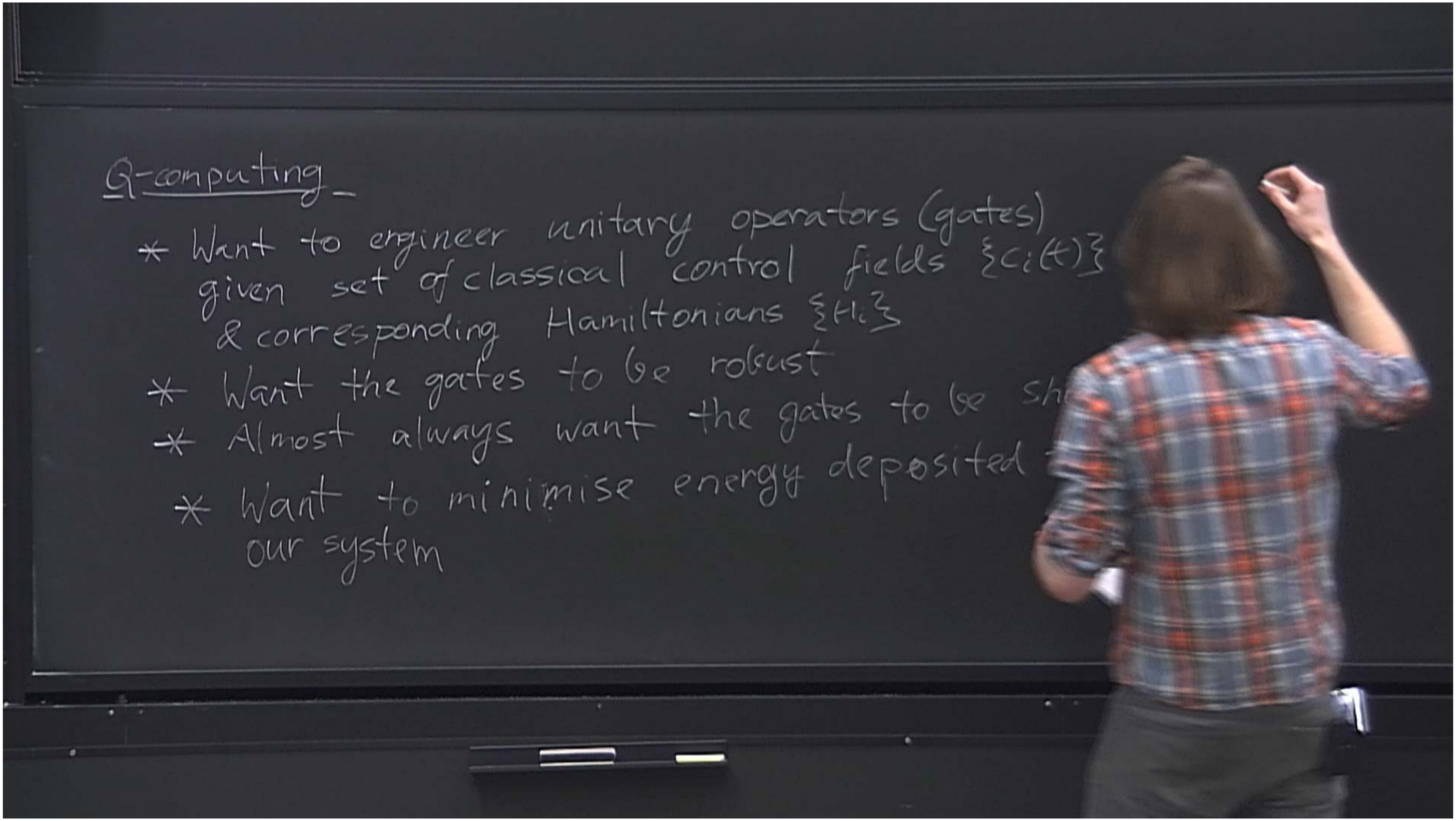
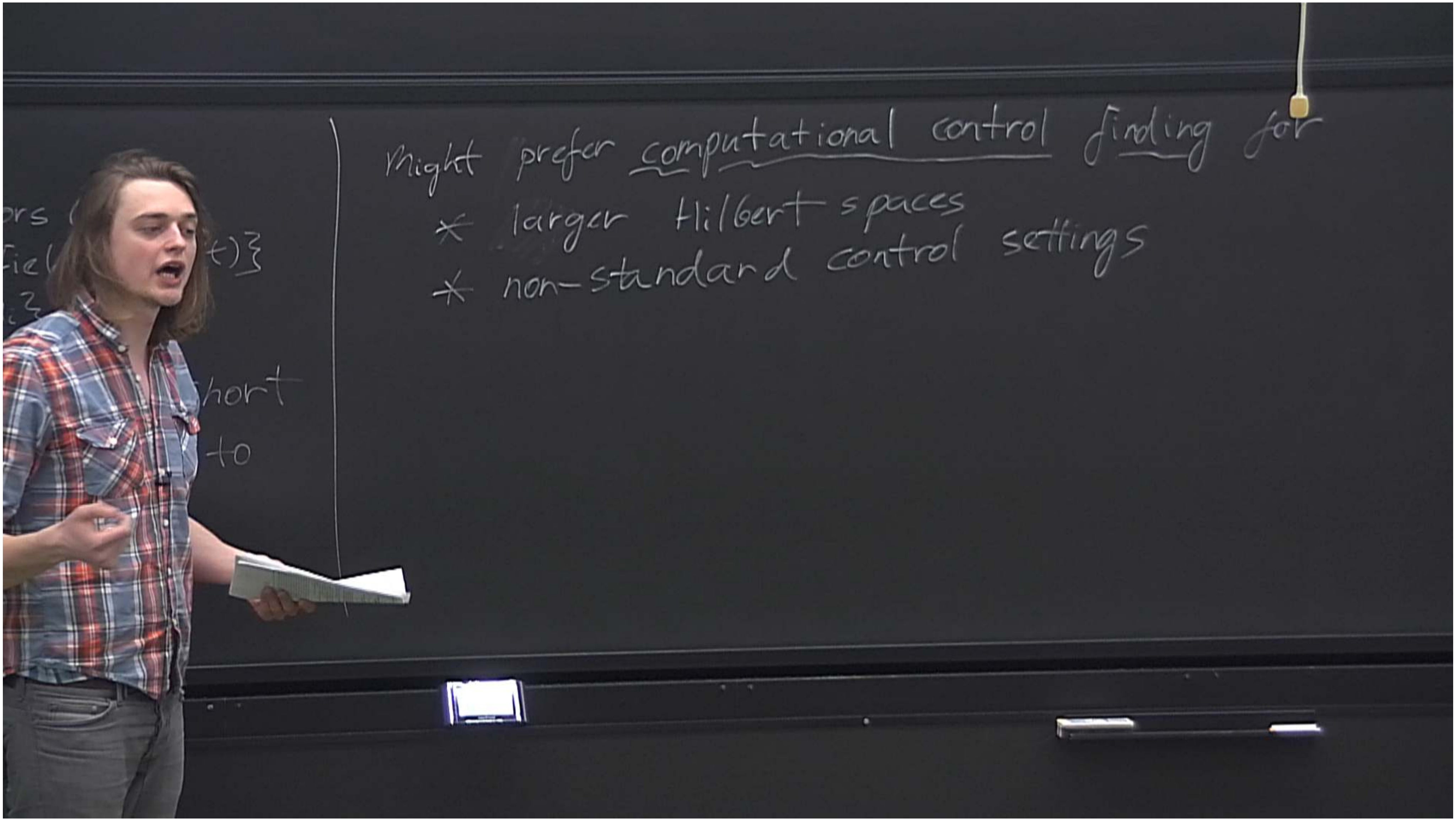
Title: Explorations in Quantum Information-11

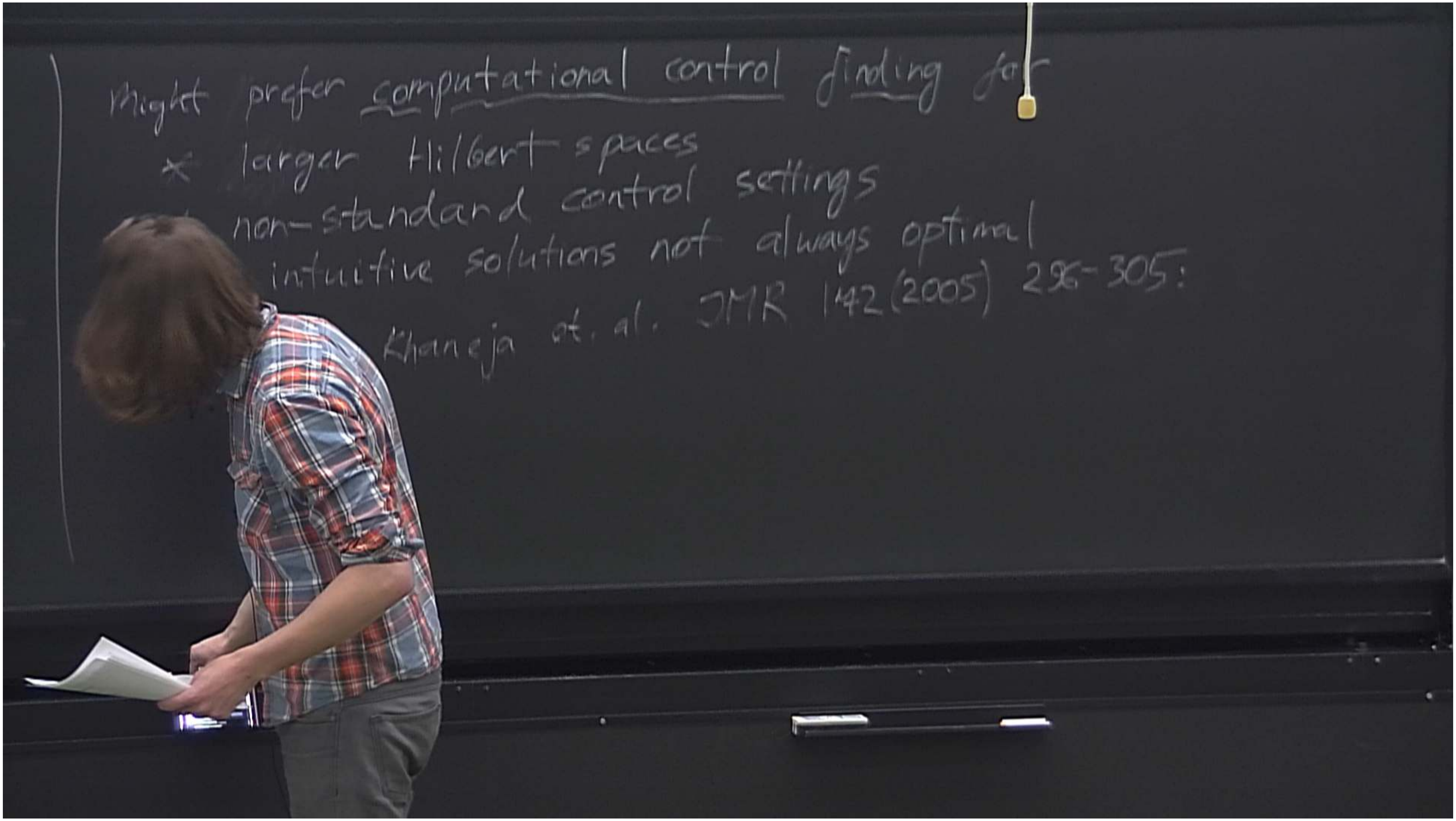
Date: Mar 30, 2015 09:00 AM

URL: http://pirsa.org/15030032

Abstract:


Q-computing
* Want to engineer unitary operators (gates)
given set of classical control fields $\{c_i(t)\}$
& corresponding Hamiltonians $\{H_i\}$
* Want the gates to be robust
* Almost always want the gates to be
* Want to minimise energy deposited
our system


Might prefer computational control finding for
* larger Hilbert spaces
* non-standard control settings

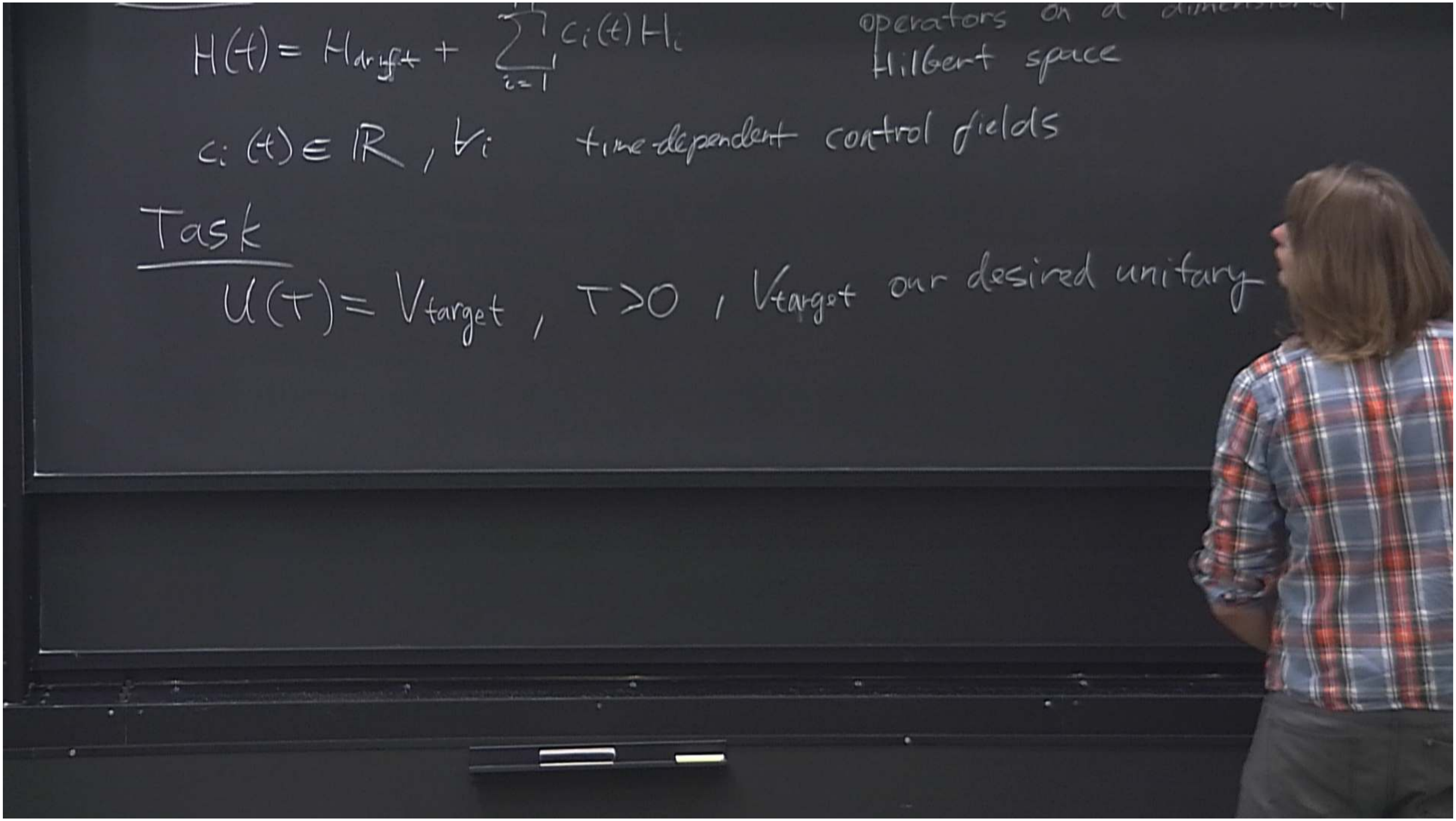

might prefer computational control finding for
* larger Hilbert spaces
non-standard control settings
intuitive solutions not always optimal
Khaneja et. al. JMR 142 (2005) 296-305:

Model

$$H(t) = H_{drift} + \sum_{i=1}^{M} c_i(t) H_i$$

H_{drift}, $\{H_i\}$ Hermitian operators on d dimensional Hilbert space

$c_i(t) \in \mathbb{R}$, $\forall i$ time-dependent control fields


$H(t) = H_{drift} + \sum_{i=1} c_i(t) H_i$
operators on a d
Hilbert space
$c_i(t) \in \mathbb{R}, \forall i$ time-dependent control fields
Task
$U(T) = V_{target}, \; T > 0, \; V_{target}$ our desired unitary

Model

$$H(t) = H_{drift} + \sum_{i=1}^{M} c_i(t) H_i$$

H_{drift}, $\{H_i\}$ Hermitian operators on d dimensional Hilbert space

$c_i(t) \in \mathbb{R}$, $\forall i$ time-dependent control fields

Task

$U(T) = V_{target}$, $T > 0$, V_{target} our desired unitary,

while $\frac{d}{dt} U(t) = -i H(t) U(t)$ & $U(0) = \mathbb{1}$

and $\{c_i(t)\}$ satisfy all constraints imposed on them

Model

H_{drift}, $\{H_i\}$ Hermitian operators on d dimensional Hilbert space

$$H(t) = H_{drift} + \sum_{i=1}^{M} c_i(t) H_i$$

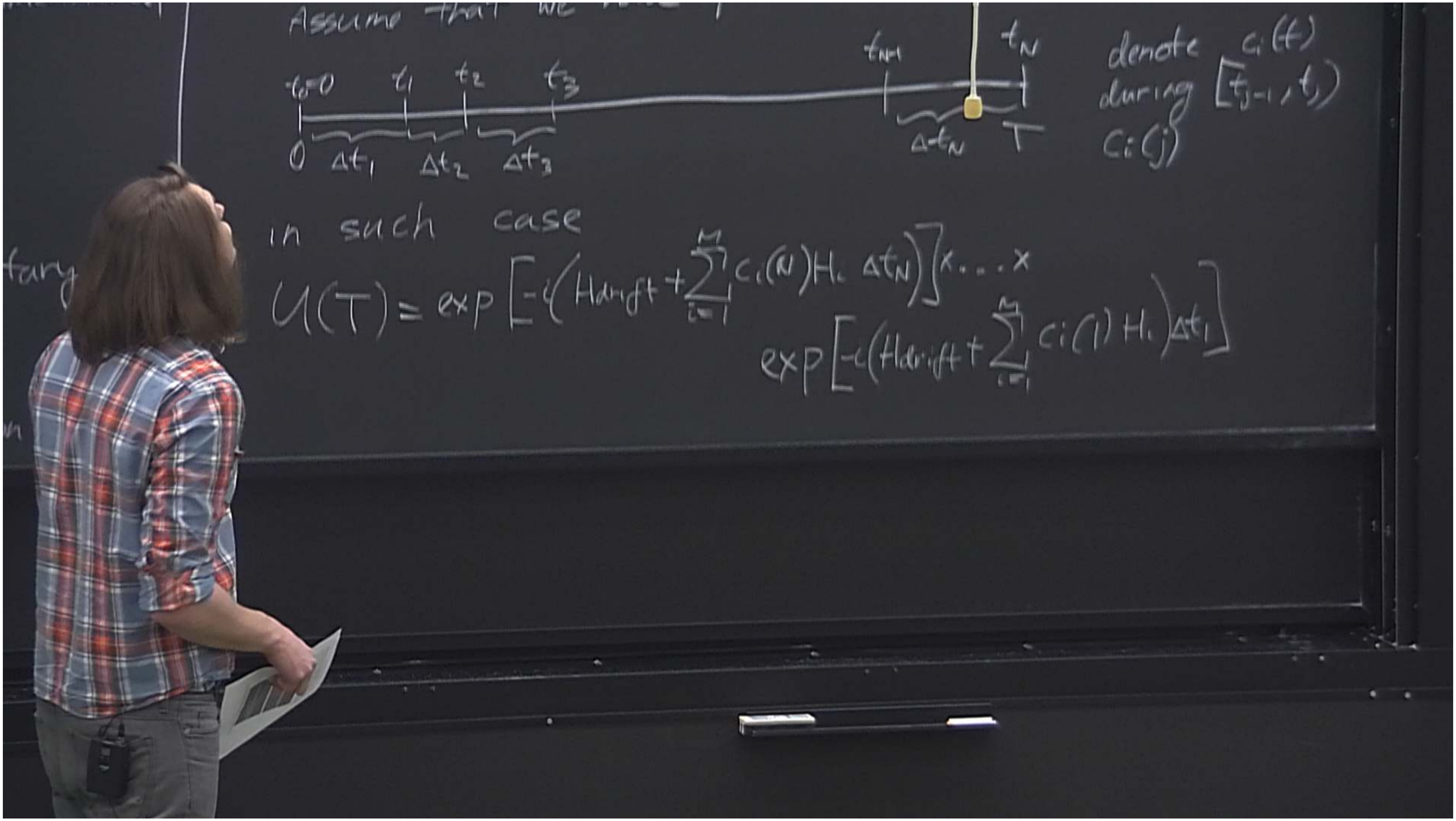
$c_i(t) \in \mathbb{R}$, $\forall i$ time-dependent control fields

Task

$U(T) = V_{target}$, $T > 0$, V_{target} our desired unitary,

while $\frac{d}{dt} U(t) = -i H(t) U(t)$ & $U(0) = \mathbb{1}$

and $\{c_i(t)\}$ satisfy all constraints imposed on them


Assume that we have
$t_0 = 0$ t_1 t_2 t_3 t_{N-1} t_N
0 Δt_1 Δt_2 Δt_3 Δt_N T
denote $c_i(t)$
during $[t_{j-1}, t_j)$
$c_i(j)$
in such case
$U(T) = \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(N) H_i\right) \Delta t_N\right] \times \dots \times$
$\exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(1) H_i\right) \Delta t_1\right]$

and $\{c_i(t)\}$ satisfy all constraints imposed on them

$$0 \leq \Phi\left[U(T)\right] = \frac{1}{d^2}\,\mathrm{Tr}\left[V^{+}_{target} \cdot U(T)\right]\mathrm{Tr}\left[V_{target} \cdot U^{+}(T)\right] \leq 1$$

and $\{c_i(t)\}$ satisfy all constraints imposed on them

$$0 \leq \Phi[U(T)] = \frac{1}{d^2} \mathrm{Tr}\left[V^{\dagger}_{target} \cdot U(T)\right] \mathrm{Tr}\left[V_{target} \cdot U^{\dagger}(T)\right] \leq 1$$

Derivatives?

$$\frac{\delta}{\delta c_k(j)} \Phi[U(T)] = 2 \mathrm{Re}\left\{ \frac{1}{d^2} \mathrm{Tr}\left[V_{target}^{\dagger} \cdot \frac{\delta}{\delta c_k(j)} U(T)\right] \mathrm{Tr}\left[V_{target} \cdot U\right.\right.$$

$$\frac{\delta}{\delta c_k(j)} U(T) = \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(N) H_i\right)\Delta t_N\right] \times \ldots \times \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c\right.\right.$$

$\ldots)\}$ satisfy all constraints imposed on them

$$\frac{1}{d^2}\,\mathrm{Tr}\left[V_{target}^{+}\cdot U(T)\right]\mathrm{Tr}\left[V_{target}\cdot U^{+}(T)\right]\leq 1\,,\quad \Phi\left[U(T)\right]=$$

$$\ldots(T)\Big]=2\,\mathrm{Re}\left\{\frac{1}{d^2}\,\mathrm{Tr}\left[V_{target}^{+}\cdot\frac{\delta}{\delta c_k(j)}U(T)\right]\mathrm{Tr}\left[V_{target}\cdot U^{+}\right]\right.$$

$$G_j$$

$$\ldots)=\exp\left[-i\left(H_{drift}+\sum_{i=1}^{M}c_i(N)H_i\right)\Delta t_N\right]\times\ldots\times\exp\left[-i\left(H_{drift}+\sum_{i=1}^{M}c_i(j+1)H_i\right)\Delta t_{j+1}\right]\times\frac{\delta}{\delta c_k(j)}$$

...sfy all constraints imposed on them

$$[V^{\dagger}_{target} . U(T)] \, Tr[V_{target} . U^{\dagger}(T)] \leq 1 \, , \quad \Phi[U(T)] = \Phi(\{c_i(j ...$$

$$Re\left\{ \frac{1}{d^2} Tr\left[V_{target}^{\dagger} . \frac{\delta}{\delta c_k(j)} U(T)\right] Tr[V_{target} . U^{\dagger}]\right.$$

$$\ldots \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(N) H_i\right)\Delta t_N\right] \times \ldots \times \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(j+1) H_i\right)\Delta t_{j+1}\right] \times \frac{\delta}{\delta c_k(j)} \exp\left[-i(H \ldots\right.$$

$$\underbrace{\exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(j-1) H_i\right)\Delta t_{j-1}\right] \times \ldots \times \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(1) H_i\right)\Delta t_1\right]}_{F_{j-1}}$$

Now

$$D_{kj} = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(j) H_i\right)\Delta t - i\varepsilon H_k \Delta t_j\right]$$

Now

$$D_{kj} = \frac{d}{d\varepsilon}\bigg|_{\varepsilon=0} \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(j) H_i\right)\Delta t - i\varepsilon H_k \Delta t_j\right]$$

$\frac{d}{}$

Now

$$D_{kj} = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(j) H_i\right)\Delta t - i\varepsilon H_k \Delta t_j\right]$$

$$\frac{d}{d\varepsilon}\Big|_{\varepsilon=0} e^{X+\varepsilon Y} = \left(\int_0^1 d\alpha\, e^{\alpha X} Y e^{-\alpha X}\right) e^{X}$$

$\frac{d}{dx}\int$

Wilcox JMP 8

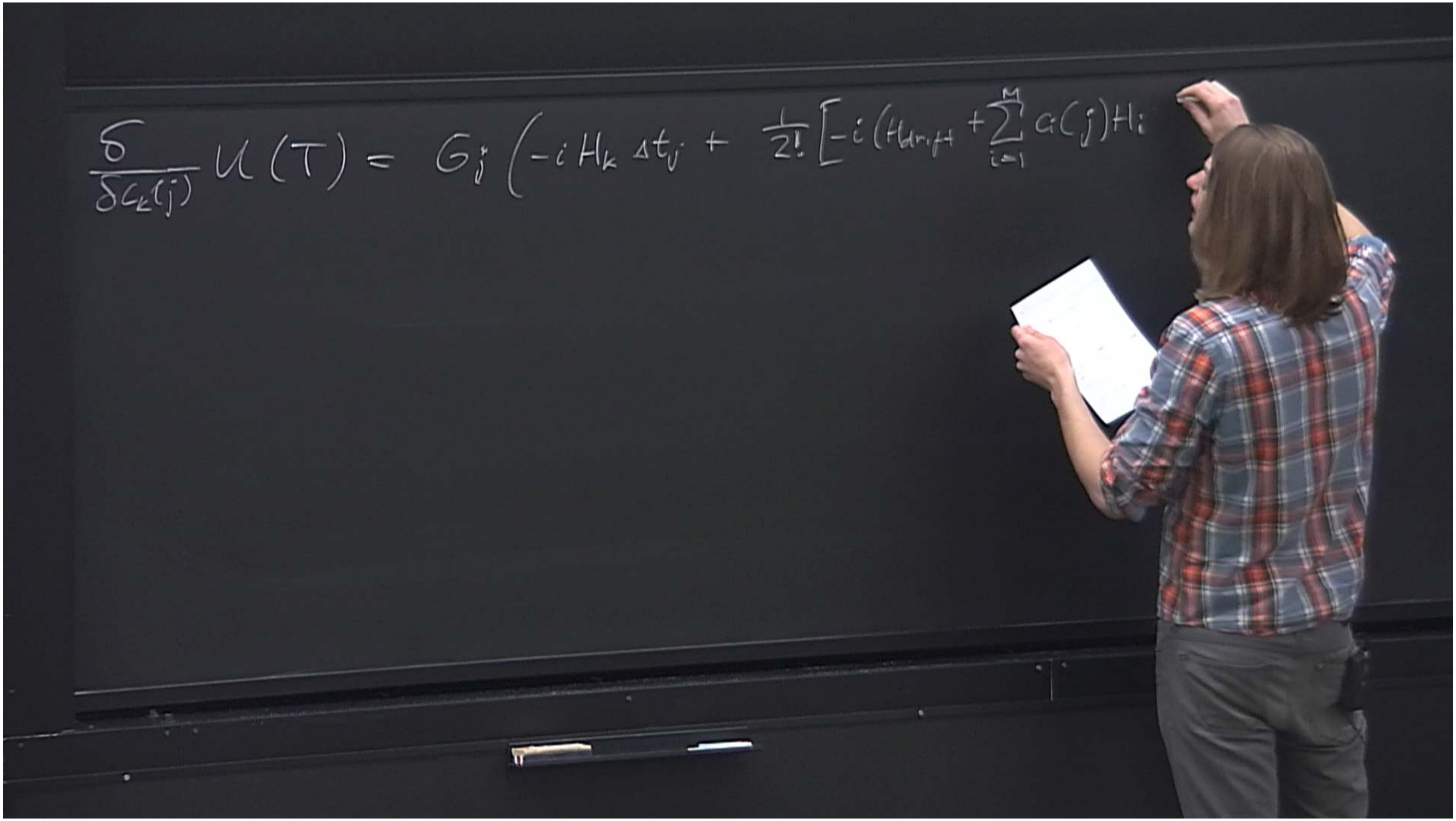
Now

$$D_{kj} = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \exp\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(j) H_i\right)\Delta t_j - i\varepsilon H_k \Delta t_j\right]$$

Wilcox

$$\frac{d}{d\varepsilon}\Big|_{\varepsilon=0} e^{X+\varepsilon Y} = \Big(\underbrace{\int_0^1 d\alpha\, e^{\alpha X} Y e^{-\alpha X}}_{Y + \frac{1}{2!}[X,Y] + \frac{1}{3!}[X,[X,Y]] + \ldots}\Big) e^{X}$$

usually pick Δt s.t. $\|X\| \ll 1$


$\frac{\delta}{\delta c_k(j)} U(T) = G_j \left(-i H_k \Delta t_j + \frac{1}{2!}\left[-i\left(H_{drift} + \sum_{i=1}^{M} c_i(j) H_i\right.\right.\right.$

$$\frac{\delta}{\delta c_k(j)} U(T) = G_j \Big(-i H_k \Delta t_j + \frac{1}{2!}\Big[-i \big(H_{drift} + \sum_{i=1}^{M} c_i(j) H_i \Delta t_j\big), -i H_k \Delta t_j \Big] + \dots$$

$$\frac{\delta}{\delta c_k(j)} U(T) = G_j \left(-i H_k \Delta t_j + \frac{1}{2!} \left[-i \left(H_{drift} + \sum_{i=1}^{M} c_i(j) H_i \Delta t_j , -i H_k \Delta t \right.\right.\right.$$

Algorithm

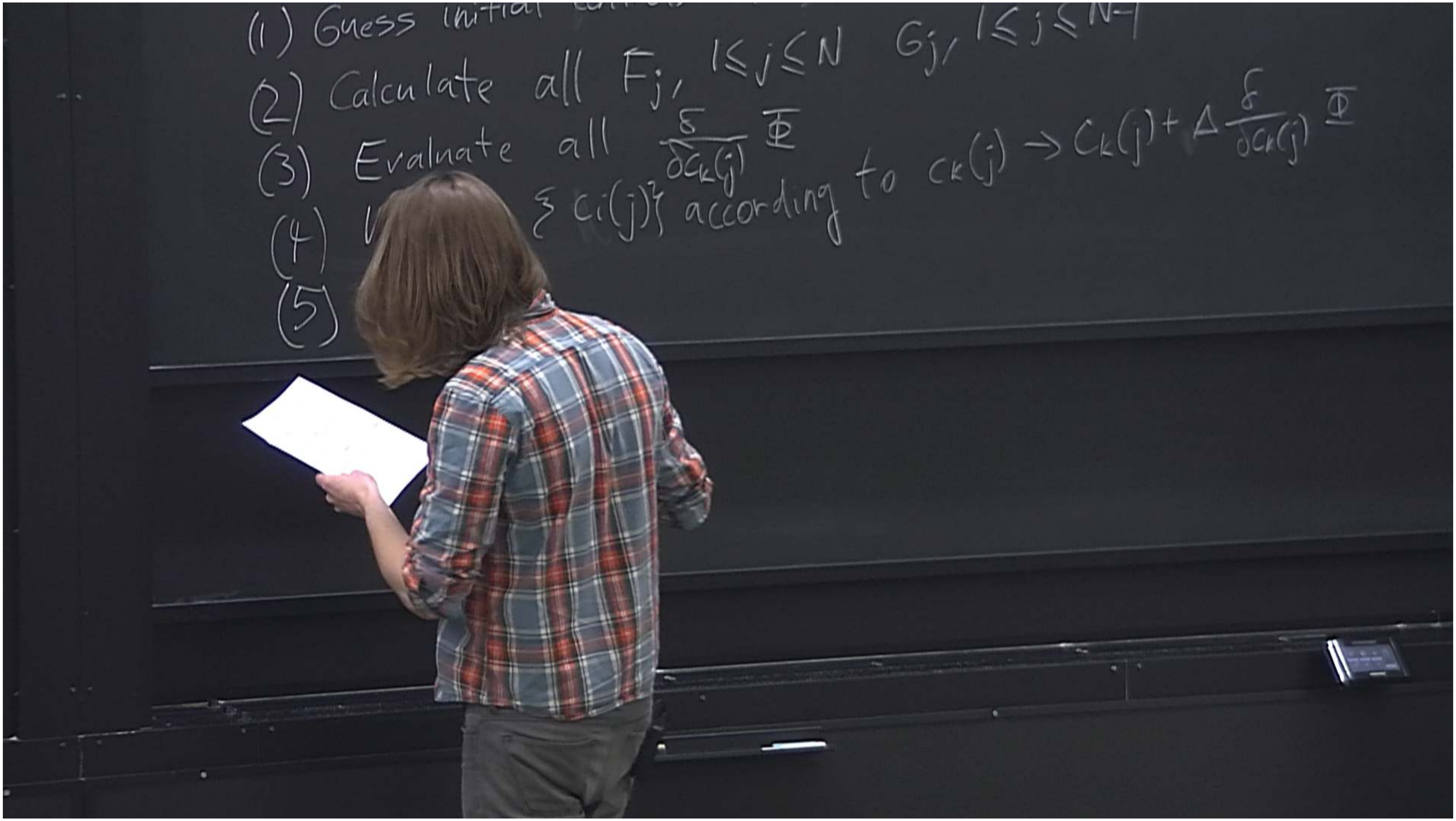
(1) Guess initial controls $\{c_i(j)\}$

(2) Calculate all F_j, $1 \leq j \leq N$ G_j, $1 \leq j \leq N-1$

(3) Evaluate all $\frac{\delta}{\delta c_k(j)} \Phi$

(4) Update $\{c_i(j)\}$ according to $c_k(j) \to c_k(j) + \Delta \frac{\delta}{\delta c_k(j)} \Phi$

(5)


(1) Guess initial
(2) Calculate all F_j, $1 \leq j \leq N$ G_j, $1 \leq j \leq N-1$
(3) Evaluate all $\frac{\delta}{\delta c_k(j)} \Phi$
(4) $\{c_i(j)\}$ according to $c_k(j) \rightarrow c_k(j) + \Delta \frac{\delta}{\delta c_k(j)} \Phi$
(5)

(4) Update $\{c_i(j)\}$ according

(5) Make sure $\{c_i(j)\}$ satisfy constraints

Example

$$H(t) = \Delta\omega \frac{\sigma_z}{2} + \gamma\left(c_x(t)\frac{\sigma_x}{2} + c_y(t)\frac{\sigma_y}{2}\right),\qquad \Delta\omega \in [-0.25, 0.25],$$

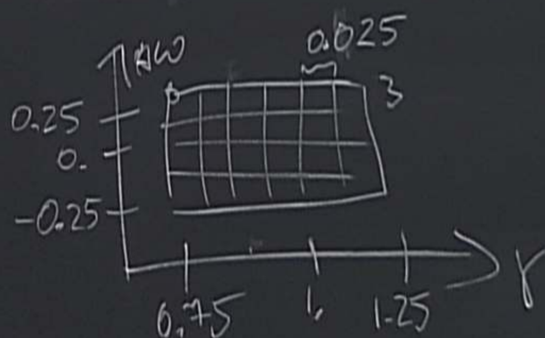
(4) Update $\{c_i(j)\}$ according

(5) Make sure $\{c_i(j)\}$ satisfy constraints

Example

$$H(t) = \Delta\omega \frac{\sigma_z}{2} + \gamma\left(c_x(t)\frac{\sigma_x}{2} + c_y(t)\frac{\sigma_y}{2}\right), \quad \Delta\omega \in [-0.25, 0.25],$$

$$-1 \le c_x(t), c_y(t) \le 1, \ \forall t$$



(4) Update $\{c_i(j)\}$ according

(5) Make sure $\{c_i(j)\}$ satisfy constraints

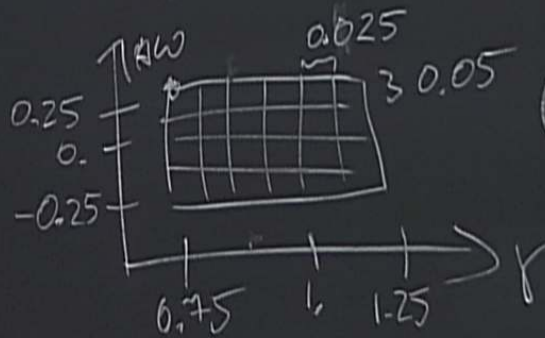
Example

$$H(t) = \Delta\omega \frac{\sigma_z}{2} + \gamma \left(c_x(t) \frac{\sigma_x}{2} + c_y(t) \frac{\sigma_y}{2} \right), \quad \Delta\omega \in [-0.25, 0.25],$$

$$-1 \leq c_x(t), c_y(t) \leq 1, \ \forall t$$

→ Sequence length $\sim 33\pi$ pul

→ Pick 100 equal time step



0.05 (231 points)

→ Evaluate fidelity against the compensated pulse

PSI lecture GRAPE.nb

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

Wolfram Mathematica | FOR STUDENTS

Demonstrations | MathWorld | Student Forum | Help

Definitions

Pulse Finder

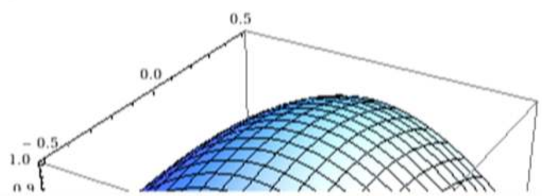
Target Functions

Searches

Fidelity Plots

- **Hard Pulse**

In[123]:=
```
ListPlot3D[
 Flatten[
  Table[
   {i, j, Abs[
     {0, 1}.GenerateLastPropagator[GenerateSolutions[{π}, {{i 1, 0, j}}, solutionUnitary]
      ].{1, 0}
    ]}, {i, 0.5, 1.5, 0.05}, {j, -0.5, 0.5, 0.05}
  ], 1
 ]
]
```



100%

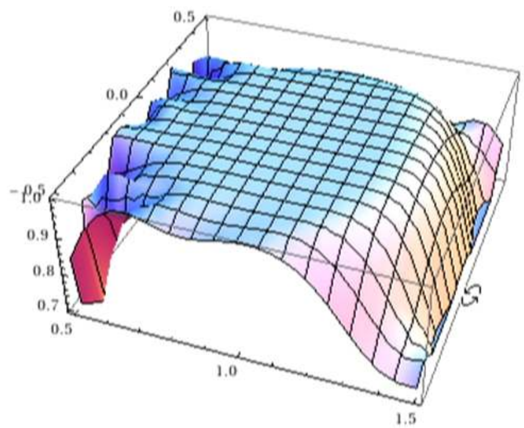
PSI lecture GRAPE.nb

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

Wolfram Mathematica | FOR STUDENTS — Demonstrations | MathWorld | Student Forum | Help

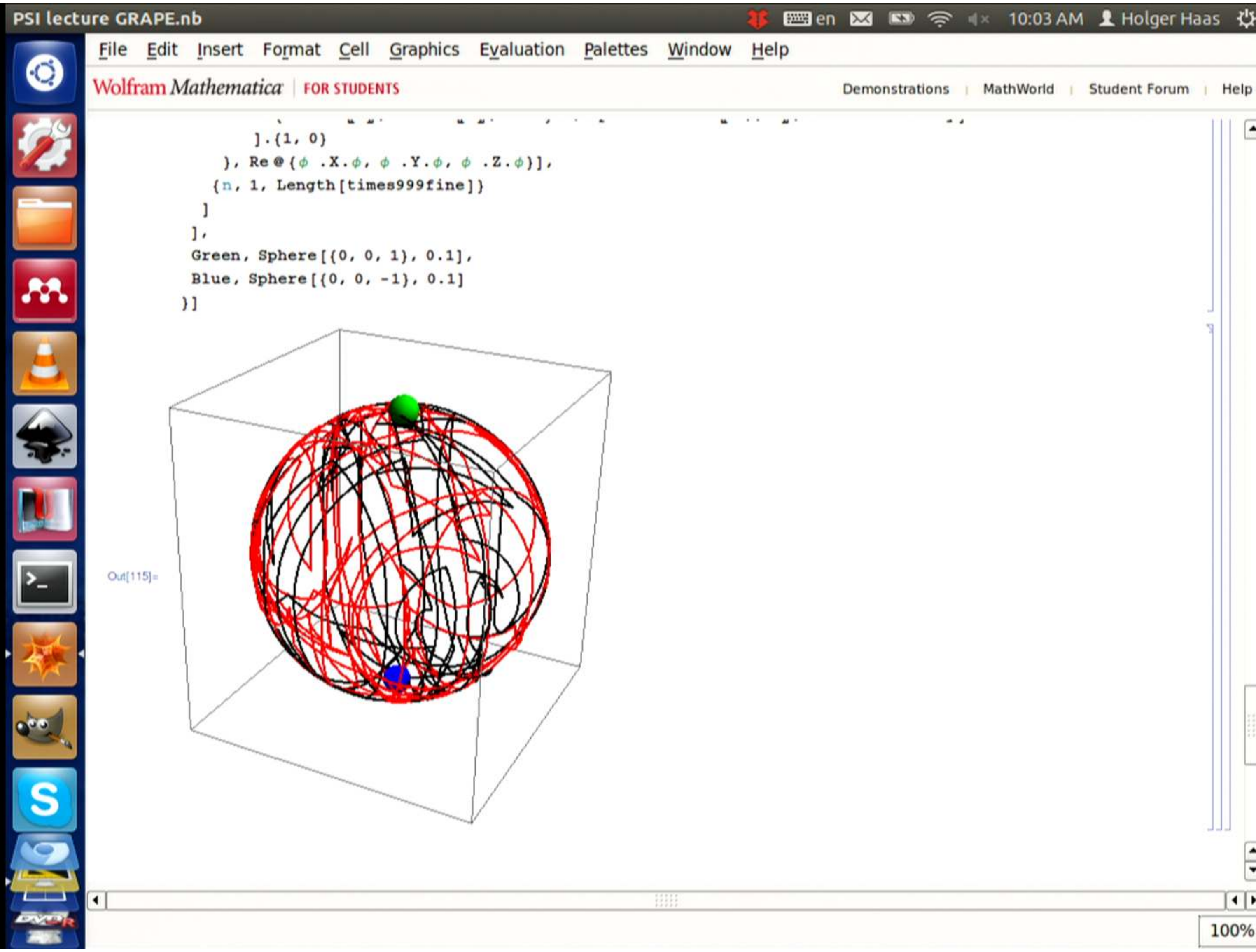
```
      {0, 1}.GenerateLastPropagator[GenerateSolutions[times999, {i #[[1]], i #[[2]], j} & /@ pulse999,
         solutionUnitary]
       ].{1, 0}
     ]}, {i, 0.5, 1.5, 0.05}, {j, -0.5, 0.5, 0.05}
   ], 1
  ]
 ]
```

Out[125]=




```
In[126]:= crossSection[Δω_] :=
  ListLinePlot[
   {Table[
     {i, Abs[
       {0, 1}.GenerateLastPropagator[GenerateSolutions[times999, {i #[[1]], i #[[2]], Δω} & /@ pulse999,
          solutionUnitary]
        ].{1, 0}
      ]}, {i, 0.5, 1.5, 0.05}
    ],
```

100%


PSI lecture GRAPE.nb
10:03 AM Holger Haas
File Edit Insert Format Cell Graphics Evaluation Palettes Window Help
Wolfram Mathematica FOR STUDENTS
Demonstrations | MathWorld | Student Forum | Help
].{1, 0}
}, Re@{ϕ .X.ϕ, ϕ .Y.ϕ, ϕ .Z.ϕ}],
{n, 1, Length[times999fine]}
]
],
Green, Sphere[{0, 0, 1}, 0.1],
Blue, Sphere[{0, 0, -1}, 0.1]
}]
Out[115]=
100%