Quantum circuits synthesis using Householder transformations

Timothée Goubault de Brugière[1,2], Marc Baboulin[1],
Benoît Valiron[1], Cyril Allouche[2]

[1] *LRI and Université Paris-Sud, Orsay, France*
[2] *Atos-Bull, Les Clayes-sous-Bois, France*

September 27, 2018

Outline

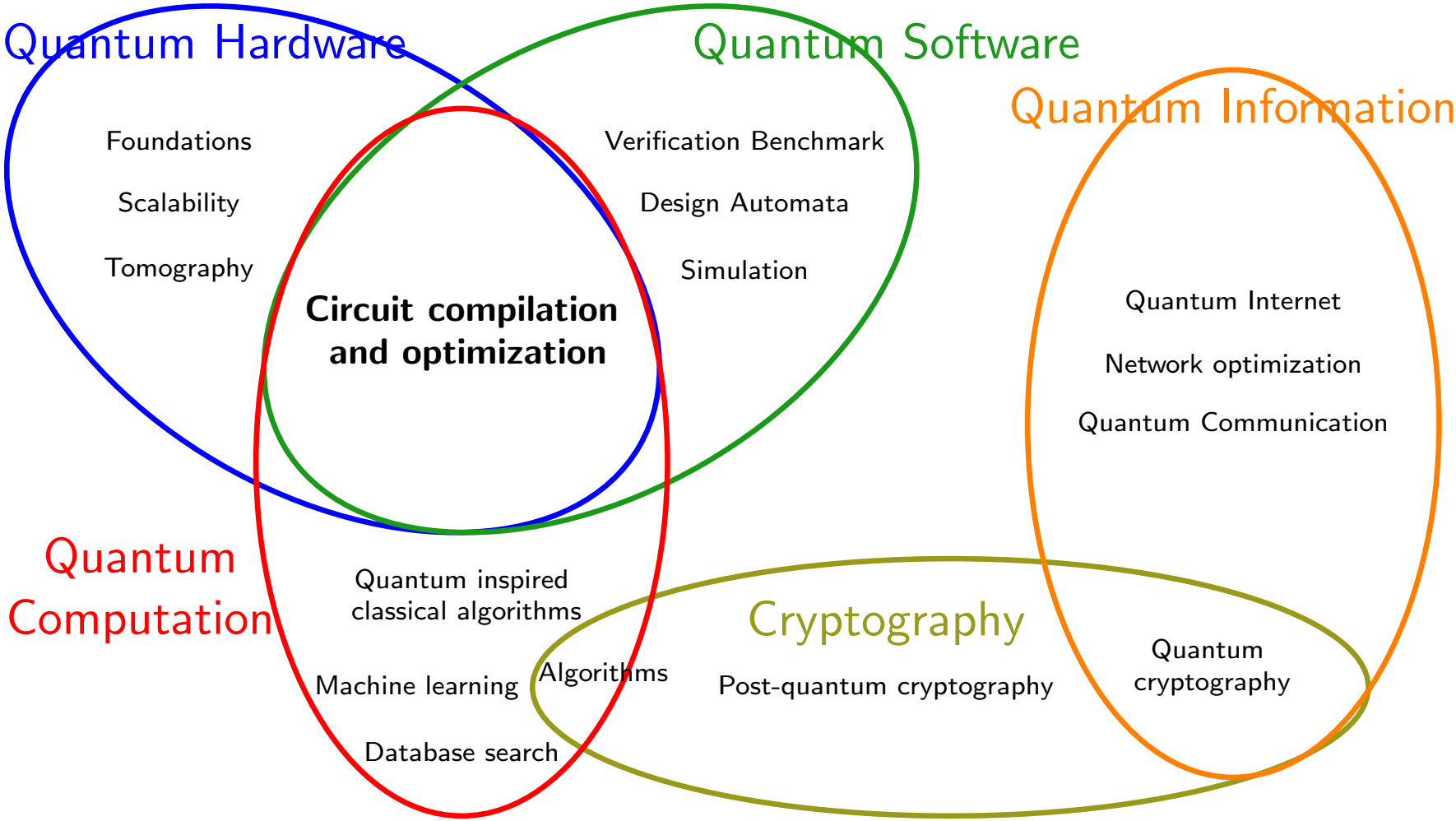
Introduction :

- Quantum computing, quantum circuits
- Existing methods for quantum circuit synthesis

Our contribution :

- QR factorization of quantum operators via Householder transformations
- From QR to circuit synthesis
- Experimental results

Fields of research



Quantum vs Classical computation

- Bit $= \{0, 1\} \rightarrow$ Qubit $\in \mathbb{C}^2$

$$|\psi\rangle = \alpha \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\text{False}} + \beta \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\text{True}}, |\alpha|^2 + |\beta|^2 = 1$$

Quantum vs Classical computation

- Bit $= \{0, 1\} \rightarrow$ Qubit $\in \mathbb{C}^2$

$$|\psi\rangle = \alpha \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\text{False}} + \beta \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\text{True}}, |\alpha|^2 + |\beta|^2 = 1$$

- All computations require linear algebra

Tensor product :

$$\underbrace{|\psi_3\rangle}_{\mathbb{C}^{2^{n+m}}} = \underbrace{|\psi_1\rangle}_{\mathbb{C}^{2^n}} \otimes \underbrace{|\psi_2\rangle}_{\mathbb{C}^{2^m}}$$

Matrix-vector multiplication :

$$|\psi_{t_2}\rangle = U\,|\psi_{t_1}\rangle \iff |\psi_{t_1}\rangle = U^H\,|\psi_{t_2}\rangle$$

$$U \in \mathcal{U}(2^n) = \{A \in \mathcal{M}_{2^n}(\mathbb{C}) | A^H A = I\}$$

Quantum vs Classical computation

- Bit $= \{0, 1\} \rightarrow$ Qubit $\in \mathbb{C}^2$

$$|\psi\rangle = \alpha \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\text{False}} + \beta \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\text{True}}, |\alpha|^2 + |\beta|^2 = 1$$

- All computations require linear algebra

Tensor product :

$$\underbrace{|\psi_3\rangle}_{\mathbb{C}^{2^{n+m}}} = \underbrace{|\psi_1\rangle}_{\mathbb{C}^{2^n}} \otimes \underbrace{|\psi_2\rangle}_{\mathbb{C}^{2^m}}$$

Matrix-vector multiplication :

$$|\psi_{t_2}\rangle = U\,|\psi_{t_1}\rangle \iff |\psi_{t_1}\rangle = U^H\,|\psi_{t_2}\rangle$$

$$U \in \mathcal{U}(2^n) = \{A \in \mathcal{M}_{2^n}(\mathbb{C}) | A^H A = I\}$$

- Examples of quantum operators on 1 qubit :
  - $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} =$ "NOT"
  - $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
  - $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

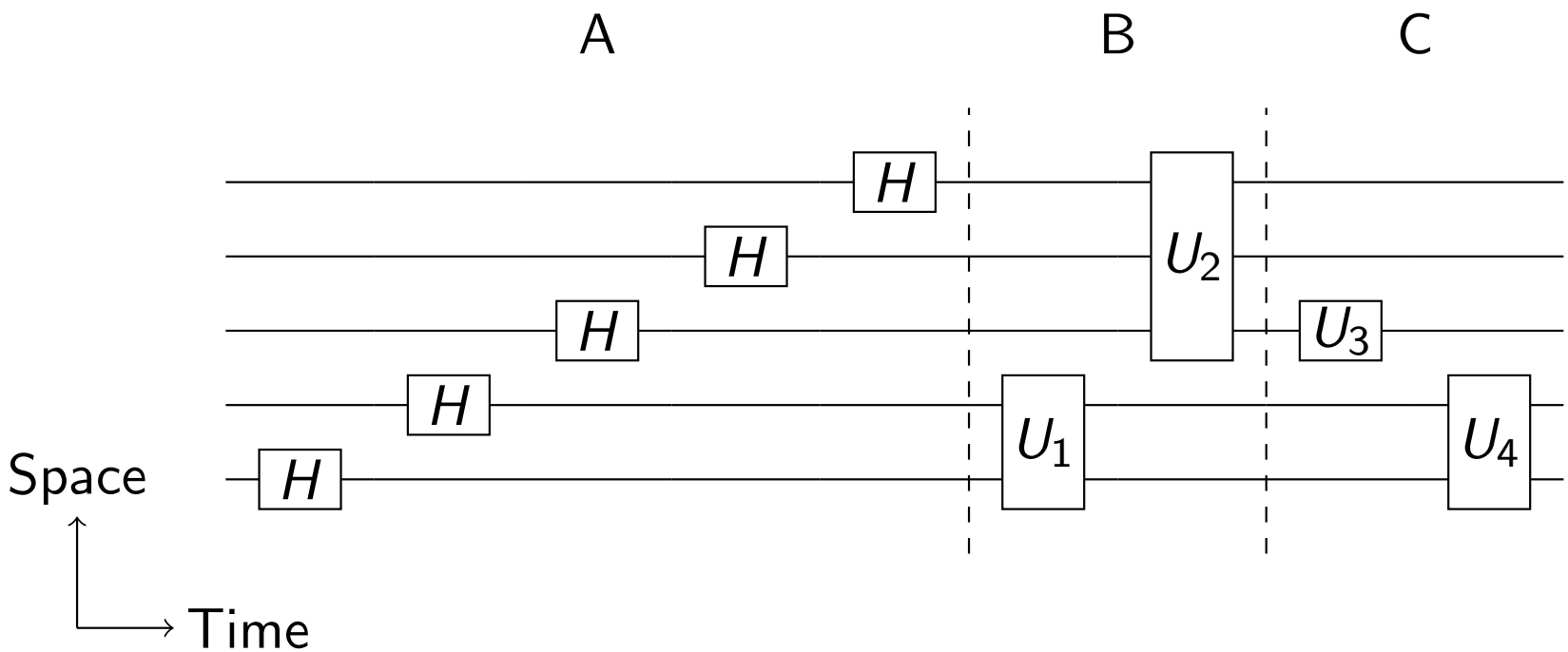
Quantum circuit

Quantum algorithm = quantum circuit = series of quantum gates

Quantum circuit

Quantum algorithm = quantum circuit = series of quantum gates

Space composition $\rightarrow$ tensor product
Time composition $\rightarrow$ matrix multiplication from the left



$$U_{circuit} = \underbrace{(I_4 \otimes U_3 \otimes U_4)}_{C} \underbrace{(U_2 \otimes U_1)}_{B} \underbrace{(H^{\otimes 5})}_{A}$$

Circuit synthesis

Problem : Given an n-qubits unitary operator $U \in \mathcal{U}(2^n)$,
find (using classical computer) a circuit that implements it with some given criteria:

- lowest number of gates,
- specific set of gates (hardware constraints),
- allowed/forbidden extra memory (auxiliary qubits),
- maximum approximation error ($\|U - U_{synth}\|_F \leq \epsilon$),
- **minimal computation time**.

Hard problem [Shende, Bullock, Markov, IEEE, 2006]

$U \in \mathcal{U}(\mathbf{2^n})$: the complexity (in gate count) is exponential in n.
We assume here that U is dense.

Synthesis Algorithm	Number of qubits and gate counts 1	2	3	4	5	6	7	n
Original QR decomp. [3, 10]					———			$O(n^3 4^n)$
Improved QR decomp. [21]					———			$O(n4^n)$
Palindrome transform [2]					———			$O(n4^n)$
QR [33, Table I]	0	4	64	536	4156	22618	108760	$O(4^n)$
CSD [22, p. 4]	0	8	48	224	960	3968	16128	$4^n - 2 \times 2^n$
QSD ($l = 1$)	0	6	36	168	720	2976	12096	$(3/4) \times 4^n - (3/2) \times 2^n$
QSD ($l = 2$)	0	3	24	120	528	2208	9024	$(9/16) \times 4^n - (3/2) \times 2^n$
QSD ($l = 2$, optimized)	0	**3**	**20**	**100**	**444**	**1868**	**7660**	$(23/48) \times 4^n - (3/2) \times 2^n + 4/3$
Lower bounds [28, 29]	0	3	14	61	252	1020	4091	$\lceil \frac{1}{4}(4^n - 3n - 1) \rceil$

Not only the number of gates matters but also the time and flops to obtain the circuit.

Existing method: Quantum Shannon Decomposition

Relies on the *Cosine-Sine Decomposition* (CSD) :

$$U = \begin{pmatrix} A_1 & \\ & B_1 \end{pmatrix} \begin{pmatrix} C & -S \\ S & C \end{pmatrix} \begin{pmatrix} A_2 & \\ & B_2 \end{pmatrix}, C^2 + S^2 = I$$

A_1, A_2, B_1, B_2 are unitaries on $n-1$ qubits, C, S are diagonal.

- gives the shortest circuits : $\frac{23}{48} \times 4^n$ gates,
- most considered method today,
- CSD is expensive ($2 \times N^3$ flops),
- $\simeq 7 \times N^3 = 7 \times 8^n$ total flops needed ($N = 2^n$).

We propose a method based on QR decomposition

How can we use QR for quantum circuit synthesis ?

- U is unitary: R is diagonal
  $\Longrightarrow$ the process is a progressive diagonalization of U.
- not all Q and R are of interest ($Q = U$, $R = I$ works)
  $\Longrightarrow$ we need a series of elementary transformations,
  $\Longrightarrow$ each of them will be synthesized.

Examples :

- Givens rotations: gives very large circuits.
- Householder transformation: so far, only as a theoretical work.
  QR via Householder is numerically stable and rich in BLAS3.

General objective

$\Longrightarrow$ What are the properties of a quantum circuit synthesis framework using Householder transformations ?

Two metrics :

- quantum resources (= number of gates)
- classical resources (= number of flops and flop/s)

The algorithm - General presentation

$$A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \xrightarrow{H_1 A} \begin{pmatrix} e^{i\theta_1} & 0 & 0 & 0 \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix} \xrightarrow{H_2 H_1 A} \begin{pmatrix} e^{i\theta_1} & 0 & 0 & 0 \\ 0 & e^{i\theta_2} & 0 & 0 \\ 0 & 0 & * & * \\ 0 & 0 & * & * \end{pmatrix}$$

$$\xrightarrow{H_3 H_2 H_1 A} \begin{pmatrix} e^{i\theta_1} & 0 & 0 & 0 \\ 0 & e^{i\theta_2} & 0 & 0 \\ 0 & 0 & e^{i\theta_3} & 0 \\ 0 & 0 & 0 & e^{i\theta_4} \end{pmatrix}$$

Two steps, :

1) choice of the Householder vector v_i ($H_i = I - \frac{2}{\|v_i\|^2} v_i v_i^H$)

2) update of the rest of the matrix

The algorithm - updating the matrix

$$A = \left(\begin{array}{c|c} a_{11} & r \\ \hline c & A'' \end{array} \right)$$

(v_1 : first column of A; A' : remaining columns)

Standard update rule :

$$HA' = A' - \frac{2}{\|v_1\|^2} v_1 \times (A'^H v_1)^H$$

New update rule (using orthonormality of the columns of A) :

$$HA'' = A'' - c \times r$$

Overall :

- the matrix/vector product is replaced by a rank-one update,
- twice less flops is needed : $\frac{2}{3} \times N^3 = \frac{2}{3} \times 8^n$ (QSD: 7×8^n)

Yet this update involves BLAS 2 operations.
$\rightarrow$ any block version to use BLAS 3 ?

The algorithm - block version

$$A = \begin{array}{c} \overset{k}{\longleftrightarrow} \\ \left[\begin{array}{c|c} A_1 & A_3 \\ \hline A_2 & A_4 \end{array}\right] \end{array} \quad \begin{array}{l} \\ \updownarrow k \\ \\ \end{array}$$

Once A_1 is factorized :

- there exist $k \times k$ triangular matrices T_2, T_3 *only depending on the factorized matrix* A_1 such that

$$A_2 \leftarrow T_2 \times A_2,$$

$$A_3 \leftarrow A_3 \times T_3,$$

- with the new updated matrices A_2, A_3 :

$$A_4 \leftarrow A_4 - A_3 \times A_2.$$

The algorithm - pseudo code

Algorithm 1 Householder factorization of a unitary matrix A

Require: $N \geq 0$, A$\in \mathcal{U}_N$
Ensure: $A = QR$
  {NX determines when to switch from blocked to unblocked code}
  {NB is the block size}
  for $I = 1, NX, NB$ **do**
    $IB \leftarrow MIN(N - I + 1, NB)$
    call ZUNQR2(IB, IB, A(I, I), TAU(I))
    $T_1, T_2 \leftarrow$ ZLARFT2(N, IB, A(I, I), TAU(I))
    update $A(I : N, I : I + IB)$ via a call to ZTRMM
    if $I + IB \leq N$ **then**
      update $A(I : I + IB, I : N)$ via a call to ZTRMM
      update $A(I + IB : N, I + IB : N)$ via a call to ZGEMM
    end if
  end for
  if $I \leq N$ **then**
    call ZUNQR2(N-I+1, N-I+1, A(I, I), TAU(I))
  end if

Producing a quantum circuit

Step I :

$$U = \prod_{i=1}^{2^n} H_i \times D$$

with D diagonal gate.

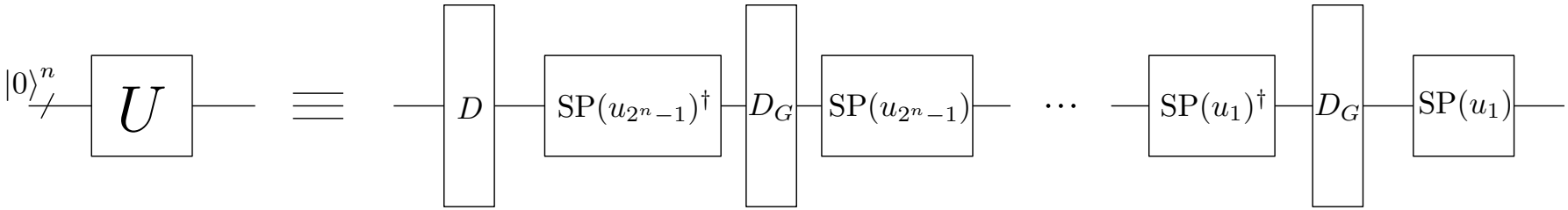
Step II :

$$H_i = I - \frac{2}{\|v_i\|^2} v_i \times v_i^H = SP(v_i) \times D_G \times SP(v_i)^H$$

where :

- $D_G = \begin{pmatrix} -1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$ (also called Grover shift operator),
- $SP(v_i) = \begin{pmatrix} v_i & | & * \end{pmatrix}$ prepares the Householder vector $v_i/\|v_i\|$.

Producing a quantum circuit



Step III :

$$v_i = \begin{pmatrix} e^{i\theta_1}\rho_1 \\ \vdots \\ e^{i\theta_{2^n}}\rho_{2^n} \end{pmatrix} = \underbrace{\begin{pmatrix} e^{i\theta_1} & & \\ & \ddots & \\ & & e^{i\theta_{2^n}} \end{pmatrix}}_{\text{diagonal gate}} \underbrace{\begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{2^n} \end{pmatrix}}_{\text{real state preparation}}$$

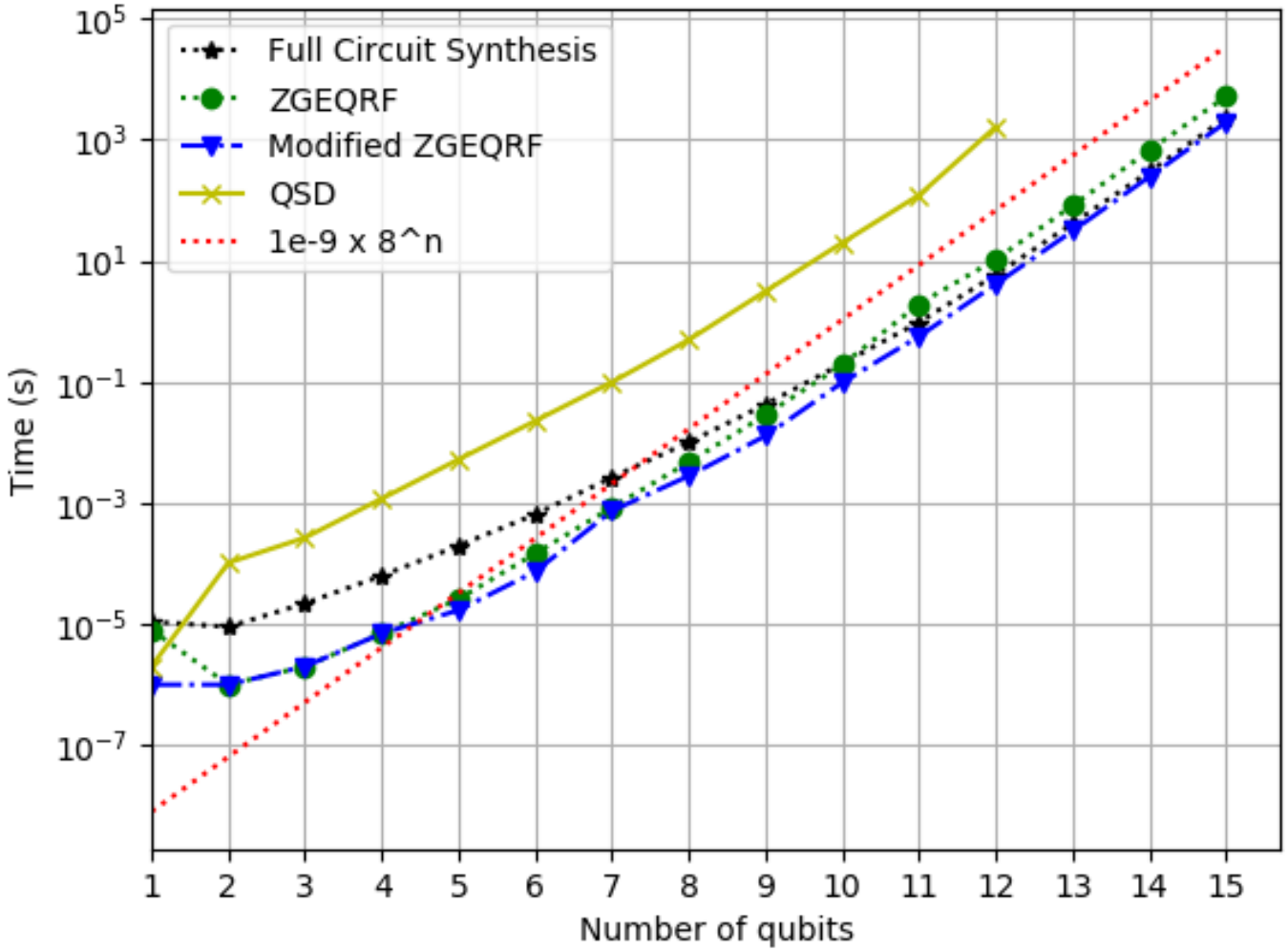
Gate and flop count for synthesis only :

Gates : 2×4^n (QSD: $23/48 \times 4^n$) flops : $O(4^n)$

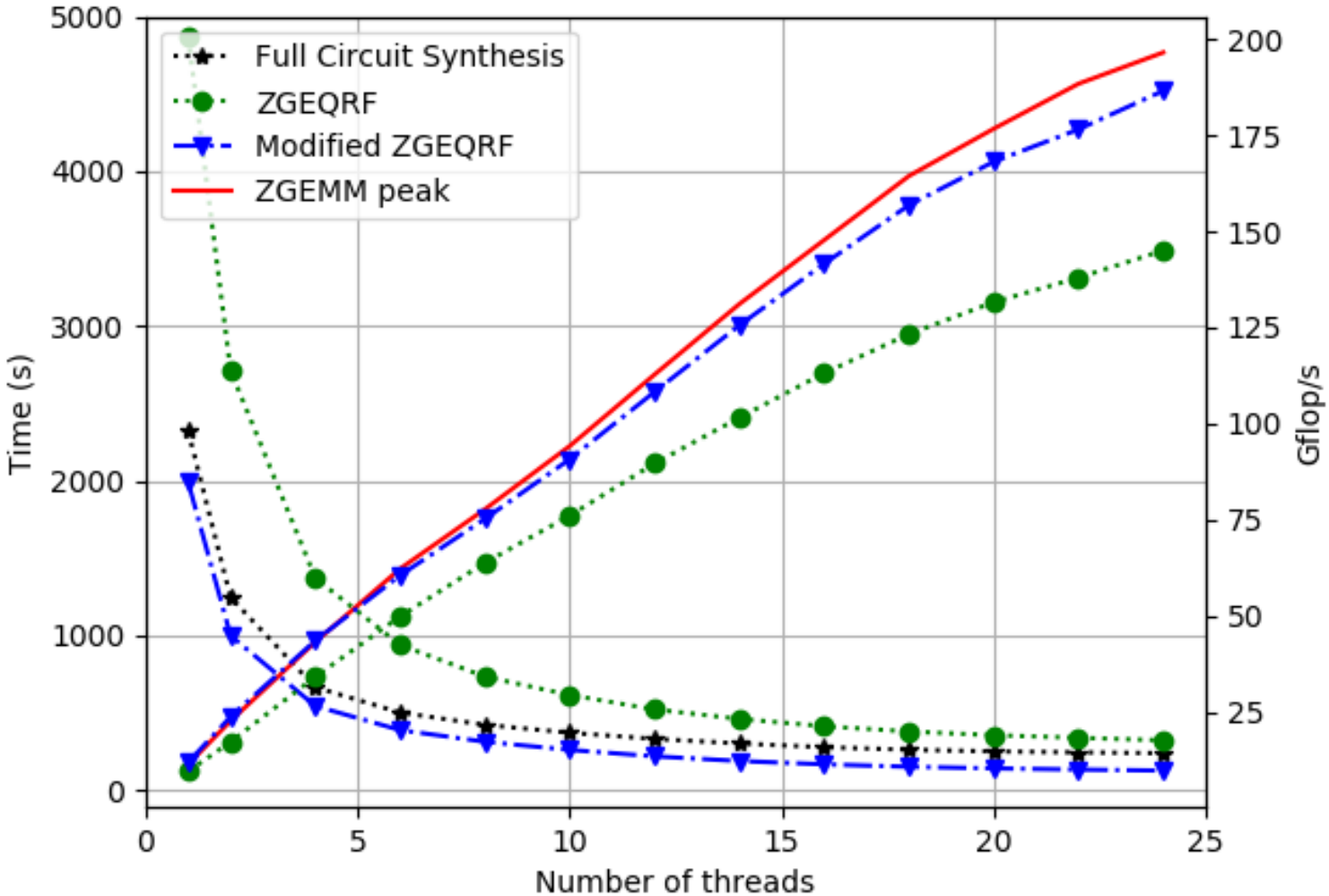
Experimental context

- We start from the original LAPACK routine ZGEQRF.
  $\rightarrow$ we modified it according to our new algorithm
- Test matrices are random unitary (Haar measure).
- Linked with the MKL multithreaded BLAS (hyper-threading disabled).
- Experiments achieved using a 24-core Intel Xeon(R) E7-8890 v4 processor at 2.4 GHz 24-cores.
- We evaluate our modified QR and the resulting full circuit synthesis.

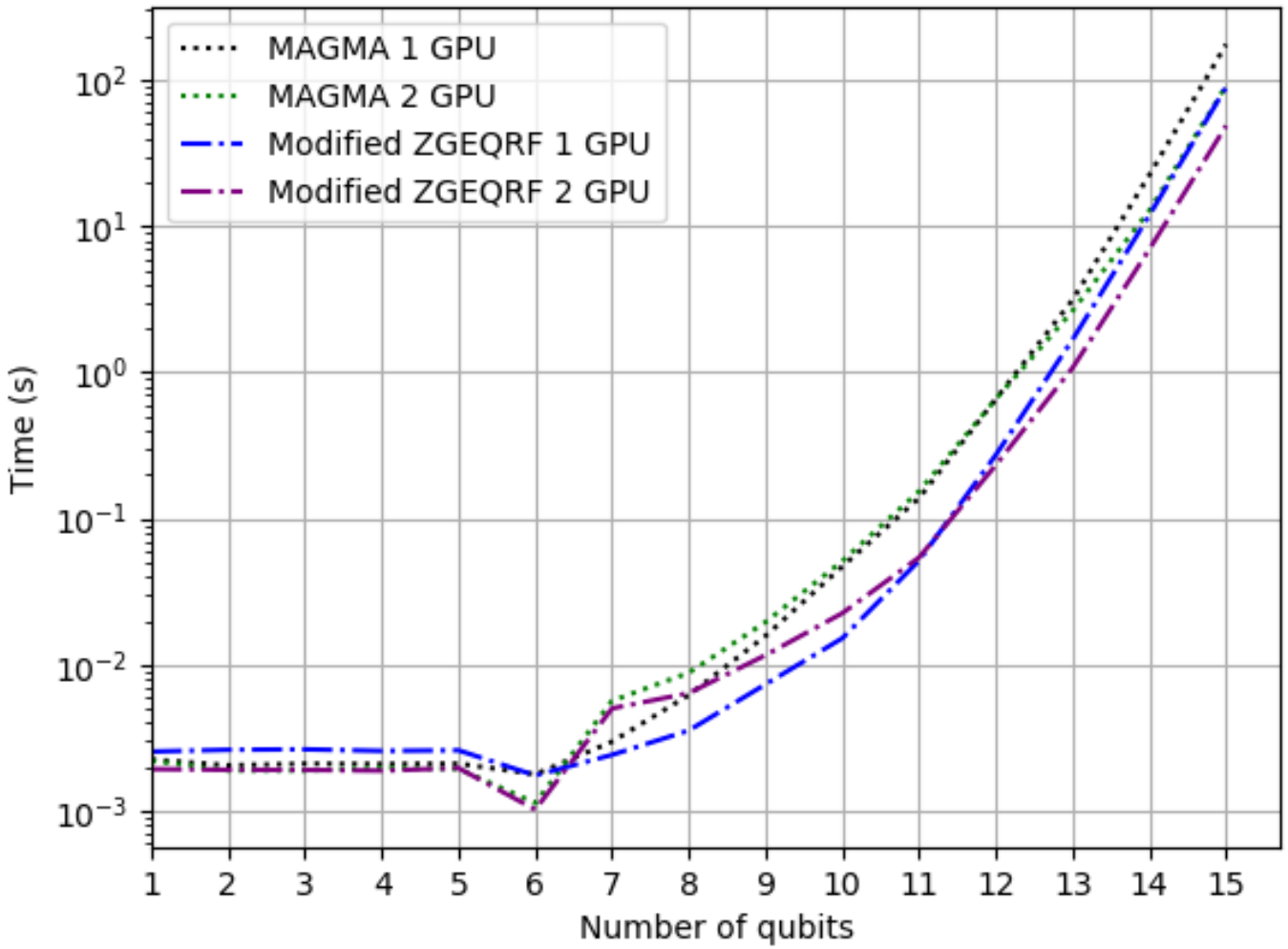
Experimental results - sequential



Experimental results - strong scaling



Preliminary results - 1 and 2 GPU (Kepler)



Conclusion and future work

During this talk :

- we introduced the main issues of quantum circuit synthesis,
- we adapted the classical QR factorization to unitary operators,
- we decreased the computational time:
  - ⋆ by a factor of 2 with the standard QR,
  - ⋆ by a factor of 100 with the QSD (sequential),
- we provided an efficient procedure to obtain a quantum circuit.

Future work :

- using more GPUs to address larger problems,
- special cases : sparse unitary matrices,
- tradeoff classical/quantum resources.