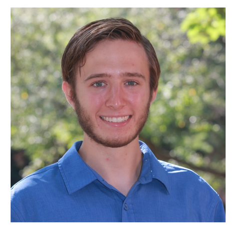
Numerical Methods for Highly Oscillatory Problems in QC

Quantum State Evolution Simulated using Implicit Filon Quadrature



Spencer Lee
Michigan State University

BACKGROUND: Quantum algorithms are realized by applying control functions. Quantum algorithms must be fast or else the system collapses into a classical state. Therefore our goal is to perform fast (classical) simulations of the quantum state of a system, governed by Schrödinger's equation:

$$\dot{\psi}(t) = -iH(t)\psi(t), \quad 0 \leq t \leq T, \psi(0) = \psi_0.$$

Approach

Filon quadrature makes the approximation

$$\int_{-1}^{1} f(t)e^{i\omega t}dt \approx \int_{-1}^{1} p(t)e^{i\omega t}dt,$$

where *p(t)* is the unique degree *2m+1* Hermite interpolation polynomial such that

$$p^{(l)}(\pm 1) = f^{(l)}(\pm 1), \quad l = 0, \dots, m.$$

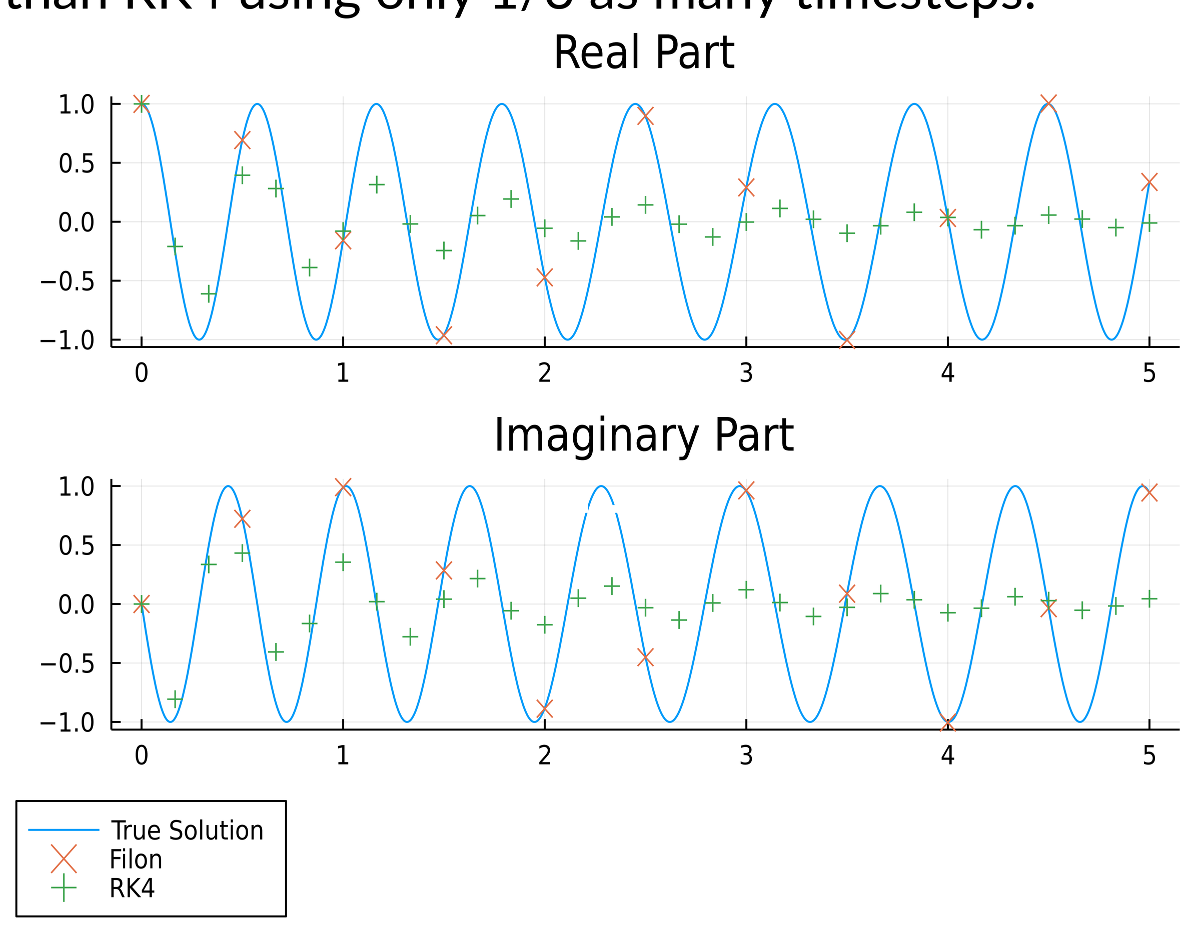
Highly effective for *f* non-oscillatory and *ω >> 1*. We assume ψ is highly oscillatory with frequency ω, so that

$$\int_{-1}^{1} \left(H(t)e^{-i\omega t}\psi(t)\right) e^{i\omega t}dt$$

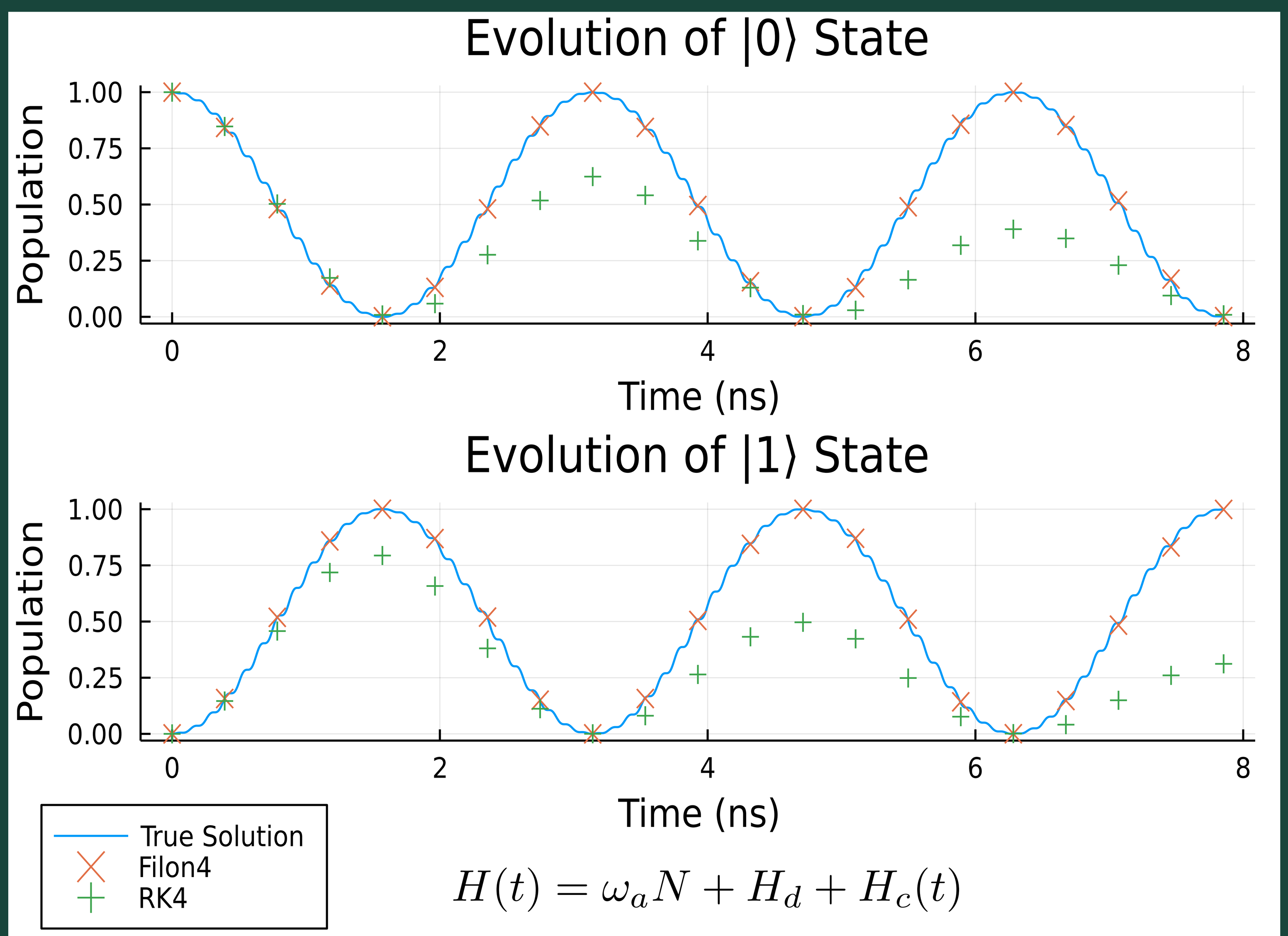
is suitable for integration using Filon quadrature. This assumption is accurate if the control is small in magnitude relative to the entire Hamiltonian.

RESULTS

For the simple case of $\dot{\psi}(t) = i\left(\cos(t) + \omega\right)\psi(t)$ with ω=10, 4th-order Filon is much more accurate than RK4 using only 1/3 as many timesteps:

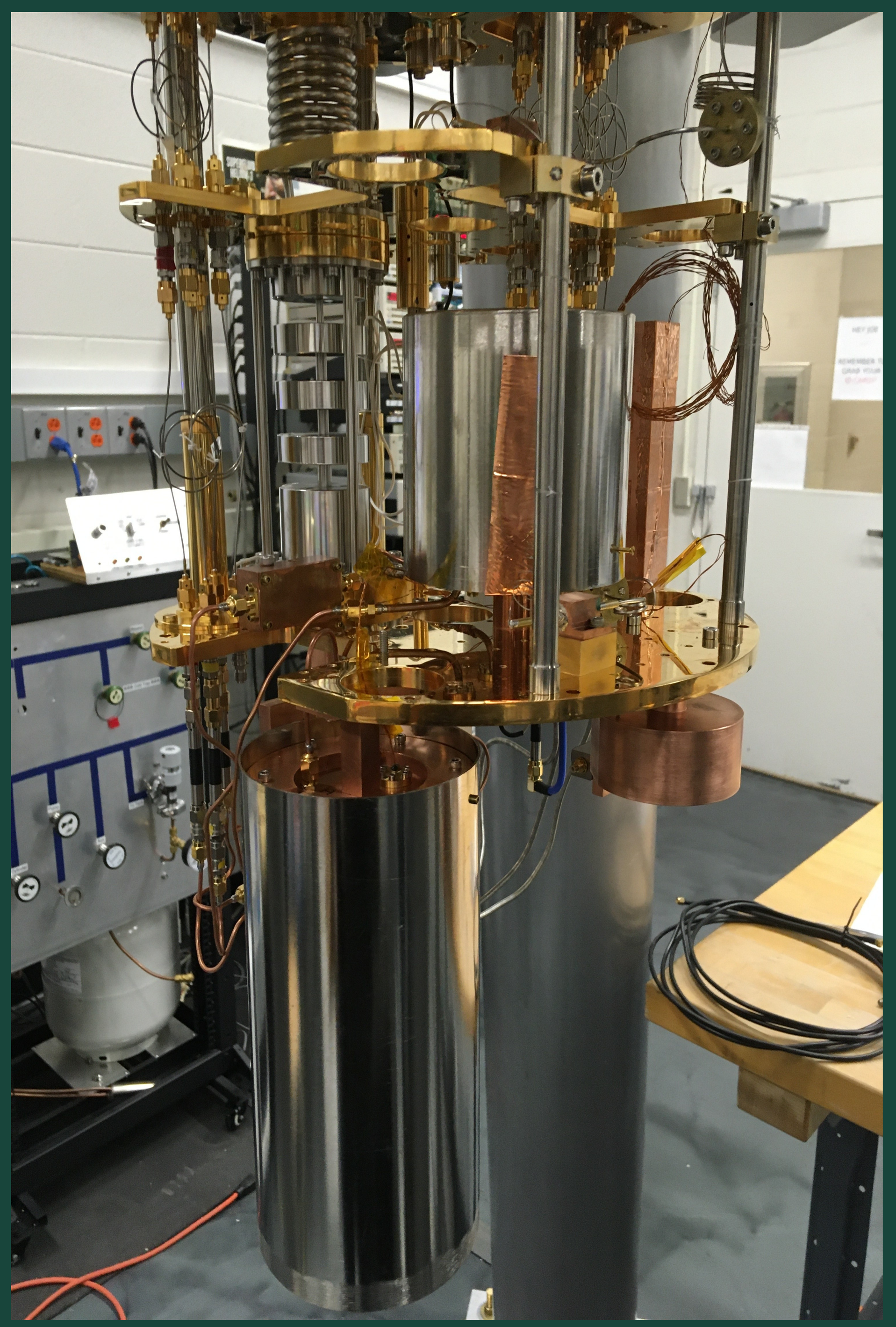


How We Use Filon Quadrature to Solve High Frequency ODEs, Design Controls and Gates for Quantum Computers



$$H(t) = \omega_a N + H_d + H_c(t)$$

Simulation of Rabi oscillator, used to swap states, for a single qubit (d=2). 4th-order implicit Filon is significantly more accurate than 4th-order Runge-Kutta while using the same number of timesteps (every 10th timestep shown above).



Photograph Courtesy of Johannes Pollanen, Michigan State University

SCAN ME

Take a picture to **download** the **full paper**

$$\frac{d\mathbf{u}}{dt} = H(t)\mathbf{u}(t)$$

$$\rightarrow \mathbf{u}_{n+1} = \mathbf{u}_n + \int_{t_n}^{t_{n+1}} \mathbf{u}_t dt$$

$$\approx \mathbf{u}_n + \begin{bmatrix} \sum_{k=1}^{m} \int_{t_n}^{t_{n+1}} h_{1k}e^{-i\omega_k t}u_k e^{i\omega_k t}dt \\ \vdots \\ \sum_{k=1}^{m} \int_{t_n}^{t_{n+1}} h_{mk}e^{-i\omega_k t}u_k e^{i\omega_k t}dt \end{bmatrix}$$

$$b_{1,0}(\omega) = \frac{ie^{-i\omega}}{\omega} + \frac{3i\cos\omega}{\omega^3} - \frac{3i\sin\omega}{\omega^4}$$

$$b_{1,1}(\omega) = -\frac{e^{-i\omega}}{\omega^2} + \frac{i(2e^{-i\omega} + e^{i\omega})}{\omega^3} - \frac{3i\sin\omega}{\omega^4}$$

$$b_{2,0}(\omega) = \frac{-ie^{i\omega}}{\omega} - \frac{3i\cos\omega}{\omega^3} + \frac{3i\sin\omega}{\omega^4}$$

$$b_{2,1}(\omega) = \frac{e^{i\omega}}{\omega^2} + \frac{i(e^{-i\omega} + 2e^{i\omega})}{\omega^3} - \frac{3i\sin\omega}{\omega^4}$$

$$F_\omega^n[f] - I_\omega[f] = \int_{-1}^{1} p(x)e^{i\omega x}dx - \int_{-1}^{1} f(x)e^{i\omega x}dx$$

$$= \int_{-1}^{1} (p(x) - f(x))e^{i\omega x}dx$$

$$= -\sum_{k=0}^{\infty} \frac{1}{(-i\omega)^{k+1}} \left[(p^{(k)}(x) - f^{(k)}(x))e^{i\omega x}\right]_{-1}^{1}$$

$$\lim_{\omega\to 0} F_\omega^n[f] = \int_{-1}^{1} p(x)dx$$

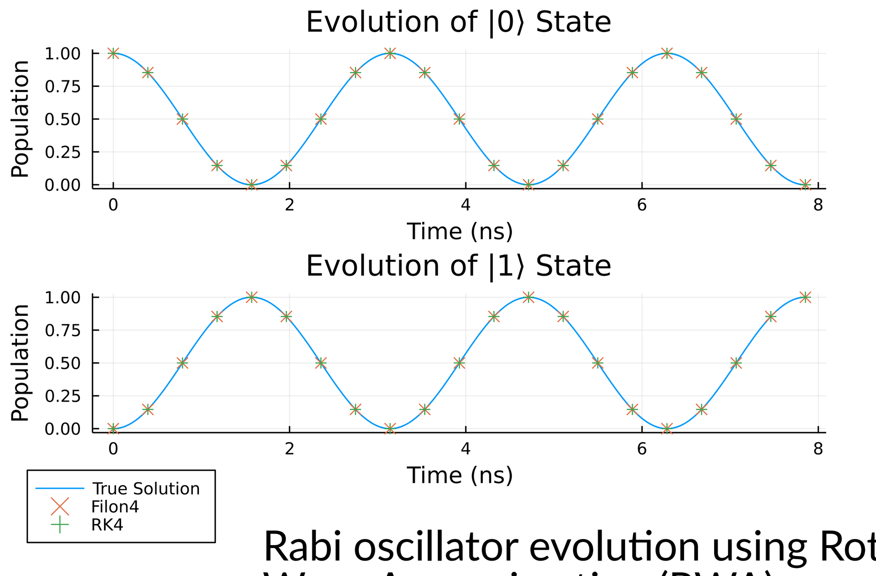
$$\lim_{\omega\to 0} I_\omega[f] = \int_{-1}^{1} f(x)dx$$

$$u_{n+1} = u_n + \frac{\Delta t}{2}e^{i\omega(t_n + \Delta t/2)} \times$$

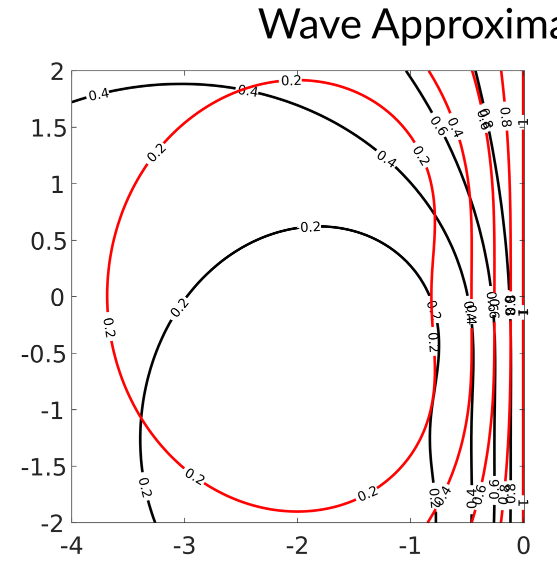
$$\left[b_{-1,0}\left(\frac{\Delta t}{2}\omega\right) f(t_n) + b_{1,0}\left(\frac{\Delta t}{2}\omega\right) f(t_{n+1}) + \frac{\Delta t}{2}\left(b_{-1,1}\left(\frac{\Delta t}{2}\omega\right) f'(t_n) + b_{1,1}\left(\frac{\Delta t}{2}\omega\right) f'(t_{n+1})\right)\right]$$

$$H_c(t) = f(t)(a + a^\dagger), \quad a = \begin{bmatrix} 0 & 1 & & \\ & 0 & \sqrt{2} & \\ & & \ddots & \ddots \end{bmatrix}$$

Rabi Oscillator: $f(t) = 2\Omega\cos(\omega_a t)$



Rabi oscillator evolution using Rotating Wave Approximation (RWA).



Contours for stability function Q for ω=0 (red) and ωΔt/2=-1.5 (black). The x and y axes correspond to the real and imaginary parts of λΔ/2. Results suggest implicit Filon is A-stable.