

Quantum Optimal Control via Semi-Automatic Differentiation

— Optimizing Metrological Devices with Memory-Efficient Automatic Differentiation

Michael H. Goerz, Sebastián C. Carrasco, Vladimir S. Malinovsky

DEVCOM Army Research Lab, Adelphi, MD 20783, USA



① Gradient-Based Optimal Control

Optimization Functional

$$J(\{\epsilon_{nl}\}) = J_T(\{|\psi_k(T)\rangle\}) + \int_0^T g_a(\{\epsilon_l(t)\}, t) dt + \int_0^T g_b(\{|\psi_k(t)\rangle\}, t) dt$$

For example, square modulus gate fidelity:

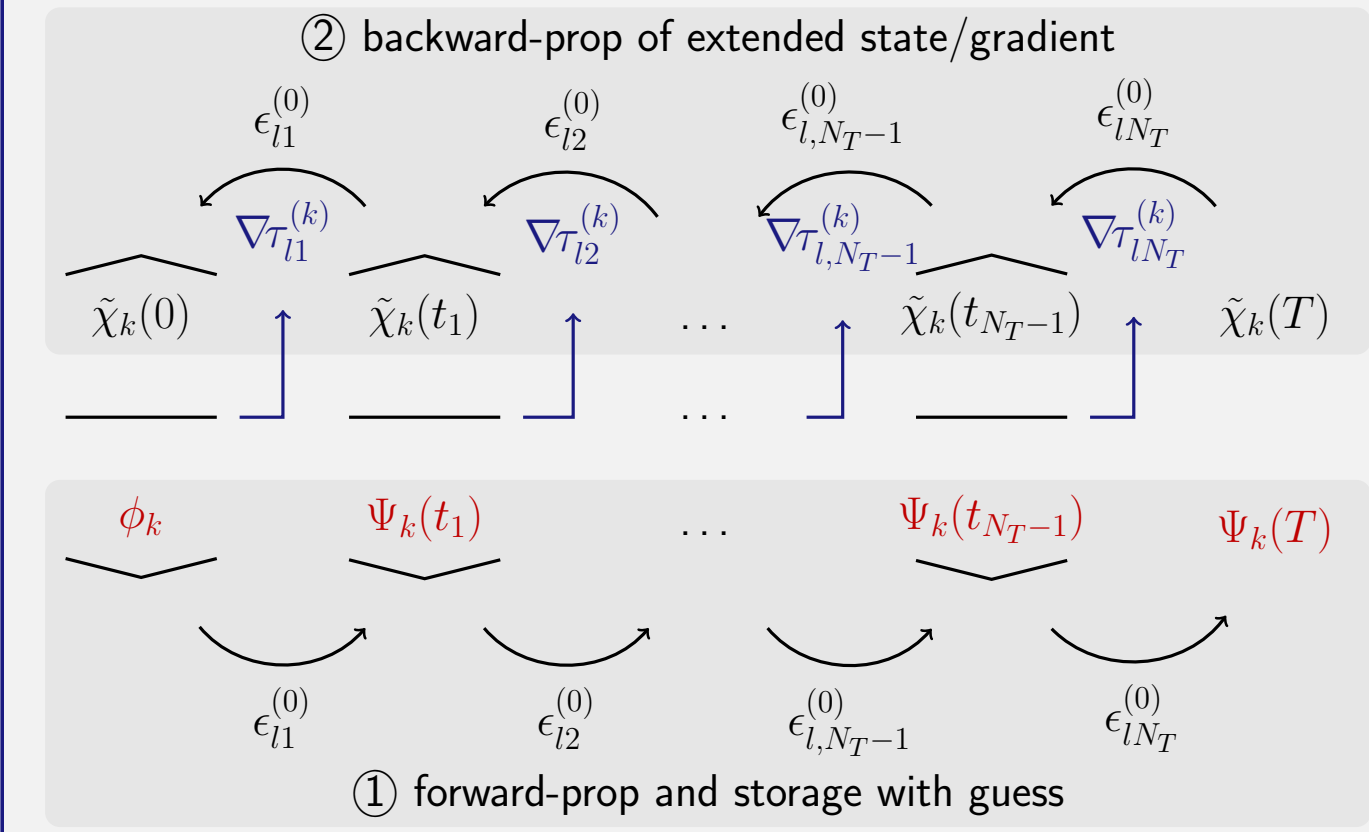
$$J_T = J_{T,sm} = 1 - \left| \frac{1}{N} \sum_{k=1}^N \langle \phi_k^{tgt} | \hat{U} | \phi_k \rangle \right|^2$$

gradient of complex overlap:

$$\nabla_{\tau_{ln}}^{(k)} = \frac{\partial}{\partial \epsilon_{nl}} \langle \phi_k^{tgt} | \hat{U}_{N_T} \dots \hat{U}_n \dots \hat{U}_1 | \phi_k \rangle = \left\langle \chi_k(t_n) \left| \frac{\partial \hat{U}_n}{\partial \epsilon_{nl}} \right| \psi_k(t_{n-1}) \right\rangle$$

GRAPE

— Khaneja *et. al.* J. Mag. Res. 172, 296 (2005)



Update in direction of the gradient

$$(\nabla J)_{ln} \equiv \frac{\partial J}{\partial \epsilon_{ln}}$$

- first order: $\epsilon_{nl}^{(i+1)} = \epsilon_{nl}^{(i)} - \alpha (\nabla J)_{ln}$
- second order: L-BFGS-B (or other optimizer)

backward propagation of *target states*

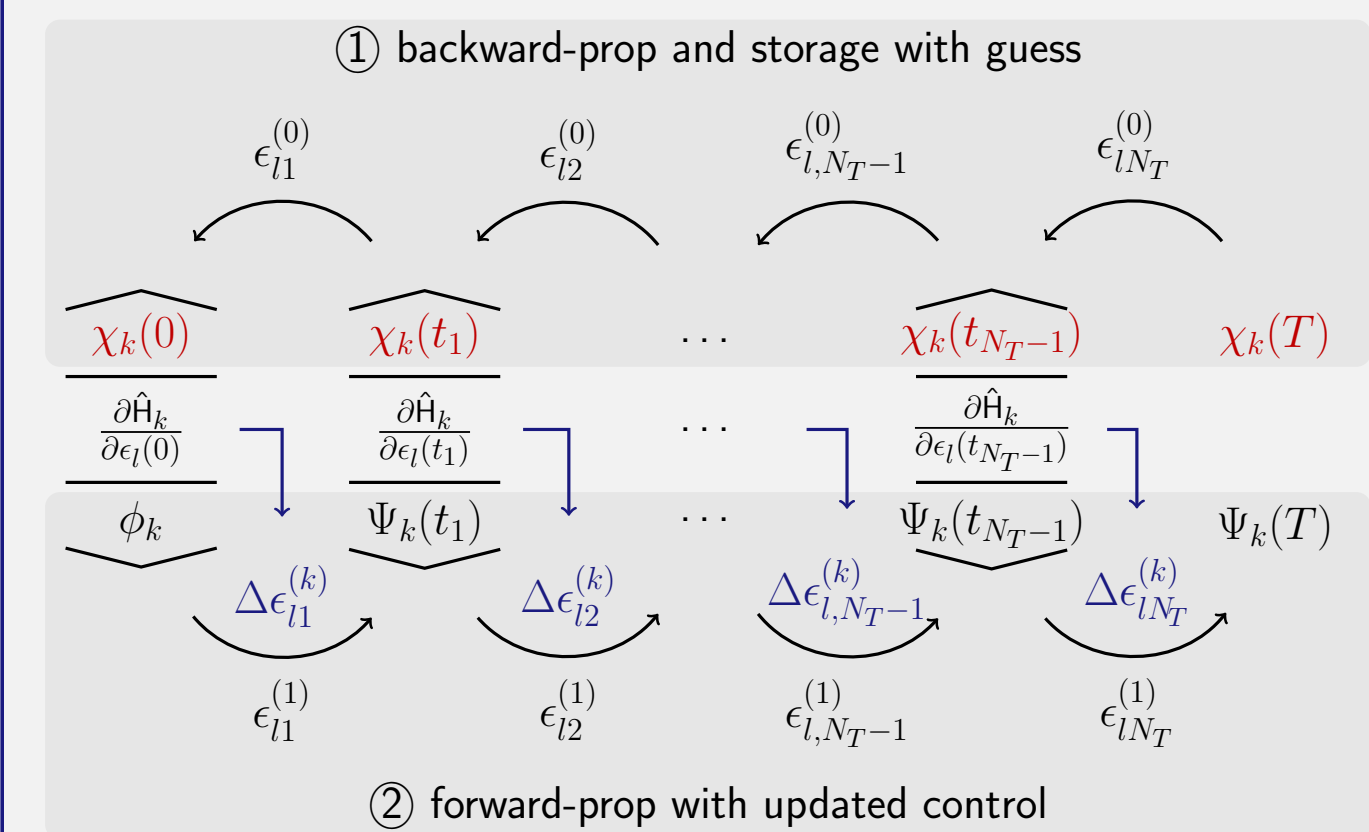
Gradient Evaluation

— Goodwin, Kuprov, J. Chem. Phys. 143, 084113 (2015)

$$\begin{pmatrix} \frac{\partial \hat{U}_n^\dagger}{\partial \epsilon_{nl}} |\chi_k(t_n)\rangle \\ \vdots \\ \frac{\partial \hat{U}_1^\dagger}{\partial \epsilon_{nl}} |\chi_k(t_n)\rangle \end{pmatrix} = \exp \left[-i \begin{pmatrix} \hat{H}_n^\dagger & 0 & \dots & 0 & \hat{H}_n^{(1)\dagger} \\ 0 & \hat{H}_n^\dagger & \dots & 0 & \hat{H}_n^{(2)\dagger} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \hat{H}_1^\dagger & \hat{H}_1^{(L)\dagger} \\ 0 & 0 & \dots & 0 & \hat{H}_1^\dagger \end{pmatrix} dt_n \right] \begin{pmatrix} 0 \\ \vdots \\ 0 \\ |\chi_k(t_n)\rangle \end{pmatrix}$$

Krotov's method

— Reich *et al.* J. Chem. Phys. 136, 104103 (2012)



construction of $\Delta \epsilon(t)$ for monotonic convergence

$$\Delta \epsilon_l(t) \propto \sum_{k=1}^N \left\langle \chi_k^{(0)}(t) \left| \frac{\partial \hat{H}}{\partial \epsilon_l(t)} \right| \psi_k^{(1)}(t) \right\rangle$$

$$|\chi_k^{(0)}(T)\rangle = -\frac{\partial J_T}{\partial \langle \psi_k^{(0)}(T) |}$$

sequential update!

References

- JuliaQuantumControl Organization: — <https://github.com/JuliaQuantumControl>
- GRAPE.jl: Gradient-ascent pulse engineering with Semi-AD
 - Krotov.jl: Krotov's method with Semi-AD

[1] Goerz *et al.* arXiv:2205.15044 (2022)

[3] Watts *et al.* Phys. Rev. A 91, 062306 (2015)

[2] Carrasco *et al.* arXiv:2201.01744 (2022)

[4] Goerz *et al.* Phys. Rev. A 91, 062307 (2015)

② Semi-Automatic Differentiation

Automatic Differentiation

— Leung *et al.* Phys. Rev. A 95, 042318 (2017)

forward mode “tangent”

$$\dot{v}_j \equiv \frac{\partial v_j}{\partial \epsilon_n}$$

$$\dot{v}_j = \sum_i \frac{\partial v_j}{\partial v_i} \dot{v}_i$$

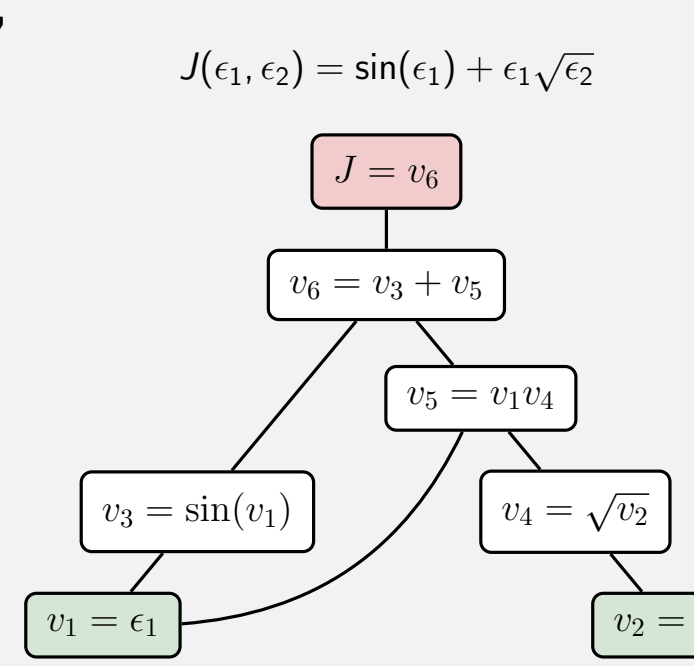
sum over all v_i on which v_j depends

backward mode “adjoint”

$$\bar{v}_j \equiv \frac{\partial J}{\partial v_j}$$

$$\bar{v}_j = \sum_i \bar{v}_i \frac{\partial v_i}{\partial v_j}$$

sum over all v_i which depend on v_j



Problems with backward mode:

- Need to store adjoint for every node in comput. graph
- Need to implement adjoint for every operation (sparse, complex, etc.)

Solution: Semi-Automatic Differentiation [1]

$$J(\{\epsilon_{nl}\} \rightarrow J(\{u_j\}) \Rightarrow (\nabla J)_{nl} \equiv \frac{\partial J}{\partial \epsilon_{ln}} = \sum_j \frac{\partial J}{\partial u_j} \frac{\partial u_j}{\partial \epsilon_{ln}}$$

1. Overlap Fidelities: $J(\{\epsilon_{nl}\} \rightarrow J(\{\tau_k\})$

$$\Rightarrow (\nabla J_T)_{nl} = 2\text{Re} \sum_k \frac{\partial J_T}{\partial \tau_k} \frac{\partial \tau_k}{\partial \epsilon_{ln}} \text{ with Wirtinger deriv. } \frac{\partial J_T}{\partial \tau_k} \equiv \frac{1}{2} \left(\frac{\partial J_T}{\partial \text{Re}[\tau_k]} - i \frac{\partial J_T}{\partial \text{Im}[\tau_k]} \right)$$

Use AD to evaluate Wirtinger derivative (Zygote gradient up to factor 2)

2. State Functionals: $J(\{\epsilon_{nl}\} \rightarrow J(\{|\psi_k\rangle\})$

$$\Rightarrow (\nabla J_T)_{nl} = 2\text{Re} \sum_k \left(\frac{\partial J_T}{\partial |\psi_k(T)\rangle} \frac{\partial |\psi_k(T)\rangle}{\partial \epsilon_{nl}} \right) = -2\text{Re} \sum_k \frac{\partial}{\partial \epsilon_{nl}} \langle \chi_k(T) | \psi_k(T) \rangle$$

$$\text{with } |\chi_k(T)\rangle \equiv -\frac{\partial J_T}{\partial \langle \psi_k(T) |}$$

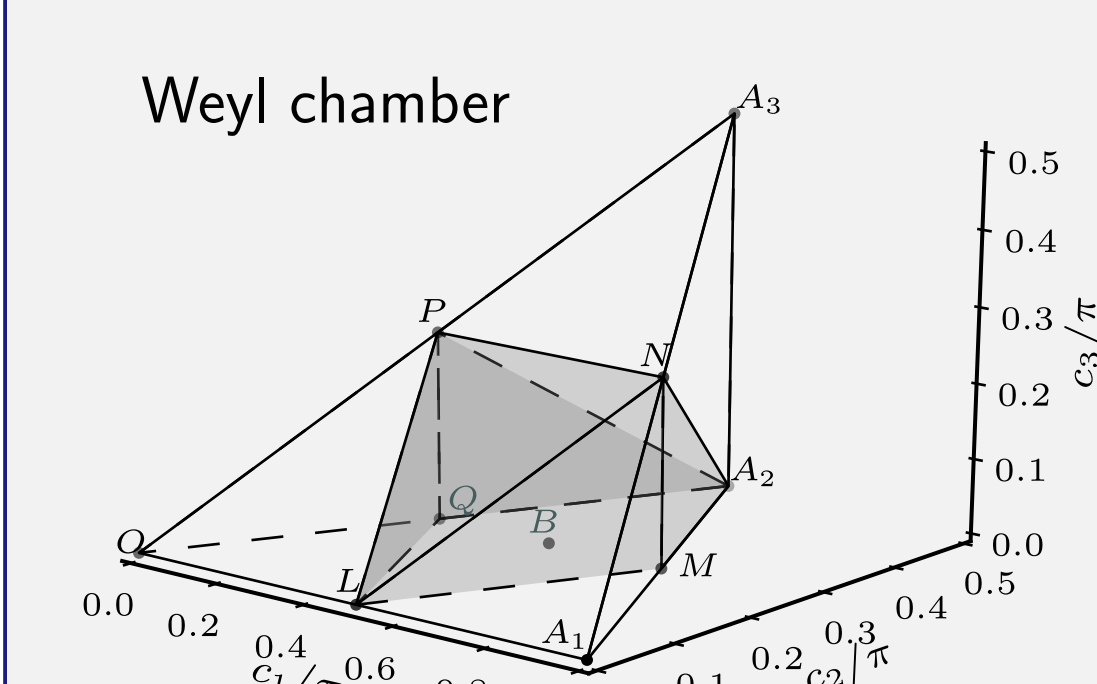
3. Gate Functionals: $J(\{\epsilon_{nl}\} \rightarrow J(\{|\psi_k\rangle\}) \rightarrow J(\hat{U}_L)$

$$\text{Use } (\hat{U}_L)_{ij} = \langle \phi_i | \psi_j(T) \rangle \Leftrightarrow (\hat{U}_L)_{ij}^* = \langle \psi_j(T) | \phi_i \rangle$$

$$\Rightarrow |\chi_k\rangle = -\sum_{ij} \left(\frac{\partial J_T}{\partial (U_L)_{ij}} \frac{\partial (U_L)_{ij}}{\partial \langle \psi_k(T) |} + \frac{\partial J_T}{\partial (U_L)_{ij}^*} \frac{\partial (U_L)_{ij}^*}{\partial \langle \psi_k(T) |} \right) = -\frac{1}{2} \sum_i (\nabla_{U_L} J_T)_{ik} |\phi_i\rangle$$

③ Optimizing Entangling Quantum Gates

Weyl chamber



Cartan decomposition

$$\hat{U}_L = \hat{k}_1 \exp \left[\frac{i}{2} (c_1 \hat{\sigma}_x \hat{\sigma}_x + c_2 \hat{\sigma}_y \hat{\sigma}_y + c_3 \hat{\sigma}_z \hat{\sigma}_z) \right] \hat{k}_2$$

Gate concurrence — not analytical

$$C(\hat{U}_L) = \max |\sin(c_{1,2,3} \pm c_{3,1,2})|$$

PE functional [3, 4] — analytical

$$D_{PE}(\hat{U}_L) = g_3 \sqrt{g_1^2 + g_2^2} - g_1$$

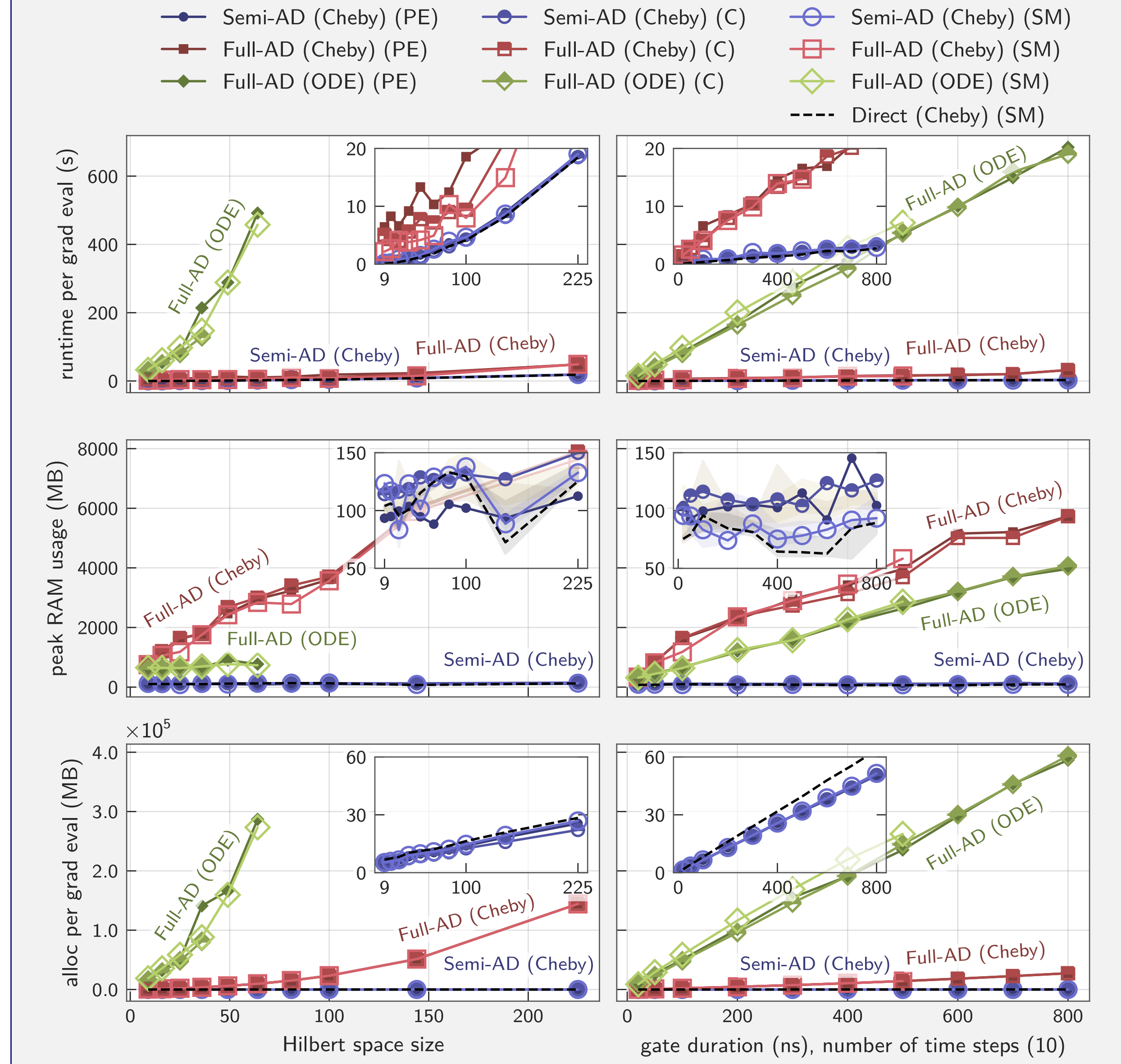
$$g_1 = \frac{1}{16} \text{Re} [\text{tr}^2(\hat{m})], \quad g_2 = \frac{1}{16} \text{Im} [\text{tr}^2(\hat{m})], \quad g_3 = \frac{1}{4} [\text{tr}^2(\hat{m}) - \text{tr}(\hat{m}^2)]$$

with $\hat{m} = \hat{U}_B^T \hat{U}_B$. $\hat{U}_B$ is $\hat{U}_L$ in Bell basis.

④ Benchmarks

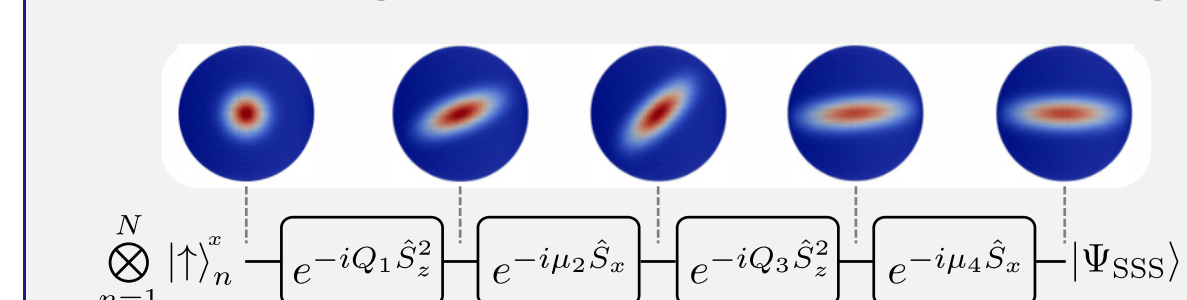
Compare runtime and RAM usage for perfect entangler optimization on two transmons [1]

- Full automatic differentiation vs Semi-automatic differentiation
- Chebyshev propagator vs ODE Solver



⑥ Metrology & Quantum Fisher Information

Generating Extreme Spin Squeezing [2]



- Generation of optimal squeezed states for Ramsey interferometry
- Extreme-SSS are eigenstates of

$$\hat{H} = \chi \hat{S}_z^2 - \Omega \hat{S}_x$$

- Alternate OAT squeezing pulses and rotations
- Reaches Heisenberg scaling

Optimize for contrast loss due to photon scattering and dephasing $\Rightarrow$ open system dynamics

Quantum Fisher Information

— Ma *et. al.* Phys. Rep. 509, 89 (2011)

$$F(|\psi\rangle) = 4 \left(\text{Re} \langle \hat{S}_z^2 \rangle_\psi - \text{Re} \langle \hat{S}_z \rangle_\psi^2 \right) \quad \text{closed system}$$

$$F(\hat{\rho}) = \sum_{i \neq j} \frac{2(p_i - p_j)^2}{p_i + p_j} \left| \langle \phi_i | \hat{S}_z | \phi_j \rangle \right|^2 \quad \text{open system}$$

where $p_i, |\phi_i\rangle$ are eigenvalues / eigenstates of $\hat{\rho}$ — non-analytic!