

Linear Algebra Operations in a Quantum Control Problem

Author:

Kaveh Rahnema

rahnema@in.tum.de

Advisor:

Dr. Michael Bader

bader@in.tum.de

Ferienakademie, Sarntal/Süd Tirol , Italy

21st September-3rd October 2008

Contents

1. Classical Mechanics
 - 1.1. Newton's Laws of motion
 - 1.2. Lagrange Equations
2. Quantum Mechanics
 - 2.1. Uncertainty Relations
 - 2.2. Schrödinger Equation
3. Spin
 - 3.1. Discovery
 - 3.2. Spin Operators
 - 3.3. Pauli Matrices
 - 3.4. Spin Coupling
4. NMR
 - 4.1. Physics Behind it
 - 4.2. The GRAPE Algorithm
 - 4.3. Challenges of GRAPE
5. Matrix Exponential
 - 5.1. Definition
 - 5.2. Diagonalisation, Decomposition
 - 5.3. Approximating Matrix Exponential
 - 5.4. Scaling and Squaring
 - 5.5. Chebyshev Series Expansion
 - 5.6. Pade Approximation

Classical Mechanics

Newton stated that the sum of all of the forces acting on a solid body is proportional to the acceleration. The proportionality constant is mass of the body.

$$\sum f = m \ddot{x}$$

not bringing friction into account this formula can be replaced by Lagrange Equations

$$\frac{d}{dt} \left(\frac{\partial l}{\partial \dot{x}} \right) - \frac{\partial l}{\partial x} = 0, \quad l = \frac{1}{2} m v^2 - V(x)$$

where $V(x)$ is the potential energy.

Hamilton has shown that the Lagrange equation is equivalent to this system of two partial differential equations:

$$\frac{\partial H}{\partial p_k} = \dot{q}_k, \quad \frac{\partial H}{\partial q_k} = -\dot{p}_k, \quad H = \sum_{m=1}^k \frac{1}{2} m \dot{q}_m^2 + V(q_1, q_2, \dots, q_k)$$

$$p_k = m \dot{q}_k$$

P being the momentum, q the position and H the energy of the system.

Quantum Mechanics

In classical physics, $x(t)$ is a function which describes the trajectory of the center of mass exactly, where in quantum mechanics $x(t)$ is replaced by the wave function $\psi(x, t)$.

There will be no more classical functions and operators whose eigenvalues are the observable value take their places.

$$x \rightarrow \hat{x}, \quad p \rightarrow -i \hbar \nabla, \quad E \rightarrow i \hbar \partial_t$$

As it is seen in the formulas above there is a correlation between **Position-Momentum** and **Energy-Time**.

Using the transformations above and placing them into Hamilton's Equation will lead us to one of the most important Equations of Quantum Mechanics, The Schrödinger Equation.

$$\hat{H} \psi(x, t) = \left(\frac{-\hbar^2}{2m} \nabla^2 + \hat{V}(x) \right) \psi(x, t) = i \hbar \partial_t \psi(x, t)$$

As said before there will be a correlation between some operators in Quantum, when changing into Quantum Mechanics we will be dealing with probabilities not exact values, and the Heisenberg's uncertainty relations show us how exact we can be and at what costs.

$$\Delta x \Delta p \geq \hbar \quad , \quad \Delta E \Delta t \geq \hbar$$

This means if we want more accuracy in Position then we will lose accuracy in Momentum and vice versa.

The same goes for Energy Time but with slightly different interpretation (depending on the experiment) while the second one is derived from experiments and the first from Theory.

Spin

In 1922, Otto Stern and Walther Gerlach made an experiment with accelerated silver-atoms in an inhomogeneous magnetic field and found that the ray got split in two parts along the magnetic field axis. This was unexpected since the only relevant 5s valence electron has no orbital angular momentum and hence no magnetic dipole moment which could interact with the magnetic field. Today we know that electrons have an intrinsic attribute we call spin and which is correlated with a magnetic dipole moment with the so called g-factor. Spin is not simple angular momentum, because electrons are point-shaped and have no volume (as far as we know). The Schrödinger equation does not predict spin as there is no classical analogon to it and we derived the equation by applying the correspondence principle to the Hamilton function. To completely understand spin, one has to do relativistic calculations and use the Dirac equation, but this is beyond the scope of this paper.

Let z be the distinguished axis. From the Stern-Gerlach experiment we know that the eigenvalues of \hat{S}_z have to be $\pm \frac{\hbar}{2}$

Hence there have to be two different linear independent eigenvectors which we call (for historical reasons) $|\downarrow\rangle$, $|\uparrow\rangle$

Therefore we can write the spin state of our electron as a complex linear combination of these two vectors

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |\uparrow\rangle + \beta |\downarrow\rangle \in \mathbb{C}$$

Because $|\alpha|^2$ equals the probability of finding $|\uparrow\rangle$ in an experiment and $|\beta|^2$ equals the probability of finding $|\downarrow\rangle$, the normalization condition is

$$|\alpha|^2 + |\beta|^2 = 1$$

The spin operator has to satisfy $[\hat{S}_x, \hat{S}_y] = i\hbar\hat{S}_z$ and cyclical with $[A, B] = AB - BA$ being the commutator.

The spin operators in the three dimensions can be written as matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad , \quad \hat{S}_i = \frac{\hbar}{2}\sigma_i$$

which are called the Pauli matrices.

For coupling spins one needs to know about Kronecker tensor product and by applying it to each 2 single spin states one can get to higher dimensions in the spin space.

$$\begin{aligned} \langle \uparrow | \otimes \langle \downarrow | &=: \langle \uparrow \downarrow | \\ \langle \uparrow | \otimes \langle \uparrow | &=: \langle \uparrow \uparrow | \\ \langle \downarrow | \otimes \langle \downarrow | &=: \langle \downarrow \downarrow | \\ \langle \downarrow | \otimes \langle \uparrow | &=: \langle \downarrow \uparrow | \end{aligned}$$

In general, one can couple n spins by producing the Kronecker product of all basis vectors, yielding 2^n basic states.

Now that we know how to construct spin product, we can try to model the potential energy due to spins.

The potential between two spins is direct proportional to the scalar product of the two spin operators or to be more exactly their eigenvalues:

$$\hat{V} = \mu \hat{S}^{(1)} \otimes \hat{S}^{(2)} = \mu (\hat{S}_z^{(1)} \otimes \hat{S}_z^{(2)} + \frac{1}{2} (\hat{S}_-^{(1)} \otimes \hat{S}_+^{(2)} + \hat{S}_+^{(1)} \otimes \hat{S}_-^{(2)}))$$

With μ being a constant and $\hat{S}_\pm = \hat{S}_x \pm i\hat{S}_y$ being the Ladder

Operators.

$$\begin{aligned} \hat{S}_+ |\uparrow\rangle &= 0 & \hat{S}_+ |\downarrow\rangle &= \hbar |\uparrow\rangle \\ \hat{S}_- |\uparrow\rangle &= \hbar |\downarrow\rangle & \hat{S}_- |\downarrow\rangle &= 0 \end{aligned}$$

Nuclear Magnetic Resonance

Nuclei of atoms have like electrons their own spin. One can couple multiple spins in an experimental setup and manipulate them by external magnetic fields. Spins can be measured by stimulated emission of radiation. The technical challenges include the creation of very strong magnetic fields ($\approx 20\text{T}$) and the compensation of the energy relaxation as well as avoiding of the decoherence. Each physical system propagates towards its energetic ground state. This energy relaxation called phenomena erases the qubits after a certain amount of time, which makes the quantum computer unusable. Also, the superposition of multiple spins can be destroyed by interaction with the environment; this is called decoherence and to avoid it, one has to isolate the experimental setup very carefully from the outside.

As said before the Schrödinger Equation gives us the solutions to Quantum problems:

$$\hat{H} \psi(x, t) = \left(\frac{-\hbar^2}{2m} \nabla^2 + \hat{V}(x) \right) \psi(x, t) = i \hbar \partial_t \psi(x, t)$$

In NMR the particles don't move so we won't need the $\frac{-\hbar^2}{2m} \nabla^2$ term Which is the Momentum part.

Furthermore we need to construct the Potential Operator, for that we already know the Potential between 2 particles and we can construct the Potential Operator as follows:

$$\hat{V}(x) = \frac{1}{2} \sum_{i \neq j} \mu_{ij} \hat{S}^{(i)} \otimes \hat{S}^{(j)}$$

This is a $2^n \times 2^n$ Matrix which can be diagonalized and we will refer to it as H_d

For more than 2 spins the general formula will look like:

$$\hat{V}_c = \sum_{k=0}^{n-1} (a_k 1_{2^k} \otimes \sigma_x \otimes 1_{2^{n-k-q}} + b_k 1_{2^k} \otimes \sigma_y \otimes 1_{2^{n-k-q}})$$

which is the control part and we will call it H_c

An easier way to construct the H_c is to use the recursion formula below

$$A_{n+1} = \begin{pmatrix} A_n & 1_{2^n} \\ 1_{2^n} & A_n \end{pmatrix} \quad A_0 = 0$$

A quantum gate is an operation on the spin state of the system which performs a desired change in it, e.g. NOT, NAND, XOR, . . . For each of these gates the desired operation can be described by a matrix U_G . So the challenge is: adjusting $H_j(t_k)$ so that $U(t)$ overlaps best with U_G for a given time $t=T$. It can be shown that maximizing $\Re \text{tr}(U_G^\dagger U(t))$ subject to $\partial_t U(t) = -i \hat{H} U(t)$ optimizes the propagator. To solve this task, Khajena and Glaser came up with a gradient flow algorithm they called "gradient ascent pulse engineering" (GRAPE):

1) Set initial controls $u_j^{(r)}(t_k)$ for all times t_k at random or by guess

2) For each $k \in \{1, 2, \dots, M-1, M\}$ do:

2.1) Calculate the forward-propagation

$$u(t_k) = e^{-i \Delta \hat{H}(t_k)} e^{-i \Delta \hat{H}(t_{k-1})} \dots e^{-i \Delta \hat{H}(t_1)}$$

2.2) Calculate the backward-propagation

$$\Lambda(t_k) = e^{-i \Delta \hat{H}(t_k)} e^{-i \Delta \hat{H}(t_{k+1})} \dots e^{-i \Delta \hat{H}(t_M)}$$

2.3) Update $u_j^{(r+1)}(t_k) = u_j^{(r)}(t_k) + \epsilon \Re(\text{tr}(\Lambda^\dagger(t_k) (-i \hat{H}_j) u(t_k)))$

3) Return to step 2 with the new controls $u_j^{(r+1)}$

challenges of the GRAPE

GRAPE converges to a local optimum of $u(t)$. It is necessary to re-run it a couple of times with different initial values to confirm that the global maximum is reached although this cannot be proven.

One has to calculate the exponential of a sparse matrix

$$u_k := e^{-i\Delta t \hat{H}(t_k)}$$

One has to calculate the product of many different matrices

$$u(t_k) = u_k \cdot u_{k-1} \dots u_1$$

One has to calculate the trace

$$\text{tr} \{ (u_k u_{k+1} \dots u_m) (-i \hat{H}_j) (u_k u_{k-1} \dots u_1) \}$$

Matrix Exponential

As we saw in the GRAPE Algorithm we need to calculate Matrix Exponential and the problem is that we need to calculate all intermediate matrices

So we need to find some ways to do this part fast enough.

- 1) Diagonalize A (Eigendecomposition)
- 2) Approximate the exponential function
 - i) with polynomials
 - a) TAYLOR series
 - b) CHEBYSHEV series expansion
 - ii) with rational functions
PADE approximation

Eigendecomposition

A Diagonal matrix exponential is trivial

$$A = \text{diag}(d_1, d_2, \dots, d_n)$$

$$e^A = \text{diag}(e^{d_1}, e^{d_2}, \dots, e^{d_n})$$

And if $H = SDS^{-1}$ then we can calculate the exponential in the following way.

$$e^A = S(\text{diag}(e^{d_1}, \dots, e^{d_n}))S^{-1}$$

But eigendecomposition is expensive.

So we need to approximate the matrix exponential.

Scaling and squaring

The quality of the approximations of either one of the two approximations discussed below is strongly dependent on the norm of the exponent being smaller than one. As one needs to compute the exponential of a matrix of arbitrary norm, a method to transform the general case of the problem to the special case for which the approximations work is needed. Fortunately, such a method exists. It is called scaling and squaring. The following equation is true for the matrix exponential:

$$e^A = (e^{A/r^k})^{r^k}$$

That means that one can multiply the exponent with a factor of $\frac{1}{r^k}$ to decrease its norm, then use the approximation to calculate the exponential, and square that result k times to obtain the exponential.

Of course, these calculations increase both the time needed to run and the numerical error, but they mean that the following two approximations can generally be used.

Chebyshev series expansion

The Chebyshev polynomials form an orthogonal basis in the interval $[-1, 1]$ with a slightly unusual metric. Any well behaved function — particularly the exponential function — can be expanded in this basis.

The basic idea of the Chebyshev series expansion is to approximate the exponential function by a partial sum of this expansion. The coefficients decrease as $\frac{1}{2^k k!}$, so even a polynomial of comparably low order can give a very good approximation.

Furthermore, this method can also be used to calculate matrix exponentials, provided that the norm of the argument matrix is smaller than one. Using the “Scaling and Squaring” technique, this prerequisite can be ensured.

As matrix polynomials of sparse matrices are comparably inexpensive to compute, especially when using elaborate multiplication schemes, the efficiency of this algorithm is rather high.

A well-behaved function $f: [-1,1] \rightarrow \mathbb{C}$ can be approximated by Chebychev polynomials $T_k(x)$

$$f(x) \approx \frac{a_0}{2} + \sum_{k=1}^m a_k T_k(x)$$

$$a_k := \frac{2}{\pi} \int_{-1}^1 f(x) T_k(x) \frac{dx}{\sqrt{1-x^2}}$$

For the exponential function, a_k decreases as $\frac{1}{2^k k!}$

PADE approximation

Like the Taylor series expansion, the Pade expansion approximates a well-behaved function, like the exponential function, at a single point, which will be zero in the given case. But unlike the Taylor series expansion, the Pade approximation uses a rational function instead of a polynomial to emulate the given function.

As the Taylor method did completely fail in terms of numerical stability, it is somewhat surprising that the Pade method actually works quite well.

Like the Chebyshev method, the Pade method can be generalized to matrices. But there are similar restrictions on the norms of the argument matrices, as the Pade approximation is good only near zero.

Thus the “Scaling and Squaring” steps have to be applied before and afterward respectively.

The division is replaced by a matrix inversion when switching from real numbers to matrices. Of course, the inverse matrix, which has rather poor numerical properties, is never calculated explicitly, but rather indirectly by solving a corresponding system of linear equations.

It should be noted that the matrix inversion is more costly than matrix-matrix-multiplications for sparse matrices, which means that the Pade approximation has some initial disadvantage when compared to the Chebyshev method. Whether or not this is balanced by better approximations with smaller polynomials is dependent on the size of the matrices.

For $x \in \mathbb{C}$ the Pade approximation $r_m(x)$ of e^x is given by

$$r_m(x) = \frac{p_m(x)}{q_m(x)}$$

$$p_m(x) = \sum_{j=0}^m \frac{(2m-j)! m!}{(2m)! (m-j)! j!} x^j,$$

$$q_m(x) = \sum_{j=0}^m \frac{(2m-j)! m! (-1)^j}{(2m)! (m-j)! j!} x^j$$

References

- 1) Optimal Control of coupled spin dynamics : design of NMR pulse sequences by gradient ascent algorithm, science direct, JMR 172 (2005) 296–305
- 2) Parallelizing Matrix Operations on Cluster for an Optimal Control-based Quantum Compiler, T. Gradl, A. Spörl, T. Huckle.
- 3) Maximilian Fischer's Presentation in JASS 2008
- 4) Philip Klenze's Presentation in JASS 2008
- 5) Quantum Mechanics 2nd ed 1996, Gasiorowitz, Stephan