IV. FINAL ASSIGNMENT (DUE APRIL 13)

In all the problems involving numerical calculations, please make sure that your computations are converged, i.e., the results of calculations weakly depend on the choice of the coordinate and time step sizes, *etc.* When submitting the assignment include your codes as well as numerical and graphical outputs confirming your conclusions. Your are allowed to use codes provided in the class [1] via the Pythonic *import* statement or in any other way.

Problem 20. Numerically calculate energies of the Hamiltonian

$$\hat{H} = \frac{\hat{p}^2}{2} + 8\hat{x}^2 + \frac{1}{4\hat{x}^2}, \qquad 0 \leq x < \infty. \tag{14}$$

How is the spectrum spaced?

Problem 21. Let the quantum time independent Hamiltonian be an $N \times N$ Hermitian matrix $\hat{H}$. Then the Schrödinger equation for the wave function $\vec{\psi}(t) = (\psi_1(t), \ldots, \psi_N(t))^T$ reads

$$i\frac{d}{dt}\vec{\psi}(t) = \hat{H}\vec{\psi}(t). \tag{15}$$

Randomly generate the matrix $\hat{H}$ for $N = 3$ and select a random initial condition $\vec{\psi}(t = 0)$. Solve Eq. (15) using an ODE solver in Scipy (see, e.g., [2]). Numerically verify that the obtained solution $\vec{\psi}(t)$ coincides with $\vec{\psi}(t) = e^{-it\hat{H}}\vec{\psi}(t = 0)$, where the matrix exponent can be evaluated via *scipy.linalg.expm*[3].

Problem 22. (**Mandatory for graduate students; extra credit for undergraduates**) Assuming that the generator of motion $\hat{\mathcal{G}}(\tau)$ at time τ commutes with $\hat{\mathcal{G}}(\tau')$ at another time τ' [i.e., $\hat{\mathcal{G}}(\tau)\hat{\mathcal{G}}(\tau') = \hat{\mathcal{G}}(\tau')\hat{\mathcal{G}}(\tau)$ for all $t' \leq \tau' \leq t$ and $t' \leq \tau \leq t$] show that

$$\hat{\mathcal{T}} \exp\left[-i\int_{t'}^{t} \hat{\mathcal{G}}(\tau)d\tau\right] = \exp\left[-i\int_{t'}^{t} \hat{\mathcal{G}}(\tau)d\tau\right], \tag{16}$$

i.e., the time ordered exponent coincides with the operatorial exponent.

Problem 23. (**Mandatory for graduate students; extra credit for undergraduates**) Implement the split-operator propagator for the 1D Schrödinger equation in the momentum representation. You may omit the verification of the Ehrenfest theorems.

Problem 24. (**Mandatory for graduate students; extra credit for undergraduates**) In the class we studied the symplectic integrators for classical dynamics (in particular, we derived and implemented the Verlet integrator). In this assignment, we will see how to use this method to propagate Schrödinger equation for N-level system with a time independent Hamiltonian.

First let's see how the Schrodinger equation (15) reduces to a set of classical equations of motion. Introduce the classical coordinates $x_j(t)$ and momenta $p_j(t)$ as

$$x_j(t) = \sqrt{2}\mathrm{Re}\,\psi_j(t), \qquad p_j(t) = \sqrt{2}\mathrm{Im}\,\psi_j(t), \qquad j = 1, \ldots N. \tag{17}$$

Show that Eq. (15) is equivalent to the following Newton's equations of motion [i.e., Hamilton's equations with the classical Hamiltonian $h(\vec{x}, \vec{p})$]

$$\frac{dx_j(t)}{dt} = \frac{\partial h(\vec{x}, \vec{p})}{\partial p_j}, \qquad \frac{dp_j(t)}{dt} = -\frac{\partial h(\vec{x}, \vec{p})}{\partial x_j}, \qquad j = 1, \ldots, N, \qquad h(\vec{x}, \vec{p}) = \frac{1}{2}\sum_{k,l=1}^{N} H_{kl}\,(x_k - ip_k)\,(x_l + ip_l)\,. \tag{18}$$

[1] https://github.com/dibondar/QuantumClassicalDynamics
[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html
[3] http://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.expm.html

Furthermore, assume that $\hat{H}$ is a real symmetric matrix (i.e., $H_{kl} = H_{lk}$) show that

$$h(\vec{x}, \vec{p}) = \frac{1}{2} \sum_{k,l=1}^{N} H_{kl} \left(x_k x_l + p_k p_l \right). \tag{19}$$

The symplectic integrators are well applicable to the case of Eq. (19).