# Adiabatic quantum computing

quantum system.

When the Hamiltonian H of a system is time independent, the solution of the Schrödinger equation ( $\hbar = 1$ )

is

The evolved state in the Hamiltonian eigenbasis  $H|n\rangle = E_n|n\rangle$  is

 $|\psi(t)\rangle =$ 



- The adiabatic quantum computing (AQC) is based on the adiabatic evolution of the
  - **Time-independent Hamiltonian** 

    - $i\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle$

- $|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle.$  $(\dot{H}=0)$

$$\sum c_n(0)e^{-iE_nt}|n\rangle.$$



If the initial state of the system is a Hamiltonian eigenvector,  $c_n(0) = \delta_{mn}$ , then, there is no transition among its eigenstates for t > 0

 $|\psi(t)\rangle =$ 

### **Time-dependent Hamiltonian**

For time-dependent Hamiltonians, the Schrödinger equation becomes  $i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle$ 

and the corresponding solution can be written as

$$|\psi(t)\rangle = \sum_{n} c$$

To solve the Schrödinger equation we need to find the coefficients  $c_n(t)$ .



$$e^{-iE_nt}|n\rangle$$
.

 $c_n(t)e^{-i\int_0^t E_n(\tau)d\tau} |n(t)\rangle$ 

### **Exercise:** Show that the coefficients of the state expansion $c_n(t)$ are now coupled to each other by<sup>1</sup>

# $\dot{c}_k = -c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1} c_k \langle k \, | \, \dot{k} \rangle - \sum_{k=1}^{n-1}$

### in which



### **Adiabatic evolution**

# provided the Hamiltonian changes sufficiently slowly.

$$\sum_{\substack{\neq k}} c_n \frac{\langle k | \dot{H} | n \rangle}{g_{nk}} e^{\frac{-i}{\hbar} \int_0^t g_{nk}(\tau) d\tau}$$

$$E_n(t)-E_k(t).$$

A quantum system that begins in the nondegenerate ground state of a timedependent Hamiltonian will remain in the instantaneous ground state

<sup>1</sup>M. S. Sarandy, L. A. Wu, and D. A. Lidar, Quantum Inf Process **3**, 331 (2004).







<sup>1</sup>M. S. Sarandy, L. A. Wu, and D. A. Lidar, *Quantum Inf Process* **3**, 331 (2004).

### **Adiabatic Theorem**



 $(n \neq k)$ 

$$\leq 1 |c_n(0)\langle k(s)| \frac{dH(s)}{ds} |n(s)\rangle|$$

$$\min_{0\leq s\leq 1}|g_{nk}(s)|^2$$

and 
$$s = \frac{t}{T}$$
 is the parametrized time.

AQC is a universal computational model;

In terms of computational complexity, it is polynomially equivalent to gate-based quantum computing<sup>1</sup>.

The solution to a desired problem is encoded in the ground state of a Hamiltonian (cost function). It is more robust to decoherence!

<sup>1</sup>D. Aharonov, et al., SIAM Journal on Computing **37**, 166 (2007). arXiv:quant-ph/0405098.







### Adiabatic quantum computing<sup>1</sup>

can be cast as the *minimization of some cost function* h.

contains the solution to the desired problem, as



The real time dependent coefficients A(t) and B(t) must satisfy

B(0) = 0 B(T) = 1

and

<sup>1</sup>E. Farhi, et al., arXiv:quant-ph/0001106v1 (2000); T. Albash and D. Lidar, Rev. Mod. Phys. **90**, 1 (2018).



- In the AQC model we solve optimization problems. Many computational problems
- The idea is to interpolate between an easy to prepare Hamiltonian  $H_I$  to a final one that

## $H(t) = A(t)H_I + B(t)H_P$

- A(0) = 1 A(T) = 0

- $[H_I, H_P] \neq 0.$





**Initial Hamiltonian:** In general, the initial Hamiltonian is encoded as

whose ground state is

**Final Hamiltonian:**  $H_P$  is called *problem Hamiltonian* and its ground state encodes the solution of the problem.

Adiabatic evolution path: In general, the functions A(t) and B(t) are chosen as

A(t) = s(t) and B(t) = 1 - s(t).

A common expression for the interpolation function is

 $s(t) = \frac{l}{T}$ 





### $|\psi_{GS}(0)\rangle = |+++...+\rangle.$

T is the total computing time.





### The running time and the gap

The final (solution) state is achieved with high probability if the runtime is sufficiently large. But how long is long enough? It depends on each specific problem.  $\bigcirc$  The total computing time T determines the efficiency of the algorithm, which is constrained by the adiabatic theorem. Output However, it is strongly dependent on the energy gap between the ground and first excited states  $g_{01} = |E_0 - E_1| \Rightarrow T \gg g_{01}^{-2}$ Only in simple cases it is possible to estimate de energy gap.  $\bigcirc$  There is no easy recipe to compute the runtime in the AQC model.  $\bigcirc$ 

$$T \gg \frac{\max_{0 \le s \le 1} |c_n(0)\langle k(s)| \frac{dH(s)}{ds} |n(s)\rangle|}{\min_{0 \le s \le 1} |g_{nk}(s)|^2}$$





### Some strategies to improve the computational time

Optimize the interpolation function: new time dependency; parameters with different speeds.

- Add intermediate hamiltonians to avoid energy gap closure.  $\bigcirc$  Choose different (simpler) final Hamiltonians, but that commute with  $H_P$  and have
- the same ground state.

T. Albash and D. Lidar, Rev. Mod. Phys. 90, 1 (2018).





Quantum annealing<sup>1</sup> (QA) is an optimization process for finding the global minimum of a given objective function over a given set of candidate solutions (candidate states), by a process using quantum fluctuations.

This method is strongly inspired on the classical method for optimization named Simulated Annealing. In this case, the thermal fluctuations are used to overcome the potential barriers.

**Annealing:** The goal is to find the equilibrium state of a system in the limit of low temperature. Steps:

(a) melting the material under study; (b) lowering the temperature very slowly; (c) spending a considerable amount of time in a range of temperatures close to freezing point.

<sup>1</sup>T. Kadowaki and H. Nishimori, *Phys. Rev. E*. **58**, 5355 (1998).

## **Quantum annealing**





Figure taken from C. C. McGeoch , Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice (2014).





### QA: thermal fluctuations are replaced by quantum fluctuations



problem.



In general, QA is recommended for applications where the search space is discrete, such as combinatorial optimization problems. Ex.: spin glass or the traveling salesman



### **Implementing Quantum Annealing**

contains the solution for an optimization problem.

2 - The system starts in an *arbitrary initial state*.

3 - Its adiabatic evolution is governed by the time-dependent Hamiltonian

zero over time.

 $H_D$  is the transverse field Hamiltonian, which does commute with  $H_F$ 

S. Aaronson, *Quantum Computing*, Lecture 15.



- 1 The objective function is encoded in the final Hamiltonian  $H_F$ , whose ground state

  - $H(t) = H_F + \Gamma(t)H_D,$
- where  $\Gamma(t)$  is the transverse field coefficient, which is initially very high, and reduces to





### The transverse field Hamiltonian

The disordering Hamiltonian  $H_D$  introduces kinetic energy to the annealing process in the form of quantum fluctuations of the solution space.

Decreasing  $\Gamma(t)$  moves the system closer to  $H_F$  while dampening the quantum fluctuations.

High values of  $\Gamma(t)$  allow the computation to escape local minima by "tunneling through" hills instead of climbing over them incrementally, as Simulated Annealing does. This is allows the algorithm to move "*faster*" and "*further*" across the landscape early in the process.

C. C. McGeoch, *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*. Synthesis Lectures on Quantum Computing **5**,1 (2014).



### **Practical implementation of Quantum Annealing:** DWave Systems<sup>1</sup>

In the DWave quantum annealers, the final Hamiltonian  $H_F$  is written as an Ising Hamiltonian

$$H_F = H_{Ising} =$$

Hamiltonian is given by

where s(t) is the scheduling function. The initial Hamiltonian is

<sup>1</sup>D-Wave Systems Inc., https://dwavesys.com. <sup>2</sup>F. Glover, G. Kochenberger, and Y. Du, 4OR-Q J Oper Res 17, 335 (2019).



$$\sum_{i} h_{i} \sigma_{z}^{i} + \sum_{ij} J_{ij} \sigma_{z}^{i} \sigma_{z}^{j},$$

- in which the coefficients  $h_i$  and  $J_{ij}$  encode the desired problem. The time-dependent
  - $H(t) = [1 s(t)]H_I + s(t)H_F,$

$$-\sum \sigma_x^n$$
.

n

After the total evolution time T, the qubits are measured in the computational basis, returning a string of bits

superconducting qubits are cooled down approximately 15 mK.

### **Observations:**

good.

2 - AQC and QA methods are sometimes treated as equals.

3 - Due to  $H_F = H_{Ising}$  the DWave quantum annealers are dedicated quantum computers, not universal ones.



- $|z_1 z_2 ... z_n\rangle$  s.t.  $z_i \in \{0, 1\}$
- However, beyond quantum fluctuations, there are thermal fluctuations, once the
- After many runs, the lowest energy eigenstate is taken as the solution to the problem.

- 1- Due to decoherence and several experimental imperfections, the results cannot be





### Finding the prime factors of N = pq via QA<sup>1</sup>

Our first task is to build a cost function to this problem,  $f_N(x,y)$ 

The solution to the problem is x = p and y = q s.t.  $f_N(x, y) = 0$ . Expanding x and y in binary representation  $x_i, y_k \in \{0,1\}$ 

$$x = \sum_{i=1}^{n_x} 2^i x_i + 1$$

$$F_N(x_1, \dots, x_{n_x}, y_1, \dots, y_{n_y}) = \left[ N - \left( \sum_{i=1}^{n_x} 2^i x_i + 1 \right) \left( \sum_{k=1}^{n_y} 2^k y_k + 1 \right) \right]^2$$

<sup>1</sup>X. Peng, et al., Phys. Rev. Lett. **101**, 220405 (2008).

$$= (N - xy)^2$$

$$y = \sum_{k=1}^{n_y} 2^k y_k + 1$$

We considering only odd numbers.





Quantizing the binary variables:

$$\hat{x_i} = \frac{I - \sigma_i^z}{2}$$

we obtain the problem Hamiltonian

$$H_N = \left[ NI - \left( \sum_{i=1}^{n_x} 2^i \hat{x}_i + I \right) \left( \sum_{k=1}^{n_y} 2^k \hat{y}_k + I \right) \right]^2$$

Bounds on the values of  $n_x$  and  $n_y$ <sup>1</sup>

$$n_x = \lceil \log_2 \lfloor \sqrt{N} \rfloor_{odd} \rceil - 1$$

 $[w]_{odd}$  means the largest odd integer not larger than w. [w] means the smallest integer larger than or equal to w.

<sup>1</sup>N. N. Hegade, et al., Phys. Rev. A **104**, L050403 (2021).



$$\hat{y_k} = \frac{I - \sigma_k^z}{2}$$

$$n_y = \lceil \log_2 \lfloor \frac{N}{3} \rfloor \rceil - 1$$

### Example: N=21

$$H_{21} = 16x_1^2y_1^2 + 64x_1^2y_1y_2 + 16x_1^2y_1 + 64x_1^2y_2^2 + 32x_1^2y_2 + 4x_1^2$$
  
+  $16x_1y_1^2 + 64x_1y_1y_2 - 152x_1y_1 + 64x_1y_2^2 - 304x_1y_2$   
-  $80x_1 + 4y_1^2 + 16y_1y_2 - 80y_1 + 16y_2^2 - 160y_2 + 400$ 

We can simplify the Hamiltonian taking into account the repeated binary variables:

$$x_i^2 = x_i$$

The simplified Hamiltonian is

$$H_{21} = 128x_1y_1y_2 - 104x - 76y_1 - 144y_2 + 4$$

Its ground state is  $|x_1y_1y_2\rangle = |111\rangle \Rightarrow \begin{cases} x = \\ y = \end{cases}$ 



$$y_k^2 = y_k$$

 $\frac{x_1y_1 - 144x_1y_2 - 76x_1 + 16y_1y_2}{400}$ 

$$= 1.2^{1} + 1 = 3$$
$$= 1.2^{2} + 1.2^{1} + 1 = 7$$

# Solving differential equations on quantum annealers: 2D heat equation

Heat equation in the stationary regime for a square plate of length L with multiple heat sources and sinks is

$$k\left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right] + \dot{q} = 0$$

### $\dot{q}(x,y)$ - heat-source function

k is the thermal conductivity of the material (we use k = 1 W/m K).

G. G. Pollachini, Phys. Rev. A **104**, 032426 (2021)



**Boundary conditions**  $T(x, 0) = 0^{\circ}C, x \in [0, L],$  $T(x, L) = \frac{x}{L} \times 100 \,^{\circ}\text{C}, \ x \in [0, L],$  $T(0, y) = 0^{\circ}C, y \in [0, L],$  $T(L, y) = \frac{y}{L} \times 100 \,^{\circ}\text{C}, \ y \in [0, L],$ 



### Using the finite element method, the plate can be divided in smaller squares

k[





$$\frac{\partial^2 T}{\partial x^2}(x_i, y_j) = \frac{T(x_{i+1}, y_j) - 2T(x_i, y_j) + T(x_{i-1}, y_j)}{(L/m)^2} + \frac{(L/m)^2}{12} \frac{\partial^4 T}{\partial x^4}(x_i^*, y_j),$$

$$\frac{\partial^2 T}{\partial y^2}(x_i, y_j) = \frac{T(x_i, y_{j+1}) - 2T(x_i, y_j) + T(x_i, y_j)}{(L/m)^2} + \frac{(L/m)^2}{12} \frac{\partial^4 T}{\partial y^4}(x_i, y_j^*)$$

$$T(x_{i+1}, y_j) - 4T(x_i, y_j) + T(x_{i-1}, y_j) + T(x_i)$$

+  $T(x_i, y_{j-1})$ ] +  $Q(x_i, y_j) = 0$ ,

 $Q = (L/m)^2 \dot{q}$ m = 11





### **Solving linear systems on quantum annealers**

The solution of the system of coupled linear equations can be found by minimizing the function

$$H(\mathbf{x}) = \left(A\mathbf{x} - \mathbf{b}\right)^{\dagger} \left(A\mathbf{x} - \mathbf{b}\right)$$

Each component can be represented in floating-point notation as

$$x_i \in \left[-d_i, 2c_i - d_i\right)$$

*R* describes the precision of the approximation.

D. O'Malley and V. V. Vesselinov, in 2016 IEEE High Performance Extreme Computing Conference (HPEC) (IEEE, 2016); M. L. Rogers and R. L. Singleton Jr, Frontiers in Physics 8, 265 (2020).



$$x_i = c_i \sum_{r=0}^{R-1} q_r^i 2^{-r} - d_i \qquad q_r^i \in \{0\}$$







### **Encoding the problem - discrete Hamiltonian**

$$H(q) = \sum_{r=0}^{R-1} \sum_{i=0}^{N-1} h_r^i q$$

$$h_{r}^{i} = -2\left(\sum_{j,k=0}^{N-1} A_{ki}A_{kj}c_{i}d_{j} + \sum_{j=0}^{N-1} A_{ji}c_{i}b_{j}\right)2^{-r}$$
$$J_{rs}^{ij} = \left(\sum_{k=0}^{N-1} A_{ki}A_{kj}c_{i}c_{j}\right)2^{-(r+s)}.$$



R - 1 N - 1 $q_r^i + \sum J_{rs}^{ij} q_r^i q_s^j$ r,s=0 i, j=0

### Coefficients

### Two indices for each qubit?

### Mapping the indexes

## $\ell(i, r) = iR + r$

### in which



### The inverse map

### The QUBO Hamiltonian





$$\ell = 0, \ldots, NR - 1$$

$$1, \ldots, N - 1$$

$$1, \ldots, R - 1$$

$$i_{\ell} = \lfloor \ell / R 
floor,$$
  
 $r_{\ell} = \ell \mod K$ 

$$q_{\ell} + \sum_{\ell,\kappa} b_{\ell,\kappa} q_{\ell} q_{\kappa}$$

DWave needs just the Hamiltonian coefficients.



# $\begin{array}{c|c} N/2 & A_{11} & A_{12} & \mathbf{x}_1 \\ N/2 & A_{21} & A_{22} & \mathbf{x}_2 & \mathbf{b}_2 \end{array}$ *N*/2 *N*/2

# Iterative solution $A_{11}\mathbf{x}_1^{(1)} = \mathbf{b}_1 - A_{12}\mathbf{x}_2^{(0)}$

 $A_{22}\mathbf{x}_{2}^{(1)} = \mathbf{b}_{2} - A_{21}\mathbf{x}_{1}^{(1)}$ 







### Initial conditions Distribution of sources and sinks





### **Relative error after steps** (Benchmarking the 2000Q and Advantage Systems)







### Relative error after steps with a shrink factor $\gamma = 0.8$ of at each step



the interval for estimating the solution at the kth iteration

 $[x_i^{k-1} - c_i \gamma^{k-1}]$ 



$$[x_i^{k-1} + c_i \gamma^{k-1}]$$

# Solution: temperature distribution on the plate







### Masters Scholarship in Quantum Computing

Start: beginning of 2023



## **Conclusions and perspectives**

Quantum computing is a very exciting field!!! This mini course is a first step towards the development of more sophisticated quantum algorithms. There are interesting open questions. I hope you have enjoyed ;)





