Numerical Methods for Quantum Computing

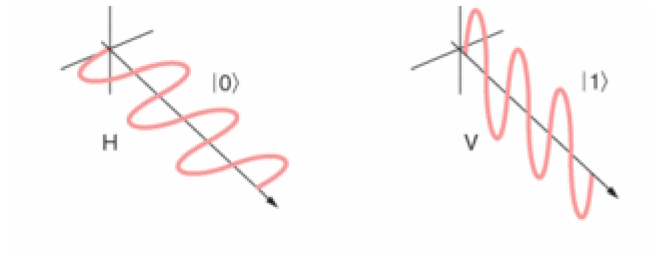
Thomas Huckle

Computer Science / Mathematics

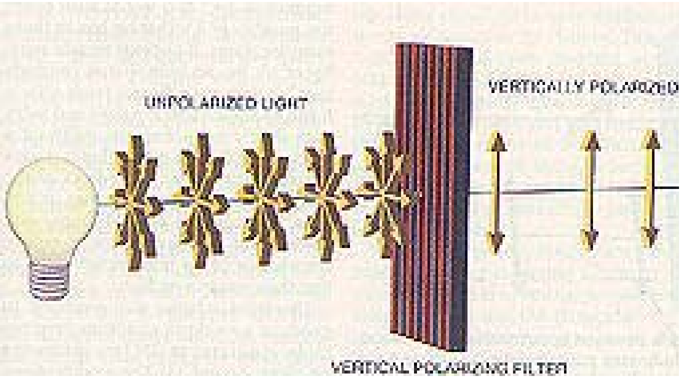
Technische Universität München

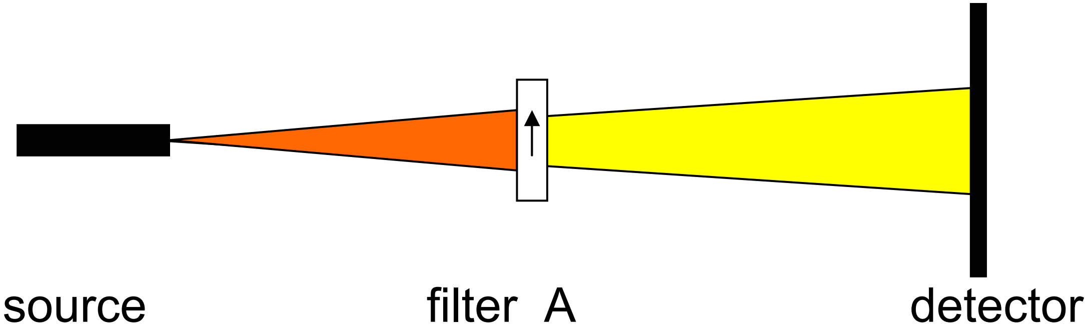
Introduction to Quantum Computing

Photon: Polarization direction

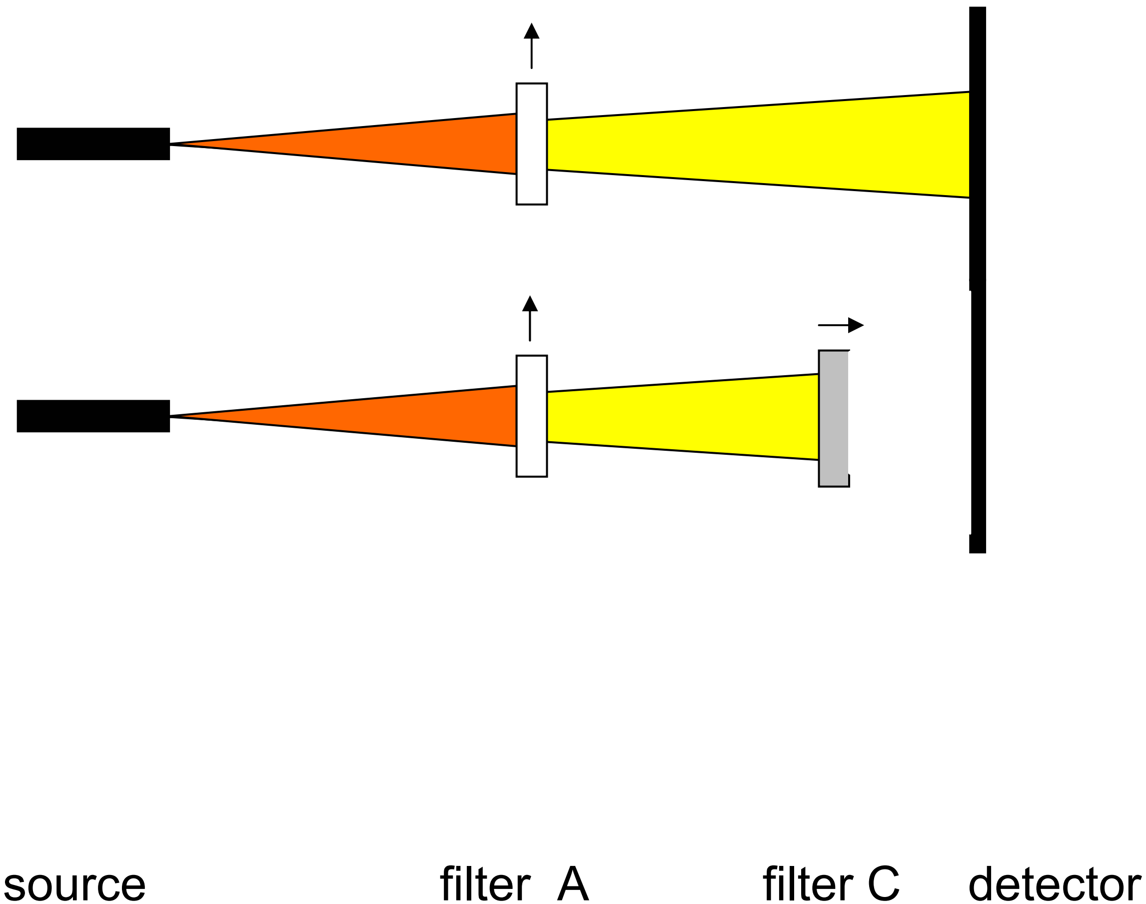


Photon Polarization:

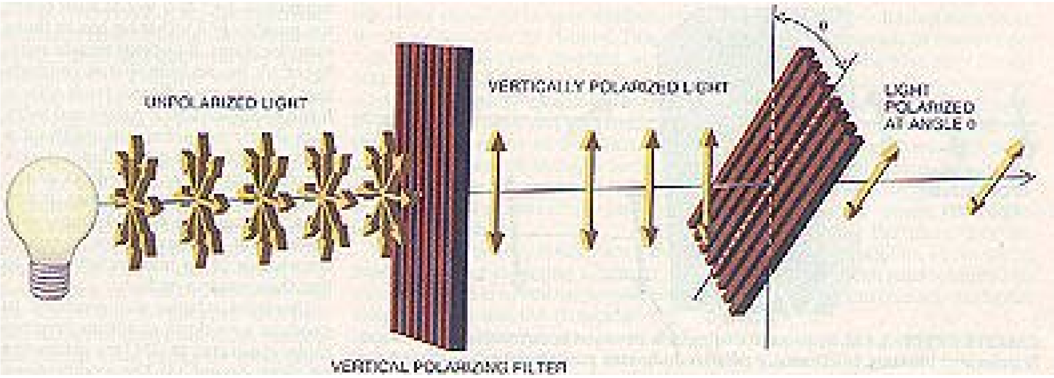


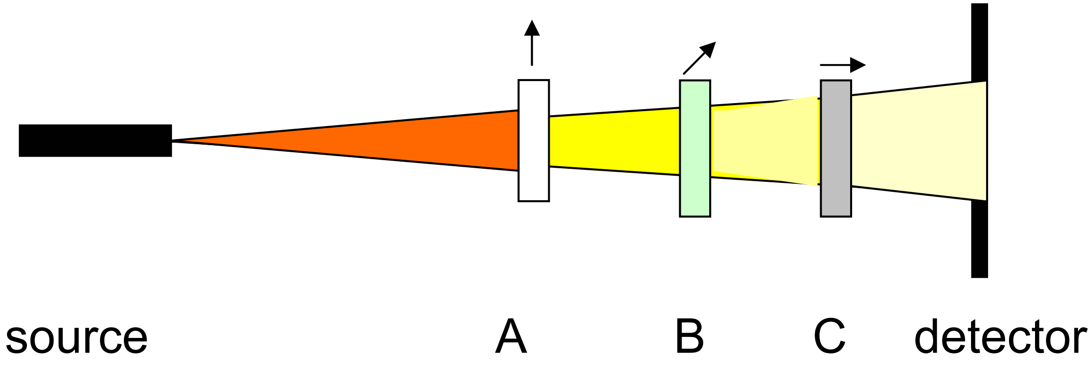


Two orthogonal Filter:



Three filter in 45°





Explanation:

Photon polarization can be described as direction
That is a linear combination of up↑ or right→ .

Description in a vector space of the form

$$\psi = a\,|\uparrow\rangle + b\,|\rightarrow\rangle \quad \text{with} \quad |a|^2 + |b|^2 = 1$$

$|a|^2$ the probability for state $|\uparrow\rangle$

The three filters in the previous example
are related to polarization direction

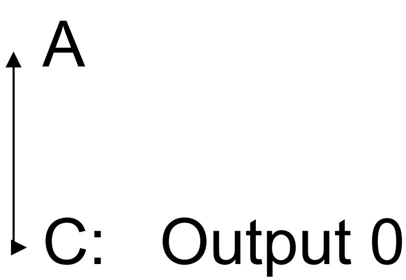
↑ (A) , ↗(B) , and → (C).

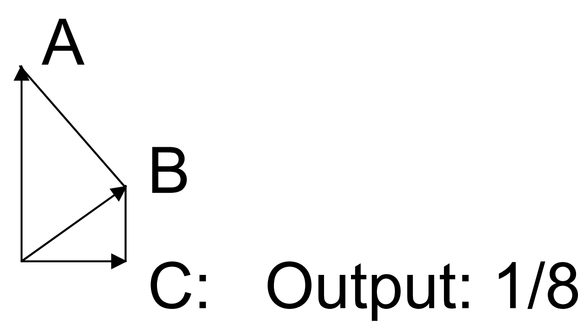
Filter A restricts the photon ψ to ist component $|\uparrow>$.

If this photon beam reaches filter C, it is restricted to its $\rightarrow$ orthogonal component, that is zero.

With additional filter B in between, $|\uparrow>$ is restricted to its 45° $\nearrow$ component φ.

Additional filter C restricts φ to $\rightarrow$.



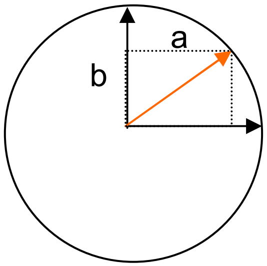


State space representation for quantum system:

Quantum state can be measured as |0> or |1>.

General state can be described as   a |0> + b |1>
with $|a|^2 + |b|^2 = 1$ or

$$a\begin{pmatrix}1\\0\end{pmatrix}+b\begin{pmatrix}0\\1\end{pmatrix}$$



Measuring forces the state to be |0>   or |1> .

Inner product between state vectors |x> and |y>:
<x| |y> = <x|y>   in bra-ket notation (equiv. to $x^T y$).

Outer product (matrix): $|x><y|$, e.g. $|0><1| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$

Outer product can also be used to describe transformations of quantum states:

$$X = |0><1| + |1><0| = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Different descriptions for operator X:

$$\left\{ \begin{aligned} & X\left(a|0> + b|1>\right) = \left(|0><1| + |1><0|\right)\left(a|0> + b|1>\right) = \\ & \qquad = a|1> + b|0> \\ & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix} \end{aligned} \right.$$

$$X: \quad \begin{aligned} |0> &\rightarrow |1> \\ |1> &\rightarrow |0> \end{aligned}$$

Quantum Bit = Qubit

Qubit is a unit vector in a two-dimensional complex vector space with fixed basis, denoted by { |0>, |1> }

The orthonormal basis can be related e.g. to
- polarization ↑ and →
- spin up and down (1/2 or -1/2) of an electron or a nucleus.

The basis states |0> and |1> represent the classical bit values 0 and 1. Each measurement gives only |0> or |1>.

But qubits can be in a superposition of |0> and |1>, e.g. a |0> + b |1> with complex a,b of norm 1.

$|a|^2$ and $|b|^2$ giving the probabilities of state |0> or |1>.

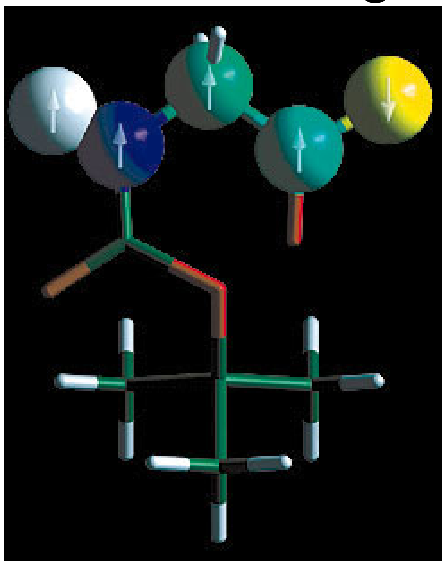
Quantum bit can be in infinitely many superposition states, but we can extract only a single bit‘s worth of information:

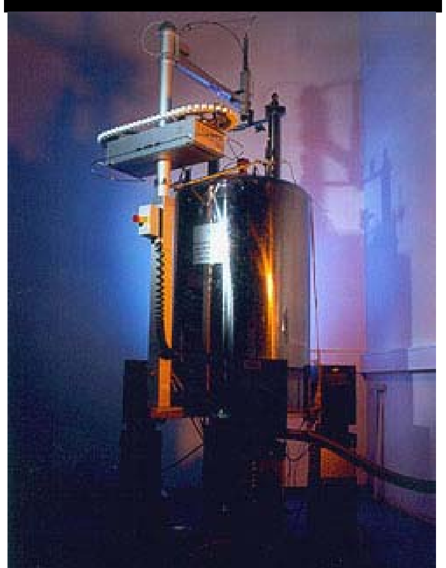
Measurement forces the state to |0> or |1> with only two possible results.

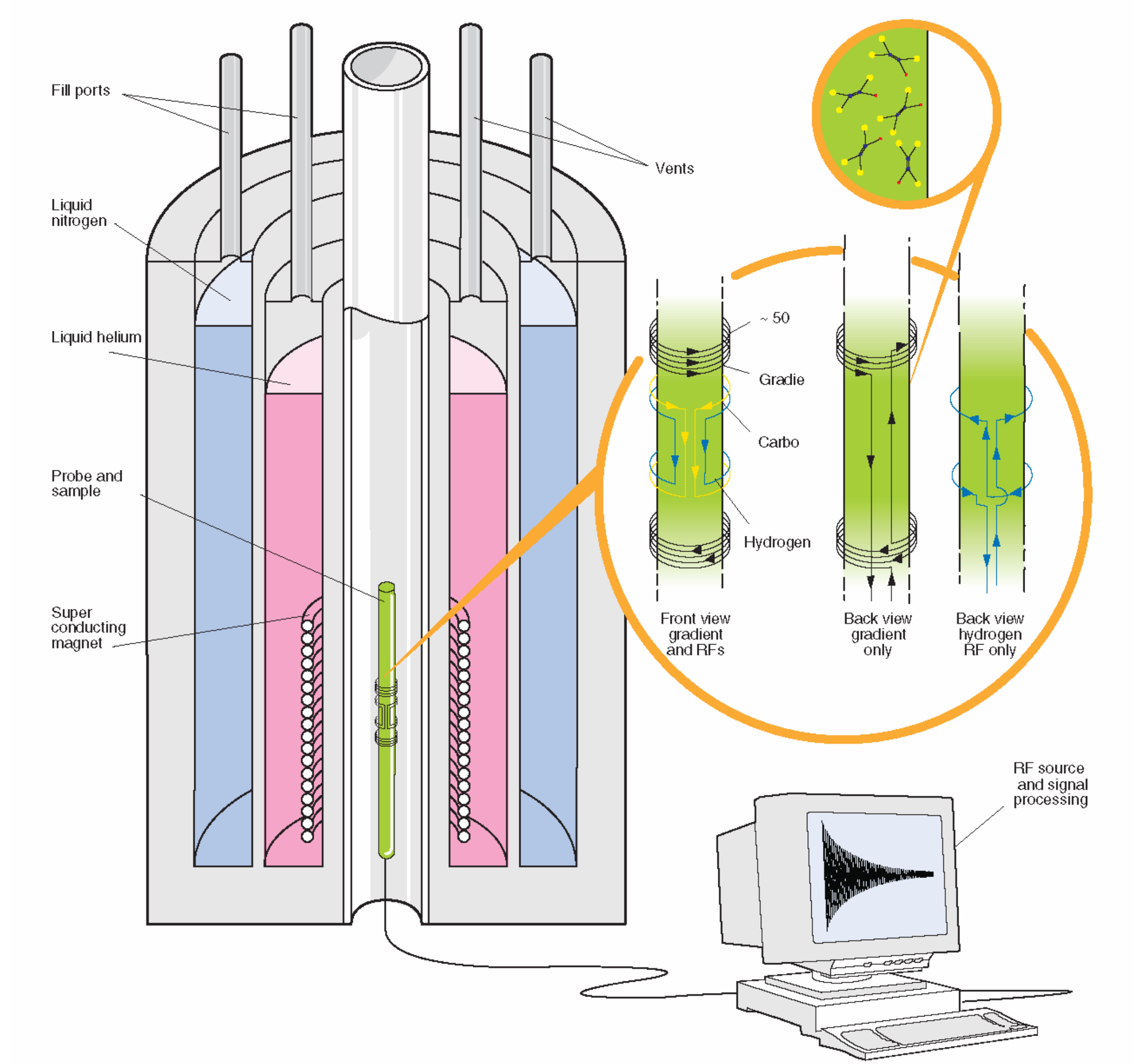
Measurement changes the system!

Realization of Qubits?

Qubits can be realized by Nuclear Magnetic Resonance NMR as spin of a number of nuclei of a molecule in a liquid that contains a large number of these molecules.







Molecule: ^{13}C trichloroethylene (TCE).

Nuclei: hydrogen nucleus (proton) has strong magnetic moment.

Inside powerful external magnetic field, each proton‘s spin prefers to align itself with the field.

By RF pulses spin direction can be induced to tip off-axis → static field leads to precession around the main axis.

The magnetic field induces by the precessing protons can be detected by magnetic induction.

Two possible states of spin can have different energy level in view of the external magnetic field.

Multiple Qubits:

Consider n-particle quantum system.

Classical: the space of n-particel system, where each
particel has two possible states, can be described
by a 2n-dimensional vector space.
Cartesian product: $\dim(X \times Y) = \dim(X) + \dim(Y)$

Quantum system: Description in 2^n-dimensional vector space!
Tensor product: $\dim(X \otimes Y) = \dim(X) \cdot \dim(Y)$

Four basis vector for two states:

$$|0> \otimes |0>, \quad |0> \otimes |1>, \quad |1> \otimes |0>, \quad |1> \otimes |1>$$

Excursion: Tensor product of matrices and vectors:

Matrices $$A=\left(a_{j,k}\right)_{j,k=1}^{n}, B=\left(b_{r,s}\right)_{r,s=1}^{m} \Rightarrow A\otimes B=\left(a_{j,k}\cdot B\right)_{j,k=1}^{n}$$

Example:

$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}\otimes\begin{pmatrix}10 & 11\\ 12 & 13\end{pmatrix}=\left(\begin{array}{cc|cc}10 & 11 & 20 & 22\\ 12 & 13 & 24 & 26\\ \hline 30 & 33 & 40 & 44\\ 36 & 39 & 48 & 52\end{array}\right)$$

Vectors $$x=\left(x_j\right)_{j=1}^{n}, y=\left(y_r\right)_{r=1}^{m} \Rightarrow x\otimes y=\begin{pmatrix}x_1 y & \cdots & x_n y\end{pmatrix}$$

$$\begin{pmatrix}1 & 2 & 3\end{pmatrix}\otimes\begin{pmatrix}10 & 11\end{pmatrix}=\left(\begin{array}{cc|cc|cc}10 & 11 & 20 & 22 & 30 & 33\end{array}\right)$$

Abbreviation: $|0> \otimes |0> = |00>, \quad |01>, \quad |10>, \quad |11>$

first second qubit

4 basis states for two qubits:

$$|x> = a|00> + b|01> + c|10> + d|11>$$

Dimension of two qubit state space: 2*2=4

Example: Three qubits

Basis is given by |000> , … , |111> and each state can be described in this basis as superposition by

a_1 |000> + a_2 |001> + … + a_8 |111> with a_1 … $a_{8,}$, ||a||=1.

$$|000>=\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}=|0>|0>|0>$$

3 qubits lead to 2^3=8 dimensional space,
q qubits to 2^q dimensional space.

Entangled States:

Some states cannot be described by the decomposition into two separated component states:

$$|y> = |00> + |11>$$

$$(a|0> + b|1>) \otimes (c|0> + d|1>) = ac|00> + ad|01> + bc|10> + bd|11> \neq y$$

Such states are called entangled states, they have no classical counterpart.

Exponential growth of state space suggests possible exponential speed-up of computations
on quantum computers!

Quantum Gates – Pauli Matrices

$I:$ $\quad |0> \;\rightarrow\; |0>$ $\quad |1> \;\rightarrow\; |1>$ $\quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ Identity

$X:$ $\quad |0> \;\rightarrow\; |1>$ $\quad |1> \;\rightarrow\; |0>$ $\quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ Negation, P_x

$Y:$ $\quad |0> \;\rightarrow\; -i\,|1>$ $\quad |1> \;\rightarrow\; i\,|0>$ $\quad i\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ Phase shift, P_y

$Z:$ $\quad |0> \;\rightarrow\; |0>$ $\quad |1> \;\rightarrow\; -|1>$ $\quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ Combination of X and Y, P_z

Possible transformations (quantum gate) for one qubit.

Four unitary basis matrices for four-dim. space of unitary 2 x 2 matrices.

Two-Qubit Gate: C$_{not}$

$$
C_{not}: \quad |00> \;\rightarrow\; |00> \qquad |01> \;\rightarrow\; |01> \qquad |10> \;\rightarrow\; |11> \qquad |11> \;\rightarrow\; |10>
$$

$$
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}
$$

C$_{not}$ acts as the identity on the second qubit iff the first qubit is in state |0>;
If the first qubit is |1>, the second qubit is changed like X.

Representation by tensor product and Pauli matrices:

$$
C_{not} = \left(I \otimes I + Z \otimes I + I \otimes X - Z \otimes X \right) / 2 =
$$

$$
= \left((I + Z) \otimes I + (I - Z) \otimes X \right) / 2
$$

Two-Qubit Gate: Walsh-Hadamard Transformation

One-dim. case:

$$H: \quad |0> \quad \to \quad \frac{1}{\sqrt{2}}\left(|0>+|1>\right)$$
$$|1> \quad \to \quad \frac{1}{\sqrt{2}}\left(|0>-|1>\right)$$

n-dim. case:

$$W = H \otimes H \otimes \cdots \otimes H$$

Application generates a superposition of all 2^n possible states:

$$\left(H \otimes H \otimes \cdots \otimes H\right)|00\cdots 0>=$$
$$=\frac{1}{\sqrt{2^n}}\left(\left(|0>+|1>\right)\otimes\left(|0>+|1>\right)\otimes\cdots\otimes\left(|0>+|1>\right)\right)=$$
$$=\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}|j>$$

Quantum Gate Arrays

Quantum gates are always unitary operators.
In order to model a classical function f(x) we consider a quantum gate array U_f defined by

$$U_f : |x, y> \quad \rightarrow \quad |x, y \oplus f(x)>$$

where $\oplus$ denotes the bitwise exclusive-OR.

U_f is unitary and can be realized by quantum gate array.

To computed f(x) we apply U_f on |x,0>.
The result f(x) can be read of as the value y with

$$f(x) \oplus f(x) = 0 = y \oplus f(x)$$

Quantum Parallelism

U_f is applied to an input vector in superposition.
Hence, U_f is applied to all basis vectors in the superposition simultaneously and will generate a superposition of the results.
In this way it is possible to compute f(x) for n values of x in a single application of U_f.

Start with n-qubit state |00…0>.
Apply Walsh-Hadamard transformation W → superposition

$$\left(H \otimes H \otimes \cdots \otimes H\right) | 00 \cdots 0 > =$$

$$= \frac{1}{\sqrt{2^n}} \left(\left(| 0 > + | 1 > \right) \otimes \cdots \otimes \left(| 0 > + | 1 > \right) \right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n - 1} | j >$$

$$U_f\left(\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}|x,0>\right)=\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}U_f(|x,0>)=$$

$$=\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}|x,f(x)>$$

computes f(x) by n qubits with 2^n states simultaneously.
Problem: Measurement and interpretation of output.

Famous quantum algorithms:
-Quantum Fourier Transform
(of length 2^m with m(m+1)/2 gates)
-Shor‘s algorithm for factoring n-digit numbers
(in polynomial time)
- Grover‘s search algorithm (O(√n))

Unitary Matrices and Lie Algebras

Quantum Algorithm ←→ Unitary Matrix

Space of unitary matrices ***U*** is a Lie Group
U is a smooth manifold and a group where multiplication and inversion are smooth mappings.

The tangential space ***T*** to ***U*** in identity I is a Lie Algebra:

Mapping from the vector space of hermitian matrices in the unitary Lie group:

$$U = \exp(iH), \quad U \in \mathcal{U}, \quad H \quad hermitian$$

is called the exponential mapping.

Important Lie Groups and Lie Algebras:

U(n): unitary matrices
u(n): hermitian matrices

exp(iH): ***u***(n) $\rightarrow$ ***U***(n)

$$\exp(iH) = \exp(iU\Lambda U^H) = U \cdot \exp(i\Lambda) \cdot U^H$$

SU(n): unitary matrices with det(U)=1
su(n): hermitian matrices with trace = 0

exp(iH): ***su***(n) $\rightarrow$ ***SU***(n)

trace(H)=0 $\rightarrow \det(\exp(iH)) = \det(\exp(i\Lambda)) = \prod_j \exp(i\lambda_j) = \exp\left(\sum_j i\lambda_j\right) = \exp(0) = 1$

Quantum Dynamics

Wave function ψ(t) describes the state of a quantum system depending of time t.
Vector |ψ(t)> element of Hilbert space with orthogonal basis |ψ_k(t)>, k=1,2,…

$$|\psi(t)>=\sum_k c_k\,|\psi_k(t)>, \qquad c_k=<\psi\,|\,\psi_k>$$

Change of |ψ(t)> in time is described by the Hamilton operator H and follows the Schrödinger equation

$$\frac{\partial}{\partial t}|\psi(t)>=\frac{i}{\hbar}H(t)|\psi(t)>$$

Stationary solution: $|\psi(t)>=U(t)\,|\,\psi(0)>=\exp(-iHt)\cdot|\,\psi(0)>$

Hamiltonian H is Hermitian.

The real eigenvalues of H are the energy levels of stationary states described by the related eigenvectors:

$$H \mid \psi_k >= E_k \mid \psi_k >$$

Hamiltonian $H = H_{drift} + H_{control}$

internal coupling (drift)   external pulse (control)

Numerical Problems

1. Mathematical properties of related matrices
2. Matrix exponential
3. Quantum compiler and parallel matrix multiplication
4. Approximating the smallest eigenvalue of huge H
5. Solving linear systems

1. Properties of Matrices in Quantum Computing

Numerical methods should take into account special properties of the considered matrices, as

-sparsity (e.g. PDE,…)

-structure (e.g. FFT, symplectic)

-general dense

Typical Matrices I

$$\sum_k \alpha_k Q_1^{(k)} \otimes Q_2^{(k)} \otimes \cdots \otimes Q_p^{(k)}$$

$$Q_j^{(k)} \in \{I, P_x, P_y, P_z\} : Pauli\ \ matrices$$

$Q_j^{(k)}$ describing the interaction between different quantum states.

Usually, most of the $Q_j^{(k)}$ are I.

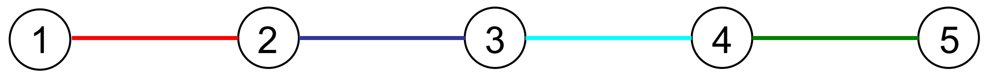
The other are P_x, P_y, or P_z

$$I_2 \otimes \cdots \otimes I_2 \otimes \overset{j}{P_x} \otimes I_2 \otimes \cdots \otimes I_2$$

$$I_2 \otimes \cdots I_2 \otimes \overset{j}{P_z} \otimes I_2 \otimes \cdots I_2 \otimes \overset{k}{P_z} \otimes I_2 \otimes \cdots I_2$$

Typical Matrices II

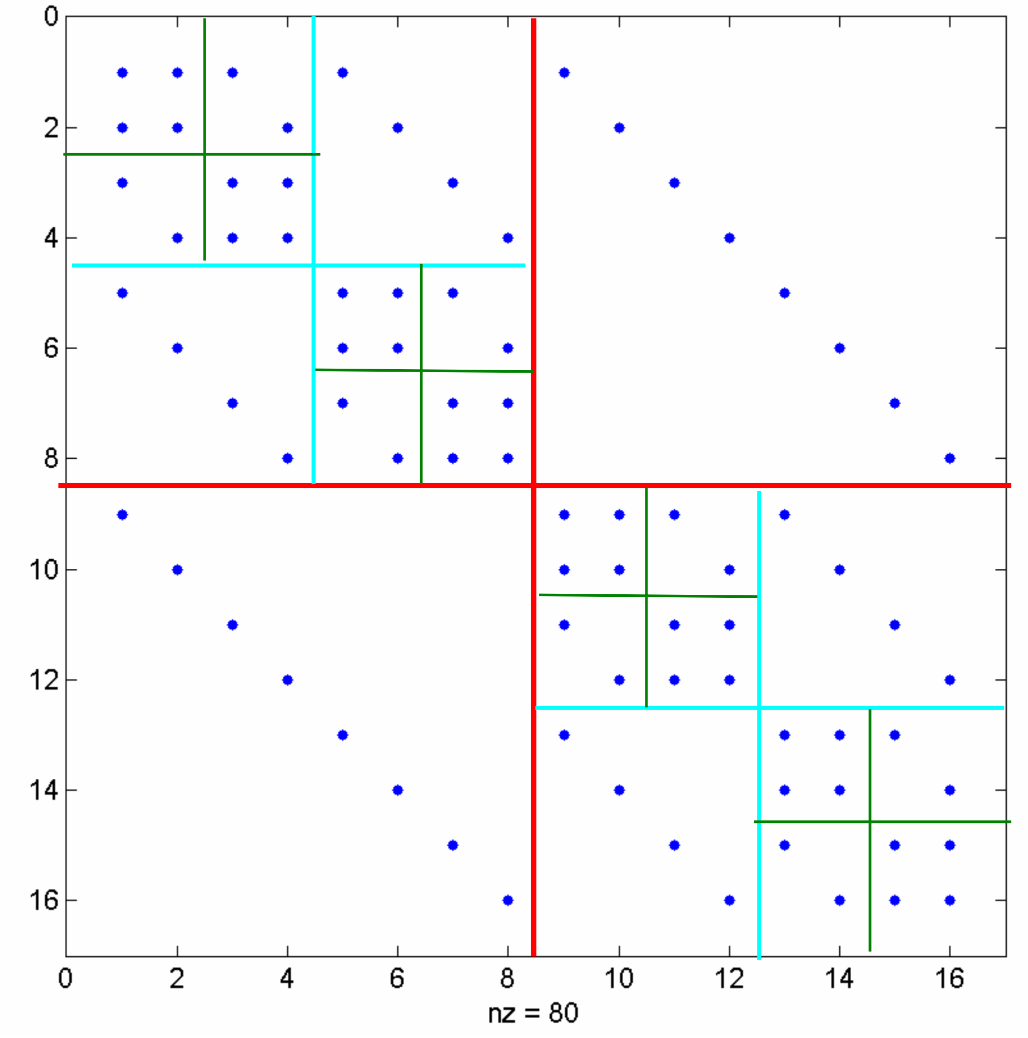
1D spin chain, drift Hamiltonian



$$
\begin{aligned}
H := & \boxed{\alpha_1 \cdot P_x \otimes P_x \otimes I \otimes I \otimes I} \\
+ & \boxed{\alpha_2 \cdot I \otimes P_x \otimes P_x \otimes I \otimes I} \\
+ & \boxed{\alpha_3 \cdot I \otimes I \otimes P_x \otimes P_x \otimes I} \\
+ & \boxed{\alpha_4 \cdot I \otimes I \otimes I \otimes P_x \otimes P_x}
\end{aligned}
$$

Typical Pattern

1D spin chain, control Hamiltonian



Sparsity: O(n log(n))

Structured: constant along diagonals

Typical matrices III

$$H_x = \sum_{j=1}^{p} a_j \cdot \left(I_2 \otimes \cdots \otimes I_2 \otimes \overset{j}{P_x} \otimes I_2 \otimes \cdots \otimes I_2 \right)$$

$$H_y = \sum_{j=1}^{p} b_j \cdot \left(I_2 \otimes \cdots \otimes I_2 \otimes \overset{j}{P_y} \otimes I_2 \otimes \cdots \otimes I_2 \right)$$

$$H_{zz} = \sum_{j<k=1}^{p} c_{j,k} \cdot \left(I_2 \otimes \cdots I_2 \otimes \overset{j}{P_z} \otimes I_2 \otimes \cdots I_2 \otimes \overset{k}{P_z} \otimes I_2 \otimes \cdots I_2 \right)$$

Properties of matrices

All Pauli matrices are unitary and hermitian

$$P_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}:$$

$$P_y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} = i\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}:$$

$$P_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}:$$

$$P_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}:$$

P_x is circulant and symmetric persymmetric:

Circulant:

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_0 & c_1 & & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & c_1 \\ c_1 & \cdots & c_{n-2} & c_{n-1} & c_0 \end{pmatrix}$$

Symmetric Persymmetric:

$$J \cdot C \cdot J = C^T \left(= C\right)$$

J is Anti-Identity

$$J = \begin{pmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{pmatrix}$$

Circulant Matrices

Fourier-Matrix F_n , unitary, symmetric,
are closely related to the Discrete Fourier Transform (FFT)
(All computations can be done in O(n log(n)))

$$F_n = \left(\exp\left(\frac{2\pi i}{n} jk \right) \right)_{j,k=0}^{n-1} = \left(\omega^{jk} \right)_{j,k=0}^{n-1} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & & \omega^{n-1} \\ \vdots & & \ddots & \vdots \\ 1 & \omega^{n-1} & \cdots & \omega^{(n-1)(n-1)} \end{pmatrix}$$

Circulant matrix describes convolution: $C = F_n^H \cdot \Lambda \cdot F_n$

$$F_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad P_x = F_2 \cdot D_2 \cdot F_2$$

Symmetric Persymmetric

P_x is symmetric with respect to the main diagonal and with respect to the anti-diagonal:

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix} = \begin{pmatrix} & J \\ J & \end{pmatrix} \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} & J \\ J & \end{pmatrix} = \begin{pmatrix} JCJ & JB^T J \\ JBJ & JAJ \end{pmatrix}$$

$$\Downarrow$$

$$A = JCJ \qquad and \qquad B = JB^T J$$

$$\begin{pmatrix} I & J \\ I & -J \end{pmatrix} \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} I & I \\ J & -J \end{pmatrix} = U^T \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} U =$$

$$= \begin{pmatrix} A + BJ + JB^T + JCJ & A - BJ + JB^T - JCJ \\ A + BJ - JB^T - JCJ & A - BJ - JB^T + JCJ \end{pmatrix} = 2 \begin{pmatrix} A + BJ & 0 \\ 0 & A - BJ \end{pmatrix}$$

H_x is symmetric, persymmetric, p-level circulant

$$H_x = \sum_{j=1}^{p} a_j \cdot \left(I_2 \otimes \cdots \otimes I_2 \otimes \overset{j}{P_x} \otimes I_2 \otimes \cdots \otimes I_2 \right)$$

can be diagonalized in the form $F_2 \otimes \cdots \otimes F_2$

Furthermore $J = J_2 \otimes \cdots \otimes J_2$

By this formula $J\,H_x\,J = H_x$ can be reduced to $J\,P_x\,J = P_x$.

$$P_y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}:$$

P_y is skewcirculant and antisymmetric persymmetric:

Skewcirculant:
(can be reduced to the circulant case by a diagonal Transformation!)

$$\begin{pmatrix} s_0 & s_1 & \cdots & s_{n-2} & s_{n-1} \\ -s_{n-1} & s_0 & s_1 & & s_{n-2} \\ -s_{n-2} & -s_{n-1} & s_0 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & s_1 \\ -s_1 & \cdots & -s_{n-2} & -s_{n-1} & s_0 \end{pmatrix} = \overline{\Omega} \cdot C \cdot \Omega$$

Skewsymmetric Persymmetric:

$$U^T P_y U = \begin{pmatrix} 0 & * \\ * & 0 \end{pmatrix}$$

H_y is skewsymmetric persymmetric, p-level skewcirculant

$$H_y = \sum_{j=1}^{p} b_j \cdot \left(I_2 \otimes \cdots \otimes I_2 \otimes \overset{j}{P_y} \otimes I_2 \otimes \cdots \otimes I_2 \right)$$

can be diagonalized by $\left(\overline{\Omega_2} F_2 \Omega_2\right) \otimes \cdots \otimes \left(\overline{\Omega_2} F_2 \Omega_2\right)$

Furthermore $U^T \cdot H_y \cdot U = \begin{pmatrix} 0 & B_1 \\ B_2 & 0 \end{pmatrix}$

H_{zz} is diagonal and symmetric persymmetric

$$P_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}: \qquad H_{zz} = \sum_{j<k=1}^{p} c_{j,k} \cdot \left(I_2 \otimes \cdots I_2 \otimes \overset{j}{P_z} \otimes I_2 \otimes \cdots I_2 \otimes \overset{k}{P_z} \otimes I_2 \otimes \cdots I_2 \right)$$

P_z is skewpersymmetric (change of sign).

Therefore, H_{zz} is again persymmetric.

Matrix $H=H_{zz}+H_x+H_y$?

Consider $$aP_x + bP_y = \begin{pmatrix} 0 & c \\ \overline{c} & 0 \end{pmatrix} = \begin{pmatrix} 0 & re^{i\varphi} \\ re^{-i\varphi} & 0 \end{pmatrix}$$

aP_x+bP_y is ω-zirkulant:

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}\begin{pmatrix} 0 & re^{i\varphi} \\ re^{-i\varphi} & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ & e^{-i\varphi} \end{pmatrix} = \overline{D}\left(aP_x + bP_y\right)D = \begin{pmatrix} 0 & r \\ r & 0 \end{pmatrix} = C$$

Therefore, $$\left(\overline{D_1} \otimes \cdots \otimes \overline{D_p}\right)\cdot\left(H_x + H_y\right)\cdot\left(D_1 \otimes \cdots \otimes D_p\right)$$

is real symmetric p-level circulant.
Furthermore:

$$\left(\overline{D_1} \otimes \cdots \otimes \overline{D_p}\right)\cdot H_{zz} \cdot \left(D_1 \otimes \cdots \otimes D_p\right) = H_{zz}$$

$H = H_{zz} + H_x + H_y$?

Therefore, $$\left(\overline{D}_1 \otimes \cdots \otimes \overline{D}_p\right) \cdot \left(\boxed{H_x + H_y} + \boxed{H_{zz}}\right) \cdot \left(D_1 \otimes \cdots \otimes D_p\right)$$

is real symmetric and is build from two matrices,

that are both persymmetric.

It holds

$$U^T \cdot \left(\overline{D}_1 \otimes \cdots \otimes \overline{D}_p\right) \cdot \left(H_x + H_y + H_{zz}\right) \cdot \left(D_1 \otimes \cdots \otimes D_p\right) \cdot U = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$$

All computations for $H_x + H_y + H_{zz}$ can be reduced to two real matrices A_1 and A_2 of half size.
Improvement of upto a factor 16.

Computation of eigendecoposition

$$H = H_x + H_y + H_{zz}$$

By diagonal matrix D the matrix H can be transformed into a real symmetric persymmetric matrix R, which can be reduced by U into a real Block matrix diag(A,B):

$$H = D\left(\overline{D}HD\right)\overline{D} = DR\overline{D} = DU\left(U^T\overline{D}HDU\right)U^T\overline{D} = DU\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}U^T\overline{D}$$

Eigendecomposition of A and B → eigendecomposition of H.

2. Computation of exponential of a matrix

Definition: $$\exp(A) = e^{A} = \sum_{k=0}^{\infty} A^{k} / k!$$

„19 dubious ways to compute the exponential of a matrix“
Moler, van Loan

Example: Taylor expansion is numerically instable

$$\exp\begin{pmatrix} 100 & 0 \\ 0 & -100 \end{pmatrix} = \sum \begin{pmatrix} 100^{k} & 0 \\ 0 & (-100)^{k} \end{pmatrix} / k! = \begin{pmatrix} \sum \frac{100^{k}}{k!} & 0 \\ 0 & \sum \frac{(-100)^{k}}{k!} \end{pmatrix}$$

Cancellation for exp(-x)

Basic facts:

$$\exp(A+B) \neq \exp(A)\cdot\exp(B)$$

Scaling and Squaring:

$$\exp(A) = \exp(A/2^p)^{2^p} = B^{2^p}$$

Compute $\exp(A/2^p)$ and recover exp(A)!

Allows numerical stable computation by

```
B = exp(A/2^p);
for j = 1 : p
        B = B*B;
end
```

1. Padé approximation:

$$\exp(x) \approx \frac{p_n(x)}{q_n(x)} \quad \Longrightarrow \quad \exp(A) \approx q_n^{-1}(A) \cdot p_n(A)$$

with polynomials p_n and q_n such that the series expansion of $p_n(x)/q_n(x)$ coincides with exp(x) for the first 2n coefficients.

2. Eigendecomposition of A:

$$\exp(A) = \exp(U\Lambda U^H) = U \exp(\Lambda) U^H = U \begin{pmatrix} \exp(\lambda_1) & & \\ & \ddots & \\ & & \exp(\lambda_n) \end{pmatrix} U^H$$

3. Chebychev Expansion:

For -1<x<1: $$e^x = J_0(i) + 2 \cdot \sum_{k=1}^{\infty} i^k J_k(-i) T_k(x)$$

with $T_k(x)$ Chebychev polynomial of first kind
and $J_k(x)$ Bessel function.

Finite Chebychev expansion $s_n(x)$ is defined as
Least Squares approximation of the form

$$\int_{-1}^{1} \left| e^x - s_n(x) \right|^2 \cdot \frac{dx}{\sqrt{1-x^2}} = \min_{p \in P_n} \int_{-1}^{1} \left| e^x - p(x) \right|^2 \cdot \frac{dx}{\sqrt{1-x^2}}$$

Chebychev expansion for matrix A:

$$\exp(A) \approx J_0(i) \cdot I + 2\sum_{k=1}^{n} i^k J_k(-i) T_k(A) = s_n(A)$$

Three-term recursion for $T_k(x)$ gives polynomial coefficients of $s_n(x)$ with O(n-1) matrix multiplications:

$$s_n(A) = a_0 I + a_1 A + \cdots + a_n A^n$$

Faster methods for computing $s_n(A)$ by partitioning, e.g.:

$$s_n(A) = a_0 I + a_2 A^2 + \cdots + a_{2m} A^{2m} + A \cdot \left(a_1 I + a_3 A^2 + \cdots a_{2m-1} A^{2m} \right)$$

Takes only n/2+1 matrix products.

Fastest evaluation of matrix polynomial:

$$k: \qquad n = k^2 - 1$$

$$\begin{aligned} s_n(x) = a_{0,0}I + \cdots + a_{k-1,0}A^{k-1} + A^k \cdot \left(a_{0,1}I + \cdots + a_{k-1,1}A^{k-1}\right) + \\ + A^{2k} \cdot \left(a_{0,2}I + \cdots + a_{k-1,2}A^{k-1}\right) + \cdots + \\ + A^{(k-1)k} \cdot \left(a_{0,k-1}I + \cdots + a_{k-1,k-1}A^{k-1}\right) \end{aligned}$$

takes matrix multiplications: $A^2, \ldots, A^{k-1}, A^k, \ldots, A^{(k-1)k}$,
and the products of powers of A with partial polynomials:

$(k-2) + (k-1) + (k-1) = 3k-2 = O(\sqrt{n})$ matrix multiplications.

Example: n=8 and k=3

$$s_n(x) = a_{0,0}I + a_{1,0}A + a_{2,0}A^2 + A^3 \cdot \left(a_{0,1}I + a_{1,1}A + a_{2,1}A^2\right) +$$
$$+ A^6 \cdot \left(a_{0,2}I + a_{1,2}A + a_{2,2}A^2\right)$$

takes matrix multiplications: A^2 , A^3, A^6,
and the products of A^3 and A^6 with partial polynomials:

5 matrix multiplications (instead of 7 with Horner).

3. Quantum Compiler

Quantum algorithm is described by unitary matrix U_G .

Find optimal implementation of U_G , e.g. on NMR, using elementary Quantum gates!

Find short factorization of U_G in terms of elementary tensor products of Pauli matrices.

Leads to numerical optimization problem.

Compilization

Elementary Quantum transformations represented by unitary matrices exp(i*H_j) with $H_j = \sum \alpha_{k,j} Q_1^{(k)} \otimes \cdots \otimes Q_p^{(k)}$

Sequence of Quantum transformations by

exp(i*H_1)*…*exp(i*H_m) =! exp(i*H) = U_G

Find smallest number of factors, defined by $\alpha_{k,j}$.

Numerical tasks connected with this optimization problem:

Compute U_j = exp(i*H_j)

Compute all products U_1*U_2, U_1*U_2*U_3, … U_1*U_2*…*U_m

Parallel Multiple Matrix Multiplication

Compute $$H_{1,k} = U_1 \cdot U_2 \cdot \cdots \cdot U_k$$

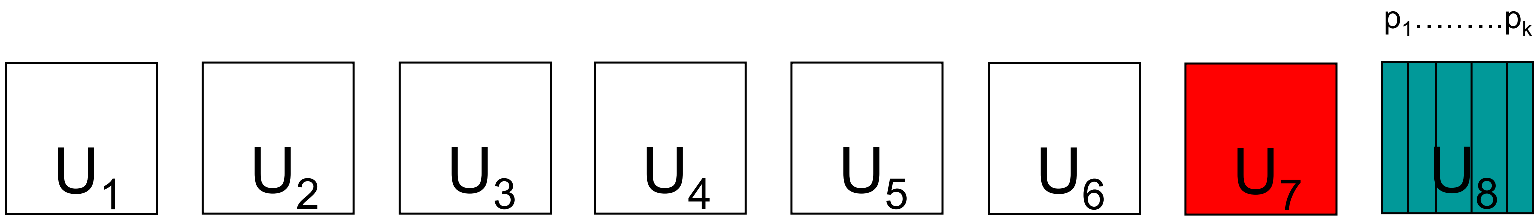
for all k=1,2,…,N

with n x n – matrices $U_1, \ldots, U_N$

Total costs sequentially: $N*n^3$

There exist fast matrix-matrix algorithms that are faster than n^3 (Strassen, group-theoretic)
Conjecture: $O(n^{2+\varepsilon})$

Block Column Parallel

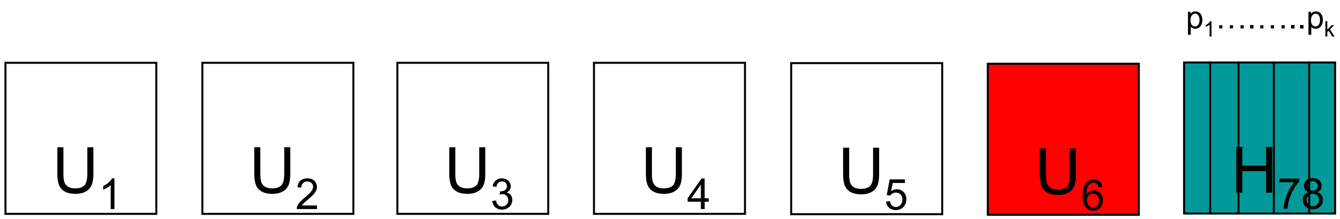


Distribute U_8 on k processors $p_1 \dots p_k$ together with full U_7 .

$$
\begin{array}{c|c|c|c}
p_1 & p_2 & \cdots & p_k \\
 & & & \\
U_7 \cdot U_8(:,1:n_1) & U_7 \cdot U_8(:,n_1+1:n_2) & \cdots & U_7 \cdot U_8(:,n_{k-1}+1:n)
\end{array}
$$

Gives $H_{7,8} = U_7\, U_8$

Block Column Parallel

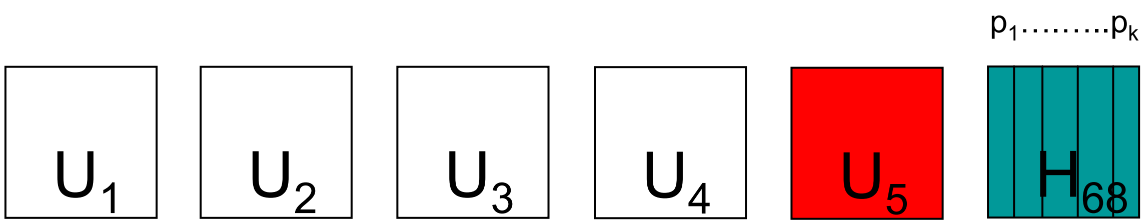


Send full U_6 to all processors $p_1 \ldots p_k$.

p_1	p_2	$\cdots$	p_k
$U_6 \cdot H_{78}(:,1:n_1)$	$U_6 \cdot H_{78}(:,n_1+1:n_2)$	$\cdots$	$U_6 \cdot H_{78}(:,n_{k-1}+1:n)$

Gives $H_{6,8} = U_6\, U_7\, U_8$

Block Column Parallel

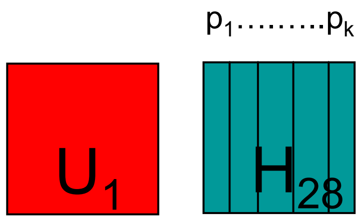


Send full U_5 to all processors $p_1 \ldots p_k$.

$$
\begin{array}{c|c|c|c}
p_1 & p_2 & \cdots & p_k \\
& & & \\
U_5 \cdot H_{68}(:,1:n_1) & U_5 \cdot H_{68}(:,n_1+1:n_2) & \cdots & U_5 \cdot H_{68}(:,n_{k-1}+1:n)
\end{array}
$$

Gives $H_{5,8} = U_5\, U_6\, U_7\, U_8$

Block Column Parallel

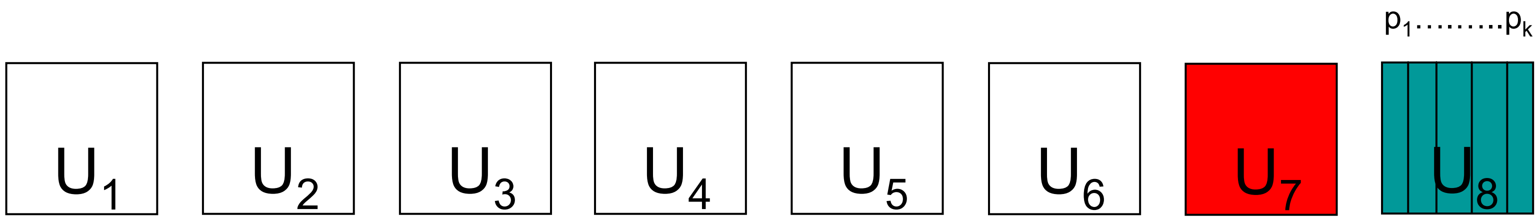


Send full U_1 to all processors $p_1 \ldots p_k$.

$$
\begin{array}{c|c|c|c}
p_1 & p_2 & \cdots & p_k \\
 & & & \\
U_1 \cdot H_{28}(:,1:n_1) & U_1 \cdot H_{28}(:,n_1+1:n_2) & \cdots & U_1 \cdot H_{28}(:,n_{k-1}+1:n)
\end{array}
$$

Gives $H_{1,8} = U_1 \ldots U_6\, U_7\, U_8$

Costs in Parallel:

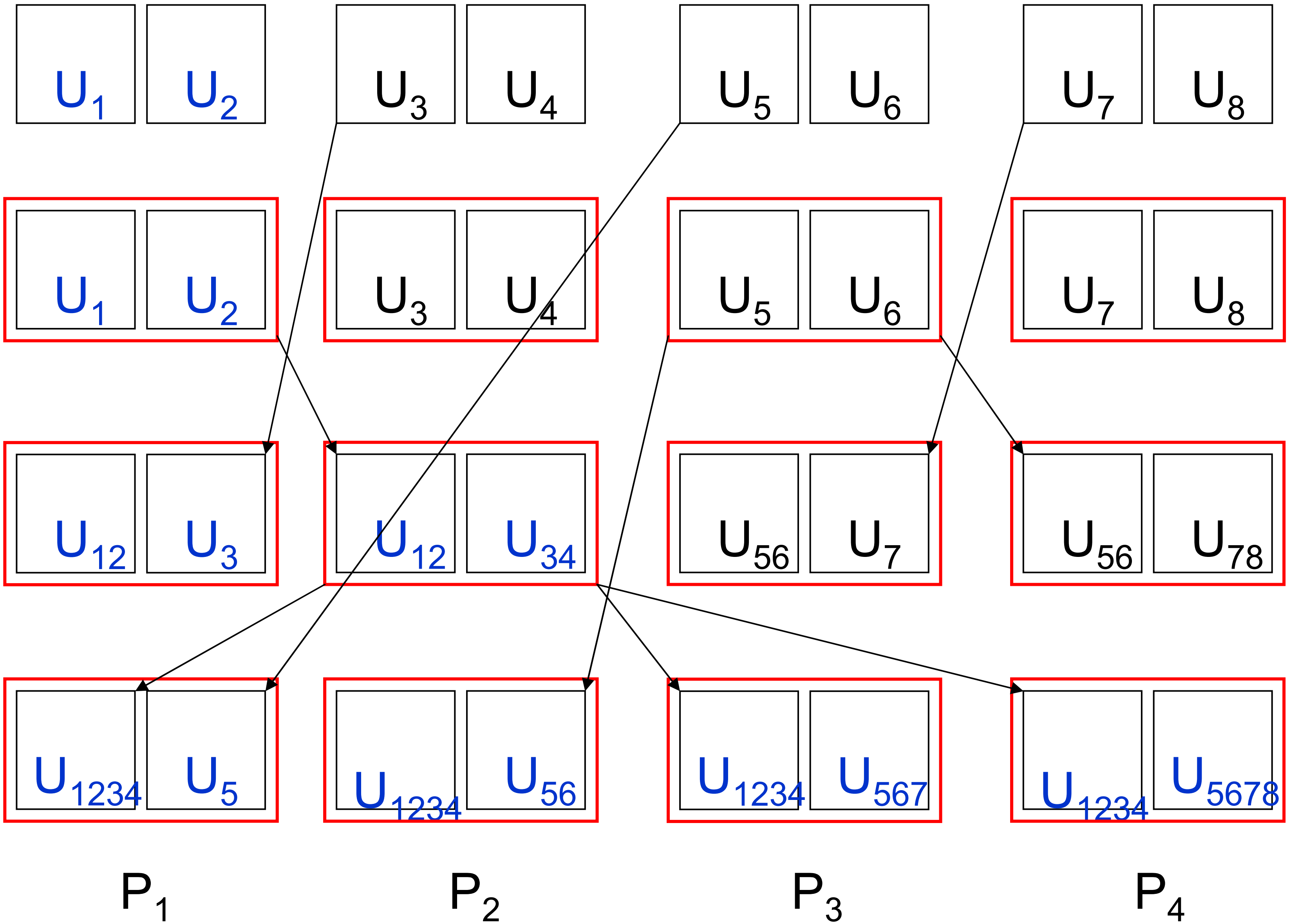


N-1 times $n^2 * n/k$ = $(N-1)*n^3 / k$

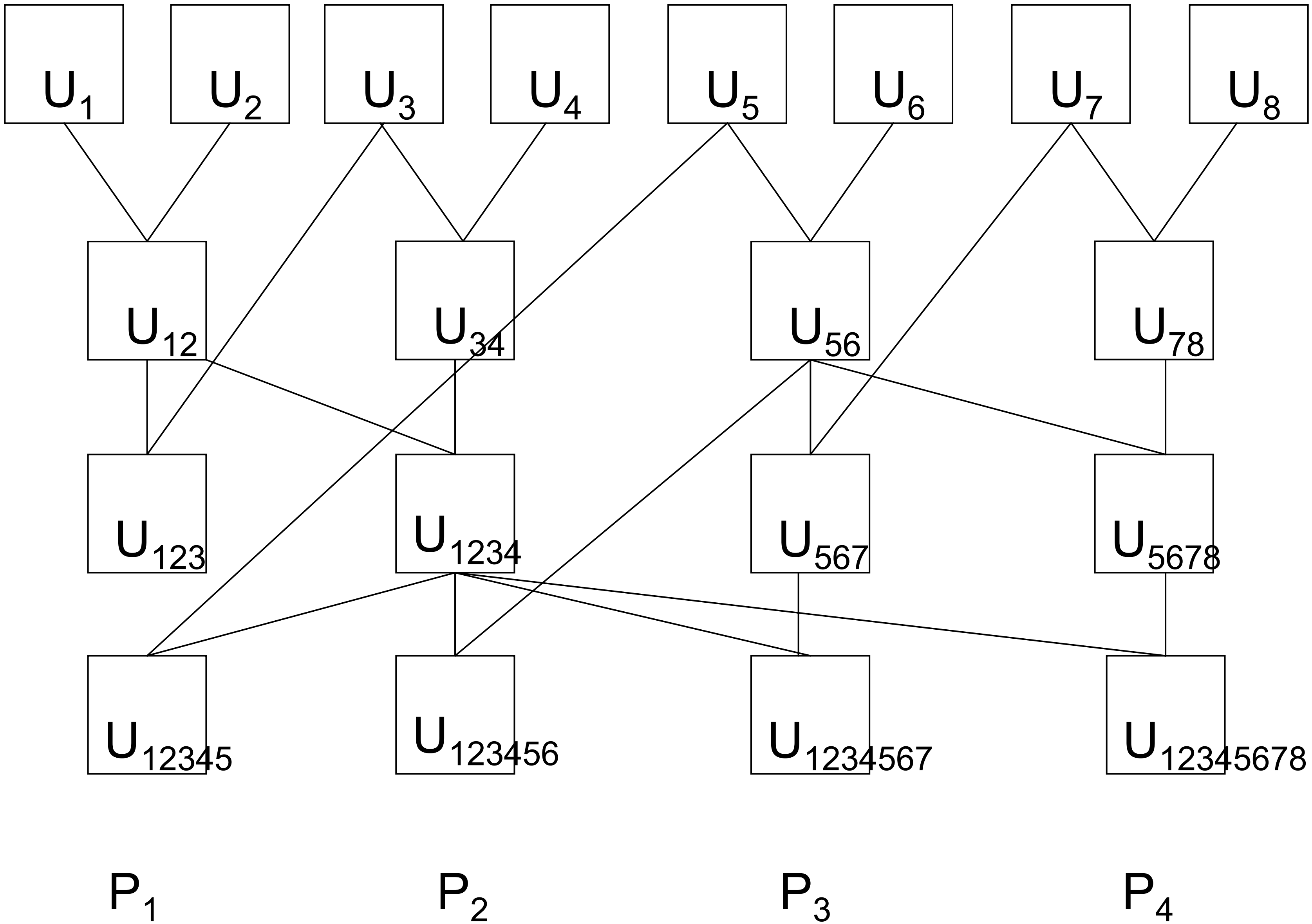
For N matrices of n x n size with k processors.

Especially for 8 matrices and 4 processors: $(7/4)*n^3$

Parallel Prefix Tree



Parallel Prefix Tree



Parallel Prefix Tree

Costs: $\log(N)*n^3$ with N/2 processors

Especially: $3*n^3$

A little bit more expensive than the columnwise method,

but less communication/storage.

4. Smallest Eigenvalue

Given Hamiltonian H for p qubits, p more than 50.

Problem: Size of H is $2^{50} \sim 1.1 \cdot 10^{15}$

Lowest energy level is given by the smallest eigenvalue

Numerical task:
Compute approximation to smallest eigenvalue of H, but H is so large that it is not possible to build vector of this size!

Use Rayleigh Quotient based on special vectors

$$\min_{x \neq 0} \frac{x^H H x}{x^H x} = \lambda_{\min}(H)$$

$$H = \sum_{k=1}^{M} \alpha_k \cdot Q_1^{(k)} \otimes \cdots \otimes Q_p^{(k)}$$

Consider only vectors such that Hx is computable, e.g.

$$x = \sum_k x_1^{(k)} \otimes \cdots \otimes x_p^{(k)}$$

Find $x_j^{(k)}$ that minimize the Rayleigh Quotient

5. Linear Systems

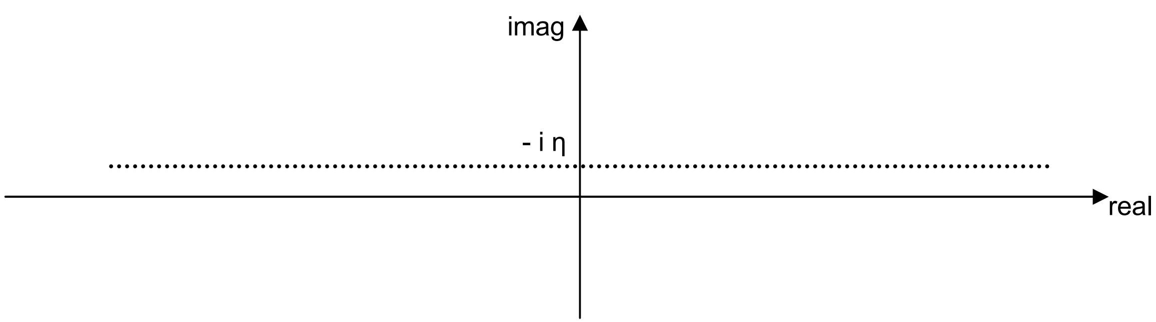
Solve $$\left(H-\omega-i\eta\right)x=b$$

for $\eta \to 0$

with H hermitian indefinite.

Use iterative solver GMRES, Bicgstab,…?

Spectrum:



$$\left(H-\omega\right)-i\eta\cdot I$$

Hermitian part: H-ω, indefinite

Skewhermitian part: -η·I, positive definite

Because of the special hermitian/skewhermitian structure GMRES (Arnoldi) leads to a tridiagonal upper Hessenberg matrix:

Short recurrence, cheap iterative step.

GMRES ←→ MINRES ←→ cg on normal equations