Stoquastic Hamiltonians, a Leisurely Introduction

The motivation for stoquastic Hamiltonians comes from the success of quantum monte carlo methods.

There are many variants of quantum monte carlo but all of them have a similar character, which I will outline here.

We often want to evaluate expressions like this:

$$\frac{Tr[Oe^{-\beta H}]}{Tr[e^{-\beta H}]}$$

But accessing $e^{-\beta H}$ can be nasty.
We can break up the exponentials into small steps

$$Tr[Oe^{-\beta H}] = Tr[O(e^{-\frac{\beta}{L}H})^L]$$

and insert identities.

$$= Tr\left[O\left(\sum_{x_1}|x_1\rangle\langle x_1|\right)e^{-\frac{\beta}{L}H}\left(\sum_{x_2}|x_2\rangle\langle x_2|\right)e^{-\frac{\beta}{L}H}\left(\sum_{x_3}|x_3\rangle\langle x_3|\right)..\right]$$

$$= Tr\left[O\left(\sum_{x_1}|x_1\rangle\langle x_1|\right)e^{-\frac{\beta}{L}H}\left(\sum_{x_2}|x_2\rangle\langle x_2|\right)e^{-\frac{\beta}{L}H}\left(\sum_{x_3}|x_3\rangle\langle x_3|\right)..\right]$$

If L is large enough, and H is efficiently expressible, then $e^{-\frac{\beta}{L}H} = (\mathbb{I} - \frac{\beta}{L}H) = G$ is easier to handle.

We can factor out the sums, to transform the expression into a sum over "paths" $\vec{x} = (x_0, x_1, x_2..)$

$$Tr[Oe^{-\beta H}] = \sum_{\vec{x}|x_0=x_L}\langle x_0|O|x_1\rangle\prod_{i=1}^{L-1}\langle x_i|G|x_{i+1}\rangle$$

$$Tr[Oe^{-\beta H}] = \sum_{x} O(x)w(x)$$

We don't want to sum over these paths

$$Tr[Oe^{-\beta H}] = \sum_x O(x)w(x)$$

The strategy of quantum Monte Carlo is to sample from these paths in a way that is faithful to this weighting.

$$Tr[Oe^{-\beta H}] = \sum_{\vec{x}|x_0=x_L} \langle x_0|O|x_1\rangle \prod_{i=1}^{L-1} \langle x_i|G|x_{i+1}\rangle$$

For example [Sorella, Capriotti (2013)], one might have a walker (w, x), and one might start at a random x and perform a random walk informed by the matrix elements $\langle x|G|y\rangle = G_{xy}$:

$$x \to y \text{ w/ prob. } \frac{|G_{xy}|}{\sum_y |G_{xy}|}$$

$$w \to w * \text{sign}(G_{xy}) * \left(\sum_y |G_{xy}|\right)$$

Sampling in this way will reproduce the distribution, but there is a problem.

$$\langle O \rangle = \frac{\sum_x O(x) w(x)}{\sum_x w(x)}$$

$$\langle O \rangle = \frac{\sum_x O(x)\ sign(w(x))\ |w(x)|}{\sum_x\ sign(w(x))\ |w(x)|} \left(\frac{\sum_x |w(x)|}{\sum_x |w(x)|} \right)$$

$$\text{Define: } P(x) := \frac{|w(x)|}{\sum_x |w(x)|}\ ,\ \delta(x) := sign(w(x))$$

$$\langle O \rangle = \frac{\sum_x O(x)\ \delta(x)\ P(x)}{\sum_x \delta(x)\ P(x)} = \frac{\langle O\ \delta \rangle}{\langle \delta \rangle}$$

If there are negative signs in G, then for long path lengths the average sign will tend to zero, and relative errors can blow up.

How do we avoid this "sign problem"?

If H is real and has non-positive entries in its off-diagonals then for some sufficiently large L, $G = \mathbb{I} - \frac{\beta}{L}H$ is entrywise non-negative and real,

so all path weights are positive and real. $\langle \delta \rangle = 1$

We call such H globally stoquastic (in the standard basis).

In fact, $e^{-\beta H}$ is an entrywise non-negative matrix for all β if and only if H is globally stoquastic.

— If H is stoquastic, G is non-negative for large L, therefore $e^{-\beta H} = G^L$ is non-negative.

— If $e^{-\beta H}$ is non-negative for all β, then choose sufficiently small β: $e^{-\beta H} = \mathbb{I} - \beta H + O(\beta^2 ||H||^2)$ and so H must have non-positive off-diagonals.

If $e^{-\beta H}$ is positive and real, then the Perron-Frobenius theorem tells us that the ground state of H is a vector with all positive and real weights.

$$\langle\psi|H|\psi\rangle = \sum_x |\psi(x)|^2\langle x|H|x\rangle + \sum_{x\neq y}\psi(x)\psi^*(y)\langle x|H|y\rangle$$

$$\langle\psi|H|\psi\rangle = \sum_x |\psi(x)|^2\langle x|H|x\rangle - \sum_{x\neq y}\psi(x)\psi^*(y)|\langle x|H|y\rangle|$$

$$\langle|\psi||H||\psi|\rangle = \sum_x |\psi(x)|^2\langle x|H|x\rangle - \sum_{x\neq y}|\psi(x)||\psi^*(y)||\langle x|H|y\rangle|$$

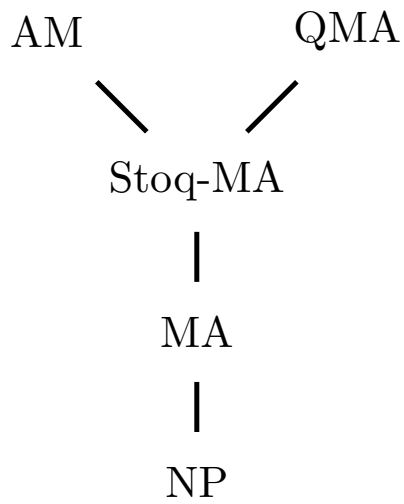
$$\langle|\psi||H||\psi|\rangle \leq \langle\psi|H|\psi\rangle$$

(Thanks to Alex for that proof)
We can think of the ground state as the stationary probability distribution of a quantum monte carlo process

Some comments on terminology:

- Globally stoquastic in standard basis: $\langle x|H|y\rangle \leq 0 \;\; x \neq y$
- (termwise) Stoquastic in standard basis: $H = \sum_k H_k \;\; : \langle x|H_k|y\rangle \leq 0 \;\; x \neq y$
- Globally stoquastic $\neq$ termwise stoquastic in general
- For 2-local multi-qubit Hamiltonians they are the same. (but not for 3-local)
- Computer scientists seem to care about termwise stoquastic
- Monte Carlo community it is not so clear to me. Seems like it might depend on the method.

Stoquastic Hamiltonians form a distinct complexity class called Stoq-MA.



The decision problem is the ground energy of a stoquastic Hamiltonian. The transverse field Ising model is complete for this class.

Stoquastic k-sat ($H = \sum H_a$, there exists a $|\psi\rangle$ such that $H_a|\psi\rangle = 0$) is MA-complete

The idea behind our research program is that stoquasticity is basis dependent. So for which Hamiltonians can we find a basis that makes them stoquastic?

Deciding Stoquasticity of 2-Local Hamiltonians

Joel Klassen

2018

Hello my name is Joel.

I am a postdoc at QuTech

Working with Barbara Terhal

Thanks for hosting me.

Introduction

- This is joint work largely done by myself and Milad Marvian. Our collaborators are Barbara Terhal, Marios Iannous, Itay Hen and Daniel Lidar.
- Our research has focused on stoquastic Hamiltonians
- In particular I have been trying to develop algorithms for deciding when a Hamiltonian is stoquastic
- I will explain what a stoquastic Hamiltonian is
- I will give some motivation for why it is interesting
- I will present a polynomial time algorithm for deciding if a two-local multiqubit Hamiltonain, with no one-local terms, can be made stoquastic in some local basis.

Yeah but like, what even is a stoquastic Hamiltonian anyways?

Did you say stochastic?

- A stochastic process is a random process evolving in time.
- A quantum process is not a stochastic process in general.
- Indeed unitary time evolution is deterministic.
- However some quantum systems can be modelled by stochastic processes.

$$\textbf{Stochastic} + \textbf{Quantum} = \textbf{Stoquastic}\ ^{1}$$

[1] [Bravyi et.al. (2006)]

We often want to evaluate expressions like this:

$$Tr[Oe^{-\beta H}]$$

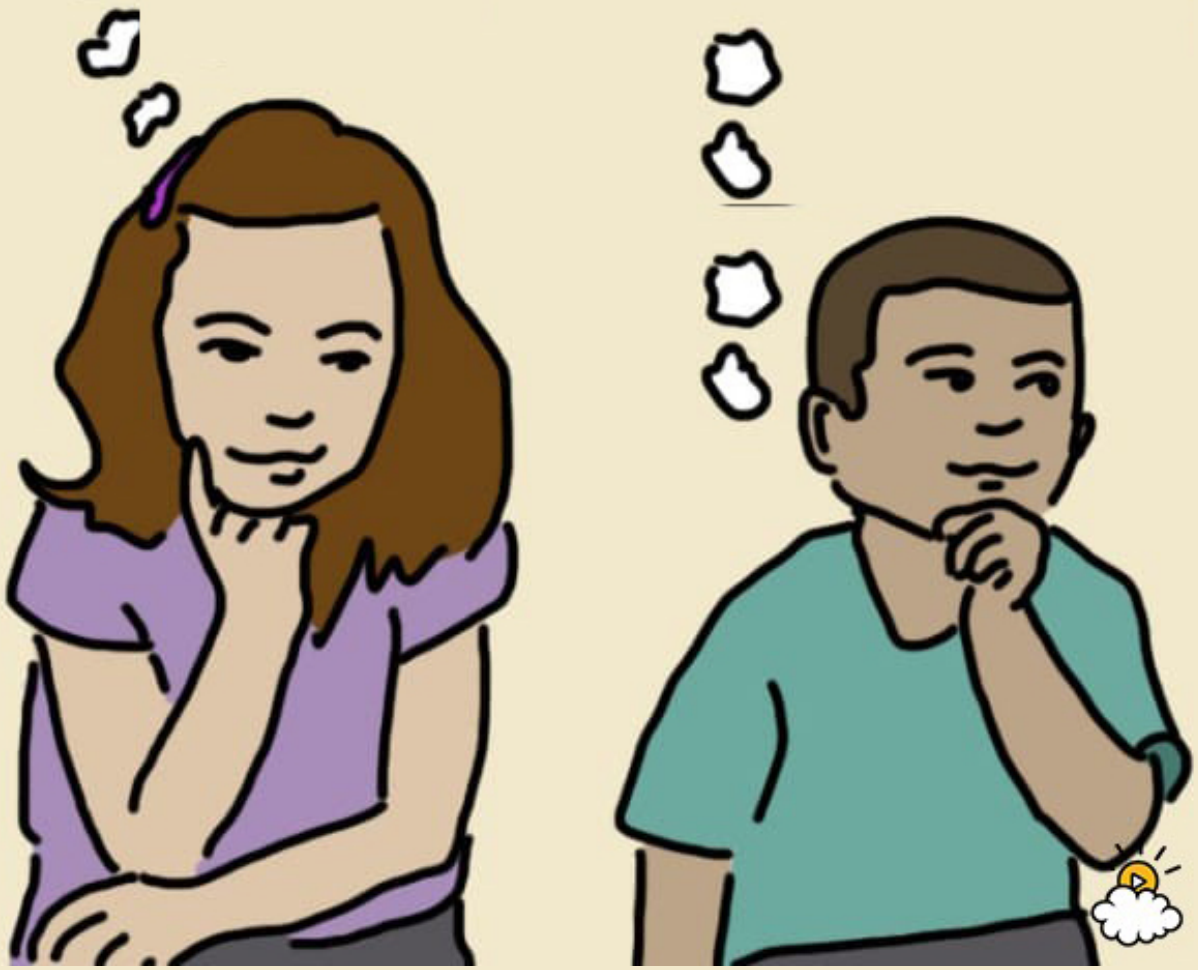
One way is to break up the exponentials into small steps, and insert identities.

$$Tr[Oe^{-\beta H}] = \sum_{p|p_0=p_L} \langle p_0|O|p_1\rangle \prod_{i=1}^{L-1} \langle p_i|\mathbb{I} - \frac{\beta}{L} H|p_{i+1}\rangle$$

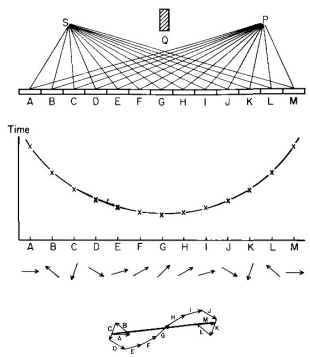
Terms can be evaluated if O and H are local in the basis p_i. Morally, this is path integration, with p representing a particular path.

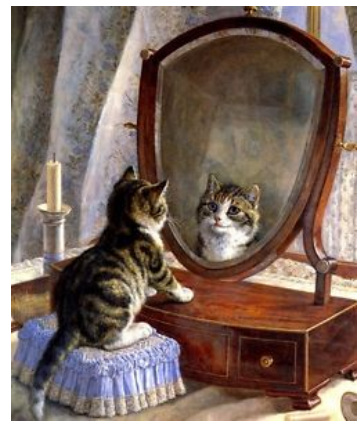
$$\sum \langle \text{end}|O|\text{beginning}\rangle \times \text{ amplitudes of paths}$$

Boy, it sure would be nice if we didn't have to evaluate all of those paths...



- What if we just sample from these paths according to their weights? Will our answers be faithful?
- Not if our amplitudes interfere! Random sampling can obscure important coherence effects.
- This is called the sign problem.
- Its very much like the difference between burnished metal and a polished mirror.

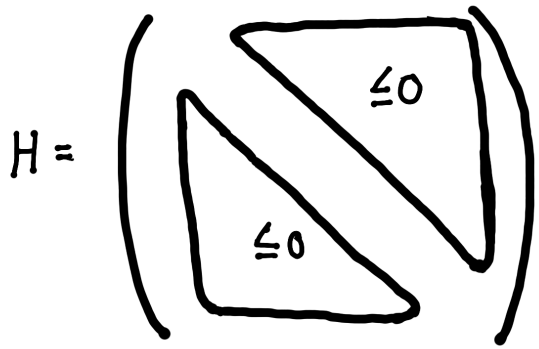




- However if all of our amplitudes are positive and real... then we don't have this problem.

Enter Stoquastic Hamiltonians

Consider a Hamiltonian H such that all of its off diagonal elements are non-positive and real in some basis $\{|i\rangle\}$.



- For all values of $\beta \geq 0$: $\langle i|e^{-\beta H}|j\rangle \geq 0$
- **Path amplitudes will be positive and real**, and we can perform stochastic sampling of our path integrals.
- Such a matrix H is an instance of a "Z-matrix"
- Matrices of this type are also employed in the study of economics, control theory, and population dynamics.

A Z-matrix in any other basis would smell as sweet.

- Critically, a Z-matrix is basis dependent!
- Generally one wants to say that when a Hamiltonian can be efficiently transformed into a Z-matrix while preserving sparsity (ie local structure), then it is "stoquastic" (Quantum Stochastic) under that transformation.

Stoquastic $\simeq$ Z-matrix in some efficient representation

- There is a subtlety here. We can ask that each k-local term be a Z-matrix, or we can ask that the whole Hamiltonian be a Z-matrix
- These two questions are distinct! But for two-local qubit Hamiltonians they are the same.

Okay but I mean who cares?

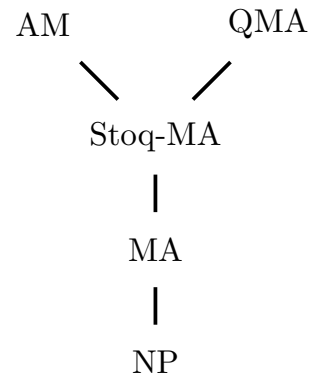
Motivation

The Quantum Monte Carlo Community

Stoquastic Hamiltonians avoid the sign-problem and thus are more amenable to quantum Monte Carlo methods.

Computational Complexity Theorists

Stoquastic Hamiltonians constitutes a distinct and interesting computational complexity class: Stoq-MA [Bravyi et.al. (2006)(2008)] [Aharanov, Grillo (2019)]



Adiabatic evolution of frustration free stoquastic Hamiltonians can be simulated efficiently. [Bravyi, Terhal (2008)]

Motivation

Experimentalists and Engineers

- It seems as though finding ground states and ground energies of stoquastic Hamiltonians is easier than for generic Hamiltonians.
- Perhaps in adiabatic quantum computation we want to build devices that are not stoquastic. (eg. TFIM is stoquastic)

Theoretical Physicists

- Many natural systems are manifestly stoquastic in what is considered a natural basis. (spinless destinguishable particles, hopping bosons)
- Is there something deep behind this?

General Problem Statement & Prior Work

Stoquasticity of 2-local multiqubit Hamiltonians

When is a 2-local Hamiltonian acting on n qubits stoquastic in some local basis, and what is that basis?

Lots of work in the QMC community on avoiding the sign problem. Few systematic approaches in this stoquastic picture.

Some limited strategies for choosing right basis:

- in certain regimes of the antiferromagnetic XXZ model on a triangular lattice [Hatano, Suzuki 1992]
- in certain regimes of generalized XYZ Heisenberg Hamiltonian on bipartite lattice, transverse field Ising model and single-ion anisotropy model (transformations are called Marshall-Peierls sign rules in this context) [Bishop, Farnell 2001]
- Any Hamiltonian which is tridiagonal in some local basis is stoquastic [Bausch, Crosson 2018]

Prior Complexity Results

- Deciding if 3-local Hamiltonian is stoquastic under one-local Cliffords is NP-complete
- Deciding if 6-local Hamiltonian is stoquastic under one-local orthogonal transformations is NP-complete [Marvian, Lidar, Hen 2018]
- There is an efficient algorithm for deciding if a generalized XYZ-Hamiltonian ($\sum aXX + bYY + cZZ$) is stoquastic under one-local unitaries. [Klassen, Terhal 2018]
- **The XYZ-algorithm** $\leftarrow$ Will use this later

Watch our talk on youtube! Search: QIP19 Marvian Klassen

Can we more clearly delineate the boundary between hardness and easiness?

Can we efficiently decide beyond the XYZ algorithm?

So we can decide if an XYZ-Hamiltonian is stoquastic in some basis...

"But I want a Hamiltonian that has an XY term in it!"

Main Result: Going Beyond the XYZ Hamiltonian

Exactly 2-local Hamiltonian on n qubits:

$$H = \sum_{uv} \sum_{ij \in \{1,2,3\}} \beta_{ij}^{uv} P_i P_j$$

$$P_1 = X, P_2 = Y, P_3 = Z, \quad \beta_{ij}^{uv} \in \mathbb{R}$$

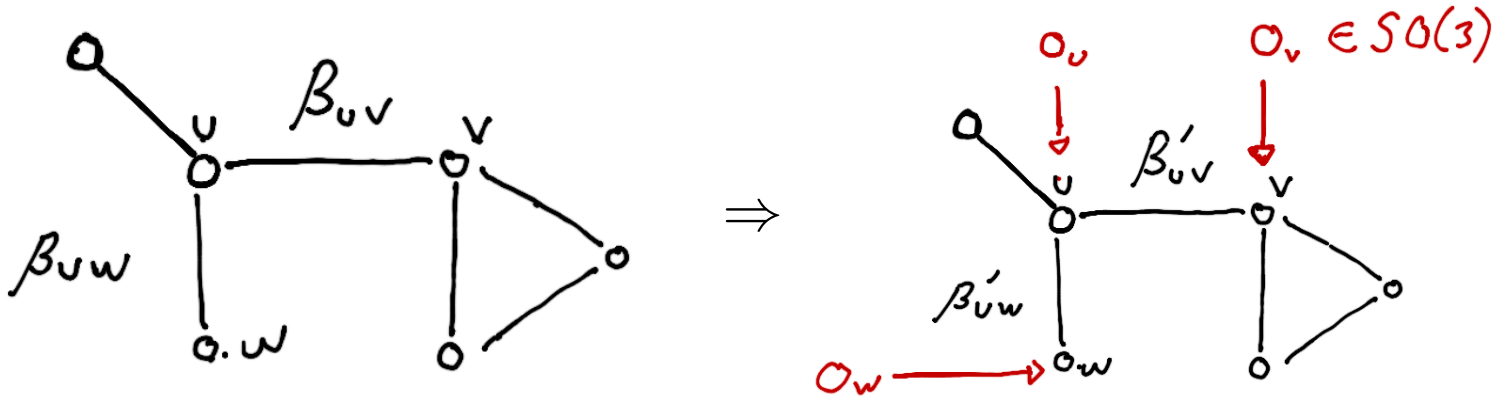
Theorem

There is an efficient algorithm that runs in time $O(n^3)$ that decides whether or not H is stoquastic in some local basis. The algorithm finds the local basis, or decides that no such local basis exists.

Graph Representation

$$H = \sum_{uv} \sum_{ij \in \{1,2,3\}} \beta_{ij}^{uv} P_i P_j$$

- β^{uv} is a matrix.
- $SU(2) \leftrightarrow SO(3)$. So $H \to (U_1 \otimes U_2) H (U_1 \otimes U_2)^\dagger$ corresponds to $\beta \to O_1^T \beta O_2$
- Our Hamiltonian looks like a matrix weighted graph, and we are applying $SO(3)$ rotations at the vertices:



$$(\beta^{uv})' = O_u^{\mathsf{T}} \beta^{uv} O_v, \quad (\beta^{uw})' = O_u^{\mathsf{T}} \beta^{uw} O_w, \quad \text{etc.}$$

Conditions

$$H_2 = \begin{pmatrix} \bullet & \beta_{ZX} & \beta_{XZ} & \beta_{XX} - \beta_{YY} \\ \bullet & \bullet & \beta_{XX} + \beta_{YY} & -\beta_{XZ} \\ \bullet & \bullet & \bullet & -\beta_{ZX} \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix} \tag{1}$$

$$+i \begin{pmatrix} \bullet & -\beta_{ZY} & -\beta_{YZ} & -\beta_{XY} - \beta_{YX} \\ \bullet & \bullet & \beta_{XY} - \beta_{YX} & \beta_{YZ} \\ \bullet & \bullet & \bullet & \beta_{ZY} \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix} \tag{2}$$

- terms associated with XY, ZY, YX, YZ must be zero, to keep things real.
- terms associated with XZ, ZX must be zero, and $\beta_{XX} \pm \beta_{YY}$ must be less than zero, to keep things negative.
- So to be stoquastic, all $\beta' = O_1^{\mathsf{T}} \beta O_2$ must be diagonal, and $\beta'_{11} \leq -|\beta'_{22}|$

Roadblock

- If all β are diagonal, then their corresponding interactions are $(\beta_{11} XX + \beta_{22} YY + \beta_{33} ZZ)$
- therefore the XYZ-algorithm solves the problem
- naively just need to find a simultaneous diagonalization of the matrix weights.
- **problem:** simultaneously diagonalizing all β in a graph is an NP-hard problem. [Klassen, Terhal 2018]
- **workaround:** ignore some cases that will not have a chance of being stoquastic, no longer NP-hard!

Problem Statement

Problem Statement

Find a set of orthogonal rotations $\{O_u\}$ such that:

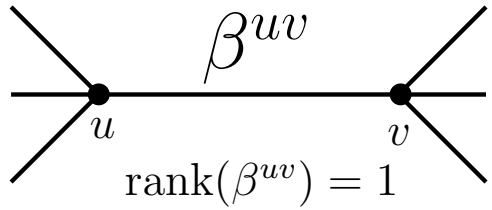
1. $O_u^{\mathsf{T}} \beta^{uv} O_v$ is diagonal for all u, v.
2. $[O_u^{\mathsf{T}} \beta^{uv} O_v]_{22} = 0$ for all β^{uv} with $\mathrm{rank}(\beta^{uv}) = 1$.

Or show that none exists.

- Both conditions are necessary. $(\beta'_{11} \leq -|\beta'_{22}|)$
- If you find such a set, pass the solution to the XYZ algorithm
- Turns out it suffices to consider $O(3)$ rotations instead of $SO(3)$ rotations (we can dump signs into the ZZ terms.)
- **!!** If $\{O_u\}$ is a solution, then so is $\{O_u \mathrm{diag}(\pm 1, \pm 1, \pm 1)\}$**!!**

Illustrative Subcase

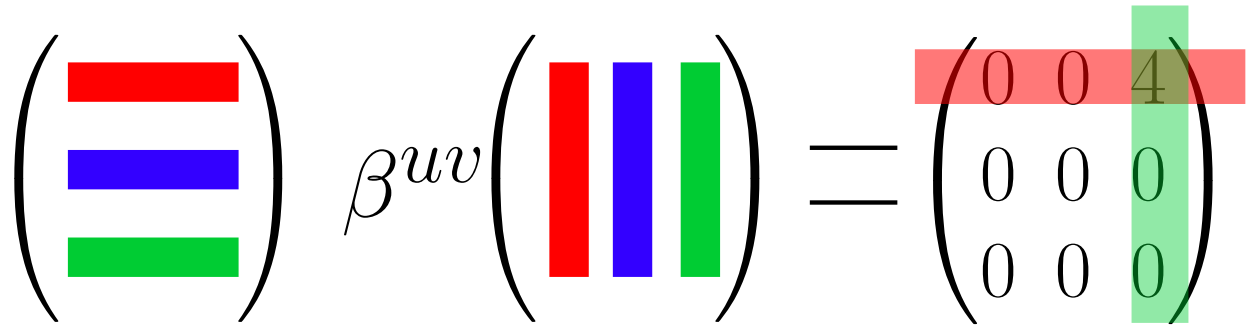
- What if every edge in our graph has rank-1 β?



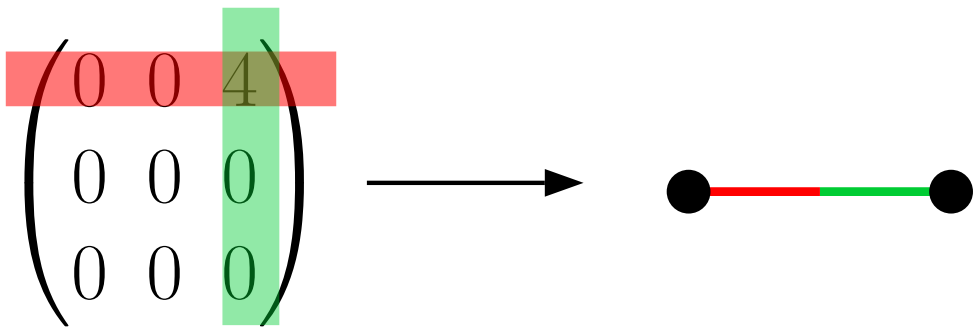
- For every vertex u find an O_u that simultaneously diagonalizes $\beta^{uv}(\beta^{uv})^{\mathsf{T}}$ for all v adjacent to u. This is not hard.
- If you can't find such O_u, then H is not stoquastic.
- Otherwise we can apply them and then $O_u^{\mathsf{T}} \beta^{uv} O_v$ must have a single non-zero entry, but maybe not in the right place:

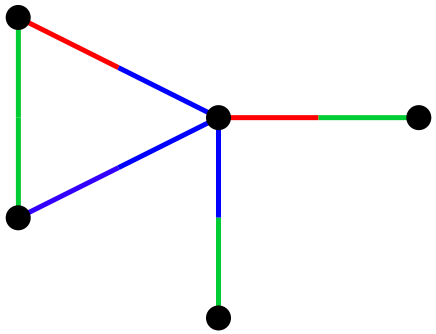
$$O_u^{\mathsf{T}} \beta^{uv} O_v = \begin{pmatrix} 0 & 0 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

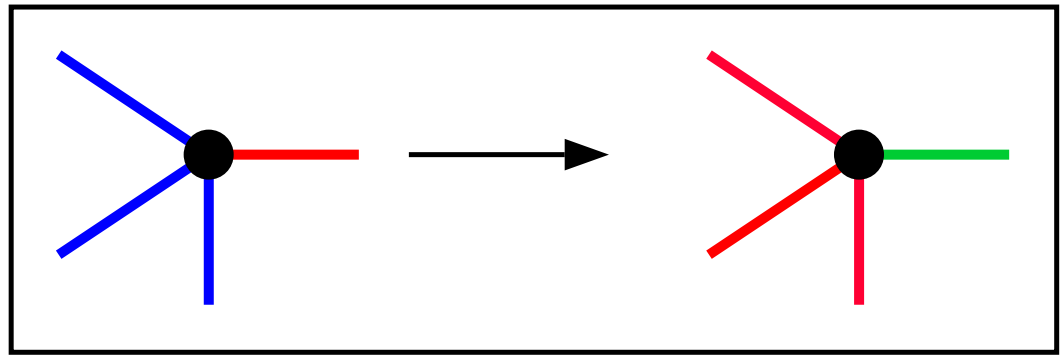
- We want to move that non-zero entry into the first or third position!



- 3rd column of O_v and 1st column of O_u are the right and left singular vectors not in the null space of β^{uv}.
- To move the non-zero entry, permute the columns of O_u and O_v.
- but this has an effect on the other edges!
- We can bi-colour each edge in terms of the positions of the columns which have non-zero singular values.

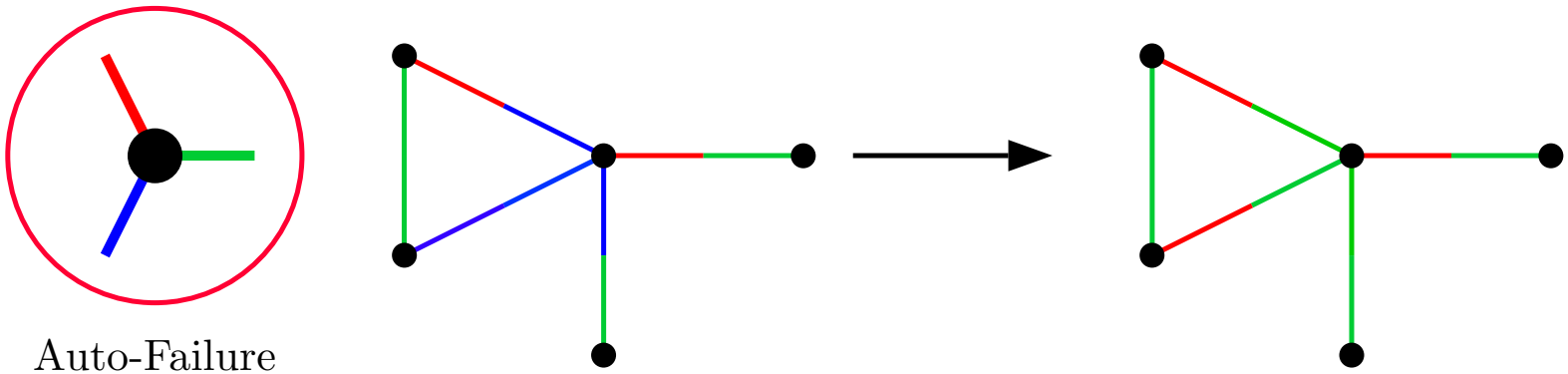






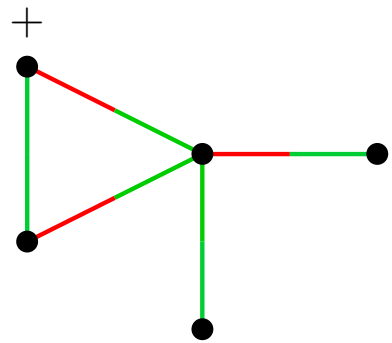
- Now we have a bicoloured graph.
- We can permute the colours at each vertex (permuting columns of O_u)
- We want every edge uniform, and to remove all blue edges (second position)
- Just making edges uniform is NP-complete.
- But if we also want to remove all blue edges, then it becomes easy.

- Flip all blue edges to whichever colour is not present at the vertex:



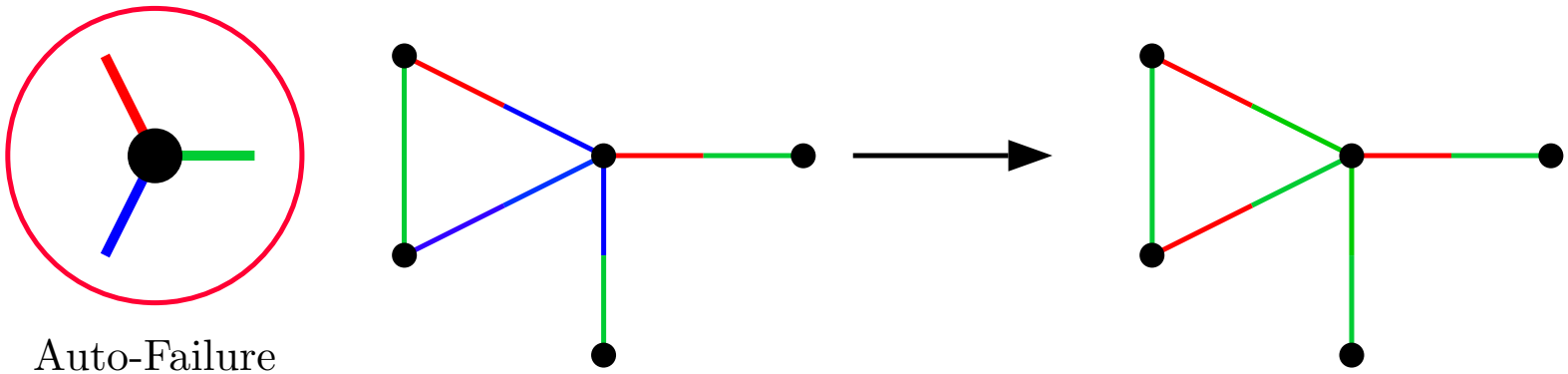
Auto-Failure

- can either permute green and red $(-)$, or not $(+)$. (think Ising model)
- choose $(+)$ or $(-)$ at one vertex, then propagate:



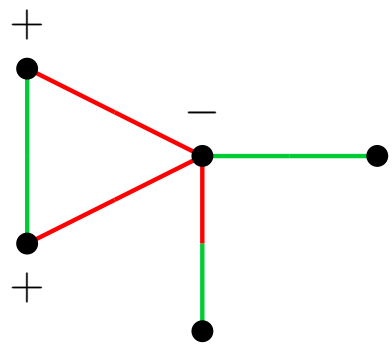
- If you run into a contradiction, then no other choice would solve it.

- Flip all blue edges to whichever colour is not present at the vertex:



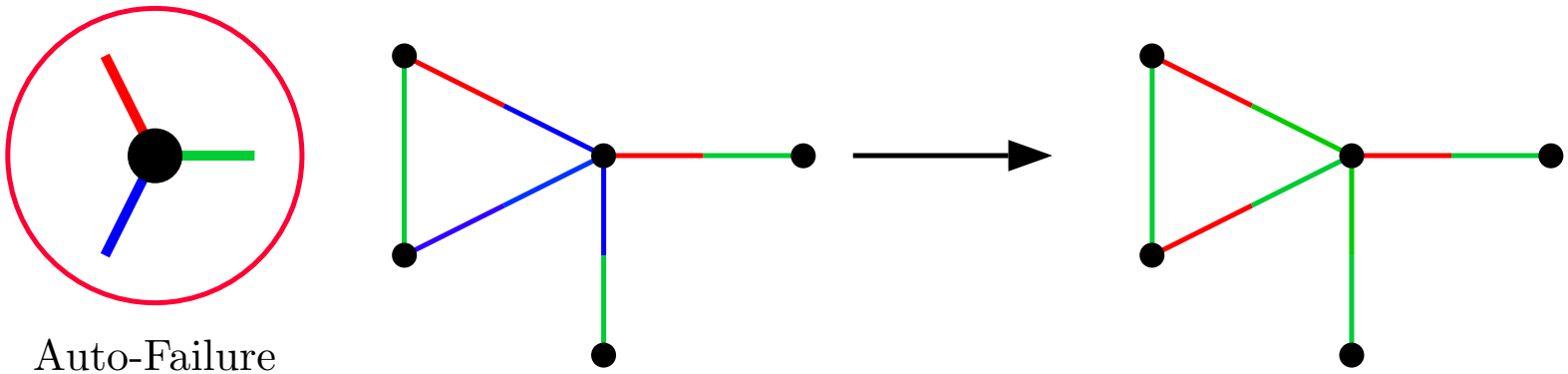
Auto-Failure

- can either permute green and red $(-)$, or not $(+)$. (think Ising model)
- choose $(+)$ or $(-)$ at one vertex, then propagate:



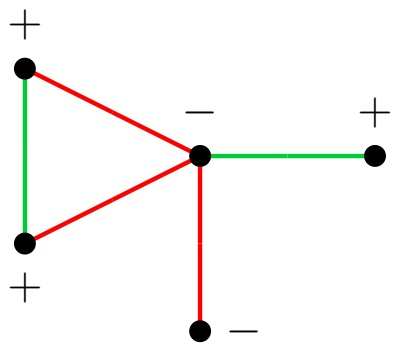
- If you run into a contradiction, then no other choice would solve it.

- Flip all blue edges to whichever colour is not present at the vertex:



Auto-Failure

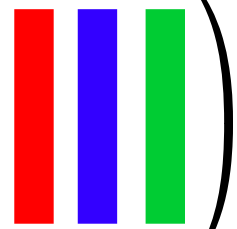
- can either permute green and red $(-)$, or not $(+)$. (think Ising model)
- choose $(+)$ or $(-)$ at one vertex, then propagate:



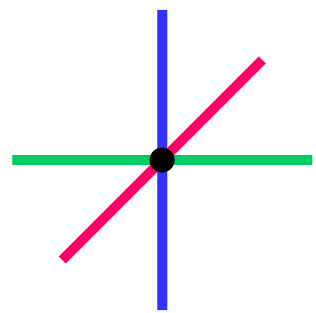
- If you run into a contradiction, then no other choice would solve it.

Incorporating Rank>1 Edges

- Let us now consider the most general case.
- continue thinking of O_u in terms of orthonormal column vectors.

$$O_u = \left(\right.$$  $$\left. \right)$$

- Recall that we don't care about the signs of these vectors so let's represent them projectively (represent columns as 1-d subspaces):

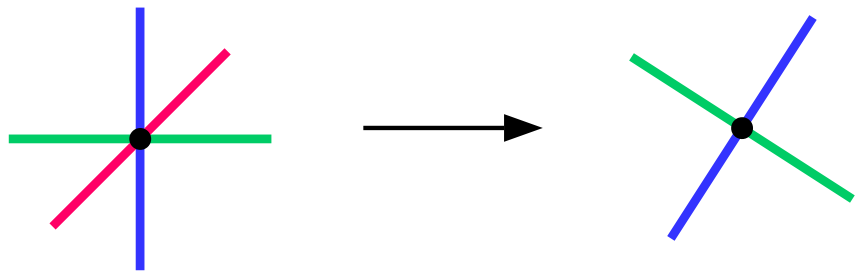
$$O_u =$$ 

- we still need to care about the ordering of the columns!

If $O_u^\mathsf{T} \beta^{uv} O_v$ is diagonal, then the ith column e_v^i of O_v is either in the kernel of β^{uv}, or $\beta^{uv} e_v^i \propto e_u^i$, the ith column of O_u.

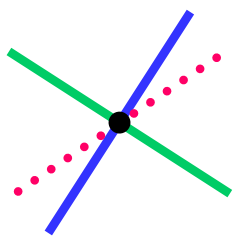
So if O_v diagonalizes $(\beta^{uv})^\mathsf{T} \beta^{uv}$ then we can think of β^{uv} as a rotation on some of these 1-d subspaces:

β^{uv}:



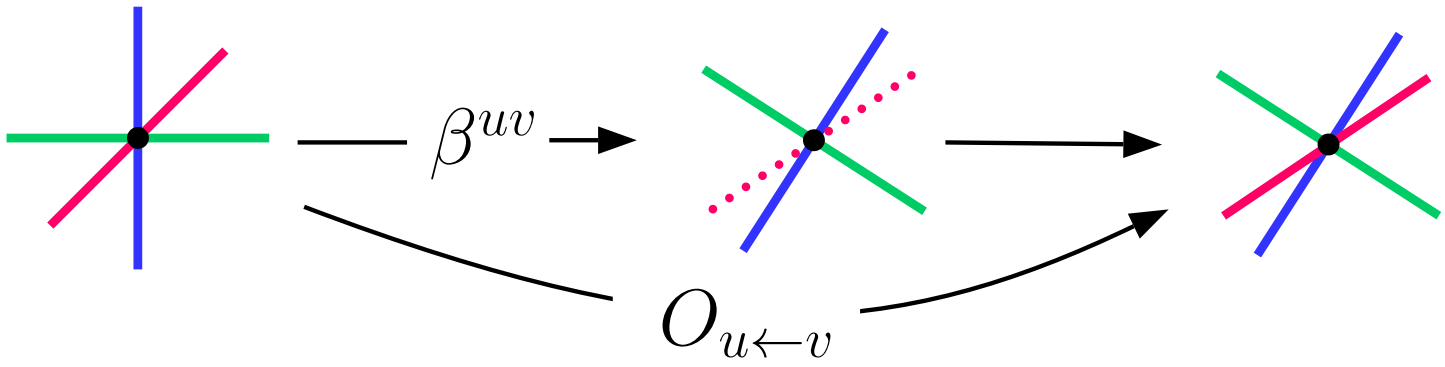
And this rotation partially specifies what O_u must be in order for O_u, O_v to diagonalize β^{uv}:

$O_u =$

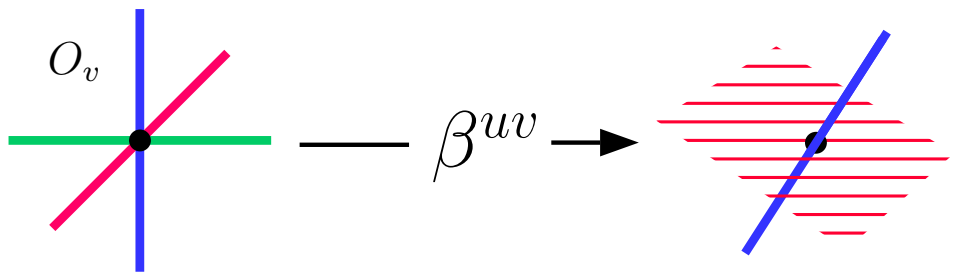


Rank-1 Edges and Rank>1 Edges

If $\mathrm{rank}(\beta^{uv}) > 1$ then the choice of O_v completely specifies O_u by the mapping β^{uv}:

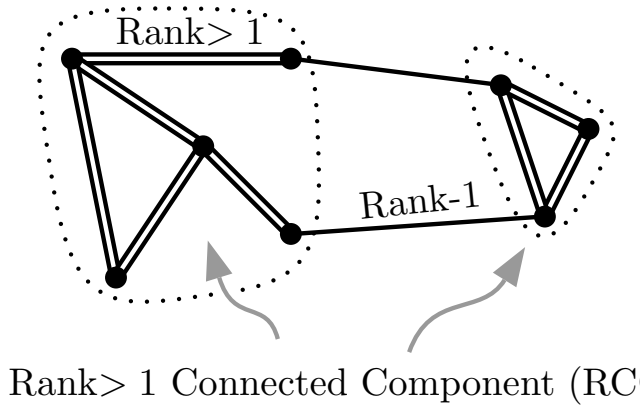


But if $\mathrm{rank}(\beta^{uv}) = 1$ then O_u is underspecified:

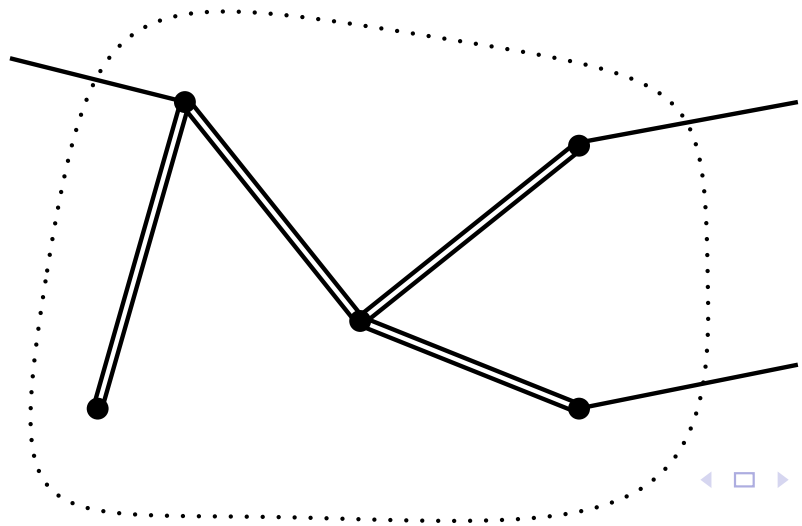


Rank>1 Connected components

Returning to our graph picture, this important distinction between Rank>1 and Rank-1 matrix weights adds structure to the graph:

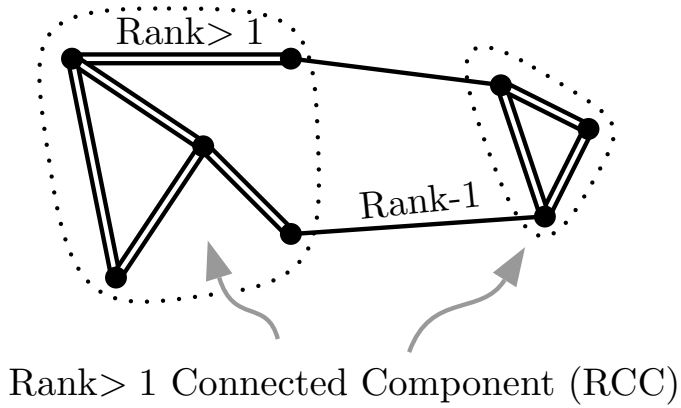


If we wisely choose the rotation at some starting vertex in an RCC, then that specifies the rotations at all other vertices in the RCC:

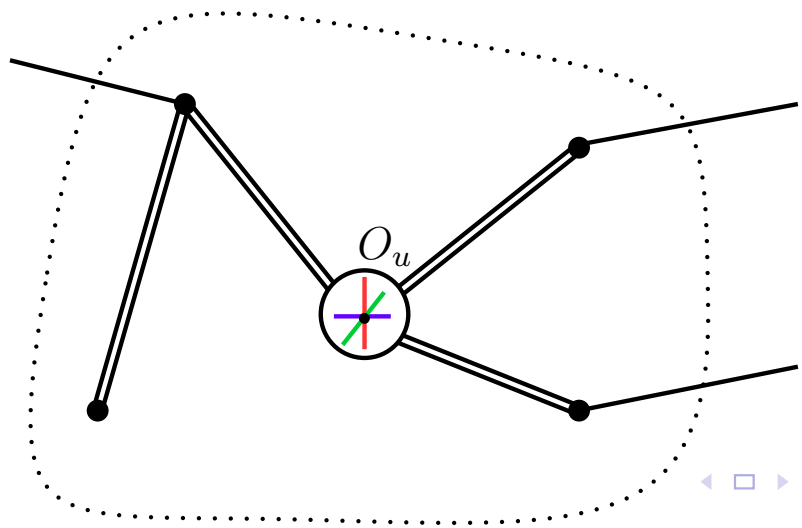


Rank>1 Connected components

Returning to our graph picture, this important distinction between Rank>1 and Rank-1 matrix weights adds structure to the graph:

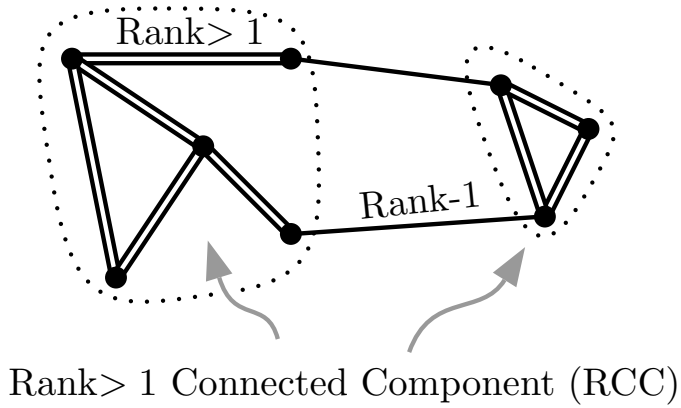


If we wisely choose the rotation at some starting vertex in an RCC, then that specifies the rotations at all other vertices in the RCC:

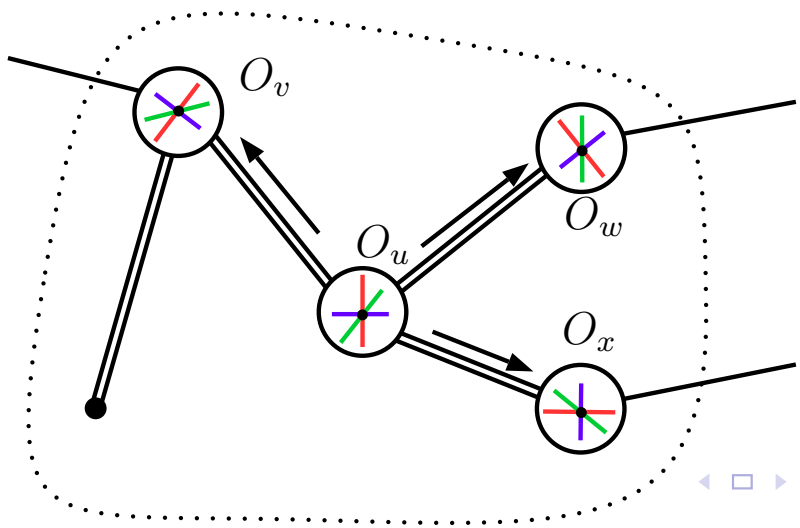


Rank> 1 Connected components

Returning to our graph picture, this important distinction between Rank> 1 and Rank-1 matrix weights adds structure to the graph:

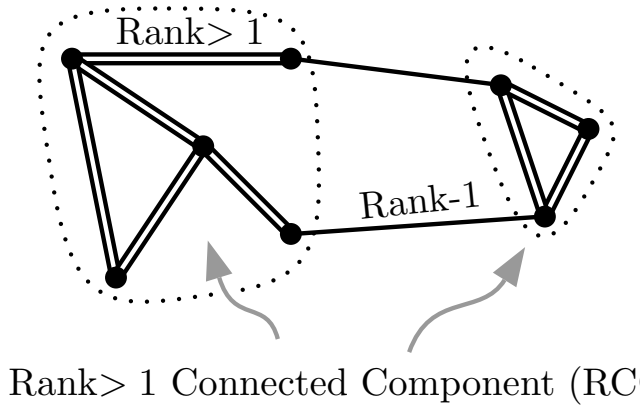


If we wisely choose the rotation at some starting vertex in an RCC, then that specifies the rotations at all other vertices in the RCC:

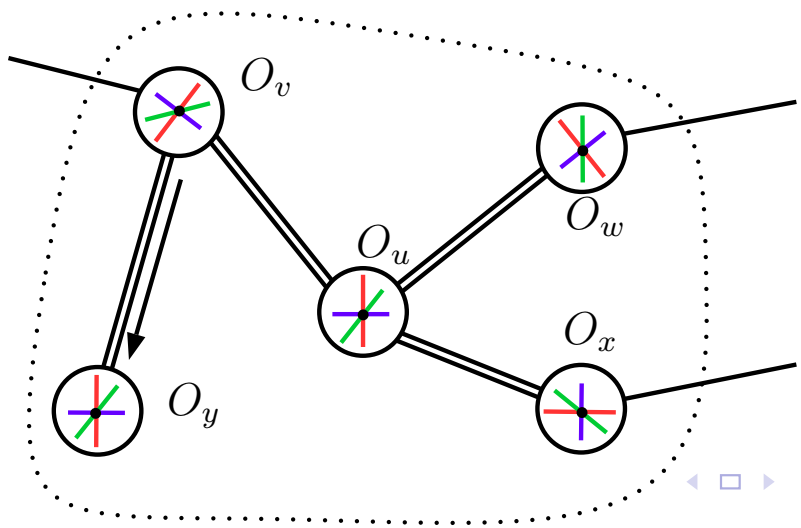


Rank>1 Connected components

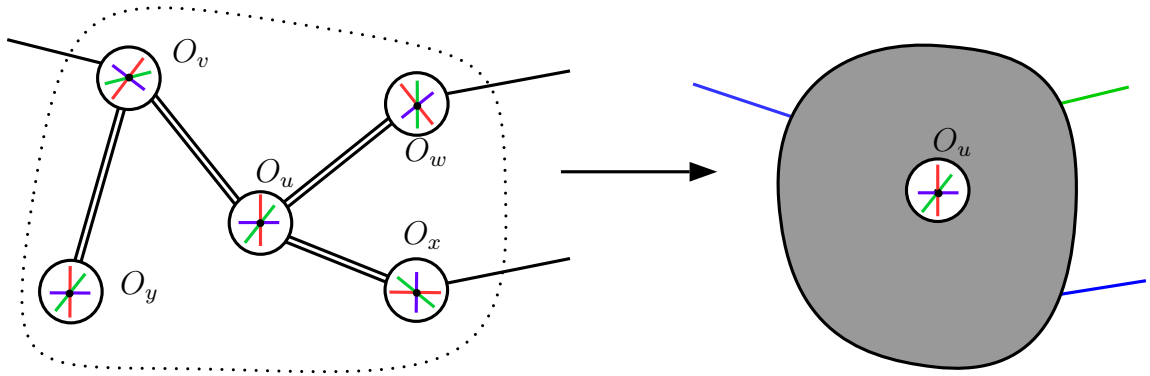
Returning to our graph picture, this important distinction between Rank>1 and Rank-1 matrix weights adds structure to the graph:



If we wisely choose the rotation at some starting vertex in an RCC, then that specifies the rotations at all other vertices in the RCC:



- Thus we can think of an RCC as a single site on which to apply an $O(3)$ rotation.



- All the edges in the RCC will be diagonalized
- And we pick up the same edge bi-colouring property for rank-1 edges.
- permuting the ordering of the columns of the starting choice propagates through the RCC in such a way that the colourings of the rank-1 edges permutes in a similar fashion.
- So if we can choose a good starting rotation for each RCC, then our problem reduces to the previous case.

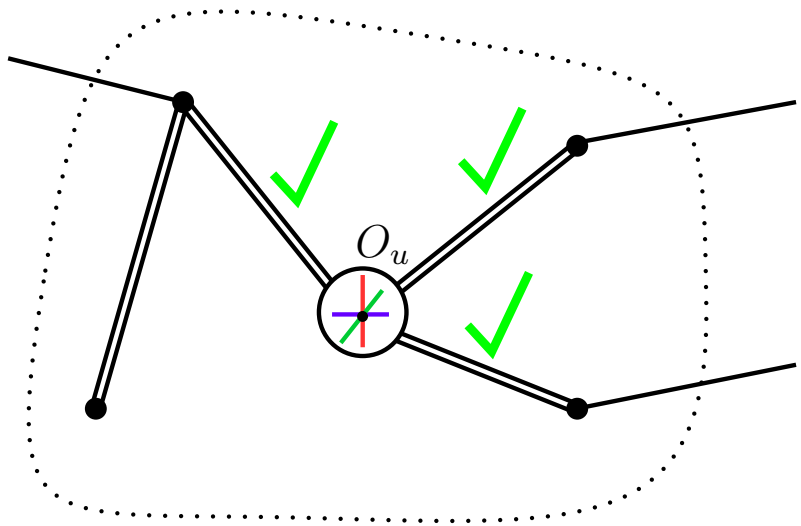
Boy, it sure would be nice to know what you mean by a good starting rotation...

Choosing a Good Starting Rotation

- The notion of O_u transporting along the rank>1 edge...
- and the notion of O_u inducing a bicolouring on its neighbouring rank-1 edge rely on...

$$\underline{O_u \text{ diagonalizing } \beta^{uv}(\beta^{uv})^{\mathsf{T}}}$$

- We need O_u to not only have this property for all of its neighbouring edges, but also all of the transported rotations must have this property!

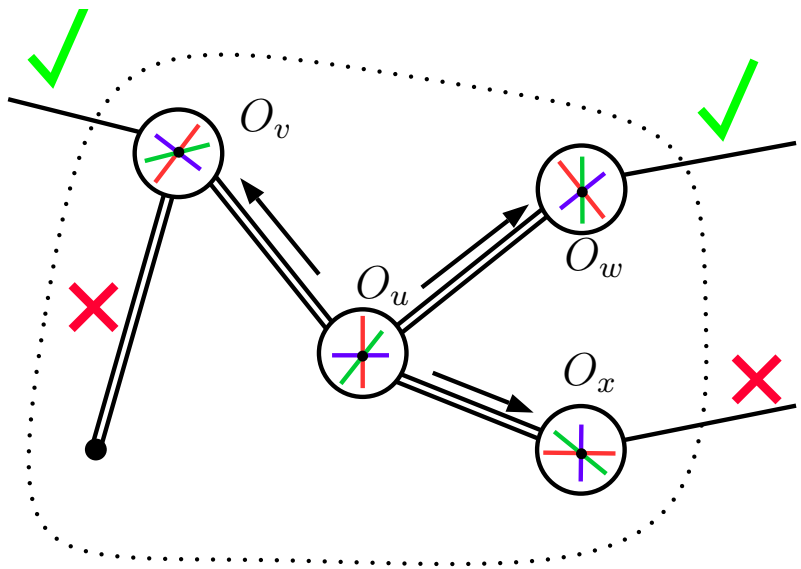


Choosing a Good Starting Rotation

- The notion of O_u transporting along the rank>1 edge...
- and the notion of O_u inducing a bicolouring on its neighbouring rank-1 edge rely on...

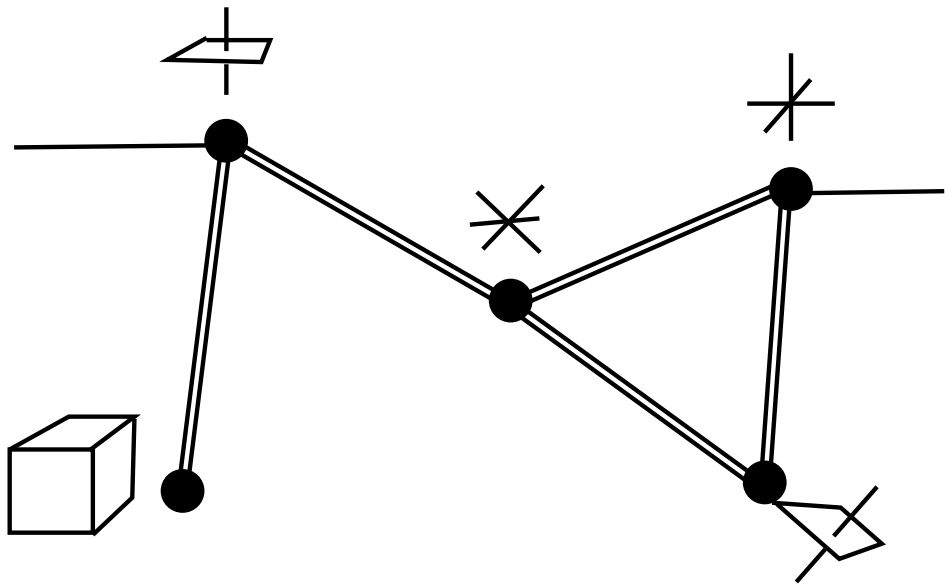
$$\underline{O_u \text{ diagonalizing } \beta^{uv}(\beta^{uv})^{\mathsf{T}}}$$

- We need O_u to not only have this property for all of its neighbouring edges, but also all of the transported rotations must have this property!



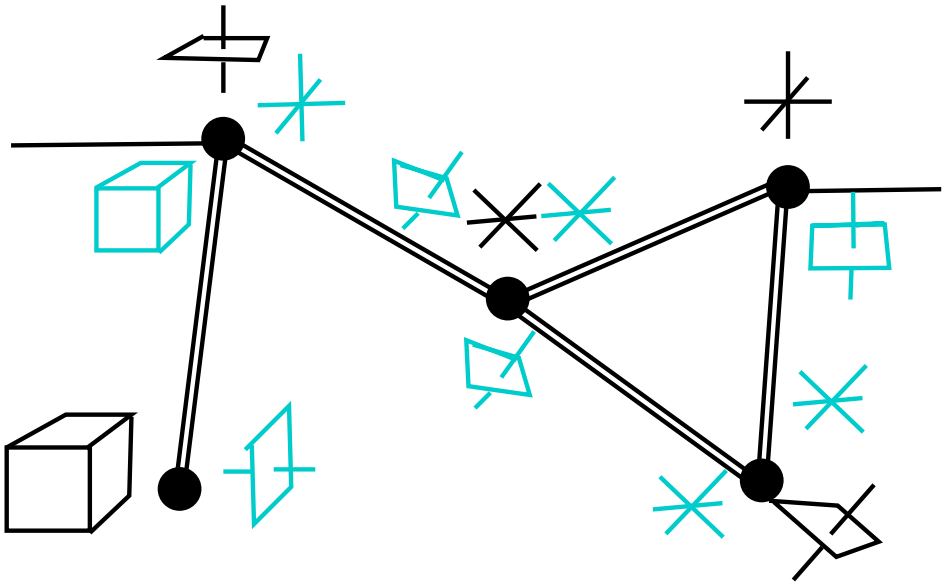
Choosing a Good Starting Rotation

Compute the simultaneous eigenspaces (eigenspaces $\beta^{uv}(\beta^{uv})^{\mathsf{T}}$ of every adjacent edge) for each vertex:



Choosing a Good Starting Rotation

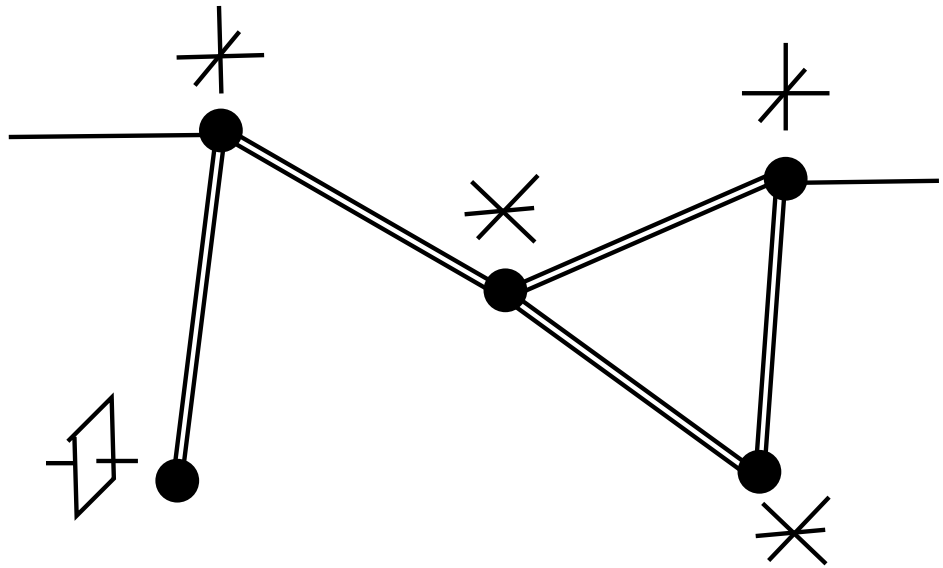
Compute the simultaneous eigenspaces (eigenspaces $\beta^{uv}(\beta^{uv})^{\mathsf{T}}$ of every adjacent edge) for each vertex:



Transport the eigenspaces to neighbouring edges and take intersections

Choosing a Good Starting Rotation

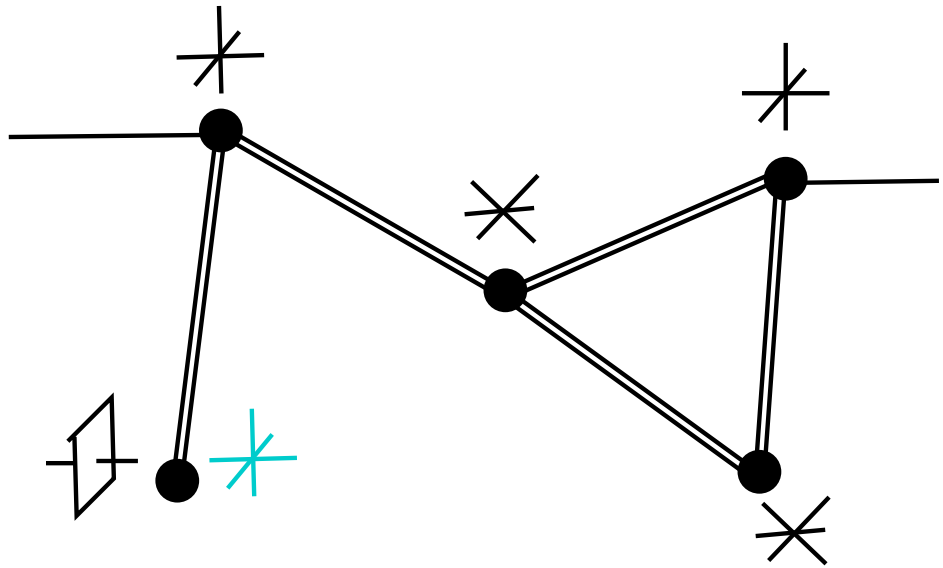
Compute the simultaneous eigenspaces (eigenspaces $\beta^{uv}(\beta^{uv})^{\mathsf{T}}$ of every adjacent edge) for each vertex:



Transport the eigenspaces to neighbouring edges and take intersections

Choosing a Good Starting Rotation

Compute the simultaneous eigenspaces (eigenspaces $\beta^{uv}(\beta^{uv})^{\mathsf{T}}$ of every adjacent edge) for each vertex:

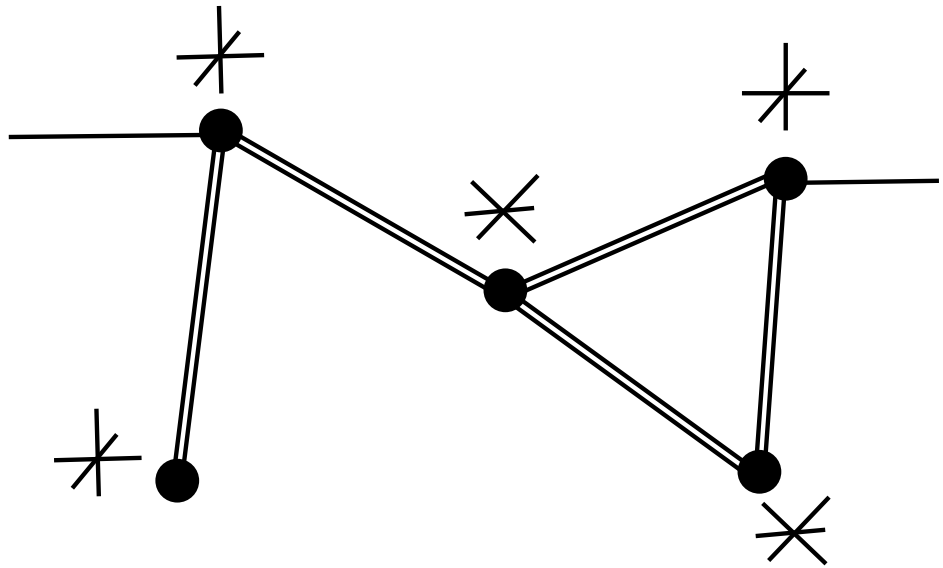


Transport the eigenspaces to neighbouring edges and take intersections

Repeat until you reach a fixed point.

Choosing a Good Starting Rotation

Compute the simultaneous eigenspaces (eigenspaces $\beta^{uv}(\beta^{uv})^{\mathsf{T}}$ of every adjacent edge) for each vertex:

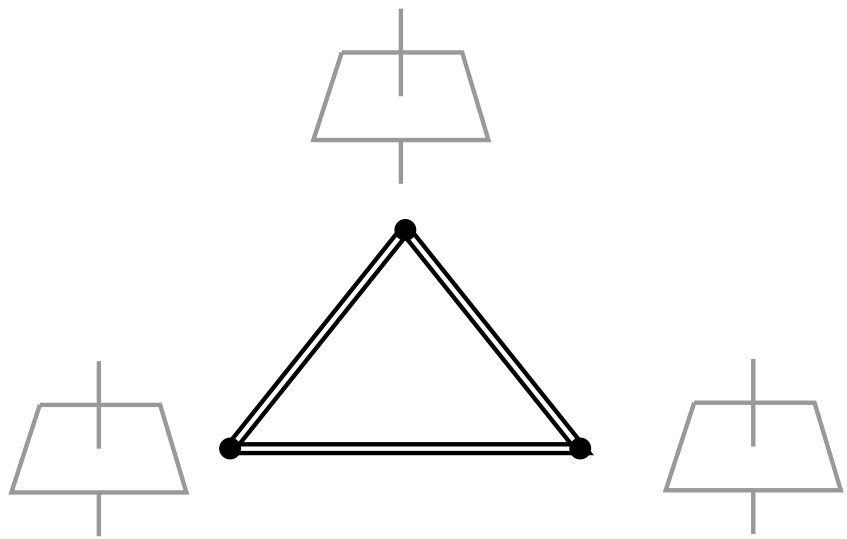


Transport the eigenspaces to neighbouring edges and take intersections

Repeat until you reach a fixed point.
A good starting rotation must be drawn from this final set.

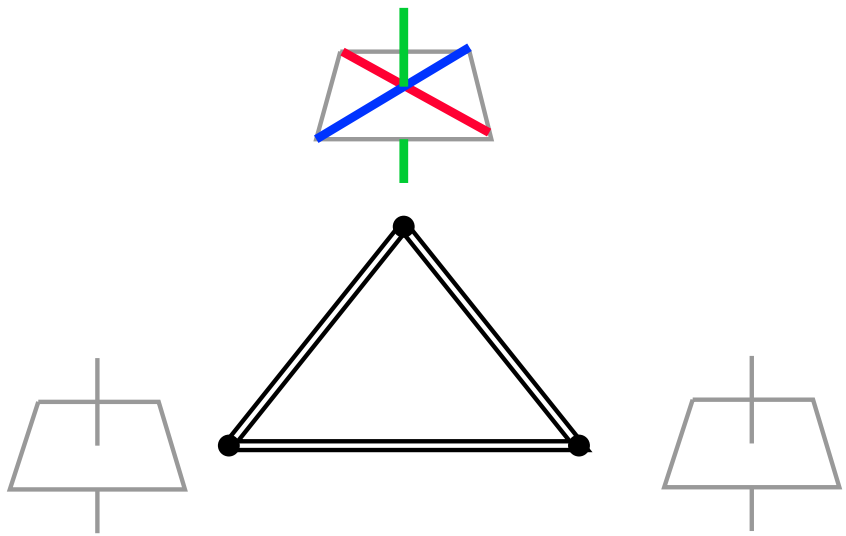
One Last Consideration: Topological Frustrations

- The previous illustration suggested a trivial choice for the starting rotation.
- Sometimes you have bigger subspaces to choose from:



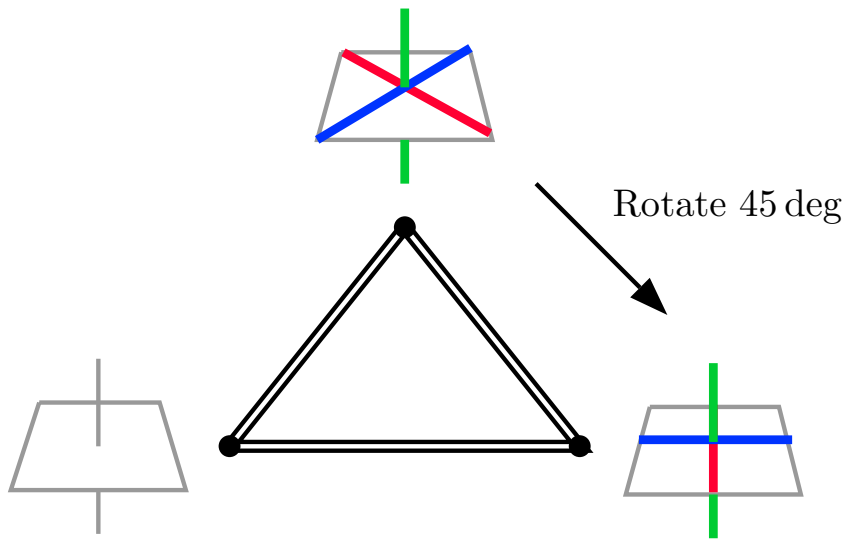
One Last Consideration: Topological Frustrations

- The previous illustration suggested a trivial choice for the starting rotation.
- Sometimes you have bigger subspaces to choose from:



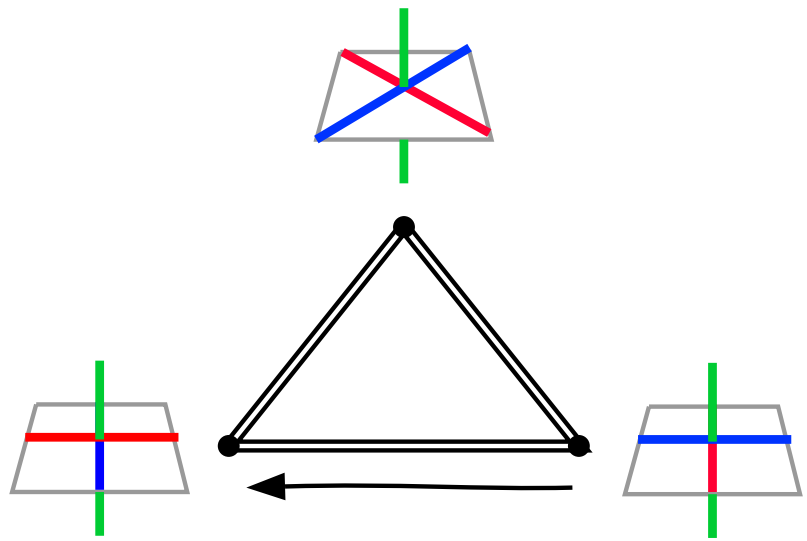
One Last Consideration: Topological Frustrations

- The previous illustration suggested a trivial choice for the starting rotation.
- Sometimes you have bigger subspaces to choose from:



One Last Consideration: Topological Frustrations

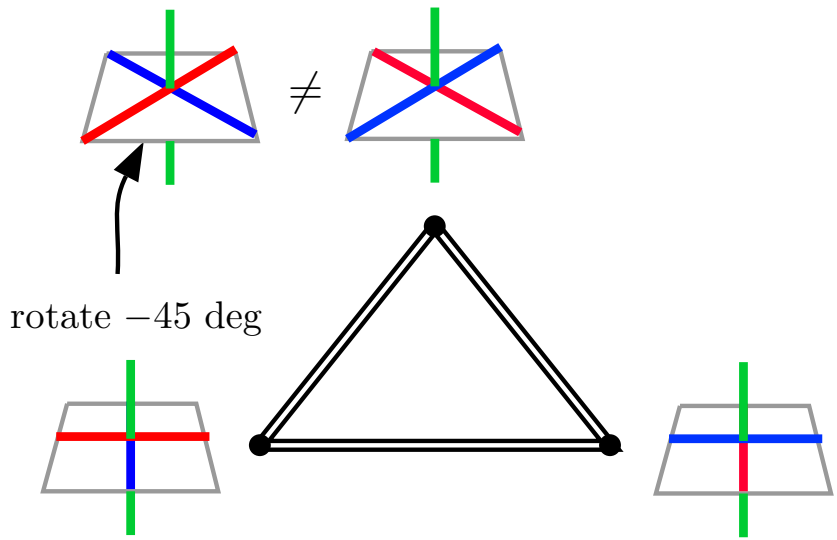
- The previous illustration suggested a trivial choice for the starting rotation.
- Sometimes you have bigger subspaces to choose from:



Flip along diagonal

One Last Consideration: Topological Frustrations

- The previous illustration suggested a trivial choice for the starting rotation.
- Sometimes you have bigger subspaces to choose from:

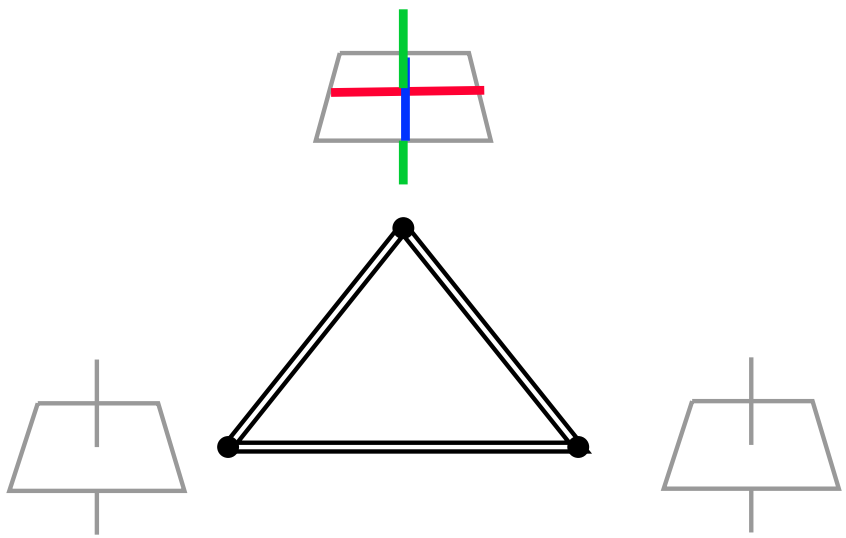


- there are cases where you can still make a bad starting choice!

One Last Consideration: Topological Frustrations

So for the final step:

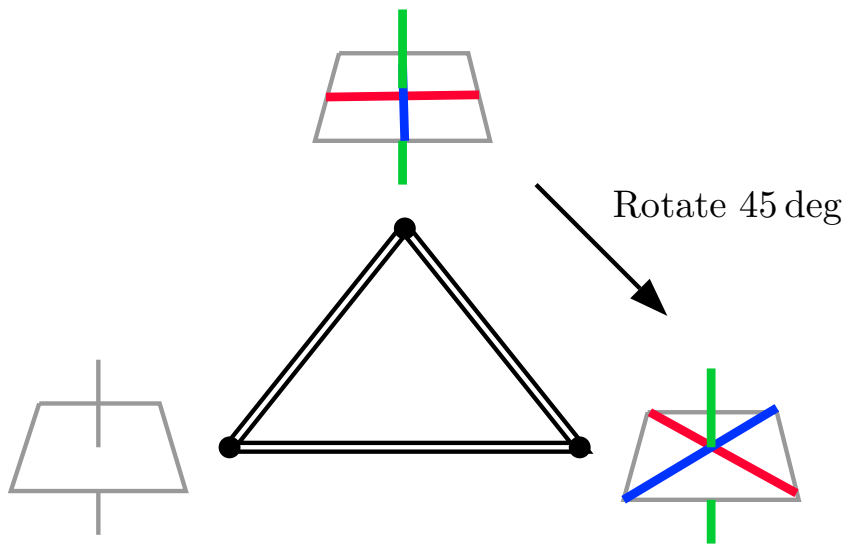
- Identify all the fundamental cycles in the RCC
- A good choice must also be from the eigenspaces of the transport operators associated with those cycles.
- Lets try the previous example again:



One Last Consideration: Topological Frustrations

So for the final step:

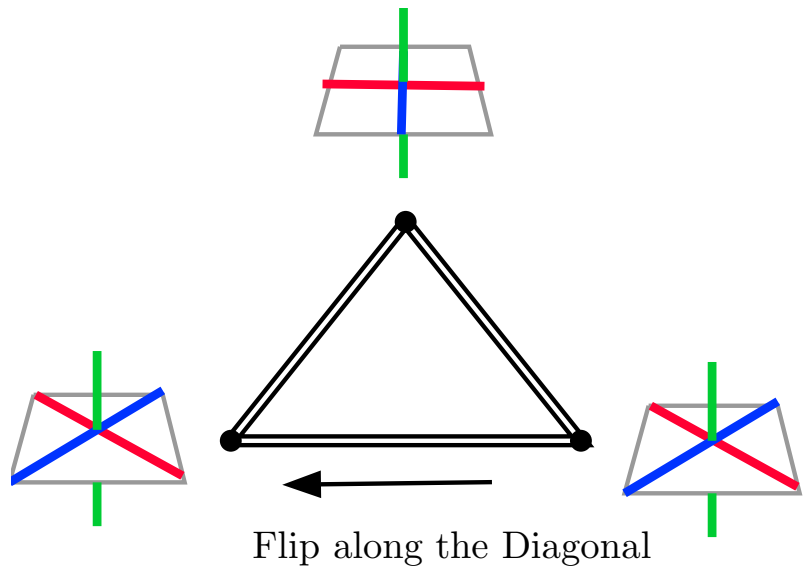
- Identify all the fundamental cycles in the RCC
- A good choice must also be from the eigenspaces of the transport operators associated with those cycles.
- Lets try the previous example again:



One Last Consideration: Topological Frustrations

So for the final step:

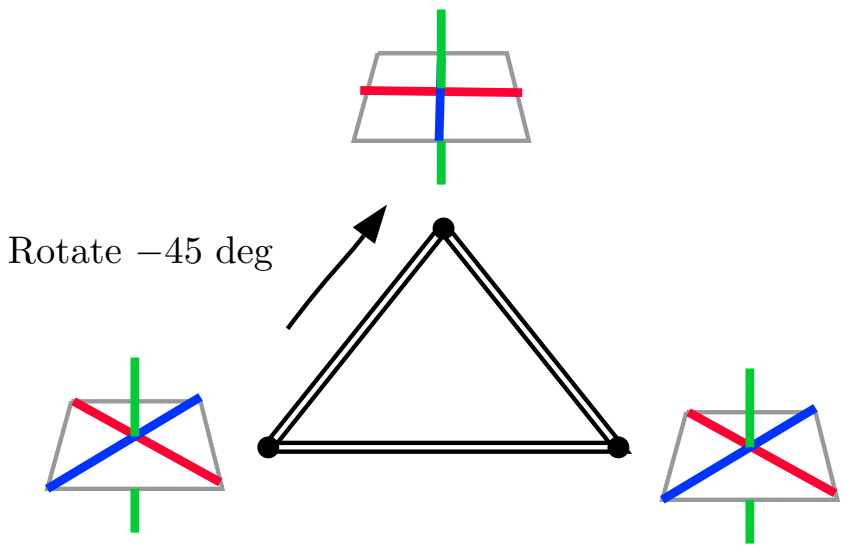
- Identify all the fundamental cycles in the RCC
- A good choice must also be from the eigenspaces of the transport operators associated with those cycles.
- Lets try the previous example again:



One Last Consideration: Topological Frustrations

So for the final step:

- Identify all the fundamental cycles in the RCC
- A good choice must also be from the eigenspaces of the transport operators associated with those cycles.
- Lets try the previous example again:



Recap

- Identify rank>1 connected components (RCCs)
- Find a good initial choice of rotation for each RCC by:
  - computing the intersections of the transported eigenspaces in the RCC
  - computing the real eigenspaces of the fundamental cycles, centered at a vertex.
  - selecting basis elements from the intersection of these sets, and making those the columns of your initial rotation choice
  - propagating the choice through the RCC
- Treat the RCC as a single vertex in a graph with only rank-1 edges, and solve the bicoloured graph problem.
- The solution specifies how the columns of the orthogonal rotations on each RCC should be permuted.

If at any point a step fails, then no solution exists (this is non-trivial)

Can we add 1-local terms?

Can we generalize this algorithm to the generic 2-local Hamiltonian, which includes 1-local terms?

The answer is no.

We show that adding 1-local terms makes the problem NP-hard.

We do this by constructing a reduction to 3-SAT

This is possible thanks in part to the fact that there's a freedom in how 1-local terms can be grouped with 2-local terms.

So we have found a nice delineation! The presence or absence of 1-local terms determines if deciding stoquasticity is hard or easy (for 2-local qubit Hamiltonians).

Future Directions

- Recently D-Wave announced that they had engineered a non-stoquastic interaction. [arxiv:1903.06139]
- In the low energy space they do indeed get a 2-qubit interaction that is not stoquastic in any local basis.
- But the full circuit Hamiltonian can always be made stoquastic by a canonical transformation.
- Under what circumstances can we lift non-stoquastic Hamiltonians to stoquastic ones? What are the consequences of this?

Future Directions

- Global and termwise stoquasticity is different
- What can we say about this distinction?
- When is it important?
- Masters student Marios Iannous is thinking about this.