QuTiP Control

Alexander Pitchford

30 June 2015

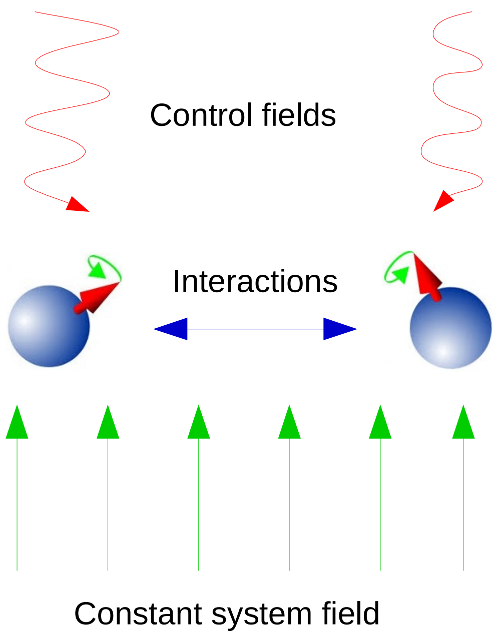
Tutorial Overview

- QuTiP introduction
- Quantum Control
- GRAPE
- QuTiP control modules
  - History
  - Features
  - Code object model
- Worked examples

QuTiP

- Quantum Toolbox in Python
- Simulate dynamics of quantum systems
  - Open systems solvers
  - Visualisation tools
- Open source
- Open platform – Linux, Mac OS, MS Windows
- License free

Quantum Control



- System or Drift Hamiltonian $$H_0 = H_i + H_s$$
- Control fields $$H(t) = H_0 + \sum_{j=1} u_j(t) H_j$$
- Combined Hamiltonian
- Schrödinger equation $$\dot{U}(t) = -iH(t)U(t)$$
- System evolves over time $$U(T)$$
- Target evolution $$U_{target}$$
- Can we determine pulses?
  - Maximise fidelity
  - Minimise energy
  - Minimise time

GRAPE Algorithm

- GRadient Ascent Pulse Engineering [1]
- Timesliced control amplitudes
- Hamiltonian constant within the timeslot
- Propagation
- Time evolution
- Fidelity error $\varepsilon = 1 - f$
- Normalisation
  - Unitary dynamics
  - Frobenius norm [3]

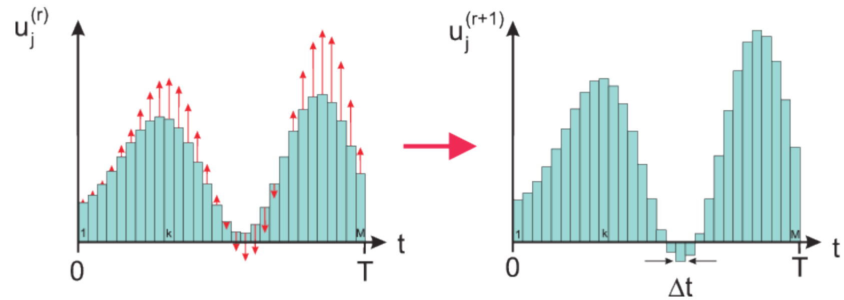


Diagram taken from [2]

$$H_{t_k} \approx H(t)$$

$$X_k := e^{-iH_{t_k}\Delta t}$$

$$X(t_k) := X_k X_{k-1} \cdots X_1 X_0$$

$$f = \frac{1}{N}\{X_{\text{target}}^{\dagger} X(T)\} \qquad f_{PSU}^2 = \tfrac{1}{N^2}|\{X_{\text{target}}^{\dagger} X(T)\}|^2$$

$$\varepsilon = \lambda\{\Lambda^{\dagger}\Lambda\} \qquad \Lambda = X_{\text{target}} - X(T)$$

References:

1. N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbruggen, and S. J. Glaser, J. Magn. Reson. **172**, 296 (2005).
2. S. Machnes, U. Sander, S. J. Glaser, P. De Fouquieres, A. Gruslys, S. Schirmer, and T. Schulte-Herbrueggen, Phys. Rev. A **84**, 1 (2010).
3. F. F. Floether, P. de Fouquieres, and S. G. Schirmer, New J. Phys. **14**, 073023 (2012)

GRAPE – 'Hill climbing'

Minimise fidelity error w.r.t. the piece-wise control amplitudes

- Multi-variable function optimisation problem
- BFGS
  - Quasi 2nd-order Newton method
- Gradient calculation
- Propagator gradient
  - Unitary – spectral theorem [2]
  - General – augmented matrix [3]
  - Frechet derivative

$$\frac{\partial f}{\partial u_j} = \frac{1}{N}\{\Lambda^{\dagger}_{M:k+1}\left(\frac{\partial X_k}{\partial u_j}\right)X_{k-1:0}\} \qquad \Lambda^{\dagger}_{M:k+1} = U^{\dagger}_{targ}X_{M:k+1}$$

$$\frac{\partial X}{\partial u_j} \qquad \begin{pmatrix} X_k & \frac{\partial X}{\partial u_j} \\ 0 & X_k \end{pmatrix} = exp\begin{pmatrix} A_u\Delta t & A_{u_j}\Delta t \\ 0 & A_u\Delta t \end{pmatrix}$$

History

- 2012 - Group working with DYNAMO [2]
- Limitations of MATLAB licenses
- HPC Wales opportunity
- Jan 2014 – Started Qtrl development
  - Porting DYNAMO to Python
  - Open source, open platform – license free
- Oct 2014 – Merged into QuTiP
- Jan 2015 – Released in QuTiP 3.10
- April 2015 – Incorporated CRAB algorithm [4]

References:
2. S. Machnes, U. Sander, S. J. Glaser, P. De Fouquieres, A. Gruslys, S. Schirmer, and T. Schulte-Herbrueggen, Phys. Rev. A **84**, 1 (2010).
4. T. Caneva, T. Calarco, and S. Montangero, Phys. Rev. A - At. Mol. Opt. Phys. **84**, (2011).

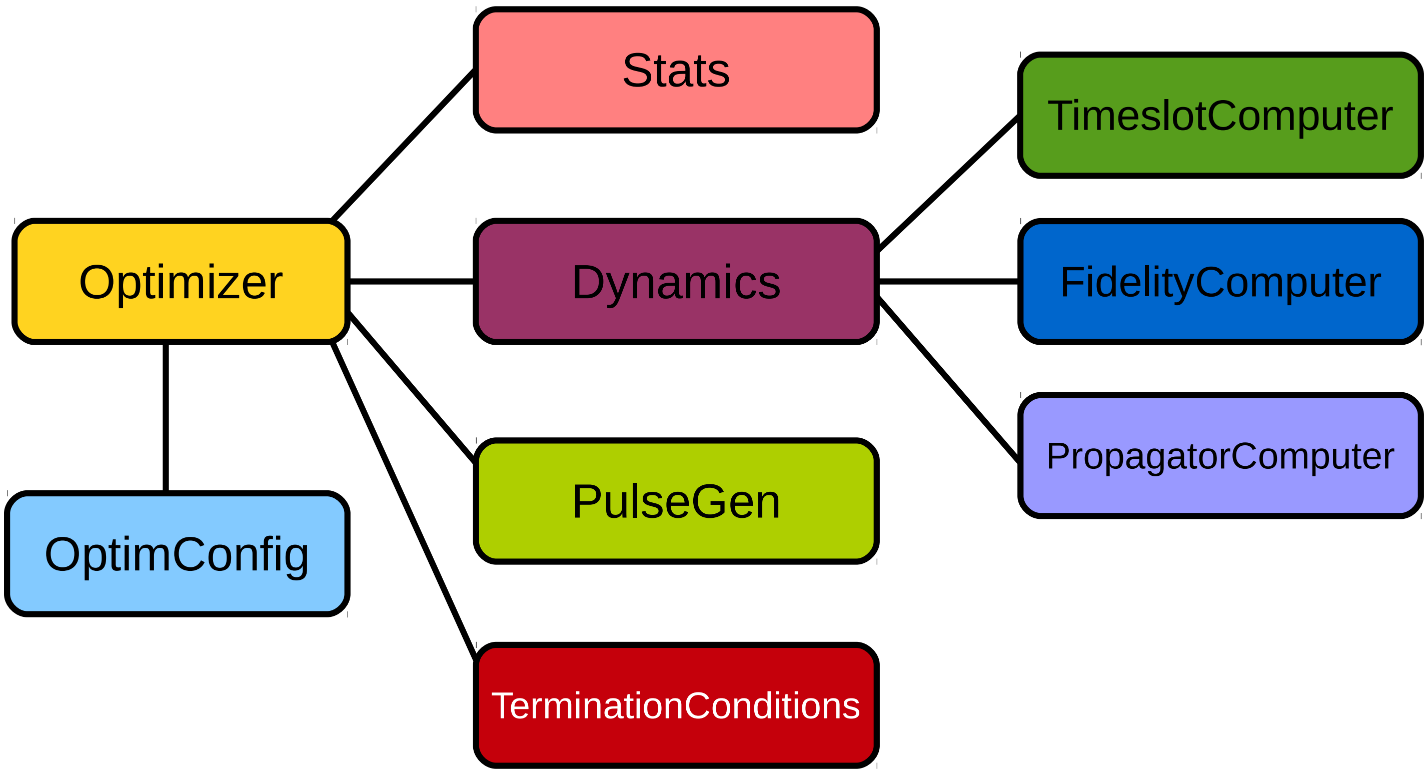
`qutip.control`

Features

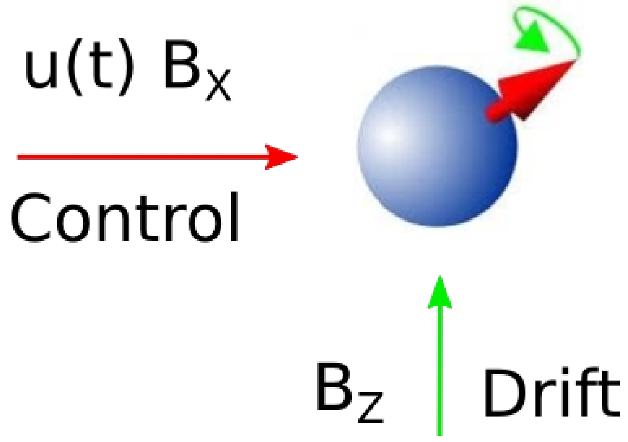
- Algorithms
  - GRAPE
  - CRAB
- Minimisation functions
  - L-BFGS-B
  - Any scipy.optimize
- Quantum dynamics
  - Closed systems – Unitary
  - Open systems – Lindbladian
  - QHO – Sympletic
- Fidelity measures
  - Unitary (overlap)
  - General (Frobenius)
  - Custom
- Gradient methods
  - Spectral theorem
  - Frechet derivative
  - Approximate methods

`qutip.control`

Code Object Model



Example
single qubit Hadamard



Unitary evolution

$$U_{targ} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

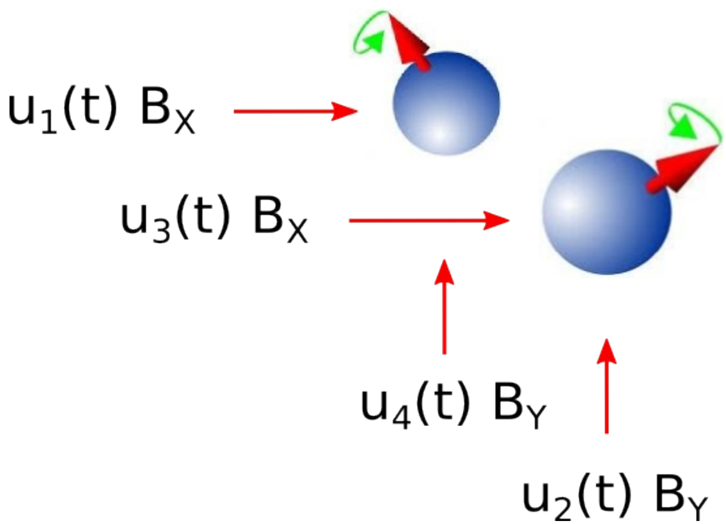
$$H_0 = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$H_1 = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Example
two qubit QFT

Unitary evolution

Target: Quantum Fourier Transform



$$H_1 = \sigma_x \otimes \mathbb{1}$$

$$H_2 = \sigma_y \otimes \mathbb{1}$$

$$H_3 = \mathbb{1} \otimes \sigma_x$$

$$H_4 = \mathbb{1} \otimes \sigma_y$$

$$H_0 = \frac{1}{2}(\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y + \sigma_z \otimes \sigma_z)$$

Example
Amplitude damping channel

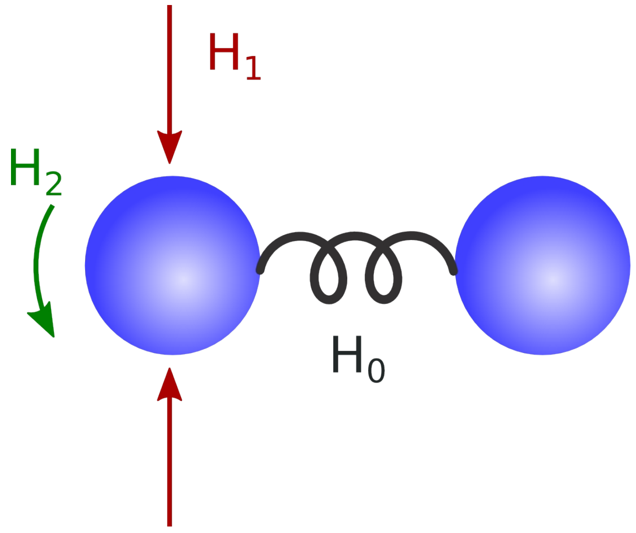
Markovian dynamics:
described by some Lindblad operator

Lindblad operator:
vectorising the density operator

$$\mathcal{L} = \gamma(2\sigma \otimes (\sigma^\dagger)^T - (\sigma^\dagger\sigma \otimes \mathbb{1} + \mathbb{1} \otimes (\sigma^\dagger\sigma)^T))$$

Controls added similarly

Example
Coupled oscillators



- Continuous variable
- Gaussian states
- Symplectic dynamics

Controls on first oscillator:
- Rotation
- Squeezing

Target: Boson swap

Finally

- Summary:
  - QuTiP control modules available for research and teaching
  - License free
  - Open source, open platform
- Future development plans:
  - Tighter integration with QuTiP
  - More fidelity types
  - Additional algorithms (Krotov?)
- Collaborators welcome