1 — PHY 494: Makeup assignment 1 (15 points total)

Due Thursday, April 5, 2018, 11:59pm.

This is a **optional Makeup Assignment**. If you choose to hand it in by the deadline, it will be graded like a normal homework assignment. If its grade is better than your worst homework grade then it will replace that grade.

Submission is to your **private GitHub repository**.

Enter the repository and run the script `scripts/update.sh` (replace *YourGitHubUsername* with your GitHub username):

```
cd assignments-2018-YourGitHubUsername
bash ./scripts/update.sh
```

It should create three subdirectories[1] `makeup_01/Submission`, `makeup_01/Grade`, and `makeup_01/Work` and also pull in the PDF of the makeup and an additional file.

To submit your makeup assignment, commit and push Python code inside the `makeup_01/Submission` directory. *Commit any other additional files exactly as required in the problems.*

Failure to adhere to the following requirements may lead to homework being returned ungraded with 0 points for the problem.

- Only submit code.

- All code should be in a file `makeup01.py`.

- Code will be tested against the unit tests in `test_makeup01.py`. The grade will be approximately proportional to the number of tests that pass successfully so your code *must* be able run under the tests. (Failing tests for the Bonus problem can be ignored.)

1.1 ODE with Scipy (15 points)

In this problem you should learn how to use existing functions in a library, namely in the `scipy` package. Part of the problem is to find and read the documentation for the function and figure out how to use it.

Use the Scipy function `scipy.integrate.odeint()` to solve the following ordinary differential equation

$$-\frac{1}{2}\frac{d^2\psi_n(x)}{dx^2} + \left[\frac{1}{2}x^2 - E_n\right]\psi_n(x) = 0 \tag{1}$$

$$\psi_n(0) = 1; \quad \frac{d\psi_n(0)}{dx} = 0 \tag{2}$$

$$E_n = n + \frac{1}{2}, \quad n = 0, 2, 4, 6, \ldots \tag{3}$$

---

[1]If the script fails, file an issue in the Issue Tracker for PHY494-assignments-skeleton and just create the directories manually.

for the real function $\psi_n(x)$[2] and the three values $n = 0$, $n = 2$, and $n = 8$.

The code should contain the following functions:

1. **ode_qmhosc()** solves the ODE Eq. 1:

   ```
   psi = ode_qmhosc(x, psi0, dpsidx0, n=n)
   ```

   that takes as arguments

   - all the values x at which the solution should be evaluated (the first value *must* be the one for which the initial conditions are given, i.e., $x = 0$ for this problem),
   - as initial conditions the function value and first derivative at $x = 0$ (Eq. 2), and
   - the value of n that determines E_n (Eq. 3).

   It should return the function values $\psi_n(x)$ as a numpy array.

2. **f_qmhosc()** is needed to transform the ODE Eq. 1 into standard form so that it can be solved with `scipy.integrate.odeint()`.

   ```
   f_qmhosc(y, t, E=0)
   ```

   should produce the *ODE standard form*[3] of the derivative vector **f** when provided with

   - the current values of **y**
   - the current value of t
   - and the parameter E_n (see Eq. 1).

(a) Numerically solve Eq. 1 on a lattice with $\Delta x = 0.01$ for $0 \leqslant x < 6$ for

   - $n = 0$
   - $n = 2$
   - $n = 8$

   (i) The tests in `test_makeup01.py` should all pass to show that you correctly implemented a solution. [**12 points**]

   (ii) Plot all your solutions in one figure and include the figure as a PDF named `qmhosc.pdf`. [**3 points**]

---

[2]If you have done Quantum Mechanics 1 (PHY 314) then you should recognize it as the Schrödinger equation for the simple harmonic oscillator.

[3]Remember that the ODE standard form is

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}$$

and in the case of Eq. 1, we would have $t = x$, $y_0 = \psi_n(x)$, and $y_1 = \psi'_n(x) \equiv \frac{d\psi_n}{dx}$.

(b) BONUS: Correct physical[4] solutions for Eq. 1 are *symmetric* ($\psi(x) = \psi(-x)$) for even n and *antisymmetric* ($\psi(x) = -\psi(-x)$) for odd n. This implies that the *initial conditions* for $n = 1, 3, 5, \ldots$ are different from the ones for even n (Eq. 2). Choose appropriate initial conditions[5] and write a function

```
phi0, dphidx0 = initial_values_qmhosc(n)
```

that produces the correct initial values depending on the value of n.

Using your initial values, solve Eq. 1 for $n = 3$ and plot the solution (include a PDF named qmhosc_odd.pdf). [**bonus +4***]

---

[4] Physical solutions for the wavefunction $\psi_n(x)$ have to be normalizable, i.e., they have to decay to zero for $x \to \pm\infty$. The numerical solutions only fulfill this requirement for small x and even for moderately large x they start to diverge. It would be better to use specialized algorithms that have the normalizability requirement built in.

[5] Choose $\psi(0) \geqslant 0$, $\frac{d\psi(0)}{dx} \geqslant 0$ and either use values of 0 or 1 because this is the convention employed in the tests.