# Problem set 2

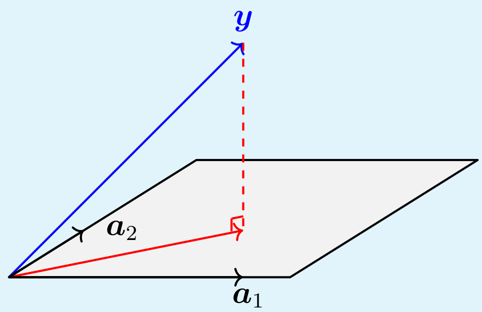
**Workout 0.1.** Compute the condition number of $A$ in norm infinity, norm 1 and Frobenius norm. You can use Python to compute the inverse of $A$:

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

**Workout 0.2.** Use least squares approximation to fit a straight line (linear polynomial) to the data set:

| $x$ | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|
| $y$ | 5.0 | 5.0 | 4.0 | 3.0 | 1.0 |

1. Use the ansatz $p_1(x) = a_0 + a_1 \cdot x$, formulate the normal equations $A^T A\boldsymbol{a} = A^T\boldsymbol{y}$, and solve them. Write down all the steps from the polynomial ansatz, to the matrix $A$ and the normal equations, and finally the resulting polynomial. Also, draw the data set and the straight line in a figure. You can use Python to create the figure, or you can draw it by hand.
2. Calculate the residual vector $\boldsymbol{e} = \boldsymbol{y} - A\boldsymbol{a}$, where $A$ is the matrix you construct in part 1, and $\boldsymbol{a}$ is the solution of the normal equations (i.e. the polynomial coefficients). What does the residual mean in relation to the figure you draw in part 1? Finally compute $residual = \|\boldsymbol{e}\|_2$.
3. Below is an image describing what we are doing when we form the Normal equations.




Try to relate the image to the particular case in this task. What are the different vectors in the image in this particular case? Also, what is the length of the error vector (dashed red line)?

**Workout 0.3.**

1. Using the same data set as in previous task, do a least squares approximation again but now center the data. You can do that by using the ansatz $p_1(x) = a_0 + a_1 \cdot (x - \bar{x})$, where $\bar{x}$ is the mean over $x$ values. Form the normal equations for this case and find the polynomial.
2. For the two cases above, parts 1 of this and the previous task, calculate the 2-norm condition numbers of the two matrices in the normal equations. You can use Python to do the calculations.
3. Assume the relative error in $\boldsymbol{y}$ (in 2-norm) is $\approx 10^{-3}$. In the worst case, what relative error in the solution of the normal equations (i.e. in the polynomial coefficients) can we get in both cases? Answer in percent.

**Workout 0.4.** If you would like to fit a 2nd-degree polynomial to the data above, what would the matrix $A$ and the normal equations look like? You do not need to solve the normal equation, but form the normal equations for this case. Again, you can use Python as a pocket calculator but write down the answers on paper.

**Workout 0.5.** Repeat parts 1 and 2 of Workout 2, but now based on the QR-factorization, $A = QR$. You can do the calculations in Python, but write down the steps you've done also on paper (you don't need to write down all the decimals, you can for example round to 2 decimal places). Use the "reduced" form of QR-factorization.

**Workout 0.6.** Assume that the first step of the Householder method is applied to a matrix $A$, resulting in the matrix $A^{(1)}$ below. Apply the second step to zero the elements in the second column of $A^{(1)}$ below $a_{22}$. Use Python as a calculator, but write down all the computation on paper, rounding to 3 decimal places.

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

**Non-mandatory workouts:**

**Workout 0.7.** The molecular weight (molar mass, g/mol) of nitrogen oxides are tabulated below (three decimal places accuracy):

| | | | | | |
|---|---|---|---|---|---|
| $NO$ | 30.006 | $N_2O$ | 44.013 | $NO_2$ | 46.006 |
| $N_2O_3$ | 76.012 | $N_2O_5$ | 108.010 | $N_2O_4$ | 92.001 |

Based on the data, do a least squares approximation to calculate the molecular weight for

nitrogen and oxygen. All data must be used.
*Hint:* Let $x$ and $y$ denote the molecular weight for nitrogen and oxygen, respectively. Using the table data we get $x + y = 30.006$, $2x + y = 44.013$, and so forth. That way, the table values will lead to six equations in two unknowns. Use QR factorization to obtain the solution. Use Python and write down all matrices and systems but round all numbers to 3 decimal places.

**Workout 0.8.** A time series have the following measured values:

| $t$ | 0 | 0.50 | 1.00 | 2.00 | 4.00 |
|---|---|---|---|---|---|
| $r$ | 2.01 | 1.79 | 1.42 | 0.99 | 0.42 |

It's known that correlation between $r$ and $t$ can be expressed as

$$r(t) = \alpha e^{\beta t}.$$

Find the best approximation to $\alpha$ and $\beta$ in least squares sense.
*Hint:* Take the logarithm of both sides in the expression and use it as the ansatz.

**Workout 0.9.** Assume that you work in an institute that do socioeconomic studies. One researcher at the institute has proposed a hypothesis that the apartment price $p$ in certain cities can *approximately* be expressed as

$$p = \frac{c}{r^2}$$

where $r$ is the distance between the apartment and the city centre. $c$ is a constant that depend on the particular city that is studied. Thus, $c$ is constant for a certain city, but differ between different cities.
As a new employee you are supposed to test the hypothesis, and find the $c$-value for Uppsala. Describe how you would set up the test, and outline a Python program for that find the $c$-value.

**Workout 0.10.** Show that the following properties hold true for the condition number:

1. For identity matrix, $\mathrm{cond}(I) = 1$
2. For any matrix A, $\mathrm{cond}(A) \geqslant 1$
3. For any matrix $A$ and scalar $\alpha$, $\mathrm{cond}(\alpha A) = \mathrm{cond}(A)$
4. For any diagonal matrix $D = \mathrm{diag}\{d_1, \ldots, d_n\}$, $\mathrm{cond}(D) = (\max |d_k| / \min |d_k|)$
5. If $Q$ is an orthogonal matrix then $\mathrm{cond}_2(Q) = 1$ (an important property)
6. If $A$ is square and non-singular then $\mathrm{cond}_2(A) = \mathrm{cond}_2(R)$ where $R$ is the triangular

factor in the QR factorization of $A$.