

AM205 HW2. Numerical linear algebra

P1. Equations with vector norms [8 pts]

Define a matrix

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

representing a linear transformation Av for $v = (x, y)^T \in \mathbb{R}^2$. Here you will consider a system of equations $\|v\|_p = 1$ and $\|Av\|_p = 2$ and see that the system has only four solutions for any finite p but has infinitely many solutions for $p = \infty$.

(a) [3 pts] Plot two curves $\|v\|_2 = 1$ and $\|Av\|_2 = 2$. Analytically find the coordinates of four points $v \in \mathbb{R}^2$ where the two curves intersect and mark them on the plot.

(b) [1 pts] Plot two curves $\|v\|_4 = 1$ and $\|Av\|_4 = 2$.

(c) [2 pts] Plot two curves $\|v\|_\infty = 1$ and $\|Av\|_\infty = 2$. Analytically show that both curves contain the entire line segment connecting the points $(1, \frac{2}{3})$ and $(1, 1)$. Mark the line segment on the plot.

(d) [2 pts] For any finite $p \geq 1$, analytically show that if $v \in \mathbb{R}^2$ satisfies both $\|v\|_p = 1$ and $\|Av\|_p = 2$, then either $4x = y$ or $4x = 5y$, i.e. the solutions always lie on the two straight lines. Draw the straight lines on your plots from the previous parts.

P2. Condition number of LU factorization [15 pts]

(a) [5 pts] Implement the LU factorization algorithm for a square matrix $A \in \mathbb{R}^{n \times n}$ with partial pivoting and without pivoting. Do not use library functions for the LU factorization.

(b) [3 pts] In the same log-linear plot, show the condition number $\kappa(A)$ and the condition numbers $\kappa(L)$ and $\kappa(U)$ after each step of the algorithm (i.e. after eliminating the elements below the diagonal in one column) with and without pivoting for the Vandermonde matrix on 10 equidistant points in the range $[1, 2]$ (including the endpoints). Use the same ordering as in `numpy.vander()` by default, i.e. decreasing powers from left to right.

(c) [3 pts] Repeat part **(b)** for the following “pseudorandom” matrix $A \in \mathbb{R}^{n \times n}$

$$A_{ij} = \cos(37 \cos(37(i + nj))) \quad \text{for } i, j = 1, \dots, n$$

with $n = 10$.

(d) [4 pts] Sample 1000 random 10×10 matrices A with elements distributed uniformly in $[-1, 1]$ and plot the histograms of $\log_{10} \kappa(A)$, $\log_{10} \kappa(U)$ with pivoting, and $\log_{10} \kappa(U)$ without pivoting. To compute the histograms, use 50 equal-width bins in the range $[0, 5]$. Report the most probable value of κ , i.e. value from the bin with the largest number of samples. Which algorithm (with or without pivoting) is expected to provide a smaller condition number $\kappa(U)$? Are your results consistent with this expectation?

P3. Sparse linear algebra [19 pts]

Here you will implement common operations with sparse matrices in different formats and analyze their cost. Consider three different storage formats: two-dimensional array (dense), compressed sparse row (CSR), and compressed sparse column (CSC). Do not use library functions or operators. However, you may and should use existing data structures (e.g. `scipy.sparse.csr_array`).

(a) [3 pts] Write three functions that implement the matrix-vector product Ax for $x \in \mathbb{R}^n$ and matrix $A \in \mathbb{R}^{n \times n}$ stored in each of the three formats (dense, CSR, CSC). Store vectors as one-dimensional arrays.

(b) [5 pts] Consider a matrix $A \in \mathbb{R}^{n \times n}$ stored in the CSR format and matrix $B \in \mathbb{R}^{n \times n}$ stored in the CSC format. Write a function that returns the product AB in the CSC format.

(c) [5 pts] Write a function that takes matrix $A \in \mathbb{R}^{n \times n}$ in the CSC format and column index j and returns the elementary elimination matrix

$$L_j = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -a_{j+1,j}/a_{jj} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -a_{nj}/a_{jj} & 0 & \cdots & 1 \end{bmatrix}$$

in the CSR format, such that $L_j A$ contains only zeros below the diagonal in column j .

(d) [2 pts] Write a function `sparse_lu()` that takes a matrix $A \in \mathbb{R}^{n \times n}$ stored in the CSC format and returns an upper triangular matrix $U = L_{n-1} \cdots L_2 L_1 A$ stored in the CSC format. Reuse the two functions from parts **(b)** and **(c)**.

(e) [4 pts] By adding a counter to your functions (use a global variable or return the number of operations together with the result), measure the number of floating point operations $K(n)$ required to complete `sparse_lu()` for $n = 5, 6, \dots, 50$ in each of the following cases

- A is a tridiagonal matrix

$$A = \begin{bmatrix} 3 & 1 & & & \\ 1 & 3 & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 3 \end{bmatrix},$$

- $A = I + H$, where H is a Hilbert matrix with elements

$$H_{ij} = \frac{1}{i+j-1} \quad \text{for } i, j = 1, \dots, n.$$

You may ignore those operations and loops that do not contribute to the leading term of $K(n)$, i.e. measure it up to asymptotic equivalence. Make a log-log plot of the number of operations $K(n)$ as a function of n . Fit a straight line $\log C + q \log n \approx \log K(n)$ and report the obtained C and q .

P4. Unstable LU factorization [15 pts]

Consider matrices $G(n, c) \in \mathbb{R}^{n \times n}$ of the form

$$G(n, c) = \begin{bmatrix} c & 0 & 0 & \dots & 0 & c \\ -1 & 1 & 0 & \dots & 0 & 0 \\ -1 & -1 & 1 & \dots & 0 & 0 \\ \vdots & & & \ddots & \vdots & \vdots \\ -1 & -1 & -1 & \dots & 1 & 0 \\ -1 & -1 & -1 & \dots & -1 & 0 \end{bmatrix}$$

with $n \geq 3$ and $c \in \mathbb{R}, c > 0$. In the following, you will perform the LU factorization of $G(n, c)$ with partial pivoting, i.e. by selecting the row with the largest absolute value in the corresponding column. If the choice of the pivot is non-unique, i.e. multiple rows contain elements with the same absolute value, select the one with the minimal row index (however, if you use a library implementation in part (d), any choice of the row is allowed).

(a) [3 pts] Consider two cases $0 < c < 1$ and $c > 1$. In each case, perform the LU factorization of $G(4, c)$ with partial pivoting (either by hand or using symbolic computation). Report matrices U_1, U_2 , and $U_3 = U$ obtained after each step of the algorithm, i.e. after eliminating the elements below the diagonal in one column. The results should contain c as a parameter.

(b) [3 pts] Consider two cases $0 < c < 1$ and $c > 1$. For each case, perform the LU factorization of $G(n, c)$ with partial pivoting in general for all $n \geq 3$ and report the resulting matrix U . The result should contain c as a parameter. Showing the results of intermediate steps is not necessary.

(c) [2 pts] Write a function that returns $G(n, c)$.

(d) [7 pts] Consider two cases $c = 0.95$ and $c = 1.05$. For each case and for $n = 3, 4, \dots, 80$, let $G = G(n, c)$, define the reference solution $x \in \mathbb{R}^n$ as n equidistant points in the range $[1, 2]$ (including the endpoints), and construct a right-hand side vector $b = Gx$. Perform the LU factorization of G with partial pivoting numerically (e.g. using `scipy.linalg.lu`) and solve the system $G\tilde{x} = b$ by solving two triangular systems (e.g. using `scipy.linalg.solve_triangular`). Plot the 2-norm relative error $\|x - \tilde{x}\|_2 / \|x\|_2$ as a function of n for each c . Explain if the observed behavior of the error is consistent with the form of the matrix U obtained in parts **(a)** or **(b)**.

P5. QR factorization applied to a bouncing ball [15 pts]

Here you will analyze the trajectories of a bouncing [Super Ball](#). In the data archive `am205_hw2_data.zip`, there is a directory `super` containing twenty images showing two bounces of the ball on a table top. The images are given for illustration only, you will not need to process them. The duration between two frames is $7/120$ s. The diameter of the ball is 42.5 mm corresponding to 43.5 pixels. In the same archive, you will find a text file `super_ypos.txt` that lists the y positions of the ball center measured in pixels for each frame k , which were calculated from the images.

(a) [6 pts] Write a function that performs the QR factorization using Givens rotations for an arbitrary rectangular matrix $A \in \mathbb{R}^{m \times n}$ where $m \geq n$. Test your function on 10 random 11×7 matrices, with elements drawn from a uniform distribution in $[-1, 1]$, by verifying that the Frobenius norm $\|A - QR\|_F$ is small. Do not use library functions for the QR factorization.

(b) [6 pts] During the experiment, the ball follows three parabolic arcs separated from each other by the two bounces. Using your QR factorization function, fit each of the three different arcs to

$$y(k) = \alpha k^2 + \beta k + \gamma,$$

where k is the frame index. Report the obtained coefficients and plot the fitted model together with the data points.

(c) [3 pts] Using your fitted models from part (b), calculate and report the gravitational acceleration g from each of the three parabolic arcs. Using the fits to the first two arcs, calculate the height h above the table top from which the ball is released. This should be the distance from the table top to the bottom of the ball. All quantities should be reported in the SI units.

P6. Traffic light images from PCA [20 pts]

Here you will apply the principal component analysis (PCA) to images showing a traffic light, calibrate the components using labeled images, and generate new images not observed in practice. The images are extracted from a [time-lapse video](#) produced in Italy.

In the archive `am205_hw2_data.zip`, you will find a directory `traffic` with images `a_*.png` numbered by the image index. Each image has a resolution of 240×135 pixels with RGB channels. In the following, each image is represented as a vector $p \in \mathbb{R}^n$ containing the color intensity values between 0 and 1, where $n = 240 \times 135 \times 3$. The dataset consists of $m = 64$ images p_i for $i = 0, \dots, m - 1$.

(a) [4 pts] Construct the image $p_{\min} \in \mathbb{R}^n$ as the element-wise minimum over all images. Visualize p_{\min} . In the following, vectors $p_i - p_{\min}$ are referred to as relative images.

(b) [5 pts] Assemble a matrix $A \in \mathbb{R}^{m \times n}$ with rows $(p_i - p_{\min})^T$ for $i = 0, \dots, m - 1$. Using the reduced SVD of A , extract the first three principal components $v_1, v_2, v_3 \in \mathbb{R}^n$ which are orthonormal vectors. It is important to use the reduced SVD since the matrices generated by the full SVD will not fit in memory. Note that unlike the case discussed in the lecture, the reduced SVD is applied to matrix A with $m < n$ and returns you a non-square matrix of the right singular vectors. Visualize the images defined as $0.5(1 + v_k / \|v_k\|_\infty)$ for $k = 1, 2, 3$.

(c) [8 pts] Three images with indices 0, 37, and 5 are used for calibration. They show three states of the traffic light: $g_1 = (1, 0, 0)^T$ (red), $g_2 = (0, 1, 0)^T$ (yellow), and $g_3 = (0, 0, 1)^T$ (green).



Denote the respective relevant images as $q_1 = p_0 - p_{\min}$, $q_2 = p_{37} - p_{\min}$, and $q_3 = p_5 - p_{\min}$. Assemble a matrix $V \in \mathbb{R}^{3 \times n}$ with rows v_1, v_2, v_3 from part (b). Define a new

of set vectors $w_1, w_2, w_3 \in \mathbb{R}^n$ and the corresponding matrix $W \in \mathbb{R}^{3 \times n}$ as

$$W = FV \tag{1}$$

with an unknown matrix $F \in \mathbb{R}^{3 \times 3}$. The state vectors will be used as coordinates in the basis W . The relative image corresponding to a state vector $g \in \mathbb{R}^3$ is recovered by $W^T g$. The unknown matrix F can be found by minimizing the least-squares error

$$S(F) = \sum_{k=1}^3 \|q_k - W^T g_k\|^2 = \sum_{k=1}^3 \|q_k - V^T F^T g_k\|^2,$$

which is equivalent to solving the normal equations

$$V(q_k - V^T F^T g_k) = 0, \quad k = 1, 2, 3.$$

Note that V consists of orthonormal rows and the state vectors g_1, g_2, g_3 are linearly independent. Solve these equations to obtain F , then compute W from (1) that will give you w_1, w_2, w_3 . Report the obtained F . Visualize the images defined as $0.5(1 + w_k / \|w_k\|_\infty)$ for $k = 1, 2, 3$.

(d) [3 pts] Some state vectors correspond to abnormal states of the traffic light. Consider new state vectors $g = (0, 1, 1)$, $(1, 0, 1)$, and $(1, 1, 0)$. Visualize the corresponding images $p_{\min} + W^T g$ after clipping the values to $[0, 1]$.