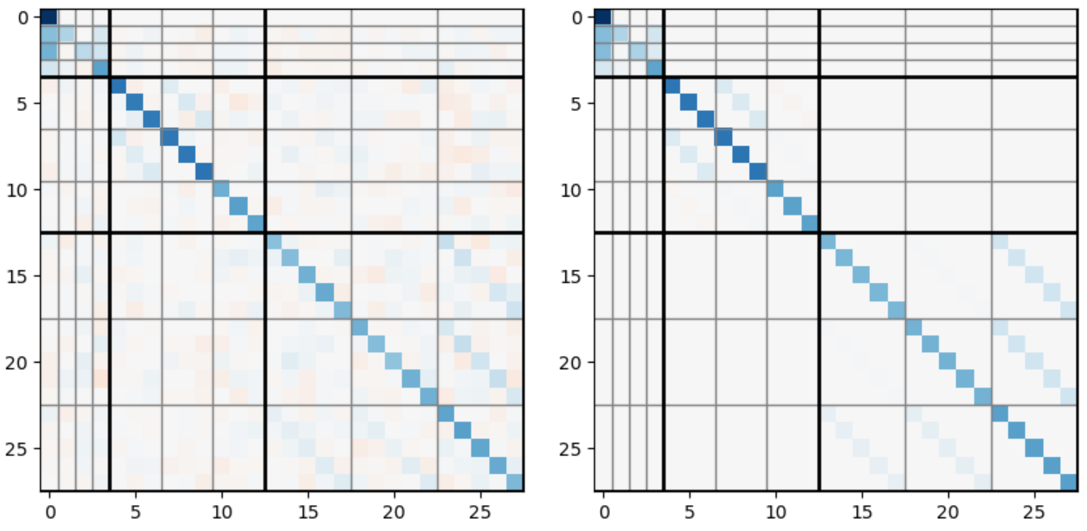
TICA and the VAMP-score in Situations with Symmetry

Phillip Bement

September 30, 2025



1 Introduction

Given sufficient training data about the (stochastic) dynamics of some physical system, TICA[1] (short for "time-lagged independent component analysis") and VAMP-nets[2] ("VAMP" is short for "variational approach to markov processes") provide a method to find slowly changing functions of the system coordinates. This is useful because the slowest-changing aspects of the system's state are the most useful for predicting what it will do far in the future, and often have scientific significance. (For example, some proteins have two alternate conformations and a function that indicates which one it occupies tends to be slowly-changing.)

[1]Molgedey, L. and Schuster, H. G., *Separation of a mixture of independent signals using time delayed correlations.* 10.1103/PhysRevLett.72.3634

[2]Hao Wu and Frank Noé, *Variational approach for learning Markov processes from time series data.* arxiv.org/abs/1707.04659

VAMP-nets learn such slowly-changing functions by optimizing a VAMP-score, while the simpler TICA technique finds slowly-changing linear combinations of a fixed set of basis functions. Both methods are based on the mathematical formalism of the Koopman operator, a linear operator on functions that captures the dynamics of a physical system.

This blog-post-in-a-pdf contains notes on how the Koopman operator interacts with symmetry, and how TICA and the VAMP-score can be generalized to physical systems that are symmetric in some way. We'll find that Koopman eigenfunctions are equivariant under the symmetry and by applying prior knowledge that our system is symmetric in some way, we can obtain estimates of covariances with less statistical noise.

1.1 The Koopman Operator

It's probably advisable for a reader who is new to the Koopman operator to read a true introduction to the subject before returning to this post[3]. But, to recap: The Koopman operator maps functions at one time to their expected values at a future time. In other words, the Koopman operator $\mathcal{K}$ for a given lag time Δt is a linear operator on functions ψ of the system coordinates $\mathbf{X}$ defined such that:

$$(\mathcal{K}\psi)(\mathbf{X}_t) = \mathbb{E}_{p(\mathbf{X}_{t+\Delta t}|\mathbf{X}_t)}\left[\psi(\mathbf{X}_{t+\Delta t})\right]$$

where p is the conditional probability distribution of later states conditional on earlier states implied by the dynamics of the system. For a system that obeys detailed-balance[4] with equilibrium distribution $q(\mathbf{X})$, the Koopman operator is self-adjoint under the inner product

$$\langle \phi, \psi \rangle = \mathbb{E}_{q(\mathbf{X})}\left[\phi(\mathbf{X})^* \psi(\mathbf{X})\right]$$

From this, we find that the operator has real eigenvalues and associated eigenfunctions. These eigenvalues and eigenfunctions characterize the dynamics of the system. Eigenvalues are ≤ 1 and larger eigenvalues are associated with aspects of the system with slower dynamics. There is always an eigenvalue of 1, which is associated with functions that are constant (i.e. scalings of $\psi(\mathbf{X}) = 1$).

The Koopman operator formalism also applies to systems that do not obey detailed balance, or even have an equilibrium distribution. But it's simpler if we focus here on system that do.

1.2 VAMP-nets and TICA

It's usually important that machine learning methods can fit in a computer, so we'd like to find finite-dimensional approximations of the Koopman operator. If we can only pick

[3]While we're talking about pre-reqs, you should also know basic representation theory. Just unitary irreps and Schur's lemma is fine, we won't need to talk about characters here.

[4]Detailed balance means that not only does the system have an equilibrium distribution q, but at equilibrium each flow of probability mass from state $\mathbf{X}$ to $\mathbf{X}'$ is balanced by an equal flow in the other direction. So p and q satisfy: $p(\mathbf{X}'|\mathbf{X})q(\mathbf{X}) = p(\mathbf{X}|\mathbf{X}')q(\mathbf{X}')$ for any two states $\mathbf{X}, \mathbf{X}'$.

a finite number N of basis functions, we want them to be the eigenfunctions with the N largest eigenvalues.

TICA is given a fixed set of basis functions, and finds an approximate Koopman operator on the associated subspace. How good an approximation this is depends heavily on how well the user chose the input basis functions. TICA will at least tell us what linear combinations are the slowest-changing, though.

VAMP-nets instead try to learn a set of basis functions, with the loss function provided by the VAMP-score. This is less dependent on choosing a good set of basis functions to start out with, but it does involve training a neural net.

1.3 Symmetry

A physical system having symmetry means that there is some group G describing that symmetry, and that this group acts on the system coordinates: i.e. we can write $g\mathbf{X}$ for some $g \in G$, and this will give a transformed system state. This is not sufficient, though. We also need the dynamics to respect the symmetry, i.e. for any $g \in G$:

$$p(g\mathbf{X}_{t+\Delta t}|g\mathbf{X}_t) = p(\mathbf{X}_{t+\Delta t}|\mathbf{X}_t)$$

This in turn implies that the equilibrium distribution q is also symmetric:

$$q(g\mathbf{X}) = q(\mathbf{X})$$

If a system has symmetry, we can immediately know some things about the spectrum of its Koopman operator.

1.4 Eigenspaces and Symmetry

For any $g \in G$, we can define a linear operator $\mathcal{D}_g$ that acts on functions of system coordinates and is equivalent to applying the symmetry g:

$$(\mathcal{D}_g\psi)(\mathbf{X}) = \psi(g\mathbf{X})$$

So $\mathcal{D}$ is a representation of G. A very reducible, infinite dimensional representation, but a representation nonetheless. This provides our first hint that representation theory will be important here. After all, if the entire space of functions transforms by a representation of G, then surely there are finite-dimensional subspaces that transform by irreps of G. And yes, this is true, at least in most naturally occurring cases.

How about dynamics? If the physics is symmetric under the action of G, then applying a symmetry g before or after time-evolution is equivalent, so we have:

$$\forall g \in G: \;\; \mathcal{D}_g\mathcal{K} = \mathcal{K}\mathcal{D}_g \qquad \text{i.e.} \qquad \forall g \in G: \;\; [\mathcal{D}_g, \mathcal{K}] = 0$$

so if ψ is an eigenstate of $\mathcal{K}$ with $\mathcal{K}\psi = \lambda\psi$, then

$$\mathcal{K}\mathcal{D}_g\psi = \mathcal{D}_g\mathcal{K}\psi = \lambda\mathcal{D}_g\psi$$

i.e. $\mathcal{D}_g\psi$ is also a member of the eigenspace associated with λ. So eigenspaces must transform by a representation of G. In quantum mechanics, we learn that energy eigenspaces also transform by representations of symmetries whose actions commute with the Hamiltonian $\mathcal{H}$. So this is kind-of the same. Anyway, because TICA and the VAMP-score are children of the Koopman operator, they inherit some nice additional properties from the fact that $[\mathcal{K}, \mathcal{D}_g] = 0$.

1.5 Plan for the rest of this post

This post will show how to generalize TICA and the VAMP-score to situations with some kind of symmetry. We'll get there in a series of steps. First, we'll provide analytic examples where we find Koopman eigenfunctions of a simple system, the 1d Ornstein-Uhlenbeck process. Then we'll do the same for the 3d Ornstein-Uhlenbeck process, which has $SO(3)$ symmetry. This will show us how the Koopman operator looks when the physical system it describes is symmetric, and demonstrate that non-trivial irreps are indeed important and there are slow-changing equivariant[5] functions.

We'll then prove some results about equilibrium and time-lagged covariances of equivariant functions.

Then it will be clear how TICA needs to be modified and from there it is a small further jump to the VAMP-score. We'll do a numeric demo of this symmetric TICA on a system that has $SO(3)$ symmetry, but is non-linear so that its set of Koopman eigenfunctions is not possible to find analytically. Finally, we'll mention some relevant symmetry groups besides $SO(3)$ that might be commonly useful.

2 Koopman Eigenfunctions of the Ornstein-Uhlenbeck Process

The Ornstein-Uhlenbeck process is one of the simplest possible stochastic processes. Sample trajectories are shown in figure 1. It describes a particle undergoing Brownian motion in a 1d harmonic potential $U(x) = \frac{1}{2}x^2$.

Alternately, with the appropriate non-dimensionalization, we can describe the fluctuation of charge on the capacitor in fig 2 as an Ornstein-Uhlenbeck process. In any case, the process is described by the following stochastic differential equation:

$$dx = -x dt + dW$$

2.1 Infinitesimal Generator

In this subsection we compute the infinitesimal generator for an Ornstein-Uhlenbeck process. You can skip the derivation and just take the final expression for the generator on faith, if you like.

If we define the operator $\mathcal{G}$ by:

[5] An equivariant function is a vector-valued function of system coordinates that transforms under a representation of the group G. i.e. if $\boldsymbol{\psi}(g\mathbf{X}) = D_g\boldsymbol{\psi}(\mathbf{x})$ for some representation D, then $\boldsymbol{\psi}$ is an equivariant function.

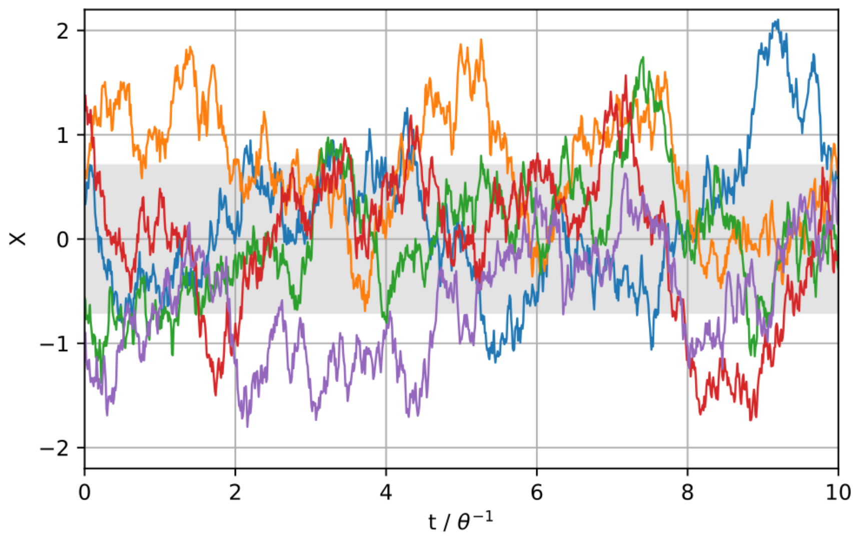


Figure 1: Some sample trajectories of the Ornstein-Uhlenbeck process.

Geek3, CC BY-SA 4.0, via Wikimedia Commons

$$(\mathcal{G}\psi)(x_t) = \lim_{\delta t \to 0} \int dx_{t+\delta t}\ p(x_{t+\delta t}|x_t)\ \frac{\psi(x_{t+\delta t}) - \psi(x_t)}{\delta t}$$

then for a lag time of Δt,

$$\mathcal{K} = \exp\left(\mathcal{G}\Delta t\right)$$

i.e. the operator $\mathcal{G}$ generates $\mathcal{K}$. In particular, this implies that the two operators have the same eigenvectors. So our next task is to find the eigenvectors of $\mathcal{G}$ for the Ornstein-Uhlenbeck process.

In the limit of small δt:

$$x_{t+\delta t} = x_t - x_t\ \delta t + \epsilon\sqrt{\delta t}$$

with $\epsilon \sim \mathcal{N}(0,1)$. Plugging $x_{t+\delta t}$ into some arbitrary function ψ gives:

$$(\mathcal{G}\psi)(x_t) = \int \frac{e^{-\epsilon^2/2}}{\sqrt{2\pi}}\ d\epsilon\ \ \frac{\psi\left(x_t - x_t\ \delta t + \epsilon\sqrt{\delta t}\right) - \psi(x_t)}{\delta t}$$

we can now use the 2nd order Taylor expansion of $g(x)$:

$$\psi(x + \delta x) = \psi(x) + \psi'(x)\delta x + \frac{1}{2}\psi''(x)\delta x^2$$

and plug this in to our expression above and only keep terms up to order δt:

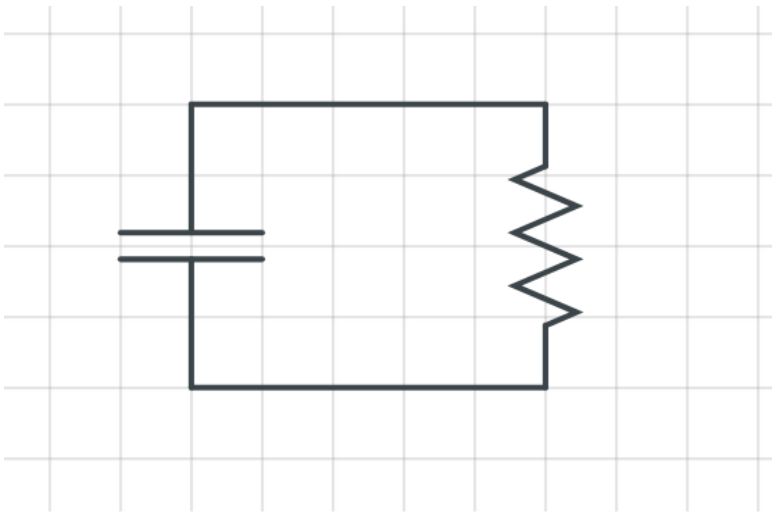


Figure 2: The Ornstein-Uhlenbeck process describes the fluctuation of charge over time on the capacitor in this circuit due to thermal fluctuations (Johnson-Nyquist noise).

$$(\mathcal{G}\psi)(x_t) = \int \frac{e^{-\epsilon^2/2}}{\sqrt{2\pi}}\, d\epsilon \;\; \frac{\psi'(x_t)\left(-x_t\ \delta t + \epsilon\sqrt{\delta t}\right) + \frac{1}{2}\psi''(x_t)\left(\epsilon^2\ \delta t\right)}{\delta t}$$

The $\epsilon\sqrt{\delta t}$ is odd, so it disappears and after doing the Gaussian integrals on the other two terms we're left with:

$$(\mathcal{G}\psi)(x) = -x\ \psi'(x) + \frac{1}{2}\psi''(x)$$

or, as an operator:

$$\mathcal{G} = -x\ \partial_x + \frac{1}{2}\partial_{xx}$$

2.2 Eigenfunctions

We'd like to solve the following eigenvalue problem:

$$\mathcal{G}\psi = \lambda\psi$$

$$-x\ \partial_x\psi(x) + \frac{1}{2}\partial_{xx}\psi(x) = \lambda\ \psi(x)$$

This equation can be solved by using the Hermite polynomials (see Appendix), which obey:

$$-x\partial_x\ \mathrm{He}_n(x) + \partial_{xx}\ \mathrm{He}_n(x) = -n\ \mathrm{He}_n(x)$$

for $n = 0, 1, 2, 3 \dots$. The first few Hermite polynomials are:

$$\begin{cases} \mathrm{He}_0(x) = 1 & \mathrm{He}_4(x) = x^4 - 6x^2 + 3 \\ \mathrm{He}_1(x) = x & \mathrm{He}_5(x) = x^5 - 10x^3 + 15x \\ \mathrm{He}_2(x) = x^2 - 1 & \mathrm{He}_6(x) = x^6 - 15x^4 + 45x^2 - 15 \\ \mathrm{He}_3(x) = x^3 - 3x & \dots \end{cases}$$

To solve the equation given above, we rescale x by $1/\sqrt{2}$. Then the eigenfunctions of $\mathcal{G}$ are given by:

$$\psi_n(x) = \mathrm{He}_n\left(\frac{x}{\sqrt{2}}\right)$$

with eigenvalue $-n$. As a result, $\mathcal{K}$ has the same eigenfunctions, with eigenvalues:

$$\mathcal{K}\psi_n = \exp(-n\Delta t)\psi_n$$

2.3 3d Ornstein-Uhlenbeck Process

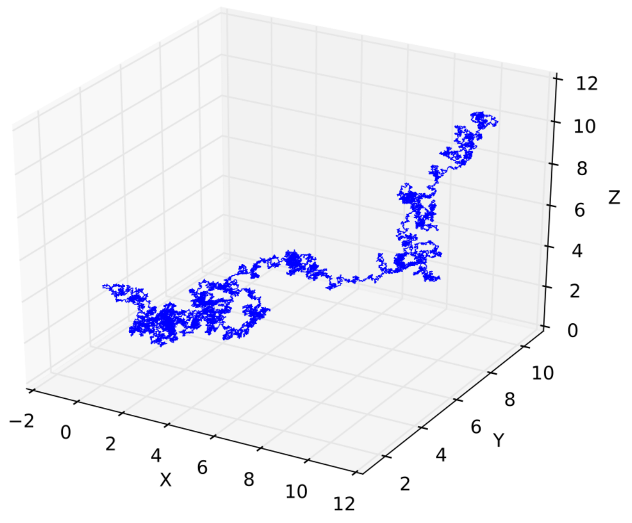


Figure 3: Scaled 3d Ornstein-Uhlenbeck process. The particle was initially started far away from the bottom of the potential well.

We can generalize the Ornstein-Uhlenbeck process to a 3d version by simply using a 3d rotationally symmetric harmonic potential $U(\mathbf{x}) = \frac{1}{2}\mathbf{x}^2$ instead of a 1d potential.

The equation of motion is:

$$d\mathbf{x} = -\mathbf{x}dt + d\mathbf{W}$$

By a similar derivation to above, the infinitesimal generator for this system is:

$$\mathcal{G} = -\mathbf{x}\cdot\nabla + \frac{1}{2}\nabla^2$$

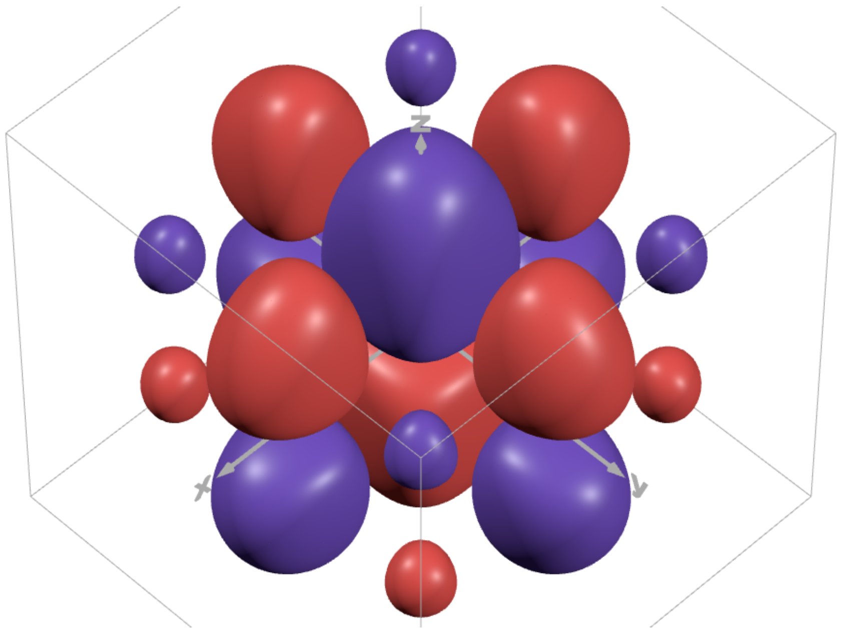


Figure 4: Plot of the level sets $\mathrm{He}_{[2,2,1]}(\mathbf{x})e^{-\frac{1}{2}\mathbf{x}^2} = \{0.1, -0.1\}$. The $[2,2,1]$ 3d-Hermite polynomial is multiplied by a $e^{-\frac{1}{2}\mathbf{x}^2}$ factor (proportional to the equilibrium distribution) to make the visualization clearer.

To find the eigenfunctions of this operator, we can use the 3d Hermite polynomials (see Appendix). They satisfy a 3d version of the Hermite differential equation:

$$-\mathbf{x}\cdot\nabla\mathrm{He}_{\mathbf{n}}(\mathbf{x}) + \nabla^2\mathrm{He}_{\mathbf{n}}(\mathbf{x}) = -(n_0+n_1+n_2)\mathrm{He}_{\mathbf{n}}(\mathbf{x})$$

Similar to before, the Koopman eigenfunctions are given by:

$$\psi_{\mathbf{n}}(\mathbf{x}) = \mathrm{He}_{\mathbf{n}}\left(\frac{\mathbf{x}}{\sqrt{2}}\right)$$

with eigenvalues:

$$\mathcal{K}\psi_{\mathbf{n}} = \exp(-(n_0+n_1+n_2)\Delta t)\psi_{\mathbf{n}}$$

So far, we have found the eigenvalues/eigenfunctions. In accordance with the previous discussion, the eigenspaces should correspond to equivariant functions, i.e. functions that transform under representations of $SO(3)$. So we should probably quickly check what the irreps of $SO(3)$ are.

2.4 Representation Theory of $SO(3)$

Skipping directly to the answer: The space of traceless fully symmetric tensors with l indices transforms according to a $2l+1$-dimensional irrep of $SO(3)$.

First, we should check that the dimensions of this vector space add up correctly. A tensor $T_{i_1 \dots i_l}$ has 3^n elements. If it is fully symmetric, then swapping any two indices leaves the tensor invariant. This fact reduces the number of truly independent elements. Thanks to reordering, the truly independent elements are indexed by multisets of size l chosen from $\{0, 1, 2\}$. The number of such multisets is given by:

$$\binom{l+2}{2} = \frac{(l+2)(l+1)}{2}$$

(This is a well-known fact in combinatorics. To save you from looking it up, you can prove it to yourself by letting strings containing l $\star$-symbols and 2 $|$-symbols represent multisets of size l (so for example $\star\star\star|\star|\star\star$ represents the multiset $\{0, 0, 0, 1, 2, 2\}$) and counting the permutations of these symbols.)

So now we know that the number of truly independent elements of a fully symmetric l-index tensor is $(l+2)(l+1)/2$. However, the tensor must also be traceless (the contraction of any two of its indices must yield 0). Since the tensor is fully symmetric, it doesn't matter which two indices we pick. We'll then need to subtract off the dimension of the trace of the tensor, or 0 if the tensor has less than 2 indices and thus no way to compute a trace. The trace of a fully symmetric tensor remains fully symmetric, and so we can reuse the previous formula for the dimension of a fully symmetric tensor with two fewer indices, and we get $l(l-1)/2$. So the dimension of the space of fully symmetric tensors is:

$$\frac{(l+2)(l+1)}{2} - \frac{l(l-1)}{2} = 2l + 1$$

as expected. For $l = 0, 1$, note that we end up subtracting 0, which is consistent with not being able to take a trace.

Proving that tensors transform under representations is simple: Given $R \in SO(3)$,

$$T_{i_1 \dots i_l} \to R_{i_1 j_1} \dots R_{i_l j_l} T_{j_1 \dots j_l}$$

This transformation is linear in T, therefore it constitutes a representation. It is also clear that if T is fully symmetric, then its transformed version will be too, so fully symmetric tensors are also representations. Finally, we note that a tensor trace transforms like a tensor does:

$$T_{kki_3 \dots i_l} \to R_{kj_1} R_{kj_2} R_{i_3 j_3} \dots R_{i_l j_l} T_{j_1 j_2 j_3 \dots j_l} = R_{i_3 j_3} \dots R_{i_l j_l} T_{kkj_3 \dots j_l}$$

This is why to get an irrep, our tensor has to be traceless: If it was not, its trace would be a sub-representation. Anyway, by linearity, if the trace of a tensor is 0, the trace of its transformed version will be too. Thus, fully-symmetric traceless tensors with l indices transform under representations of $SO(3)$.

The first four such representations are:

l	dim	tensor	constraints	name	irrep symbol
0	1	T		scalar	D^0
1	3	T_i		vector	D^1
2	5	T_{ij}	$T_{ii} = 0, \quad T_{ij} = T_{ji}$	symmetric traceless matrix	D^2
3	7	T_{ijk}	$T_{iij} = 0, \quad T_{ijk} = T_{jik} = T_{jki}$	–	D^3

where we use D^l to refer to the irrep itself.

Proving that these representations are now truly irreducible and that there are no remaining irreps besides the ones we can make from symmetric traceless tensors is not central to this post, so we leave it as an optional exercise.

2.5 3d Hermite Polynomial Irreps

Let $\mathbf{s} = [1, 1, 1]$ so that $\mathbf{s} \cdot \mathbf{n} = n_0 + n_1 + n_2$. Then all the 3d Hermite polynomials $\mathrm{He}_\mathbf{n}$ with a given value of $\mathbf{s} \cdot \mathbf{n}$ live in the same eigenspace. This eigenspace transforms according to a representation of $SO(3)$, so we should be able to break it down into its constituent irreps.

Define

$$H_{i_1 \cdots i_l}(\mathbf{x}) = \mathrm{He}_\mathbf{n}(\mathbf{x}) \qquad \text{where} \qquad \mathbf{n} = \begin{pmatrix} \sum_{a=1}^{l} [i_a = 0] \\ \sum_{a=1}^{l} [i_a = 1] \\ \sum_{a=1}^{l} [i_a = 2] \end{pmatrix}$$

($\mathbf{n}$ counts the number of indices that are each of 0, 1, 2.) This object (which we'll show soon is a tensor) is fully symmetric and has $l = \mathbf{s} \cdot \mathbf{n}$ indices. So each eigenspace is associated with a particular l-index tensor $H_{i_1 \ldots i_l}$.

If we recall the generating function for Hermite polynomials,

$$\sum_{n_0=0}^{\infty} \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \mathrm{He}_\mathbf{n}(\mathbf{x}) \frac{t_0^{n_0}}{n_0!} \frac{t_1^{n_1}}{n_1!} \frac{t_2^{n_2}}{n_2!} = e^{-\frac{1}{2}\mathbf{t}\cdot\mathbf{t}} e^{\mathbf{x}\cdot\mathbf{t}}$$

then we have:

$$H_{i_1 \cdots i_l}(\mathbf{x}) = \frac{\partial}{\partial t_{i_1}} \cdots \frac{\partial}{\partial t_{i_l}} e^{-\frac{1}{2}\mathbf{t}\cdot\mathbf{t}} e^{\mathbf{x}\cdot\mathbf{t}} \bigg|_{\mathbf{t}=\mathbf{0}}$$

Now let $R \in SO(3)$ and we'll see what happens when we rotate space:

$$H_{i_1 \cdots i_l}(R\mathbf{x}) = \frac{\partial}{\partial t_{i_1}} \cdots \frac{\partial}{\partial t_{i_l}} e^{-\frac{1}{2}\mathbf{t}\cdot\mathbf{t}} e^{(R\mathbf{x})\cdot\mathbf{t}} \bigg|_{\mathbf{t}=\mathbf{0}}$$

$$= \frac{\partial}{\partial t_{i_1}} \dots \frac{\partial}{\partial t_{i_l}} e^{-\frac{1}{2}(R\mathbf{t}')\cdot(R\mathbf{t}')} e^{(R\mathbf{x})\cdot(R\mathbf{t}')} \Big|_{\mathbf{t}'=\mathbf{0}} \qquad \text{where} \qquad \mathbf{t} = R\mathbf{t}'$$

$$= \frac{\partial}{\partial t_{i_1}} \dots \frac{\partial}{\partial t_{i_l}} e^{-\frac{1}{2}\mathbf{t}'\cdot\mathbf{t}'} e^{\mathbf{x}\cdot\mathbf{t}'} \Big|_{\mathbf{t}'=\mathbf{0}}$$

$\frac{\partial}{\partial t_i} = R_{ij} \frac{\partial}{\partial t'_j}$, so:

$$H_{i_1 \cdots i_l}(R\mathbf{x}) = R_{i_1 j_1} \frac{\partial}{\partial t'_{j_1}} \dots R_{i_l j_l} \frac{\partial}{\partial t'_{j_l}} e^{-\frac{1}{2}\mathbf{t}'\cdot\mathbf{t}'} e^{\mathbf{x}\cdot\mathbf{t}'} \Big|_{\mathbf{t}'=\mathbf{0}}$$

$$= R_{i_1 j_1} \dots R_{i_l j_l} H_{j_1 \dots j_l}(\mathbf{x})$$

From this, we see that H is a tensor, and therefore transforms under a representation of $SO(3)$. While H is fully symmetric, it is not traceless in general, and thus the representation it transforms under might not be an irrep. So what is the representation associated with a fully symmetric and not traceless tensor? Well, it's the direct sum of the representation of the trace and the traceless part:

$$\text{rep}(l) = \begin{cases} D^l & \text{if } l < 2 \\ D^l \oplus \text{rep}(l-2) & \text{otherwise} \end{cases}$$

We expect that these should add up to $(l+2)(l+1)/2$ dimensions, the size of a fully symmetric tensor and also the number of choices of $\mathbf{n}$ that satisfy $\mathbf{s}\cdot\mathbf{n} = l$. Here are the first few representations of the eigenspaces:

$\mathbf{s}\cdot\mathbf{n}$	representation	dimension	
0	D^0	1	$= 1$
1	D^1	3	$= 3$
2	$D^2 \oplus D^0$	$5 + 1$	$= 6$
3	$D^3 \oplus D^1$	$7 + 3$	$= 10$
4	$D^4 \oplus D^2 \oplus D^0$	$9 + 5 + 1$	$= 15$
5	$D^5 \oplus D^3 \oplus D^1$	$11 + 7 + 3$	$= 21$

So far then, we've confirmed that eigenspaces do indeed correspond to representations of $SO(3)$ in this system. These representations can be further broken down into irreps. One interesting thing to note is that because the Koopman operator is self-adjoint, its eigenfunctions are orthogonal. In particular, this means that:

$$\langle \phi, \psi \rangle = 0 = \mathbb{E}_q \left[\phi^* \psi \right]$$

for eigenfunctions ϕ, ψ with different eigenvalues. In the next section, we'll look further into covariances of equivariant functions.

3 Symmetry and Covariances

We'll just be looking at unitary representations here, i.e. representations D where $D_{g^{-1}} = D_g^\dagger$. This is the nicest class of representations, and is in some sense fully sufficient for describing finite groups. (And even for many infinite groups, eg $SO(3)$, in practice we don't need to use any non-unitary representations they may have.)

Let $\boldsymbol{\phi}, \boldsymbol{\psi}$ be some (vector-valued) functions that transform under the irreps D^ϕ, D^ψ of our symmetry group G. We've specified that these functions are irrep-equivariant, but nothing else about them. They need not be eigenfunctions. We'll consider two cases: First, the case where $D^\phi \neq D^\psi$, and then the case where $D^\phi = D^\psi$.

3.1 When $D^\phi \neq D^\psi$

Firstly, let's clarify that by saying the irreps are not equal, we mean that they are not even isomorphic. They are truly different representations of G. Let A be some (possibly rectangular) matrix so that we can take the product $\boldsymbol{\phi}^\dagger A \boldsymbol{\psi}$ and get a scalar. We'll compute the expected value:

$$\langle \boldsymbol{\phi}, A\boldsymbol{\psi} \rangle = \mathbb{E}_q \left[\boldsymbol{\phi}^\dagger A \boldsymbol{\psi} \right] = \int q(\mathbf{X}) d\mathbf{X} \; \boldsymbol{\phi}^\dagger(\mathbf{X}) A \boldsymbol{\psi}(\mathbf{X})$$

Now because q is symmetric under the action of G, we can average over group elements (replace this with a Haar-measure integral if the group is continuous) to get:

$$= \int q(\mathbf{X}) d\mathbf{X} \; \frac{1}{|G|} \sum_{g \in G} \; \boldsymbol{\phi}^\dagger(g\mathbf{X}) \; A \; \boldsymbol{\psi}(g\mathbf{X})$$

now we can apply the fact that the functions are representations:

$$= \int q(\mathbf{X}) d\mathbf{X} \; \frac{1}{|G|} \sum_{g \in G} \; \boldsymbol{\phi}^\dagger(\mathbf{X}) D_g^{\phi\,\dagger} \; A \; D_g^\psi \boldsymbol{\psi}(\mathbf{X})$$

$$= \int q(\mathbf{X}) d\mathbf{X} \; \boldsymbol{\phi}^\dagger(\mathbf{X}) \left(\frac{1}{|G|} \sum_{g \in G} D_g^{\phi\,\dagger} \; A \; D_g^\psi \right) \boldsymbol{\psi}(\mathbf{X})$$

By Schur's lemma, this inner sum over group elements is 0, and thus, so is the whole integral.

$$\mathbb{E}_q \left[\boldsymbol{\phi}^\dagger A \boldsymbol{\psi} \right] = 0$$

So the expected value of this sandwich is 0, regardless of the value of A. Now we'll take an outer product of the two functions to get a covariance matrix. (Computing a covariance usually requires us to subtract the mean first, but here we don't. TICA has its own way to deal with functions that have non-zero mean, which will be described later. We still write "covariance" instead of "expected outer product" because it's less clunky.) We can figure out what each element should be by making A all zeros except for a single element that is set to 1.

$$\mathbb{E}_q\left[\boldsymbol{\psi}\boldsymbol{\phi}^\dagger\right] = 0$$

So we find no covariance between functions that transform under different irreps. (This doesn't necessarily mean they are independent, though!) Note that we didn't need the functions to be eigenfunctions for this to be true, just the fact that the irreps are different is enough.

3.2 When $D^\phi = D^\psi$

Firstly, let's clarify that by saying the irreps are equal, we mean that they are precisely equal, not just isomorphic. This will simplify the following derivation. Once again, let A be any matrix that can be sandwiched between the irreps. This time, we know that A must be square. We'll compute:

$$\langle \boldsymbol{\phi}, A\boldsymbol{\psi}\rangle = \mathbb{E}_q\left[\boldsymbol{\phi}^\dagger A\boldsymbol{\psi}\right] = \int q(\mathbf{X})d\mathbf{X}\ \boldsymbol{\phi}^\dagger(\mathbf{X})A\boldsymbol{\psi}(\mathbf{X})$$

as before, we average over group elements:

$$= \int q(\mathbf{X})d\mathbf{X}\ \frac{1}{|G|}\sum_{g\in G}\ \boldsymbol{\phi}^\dagger(g\mathbf{X})\ A\ \boldsymbol{\psi}(g\mathbf{X})$$

and apply the representations:

$$= \int q(\mathbf{X})d\mathbf{X}\ \frac{1}{|G|}\sum_{g\in G}\ \boldsymbol{\phi}^\dagger(\mathbf{X})D_g^{\phi\ \dagger}\ A\ D_g^\psi\boldsymbol{\psi}(\mathbf{X})$$

$$= \int q(\mathbf{X})d\mathbf{X}\ \boldsymbol{\phi}^\dagger(\mathbf{X})\left(\frac{1}{|G|}\sum_{g\in G} D_g^{\phi\ \dagger}\ A\ D_g^\psi\right)\boldsymbol{\psi}(\mathbf{X})$$

Now, the fact that the representations D^ϕ, D^ψ are actually exactly the same means that Schur's lemma tells us that:

$$\frac{1}{|G|}\sum_{g\in G} D_g^{\phi\ \dagger}\ A\ D_g^\psi = I\frac{\operatorname{Tr} A}{\operatorname{Tr} I} = I\frac{\operatorname{Tr} A}{\dim D^\phi}$$

so we get:

$$\mathbb{E}_q\left[\boldsymbol{\phi}^\dagger A\boldsymbol{\psi}\right] = \frac{\operatorname{Tr} A}{\dim D^\phi}\int q(\mathbf{X})d\mathbf{X}\ \boldsymbol{\phi}^\dagger(\mathbf{X})\boldsymbol{\psi}(\mathbf{X})$$

If we let $A = I$, then we get:

$$\mathbb{E}_q\left[\boldsymbol{\phi}^\dagger\boldsymbol{\psi}\right] = \int q(\mathbf{X})d\mathbf{X}\ \boldsymbol{\phi}^\dagger(\mathbf{X})\boldsymbol{\psi}(\mathbf{X}) = C_{\phi\psi}$$

which is as expected. We assign this value the short name $C_{\phi\psi}$.

Now we'll take an outer product of the two functions to get a covariance matrix. We can figure out what each element should be by making A all zeros except for a single element that is set to 1. If the non-zero element is on the diagonal we get trace 1, otherwise trace 0. Thus, the resulting covariance matrix is:

$$\mathbb{E}_q\left[\boldsymbol{\psi}\boldsymbol{\phi}^\dagger\right] = I\frac{1}{\dim D^\phi}\int q(\mathbf{X})d\mathbf{X}\ \boldsymbol{\phi}^\dagger(\mathbf{X})\boldsymbol{\psi}(\mathbf{X}) = I\frac{C_{\phi\psi}}{\dim D^\phi}$$

i.e. it is just a scaled copy of the identity matrix.

3.3 Symmetry and $\mathcal{K}$

Now we'll bring dynamics into the preceding discussion. Let's try applying the Koopman operator $\mathcal{K}$ to some function $\boldsymbol{\psi}(\mathbf{X})$ that transforms under an irrep D^ψ. Before applying any symmetry, we have:

$$\mathcal{K}\boldsymbol{\psi}(\mathbf{X}_t) = \int p(\mathbf{X}_{t+\Delta t}|\mathbf{X}_t)\ d\mathbf{X}_{t+\Delta t}\ \ \boldsymbol{\psi}(\mathbf{X}_{t+\Delta t})$$

and after applying a symmetry g, we have:

$$\mathcal{K}\boldsymbol{\psi}(g\mathbf{X}_t) = \int p(g\mathbf{X}_{t+\Delta t}|g\mathbf{X}_t)\ d\mathbf{X}_{t+\Delta t}\ \ \boldsymbol{\psi}(g\mathbf{X}_{t+\Delta t})$$

By the fact the the dynamics is unchanged under the symmetry group G, we get:

$$\mathcal{K}\boldsymbol{\psi}(g\mathbf{X}_t) = \int p(\mathbf{X}_{t+\Delta t}|\mathbf{X}_t)\ d\mathbf{X}_{t+\Delta t}\ \ \boldsymbol{\psi}(g\mathbf{X}_{t+\Delta t})$$

Now using the representation D^ψ, we get:

$$\mathcal{K}D_g^\psi\boldsymbol{\psi}(\mathbf{X}) = \int p(\mathbf{X}_{t+\Delta t}|\mathbf{X}_t)\ d\mathbf{X}_{t+\Delta t}\ \ D_g^\psi\boldsymbol{\psi}(\mathbf{X}_{t+\Delta t}) = D_g^\psi\mathcal{K}\boldsymbol{\psi}(\mathbf{X})$$

This is a little different from our finding that $\mathcal{D}_g$ commutes with $\mathcal{K}$. $\mathcal{D}_g$ acts on functions, while D_g^ψ is a simple matrix that only acts on finite-dimensional vectors. Essentially what it means is that $\mathcal{K}\boldsymbol{\psi}$ still transforms under the same irrep D^ψ that $\boldsymbol{\psi}$ originally transformed under. We've already seen how to find the covariance between two irrep-transforming functions $\boldsymbol{\phi}, \boldsymbol{\psi}$. But thanks to the above fact, applying the Koopman operator to one of these functions results in an outcome that is basically the same!

$$\mathbb{E}_q\left[\mathcal{K}\boldsymbol{\psi}\ \boldsymbol{\phi}^\dagger\right] = \begin{cases} 0 & \text{if } D^\phi \neq D^\psi \\ I\frac{C_{\phi,\mathcal{K}\psi}}{\dim D^\phi} & \text{if } D^\phi = D^\psi \end{cases}$$

4 Symmetric TICA

4.1 How TICA Originally Worked

Recall that the procedure for TICA was to restrict our attention to some preselected set of basis functions $\{\psi_j\}$. Then, we empirically estimate both lagged and non-lagged covariances. The non-lagged covariances estimate is:

$$C_{ij} \approx \langle \psi_i, \psi_j \rangle$$

while the lagged covariances estimate is:

$$C_{ij}^{\Delta t} \approx \langle \psi_i, \mathcal{K}\psi_j \rangle$$

From these two covariance matrices, it's possible to compute a matrix K that is in some sense the best approximation of the Koopman operator available on the given set of basis functions. In particular, we perform a multivariable linear regression to find the K that minimizes

$$\mathbb{E}_{p(\mathbf{X}_{t+\Delta t}|\mathbf{X}_t)q(\mathbf{X}_t)} \left[\left(\boldsymbol{\psi}(\mathbf{X}_{t+\Delta t}) - K\boldsymbol{\psi}(\mathbf{X}_t) \right)^2 \right]$$

and the resulting K can be written in terms of the covariance matrices:

$$K = \left(C^{-1} C_{\Delta t} \right)^\top$$

The "independent component analysis" part of this is that we do a spectral decomposition on K. Here we have to be a little careful. The Koopman operator was self-adjoint under a very particular inner product, an expectation under q. The spectral decomposition is inner product dependent, and the usual computer routines assume the regular dot product. So we have to perform the decomposition on K, but with an inner product given by $\langle \boldsymbol{\phi}, \boldsymbol{\psi} \rangle = \phi^\dagger C \psi$. Computationally, this is equivalent to decomposing the "de-correlated" version of K, called K^{dc}, with the usual dot product:

$$K^{\text{dc}} = C^{-1/2}\, C_{\Delta t}^\top\, C^{-1/2} = USU^\top$$

Using a de-correlated (sometimes called whitened) basis means that even applying a linear transformation to our basis functions leaves the singular values produced by the analysis invariant. And the basis functions of the de-correlated basis are orthonormal under the equilibrium-distribution inner product.

4.2 Symmetric TICA

The modifications needed to perform TICA where there's a symmetry are actually pretty small. They are:

- For each irrep, we have a set of basis functions that transform under that irrep, which we'll call an "irrep block".

- Covariances between functions from different irrep blocks are identically 0, so we might as well only measure covariances between functions in the same block. We have an entirely separate covariance matrix for each irrep block.
- For functions from the same irrep block, covariances just look like a scaled identity matrix. We can take advantage of this by computing $C_{\phi\psi}$ where $\boldsymbol{\phi}, \boldsymbol{\psi}$ are functions that transform under the same irrep. Then we know the covariance matrix between them is given by $IC_{\phi\psi}/\dim D^{\phi}$. Really, we just need to store the $C_{\phi\psi}$, as this compresses all the relevant covariance information that it's possible for a symmetric system to have.
- Since the overall covariance matrix is block-diagonal in terms of the irrep blocks, we can compute products like $\left(C^{-1}C_{\Delta t}\right)^{\top}$ and $C^{-1/2}\, C_{\Delta t}^{\top}\, C^{-1/2}$ separately for each irrep block. We can also compute matrix decompositions separately for each irrep block. When we get eigenvalues out of TICA, they have degeneracy equal to the dimension of the irrep whose block they came from. So just like the Koopman operator itself, TICA will spit out duplicate eigenvalues when symmetry is present.
- When inverting and multiplying to compute K or K^{dc} as described in the previous step, we can just treat everything that looks like $IC_{\phi\psi}/\dim D^{\phi}$ as equalling just the number $C_{\phi\psi}$. This makes our computations less redundant and removes opportunity for numerical errors. We can ignore the I because it just acts like a 1, and add it back at the end. We can even ignore the $\dim D^{\phi}$, since the factor of $1/\dim D^{\phi}$ from $C_{\Delta t}$ and the factor of $\dim D^{\phi}$ from C^{-1} cancel out.

That might all sound pretty abstract, so we'll look at a concrete example.

4.3 Symmetric TICA: Example

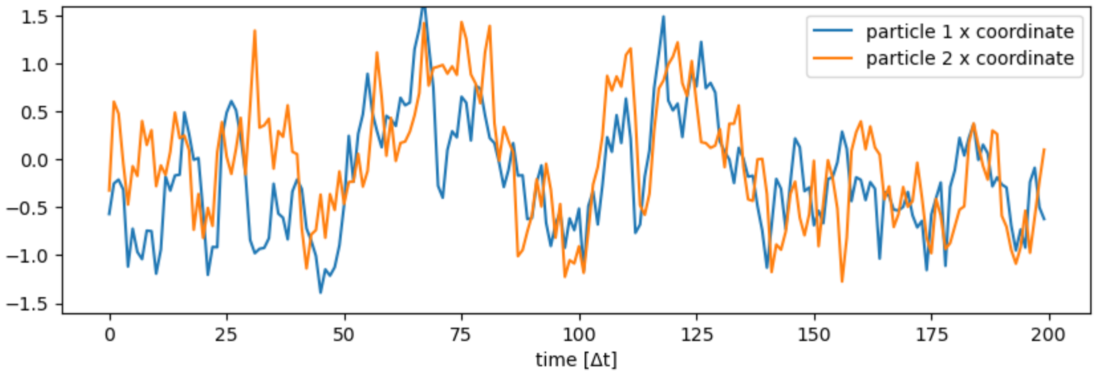


Figure 5: An example trajectory for our dimer system. Only the projection of the 3d position of each particle to the x-axis is shown.

As our example, we'll take a system that obeys detailed balance and has $SO(3)$ symmetry. It has two particles, connected by a harmonic spring, and they occupy a quartic sombrero potential. The dynamics is Brownian (so it obeys detailed balance) and the overall potential is:

$$U(\mathbf{r}_1,\mathbf{r}_2) = \frac{1}{4}\left(\mathbf{r}_1^2 - 1\right)^2 + \frac{1}{4}\left(\mathbf{r}_2^2 - 1\right)^2 + \frac{1}{2}\left(\mathbf{r}_1 - \mathbf{r}_2\right)^2$$

We generate a bunch of trajectory data with a lag time of $\Delta t = 0.2$ and this can be used in our symmetric TICA analysis. An example of what this data looks like is shown in figure 5. To ensure we get a symmetric time-lagged covariance matrix, we duplicate our data set so that both the forwards and backwards version of each pair $(\mathbf{X}_t, \mathbf{X}_{t+\Delta t})$ is included. (That step should be skipped if we're not totally sure the system obeys detailed balance.)

TICA needs a set of basis functions, and symmetric TICA needs these functions to be irreps. We'll take our basis to be polynomials in the system coordinates up to degree 2. These can certainly be divided into irreps, the ones we'll need are D^0, D^1, D^2. The following table lists them, categorized by irrep block:

irrep block	polynomial functions	dim
D^0	$1,\ \mathbf{r}_1^2,\ \mathbf{r}_2^2,\ \mathbf{r}_1 \cdot \mathbf{r}_2$	$4 \times 1 = 4$
D^1	$\mathbf{r}_1,\ \mathbf{r}_2,\ \mathbf{r}_1 \times \mathbf{r}_2$	$3 \times 3 = 9$
D^2	$\text{sst}(\mathbf{r}_1 \otimes \mathbf{r}_1),\ \text{sst}(\mathbf{r}_1 \otimes \mathbf{r}_2),\ \text{sst}(\mathbf{r}_2 \otimes \mathbf{r}_2)$	$3 \times 5 = 15$

where sst means turning a matrix into an D^2 irrep by symmetrizing it and subtracting the trace. We can check that the number of dimensions matches. The system has 6 coordinates, so there is 1 degree-0 polynomial, 6 degree-1 polynomials, and 21 degree-2 polynomials in these coordinates. This totals 28, which matches the 4+9+15 dimensions we see in the table.

The trajectory data is fed into these functions to produce a time-series. Then we can compute stationary and lagged covariances to get corresponding covariance matrices. From these, we compute K, K^{dc}. See figure 6, where we can see that the estimate of K^{dc} has less statistical noise when we use symmetric TICA. The image at the start of this document shows the same thing for K.

Figure 7 shows K^{dc} in the non-redundant representation that we actually use to do computations in symmetric TICA. There are separate K^{dc} matrices for each irrep block. Also note that the D^1 irrep block's matrix, for example, only has entries for each pair of the 3 vector basis functions, not for each pair of the 9 elements of the 3 vector basis functions.

Now the final step in TICA is finding spectral decomposition of K^{dc}.[6] This produces a bunch of eigenvalues. The D^0 irrep block should have 4 eigenvalues[7] of multiplicity 1, the D^1 irrep block should have 3 eigenvalues of multiplicity 3, and the D^2 irrep block should have 3 eigenvalues of multiplicity 5. In symmetric TICA, we compute eigenvalues for each of the non-redundant K^{dc} matrices shown in figure 7, and then just set the multiplicity equal to the dimension of its irrep.

[6]The matrix K^{dc} should be symmetric in principle (we symmetrized our data so that $C_{\Delta t}$ would be symmetric). Due to numerical errors, it's not exactly symmetric in the computer's memory, so we use a singular-value-decomposition routine and call the resulting singular values our "eigenvalues".

[7]one of which is 1, associated with the constant eigenfunction

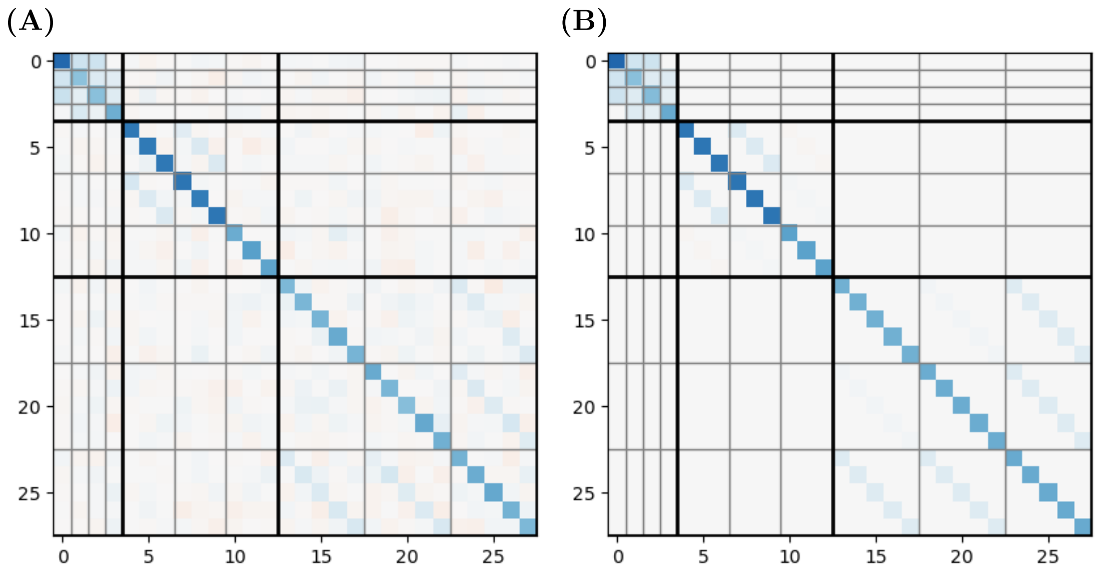


Figure 6: (A) shows K^{dc} where we compute covariances of basis functions without any consideration for symmetry. (B) shows the K^{dc} produced by symmetric TICA. In both matrices, the thick lines separate irrep blocks, while the thin lines separate the individual equivariant functions in each block.

Figure 8 shows the results of this, for both raw TICA, and symmetric TICA. The trivial 1 eigenvalue is the first, followed by a multiplicity-3 eigenvalue, associated with a D^1 equivariant eigenfunction. We can see a matching group of 3 eigenvalues from the raw TICA analysis, though the degeneracy is slightly split due to statistical errors. The next highest eigenvalue produced by symmetric TICA is of multiplicity 5 (arising from some D^2 equivariant function), and so on. There are no more clear groups of eigenvalues produced by regular TICA, perhaps because the spacing has become too fine. However, the general trends of the spectra match. From symmetric TICA, we get a less-noisy, symmetry-respecting spectrum.

One other interesting thing to note is that while the top eigenvalue of 1 is always from the trivial representation, in this case the next 3 distinct eigenvalues are from the D^1, D^2, D^1 representations, in that order, and only the next distinct eigenvalue after that is from the D^0 representation again. This author has observed that some works tend to handle symmetry by only choosing invariant basis functions for TICA. However, this example shows that sometimes the slowest-changing function is actually an equivariant function. We also saw this with the 3d Ornstein-Uhlenbeck process, where the slowest non-trivial mode (H_i with $l = \mathbf{s} \cdot \mathbf{n} = 1$) was also D^1 equivariant.

4.4 Mean-Subtraction and the $\psi_1(\mathbf{X}) = 1$ Eigenfunction

One thing to note here is that we should always include the constant $\psi_1(\mathbf{X}) = 1$ function in our basis. It is always a Koopman eigenfunction, regardless of the physical system.

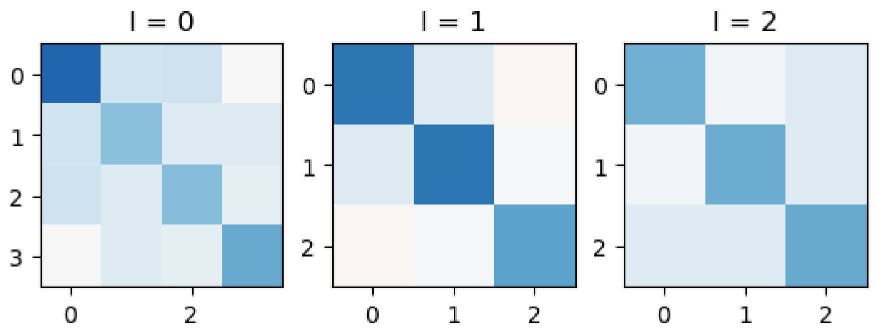


Figure 7: The data contained in symmetric TICA's K^{dc} matrix, shown in figure 6 (B), can be compressed into a separate matrix for each irrep block. The dimension of each of these is equal to the number of equivariant functions in its irrep block. It doesn't matter how high-dimensional the irrep itself is.

The matrix K produced by the linear regression step of TICA will have a $[1, 0, 0, 0 \ldots]$ row that it uses to estimate $\mathbb{E}\left[\psi_1(\mathbf{X}_{t+\Delta t})\right]$. For any other basis function ϕ, we have $\langle \psi_1, \phi \rangle = \mathbb{E}_q\left[\phi\right]$, the mean value of ϕ. If some other functions have non-zero mean, this is corrected for by including ψ_1 in the basis.

Some other formulations of TICA instead just subtract the mean from all basis functions before performing their analysis. We feel it is often nicer to explicitly include it as one of the basis functions, however. For example, it's the easiest way to do symmetric TICA.

ψ_1 transforms under the trivial irrep (the one-dimensional irrep where $D_g = 1$ for all g; it exists for any possible group G). This tells us that we must put ψ_1 into the trivial irrep block. So all the invariant functions (i.e. the ones that are trivial-irrep-equivariant) will have their mean corrected for. And then because of the rule that there are no correlations between different irrep blocks, any nontrivial-irrep-equivariant functions will not have their mean corrected for. This is fine, however, because group theory tells us that such functions should have a mean of exactly 0 anyway.

4.5 Symmetric VAMP-score

Now computing a symmetric VAMP-score is not much harder to do than symmetric TICA. The first step here is to perform symmetric TICA as usual on the functions we're trying to score. So we run our trajectories through the functions and compute (symmetry-aware) covariances as usual. There are several different VAMP scores, but eg. the VAMP-2 score can be computed as:

$$\mathrm{VAMP}_2 = \mathrm{Tr}\left[K^{\mathrm{dc}\,\top} K^{\mathrm{dc}}\right]$$

Just like regular TICA, symmetric TICA's computation of K^{dc} is differentiable, and so there is no problem propagating gradients backwards through this score. The functions

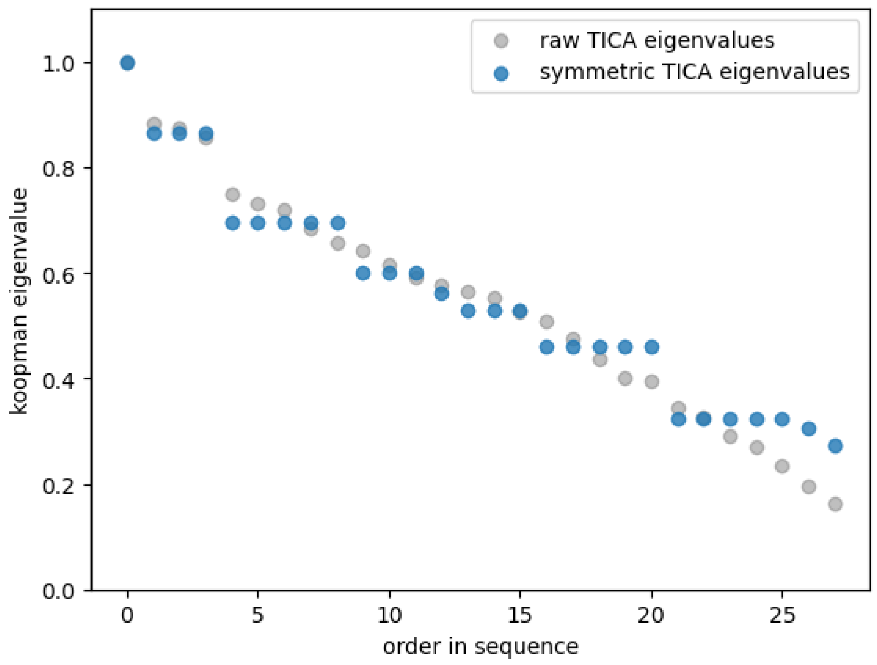


Figure 8: The spectrum (with eigenvalues simply put into sorted order) produced by symmetric TICA, overlaid onto the spectrum produced by regular TICA.

used as input must be equivariant, and so if training a VAMP-net, one must construct an equivariant neural network to supply them. The python library `e3nn`[8] provides a framework for this for 3d symmetry groups like $SO(3)$ or $O(3)$.

4.6 Other Useful Symmetry Groups

$SO(2)$ aka $U(1)$ aka Periodic Translation Group

This group is abelian so all irreps are 1d. We can write elements of this group as rotations through an angle of θ. There's an irrep D^k for even integer $k \in \mathbb{Z}$, given by:

$$D^k_\theta = e^{ik\theta}$$

The trivial irrep is D^0.

Lattice Translations

If we have an n dimensional crystal lattice whose points are given by $A\mathbb{Z}^n$, then the group of lattice translations is also abelian and so has 1d irreps. Let $\mathbf{k}$ be a wavevector in $\mathbb{R}^n$. Then the irreps are given by:

[8]Mario Geiger and Tess Smidt, *e3nn: Euclidean Neural Networks*. arxiv.org/abs/2207.09453

$$D^{\mathbf{k}}_{\mathbf{a}} = e^{i\mathbf{k}\cdot\mathbf{a}}$$

where $\mathbf{a}$ is a lattice translation vector. There is some redundancy in the definition of $\mathbf{k}$ (which can be removed by specifying that it is contained within the Brillouin zone), where we say that $\mathbf{k}$ and $\mathbf{k}'$ give equivalent representations iff $\frac{1}{2\pi}A^\top(\mathbf{k}-\mathbf{k}') \in \mathbb{Z}^n$. The trivial irrep is $D^{\mathbf{0}}$.

Cyclic Groups $\mathbb{Z}_n$

The cyclic group $\mathbb{Z}_n$ is finite and abelian, and it has a finite set of irreps. Let k be an integer. Then the irreps are given by:

$$D^k_m = e^{2\pi i\ km/n}$$

where m is a group element written as an integer $\mod n$. The is some redundancy in the definition of k, where we say that k and k' given equivalent representations iff $k = k' \mod n$. The trivial irrep is D^0.

The Group $SU(2)$

$SU(2)$ double-covers $SO(3)$, and so the irreps for this group include the irreps for $SO(3)$, except there are more of them: We are now allowed half-integer l along with integer l. If $U \in SU(2)$, then:

$$D^{1/2}_U = U$$

and we can build up larger irreps by taking tensor products of this with itself and decomposing the resulting representations into their irreducible parts.

Direct Products of Multiple Groups

What about if we have periodic boundary conditions on some 3d system? The symmetry group for this can be written as $U(1)\times U(1)\times U(1)$. How do we get the irreps? We use the rule for direct products of irreps: If D^G is an irrep of G and D^H is an irrep of H then $D^G \otimes D^H$ is an irrep of $G \times H$, and all irreps of $G\times H$ can be written in this form. To make sure the notation is clear: If $\mathbf{u}$ is D^G-equivariant and $\mathbf{v}$ is D^H-equivariant, and $g \in G$ and $h \in H$, then:

$$\left(D^G \otimes D^H\right)_{(g,h)}(\mathbf{u}\otimes\mathbf{v}) = (D^G_g\mathbf{u}) \otimes (D^H_h\mathbf{v})$$

This applies regardless of whether G, H are abelian or not. The trivial irrep is the tensor product of the trivial irreps of G and H.

Code Link:

github.com/pb1729/symmetric-tica

Apprendix: The Hermite Polynomials

The 1d Hermite Polynomials

The 1d Hermite polynomials, denoted $\mathrm{He}_n(x)$ for $n \in \mathbb{N} = \{0, 1, 2, 3 \ldots\}$ can be defined using a generating function as follows:

$$\sum_{n=0}^{\infty} \mathrm{He}_n(x) \frac{t^n}{n!} = e^{-\frac{1}{2}t^2} e^{xt}$$

The Hermite polynomials have many interesting properties, but the one that is relevant for our purposes here is that they satisfy the Hermite differential equation, namely:

$$\left(-x\frac{d}{dx} + \frac{d^2}{dx^2}\right) \mathrm{He}_n(x) = -n\mathrm{He}_n(x)$$

We can show this using the generating function definition as follows:

$$\sum_{n=0}^{\infty} \left(-x\frac{\partial}{\partial x} + \frac{\partial^2}{\partial x^2}\right) \mathrm{He}_n(x) \frac{t^n}{n!} = \left(-x\frac{\partial}{\partial x} + \frac{\partial^2}{\partial x^2}\right) e^{-t^2/2} e^{xt}$$

$$= \left(-xt + t^2\right) e^{-t^2/2} e^{xt} = -t\frac{\partial}{\partial t} e^{-t^2/2} e^{xt}$$

$$= -\sum_{n=0}^{\infty} \mathrm{He}_n(x) t \frac{\partial}{\partial t} \frac{t^n}{n!} = -\sum_{n=0}^{\infty} \mathrm{He}_n(x) n \frac{t^n}{n!}$$

Therefore:

$$\left(-x\frac{\partial}{\partial x} + \frac{\partial^2}{\partial x^2}\right) \mathrm{He}_n(x) = -n\mathrm{He}_n(x)$$

as expected.

The 3d Hermite Polynomials

The Hermite polynomials can be generalized to a 3d version using the following generating function:

$$\sum_{n_0=0}^{\infty} \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \mathrm{He}_{\mathbf{n}}(\mathbf{x}) \frac{t_0^{n_0}}{n_0!} \frac{t_1^{n_1}}{n_1!} \frac{t_2^{n_2}}{n_2!} = e^{-\frac{1}{2}\mathbf{t}\cdot\mathbf{t}} e^{\mathbf{x}\cdot\mathbf{t}}$$

where $[n_0, n_1, n_2] = \mathbf{n} \in \mathbb{N}^3$. The resulting polynomials are just products of any 3 chosen 1d Hermite polynomials. By a similar argument to above, we can check that a given 3d Hermite polynomial satisfies a 3d version of the Hermite differential equation:

$$\sum_{n_0=0}^{\infty} \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \left(-\mathbf{x} \cdot \nabla_{\mathbf{x}} + \nabla_{\mathbf{x}}^2\right) \mathrm{He}_{\mathbf{n}}(\mathbf{x}) \frac{t_0^{n_0}}{n_0!} \frac{t_1^{n_1}}{n_1!} \frac{t_2^{n_2}}{n_2!} = \left(-\mathbf{x} \cdot \nabla_{\mathbf{x}} + \nabla_{\mathbf{x}}^2\right) e^{-\frac{1}{2}\mathbf{t}\cdot\mathbf{t}} e^{\mathbf{x}\cdot\mathbf{t}}$$

$$= \left(-\mathbf{x}\cdot\mathbf{t} + \mathbf{t}\cdot\mathbf{t}\right) e^{-\frac{1}{2}\mathbf{t}\cdot\mathbf{t}} e^{\mathbf{x}\cdot\mathbf{t}} = -\mathbf{t}\cdot\nabla_{\mathbf{t}} e^{-\frac{1}{2}\mathbf{t}\cdot\mathbf{t}} e^{\mathbf{x}\cdot\mathbf{t}}$$

$$= -\sum_{n_0=0}^{\infty}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} \mathrm{He}_{\mathbf{n}}(\mathbf{x}) \left(\mathbf{t}\cdot\nabla_{\mathbf{t}}\right) \frac{t_0^{n_0}}{n_0!}\frac{t_1^{n_1}}{n_1!}\frac{t_2^{n_2}}{n_2!}$$

$$= -\sum_{n_0=0}^{\infty}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} \mathrm{He}_{\mathbf{n}}(\mathbf{x}) \left(n_0 + n_1 + n_2\right) \frac{t_0^{n_0}}{n_0!}\frac{t_1^{n_1}}{n_1!}\frac{t_2^{n_2}}{n_2!}$$

Therefore:

$$\left(-\mathbf{x}\cdot\nabla_{\mathbf{x}} + \nabla_{\mathbf{x}}^2\right) \mathrm{He}_{\mathbf{n}}(\mathbf{x}) = -\left(n_0 + n_1 + n_2\right) \mathrm{He}_{\mathbf{n}}(\mathbf{x})$$

as expected.