EPDiff and Optimal Control of Shapes

Colin Cotter, Department of Aeronautics,
Imperial College London

Talk Outline

Talk Outline

- How does EPDiff arise in the analysis of shape? A Darryl-eyed view

Talk Outline

- How does EPDiff arise in the analysis of shape? A Darryl-eyed view
- Particle-mesh discretisation

Talk Outline

- How does EPDiff arise in the analysis of shape? A Darryl-eyed view
- Particle-mesh discretisation
- Numerical examples

EP equation and optimal control

Example: control of rigid body

Example: control of rigid body

$$\dot{Q} = Q\Omega \quad \text{with} \quad \Omega^T = -\Omega \in \mathfrak{so}(3).$$

Example: control of rigid body

$$\dot{Q} = Q\Omega \quad \text{with} \quad \Omega^T = -\Omega \in \mathfrak{so}(3).$$

Minimise $\displaystyle\int_0^1 \frac{1}{2}\Omega \cdot I\Omega \,\mathrm{d}\, t$

Example: control of rigid body

$$\dot{Q} = Q\Omega \quad \text{with} \quad \Omega^T = -\Omega \in \mathfrak{so}(3).$$

Minimise $\displaystyle\int_0^1 \frac{1}{2}\Omega \cdot I\Omega \,\mathrm{d}\,t$

$$Q(0) = Q_0, \quad Q(1) = Q_1$$

Example: control of rigid body

$$\dot{Q} = Q\Omega \quad \text{with} \quad \Omega^T = -\Omega \in \mathfrak{so}(3).$$

Minimise $\displaystyle\int_0^1 \frac{1}{2}\Omega \cdot I\Omega \,\mathrm{d}\,t$

$$Q(0) = Q_0, \quad Q(1) = Q_1$$

see Bloch, Crouch, Marsden & Ratiu, 1998

$$\delta \int_0^1 \frac{1}{2}\Omega \cdot I\Omega + P^T \cdot (\dot{Q} - Q\Omega) \, \mathrm{d}\, t = 0$$

$$\delta \int_0^1 \frac{1}{2}\Omega \cdot I\Omega + P^T \cdot (\dot{Q} - Q\Omega)\,\mathrm{d}\,t = 0$$

$$\begin{aligned} I\Omega &= Q^T P, \\ \dot{Q} &= Q\Omega, \\ \dot{P} &= P\Omega. \end{aligned}$$

$$\delta \int_0^1 \frac{1}{2}\Omega \cdot I\Omega + P^T \cdot (\dot{Q} - Q\Omega)\,\mathrm{d}\,t = 0$$

$$\begin{aligned} I\Omega &= Q^T P, \\ \dot{Q} &= Q\Omega, \\ \dot{P} &= P\Omega. \end{aligned}$$

$$\begin{aligned} \tfrac{d}{dt} I\Omega &= \dot{Q}^T P + Q^T \dot{P}, \\ &= (Q\Omega)^T P + Q^T P\Omega, \\ &= \Omega^T (Q^T P) + (Q^T P)\Omega, \\ &= \Omega^T (I\Omega) + (I\Omega)\Omega. \end{aligned}$$

When in general can P and Q be eliminated from the dynamical equations?

$$Q \in M \quad \xi \in V \quad \dot{Q} = L_{\xi} Q$$

$$Q \in M \quad \xi \in V \quad \dot{Q} = L_\xi Q$$

$$\delta \int_0^1 \ell(\xi) + \langle P, \dot{Q} - L_\xi Q \rangle \, \mathrm{d}\, t = 0$$

$$Q \in M \quad \xi \in V \quad \dot{Q} = L_\xi Q$$

$$\delta \int_0^1 \ell(\xi) + \langle P, \dot{Q} - L_\xi Q \rangle \, \mathrm{d}\, t = 0$$

$$\begin{aligned} \frac{\delta \ell}{\delta \xi} &= -P \diamond Q, \\ Q &= \mathcal{L}_\xi Q, \\ \dot{P} &= -\left(T_Q \mathcal{L}_\xi\right)^T P, \end{aligned}$$

$$Q \in M \quad \xi \in V \quad \dot{Q} = L_\xi Q$$

$$\delta \int_0^1 \ell(\xi) + \langle P, \dot{Q} - L_\xi Q \rangle \, \mathrm{d}\, t = 0$$

$$\begin{aligned} \frac{\delta \ell}{\delta \xi} &= -P \diamond Q, \\ Q &= \mathcal{L}_\xi Q, \\ \dot{P} &= -\left(T_Q \mathcal{L}_\xi\right)^T P, \end{aligned}$$

$$\langle \boldsymbol{P} \diamond \boldsymbol{Q}, \boldsymbol{\xi} \rangle_V = -\langle \boldsymbol{P}, \mathcal{L}_{\boldsymbol{\xi}} \boldsymbol{Q} \rangle_{T^* M}$$

$$Q \in M \quad \xi \in V \quad \dot{Q} = L_\xi Q$$

$$\delta \int_0^1 \ell(\xi) + \langle P, \dot{Q} - L_\xi Q \rangle \, \mathrm{d}\, t = 0$$

$$\begin{aligned} \frac{\delta \ell}{\delta \xi} &= -P \diamond Q, \\ Q &= \mathcal{L}_\xi Q, \\ \dot{P} &= -\left(T_Q \mathcal{L}_\xi\right)^T P, \end{aligned}$$

$$\langle \boldsymbol{P} \diamond \boldsymbol{Q}, \boldsymbol{\xi} \rangle_V = - \langle \boldsymbol{P}, \mathcal{L}_{\boldsymbol{\xi}} \boldsymbol{Q} \rangle_{T^* M}$$

Hamiltonian $H = \ell(\xi(P, Q))$

Thm: The cotangent variables (P, Q) may be eliminated to give an equation for ξ only if and only if the velocity map L_ξ is a Lie algebra action

Thm: The cotangent variables (P, Q) may be eliminated to give an equation for ξ only if and only if the velocity map L_ξ is a Lie algebra action

In that case, the evolution equation obtained is the Euler-Poincaré equation

Thm: The cotangent variables (P, Q) may be eliminated to give an equation for ξ only if and only if the velocity map L_ξ is a Lie algebra action

In that case, the evolution equation obtained is the Euler-Poincaré equation

$$\frac{d}{dt}\frac{\delta l}{\delta \xi} + \mathrm{ad}^*_\xi \frac{\delta l}{\delta \xi} = 0$$

Thm: The cotangent variables (P, Q) may be eliminated to give an equation for ξ only if and only if the velocity map L_ξ is a Lie algebra action

In that case, the evolution equation obtained is the Euler-Poincaré equation

$$\frac{d}{dt}\frac{\delta l}{\delta \xi} + \mathrm{ad}^*_\xi \frac{\delta l}{\delta \xi} = 0$$

and $P \diamond Q$ is a cotangent-lifted momentum map

Variational integrators

Variational integrators

- This allows a method for designing variational integrators for the EP equation

Variational integrators

- This allows a method for designing variational integrators for the EP equation
- Simply discretise the variational principle and derive the equations

Variational integrators

- This allows a method for designing variational integrators for the EP equation
- Simply discretise the variational principle and derive the equations
- The resulting numerical method can be reduced to a discrete EP equation provided that the discretisation is left-invariant

Variational integrators

- This allows a method for designing variational integrators for the EP equation
- Simply discretise the variational principle and derive the equations
- The resulting numerical method can be reduced to a discrete EP equation provided that the discretisation is left-invariant
- See: Marsden and Bou-Rabee (2007) and Cotter and Holm (submitted 2007)

Variational image matching

When the control variables are vector fields on a manifold, they can act on a number of different spaces.

When the control variables are vector fields on a manifold, they can act on a number of different spaces.

For a velocity field $\boldsymbol{u}_t(\boldsymbol{x})$ defined on the domain Ω with appropriate boundary conditions:

When the control variables are vector fields on a manifold, they can act on a number of different spaces.

For a velocity field $\boldsymbol{u}_t(\boldsymbol{x})$ defined on the domain Ω with appropriate boundary conditions:

- Embeddings $\boldsymbol{Q} : S \mapsto \Omega, \qquad \frac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \boldsymbol{u}_t(\boldsymbol{Q}_t(s))$

When the control variables are vector fields on a manifold, they can act on a number of different spaces.

For a velocity field $\boldsymbol{u}_t(\boldsymbol{x})$ defined on the domain Ω with appropriate boundary conditions:

- Embeddings $\boldsymbol{Q} : S \mapsto \Omega, \qquad \frac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \boldsymbol{u}_t(\boldsymbol{Q}_t(s))$
- Functions $f : \Omega \mapsto \mathbb{R}, \qquad \frac{\partial}{\partial t} f_t(\boldsymbol{x}) + \boldsymbol{u}_t(\boldsymbol{x}) \cdot \nabla \boldsymbol{f}_t(\boldsymbol{x}) = 0$

When the control variables are vector fields on a manifold, they can act on a number of different spaces.

For a velocity field $\boldsymbol{u}_t(\boldsymbol{x})$ defined on the domain Ω with appropriate boundary conditions:

- Embeddings $\boldsymbol{Q} : S \mapsto \Omega,$ $\quad \dfrac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \boldsymbol{u}_t(\boldsymbol{Q}_t(s))$
- Functions $f : \Omega \mapsto \mathbb{R},$ $\quad \dfrac{\partial}{\partial t} f_t(\boldsymbol{x}) + \boldsymbol{u}_t(\boldsymbol{x}) \cdot \nabla \boldsymbol{f}_t(\boldsymbol{x}) = 0$
- Vector fields $\boldsymbol{\xi} : \Omega \mapsto T\Omega,$ $\quad \dfrac{\partial}{\partial t}\boldsymbol{\xi}_t + (\boldsymbol{u}_t \cdot \nabla)\boldsymbol{\xi}_t = (\boldsymbol{\xi}_t \cdot \nabla)\boldsymbol{u}_t$

When the control variables are vector fields on a manifold, they can act on a number of different spaces.

For a velocity field $\boldsymbol{u}_t(\boldsymbol{x})$ defined on the domain Ω with appropriate boundary conditions:

- Embeddings $\boldsymbol{Q} : S \mapsto \Omega, \qquad \frac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \boldsymbol{u}_t(\boldsymbol{Q}_t(s))$
- Functions $f : \Omega \mapsto \mathbb{R}, \qquad \frac{\partial}{\partial t} f_t(\boldsymbol{x}) + \boldsymbol{u}_t(\boldsymbol{x}) \cdot \nabla \boldsymbol{f}_t(\boldsymbol{x}) = 0$
- Vector fields $\boldsymbol{\xi} : \Omega \mapsto T\Omega, \qquad \frac{\partial}{\partial t}\boldsymbol{\xi}_t + (\boldsymbol{u}_t \cdot \nabla)\boldsymbol{\xi}_t = (\boldsymbol{\xi}_t \cdot \nabla)\boldsymbol{u}_t$
- ...

When the control variables are vector fields on a manifold, they can act on a number of different spaces.

For a velocity field $\boldsymbol{u}_t(\boldsymbol{x})$ defined on the domain Ω with appropriate boundary conditions:

- Embeddings $\boldsymbol{Q} : S \mapsto \Omega,\qquad \frac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \boldsymbol{u}_t(\boldsymbol{Q}_t(s))$
- Functions $f : \Omega \mapsto \mathbb{R},\qquad \frac{\partial}{\partial t} f_t(\boldsymbol{x}) + \boldsymbol{u}_t(\boldsymbol{x}) \cdot \nabla \boldsymbol{f}_t(\boldsymbol{x}) = 0$
- Vector fields $\boldsymbol{\xi} : \Omega \mapsto T\Omega,\qquad \frac{\partial}{\partial t}\boldsymbol{\xi}_t + (\boldsymbol{u}_t \cdot \nabla)\boldsymbol{\xi}_t = (\boldsymbol{\xi}_t \cdot \nabla)\boldsymbol{u}_t$
- ...

All of these spaces have roles in imaging

When the control variables are vector fields on a manifold, they can act on a number of different spaces.

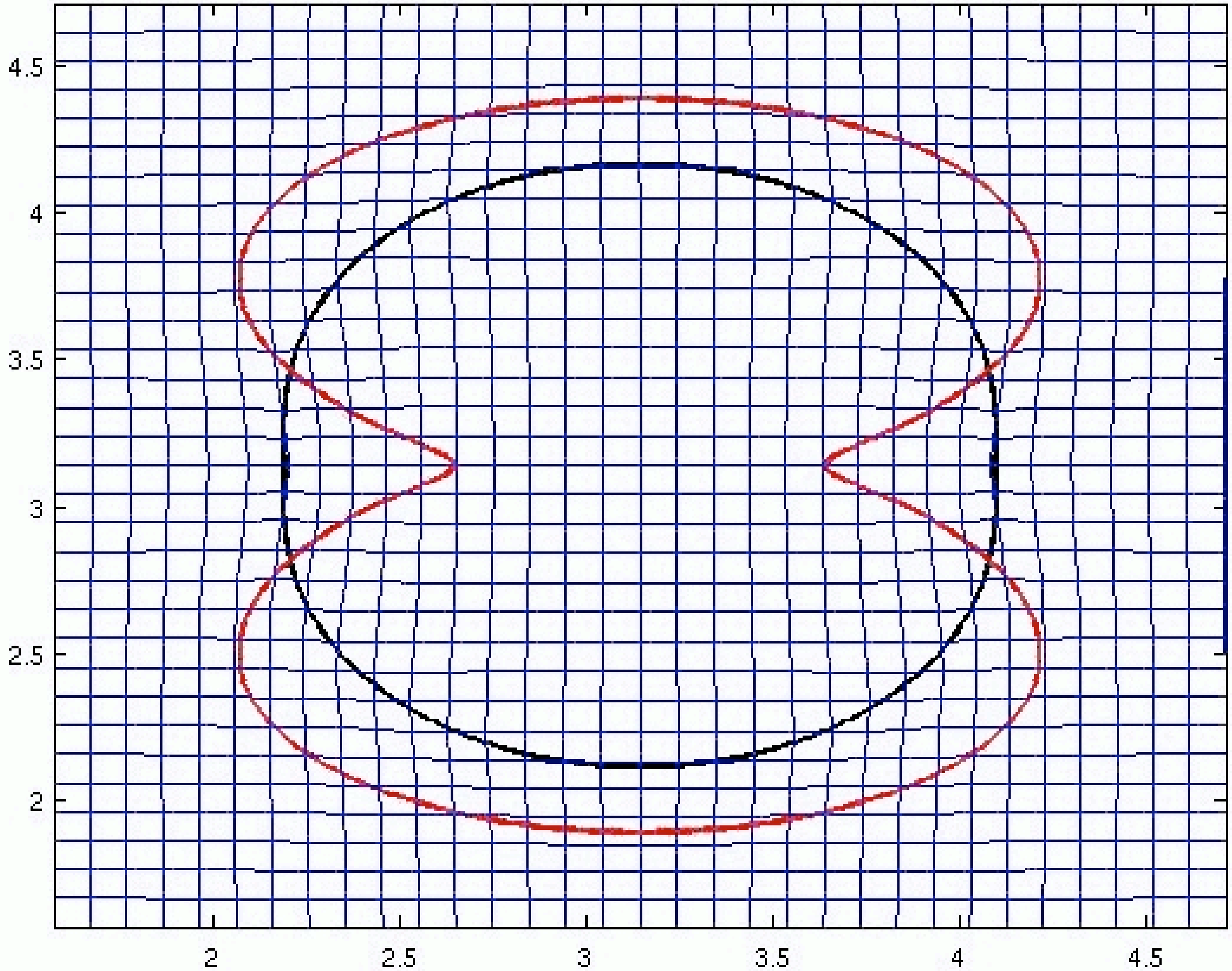
For a velocity field $\boldsymbol{u}_t(\boldsymbol{x})$ defined on the domain Ω with appropriate boundary conditions:

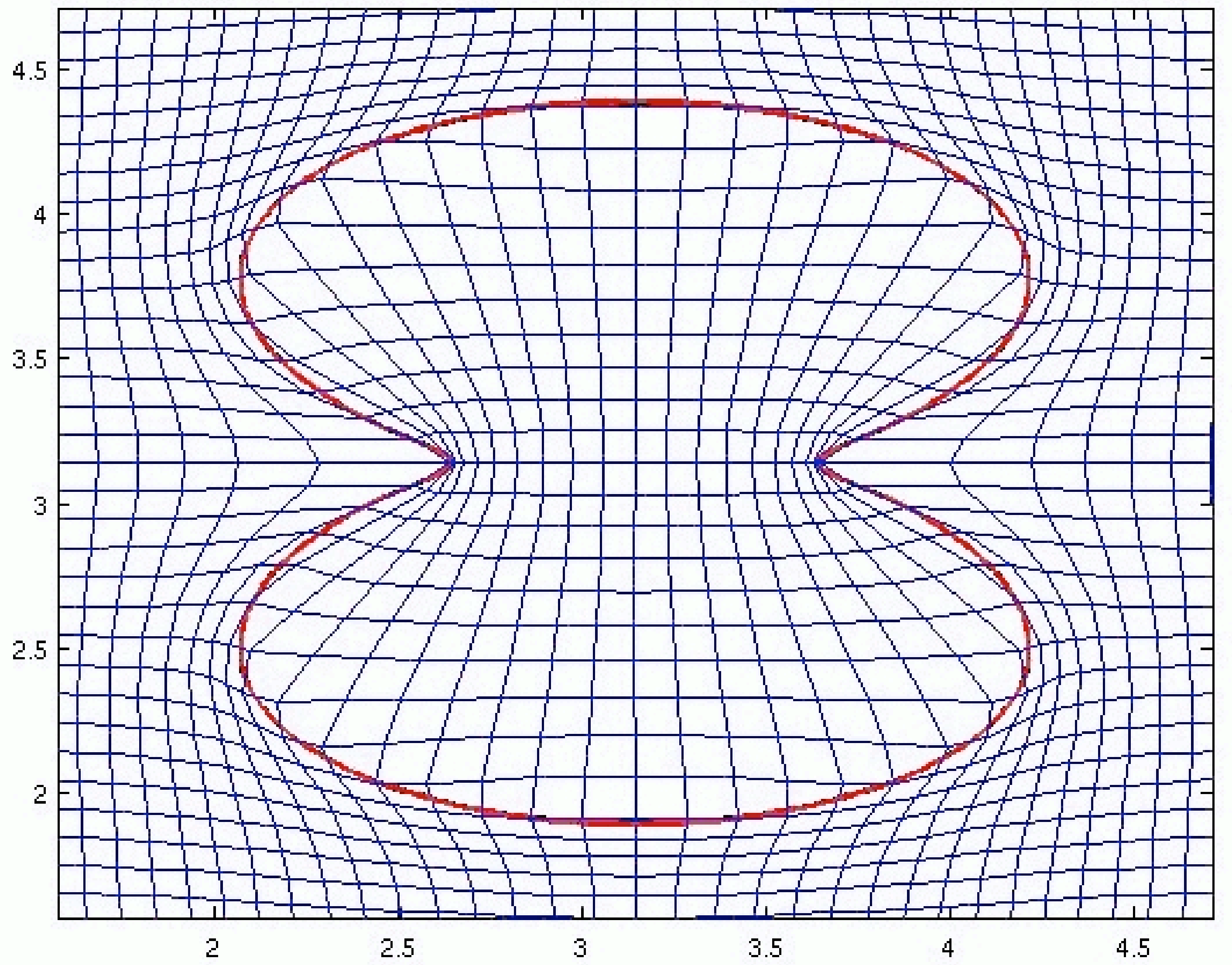
- Embeddings $\boldsymbol{Q} : S \mapsto \Omega, \qquad \frac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \boldsymbol{u}_t(\boldsymbol{Q}_t(s))$
- Functions $f : \Omega \mapsto \mathbb{R}, \qquad \frac{\partial}{\partial t} f_t(\boldsymbol{x}) + \boldsymbol{u}_t(\boldsymbol{x}) \cdot \nabla \boldsymbol{f}_t(\boldsymbol{x}) = 0$
- Vector fields $\boldsymbol{\xi} : \Omega \mapsto T\Omega, \qquad \frac{\partial}{\partial t}\boldsymbol{\xi}_t + (\boldsymbol{u}_t \cdot \nabla)\boldsymbol{\xi}_t = (\boldsymbol{\xi}_t \cdot \nabla)\boldsymbol{u}_t$
- ...

All of these spaces have roles in imaging

For the rest of the talk specialise to embeddings

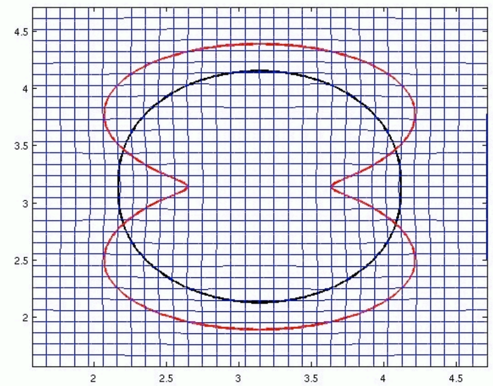
What does the shortest path between two shapes look like?

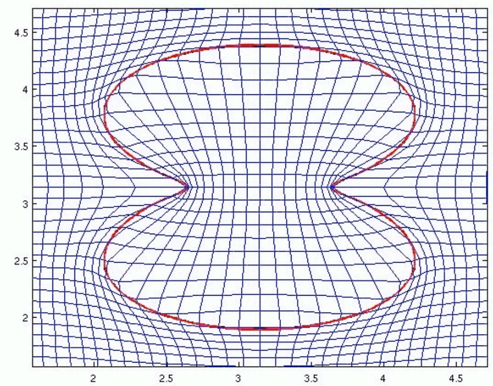

4.5
4
3.5
3
2.5
2
2
2.5
3
3.5
4
4.5


4.5
4
3.5
3
2.5
2
2
2.5
3
3.5
4
4.5

What can we do with these paths?

- Length along path measures amount of **deformation** from one shape to another
- A way of **comparing** and **classifying** shapes
- Deformation from one shape to another is encoded in **initial conditions** for deformation velocity field, where we can perform linear statistics





Optimisation problem

Solve

$$\min\left\{\int_0^1 \|\boldsymbol{u}_t\|^2 \mathrm{d}t\right\}$$

Optimisation problem

Solve

$$\min \left\{ \int_0^1 \|\boldsymbol{u}_t\|^2 \mathrm{d}t \right\}$$

subject to

$$\frac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \boldsymbol{u}_t(\boldsymbol{Q}_t(s)), \quad \forall s \in (0, 2\pi]$$

Optimisation problem

Solve

$$\min \left\{ \int_0^1 \|\boldsymbol{u}_t\|^2 \mathrm{d}t \right\}$$

subject to

$$\frac{\partial}{\partial t} \boldsymbol{Q}_t(s) = \boldsymbol{u}_t(\boldsymbol{Q}_t(s)), \quad \forall s \in (0, 2\pi]$$

and

$$\boldsymbol{Q}_0(s) = \boldsymbol{Q}^A(s), \quad \forall s \in (0, 2\pi], \qquad \boldsymbol{Q}_1 \equiv \boldsymbol{Q}^B$$

Constraints

Constraints

- **Dynamical constraint:** shape is moved by the flow

Constraints

- **Dynamical constraint:** shape is moved by the flow
- **Matching condition:** shapes are “equivalent” after deformation

Constraints

- **Dynamical constraint:** shape is moved by the flow
- **Matching condition:** shapes are “equivalent” after deformation
- For now, matching condition is a “soft constraint” enforced *via* penalty term

Dynamical constraints

Enforce dynamical equation using **Lagrange multipliers**

Dynamical constraints

Enforce dynamical equation using **Lagrange multipliers**

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\int_0^1 \left(\|\boldsymbol{u}_t\|^2 + \int_S \boldsymbol{P}_t(s) \cdot \left(\frac{\partial}{\partial t} \boldsymbol{Q}_t(s) - \boldsymbol{u}_t(\boldsymbol{Q}_t(s)) \right) \mathrm{d}s \right) \mathrm{d}t + \frac{1}{\sigma^2} f[\boldsymbol{Q}_1] \right)$$

Dynamical constraints

Enforce dynamical equation using **Lagrange multipliers**

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\int_0^1 \left(\|\boldsymbol{u}_t\|^2 + \int_S \boldsymbol{P}_t(s) \cdot \left(\frac{\partial}{\partial t}\boldsymbol{Q}_t(s) - \int_{\boldsymbol{\Omega}} \boldsymbol{u}_t(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{Q}_t(s))\mathrm{d}V(\boldsymbol{x}) \right) \mathrm{d}s \right) \mathrm{d}t + \frac{1}{\sigma^2} f[\boldsymbol{Q}_1] \right)$$

Dynamical constraints

Enforce dynamical equation using **Lagrange multipliers**

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\int_0^1 \left(\|\boldsymbol{u}_t\|^2 + \int_S \boldsymbol{P}_t(s) \cdot \left(\frac{\partial}{\partial t} \boldsymbol{Q}_t(s) - \int_{\boldsymbol{\Omega}} \boldsymbol{u}_t(\boldsymbol{x}) \delta(\boldsymbol{x} - \boldsymbol{Q}_t(s)) \mathrm{d}V(\boldsymbol{x}) \right) \mathrm{d}s \right) \mathrm{d}t + \frac{1}{\sigma^2} f[\boldsymbol{Q}_1] \right)$$

Euler-Lagrange equations for t<1

Dynamical constraints

Enforce dynamical equation using **Lagrange multipliers**

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\int_0^1 \left(\|\boldsymbol{u}_t\|^2 + \int_S \boldsymbol{P}_t(s) \cdot \left(\frac{\partial}{\partial t} \boldsymbol{Q}_t(s) - \int_{\boldsymbol{\Omega}} \boldsymbol{u}_t(\boldsymbol{x}) \delta(\boldsymbol{x} - \boldsymbol{Q}_t(s)) \mathrm{d}V(\boldsymbol{x}) \right) \mathrm{d}s \right) \mathrm{d}t + \frac{1}{\sigma^2} f[\boldsymbol{Q}_1] \right)$$

Euler-Lagrange equations for $t<1$

$$\begin{aligned} \frac{\partial}{\partial t} \boldsymbol{Q}_t(s) &= \boldsymbol{u}_t(\boldsymbol{Q}_t(s)) \\ \frac{\partial}{\partial t} \boldsymbol{P}_t(s) &= -(\nabla \boldsymbol{u}(\boldsymbol{Q}_t(s)))^T \boldsymbol{P}_t(s) \\ \boldsymbol{m}_t(\boldsymbol{x}) &= \int_S \boldsymbol{P}_t(s) \delta(\boldsymbol{x} - \boldsymbol{Q}_t(s)) \, \mathrm{d}\, s \end{aligned}$$

Dynamical constraints

Enforce dynamical equation using **Lagrange multipliers**

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\int_0^1 \left(\|\boldsymbol{u}_t\|^2 + \int_S \boldsymbol{P}_t(s) \cdot \left(\frac{\partial}{\partial t} \boldsymbol{Q}_t(s) - \int_{\boldsymbol{\Omega}} \boldsymbol{u}_t(\boldsymbol{x}) \delta(\boldsymbol{x} - \boldsymbol{Q}_t(s)) \mathrm{d}V(\boldsymbol{x}) \right) \mathrm{d}s \right) \mathrm{d}t + \frac{1}{\sigma^2} f[\boldsymbol{Q}_1] \right)$$

Euler-Lagrange equations for *t*<1

$$\begin{aligned} \frac{\partial}{\partial t} \boldsymbol{Q}_t(s) &= \boldsymbol{u}_t(\boldsymbol{Q}_t(s)) \\ \frac{\partial}{\partial t} \boldsymbol{P}_t(s) &= -(\nabla \boldsymbol{u}(\boldsymbol{Q}_t(s)))^T \boldsymbol{P}_t(s) \\ \boldsymbol{m}_t(\boldsymbol{x}) &= \int_S \boldsymbol{P}_t(s) \delta(\boldsymbol{x} - \boldsymbol{Q}_t(s)) \, \mathrm{d}\, s \end{aligned}$$

where

$$\|\boldsymbol{u}_t + \epsilon \hat{\boldsymbol{u}}\|^2 = \|\boldsymbol{u}_t\|^2 + \epsilon \int_\Omega \boldsymbol{m}_t \cdot \hat{\boldsymbol{u}} \mathrm{d}V(\boldsymbol{x}) + \mathcal{O}(\epsilon^2)$$

Momentum formula

Momentum formula

For example: $\|\boldsymbol{u}\|^2 = \frac{1}{2}\int_\Omega |\boldsymbol{u}|^2 + \alpha^2 \|\nabla \boldsymbol{u}\|^2 \mathrm{d}V(\boldsymbol{x})$

Momentum formula

For example: $\|\boldsymbol{u}\|^2 = \frac{1}{2}\int_\Omega |\boldsymbol{u}|^2 + \alpha^2 \|\nabla \boldsymbol{u}\|^2 \mathrm{d}V(\boldsymbol{x})$

$$\|\boldsymbol{u} + \epsilon\hat{\boldsymbol{u}}\|^2 = \|\boldsymbol{u}\|^2 + \epsilon \int_\Omega (1 - \alpha^2\nabla^2)\boldsymbol{u} \cdot \hat{\boldsymbol{u}} \mathrm{d}V(\boldsymbol{x}) + \mathcal{O}(\epsilon^2)$$

Momentum formula

For example: $\|\boldsymbol{u}\|^2 = \frac{1}{2}\int_\Omega |\boldsymbol{u}|^2 + \alpha^2\|\nabla\boldsymbol{u}\|^2 \mathrm{d}V(\boldsymbol{x})$

$$\|\boldsymbol{u}+\epsilon\hat{\boldsymbol{u}}\|^2 = \|\boldsymbol{u}\|^2 + \epsilon\int_\Omega (1-\alpha^2\nabla^2)\boldsymbol{u}\cdot\hat{\boldsymbol{u}}\mathrm{d}V(\boldsymbol{x}) + \mathcal{O}(\epsilon^2)$$

So $\boldsymbol{m} = (1-\alpha^2\nabla^2)\boldsymbol{u}$

Momentum formula

For example: $\|\boldsymbol{u}\|^2 = \frac{1}{2}\int_\Omega |\boldsymbol{u}|^2 + \alpha^2\|\nabla\boldsymbol{u}\|^2 \mathrm{d}V(\boldsymbol{x})$

$$\|\boldsymbol{u}+\epsilon\hat{\boldsymbol{u}}\|^2 = \|\boldsymbol{u}\|^2 + \epsilon\int_\Omega (1-\alpha^2\nabla^2)\boldsymbol{u}\cdot\hat{\boldsymbol{u}}\mathrm{d}V(\boldsymbol{x}) + \mathcal{O}(\epsilon^2)$$

So $\boldsymbol{m} = (1-\alpha^2\nabla^2)\boldsymbol{u}$

Equations become

$$\begin{aligned}
\frac{\partial}{\partial t}\boldsymbol{Q}_t(s) &= \boldsymbol{u}_t(\boldsymbol{Q}_t(s)) \\
\frac{\partial}{\partial t}\boldsymbol{P}_t(s) &= -(\nabla\boldsymbol{u}(\boldsymbol{Q}_t(s)))^T\boldsymbol{P}_t(s) \\
(1-\alpha^2\nabla^2)\boldsymbol{u}_t(\boldsymbol{x}) &= \int_S \boldsymbol{P}_t(s)\delta(\boldsymbol{x}-\boldsymbol{Q}_t(s))\,\mathrm{d}\,s
\end{aligned}$$

Elimination

For any choice of norm, we can eliminate $\boldsymbol{P}$ and $\boldsymbol{Q}$ to get an equation purely in terms of $\boldsymbol{u}$

$$\frac{\partial}{\partial t}\boldsymbol{m} + \mathrm{ad}^*_{\boldsymbol{u}}\,\boldsymbol{m} = \frac{\partial}{\partial t}\boldsymbol{m} + \nabla\cdot(\boldsymbol{u}\boldsymbol{m}) + (\nabla\boldsymbol{u})^T\boldsymbol{m} = 0$$

Elimination

For any choice of norm, we can eliminate $\boldsymbol{P}$ and $\boldsymbol{Q}$ to get an equation purely in terms of $\boldsymbol{u}$

$$\frac{\partial}{\partial t}\boldsymbol{m} + \mathrm{ad}^*_{\boldsymbol{u}}\boldsymbol{m} = \frac{\partial}{\partial t}\boldsymbol{m} + \nabla\cdot(\boldsymbol{u}\boldsymbol{m}) + (\nabla\boldsymbol{u})^T\boldsymbol{m} = 0$$

This is the equation for geodesics on the diffeomorphism group with metric $\|\boldsymbol{u}_t\|^2$, known as EPDiff (Holm and Marsden, 2004)

Elimination

For any choice of norm, we can eliminate $\boldsymbol{P}$ and $\boldsymbol{Q}$ to get an equation purely in terms of $\boldsymbol{u}$

$$\frac{\partial}{\partial t}\boldsymbol{m} + \mathrm{ad}^*_{\boldsymbol{u}}\,\boldsymbol{m} = \frac{\partial}{\partial t}\boldsymbol{m} + \nabla\cdot(\boldsymbol{u}\boldsymbol{m}) + (\nabla\boldsymbol{u})^T\boldsymbol{m} = 0$$

This is the equation for geodesics on the diffeomorphism group with metric $\|\boldsymbol{u}_t\|^2$, known as EPDiff (Holm and Marsden, 2004)

For our example norm, we get the *n*-dimensional Camassa-Holm equation (Holm, Marsden and Ratiu, 1998)

Elimination

For any choice of norm, we can eliminate $\boldsymbol{P}$ and $\boldsymbol{Q}$ to get an equation purely in terms of $\boldsymbol{u}$

$$\frac{\partial}{\partial t}\boldsymbol{m} + \mathrm{ad}^*_{\boldsymbol{u}}\boldsymbol{m} = \frac{\partial}{\partial t}\boldsymbol{m} + \nabla \cdot (\boldsymbol{u}\boldsymbol{m}) + (\nabla \boldsymbol{u})^T \boldsymbol{m} = 0$$

This is the equation for geodesics on the diffeomorphism group with metric $\|\boldsymbol{u}_t\|^2$, known as EPDiff (Holm and Marsden, 2004)

For our example norm, we get the *n*-dimensional Camassa-Holm equation (Holm, Marsden and Ratiu, 1998)

In 1-dimension this becomes (Camassa and Holm, 1993)

$$m_t + \frac{\partial m}{\partial x}u + 2\frac{\partial u}{\partial x}m = 0, \qquad m = u - \alpha^2 \frac{\partial^2 u}{\partial x^2}$$

Matching condition

We don't want to require that specific points on the shapes get matched to each other (landmark matching)

Matching condition

We don't want to require that specific points on the shapes get matched to each other (landmark matching)

Density matching condition (Glaunes et al, 2005)

$$\mathrm{d}\mu_A(\boldsymbol{x}) = \int_S \hat{\mu}_A(s)\delta(\boldsymbol{x} - \boldsymbol{Q}^A(s))\,\mathrm{d}s\,\mathrm{d}V(\boldsymbol{x})$$

$$\mathrm{d}\mu_B(\boldsymbol{x}) = \int_S \hat{\mu}_B(s)\delta(\boldsymbol{x} - \boldsymbol{Q}^B(s))\,\mathrm{d}s\,\mathrm{d}V(\boldsymbol{x})$$

$$\mathrm{d}\mu_1(\boldsymbol{x}) = \int_S \hat{\mu}_A(s)\delta(\boldsymbol{x} - \boldsymbol{Q}_1(s))\,\mathrm{d}s\,\mathrm{d}V(\boldsymbol{x})$$

Matching condition

We don't want to require that specific points on the shapes get matched to each other (landmark matching)

Density matching condition (Glaunes et al, 2005)

$$\mathrm{d}\mu_A(\boldsymbol{x}) = \int_S \hat{\mu}_A(s)\delta(\boldsymbol{x} - \boldsymbol{Q}^A(s))\,\mathrm{d}s\,\mathrm{d}V(\boldsymbol{x})$$

$$\mathrm{d}\mu_B(\boldsymbol{x}) = \int_S \hat{\mu}_B(s)\delta(\boldsymbol{x} - \boldsymbol{Q}^B(s))\,\mathrm{d}s\,\mathrm{d}V(\boldsymbol{x})$$

$$\mathrm{d}\mu_1(\boldsymbol{x}) = \int_S \hat{\mu}_A(s)\delta(\boldsymbol{x} - \boldsymbol{Q}_1(s))\,\mathrm{d}s\,\mathrm{d}V(\boldsymbol{x})$$

Penalty functional is

$$f[\boldsymbol{Q}_1] = \int_\Omega \int_\Omega K(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\nu(\boldsymbol{x})\mathrm{d}\nu(\boldsymbol{y}), \quad \mathrm{d}\nu = \mathrm{d}\mu_B - \mathrm{d}\mu_1$$

Momentum

- Geodesic is determined by initial conditions for shape coordinates Q and conjugate momenta P
- Given a reference shape, have an isomorphism between any topologically equivalent shape and initial conditions for P
- P is in a linear space so we can apply linear statistical techniques to this representation

Relabelling symmetry

Relabelling symmetry

- As noted by Holm and Marsden (2004) in the context of the n-dimensional CH equation, the Lagrangian is invariant to reparameterisations of the curve

Relabelling symmetry

- As noted by Holm and Marsden (2004) in the context of the n-dimensional CH equation, the Lagrangian is invariant to reparameterisations of the curve
- Corresponding conserved momenta is

Relabelling symmetry

- As noted by Holm and Marsden (2004) in the context of the n-dimensional CH equation, the Lagrangian is invariant to reparameterisations of the curve
- Corresponding conserved momenta is

$$\boldsymbol{P} \cdot \mathrm{d}\boldsymbol{Q}$$

Relabelling symmetry

Relabelling symmetry

- This relabelling symmetry plays an important role in diffeomorphic matching of images

Relabelling symmetry

- This relabelling symmetry plays an important role in diffeomorphic matching of images
- We have relaxed the endpoint condition by requiring that embeddings are equivalent rather than setting boundary condition for each parameter value

Relabelling symmetry

- This relabelling symmetry plays an important role in diffeomorphic matching of images
- We have relaxed the endpoint condition by requiring that embeddings are equivalent rather than setting boundary condition for each parameter value
- The optimal solution has momentum which is normal to the shape (see Miller, Trouvé and Younes (2003))

Momentum condition

Momentum condition

Easiest to see this for the case of advected functions

Momentum condition

Easiest to see this for the case of advected functions

$$\frac{\delta \ell}{\delta \boldsymbol{u}} = -\pi \nabla f$$

Momentum condition

Easiest to see this for the case of advected functions

$$\frac{\delta \ell}{\delta \boldsymbol{u}} = -\pi \nabla f$$

The momentum field is normal to level sets of the image function

Momentum condition

Easiest to see this for the case of advected functions

$$\frac{\delta \ell}{\delta \boldsymbol{u}} = -\pi \nabla f$$

The momentum field is normal to level sets of the image function

This is closely linked to the Clebsch representation of fluid dynamics, and the Kelvin circulation theorem

Particle-mesh discretisation

Particle-mesh method

Particle-mesh method

- Lagrangian **particles** moving with the flow represent the movement of the embedded curve/surface

Particle-mesh method

- Lagrangian **particles** moving with the flow represent the movement of the embedded curve/surface
- Velocity represented on a fixed **mesh**

Particle-mesh method

- Lagrangian **particles** moving with the flow represent the movement of the embedded curve/surface
- Velocity represented on a fixed **mesh**
- Closely related to Hamiltonian particle-mesh method (HPM) for shallow-water equations (Frank, Gottwald, Reich)

Interpolation

Given a set of velocities $\{\boldsymbol{u}_k\}_{k=1}^{n_g}$ stored at grid points $\{\boldsymbol{x}_k\}_{k=1}^{n_g}$ define a velocity field everywhere on Ω

Interpolation

Given a set of velocities $\{\boldsymbol{u}_k\}_{k=1}^{n_g}$ stored at grid points $\{\boldsymbol{x}_k\}_{k=1}^{n_g}$ define a velocity field everywhere on Ω

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{x})$$

Interpolation

Given a set of velocities $\{\boldsymbol{u}_k\}_{k=1}^{n_g}$ stored at grid points $\{\boldsymbol{x}_k\}_{k=1}^{n_g}$ define a velocity field everywhere on Ω

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{x})$$

Dynamical condition becomes

$$\frac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_t(s))$$

Interpolation

Given a set of velocities $\{\boldsymbol{u}_k\}_{k=1}^{n_g}$ stored at grid points $\{\boldsymbol{x}_k\}_{k=1}^{n_g}$

define a velocity field everywhere on Ω

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{x})$$

Dynamical condition becomes

$$\frac{\partial}{\partial t}\boldsymbol{Q}_t(s) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_t(s))$$

Now restrict to a finite set of points $\{\boldsymbol{Q}_\beta\}_{\beta=1}^{n_p}$

$$\dot{\boldsymbol{Q}}_\beta = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_\beta)$$

Particle-mesh equations

We adopt a **geometric approach** by trying to exactly optimise a discretised functional

would be momentum map if action

Particle-mesh equations

We adopt a **geometric approach** by trying to exactly optimise a discretised functional

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}}\left(\int_0^1\left(\|\boldsymbol{u}_t\|^2+\int_S \boldsymbol{P}_t(s)\cdot\left(\frac{\partial}{\partial t}\boldsymbol{Q}_t(s)-\int_{\boldsymbol{\Omega}}\boldsymbol{u}_t(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{Q}_t(s))\mathrm{d}V(\boldsymbol{x})\right)\mathrm{d}s\right)\mathrm{d}t+\frac{1}{\sigma^2}f[\boldsymbol{Q}_1]\right)$$

would be momentum map if action

Particle-mesh equations

We adopt a **geometric approach** by trying to exactly optimise a discretised functional

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\int_0^1 \left(\|\boldsymbol{u}\|_g^2 + \sum_\beta \boldsymbol{P}_\beta \cdot \left(\dot{\boldsymbol{Q}}_\beta - \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_\beta) \right) \right) \mathrm{d}t + \frac{1}{\sigma^2} \hat{f}(\boldsymbol{Q}(1)) \right)$$

would be momentum map if action

Particle-mesh equations

We adopt a **geometric approach** by trying to exactly optimise a discretised functional

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\int_0^1 \left(\|\boldsymbol{u}\|_g^2 + \sum_\beta \boldsymbol{P}_\beta \cdot \left(\dot{\boldsymbol{Q}}_\beta - \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_\beta) \right) \right) \mathrm{d}t + \frac{1}{\sigma^2} \hat{f}(\boldsymbol{Q}(1)) \right)$$

Semi-discrete equations are

$$\begin{aligned} \dot{\boldsymbol{Q}}_\beta &= \textstyle\sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_\beta) \\ \dot{\boldsymbol{P}}_\beta &= -\boldsymbol{P}_\beta \cdot \textstyle\sum_k \boldsymbol{u}_k \nabla \psi_k(\boldsymbol{Q}_\beta) \\ \frac{\partial}{\partial \boldsymbol{u}_k} \|\boldsymbol{u}\|_g^2 &= \textstyle\sum_\beta \boldsymbol{P}_\beta \psi_k(\boldsymbol{Q}_\beta) \end{aligned}$$

would be momentum map if action

Particle-mesh equations

We adopt a **geometric approach** by trying to exactly optimise a discretised functional

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\int_0^1 \left(\|\boldsymbol{u}\|_g^2 + \sum_\beta \boldsymbol{P}_\beta \cdot \left(\dot{\boldsymbol{Q}}_\beta - \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_\beta) \right) \right) \mathrm{d}t + \frac{1}{\sigma^2} \hat{f}(\boldsymbol{Q}(1)) \right)$$

Semi-discrete equations are

$$\begin{aligned} \dot{\boldsymbol{Q}}_\beta &= \textstyle\sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_\beta) \\ \dot{\boldsymbol{P}}_\beta &= -\boldsymbol{P}_\beta \cdot \textstyle\sum_k \boldsymbol{u}_k \nabla \psi_k(\boldsymbol{Q}_\beta) \\ \frac{\partial}{\partial \boldsymbol{u}_k} \|\boldsymbol{u}\|_g^2 &= \textstyle\sum_\beta \boldsymbol{P}_\beta \psi_k(\boldsymbol{Q}_\beta) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial t} \boldsymbol{Q}_t(s) &= \boldsymbol{u}_t(\boldsymbol{Q}_t(s)) \\ \frac{\partial}{\partial t} \boldsymbol{P}_t(s) &= -(\nabla \boldsymbol{u}(\boldsymbol{Q}_t(s)))^T \cdot \boldsymbol{P}_t(s) \\ \boldsymbol{m}_t &= \textstyle\int_S \boldsymbol{P}_t(s) \delta(\boldsymbol{x} - \boldsymbol{P}_t(s)) \mathrm{d}s \end{aligned}$$

would be momentum map if action

Discrete matching condition

Discrete matching condition

Use a particle-mesh approach

Discrete matching condition

Use a particle-mesh approach

$$\mu_k^1 = \sum_\beta \hat{\mu}_\beta^A \psi_k(\boldsymbol{Q}_\beta(1))$$

$$\mu_k^B = \sum_\beta \hat{\mu}_\beta^B \psi_k(\boldsymbol{Q}_\beta^B)$$

Discrete matching condition

Use a particle-mesh approach

$$\mu_k^1 = \sum_\beta \hat{\mu}_\beta^A \psi_k(\boldsymbol{Q}_\beta(1))$$

$$\mu_k^B = \sum_\beta \hat{\mu}_\beta^B \psi_k(\boldsymbol{Q}_\beta^B)$$

$$\hat{f}(\boldsymbol{Q}(1)) = \sum_{kl} K_{kl}\nu_k\nu_l, \quad \nu_k = \mu_k^1 - \mu_k^B$$

Time discretisation

Discretise in time in the functional

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\Delta t \sum_{n=1}^{N} \left(\|\boldsymbol{u}^{n-1}\|_g^2 + \sum_\beta \boldsymbol{P}_\beta^{n-1} \cdot \left(\frac{\boldsymbol{Q}_\beta^n - \boldsymbol{Q}_\beta^{n-1}}{\Delta t} - \sum_k \boldsymbol{u}_k^{n-1} \psi_k(\boldsymbol{Q}_\beta^{n-1}) \right) \right) \mathrm{d}t + \frac{1}{\sigma^2} \hat{f}(\boldsymbol{Q}^N) \right)$$

Get symplectic Euler discretisation

$$\begin{aligned}
\boldsymbol{Q}_\beta^n &= \boldsymbol{Q}_\beta^{n-1} + \Delta t \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_\beta^{n-1}) \\
\boldsymbol{P}_\beta^n &= \boldsymbol{P}_\beta^{n-1} - \Delta t \boldsymbol{P}_\beta^n \cdot \sum_k \boldsymbol{u}_k \nabla \psi_k(\boldsymbol{Q}_\beta^{n-1}) \\
\frac{\partial}{\partial \boldsymbol{u}_k} \|\boldsymbol{u}^{n-1}\|_g^2 &= \sum_\beta \boldsymbol{P}_\beta^{n-1} \psi_k(\boldsymbol{Q}_\beta^{n-1})
\end{aligned}$$

Time discretisation

Discretise in time in the functional

$$\min_{\boldsymbol{u},\boldsymbol{Q},\boldsymbol{P}} \left(\Delta t \sum_{n=1}^{N} \left(\|\boldsymbol{u}^{n-1}\|_g^2 + \sum_\beta \boldsymbol{P}_\beta^{n-1} \cdot \left(\frac{\boldsymbol{Q}_\beta^n - \boldsymbol{Q}_\beta^{n-1}}{\Delta t} - \sum_k \boldsymbol{u}_k^{n-1} \psi_k(\boldsymbol{Q}_\beta^{n-1}) \right) \right) \mathrm{d}t + \frac{1}{\sigma^2} \hat{f}(\boldsymbol{Q}^N) \right)$$

Get symplectic Euler discretisation

$$\begin{aligned} \boldsymbol{Q}_\beta^n &= \boldsymbol{Q}_\beta^{n-1} + \Delta t \textstyle\sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}_\beta^{n-1}) \\ \boldsymbol{P}_\beta^n &= \boldsymbol{P}_\beta^{n-1} - \Delta t \boldsymbol{P}_\beta^n \cdot \textstyle\sum_k \boldsymbol{u}_k \nabla \psi_k(\boldsymbol{Q}_\beta^{n-1}) \\ \frac{\partial}{\partial \boldsymbol{u}_k} \|\boldsymbol{u}^{n-1}\|_g^2 &= \textstyle\sum_\beta \boldsymbol{P}_\beta^{n-1} \psi_k(\boldsymbol{Q}_\beta^{n-1}) \end{aligned}$$

Can get higher-order methods by discretising the dynamical equation with a RK method; get a symplectic PRK method

Discrete conservation

Although we have a discrete set of particles, we still have continuous velocity fields defined by

Discrete conservation

Although we have a discrete set of particles, we still have continuous velocity fields defined by

$$\boldsymbol{u}(\boldsymbol{x}, t) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{x})$$

Discrete conservation

Although we have a discrete set of particles, we still have continuous velocity fields defined by

$$\boldsymbol{u}(\boldsymbol{x}, t) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{x})$$

Hence it is possible to choose a continuous curve through the set of points and to ask how it evolves

Discrete conservation

Although we have a discrete set of particles, we still have continuous velocity fields defined by

$$\boldsymbol{u}(\boldsymbol{x},t) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{x})$$

Hence it is possible to choose a continuous curve through the set of points and to ask how it evolves

$$\frac{\partial}{\partial t}\boldsymbol{Q}(s,t) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{Q}(s,t))$$

Discrete conservation

Discrete conservation

More generally, we can follow how the flow map evolves over the whole space

Discrete conservation

More generally, we can follow how the flow map evolves over the whole space

$$\frac{\partial}{\partial t}\boldsymbol{g}(\boldsymbol{x},t) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{g}(\boldsymbol{x},t))$$

Discrete conservation

More generally, we can follow how the flow map evolves over the whole space

$$\frac{\partial}{\partial t}\boldsymbol{g}(\boldsymbol{x},t) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{g}(\boldsymbol{x},t))$$

and we can follow the Jacobian of this map

Discrete conservation

More generally, we can follow how the flow map evolves over the whole space

$$\frac{\partial}{\partial t}\boldsymbol{g}(\boldsymbol{x},t) = \sum_k \boldsymbol{u}_k \psi_k(\boldsymbol{g}(\boldsymbol{x},t))$$

and we can follow the Jacobian of this map

$$\frac{\partial}{\partial t}\frac{\partial \boldsymbol{g}(\boldsymbol{x},t)}{\partial \boldsymbol{x}} = \sum_k \boldsymbol{u}_k \nabla\psi_k(\boldsymbol{g}(\boldsymbol{x},t)) \cdot \frac{\partial \boldsymbol{g}(\boldsymbol{x},t)}{\partial \boldsymbol{x}}$$

Discrete conservation

Discrete conservation

As a post-processing step, this equation is independent for each point, and in particular:

Discrete conservation

As a post-processing step, this equation is independent for each point, and in particular:

$$\frac{\partial}{\partial t} J_\beta = \sum_k \boldsymbol{u}_k \nabla \psi_k(\boldsymbol{Q}_\beta) \cdot J_\beta, \quad J_\beta = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}(\boldsymbol{Q}_\beta(t), t)$$

Discrete conservation

As a post-processing step, this equation is independent for each point, and in particular:

$$\frac{\partial}{\partial t} J_\beta = \sum_k \boldsymbol{u}_k \nabla \psi_k(\boldsymbol{Q}_\beta) \cdot J_\beta, \quad J_\beta = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}(\boldsymbol{Q}_\beta(t), t)$$

This gives a simple way of computing Jacobi information

Discrete conservation

Discrete conservation

A variation $\delta \boldsymbol{Q}_\beta(0)$ in the initial conditions $\boldsymbol{Q}_\beta(0)$

Discrete conservation

A variation $\delta \boldsymbol{Q}_\beta(0)$ in the initial conditions $\boldsymbol{Q}_\beta(0)$ leads to a variation in the entire trajectory given by

Discrete conservation

A variation $\delta \boldsymbol{Q}_\beta(0)$ in the initial conditions $\boldsymbol{Q}_\beta(0)$ leads to a variation in the entire trajectory given by

$$\delta \boldsymbol{Q}_\beta(t) = J_\beta(t) \boldsymbol{Q}_\beta(0)$$

Discrete conservation

A variation $\delta \boldsymbol{Q}_\beta(0)$ in the initial conditions $\boldsymbol{Q}_\beta(0)$ leads to a variation in the entire trajectory given by

$$\delta \boldsymbol{Q}_\beta(t) = J_\beta(t) \boldsymbol{Q}_\beta(0)$$

This variation is a symmetry of the action principle, with corresponding conserved momentum

Discrete conservation

A variation $\delta \boldsymbol{Q}_\beta(0)$ in the initial conditions $\boldsymbol{Q}_\beta(0)$ leads to a variation in the entire trajectory given by

$$\delta \boldsymbol{Q}_\beta(t) = J_\beta(t) \boldsymbol{Q}_\beta(0)$$

This variation is a symmetry of the action principle, with corresponding conserved momentum

$$\boldsymbol{P}_\beta \cdot J_\beta(t)$$

Discrete conservation

A variation $\delta \boldsymbol{Q}_\beta(0)$ in the initial conditions $\boldsymbol{Q}_\beta(0)$ leads to a variation in the entire trajectory given by

$$\delta \boldsymbol{Q}_\beta(t) = J_\beta(t) \boldsymbol{Q}_\beta(0)$$

This variation is a symmetry of the action principle, with corresponding conserved momentum

$$\boldsymbol{P}_\beta \cdot J_\beta(t)$$

This property is preserved by any variational integrator

Discrete conservation

A variation $\delta \boldsymbol{Q}_\beta(0)$ in the initial conditions $\boldsymbol{Q}_\beta(0)$ leads to a variation in the entire trajectory given by

$$\delta \boldsymbol{Q}_\beta(t) = J_\beta(t) \boldsymbol{Q}_\beta(0)$$

This variation is a symmetry of the action principle, with corresponding conserved momentum

$$\boldsymbol{P}_\beta \cdot J_\beta(t)$$

This property is preserved by any variational integrator

This means that, if the momentum starts normal to the shape it will stay normal to the shape in the numerical solution

Solution methods

Three different approaches

Solution methods

Three different approaches

Find initial conditions $\boldsymbol{P}_\beta^0$ which minimise $\hat{f}(\boldsymbol{Q}^N)$ subject to the dynamical equations for $\boldsymbol{P}, \boldsymbol{Q}$

Solution methods

Three different approaches

Find initial conditions $\boldsymbol{P}_\beta^0$ which minimise $\hat{f}(\boldsymbol{Q}^N)$ subject to the dynamical equations for $\boldsymbol{P}, \boldsymbol{Q}$

Find a time series $\{\boldsymbol{P}_\beta^n\}_{n=0}^{N-1}$ which minimises the matching penalty term subject to the dynamical equations for $\boldsymbol{Q}$

Solution methods

Three different approaches

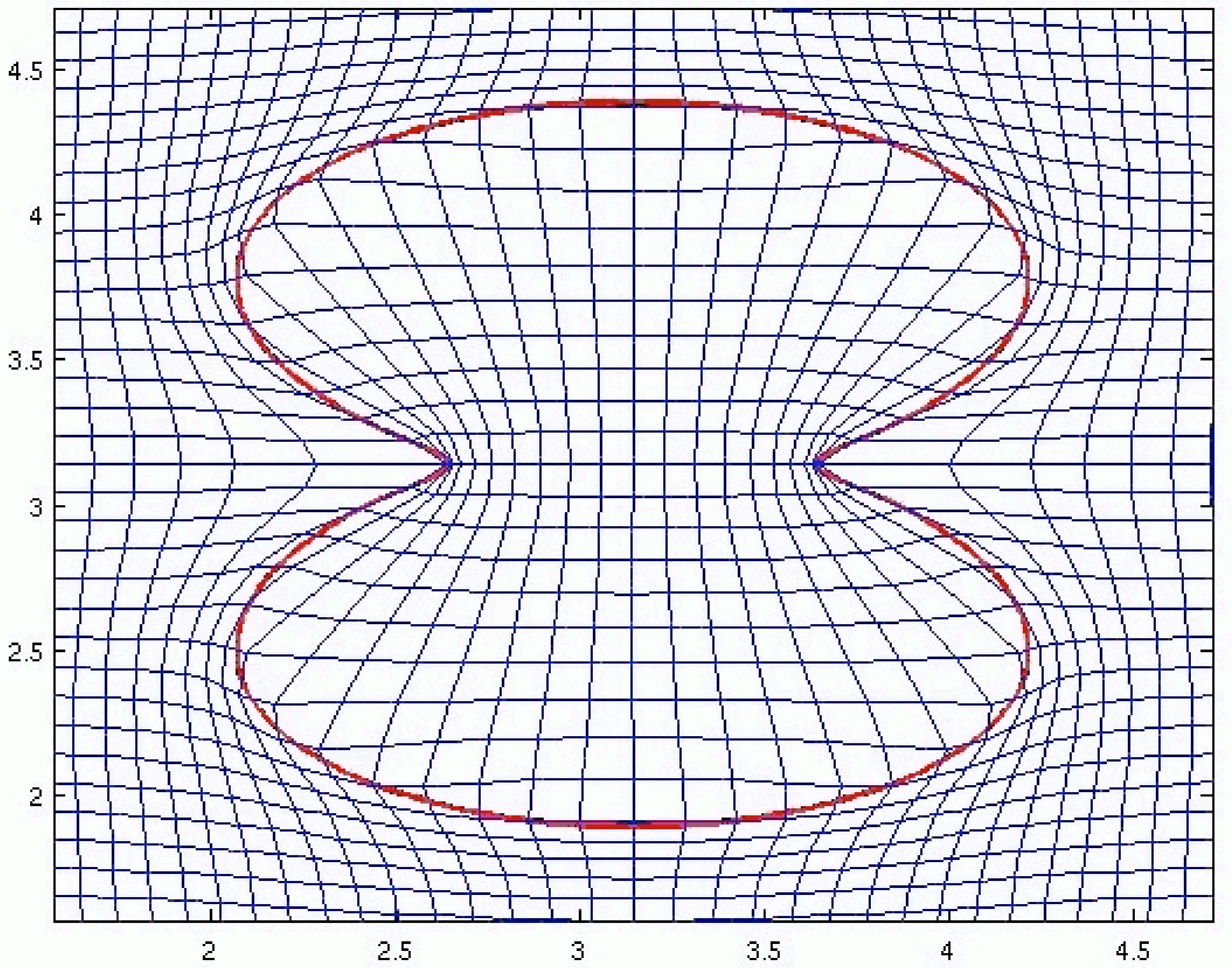
Find initial conditions $\boldsymbol{P}_\beta^0$ which minimise $\hat{f}(\boldsymbol{Q}^N)$ subject to the dynamical equations for $\boldsymbol{P}, \boldsymbol{Q}$

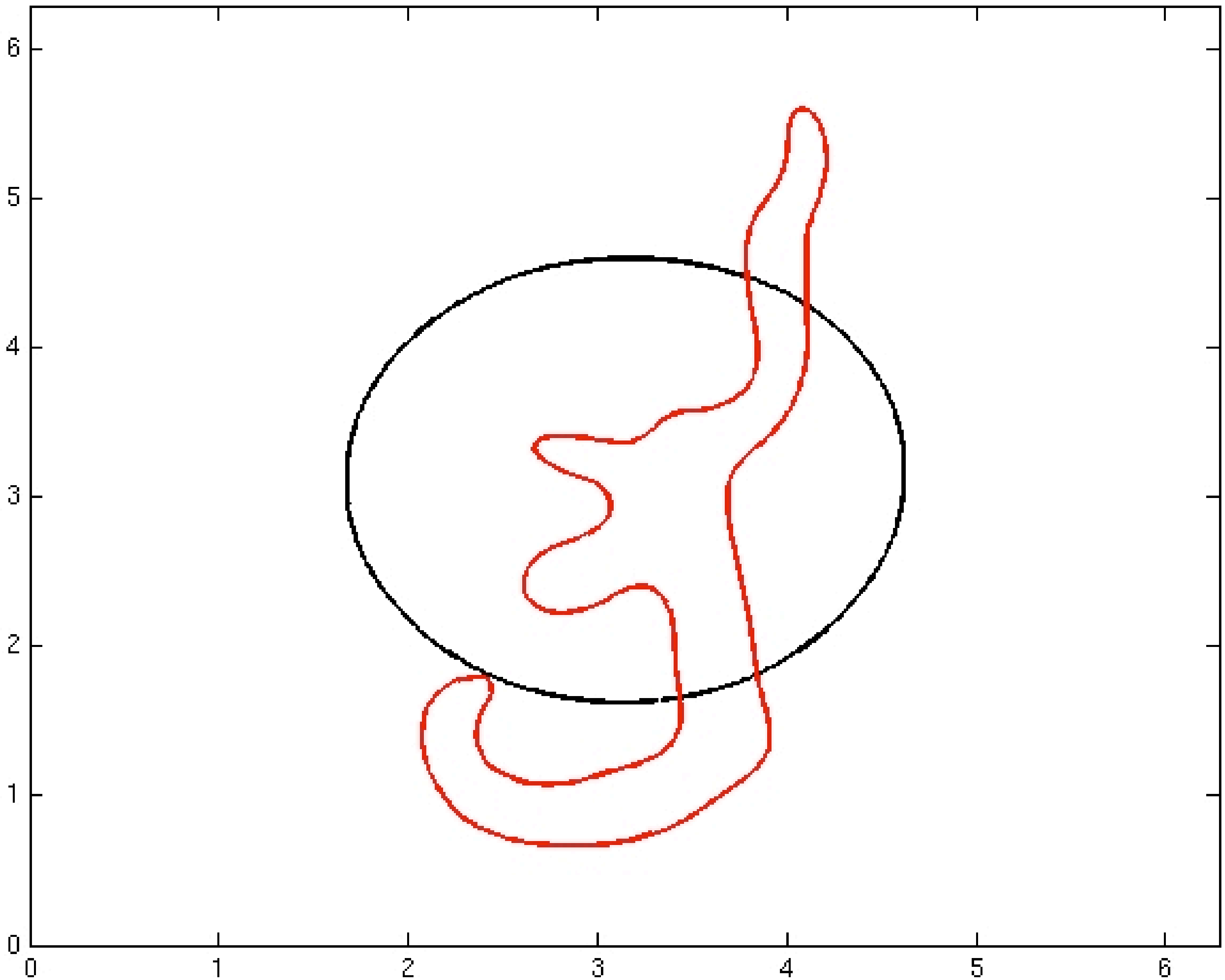
Find a time series $\{\boldsymbol{P}_\beta^n\}_{n=0}^{N-1}$ which minimises the matching penalty term subject to the dynamical equations for $\boldsymbol{Q}$

Extremise the discrete functional directly

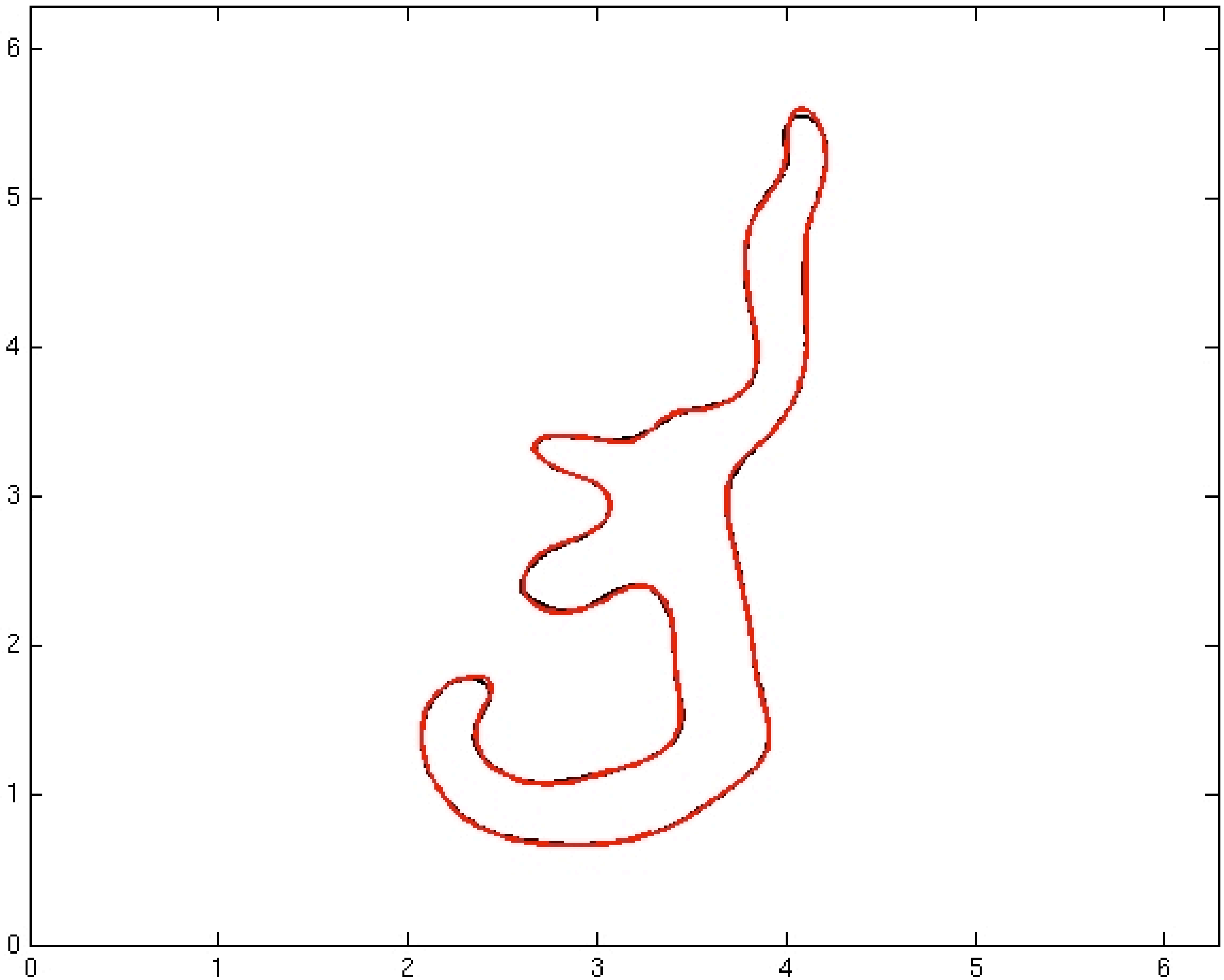
Some movies

What does the shortest path between two shapes look like?


4.5
4
3.5
3
2.5
2
2
2.5
3
3.5
4
4.5



6
5
4
3
2
1
0
0
1
2
3
4
5
6



0
1
2
3
4
5
6
0
1
2
3
4
5
6

Summary

- Variational shape matching is useful in a wide variety of applications: analysis of shape datasets and shape optimisation
- Necessary when large deformations are needed
- Shapes are embedded in a flow which follows geodesics in diffeomorphism group
- Particle-mesh method gives simple discretisation with geometric properties for matching curves/surfaces

Outlook

- Collaborations with engineers at ICL and elsewhere
- Parallel algorithm for matching complex shapes
- Developing statistical analysis of shapes with Sofia Olhede (Maths, ICL→UCL)

The End