ASSIGNMENT 1

1. **(5)** Consider the matrix $H(v) = \hat{1} - 2vv^T$, where v is a unit column vector. What is the rank of the matrix $H(v)$? Prove that it is orthogonal.

   Solution 1 *Let's prove that* $A = A^T$

   $$A^T = \left(I_n - 2vv^T\right)^T = I_n - 2(vv^T)^T = I_n - 2vv^T$$

   So $A^T = A$. *Calculating* AA^T

   $$AA^T = A^2 = \left(I_n - 2vv^T\right)\left(I_n - 2vv^T\right) = I_n - 4vv^T + 4vv^Tvv^T = |v^Tv = 1| = I_n$$

   from which we conclude that A *is both orthogonal and indepotent matrix.*
   For indepotent matrices following statement is true:

   $$\operatorname{tr} A = \operatorname{rank} A$$
   $$\operatorname{tr} A = n - 2v^Tv = n - 2$$
   $$\operatorname{rank} A = n - 2$$

   Solution 2

   $$Hv = -v$$

   So v *is an eigenvector with eigenvalue* $\lambda = -1$. *If we take any another vector* u *such that* $u \perp v$, *we get* $Hu = u$. *So there are only two eigenvalues* $\lambda_1 = -1$ *and* $\lambda_2 = 1$. *It means that there exists some transformation* V *which makes* H *diagonal matrix* $\operatorname{diag}(-1, 1, \ldots 1)$. *This matrix is it's own inverse hence orthogonal. Beside that it is indepotent matrix so* $\operatorname{rank} A = n - 2$

2. **(10)** Prove the following inequalities and provide examples of x and A when they turn into equalities:

   - $\|x\|_2 \leq \sqrt{m}\,\|x\|_\infty$
   - $\|A\|_\infty \leq \sqrt{n}\,\|A\|_2$

   where x is a vector of m components and A is $m \times n$ matrix.

   Solution 3

   - $$|x_i| \leq \sup_j |x_j| := \|x\|_\infty \Rightarrow x_i^2 \leq \|x\|_\infty^2$$
     $$\|x\|_2 = \sqrt{\sum_i^m x_i^2} \leq \sqrt{\sum_i^m \|x\|_\infty^2} = \sqrt{m\|x\|_\infty^2} = \sqrt{m}\|x\|_\infty$$

   - $$\|A\|_2 = \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} \leq \sup_{x\neq 0} \frac{\sqrt{m}\|Ax\|_\infty}{\|x\|_2} \leq \frac{\sqrt{m}\|Ax\|_\infty}{\|x\|_\infty} = \sqrt{m}\|A\|_\infty$$

     $$\|Ax\|_\infty \leq \|Ax\|_2 \leq \|A\|_2\|x\|_2 \leq \|A\|_2\sqrt{n}\|x\|_\infty \Rightarrow \|A\|_\infty \leq \sqrt{n}\,\|A\|_2$$

     $$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{m}\,\|x\|_\infty$$
     $$\|A\|_2 = \frac{\|Ax\|_2}{\|x\|_2} \leq \frac{\sqrt{n}\,\|Ax\|_\infty}{\|x\|_\infty} = \sqrt{n}A_\infty$$
     $$\|A\|_\infty = \frac{\|Ax\|_\infty}{\|x\|_\infty} \leq \frac{\sqrt{m}\,\|Ax\|_2}{\|x\|_2} = \sqrt{m}A_2$$

3. **(5)** Assuming u and v are m-vectors, consider the matrix $A = 1 + uv^T$ which is a rank-one perturbation of identity. Can it be singular? Assuming it is not, compute its inverse. You may look for it in a form of $A^{-1} = 1 + \alpha uv^T$ for some scalar α and evaluate α.

4. **(5)** Prove that for any unitary matrix U one has $\|UA\|_F = \|AU\|_F = \|A\|_F$.

5. **(10)** In this exercise your goal will be to study and speed up an implementation of K-means algorithm. In the notebook `kmeans.ipynb`, you can find a naive implementation. Explore the code, make sure you understand it. You will find there two functions `dist_i` and `dist_ij` which are (on purpose) implemented in a rather inefficient way. Improve them by getting rid of the loops in the favor of a proper numpy vectorized implementation and measure the speed-up of the full algorithm for $N = 10000$.

6. **(10)** Some things just can not be vectorized but still can be sped up compared to naive implementation. For example, consider computation of the Hofstadter-Conway sequence $a(n)$ such that $a(1) = 1$, $a(2) = 1$ and

$$a(n) = a(a(n-1)) + a(n - a(n-1)), \quad n > 2 \tag{1}$$

Write three functions, computing the sequence up to n-th element in three ways: i) pre-allocating `numpy` array and filling it using `for` loop, ii) cumulatively appending `python` list and converting it to `numpy` array, iii) same as i) but compiled (`jit`) version. Time the resulting implementations and conclude which is preferable. With the optimal one, compute $a(10^8)$.

7. **(15*)** Consider a function mapping six tensors to one tensor: $Z\left(\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)}, \Gamma^{(1)}, \Gamma^{(2)}, U\right)$, with

$$Z_{ahij} = \sum_{bcdefg} \lambda^{(1)}{}_{ab}\Gamma^{(1)}{}_{cbd}\lambda^{(2)}{}_{de}\Gamma^{(2)}{}_{feg}\lambda^{(3)}{}_{gh}U_{ijcf}. \tag{2}$$

Assume that all indices of the tensors appearing above take values from 1 to χ. Running the numerical experiments, explore the values of χ in the range 3–50 (from slowest to fastest implementation).

- In the notebook `convolution.ipynb` you may find implemented a *stupid* way to compute this convolution, which takes $\chi^4 \times \chi^6 = \chi^{10}$ flops. In fact, this can be computed much faster!
- Using the function `numpy.einsum` (its crucial to use the `optimize` argument), you can actually achieve a much faster implementation. In order to understand what it is doing under the hood, explore the function `numpy.einsum_path`. What is the minimal number of flops required for computation of Z?
- Using the understanding of the output of `numpy.einsum_path`, implement an algorithm to compute Z, which is as effective as `numpy.einsum`, but relying only on more elementary `numpy.dot` and `numpy.tensor_dot`.