Title: Quantum computation as geometry

Date: Aug 02, 2006 02:00 PM

URL: http://pirsa.org/06080007

Abstract: How should we think about quantum computing? The usual answer to this question is based on ideas inspired by computer science, such as qubits, quantum gates, and quantum circuits. In this talk I will explain an alternate geometric approach to quantum computation. In the geometric approach, an optimal quantum computation corresponds to "free falling" along the minimal geodesics of a certain Riemannian manifold. This reformulation opens up the possibility of using tools from geometry to understand the strengths and weaknesses of quantum computation, and perhaps to understand what makes certain physical operations difficult (or easy) to synthesize.

The overall picture: spoiling the plot

The overall picture: spoiling the plot

Quantum computing

Finding small quantum circuits.

⟷

Riemannian geometry

Finding shortest paths on a particular manifold.

The overall picture: spoiling the plot

Quantum computing ⟷ **Riemannian geometry**

Finding small quantum circuits.

Finding shortest paths on a particular manifold.

The overall picture: spoiling the plot

Quantum computing ⟷ **Riemannian geometry**

Finding small quantum circuits.

Finding shortest paths on a particular manifold.

Motivation: Church-Turing-Deutsch Principle.

The overall picture: spoiling the plot

Quantum computing

Finding small quantum circuits.

Riemannian geometry

Finding shortest paths on a particular manifold.

Motivation: Church-Turing-Deutsch Principle.

I'll describe the equivalence.

The overall picture: spoiling the plot

Quantum computing ⟷ **Riemannian geometry**

Finding small quantum circuits.

Finding shortest paths on a particular manifold.

Motivation: Church-Turing-Deutsch Principle.

I'll describe the equivalence.

I'll overview what we know about the geometry (not much!)

The overall picture: spoiling the plot

Quantum computing ⟷ **Riemannian geometry**

Finding small quantum circuits.

Finding shortest paths on a particular manifold.

Motivation: Church-Turing-Deutsch Principle.

I'll describe the equivalence.

I'll overview what we know about the geometry (not much!)

I'll speculate on how ideas from geometry may be used to gain insight into quantum computing.

Finding minimal circuits is hard

Finding minimal circuits is hard

Shannon (1937): Most Boolean functions require exponential size circuits.

Finding minimal circuits is hard

Shannon (1937): Most Boolean functions require exponential size circuits.

"However all attempts to find even superlinear lower bounds for unrestricted circuits for `explicitly given' Boolean functions have met with total failure; the best such lower bound given so far is about 4n." – Steven Cook (2003)

Finding minimal circuits is hard

Shannon (1937): Most Boolean functions require exponential size circuits.

"However all attempts to find even superlinear lower bounds for unrestricted circuits for `explicitly given' Boolean functions have met with total failure; the best such lower bound given so far is about 4n." – Steven Cook (2003)

The situation is similar for implementing unitary operations with quantum circuits.

Finding minimal circuits is hard

Shannon (1937): Most Boolean functions require exponential size circuits.

"However all attempts to find even superlinear lower bounds for unrestricted circuits for `explicitly given' Boolean functions have met with total failure; the best such lower bound given so far is about 4n." – Steven Cook (2003)

The situation is similar for implementing unitary operations with quantum circuits.

Virtually all the major problems in computational complexity theory are still open: separate P from NP, or PSPACE.

Finding minimal circuits is hard

Finding minimal circuits is hard

Some researchers suspect that whether P = NP may be independent of the usual axioms of mathematics.

Finding minimal circuits is hard

Some researchers suspect that whether P = NP may be independent of the usual axioms of mathematics.

Razborov-Rudich (very loosely): If good pseudorandom number generators exist, then it's impossible to efficiently distinguish hard-to-compute functions from easy to compute functions.

Finding minimal circuits is hard

Some researchers suspect that whether P = NP may be independent of the usual axioms of mathematics.

Razborov-Rudich (very loosely): If good pseudorandom number generators exist, then it's impossible to efficiently distinguish hard-to-compute functions from easy to compute functions.

Why (loosely): Because a good pseudorandom number generator can be used to generate a pseudorandom function which is (1) easy to compute, and (2) impossible to efficiently distinguish from a truly random function, which is hard to compute.

Finding minimal circuits is hard

Some researchers suspect that whether P = NP may be independent of the usual axioms of mathematics.

Razborov-Rudich (very loosely): If good pseudorandom number generators exist, then it's impossible to efficiently distinguish hard-to-compute functions from easy to compute functions.

Finding minimal circuits is hard

Some researchers suspect that whether P = NP may be independent of the usual axioms of mathematics.

Razborov-Rudich (very loosely): If good pseudorandom number generators exist, then it's impossible to efficiently distinguish hard-to-compute functions from easy to compute functions.

Why (loosely): Because a good pseudorandom number generator can be used to generate a pseudorandom function which is (1) easy to compute, and (2) impossible to efficiently distinguish from a truly random function, which is hard to compute.

Finding minimal circuits is hard

Some researchers suspect that whether $P = NP$ may be independent of the usual axioms of mathematics.

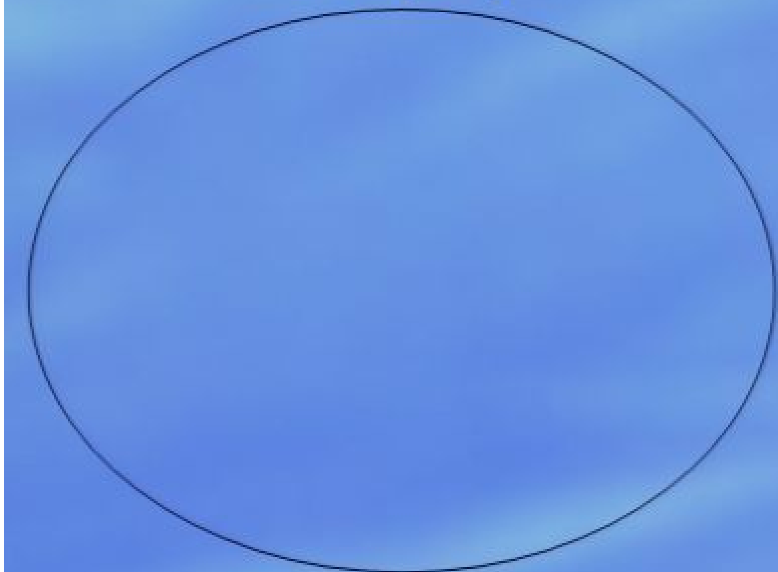
Razborov-Rudich (very loosely): If good pseudorandom number generators exist, then it's impossible to efficiently distinguish hard-to-compute functions from easy to compute functions.

Why (loosely): Because a good pseudorandom number generator can be used to generate a pseudorandom function which is (1) easy to compute, and (2) impossible to efficiently distinguish from a truly random function, which is hard to compute.

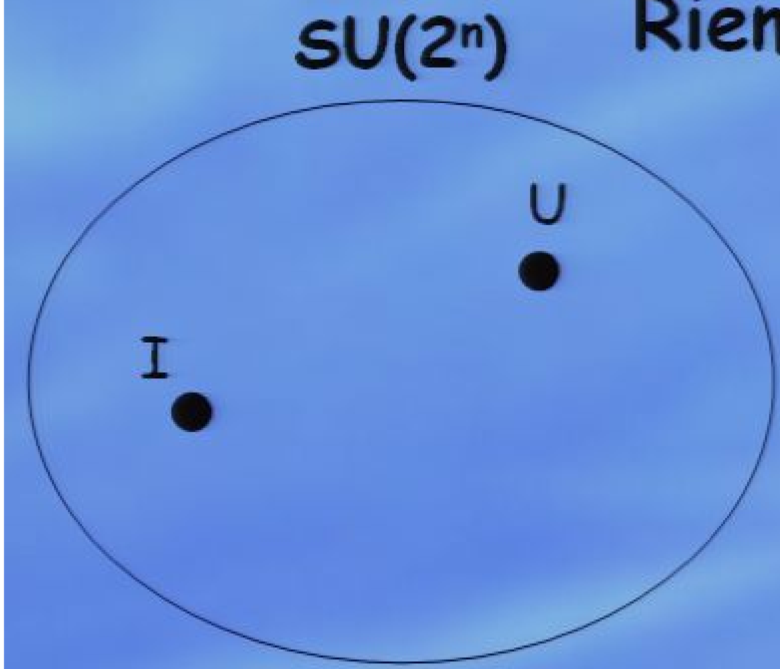
So any proof that (for example) $P \neq NP$ must not rely on being able to efficiently distinguish easy- and hard-to-compute functions.

Quantum computing and Riemannian geometry

$SU(2^n)$

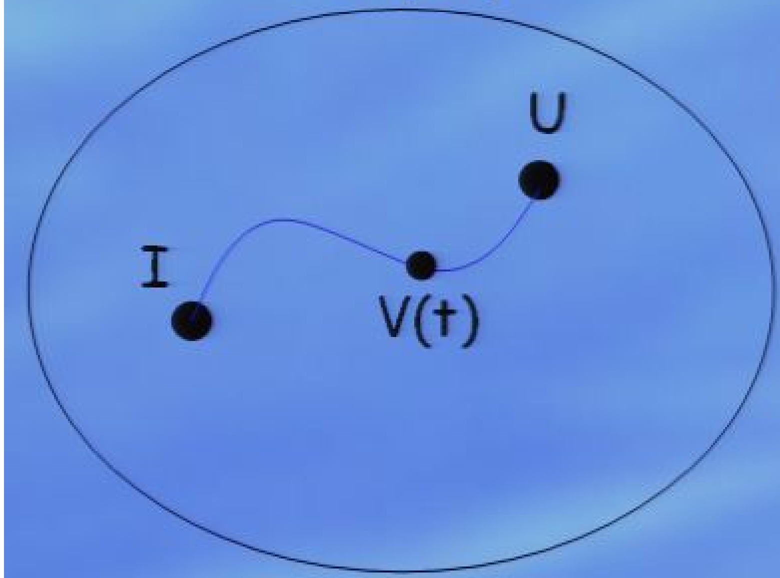


Quantum computing and Riemannian geometry



Quantum computing and Riemannian geometry

$SU(2^n)$



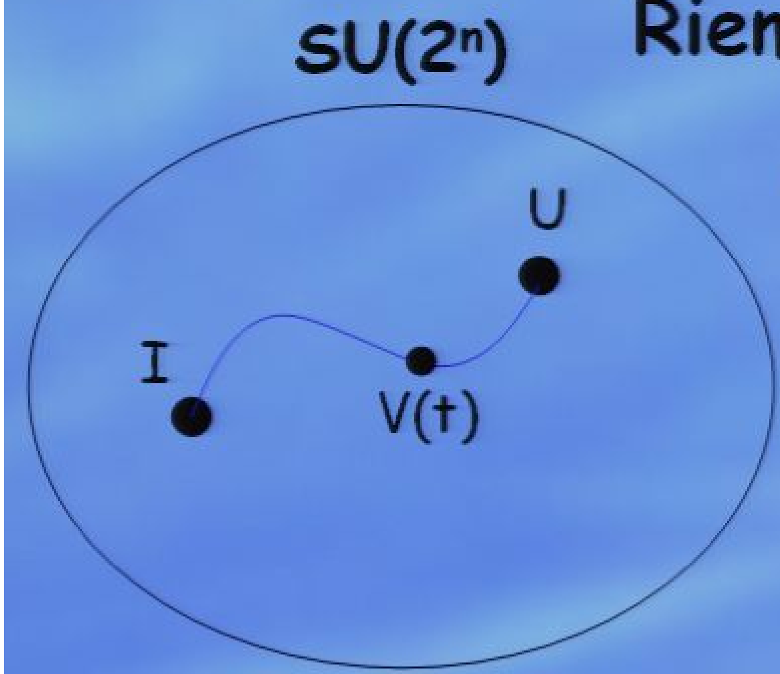
We can think of U as being generated by an n-qubit time-dependent Hamiltonian:

$$H(t) = \sum_\sigma h_\sigma(t)\, \sigma$$

$$V(0) = I; \qquad V(1) = U$$

Notation: σ = tensor product of the Pauli matrices or I.

Quantum computing and Riemannian geometry



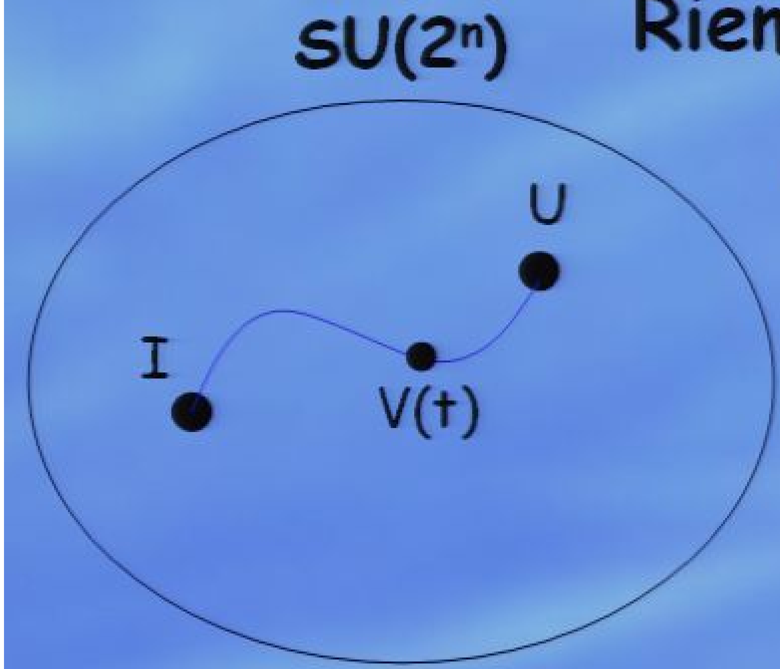
We can think of U as being generated by an n-qubit time-dependent Hamiltonian:

$$H(t) = \sum_\sigma h_\sigma(t)\, \sigma$$

$$V(0) = I; \qquad V(1) = U$$

Notation: σ = tensor product of the Pauli matrices or I.

Quantum computing and Riemannian geometry



We can think of U as being generated by an n-qubit time-dependent Hamiltonian:

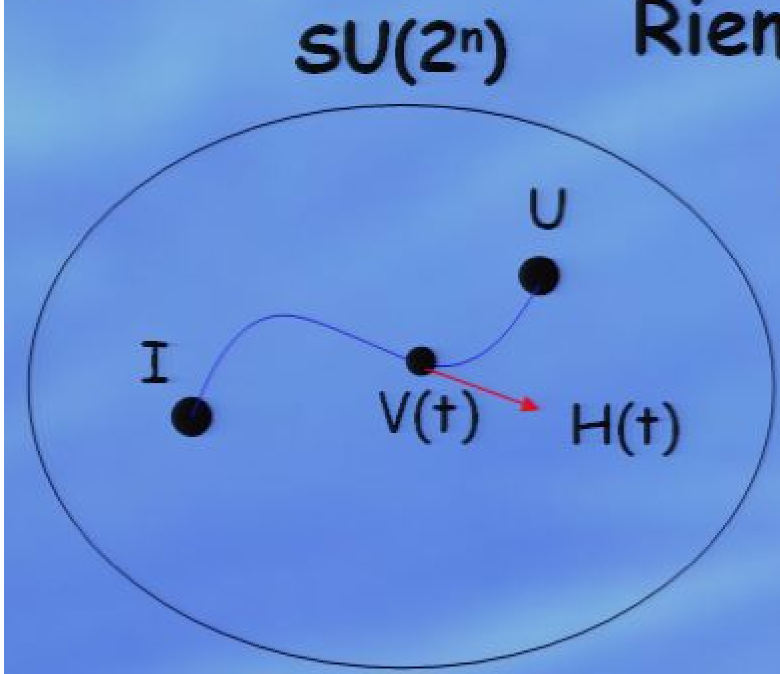
$$H(t) = \sum_\sigma h_\sigma(t)\,\sigma$$

$$V(0) = I; \qquad V(1) = U$$

The Hamiltonian H(t) generates "small displacements" near V(t).

Notation: σ = tensor product of the Pauli matrices or I.

Quantum computing and Riemannian geometry



We can think of U as being generated by an n-qubit time-dependent Hamiltonian:

$$H(t) = \sum_\sigma h_\sigma(t)\,\sigma$$

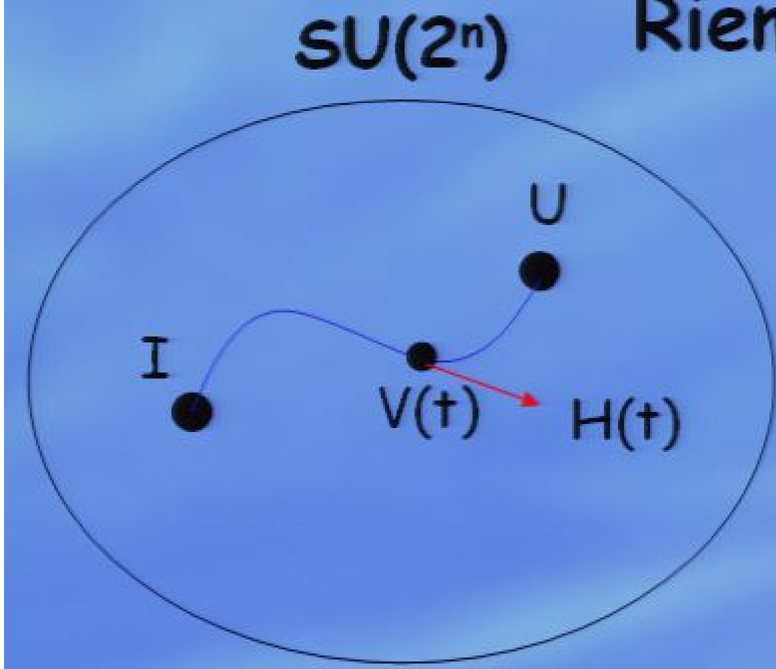
$$V(0) = I; \qquad V(1) = U$$

The Hamiltonian H(t) generates "small displacements" near V(t).

We define a metric: $c(V,H) \equiv \sqrt{\sum'_\sigma h_\sigma^2 + p^2 \sum''_\sigma h_\sigma^2}$

Notation: σ = tensor product of the Pauli matrices or I.

Quantum computing and Riemannian geometry



We can think of U as being generated by an n-qubit time-dependent Hamiltonian:

$$H(t) = \sum_\sigma h_\sigma(t)\,\sigma$$

$$V(0) = I; \qquad V(1) = U$$

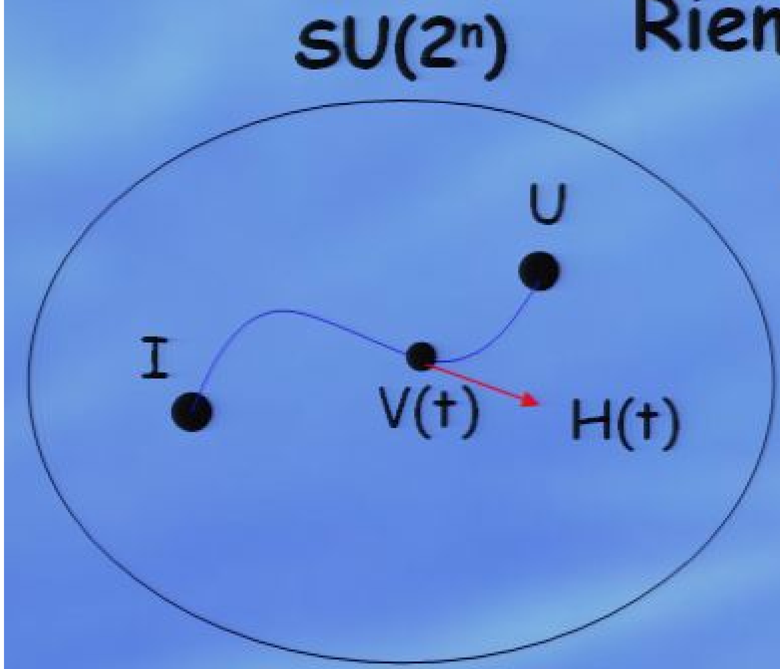
The Hamiltonian H(t) generates "small displacements" near V(t).

We define a metric: $c(V,H) \equiv \sqrt{\sum_\sigma' h_\sigma^2 + p^2 \sum_\sigma'' h_\sigma^2}$

Length of the curve V: $l(V) \equiv \int dt\; c(V(t),H(t))$.

Notation: σ = tensor product of the Pauli matrices or I.

Quantum computing and Riemannian geometry



We can think of U as being generated by an n-qubit time-dependent Hamiltonian:

$$H(t) = \sum_\sigma h_\sigma(t)\,\sigma$$

$$V(0) = I; \qquad V(1) = U$$

The Hamiltonian H(t) generates "small displacements" near V(t).

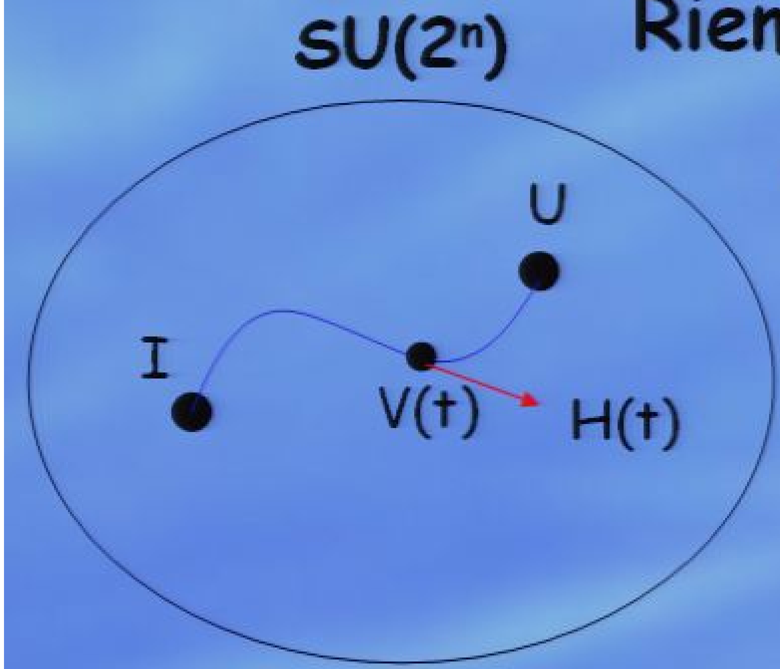
We define a **metric**: $c(V,H) \equiv \sqrt{\sum'_\sigma h_\sigma^2 + p^2 \sum''_\sigma h_\sigma^2}$

Length of the curve V: $l(V) \equiv \int dt\, c(V(t),H(t)).$

Distance between I and U: $d(I,U) \equiv \inf l(V)$

Notation: σ = tensor product of the Pauli matrices or I.

Quantum computing and Riemannian geometry



We can think of U as being generated by an n-qubit time-dependent Hamiltonian:

$$H(t) = \sum_\sigma h_\sigma(t)\,\sigma$$

$$V(0) = I; \qquad V(1) = U$$

The Hamiltonian H(t) generates "small displacements" near V(t).

We define a metric: $c(V,H) \equiv \sqrt{\sum'_\sigma h_\sigma^2 + p^2 \sum''_\sigma h_\sigma^2}$

Length of the curve V: $l(V) \equiv \int dt\; c(V(t),H(t))$.

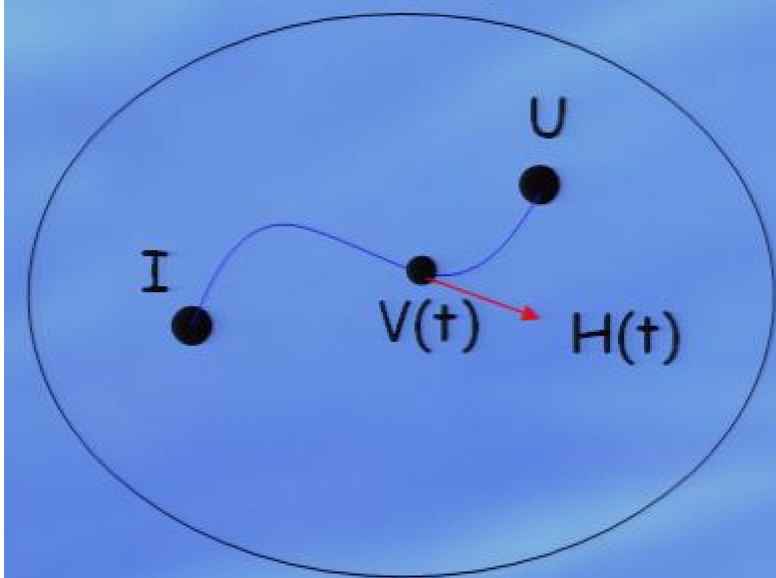
Distance between I and U: $d(I,U) \equiv \inf l(V)$

Notation: σ = tensor product of the Pauli matrices or I.

Actual Riemannian metric:
$\langle H,J \rangle \equiv tr(H\,P(J)) + p^2\, tr(H\,Q(J))$

Quantum computing and Riemannian geometry

SU(2^n)



We can think of U as being generated by an n-qubit time-dependent Hamiltonian:

$$H(t) = \sum_\sigma h_\sigma(t)\, \sigma$$

$$V(0) = I; \qquad V(1) = U$$

The Hamiltonian H(t) generates "small displacements" near V(t).

We define a **metric**: $c(V,H) \equiv \sqrt{\sum'_\sigma h_\sigma^2 + p^2 \sum''_\sigma h_\sigma^2}$

Length of the curve V: $l(V) \equiv \int dt\, c(V(t),H(t))$.

Distance between I and U: $d(I,U) \equiv \inf l(V)$

Notation: σ = tensor product of the Pauli matrices or I.

Actual Riemannian metric:
$\langle H,J \rangle \equiv tr(H\, P(J)) + p^2\, tr(H\, Q(J))$

P(XII+ZZI+XYZ) = XII+ZZI

Quantum computing and Riemannian geometry

Claim: For any $U \in SU(2^n)$:

(a) The minimal number of gates required to exactly synthesize U satisfies:

$$d(I,U) \leq m(U)$$

(b) We can synthesize U to high accuracy using a number of quantum gates polynomial in $d(I,U)$.

Quantum computing and Riemannian geometry

Motivating idea: When we think about finding optimal circuits, we are usually thinking in terms of optimizing over a *discrete* space.

Quantum computing and Riemannian geometry

Motivating idea: When we think about finding optimal circuits, we are usually thinking in terms of optimizing over a *discrete* space.

It's much better to minimize *smooth* functions over smooth spaces: we can use calculus!

Quantum computing and Riemannian geometry

Motivating idea: When we think about finding optimal circuits, we are usually thinking in terms of optimizing over a *discrete* space.

It's much better to minimize *smooth* functions over smooth spaces: we can use calculus!

⇒ If we can work on a Riemannian manifold this leads to the geodesic equation.

$$\frac{d^2x^\alpha}{dt^2} + \sum_{\beta\gamma} \Gamma^\alpha_{\beta\gamma} \frac{dx^\beta}{dt}\frac{dx^\gamma}{dt} = 0.$$

Quantum computing and Riemannian geometry

Motivating idea: When we think about finding optimal circuits, we are usually thinking in terms of optimizing over a *discrete* space.

It's much better to minimize *smooth* functions over smooth spaces: we can use calculus!

If we can work on a Riemannian manifold this leads to the geodesic equation.

$$\frac{d^2x^\alpha}{dt^2} + \sum_{\beta\gamma} \Gamma^\alpha_{\beta\gamma} \frac{dx^\beta}{dt}\frac{dx^\gamma}{dt} = 0.$$

The geodesic equation is a second order ODE, so given an initial position and velocity, subsequent motion is **completely determined** by the geometry.

Quantum computing and Riemannian geometry

Motivating idea: When we think about finding optimal circuits, we are usually thinking in terms of optimizing over a *discrete* space.

It's much better to minimize *smooth* functions over smooth spaces: we can use calculus!

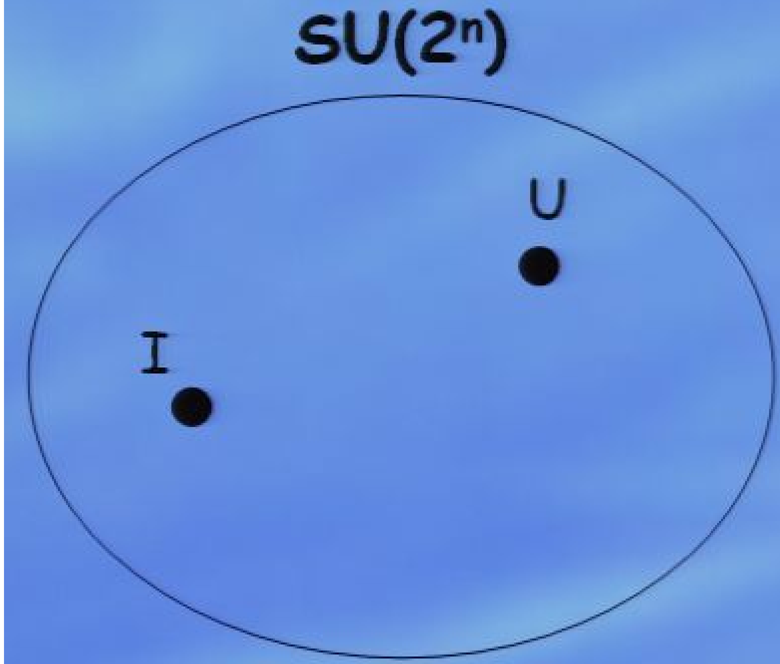
If we can work on a Riemannian manifold this leads to the geodesic equation.

$$\frac{d^2x^\alpha}{dt^2} + \sum_{\beta\gamma} \Gamma^\alpha_{\beta\gamma} \frac{dx^\beta}{dt}\frac{dx^\gamma}{dt} = 0.$$

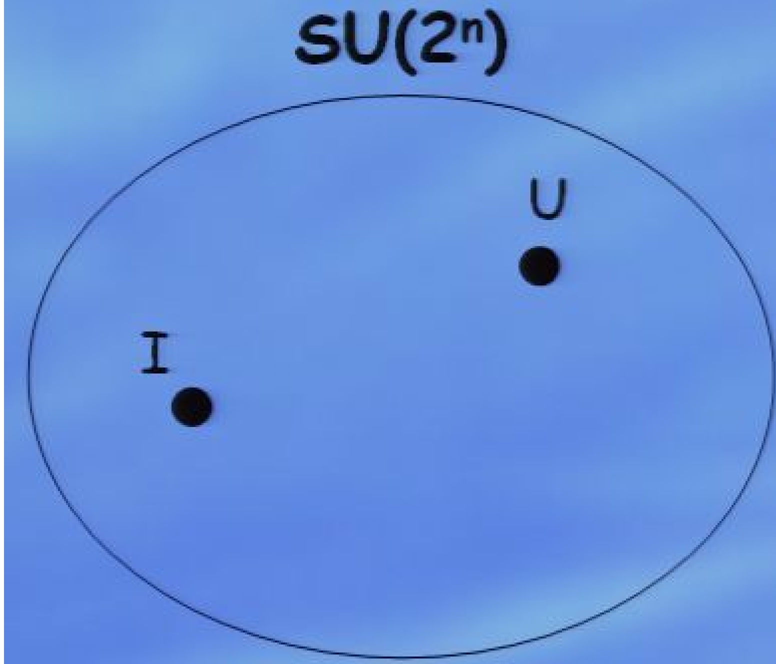
The geodesic equation is a second order ODE, so given an initial position and velocity, subsequent motion is **completely determined** by the geometry.

Compare, e.g., with being given part of an optimal circuit.

$$d(I, U) \leq m(U)$$

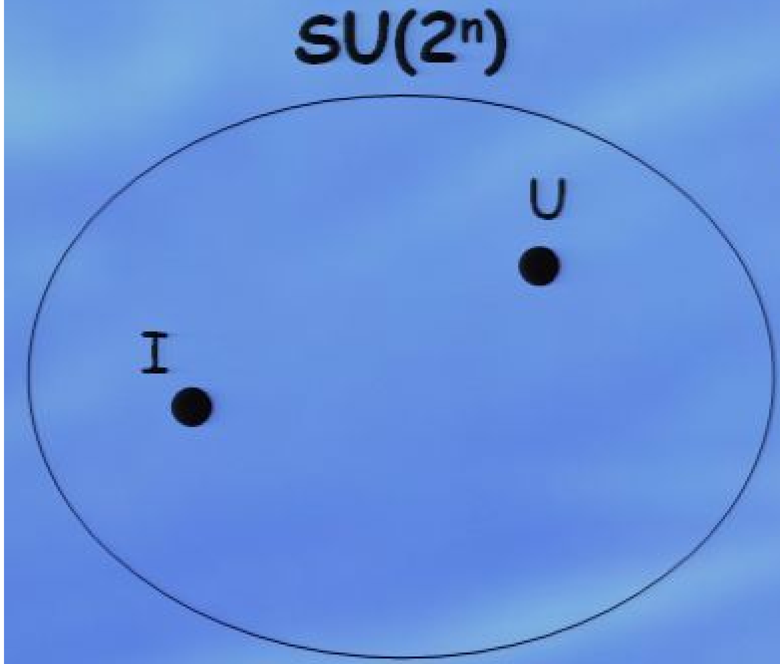


$d(I,U) \leq m(U)$



Suppose we use gates of the form $\exp(-i\alpha\,\sigma)$ as our gate set, where $|\alpha| \leq 1$.

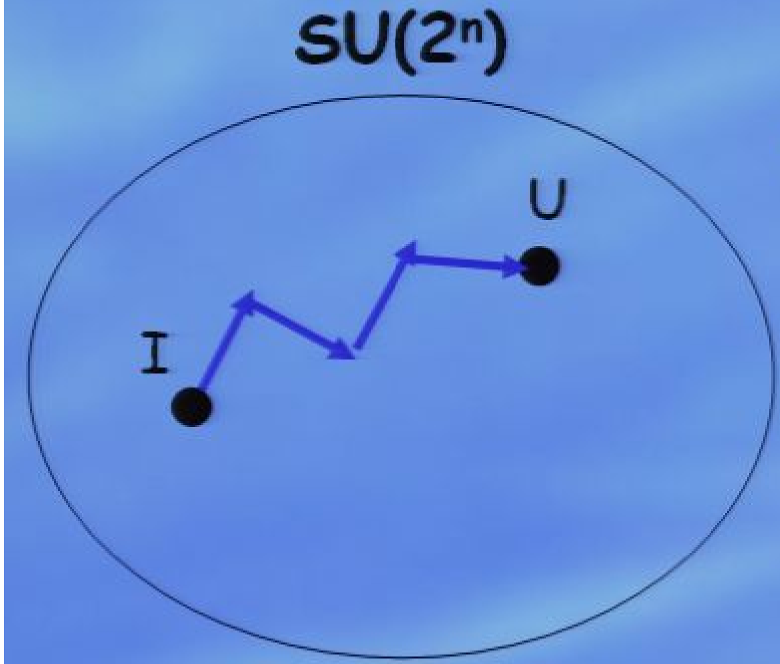
$d(I,U) \leq m(U)$



Suppose we use gates of the form $\exp(-i\alpha\,\sigma)$ as our gate set, where $|\alpha| \leq 1$.

Let $m(U)$ be the minimal number of gates of this form necessary to build up U.

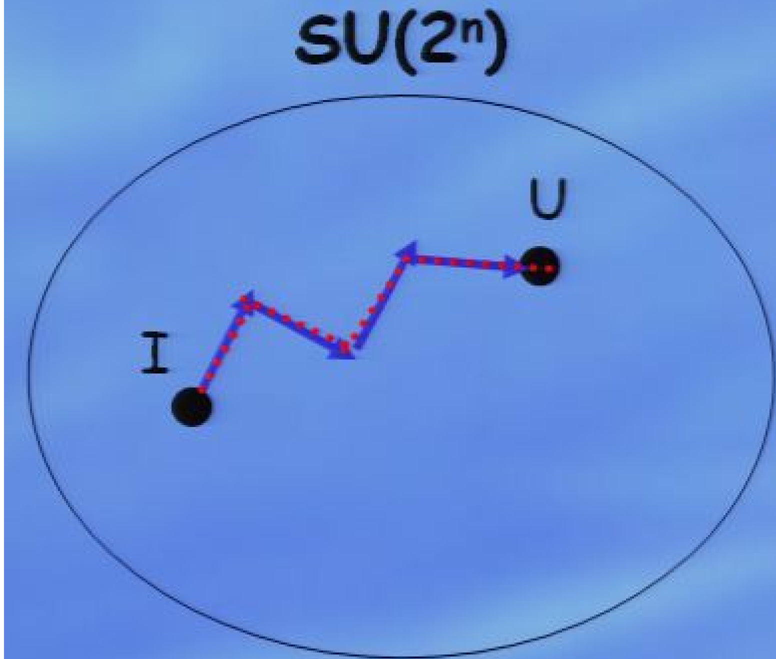
$d(I,U) \leq m(U)$



Suppose we use gates of the form $\exp(-i\alpha\,\sigma)$ as our gate set, where $|\alpha| \leq 1$.

Let $m(U)$ be the minimal number of gates of this form necessary to build up U.

$d(I,U) \leq m(U)$

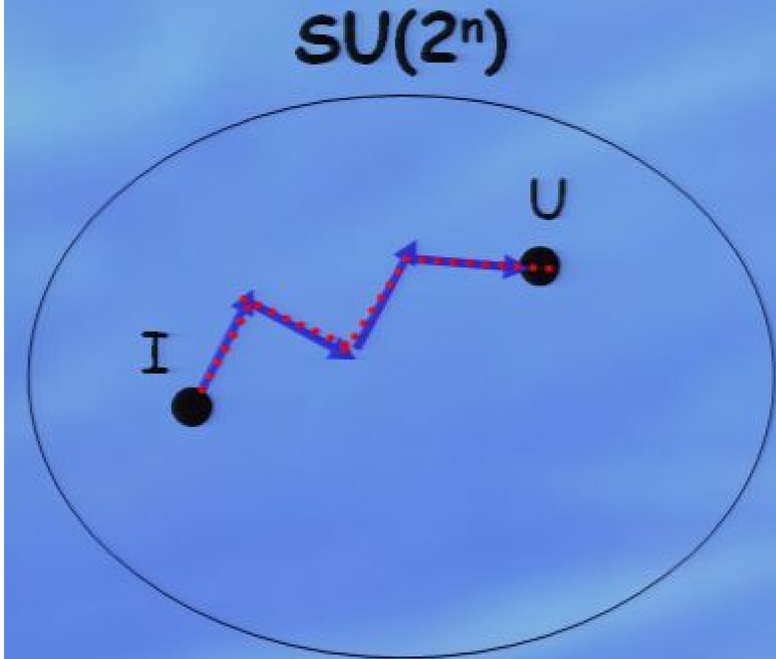


Suppose we use gates of the form $\exp(-i\alpha\,\sigma)$ as our gate set, where $|\alpha| \leq 1$.

Let m(U) be the minimal number of gates of this form necessary to build up U.

We can construct a control function h(t) which replicates this path.

$d(I,U) \leq m(U)$

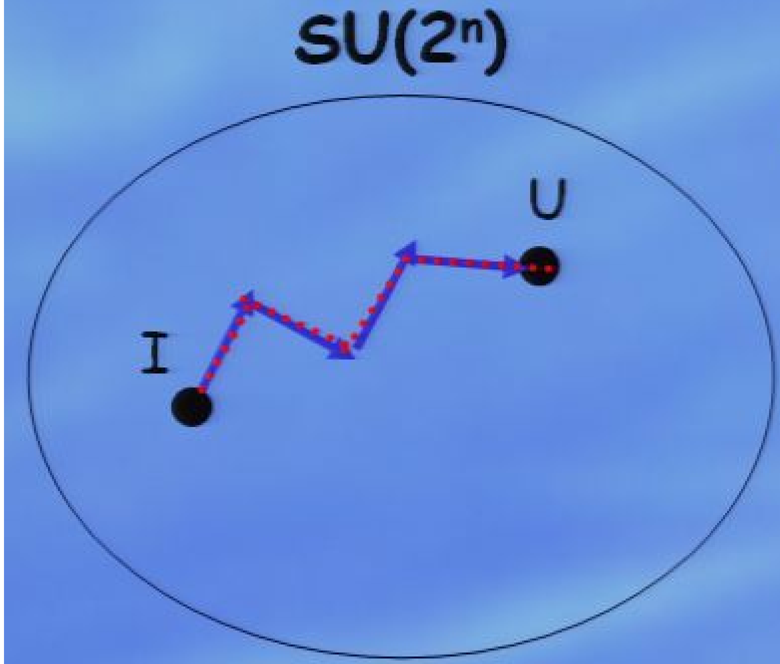


Suppose we use gates of the form $\exp(-i\alpha\,\sigma)$ as our gate set, where $|\alpha| \leq 1$.

Let m(U) be the minimal number of gates of this form necessary to build up U.

We can construct a control function h(t) which replicates this path.

$d(I,U) \leq m(U)$



Suppose we use gates of the form $\exp(-i\alpha\,\sigma)$ as our gate set, where $|\alpha| \leq 1$.

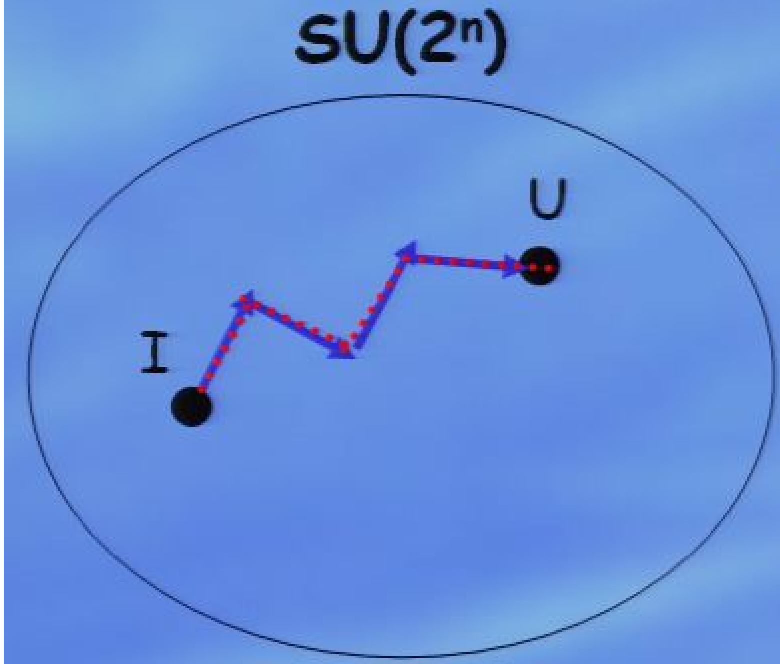
Let m(U) be the minimal number of gates of this form necessary to build up U.

We can construct a control function h(t) which replicates this path.

The distance along the induced path is:

$$|\alpha_1|+|\alpha_2|+\ldots+|\alpha_{m(U)}| \leq m(U)$$

$d(I,U) \le m(U)$



Suppose we use gates of the form $\exp(-i\alpha\,\sigma)$ as our gate set, where $|\alpha| \le 1$.

Let m(U) be the minimal number of gates of this form necessary to build up U.

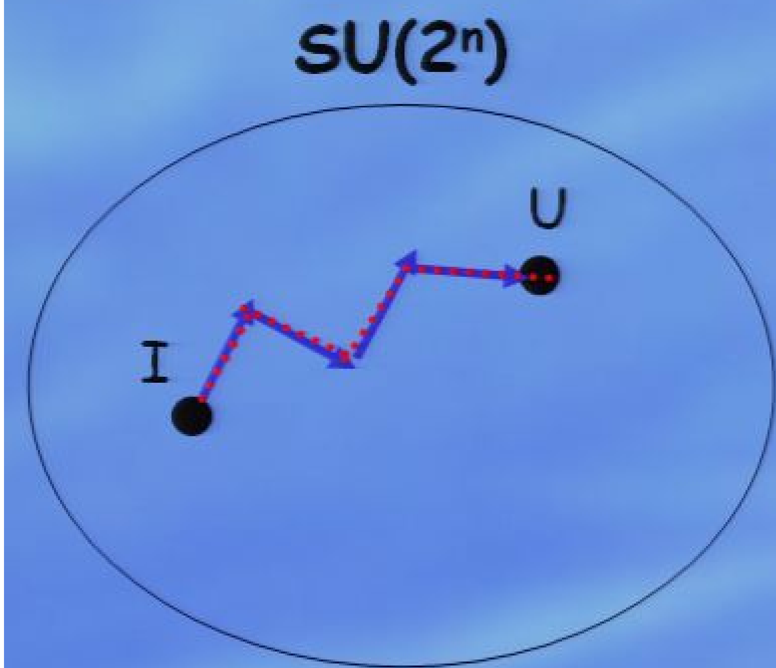
We can construct a control function h(t) which replicates this path.

The distance along the induced path is:

$$|\alpha_1|+|\alpha_2|+\ldots+|\alpha_{m(U)}| \le m(U)$$

It follows that: $d(I,U) \le m(U)$

$d(I,U) \leq m(U)$



Suppose we use gates of the form $\exp(-i\alpha\,\sigma)$ as our gate set, where $|\alpha| \leq 1$.

Let $m(U)$ be the minimal number of gates of this form necessary to build up U.

We can construct a control function h(t) which replicates this path.

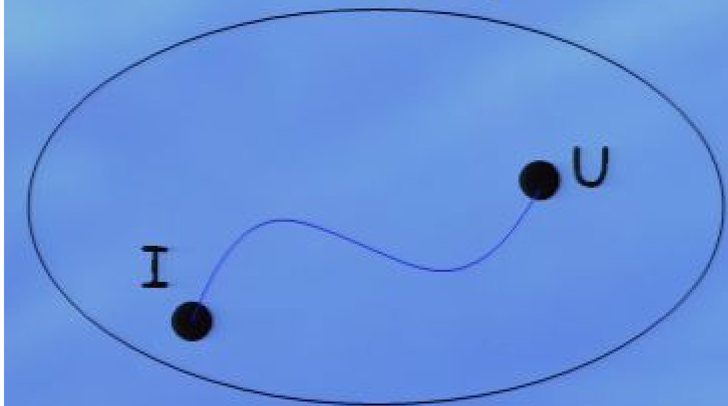
The distance along the induced path is:

$$|\alpha_1|+|\alpha_2|+\ldots+|\alpha_{m(U)}| \leq m(U)$$

It follows that: $d(I,U) \leq m(U)$

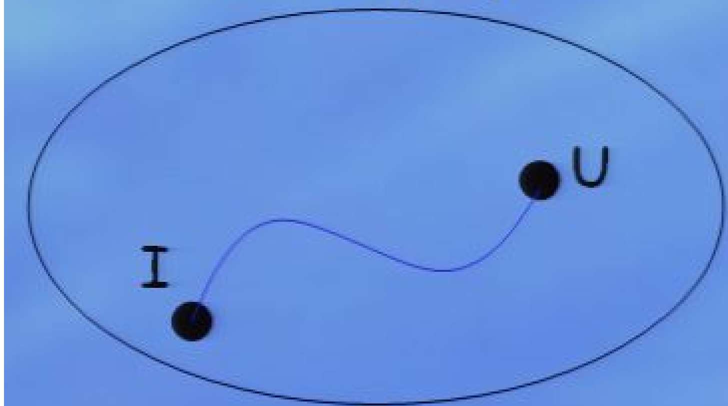
The upper bound: $m(U') \leq \text{poly}(d(I,U))$

$SU(2^n)$



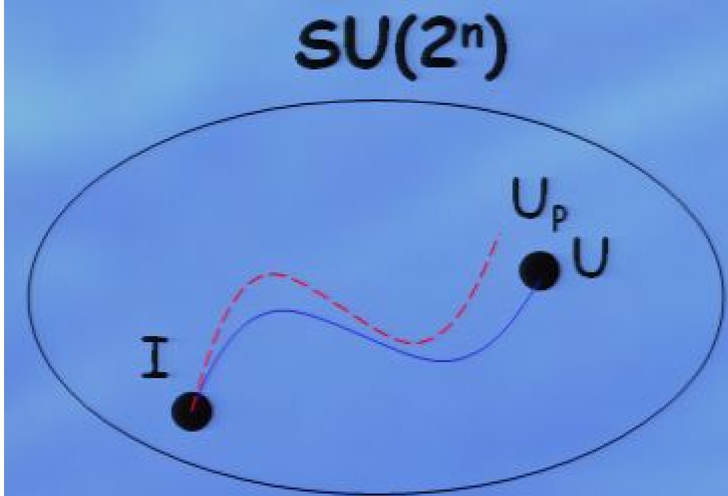
The upper bound: $m(U') \leq poly(d(I,U))$

$SU(2^n)$



Idea 1: Let $H_p(t)$ be the "projected Hamiltonian" obtained by removing all 3- and more-qubit terms from $H(t)$.

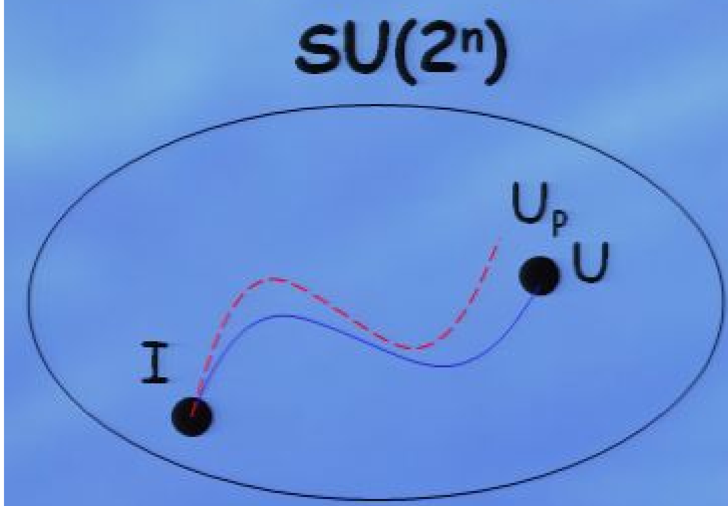
The upper bound: $m(U') \leq \text{poly}(d(I,U))$



Idea 1: Let $H_p(t)$ be the "projected Hamiltonian" obtained by removing all 3- and more-qubit terms from $H(t)$.

By making the penalty p in the metric very large, we can ensure that U_p is as close to U as desired.

The upper bound: $m(U') \leq poly(d(I,U))$



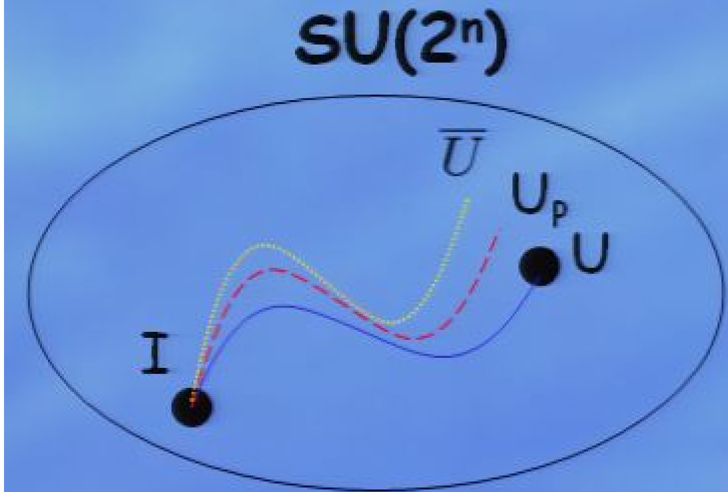
Idea 1: Let $H_p(t)$ be the "projected Hamiltonian" obtained by removing all 3- and more-qubit terms from $H(t)$.

By making the penalty p in the metric very large, we can ensure that U_p is as close to U as desired.

Idea 2: Break the path to U_p up into small intervals of size Δ. Then approximate evolution by $H_p(t)$ over that interval by evolution due to the "average" Hamiltonian:

$$\overline{H} = \frac{1}{\Delta}\int_{t_0}^{t_0+\Delta} dt\, H_P(t)$$

The upper bound: $m(U') \leq poly(d(I,U))$



Idea 1: Let $H_p(t)$ be the "projected Hamiltonian" obtained by removing all 3- and more-qubit terms from H(t).

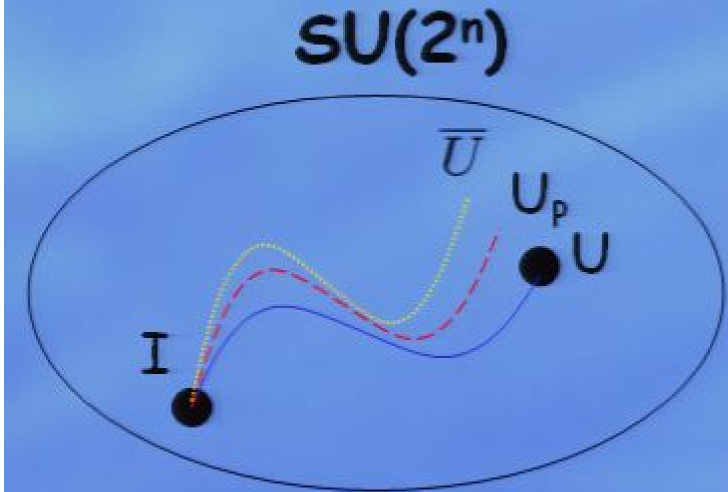
By making the penalty p in the metric very large, we can ensure that U_p is as close to U as desired.

Idea 2: Break the path to U_p up into small intervals of size Δ. Then approximate evolution by $H_p(t)$ over that interval by evolution due to the "average" Hamiltonian:

$$\overline{H} = \frac{1}{\Delta} \int_{t_0}^{t_0+\Delta} dt\, H_P(t)$$

Key point: To get a good approximation to U_p, we need only choose Δ polynomially small. Proved using Dyson formula.

The upper bound: $m(U') \leq poly(d(I,U))$



Idea 1: Let $H_P(t)$ be the "projected Hamiltonian" obtained by removing all 3- and more-qubit terms from $H(t)$.

By making the penalty p in the metric very large, we can ensure that U_P is as close to U as desired.

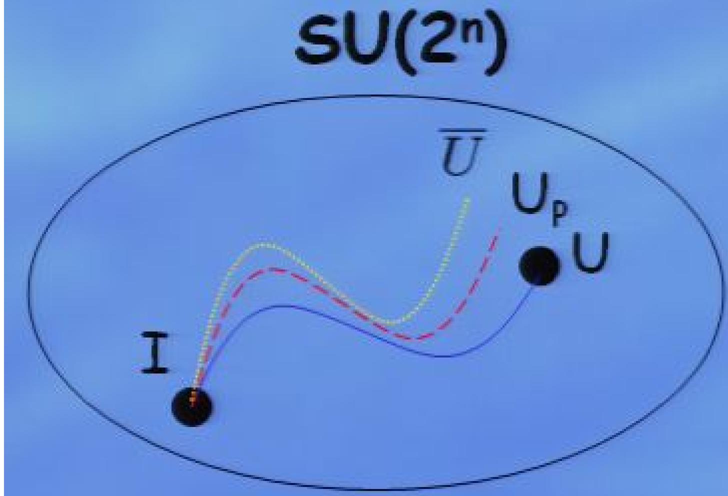
Idea 2: Break the path to U_P up into small intervals of size Δ. Then approximate evolution by $H_P(t)$ over that interval by evolution due to the "average" Hamiltonian:

$$\overline{H} = \frac{1}{\Delta}\int_{t_0}^{t_0+\Delta} dt\, H_P(t)$$

Key point: To get a good approximation to U_P, we need only choose Δ polynomially small. Proved using Dyson formula.

Idea 3: Now simulate the average Hamiltonian over a time Δ using standard quantum computing techniques.

The upper bound: $m(U') \leq poly(d(I,U))$



Idea 1: Let $H_p(t)$ be the "projected Hamiltonian" obtained by removing all 3- and more-qubit terms from $H(t)$.

By making the penalty p in the metric very large, we can ensure that U_p is as close to U as desired.

Idea 2: Break the path to U_p up into small intervals of size Δ. Then approximate evolution by $H_p(t)$ over that interval by evolution due to the "average" Hamiltonian:

$$\overline{H} = \frac{1}{\Delta} \int_{t_0}^{t_0+\Delta} dt\, H_P(t)$$

Key point: To get a good approximation to U_p, we need only choose Δ polynomially small. Proved using Dyson formula.

Idea 3: Now simulate the average Hamiltonian over a time Δ using standard quantum computing techniques.

Easy to do accurately since (a) the average Hamiltonian is two-body, and (b) it has bounded strength.

The geodesic equation

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K\rangle = \mathrm{tr}(L\ K)$

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K\rangle = \mathrm{tr}(L\,K)$

$$L = P(H) + p^{-2}\,Q(H)$$

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K\rangle = \mathrm{tr}(L\,K)$

$L = P(H) + p^{-2}\, Q(H)$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K \rangle = tr(L\ K)$

$$L = P(H) + p^{-2}\,Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

$\dot{M} = i[M, P(M)]$ where $M = -2^n(1-p^{-2})L$

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K \rangle = tr(L\ K)$

$$L = P(H) + p^{-2}\, Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

$\dot{M} = i[M, P(M)]$ where $M = -2^n(1-p^{-2})L$

This is a type of Lax equation.

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K \rangle = tr(L\,K)$

$$L = P(H) + p^{-2}\,Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

$\dot{M} = i[M, P(M)]$ where $M = -2^n(1-p^{-2})L$

This is a type of Lax equation.

Simplest possible equation involving both the commutator and the projection P.
Independent of the penalty parameter p.

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K\rangle = tr(L\ K)$

$$L = P(H) + p^{-2}\, Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L,P(L)]$

$$\dot{M} = i[M,P(M)] \text{ where } M = -2^n(1-p^{-2})L$$

This is a type of Lax equation.

Simplest possible equation involving both the commutator and the projection P.
Independent of the penalty parameter p.

By substitution into the Lax equation we can verify that

$$L(t) = U(t)^{+}\, L(0)\, U(t)$$

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K \rangle = tr(L\ K)$

$$L = P(H) + p^{-2}\, Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

$\dot{M} = i[M, P(M)]$ where $M = -2^n(1-p^{-2})L$

This is a type of Lax equation.

Simplest possible equation involving both the commutator and the projection P.
Independent of the penalty parameter p.

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K\rangle = tr(L\,K)$

$$L = P(H) + p^{-2}\,Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

$$\dot{M} = i[M, P(M)] \text{ where } M = -2^n(1-p^{-2})L$$

This is a type of Lax equation.

Simplest possible equation involving both the commutator and the projection P.
Independent of the penalty parameter p.

By substitution into the Lax equation we can verify that

$$L(t) = U(t)^{+}\,L(0)\,U(t)$$

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K \rangle = \mathrm{tr}(L\,K)$

$$L = P(H) + p^{-2}\,Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

$\dot{M} = i[M, P(M)]$ where $M = -2^n(1-p^{-2})L$

This is a type of Lax equation.

Simplest possible equation involving both the commutator and the projection P.
Independent of the penalty parameter p.

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K\rangle = tr(L\ K)$

$$L = P(H) + p^{-2}\, Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

$$\dot{M} = i[M, P(M)] \text{ where } M = -2^n(1-p^{-2})L$$

This is a type of Lax equation.

Simplest possible equation involving both the commutator and the projection P.
Independent of the penalty parameter p.

By substitution into the Lax equation we can verify that

$$L(t) = U(t)^{+}\, L(0)\, U(t)$$

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K\rangle = \mathrm{tr}(L\,K)$

$$L = P(H) + p^{-2}\,Q(H)$$

Geodesic equation: $\dot{L} = -i2^n(1-p^{-2})[L, P(L)]$

$\dot{M} = i[M, P(M)]$ where $M = -2^n(1-p^{-2})L$

This is a type of Lax equation.

Simplest possible equation involving both the commutator and the projection P.
Independent of the penalty parameter p.

By substitution into the Lax equation we can verify that

$$L(t) = U(t)^{+}\,L(0)\,U(t)$$

Thus $U(t)\,L(t)\,U(t)^{+} = L(0)$ is a constant of the motion.

The geodesic equation

Simplest form is in terms of the "dual" L to H: $\langle H,K\rangle = \mathrm{tr}(L\,K)$

$$L = P(H) + p^{-2}\,Q(H)$$

$$\text{Geodesic equation: } \dot{L} = -i2^n(1-p^{-2})[L, P(L)]$$

$$\dot{M} = i[M, P(M)] \text{ where } M = -2^n(1-p^{-2})L$$

This is a type of Lax equation.

Simplest possible equation involving both the commutator and the projection P.
Independent of the penalty parameter p.

By substitution into the Lax equation we can verify that

$$L(t) = U(t)^{+}\,L(0)\,U(t)$$

Thus $U(t)\,L(t)\,U(t)^{+} = L(0)$ is a constant of the motion.

⇒ The system is integrable.

Big picture

What we've learnt

d(I,U) $\leq$ m(U) and m(U') $\leq$ d(I,U) for some U' $\approx$ U.

Geodesic eqtn is a Lax equation: $\dot{M} = i[M, P(M)]$

U(t) M(t) U(t)$^{+}$ is a complete set of constants of the motion.

Big picture

What we've learnt

$d(I,U) \leq m(U)$ and $m(U') \leq d(I,U)$ for some $U' \approx U$.

Geodesic eqtn is a Lax equation: $\dot{M} = i[M, P(M)]$

$U(t)\, M(t)\, U(t)^{+}$ is a complete set of constants of the motion.

What can be done?

Geometry provides a lens to re-examine the problems of quantum computing.

Big picture

What we've learnt

d(I,U) $\leq$ m(U) and m(U') $\leq$ d(I,U) for some U' $\approx$ U.

Geodesic eqtn is a Lax equation: $\dot{M} = i[M, P(M)]$

$U(t)\, M(t)\, U(t)^{+}$ is a complete set of constants of the motion.

What can be done?

Geometry provides a lens to re-examine the problems of quantum computing.

Study solutions to the geodesic equation, analytically and numerically.

Big picture

What we've learnt

d(I,U) $\leq$ m(U) and m(U′) $\leq$ d(I,U) for some U′ $\approx$ U.

Geodesic eqtn is a Lax equation: $\dot{M} = i[M, P(M)]$

U(t) M(t) U(t)$^{+}$ is a complete set of constants of the motion.

What can be done?

Geometry provides a lens to re-examine the problems of quantum computing.

Study solutions to the geodesic equation, analytically and numerically.

Use theory of conjugate points to study when geodesics are minimizing.

Big picture

What we've learnt

$d(I,U) \leq m(U)$ and $m(U') \leq d(I,U)$ for some $U' \approx U$.

Geodesic eqtn is a Lax equation: $\dot{M} = i[M, P(M)]$

$U(t)\, M(t)\, U(t)^{+}$ is a complete set of constants of the motion.

What can be done?

Geometry provides a lens to re-examine the problems of quantum computing.

Study solutions to the geodesic equation, analytically and numerically.

Use theory of conjugate points to study when geodesics are minimizing.

Study quantum algorithms (e.g. factoring) using geometry.

Big picture

What we've learnt

$d(I,U) \leq m(U)$ and $m(U') \leq d(I,U)$ for some $U' \approx U$.

Geodesic eqtn is a Lax equation: $\dot{M} = i[M, P(M)]$

$U(t)\, M(t)\, U(t)^{+}$ is a complete set of constants of the motion.

What can be done?

Geometry provides a lens to re-examine the problems of quantum computing.

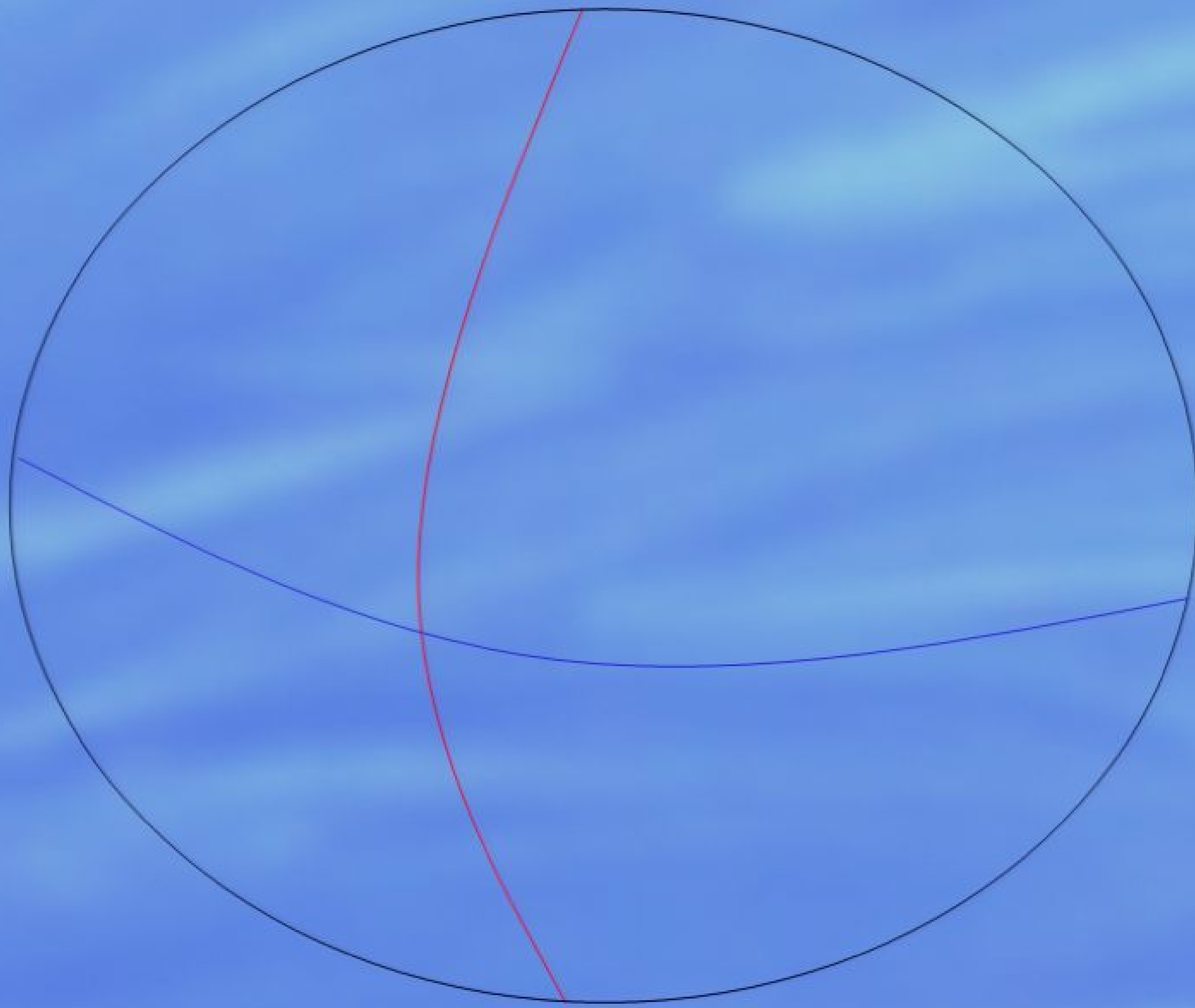
Study solutions to the geodesic equation, analytically and numerically.

Use theory of conjugate points to study when geodesics are minimizing.

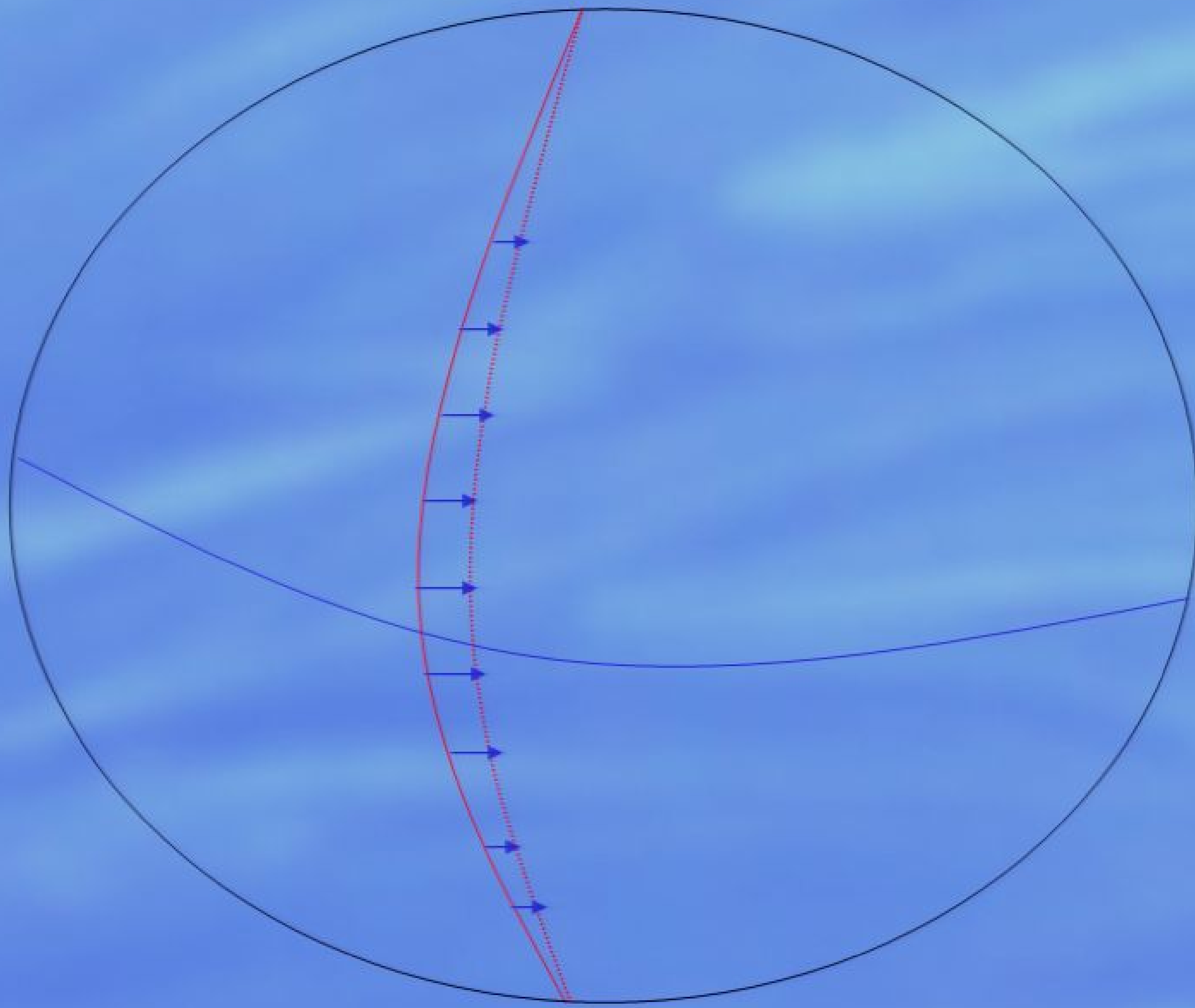
Study quantum algorithms (e.g. factoring) using geometry.

Study impact of ancilla on quantum complexity

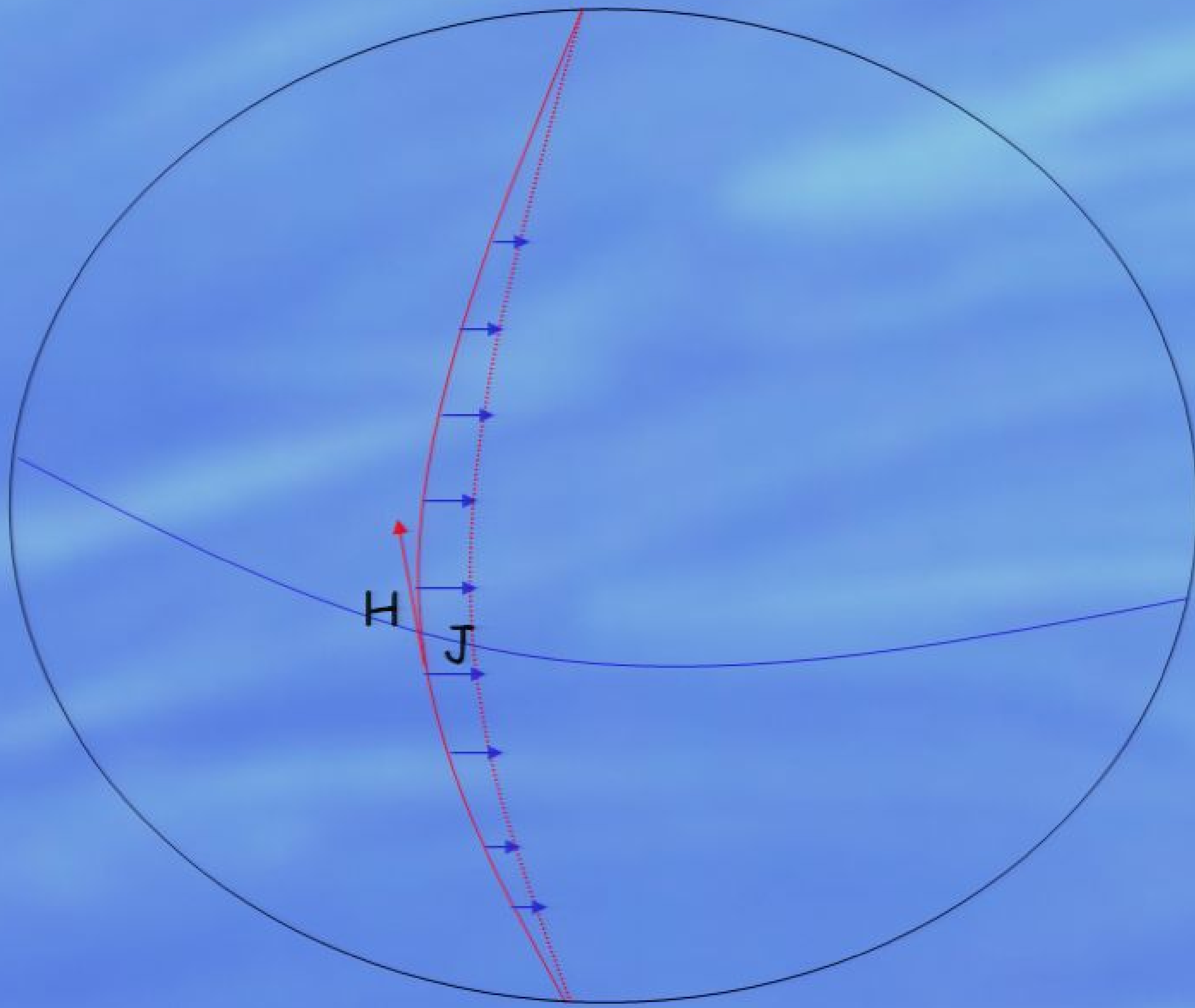
Conjugate points



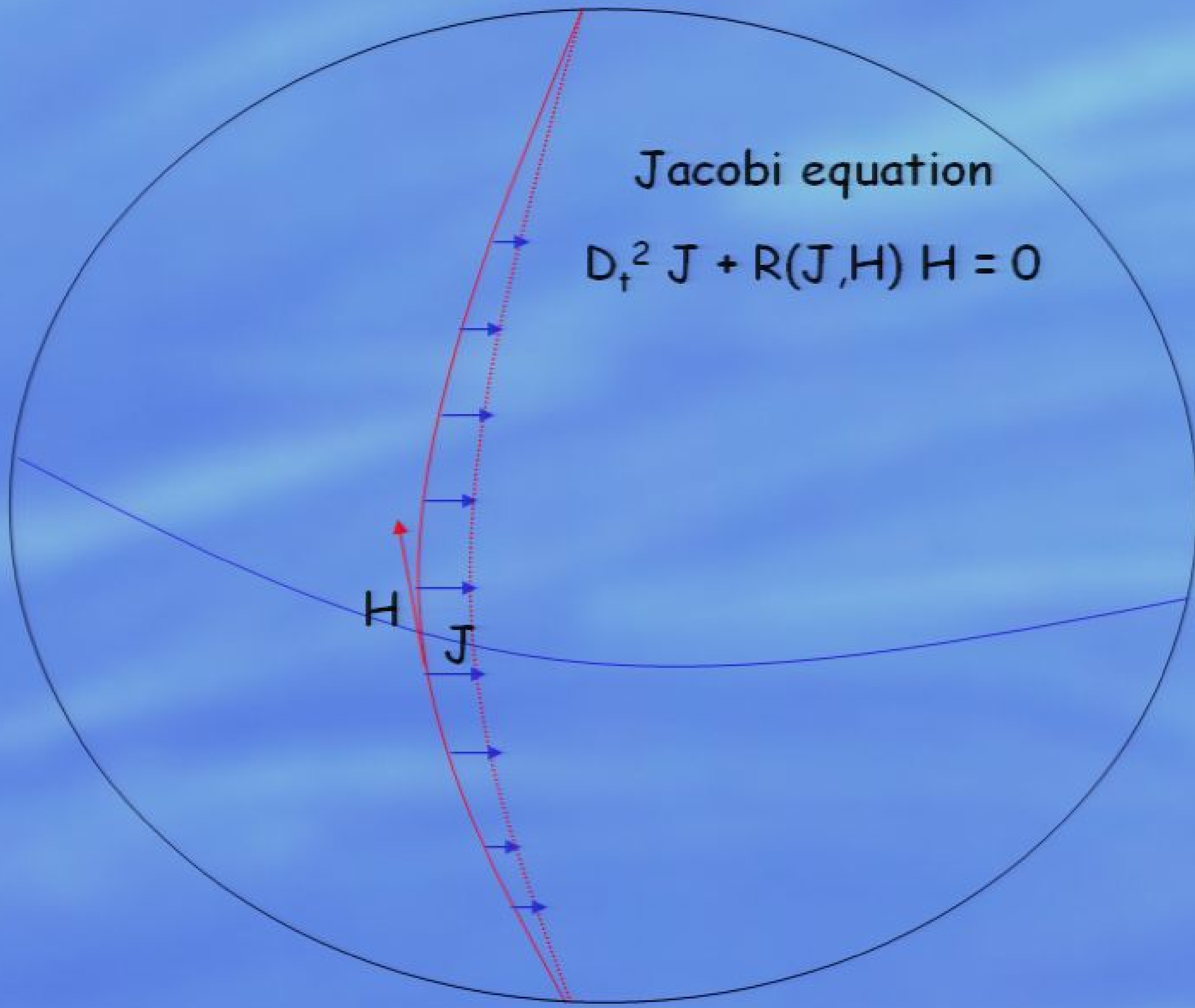
Conjugate points



Conjugate points



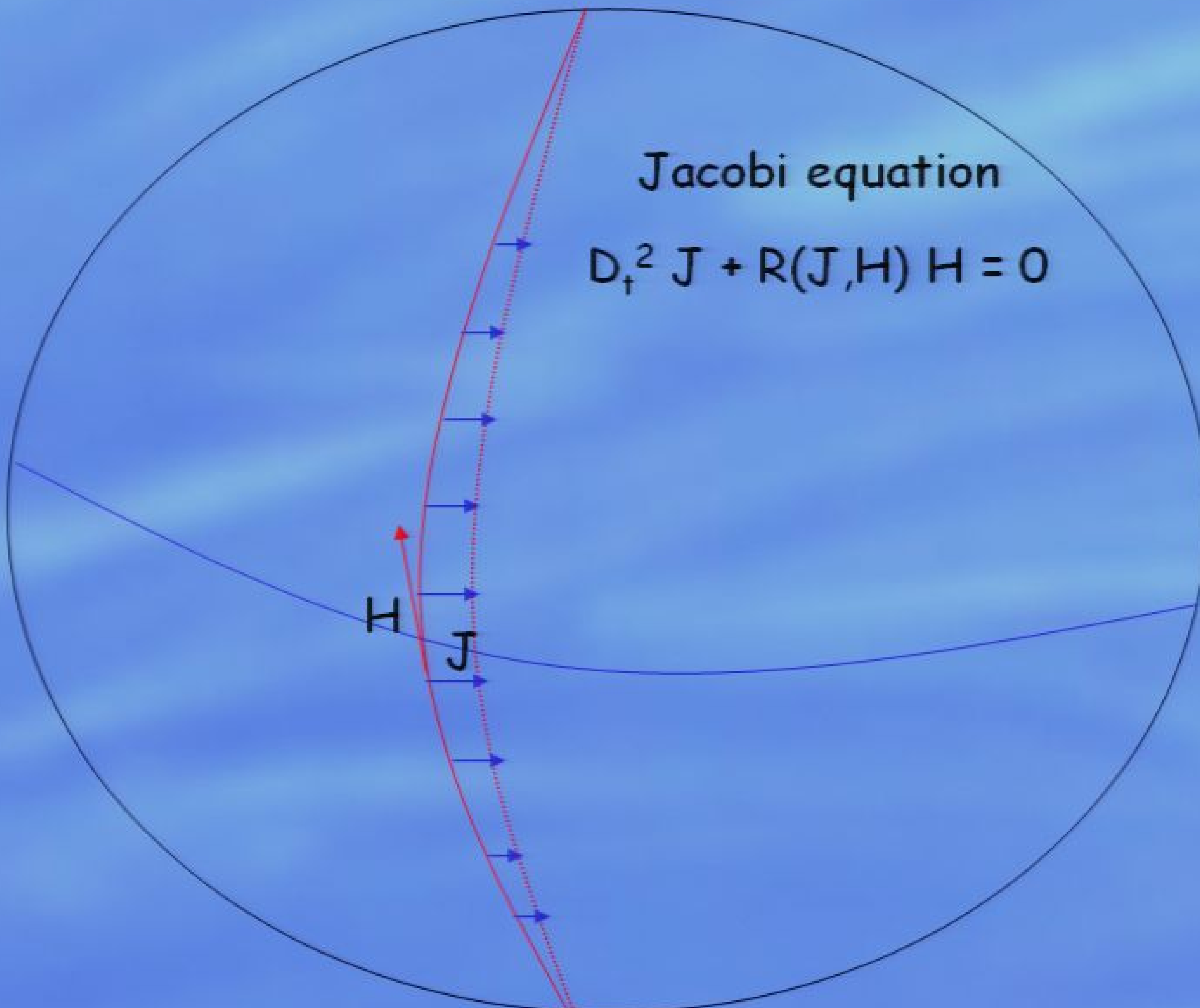
Conjugate points



Jacobi equation

$$D_t^2 J + R(J,H) H = 0$$

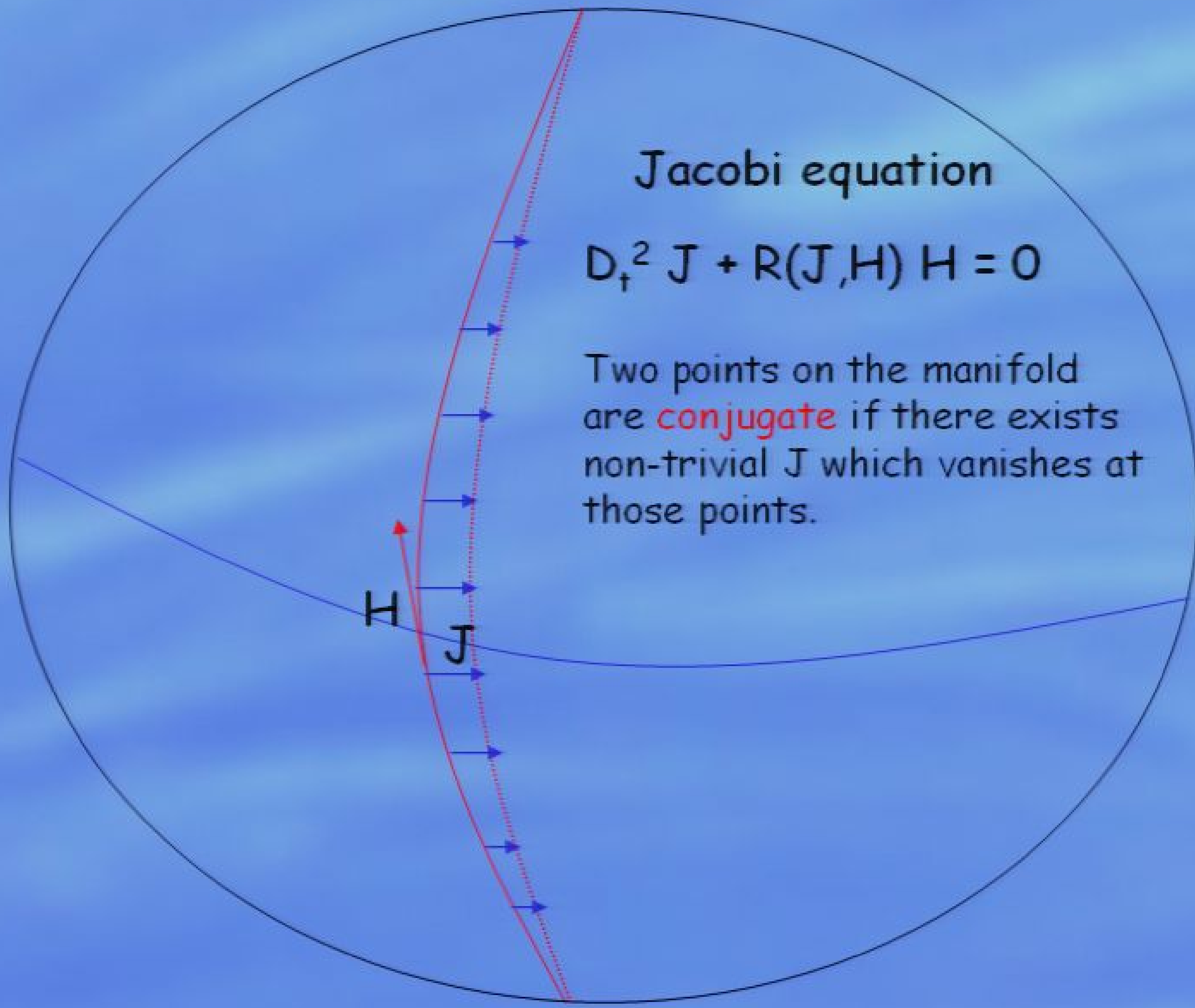
Conjugate points



Jacobi equation

$$D_t^2 J + R(J,H)\,H = 0$$

Conjugate points

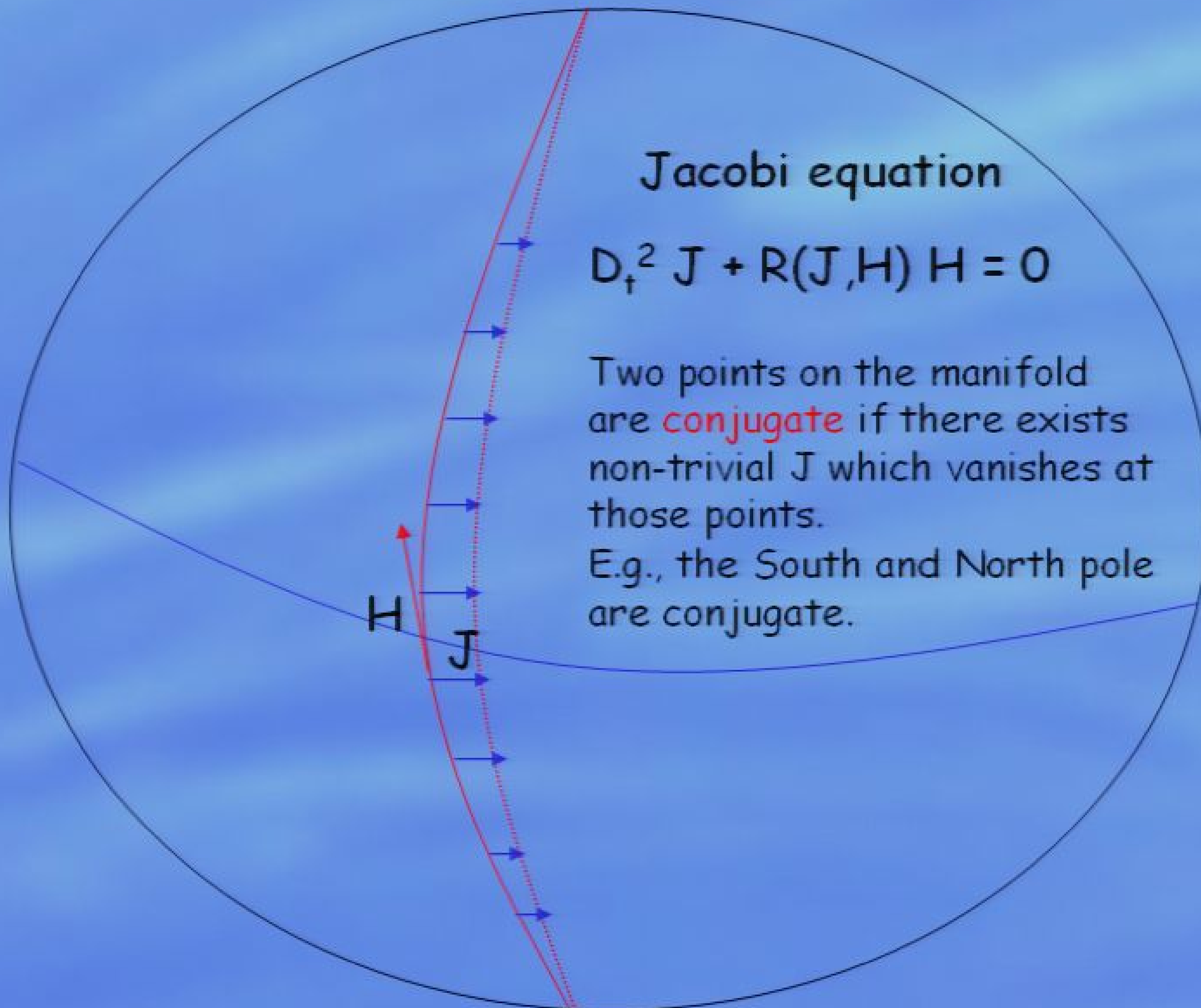


Jacobi equation

$$D_t^2 J + R(J,H)\, H = 0$$

Two points on the manifold are conjugate if there exists non-trivial J which vanishes at those points.

Conjugate points

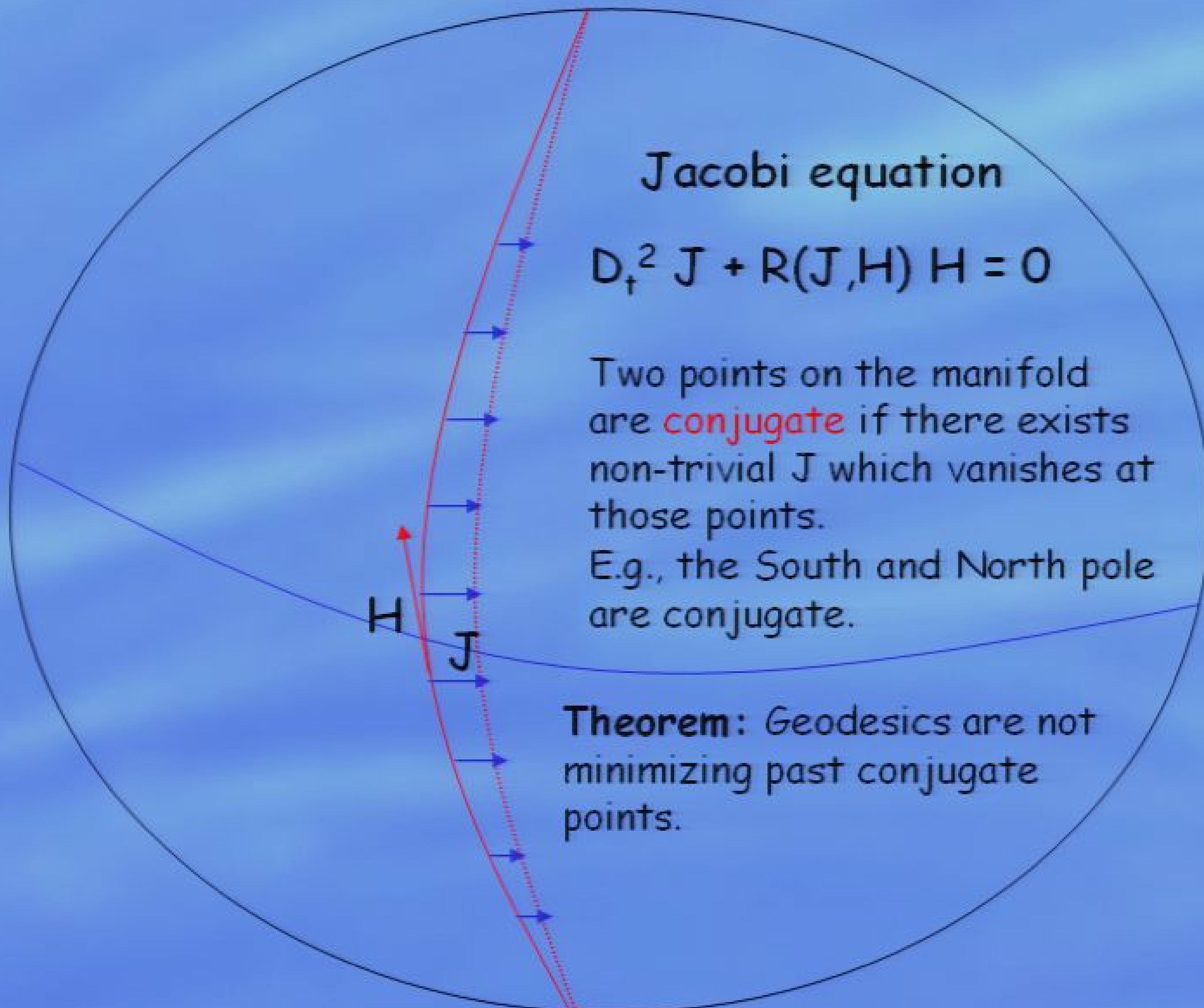


Jacobi equation

$$D_t^2 J + R(J,H)\,H = 0$$

Two points on the manifold are conjugate if there exists non-trivial J which vanishes at those points.
E.g., the South and North pole are conjugate.

Conjugate points



Jacobi equation

$$D_t^2 J + R(J,H)\,H = 0$$

Two points on the manifold are conjugate if there exists non-trivial J which vanishes at those points.
E.g., the South and North pole are conjugate.

Theorem: Geodesics are not minimizing past conjugate points.

Conjugate points

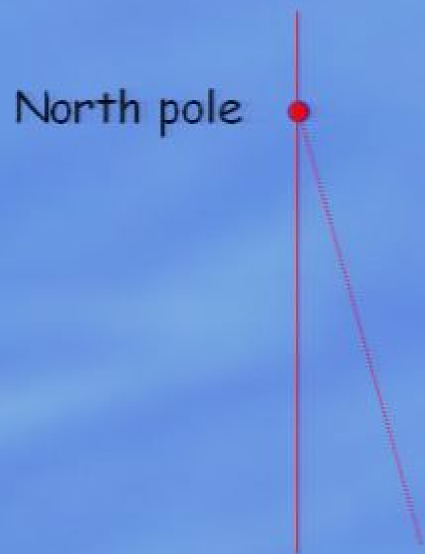
Theorem: Geodesics are not minimizing past conjugate points.

Conjugate points

Theorem: *Geodesics are not minimizing past conjugate points.*

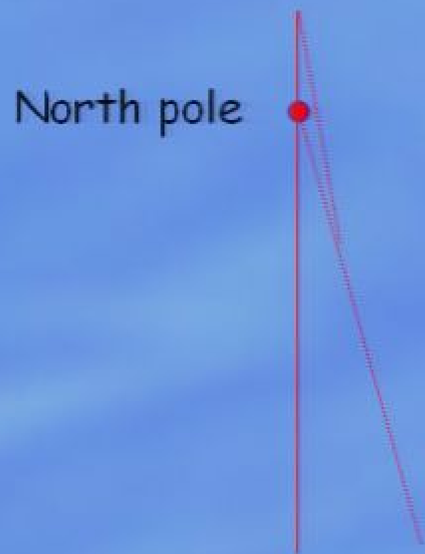
Conjugate points

Theorem: *Geodesics are not minimizing past conjugate points.*



Conjugate points

Theorem: *Geodesics are not minimizing past conjugate points.*



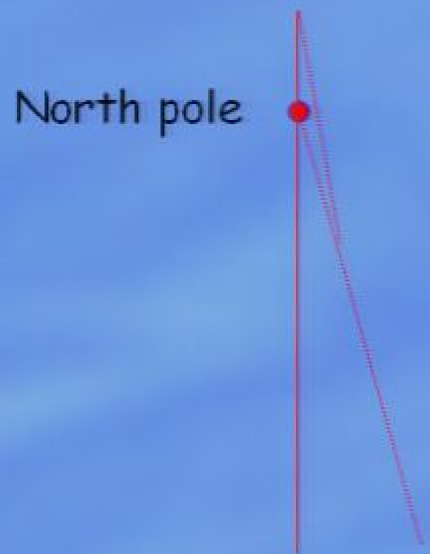
Conjugate points

Theorem: *Geodesics are not minimizing past conjugate points.*

The geodesic is no longer a local minimum, but merely a local extremum.

Conjugate points

Theorem: *Geodesics are not minimizing past conjugate points.*



The geodesic is no longer a local minimum, but merely a local extremum.

Therefore, the global minimum must be elsewhere.

Application of conjugate points

Application of conjugate points

Suppose H = sum of 1- and 2-body terms.

Then M = sum of 1- and 2-body terms.

Application of conjugate points

Suppose H = sum of 1- and 2-body terms.

Then M = sum of 1- and 2-body terms.

$$\dot{M} = i[P(M), M] = 0$$

So H = const solves the geodesic equation $\Rightarrow$ e^{-iHt} is a geodesic

Application of conjugate points

Suppose H = sum of 1- and 2-body terms.

Then M = sum of 1- and 2-body terms.

$$\dot{M} = i[P(M), M] = 0$$

So H = const solves the geodesic equation $\Rightarrow$ e^{-iHt} is a geodesic

It follows that for short times e^{-iHt} is a minimal curve.

Application of conjugate points

Suppose H = sum of 1- and 2-body terms.

Then M = sum of 1- and 2-body terms.

$$\dot{M} = i[P(M), M] = 0$$

So H = const solves the geodesic equation $\Rightarrow$ e^{-iHt} is a geodesic

It follows that for short times e^{-iHt} is a minimal curve.

Conjugate points provide us with a calculational means of proving that a curve is no longer minimizing.

Application of conjugate points

Suppose H = sum of 1- and 2-body terms.

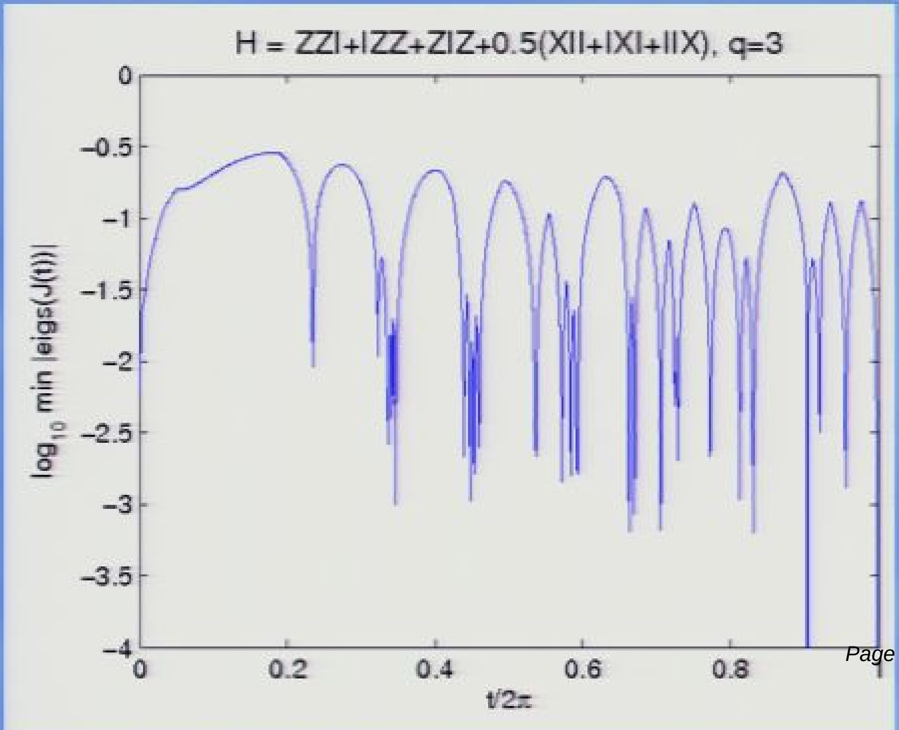
Then M = sum of 1- and 2-body terms.

$$\dot{M} = i[P(M), M] = 0$$

So H = const solves the geodesic equation $\Rightarrow$ e^{-iHt} is a geodesic

It follows that for short times e^{-iHt} is a minimal curve.

Conjugate points provide us with a calculational means of proving that a curve is no longer minimizing.



Application of conjugate points

Suppose H = sum of 1- and 2-body terms.

Then M = sum of 1- and 2-body terms.

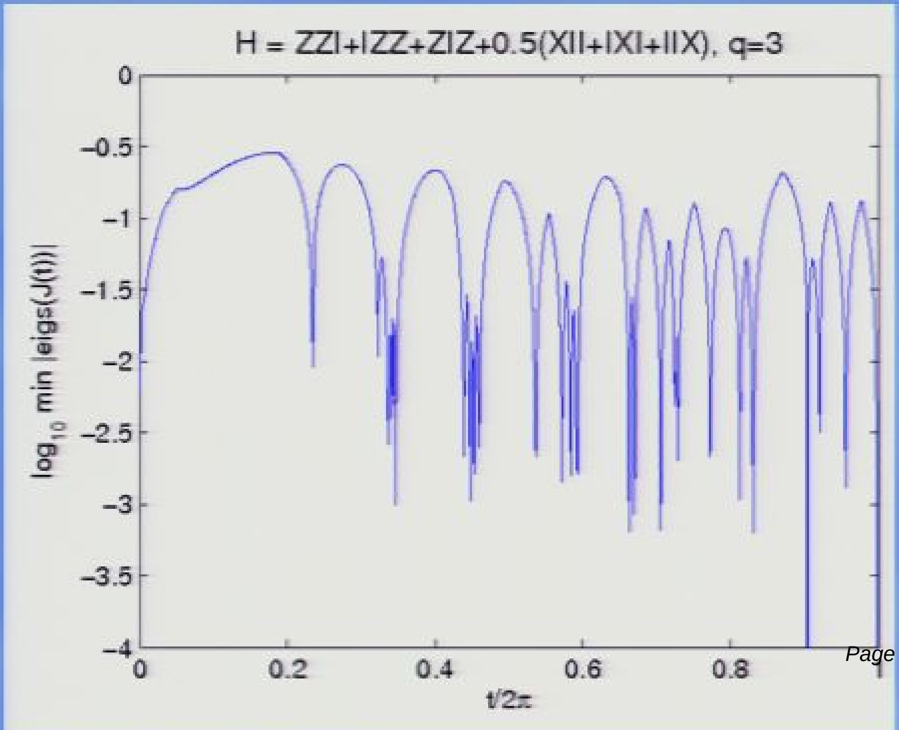
$$\dot{M} = i[P(M), M] = 0$$

So H = const solves the geodesic equation $\Rightarrow$ e^{-iHt} is a geodesic

It follows that for short times e^{-iHt} is a minimal curve.

Conjugate points provide us with a calculational means of proving that a curve is no longer minimizing.

We are using conjugate points to study the impact of ancilla on distance.



Application of conjugate points

Suppose H = sum of 1- and 2-body terms.

Then M = sum of 1- and 2-body terms.

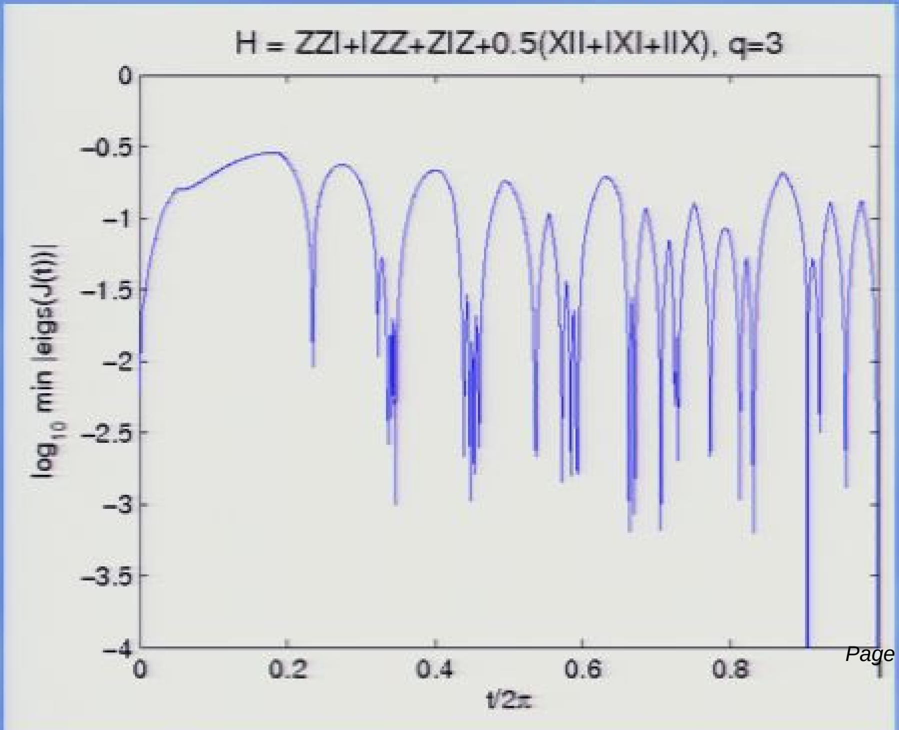
$$\dot{M} = i[P(M), M] = 0$$

So H = const solves the geodesic equation $\Rightarrow$ e^{-iHt} is a geodesic

It follows that for short times e^{-iHt} is a minimal curve.

Conjugate points provide us with a calculational means of proving that a curve is no longer minimizing.

We are using conjugate points to study the impact of ancilla on distance.



Why conjugate points

Why conjugate points

Suppose H = sum of 1- and 2-body terms.

Then M = sum of 1- and 2-body terms.

What's the connection to Razborov-Rudich?

What's the connection to Razborov-Rudich?

Recall: If good pseudorandom number generators exist, then it's impossible to efficiently distinguish easy- and hard-to-compute functions.

What's the connection to Razborov-Rudich?

Recall: If good pseudorandom number generators exist, then it's impossible to efficiently distinguish easy- and hard-to-compute functions.

What's the connection to Razborov-Rudich?

Recall: If good pseudorandom number generators exist, then it's impossible to efficiently distinguish easy- and hard-to-compute functions.

Corollary: If good pseudorandom number generators exist, then it's impossible to efficiently determine distances on Riemannian manifolds.

What's the connection to Razborov-Rudich?

Recall: If good pseudorandom number generators exist, then it's impossible to efficiently distinguish easy- and hard-to-compute functions.

Corollary: If good pseudorandom number generators exist, then it's impossible to efficiently determine distances on Riemannian manifolds.

Could we ever use the geometric point of view to find a hard-to-compute unitary operation?

What's the connection to Razborov-Rudich?

Recall: If good pseudorandom number generators exist, then it's impossible to efficiently distinguish easy- and hard-to-compute functions.

Corollary: If good pseudorandom number generators exist, then it's impossible to efficiently determine distances on Riemannian manifolds.

Could we ever use the geometric point of view to find a hard-to-compute unitary operation?