Lab assignment #4: Solution

Nico Grisouard (Instructor), Pascal Hogan-Lamarre (TA & Marker),
Mikhail Schee (TA)

Due Friday, October 9th 2020, 5 pm

---

1. **[35%] Solving linear systems**

   (a) Nothing to submit.

   (b) The code (not required) can be found in `L04-407-2020-Q1ab.py`. This code imports `SolveLinear_sol.py` which implements partial pivoting.

   The errors and timings for Gaussian elimination, partial pivoting, and LU decompositions for the random matrices are plotted on fig. 1. The figure shows that for $N > 50$, the pivot and LU methods are much more accurate than the standard Gaussian elimination, and yield similar answers, by a factor of 100-1000. For these large N, the LU method is about a 1,000 times faster than the other two methods, which have similar timings.

   For the small systems we are dealing with this lab, there will be no practical difference between your method and the `linalg` method.

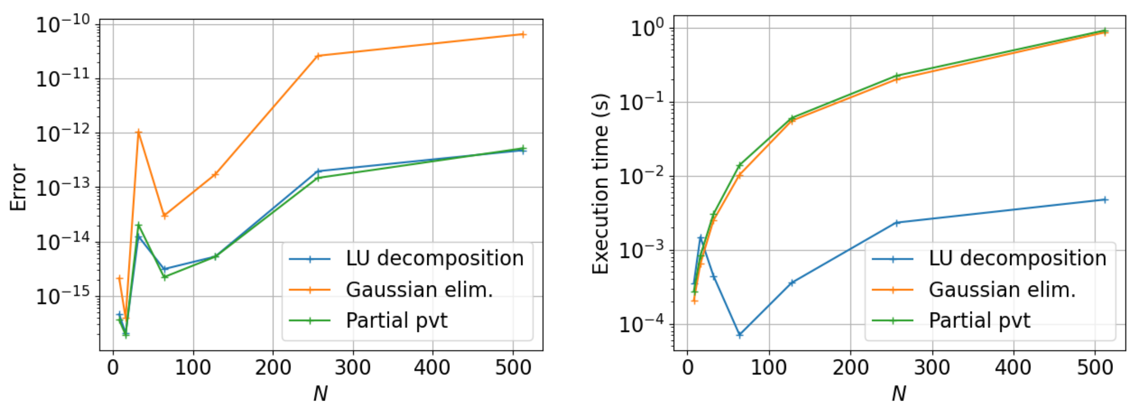

Figure 1: Solutions to Q1b. Left: error; right: execution time.

   (c) The code is in `L04-407-2020-Q1c.py`. There are two examples, first, for $R_6 = 2\mathrm{k}\Omega$:

```
Amplitudes and phases of voltages with resistor R6:
  |V1| = 1.70e+00 V, phi1(t=0) = -5 degrees
```

```
  |V2| = 1.48e+00 V, phi2(t=0) = 11 degrees
  |V3| = 1.86e+00 V, phi3(t=0) = -4 degrees
```

associated with the time series plotted on the left panel of fig. 2.

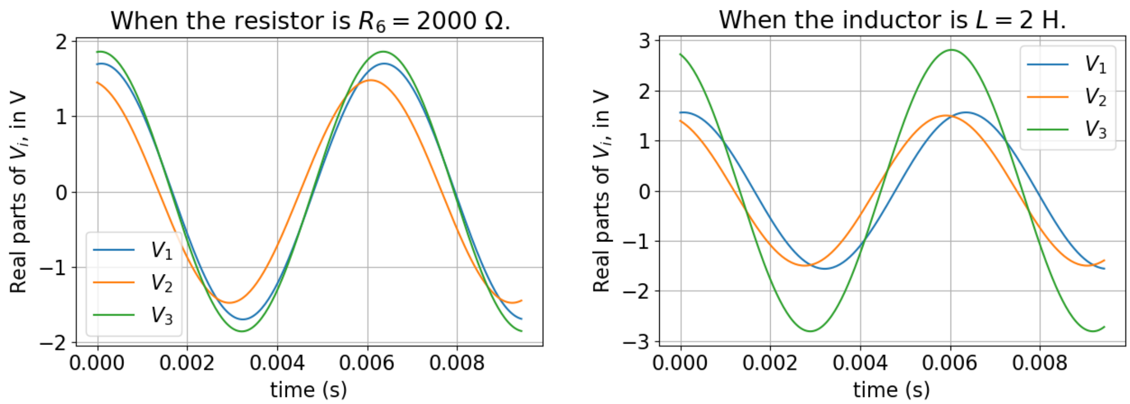

Figure 2: Solutions to Q1c. Left: first example, with R_6 a resistor; right: second example, with iR_6 an inductor.

Second, for R_6 replaced by $\mathrm{i}\omega L$, with $L = 2$H,

```
Amplitudes and phases of voltages with inductor:
  |V1| = 1.56e+00 V, phi1(t=0) = -4 degrees
  |V2| = 1.50e+00 V, phi2(t=0) = 21 degrees
  |V3| = 2.81e+00 V, phi3(t=0) = 14 degrees
```

associated with the time series plotted on the right panel of fig. 2.

Compared to the second case, the amplitude $|V_3|$ is quite large, comparable to the 3V amplitude driving voltage, indicating weaker resistance overall. The signal in V_3 has shifted ahead, reflecting the phase change owing to the inductor. The other voltages V_1 and V_2 are not affected as much.

2. **[30%] Asymmetric quantum well**. For reference, we include the code in `L04-407-2020-Q2.py`.

   (a) Here are the first 10 eigenvalues:

```
First 10 energy levels:
E1 = 5.83638e+00 eV
E2 = 1.11811e+01 eV
E3 = 1.86629e+01 eV
E4 = 2.91442e+01 eV
E5 = 4.26551e+01 eV
E6 = 5.91853e+01 eV
E7 = 7.87294e+01 eV
E8 = 1.01285e+02 eV
E9 = 1.26851e+02 eV
E10 = 1.55555e+02 eV
```

(b) If the matrix is 100x100, the first 10 eigenvalues are very similar to the case where the matrix is 10x10. For example, the ground state energy is 7e-6% different and the 10th state energy level is 0.08% different. (also ok if you mentioned the number of significant figures instead).

```
First 10 energy levels:
E1 = 5.83638e+00 eV
E2 = 1.11811e+01 eV
E3 = 1.86629e+01 eV
E4 = 2.91442e+01 eV
E5 = 4.26551e+01 eV
E6 = 5.91852e+01 eV
E7 = 7.87293e+01 eV
E8 = 1.01285e+02 eV
E9 = 1.26851e+02 eV
E10 = 1.55426e+02 eV
```

(c) The plot is fig. 3.

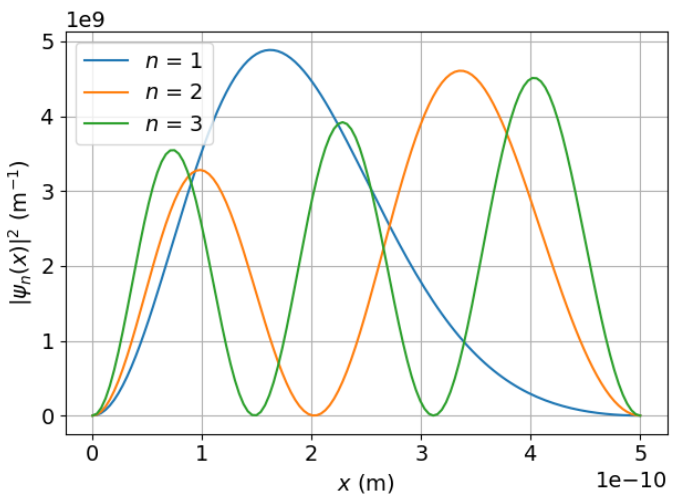

Figure 3: First three excited states: ground state ($n = 1$) and next two states.

3. **[35%] Solving non-linear equations**

(a) The plot is fig. 4.

(b) Newman, exercise 6.11:

Parts b, c See `L04-407-2020-Q3ab.py` for code. For part (b), starting from $x_0 = 0.5$, it took 15 iterations to converge without over relaxation. With $\omega = 0.5$, it converged in 7 iterations. Playing with ω a bit, I found $\omega = 0.7$ to be roughly the value where I got the minimum number of iterations, which was 4. E.g., $\omega = 0.6$ produced 6 iterations and $\omega = 0.8$ produced 5 iterations. However, this will depend on the initial $x_0$value chosen.

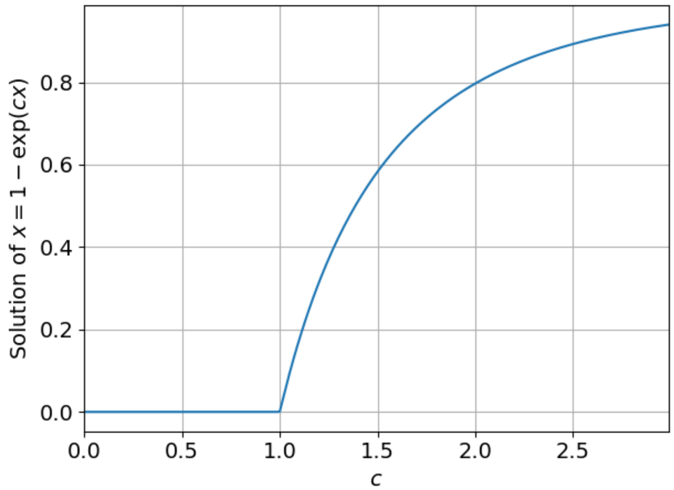

Figure 4: Solution to Q3a.

Part d $\omega < 0$ would help if $f' < 0$ since you would be damping out the overshoot of the oscillation in each case.

(c) See `L04-407-2020-Q3c.py` for code.

```
Binary search: x = 4.965114, converged in 24 iterations
Newton's method: x = 4.965114, converged in 4 iterations
Secant method: x = 4.965114, converged in 8 iterations

In any case, the temperature of the Sun is 5.772456e+03 K
```

Newton's method found the root in just a few iterations. Secant is not too shabby, binary search did its best.