Winfried Auzinger
Dirk Praetorius

Sommersemester 2015
20.–22.05.2015

Übungsaufgaben zur VU Computermathematik
Serie 8

In all examples we use the package `LinearAlgebra` *and the data types* `Vector` *and* `Matrix`. *Some of these exercises also serve to illustrate how [operations on] vectors and matrices are denoted and handled in numerical linear algebra; see also the exercises on* MATLAB. *In particular, column vectors are often identified with $n \times 1$ matrices, and row vectors are identified with $1 \times n$ matrices, Here, only the case of real vectors and matrices is considered.*

It is assumed that you are familiar with basic properties of the Euclidean inner product $u \cdot v$ and its geometric meaning in $\mathbb{R}^2$ and $\mathbb{R}^3$. Two vectors u, v are called orthogonal, $u \perp v$, if $u \cdot v = 0$.

Vectors u are generally to be understood as columns vectors, and u^T is the corresponding row vector. If u and v have the same dimension, $v^T u = u^T v$ is the dot product (Euclidean inner product) $u \cdot v$. For arbitrary dimensions, $u\,v^T$ is the outer product (or dyadic product), which is a matrix.

$\|u\| = \sqrt{u \cdot u} = \sqrt{u^T u} = \sqrt{\sum_i u_i^2}$ *is the Euclidean norm of a vector u.*

Several exercises are based on assertions from linear algebra which you may be aware of (or not). Some of these assertions are easy to prove; others not. You may try to think about some of these proofs, but this is not essential here. For special cases one may give a (brute-force) 'computer-aided proof'; see for instance Exercise 8.4 b).

'Verify' means: verify by testing on examples.

Exercise 8.1. *Investigation of a parameter-dependent matrix.*

Consider the matrix

$$A = \begin{pmatrix} 0 & a & 1 & 0 & b \\ 1 & 0 & 0 & b & 0 \\ 0 & 1 & b & 0 & 1 \\ b & 0 & 0 & 1 & 0 \\ 0 & b & 1 & 0 & b \end{pmatrix}$$

depending on two parameters a and b. Use Maple / `LinearAlgebra`:

a) For which values a, b is A invertible? Determine the inverse of A.

b) Same question as in **a)**, for the symmetric part $(A + A^T)/2$ instead of A.

c) Same question as in **a)**, for the skew-symmetric part $(A - A^T)/2$ instead of A.

Exercise 8.2. *Basic operations with vectors and matrices.*

a) *Assertion: Given two vectors $0 \neq u, v \in \mathbb{R}^n$, the rank of the $n \times n$ matrix $u\,v^T$ is 1.*

- 'Verify' this for the case $n = 3$ and arbitrary vectors $u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$, $v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$. Furthermore, compute a basis for the kernel (also called *nullspace*) of this matrix, and comment on the result. (You may begin with $n = 2$.)

  Hint: Use `Rank` and `NullSpace`.

b) (**a)** continued:) 'Verify' the elementary identity $(u\,v^T)\,x = (v^T x)\,u$ for vectors $u, v, x \in \mathbb{R}^n$. Also, explain why this identity holds true.

c) *Assertion: Given two column vectors $u, v \in \mathbb{R}^n$ satisfying $v^T u \neq 1$, the $n \times n$ matrix $I - u\,v^T$ is invertible, with*

$$\left(I - u\,v^T\right)^{-1} = I - \frac{u\,v^T}{v^T u - 1}$$

- 'Verify' this identity for the case $n = 3$ and arbitrary symbolic vectors $u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$, $v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$.

  (You may begin with $n = 2$.)

d) Design two functions which expect three vectors `u,v,x` as its arguments and which evaluate

$$\left(I - u\,v^T\right)x \quad \text{and} \quad \left(I - u\,v^T\right)^{-1}x$$

without explicitly forming the matrices.

Exercise 8.3. *Understanding orthogonal projectors.*

For the numerical computations, use simple data and exact arithmetic (not `evalf`*).*

a) Design a procedure `Umatrix(ulist)` which expects a list `ulist` consisting of $1 \le m \le n$ column vectors $u_i \in \mathbb{R}^n$ as its argument and returns the matrix

$$U = \left(\begin{array}{c|c|c|c} u_1 & u_2 & \ldots & u_m \end{array} \right) \in \mathbb{R}^{n\times m}$$

In the following, let $u_1, \ldots, u_m$ be a set of $1 \le m \le n$ orthonormal vectors in $\mathbb{R}^n$, i.e., $u_i \cdot u_j = u_j^T u_i = \delta_{ij}$.

b) *Assertion: The matrix* $P := UU^T$ *represents an orthogonal projector onto the* m*-dimensional subspace* $\{\mathcal{U} := \lambda_1 u_1 + \ldots + \lambda_m u_m,\ \lambda_i \in \mathbb{R}\}$ *of* $\mathbb{R}^n$*, i.e.,* $P\,x = x$ *for* $x \in \mathcal{U}$ *and* $P\,x = 0$ *for* $x \perp \mathcal{U}$*.*

- Choose two orthonormal numerical vectors $u_1, u_2 \in \mathbb{R}^3$ and illustrate the behavior of the mapping $x \mapsto P\,x$ for some numerical vectors x. What is the rank of P? Also, verify $P = P^T = P^2$. What is $U^T U$?

c) 'Verify' the identity [1]

$$P\,x = \sum_{i=1}^{m} (x^T u_i)\,u_i \quad \text{for} \quad x \in \mathbb{R}^n\,.$$

and implement evaluation of $P\,x$ in this way, without explicitly forming the matrix P.

d) *Assertion: The matrix* $Q := I - UU^T$ *represents an orthogonal projector onto the* $(n-m)$*-dimensional orthogonal complement of* $\mathcal{U}$*, i.e.,* $Q\,x = 0$ *for* $x \in \mathcal{U}$ *and* $Q\,x = x$ *for* $x \perp \mathcal{U}$*.*

- Choose two orthonormal numerical vectors $u_1, u_2 \in \mathbb{R}^3$ (see **b)**) and illustrate this behavior of the mapping $x \mapsto Q\,x$ for some numerical vectors x. What is the rank of Q? Also, verify $Q = Q^T = Q^2$.

Exercise 8.4. *A formula for the inverse of a matrix after a low-rank perturbation.*

Let $A \in \mathbb{R}^{n\times n}$ be invertible, and $U, V \in \mathbb{R}^{n\times k}$. Then, the *Sherman-Morrison-Woodbury (SMW)* formula holds: $A + UV^T \in \mathbb{R}^{n\times n}$ is invertible if and only if $I + V^T A^{-1} U \in \mathbb{R}^{k\times k}$ is invertible, with [2]

$$\left(A + UV^T\right)^{-1} = A^{-1} - A^{-1}U\left(I + V^T A^{-1} U\right)^{-1} V^T A^{-1}\,.$$

a) Implement this formula in form of a procedure

`SMW_inverse(AI::Matrix,U::{Matrix,Vector[column]},V::{Matrix,Vector[column]})`

In `AI`, the given inverse A^{-1} is passed. For the case $k = 1$, admit that `U,V` are specified in form of objects of type `Vector[column]` instead of `Matrix` and treat the case $k = 1$ separately (1D inverse!).

b) Try to give a computer-aided proof of the SMW formula for the case $n = 2$ and $k = 1$, i.e., for a symbolic 2×2 matrix `A` and two symbolic column vectors $\texttt{U,V} \in \mathbb{R}^2$. (You may also try the case $n = 3$, $k = 1$.)

c) Choose a numerical example (e.g., $n = 9$, $k = 3$) and compare with direct inversion. Use floating point arithmetic (`evalf`).

[1] From this identity you can understand why UU^T is an orthogonal projector.

[2] See **2 c)** for a special case. The SMW formula can be used to compute the inverse $\left(A + UV^T\right)^{-1}$, assuming A^{-1} is already known. The additional effort involves only a smaller inverse $\left(I + V^T A^{-1} U\right)^{-1} \in \mathbb{R}^{k\times k}$, and using the SMW formula is more efficient than direct inversion of $(I + V^T A^{-1} U)$ if $k \ll n$ (i.e., if the perturbation UV^T is of low rank $\le k \ll n$).

Exercise 8.5. *Playing with determinants.*

The well-known formula for the determinant of a 2×2 matrix generalized as follows: Consider a matrix block-partitioned according to

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathbb{R}^{n \times n},$$

where $A \in \mathbb{R}^{k \times k}, \ D \in \mathbb{R}^{(n-k) \times (n-k)}, \ k < n.$

Assertion: Suppose A is invertible. Then the determinant of M equals [3]

$$\det(M) = \det(A)\,\det(D - C\,A^{-1}\,B)\,.$$

- 'Verify' this identity for an example of your choice with integer coefficients. Use `Determinant`.
- An other variant reads (for D invertible)

$$\det(M) = \det(D)\,\det(X)$$

What is X? Think about it and test.

Exercise 8.6. *Quadratic forms on* $\mathbb{R}^2$.

For a (symmetric) 2×2 matrix A, the function $f\colon \mathbb{R}^2 \to \mathbb{R}$ defined by

$$q(x) = (A\,x)^T x$$

is called a *quadratic form.* It is a bivariate polynomial in the variables x_1 and x_2 $(x = (x_1, x_2))$.

a) Design a function `q(A,x1,x2)` which evaluates this quadratic form for given $x =$ (`x1`, `x2`).

Remark: Choosing the names `x,y` instead of `x1,x2` will be more convenient here.

b) For given $c \in \mathbb{R}$, the solutions x of the equation $q(x) = c$ are located on a conic section (*Kegelschnitt*) in the plane.

- Choose several examples (i.e,, choose A and c), and use your function `q` from **a)** and `plots[implicitplot]` to visualize the corresponding conic section.

Hint: When using `implicitplot`, increasing the value of the parameter `numpoints` may be essential to obtain a good resolution.

Exercise 8.7. *Visualization of linear mappings.*

a) With `plots[arrow]` you can draw arrows. Use this to visualize the behavior of a linear mapping $\psi\colon \mathbb{R}^3 \to \mathbb{R}^3$ represented by a coefficient matrix A, by drawing the parallelepiped spanned by the image of the unit vectors $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ under the mapping. Choose an example and produce a nice plot.

b) $(*)$ Another visualization is provided by the image of the unit sphere under the mapping. To this end, use spherical coordinates

$$x = \cos\theta\,\cos\varphi,$$
$$y = \cos\theta\,\sin\varphi,$$
$$z = \sin\theta,$$

with $\varphi \in [0, 2\,\pi]$ and $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, and use `plot3d`.

Produce a nice plot. Also use `display[3d]` to combine this with a plot of the unit sphere. Use different colors and set the option `transparency=0.5`.

Hint: With `convert(...,list)` you can convert a `Vector` into a list.

Exercise 8.8. *Compressed representation of sparse matrices.* [4]

a) Check the help page for `LinearAlgebra[CompressedSparseForm]`, understand what it means, and explain by means of an example.

Remark: This only works for matrices where the entries are of hardware type. Use double precision (`datatype=hfloat`).

b) Same as **a)**, for `LinearAlgebra[FromCompressedSparseForm]`.

c) $(*)$ Implement matrix-vector multiplication assuming that the matrix is given in compressed sparse form. (The vector is assumed to be of the normal type `Vector`.)

---

[3] $D - C\,A^{-1}\,B$ is called the *Schur complement* of A.

[4] This works in a similar way as storage of sparse matrices in MATLAB.