

Rotational Tractor Atom Interferometry with the QuantumControl.jl Framework



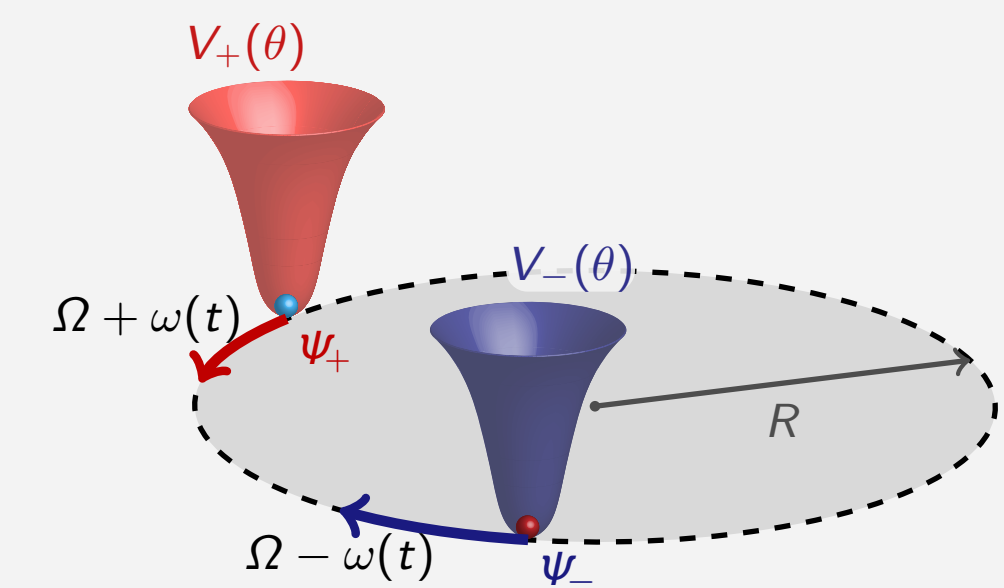
Michael H. Goerz¹, B. Dash², A. Duspayev², S. C. Carrasco¹, G. Raithel², and V. S. Malinovsky¹

¹DEVCOM Army Research Laboratory, Adelphi, MD 20783 ²Department of Physics, University of Michigan, Ann Arbor, MI 48109

① Tractor Atom Interferometry [1]

Uninterrupted three-dimensional confinement and guidance (1) guarantees recombination, (2) allows arbitrary holding times and complex trajectory patterns, and (3) limits dispersion.

Rotation sensing [2]



$$H_{\pm}(\theta, t) = -\frac{\hbar^2}{2M}\frac{\partial^2}{\partial\theta^2} + V_0 \cos(m(\theta + \phi_{\pm}(t)))$$

$$\phi_{\pm}(t) = \int_0^t \omega_{\pm}(t') dt' = \int_0^t (\Omega \pm \omega(t')) dt'$$

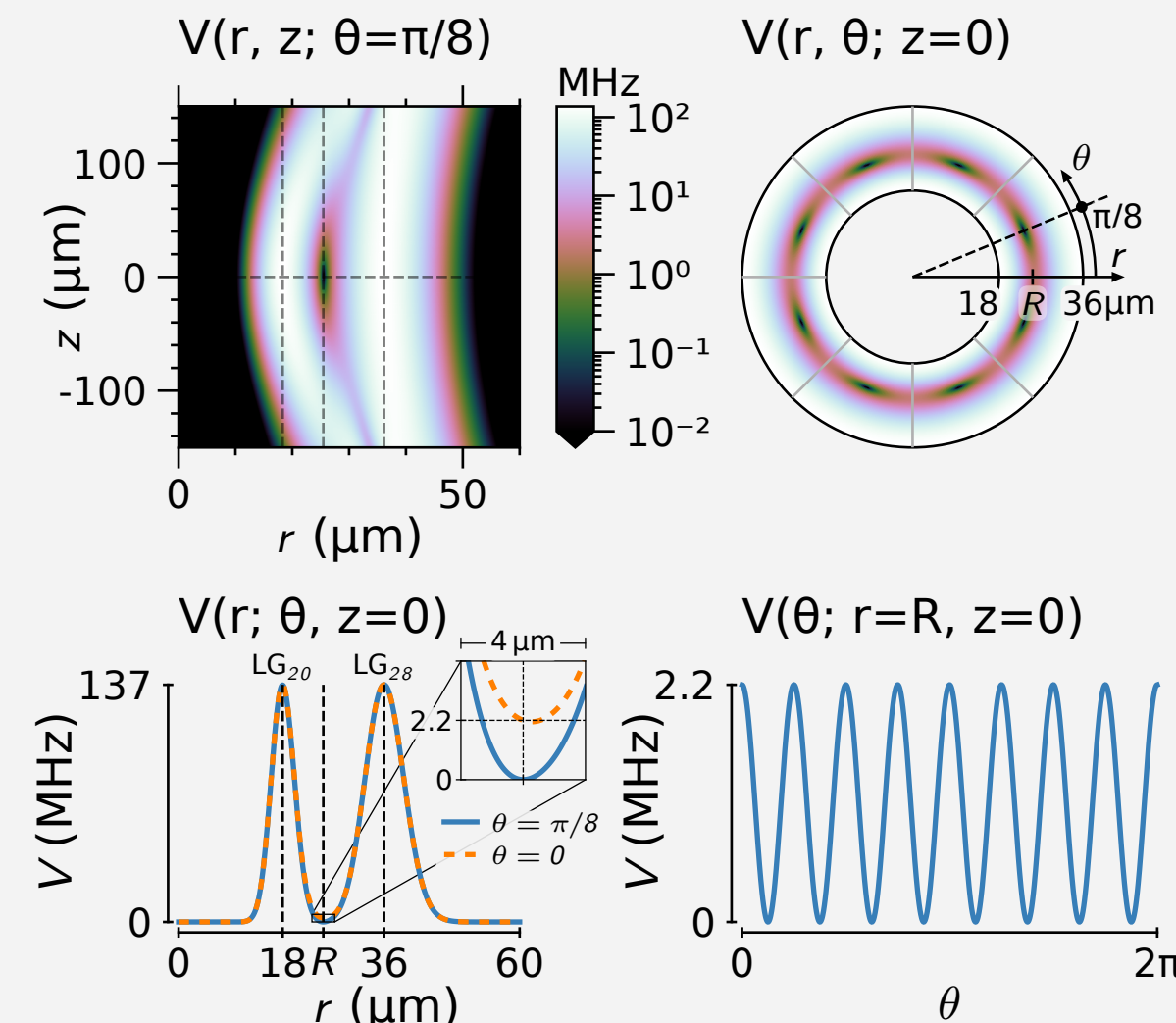
for “spin” states in Rb: $|+\rangle = |5S_{1/2}, F=1, m_F=0\rangle$; $|-\rangle = |5S_{1/2}, F=2, m_F=0\rangle$

Goal: measure background rotation Ω

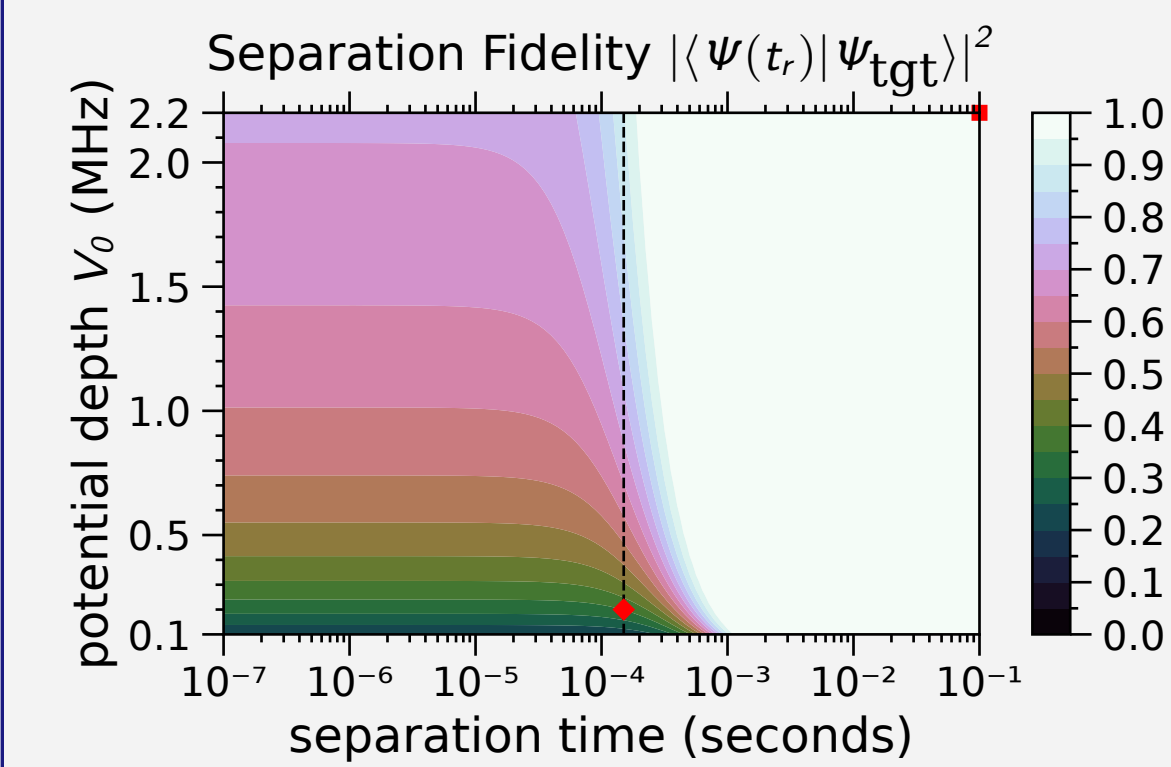
Transformation to “mixed frame” — position relative to moving potentials

$$\tilde{H}_{\pm}(t) = -\frac{\hbar^2}{2M}\frac{\partial^2}{\partial\theta^2} + V_0 \cos(m\theta) - i\hbar\omega_{\pm}(t)\frac{\partial}{\partial\theta}$$

Counter-rotating pinwheel optical lattices [3]



③ Non-Adiabatic Operation



For $t_r < 1$ ms and weak trapping potential:

- $\Psi_0(\theta)$ stays frozen in lab frame
- Wavepackets oscillate with ω_0 during constant rotation t_{loop} time
- After de-acceleration, wave packets are displaced relative to ground state ($\eta < 1$)

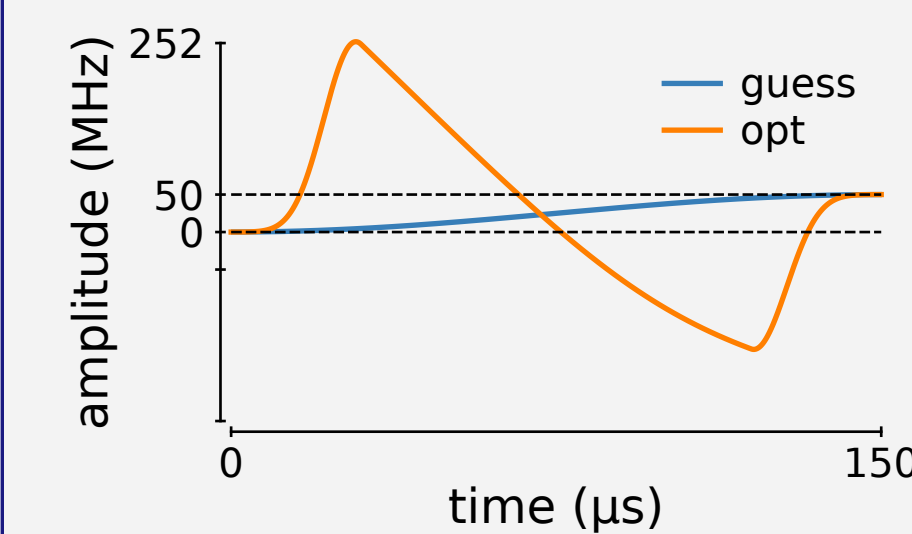
Recombined population for non-adiabatic evolution:

$\Rightarrow$ reduced contrast

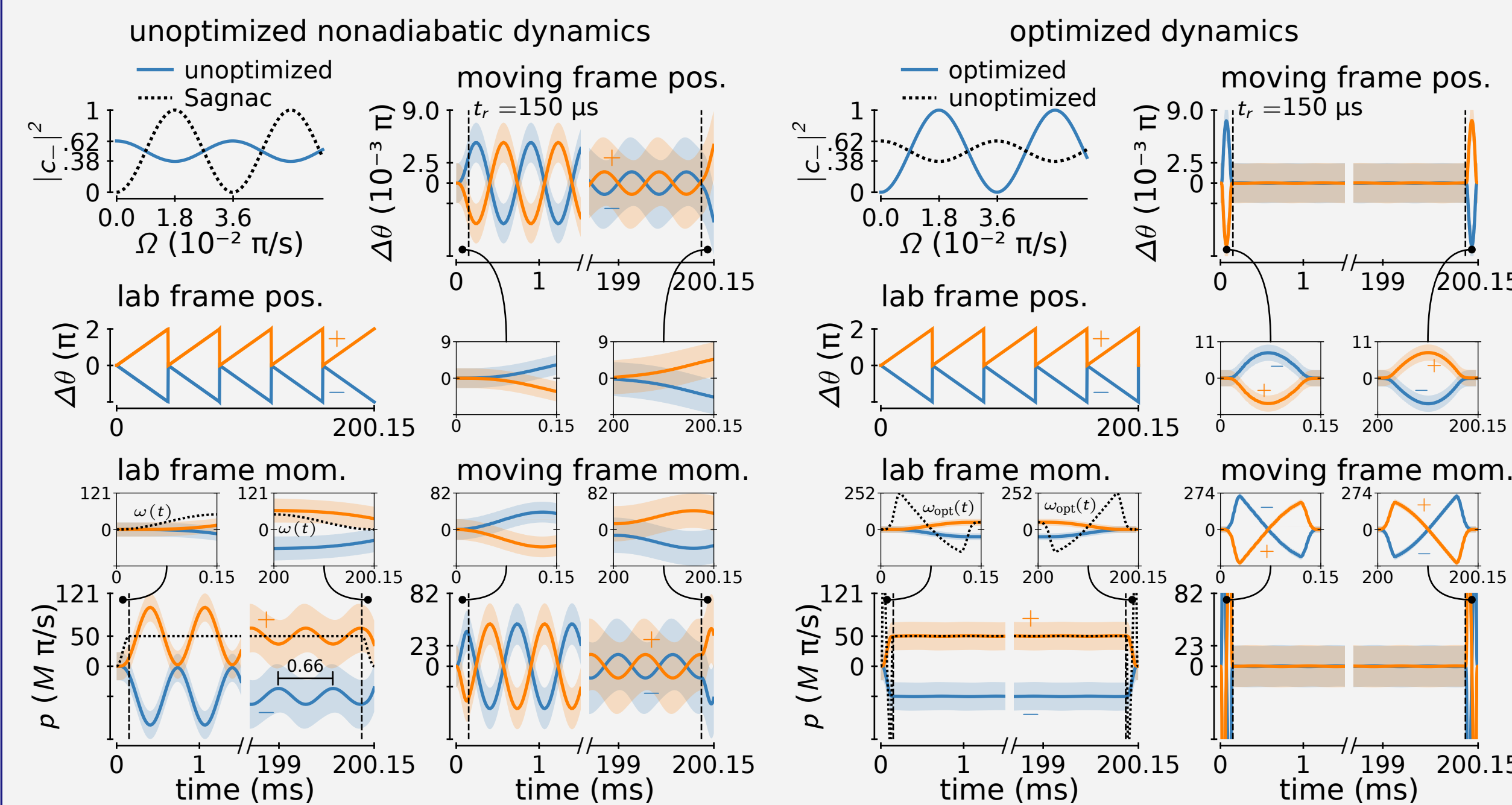
$$|c_{\pm}|^2 = \frac{1}{2} \pm \frac{1}{2} \text{Re} \left[\eta e^{-i\Delta\phi} \right]; \quad \eta = \langle \Psi_-(\theta, T) | \Psi_+(\theta, T) \rangle$$

④ Optimal Control

- Use Krotov’s methods to optimize for reaching ground state of rotating potential at $t = t_r$
- Parametrized “guided” amplitude $\omega_{\text{opt}}(t) = \omega(t) + S(t)\delta\omega(t)$; control only $\delta\omega(t)$
- De-acceleration flips acceleration pulse



- “kick and catch” movement of potential
- intermittently switches rotational direction!
- shorter $t_r \rightarrow$ higher amplitudes
- separation time limited only by feasible amplitude



⑤ QuantumControl.jl Framework

- JuliaQuantumControl Organization
— <https://github.com/JuliaQuantumControl>
- QuantumPropagators.jl: Simulate time dynamics (Chebychev, Newton, etc.)
 - GRAPE.jl: Gradient-ascent pulse engineering with Semi-AD
 - Krotov.jl: Krotov’s method with Semi-AD
 - QuantumControl.jl: High-level framework for dynamics and control

⑥ Design Principles

Generators: No assumptions on structure of Hamiltonian

$$\hat{H} = \hat{H}_0 + \sum_l \frac{\text{control amplitude}}{\text{control function}} \hat{H}_l$$

- Built-in “generator” hamiltonian/liouvillian:

$$\hat{H} = \hat{H}_0 + \sum_l \frac{\text{control amplitude}}{\text{control function}} \hat{H}_l$$

- “Control amplitude” on top of “control functions”: noise, non-linear coupling, “guided” controls, etc.

Interfaces: Implementation in terms of abstract interfaces only

```
@test check_generator(generator, state, tlist,
    for mutable_state=true, for immutable,
    for_expval=true, atol=1e-15)

verifies the given generator:
• get_controls(generator) must be defined and return a tuple
• all controls returned by get_controls(generator) must pass
  check_control
• evaluate(generator, tlist, n) must return a valid operator
  (check_operator), with forwarded keyword arguments (including
  for_expval)
• evaluate(top, generator, tlist, n) must be defined
• substitute(generator, replacements) must be defined
```

Multiple Dispatch: Efficient custom data structures, dynamically compiled

- Project-specific Hamiltonian, states, etc.
- GPU computing, automatic differentiation, ...

$\Rightarrow$ Separation of layers:

- control algorithms (QuantumControl.jl),
- time propagation (QuantumPropagators.jl),
- model representation (Julia ecosystem)

Semi-Automatic Differentiation [4]

Optimize any computable functional by writing functional as explicitly dependent on the states $\Rightarrow$ Complex matrix-calculus chain rule:

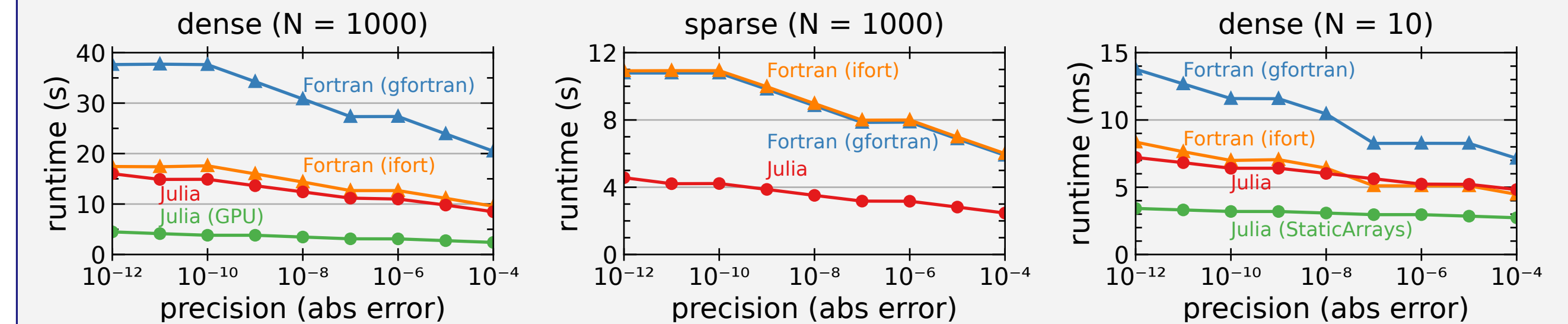
$$\nabla J_T = \frac{\partial J_T(\{\Psi_k(T)\})}{\partial \epsilon_{nl}} = 2 \text{Re} \left[\sum_k \frac{\partial J_T}{\partial |\Psi_k(T)\rangle} \frac{\partial |\Psi_k(T)\rangle}{\partial \epsilon_{nl}} \right]$$

$$= 2 \text{Re} \left[\sum_k \frac{\partial}{\partial \epsilon_{nl}} \langle \chi_k(T) | \Psi_k(T) \rangle \right]$$

- Use automatic differentiation for $\chi = \frac{\partial J_T}{\partial |\Psi_k(T)\rangle}$ (or further simplify via U_{ij}, τ_i)
- Standard GRAPE scheme for $\frac{\partial}{\partial \epsilon_{nl}} \langle \chi_k(T) | \Psi_k(T) \rangle \rightarrow$ no computational overhead from AD

⑦ Performance

Propagation with Chebychev propagator over 1000 time steps (randomized pulses)



- Matches Fortran (ifort) performance out of the box
- Better performance for specialized data structures (multiple dispatch!)

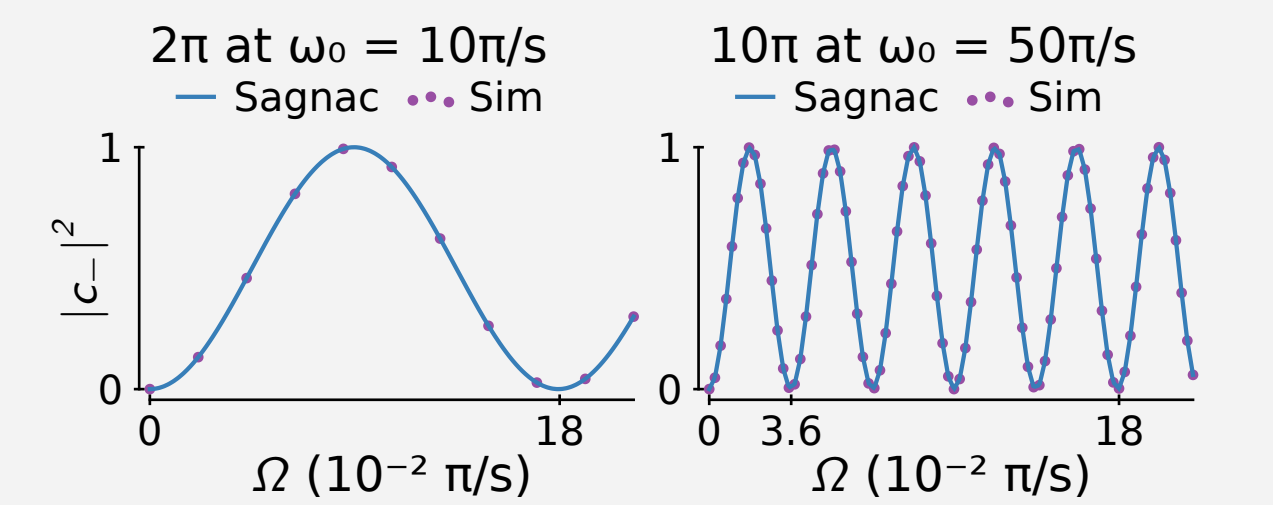
② Adiabatic Operation

Interferometric scheme

1. Accelerate from rest to $\pm\omega_0$
2. Constant rotation
3. De-accelerate back to rest

$$\omega(t) = \begin{cases} \omega_0 \sin^2\left(\frac{\pi t}{2t_r}\right) & 0 \leq t < t_r \\ \omega_0 & t_r \leq t < t_r + t_{\text{loop}} \\ \omega_0 \cos^2\left(\frac{\pi t}{2t_r}\right) & T - t_r \leq t \leq T \end{cases}$$

Interferometric response:



$$\Delta\phi = \frac{4m_{\text{Rb}}\Omega A}{\hbar}, \quad A = \frac{R^2}{2} \int_0^T \omega(t') dt'$$

$$|c_-|^2 = \frac{1}{2} - \frac{\cos \Delta\phi}{2} = \sin^2\left(\frac{\Delta\phi}{2}\right), \quad |c_+|^2 = 1 - |c_-|^2 = \cos^2\left(\frac{\Delta\phi}{2}\right)$$

References

- [1] Duspayev, Raithel, Phys. Rev. A **104**, 013307 (2021) [3] Franke-Arnold *et al.*, Opt. Express **15**, 8619 (2007)
[2] Dash, *et al.* arXiv:2308. (2023) [4] Goerz *et al.* Quantum **6**, 871 (2022)