



Quantum Optimal Control Theory

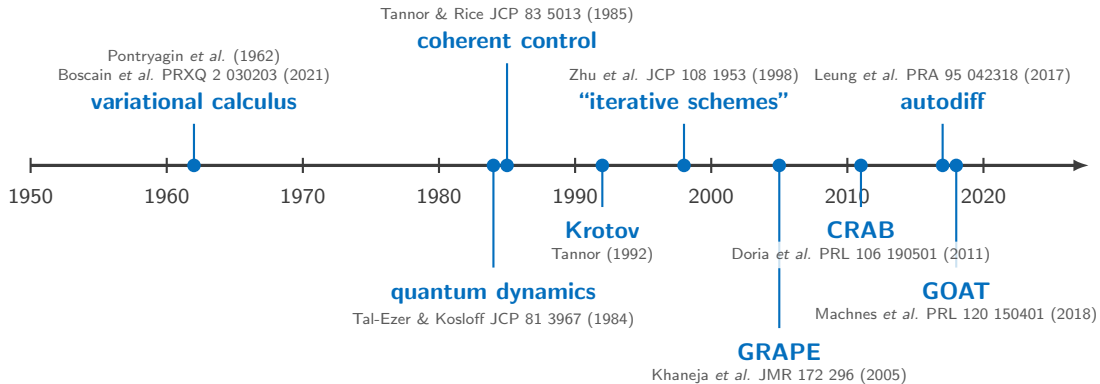
A Historical Overview

Michael H. Goerz

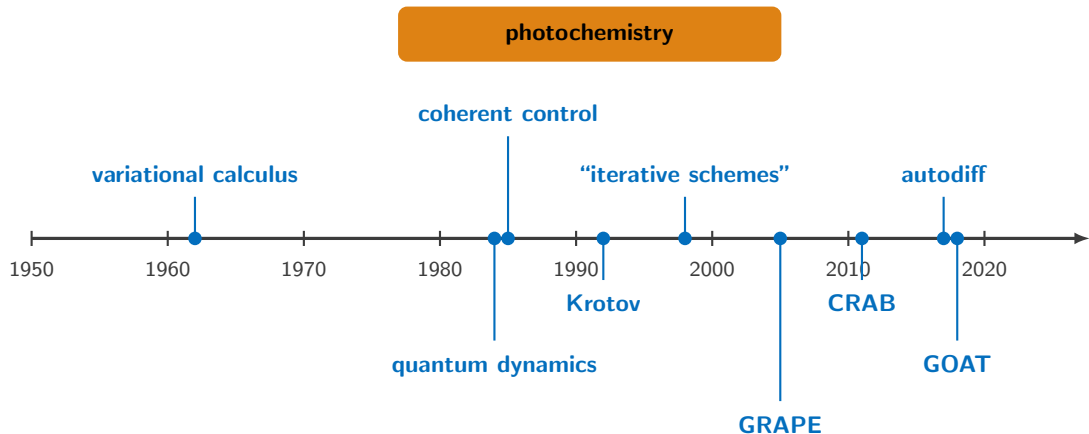
DEVCOM Army Research Lab

QNumerics Summer School, Amherst, MA, June 25, 2026

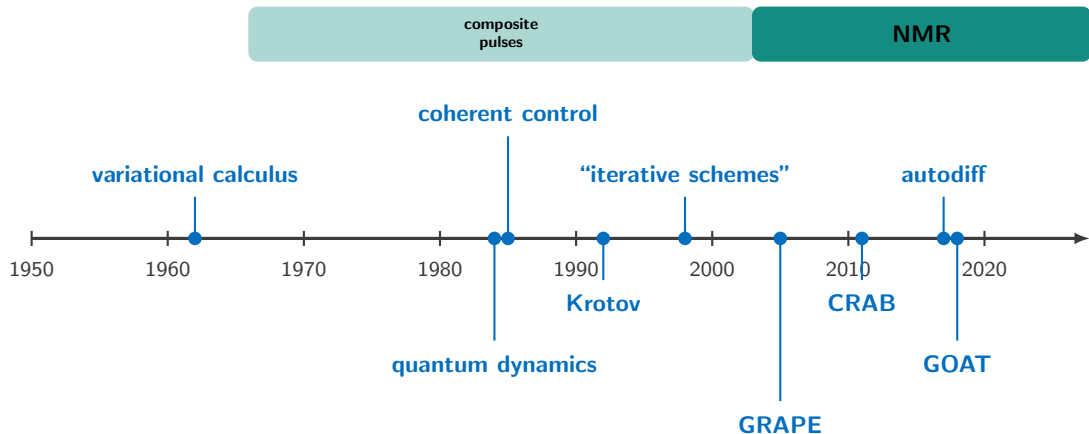
Timeline – Methods



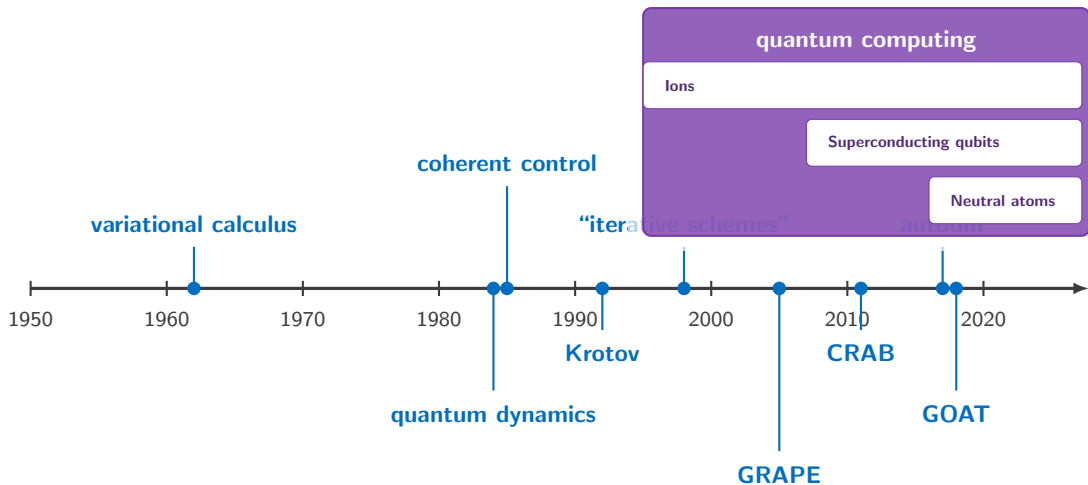
Timeline – Applications: Photochemistry



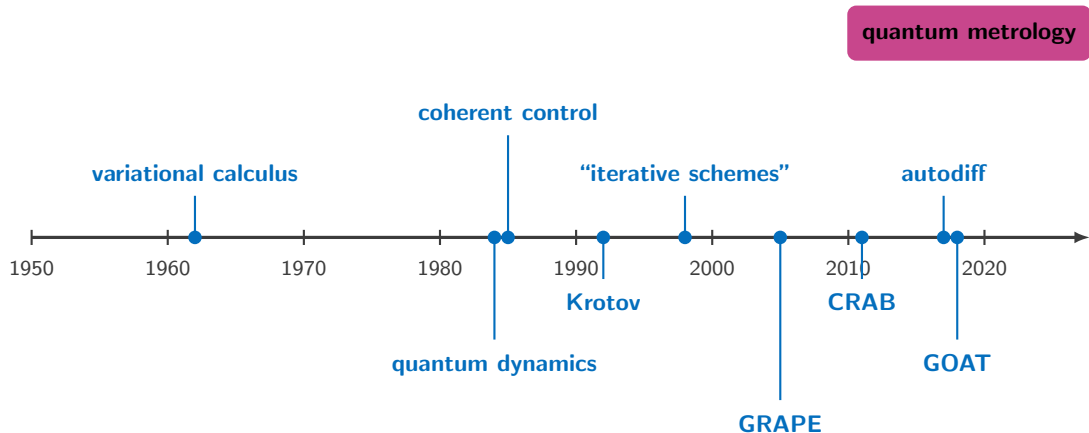
Timeline – Applications: NMR



Timeline – Applications: Quantum Computing



Timeline – Applications: Quantum Metrology



Quantum Optimal Control Theory

Steer a quantum system in some desired way

Optimization Functional:

$$J = J_T(\{|\psi_k(T)\rangle\}) + \int_0^T g_a(\{\epsilon_I(t)\}) dt + \int_0^T g_b(\{|\psi_k(t)\rangle\}) dt$$

$$\text{subject to } i\hbar \frac{\partial}{\partial t} |\psi_k(t)\rangle = \hat{H} |\psi_k(t)\rangle ; \quad \hat{H} = \hat{H}_0 + \hat{H}_1(\epsilon(t))$$

Quantum Optimal Control Theory

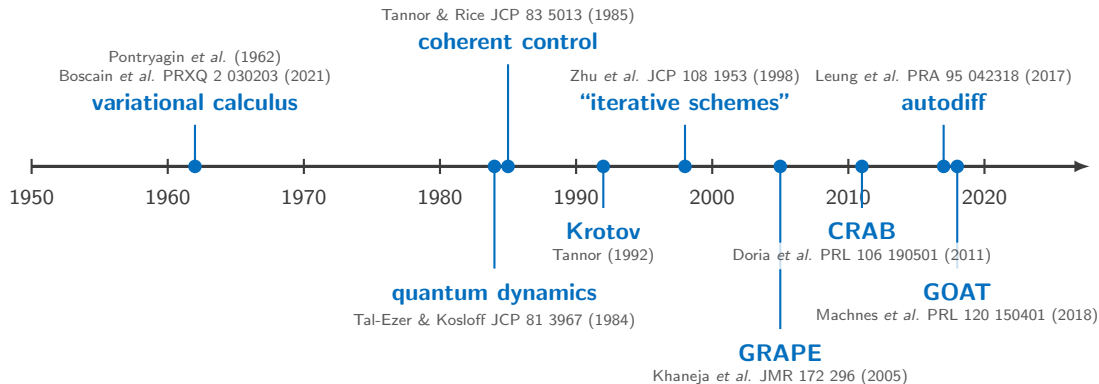
Steer a quantum system in some desired way

Optimization Functional:

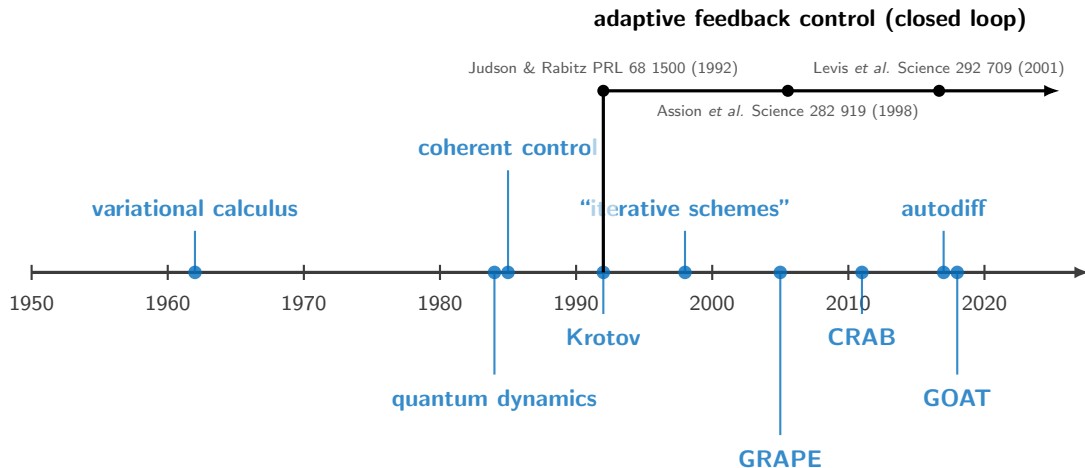
$$J = J_T(\{|\psi_k(T)\rangle\}) + \int_0^T g_a(\{\epsilon_I(t)\}) dt + \int_0^T g_b(\{|\psi_k(t)\rangle\}) dt$$

$$\text{subject to } i\hbar \frac{\partial}{\partial t} |\psi_k(t)\rangle = \hat{H} |\psi_k(t)\rangle ; \quad \hat{H} = \hat{H}_0 + \hat{H}_1(\epsilon(t))$$

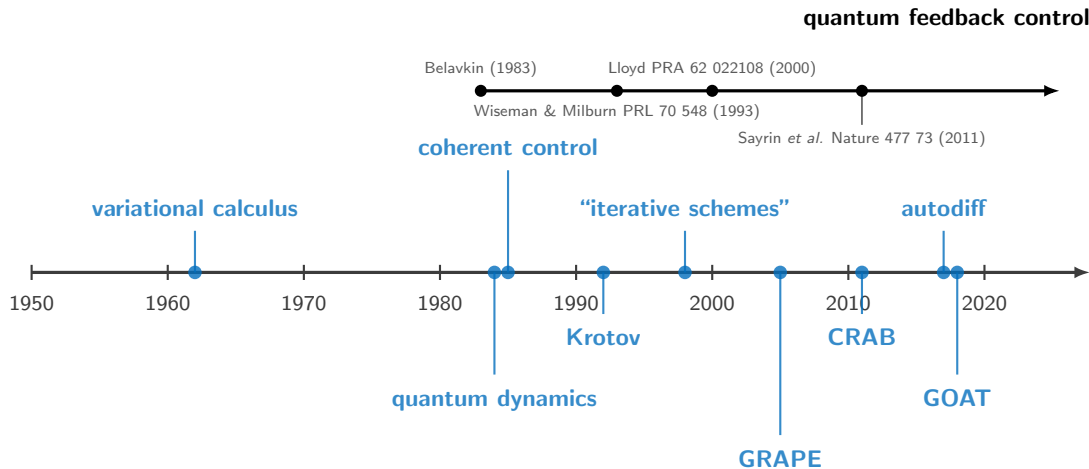
Timeline – Methods



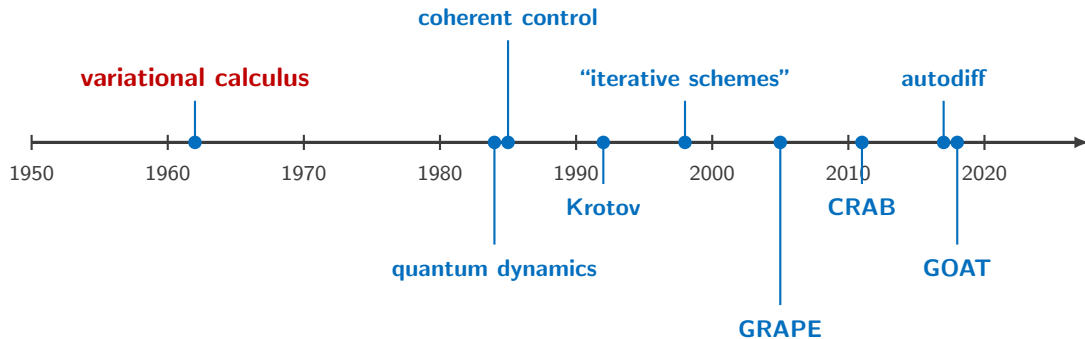
Timeline – Open Loop vs. Closed Loop



Timeline – Quantum Feedback Control



Origins: Variational Calculus



Origins: Variational Calculus

Optimization Functional:

$$J = J_T(\{|\psi_k(T)\rangle\}) + \int_0^T g_a(\{\epsilon_l(t)\}) dt + \int_0^T g_b(\{|\psi_k(t)\rangle\}) dt$$

$$\text{subject to } i\hbar \frac{\partial}{\partial t} |\psi_k(t)\rangle = \hat{H} |\psi_k(t)\rangle ; \quad \hat{H} = \hat{H}_0 + \hat{H}_1(\epsilon(t))$$

Origins: Variational Calculus

Optimization Functional:

$$J = J_T(\{|\psi_k(T)\rangle\}) + \int_0^T g_a(\{\epsilon_I(t)\}) dt + \int_0^T g_b(\{|\psi_k(t)\rangle\}) dt$$

$$\text{subject to } i\hbar \frac{\partial}{\partial t} |\psi_k(t)\rangle = \hat{H} |\psi_k(t)\rangle ; \quad \hat{H} = \hat{H}_0 + \hat{H}_1(\epsilon(t))$$

Lagrange Multipliers:

- Adjoin equation of motion to J via co-states $|\chi_k(t)\rangle$
- $|\psi_k\rangle$, $|\chi_k\rangle$, ϵ_I now independent
- Optimum: $\delta J = 0$

Pontryagin Maximum Principle

$$\mathcal{H} = \sum_k \operatorname{Im} \langle \chi_k | \hat{H} | \psi_k \rangle - g(\{\epsilon_I\}, t) \quad (\text{Pontryagin function})$$

$$\frac{\partial}{\partial \epsilon_I} \mathcal{H} = 0 \quad \Longrightarrow \quad \frac{\partial g}{\partial \epsilon_I} = \sum_k \operatorname{Im} \left\langle \chi_k \left| \frac{\partial \hat{H}}{\partial \epsilon_I} \right| \psi_k \right\rangle$$

$$i\hbar \frac{\partial}{\partial t} |\chi_k(t)\rangle = \hat{H}^\dagger |\chi_k(t)\rangle, \quad |\chi_k(T)\rangle = -\frac{\partial J_T}{\partial \langle \psi_k(T) |}$$

Pontryagin Maximum Principle

“The application of the PMP is not so straightforward. Indeed, there are many conditions to satisfy and all of them are coupled.”

— Boscain *et al.*, PRX Quantum 2, 030203 (2021)

Pontryagin Maximum Principle

$$\mathcal{H} = \sum_k \operatorname{Im} \langle \chi_k | \hat{H} | \psi_k \rangle - g(\{\epsilon_I\}, t) \quad (\text{Pontryagin function})$$

$$\frac{\partial}{\partial \epsilon_I} \mathcal{H} = 0 \quad \Longrightarrow \quad \frac{\partial g}{\partial \epsilon_I} = \sum_k \operatorname{Im} \left\langle \chi_k \left| \frac{\partial \hat{H}}{\partial \epsilon_I} \right| \psi_k \right\rangle$$

$$i\hbar \frac{\partial}{\partial t} |\chi_k(t)\rangle = \hat{H}^\dagger |\chi_k(t)\rangle, \quad |\chi_k(T)\rangle = -\frac{\partial J_T}{\partial \langle \psi_k(T) |}$$

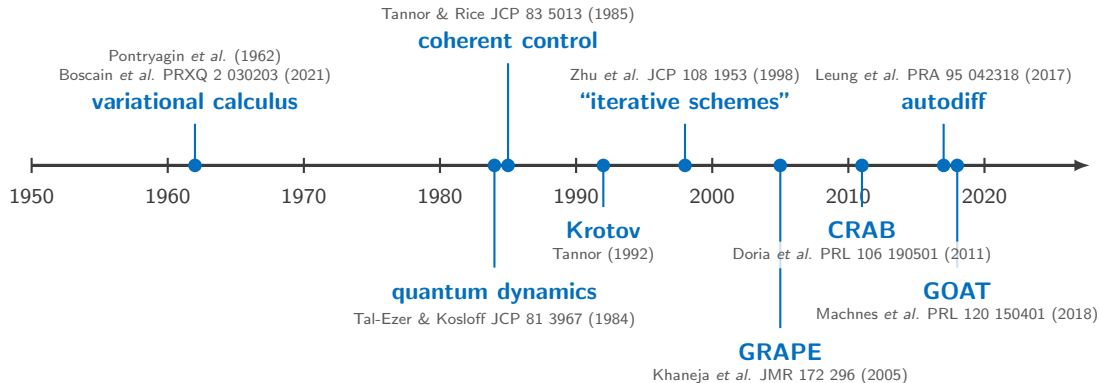
Pontryagin Maximum Principle

$$\mathcal{H} = \sum_k \operatorname{Im} \langle \chi_k | \hat{H} | \psi_k \rangle - g(\{\epsilon_I\}, t) \quad (\text{Pontryagin function})$$

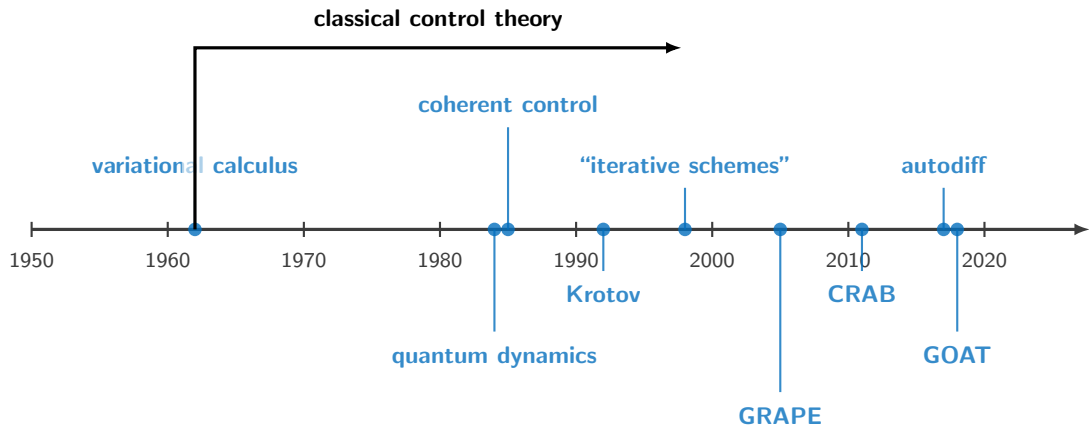
$$\frac{\partial}{\partial \epsilon_I} \mathcal{H} = 0 \quad \Longrightarrow \quad \frac{\partial g}{\partial \epsilon_I} = \sum_k \operatorname{Im} \left\langle \chi_k \left| \frac{\partial \hat{H}}{\partial \epsilon_I} \right| \psi_k \right\rangle$$

$$i\hbar \frac{\partial}{\partial t} |\chi_k(t)\rangle = \hat{H}^\dagger |\chi_k(t)\rangle, \quad |\chi_k(T)\rangle = -\frac{\partial J_T}{\partial \langle \psi_k(T) |}$$

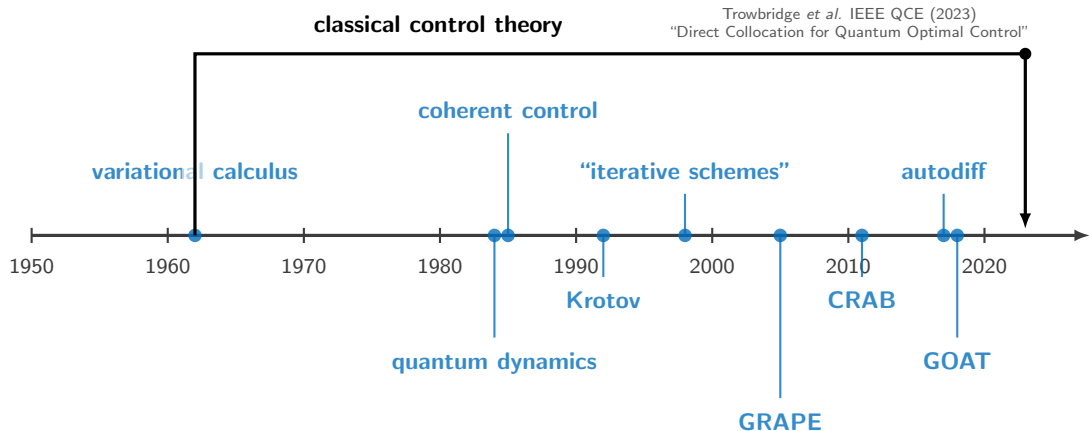
Timeline – Methods



Timeline – Classical Control Theory



Timeline – Classical Control Theory



Quantum Optimal Control Theory

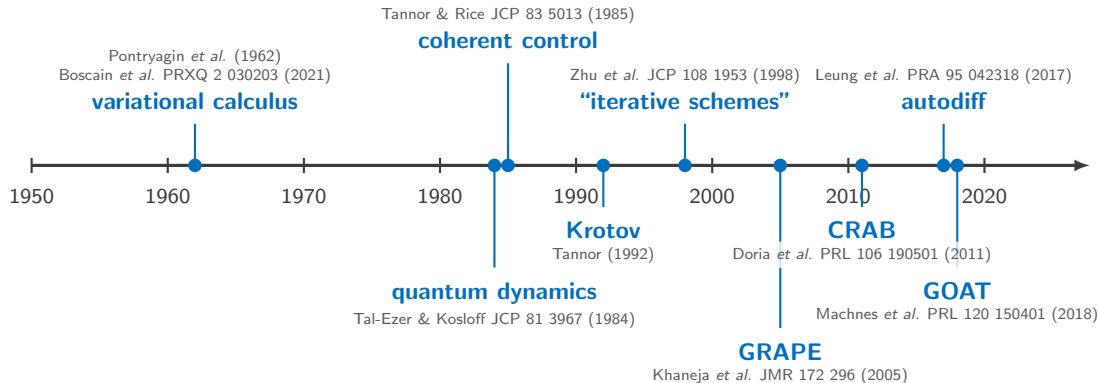
Steer a quantum system in some desired way

Optimization Functional:

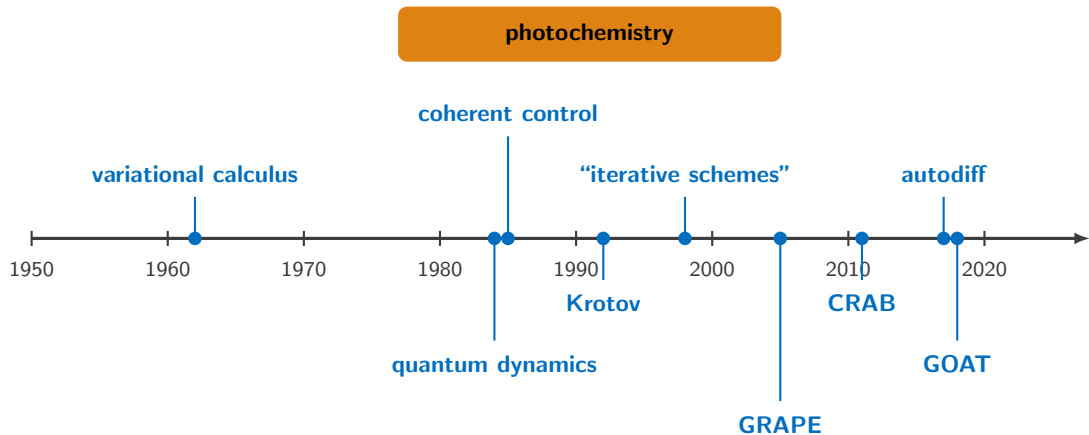
$$J = J_T(\{|\psi_k(T)\rangle\}) + \int_0^T g_a(\{\epsilon_I(t)\}) dt + \int_0^T g_b(\{|\psi_k(t)\rangle\}) dt$$

$$\text{subject to } i\hbar \frac{\partial}{\partial t} |\psi_k(t)\rangle = \hat{H} |\psi_k(t)\rangle ; \quad \hat{H} = \hat{H}_0 + \hat{H}_1(\epsilon(t))$$

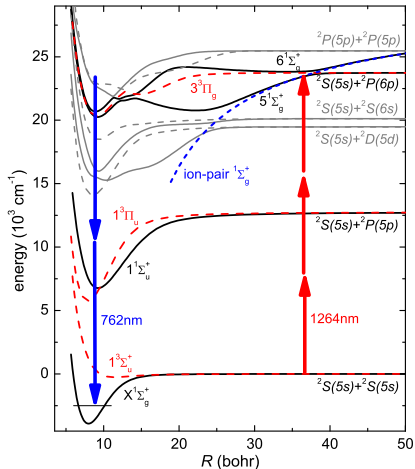
Timeline – Methods



Timeline – Applications: Photochemistry



Photochemistry



— Tomza *et al.*, Phys. Rev. A 86, 043424 (2012)

- Reaction selectivity (pump–dump)
Tannor, Rice, J. Chem. Phys. 83, 5013 (1985)
- Coherent control of products
Brumer, Shapiro, Chem. Phys. Lett. 126, 541 (1986)
- Selectivity via pulse shaping
Kosloff *et al.*, Chem. Phys. 139, 201 (1989)
- Bond dissociation & rearrangement
Levis *et al.*, Science 292, 709 (2001)
- Monograph
Brumer, Shapiro, *Principles and Applications of the Quantum Control of Molecular Processes* (Wiley, 2003)

Photochemistry

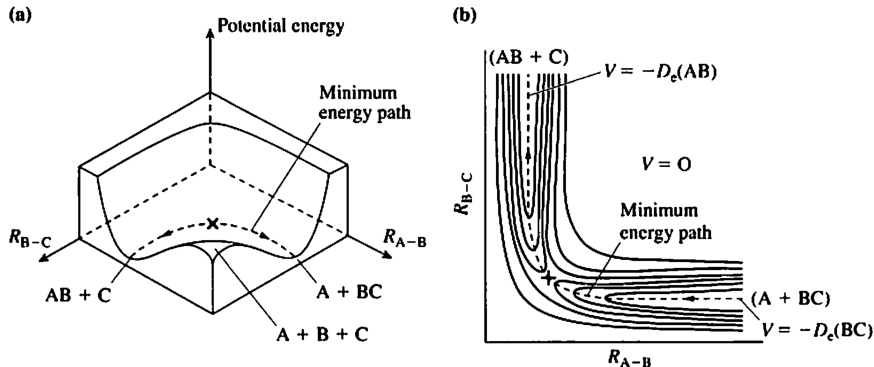
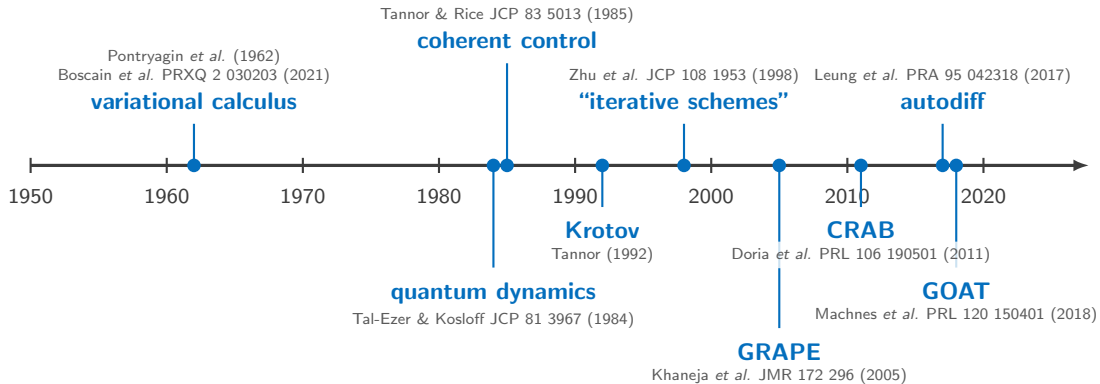


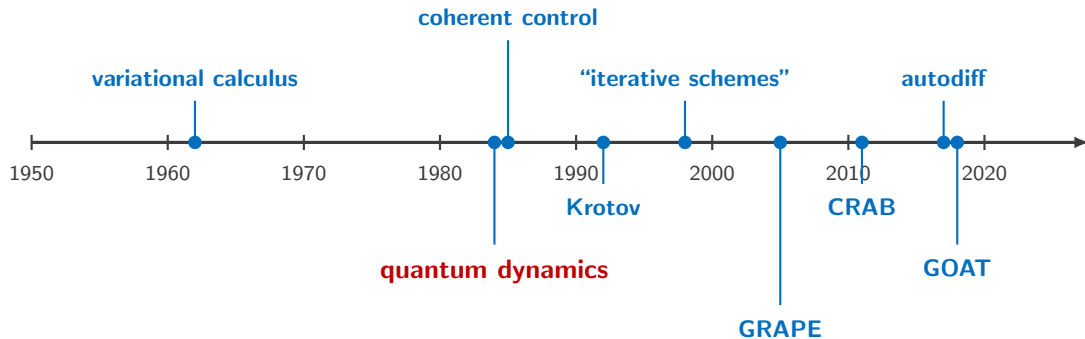
Figure 12.8 Potential energy surface for collinear $A + BC \rightarrow AB + C$. (a) stereoscopic projection; (b) equipotential contours. Adapted from Levine (1974). Note that the “transition state” between reactants and products has a saddle-like structure: a maximum along the reaction coordinate and a minimum in the perpendicular coordinate(s).

— Tannor, Introduction to Quantum Mechanics: A Time-Dependent Perspective (2007)

Timeline – Methods



Quantum Dynamics



Quantum Dynamics

1984

An accurate and efficient scheme for propagating the time dependent Schrödinger equation

H. Tal-Ezer

School of Mathematical Science, Tel Aviv University, Tel Aviv, Israel

R. Kosloff

Department of Physical Chemistry and the Fritz Haber Molecular Dynamics Research Center, The Hebrew University, Jerusalem, 91904 Israel

J. Chem. Phys. **81** (9), 1 November 1984

0021-9606/84/213967-05\$02.10

© 1984 American Institute of Physics

3967

Quantum Dynamics

1984

An accurate and efficient scheme for propagating the time dependent Schrödinger equation

1986

**Coherent pulse sequence induced control of selectivity of reactions:
Exact quantum mechanical calculations**

David J. Tannor

The Department of Chemistry and The James Franck Institute, The University of Chicago, Chicago, Illinois 60637

Ronnie Kosloff

Fritz Haber Institute for Molecular Dynamics, Hebrew University, Jerusalem, Israel 91904

Stuart A. Rice

The Department of Chemistry and The James Franck Institute, The University of Chicago, Chicago, Illinois 60637

J. Chem. Phys. **85** (10), 15 November 1986 0021-9606/86/225805-16\$02.10

© 1986 American Institute of Physics

5805

Quantum Dynamics

1984

An accurate and efficient scheme for propagating the time dependent Schrödinger equation

1986

**Coherent pulse sequence induced control of selectivity of reactions:
Exact quantum mechanical calculations**

1988

J. Phys. Chem. **1988**, 92, 2087–2100

2087

Time-Dependent Quantum-Mechanical Methods for Molecular Dynamics

Ronnie Kosloff

*Department of Physical Chemistry and The Fritz Haber Research Center for Molecular Dynamics,
The Hebrew University, Jerusalem 91904, Israel (Received: December 7, 1987)*

J. Chem. Phys. **85** (10), 15 November 1986 0021-9606/86/225805-16\$02.10

© 1986 American Institute of Physics

5805

Quantum Dynamics

1984

An accurate and efficient scheme for propagating the time dependent Schrödinger equation

1986

**Coherent pulse sequence induced control of selectivity of reactions:
Exact quantum mechanical calculations**

1988

J. Phys. Chem. **1988**, 92, 2087–2100

2087

Time-Dependent Quantum-Mechanical Methods for Molecular Dynamics

1992

J. Phys. A: Math. Gen. **25** (1992) 1283–1307. Printed in the UK

**Solution of the time-dependent Liouville–von Neumann
equation: dissipative evolution**

Michael Berman[†], Ronnie Kosloff[†] and Hillel Tal-Ezer[‡]

Quantum Dynamics

1984

An accurate and efficient scheme for propagating the time dependent Schrödinger equation

1986

**Coherent pulse sequence induced control of selectivity of reactions:
Exact quantum mechanical calculations**

1988

J. Phys. Chem. **1988**, 92, 2087–2100

2087

Time-Dependent Quantum-Mechanical Methods for Molecular Dynamics

1992

J. Phys. A: Math. Gen. **25** (1992) 1283–1307. Printed in the UK

Solution of the time-dependent Liouville–von Neumann

1994

Annu. Rev. Phys. Chem. **1994**, 45: 145–78

Copyright © 1994 by Annual Reviews Inc. All rights reserved

**PROPAGATION METHODS FOR
QUANTUM MOLECULAR
DYNAMICS**

Ronnie Kosloff

Chebyshev Propagation

Chebyshev Polynomials

$$P_0(x) = 1; \quad P_1(x) = x; \quad P_n(x) = 2xP_{n-1}(x) - P_{n-2}(x)$$

$P_n(x)$ are defined for $x \in [-1, 1]$

Chebyshev Propagation

Chebyshev Polynomials

$$P_0(x) = 1; \quad P_1(x) = x; \quad P_n(x) = 2xP_{n-1}(x) - P_{n-2}(x)$$

$P_n(x)$ are defined for $x \in [-1, 1]$

$$|\Psi(t + dt)\rangle = e^{-i\hat{H} dt} |\Psi(t)\rangle = \sum_n a_n \underbrace{P_n(-i\hat{H}_{\text{norm}})}_{\equiv |\Phi_n\rangle} |\Psi(t)\rangle,$$

Chebyshev Propagation

Chebyshev Polynomials

$$P_0(x) = 1; \quad P_1(x) = x; \quad P_n(x) = 2xP_{n-1}(x) - P_{n-2}(x)$$

$P_n(x)$ are defined for $x \in [-1, 1]$

$$|\Psi(t + dt)\rangle = e^{-i\hat{H} dt} |\Psi(t)\rangle = \sum_n a_n \underbrace{P_n(-i\hat{H}_{\text{norm}})}_{\equiv |\Phi_n\rangle} |\Psi(t)\rangle,$$

$$\hat{H}_{\text{norm}} = 2 \frac{\hat{H} - E_{\min} \mathbf{1}}{\Delta} - \mathbf{1}, \quad a_n = (2 - \delta_{n0}) e^{-\frac{i}{\hbar} \left(\frac{\Delta}{2} + E_{\min} \right) dt} J_n(\alpha),$$

$$|\Phi_0(x)\rangle = |\Psi(t)\rangle; \quad |\Phi_1(x)\rangle = -i\hat{H}_{\text{norm}} |\Phi_0\rangle; \quad |\Phi_n(x)\rangle = -2i\hat{H}_{\text{norm}} |\Phi_{n-1}\rangle + |\Phi_{n-2}\rangle$$

Chebyshev Propagation – Pseudocode

Algorithm 2 CHEBYCHEV-PROPAGATOR Evaluate $\vec{w} = f(\pm \hat{A} dt) \vec{v}$, with $f(\pm \hat{A} dt) = e^{\pm i \hat{A} dt}$.

Input: input vector $\vec{v} \in \mathbb{C}^N$; operator $\hat{A} \in \mathbb{C}^{N \times N}$; time step dt ;
Output: Approximation of propagated vector $\vec{w} = e^{-i \hat{A} dt} \vec{v} \in \mathbb{C}^N$

```

1: procedure CHEBY( $\vec{v}, \hat{A}, dt$ )
2:    $\Delta =$  spectral radius of  $\hat{A}$ 
3:    $E_{\min} =$  minimum eigenvalue of  $\hat{A}$ 
4:    $[a_0 \dots a_n] = \text{EXPCHEBYCOEFFS}(\Delta, E_{\min}, dt)$ 
5:    $d = \frac{1}{2}\Delta$ ;  $\beta = d + E_{\min}$ 
6:    $\vec{v}_0 = \vec{v}$ 
7:    $\vec{w}^{(0)} = a_0 \vec{v}_0$ 
8:    $\vec{v}_1 = \pm \frac{i}{d} (\hat{A} \vec{v}_0 - \beta \vec{v}_0)$ 
9:    $\vec{w}^{(1)} = \vec{w}^{(0)} + a_1 \vec{v}_1$ 
10:  for  $i = 2 : n$  do
11:     $\vec{v}_i = \pm \frac{2i}{d} (\hat{A} \vec{v}_{i-1} - \beta \vec{v}_{i-1}) + \vec{v}_{i-2}$ 
12:     $\vec{w}^{(i)} = \vec{w}^{(i-1)} + a_i \vec{v}_i$ 
13:  end for
14:  return  $e^{\pm i \beta dt} \vec{w}^{(n)}$ 
15: end procedure

```

Algorithm 3 CHEBYCHEVCOEFFICIENTS for $f(\pm \hat{A} dt) = e^{\pm i \hat{A} dt}$.

Input: spectral radius Δ of $\hat{A}$; minimum eigenvalue $E_{\min}$ of $\hat{A}$; time step dt

Output: Array of Chebyshev coefficients $[a_0 \dots a_n]$ allowing to approximate $f(\hat{A} dt)$ to pre-defined precision.

```

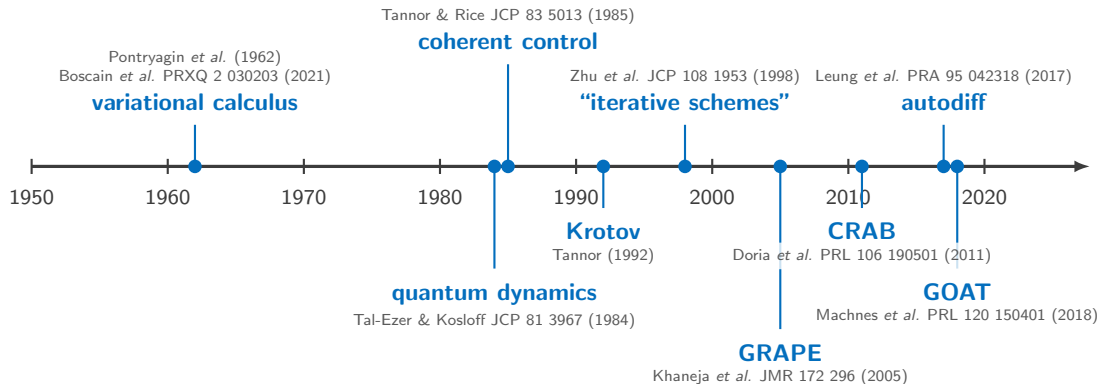
1: procedure EXPCHEBYCOEFFS( $\Delta, E_{\min}, dt$ )
2:    $\alpha = \frac{1}{2}\Delta dt$ 
3:    $a_0 = J_0(\alpha)$   $\triangleright$  0'th order Bessel-function of first kind
4:   for  $i = 1 : n_{\max} \approx 4 \lfloor \alpha \rfloor$  do
5:      $a_i = 2J_i(\alpha)$   $\triangleright$   $i$ 'th order Bessel-function of first kind
6:     if  $|a_i| < \text{limit}$  then exit loop with  $n = i$ 
7:   end for
8:   return  $[a_0, \dots, a_n]$ 
9: end procedure

```

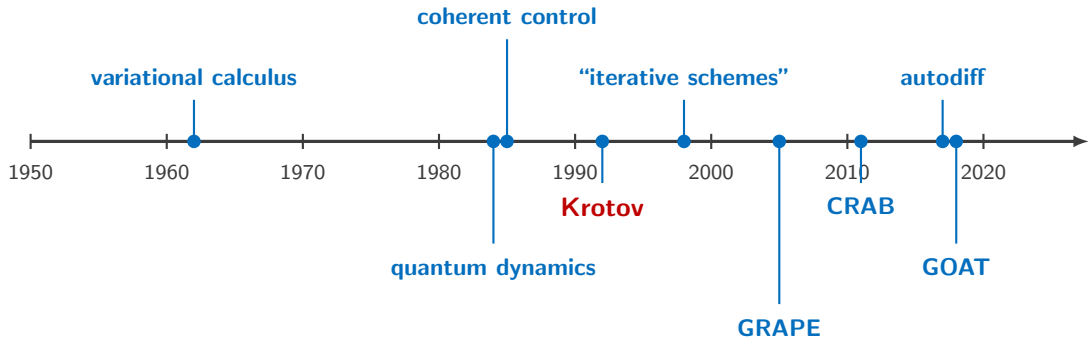
Goerz, PhD Thesis, Appendix F

<https://michaelgoerz.net>

Timeline – Methods



Krotov's Method



Krotov's Method

Classical Control Theory

- Krotov & Feldman, Eng. Cybern. 21, 123 (1983)
- Krotov, Control & Cybern. 17, 115 (1988)
- Krotov, *Global Methods in Optimal Control Theory* (Dekker, 1996)
- Konnov & Krotov, Autom. Remote Control 60, 1427 (1999)

Krotov's Method

Classical Control Theory

- Krotov & Feldman, Eng. Cybern. 21, 123 (1983)
- Krotov, Control & Cybern. 17, 115 (1988)
- Krotov, *Global Methods in Optimal Control Theory* (Dekker, 1996)
- Konnov & Krotov, Autom. Remote Control 60, 1427 (1999)

Quantum Control

- Tannor *et al.*, in *Time-Dependent Quantum Molecular Dynamics* (Plenum, 1992)
- Somló *et al.*, Chem. Phys. 172, 85 (1993)
- Palao & Kosloff, Phys. Rev. A 68, 062308 (2003)
- Eitan *et al.*, Phys. Rev. A 83, 053426 (2011)
- Reich *et al.*, J. Chem. Phys. 136, 104103 (2012)

Krotov's Method

$$J = J_T(\{|\psi_k(T)\rangle\}) + \int_0^T g_a(\{\epsilon_I(t)\}) dt + \int_0^T g_b(\{|\psi_k(t)\rangle\}) dt$$

Given: guess $\epsilon_I^{(0)}(t)$. **Find:** new field $\epsilon_I^{(1)}(t)$ with $J(\{\epsilon_I^{(1)}(t)\}) \leq J(\{\epsilon_I^{(0)}(t)\})$

Necessary & sufficient condition:

$$\frac{\partial g_a}{\partial \epsilon_I^{(1)}} = 2\text{Im} \sum_k \left\langle \chi_k^{(0)}(t) \left| \frac{\partial \hat{H}}{\partial \epsilon_I} \right| \psi_k^{(1)}(t) \right\rangle, \quad |\chi_k^{(0)}(T)\rangle = -\frac{\partial J_T}{\partial \langle \psi_k(T) |}$$

Choose g_a quadratic in the update:

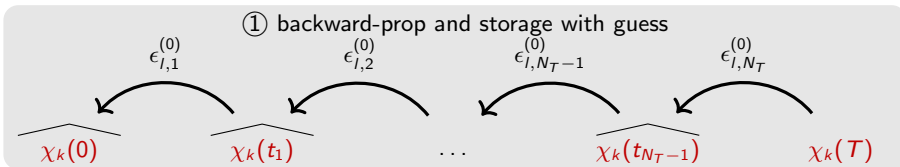
$$g_a = \frac{\lambda_a}{S(t)} (\Delta \epsilon_I(t))^2, \quad \Delta \epsilon_I(t) = \epsilon_I^{(1)}(t) - \epsilon_I^{(0)}(t)$$

$$\Rightarrow \Delta \epsilon_I(t) = \frac{S(t)}{\lambda_a} \text{Im} \sum_k \left\langle \chi_k^{(0)}(t) \left| \frac{\partial \hat{H}}{\partial \epsilon_I} \right| \psi_k^{(1)}(t) \right\rangle$$

Krotov Numerical Scheme

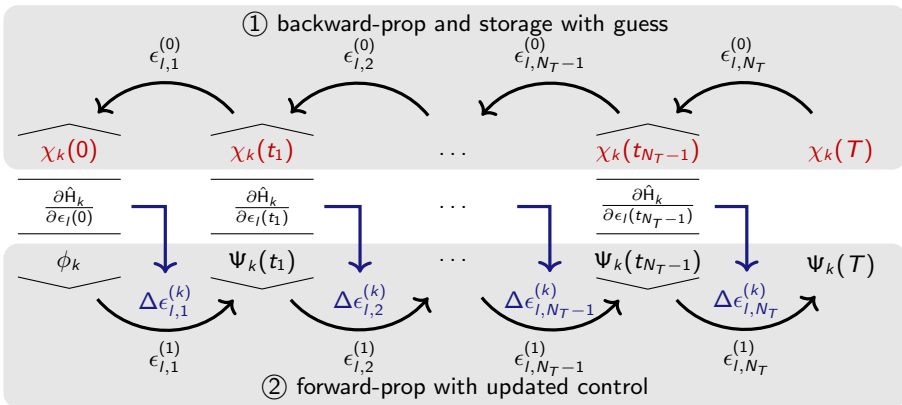
— Goerz *et al.* Quantum 6, 871 (2022)

Krotov Numerical Scheme



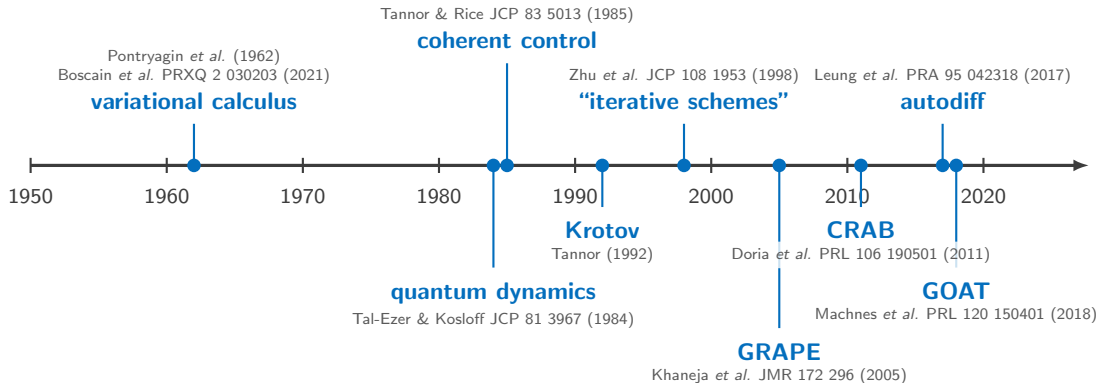
— Goerz *et al.* Quantum 6, 871 (2022)

Krotov Numerical Scheme

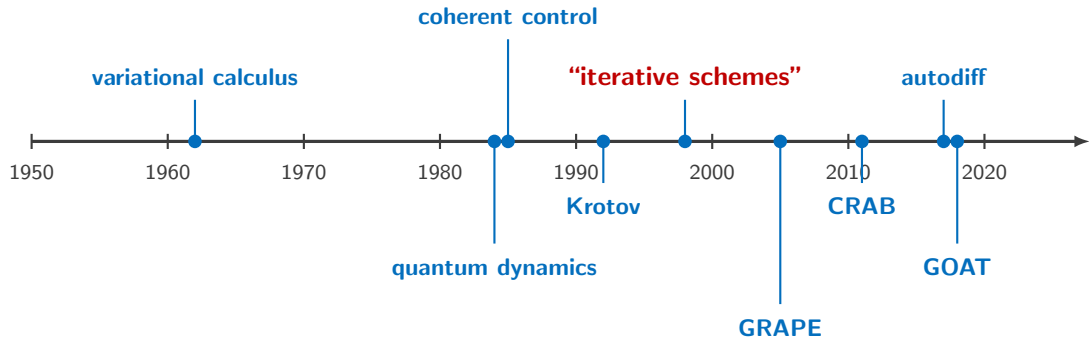


— Goerz *et al.* Quantum 6, 871 (2022)

Timeline – Methods



Iterative Schemes



Iterative Schemes

Concurrent update:

$$\Delta\epsilon_l^{(i)}(t) \propto \text{Im} \langle \chi_k^{(i-1)}(t) | \frac{\partial \hat{H}}{\partial \epsilon_l(t)} | \phi_k^{(i-1)}(t) \rangle$$

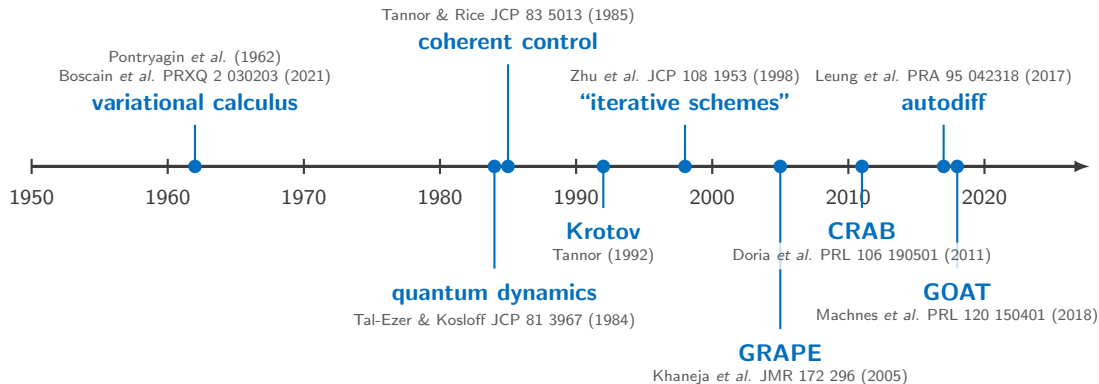
Monotonic convergence (Zhu–Rabitz):

$$\epsilon_l^{(i)}(t) = (1 - \delta) \tilde{\epsilon}_l^{(i-1)}(t) - \frac{\delta}{\alpha} \text{Im} \langle \chi_k^{(i-1)}(t) | \frac{\partial \hat{H}}{\partial \epsilon_l(t)} | \phi_k^{(i)}(t) \rangle$$

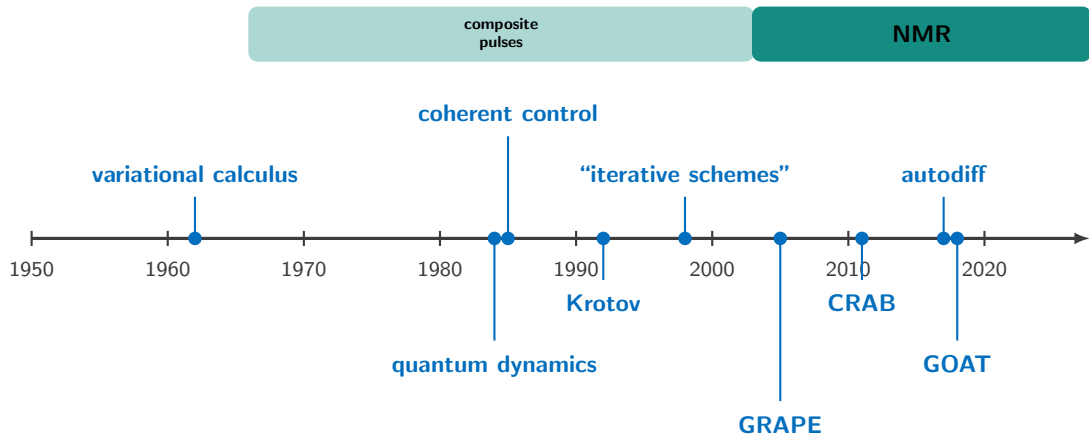
$$\tilde{\epsilon}_l^{(i)}(t) = (1 - \eta) \epsilon_l^{(i-1)}(t) - \frac{\eta}{\alpha} \text{Im} \langle \chi_k^{(i)}(t) | \frac{\partial \hat{H}}{\partial \epsilon_l(t)} | \phi_k^{(i)}(t) \rangle$$

— Zhu *et al.*, J. Chem. Phys. 108, 1953 (1998)

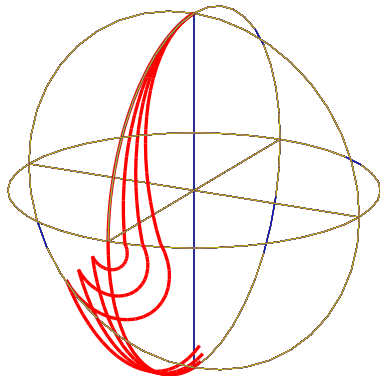
Timeline – Methods



Timeline – Applications: NMR



Nuclear Magnetic Resonance



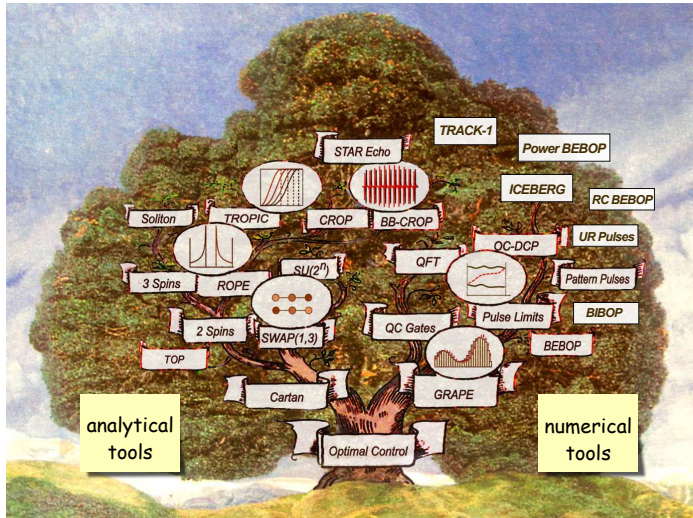
Composite 180° rotation: pulse errors cancel

— Vandersypen & Chuang,
Rev. Mod. Phys. 76, 1037 (2005)

Pulse design since the 1960s

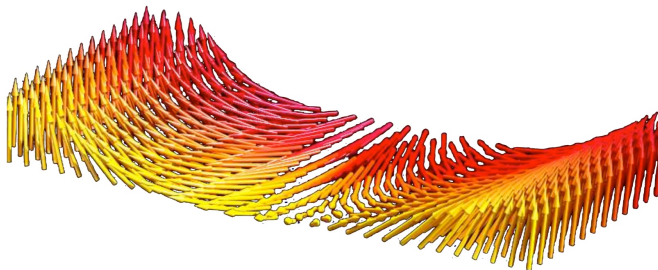
- Bloch equation: RF field steers spin
 - Robust over ensembles:
chemical-shift offset, B_1 inhomogeneity
 - Analytic pulse families:
composite, adiabatic, shaped
-
- Bloch, Phys. Rev. 70, 460 (1946)
 - Conolly *et al.*, IEEE Trans. Med. Imaging 5, 106 (1986)
 - Levitt, Prog. NMR Spectrosc. 18, 61 (1986)
 - Cory *et al.*, Physica D 120, 82 (1998)
 - Glaser *et al.*, Science 280, 421 (1998)
 - Skinner *et al.*, J. Magn. Reson. 163, 8 (2003)

Nuclear Magnetic Resonance



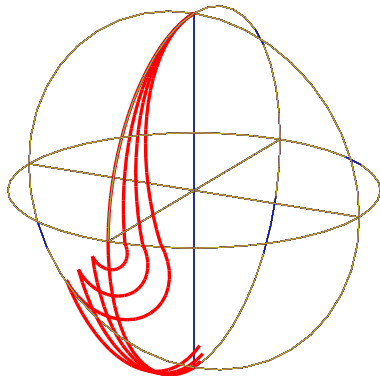
— Glaser, “NMR Control Overview,” KITP (2009)

Nuclear Magnetic Resonance



— Glaser, "NMR Control Overview," KITP (2009)

Nuclear Magnetic Resonance



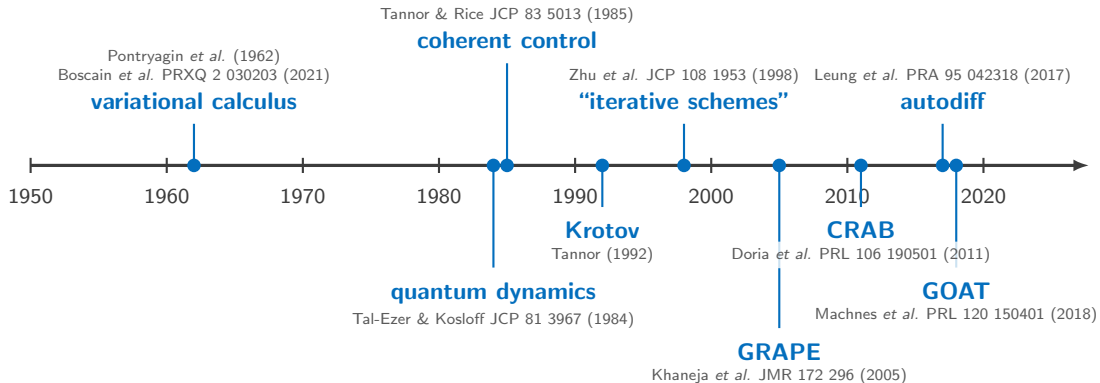
Composite 180° rotation: pulse errors cancel

— Vandersypen & Chuang,
Rev. Mod. Phys. 76, 1037 (2005)

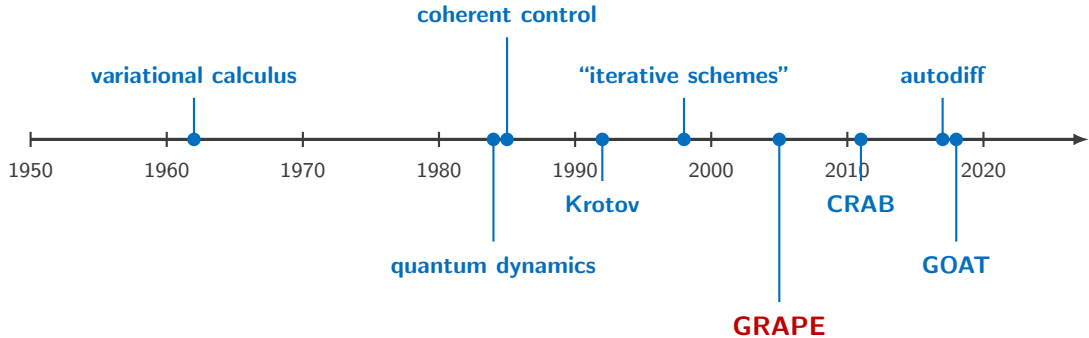
Pulse design since the 1960s

- Bloch equation: RF field steers spin
 - Robust over ensembles:
chemical-shift offset, B_1 inhomogeneity
 - Analytic pulse families:
composite, adiabatic, shaped
-
- Bloch, Phys. Rev. 70, 460 (1946)
 - Conolly *et al.*, IEEE Trans. Med. Imaging 5, 106 (1986)
 - Levitt, Prog. NMR Spectrosc. 18, 61 (1986)
 - Cory *et al.*, Physica D 120, 82 (1998)
 - Glaser *et al.*, Science 280, 421 (1998)
 - Skinner *et al.*, J. Magn. Reson. 163, 8 (2003)

Timeline – Methods

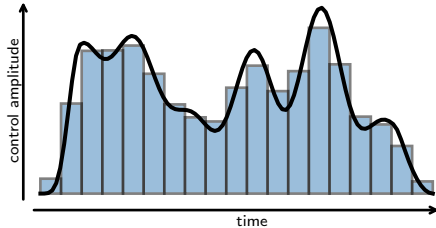


GRAPE



GRAPE

Khaneja *et al.*, J. Magn. Reson. 172, 296 (2005)



Discretize first!

Piecewise constant: $\hat{H}_n = \hat{H}(\{\epsilon_{nl}\})$ with $\epsilon_{nl} = \epsilon_l(t = t_n)$ for n 'th time slice

$$J(\{\epsilon_{nl}\}) = J_T(\{|\Psi_k(T)\rangle\}) + \int_0^T \dots dt$$

Gradient $\nabla J \equiv \frac{\partial J}{\partial \epsilon_{nl}} \Rightarrow$ LBFGS

Aside: Wirtinger Derivatives — Derivatives w.r.t. Complex Numbers

$$J_T(\{z_k\}) = J_T(\{\operatorname{Re}[z_k], \operatorname{Im}[z_k]\}); \quad J_T \in \mathbb{R}, \quad z_k \in \mathbb{C}$$

Aside: Wirtinger Derivatives — Derivatives w.r.t. Complex Numbers

$$J_T(\{z_k\}) = J_T(\{\operatorname{Re}[z_k], \operatorname{Im}[z_k]\}); \quad J_T \in \mathbb{R}, \quad z_k \in \mathbb{C}$$

$$\frac{\partial J_T(\{z_k\})}{\partial \epsilon_{nl}} = \sum_k \left(\frac{\partial J_T}{\partial \operatorname{Re}[z_k]} \frac{\partial \operatorname{Re}[z_k]}{\partial \epsilon_{nl}} + \frac{\partial J_T}{\partial \operatorname{Im}[z_k]} \frac{\partial \operatorname{Im}[z_k]}{\partial \epsilon_{nl}} \right); \quad \epsilon_{nl} \in \mathbb{R}$$

Aside: Wirtinger Derivatives — Derivatives w.r.t. Complex Numbers

$$J_T(\{z_k\}) = J_T(\{\operatorname{Re}[z_k], \operatorname{Im}[z_k]\}); \quad J_T \in \mathbb{R}, \quad z_k \in \mathbb{C}$$

$$\frac{\partial J_T(\{z_k\})}{\partial \epsilon_{nl}} = \sum_k \left(\frac{\partial J_T}{\partial \operatorname{Re}[z_k]} \frac{\partial \operatorname{Re}[z_k]}{\partial \epsilon_{nl}} + \frac{\partial J_T}{\partial \operatorname{Im}[z_k]} \frac{\partial \operatorname{Im}[z_k]}{\partial \epsilon_{nl}} \right); \quad \epsilon_{nl} \in \mathbb{R}$$

Define

$$\frac{\partial J_T(\{z_k\})}{\partial z_k} \equiv \frac{1}{2} \left(\frac{\partial J_T}{\partial \operatorname{Re}[z_k]} - i \frac{\partial J_T}{\partial \operatorname{Im}[z_k]} \right)$$

$$\frac{\partial J_T(\{z_k\})}{\partial z_k^*} \equiv \frac{1}{2} \left(\frac{\partial J_T}{\partial \operatorname{Re}[z_k]} + i \frac{\partial J_T}{\partial \operatorname{Im}[z_k]} \right) = \left(\frac{\partial J_T}{\partial z_k} \right)^*$$

Aside: Wirtinger Derivatives — Derivatives w.r.t. Complex Numbers

$$J_T(\{z_k\}) = J_T(\{\operatorname{Re}[z_k], \operatorname{Im}[z_k]\}); \quad J_T \in \mathbb{R}, \quad z_k \in \mathbb{C}$$

$$\frac{\partial J_T(\{z_k\})}{\partial \epsilon_{nl}} = \sum_k \left(\frac{\partial J_T}{\partial \operatorname{Re}[z_k]} \frac{\partial \operatorname{Re}[z_k]}{\partial \epsilon_{nl}} + \frac{\partial J_T}{\partial \operatorname{Im}[z_k]} \frac{\partial \operatorname{Im}[z_k]}{\partial \epsilon_{nl}} \right); \quad \epsilon_{nl} \in \mathbb{R}$$

Define

$$\frac{\partial J_T(\{z_k\})}{\partial z_k} \equiv \frac{1}{2} \left(\frac{\partial J_T}{\partial \operatorname{Re}[z_k]} - i \frac{\partial J_T}{\partial \operatorname{Im}[z_k]} \right)$$

$$\frac{\partial J_T(\{z_k\})}{\partial z_k^*} \equiv \frac{1}{2} \left(\frac{\partial J_T}{\partial \operatorname{Re}[z_k]} + i \frac{\partial J_T}{\partial \operatorname{Im}[z_k]} \right) = \left(\frac{\partial J_T}{\partial z_k} \right)^*$$

$$\frac{\partial J_T(\{z_k\})}{\partial \epsilon_{nl}} = \sum_k \left(\frac{\partial J_T}{\partial z_k} \frac{\partial z_k}{\partial \epsilon_{nl}} + \frac{\partial J_T}{\partial z_k^*} \frac{\partial z_k^*}{\partial \epsilon_{nl}} \right) = 2\operatorname{Re} \left[\sum_k \frac{\partial J_T}{\partial z_k} \frac{\partial z_k}{\partial \epsilon_{nl}} \right]$$

Aside: Complex Matrix Calculus

$|\Psi\rangle \hat{=} \text{complex column vector} \hat{=} z$

Aside: Complex Matrix Calculus

$|\Psi\rangle \hat{=} \text{complex column vector} \hat{=} z$

$\langle\Psi| \hat{=} \text{conjugate-transpose row vector} \hat{=} z^*$

Aside: Complex Matrix Calculus

$$|\Psi\rangle \hat{=} \text{complex column vector} \hat{=} z$$

$$\langle\Psi| \hat{=} \text{conjugate-transpose row vector} \hat{=} z^*$$

$$\langle\Psi|\phi\rangle \hat{=} \text{scalar}$$

$$\Rightarrow \frac{\partial\langle\Psi|\phi\rangle}{\partial\langle\Psi|} = |\phi\rangle \quad \frac{\partial\langle\Psi|\phi\rangle}{\partial|\phi\rangle} = \langle\Psi|$$

GRAPE Derivation

$$J(\{\epsilon_{ne}\}) = J_T(\{\chi_u(\tau)\}) + \dots$$

$$\nabla J = \frac{\partial J_T}{\partial \epsilon_{ne}}$$

$$\frac{\partial J_T}{\partial \epsilon_{ne}} = 2 \operatorname{Re} \left[\sum_u \underbrace{\frac{\partial J_T}{\partial \langle \chi_u(\tau) |}}_{\langle \chi_u(\tau) |} \cdot \frac{\partial |\chi_u(\tau)\rangle}{\epsilon_{ne}} \right] ; \quad |\chi_u(\tau)\rangle = \frac{\partial J_T}{\partial \langle \chi_u(\tau) |}$$

$$= 2 \operatorname{Re} \left[\sum_u \frac{\partial}{\partial \epsilon_{ne}} \langle \chi_u(\tau) | \chi_u(\tau) \rangle \right]$$

— Goerz *et al.*, Quantum 6, 871 (2022)

GRAPE Derivation

$$\frac{\partial}{\partial \epsilon_{uv}} \langle \chi_u(\tau) | \hat{U}_N \dots \hat{U}_1 | \chi_u(0) \rangle \quad ; \quad \hat{U}_n = e^{-i \hat{H}_n \Delta t}$$

$$= \underbrace{\langle \chi_u(\tau) | \hat{U}_N \dots \hat{U}_{u+1}}_{bv} \frac{\partial \hat{U}_u}{\partial \epsilon_{uv}} \underbrace{\hat{U}_{u-1} \dots \hat{U}_1 | \chi_u(0) \rangle}_{fw}$$

$$| \chi_u(\tau) \rangle = \frac{\partial \mathcal{J}_T}{\partial \langle \chi_u(\tau) |}$$

$$\mathcal{J}_T(\hat{A})$$

$$\hat{A}_{i,u} = \langle q_i | \chi_u \rangle$$

$$\frac{\partial \mathcal{J}_T}{\partial \langle \chi_u(\tau) |} = \sum_i \frac{\partial \mathcal{J}_T}{\partial u_{i,u}} |q_i\rangle$$

— Goerz et al., Quantum 6, 871 (2022)

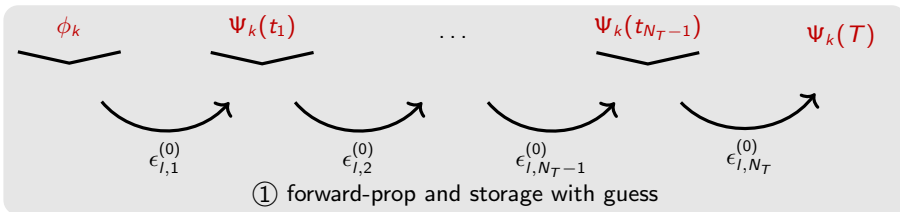
Gradient of Time Evolution Operator

$$\begin{pmatrix} \frac{\partial \hat{U}_n^\dagger}{\partial \epsilon_{n1}} |\chi_k(t_n)\rangle \\ \vdots \\ \frac{\partial \hat{U}_n^\dagger}{\partial \epsilon_{nL}} |\chi_k(t_n)\rangle \\ \hat{U}_n^\dagger |\chi_k(t_n)\rangle \end{pmatrix} = \exp \left[-i \begin{pmatrix} \hat{H}_n^\dagger & 0 & \dots & 0 & \hat{H}_n^{(1)\dagger} \\ 0 & \hat{H}_n^\dagger & \dots & 0 & \hat{H}_n^{(2)\dagger} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & \hat{H}_n^\dagger & \hat{H}_n^{(L)\dagger} \\ 0 & 0 & \dots & 0 & \hat{H}_n^\dagger \end{pmatrix} dt_n \right] \begin{pmatrix} 0 \\ \vdots \\ 0 \\ |\chi_k(t_n)\rangle \end{pmatrix}$$

$$\hat{U}_n = \exp[-i\hat{H}_n dt_n]; \quad \hat{H}_n^{(l)} = \frac{\partial \hat{H}_n}{\partial \epsilon_l(t)}$$

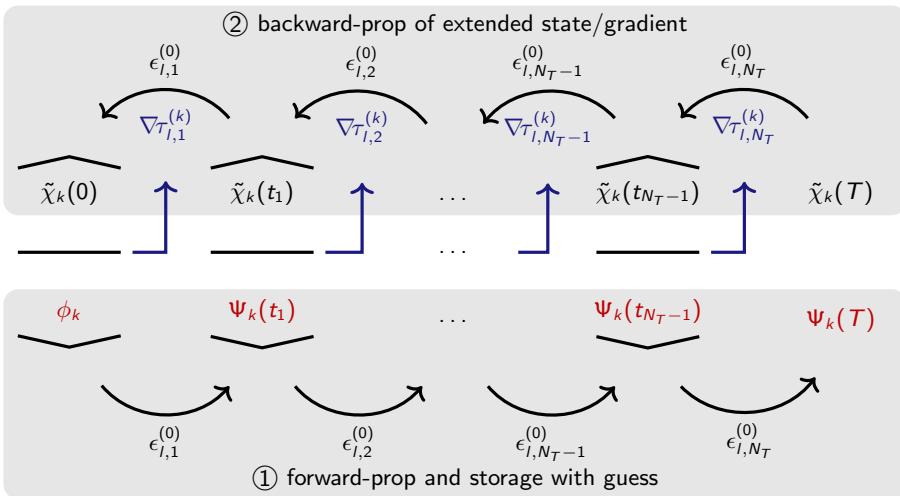
— Goodwin & Kuprov, J. Chem. Phys. 143, 084113 (2015)

Generalized GRAPE Scheme



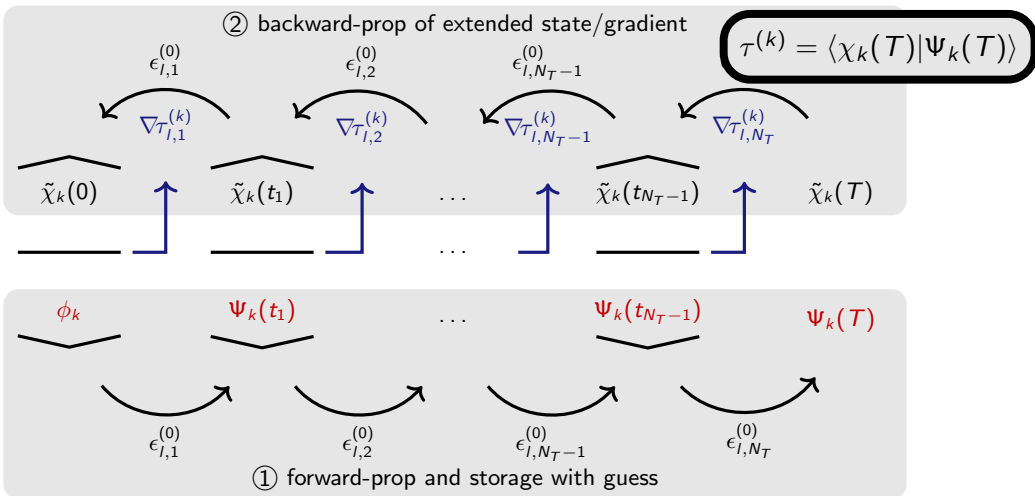
— Goerz *et al.* Quantum 6, 871 (2022)

Generalized GRAPE Scheme



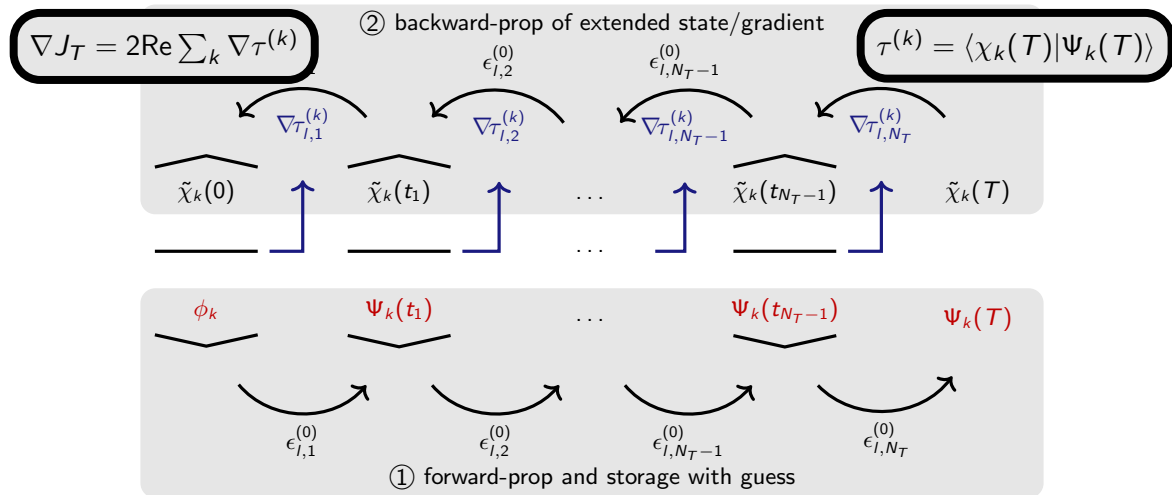
— Goerz *et al.* Quantum 6, 871 (2022)

Generalized GRAPE Scheme

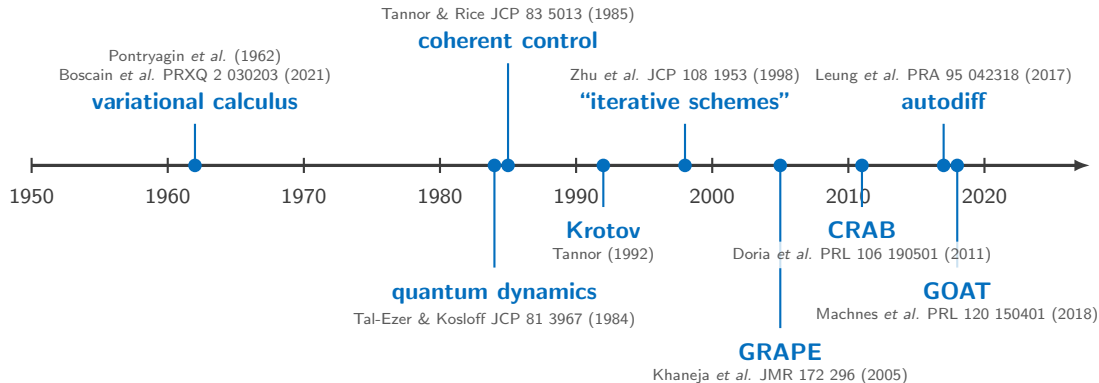


— Goerz *et al.* Quantum 6, 871 (2022)

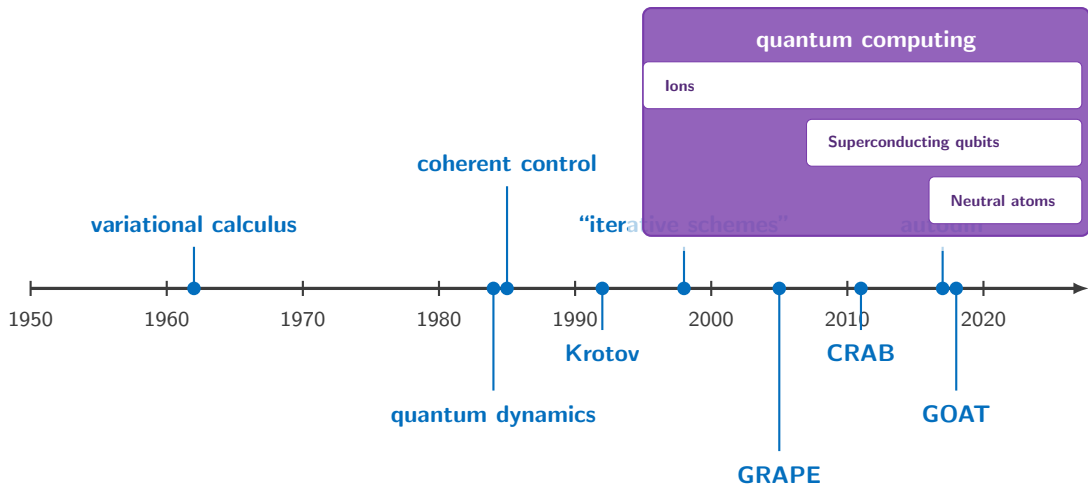
Generalized GRAPE Scheme

— Goerz *et al.* Quantum 6, 871 (2022)

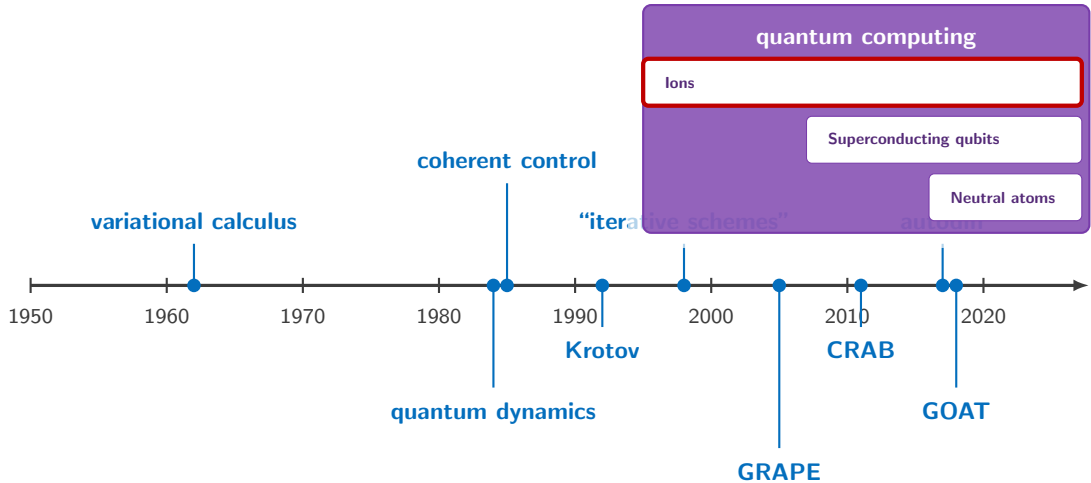
Timeline – Methods



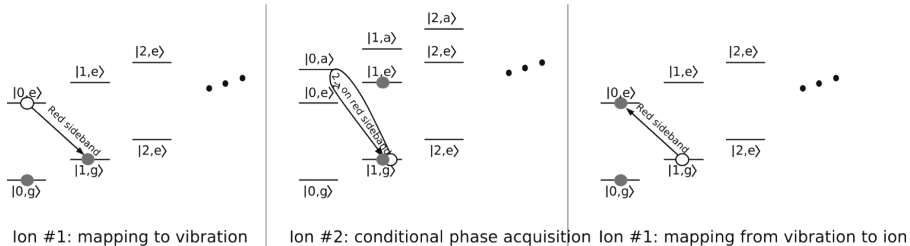
Timeline – Applications: Quantum Computing



Quantum Computing – Trapped Ions



Cirac-Zoller Gate



— Häffner *et al.*, Phys. Rep. 469, 155 (2008)

Idea: map qubit of ion 1 onto the shared (COM) vibrational mode; single-site addressed gate on ion 2 conditioned on the mode $\Rightarrow$ entanglement

— Cirac & Zoller, Phys. Rev. Lett. 74, 4091 (1995)

Mølmer-Sørensen Gate

Direct $|gg\rangle \leftrightarrow |ee\rangle$ coupling via blue/red-sideband interference

— Sørensen & Mølmer, Phys. Rev. Lett. 82, 1971 (1999)

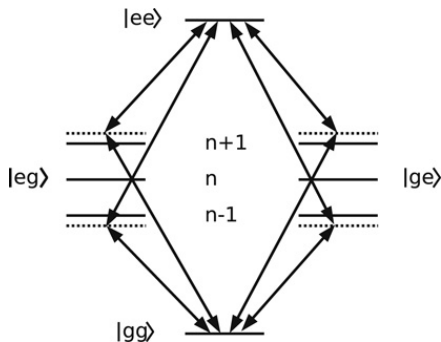
$$|gg\rangle \rightarrow \cos\left(\frac{\tilde{\Omega}T}{2}\right)|gg\rangle + i \sin\left(\frac{\tilde{\Omega}T}{2}\right)|ee\rangle,$$

$$|ee\rangle \rightarrow \cos\left(\frac{\tilde{\Omega}T}{2}\right)|ee\rangle + i \sin\left(\frac{\tilde{\Omega}T}{2}\right)|gg\rangle,$$

$$|ge\rangle \rightarrow \cos\left(\frac{\tilde{\Omega}T}{2}\right)|ge\rangle - i \sin\left(\frac{\tilde{\Omega}T}{2}\right)|eg\rangle,$$

$$|eg\rangle \rightarrow \cos\left(\frac{\tilde{\Omega}T}{2}\right)|eg\rangle - i \sin\left(\frac{\tilde{\Omega}T}{2}\right)|ge\rangle.$$

effective coupling $\tilde{\Omega}$ from 2nd-order perturbation theory — independent of phonon number n

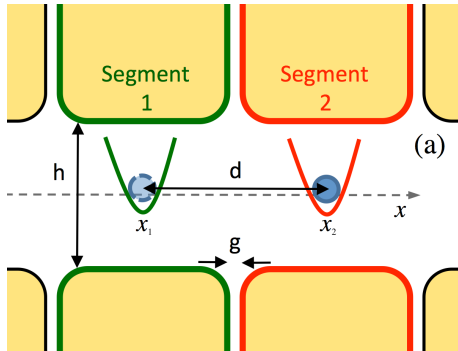


— Häffner *et al.*, Phys. Rep. 469, 155 (2008)

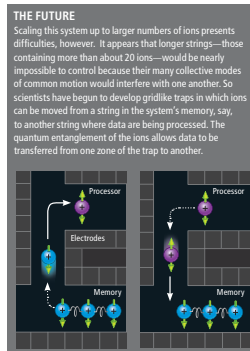
Ion Transport

“Shuffling atoms around” (QCCD)

move ions between trap zones fast, without heating the motional modes

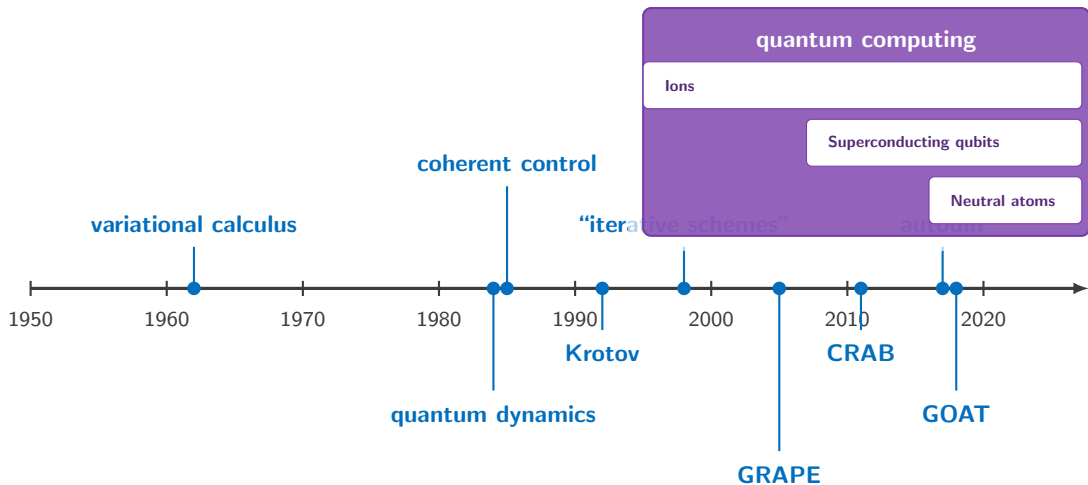


— Fürst *et al.*, New J. Phys. 16, 075007 (2014)

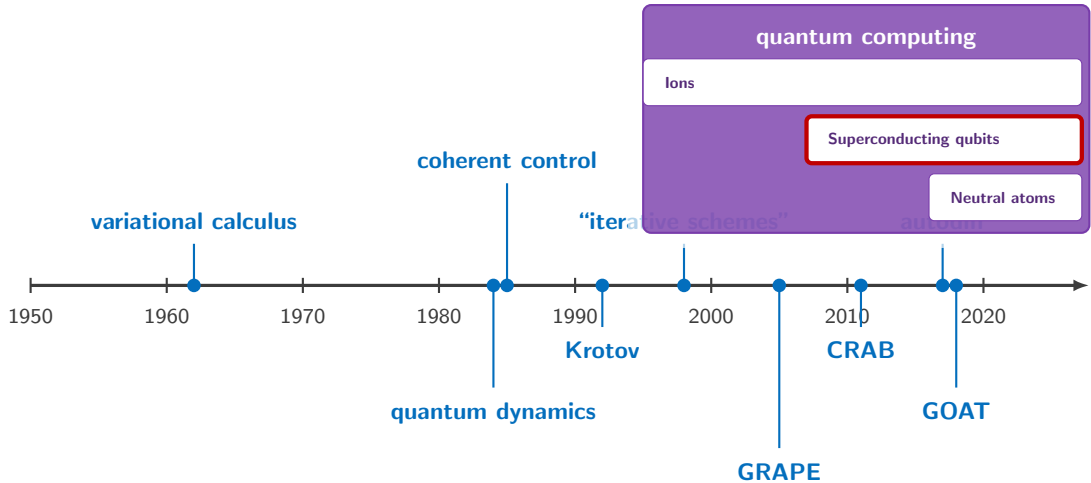


— Monroe & Wineland, Sci. Am. (2008)

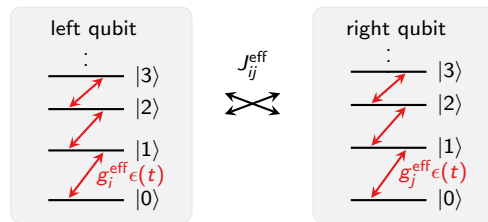
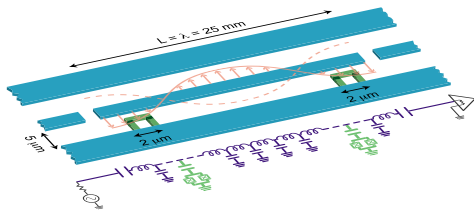
Timeline – Applications: Quantum Computing



Quantum Computing – Superconducting Qubits

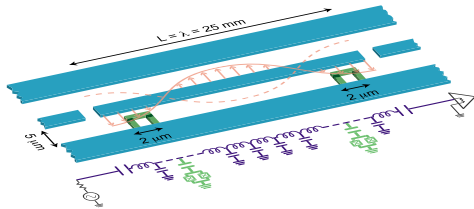


Two Transmon Qubits with Shared Transmission Line



Blais *et al.*, Phys. Rev. A 75, 032329 (2007)

Two Transmon Qubits with Shared Transmission Line

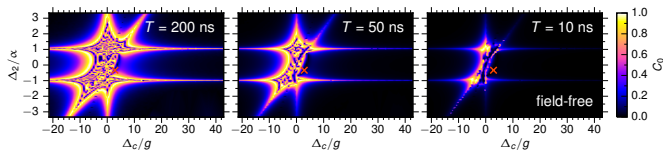


Blais *et al.*, Phys. Rev. A 75, 032329 (2007)

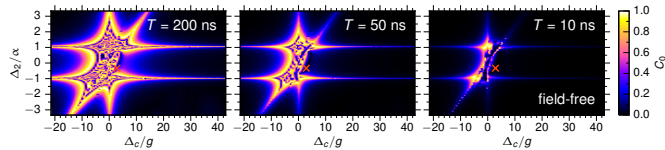
QOCT for superconducting qubits

- DRAG: Motzoi *et al.*, Phys. Rev. Lett. 103, 110501 (2009)
- Perfect entanglers: Goerz *et al.*, Phys. Rev. A 91, 062307 (2015)
- Design landscape: Goerz *et al.*, npj Quantum Inf. 3, 37 (2017)
- Qiskit Pulse: Alexander *et al.*, Quantum Sci. Technol. 5, 044006 (2020)
- Cloud OC: Carvalho *et al.*, Phys. Rev. Applied 15, 064054 (2021)
- Cross-resonance: Li *et al.*, npj Quantum Inf. 10, 66 (2024)

Mapping the Transmon Landscape with QOCT

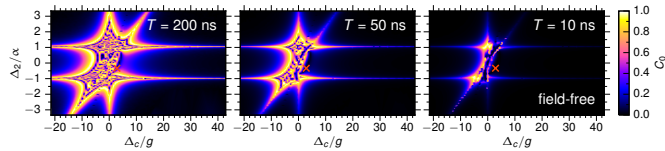


Mapping the Transmon Landscape with QOCT



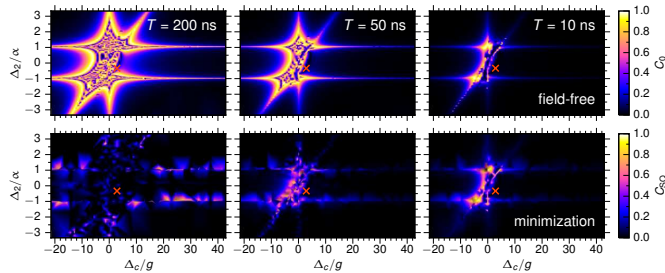
universal quantum computing: perfect entangler *and* local gates

Mapping the Transmon Landscape with QOCT

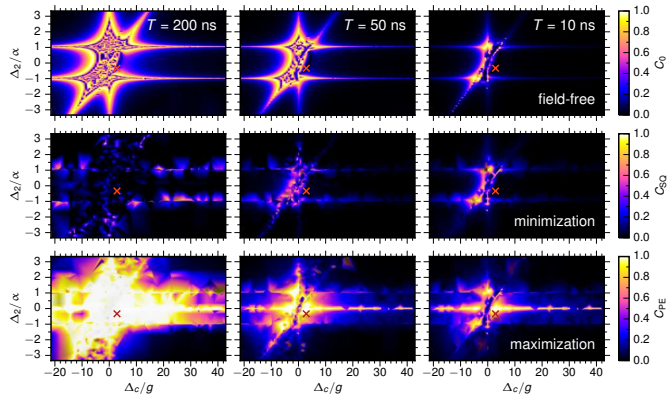


optimize for maximum / minimum entanglement

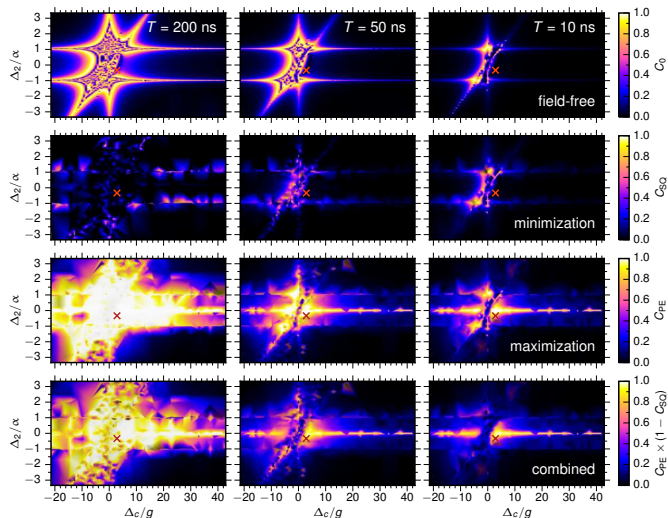
Mapping the Transmon Landscape with QOCT



Mapping the Transmon Landscape with QOCT

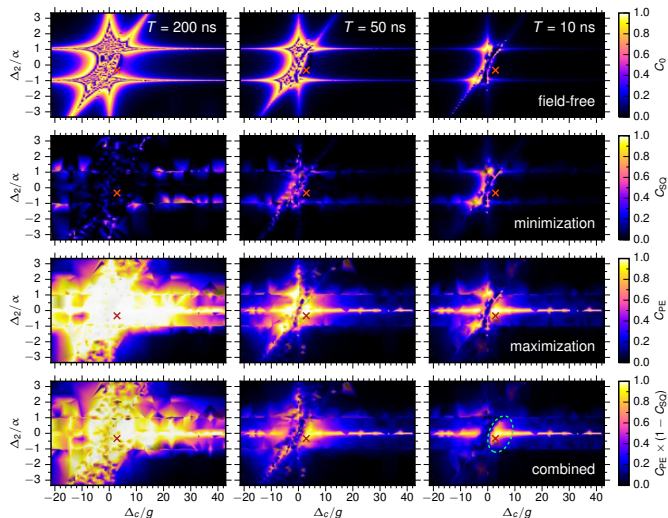


Mapping the Transmon Landscape with QOCT



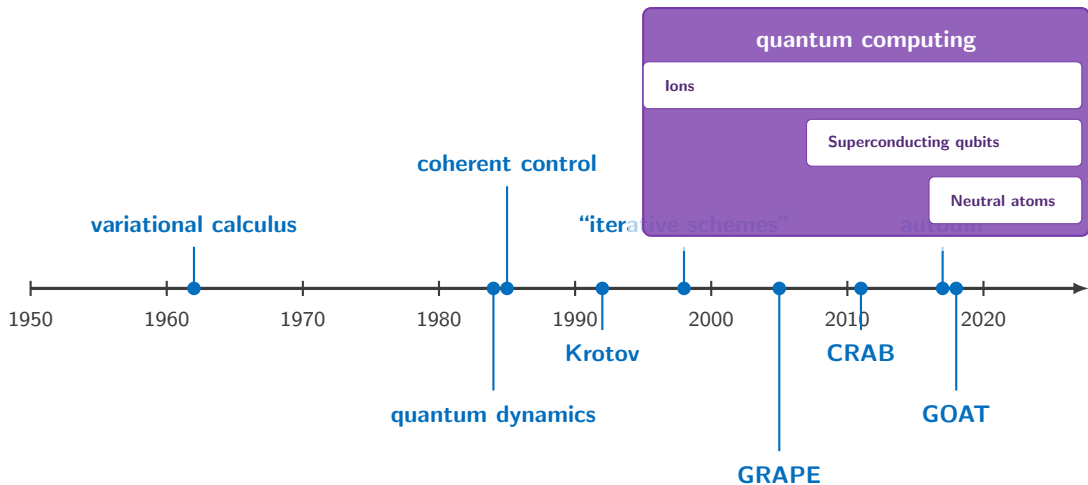
— Goerz *et al.*, npj Quantum Inf. 3, 37 (2017)

Mapping the Transmon Landscape with QOCT

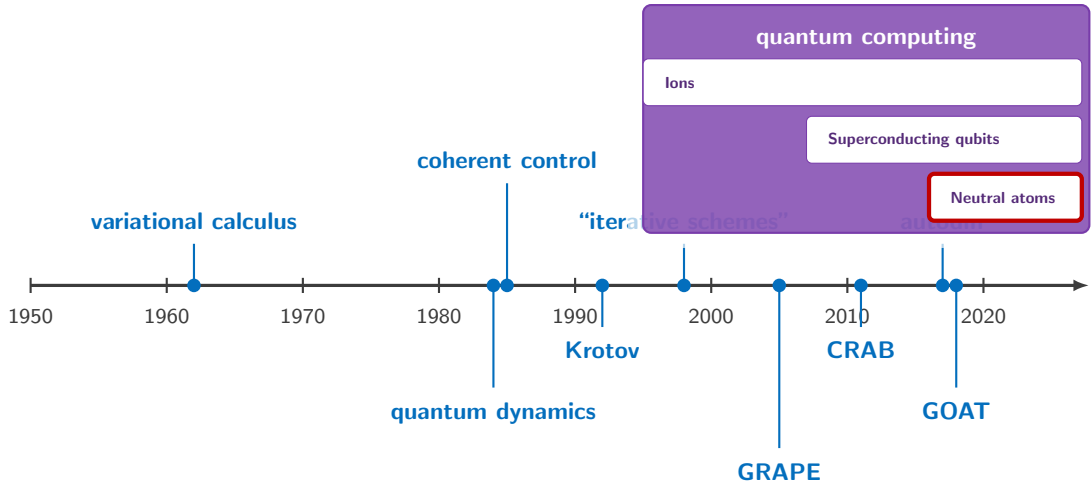


— Goerz *et al.*, npj Quantum Inf. 3, 37 (2017)

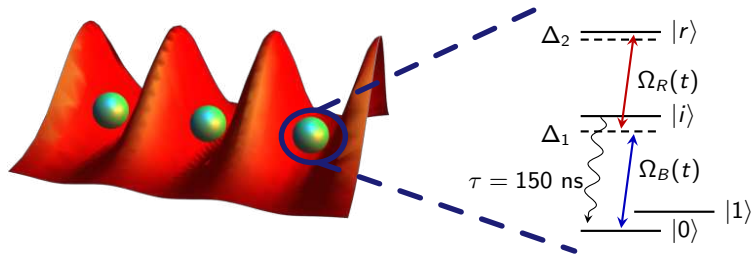
Timeline – Applications: Quantum Computing



Quantum Computing – Neutral Atoms

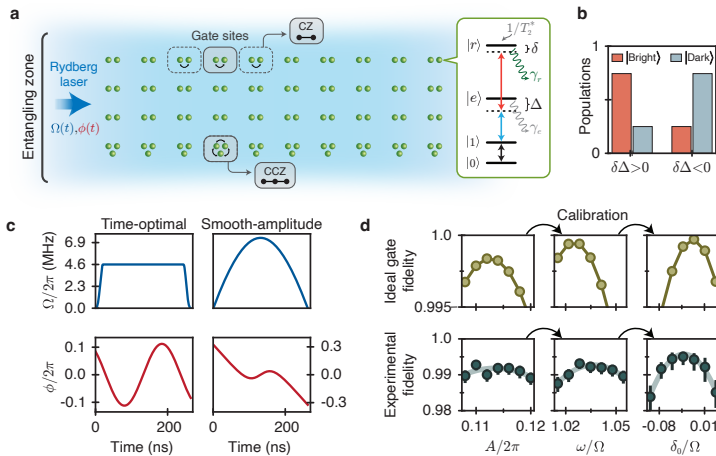


Rydberg Atoms



— Goerz *et al.*, Phys. Rev. A 90, 032329 (2014)

Neutral-Atom Rydberg Gates



— Evered *et al.*, Nature 622, 268 (2023)

Optimal Control in Industry

Two paradigms: open-loop, model-based vs. closed-loop, model-free

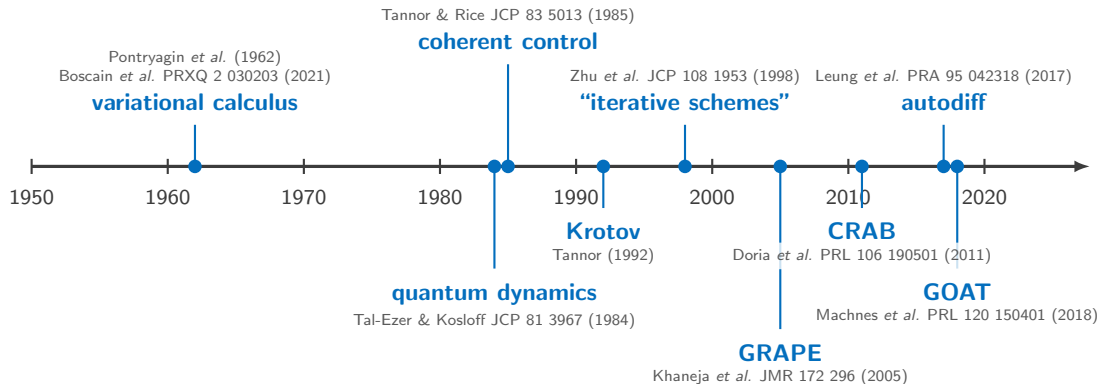
Open loop (model-based)

- Q-CTRL Boulder Opal
Carvalho *et al.*, Phys. Rev. Applied 15, 064054 (2021)
- IonQ power-optimal MS gate
Blümel *et al.*, npj Quantum Inf. 7, 147 (2021)
- Rydberg / time-optimal pulses
Levine *et al.*, Phys. Rev. Lett. 123, 170503 (2019)
- Multi-derivative DRAG
Li *et al.*, npj Quantum Inf. 10, 66 (2024)

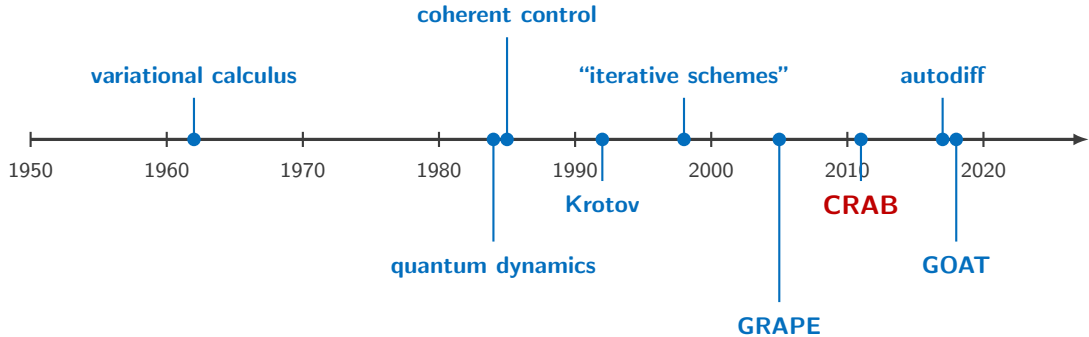
Closed loop (model-free)

- Google frequency allocation (Snake)
Klimov *et al.*, Nat. Commun. 15, 2442 (2024)
- Q-CTRL deep reinforcement learning
Baum *et al.*, PRX Quantum 2, 040324 (2021)
- QuEra analog schedule optimization
Ebadi *et al.*, Science 376, 1209 (2022)

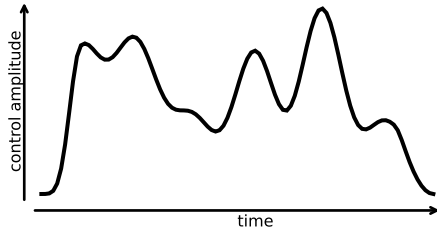
Timeline – Methods



CRAB



CRAB



piecewise-constant pulses
 $\Rightarrow$ parametrized continuous controls

$$\epsilon(t) = \epsilon(\{u_n\}, t)$$

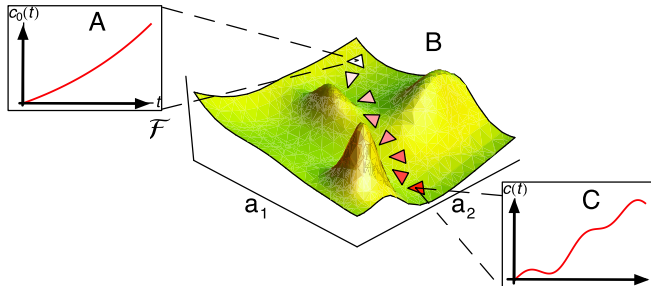
E.g. CRAB — Chopped Random (spectral) Basis

$$\epsilon(t) = \sum_{n=1}^{10} (a_n \cos(\omega_n t) + b_n \sin(\omega_n t))$$

— Caneva *et al.*, Phys. Rev. A 84, 022326 (2011)

CRAB

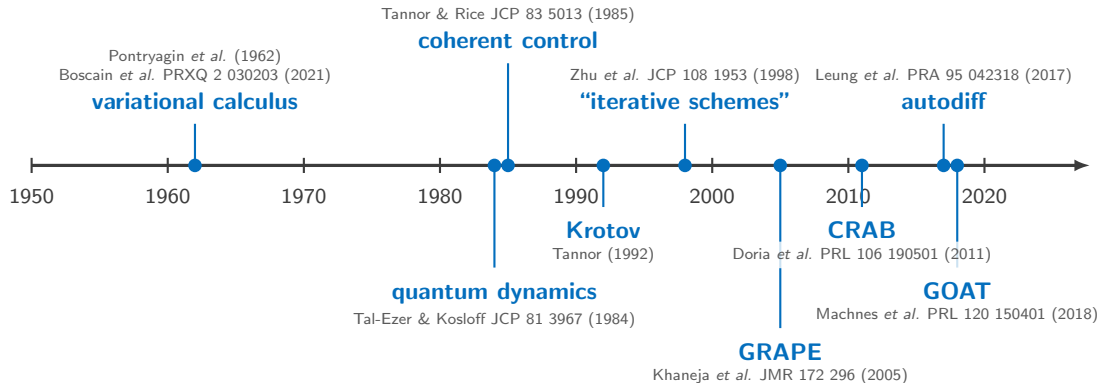
Gradient-free optimization



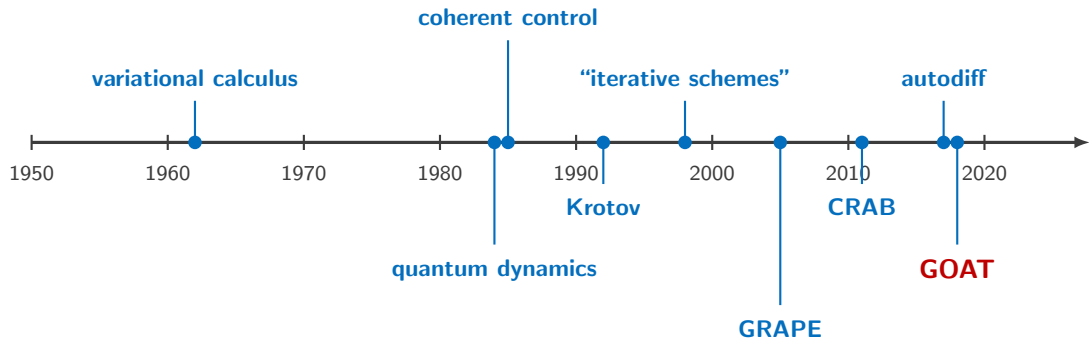
e.g. Nelder-Mead (simplex), genetic algorithms...

— Doria *et al.*, Phys. Rev. Lett. 106, 190501 (2011)

Timeline – Methods



GOAT



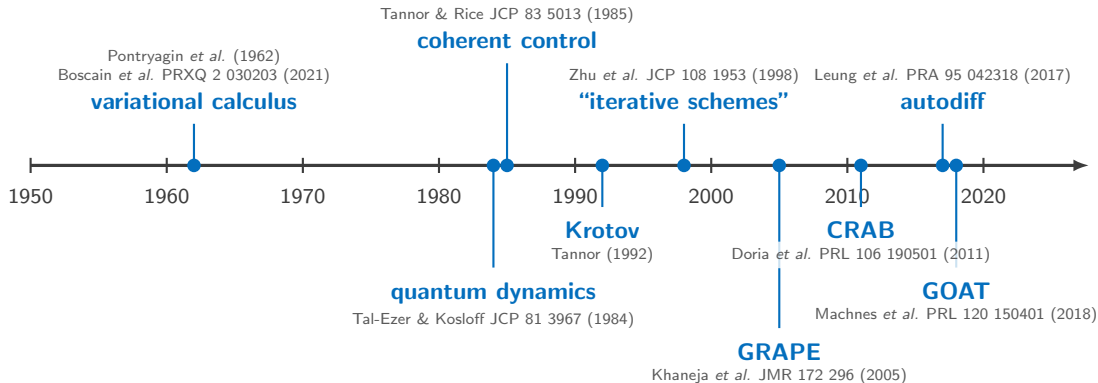
GOAT

$$\begin{pmatrix} \frac{\partial \hat{U}}{\partial u_1} |\psi_k\rangle \\ \vdots \\ \frac{\partial \hat{U}}{\partial u_N} |\psi_k\rangle \\ \hat{U} |\psi_k\rangle \end{pmatrix} = \exp \left[-i\mathcal{T} \int_0^T \begin{pmatrix} \hat{H}(t) & 0 & \dots & 0 & \hat{H}^{(1)}(t) \\ 0 & \hat{H}(t) & \dots & 0 & \hat{H}^{(2)}(t) \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & \hat{H}(t) & \hat{H}^{(N)}(t) \\ 0 & 0 & \dots & 0 & \hat{H}(t) \end{pmatrix} dt \right] \begin{pmatrix} 0 \\ \vdots \\ 0 \\ |\psi_k\rangle \end{pmatrix}$$

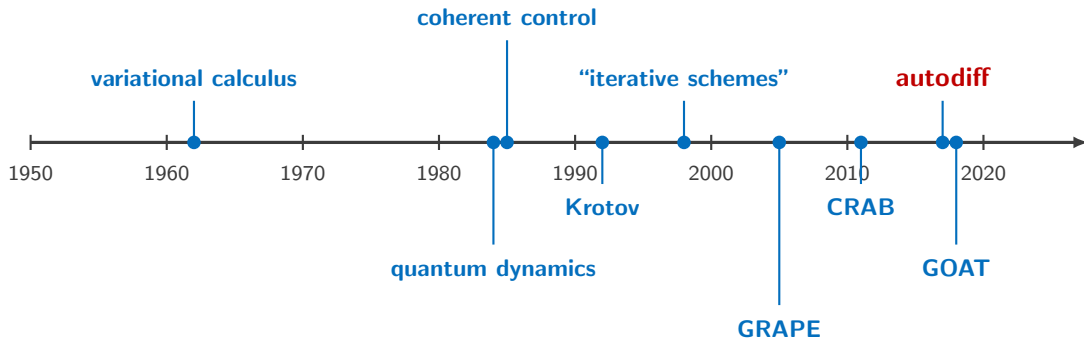
$$\text{with } \hat{H}^{(n)}(t) = \frac{\partial \hat{H}(t)}{\partial u_n}$$

— “GOAT”: Machnes *et al.*, Phys. Rev. Lett. 120, 150401 (2018)

Timeline – Methods



Automatic Differentiation



Automatic Differentiation (AD)

- Build computational graph for time propagation
 - Elementary operations have known derivatives
 - Let computer apply chain rule at each node
 - Backward pass to accumulate gradient
-
- Leung *et al.*, Phys. Rev. A 95, 042318 (2017)
 - Abdelhafez *et al.*, Phys. Rev. A 99, 052327 (2019)
 - Schäfer *et al.*, Mach. Learn.: Sci. Technol. 1, 035009 (2020)
 - Abdelhafez *et al.*, Phys. Rev. A 101, 022321 (2020)

Automatic Differentiation (AD)

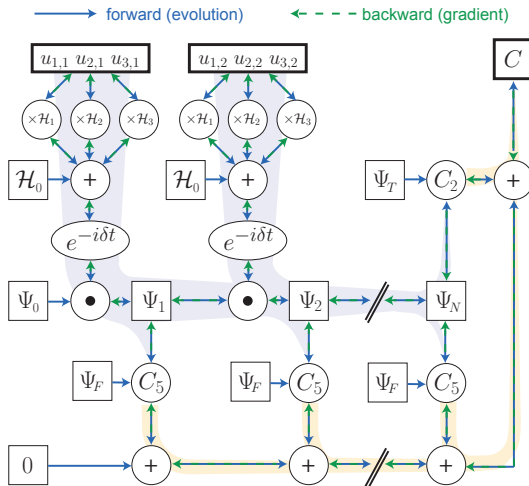


Fig. 2 in Leung *et al.*, Phys. Rev. A 95, 042318 (2017)

AD Advantages

■ Arbitrary functionals

μ	Cost-function contribution	$C_\mu(\mathbf{u})$
1	Target-gate infidelity	$1 - \text{tr}(K_T^\dagger K_N)/D ^2$
2	Target-state infidelity	$1 - \langle \Psi_T \Psi_N \rangle ^2$
3	Control amplitudes	$ \mathbf{u} ^2$
4	Control variations	$\sum_{j,k} u_{k,j} - u_{k,j-1} ^2$
5	Occupation of forbidden state	$\sum_j \langle \Psi_F \Psi_j \rangle ^2$
6	Evolution time (target gate)	$1 - \frac{1}{N} \sum_j \text{tr}(K_T^\dagger K_j)/D ^2$
7	Evolution time (target state)	$1 - \frac{1}{N} \sum_j \langle \Psi_T \Psi_j \rangle ^2$

Table 1 in Leung *et al.*, Phys. Rev. A 95, 042318 (2017)

AD Advantages

■ Arbitrary functionals

μ	Cost-function contribution	$C_\mu(\mathbf{u})$
1	Target-gate infidelity	$1 - \text{tr}(K_T^\dagger K_N)/D ^2$
2	Target-state infidelity	$1 - \langle \Psi_T \Psi_N \rangle ^2$
3	Control amplitudes	$ \mathbf{u} ^2$
4	Control variations	$\sum_{j,k} u_{k,j} - u_{k,j-1} ^2$
5	Occupation of forbidden state	$\sum_j \langle \Psi_F \Psi_j \rangle ^2$
6	Evolution time (target gate)	$1 - \frac{1}{N} \sum_j \text{tr}(K_T^\dagger K_j)/D ^2$
7	Evolution time (target state)	$1 - \frac{1}{N} \sum_j \langle \Psi_T \Psi_j \rangle ^2$

Table 1 in Leung *et al.*, Phys. Rev. A 95, 042318 (2017)

■ Arbitrary equations of motion

e.g. quantum trajectories — Abdelhafez *et al.*, Phys. Rev. A 99, 052327 (2019)

AD Advantages

■ Arbitrary functionals

μ	Cost-function contribution	$C_\mu(\mathbf{u})$
1	Target-gate infidelity	$1 - \text{tr}(K_T^\dagger K_N)/D ^2$
2	Target-state infidelity	$1 - \langle \Psi_T \Psi_N \rangle ^2$
3	Control amplitudes	$ \mathbf{u} ^2$
4	Control variations	$\sum_{j,k} u_{k,j} - u_{k,j-1} ^2$
5	Occupation of forbidden state	$\sum_j \langle \Psi_F \Psi_j \rangle ^2$
6	Evolution time (target gate)	$1 - \frac{1}{N} \sum_j \text{tr}(K_T^\dagger K_j)/D ^2$
7	Evolution time (target state)	$1 - \frac{1}{N} \sum_j \langle \Psi_T \Psi_j \rangle ^2$

Table 1 in Leung *et al.*, Phys. Rev. A 95, 042318 (2017)

■ Arbitrary equations of motion

e.g. quantum trajectories — Abdelhafez *et al.*, Phys. Rev. A 99, 052327 (2019)

■ GPU support

Entanglement Measures

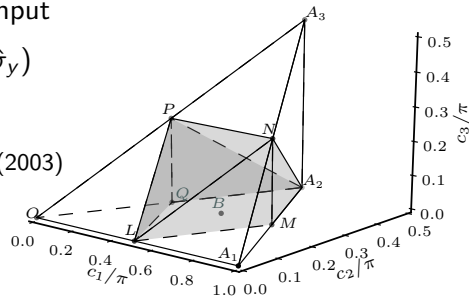
Quantum Gate Concurrence

Max concurrence that can be generated for a separable input

1 $c_1, c_2, c_3 \propto \text{eigvals}(\hat{U}\tilde{U}); \quad \tilde{U} = (\hat{\sigma}_y \otimes \hat{\sigma}_y)\hat{U}(\hat{\sigma}_y \otimes \hat{\sigma}_y)$

2 $C(\hat{U}) = \max |\sin(c_{1,2,3} \pm c_{3,1,2})|$

Childs *et al.*, Phys. Rev. A 68, 052311 (2003)



Entanglement Measures

Quantum Gate Concurrence

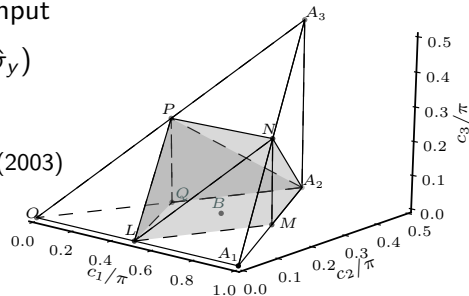
Max concurrence that can be generated for a separable input

1 $c_1, c_2, c_3 \propto \text{eigvals}(\hat{U}\tilde{U}); \quad \tilde{U} = (\hat{\sigma}_y \otimes \hat{\sigma}_y)\hat{U}(\hat{\sigma}_y \otimes \hat{\sigma}_y)$

2 $C(\hat{U}) = \max |\sin(c_{1,2,3} \pm c_{3,1,2})|$

Childs *et al.*, Phys. Rev. A 68, 052311 (2003)

Not analytic!



Entanglement Measures

Quantum Gate Concurrence

Max concurrence that can be generated for a separable input

$$1 \quad c_1, c_2, c_3 \propto \text{eigvals}(\hat{U}\tilde{U}); \quad \tilde{U} = (\hat{\sigma}_y \otimes \hat{\sigma}_y)\hat{U}(\hat{\sigma}_y \otimes \hat{\sigma}_y)$$

$$2 \quad C(\hat{U}) = \max |\sin(c_{1,2,3} \pm c_{3,1,2})|$$

Childs *et al.*, Phys. Rev. A 68, 052311 (2003)

Perfect Entanglers Functional

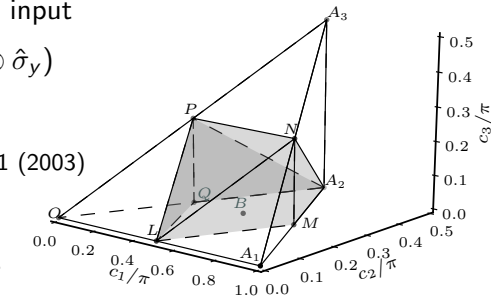
Find a two-qubit gate with maximum entangling power

$$F_{PE} = \left(\frac{1}{\det U_B} \right) \left(\frac{1}{4} (\text{tr}^2[U_B^T U_B] - \text{tr}[U_B^T U_B U_B^T U_B]) \right) \left(\frac{1}{16} \text{Re}^2[\text{tr}[U_B^T U_B]] \right) +$$

$$+ \left(\frac{2}{\det U_B} \right) \left(\frac{1}{4} (\text{tr}^2[U_B^T U_B] - \text{tr}[U_B^T U_B U_B^T U_B]) \right) \left(\frac{1}{16} \text{Im}^2[\text{tr}[U_B^T U_B]] \right)$$

$$\left(\frac{1}{16} \text{Re}[\text{tr}^2[U_B^T U_B]] \right)$$

U_B : projection into logical subspace, in Bell basis



Watts *et al.*, Phys. Rev. A 91, 062306 (2015)

Goerz *et al.*, Phys. Rev. A 91, 062307 (2015)

Entanglement Measures

Quantum Gate Concurrence

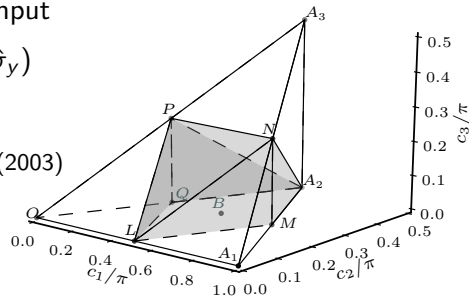
Max concurrence that can be generated for a separable input

$$1 \quad c_1, c_2, c_3 \propto \text{eigvals}(\hat{U}\tilde{U}); \quad \tilde{U} = (\hat{\sigma}_y \otimes \hat{\sigma}_y)\hat{U}(\hat{\sigma}_y \otimes \hat{\sigma}_y)$$

$$2 \quad C(\hat{U}) = \max |\sin(c_{1,2,3} \pm c_{3,1,2})|$$

Childs *et al.*, Phys. Rev. A 68, 052311 (2003)

To a computer, everything is analytic!



Entanglement Measures

Quantum Gate Concurrence

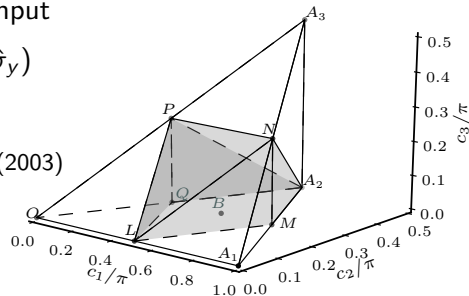
Max concurrence that can be generated for a separable input

$$1 \quad c_1, c_2, c_3 \propto \text{eigvals}(\hat{U}\tilde{U}); \quad \tilde{U} = (\hat{\sigma}_y \otimes \hat{\sigma}_y)\hat{U}(\hat{\sigma}_y \otimes \hat{\sigma}_y)$$

$$2 \quad C(\hat{U}) = \max |\sin(c_{1,2,3} \pm c_{3,1,2})|$$

Childs *et al.*, Phys. Rev. A 68, 052311 (2003)

To a computer, everything is analytic!



Quantum Fisher Information

$$F(\hat{\rho}) = \sum_{i \neq j} \frac{2(p_i - p_j)^2}{p_i + p_j} \left| \langle \phi_i | \hat{S}_z | \phi_j \rangle \right|^2$$

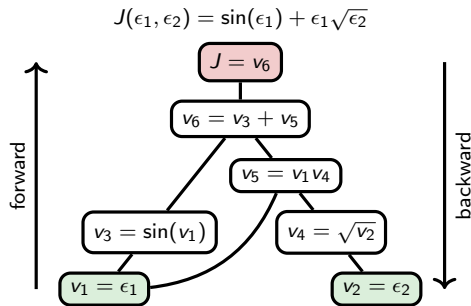
where p_i , $|\phi_i\rangle$ are eigenvalues / eigenstates of $\hat{\rho}$

— Ma *et al.*, Phys. Rep. 509, 89 (2011)

AD Compromises

1 Numerical scaling

- AD memory overhead
- computational overhead (at least on CPU)



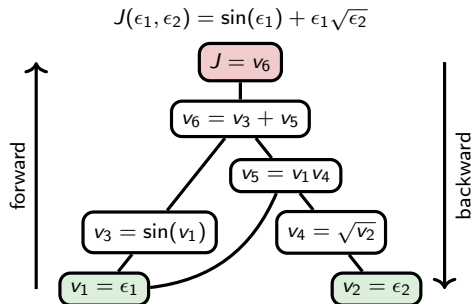
AD Compromises

1 Numerical scaling

- AD memory overhead
- computational overhead (at least on CPU)

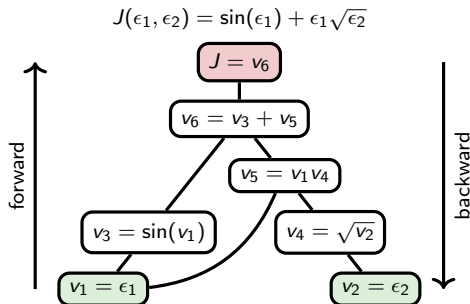
2 Framework limitations

- Complex numbers?
- In-place operations?
- Double-precision?



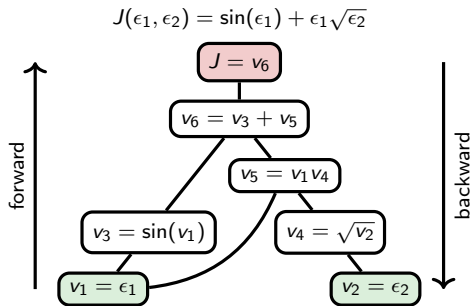
AD Compromises

- 1 Numerical scaling
 - AD memory overhead
 - computational overhead (at least on CPU)
- 2 Framework limitations
 - Complex numbers?
 - In-place operations?
 - Double-precision?
- 3 Code reuse
 - Re-implement propagation methods?
 - Re-use existing GRAPE implementation?



AD Compromises

- 1 Numerical scaling
 - AD memory overhead
 - computational overhead (at least on CPU)
- 2 Framework limitations
 - Complex numbers?
 - In-place operations?
 - Double-precision?
- 3 Code reuse
 - Re-implement propagation methods?
 - Re-use existing GRAPE implementation?



We don't have to compromise!

Semi-Automatic Differentiation

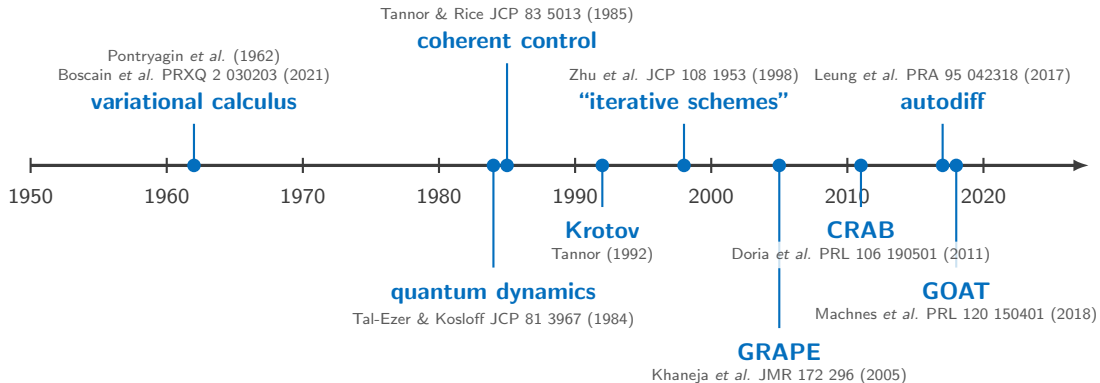
Best of both: GRAPE efficiency + AD flexibility

$$\nabla J = \frac{\partial J_T}{\partial \epsilon_{nl}} = 2\text{Re} \sum_k \frac{\partial}{\partial \epsilon_{nl}} \langle \chi_k(T) | \Psi_k(T) \rangle; \quad |\chi_k(T)\rangle = \frac{\partial J_T}{\partial \langle \Psi_k(T) |}$$

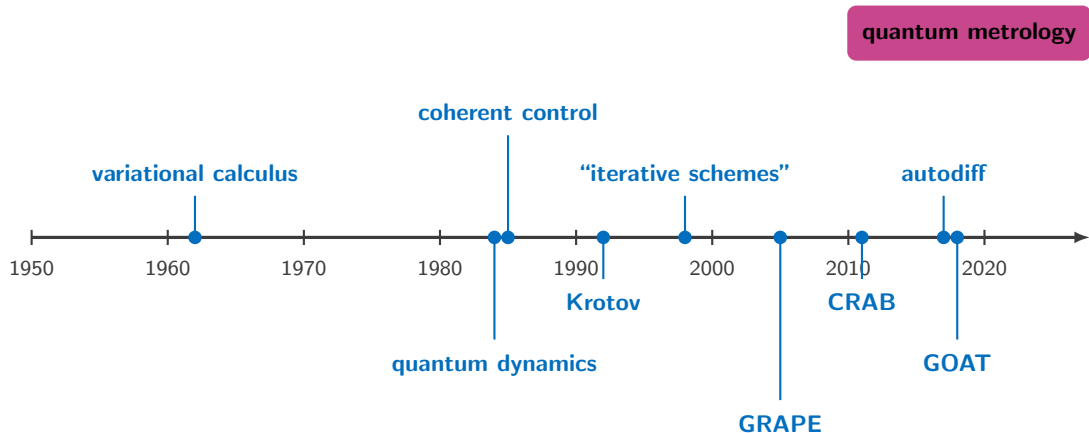
- gradient of the propagation, $\partial |\Psi_k(T)\rangle / \partial \epsilon_{nl}$ — analytic GRAPE sweep (efficient)
- gradient of the functional, $\partial J_T / \partial \langle \Psi_k(T) |$ — automatic differentiation (arbitrary functionals)

— Goerz *et al.*, Quantum 6, 871 (2022)

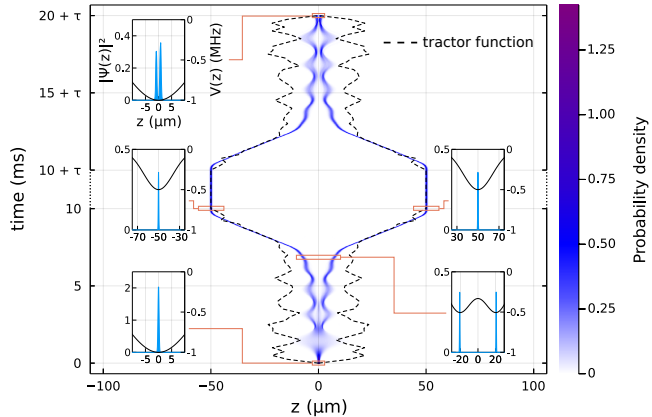
Timeline – Methods



Timeline – Applications: Quantum Metrology



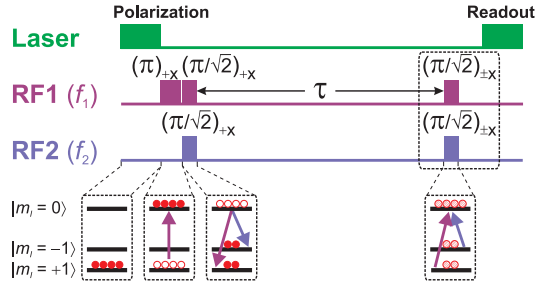
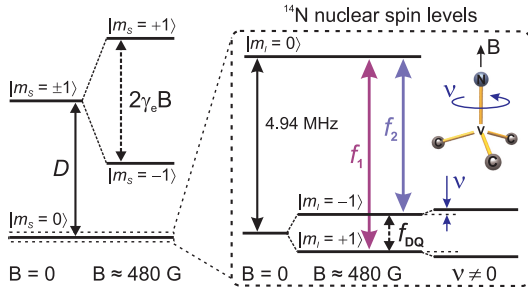
Tractor Atom Interferometry



Find non-adiabatic tractor potential closing the interferometric path

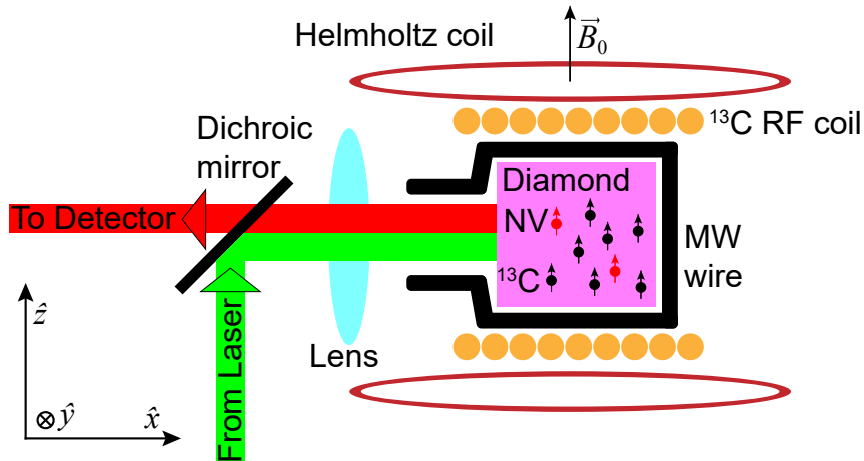
— Raithel *et al.*, Quantum Sci. Technol. 8, 014001 (2023)

Nuclear Spin Gyroscope



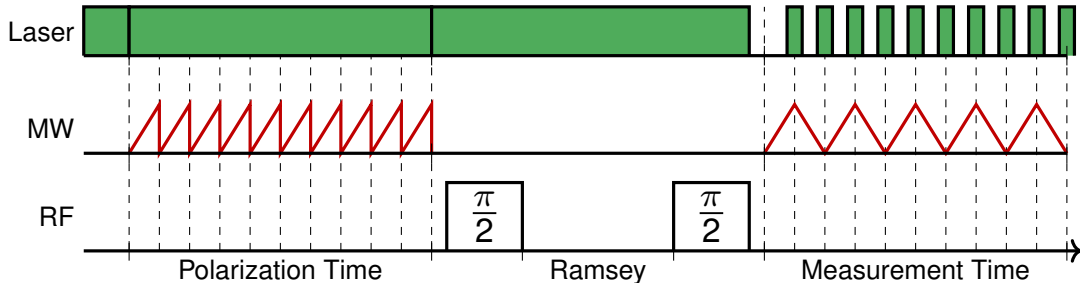
— Adapted from Jarmola *et al.*, Sci. Adv. 7, eabl3840 (2021)

Ramsey Metrology of ^{13}C in Diamond



— Aiello *et al.*, arXiv:2605.00152 (2026)

Ramsey Metrology of ^{13}C in Diamond



Gordon Research Conference



Quantum Control of Light and Matter

Gordon Research Conference

August 1 - 6, 2027

[Apply Now](#)

GRC Education Requirements: Undergraduates or those who have not obtained a bachelor's degree in science/engineering (or acceptable equivalent) are not eligible to apply to attend Gordon Research Conferences or Seminars.

Chairs

Rosario Gonzalez-Ferez and
Patrick Nuernberger

Vice Chairs

Ignacio R. Sola and Lindsay J. LeBlanc

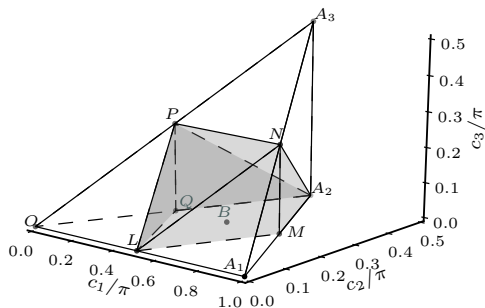
Salve Regina University

36 Ochre Point Avenue
Newport, Rhode Island, United
States

Backup Slides

Geometry of Two-Qubit Gates: The Weyl Chamber

- Local invariants: gates equal up to single-qubit operations
- Each class $\rightarrow$ one point in the Weyl chamber
- Half of all two-qubit gates are perfect entanglers
- Target the *class*, not a specific unitary



Zhang *et al.*, Phys. Rev. A 67, 042313 (2003)

Goerz *et al.*, Phys. Rev. A 91, 062307 (2015)

— Watts *et al.*, Phys. Rev. A 91, 062306 (2015)

QuantumControl.jl

github.com



JuliaQuantumControl

Julia Framework for Quantum Optimal Control

 57 followers  <https://juliaquantumcontrol.github.io...>

 Overview  Repositories 20  Discussions  Projects  Packages  People 4

README.md

A Julia Framework for Quantum Optimal Control.

[docs](#) [stable](#) [docs](#) [dev](#)

The [JuliaQuantumControl](#) organization collects packages implementing a comprehensive collection of methods of open-loop quantum optimal control.

[Quantum optimal control theory](#) attempts to steer a quantum system in some desired way by finding optimal control parameters or control fields inside the system Hamiltonian or Liouvillian. Typical control tasks are the preparation of a specific quantum state or the realization of a logical gate in a quantum computer. Thus, quantum control theory is a critical part of realizing quantum technologies, at the lowest level. Numerical methods of *open-loop* quantum control (methods that do not involve measurement feedback from a physical quantum device) such as [Krotov's method](#) and [GRAPE](#) address the control problem by [simulating the dynamics of the system](#) and then iteratively improving the value of a functional that encodes the desired outcome.

People



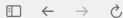
Top languages

 Julia  TeX  Jupyter Notebook

Most used topics

[julia](#) [quantum](#) [grape](#) [optimal-control](#) [quantum-computing](#)

QuantumControl.jl



github.com



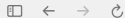
Packages

Package	Version	CI Status	Coverage	Description
★ QuantumPropagators.jl	Jun 2026 v0.9.0	CI passing	codecov 91%	Simulate the time evolution of quantum systems (docs)
★ QuantumControl.jl	Jun 2026 v0.11.4	CI passing	codecov 88%	Framework for Quantum Dynamics and Control (docs)
QuantumGradientGenerators.jl	Jun 2026 v0.1.9	CI passing	codecov 88%	Dynamic Gradients for Quantum Control (docs)
Krotov.jl	Sep 2024 v0.7.2	CI passing	codecov 85%	Krotov's method of optimal control (docs)
GRAPE.jl	Jun 2026 v1.1.0	CI passing	codecov 88%	Gradient Ascent Pulse Engineering method (docs)
TwoQubitWeylChamber.jl	Sep 2024 v0.2.1	CI passing	codecov 97%	Optimizing two-qubit gates in the Weyl chamber (docs)
QuantumControlTestUtils.jl	Jun 2026 v0.3.2	CI passing		Tools for testing and benchmarking (docs)

Documentation

Read the [QuantumControl.jl documentation](#) for a comprehensive description of the framework.

Control Problem



juliaquantumcontrol.github.io



✓ QuantumControlBase.ControlProblem — Type

A full control problem with multiple trajectories.

```
ControlProblem(  
    trajectories,  
    tlist;  
    kwargs...  
)
```



The `trajectories` are a list of `Trajectory` instances, each defining an initial state and a dynamical generator for the evolution of that state. Usually, the trajectory will also include a target state (see `Trajectory`) and possibly a weight. The `trajectories` may also be given together with `tlist` as a mandatory keyword argument.

The `tlist` is the time grid on which the time evolution of the initial states of each trajectory should be propagated. It may also be given as a (mandatory) keyword argument.

The remaining `kwargs` are keyword arguments that are passed directly to the optimal control method. These typically include e.g. the optimization functional.

Dynamical Generator


juliaquantumcontrol.github.io


Glossary



Generator — Dynamical generator (Hamiltonian / Liouvillian) for the time evolution of a state, i.e., the right-hand-side of the equation of motion (up to a factor of i) such that $|\Psi(t + dt)\rangle = e^{-i\hat{H}dt}|\Psi(t)\rangle$ in the infinitesimal limit. We use the symbols G , $\hat{H}$, or L , depending on the context (general, Hamiltonian, Liouvillian). Examples for supported forms a Hamiltonian are the following, from the most general case to simplest and most common case of linear controls,

$$\hat{H} = \overbrace{\hat{H}_0}^{\text{drift term}} + \sum_l \overbrace{\hat{H}_l(\{\epsilon_l(t)\}, t)}^{\text{control term}} \quad (\text{G1})$$

$$\hat{H} = \hat{H}_0 + \sum_l \overbrace{a_l(\{\epsilon_l(t)\}, t)}^{\text{control amplitude}} \underbrace{\hat{H}_l}_{\text{control function}} \quad (\text{G2})$$

$$\hat{H} = \hat{H}_0 + \sum_l \underbrace{\epsilon_l(t)}_{\text{control operator}} \hat{H}_l \quad (\text{G3})$$

Dynamical Generator



juliaquantumcontrol.github.io



Glossary



Generator — Dynamical generator (Hamiltonian / Liouvillian) for the time evolution of a state, i.e., the right-hand-side of the equation of motion (up to a factor of i) such that $|\Psi(t + dt)\rangle = e^{-i\hat{H}dt}|\Psi(t)\rangle$ in the infinitesimal limit. We use the symbols G , $\hat{H}$, or L , depending on the context (general, Hamiltonian, Liouvillian). Examples for supported forms a Hamiltonian are the following, from the most general case to simplest and most common case of linear controls,

drift term
control term

```
return hamiltonian( $\hat{H}_0$ , ( $\hat{H}_{1re}$ ,  $\Omega_{re}$ ), ( $\hat{H}_{1im}$ ,  $\Omega_{im}$ ))
```

$$\hat{H} = \hat{H}_0 + \sum_l \underbrace{a_l(\{\epsilon_l(t)\}, t)}_{\text{control function}} \hat{H}_l \quad (G2)$$

$$\hat{H} = \hat{H}_0 + \sum_l \underbrace{\epsilon_l(t)}_{\text{control operator}} \hat{H}_l \quad (G3)$$

Dynamical Generator



juliaquantumcontrol.github.io



Glossary



Generator — Dynamical generator (Hamiltonian / Liouvillian) for the time evolution of a state, i.e., the right-hand-side of the equation of motion (up to a factor of i) such that $|\Psi(t + dt)\rangle = e^{-i\hat{H}dt}|\Psi(t)\rangle$ in the infinitesimal limit. We use the symbols G , $\hat{H}$, or L , depending on the context (general, Hamiltonian, Liouvillian). Examples for supported forms a Hamiltonian are the following, from the most general case to simplest and most common case of linear controls,

$$\hat{H} = \overbrace{\hat{H}_0}^{\text{drift term}} + \sum_l \overbrace{\hat{H}_l(\{\epsilon_l(t)\}, t)}^{\text{control term}} \quad (\text{G1})$$

$$\hat{H} = \hat{H}_0 + \sum_l \overbrace{a_l(\{\epsilon_l(t)\}, t)}^{\text{control amplitude}} \hat{H}_l \quad (\text{G2})$$

$$\hat{H} = \hat{H}_0 + \sum_l \overbrace{\epsilon_l(t)}^{\text{control function}} \underbrace{\hat{H}_l}_{\text{control operator}} \quad (\text{G3})$$

Dynamical Generator


juliaquantumcontrol.github.io


Glossary



Generator — Dynamical generator (Hamiltonian / Liouvillian) for the time evolution of a state, i.e., the right-hand-side of the equation of motion (up to a factor of i) such that $|\Psi(t + dt)\rangle = e^{-i\hat{H}dt}|\Psi(t)\rangle$ in the infinitesimal limit. We use the symbols G , $\hat{H}$, or L , depending on the context (general, Hamiltonian, Liouvillian). Examples for supported forms a Hamiltonian are the following, from the most general case to simplest and most common case of linear controls,

$$\hat{H} = \overbrace{\hat{H}_0}^{\text{drift term}} + \sum_l \overbrace{\hat{H}_l(\{\epsilon_l(t)\}, t)}^{\text{control term}} \quad (\text{G1})$$

$$\hat{H} = \hat{H}_0 + \sum_l \overbrace{a_l(\{\epsilon_l(t)\}, t)}^{\text{control amplitude}} \hat{H}_l \quad (\text{G2})$$

$$\hat{H} = \hat{H}_0 + \sum_l \overbrace{\epsilon_l(t)}^{\text{control function}} \underbrace{\hat{H}_l}_{\text{control operator}} \quad (\text{G3})$$

Generator Interface



juliaquantumcontrol.github.io



API / Subpackages / QuantumPropagators



```
@test check_generator(generator; state, tlist,  
                      for_mutable_state=true, for_immutable_state=true,  
                      for_expval=true, atol=1e-15)
```



verifies the given generator:

- `get_controls(generator)` must be defined and return a tuple
- all controls returned by `get_controls(generator)` must pass `check_control`
- `evaluate(generator, tlist, n)` must return a valid operator (`check_operator`), with forwarded keyword arguments (including `for_expval`)
- `evaluate!(op, generator, tlist, n)` must be defined
- `substitute(generator, replacements)` must be defined
- If `generator` is a `Generator` instance, all elements of `generator.amplitudes` must pass `check_amplitude`.

[source](#)

Generator Interface



juliaquantumcontrol.github.io



API / Subpackages / QuantumPropagators



```
@test check_generator(generator; state, tlist,  
                      for_mutable_state=true, for_immutable_state=true,  
                      for_expval=true, atol=1e-15)
```

verifies the given generator:

- 
- `get_controls(generator)` must be defined and return a tuple
 - all controls returned by `get_controls(generator)` must pass `check_control`
 - `evaluate(generator, tlist, n)` must return a valid operator (`check_operator`), with forwarded keyword arguments (including `for_expval`)
 - `evaluate!(op, generator, tlist, n)` must be defined
 - `substitute(generator, replacements)` must be defined
 - If `generator` is a `Generator` instance, all elements of `generator.amplitudes` must pass `check_amplitude`.

[source](#)

Generator Interface



juliaquantumcontrol.github.io



API / Subpackages / QuantumPropagators



```
@test check_generator(generator; state, tlist,  
                      for_mutable_state=true, for_immutable_state=true,  
                      for_expval=true, atol=1e-15)
```

verifies the given generator:

- `get_controls(generator)` must be defined and return a tuple
- all controls returned by `get_controls(generator)` must pass `check_control`
- `evaluate(generator, tlist, n)` must return a valid operator (`check_operator`), with forwarded keyword arguments (including `for_expval`)
- `evaluate!(op, generator, tlist, n)` must be defined
- `substitute(generator, replacements)` must be defined
- If `generator` is a `Generator` instance, all elements of `generator.amplitudes` must pass `check_amplitude`.

[source](#)

Propagator Interface



juliaquantumcontrol.github.io



Overview



The Propagator interface

As a lower-level interface than `propagate`, the `QuantumPropagators` package defines an interface for "propagator" objects. These are initialized via `init_prop` as, e.g.,

```
using QuantumPropagators: init_prop

propagator = init_prop( $\Psi_0$ , H, tlist)
```

The `propagator` is a propagation-method-dependent object with the interface described by `AbstractPropagator`.

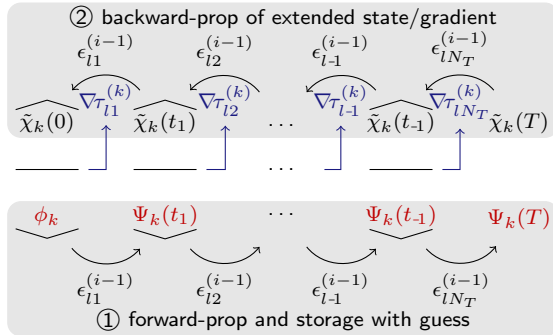
The `prop_step!` function can then be used to advance the `propagator`:

```
using QuantumPropagators: prop_step!

 $\Psi$  = prop_step!(propagator) # single step
```

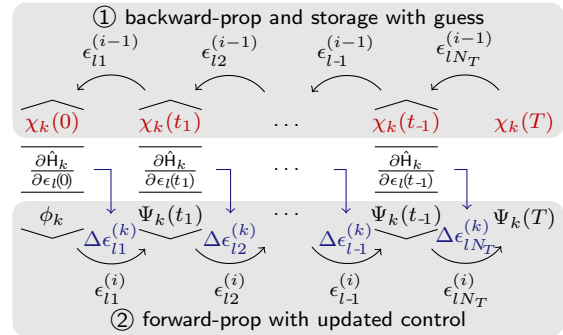
GRAPE and Krotov Numerical Scheme Comparison

(a) GRAPE



concurrent update

(b) Krotov's method



sequential update

— Goerz *et al.* Quantum 6, 871 (2022)

Open Quantum Systems

Lindblad equation:

$$\begin{aligned}\frac{d}{dt}\hat{\rho}(t) &= -i \left[\hat{H}, \hat{\rho}(t) \right] + \mathcal{L}_D(\hat{\rho}(t)) \\ &= -i \left[\hat{H}, \hat{\rho}(t) \right] + \sum_k \left(\hat{A}_k \hat{\rho} \hat{A}_k^\dagger - \frac{1}{2} \hat{A}_k^\dagger \hat{A}_k \hat{\rho} - \frac{1}{2} \hat{\rho} \hat{A}_k^\dagger \hat{A}_k \right)\end{aligned}$$

Vectorization rule:

$$\text{vec} \left(\hat{A} \hat{\rho} \hat{B} \right) = \left(\hat{B}^T \otimes \hat{A} \right) \vec{\rho}$$

Matrix representation of Lindbladian:

$$\hat{L} = -i(\mathbf{1} \otimes \hat{H}) + i(\hat{H}^T \otimes \mathbf{1}) + \sum_k \left[(\hat{A}_k^\dagger)^T \otimes \hat{A}_k - \frac{1}{2} \left(\mathbf{1} \otimes \hat{A}_k^\dagger \hat{A}_k \right) - \frac{1}{2} \left((\hat{A}_k^\dagger \hat{A}_k)^T \otimes \mathbf{1} \right) \right]$$

— Goerz *et al.*, arXiv:1312.0111v2 (2021), Appendix B