# Exercises for the MATLAB catch-up course: Linear algebra, signals, and systems

Ivan Markovsky

## Contents



## 1 Matrix/vector manipulation

### Linear progression

Construct the vector $\begin{bmatrix} 100 & 99 & \cdots & 1 \end{bmatrix}$.

### Geometric progression

Construct the vector $\begin{bmatrix} 2^0 & 2^1 & 2^2 & \cdots & 2^9 \end{bmatrix}$.

### Inserting columns in a matrix

Given matrices `a` and `b` of the same dimension, create a matrix `c` that consists of the interlaced columns of `a` and `b`, *i.e.*, `c(:, 1)` is `a(:, 1)`, `c(:, 2)` is `b(:, 1)`, `c(:, 3)` is `a(:, 2)`, `c(:, 4)` is `b(:, 2)`, *etc.*

### Flipping a matrix

Flip a given matrix left-right and upside-down.

### Thresholding a matrix

Replace all elements in a matrix that are smaller than a specified threshold by zeros.

### Adding a vector to all columns of a matrix

Add a given vector `v` to all columns of a given matrix `a`.

### Hankel matrix

Given a vector `v` of size `m + n - 1` and an integer `m`, construct an `m`-by-`n` matrix which `(i,j)`-th element is `v(i + j - 1)`.

### Toeplitz matrix

Given a vector `v` of size `m + n - 1` and an integer `m`, construct a matrix which `(i,j)`-th element is `v(i - j + n)`.

### Vandermonde matrix

Given a vector `v` of size `n` and an integer `m`, construct a matrix which `(i,j)`-th element is `v(j) ^ (i - 1)`.

## 2 Solving equations

### General solution of systems of linear equations

Solve the systems of linear equations

1. $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} x = 0$
2. $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$

### Solving a polynomial equation (rooting a polynomial)

Solve the polynomial equation

$$x + 2x^2 + 3x^3 + 4x^4 = 5.$$

### Particular solution of an ordinary differential equation (initial value problem)

Solve the ordinary differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t} y - (1 - y^2) y + y = 0,$$

over the interval `[0 10]`, starting from the initial condition `y(0) = 1` and `dy(0) = 1`.

## 3 Computational cost

- How many scalar multiplications requires the computation of $A^{100}$ by direct multiplication $A \cdots A$ for a $2 \times 2$ matrix $A$? Suggest a faster method. Apply the method on $A = \begin{bmatrix} -1/4 & 1/4 \\ -3/2 & 1 \end{bmatrix}$.
- The function `dftmtx` constructs a matrix representation $F$ of the discrete Fourier transform.
  - What is the number of multiplications needed to compute the DFT $\widehat{x} = Fx$ by matrix-vector multiplication? Compare this number with the $n \log_2(n)$ multiplications needed for the same computation by the fast Fourier transform.
  - Using Matlab's `tic` and `toc` functions, measure the computation times of the matrix-vector multiplication and the fast Fourier transform methods for the computation of the discrete Fourier transform of $x$. (Use a random $x$.) Repeat the experiment for different size $n$ of $x$ and plot the results. Do the empirical observations match the theoretical predictions?
- Computation time for the convolution operation
  1. Find a matrix representation $M_h$ of the convolution $h \star x$ of $h \in \mathbb{R}^n$ and $x \in \mathbb{R}^n$.
  2. What is the number of multiplications needed to compute $y = M_h x$ by matrix-vector multiplication?
  3. Propose a fast method for convolution based on the fast Fourier transform. What is the computational cost of this method?

4. Measure the computation times of the matrix-vector multiplication and the fast Fourier transform methods for the computation of the convolution

$$y = \exp_\lambda \star x, \qquad \text{where } h(t) = \exp_\lambda(t) = e^\lambda t.$$

(Use a random $x$.) Repeat the experiment for different size $n$ and plot the results. Do the empirical observations match the theoretical predictions?

5. For the special case of exponential $h$, can you propose another fast method?

# 4 Identification of a linear function

Let $\mathbb{R}^n$ be the $n$-dimensional real vector space. A linear function $f$ from $\mathbb{R}^m$ to $\mathbb{R}^n$ has a matrix representation $f(x) = Ax$, where $A$ is an $m \times n$ real matrix. The problem is to find $A$ from observed data

$$\mathscr{D} := \{ (x^1, y^1), \dots (x^N, y^N) \}, \qquad \text{where } y^i = f(x^i). \tag{1}$$

1. Give conditions under which it is possible to find $A$ from $\mathscr{D}$. Describe a computational method that does the job. Implement the method in a language of your choice and test it on an example.

2. If you can choose the points $x^1, \dots, x^N$ how many and what points would you choose?

3. If the $y^i$'s are corrupted by additive zero mean, uncorrelated, Gaussian noise $e^i$, *i.e.*, in (1)

$$y^i = f(x^i) + e^i, \qquad \text{for } i = 1, \dots, N,$$

how would you estimate $A$ from $\mathscr{D}$?

# 5 Empirical validation of estimator's consistency property

Consider the standard *linear model* $Ax = b$, where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ is the given data and $x$ is the to-be-found *parameter*. Assume that $A$ is "exact" and $b$ is "noisy", *i.e.*, there is a "true" values $\bar{b}$ of $b$ such that $b = \bar{b} + e$ and $A\bar{x} = \bar{b}$. Here $e$ is the *measurement noise* and $\bar{x}$ is the *true value of the parameter* $x$. The goal is to estimate $\bar{x}$ from the data $(A, b)$. A procedure that maps $(A, b)$ to the estimate $\widehat{x}$ is called an *estimator*. Assuming that $m > n$ and $A$ is full rank, the *least squares estimator* is given by $\widehat{x} = (A^\top A)^{-1} A^\top b$.

Consistency is the property of an estimator $\widehat{x}$, that it converges to the true value $\bar{x}$ as the *sample size* $m$ goes to infinity. An important result in least-squares estimation is that the least squares estimator is consistent in the linear model assuming that the error $e$ is zero mean Gaussian with a covariance matrix that is a multiple of the identity. The task of this exercise is to demonstrate the consistency of the least-squares estimation *empirically*, i.e., on simulation examples. For this purpose:

1. generate true data according to the model (choose a random model),

2. estimate the parameter of interest for different noise realizations, and

3. plot the average estimation error $\|\bar{x} - \widehat{x}\|$ as a function of the sample size $m$.

In order to make sure that the result is representative, vary the simulation parameters (e.g., the $n$ and the $A$) and recompute the error vs number of samples curve.

The theoretical convergence rate of the least-squares estimator is $1/\sqrt{m}$. Verify that it is consistent with the empirical results, *i.e.*, fit the function $c/\sqrt{m}$ to the empirical estimation error.