Introduction to the GRAPE Algorithm

Michael Goerz

June 8, 2010
Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms

Navin Khaneja\textsuperscript{a,*}, Timo Reiss\textsuperscript{b}, Cindie Kehlet\textsuperscript{b}, Thomas Schulte-Herbrüggen\textsuperscript{b}, Steffen J. Glaser\textsuperscript{b,*}

\textsuperscript{a} Division of Applied Sciences, Harvard University, Cambridge, MA 02138, USA
\textsuperscript{b} Department of Chemistry, Technische Universität München, 85747 Garching, Germany

Received 27 June 2004; revised 23 October 2004
Available online 2 December 2005
The Basic Idea

**Acronym**

**GRAPE:** Gradient Ascent Pulse Engineering

---

**Pulse Update**

\[ u_k(j) \rightarrow u_k(j) + \epsilon \frac{\partial \Phi_0}{\partial u_k(j)} \]

---

![Fig. 1. Schematic representation of a control amplitude \( u_k(t) \), consisting of \( N \) steps of duration \( \Delta t = T/N \). During each step \( j \), the control amplitude \( u_k(j) \) is constant. The vertical arrows represent gradients \( \delta \Phi_0/\delta u_k(j) \), indicating how each amplitude \( u_k(j) \) should be modified in the next iteration to improve the performance function \( \Phi_0 \).](image-url)
Working in Liouville Space

Density Matrix

$$|\psi\rangle \rightarrow \rho = |\psi\rangle \langle \psi|$$

Liouville-von Neumann Equation

$$\dot{\rho}(t) = -i [H, \rho(t)] = -i \left[ \left( H_0 + \sum_{k=1}^{m} u_k(t) H_k \right), \rho \right]$$

Time Propagation

$$U_j = \exp \left\{ -i \Delta t \left( H_0 + \sum_{k=1}^{m} u_k(j) H_k \right) \right\}$$

$$\rho(T) = U_N \cdots U_1 \rho(0) U_1^\dagger \cdots U_N^\dagger$$

$$= |\psi(T)\rangle \langle \psi(T)| \quad \text{with} \quad \psi(T) = U_n \cdots U_1 \psi(0)$$
Fidelity in Liouville space is defined in analogy to fidelity in Hilbert space: as the overlap between the propagated state with the optimal state.

<table>
<thead>
<tr>
<th>Fidelity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi_0 = \langle C</td>
</tr>
<tr>
<td>$C = O</td>
</tr>
<tr>
<td>$\rho(T) = U</td>
</tr>
</tbody>
</table>
Definition of Fidelity

Fidelity in Liouville space is defined in analogy to fidelity in Hilbert space: as the overlap between the propagated state with the optimal state.

\[ \Phi_0 = \langle C | \rho(\mathcal{T}) \rangle \equiv \text{tr} \left( C^\dagger \rho(\mathcal{T}) \right) \]

\[ C = O \left| \Psi(0) \right\rangle \left\langle \Psi(0) \right| O^\dagger \quad \rho(\mathcal{T}) = U \left| \Psi(0) \right\rangle \left\langle \Psi(0) \right| U^\dagger \]

Equivalence to “normal” fidelity

\[
\text{tr} \left( \left( C^\dagger \rho(\mathcal{T}) \right) \right) = \sum_n \left\langle n \left| O \right\rangle \left\langle \Psi(0) \right| \left\langle \Psi(0) \right| O^\dagger U \left| \Psi(0) \right\rangle \left\langle \Psi(0) \right| U^\dagger \left| n \right\rangle \\
= \left\langle \Psi(0) \right| U^\dagger \left( \sum_n \left\langle n \left| O \right\rangle \left\langle \Psi(0) \right| \left\langle \Psi(0) \right| O^\dagger U \left| \Psi(0) \right\rangle \left\langle \Psi(0) \right| U^\dagger \left| n \right\rangle \\
= \left\langle \Psi(0) \right| U^\dagger O \left| \Psi(0) \right\rangle \left\langle \Psi(0) \right| O^\dagger U \left| \Psi(0) \right\rangle \\
= \left| \left\langle \Psi(0) \right| O^\dagger U \left| \Psi(0) \right\rangle \right|^2
\]
A trace is invariant under cyclic permutation of its factors!

**Fidelity at T**

\[ \Phi_0 = \langle C | \rho(T) \rangle = \langle C | U_N \ldots U_1 \rho(0) U_1^\dagger \ldots U_N^\dagger \rangle = \langle U_{j+1}^\dagger \ldots U_N^\dagger C U_N \ldots U_{j+1} \rho(0) U_1 \ldots U_j^\dagger \rangle \]

**Propagated States \(\rightarrow\) Fidelity at \(t_j\)**

\[ \lambda_j \equiv U_{j+1}^\dagger \ldots U_N^\dagger C U_N \ldots U_{j+1} \quad \text{bw. propagated optimal state} \]
\[ \rho_j \equiv U_j \ldots U_1 \rho(0) U_1^\dagger \ldots U_j^\dagger \quad \text{fw. propagated initial state} \]

\[ \Phi_0 = \langle C | \rho(T) \rangle = \langle \lambda_j | \rho_j \rangle \]

Note: all propagations with guess pulse!
Calculation of Pulse Update

\[ u_k(j) \rightarrow u_k(j) + \epsilon \frac{\partial \Phi_0}{\partial u_k(j)} \]  
We need to calculate \( \frac{\partial \Phi_0}{\partial u_k(j)} \)

Two steps:
- For a variation \( \delta u_k(j) \), calculate \( \delta U_j \)
- Use \( \delta U_j \) to calculate \( \frac{\partial \Phi_0}{\partial u_k(j)} \)

Calculations are not completely trivial.

Solution:

\[ \frac{\partial \Phi_0}{\partial u_k(j)} = -\langle \lambda_j | i \Delta t [H_k, \rho_j] - \rangle \]
Grape Algorithm

Pulse Update

\[ u_k(j) \rightarrow u_k(j) + \epsilon \frac{\partial \Phi_0}{\partial u_k(j)}; \quad \frac{\partial \Phi_0}{\partial u_k(j)} = -\langle \lambda_j | i \Delta t [H_k, \rho_j] \rangle \]

1. Guess initial controls \( u_k(j) \)
2. Update pulse according to gradient:
   - Forward propagation of \( \rho(0) \): calculate and store all \( \rho_j = U_j \ldots U_1 \rho(0) U_1^\dagger \ldots U_j^\dagger \)
     for \( j \in [1, N] \)
   - Backward propagation of \( C \): calculate and store all \( \lambda_j = U_{j+1}^\dagger \ldots U_N^\dagger C U_N \ldots U_{j+1} \)
     for \( j \in [1, N] \)
   - Evaluate \( \frac{\partial \Phi_0}{\partial u_k(j)} \) and update the \( m \times N \) control amplitudes \( u_k(j) \)
3. Done if fidelity converges

Michael Goerz | Introduction to the GRAPE Algorithm
### Non-Hermitian Operators

\[
\Phi_1 = \Re[\Phi_0]; \quad \frac{\partial \Phi_1}{\partial u_k(j)} = -\langle \lambda_j^x | i \Delta t [H_k, \rho_j^x] \rangle - \langle \lambda_j^y | i \Delta t [H_k, \rho_j^y] \rangle
\]

\[
\Phi_2 = |\Phi_0|^2; \quad \frac{\partial \Phi_2}{\partial u_k(j)} = -2\Re \{ \langle \lambda_j | i \Delta t [H_k, \rho_j] \rangle \langle \rho_N^x | C \rangle \}
\]

### Unitary Transformations

\[
\Phi_3 = \Re \langle U_F | U(T) \rangle = \Re \langle U_{j+1}^\dagger \ldots U_N^\dagger U_F | U_j \ldots U_1 \rangle = \Re \langle P_j | X_j \rangle
\]

\[
\frac{\partial \Phi_3}{\partial u_k(j)} = -\Re \langle P_j | i \Delta t H_k X_j \rangle
\]

\[
\Phi_4 = |\langle U_F | U(T) \rangle|^2 = \langle P_j | X_j \rangle \langle X_j | P_j \rangle
\]

\[
\frac{\partial \Phi_4}{\partial u_k(j)} = -2\Re \{ \langle P_j | i \Delta t H_k X_j \rangle \langle X_j | P_j \rangle \}
\]

Also works with Lindblad-Operators. Additional energy constraints are possible.
Comparison with OCT

![Diagram of control amplitude](image)

Fig. 1. Schematic representation of a control amplitude $u_k(t)$, consisting of $N$ steps of duration $\Delta t = T/N$. During each step $j$, the control amplitude $u_k(j)$ is constant. The vertical arrows represent gradients $\delta \Phi_0/\delta u_k(j)$, indicating how each amplitude $u_k(j)$ should be modified in the next iteration to improve the performance function $\Phi_0$.

$$\Delta u(j) \sim \langle \Psi_{bw}(t_j) | \mu | \Psi_{fw}(t_j) \rangle$$

- GRAPE also needs forward- and backward-propagation, but only with old pulse. Propagated states also need to be stored.
- Pulse update at point $j$ in the current iteration does not depend on other updated pulse values (non-sequential update)
- All updates in GRAPE can in principle be calculated in parallel.
- Convergence tends to be pretty lousy (so I’m told)
- What about the choice of $\epsilon$?
Thank You!