Interpolation and more pseudocode

Due: Friday, March 1st, 12:30. Deposit the written part of your assignment in the designated drop box. Write your answers out carefully and clearly. Upload the following files to Blackboard:

- the function Hess_LUP.py and the script test_Hess.py for 1(**d**).
- a Python script for question 2c.

Be sure all files submitted include a comment line with your name and student number, e.g.,

```
# Johannes Vanderbroecke
# 100456789
```

Also, be a **good programmer** and include comments with a brief description of the functionality, input and output arguments and usage of each function or script. Add some comments that explain what steps are taken. Marks will be awarded or subtracted based on the readability and transparency of your code.
If you worked together with class mates on your code, and a substantial part of the code you submit coincides with theirs, you must list their names in a comment, e.g.

```
# Written in collaboration with Shawn Shawnson and Lea Leason.
```

Failure to do so may qualify your work as plagiarism.
A discussion thread for this assignment is available on Slack. Pose your questions there before approaching the lecturer or TA.

Question 1 **15 marks**

An $n \times n$ *upper Hessenberg* matrix A has the property that $A_{ij} = 0$ if $i > j + 1$.

(**a**) Compute is PLU decomposition by hand for the following example matrix:

$$A = \begin{bmatrix} 3 & -1 & 1 & 2 \\ 1 & 4 & -2 & 3 \\ 0 & -3 & 5 & -5 \\ 0 & 0 & 3 & 1 \end{bmatrix}$$

You will find that a lot of the multipliers are zero.

(**b**) Write a pseudo-code for the LUP decomposition of an upper Hessenberg matrix that avoids unnecessary FLOPs (i.e. adding and multiplying by zero).

(**c**) Compute the number of FLOPs necessary to complete the pseudo-code.

(**d**) Write a function in Python that implements the pseudo-code of part (**b**). Also write a test script that

1. generates a $n \times n$ random upper Hessenberg matrix,
2. computes its PLU decomposition,
3. measures the time it takes to compute the decomposition for $n = 2^p$, $p = 4, \ldots, 11$ and
4. plots the wall time on an appropriate scale to see the behaviour predicted by the FLOP count of part (**c**).

Question 2 **10 marks**

Consider the problem of interpolating the data

k	0	1	2	3	4
x_k	0	1	2	3	4
y_k	0	1	5	14	35

using a polynomial interpolant $\Pi_4(x)$

(**a**) Write out the interpolating conditions that $\Pi_4(x)$ satisfies.

(**b**) Construct the Vandermonde matrix V associated with the data $\{x_k\}_{k=0}^{4}$ from the table above. Work with pen(cil) & paper.

(**c**) Write a Python script that does the following:

1. Forms the Vandermonde matrix.
2. Solves the linear system corresponding to the interpolation conditions.
3. Produces a figure that shows the interpolant as well as the points in the table.