Part I:
Optimization

Part II:
Constrained optimization

Part III:
Piccolo.jl

Aaron Trowbridge

Staff, Robotics
Carnegie Mellon

Andy Goldschmidt

Postdoc, Comp Sci
UChicago

Jack Champagne

MSc, Comp Sci
Carnegie Mellon

Aditya Bhardwaj

PhD student,
Caltech

Piccolo.jl

Piccolo.jl

> QuantumCollocation.jl

> NamedTrajectories.jl

> TrajectoryIndexingUtils.jl

> QuantumCalibration.jl*

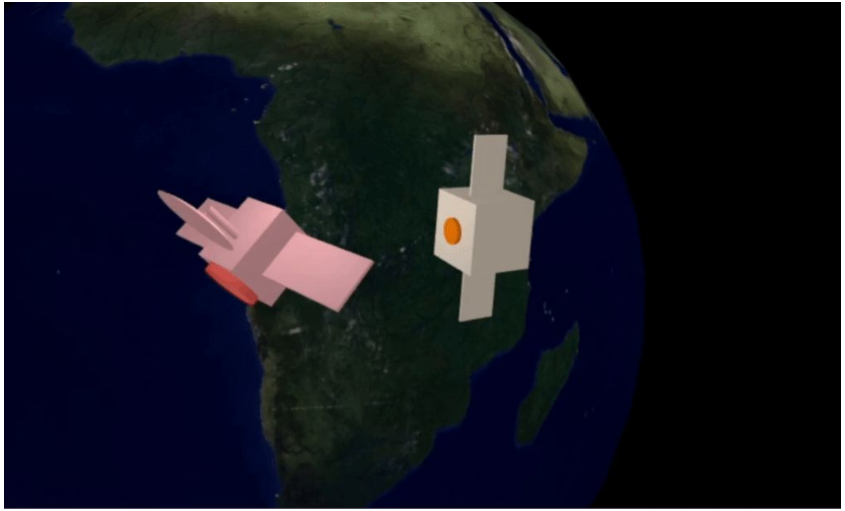
REXLAB

EPiQC

schuster.lab

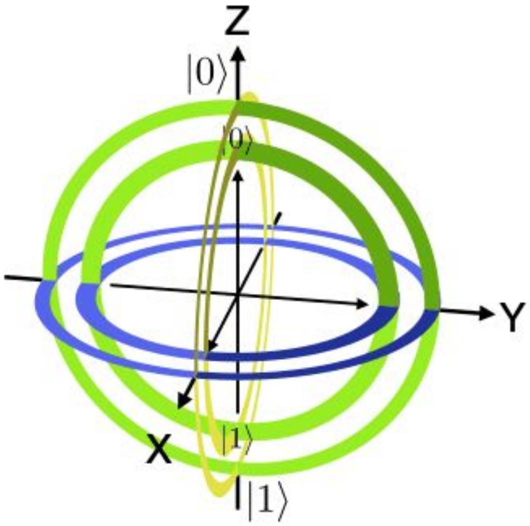
Unitary Fund

Optimal control



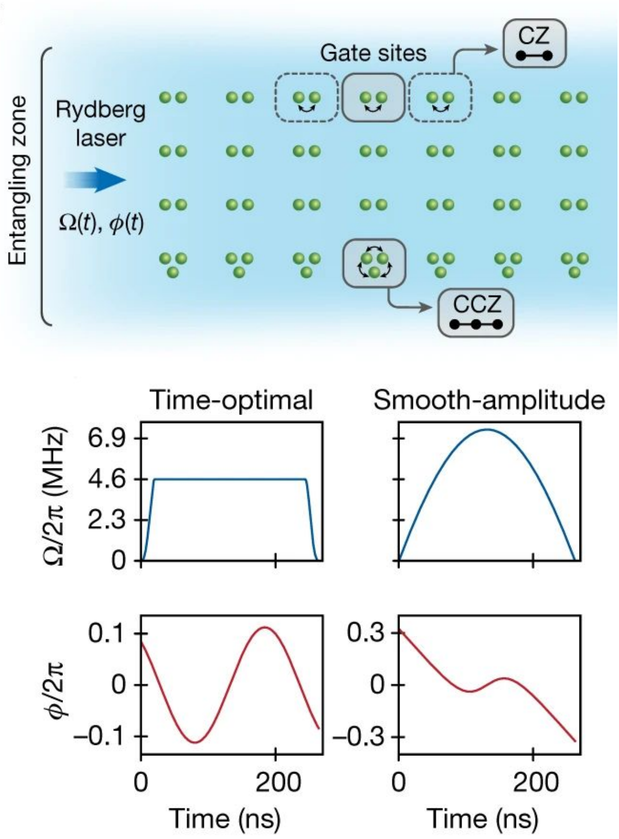
Control affine: $$\dot{\mathbf{x}}(t) = \mathbf{f}_0(\mathbf{x}(t)) + \sum_j u_j(t)\mathbf{f}_j(\mathbf{x}(t))$$

Quantum optimal control



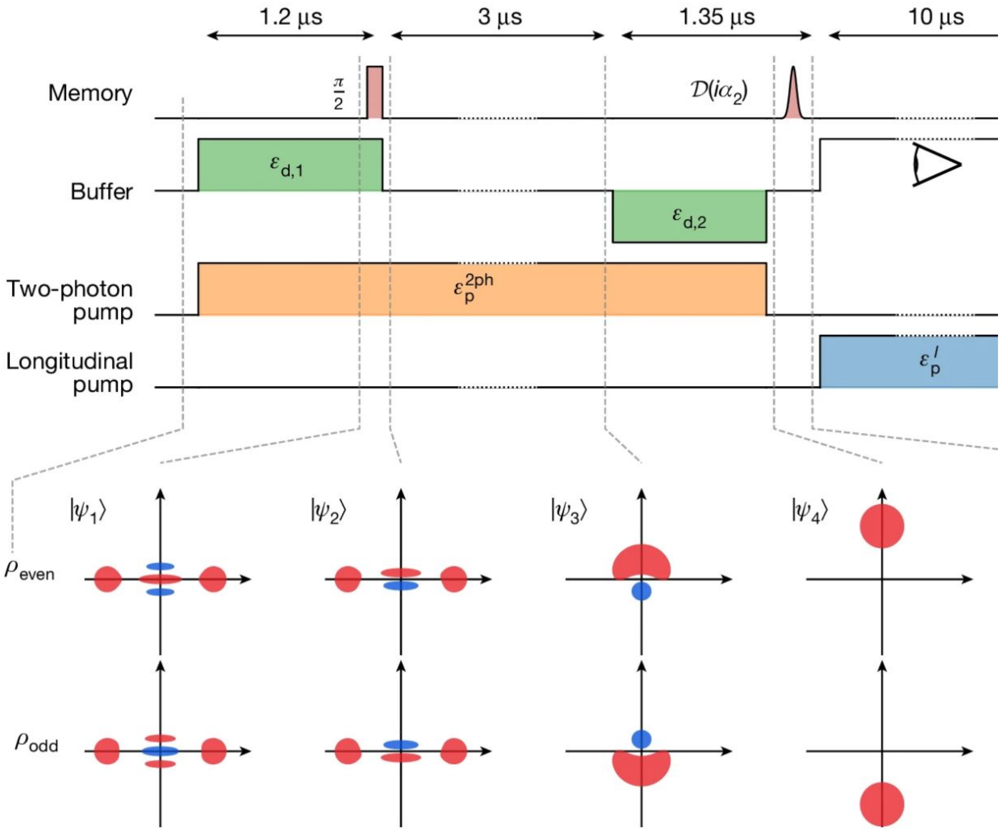
Control affine: $$i\frac{d}{dt}|\boldsymbol{\psi}(t)\rangle = \mathbf{H}_0|\boldsymbol{\psi}(t)\rangle + \sum_j u_j(t)\mathbf{H}_j|\boldsymbol{\psi}(t)\rangle$$

(a) Neutral atoms ⚛️



S. J. Evered et al., Nature 622, 268 (2023)

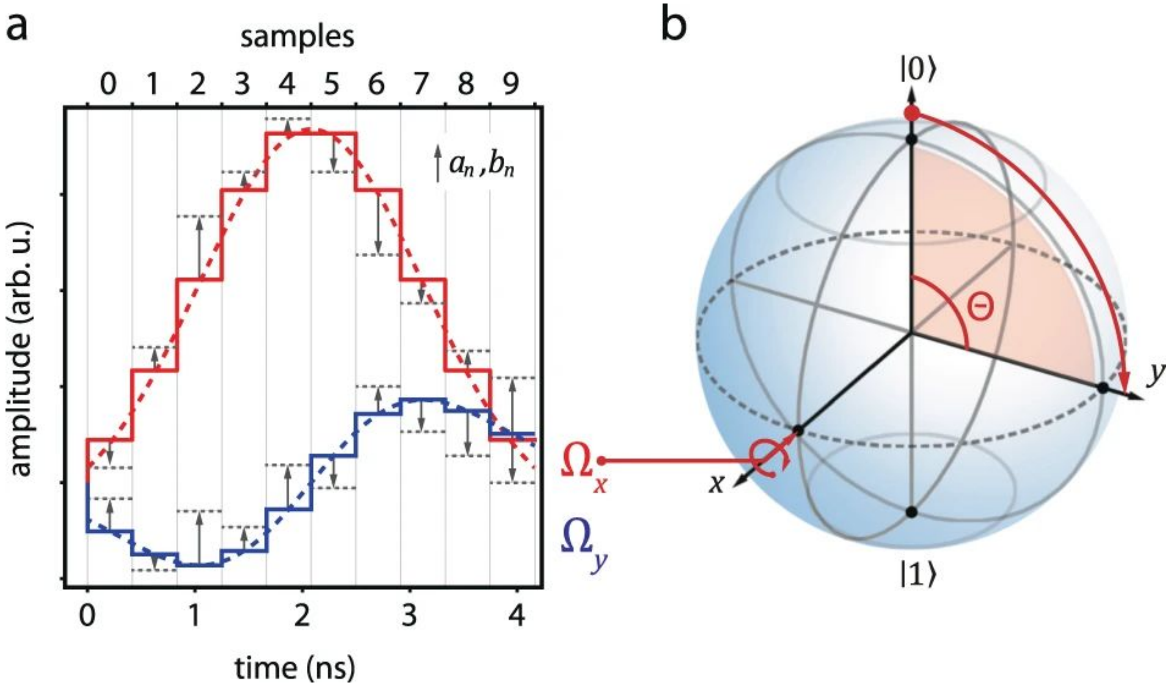
(b) Cat qubits 🙀



U. Réglade et al., Nature 629, 778 (2024)

(c) Transmons 🥶

C. P. Koch et al., EPJ Quantum Technology 9, (2022)



(d) Ions 🔋

(e) Spins 😵‍💫

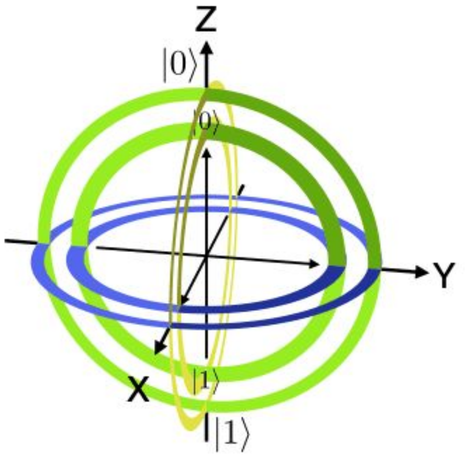
… and more!

M. Werninghaus et al., npj Quantum Info 7, (2021)

Quantum control by hand

Hamiltonians generate rotations.

- Commutators make orthogonal Hamiltonians.
- Computing all possible commutators reveals all possible rotations.



$$e^{t(X+Y)} = e^{tX}\ e^{tY}\ e^{-\frac{t^2}{2}[X,Y]}\ e^{\frac{t^3}{6}(2[Y,[X,Y]]+[X,[X,Y]])}\ \ldots$$

Part I

optimization

quantum optimal control

Optimization

$$\min_{\mathbf{x}} J(\mathbf{x})$$

Gradient descent & Newton's method

Necessary condition: $\nabla \mathbf{f}(\mathbf{x}^*) = 0$

- First-order methods: gradient.
- Second-order methods: Hessian.

$$0 \stackrel{!}{=} \nabla \mathbf{f}(\mathbf{x} + \Delta \mathbf{x}) \approx \nabla \mathbf{f}(\mathbf{x}) + \nabla^2 \mathbf{f}(\mathbf{x}) \Delta \mathbf{x}$$

$$\Delta \mathbf{x} \stackrel{!}{=} -(\nabla^2 \mathbf{f}(\mathbf{x}))^{-1} \nabla \mathbf{f}(\mathbf{x})$$

$$\min_{\mathbf{x}} J(\mathbf{x})$$

Example I.1

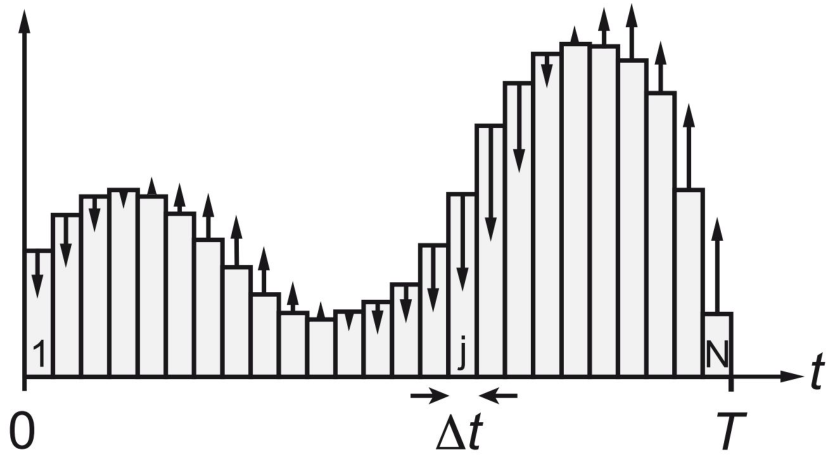
Example I.1

Takeaway: How to evaluate Newton's method? Regularization and line search.

GRadient Ascent Pulse Engineering

$$\min_{\mathbf{a}_{1:\mathrm{T}}} 1 - \frac{1}{N}\mathcal{F}\left(\mathbf{U}_{\mathrm{goal}}, \mathbf{U}_{\mathrm{T}}(\mathbf{a}_{1:\mathrm{T}})\right)$$

- It's just $\min_{\mathbf{x}} J(\mathbf{x})$
- *Indirect* method



Example I.2

Example I.2

Takeaway: How to set up and solve a GRAPE problem?

Rollouts and **Optim.jl**.

Exercises

- Adding Piccolo
- Modifying GRAPE

Part II

constrained optimization

quantum collocation

Constrained optimization

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}) = 0 \\ & \mathbf{C}(\mathbf{x}) \leq 0 \end{aligned}$$

Equality constraints

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}) = 0 \end{aligned}$$

Equality constraints

Remember your classical mechanics: Lagrange multipliers!

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & J(\mathbf{x}) \\
\text{s.t.} \quad & \mathbf{f}(\mathbf{x}) = 0
\end{aligned}
$$

Equality constraints

Remember your classical mechanics: Lagrange multipliers!

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = J(\mathbf{x}) + \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}} \boldsymbol{\lambda}$$

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}) = 0 \end{aligned}$$

Necessary conditions

Compute both gradients of the Lagrangian

$$\nabla_{\mathbf{x}}\mathcal{L} = \nabla_{\mathbf{x}} J(\mathbf{x}) + \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}} \boldsymbol{\lambda}$$

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L} = \mathbf{f}(\mathbf{x}) = 0$$

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}) = 0 \end{aligned}$$

Newton's method ⇒ KKT system

Compute the Newton step toward $\nabla \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0$

Give it a try!

$$\nabla_{\mathbf{x}} \mathcal{L} = \nabla_{\mathbf{x}} J(\mathbf{x}) + \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}} \boldsymbol{\lambda}$$

$$\nabla_{\lambda} \mathcal{L} = \mathbf{f}(\mathbf{x}) = 0$$

$$\min_{\mathbf{x}} \quad J(\mathbf{x})$$
$$\text{s.t.} \quad \mathbf{f}(\mathbf{x}) = 0$$

Newton's method ⇒ KKT system

Compute the Newton step toward $\nabla \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0$

$$\begin{bmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{x}^2} & \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \\ \frac{\partial \mathbf{f}}{\partial \mathbf{x}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \\ \mathbf{f}(\mathbf{x}) \end{bmatrix}$$

$$\nabla_{\mathbf{x}} \mathcal{L} = \nabla_{\mathbf{x}} J(\mathbf{x}) + \left(\frac{\partial \mathrm{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}} \boldsymbol{\lambda}$$

$$\nabla_{\lambda} \mathcal{L} = \mathrm{f}(\mathbf{x}) = 0$$

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}) = 0 \end{aligned}$$

Example II.1

Example II.1

Takeaway: How the KKT conditions work

Inequality constraints

KKT conditions (*necessary* conditions)

1. $\nabla f(\mathbf{x}) + \left(\frac{\partial \mathbf{C}}{\partial \mathbf{x}}\right)^{\mathrm{T}} \boldsymbol{\lambda} = 0$
2. $\mathbf{C}(\mathbf{x}) \leq 0$
3. $\boldsymbol{\lambda} \geq 0$
4. $\mathbf{C}(\mathbf{x}) \odot \boldsymbol{\lambda} = 0$

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{C}(\mathbf{x}) \leq 0 \end{aligned}$$

Inequality constraints

KKT conditions (*necessary* conditions)

1. $\nabla f(\mathbf{x}) + \left(\frac{\partial \mathbf{C}}{\partial \mathbf{x}}\right)^{\mathrm{T}} \boldsymbol{\lambda} = 0$ Stationarity
2. $\mathbf{C}(\mathbf{x}) \leq 0$ Primal feasibility
3. $\boldsymbol{\lambda} \geq 0$ Dual feasibility
4. $\mathbf{C}(\mathbf{x}) \odot \boldsymbol{\lambda} = 0$ Complementarity

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{C}(\mathbf{x}) \leq 0 \end{aligned}$$

How to solve

Active set: Guess when constraints are active / inactive.

How to solve

Active set: Guess when constraints are active / inactive.

Penalty methods: Add regularizers on constraints.

How to solve

Active set: Guess when constraints are active / inactive.

~~Penalty methods~~: Add regularizers on constraints.

How to solve

Active set: Guess when constraints are active / inactive.

~~**Penalty methods**~~: Add regularizers on constraints.

Augmented Lagrangian: Add Lagrange multipliers on constraints.

How to solve

Active set: Guess when constraints are active / inactive.

~~**Penalty methods**~~: Add regularizers on constraints.

Augmented Lagrangian: Add Lagrange multipliers on constraints.

Interior-point: Add log-barriers on constraints.

How to solve

Active set: Guess when constraints are active / inactive.

~~Penalty methods~~: Add regularizers on constraints.

Augmented Lagrangian: Add Lagrange multipliers on constraints.

Interior-point: Add log-barriers on constraints.

`Piccolo.jl`

Quantum COLLOocation

Objectives

$$\min_{\mathbf{x}_{1:\mathrm{T}},\,\mathbf{a}_{1:\mathrm{T}},\,\Delta t} \quad \boxed{J(\mathbf{x}_{1:\mathrm{T}},\, \mathbf{a}_{1:\mathrm{T}},\, \Delta t)}$$

Integrators

$$\text{s.t.} \quad \boxed{\mathbf{f}(\mathbf{x}_{n+1},\, \mathbf{x}_n,\, \mathbf{a}_n,\, \Delta \mathbf{t}) = 0}$$

$$\boxed{\mathbf{C}(\mathbf{x}, \mathbf{a}, \Delta \mathbf{t}) \in C}$$

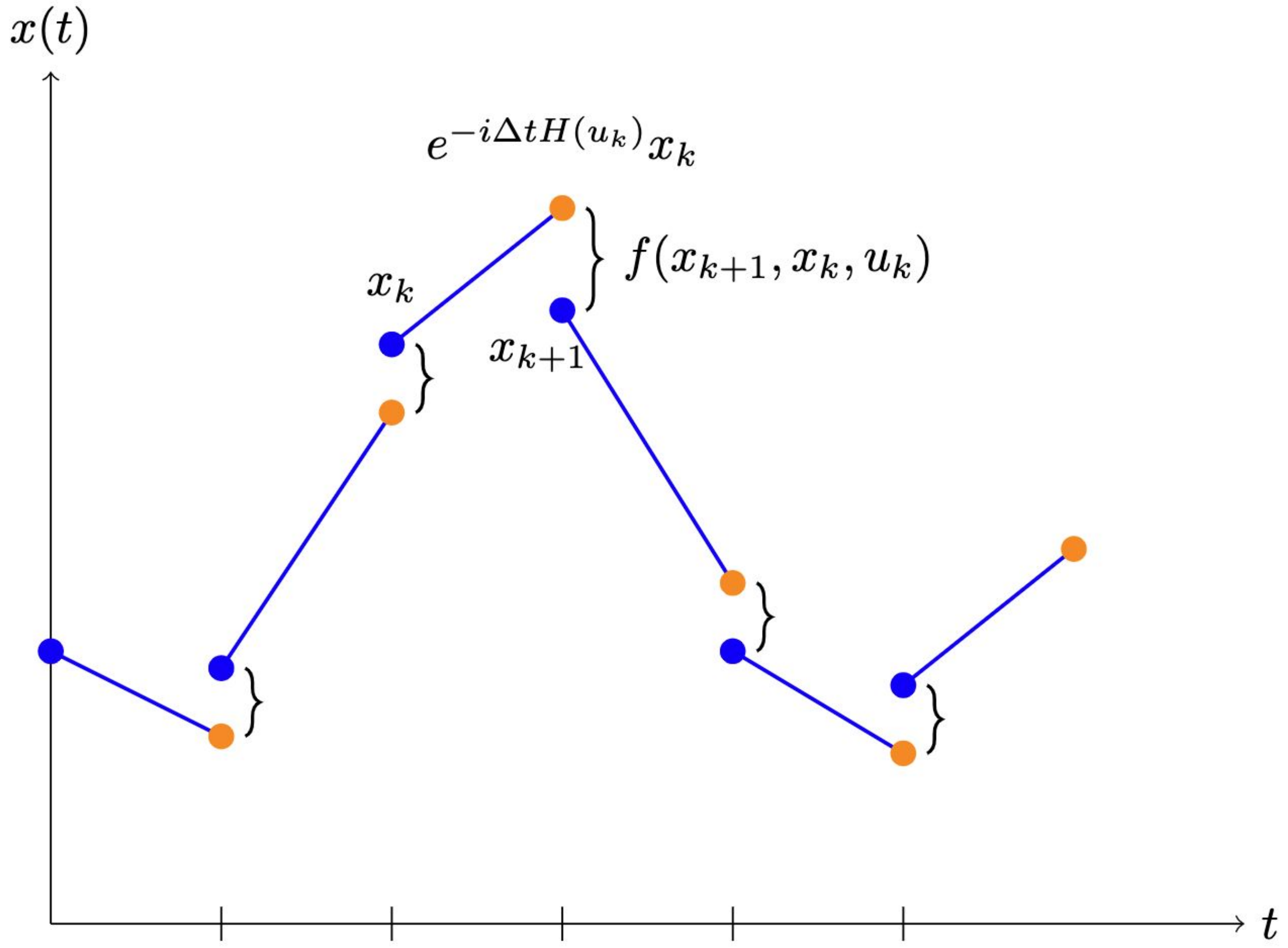
Constraints

Piccolo.jl

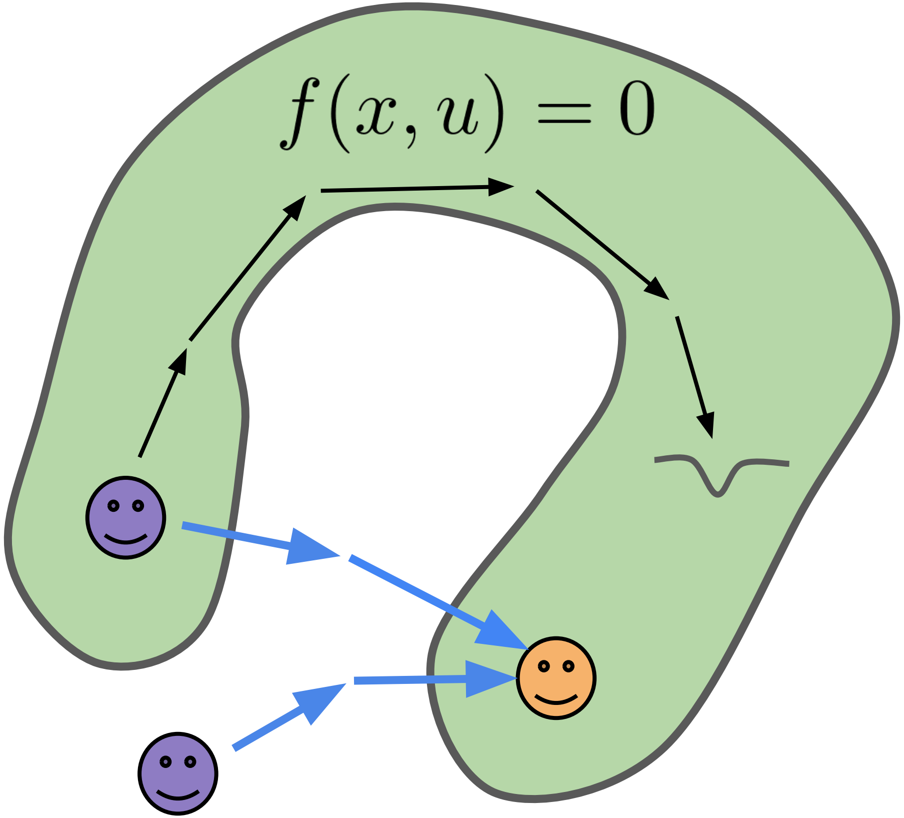
Example II.2

Example II.2

Takeaway: How to interpret Piccolo outputs



Piccolo.jl



Piccolo.jl

Exercises

- Gauss-Newton reminder
- Augmented Lagrangians

Part III

integrators

objectives

constraints

Parts of a QuantumControlProblem

Objectives

$$\min_{\mathbf{x}_{1:\mathrm{T}}, \mathbf{a}_{1:\mathrm{T}}, \Delta t} \quad \boxed{J(\mathbf{x}_{1:\mathrm{T}}, \mathbf{a}_{1:\mathrm{T}}, \Delta t)}$$

Integrators

$$\text{s.t.} \quad \boxed{\mathbf{f}(\mathbf{x}_{n+1}, \mathbf{x}_n, \mathbf{a}_n, \Delta \mathbf{t}) = 0}$$

$$\boxed{\mathbf{C}(\mathbf{x}, \mathbf{a}, \Delta \mathbf{t}) \in C}$$

Constraints

Objectives, Constraints, and Losses

Trajectories

- Objectives

$$J(\vec{\mathbf{Z}}),\ \nabla J(\vec{\mathbf{Z}})$$

- Constraints

$$\mathbf{C}(\vec{\mathbf{Z}}),\ \frac{\partial \mathbf{C}}{\partial \vec{\mathbf{Z}}}$$

Knot points

- Losses

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma, \ldots),$$
$$\nabla L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \ldots)$$

Pade Integrator COLLOcation

19 dubious ways to compute a matrix exponential

Explicit integrator: $$\mathbf{U}_{t+1} = \exp(-i\Delta t_t \mathbf{H}(\mathbf{a}_t))\mathbf{U}_t$$

Pade Integrator COLLOcation

19 dubious ways to compute a matrix exponential

Explicit integrator: $$\mathbf{U}_{t+1} = \exp(-i\Delta t_t \mathbf{H}(\mathbf{a}_t))\mathbf{U}_t$$

Implicit integrator: $$\mathbf{U}_{t+1} - \exp(-i\Delta t_t \mathbf{H}(\mathbf{a}_t))\mathbf{U}_t = 0$$

Pade Integrator COLLOcation

19 dubious ways to compute a matrix exponential

Explicit integrator: $$\mathbf{U}_{t+1} = \exp(-i\Delta t_t \mathbf{H}(\mathbf{a}_t))\mathbf{U}_t$$

Implicit integrator: $$\mathbf{U}_{t+1} - \exp(-i\Delta t_t \mathbf{H}(\mathbf{a}_t))\mathbf{U}_t = 0$$

$$\mathbf{U}_{t+1} - \mathbf{B}^{-1}(\mathbf{a}_t, \Delta t_t)\mathbf{F}(\mathbf{a}_t, \Delta t_t)\mathbf{U}_t \approx 0$$

C. Moler and C. Van Loan, SIAM Review 20, 801 (1978)

Pade Integrator COLLOcation

19 dubious ways to compute a matrix exponential

Explicit integrator: $$\mathbf{U}_{t+1} = \exp(-i\Delta t_t \mathbf{H}(\mathbf{a}_t))\mathbf{U}_t$$

Implicit integrator: $$\mathbf{U}_{t+1} - \exp(-i\Delta t_t \mathbf{H}(\mathbf{a}_t))\mathbf{U}_t = 0$$

$$\mathbf{U}_{t+1} - \mathbf{B}^{-1}(\mathbf{a}_t, \Delta t_t)\mathbf{F}(\mathbf{a}_t, \Delta t_t)\mathbf{U}_t \approx 0$$

$$\mathbf{B}(a_t, \Delta t_t)\mathbf{U}_{t+1} - \mathbf{F}(a_t, \Delta t_t)\mathbf{U}_t \approx 0$$

C. Moler and C. Van Loan, SIAM Review 20, 801 (1978)

Example III.1

Example III.1

Takeaway: How to build a quantum control problem

Problem templates

Packaging the creation of quantum control problems

```
function UnitarySmoothPulseProblem(
    system::AbstractQuantumSystem,
    operator::QuantumOperator,
    T::Int,
    Δt::Union{Float64, Vector{Float64}};
    ipopt_options::IpoptOptions=IpoptOptions(),
    piccolo_options::PiccoloOptions=PiccoloOptions(),...
)
```

An applications toolbox

Unitary control

UnitarySmoothPulseProblem
UnitaryBangBangProblem
UnitaryRobustnessProblem

UnitaryMinTimeProblem
UnitarySamplingProblem
UnitaryDirectSumProblem

Quantum state control

QuantumStateSmoothPulseProblem
QuantumStateMinTimeProblem

Density matrix control

Flexible design patterns

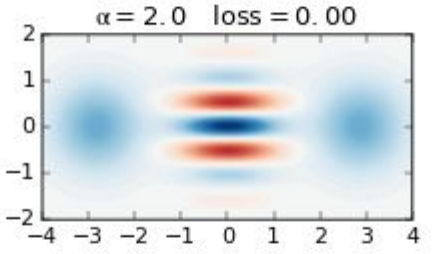
Optimize	Specialize	Coordinate
Smooth pulses	Minimum time	Direct sums
Bang-bang pulses	Hamiltonian robustness	Sampling-based robustness

Exercises

- Inspect a gradient for correctness
- Exploring problem templates

What’s next?

Piccolo.jl 1.0



Open quantum systems

Integrate with **QuantumOptics.jl, QuantumToolbox.jl**

Piccolo.jl 1.0

In progress

Summer 2024

Fall 2024

Winter 2024

End of year

Expand docs, CI, contribution, & style guides

QuantumCollocationCore.jl