

Contracting Tensor Networks from Clifford Circuits Using GF(2) Gaussian Elimination

Edgar Solomonik and Yuchen Pang

Siebel School of Computing and Data Sciences
University of Illinois, Urbana-Champaign

SIAM Annual Meeting

Montreal, Canada

July 31, 2025

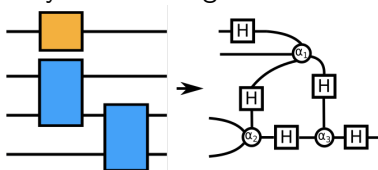
Linear Systems over $\text{GF}(2)$ and Clifford States

- The Clifford group of gates (generated by Hadamard, phase, and CNOT) leads to circuits that may be simulated classically (Gottesman-Knill theorem)
- Each stabilizer state corresponds to a quadratic form, and Clifford circuits are often represented using a tableau, which encodes their action on a set of elementary states
- Any stabilizer state of n -qubits can be described by an undirected graph G and some 1-qubit unitaries
- Gosset, Grier, Kerzner, Schaeffer (Quantum 2024), show that
 - n -qubit stabilizer states can be simulated in $O(n\tau^{\omega-1})$ time where τ is the treewidth of G
 - linear systems over $\text{GF}(2)$ can be solved in the same complexity using their algorithm
- We seek to show the other direction
 - simulation of Clifford circuits reduces to solving a linear system
 - sparse linear systems can be solved in time $O(n\tau^{\omega-1})$

ZX Calculus

A Clifford circuit consists only of Hadamard, phase, and CZ gates

- ZX-calculus provides a way to reduce any circuit with m gates into a tensor network consisting of only $2 \times \cdots \times 2$ tensors known as Z-spiders, connected by Hadamard gates.



- With 0-based indices $i_j \in \{0, 1\}$, the Z-spider (α_i nodes in diagram) is defined as

$$z_{i_1 \dots i_m} = i_1 \cdots i_m e^{i\alpha} + (1 - i_1) \cdots (1 - i_m)$$

- If $\alpha = j\pi/2$, $j \in \{0, 1, 2, 3\}$, “the tensor is Clifford”
- Further, a ZX diagram may be written as a local unitary multiplied by a graph state for a graph $G = (V, E)$, $|G\rangle = \prod_{ij \in E} U_{CZ}^{\langle i, j \rangle} |+\rangle^{\otimes |V|}$

Vertex and Edge Complementation

- We say two $|\psi\rangle$ and $|\phi\rangle$ are locally equivalent, $|\psi\rangle \equiv |\phi\rangle$, if $\exists U = U_1 \otimes \cdots \otimes U_n$, $U_1, \dots, U_n$ Clifford, $U|\psi\rangle = |\phi\rangle$
- $O(G) = \{G' : |G\rangle \equiv |G'\rangle\}$ is generated by basic complementations
- Vertex complementation on vertex v with neighborhood $N(v)$ is

$$|\tau_v(G)\rangle = R_x^{\langle v \rangle}(\pi/2) \prod_{u \in N(v)} R_z^{\langle u \rangle}(-\pi/2) |G\rangle$$

- With $G = (V, E)$, $\tau_v(G) = (V, E')$, we have

$$\forall i, j \in N(v), (i, j) \in E' \text{ iff } (i, j) \notin E$$

- Edge complementation on edge (u, v) is

$$|\epsilon_{uv}(G)\rangle = H^{\langle u \rangle} H^{\langle v \rangle} \prod_{w \in N(v) \cap N(u)} Z^{\langle w \rangle} |G\rangle$$

- With $G = (V, E)$, $\epsilon_{uv}(G) = (V, E')$, we have

$$\begin{aligned} \forall i \in N(u), j \in N(v), (i, j) \in E' \text{ iff } (i, j) \notin E, \\ N'(u) = N(v), N'(v) = N(u) \end{aligned}$$

A New Matrix Representation for Clifford States

- Given G and $y \in \mathbb{F}_2^n$, an XY-basis graph state is

$$|G, y\rangle = \left(\bigotimes_{i=1}^n H(S^\dagger)^{y_i} \right) |G\rangle$$

- Let the adjacency matrix of G be A_G , then we consider the matrix

$$A = A_G + \text{diag}(y) \text{ and denote } |A\rangle = |G, y\rangle$$

- A circuit with m Clifford gates can be reduced to an XY-basis state with $O(m)$ vertices (qubits)
- Strong simulation (computing output amplitudes) reduces to computation of amplitudes of an XY-basis state
- Weak simulation (sampling) corresponds to sampling XY-basis state with all but n qubits projected

Strong Simulation of XY-basis States

- The amplitudes of a graph state G are described by the well-known formula

$$\langle x|G\rangle = \frac{1}{2^{n/2}}(-1)^{x^T A_G x/2}$$

- For an XY-basis state, given A with inverse $A^\#$ ($AA^\# = I \pmod{2}$), after performing local complementations to remove the Hadamards, we obtain a new formula

$$\langle x|A\rangle = \frac{1}{2^{n/2}} i^{x^T A^\# x}$$

- The above follows since vertex and edge complementations correspond to Gauss-Jordan elimination

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \Rightarrow \begin{bmatrix} A_{11}^{-1} & A_{11}^{-1} A_{12} \\ A_{21} A_{11}^{-1} & A_{22} - A_{21} A_{11}^{-1} A_{12} \end{bmatrix}$$

Strong Simulation of Low-Rank XY-basis States

- When A is rank r , we may assume that A is ordered so

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{21}A_{11}^{\#}A_{12} \end{bmatrix}$$

where $A_{11} \in \mathbb{F}_2^{r \times r}$

- We have that, $\exists u \in \mathbb{F}_2^n, v, w \in \mathbb{F}_2^r$ such that $\forall x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\langle x|A \rangle$ is nonzero iff $x + u \in \text{span}(A)$, in which case

$$\langle x|A \rangle = \frac{1}{2^{r/2}} i^{x_1^T A_{11}^{\#} x_1 + (v+2w)^T x_1}$$

- Hence, to obtain amplitudes and sample $|A\rangle$, we need matrix-vector products with $A_{11}^{\#}$ (the same is needed for obtaining u, v, w)

Complexity of Gaussian Elimination

- Strassen's 1969 paper shows that Cholesky (LU without pivoting) can be done with cost $O(n^{\log_2 7})$
- In 1974, Bunch and Hopcroft proposed an algorithm with $O(n^\omega)$ complexity for LU with pivoting for square nonsingular matrices
 - They also note a suggestion by Schönhage to instead solve linear use the normal equations with Cholesky
- In 1981, Ibarra, Moran, and Hui provided a simple recursive algorithm for LU of nonsquare matrices
- For sparse Cholesky, the maximum number τ of off-diagonal nonzeros in the L factor for a sparse matrix corresponding to a graph with treewidth τ , allows for algorithms with complexity $O(n\tau^{\omega-1})$
- What about symmetric indefinite systems of equations?

LDL Factorization

- The LDL decomposition of a symmetric matrix $A \in \mathbb{F}^{n \times n}$ is

$$P^T A P = L D L^H$$

where L is lower triangular and unit-diagonal, D is block diagonal with blocks that are either 1×1 or 2×2 and antidiagonal.

- If $A = A^H$, we have $D = D^H$, while for $A^H = -A$, we would require $D^H = -D$.
- The LDL decomposition can hence preserve a variety of symmetries
 - a complex Hermitian or skew-Hermitian matrix
 - a real symmetric or skew-symmetric matrix
 - a symmetric matrix over $\mathbb{F}_2$

LDL Factorization from Elementary Eliminations

- Vertex elimination, if $a_{11} \neq 0$ compute

$$\begin{bmatrix} 0 & 0 \\ 0 & S \end{bmatrix} = A - a_{:,1} a_{11}^{-1} a_{:,1}^H$$

where S is the Schur complement of a_{11} in A

- Each successive vertex elimination defines a successive column of L ($a_{:,1} a_{11}^{-1}$) and a diagonal entry in D (a_{11})
- A Cholesky factorization may be formed from $n - 1$ vertex eliminations for any complex Hermitian positive definite matrix
- In LDL, if $a_{21} \neq 0$, and $a_{11} = 0$ or $a_{22} = 0$ an edge elimination on $(1, 2)$ is

$$\begin{bmatrix} 0 & 0 \\ 0 & S \end{bmatrix} = A - \begin{bmatrix} a_{:,1} & a_{:,2} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} a_{:,1} & a_{:,2} \end{bmatrix}^H$$

Gaussian Elimination and Graph Treewidth

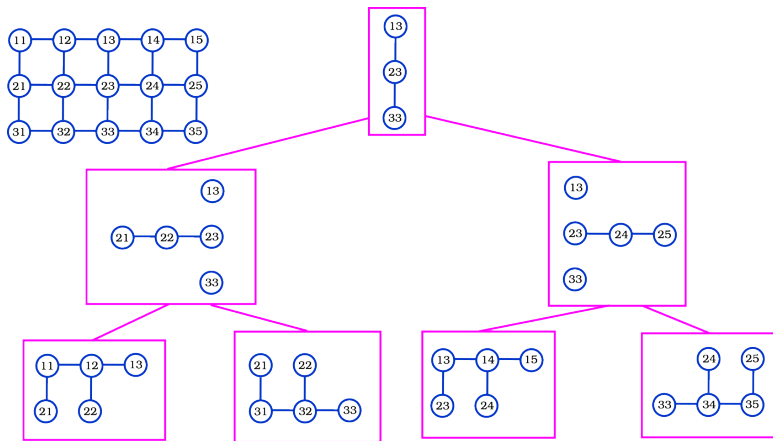
- The complexity of Cholesky of a sparse matrix without use of fast matrix multiplication is bounded by

$$O\left(\sum_{i=1}^n \text{nnz}(\ell_i)^2\right)$$

where ℓ_i is the i th column in the L factor

- The elimination width of a graph is the maximum value of any $\text{nnz}(\ell_i)$ given by the best choice of elimination ordering in Cholesky of an SPD matrix
- The elimination width is also the treewidth of a graph
- The treewidth is the least maximum bag size among possible tree decompositions of a graph
- A tree decomposition of G specifies a set of bags (subsets of vertices of G) and a tree over these bags, such that
 - 1 each vertex is assigned to at least one bag
 - 2 the set of bags to which any vertex is assigned is connected in the tree
 - 3 the end-points of any edge in G are contained within some bag

Tree Decomposition by Nested Dissection



Sparse Matrix Factorization

- Assume wlog that the tree decomposition is a full binary tree with $O(n/\tau)$ bags, each of size $O(\tau)$
- Each vertex in the graph is assigned to a subtree of bags
- Order matrix rows in columns in reverse DFS order (starting from some leaf bag) based on the roots of the subtrees to which the corresponding graph vertices are assigned
- Then, with respect to the first leaf bag $\mathcal{B}$ in the order, we have

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where $A_{11} \in \mathbb{F}^{|\mathcal{B}| \times |\mathcal{B}|}$

- If the treewidth is τ , A_{21} has at most τ nonzero rows
- If A is SPD, A_{11} is SPD, and hence we may just eliminate the block and compute the Schur complement
- What if $A_{11} = 0$? We have a saddle point system.

Γ -elimination

- Consider

$$M = \begin{bmatrix} A & B^H \\ B & 0 \end{bmatrix},$$

- We may perform edge eliminations on any nonzero in B to preserve the structure
- Permute M so that this nonzero is b_{11} and perform

$$\begin{aligned} M' &= M - \begin{bmatrix} a_{:,1} & b_{1,:}^H \\ b_{:,1} & 0 \end{bmatrix} \begin{bmatrix} a_{11} & b_{11}^* \\ b_{11} & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_{:,1} & b_{1,:}^H \\ b_{:,1} & 0 \end{bmatrix}^H \\ &= M - \begin{bmatrix} a_{:,1} & b_{1,:}^H \\ b_{:,1} & 0 \end{bmatrix} \begin{bmatrix} 0 & b_{11}^{-1} \\ (b_{11}^*)^{-1} & -a_{11}^*(b_{11}^*b_{11})^{-1} \end{bmatrix} \begin{bmatrix} a_{:,1} & b_{1,:}^H \\ b_{:,1} & 0 \end{bmatrix}^H \\ &= \begin{bmatrix} A' & B' \\ B' & 0 \end{bmatrix}, \quad \text{where} \end{aligned}$$

$$A' = A - a_{:,1}b_{11}^{-1}b_{:,1} - b_{:,1}^H(b_{11}^*)^{-1}a_{:,1}^* + b_{1,:}^H(b_{11}^*b_{11})^{-1}a_{11}b_{:,1}$$

$$B' = B - b_{:,1}(b_{11}^*)^{-1}b_{1,:}^H$$

Γ -elimination via LU

- A sequence of $\text{rank}(B)$ Γ -elimination steps computes an LU factorization of B
- Can we go the other way? Yes, given $PB^H Q^T = LU$, we may obtain the corresponding $2\text{rank}(B)$ columns of the LDL factor of M via triangular solves

Sparse LDL

Consider the block form describing a pair of leaves in the tree decomposition

$$\begin{bmatrix} A_{11} & 0 & A_{13} & A_{14} \\ 0 & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

- We may eliminate nonzeros in A_{11} and A_{22} until the blocks are zero
- Assuming $A_{11} = A_{22} = 0$, we have a saddle point system
- We may not be ready to perform eliminations of nonzeros in

$$B = \begin{bmatrix} A_{31} & A_{32} \\ A_{41} & A_{42} \end{bmatrix}$$

since the corresponding rows may be assigned to many bags in the tree decomposition

- However, we know the number of nonzero rows in B is at most τ
- If the number of nonzero nonzero columns in B is greater than τ , B and A are rank deficient

Peeling

- To resolve issues in the low-rank case, we need an implicit LDL factorization, defined based on a primitive besides vertex/edge elimination

Definition (Vertex Peeling)

Consider $A \in \mathbb{F}^{(n+m+1) \times (n+m+1)}$ and an associated graph G , where

$$A = \begin{bmatrix} A_{11} & A_{21}^H & a_{31}^H \\ A_{21} & 0 & 0 \\ a_{31} & 0 & 0 \end{bmatrix}, A_{21} \in \mathbb{F}^{m \times n}.$$

If $\exists x \in \mathbb{F}^m$, such that $A_{21}^H x = a_{31}^H$, peeling the vertex in G corresponding the the last row/column of A is defined by the factorization,

$$A = \begin{bmatrix} I & 0 \\ 0 & I \\ 0 & x^H \end{bmatrix} \tilde{A} \begin{bmatrix} I & 0 \\ 0 & I \\ 0 & x^H \end{bmatrix}^H, \tilde{A} = \begin{bmatrix} A_{11} & A_{21}^H \\ A_{21} & 0 \end{bmatrix}.$$

Full Sparse LDL Algorithm

We now have all the machinery to state how sparse LDL should work on each tree node bag $\mathcal{B}$ with parent bag $\mathcal{P}$

- 1 Recursively eliminate or peel all vertices assigned only to bags in the subtrees of the children of $\mathcal{B}$, both children return a set of uneliminated vertices connected only to vertices in $\mathcal{B}$
- 2 Perform peeling to remove any linearly dependent constraints among those returned by the children
- 3 Consider the set of graph nodes in $\mathcal{B} \setminus \mathcal{P}$ and eliminate as many as possible with vertex and edge eliminations
- 4 Perform peeling on the resulting saddle point system and return any remaining linearly independent constraints connected to vertices in $\mathcal{P}$

Complexity of Full Sparse LDL Algorithm

- The new sparse LDL algorithm consists of a sequence of $O(n/\tau)$ of lower triangular transformations $L_i W_i$, where L_i are eliminations and W_i are peeling transformations, both acting on $O(\tau)$ rows
- Hence, we may apply the inverse of L by reversing the order of these transformations.
- The eliminations $L_1, \dots, L_k$ could be concatenated to obtain the L factor of the leading $\text{rank}(A) \times \text{rank}(A)$ block of $P^T A P$
- The peeling transformations need to be multiplied by parts of L to obtain the full L factor, and we have no bound on the resulting fill/complexity

Sparse LU

Note that an edge complementation on the matrix,

$$A = \begin{bmatrix} 0 & -B^H \\ B & 0 \end{bmatrix},$$

corresponds to elementary lower/upper-triangular transformations (matrices of the form $L_i = I + C$, where C is nonzero in only one column and is strictly lower/upper-triangular) applied to both sides of B . Similarly, peeling a vertex in A corresponds to peeling a row of B .

Lemma

Given a matrix $A \in \mathbb{F}^{(n+m) \times (n+m)}$ of the form,

$$A = \begin{bmatrix} 0 & -B^H \\ B & 0 \end{bmatrix},$$

a peeled implicit LDL decomposition of A yields a peeled implicit LU factorization of B .

Conclusion and Open Problems

Summary of contributions

- new algorithms for sparse LDL and LU (over any field) of square full rank matrices that match complexity of sparse Cholesky
 - <https://arxiv.org/abs/2504.20305>
- explicit formula for amplitudes of states obtained from Clifford circuits based on matrix inversion in $\text{GF}(2)$
 - In collaboration with Yuchen Pang, manuscript in preparation

Open problems

- algorithm for explicit low rank sparse LDL or nonsquare LU
- complexity lower bounds for explicit low rank sparse LDL
- experimental evaluation of sparse factorization algorithm
- additional simplifications of amplitude formula (form of u, v, w), applications