Géométrie et espaces de formes - 2020/2021 - TP 2

1 De l'appariement linéaire aux difféomorphismes

Nous allons voir tout d'abord comment la composition de petites déformations permet de résoudre le problème de non-inversibilité de l'approche linéaire. Pour cela on reprend la situation de la section 2 du premier TP, mais au lieu de réaliser l'appariement des y_i vers les z_i en une seule étape, on introduit des points intermédiaires $y_i = q_i^1, q_i^2, \cdots, q_i^{n_t} = z_i$ régulièrement espacés le long du segment $[y_i, z_i]$ et on calcule l'appariement linéaire des q_i^t vers les q_i^{t+1}, successivement pour $1 \leq t \leq n_t - 1$. On compose ensuite simplement les déformations $\phi^t = \mathrm{id} + v^t$ obtenues : $\phi := \phi^{n_t-1} \circ \cdots \circ \phi^1$.

1. Tester l'approche proposée sur un exemple où l'appariement linéaire aboutit à une déformation ϕ non-inversible (grille de déformation "repliée"). Utiliser la fonction `LinTraj` pour construire les points intermédiaires q_i^t puis écrire une fonction `MatchingSteps(Q,h)` sur le modèle de `MatchingLinear(y,z,h)` qui calcule les $\phi(x_j)$ pour des points quelconques x_j donnés en entrée. Afficher les positions des points et les grilles de déformations obtenues. On testera plusieurs valeurs pour n_t : $n_t = 3, 5, 10$ par exemple.

2. Les positions des points intermédiaires q_i^t, $2 \leq t \leq n_t - 1$ sont en fait libres et on peut chercher à optimiser ces positions en minimisant l'énergie suivante :

$$J(q^2, \ldots, q^{n_t-1}) := \sum_{t=1}^{n_t-1} \left\langle q^{t+1} - q^t \,,\, K(q^t)^{-1}(q^{t+1} - q^t) \right\rangle .$$

   Pour éviter le calcul fastidieux du gradient de cette fonctionnelle, nous allons utiliser le module d'autodifférentiation de PyTorch. Commencer par regarder le tutoriel sur la différentiation automatique, puis écrire une fonction `Energy(Q,h)` qui renvoie la valeur de la fonctionnelle. L'entrée Q est supposée être de type torch.tensor, et la fonction h ainsi que la fonction KernelMatrix doivent être réécrites avec la syntaxe de PyTorch, de telle manière que la fonction puisse être différentiée.

3. Pour $n_t = 5$, écrire une descente de gradient à pas fixe sur la fonctionnelle précédente. Tester d'abord pour des configurations de 5 points tirés aléatoirement (faire 2000 itérations, pas de descente $\eta = 0.01$), puis pour l'exemple des poissons (prendre $\eta = 0.00001$).

4. Expliquer pourquoi le modèle précédent constitue en fait une première implémentation de l'approche difféomorphique qui définit les déformations ϕ par intégration d'une famille de champs de vecteurs $(v(t, \cdot))_{t \in [0,1]}$ dans V :

$$\begin{cases} \phi := \phi(1, \cdot), \\ \frac{\partial \phi}{\partial t}(t, x) = v(t, \phi(t, x)), \end{cases}$$

   et considère le problème d'appariement suivant :

$$(P) \quad \text{Minimiser} \quad J(v) = \int_0^1 \|v(t, \cdot)\|_V^2 dt \quad \text{sous les contraintes} \quad \phi^v(1, y_i) = z_i, \;\; 1 \leq i \leq n.$$

2 Approche hamiltonienne et équations géodésiques

5. Expliquer pourquoi le problème (P) peut s'interpréter comme un problème de recherche de trajectoire géodésique entre y et z dans l'espace des configurations de n points de $\mathbb{R}^d$ muni d'une certaine métrique locale $\| \cdot \|_q$.

Les équations géodésiques peuvent s'écrire via une formulation hamiltonienne. Pour toute trajectoire $q(t)$, on pose (en omettant la dépendance en t pour simplifier les écritures) :

$$L(q, \dot{q}) = \frac{1}{2}\|\dot{q}\|_q^2 \quad \text{(Lagrangien)},$$
$$p = \nabla_{\dot{q}} L(q, \dot{q}) \quad \text{(gradient de } L \text{ par rapport à } \dot{q} \text{ pour la métrique euclidienne)},$$
$$H(p, q) = \langle p, \dot{q} \rangle_{\mathbb{R}^{2n}} - L(q, \dot{q}) \quad \text{(Hamiltonien)}.$$

On montre alors que les trajectoires géodésiques vérifient

$$\begin{cases} \dot{p} = -\nabla_q H(p, q) \\ \dot{q} = \nabla_p H(p, q). \end{cases} \tag{1}$$

6. Identifier successivement $L(q, \dot{q})$, p, $H(p, q)$, puis écrire les équations des trajectoires géodésiques. Montrer que $H(p, q)$ est constant pour une trajectoire géodésique.
7. Écrire une fonction `Hamilt(p,q,h)` pour calculer $H(p, q)$, puis une fonction `HamiltSys(p,q,h)` pour le calcul de $(\dot{p}, \dot{q})$ d'après les équations (1), et en utilisant la différentiation automatique. Écrire ensuite une fonction `Shooting(p0,q0,h)` qui, à partir d'une configuration $p(0), q(0)$ initiale, résout numériquement l'équation (1) par un schéma d'Euler simple, puis renvoie $p(1)$, $q(1)$ ainsi que Q contenant l'ensemble des trajectoires. Tester cette fonction pour les configurations précédentes en choisissant des vecteurs moments $p(0)$ aléatoires. Afficher les trajectoires obtenues et la déformation d'une grille régulière.
8. Enfin écrire une fonction `loss(p0,q0,z,h)` renvoyant la somme des carrés des distances des $q_i(1)$ aux z_i, et réaliser une descente de gradient simple sur cette fonction afin d'optimiser sur les moments initiaux (faire 100 itérations avec pas de descente 0.01). Afficher les trajectoires $q_i(t)$ et l'appariement difféomorphique obtenu, et comparer avec les résultats de la question 3.