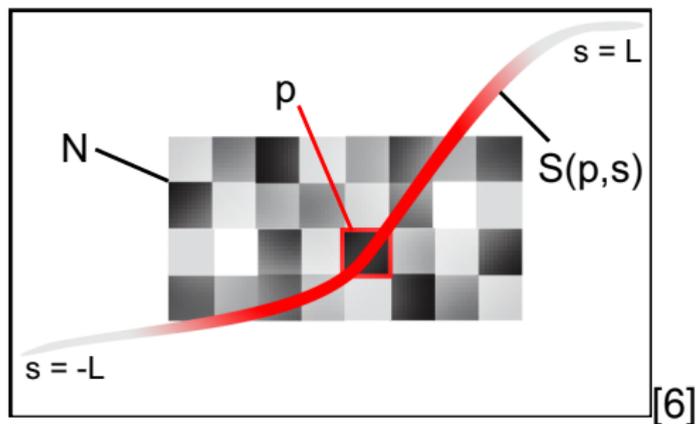
Pour chaque pixel p de ce domaine, on évalue **une ligne de courant ayant la position p comme source** pour des valeurs positives et négatives de Δt . On fixe la longueur maximale de cette ligne à L dans un sens et dans l'autre à partir de p .

On note cette ligne de courant $S(p, s)$, avec $s \in [-L, L]$.

Principe de fonctionnement

Soit T la texture de dimension $W \times H$ qui sera **la représentation visuelle du champ vectoriel**.

Chaque pixel $T(p)$ sera défini comme la somme pondérée des valeurs de N mesurées le long de la ligne de courant S passant par p .



Principe de fonctionnement

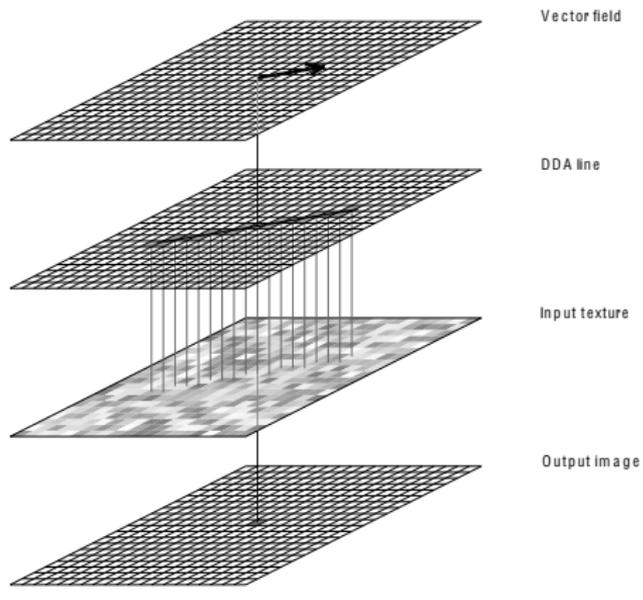
La valeur $T(p)$ s'exprime donc comme

$$T(p) = \frac{\int_{-L}^L N(S(p, s)) k(s) ds}{\int_{-L}^L k(s) ds}.$$

$k(s)$ est ici la **fonction de poids** associée aux valeurs de N . Un choix classique sera de prendre une Gaussienne :

$$k(s) = e^{-s^2}.$$

Principe de fonctionnement

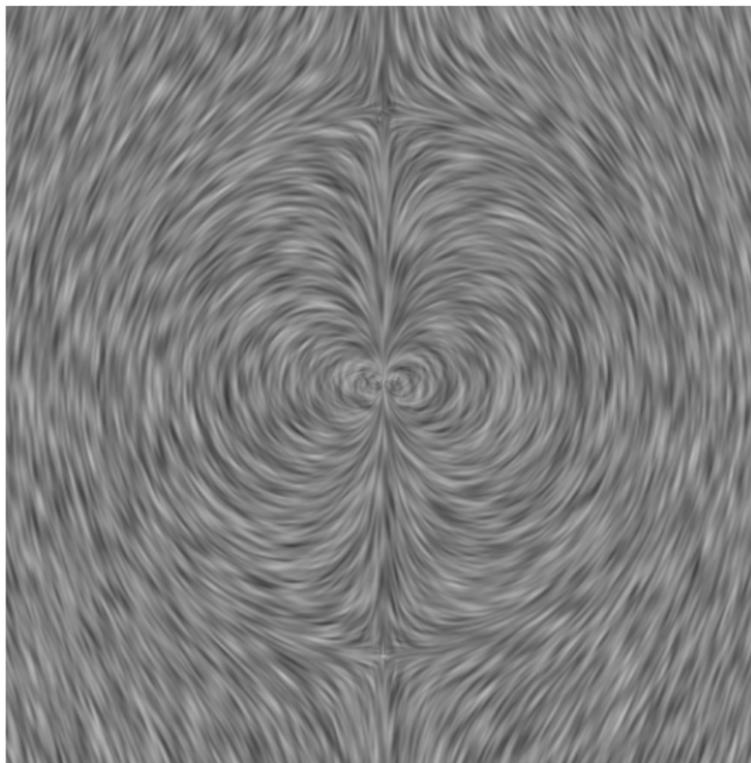


Principe de fonctionnement

Cette opération peut être vue comme la **convolution de la texture bruitée N avec le filtre k** le long de la ligne de courant S , ou encore comme une **convolution curviligne**.

Si on applique l'équation précédente sur tous les pixels, on obtient une texture T dont la couleur des pixels présente **peu de variation le long des lignes de courant**, mais affiche au contraire de grandes fluctuations entre les lignes de courant voisines.

Principe de fonctionnement



[6]

Principe de fonctionnement

On peut ainsi interpréter ce processus comme étant l'**application d'un filtre passe-bas** aux intensités de la texture bruitée le long des lignes de courant.

À la base, la visualisation de champs vectoriels basée sur la texture a été développée pour illustrer les flux de fluides, mais rien n'empêche d'utiliser cette technique sur n'importe quel champ vectoriel.

IBFV

On s'attarde maintenant à détailler une implantation possible pour la visualisation de champs vectoriels basée sur la texture.

Cette technique sera non seulement applicable à des **champs vectoriels statiques**, mais pourra aussi produire des animations pour des **champs vectoriels dynamiques**.

Cette méthode porte le nom de ***Image Based Flow Visualisation***, ou IBFV.

IBFV

La méthode présentée ici est celle qui s'applique à des **champs vectoriels 2D**, mais la généralisation au cas 3D est assez intuitive.

Pour comprendre le fonctionnement de la méthode IBVF, considérons une image I définie sur un domaine 2D D et variant dans le temps. On a donc que I définit une relation

$$I : D \times \mathbb{R}_+ \rightarrow \mathbb{R}_+ : I(x, t) \mapsto [0, 1].$$

IBFV

On définit l'**advection temporelle** de I dans un champ vectoriel dynamique $\mathbf{v} : D \times \mathbb{R}_+ \rightarrow \mathbb{R}^2$ par la formule

$$I(x + \mathbf{v}(x, t)\Delta t, t + \Delta t) = I(x, t).$$

C'est donc dire que l'intensité du pixel $x + \mathbf{v}(x, t)\Delta t$ dans l'image I au temps $t + \Delta t$ est donnée par l'intensité du pixel x au temps t .

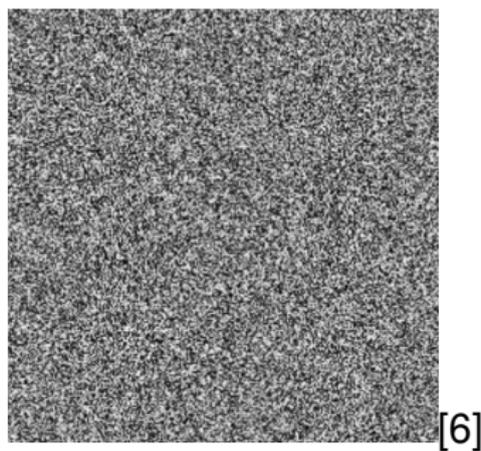
IBFV

Le pas d'incrémentation temporelle Δt sert à **discrétiser la variable temporelle** t , ce qui permet de résoudre itérativement l'équation précédente.

Comme on l'a mentionné dans la description générale de la méthode, l'objectif sera d'**appliquer l'advection temporelle sur une texture bruitée** de manière à obtenir des images présentant **peu de contraste le long des lignes de courant**, mais qui en possèdent beaucoup entre les lignes de courant voisines.

IBFV

Si on applique simplement **l'advection temporelle sur une image initiale statique** $I(x, 0) = N(x)$, où $N : D \rightarrow [0, 1]$ est une texture bruitée, on aura que, selon l'ordre de résolution de l'équation d'advection, certains points de I s'écraseront les uns les autres.



IBFV

Ce problème sera réglé en ajoutant un **terme d'injection** à l'équation d'advection. Par le fait même, on sera aussi en mesure de créer des animations de textures pour des champs vectoriels dynamiques.

On a donc

$$I(x + \mathbf{v}(x, t)\Delta t, t + \Delta t) = (1 - \alpha)I(x, t) + \alpha N(x + \mathbf{v}(x, t)\Delta t, t + \Delta t),$$

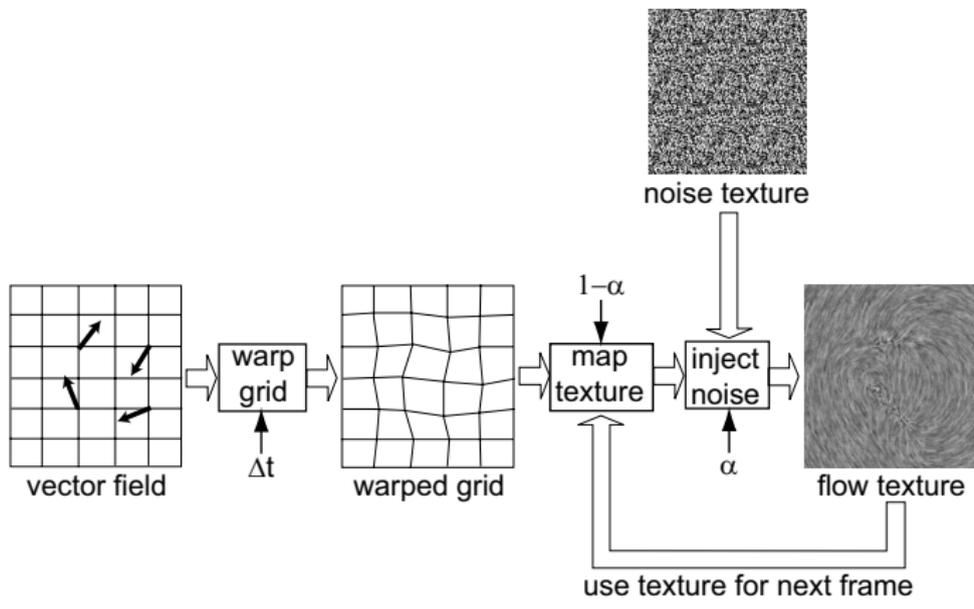
où $N(x, t)$ est le terme d'injection défini sur D et variant dans le temps.

IBFV

Le paramètre $\alpha \in [0, 1]$ contrôle le **ratio advection/injection**. En pratique, on prendra généralement une valeur $\alpha \in [0, 0.2]$.

On obtiendra ainsi une image qui présente à la fois des **variations locales** (dues au terme d'injection bruitée N) et une **cohérence le long des lignes de courant** (à cause de l'équation d'advection).

IBFV



IBFV

Une question demeure, soit celle de savoir **quelle texture bruitée** $N(x, t)$ **utiliser** comme terme d'injection.

On considère deux possibilités :

- une **texture statique** $N(x)$ définie sur D ;
- une **texture variant dans le temps** $N(x, t)$ définie sur $D \times \mathbb{R}_+$.

On s'attardera d'abord au cas de la **texture statique**.

IBFV

Pour réussir à créer une image présentant un **contraste élevé entre les différentes lignes de courant**, les pixels voisins de $N(x)$ doivent posséder des intensités différentes.

On peut facilement atteindre ce résultat en définissant N comme **un signal aléatoire composé de blocs noirs ou blancs**. En pratique, une taille $d \in [2, 10]$ pixels pour ces blocs donne de bons résultats.

IBFV

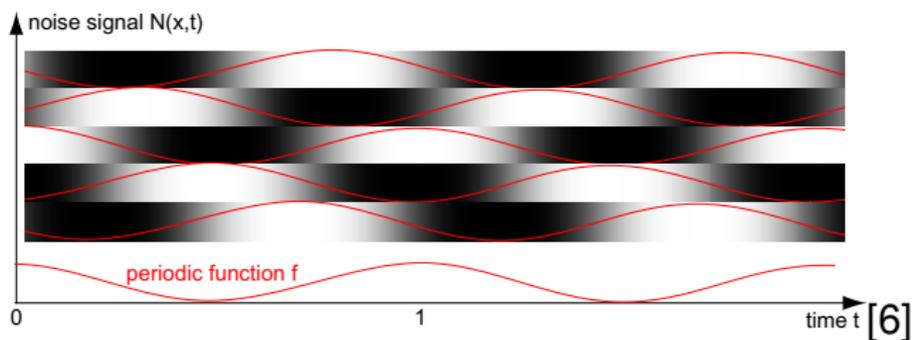
Si l'on souhaite obtenir **des animations de textures** plutôt que des représentations fixes, on doit définir un **terme d'injection dynamique** $N(x, t)$ à partir de la texture bruitée originale $N(x)$:

$$N(x, t) = f((t + N(x)) \bmod 1),$$

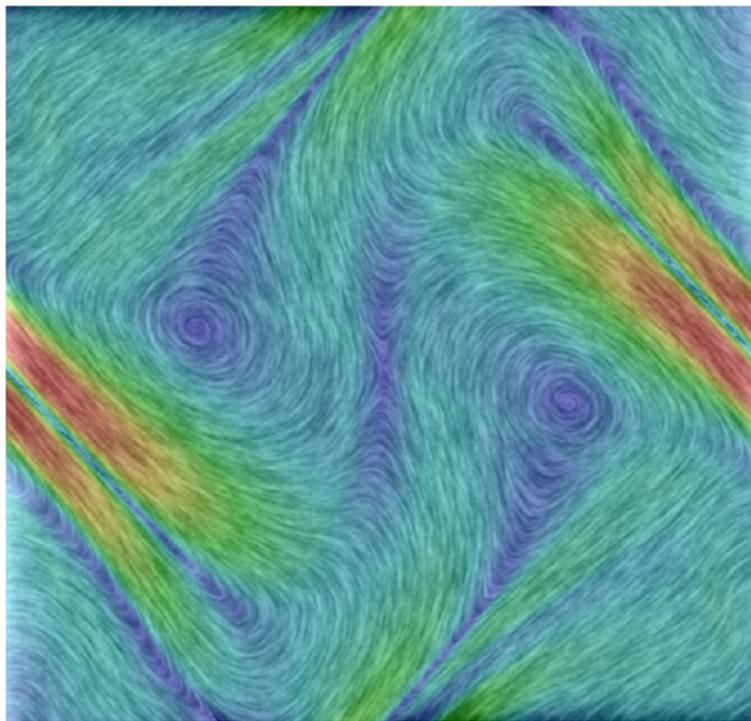
où $f : \mathbb{R}_+ \rightarrow [0, 1]$ est une fonction périodique de période 1.

IBFV

De manière intuitive, l'équation précédente dit que l'intensité de chaque pixel $x \in D$ de la texture bruitée dynamique $N(x, t)$ **oscille dans le temps selon la fonction périodique f** à partir d'un état initial aléatoire donné par la texture $N(x)$.

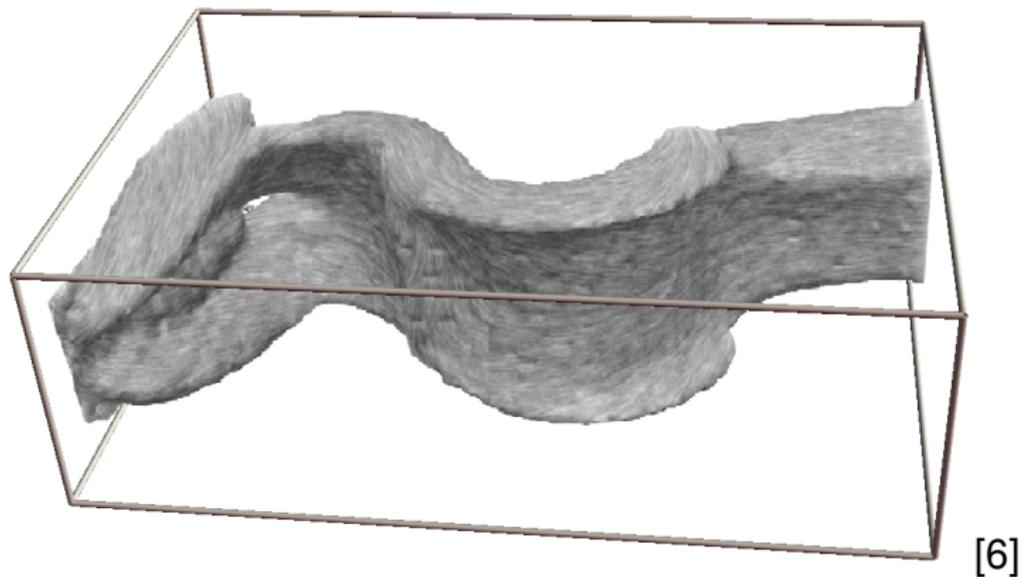


IBFV



[6]

IBFV



Visualisation de tenseurs

1 Visualisation de vecteurs

2 **Visualisation de tenseurs**

- Introduction aux tenseurs
- Tenseurs et ACP
- Représentation par glyphes
- Lignes de courant tensorielles

3 Références

Introduction aux tenseurs

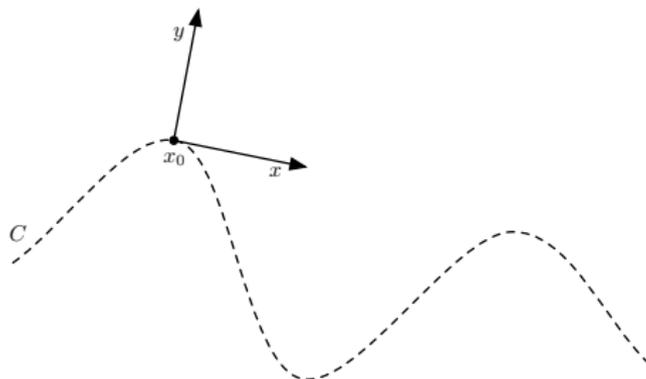
- 1 Visualisation de vecteurs
- 2 Visualisation de tenseurs**
 - Introduction aux tenseurs
 - Tenseurs et ACP
 - Représentation par glyphes
 - Lignes de courant tensorielles
- 3 Références

Qu'est-ce qu'un tenseur ?

Un tenseur est une **généralisation à haute dimension** des concepts de **vecteurs** et de **matrices**. Pour expliquer le principe, considérons la tâche de **mesurer la courbure** d'objets géométriques.

Attardons nous premièrement à une **courbe plane** C . Pour chaque point x_0 sur C , on définit un repère local xy tel que l'axe des x est tangent à la courbe et l'axe des y est normal à celle-ci.

Qu'est-ce qu'un tenseur ?



Dans le voisinage de x_0 dans ce nouveau repère, on peut définir la courbe C par une forme $y = f(x)$ avec $f(x_0) = 0$. La **courbure de C** est alors donnée par la dérivée seconde :

$$\frac{\partial^2 f}{\partial x^2}.$$

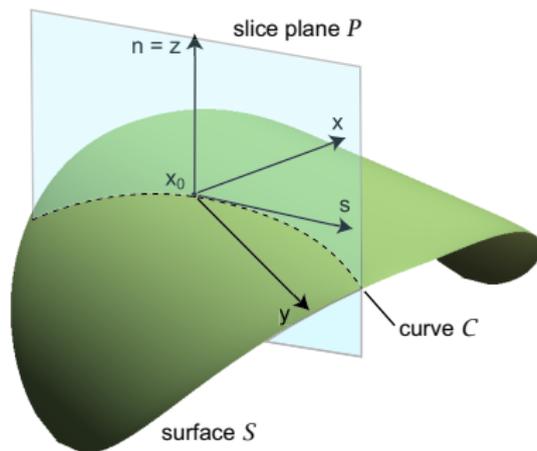
Qu'est-ce qu'un tenseur ?

La courbure d'une courbe permet de décrire comment **de petits changements dans la position sur C influencent la direction de la normale n** , ou encore de savoir à quel point l'orientation d'une courbe dévie de la droite tangente en un point donné.

Plus la normale change, plus la courbe diffère de la droite tangente et **plus la courbure est élevée**.

Qu'est-ce qu'un tenseur ?

Reprenons la même idée, mais cette fois-ci avec **une surface S à deux dimensions**. Pour chaque point x_0 sur S , on définit le repère local xyz tel que les axes x et y sont tangents à S et que l'axe des z pointe dans la direction de la normale \mathbf{n} à la surface S .



[6]

Qu'est-ce qu'un tenseur ?

Comme tout à l'heure, dans le voisinage de x_0 , on peut définir la surface S par $z = f(x, y)$ où $f(x_0) = 0$. Toujours comme dans le cas planaire, la courbure décrit **comment un petit déplacement sur S influence l'orientation de la normale n .**

Il y a toutefois un problème avec cette notion : alors que dans le cas planaire, le mouvement autour d'un point peut se faire dans seulement deux directions, dans le cas d'une surface **il y a une infinité de directions possibles.**

Qu'est-ce qu'un tenseur ?

La conséquence est que ledit mouvement **ne peut être exprimé par un seul nombre**. Pour régler ce problème, considérons le gradient \mathbf{g} de la surface S . Celui-ci pointe dans la direction de la normale à la surface.

La variation de \mathbf{g} dans une certaine direction \mathbf{s} est donnée par

$$\mathbf{g}(x_0 + \mathbf{s}) = \mathbf{g}(x_0) + H(x_0)\mathbf{s},$$

où H est une matrice 2×2 appelée le **hessien**.

Qu'est-ce qu'un tenseur ?

Cette matrice H contient **les dérivées partielles de la fonction** $f(x, y)$ dans le repère local défini au point x_0 . C'est donc dire que

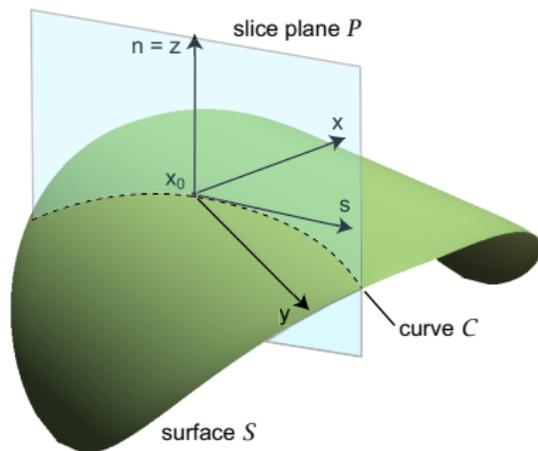
$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}.$$

Ce faisant, on a que

$$\frac{\partial^2 f}{\partial \mathbf{s}^2}(x_0) = \mathbf{s}^T H \mathbf{s}.$$

Qu'est-ce qu'un tenseur ?

L'équation précédente permet donc d'**obtenir la courbure de S** en tout point x_0 et dans toute direction \mathbf{s} . Cette valeur est identique à la courbure d'une courbe obtenue par **l'intersection de la surface S avec un plan** contenant les vecteurs \mathbf{n} et \mathbf{s} .



[6]

Qu'est-ce qu'un tenseur ?

Le problème est que, pour obtenir la courbure de S en un point quelconque x_0 , on doit **construire un repère local**, puis **évaluer la courbe** dans celui-ci.

On peut simplifier ce processus en considérant la surface S **dans un repère global plutôt que local**. C'est donc dire qu'on l'exprimera sous la forme d'une fonction implicite $f(x, y, z) = 0$.

Qu'est-ce qu'un tenseur ?

On peut ainsi obtenir la courbure de la surface S en un point x_0 et dans une direction \mathbf{s} avec la même formule que tout à l'heure, c'est-à-dire

$$\frac{\partial^2 f}{\partial \mathbf{s}^2}(x_0) = \mathbf{s}^T H \mathbf{s},$$

mais où H est cette fois-ci la **matrice hessienne** 3×3 des dérivées partielles de $f(x, y, z)$ **dans le repère global**.

Qu'est-ce qu'un tenseur ?

On a ainsi

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix}.$$

Cette matrice hessienne H est appelée le **tenseur de courbure** de la surface S .

Qu'est-ce qu'un tenseur ?

En général, un tenseur décrit le comportement d'une quantité physique qui varie selon la direction, comme **la diffusivité de l'eau** ou encore **la force appliquée sur un matériel**.

Les tenseurs sont caractérisés par leur rang : un **scalaire** est un tenseur de rang 0, un **vecteur** est un tenseur de rang 1, une **matrice** est un tenseur de rang 2, etc.

Qu'est-ce qu'un tenseur ?

En général, un tenseur de rang k est **un tableau de valeurs réelles** de dimension k .

Le tenseur de courbure associé à la matrice H est un type particulier de tenseur de rang 2, puisqu'il est **symétrique**. En visualisation, ce type de tenseurs est le plus commun et notre étude des techniques de visualisation portera essentiellement sur celui-ci.

Lorsqu'on les représente dans un repère global, ces tenseurs sont représentés par une matrice 3×3 . C'est donc dire que **chaque point dans un ensemble de données tensorielles est représenté par une telle matrice**.

Exemple d'application

Comme on l'a mentionné précédemment, les données tensorielles permettent de représenter des propriétés spatiales **variant en fonction de la position et de la direction**.

Les tenseurs sont utiles dans plusieurs domaines, mais dans le cadre de ce chapitre, on s'intéressera principalement à l'utilisation de ceux-ci pour **mesurer la diffusion de l'eau** dans des tissus organiques.

Exemple d'application

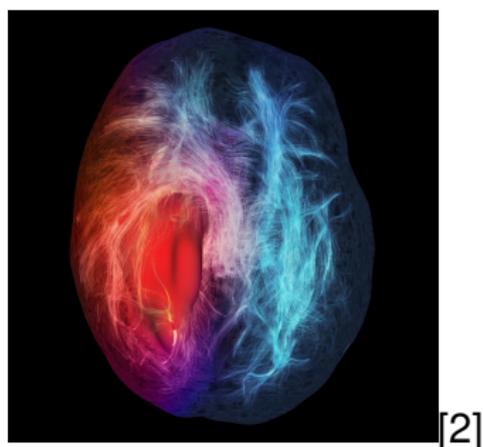
La diffusion de l'eau dans les tissus peut se faire de deux façons :

- de manière **isotrope**, c'est-à-dire à vitesse égale dans toutes les directions ; ou bien
- de manière **anisotrope**, c'est-à-dire à des vitesses différentes selon la direction.

À titre d'exemple, dans le cerveau humain, la diffusion est **plus forte dans la direction des fibres neuronales**, et plus faible à travers celles-ci.

Exemple d'application

Ces fibres, composées de groupes d'axones, forment la **matière blanche** du cerveau. Celle-ci tient son nom de la couleur caractéristique de la couche de myéline qui les recouvre.



À chaque point dans le volume de tissus, la diffusion de l'eau est représentée par **un tenseur de diffusion 3×3** .

Exemple d'application

Sachant que la diffusion est plus forte dans la direction des fibres, on peut obtenir de l'**information sur la structure des fibres neuronales** dans le cerveau humain en visualisant les données contenues dans un tenseur de diffusion.

La mesure de la diffusion de l'eau dans les tissus organiques se fait grâce à des techniques d'acquisition comme le **DT-MRI** (*diffusion tensor magnetic resonance imaging*).

Exemple d'application

On a constaté depuis le début du cours que la visualisation d'un champ vectoriel 3D n'était pas simple. On rencontrait **des problèmes d'échantillonnage et d'occultation**. Tout ça en ayant à associer à chaque point d'un volume de données un vecteur (x, y, z) .

Dans le cas de la visualisation de tenseurs, le problème devient encore plus complexe, puisqu'il faut maintenant trouver **une représentation graphique pour une matrice 3×3 à chaque point** du volume de données.

Exemple d'application

On pourrait bien entendu **visualiser chacun des éléments des tenseurs individuellement**, en utilisant une technique de visualisation scalaire. Cela ne permettrait cependant pas de saisir comment les données des tenseurs varient en fonction de la direction.

On verra plutôt comment combiner les données contenues dans un tenseur à l'aide de l'**analyse en composantes principales** pour permettre une visualisation simple et intuitive.

Tenseurs et ACP

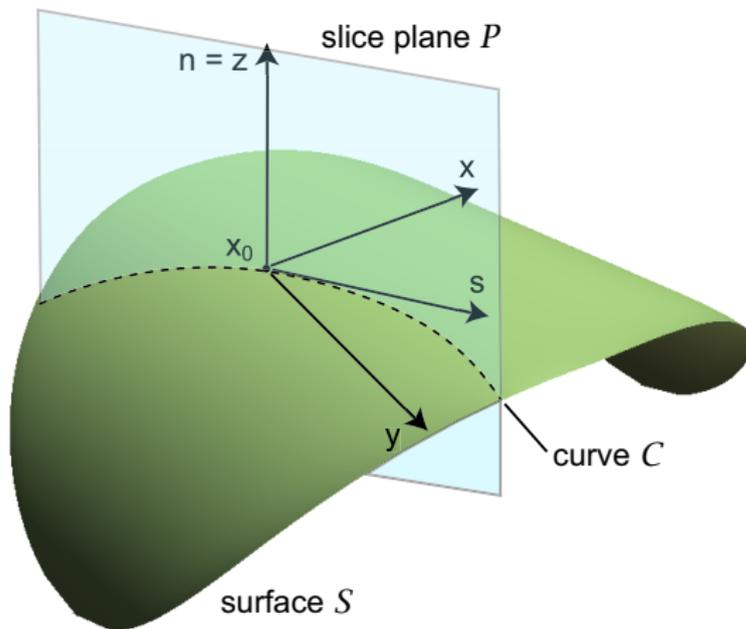
- 1 Visualisation de vecteurs
- 2 Visualisation de tenseurs**
 - Introduction aux tenseurs
 - **Tenseurs et ACP**
 - Représentation par glyphes
 - Lignes de courant tensorielles
- 3 Références

Retour sur la théorie

Revenons brièvement sur l'exemple du **tenseur de courbure d'une surface 3D** et considérons un repère xyz local centré en un point x_0 sur la surface, de telle sorte que l'axe z coïncide avec la normale \mathbf{n} de la surface au point x_0 .

On a dit que dans ce repère, on pouvait définir la surface sous la forme d'une fonction $z = f(x, y)$ avec $f(x_0) = 0$.

Retour sur la théorie



[6]

Retour sur la théorie

On a vu qu'il était possible d'évaluer la courbure de la surface en un point x_0 dans une certaine direction \mathbf{s} dans le plan tangent xy en utilisant la **matrice hessienne** 2×2 des dérivées partielles de second ordre de f .

Dans beaucoup d'applications, on ne s'intéresse toutefois qu'aux **valeurs maximale et minimale de la courbure**. On cherche donc les directions \mathbf{s} qui sont associées à ces valeurs extrêmes.

Retour sur la théorie

Comme \mathbf{s} représente une direction sur le plan tangent à la surface, on peut l'exprimer sous la forme d'un **vecteur unitaire**, c'est-à-dire

$$\mathbf{s} = \begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix},$$

où α est l'angle entre \mathbf{s} et l'axe x du repère local.

À la section précédente, on avait dit que la courbure était donnée par l'équation

$$\frac{\partial^2 f}{\partial \mathbf{s}^2} = \mathbf{s}^T H \mathbf{s}.$$

Voyons comment combiner cette équation avec la nouvelle formulation pour \mathbf{s} .

Retour sur la théorie

On a donc

$$\frac{\partial^2 f}{\partial \mathbf{s}^2} = \mathbf{s}^T H \mathbf{s} = h_{11} \cos^2 \alpha + (h_{12} + h_{21}) \sin \alpha \cos \alpha + h_{22} \cos^2 \alpha,$$

où les h_{ij} sont les éléments de la matrice hessienne 2×2 H .

On trouve **les extremums de cette fonction** quand sa dérivée est nulle, c'est-à-dire quand

$$-h_{11} \cos \alpha \sin \alpha - \frac{h_{12} + h_{21}}{2} (\sin^2 \alpha - \cos^2 \alpha) + h_{22} \sin \alpha \cos \alpha = 0.$$

Retour sur la théorie

L'équation précédente est **équivalente au système d'équations**

$$h_{11} \cos \alpha + h_{12} \sin \alpha = \lambda \cos \alpha$$

$$h_{21} \cos \alpha + h_{22} \sin \alpha = \lambda \sin \alpha$$

où λ est une quelconque valeur réelle.

Ce système d'équations peut alors être réécrit sous la forme

$$H\mathbf{s} = \lambda\mathbf{s}.$$

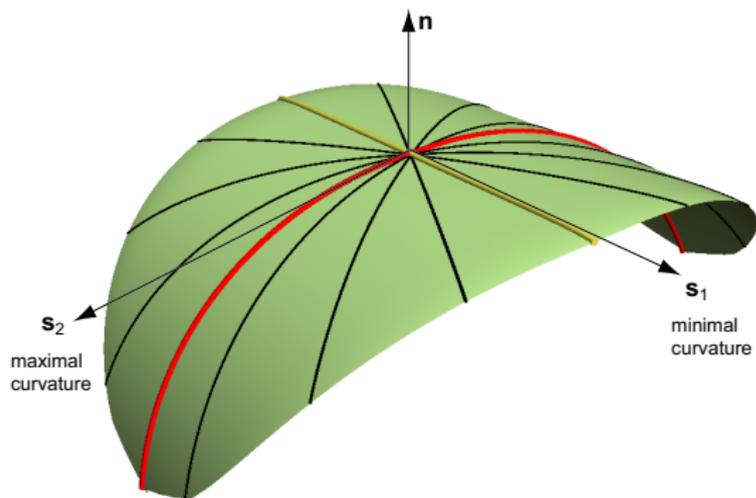
Ainsi, les directions **\mathbf{s} associées aux valeurs extrêmes de la courbure** sont les solutions du système d'équations.

Retour sur la théorie

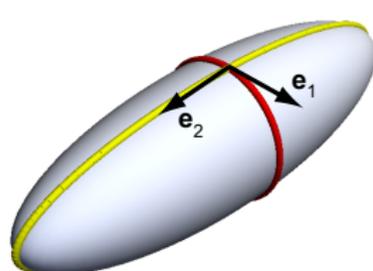
Pour une matrice 2×2 , on peut trouver **une solution analytique** au système d'équation précédent. On obtient alors deux paires de solutions : λ_1 associé à la direction \mathbf{s}_1 et λ_2 associé à la direction \mathbf{s}_2 .

\mathbf{s}_1 et \mathbf{s}_2 sont les **directions principales**, c'est-à-dire celles auxquelles on associe les valeurs maximale et minimale de la courbure. La figure suivante affiche une courbure minimale dans la direction \mathbf{s}_1 (jaune) et maximale dans la direction \mathbf{s}_2 (rouge).

Retour sur la théorie

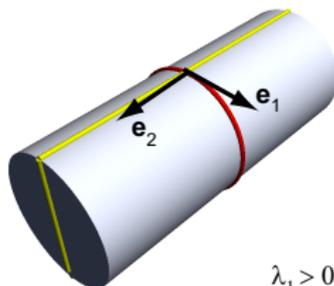


Retour sur la théorie



$$\lambda_1 > \lambda_2 > 0$$

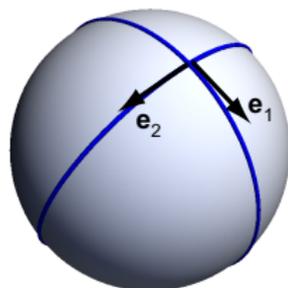
a)



$$\lambda_1 > 0$$

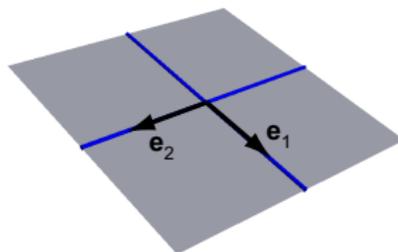
$$\lambda_2 = 0$$

b)



$$\lambda_1 = \lambda_2 > 0$$

c)



$$\lambda_1 = \lambda_2 = 0$$

d)

[6]

Retour sur la théorie

Le même raisonnement peut être appliqué dans le cas d'un **tenseur** 3×3 . Considérons par exemple le cas d'une surface 3D représentée par l'équation implicite $f(x, y, z) = 0$ dans un repère global.

Ici, la matrice hessienne H sera de dimension 3×3 et possédera **trois vecteurs propres** (directions principales). Celles-ci pourront être obtenues efficacement à l'aide de la **méthode itérative de Jacobi** qui trouve une solution numérique au système d'équation $\mathbf{s}^T H \mathbf{s}$.

Retour sur la théorie

On s'attarde maintenant à décrire des techniques de **visualisation pour des ensembles de données tensorielles**.

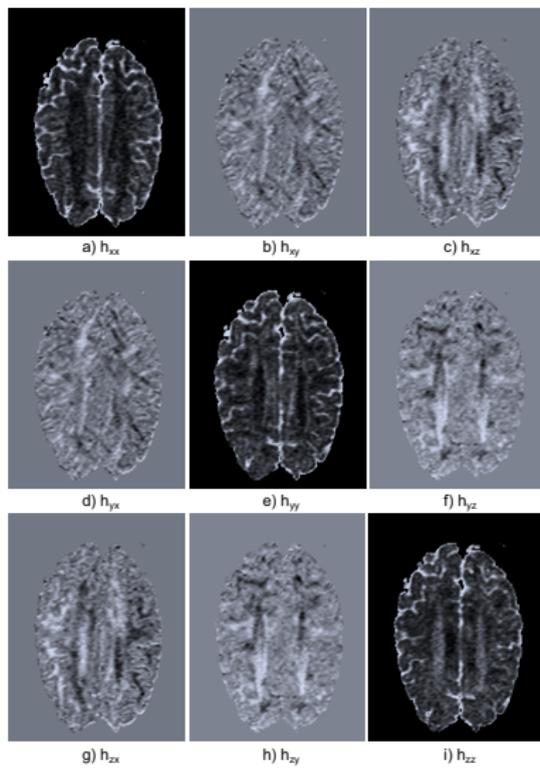
Il sera question, dans un premier temps, de la simple **visualisation scalaire** des composantes d'un tenseur, puis nous arriverons à l'utilisation des résultats obtenus avec l'**analyse en composantes principales**. Ceux-ci permettront d'exprimer davantage d'information pertinente dans une même image.

Visualisation des composantes d'un tenseur

La façon la plus simple de visualiser un ensemble de données tensorielles est de simplement **le traiter comme une série d'ensembles de données scalaires**. Par exemple, pour un ensemble de tenseurs 3×3 , on peut considérer **chacune de ses neuf composantes** comme un ensemble de données scalaires.

La figure suivante illustre ce principe pour **une tranche d'un ensemble de données tensorielles volumétriques** représentant la diffusion de l'eau dans le cerveau humain.

Visualisation des composantes d'un tenseur



[6]

Visualisation des composantes d'un tenseur

Chaque composante du tenseur est affichée en utilisant un **code de couleur associant la valeur scalaire à la luminance**.

De plus, en raison de la **symétrie du tenseur**, il n'y a que six images différentes au lieu de neuf. En effet, on a

$$h_{12} = h_{21}$$

$$h_{13} = h_{31}$$

$$h_{23} = h_{32}$$

Visualisation des composantes d'un tenseur

En général, les composantes d'un tenseur représentent **les dérivées partielles de second ordre** de la quantité qu'on souhaite illustrer, dans le repère global. Ainsi, prises séparément, les composantes d'un tenseur **n'ont pas de réelle signification**.

À l'opposé, la visualisation basée sur les vecteurs propres et les valeurs propres d'un tenseur rendront les quantités **indépendantes du repère utilisé**. C'est là que l'analyse en composantes principales entre en jeu.

Visualisation scalaire basée sur l'ACP

Une meilleure approche pour la visualisation de données tensorielles est de mettre l'emphasis sur les données obtenues par l'analyse en composantes principales, c'est-à-dire les **vecteurs propres** (directions principales) et les **valeurs propres**.

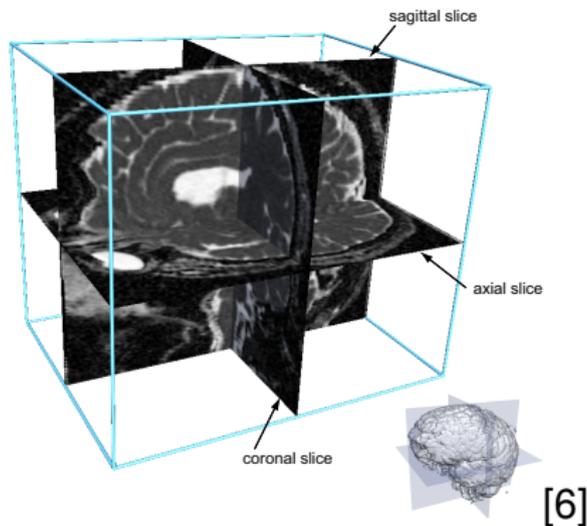
Comme premier exemple, considérons la donnée scalaire obtenue en prenant la moyenne des valeurs propres du tenseur :

$$\frac{1}{3} (\lambda_1 + \lambda_2 + \lambda_3).$$

Dans le cas d'un tenseur de diffusion, cette quantité est égale à la **diffusion moyenne** mesurée dans le tissus, en chaque point.

Visualisation scalaire basée sur l'ACP

La figure suivante illustre cette quantité scalaire sur **trois plans alignés avec les axes** en utilisant un code de couleurs basé sur la luminance.



Visualisation scalaire basée sur l'ACP

On se rappelle que les vecteurs propres d'un tenseur donnent **les directions de variation maximale et minimale** de la quantité représentée. Les valeurs propres donnent quant à elle **les valeurs de ces variations**.

Dans le cas d'un tenseur de diffusion, les valeurs propres peuvent être utilisées pour **décrire le degré d'anisotropie** du tissu organique en un point du volume.

Visualisation scalaire basée sur l'ACP

Dans un **milieu isotrope**, les directions principales auront **les mêmes valeurs propres**. Dans le cas de la diffusion, cela signifie que celle-ci sera la même dans toutes les directions autour d'un point.

Les **matériaux anisotropes** présentent quant à eux des propriétés **différentes selon la direction**. C'est la visualisation de l'anisotropie d'un tenseur de diffusion qui permet de distinguer les fibres neuronales du reste des tissus, puisque **celles-ci possèdent une grande anisotropie**.

Visualisation scalaire basée sur l'ACP

On peut par exemple définir des mesures estimant les probabilités c_l , c_p et c_s qu'un tenseur ait une forme **linéaire**, **planaire** ou **sphérique**, respectivement.

Pour cela, considérons **les trois valeurs propres** $\lambda_1 \geq \lambda_2 \geq \lambda_3$ associées à un tenseur. On définit les probabilités comme suit :

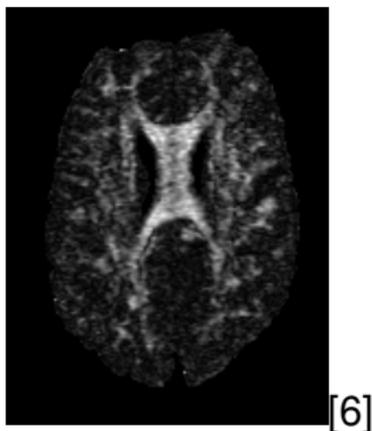
$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \quad c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

Visualisation scalaire basée sur l'ACP

Ces valeurs servent à évaluer à quel point une des valeurs propres est éloignée des plus petites. On peut ainsi savoir, avec une certaine probabilité, **comment se comportera la diffusion de l'eau** en tout point de l'ensemble de données.

Une façon simple de **visualiser l'anisotropie** d'un ensemble de données tensorielles est d'afficher la probabilité c_l à l'aide d'un code de couleur. Des valeurs élevées de c_l indiquent des régions où les fibres sont clairement définies.

Visualisation scalaire basée sur l'ACP



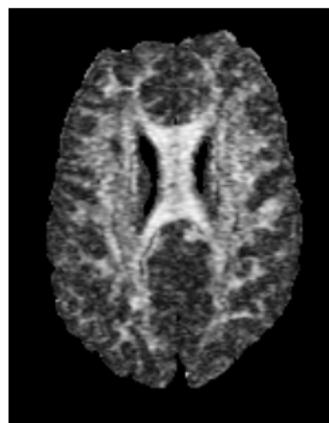
La zone blanche au milieu indique une **région de forte anisotropie** connue sous le nom de **corps calleux**. Les portions en gris indiquent quant à elle les régions de faible anisotropie et correspondent à la **matière grise**.

Visualisation scalaire basée sur l'ACP

Une autre mesure fréquemment utilisée est l'**anisotropie fractionnelle**. Cette valeur scalaire est définie par

$$FA = \sqrt{\frac{3}{2}} \frac{\sqrt{\sum_{i=1}^3 (\lambda_i - \mu)^2}}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2},$$

où $\mu = \frac{1}{3} (\lambda_1 + \lambda_2 + \lambda_3)$ est la **diffusion moyenne**.

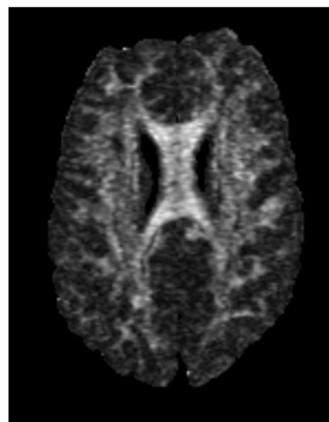


Visualisation scalaire basée sur l'ACP

Finalement, une dernière mesure scalaire communément utilisée est l'**anisotropie relative** qui est définie par

$$RA = \sqrt{\frac{3}{2} \frac{\sqrt{\sum_{i=1}^3 (\lambda_i - \mu)^2}}{\lambda_1 + \lambda_2 + \lambda_3}},$$

où $\mu = \frac{1}{3} (\lambda_1 + \lambda_2 + \lambda_3)$ est la **diffusion moyenne**.



[6]

Visualisation scalaire basée sur l'ACP

En résumé, les techniques précédentes réduisent le problème de visualiser un ensemble de données tensorielles à la **visualisation de données scalaires**.

On verra maintenant comment tirer de l'**information vectorielle** à partir des données tensorielles. La perte d'information sera ainsi moins important qu'avec la visualisation scalaire.

Visualisation vectorielle basée sur l'ACP

Avec les techniques précédentes, on a vu comment **visualiser différentes mesures d'anisotropie** calculées à partir des résultats de l'ACP appliquée à un ensemble de données tensorielles. Ces mesures étaient des scalaires.

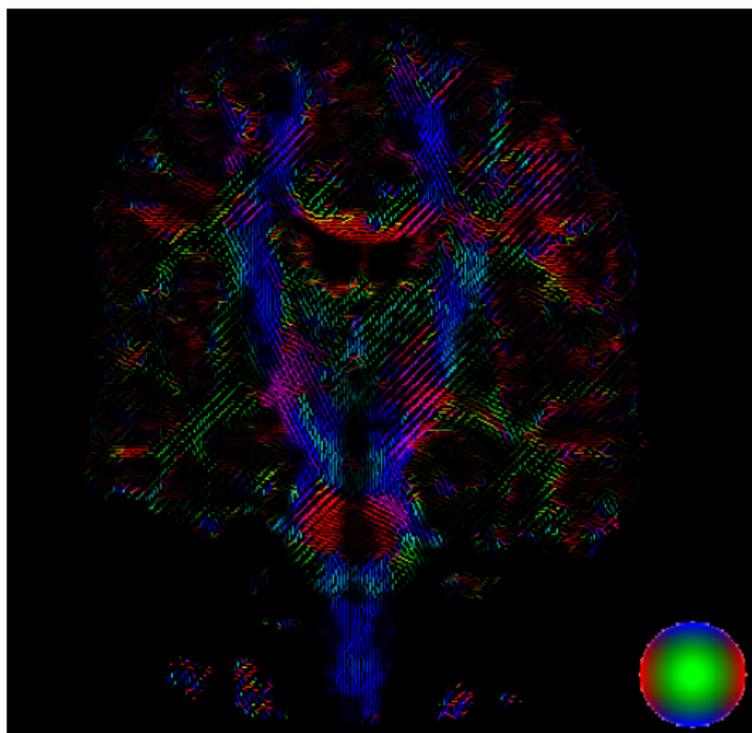
Toutefois, dans bien des cas, nous sommes non seulement intéressés par la quantité d'anisotropie dans un ensemble de données, mais aussi par **les directions prises par cette anisotropie**.

Visualisation vectorielle basée sur l'ACP

Commençons par un exemple simple, soit celui où on ne s'intéresse qu'à **la direction de variation maximale** de la quantité représentée par les tenseurs.

Pour cela, il suffit de visualiser **le vecteur propre associé à la plus grande valeur propre** en utilisant une quelconque technique de visualisation de champ vectoriel. À titre d'exemple, la figure suivante présente un diagramme en hérisson du vecteur propre principal sur une tranche coronale du cerveau humain.

Visualisation vectorielle basée sur l'ACP



[6]

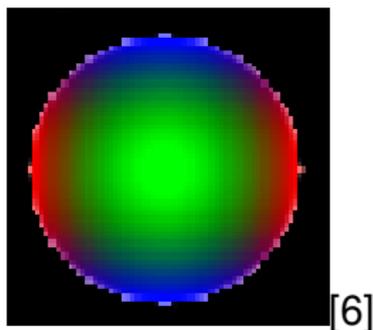
Visualisation vectorielle basée sur l'ACP

En plus de la représentation en hérisson, un **code de couleurs** a été utilisé pour mettre encore plus **l'emphase sur la direction** prise par les vecteurs propres principaux :

$$R = |\mathbf{e}_1 \cdot \mathbf{x}| \quad G = |\mathbf{e}_1 \cdot \mathbf{y}| \quad B = |\mathbf{e}_1 \cdot \mathbf{z}|$$

De cette façon, les vecteurs horizontaux apparaissent en **rouge**, les verticaux sont en **vert** et ceux alignés avec l'axe des z sont en **bleu**.

Visualisation vectorielle basée sur l'ACP



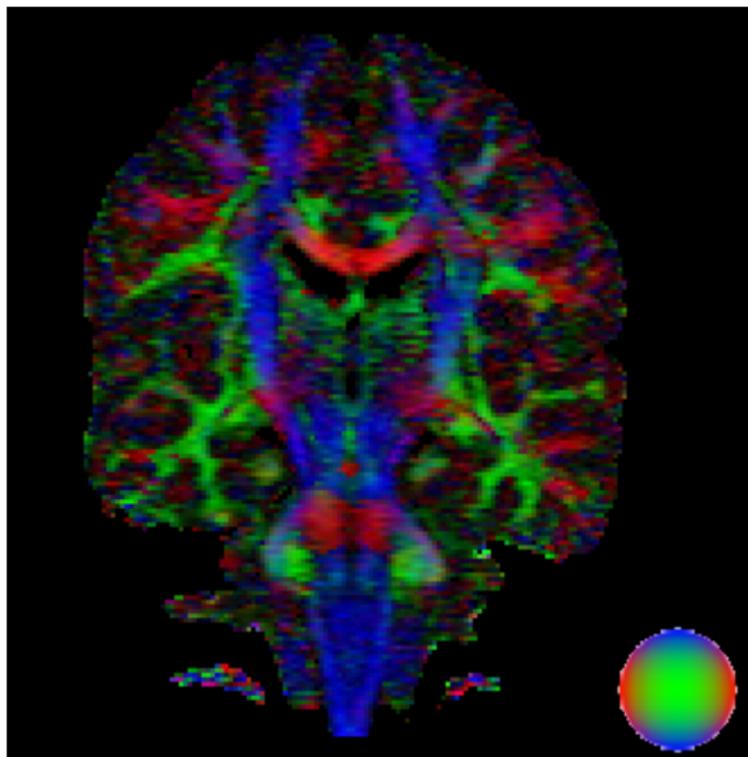
De plus, on utilise la luminance pour illustrer **le niveau de confiance** : plus les couleurs sont vives, plus la direction principale se démarque des autres.

Visualisation vectorielle basée sur l'ACP

Les diagrammes en hérisson ne sont pas la seule technique de visualisation de champs vectoriels qui soit utilisée dans un contexte de données tensorielles. Une technique populaire en imagerie médicale consiste à appliquer un simple **code de couleurs** pour **encoder la direction** du vecteur propre principal.

Pour ce faire, on utilise **le même code de couleur que précédemment**, mais cette fois-ci directement sur l'ensemble de données. La figure suivante illustre un résultat obtenu avec cette technique.

Visualisation vectorielle basée sur l'ACP



[6]

Visualisation vectorielle basée sur l'ACP

Comme mentionné dans le chapitre 2, l'avantage principal d'une telle technique est de **fournir une représentation dense** pour la visualisation d'un champ vectoriel.

Les représentations denses ont toutefois ici un intérêt limité en raison de **la faible résolution des volumes de données tensorielles** ($128 \times 128 \times 56$, par exemple). Ainsi, même avec un diagramme en hérisson, on peut afficher l'ensemble des échantillons sans qu'il y ait de réel problème de recouvrement.

Visualisation vectorielle basée sur l'ACP

Bien qu'il soit intéressant de visualiser une seule direction principale à la fois, ne serait-il pas plus pertinent de pouvoir **afficher l'ensemble de celles-ci** dans une même représentation graphique ?

On débute donc notre étude des techniques permettant d'obtenir un tel résultat avec l'exploration des **représentations par glyphes** pour les données tensorielles.

Représentation par glyphes

- 1 Visualisation de vecteurs
- 2 Visualisation de tenseurs**
 - Introduction aux tenseurs
 - Tenseurs et ACP
 - Représentation par glyphes**
 - Lignes de courant tensorielles
- 3 Références

Représentation par glyphes

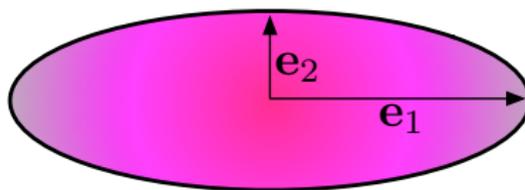
La méthode de visualisation de données tensorielles présentée ici se veut une **généralisation du concept de glyphes** introduit au chapitre 2 pour visualiser des champs vectoriels.

Le principe est le même : **on échantillonne** l'ensemble de données, et pour chaque point conservé, **on construit un glyphe** qui représente les valeurs propres et vecteurs propres du tenseur en ce point.

Représentation par glyphes

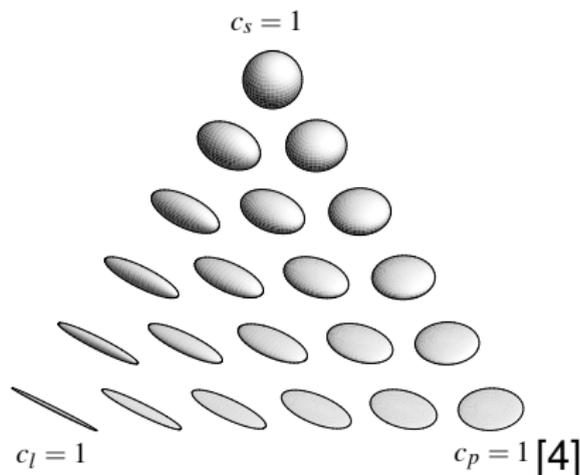
Pour un tenseur 2×2 , cela signifie de considérer **deux valeurs propres** et **deux vecteurs propres** par échantillon.

Pour y arriver, on construit une ellipse 2D dont les demi-axes sont **orientés dans la direction des vecteurs propres** avec une mise à l'échelle selon les valeurs absolues des valeurs propres.



Représentation par glyphes

Dans le cas d'un tenseur 3×3 , on procède de la même façon et on construit un ellipsoïde 3D, ce qui permet ainsi d'encoder **trois valeurs propres** et **trois vecteurs propres**.



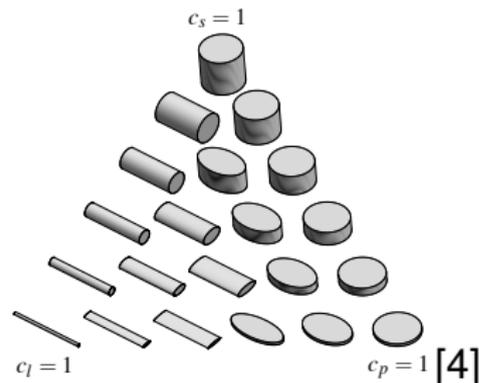
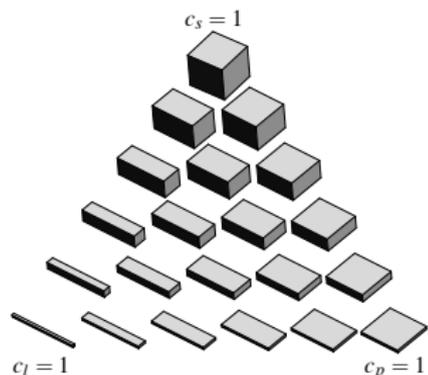
Représentation par glyphes

Dans la figure précédente, chacun des sommets du triangle représente une situation où une des mesures de probabilité c_I , c_p et c_S définies à la section précédente est **maximale, tandis que les deux autres sont nulles**.

Ces situations correspondent à des glyphes en forme de **ligne**, de **disque** et de **sphère**, respectivement.

Représentation par glyphes

En plus des ellipsoïdes, plusieurs autres formes géométriques peuvent être utilisées pour illustrer l'information contenue dans les tenseurs. À titre d'exemple, on peut utiliser les **parallélépipèdes** et les **cylindres**.

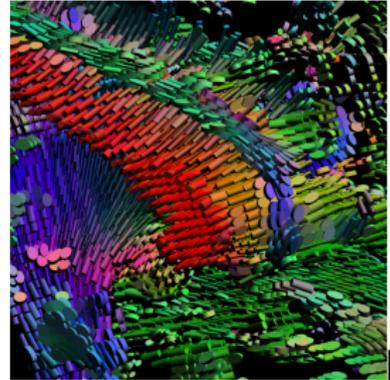
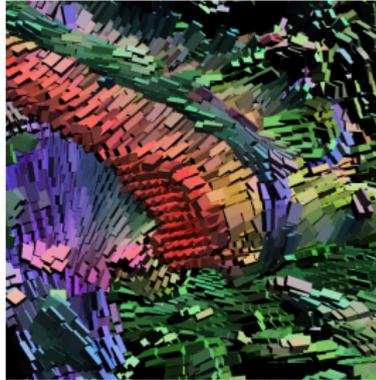
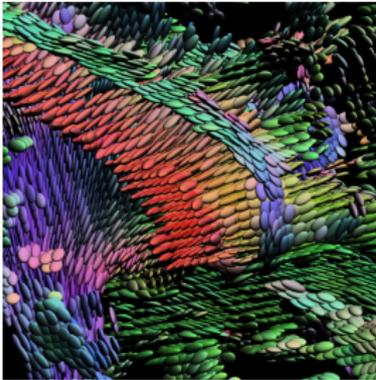


Représentation par glyphes

La figure suivante illustre l'utilisation de ces formes pour visualiser un champ de tenseurs obtenu par DT-MRI. La figure présente **un zoom sur la structure du corps calleux** qui présente une grande anisotropie.

Le code de couleur utilisé est sensiblement le même que celui de la visualisation vectorielle. C'est donc dire que les glyphes sont **colorés selon la direction du vecteur propre principal**.

Représentation par glyphes



[6]

Représentation par glyphes

Autant les ellipsoïdes que les parallélépipèdes et les cylindres ont leurs **avantages** et leurs **inconconvénients** quand vient le temps de produire une représentation visuelle.

Par exemple, les **parallélépipèdes** indiquent de façon claire **la direction des vecteurs propres** grâce à leurs surfaces planes. Ils n'expriment toutefois pas bien l'ambiguïté associée à des vecteurs propres possédant **des valeurs propres égales**.

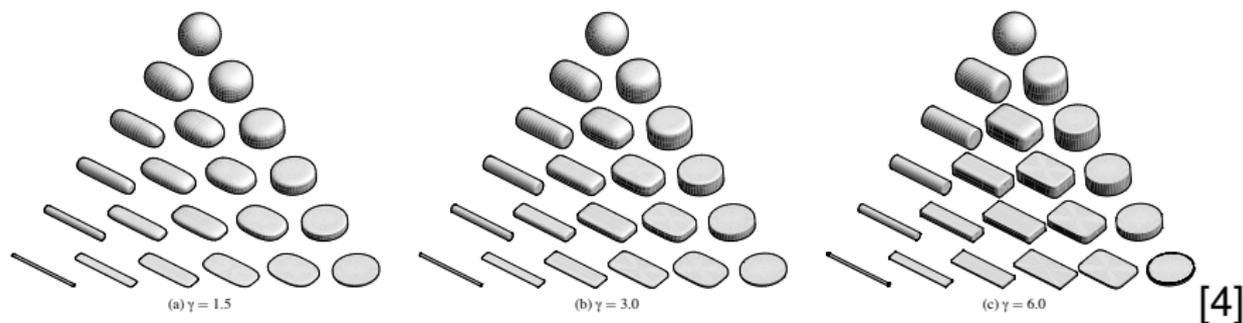
Représentation par glyphes

Les **cylindres** illustrent bien la direction du **vecteur propre principal** le long de leur axe, mais ont la fâcheuse propriété de spontanément **changer d'orientation** en raison de petites variations dans les valeurs propres qui changeraient temporairement la direction principale.

Finalement, les **ellipsoïdes** n'ont aucun de ces problèmes, mais c'est leur **projection 2D** qui peut être ambiguë et donner de fausses impressions quant à l'orientation réelle des tenseurs.

Représentation par glyphes

Pour régler ces problèmes, on a défini les glyphes en forme de **superquadriques**.



Ces formes géométriques ont comme paramètres **les mesures de probabilité** linéaire et planaire c_l et c_s , respectivement.

Représentation par glyphes

La **forme implicite** de ces surfaces est donnée par l'équation

$$\begin{cases} \left(y^{\frac{2}{\alpha}} + z^{\frac{2}{\alpha}} \right)^{\frac{\alpha}{\beta}} + x^{\frac{2}{\beta}} - 1 = 0 & \text{si } c_l \geq c_p \\ \left(x^{\frac{2}{\alpha}} + y^{\frac{2}{\alpha}} \right)^{\frac{\alpha}{\beta}} + z^{\frac{2}{\beta}} - 1 = 0 & \text{si } c_l < c_p \end{cases}$$

avec

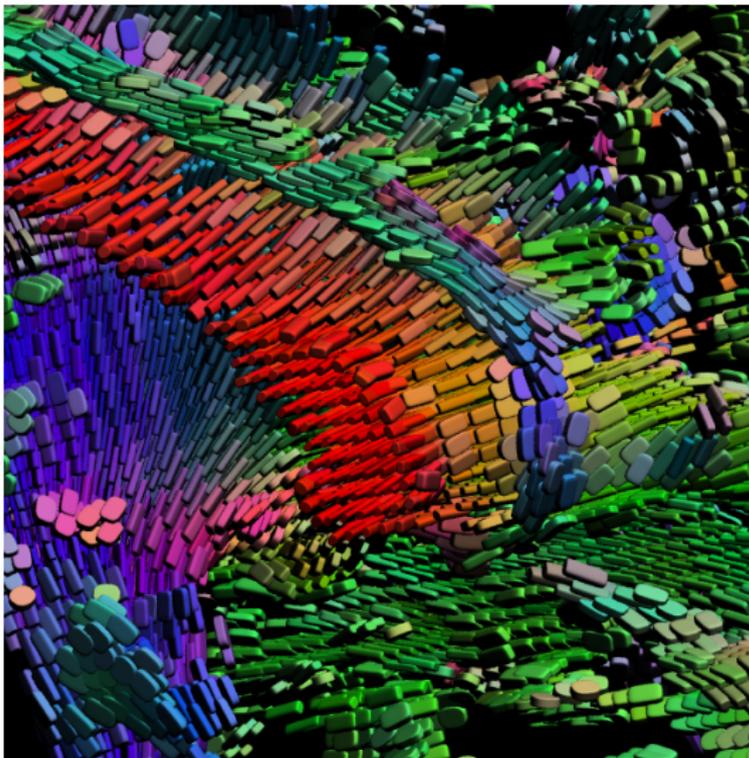
$$\alpha = (1 - c_p)^\gamma \quad \text{et} \quad \beta = (1 - c_l)^\gamma$$

dans le premier cas, et

$$\alpha = (1 - c_l)^\gamma \quad \text{et} \quad \beta = (1 - c_p)^\gamma$$

dans le deuxième.

Représentation par glyphes



Représentation par glyphes

En bout de ligne, l'utilisation des glyphes est potentiellement **la manière la plus simple** d'obtenir une représentation visuelle pour un ensemble de données tensorielles.

La visualisation par glyphes pour les tenseurs souffre cependant des **mêmes problèmes** que lorsqu'on l'utilisait pour représenter des données vectorielles, soit de fournir une **représentation discontinue** présentant du **recouvrement**.

Représentation par glyphes

La prochaine technique abordée propose de **mettre l'emphase sur certaines structures** dans l'ensemble de données tensorielles, comme les fibres neuronales, ce que ne permettent pas de faire les glyphes.

Cette méthode sera grandement inspirée des **lignes de courant** dans un champ vectoriel et permettra de faire le suivi des fibres dans le cerveau.

Lignes de courant tensorielles

1 Visualisation de vecteurs

2 **Visualisation de tenseurs**

- Introduction aux tenseurs
- Tenseurs et ACP
- Représentation par glyphes
- **Lignes de courant tensorielles**

3 Références

Tracking de fibres

Pour illustrer cette méthode, on utilisera de nouveau un ensemble de **données tensorielles obtenues par DT-MRI**.

On se souvient que les régions présentant une **anisotropie élevée** (fort coefficient c_1) correspondent au **fibres neuronales** et que celles-ci sont alignées avec le vecteur propre principal \mathbf{e}_1 .

Tracking de fibres

Une manière naturelle de visualiser **la position et l'orientation de ces fibres** est de faire le suivi des directions prises par les vecteurs propres principaux dans les régions de forte anisotropie.

Pour arriver à ce résultat, **on fixe d'abord une région source** parmi les régions où l'anisotropie est élevée. Pour trouver une telle région, on peut se baser sur les mesures servant à quantifier l'anisotropie vues précédemment.

Tracking de fibres

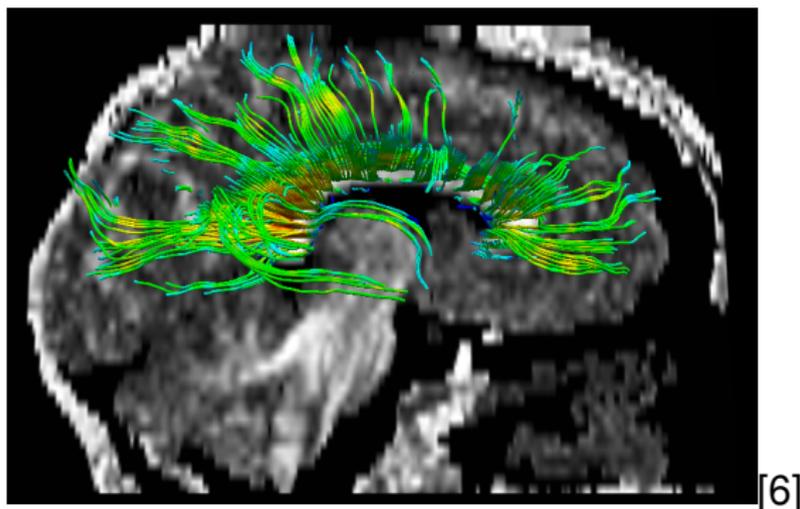
Par la suite, **on trace des lignes de courant** à partir de la région source pour des valeurs positives et négatives de Δt . Les directions prises par ces lignes sont données par le vecteur propre principal \mathbf{e}_1 .

On fait le suivi de la fibre jusqu'à ce qu'un **critère d'arrêt** soit rencontré :

- longueur maximale de la fibre ;
- sortie d'une région d'intérêt ;
- valeur maximale de Δt ;
- seuil d'anisotropie ;
- proximité des autres fibres.

Tracking de fibres

Une fois que les fibres ont été obtenues, on peut les afficher avec des **segments de droite** ou avec des **tubes de courant**.

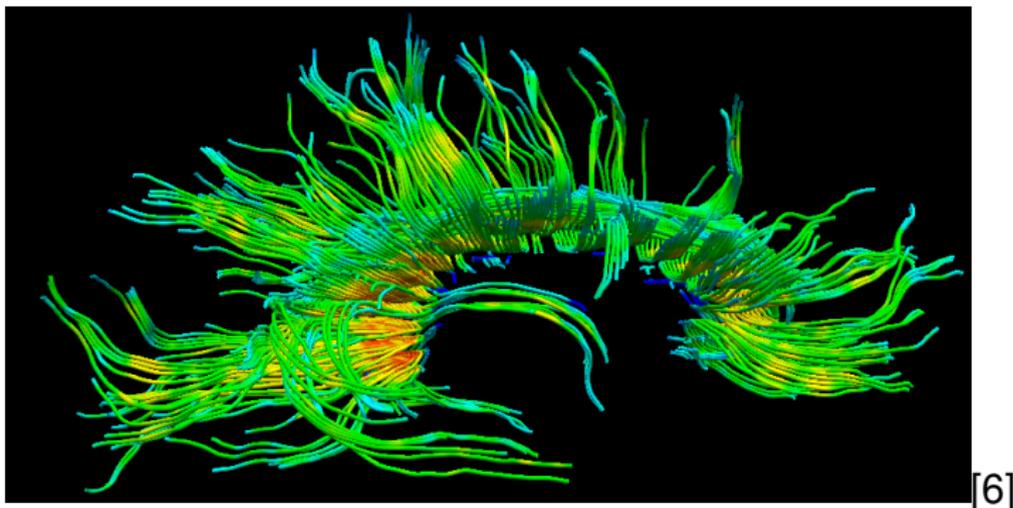


Tracking de fibres

Dans l'image précédente, un **code de couleurs basé sur l'anisotropie fractionnelle** a été utilisé. La couleur **bleu** indique une **faible anisotropie**, tandis que le **rouge** est associé à une **anisotropie très forte**.

Le traçage s'arrête lorsque la mesure de probabilité linéaire c_l **devient inférieure à 0.15**. L'incertitude quant à la direction à suivre devient alors trop grande.

Tracking de fibres



Conclusion

730 diapos plus tard...

Conclusion

Fin.

Conclusion

Merci pour



votre attention

Références

- 1 Visualisation de vecteurs
- 2 Visualisation de tenseurs
- 3 Références**

Références



B. Cabral and L. Leedom.
Imaging vector fields using line integral convolution, 1993.



M. Chamberland, D. Fortin, and M. Descoteaux.
Real-time fiber tractography : Interactive parameter tuning for neurosurgical interventions, 2012.



C. D. Hansen and C. R. Johnson.
The visualization handbook, 2004.



G. Kindlmann.
Superquadric tensor glyphs, 2004.



W. Schroeder, K. Martin, and B. Lorensen.
The visualization toolkit : An object-oriented approach to 3d graphics, 2006.



A. C. Telea.
Data visualization : Principles and practice, 2008.