# Spectral Rigid Body Dynamics

Mikola Lysenko

May 5, 2010

# Overview

Rigid Body Dynamics

Lagrangian Mechanics

Standard Collisions

Constraint Based Collisions

Fourier Methods

# Rigid Body Dynamics

An approximate model of low energy physics for stiff objects

# Rigid Body Dynamics

An approximate model of low energy physics for stiff objects

Pros:

- + Pretty accurate at human energy scales
- + Good for stiff materials (ie metals, plastics etc.)
- + Easy kinematic constraints (useful for mechanisms)
- + Standard animation tool (videogames!)

# Rigid Body Dynamics

An approximate model of low energy physics for stiff objects

Pros:

- $+$ Pretty accurate at human energy scales
- $+$ Good for stiff materials (ie metals, plastics etc.)
- $+$ Easy kinematic constraints (useful for mechanisms)
- $+$ Standard animation tool (videogames!)

Cons:

- - Inaccurate at extremely large energies
- - Bad for materials with low elastic modulus
- - Not always solvable! (See: Painleve's paradox)

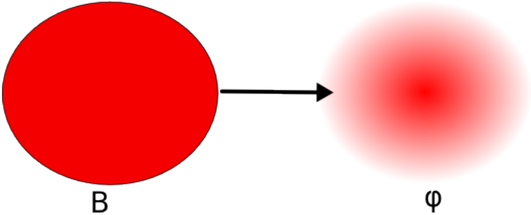
# What is a Rigid Body?

An idealized solid object with elastic modulus $= \infty$

# What is a Rigid Body?

An idealized solid object with elastic modulus $= \infty$

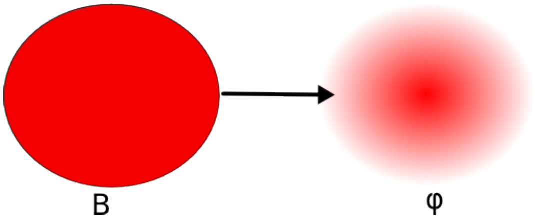
We identify a body $B$ with a scalar field, $\varphi : \mathbb{R}^d \to \mathbb{R}^+$



# What is a Rigid Body?

An idealized solid object with elastic modulus $= \infty$

We identify a body $B$ with a scalar field, $\varphi : \mathbb{R}^d \to \mathbb{R}^+$

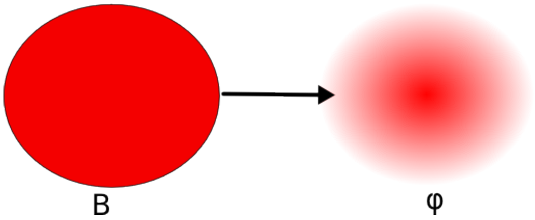



$\varphi$ represents the mass distribution of $B$

# What is a Rigid Body?

An idealized solid object with elastic modulus $= \infty$

We identify a body $B$ with a scalar field, $\varphi : \mathbb{R}^d \to \mathbb{R}^+$




$\varphi$ represents the mass distribution of $B$

$\varphi(x) = 0$ indicates $B$ does not occupy the space at $x$

# Configuration Space of a Rigid Body

Transformations of rigid mass fields must preserve distance and handedness

# Configuration Space of a Rigid Body

Transformations of rigid mass fields must preserve distance and handedness

In other words, must be a direct Euclidean isometry

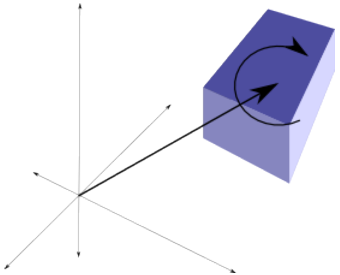
Isomorphic to finite dimensional Lie group, $SE(d) \cong SO(d) \ltimes \mathbb{R}^d$

# Configuration Space of a Rigid Body

Transformations of rigid mass fields must preserve distance and handedness

In other words, must be a direct Euclidean isometry

Isomorphic to finite dimensional Lie group, $SE(d) \cong SO(d) \ltimes \mathbb{R}^d$



Can be parameterized by a translation $t$ and a rotation $R$

Matrix: $\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$

$\binom{d+1}{2}$ degrees of freedom

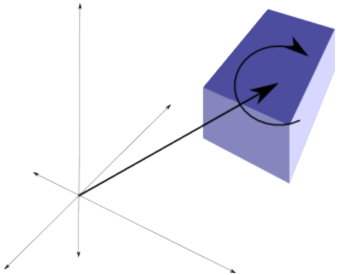
Tangent space: $\mathfrak{so}(d+1)$

# Configuration Space of a Rigid Body

Transformations of rigid mass fields must preserve distance and handedness

In other words, must be a direct Euclidean isometry

Isomorphic to finite dimensional Lie group, $SE(d) \cong SO(d) \ltimes \mathbb{R}^d$



Can be parameterized by a translation $t$ and a rotation $R$

Matrix: $\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$

$\binom{d+1}{2}$ degrees of freedom

Tangent space: $\mathfrak{so}(d+1)$

**Motions of rigid objects** $\cong$ **curves** $q(t) \subset SE(d)$

# Lagrangian Mechanics

Turns physics into an optimization problem.

# Lagrangian Mechanics

Turns physics into an optimization problem.

Given a configuration curve $q$ at time $t$, define the *Lagrangian*

$$\mathcal{L}(q, \dot{q}, t) = T(\dot{q}) - U(q, t)$$

Where $T(\dot{q}) = \frac{1}{2}\dot{q}^T M \dot{q}$ is the kinetic energy and $U$ is a potential

# Lagrangian Mechanics

Turns physics into an optimization problem.

Given a configuration curve $q$ at time $t$, define the *Lagrangian*

$$\mathcal{L}(q, \dot{q}, t) = T(\dot{q}) - U(q, t)$$

Where $T(\dot{q}) = \frac{1}{2}\dot{q}^T M\dot{q}$ is the kinetic energy and $U$ is a potential

**Hamilton's Principle of Least Action**:

*Physically plausible motions do minimal work*

# Lagrangian Mechanics

Turns physics into an optimization problem.

Given a configuration curve $q$ at time $t$, define the *Lagrangian*

$$\mathcal{L}(q, \dot{q}, t) = T(\dot{q}) - U(q, t)$$

Where $T(\dot{q}) = \frac{1}{2}\dot{q}^T M \dot{q}$ is the kinetic energy and $U$ is a potential

**Hamilton's Principle of Least Action**:

*Physically plausible motions do minimal work*

In other words:

$$\operatorname*{argmin}_{q:[t_0,t_1)\to SE(d)} \int_{t_0}^{t_1} L(q, \dot{q}, t)dt$$

# Forward Kinematics

Solve:

$$\underset{q:[t_0,t_1)\to SE(d)}{\operatorname{argmin}} \int_{t_0}^{t_1} L(q, \dot{q}, t)\,dt$$

# Forward Kinematics

Solve:

$$\underset{q:[t_0,t_1)\to SE(d)}{\operatorname{argmin}} \int_{t_0}^{t_1} L(q, \dot{q}, t)dt$$

This is a 1st order variational problem, so use Euler-Lagrange:

$$\frac{d}{dt}\left(\frac{\partial L(q, \dot{q}, t)}{\partial \dot{q}}\right) = \frac{\partial L(q, \dot{q}, t)}{\partial q}$$

# Forward Kinematics

Solve:

$$\underset{q:[t_0,t_1)\to SE(d)}{\operatorname{argmin}} \int_{t_0}^{t_1} L(q, \dot{q}, t)dt$$

This is a 1st order variational problem, so use Euler-Lagrange:

$$\frac{d}{dt}\left(\frac{\partial L(q, \dot{q}, t)}{\partial \dot{q}}\right) = \frac{\partial L(q, \dot{q}, t)}{\partial q}$$

Unpack definitions and simplify:

$$\frac{d}{dt}\left(\frac{\partial \frac{1}{2}\dot{q}^T M \dot{q}}{\partial \dot{q}}\right) = \frac{\partial U(q, t)}{\partial q}$$

# Forward Kinematics

Solve:

$$\underset{q:[t_0,t_1)\to SE(d)}{\operatorname{argmin}} \int_{t_0}^{t_1} L(q,\dot{q},t)dt$$

This is a 1st order variational problem, so use Euler-Lagrange:

$$\frac{d}{dt}\left(\frac{\partial L(q,\dot{q},t)}{\partial \dot{q}}\right) = \frac{\partial L(q,\dot{q},t)}{\partial q}$$

Unpack definitions and simplify:

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial \frac{1}{2}\dot{q}^T M \dot{q}}{\partial \dot{q}}\right) &= \frac{\partial U(q,t)}{\partial q}\\ M\ddot{q} &= \nabla U\end{aligned}$$

Newton's equations!

# Multiple Bodies

Q: How to deal with multiple independent bodies?

# Multiple Bodies

Q: How to deal with multiple independent bodies?

A: Tensor sum

Let $B_i, B_j$ be independent rigid bodies with motions $q_i, q_j$

| | |
|---|---|
| Configuration space | $SE(d)^2 \cong SE(d) \oplus SE(d)$ |
| Motion | $q(t) \cong q_i(t) \oplus q_j(t)$ |
| Lagrangian | $L(q, \dot{q}, t) = L(q_i, \dot{q}_i, t) + L(q_j, \dot{q}_j, t)$ |

# Multiple Bodies

Q: How to deal with multiple independent bodies?

A: Tensor sum

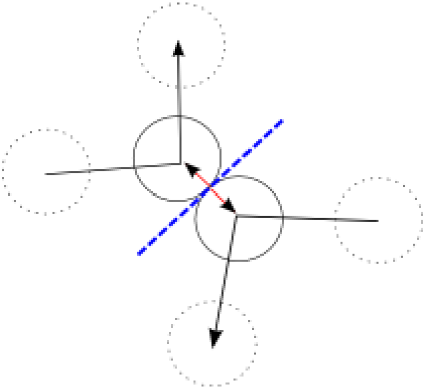
Let $B_i, B_j$ be independent rigid bodies with motions $q_i, q_j$

| | |
|---|---|
| Configuration space | $SE(d)^2 \cong SE(d) \oplus SE(d)$ |
| Motion | $q(t) \cong q_i(t) \oplus q_j(t)$ |
| Lagrangian | $L(q, \dot{q}, t) = L(q_i, \dot{q}_i, t) + L(q_j, \dot{q}_j, t)$ |

Scales to $n$ bodies, get Lagrangian in $SE(d)^n$

# Collisions

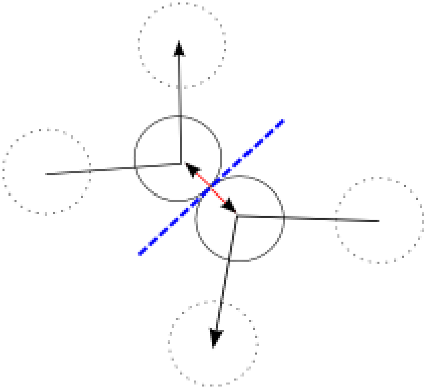
Solids can't physically occupy the same space.



Need to keep them separated

# Collisions

Solids can't physically occupy the same space.



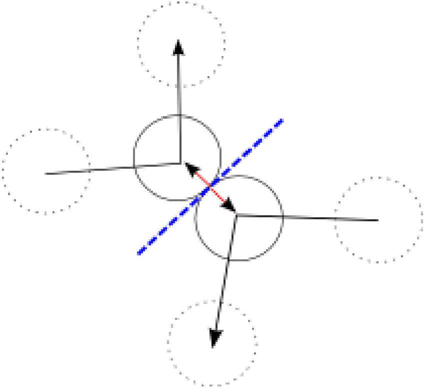
Need to keep them separated

**Standard method:**

- Time step to point of impact

# Collisions

Solids can't physically occupy the same space.



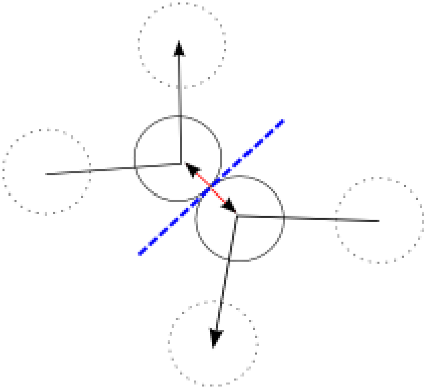
Need to keep them separated

**Standard method:**

- Time step to point of impact
- Determine contact manifold

# Collisions

Solids can't physically occupy the same space.



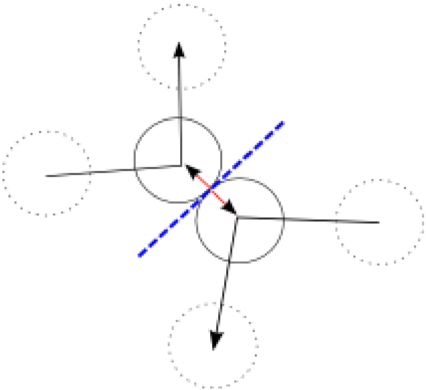
Need to keep them separated

**Standard method:**

- Time step to point of impact
- Determine contact manifold
- Apply penalty force/impulse to separate objects

# Collisions

Solids can't physically occupy the same space.



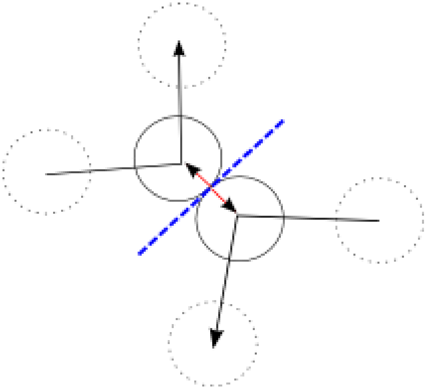
Need to keep them separated

**Standard method:**

- Time step to point of impact
- Determine contact manifold
- Apply penalty force/impulse to separate objects
- Iterate until all collisions are resolved

# Collisions

Solids can't physically occupy the same space.



Need to keep them separated

**Standard method:**

- Time step to point of impact
- Determine contact manifold
- Apply penalty force/impulse to separate objects
- Iterate until all collisions are resolved

\+ Just like high school physics

# Problems with typical collision methods

Handling collisions this way is hard

# Problems with typical collision methods

Handling collisions this way is hard

- - Finding exact point/time of impact is expensive
- - Contact manifolds can be difficult to classify
- - Normal forces are ambiguous for curvy shapes
- - Impulses are discontinuous; results in stiff system
- - Penalty methods don't gaurantee separation
- - Convergence properties not well understood
- - Unclear how to handle sharp corners
- - Discontinuity in simulation
- - Complicated, ad-hoc

# Problems with typical collision methods

Handling collisions this way is hard

- - Finding exact point/time of impact is expensive
- - Contact manifolds can be difficult to classify
- - Normal forces are ambiguous for curvy shapes
- - Impulses are discontinuous; results in stiff system
- - Penalty methods don't gaurantee separation
- - Convergence properties not well understood
- - Unclear how to handle sharp corners
- - Discontinuity in simulation
- - Complicated, ad-hoc

But can be made to work with enough hacking...

## There has got to be an easier way...

# There has got to be an easier way...

Minimal requirement for physical plausibility

*At all times no two solids intersect*

# There has got to be an easier way...

Minimal requirement for physical plausibility

*At all times no two solids intersect*

Is this really all there is to it?

# Constraint Based Impacts

For any body $B_i$ with mass field $\varphi_i$, define the set:

$$A_i = \kappa\iota \text{ supp } \varphi$$

# Constraint Based Impacts

For any body $B_i$ with mass field $\varphi_i$, define the set:

$$A_i = \kappa\iota \text{ supp } \varphi$$

So two solids, $A_i, A_j$, *collide* at a configuration $q_i, q_j$ iff:

$$\text{collide}(q_i A_i, q_j A_j) \Leftrightarrow \iota(q_i A_i \cap q_j A_j) \neq \emptyset$$

# Constraint Based Impacts

For any body $B_i$ with mass field $\varphi_i$, define the set:

$$A_i = \kappa\iota \operatorname{supp} \varphi$$

So two solids, $A_i, A_j$, *collide* at a configuration $q_i, q_j$ iff:

$$\text{collide}(q_i A_i, q_j A_j) \Leftrightarrow \iota(q_i A_i \cap q_j A_j) \neq \emptyset$$

But,

$$\iota(q_i A_i \cap q_j A_j) \neq \emptyset \Leftrightarrow \text{vol } q_i A_i \cap q_j A_j > 0$$

# Constraint Based Impacts

For any body $B_i$ with mass field $\varphi_i$, define the set:

$$A_i = \kappa\iota \text{ supp } \varphi$$

So two solids, $A_i, A_j$, *collide* at a configuration $q_i, q_j$ iff:

$$\text{collide}(q_i A_i, q_j A_j) \Leftrightarrow \iota(q_i A_i \cap q_j A_j) \neq \emptyset$$

But,

$$\iota(q_i A_i \cap q_j A_j) \neq \emptyset \Leftrightarrow \text{vol } q_i A_i \cap q_j A_j > 0$$

Define

$$C_{i,j}(q_i, q_j) \stackrel{def}{\equiv} \text{vol } q_i A_i \cap q_j A_j$$

And so we replace the impact forces with a system of differentiable holonomic inequality constraints:

$$C_{i,j} \leq 0$$

# Equations of motion revisited

New problem:

$$\text{minimize} \int_{t_0}^{t_1} L(q, \dot{q}, t)dt$$

$$\text{subject to } C_{i,j}(q_i, q_j) \leq 0 \quad \forall t \in [t_0, t_1), i \neq j$$

# Equations of motion revisited

New problem:

$$\text{minimize} \int_{t_0}^{t_1} L(q, \dot{q}, t)dt$$

$$\text{subject to } C_{i,j}(q_i, q_j) \leq 0 \quad \forall t \in [t_0, t_1), i \neq j$$

Apply KKT conditions + Euler-Lagrange to get complementarity problem:

$$\frac{d}{dt}\left(\frac{\partial T(\dot{q}_i)}{\partial \dot{q}_i}\right) - \frac{\partial U(q,t)}{\partial q_i} + \sum_{j \neq i} \mu_{i,j} \frac{\partial C_{i,j}(q_i, q_j)}{\partial q_i} = 0$$

$$0 \leq \mu_{i,j} \perp -C_{i,j} \geq 0$$

# Equations of motion revisited

New problem:

$$\text{minimize} \int_{t_0}^{t_1} L(q, \dot{q}, t)dt$$

$$\text{subject to } C_{i,j}(q_i, q_j) \leq 0 \ \ \forall t \in [t_0, t_1), i \neq j$$

Apply KKT conditions + Euler-Lagrange to get complementarity problem:

$$\frac{d}{dt}\left(\frac{\partial T(\dot{q}_i)}{\partial \dot{q}_i}\right) - \frac{\partial U(q,t)}{\partial q_i} + \sum_{j \neq i} \mu_{i,j} \frac{\partial C_{i,j}(q_i, q_j)}{\partial q_i} = 0$$

$$0 \leq \mu_{i,j} \perp -C_{i,j} \geq 0$$

Exactly elastic collision response!

Slack variables are impulse forces

# Calculating $C_{i,j}$

Need to compute:

$$C_{i,j}(q_i, q_j) = \text{vol } q_i A_i \cap q_j A_j$$

# Calculating $C_{i,j}$

Need to compute:

$$C_{i,j}(q_i, q_j) = \text{vol } q_i A_i \cap q_j A_j$$

Construct the *indicator* on $A$

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

# Calculating $C_{i,j}$

Need to compute:

$$C_{i,j}(q_i, q_j) = \text{vol } q_i A_i \cap q_j A_j$$

Construct the *indicator* on $A$

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

Observe:

$$\mathbf{1}_{A_i \cap A_j}(x) = \mathbf{1}_{A_i}(x)\mathbf{1}_{A_j}(x)$$

$$\text{vol } A_i = \int_{\mathbb{R}^d} \mathbf{1}_{A_i}(x)dx$$

# Calculating $C_{i,j}$

Need to compute:

$$C_{i,j}(q_i, q_j) = \text{vol } q_i A_i \cap q_j A_j$$

Construct the *indicator* on $A$

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

Observe:

$$\mathbf{1}_{A_i \cap A_j}(x) = \mathbf{1}_{A_i}(x)\mathbf{1}_{A_j}(x) \qquad \text{vol } A_i = \int_{\mathbb{R}^d} \mathbf{1}_{A_i}(x)dx$$

So:

$$C_{i,j}(q_i, q_j) = \int_{\mathbb{R}^d} \mathbf{1}_{A_i}(q_i^{-1}x)\mathbf{1}_{A_j}(q_j^{-1}x)dx$$

# Calculating $C_{i,j}$ (cont.)

$$C_{i,j}(q_i, q_j) = \int_{\mathbb{R}^d} \mathbf{1}_{A_i}(q_i^{-1}x)\mathbf{1}_{A_j}(q_j^{-1}x)dx$$

# Calculating $C_{i,j}$ (cont.)

$$C_{i,j}(q_i, q_j) = \int_{\mathbb{R}^d} \mathbf{1}_{A_i}(q_i^{-1}x)\mathbf{1}_{A_j}(q_j^{-1}x)dx$$

Push terms to one side:

$$\int_{\mathbb{R}^d} \mathbf{1}_{A_i}(x)\mathbf{1}_{A_j}(q_j^{-1}q_i x)dx$$

# Calculating $C_{i,j}$ (cont.)

$$C_{i,j}(q_i, q_j) = \int\limits_{\mathbb{R}^d} \mathbf{1}_{A_i}(q_i^{-1}x)\mathbf{1}_{A_j}(q_j^{-1}x)dx$$

Push terms to one side:

$$\int\limits_{\mathbb{R}^d} \mathbf{1}_{A_i}(x)\mathbf{1}_{A_j}(q_j^{-1}q_i x)dx$$

Substitute $q_j^{-1}q_i x \mapsto Rx - y$ and let $\widetilde{\mathbf{1}_{A_j}}(x) = \mathbf{1}_{A_j}(-x)$:

$$\int\limits_{\mathbb{R}^d} \mathbf{1}_{A_i}(x)\widetilde{\mathbf{1}_{A_j}}(y - Rx)dx = \int\limits_{\mathbb{R}^d} \mathbf{1}_{A_i}(R^{-1}x)\widetilde{\mathbf{1}_{A_j}}(y - x)dx$$

# Calculating $C_{i,j}$ (cont.)

$$C_{i,j}(q_i, q_j) = \int_{\mathbb{R}^d} \mathbf{1}_{A_i}(q_i^{-1}x)\mathbf{1}_{A_j}(q_j^{-1}x)dx$$

Push terms to one side:

$$\int_{\mathbb{R}^d} \mathbf{1}_{A_i}(x)\mathbf{1}_{A_j}(q_j^{-1}q_i x)dx$$

Substitute $q_j^{-1}q_i x \mapsto Rx - y$ and let $\widetilde{\mathbf{1}_{A_j}}(x) = \mathbf{1}_{A_j}(-x)$:

$$\int_{\mathbb{R}^d} \mathbf{1}_{A_i}(x)\widetilde{\mathbf{1}_{A_j}}(y - Rx)dx = \int_{\mathbb{R}^d} \mathbf{1}_{A_i}(R^{-1}x)\widetilde{\mathbf{1}_{A_j}}(y - x)dx$$

So,

$$C_{i,j}(q_i, q_j) = ((\mathbf{1}_{A_i} \circ R^{-1}) \star \widetilde{\mathbf{1}_{A_j}})(y)$$

# Fourier Methods

Convolution?

$$C_{i,j}(q_i, q_j) = ((\mathbf{1}_{A_i} \circ R^{-1}) \star \widetilde{\mathbf{1}_{A_j}})(y)$$

# Fourier Methods

Convolution? Take a Fourier transform!

$$C_{i,j}(q_i, q_j) = ((\mathbf{1}_{A_i} \circ R^{-1}) \star \widetilde{\mathbf{1}_{A_j}})(y) = \mathcal{F}^{-1}\left( (\widehat{\mathbf{1}_{A_i}} \circ R^{-1}) \overline{\widehat{\mathbf{1}_{A_j}}} \right)(y)$$

# Fourier Methods

Convolution? Take a Fourier transform!

$$C_{i,j}(q_i, q_j) = ((\mathbf{1}_{A_i} \circ R^{-1}) \star \widetilde{\mathbf{1}_{A_j}})(y) = \mathcal{F}^{-1}\left((\widehat{\mathbf{1}_{A_i}} \circ R^{-1})\overline{\widehat{\mathbf{1}_{A_j}}}\right)(y)$$

What about gradients? (ie $\frac{\partial C_{i,j}(q_i, q_j)}{\partial q_i}$)

# Fourier Methods

Convolution? Take a Fourier transform!

$$C_{i,j}(q_i, q_j) = ((\mathbf{1}_{A_i} \circ R^{-1}) \star \widetilde{\mathbf{1}_{A_j}})(y) = \mathcal{F}^{-1}\left((\widehat{\mathbf{1}_{A_i}} \circ R^{-1})\overline{\widehat{\mathbf{1}_{A_j}}}\right)(y)$$

What about gradients? (ie $\frac{\partial C_{i,j}(q_i,q_j)}{\partial q_i}$)

Fix parameters $q_i = (R_i, t_i)$,

$$q_i x = R_i x + t_i$$

$$q_i^{-1} = (R_i^{-1}, -R_i^{-1} t_i)$$

$$q_j q_i^{-1} x = Rx - y$$

# Fourier Methods

Convolution? Take a Fourier transform!

$$C_{i,j}(q_i, q_j) = ((\mathbf{1}_{A_i} \circ R^{-1}) \star \widetilde{\mathbf{1}_{A_j}})(y) = \mathcal{F}^{-1}\left((\widehat{\mathbf{1}_{A_i}} \circ R^{-1})\overline{\widehat{\mathbf{1}_{A_j}}}\right)(y)$$

What about gradients? (ie $\frac{\partial C_{i,j}(q_i,q_j)}{\partial q_i}$)

Fix parameters $q_i = (R_i, t_i)$,

$$q_i x = R_i x + t_i$$

$$q_i^{-1} = (R_i^{-1}, -R_i^{-1} t_i)$$

$$q_j q_i^{-1} x = Rx - y$$

Solve for $R, y$,

$$R = R_j R_i^{-1}$$

$$y = t_j - R_j R_i^{-1} t_i$$

# Fourier Methods

Convolution? Take a Fourier transform!

$$C_{i,j}(q_i, q_j) = ((\mathbf{1}_{A_i} \circ R^{-1}) \star \widetilde{\mathbf{1}_{A_j}})(y) = \mathcal{F}^{-1}\left((\widehat{\mathbf{1}_{A_i}} \circ R^{-1})\overline{\widehat{\mathbf{1}_{A_j}}}\right)(y)$$

What about gradients? (ie $\frac{\partial C_{i,j}(q_i, q_j)}{\partial q_i}$)

Fix parameters $q_i = (R_i, t_i)$,

$$q_i x = R_i x + t_i$$

$$q_i^{-1} = (R_i^{-1}, -R_i^{-1} t_i)$$

$$q_j q_i^{-1} x = Rx - y$$

Solve for $R, y$,

$$R = R_j R_i^{-1}$$

$$y = t_j - R_j R_i^{-1} t_i$$

Need to compute $\frac{\partial C_{i,j}(R_i, t_i, R_j, t_j)}{\partial R_i}$, $\frac{\partial C_{i,j}(R_i, t_i, R_j, t_j)}{\partial t_i}$

Or by symmetry: $C_{i,j}(q_i, q_j) = C_{j,i}(q_j, q_i)$

# Translational Gradient

Start with the translational case first.

Let $t_j^k$ denote the $k^{th}$ component of $t_j$,

# Translational Gradient

Start with the translational case first.

Let $t_j^k$ denote the $k^{th}$ component of $t_j$,then

$$\frac{\partial C_{i,j}}{\partial t_j^k} = \frac{\partial}{\partial t_i^j}\left(\int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i R_j^{-1}\omega)\overline{\widehat{\mathbf{1}_{A_j}}(\omega)}e^{2\pi i\left\langle \omega, t_j - R_j R_i^{-1} t_i\right\rangle} d\omega\right)$$

# Translational Gradient

Start with the translational case first.

Let $t_j^k$ denote the $k^{th}$ component of $t_j$,then

$$
\begin{aligned}
\frac{\partial C_{i,j}}{\partial t_j^k} &= \frac{\partial}{\partial t_i^j}\left( \int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i R_j^{-1}\omega)\overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i \left\langle \omega, t_j - R_j R_i^{-1} t_i \right\rangle} d\omega \right) \\
&= \int_{\mathbb{R}^d} 2\pi i \left\langle \omega, v^k \right\rangle \widehat{\mathbf{1}_{A_i}}(R_i R_j^{-1}\omega)\overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i \left\langle \omega, t_j - R_j R_i^{-1} t_i \right\rangle} d\omega
\end{aligned}
$$

Where $v^k$ denotes the $k^{th}$ basis vector

# Translational Gradient

Start with the translational case first.

Let $t_j^k$ denote the $k^{th}$ component of $t_j$,then

$$\begin{aligned}\frac{\partial C_{i,j}}{\partial t_j^k} &= \frac{\partial}{\partial t_i^j}\left(\int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i R_j^{-1}\omega)\overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i\left\langle \omega, t_j - R_j R_i^{-1} t_i\right\rangle} d\omega\right) \\ &= \int_{\mathbb{R}^d} 2\pi i \left\langle \omega, v^k \right\rangle \widehat{\mathbf{1}_{A_i}}(R_i R_j^{-1}\omega)\overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i\left\langle \omega, t_j - R_j R_i^{-1} t_i\right\rangle} d\omega\end{aligned}$$

Where $v^k$ denotes the $k^{th}$ basis vector

Conclusion: Translational gradient is just a multiplier

# Rotational Gradient

Parameterize $R_j = \exp(\mathfrak{r})$, where $\mathfrak{r} \in \mathfrak{so}(d)$

In otherwords $d \times d$ skew symmetric matrices, $\mathfrak{r}_{k,l} = -\mathfrak{r}_{l,k}$

# Rotational Gradient

Parameterize $R_j = \exp(\mathfrak{r})$, where $\mathfrak{r} \in \mathfrak{so}(d)$

In otherwords $d \times d$ skew symmetric matrices, $\mathfrak{r}_{k,l} = -\mathfrak{r}_{l,k}$

$$\frac{\partial C_{i,j}}{\partial \mathfrak{r}_{k,l}} = \frac{\partial}{\partial \mathfrak{r}_{k,l}} \left( \int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i \exp(-\mathfrak{r})\omega) \overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i \left\langle \omega, t_j - \exp(\mathfrak{r}) R_i^{-1} t_i \right\rangle} d\omega \right)$$

# Rotational Gradient

Parameterize $R_j = \exp(\mathfrak{r})$, where $\mathfrak{r} \in \mathfrak{so}(d)$

In otherwords $d \times d$ skew symmetric matrices, $\mathfrak{r}_{k,l} = -\mathfrak{r}_{l,k}$

$$
\begin{aligned}
\frac{\partial C_{i,j}}{\partial \mathfrak{r}_{k,l}} &= \frac{\partial}{\partial \mathfrak{r}_{k,l}} \left( \int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i \exp(-\mathfrak{r})\omega) \overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i \left\langle \omega, t_j - \exp(\mathfrak{r}) R_i^{-1} t_i \right\rangle} \, d\omega \right) \\
\ldots &= \int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(\omega) e^{2\pi i \langle \omega, y \rangle} \left( - \left\langle \overline{\nabla \widehat{\mathbf{1}_{A_j}}(R\omega)}, R \mathfrak{r}_{k,l} \omega \right\rangle \right. \\
&+ \left. \overline{\widehat{\mathbf{1}_{A_j}}(R\omega)} 2\pi i \left\langle R \mathfrak{r}_{k,l} \omega, t_j \right\rangle \right) d\omega
\end{aligned}
$$

# Rotational Gradient

Parameterize $R_j = \exp(\mathfrak{r})$, where $\mathfrak{r} \in \mathfrak{so}(d)$

In otherwords $d \times d$ skew symmetric matrices, $\mathfrak{r}_{k,l} = -\mathfrak{r}_{l,k}$

$$
\begin{aligned}
\frac{\partial C_{i,j}}{\partial \mathfrak{r}_{k,l}} &= \frac{\partial}{\partial \mathfrak{r}_{k,l}} \left( \int\limits_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i \exp(-\mathfrak{r})\omega) \overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i \left\langle \omega, t_j - \exp(\mathfrak{r}) R_i^{-1} t_i \right\rangle} d\omega \right) \\
\ldots &= \int\limits_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(\omega) e^{2\pi i \langle \omega, y \rangle} \left( - \left\langle \overline{\nabla \widehat{\mathbf{1}_{A_j}}(R\omega)}, R\mathfrak{r}_{k,l}\omega \right\rangle \right. \\
&+ \left. \overline{\widehat{\mathbf{1}_{A_j}}(R\omega)} 2\pi i \left\langle R\mathfrak{r}_{k,l}\omega, t_j \right\rangle \right) d\omega
\end{aligned}
$$

Get two terms:

## Rotational Gradient

Parameterize $R_j = \exp(\mathfrak{r})$, where $\mathfrak{r} \in \mathfrak{so}(d)$

In otherwords $d \times d$ skew symmetric matrices, $\mathfrak{r}_{k,l} = -\mathfrak{r}_{l,k}$

$$\frac{\partial C_{i,j}}{\partial \mathfrak{r}_{k,l}} = \frac{\partial}{\partial \mathfrak{r}_{k,l}} \left( \int\limits_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i \exp(-\mathfrak{r})\omega) \overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i \left\langle \omega, t_j - \exp(\mathfrak{r}) R_i^{-1} t_i \right\rangle} d\omega \right)$$

$$\ldots = \int\limits_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(\omega) e^{2\pi i \langle \omega, y \rangle} \Big( - \left\langle \overline{\nabla \widehat{\mathbf{1}_{A_j}}(R\omega)}, R \mathfrak{r}_{k,l} \omega \right\rangle$$

$$+ \overline{\widehat{\mathbf{1}_{A_j}}(R\omega)} 2\pi i \left\langle R \mathfrak{r}_{k,l} \omega, t_j \right\rangle \Big) d\omega$$

Get two terms: a multiplier (easy),

# Rotational Gradient

Parameterize $R_j = \exp(\mathfrak{r})$, where $\mathfrak{r} \in \mathfrak{so}(d)$

In otherwords $d \times d$ skew symmetric matrices, $\mathfrak{r}_{k,l} = -\mathfrak{r}_{l,k}$

$$\begin{aligned}
\frac{\partial C_{i,j}}{\partial \mathfrak{r}_{k,l}} &= \frac{\partial}{\partial \mathfrak{r}_{k,l}} \left( \int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i \exp(-\mathfrak{r})\omega) \overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i \left\langle \omega, t_j - \exp(\mathfrak{r}) R_i^{-1} t_i \right\rangle} d\omega \right) \\
\ldots &= \int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(\omega) e^{2\pi i \langle \omega, y \rangle} \left( - \left\langle \overline{\nabla \widehat{\mathbf{1}_{A_j}}(R\omega)}, R\mathfrak{r}_{k,l}\omega \right\rangle \right. \\
&+ \left. \overline{\widehat{\mathbf{1}_{A_j}}(R\omega)} 2\pi i \left\langle R\mathfrak{r}_{k,l}\omega, t_j \right\rangle \right) d\omega
\end{aligned}$$

Get two terms: a multiplier (easy), a gradient (can be precomputed).

# Rotational Gradient

Parameterize $R_j = \exp(\mathfrak{r})$, where $\mathfrak{r} \in \mathfrak{so}(d)$

In otherwords $d \times d$ skew symmetric matrices, $\mathfrak{r}_{k,l} = -\mathfrak{r}_{l,k}$

$$\begin{aligned}
\frac{\partial C_{i,j}}{\partial \mathfrak{r}_{k,l}} &= \frac{\partial}{\partial \mathfrak{r}_{k,l}} \left( \int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(R_i \exp(-\mathfrak{r})\omega) \overline{\widehat{\mathbf{1}_{A_j}}(\omega)} e^{2\pi i \left\langle \omega, t_j - \exp(\mathfrak{r}) R_i^{-1} t_i \right\rangle} d\omega \right) \\
\ldots &= \int_{\mathbb{R}^d} \widehat{\mathbf{1}_{A_i}}(\omega) e^{2\pi i \langle \omega, y \rangle} \left( - \left\langle \overline{\nabla \widehat{\mathbf{1}_{A_j}}(R\omega)}, R \mathfrak{r}_{k,l} \omega \right\rangle \right. \\
&+ \left. \overline{\widehat{\mathbf{1}_{A_j}}(R\omega)} 2\pi i \left\langle R \mathfrak{r}_{k,l} \omega, t_j \right\rangle \right) d\omega
\end{aligned}$$

Get two terms: a multiplier (easy), a gradient (can be precomputed).

Computationally not too bad, but still pretty messy in $d$-dimensional space.