Part 1: Computed tomography

In this part we will look at a least-squares problem appearing in (a simplified version of) computed tomography (CT). The objective in CT is to determine the structure of a d-dimensional object (typically, a patient) from a series of $d-1$-dimensional X-ray pictures, which are taken from various angles, see Fig. 1.

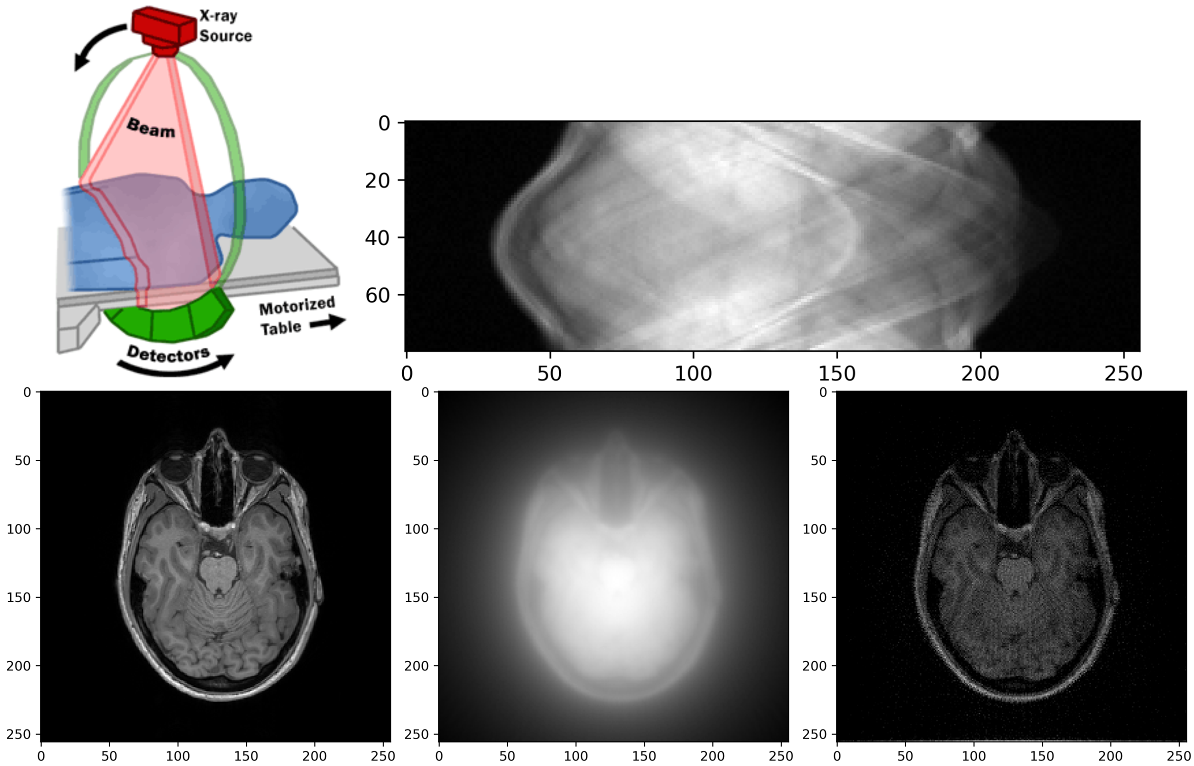

Figure 1: Illustration of the CT process. *Upper row, left-to-right*: drawing of a CT device; CT measurements, where each horizontal line represents one of 80 1-dimensional X-ray pictures of a 2-dimensional object. In this case, measurements contain 1% noise. All measurements are ultimately concatenated in a long vector b. *Lower row, left-to-right*: the unknown 2-dimensional object, which we are trying to reconstruct based on CT measurements; the right-hand side, $A^T b$, in the normal equations; the least squares reconstruction $\hat{x}$, which solves the normal equations $A^T A\hat{x} = A^T b$.

Let us consider a particularly simplified model. We focus on $d = 2$, so that our X-ray pictures are 1-dimensional. We assume that our object is given by $N \times N$ gray-scale pixels, and we take only 4 pictures of it from the angles $0°$, $90°$, $45°$, and $135°$, with the resolution of one X-ray per pixel. This situation is outlined in Fig. 2.

1. Based on the description in Fig. 2, find the matrix A representing our measurement model. That is, find A, such that the CT problem to determine $x = [x_1, x_2, x_3, x_4]$ from the measurements (without errors) reduces to solving a linear system $Ax = b$. What is the size of A?

2. Assuming now that the measurements contain unknown errors, i.e., that instead of the exact vector b we measure a different vector $\tilde{b} = b + f$. Explain why the system $Ax = \tilde{b}$ may lack solutions. Explain the concept "least squares problem" for such systems of linear algebraic equations.

3. Let us now look at an object consisting of $N \times N$ grayscale pixels, where $N \geq 2$. Assuming that we still only take 4 X-ray pictures as in Fig. 2, what are the dimensions of x, b, and A in this situation?

4. Continuing as in the previous question with a $N \times N$ object, utilize Theorem 8 from Section 1.7 in [Lay] to determine, for which $N \geq 2$ the columns in the matrix A are necessarily *linearly dependent*?

5. Explain how QR-factorization of a matrix A can be used to solve a least-squares problem $\min_x \|Ax - b\|^2$.

6. On moodle there are Python and Matlab scripts, illustrating how a QR factorization of a given matrix can be computed.[1] Compute a QR-factorization of the matrix A associated with our simplified CT

[1] They also illustrate, how to multiply matrices. To solve a system of linear algebraic equations, use "numpy.linalg.solve" in Python or "backslash" operator in Matlab.

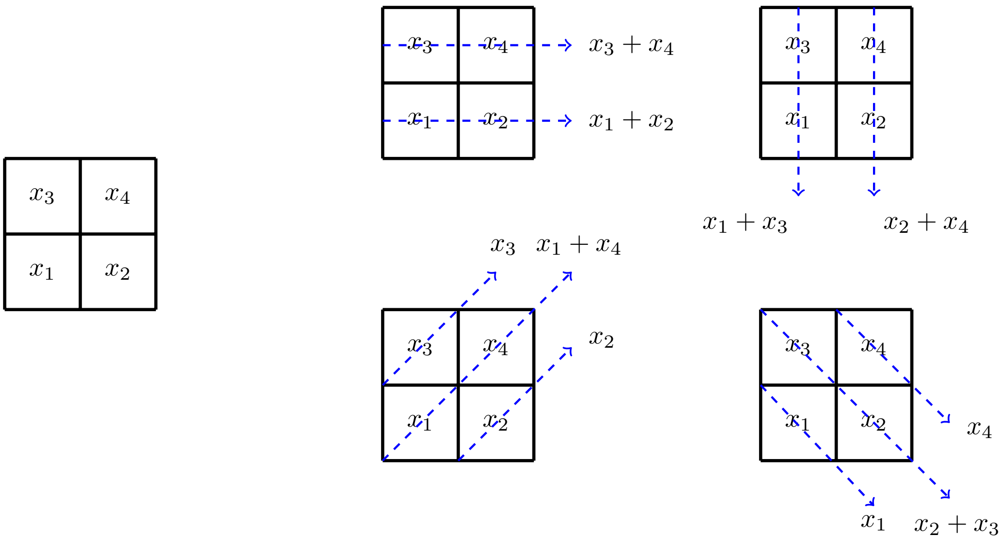

Figure 2: Our CT model with four "X-ray pictures" of a 2×2 object $[x_1, x_2, x_3, x_4] \in \mathbb{R}^4$ from four angles. In this model we measure (assuming that the measurements do not contain errors) a vector $b = [x_1 + x_2, x_3 + x_4, x_1 + x_3, x_2 + x_4, x_2, x_1 + x_4, x_3, x_1, x_2 + x_3, x_4] \in \mathbb{R}^{10}$. In reality, typically we will measure a vector $\tilde{b} = b + f \in \mathbb{R}^{10}$, where $f \in \mathbb{R}^{10}$ is a vector with unknown/random measurement errors.

measurement model for a $N \times N$ object with $N = 2$. Furthermore, using this information compute the orthogonal projection of the vector $b = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] \in \mathbb{R}^{10}$ onto the linear subspace spanned by the columns of A (recall that $\mathrm{col}(A) = \mathrm{col}(Q)$).

7. Find the solution to the least-squares problem associated with our simplified CT model for a $N \times N$ object with $N = 2$ corresponding to the measurements $b = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] \in \mathbb{R}^{10}$. Use two alternative methods: solving the normal equations and utilizing the QR-factorization.

Part 2: QR-factorization using Given's rotations

Consider a nonzero vector $[a, b] \in \mathbb{R}^2$. Given's rotation is an orthogonal transformation (as the name suggests, a rotation) which maps $[a, b]$ to a vector $[d, 0] \in \mathbb{R}^2$ aligned with the first coordinate axis.

1. Use Theorem 7 in Subsection 6.2 [Lay], to determine $|d|$ given $[a, b]$.

2. Verify that
$$G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{1}$$
where $c = a/\sqrt{a^2 + b^2}$ and $s = b/\sqrt{a^2 + b^2}$ is a Given's rotation. That is, check that G is an orthogonal matrix that maps $[a, b]$ to $[d, 0]$.

3. Let us now consider a vector $x \in \mathbb{R}^m$, $m \geq 2$, such that $x_i = a$ and $x_j = b$, $i < j$. We compute c and s as in the previous question. Let now $G(i, j, a, b)$ be an $m \times m$ matrix, with all rows/columns as in the identity matrix, with the exception of the rows/columns i and j, where we "insert" the matrix G from (1), i.e.: $G(i, j, a, b)_{ii} = G(i, j, a, b)_{jj} = c$, $G(i, j, a, b)_{ij} = -G(i, j, a, b)_{ji} = s$:
$$G(i, j, a, b) = \begin{bmatrix} I & & & & \\ & c & & s & \\ & & I & & \\ & -s & & c & \\ & & & & I \end{bmatrix}. \tag{2}$$

Verify that $G(i, j, a, b)$ is an orthogonal matrix, and that

$$G(i,j,a,b)x = \begin{bmatrix} x_1 \\ \vdots \\ x_{i-1} \\ d \\ x_{i+1} \\ \vdots \\ x_{j-1} \\ 0 \\ x_{j+1} \\ \vdots \\ x_m \end{bmatrix} \tag{3}$$

4. Explain why a product of square orthogonal matrices is itself an orthogonal matrix. I.e., assuming that $Q_1, Q_2, \ldots, Q_k$ are orthogonal matrices, explain why the matrix $Q_1 Q_2 \cdots Q_k$ is orthogonal.

5. Consider the following algorithm, where the matrix A of size $m \times n$ is given.

   $Q := I_{m\times m}$; $R := A$;
   for $i = 1, \ldots, n$ **do**
     for $j = i + 1, \ldots, m$ **do**
       $a = R_{ii}$; $b = R_{ji}$;
       if $b \neq 0$ **then**
         $R := G(i, j, a, b)R$; $Q := QG(i, j, a, b)^T$;
       end if
     end for
   end for

   where $G(i, j, a, b)$ is given by (2).

   (a) Use the previous questions to explain, why the matrix Q remains orthogonal throughout all iterations of the algorithm.

   (b) Explain why the equation $A = QR$ is fullfilled throughout all iterations of the algorithm.

   (c) Use (3) to explain, why after the algorithm terminates, the computed matrix R is upper triangular.