ION TRAP Quantum Computers

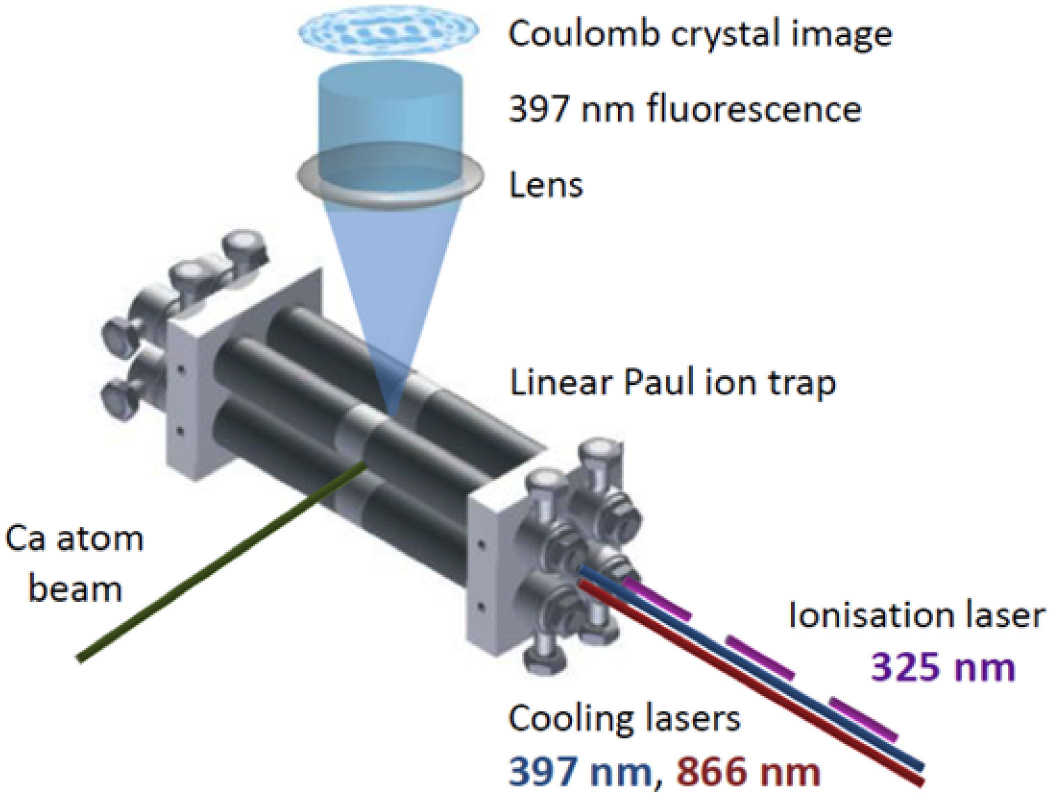
In order to build a quantum computer we need something that can serve as a qubit. It should behave as a quantum mechanical system but be scalable. Several candidates for possible quantum computer applications have been advanced. They include

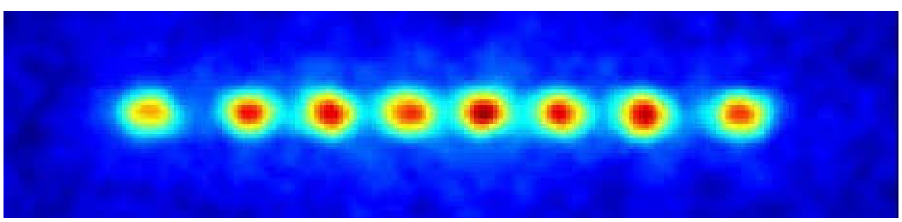
Ion - trap quantum computing (Cirac-Zoller protocol)
NMR (Nulcear Magnetic Resonance) quantum computers
Adiabatic quantum computing - Dwave
Photonic quantum computers
Cavity QED quantum computers etc.

We shall discuss the Cirac-Zoller scheme which treats a trapped atomic ion as the fundamental qubit.

A working quantum computer should allow individual qubits to be individually addressed, allow two or more qubits to "talk" to
each other (in order to built gates), and be scalable. The ion-trap quantum computer has already met these three criteria.

The linear rf-ion trap





You may remember from your physics 181 that the electric potential of a cylinder, held at a constant potential V_0 has the form

V(x,y) = $V_0 + \lambda \, \text{Log}(\frac{R}{\rho})$ for $\rho > R$

= V_0 for $\rho \leqslant R$

where $\rho=\sqrt{x^2+y^2}$ is the distance from the axis along the center of the cylinder, R is th radius of the cylinder and λ is a constant . Lets make a 3D plot of this potential.

```
V0 = 2;
v[x_, y_, x0_, y0_, R0_] := Which[ Sqrt[(x - x0) ^2 + (y - y0) ^2] > R0,
  V0 + Log[R0 / Sqrt[(x - x0) ^2 + (y - y0) ^2]] , Sqrt[(x - x0) ^2 + (y - y0) ^2] <= R0, V0]

Plot3D[v[x, y, 0, 0, 0.5], {x, -10, 10}, {y, -10, 10}, PlotPoints → 100, PlotRange → All]

dipole[x_, y_] := v[x, y, 0, 2, 0.5] - v[x, y, 0, -2, 0.5]

Plot3D[dipole[x, y], {x, -10, 10}, {y, -10, 10}, PlotPoints → 100, PlotRange → All]

quadrupole[x_, y_] :=
 v[x, y, 0, 2, 0.5] - v[x, y, -2, 0, 0.5] + v[x, y, 0, -2, 0.5] - v[x, y, 2, 0, 0.5]

Plot3D[quadrupole[x, y], {x, -4, 4}, {y, -4, 4}, PlotPoints → 100, PlotRange → All]
```

A more care full analysis shows that near the origin the potential has the form

V(x,y) = V_0 $(x^2\text{-}y^2)\ \Phi_0/2$ where Φ_0 is a constant.

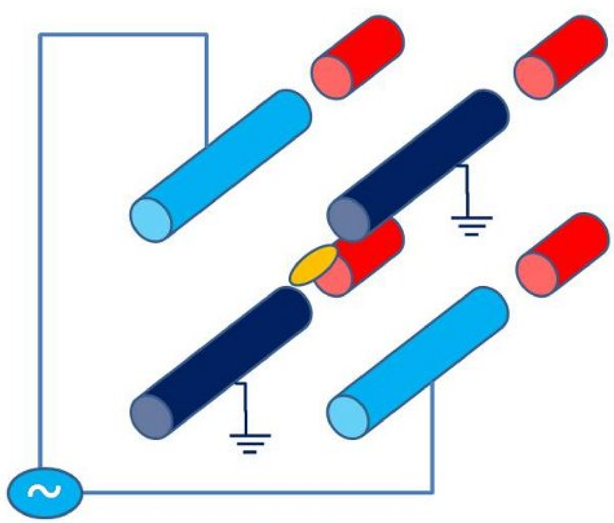
```
Vquad[x_, y_] = V0 (x^2 - y^2)
```

$2\ (x^2 - y^2)$

```
Plot3D[Vquad[x, y], {x, -4, 4}, {y, -4, 4}]
```

Now, instead of a constant voltage V_0 on the electrodes, we can change the polarity in a time-dependent way, lets consider
a Cos[ω t] dependence so that now the potential at the origin looks like,

V(x,y) = V_0 Cos(ω t) $(x^2\text{-}y^2)\ \Phi_0/2$



```
Vquad[x_, y_, t_] = V0 Cos[ω t] (x^2 - y^2) /. ω → 2 Pi
```

$2 \left(x^2 - y^2\right) \text{Cos}[2 \pi t]$

```
Manipulate[
 Plot3D[Vquad[x, y, t], {x, -4, 4}, {y, -4, 4}, PlotRange → {-30, 30}], {t, 0, 2}]
```

So how does a charged particle behave under the influence of such a "rotating saddle" potential ?
The force that a particle experiences is the negative of the gradient of the potential times its charge q i.e.

$F_x =$ -q x V_0 Cos(ω t)
$F_y =$ q y V_0 Cos(ω t)

and Newton's equation of motion are

m $\ddot{x} =$ -q x V_0 Cos(ω t)
m $\ddot{y} =$ q y V_0 Cos(ω t)

```
eq1 = x''[t] == - q / m x[t] Cos[ω t]  /. {q → 1, m → 1, ω → 2 Pi}
eq2 = y''[t] ==  q / m y[t] Cos[ω t]  /. {q → 1, m → 1, ω → 2 Pi}

(* initial conditions *)

x0 = 2.0;
y0 = -1.0;
v0x = 1.0;
v0y = 1.0;
```

$x''[t] == -\text{Cos}[2 \pi t] \, x[t]$

$y''[t] == \text{Cos}[2 \pi t] \, y[t]$

```
tend = 10;
sols = Flatten[NDSolve[{eq1, eq2, x[0] == x0, y[0] == y0, x'[0] == v0x, y'[0] == v0y},
    {x[t], y[t]}, {t, 0, tend}]];
fig1 = ParametricPlot[Evaluate[{x[t], y[t]} /. sols], {t, 0, tend}, PlotRange → All]
```

Analytic studies of the above equations show that the dynamics can be described by two types of motion, one is called
a micromotion and is evidenced above by the rapidly oscillating wiggles in the overall motion. More important is the overall
average motion that is described by a simple 2D harmonic oscillator potential that has the form

V(x,y) = 1/2 m Ω^2(x^2+ y^2)

$\Omega = \frac{q \Phi_0}{m \omega \sqrt{2}}$

this potential "binds" the system to the vicinity of the origin. We can analytically solve such a 2 D system, the result
is

x(t)= A_x Cos(Ω t + ϕ_x)
y(t)=A_y Cos(Ω t + ϕ_y)

where the quantities A_x A_y ϕ_x ϕ_y are related to the initial conditions of the ions motion

x(0)= x_0 = A_x Cos(ϕ_x)
y(0)= y_0 = A_y Cos(ϕ_y)

and $v_x(0)$ = v_{ox} = - Ω A_x Sin(ϕ_x) $v_y(0)$= v_{oy} = - Ω A_y Sin(ϕ_y) and so tan(ϕ_x) = - v_{ox} / Ω x_0
tan(ϕ_y) = - v_{oy} / Ω y_0
A_x = x_0/Cos(ϕ_x)
A_y = y_0/Cos(ϕ_x)

```
Omega = q / m / ω / Sqrt[2] /. {q → 1, m → 1, ω → 2 Pi};
phix = ArcTan[- v0x / x0 / Omega];
phiy = ArcTan[- v0y / y0 / Omega];
Ax = x0 / Cos[phix];
Ay = y0 / Cos[phiy];

xmotion[t_] = Ax Cos[Omega t + phix];
ymotion[t_] = Ay Cos[Omega t + phiy];

ion[t_] := Show[fig1, Graphics[{PointSize[0.05], Point[{xmotion[t], ymotion[t]}]}],
  PlotRange → {{-1.1 Ax, 1.1 Ax}, {-1.1 Ay, 1.1 Ay}}, Axes → True]

fig2 = Manipulate[ion[t], {t, 0, 2 Pi / Omega}]

Hyperlink["http://www.youtube.com/watch?v=XTJznUkAmIY"]
Hyperlink["http://www.youtube.com/watch?v=bkYXNeJ8IP0"]
Hyperlink["http://www.youtube.com/watch?v=PYpbKSmOnNc"]
```

```
http://www.youtube.com/watch?v=XTJznUkAmIY

http://www.youtube.com/watch?v=bkYXNeJ8IP0

http://www.youtube.com/watch?v=PYpbKSmOnNc
```