Adiabatic Quantum Computing Applied to the 3-SAT Problem

by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx

Introduction

Quantum Adiabatic Commputing encodes a problem to be solved in a Hamiltonian (energy operator) in such way that its state of minimum energy ("ground state") represents the solution. Therefore adiabatic evolution of a quantum system to the ground state of the Hamiltonian and subsequent measurement gives the solution to the problem. This document uses Quantum-`Computing` *Mathematica* add-on to implement Adiabatic Quantum Computing applied to the 3-SAT problem.

The concept of Quantum Adiabatic Commputing was created by Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Michael Sipser (2000). "Quantum Computation by Adiabatic Evolution". arXiv:quant-ph/0001106v1.

Load the Package

First load the Quantum`Computing` package. Write:
Needs["Quantum`Computing`"];
then press at the same time the keys SHIFT-ENTER to evaluate. *Mathematica* will load the package:

```
Needs["Quantum`Computing`"]
```

```
Quantum`Computing` Version 2.2.0. (June 2010)
A Mathematica package for Quantum Computing
  in Dirac bra-ket notation and plotting of quantum circuits
by José Luis Gómez-Muñoz

Execute SetComputingAliases[] in order to use
  the keyboard to enter quantum objects in Dirac's notation
SetComputingAliases[] must be executed again in each new notebook that is created
```

In order to use the keyboard to enter quantum objects write:
SetComputingAliases[ ];
then press at the same time the keys SHIFT-ENTER to evaluate. The semicolon prevents *Mathematica* from printing the help message. Remember that SetComputingAliases[ ] must be evaluated again in each new notebook:

```
SetComputingAliases[];
```

The 3-SAT Problem

A "**clause**" in the 3-SAT problem is an expression that has the form:
$x_1 \lor x_2 \lor x_3$
where each x is a boolean variable or a negation of a boolean variable.

This is an example of a clause
$q_3 \lor \neg q_6 \lor \neg q_8$
where each q_k can be either FALSE or TRUE, and $\neg q_k$ is the negation of q_k.
Clauses will be identified in the following tables with a number such that, when this number is written in binary and 0 is interpreted as FALSE and 1 as TRUE, it is the only combination of values of q_k that does **not** satisfy (i.e. makes FALSE) the corresponding clause.

These are the eight possible clauses used in the 3-SAT:

Clause 0				
q_a	q_b	q_c	$q_a \lor q_b \lor q_c$	
False	False	False	False	0
False	False	True	True	1
False	True	False	True	2
False	True	True	True	3
True	False	False	True	4
True	False	True	True	5
True	True	False	True	6
True	True	True	True	7

Clause 1				
q_a	q_b	q_c	$q_a \lor q_b \lor \neg q_c$	
False	False	False	True	0
False	False	True	False	1
False	True	False	True	2
False	True	True	True	3
True	False	False	True	4
True	False	True	True	5
True	True	False	True	6
True	True	True	True	7

Clause 2				
q_a	q_b	q_c	$q_a \lor \neg q_b \lor q_c$	
False	False	False	True	0
False	False	True	True	1
False	True	False	False	2
False	True	True	True	3
True	False	False	True	4
True	False	True	True	5
True	True	False	True	6
True	True	True	True	7

Clause 3				
q_a	q_b	q_c	$q_a \vee \neg q_b \vee \neg q_c$	
False	False	False	True	0
False	False	True	True	1
False	True	False	True	2
False	True	True	False	3
True	False	False	True	4
True	False	True	True	5
True	True	False	True	6
True	True	True	True	7

Clause 4				
q_a	q_b	q_c	$\neg q_a \vee q_b \vee q_c$	
False	False	False	True	0
False	False	True	True	1
False	True	False	True	2
False	True	True	True	3
True	False	False	False	4
True	False	True	True	5
True	True	False	True	6
True	True	True	True	7

Clause 5				
q_a	q_b	q_c	$\neg q_a \vee q_b \vee \neg q_c$	
False	False	False	True	0
False	False	True	True	1
False	True	False	True	2
False	True	True	True	3
True	False	False	True	4
True	False	True	False	5
True	True	False	True	6
True	True	True	True	7

Clause 6				
q_a	q_b	q_c	$\neg q_a \vee \neg q_b \vee q_c$	
False	False	False	True	0
False	False	True	True	1
False	True	False	True	2
False	True	True	True	3
True	False	False	True	4
True	False	True	True	5
True	True	False	False	6
True	True	True	True	7

Clause 7				
q_a	q_b	q_c	$\neg q_a \lor \neg q_b \lor \neg q_c$	
False	False	False	True	0
False	False	True	True	1
False	True	False	True	2
False	True	True	True	3
True	False	False	True	4
True	False	True	True	5
True	True	False	True	6
True	True	True	False	7

A **3-SAT problem** is to find all the combinations of values of boolean variables that satisfy (make TRUE) an expression that has the form:
$\land_{j=1}^{m} \left(x_{1\,j} \lor x_{2\,j} \lor x_{3\,j}\right)$

This is an example of a very small 3-SAT expression with four variables q_a, q_b, q_c, q_d:
$(\neg q_b \lor q_c \lor \neg q_d) \land (\neg q_a \lor \neg q_b \lor \neg q_c) \land (q_a \lor q_c \lor q_d)$

The solutions to this example are given by all the rows that give TRUE ("satisfy the expression") in the last column of the following table:

First Example				
q_a	q_b	q_c	q_d	$(\neg q_b \lor q_c \lor \neg q_d) \land (\neg q_a \lor \neg q_b \lor \neg q_c) \land (q_a \lor q_c \lor q_d)$
False	False	False	False	False
False	False	False	True	True
False	False	True	False	True
False	False	True	True	True
False	True	False	False	False
False	True	False	True	False
False	True	True	False	True
False	True	True	True	True
True	False	False	False	True
True	False	False	True	True
True	False	True	False	True
True	False	True	True	True
True	True	False	False	True
True	True	False	True	False
True	True	True	False	False
True	True	True	True	False

3-SAT problems with **only one solution** are traditionally used in Adiabatic Quantum Computing. Next expression is an example of a 3-SAT problem with four variables q_0, q_1, q_2, q_3 and only one solution (q_0, q_1, q_2, q_3)=(True, True, False, True):

3 – SAT expression with four variables and only one solution				
q_0	q_1	q_2	q_3	$(\neg q_0 \lor \neg q_1 \lor q_3) \land (\neg q_0 \lor \neg q_2 \lor \neg q_3) \land (\neg q_0 \lor q_1 \lor q_2) \land (\neg q_0 \lor q_1 \lor q_3) \land (q_0 \lor \neg q_1 \lor \neg q_3) \land (q_0 \lor \neg q_1 \lor q_3) \land (q_0 \lor q_1 \lor \neg q_2) \land (q_0 \lor q_1 \lor q_2)$
False	False	False	False	False
False	False	False	True	False
False	False	True	False	False
False	False	True	True	False
False	True	False	False	False
False	True	False	True	False
False	True	True	False	False
False	True	True	True	False
True	False	False	False	False
True	False	False	True	False
True	False	True	False	False
True	False	True	True	False
True	True	False	False	False
True	True	False	True	True
True	True	True	False	False
True	True	True	True	False

Mathematica Program for the Generation of 3-SAT Clauses

Next *Mathematica* program generates 3-SAT clauses, as shown below. Place the cursor in any place on the code and evaluate (press [SHIFT]-[ENTER]) to store it in *Mathematica*'s memory. It only uses standard *Mathematica* commands:

```
Clear[q, c];
c_n_[args__] :=
 Module[{negationTable, varTable},
  negationTable = IntegerDigits[n, 2, 3] /. {1 → Not, 0 → Identity};
  varTable = Table[q_m, {m, {args}}];
  Apply[Or,
   Table[negationTable[[j]][varTable[[j]]], {j, Length[varTable]}]]
 ]
```

This is an example of a clause generated with the program that was evaluated above:

```
c_5[2, 7, 9]
```

```
! q_2 || q_7 || ! q_9
```

TraditionalForm[] gives a formating closer to the one used in books and papers:

```
TraditionalForm[c_5[2, 7, 9]]
```

$\neg q_2 \lor q_7 \lor \neg q_9$

This is the Truth-table for the clause:

```
Grid[
 Reverse[
  BooleanTable[{{q2, q7, q9}, c5[2, 7, 9]}, {q2, q7, q9}]],
 Dividers → All]
```

{False, False, False}	True
{False, False, True}	True
{False, True, False}	True
{False, True, True}	True
{True, False, False}	True
{True, False, True}	False
{True, True, False}	True
{True, True, True}	True

Quantum *Mathematica* Program for the Generation of 3-SAT Energy Functions Used in Quantum Adiabatic Computing

Quantum Adiabatic Commputing encodes a problem to be solved in a Hamiltonian (energy operator) in such way that its state of minimum energy ("ground state") represents the solution. Therefore adiabatic evolution of a quantum system to the ground state of that Hamiltonian and subsequent measurement gives the solution to the problem.
The Hamiltonian will be the addition of "energy functions". Each of this functions will "punish" those states (kets) that violate a clause of the 3-SAT expression by increasing their energy, so that the ket that does not violate any clause has the minimum energy.

This is a quantum expression that "punishes" qubits with a value of one $|1\rangle$ by increasing their energy. Remember that $\sigma_{0,\hat{1}}$ and $\sigma_{z,\hat{1}}$ are Pauli operators:

$$\text{QuantumTableForm}\left[\frac{1}{2}\left(\sigma_{0,\hat{1}} - \sigma_{z,\hat{1}}\right)\right]$$

	Input	Output
0	$\vert 0_{\hat{1}}\rangle$	0
1	$\vert 1_{\hat{1}}\rangle$	$\vert 1_{\hat{1}}\rangle$

This is a quantum expression that "punishes" qubits with a value of zero $|0\rangle$ by increasing their energy. $\sigma_{0,\hat{1}}$ and $\sigma_{z,\hat{1}}$ are Pauli operators:

$$\text{QuantumTableForm}\left[\sigma_{0,\hat{1}} - \frac{1}{2}\left(\sigma_{0,\hat{1}} - \sigma_{z,\hat{1}}\right)\right]$$

	Input	Output
0	$\vert 0_{\hat{1}}\rangle$	$\vert 0_{\hat{1}}\rangle$
1	$\vert 1_{\hat{1}}\rangle$	0

An operator that "punishes" the ket with qubit values corresponding to the binary representation of the number three ($0_{\hat{1}}$,$1_{\hat{2}}$,$1_{\hat{3}}$) can be built from a proper combination of the quantum expressions described above. This operator is an "Energy Function":

```
QuantumTableForm[(σ_{0,1̂} - 1/2 (σ_{0,1̂} - σ_{Z,1̂})) · (1/2 (σ_{0,2̂} - σ_{Z,2̂})) · (1/2 (σ_{0,3̂} - σ_{Z,3̂}))]
```

	Input	Output
0	$\vert 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0
1	$\vert 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0
2	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
3	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$
4	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0
5	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0
6	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
7	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	0

The table above is actually a representation of the truth table for the clause 3, where True is represented by zero (no "punishment") and False by the ket times an eigenvalue of one (a "punishment" of one unit of energy):

Clause **3**				
q_1	q_2	q_3	$q_1 \vee \neg q_2 \vee \neg q_3$	
False	False	False	True	0
False	False	True	True	1
False	True	False	True	2
False	True	True	**False**	**3**
True	False	False	True	4
True	False	True	True	5
True	True	False	True	6
True	True	True	True	7

	Input	Output
0	$\vert 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0
1	$\vert 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0
2	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
3	$\boldsymbol{\vert 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle}$	$\boldsymbol{\vert 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle}$
4	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0
5	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0
6	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
7	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	0

Next *Mathematica* program generates 3-SAT energy functions, similar to the one that was shown above.
Place the cursor in any place on the code and press [SHIFT]-[ENTER] to store it in *Mathematica*'s memory. It uses standard *Mathematica* commands and Quantum *Mathematica* commands:

```
Needs["Quantum`Computing`"];
h_c_[m_, n_, p_] :=
 Module[{digits, zeropos, onepos, ope},
  digits = IntegerDigits[c, 2, 3];
  zeropos = Flatten[Position[digits, 0]];
  onepos = Flatten[Position[digits, 1]];
  ope = ReplaceAll[
    ⊗_{j}^{onepos} (1/2 (σ_{0,ĵ} - σ_{Z,ĵ})) · ⊗_{k}^{zeropos} (σ_{0,k̂} - 1/2 (σ_{0,k̂} - σ_{Z,k̂})), {1̂ → m̂, 2̂ → n̂, 3̂ → p̂}];
  Expand[ope]
 ]
```

This is an example of an energy function generated with the program. Remember that $\sigma_{0,\hat{1}}$ and $\sigma_{Z,\hat{1}}$ are Pauli operators:

```
h5[2, 7, 9]
```

$$\frac{1}{8}\sigma_{0,\hat{2}}\cdot\sigma_{0,\hat{7}}\cdot\sigma_{0,\hat{9}}-\frac{1}{8}\sigma_{Z,\hat{2}}\cdot\sigma_{0,\hat{7}}\cdot\sigma_{0,\hat{9}}+\frac{1}{8}\sigma_{0,\hat{2}}\cdot\sigma_{Z,\hat{7}}\cdot\sigma_{0,\hat{9}}-\frac{1}{8}\sigma_{Z,\hat{2}}\cdot\sigma_{Z,\hat{7}}\cdot\sigma_{0,\hat{9}}-\frac{1}{8}\sigma_{0,\hat{2}}\cdot\sigma_{0,\hat{7}}\cdot\sigma_{Z,\hat{9}}+\frac{1}{8}\sigma_{Z,\hat{2}}\cdot\sigma_{0,\hat{7}}\cdot\sigma_{Z,\hat{9}}-\frac{1}{8}\sigma_{0,\hat{2}}\cdot\sigma_{Z,\hat{7}}\cdot\sigma_{Z,\hat{9}}+\frac{1}{8}\sigma_{Z,\hat{2}}\cdot\sigma_{Z,\hat{7}}\cdot\sigma_{Z,\hat{9}}$$

TraditionalForm[] gives an output format closer to the format used in books and papers:

```
TraditionalForm[h5[2, 7, 9]]
```

$$-\frac{1}{8}\sigma_2^Z\sigma_7^0\sigma_9^0+\frac{1}{8}\sigma_2^0\sigma_7^Z\sigma_9^0-\frac{1}{8}\sigma_2^Z\sigma_7^Z\sigma_9^0-\frac{1}{8}\sigma_2^0\sigma_7^0\sigma_9^Z+\frac{1}{8}\sigma_2^Z\sigma_7^0\sigma_9^Z-\frac{1}{8}\sigma_2^0\sigma_7^Z\sigma_9^Z+\frac{1}{8}\sigma_2^Z\sigma_7^Z\sigma_9^Z+\frac{1}{8}\sigma_2^0\sigma_7^0\sigma_9^0$$

The truth table for this energy function:

```
QuantumTableForm[h5[2, 7, 9]]
```

	Input	Output
0	$\lvert 0_{\hat{2}}, 0_{\hat{7}}, 0_{\hat{9}}\rangle$	0
1	$\lvert 0_{\hat{2}}, 0_{\hat{7}}, 1_{\hat{9}}\rangle$	0
2	$\lvert 0_{\hat{2}}, 1_{\hat{7}}, 0_{\hat{9}}\rangle$	0
3	$\lvert 0_{\hat{2}}, 1_{\hat{7}}, 1_{\hat{9}}\rangle$	0
4	$\lvert 1_{\hat{2}}, 0_{\hat{7}}, 0_{\hat{9}}\rangle$	0
5	$\lvert 1_{\hat{2}}, 0_{\hat{7}}, 1_{\hat{9}}\rangle$	$\lvert 1_{\hat{2}}, 0_{\hat{7}}, 1_{\hat{9}}\rangle$
6	$\lvert 1_{\hat{2}}, 1_{\hat{7}}, 0_{\hat{9}}\rangle$	0
7	$\lvert 1_{\hat{2}}, 1_{\hat{7}}, 1_{\hat{9}}\rangle$	0

Here we use the program `cn_[args___]`, which was defined in the previous section, in order to generate a 3-SAT expression, and it is stored in the variable **myexpr**:

```
myexpr = c6[0, 1, 3] && c7[0, 2, 3] && c4[0, 1, 2] &&
    c4[0, 1, 3] && c3[0, 1, 3] && c2[0, 1, 3] && c1[0, 1, 2] && c0[0, 1, 2];
TraditionalForm[myexpr ]
```

$$(\neg q_0 \lor \neg q_1 \lor q_3) \land (\neg q_0 \lor \neg q_2 \lor \neg q_3) \land (\neg q_0 \lor q_1 \lor q_2) \land (\neg q_0 \lor q_1 \lor q_3) \land (q_0 \lor \neg q_1 \lor \neg q_3) \land (q_0 \lor \neg q_1 \lor q_3) \land (q_0 \lor q_1 \lor \neg q_2) \land (q_0 \lor q_1 \lor q_2)$$

This particular 3-SAT expression has only one solution:

3 – SAT expression with only one solution				
q_0	q_1	q_2	q_3	$(\neg q_0 \vee \neg q_1 \vee q_3) \wedge (\neg q_0 \vee \neg q_2 \vee \neg q_3) \wedge (\neg q_0 \vee q_1 \vee q_2) \wedge (\neg q_0 \vee q_1 \vee q_3) \wedge (q_0 \vee \neg q_1 \vee \neg q_3) \wedge (q_0 \vee \neg q_1 \vee q_3) \wedge (q_0 \vee q_1 \vee \neg q_2) \wedge (q_0 \vee q_1 \vee q_2)$
False	False	False	False	False
False	False	False	True	False
False	False	True	False	False
False	False	True	True	False
False	True	False	False	False
False	True	False	True	False
False	True	True	False	False
False	True	True	True	False
True	False	False	False	False
True	False	False	True	False
True	False	True	False	False
True	False	True	True	False
True	True	False	False	False
True	True	False	True	True
True	True	True	False	False
True	True	True	True	False

Writting the Hamiltonian for the 3-SAT expression is very simple: instead of writting the clauses connected with "And" operators, write the corresponding energy functions conected with additions. Here we use the program `hc_[m_, n_, p_]`, which was defined above, in order to generate the Hamiltonian for the 3-SAT expression that was stored in the variable **myexpr**, and this Hamiltonian is stored in the variable **myhamilt**.

```
myhamilt = h6[0, 1, 3] + h7[0, 2, 3] + h4[0, 1, 2] +
  h4[0, 1, 3] + h3[0, 1, 3] + h2[0, 1, 3] + h1[0, 1, 2] + h0[0, 1, 2]
```

$$\frac{3}{8}\sigma_{0,\hat{0}}\cdot\sigma_{0,\hat{1}}\cdot\sigma_{0,\hat{2}}+\frac{1}{8}\sigma_{z,\hat{0}}\cdot\sigma_{0,\hat{1}}\cdot\sigma_{0,\hat{2}}+\frac{3}{8}\sigma_{0,\hat{0}}\cdot\sigma_{z,\hat{1}}\cdot\sigma_{0,\hat{2}}+$$
$$\frac{1}{8}\sigma_{z,\hat{0}}\cdot\sigma_{z,\hat{1}}\cdot\sigma_{0,\hat{2}}+\frac{1}{2}\sigma_{0,\hat{0}}\cdot\sigma_{0,\hat{1}}\cdot\sigma_{0,\hat{3}}+\frac{1}{8}\sigma_{0,\hat{0}}\cdot\sigma_{0,\hat{2}}\cdot\sigma_{0,\hat{3}}-\frac{1}{8}\sigma_{z,\hat{0}}\cdot\sigma_{0,\hat{2}}\cdot\sigma_{0,\hat{3}}-$$
$$\frac{1}{4}\sigma_{0,\hat{0}}\cdot\sigma_{z,\hat{1}}\cdot\sigma_{0,\hat{3}}-\frac{1}{4}\sigma_{z,\hat{0}}\cdot\sigma_{z,\hat{1}}\cdot\sigma_{0,\hat{3}}-\frac{1}{8}\sigma_{0,\hat{0}}\cdot\sigma_{z,\hat{2}}\cdot\sigma_{0,\hat{3}}+\frac{1}{8}\sigma_{z,\hat{0}}\cdot\sigma_{z,\hat{2}}\cdot\sigma_{0,\hat{3}}+$$
$$\frac{1}{8}\sigma_{0,\hat{0}}\cdot\sigma_{0,\hat{1}}\cdot\sigma_{z,\hat{2}}-\frac{1}{8}\sigma_{z,\hat{0}}\cdot\sigma_{0,\hat{1}}\cdot\sigma_{z,\hat{2}}+\frac{1}{8}\sigma_{0,\hat{0}}\cdot\sigma_{z,\hat{1}}\cdot\sigma_{z,\hat{2}}-$$
$$\frac{1}{8}\sigma_{z,\hat{0}}\cdot\sigma_{z,\hat{1}}\cdot\sigma_{z,\hat{2}}+\frac{1}{4}\sigma_{0,\hat{0}}\cdot\sigma_{0,\hat{1}}\cdot\sigma_{z,\hat{3}}-\frac{1}{4}\sigma_{z,\hat{0}}\cdot\sigma_{0,\hat{1}}\cdot\sigma_{z,\hat{3}}-$$
$$\frac{1}{8}\sigma_{0,\hat{0}}\cdot\sigma_{0,\hat{2}}\cdot\sigma_{z,\hat{3}}+\frac{1}{8}\sigma_{z,\hat{0}}\cdot\sigma_{0,\hat{2}}\cdot\sigma_{z,\hat{3}}+\frac{1}{8}\sigma_{0,\hat{0}}\cdot\sigma_{z,\hat{2}}\cdot\sigma_{z,\hat{3}}-\frac{1}{8}\sigma_{z,\hat{0}}\cdot\sigma_{z,\hat{2}}\cdot\sigma_{z,\hat{3}}$$

The truth table for the Hamiltonian shows that all the base kets are eigenkets of this Hamiltonian, but **the only one that has an eigenvalue (energy) of zero is the ket that represents the solution to the 3-SAT expression**. Furthermore, the eigenvalues (energies) give the number of clauses that are violated, therefore in this example most nonsolutions violate exactly one clause, except for the clause in the eigth row, which violates two clauses:

```
QuantumTableForm[myhamilt]
```

	Input	Output
0	$\left|0_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$\left|0_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$\left|0_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$
2	$\left|0_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$\left|0_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$
3	$\left|0_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$\left|0_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$
4	$\left|0_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$\left|0_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
5	$\left|0_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$\left|0_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$
6	$\left|0_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$\left|0_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$
7	$\left|0_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$\left|0_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$
8	$\left|1_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$2\left|1_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
9	$\left|1_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$\left|1_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$
10	$\left|1_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$\left|1_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$
11	$\left|1_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$\left|1_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$
12	$\left|1_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$\left|1_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
13	$\left|1_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$	0
14	$\left|1_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$\left|1_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$
15	$\left|1_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$\left|1_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$

Another way to visualize the same information is using the command QuantumEigensystemForm, which gives its output in order of decreasing eigenvalue. In this case the eigenvalue gives the number of clauses that are not satisfied by the solution represented by the corresponding ket. Therefore, the solution to the problem is given by the ket with eigenvalue cero, at the bottom of the table:

```
QuantumEigensystemForm[myhamilt]
```

Eigenvalue	Eigenvector
2	$\left|1_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
1	$\left|1_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$
1	$\left|1_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$
1	$\left|1_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
1	$\left|1_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$
1	$\left|1_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$
1	$\left|1_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$
1	$\left|0_{\hat{0}}, 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
0	$\left|1_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$

Simulation of the Adiabatic Evolution

So far we have created a 3-SAT expression (stored in the variable **myexpr**) and the Hamiltonian (stored in the variable **myhamilt**) that codifies the solution to the problem in its lowest energy state. In this section we will simulate the adiabatic evolution that would be the actual method of solution in a real adiabatic quantum computer.

```
Column[{
  TraditionalForm[myexpr],
  TraditionalForm[myhamilt] }, Dividers → All]
```

$$(\neg q_0 \lor \neg q_1 \lor q_3) \land (\neg q_0 \lor \neg q_2 \lor \neg q_3) \land (\neg q_0 \lor q_1 \lor q_2) \land (\neg q_0 \lor q_1 \lor q_3) \land (q_0 \lor \neg q_1 \lor \neg q_3) \land (q_0 \lor \neg q_1 \lor q_3) \land (q_0 \lor q_1 \lor \neg q_2) \land (q_0 \lor q_1 \lor q_2)$$

$$\frac{1}{8}\sigma_0^Z\sigma_1^0\sigma_2^0 + \frac{3}{8}\sigma_0^0\sigma_1^Z\sigma_2^0 + \frac{1}{8}\sigma_0^Z\sigma_1^Z\sigma_2^0 - \frac{1}{8}\sigma_0^Z\sigma_2^0\sigma_3^0 - \frac{1}{4}\sigma_0^0\sigma_1^Z\sigma_3^0 - \frac{1}{4}\sigma_0^Z\sigma_1^Z\sigma_3^0 - \frac{1}{8}\sigma_0^0\sigma_2^Z\sigma_3^0 + \frac{1}{8}\sigma_0^Z\sigma_2^Z\sigma_3^0 + \frac{1}{8}\sigma_0^0\sigma_1^0\sigma_2^Z - \frac{1}{8}\sigma_0^Z\sigma_1^0\sigma_2^Z + \frac{1}{8}\sigma_0^0\sigma_1^Z\sigma_2^Z + \frac{1}{4}\sigma_0^0\sigma_1^0\sigma_3^Z - \frac{1}{4}\sigma_0^Z\sigma_1^0\sigma_3^Z - \frac{1}{8}\sigma_0^0\sigma_2^0\sigma_3^Z + \frac{1}{8}\sigma_0^Z\sigma_2^0\sigma_3^Z + \frac{1}{8}\sigma_0^0\sigma_2^Z\sigma_3^Z - \frac{1}{8}\sigma_0^Z\sigma_1^Z\sigma_2^Z - \frac{1}{8}\sigma_0^Z\sigma_2^Z\sigma_3^Z + \frac{3}{8}\sigma_0^0\sigma_1^0\sigma_2^0 + \frac{1}{2}\sigma_0^0\sigma_1^0\sigma_3^0 + \frac{1}{8}\sigma_0^0\sigma_2^0\sigma_3^0$$

In the previous section all the spectrum of the Hamiltonian was obtained with the commands QuantumTableForm and QuantumEigensystemForm. However that is **not** the way an adiabatic quantum computer would work. The actual way of solving a problem in such a computer would be:

1. A Hamiltonian H_1 is desgined so that its lowest energy eigenstate codifies the solution to the problem. Notice that we can create H_1 even if we do not have any previous idea about the actual solution. For example, in the previous section myhamilt (which plays the role of H_1) was created just by a simple recipe: instead of writting the clauses connected with "And" operators, write the corresponding energy functions conected with additions.

2. Set up a quantum system in the lowest energy state of another Hamiltonian H_0. In the original approach of quantum adiabatic computing, this initial Hamiltonian H_0 does not have any information about the specific problem to be solved.

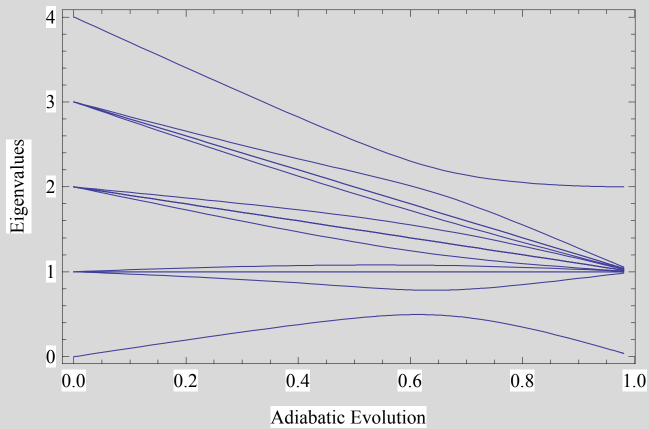
3. Adiabatically change the Hamiltonian of the system from H_0 to the Hamiltonian H_1. An adiabatic evolution ensures that the system will remain in the lowest energy state through all the process.

4. Measure the final state of the system. The result of this measurement is the solution to the problem.

The adiabatic evolution can be represented by a Hamiltonian $H(t) = (1 - t)\,H_0 + t\,H_1$ that dependes on a parameter t, with $0 \le t \le 1$. The commands below show the evolution of the eigenvalues (energy levels) from those of the initial Hamiltonian H_0 to those of the final Hamiltonian H_1. Notice the use of the standard *Mathematica* command Evaluate[], which is important in order to have a fast calculation. An important paremeter to estimate the time such an evolution would take is the minimum difference between the lowest eigenvalue (energy level) and the next one. This minimum difference is an "energy gap". A large energy gap is desirable because it means it is posible to "run" faster the evolution without loosing the adiabatic conditions.

$$\text{inihamilt} = \sum_{j=0}^{3} 1/2 \left(\sigma_{0,\hat{j}} - \sigma_{x,\hat{j}}\right);$$

```
hamilt[t_] := (1 - t) * inihamilt + t * myhamilt;
myeigenv[t_] := Evaluate[Eigenvalues[QuantumMatrix[hamilt[t]]]];
Plot[myeigenv[t], {t, 0, 0.98}, Frame -> True,
 FrameLabel → {"Adiabatic Evolution", "Eigenvalues"}]
```



Given that the system started at the lowest energy state of the initial Hamiltonian and that the evolution is adiabatic, the system will end in lowest energy state of the final Hamiltonian:

```
QuantumEigensystemForm[myhamilt, -1]
```

Eigenvalue	Eigenvector
0	$\left\| 1_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}} \right\rangle$

Then we can read (measure) the solution to the problem: $\left| 1_{\hat{0}}, 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}} \right\rangle$ means that q0 must be TRUE, q1 must be TRUE, q2 must be FALSE and q3 must be TRUE, and that is the solution to our 3-SAT problem, which was previously stored in myexpr:

```
ReplaceAll[myexpr, {q0 → True, q1 → True, q2 → False, q3 → True}]
```

```
True
```

by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx