A simple vertex reconstruction algorithm using Quantum Computing techniques

XVth Quark Confinement and the Hadron Spectrum

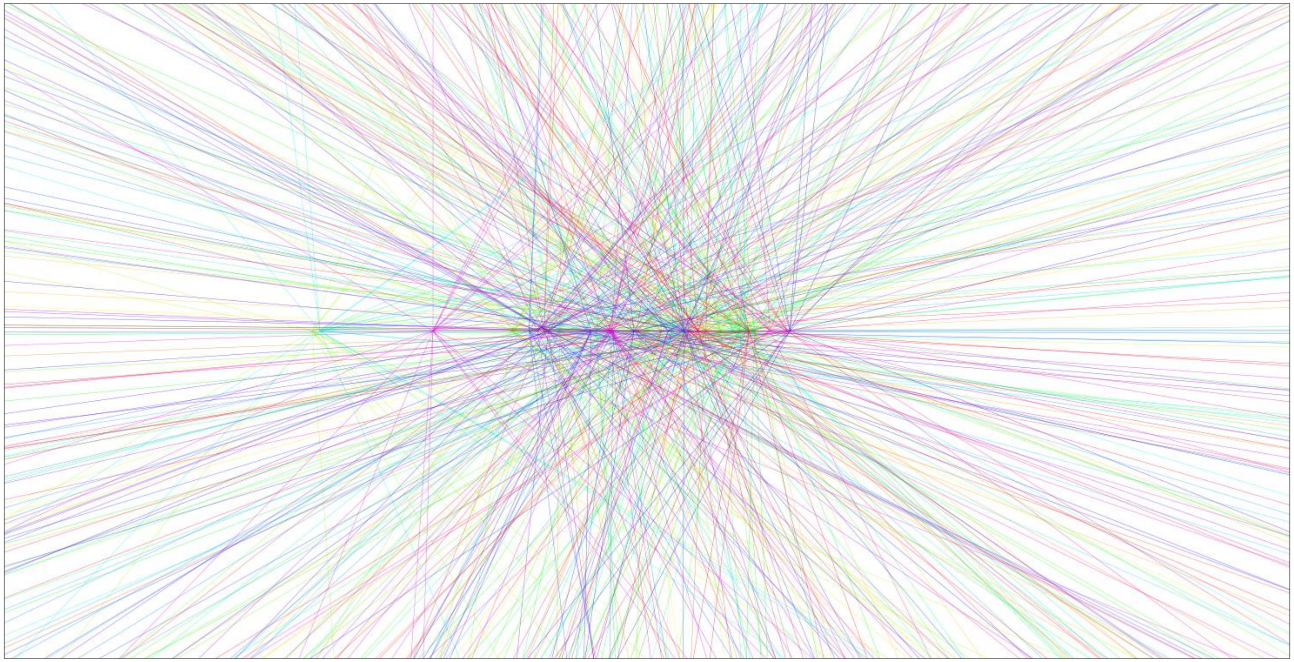
Aug 1st – 6th, 2022, University of Stavanger

F. Matorras, P. Martínez

UNIVERSIDAD DE CANTABRIA

Introduction: Vertexing in HEP

- Hadron colliders such as the LHC produce several simultaneous collisions per bunch crossing
  - The LHC during Run 2 had an average of 40 interactions per bunch crossing
  - This number is expected to scale up to 200 in the new High Luminosity LHC (HL-LHC)
- Most of these collisions are QCD interactions that largely increase the track multiplicity
- The physics program of the experiments rely on the ability to identify the primary vertex (PV)
- New, faster algorithms are being explored to reconstruct the PV in such a busy environment



Vertexing with Quantum Computing

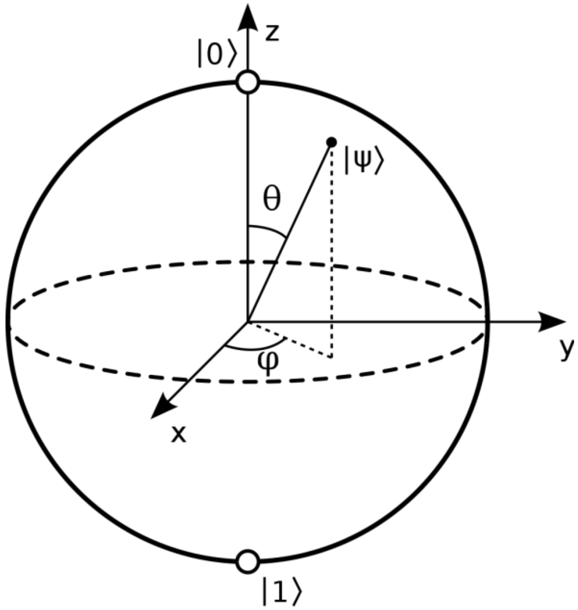
- This work aims at building a vertexing algorithm using quantum computing techniques
- A Quantum Computer uses quantum bits or qubits instead of classical bits with values 0 or 1
- Qubits are neither 0 or 1 but a linear combination of both fundamental states

$$|0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} ; |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

- Since $\alpha^2+\beta^2 = 1$, qubits are usually expressed as a direction in the Bloch sphere

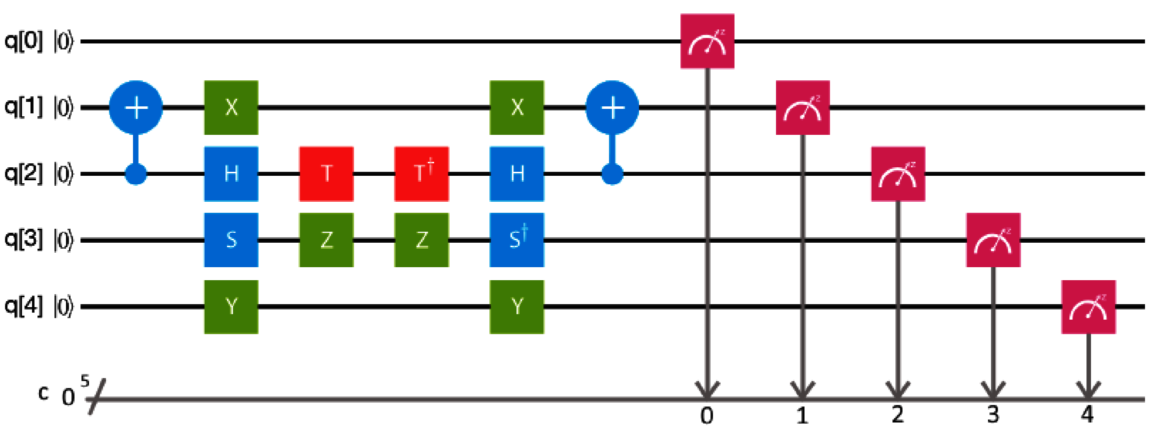
$$\alpha = cos(\theta/2)$$

$$\beta = e^{i\phi} sin(\theta/2)$$



Quantum logic gates and circuits

- Quantum logic gates are simple operations that can be applied on one or several qubits
  - The quantum states (wave function) is transformed after the application of a logic gate
  - Some common gates: Hadamard (H) or the rotation gate (R)
- Quantum computing programs are frequently written within the circuit paradigm
- A quantum circuit is a collection of qubits and logic gates that operate on them
- Circuits transform the initial state into other physical states
  - Measurements of the qubits can be done at any moment

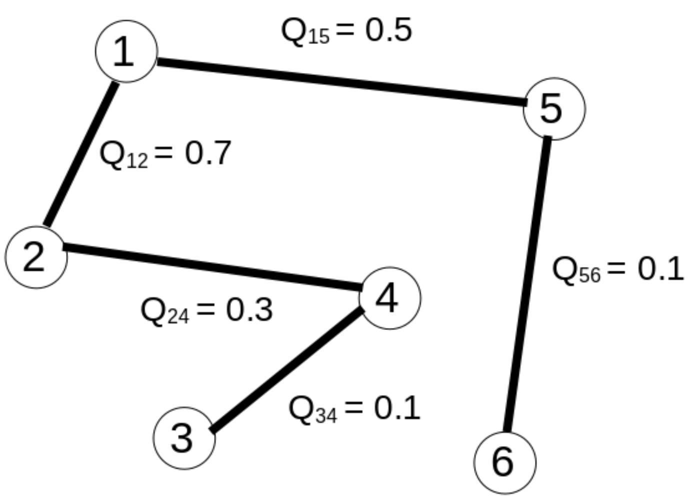


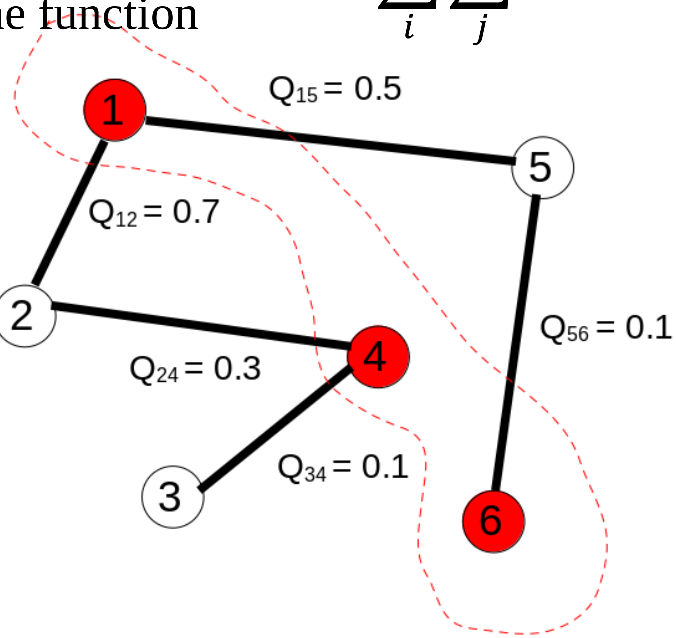
Operator	Gate(s)	Matrix
Pauli-X (X)	X, ⊕	$\begin{bmatrix}0&1\\1&0\end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix}0&-i\\i&0\end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix}1&0\\0&-1\end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix}1&0\\0&i\end{bmatrix}$
$\pi/8$ (T)	T	$\begin{bmatrix}1&0\\0&e^{i\pi/4}\end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$
Controlled Z (CZ)	Z	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}$
SWAP		$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{bmatrix}$

The Max-Cut problem

- Quantum computing algorithms are particularly good at solving combinatorial problems
- In this work we will consider the so called Max-Cut problem (NP-complete class problem)
  - Partition the nodes of a graph in 2 sets that maximize the number of edges between them
  - The edges can also have a weight in such a way that the total weighted sum is considered
- If solutions are encoded in vectors x = [0, 1, 0, 1, 1, 0] referring to which set belongs each node
  - The desired solution is the one that maximizes the function

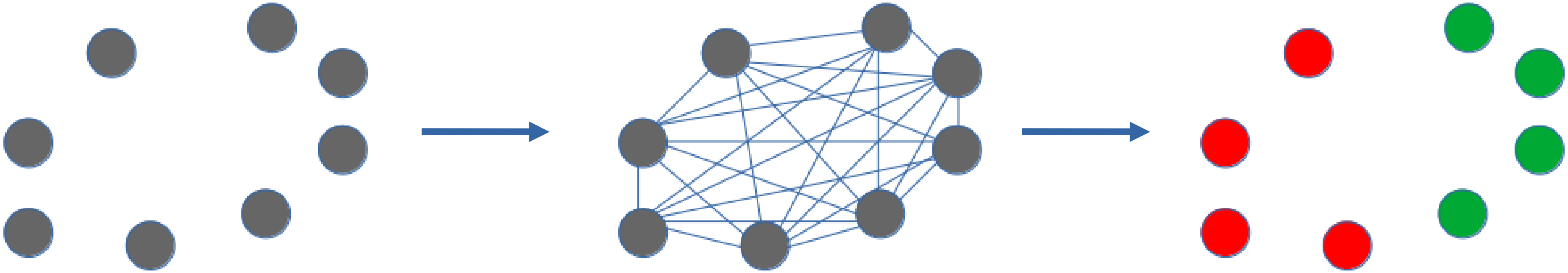
$$F = \sum_i \sum_j x_i Q_{ij} (1 - x_j)$$





From Max-Cut to vertex reconstruction

- Let’s consider a system formed by 2 vertices with a number of N tracks each
- Initially the position of the vertices is not known, nor the assignation of tracks to vertices
  - The only information we have is the coordinates of the POCA* point of every track
- Assigning tracks to vertices is in essence a clustering problem
  - Need to classify the tracks into two sub-sets, one belonging to vertex 1 and the other to 2
- Since tracks coming from a same vertex should be closer among them than other tracks
  - The clusters should maximize the distance between tracks of a cluster and the others
  - This is exactly the structure of a Max-Cut problem where all edges are taken into account
  - The weight of every edge is simply the distance bewteen nodes



POCA: Point Of Closest Approach to the beam line

Solution using Quantum Computing

- Most Quantum Computers can implement the so-called Ising Hamiltonians

$$H(\sigma) = -\sum_{\langle i\ j\rangle} J_{ij}\sigma_i\sigma_j - \mu\sum_j h_j\sigma_j$$

- Where the σ_i are the z component of the Pauli matrices for qubit number "i"
- In a Quantum Computer we can evaluate this hamiltonian for a multi-qubit state
- The cost function of the Max-Cut problem can be transformed into an Ising-like function
  - Consider the operator x_i that is 1 for qbit = |1> and 0 otherwise $\quad x_i = \frac{1+\sigma^{(z)}}{2}$
- With these operators the cost-function of the Max-Cut problem is an Ising Hamiltonian
- If the cost-function is multiplied by -1 then the solution to the Max-Cut is the minimum
  - The multi-qubit state with the lowest eigenvalue of the Ising Hamiltonian is the solution

$$F = -\sum_i\sum_j x_i Q_{ij}(1-x_j)$$

The VQE algorithm

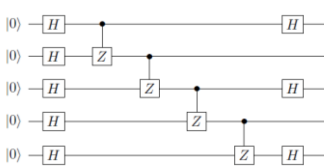
- The Variational Quantum Eigensolver (VQE) is a hybrid classic-quantum algorithm
  - Aiming at finding the multi-qubit state that minimizes a given Hamiltonian

Hamiltonian evaluation

$\langle\psi(\theta_i)|H|\psi(\theta_i)\rangle$ is computed by physically measuring the Hamiltonian a number of times

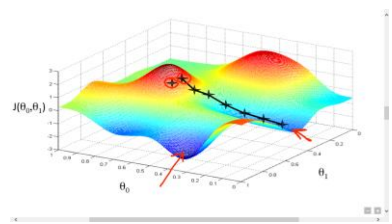
Ansatz generation

A Quantum Circuit generates an ansatz wave-function depending on parameters θ_i (randomly chosen for 1st iteration) → $|\psi(\theta_i)\rangle$



Minimization

A classic minimizer (COBYLA) finds the parameters θ_i that minimizes $\langle\psi(\theta_i)|H|\psi(\theta_i)\rangle$



Stop criteria fulfilled?

Yes

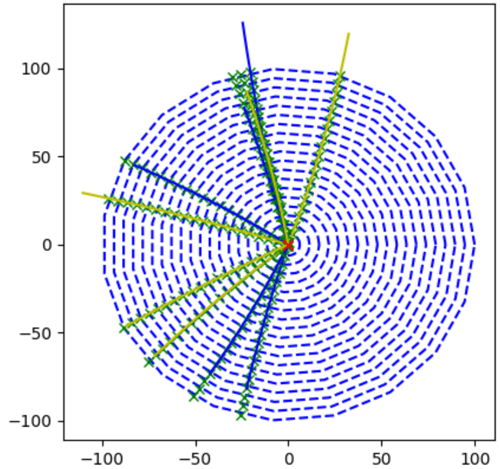
$|\psi_{min}\rangle$

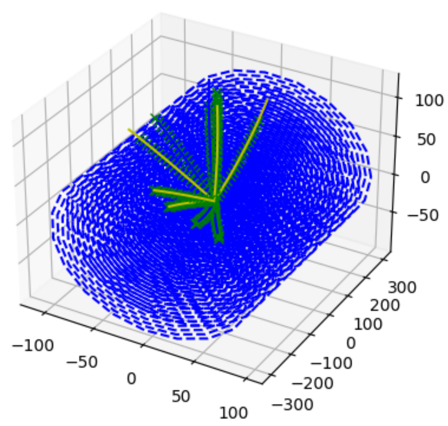
No

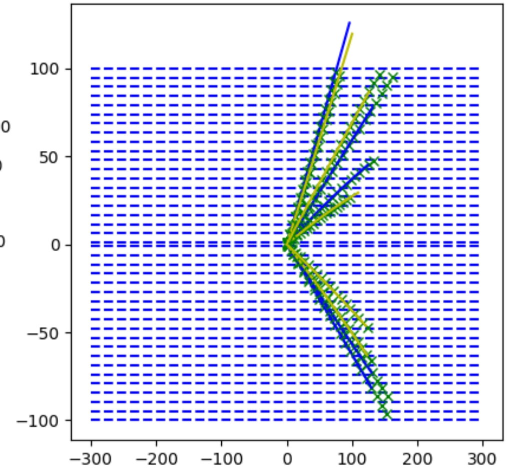
New θ_i parameters

Simulation setup (I)

- In order to test the algorithm a simple tracking system has been implemented
- A tracker with N perfectly circular layers has been considered
  - Hits are calculated from ideal track and smeared to account for resolution
  - Resolution in the rϕ and z coordinates fixed to 200 and 400 microns respectively
  - A track fit is performed and the POCA points estimated
- Two vertices are considered with randomly assigned z position and number of tracks



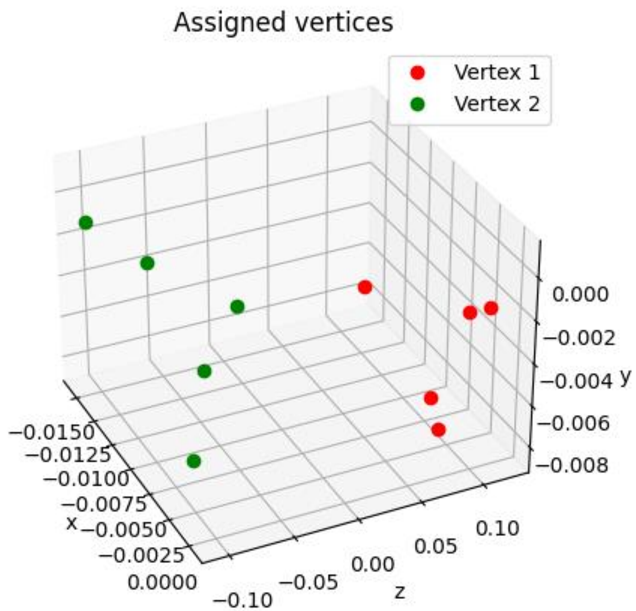




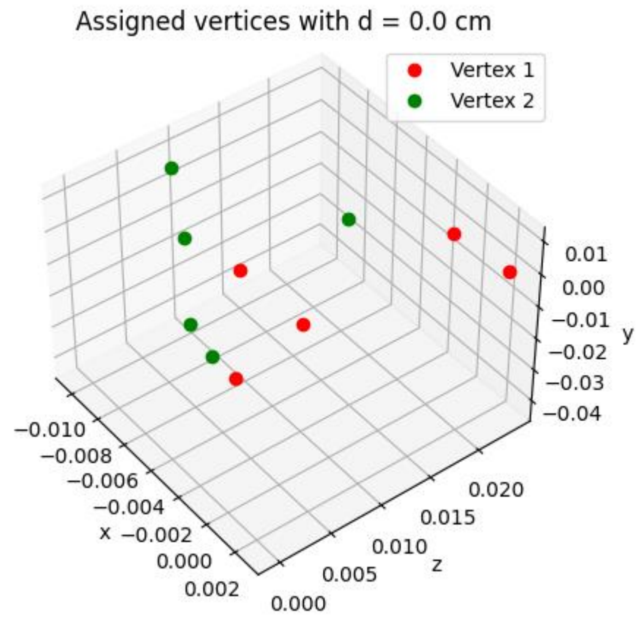
System of two vertices with five tracks each, coloured blue and yellow, meaning different origin vertex

Simulation framework (II)

- Examples of the results of POCA points for different distances between vertices



Distance of 0.1 cm to z = 0



Both vertices in 0.0 cm

Simulation framework (III)

- The VQE algorithm has been implemented and tested using Qiskit
  - The IBM quantum computing programming and simulation framework
- Parameters have been fixed in such a way that there are 2 vertices with 5 tracks each
  - Corresponding to a total of 10 qubits (one per track)
- The COBYLA algorithm has been chosen in order to perform the minimization
- Studies have been run on a GPU 3090 RTX

Figure of merit: efficiency

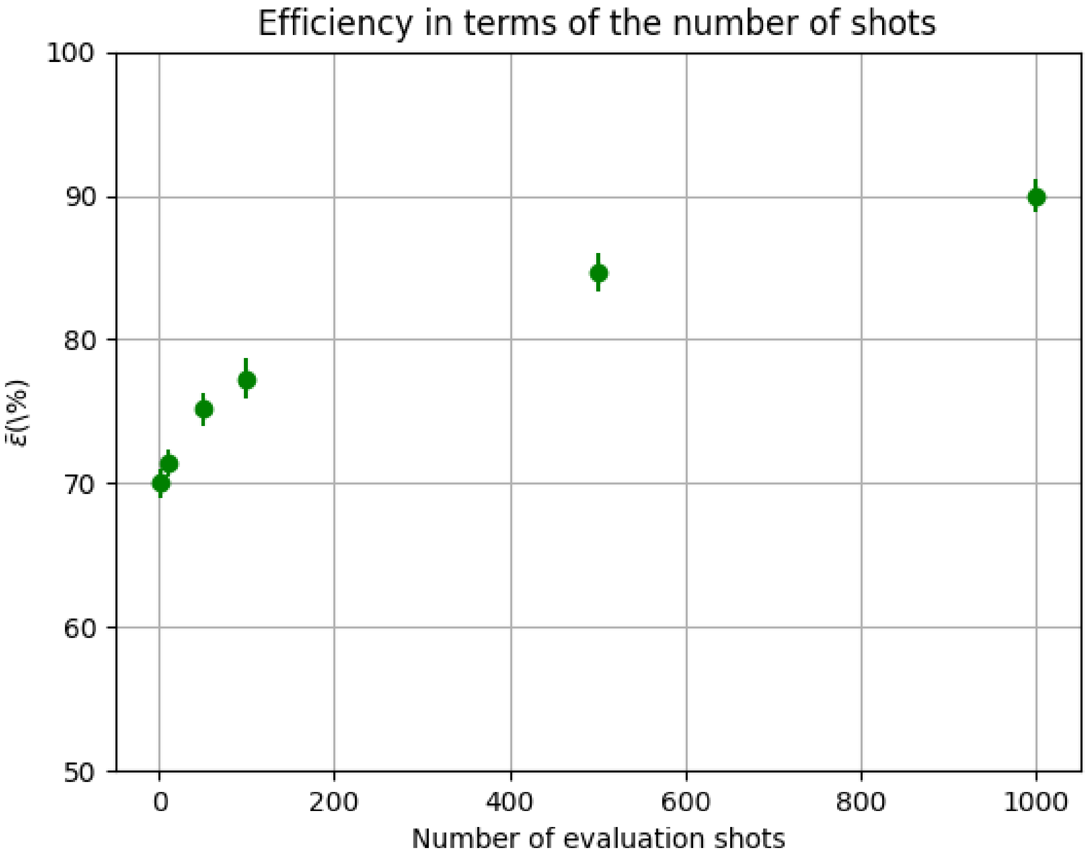
- To test the goodness of the track-vertex association a metric has been defined
- Efficiency: defined as the number of good track-to-vertex associations over number of tracks

$$\varepsilon = \frac{number of good associations}{number of tracks}$$

- It should be noted that this metric takes always values in the range [0.5, 1]
  - It is not possible to correctly assigned less than half of the tracks
- It should also be noted that a random track-vertex association would score 68% for this metric
- This metric has been studied as a function of two important parameters:
  - The distance between the vertices
  - The number of evaluations of the hamiltonian for a given ansatz (number of shots)

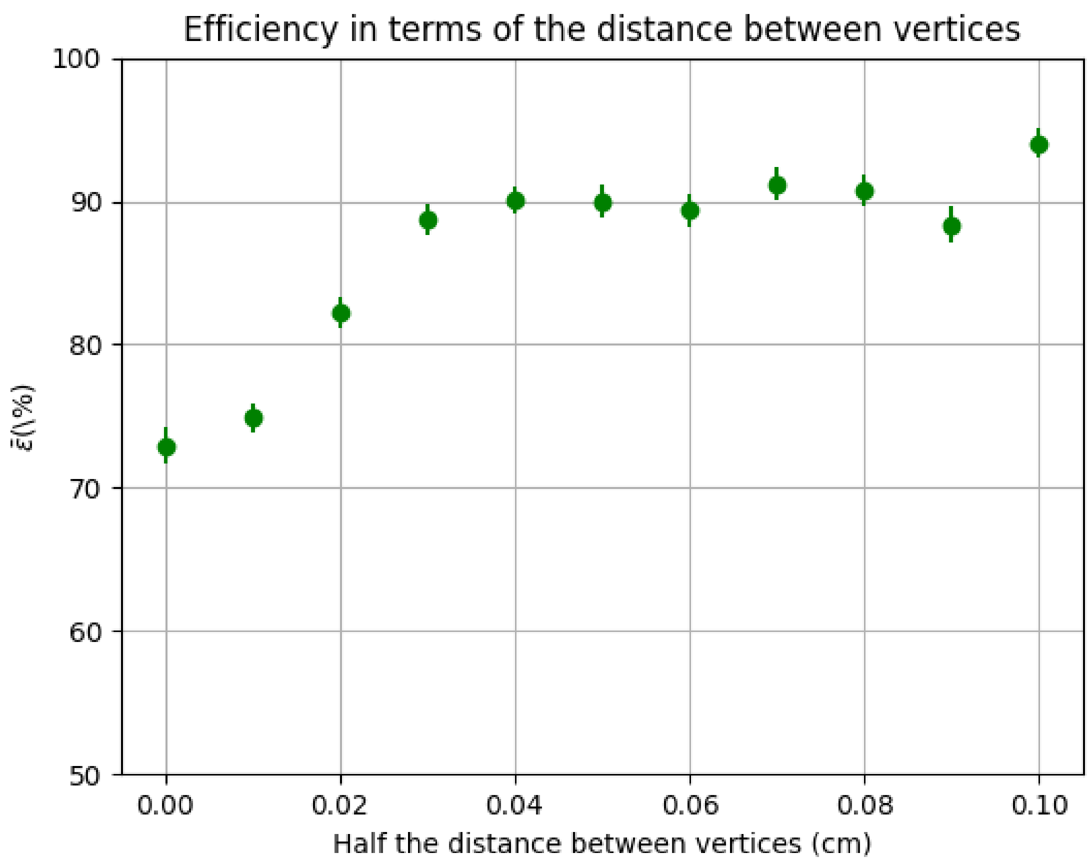
Dependency on the number of shots

- The dependency on the number of shots tested for a vertex distance of X mm
- For 1 shot the efficiency is about equal to a randomly assignment
- For 1000 shots it reaches 90% efficiency in the assignation of the tracks to the vertices



Dependency on the vertex distance

- For coincident vertices the efficiency is very similar to the random assignment
- For 0.03 cm onwards it gets about 90% of efficiency and reaches 94% for 0.1 cm
- The number of evaluations has been fixed to 1000 for this test



Conclusions

- Vertexing will be one of the most challenging problems for the incoming HL-LHC
- A quantum computing algorithm has been developed in order to solve the case of 2 vertices
  - Transforming the vertex association into a Max-Cut problem
  - And then using the VQE algorithm
- The algorithm has been observed to behave with efficiencies higher than 90%
  - For a large number of Hamiltonian evaluations (> 500)
- The algorithm reaches efficiencies of about 90% for separations beyond 200 microns
- Future studies:
  - Fully understand the saturation of the efficiency for large distances
  - Perform time measurements to quantify the speed up of the algorithm wrt a classic one