Structure-Preserving Algorithms for Quantum Digital Twins: A Numerical Analyst's Journey

Daniel Appelö

VIRGINIA TECH

Why quantum computing?

VIRGINIA TECH

Why quantum computing?

Who are these?
A bunch of people jumping on the latest hype!

VIRGINIA TECH

Forsythe, Milne

Hartree

Blanch

Taussky-Todd

Danzig, Ostrowski

Wasow, Hestenes

Lanczos Steifel John

INA 1947-1956. Primary concern: “development of mathematics pertinent to solving numerical computations. This… could only happen if some of the mathematicians were proficient in handling the electronic digital computers.”

Celebrating 70 years of the transistor

Imagination

soon 80 ...

23 December 1947
The first transistor is invented at Bell Labs, New Jersey USA

1949
‘Type A’ transistor enters limited production at Bell Labs

1952
The first application using a transistor is created – a hearing aid. Raytheon produces 10,000 CK718 hearing aids, making it the first commercial product to use transistors

1955
The first transistor calculator is created by IBM – the 608

1956
Shockley, Bardeen and Brattain awarded the Nobel Prize for the invention of the transistor

1957
Fairchild semiconductor division founded by engineers and scientists originally hired by William Shockley at Shockley Semiconductor Laboratory

1958
Fairchild begins commercial transistor shipments

1960
Sony announces the TV8 301, the first portable TV containing just 23 silicon transistors

1965
Intel’s Gordon Moore states that the number of transistors on a chip will double every year, a prediction that becomes known as Moore’s Law. (Later revised to every 18 months)

1980s
Processors increase significantly in performance. One example is the Intel 80286 16-bit microprocessor, containing over 134,000 transistors

2000s
Graphics processing units (GPUs), employing billions of transistors bring hardware 3D acceleration to the PC desktop

2010s
Transistor counts in typical smartphones typically exceed 2 billion with mobile GPUs accounting for most of this

Circuit Model - Quantum Algorithms

Definition 2.3.1. From Trefethen & Bau

Assume that $x, y \in F$. *Then in the* **standard model** *it holds that*

$$fl(x \text{ op } y) = (x \text{ op } y)(1+\delta), \quad |\delta| \leq u, \tag{2.3.1}$$

where u is the unit roundoff and "op" stands for one of the four elementary operations: $+$, $-$, $\times$, *and* $/$.

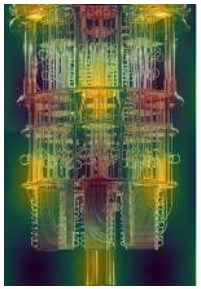
- Lin Lin's "Lecture Notes on Quantum Algorithms for Scientific Computation" arXiv:2201.08309
- Quantum Physics without the Physics: arXiv:2012.03865
- The language here is linear algebra with unitary matrices + estimates.

VIRGINIA TECH

Quantum Digital Twins - Modeling, Characterization and Control of QC

UNM

M. Motamed
Serafina Middleton (PhD student)
Chase Hodges-Heilmann (PhD student)

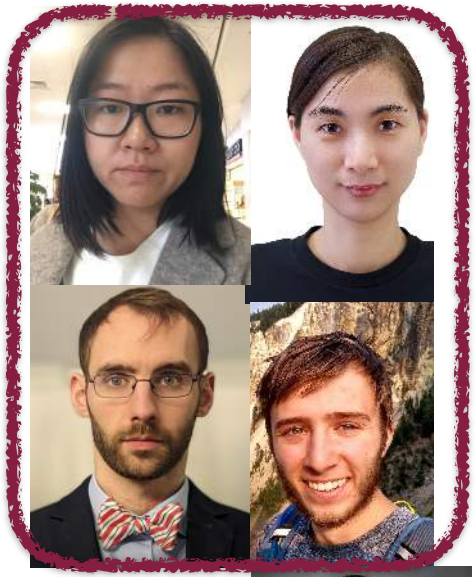


LLNL

Dr. Anders Petersson
Dr. Yujin Cho

Virginia Tech

Daniel Appelö
Xinwei Deng
Yingda Cheng
Jiuhua Hu (Postdoc)
Spencer Lee (PhD student)



SNL

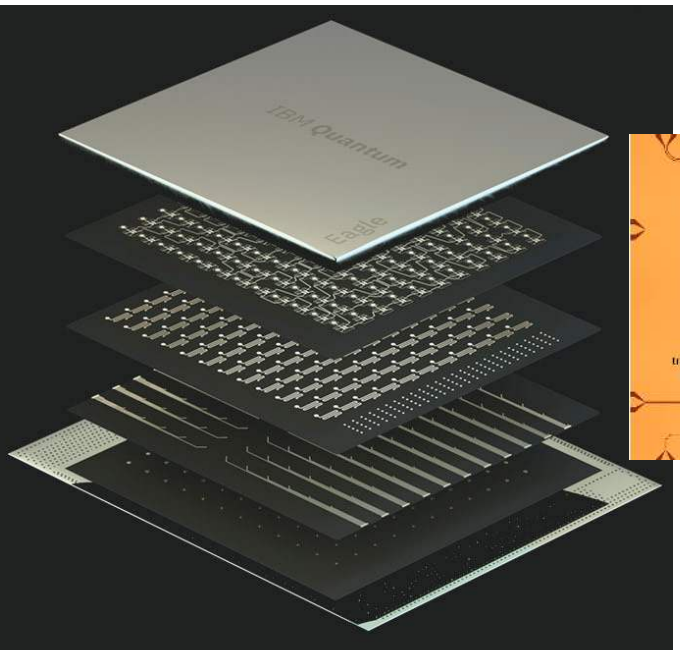
Dr. Gabriel Huerta
Dr. Bobby Middleton

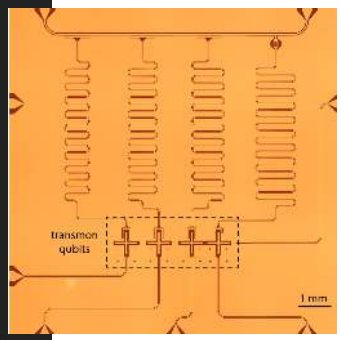
Funding

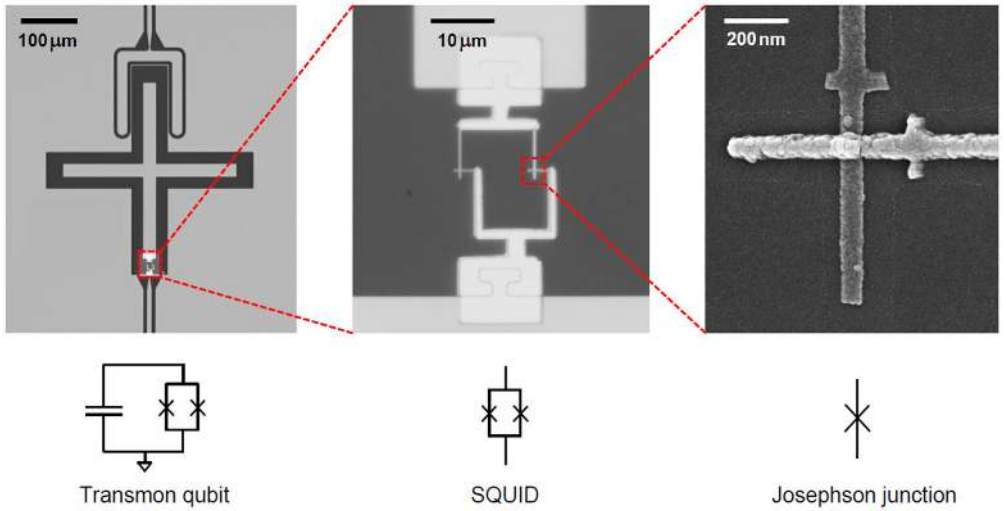
NSF DMS / AFSOR DT program
DOE Office of Science
ARO

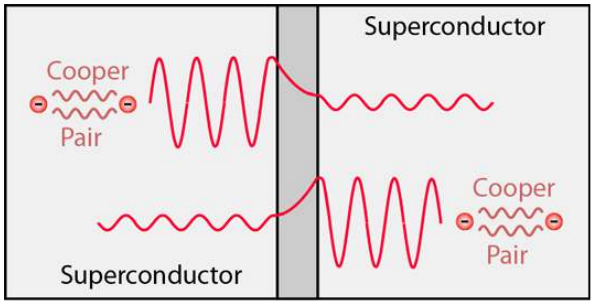
VIRGINIA TECH

Superconducting Transmon







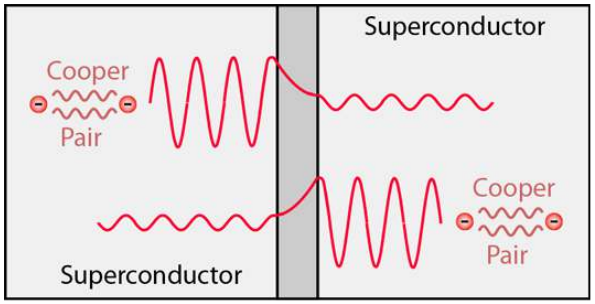


From hyperphysics.phy-astr.gsu.edu

- Requires ~K to see quantum effects ~ mK to suppress noise
- Jospehson junction allows Cooper pairs (quasiparticles) that have high correlation and who's wave functions can be measured far away.

Roth et al., An Introduction to the Transmon Qubit for Electromagnetic Engineers IEEE Antennas and Propagation Magazine, 2023

Transmon classic circuit analogy with pendulum

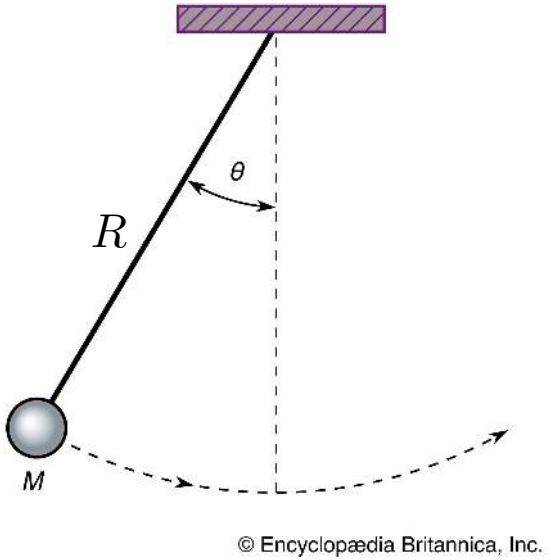


$n = N_1 - N_2$ Density difference in Cooper pairs

$\phi = \phi_1 - \phi_2$ Difference Josephson phase

Circuit Hamiltonian consists of a linear capacitor and a nonlinear inductor

$$H = H_C + H_L = 4E_C n^2 - E_J \cos\phi$$



Hamiltonian with kinetic and potential energy

L is angular momentum

$$H = \frac{1}{2MR^2} L^2 - gMR\cos\theta$$

Model for a transmon QuDit at LLNL's testbed

- Four energy levels
- State space governed by time dependent Schrödinger equation

$$i\hbar\frac{d\psi}{dt} = H\psi = (H_S + H_c(t))\psi$$

$$H_s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \omega_{0,1} & 0 & 0 \\ 0 & 0 & \omega_{0,1}+\omega_{1,2} & 0 \\ 0 & 0 & 0 & \omega_{0,1}+\omega_{1,2}+\omega_{2,3} \end{pmatrix}$$

$$H_c(t) = f(t)(a + a^\dagger)$$

$$f(t) = 2\,\mathrm{Re}\{e^{i\omega_d t} d(t)\} = 2I(t)\cos(\omega_d t) + 2Q(t)\sin(\omega_d t).$$

$$a := \begin{bmatrix} 0 & \sqrt{1} & & \\ & \ddots & \ddots & \\ & & \ddots & \sqrt{n-1} \\ & & & 0 \end{bmatrix}$$

Computational Tasks

$$i\hbar\frac{d\psi}{dt} = H\psi = (H_S + H_c(t))\psi$$

$$H_s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \omega_{0,1} & 0 & 0 \\ 0 & 0 & \omega_{0,1}+\omega_{1,2} & 0 \\ 0 & 0 & 0 & \omega_{0,1}+\omega_{1,2}+\omega_{2,3} \end{pmatrix}$$

$$H_c(t) = f(t)(a + a^\dagger)$$

$$f(t) = 2\,\mathrm{Re}\{e^{i\omega_d t}d(t)\} = 2I(t)\cos(\omega_d t) + 2Q(t)\sin(\omega_d t).$$

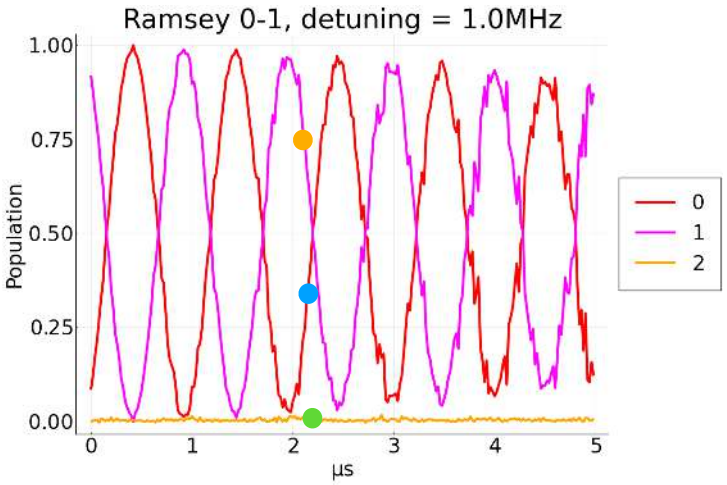
Finding the frequencies in blue is an inverse or inference problem

Given a target gate U_{Target} (a unitary matrix), finding I(t) and Q(t) so that

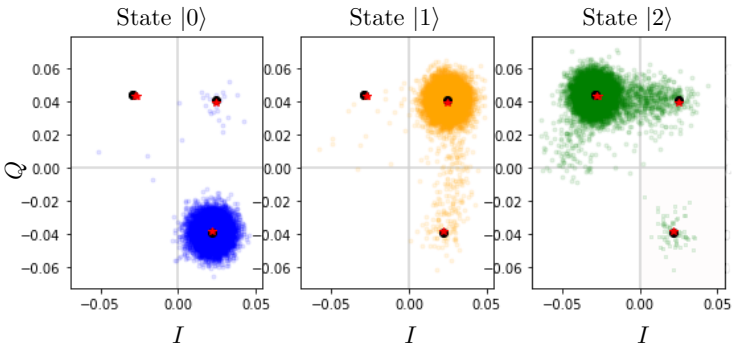
$$\psi(0) = \mathbf{e}_k \Rightarrow \psi(T) \approx U_{\text{Target}}[:, k], \forall k$$

Is the optimal control problem

Practicalities of the inverse problem



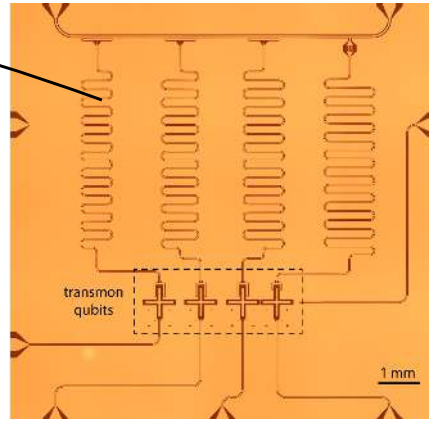
Protocol: $\frac{\pi_{0,1}}{2}$ $\xrightarrow[\text{free evolution}]{t_{\text{dark}}}$ $\frac{\pi_{0,1}}{2}$



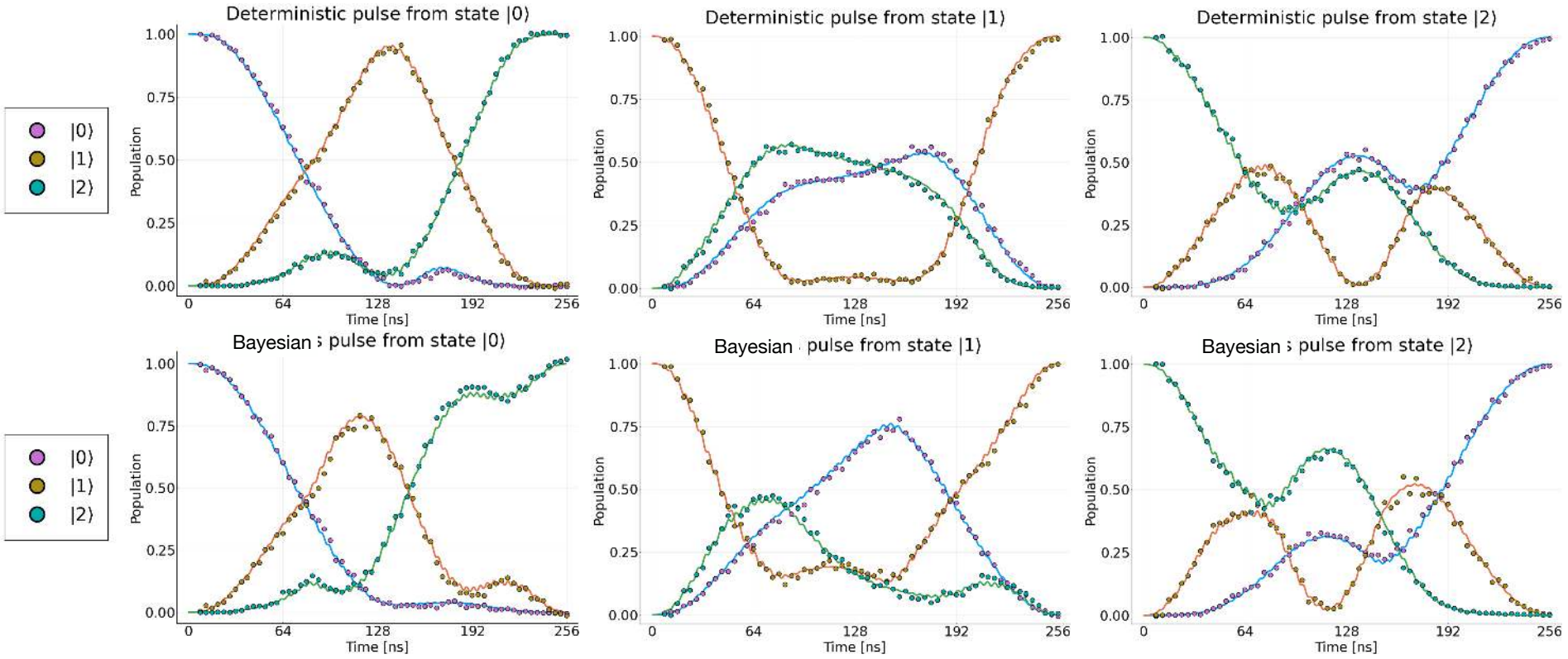
- In a real quantum computer we estimate the state by taking averages over 1k - 50k “shots”.
- In each measurement the state takes ONE value.
- The state cannot be measured directly but must be measured by demodulating the signal from a readout resonator (currently using a GMM).
- Model coax cables from 300K to 10mK.

$$I = \frac{2}{T_m}\int_0^{T_m} r(\tau)\cos(\omega_{IF}\tau)d\tau$$

$$Q = -\frac{2}{T_m}\int_0^{T_m} r(\tau)\sin(\omega_{IF}\tau)d\tau,$$



Uncertainty quantification improves robustness



	Deterministic	*Bayesian*
Gate fidelity	99.40%	99.49%
Entanglement fidelity	99.06%	99.14%

Peng, Appelö, Petersson, Garcia, Cho arXiv:2301.10712

Quantum Optimal Control Problem

Minimize the infidelity $\mathcal{I}(U(T;\theta), U_{\text{target}}) = 1 - \mathcal{F}(U(T;\theta), U_{\text{target}})$

Where the fidelity is $\mathcal{F}(U(T;\theta), U_{\text{target}}) := \frac{1}{E^2}\left|\left\langle U(T;\theta), U_{\text{target}} \right\rangle_F\right|^2$

$$\frac{d}{dt}U(t) = -iH(t;\theta)U(t), \quad U(0) = U_0 \in \mathbb{C}^{N\times E} \quad N = \prod_{q=1}^{N_Q} n_q$$

Challenges:

- State space grows exponentially
- Many forward solves
- Highly oscillatory solution
- Compute gradient of infidelity using adjoint

VIRGINIA TECH

High-Order Hermite Optimal Quantum Control

Write as a first order system and “solve”

$$\frac{d}{dt}\mathbf{w}(t) = A(t;\boldsymbol{\theta})\mathbf{w}(t) \qquad \mathbf{w}(t_{n+1}) = \mathbf{w}(t_n) + \int_{t_n}^{t_{n+1}} \frac{d\mathbf{w}(t)}{dt}\,dt$$

Approximate with Hermite interpolation and differentiate equation

$$\frac{d\mathbf{w}(t)}{dt} \approx \sum_{j=0}^{p-1} \mathfrak{Q}_{0,j}(t)\frac{d^{j+1}\mathbf{w}(t_n)}{dt^{j+1}} + \sum_{j=0}^{q-1} \mathfrak{Q}_{1,j}(t)\frac{d^{j+1}\mathbf{w}(t_{n+1})}{dt^{j+1}}$$

High-Order Hermite Optimal Quantum Control

Approximate with Hermite interpolation and differentiate equation

$$\sum_{j=0}^{q}(-1)^j c_j^{qp}\frac{\mathbf{w}_{n+1}^{(j)}}{j!}\Delta t^j = \sum_{j=0}^{p} c_j^{pq}\frac{\mathbf{w}_n^{(j)}}{j!}\Delta t^j \qquad c_j^{pq} = \binom{p}{j}\Big/\binom{p+q}{j}$$

$$\frac{d}{dt}\mathbf{w}(t) = A(t;\theta)\mathbf{w}(t) \qquad \mathbf{w}_n^{(j+1)} = \sum_{i=0}^{j}\binom{j}{i}A_n^{(j-i)}\mathbf{w}_n^{(i)}, \quad j = 0,\ldots,q-1$$

At any order the method looks like trapezoidal rule with “wrong” A

$$(I - \frac{\Delta t}{2}\tilde{A}(t_{n+1}))\mathbf{w}_{n+1} = (I + \frac{\Delta t}{2}\hat{A}(t_n))\mathbf{w}_n$$

Same system size at any order

Somewhat easy to compute adjoint

VIRGINIA TECH

CNOT Gate Example

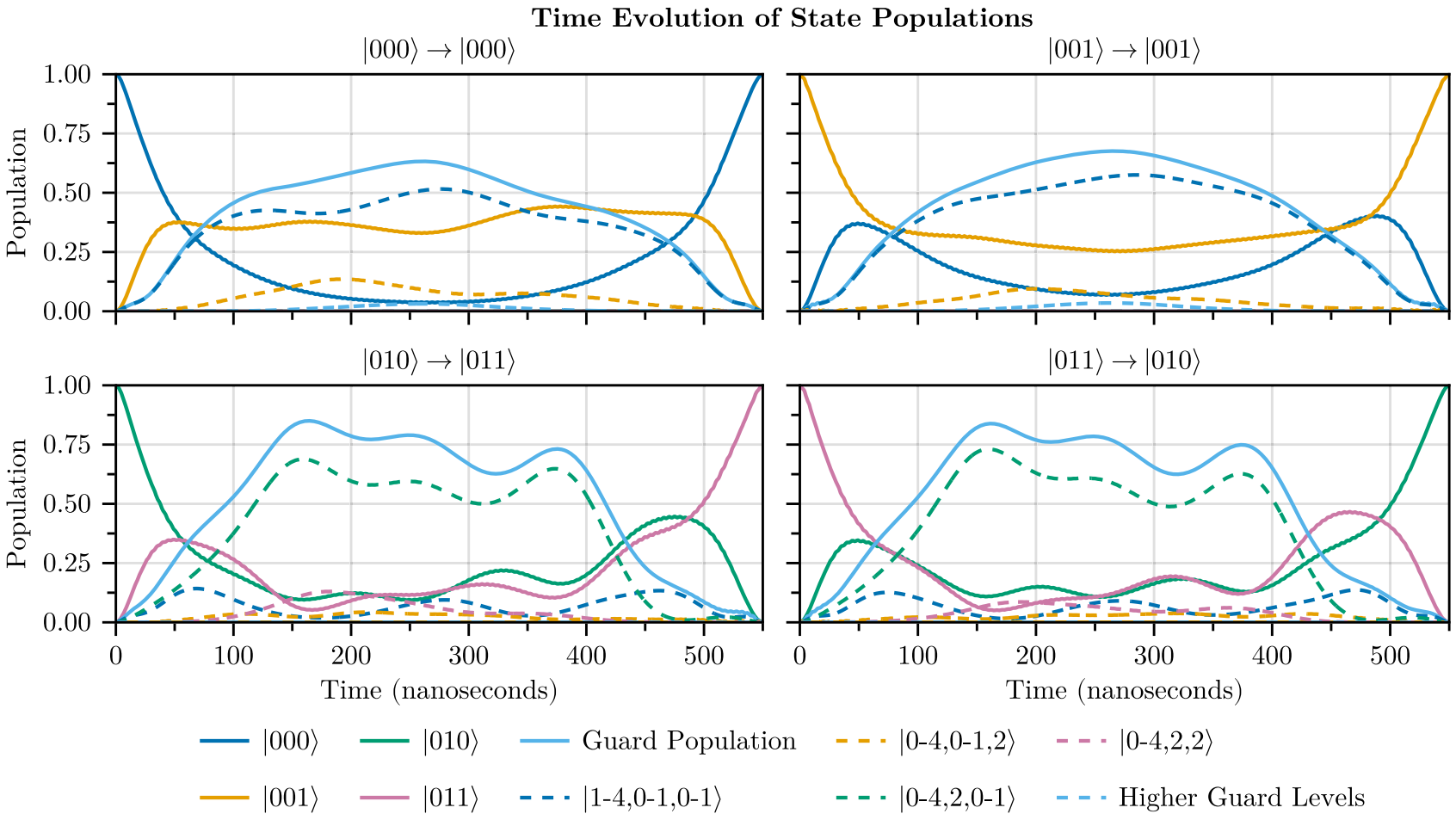
Hamiltonian for 2 transmon qubits coupled to a resonator:
(dispersive limit, rotating frame)

$$H(t) = -\frac{\xi_1}{2}a_1^\dagger a_1^\dagger a_1 a_1 - \frac{\xi_2}{2}a_2^\dagger a_2^\dagger a_2 a_2 - \frac{\xi_R}{2}a_R^\dagger a_R^\dagger a_R a_R$$
$$+ \xi_{12}a_2^\dagger a_2 a_1^\dagger a_1 + \xi_{1R}a_R^\dagger a_R a_1^\dagger a_1 + \xi_{2R}a_R^\dagger a_R a_2^\dagger a_2$$
$$+ \sum_{j\in\{1,2,R\}} p_j(t)(a_j + a_j^\dagger) + q_j(t)(a_j - a_j^\dagger)$$

System	Hilbert Space Size
Qudit 1	4
Qudit 2	4
Resonator	10
Total	160

	ξ_1	ξ_2	ξ_R
Value/2π (GHz)	1.2×10^{-1}	2.3×10^{-1}	2.8×10^{-5}
	ξ_{12}	ξ_{1R}	ξ_{2R}
Value/2π (GHz)	1×10^{-6}	2.5×10^{-3}	2.5×10^{-3}

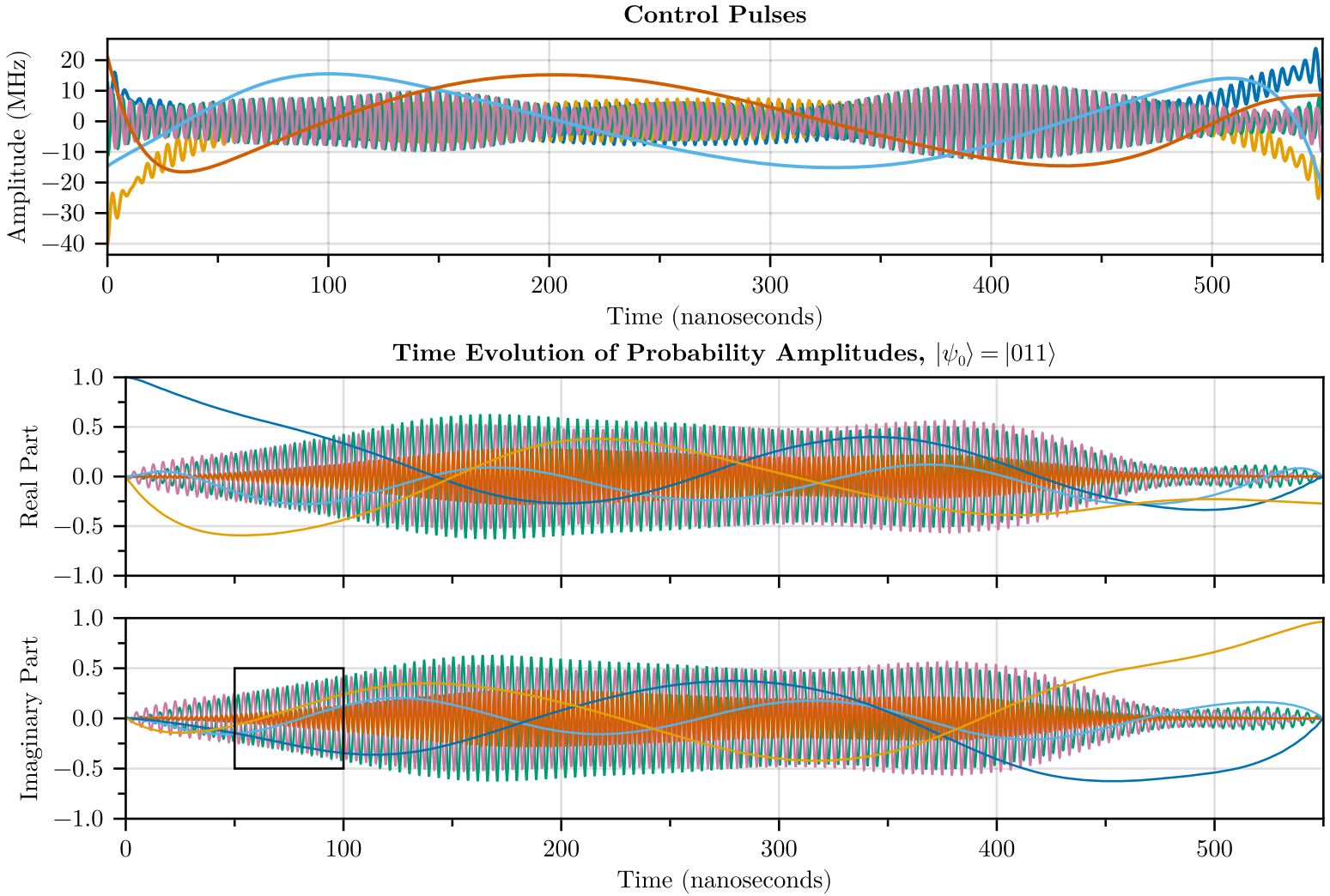
CNOT State populations



6th order
5409 steps

VIRGINIA TECH

Controls and real and imaginary part of state vector



VIRGINIA TECH

Speedup relative to Stormer-Verlet

(b) Speedup factor (relative to Störmer-Verlet) for each order of the Hermite method.

	Speedup Factor						
Target State Error	Order 2 (SV)	Order 2	Order 4	Order 6	Order 8	Order 10	Order 12
1.0(−1)	1.0	0.7	5.0	5.5	1.2	0.9	0.8
1.0(−3)	1.0	0.9	15.4	26.9	25.1	9.3	6.3
1.0(−5)	1.0	1.2	45.1	121.4	142.9	101.4	52.5
1.0(−7)	1.0	1.5	136.7	530.1	774.6	702.4	459.1

VIRGINIA TECH

Simulation of large open quantum systems

- If the state vector in a closed system is of dimension N then the density matrix has dimension N x N.
- The density matrix is **symmetric positive and has unit trace**.
- The Lindblad master equation guarantees that the density matrix remains positive as it evolves in time.
- A numerical method for solving the Lindblad equation should also have this Completely Positive Trace Preserving (**CPTP**) property.

Daniel Appelö, Yingda Cheng, **Kraus is king: High-order completely positive and trace preserving (CPTP) low rank method for the Lindblad master equation**, J. Comp. Phys., 2025.

VIRGINIA TECH

The Lindblad master equation

The evolution of our density matrix $\rho(t) \in \mathbb{C}^{N\times N}$, which is a unit-trace symmetric positive semi-definite (SPSD) matrix, satisfies

$$\frac{d}{dt}\rho(t) = -i(H\rho(t) - \rho(t)H) + \mathcal{L}_D\rho(t), \;\; \rho(0) = \rho_0.$$

Here H is the Hamiltonian, and $\mathcal{L}_D$ is on the form

$$\mathcal{L}_D\rho(t) = \sum_\alpha \gamma_\alpha \left(L_\alpha \rho(t) L_\alpha^\dagger - \frac{1}{2}\left(L_\alpha^\dagger L_\alpha \rho(t) + \rho(t) L_\alpha^\dagger L_\alpha \right) \right).$$

We can write the density matrix in its Cholesky factors

$$\rho(t) = V(t)V(t)^\dagger.$$

Here ρ is $N \times N$ but V is $N \times r$. Choose r adaptively or at the start of computation

When $r \ll N$ then the multiplication HV is much cheaper than $H\rho$.

VIRGINIA TECH

Considerations

- If the density matrix has low rank: $\boxed{\rho = \sum_{k=1}^{r} V_k V_k^\dagger}$ the method should exploit this.
- Only store the r vectors V_k, only carry out operations that scale with N .
- Our methods are exploiting Choi's theorem:
  a linear map $\mathcal{G}$ is CP iff it can be represented as $\boxed{\mathcal{G}\rho = \sum_l G_l^\dagger \rho G_l,}$
- Throughout let $\rho^n \approx \rho(t_n)$ be an approximation to the density matrix at time t_n.
- Our methods will be defined as a *Kraus map* $\mathcal{G}$ that takes ρ^n to ρ^{n+1}, that is

$$\rho^{n+1} \equiv \mathcal{G}\rho^n = \sum_l G_l^\dagger \rho^n G_l.$$

VIRGINIA TECH

Integrating Factor Induced CP Schemes

We rewrite the Lindblad equation in terms of an effective Hamiltonian J

$$\frac{d}{dt}\rho(t) = \left(J\rho(t) + \rho(t)J^\dagger\right) + L\rho(t)L^\dagger \equiv \mathcal{L}_J\rho(t) + \mathcal{L}_L\rho(t),$$

where $J = -iH_{\text{eff}},\ H_{\text{eff}} = H + \frac{1}{2i}L^\dagger L.$

- The operator $\mathcal{L}_L$ is already on Kraus form.
- The key to designing a CP scheme is the appropriate discretization of the operator $\mathcal{L}_J$.
- Using integrating factor we find

$$\frac{d}{dt}(e^{-\mathcal{L}_J t}\rho) = e^{-\mathcal{L}_J t}\mathcal{L}_L\rho.$$

VIRGINIA TECH

Integrating Factor Induced CP Schemes

Consider the application of $e^{\mathcal{L}_J \Delta t}$ onto a SPSD matrix $q(t) = V(t)V(t)^\dagger$.

We have $q(t+\Delta t) = e^{\mathcal{L}_J \Delta t} q(t) = V(t+\Delta t)V(t+\Delta t)^\dagger$.

Using the Cholesky factorization the following ODE are equivalent

$$\frac{d}{dt}q(t) = Jq(t) + q(t)J^\dagger \quad \Longleftrightarrow \quad \frac{d}{dt}V(t) = JV(t),$$

Denote $U(\Delta t)$ is the exact flow operator $e^{J\Delta t}$ so that $V(t+\Delta t) = U(\Delta t)V(t)$.

Note that $q(t+\Delta t) = U(\Delta t)q(t)U(\Delta t)^\dagger$, which is on Kraus form and thus CP.

CP is also true if we approximate the flow!

VIRGINIA TECH

Integrating Factor + Runge Kutta

$$\rho^{n+1} \equiv \mathcal{G}\rho^n = \sum_l G_l^\dagger \rho^n G_l.$$

Applying the RK method to $\frac{d}{dt}(e^{-\mathcal{L}_J t}\rho) = e^{-\mathcal{L}_J t}\mathcal{L}_L \rho$ yields

$$e^{-\mathcal{L}_J \Delta t}\rho_1 = \rho_0 + \Delta t \sum_{i=1}^{s} b_i e^{-\mathcal{L}_J c_i \Delta t}\mathcal{L}_L \rho^{(i)},$$

$$e^{-c_i \Delta t \mathcal{L}_J}\rho^{(i)} = \rho_0 + \Delta t \sum_{j=1}^{i-1} a_{ij} e^{-\mathcal{L}_J c_j \Delta t}\mathcal{L}_L \rho^{(j)}, \quad i = 1, \ldots, s.$$

Classic fourth-order method

The "original" Runge–Kutta method.[3]

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
	1/6	1/3	1/3	1/6

Regrouping terms and using the flow operator $U(\cdot)$ we get our final scheme

$$\rho^{(i)} = U(c_i\Delta t)\rho_0 U(c_i\Delta t)^\dagger + \Delta t \sum_{j=1}^{i-1} a_{ij} U((c_i - c_j)\Delta t)\mathcal{L}_L \rho^{(j)} U((c_i - c_j)\Delta t)^\dagger, \quad i = 1, \ldots, s,$$

$$\rho_1 = U(\Delta t)\rho_0 U(\Delta t)^\dagger + \sum_{i=1}^{s} b_i \Delta t U((1 - c_i)\Delta t)\mathcal{L}_L \rho^{(i)} U((1 - c_i)\Delta t)^\dagger,$$

VIRGINIA TECH

Algorithm 3.1 Low-rank CPTP time-integrator for the Lindblad equation

Input: Factor matrix $V \in \mathbb{C}^{N\times r}$ of $\rho = VV^\dagger = \rho(t)$; Butcher Tableau $\mathcal{S} = (\mathbf{A}, \mathbf{b}, \mathbf{c})$ of an s-stage explicit scheme; Step size Δt; Truncation tolerance τ.

Output: Updated factor $V' \in \mathbb{C}^{N\times r'}$ of $\rho(t + \Delta t)$.

$\tau \leftarrow \tau \,/\, (s+1)$ ▷ Tolerance per compression/truncation within the scheme

Intermediate Stages

$V^0 := V$

for $i = 1, 2, ..., s$ **do**

  $U^i = \texttt{Schrodinger-Solve}(V, c_i\Delta t)$ ▷ Solver for $\dot{\Psi} = -iH_{\text{eff}}\Psi$

  $W^{i-1} = \left[L_1 V^{i-1}, L_2 V^{i-1}, ..., L_P V^{i-1}\right]$ ▷ Applying jump operators L_p

  for $j = 1, 2, ..., i-1$ **do**

    $Y^{i,j} = \texttt{Schrodinger-Solve}(\sqrt{a_{ij}\Delta t}\; W^j,\; (c_i - c_j)\Delta t)$

  end for

  $V^i = \texttt{Compress}\left(\left[U^i,\; Y^{i,1}, ...,\; Y^{i,i-1}\right],\; \tau\right)$

end for

Final Stage

$U = \texttt{Schrodinger-Solve}(V, \Delta t)$

$W^s = [L_1 V^s, L_2 V^s, ..., L_P V^s]$

for $i = 1, 2, ..., s$ **do**

  $Y^i = \texttt{Schrodinger-Solve}(\sqrt{b_i\Delta t}\; W^i, (1 - c_i)\; \Delta t)$

end for

$V' = \texttt{Compress}\left(\left[U,\; Y^1, ...,\; Y^s\right],\; \tau\right)$

Trace Normalization

$V' \leftarrow V'/\|V'\|_F$

Addition of low rank matrices via the SVD form

$$X_1 + X_2 = U_1 S_1 V_1^T + U_2 S_2 V_2^T$$

$$X_1 + X_2 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = Q_U R_U \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} R_V^T Q_V^T$$

$$X_1+X_2 = Q_U R_U \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} R_V^T Q_V^T = Q_U \tilde{U} \tilde{S} \tilde{V}^T Q_V^T \approx U S V^T = \sum_{k=1}^{r} u_k \sigma_k v_k^T$$

Truncation based on TOL in Frobenius norm

What is the benefit? The operations produces a new SVD at cost $\mathcal{O}(nr^2 + r^3)$ instead of $\mathcal{O}(n^3)$

VIRGINIA TECH

Oscillation Revival in the Jaynes-Cumming Model

Qubit + Cavity (m levels) Hamiltonian

$$H = \lambda(b\sigma^+ + b^\dagger\sigma^-).$$

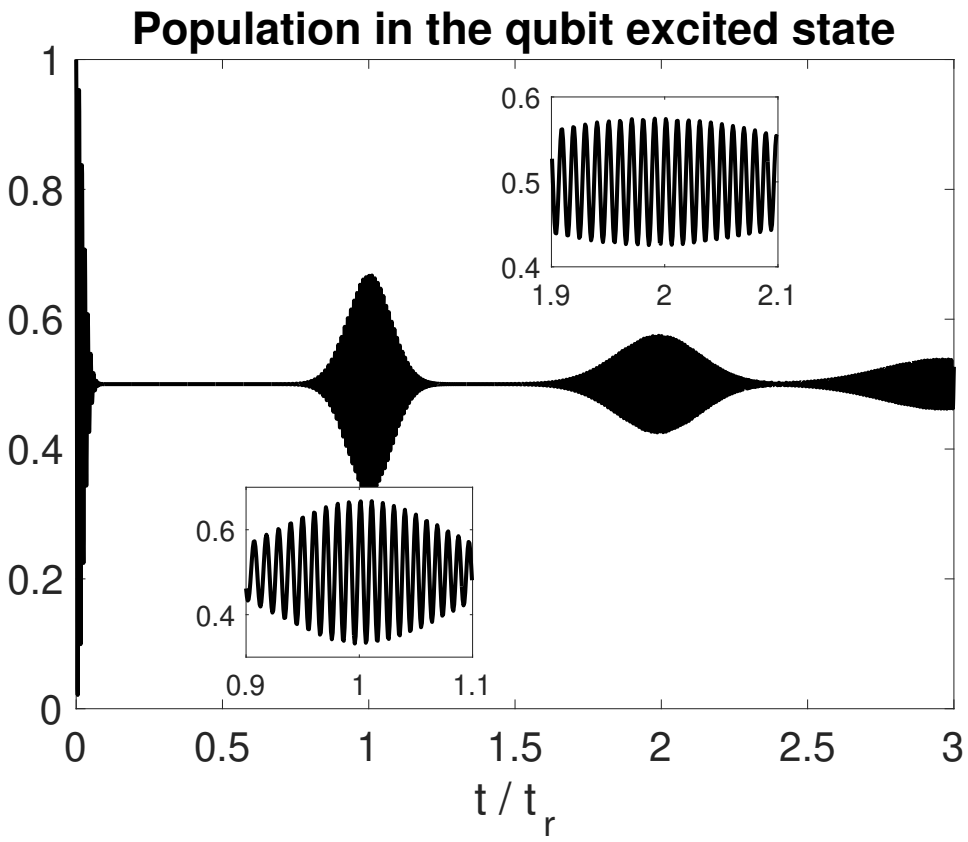
At the initial time the atom is in the excited state and the cavity is in the coherent state

$$\mathbf{v} \sim \sum_{n=0}^{m-1} \frac{|v|^n}{\sqrt{n!}}\mathbf{e}_n,$$

The initial density matrix is rank 1 with $\rho = VV^\dagger$

$$V = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \frac{\mathbf{v}}{\|\mathbf{v}\|}.$$

Here $m = 150$ so $N = 300$.

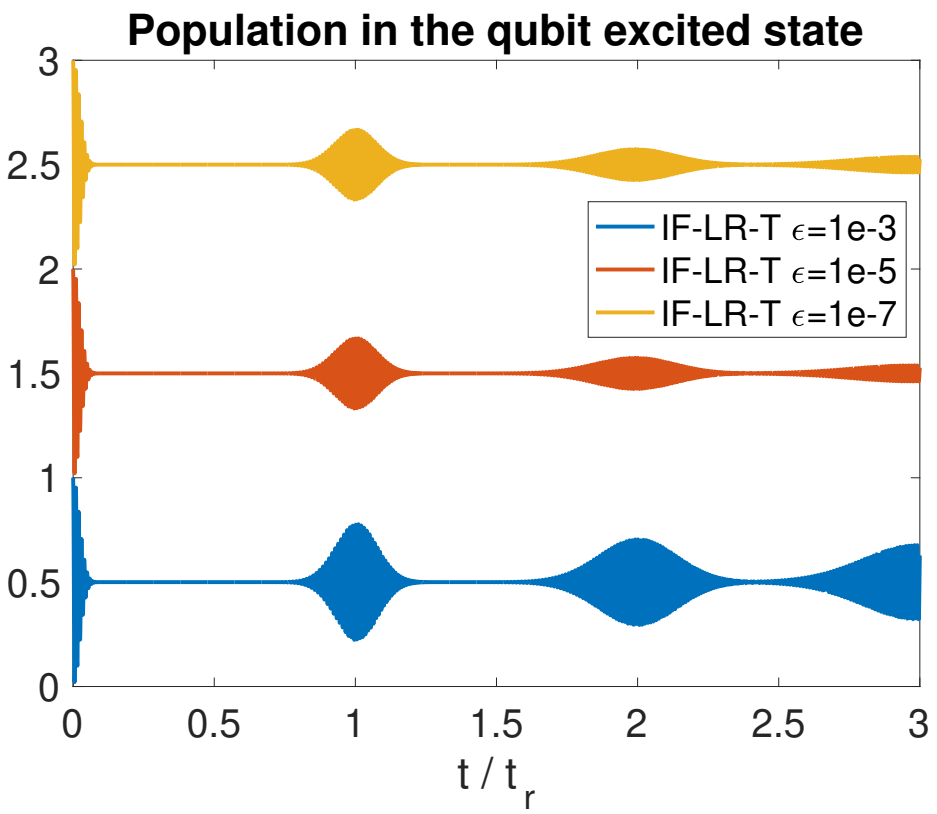


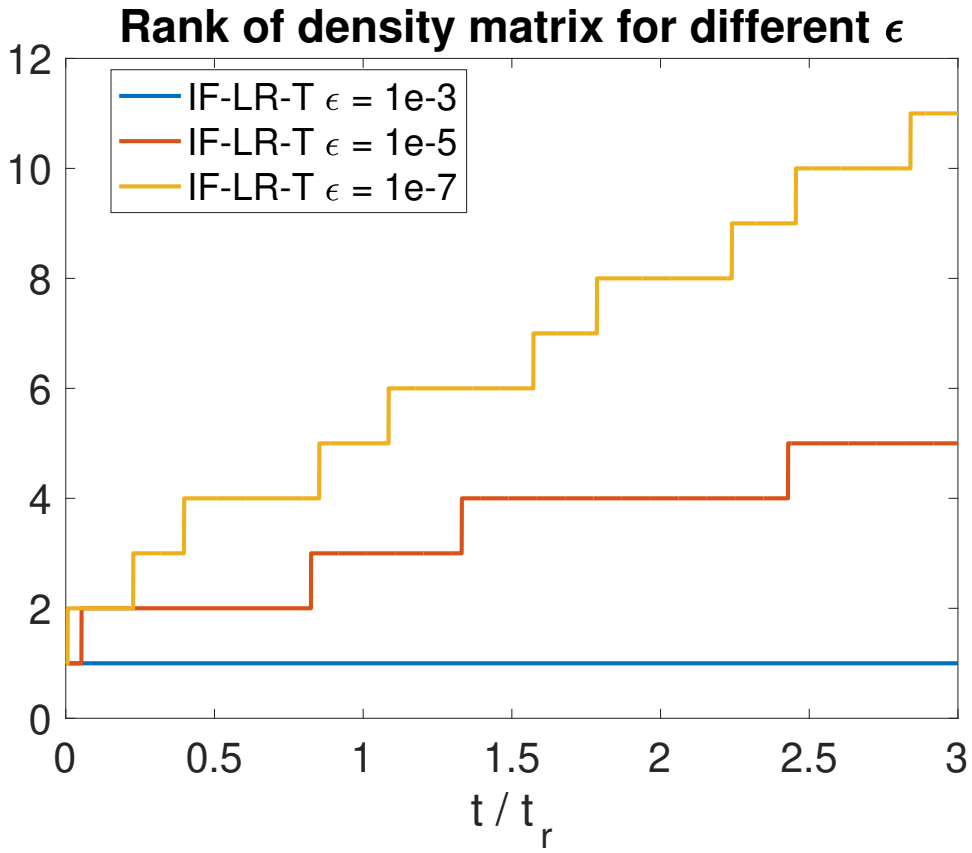
J. Gea-Banacloche, Atom- and field-state evolution in the Jaynes-Cummings model for large initial fields, Phys. Rev. A 44 (1991) 5913–5931.

VIRGINIA TECH

Oscillation Revival in the Jaynes-Cumming Model

Vary tolerance in trunc-sum, the absolute errors are around 10% and 0.1%.





VIRGINIA TECH

Tensor-train version of the method

- In principle we just add TT-
- In practice we must carefully design QR / SVD / summation.
- ODE solve is cheap.
- Trace is cheap.
- Compression is most costly and is accelerated by RNLA or cross approximation.

Tensors will be discussed in more detail on Friday.

Algorithm 4.1 Kraus is King integrator with tensor train (TT) compression

Input: Factor matrix $V \in \mathbb{C}^{N \times r}$ with columns v_i represented in TT format; Butcher Tableau $\mathcal{S} = (\mathbf{A}, \mathbf{b}, \mathbf{c})$; Step size Δt; Truncation tolerance ϵ.

Output: Factor $V' \in \mathbb{C}^{N \times r'}$ of $\rho(t + \Delta t)$

$\tau \leftarrow \tau \ / \ (3(s+1))$ ▷ Error tolerance per truncation within the scheme

Intermediate Stages

$V^0 := V$

for $i = 1, 2, ..., s$ **do**

  $U^i = \texttt{TT-Schrodinger-Solve}(V,\ c_i \Delta t,\ \tau)$

  $W^i = \texttt{TT-Compress-L}\left(\left[L_1 V^{i-1}, ..., L_P V^{i-1}\right],\ \tau\right)$

  for $j = 1, 2, ..., i-1$ **do**

    $Y^{ij} = \texttt{TT-Schrodinger-Solve}(\sqrt{a_{ij}\Delta t}\ W^j,\ (c_i - c_j)\Delta t,\ \tau/i)$

  end for

  $V^i = \texttt{TT-Compress}\left(\left[U^i, Y^{i,1}, ...,\ Y^{i,i-1}\right],\ \tau\right)$

end for

Final Stage

$U = \texttt{TT-Schrodinger-Solve}(V, \Delta t,\ \tau)$

$W^s = \texttt{TT-Compress-L}([L_1 V^s, L_2 V^s, ..., L_P V^s],\ \tau)$

for $i = 1, 2, ..., s$ **do**

  $Y^i = \texttt{TT-Schrodinger-Solve}(\sqrt{b_i \Delta t}\ W^i,\ (1 - c_i)\Delta t,\ \tau/s)$

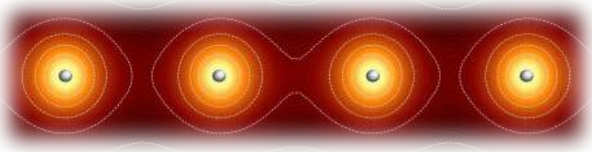
end for

$V' = \texttt{TT-Compress}\left(\left[U,\ Y^1, ...,\ Y^s\right],\ \tau\right)$

Trace Normalization

$V' \leftarrow V' / \|V'\|_F$

Dissipative Spin Chains



Spin-1/2 XX Heisenberg Chain

d magnetic spins arranged in a chain

Excitations (spin-up) propagate through nearest neighbors interactions

$$\text{Hilbert Space } \mathcal{H} = \mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_d \ , \quad \mathcal{H}_i = \text{Span}\left\{|\uparrow\rangle, |\downarrow\rangle\right\} \ , \quad |\uparrow\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ , \ |\downarrow\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\text{Hamiltonian } H = \sum_{i=1}^{d-1} (a_i^\dagger a_{i+1} + a_{i+1}^\dagger a_i) \ ,$$

a_j annihilates an excitation $\uparrow$ at site j and $a_i^\dagger$ creates one at site i

$$a_j^\dagger a_i = \cdots \otimes I \otimes \underset{\text{Site } j}{a^\dagger} \otimes I \otimes \cdots \otimes I \otimes \underset{\text{Site } i}{a} \otimes I \otimes \cdots$$

Lindbladian Dissipation

$$\text{Jump operators } L_i = \frac{1}{\sqrt{T}} a_i \ (i = 1, ..., d) \text{ dissipative excitations over time}$$

Operator Splitting for Spin Chains

Partition Hamiltonian into two sets of pairwise commuting terms

$$H = \sum_{i=1}^{d-1} H_{i,i+1} = \sum_{i \text{ odd}} H_{i,i+1} + \sum_{i \text{ even}} H_{i,i+1}$$
$$= (H_{12} + H_{34} + H_{56} + \cdots) + (H_{23} + H_{45} + H_{67} + \cdots)$$

$$H_{23} = I \otimes (a^\dagger \otimes a + a \otimes a^\dagger) \otimes I \otimes \cdots \otimes I \quad \text{for the Heisenberg model}$$

Matrix exponential of each set can efficiently represented in tensor format called a matrix product operator (MPO)

$$\Phi_{\text{odd}}(\Delta t) = \exp\left\{\Delta t \sum_{i \text{ odd}} H_{i,i+1}\right\} = \prod_{i \text{ odd}} e^{\Delta t H_{i,i+1}} = \bigotimes_{i \text{ odd}} \Phi_{i,i+1}(\Delta t)$$

$$\Phi^{\text{Heis}}_{i,i+1}(\Delta t) = e^{\Delta t(a^\dagger \otimes a + a \otimes a^\dagger)} \in \mathbb{C}^{4\times 4} \text{ acts on two adjacent TT cores}$$

Low-Rank Time Evolution

Length 25 Heisenberg model with dissipation timescale $T = 10$

Rank-1 initial state with 2 excitations

$$\rho(0) = |\psi_0\rangle\langle\psi_0| \ , \quad |\psi_0\rangle = |\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\uparrow\downarrow\downarrow\downarrow\downarrow\downarrow\uparrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\rangle$$

2nd order CPTP scheme based on midpoint rule, $\Delta t = 0.01$

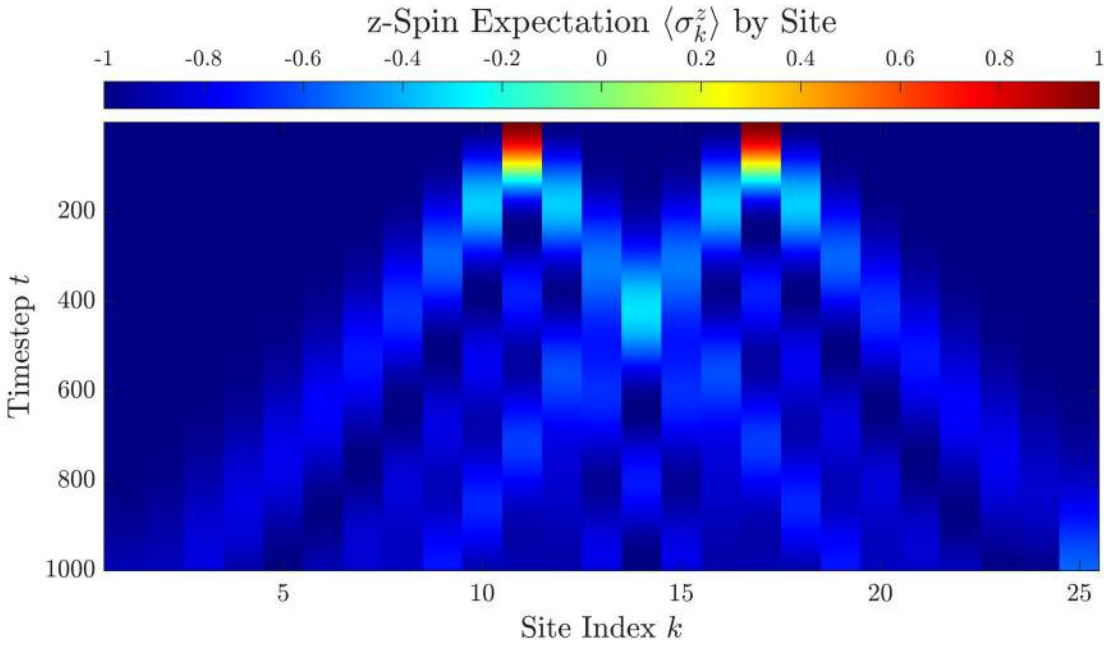
Tracking *z*-spin expectation by site to show dynamics of the excitation

$$\sigma^z = +1\,|\uparrow\rangle\langle\uparrow| + (-1)\,|\downarrow\rangle\langle\downarrow|$$

$$\langle\sigma_k^z\rangle = \mathrm{Tr}(\rho\sigma_k^z) = \sum_{i=1}^{\mathrm{rank}(V)} \langle v_i|\sigma_k^z|v_i\rangle$$

Excitations propagate, dissipate due to Lindblad dynamics

By $T = 10$, the total *z*-spin $\sum_k \langle\sigma_k^z\rangle$ has decreased by ~ e^{-1} (hard to see here)



- 64 qubits
- Hilbert space > 10^{19}
- The required memory in this simulation is
  < 3200 doubles x 21

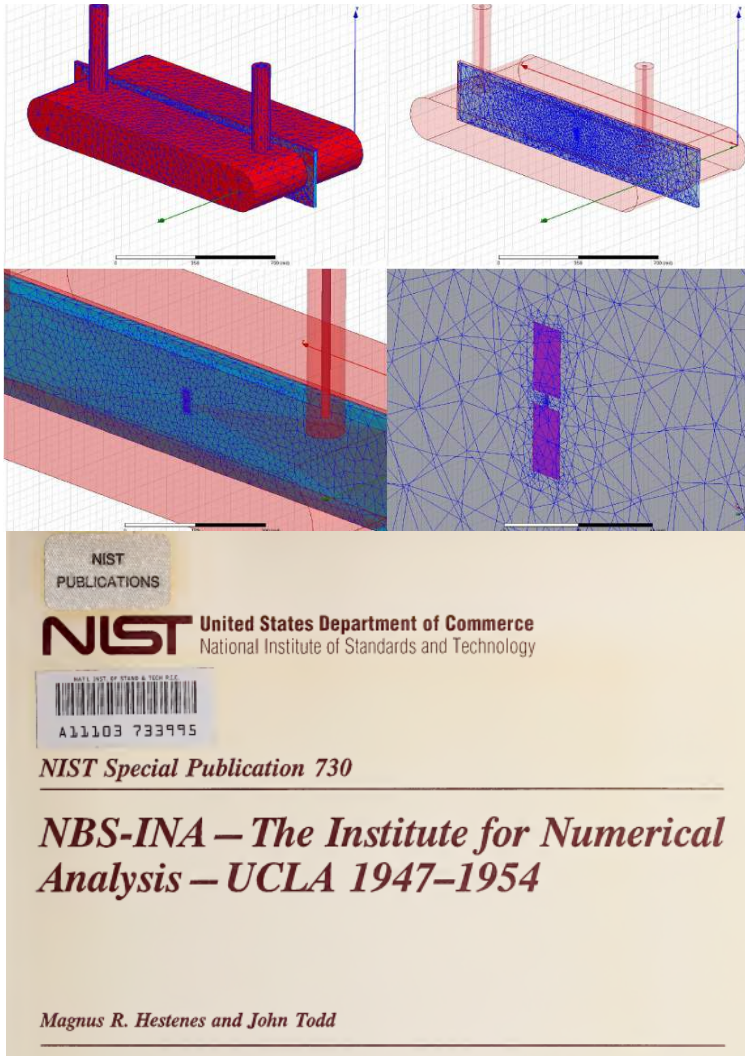
Heavy Hex Quantum Circuit

- 25 qubits with non-trivial coupling
- Hilbert space ~ 4×10^8
- Compare 190,000 doubles in low-rank
- 10h on a single node

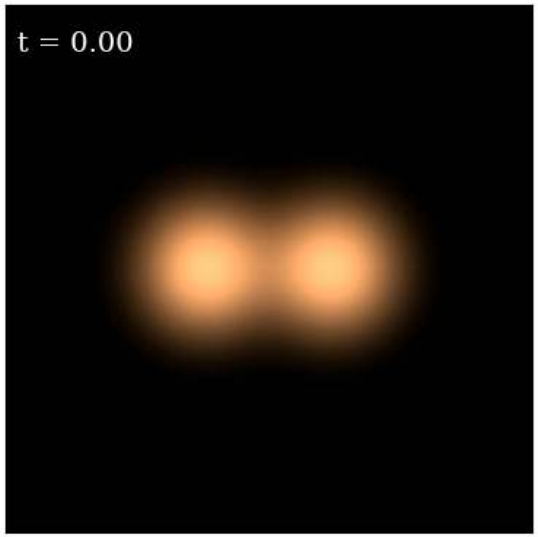
VIRGINIA TECH

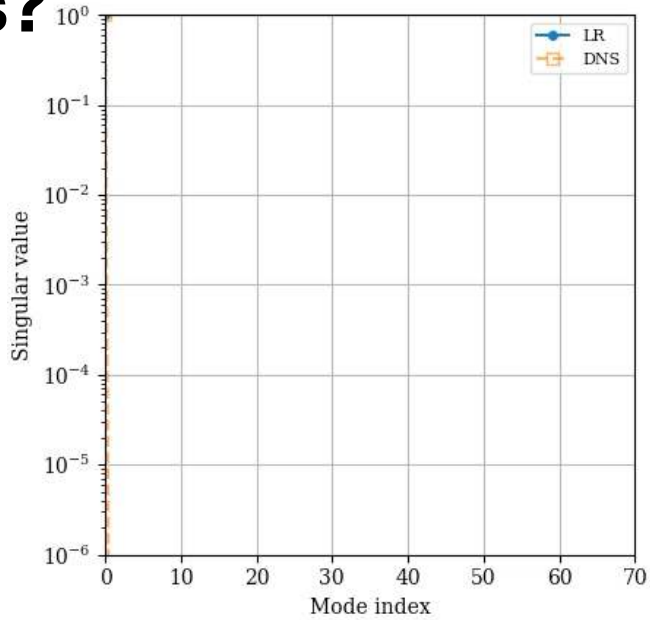
Summary

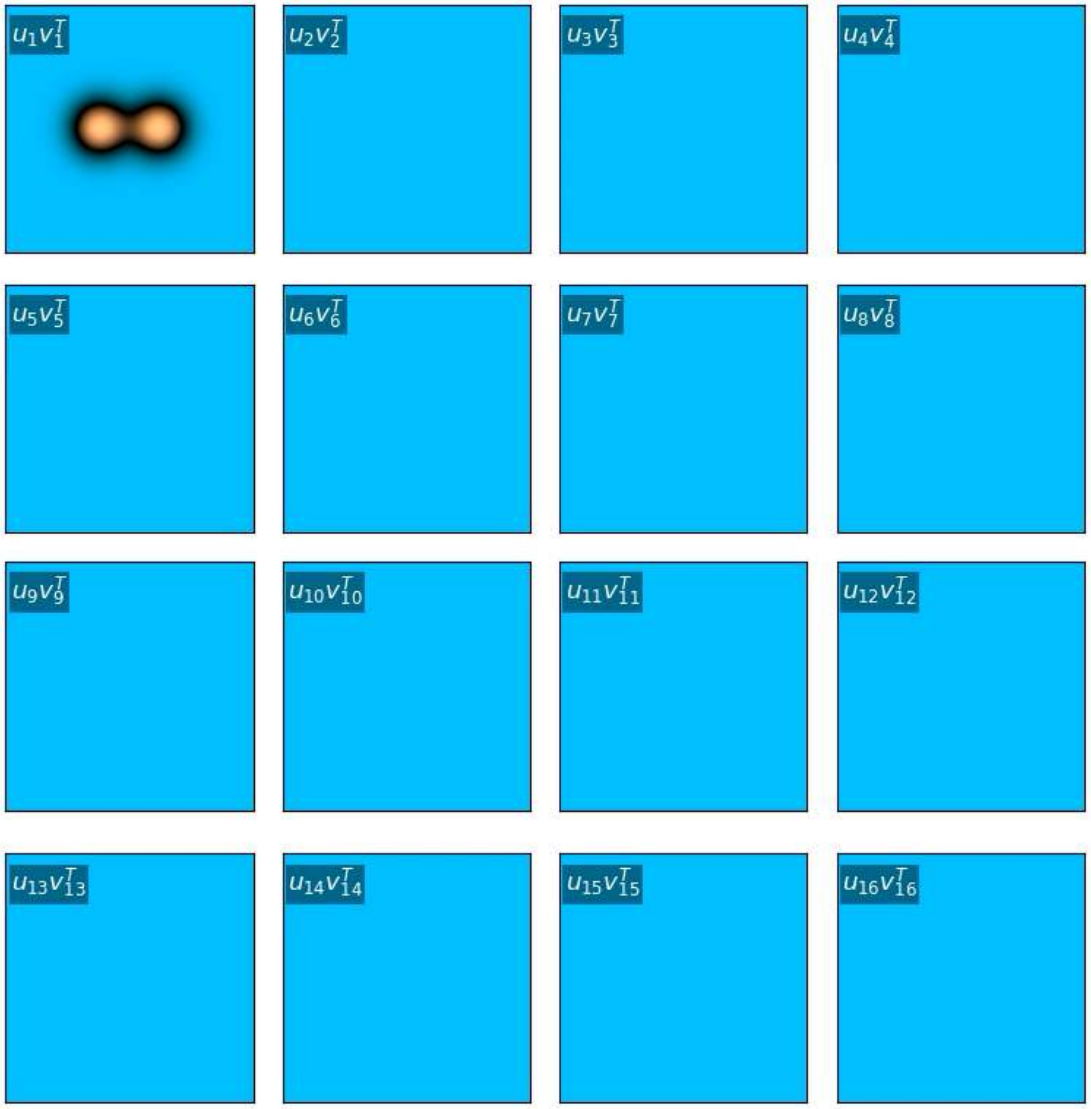
- Quantum computing is a rich area for modeling and can benefit from better numerical algorithms.
- Analysis of quantum algorithms has the same flavor as analysis of numerical methods (can you invent the next GMRES?).
- Better computational methods for EM needed as well.
- It is easy to simulate a few QUBITS but it becomes exponentially harder.
- Scaling up systems in a reliable way is an area where quantum digital twins can help.
- Let's join physicists, computer scientists and engineers to solve this!



Questions?


t = 0.00


LR
DNS
Singular value
Mode index
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
10
20
30
40
50
60
70


$u_1v_1^T$
$u_2v_2^T$
$u_3v_3^T$
$u_4v_4^T$
$u_5v_5^T$
$u_6v_6^T$
$u_7v_7^T$
$u_8v_8^T$
$u_9v_9^T$
$u_{10}v_{10}^T$
$u_{11}v_{11}^T$
$u_{12}v_{12}^T$
$u_{13}v_{13}^T$
$u_{14}v_{14}^T$
$u_{15}v_{15}^T$
$u_{16}v_{16}^T$

CPTP Nested Picard Iteration

Pretend for a second we have the equation

$$\frac{d\rho(t)}{dt} = J(t)\rho(t) + \rho(t)J^{\dagger}(t)$$

Then we had seen that we could solve for Cholesky factor instead

$$\frac{dV(t)}{dt} = J(t)V(t), \quad \Longleftrightarrow \quad V(t) = U(t,s)V(s)$$
$$V(t^0) = V_0,$$

The flow operator gives the solution for the density matrix

$$\rho(t) = U(t,s)\rho(s)U^{\dagger}(t,s)$$

VIRGINIA TECH

CPTP Nested Picard Iteration

We really want to solve this equation

$$\frac{d\rho(t)}{dt} = J(t)\rho(t) + \rho(t)J^\dagger(t) + \sum_\alpha \mathcal{L}_\alpha \rho(t) \mathcal{L}_\alpha^\dagger,$$

We can think of the last term as a forcing and use Duhamel's principle

$$\rho(t) = U(t,\tau)\rho(\tau)U^\dagger(t,\tau) + \int_\tau^t U(t,s)\mathcal{L}_L\rho(s)U^\dagger(t,s)\,ds$$

$$= U(t,\tau)\rho(\tau)U^\dagger(t,\tau) + \sum_\alpha \int_\tau^t [U(t,s)\mathcal{L}_\alpha]\rho(s)[U(t,s)\mathcal{L}_\alpha]^\dagger\,ds.$$

This is on Kraus form

VIRGINIA TECH

CPTP Nested Picard Iteration

Writing out the formula as a numerical method we find

$$\rho(t^{n+1}) = U(t^{n+1}, t^n)\rho(t^n)U^\dagger(t^{n+1}, t^n) + \int_{t^n}^{t^{n+1}} U(t^{n+1}, s)\mathcal{L}_L\rho(s)U^\dagger(t^{n+1}, s)\, ds.$$

Problem is that we don't know the solution inside the integral at future times!

The Δt in front of the integral allows us to raise the order by one in the Nested Picard Iteration

$$\rho^{n+1,k+1} := U^{(k+1)}(t^{n+1}, t^n)\rho^n U^{(k+1)}(t^{n+1}, t^n)^\dagger + \Delta t \sum_j \omega_j U^{(k)}(t^{n+1}, t^{n+c_j\Delta t})\mathcal{L}_L\rho^{n+c_j,k}U^{(k)}(t^{n+1}, t^{n+c_j\Delta t})^\dagger$$

VIRGINIA TECH

CPTP Nested Picard Iteration

As an example - here is the fourth order method in terms of the third order method

$$\rho^{n+1,4} := U^{(4)}(t^{n+1}, t^n)\rho^n U^{(4)}(t^{n+1}, t^n)^\dagger$$
$$+ \frac{\Delta t}{2}\Big(U^{(3)}(t^{n+1}, t^{n+c_1\Delta t})\mathcal{L}_L\rho^{n+c_1,3}U^{(3)}(t^{n+1}, t^{n+c_1\Delta t})^\dagger +$$
$$U^{(3)}(t^{n+1}, t^{n+c_2\Delta t})\mathcal{L}_L\rho^{n+c_2,3}U^{(3)}(t^{n+1}, t^{n+c_2\Delta t})^\dagger\Big)$$

If the last flow is done by an implicit method the overall method appears to be unconditionally stable!

We can prove this for a test equation and observe in general.

VIRGINIA TECH

Optimal Control

$$i\hbar\frac{d\psi}{dt} = H\psi = (H_S + H_c(t))\psi$$

$$H_s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \omega_{0,1} & 0 & 0 \\ 0 & 0 & \omega_{0,1}+\omega_{1,2} & 0 \\ 0 & 0 & 0 & \omega_{0,1}+\omega_{1,2}+\omega_{2,3} \end{pmatrix}$$

$$H_c(t) = f(t)(a + a^\dagger)$$

$$f(t) = 2\,\mathrm{Re}\{e^{i\omega_d t}d(t)\} = 2I(t)\cos(\omega_d t) + 2Q(t)\sin(\omega_d t).$$

Finding the frequencies in blue is an inverse or inference problem

Given a target gate U_{Target} (a unitary matrix), finding I(t) and Q(t) so that

$$\psi(0) = \mathbf{e}_k \Rightarrow \psi(T) \approx U_{\text{Target}}[:, k], \forall k$$

Is the optimal control problem