Tutorials for Lecture Course on Modeling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2016-2017

Emergency Guide to Linear Algebra:
Recall of important Matrix Properties and Operations

Prof. Dr. Moritz Diehl, Robin Verschueren, Tobias Schöls, Rachel Leuthold, and Mara Vaihinger

1 Motivation (or why would you do this?)

Matrices are common in many fields of engineering, i.e. measurements are often stored as a matrix, for example series of voltage measurements. On top of that formulating the math that is used to process these data as matrix operations is usually more compact and convenient. Therefore you will have to deal with matrices a lot during this course. However we understand that matrices might not be intuitive for everyone, especially if you have not dealt with them in a long time. This tutorial is meant to get you used to working with matrices (again).

Please do not hesitate to ask us any questions!

1.1 Warm-Up Exercises

The following exercises are meant to refresh your memory and get you used to matrices again. We recommend to calculate the tasks by hand first and then check the result using `MATLAB`.

$$A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 7 \\ 8 & 6 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$(A+B)v = \tag{1}$$

$$Av + Bv = \tag{2}$$

$$(A+B)C = \tag{3}$$

$$AC + BC = \tag{4}$$

$$CA + CB = \tag{5}$$

$$AA^{-1} = \tag{6}$$

$$v^{\mathrm{T}}v = \tag{7}$$

$$vv^{\mathrm{T}} = \tag{8}$$

$$A(BC) = \tag{9}$$

$$(AB)C = \tag{10}$$

$$CBA = \tag{11}$$

$$A^{\mathrm{T}} = \tag{12}$$

$$(Av)^{\mathrm{T}} = \tag{13}$$

$$v^{\mathrm{T}}A^{\mathrm{T}} = \tag{14}$$

$$v^{\mathrm{T}}A^{\mathrm{T}}Av = \tag{15}$$

$$\sum_{i=1}^{2} v_i = \tag{16}$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix} v = \tag{17}$$

2 Linear Algebra for MSI

To mix things up a bit we will use an example that will take us through the material: Say hi to Max!
Max is a passionate bicyclist who likes to record his tracks using a GPS device that records his position every 5 seconds [s]. The position is captured in three dimensions, i.e. x, y, and z (longitude, latitude, and altitude) all in meters [m].
He imported the measurements from his last trip into `MATLAB` and shared them with us to help him with the analysis.

Please start `MATLAB` now, open `linearAlgebra.m` and run the first section to see how far he got.
In the `MATLAB` Tutorial Max learned how to plot his data. He recorded data for a series of times t_k, these he stored in a variable called `timeStamps`, the GPS positions he stored in `positions`.
The positions Max recorded can be understood as evaluations of a function $\mathbf{f}$ times t_k [s]. Written in mathematical terms

$$\mathbf{f}(t) = \begin{bmatrix} f_x(t) \\ f_y(t) \\ f_z(t) \end{bmatrix}$$

where $\mathbf{f}(t)$ is Max's 3-D position for a given time t. Based on his measurements Max wants to compute his speed. He recalls that velocity is the first derivative of the position with respect to time from which he can then compute his speed.

2.1 Advanced Matrix Operations

Derivatives

Derivatives are very common and have many applications. Please recall that a derivative is always with regard to some variable, e.g. for a function $f(t)$ the total derivative with respect to t (e.g. time) is defined as $\frac{\mathrm{d}f}{\mathrm{d}t}$. For a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ we define the derivative with respect to its parameter vector $\mathbf{x}$ as (instead $\mathbf{f}(\mathbf{x})$ we write $\mathbf{f}$ here for cleaner notation)

$$\nabla \mathbf{f} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

The matrix above is called the Jacobian matrix. Based on this definition we can derive a number of properties for different functions $\mathbf{f}$. The list below includes some important rules that will be handy for this course. Let A be a matrix of appropriate size.

$$\begin{aligned}
\mathbf{f} = \mathbf{x} : \quad & \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}} = \mathbb{I}_n \\
\mathbf{f} = A\mathbf{x} : \quad & \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}} = A \\
\mathbf{f} = \mathbf{x}^{\mathrm{T}} A\mathbf{x} : \quad & \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}} = \mathbf{x}^{\mathrm{T}} A + (A\mathbf{x})^{\mathrm{T}} = \mathbf{x}^{\mathrm{T}}(A + A^{\mathrm{T}}) \\
\mathbf{f} = \mathbf{x}^{\mathrm{T}} A^{\mathrm{T}} A\mathbf{x} : \quad & \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}} = \mathbf{x}^{\mathrm{T}} A^{\mathrm{T}} A + (A^{\mathrm{T}} A\mathbf{x})^{\mathrm{T}} = 2\mathbf{x}^{\mathrm{T}} A^{\mathrm{T}} A
\end{aligned}$$

Back to Max, he has read the above and wrote down the following

$$\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}t} = \begin{bmatrix} \frac{\mathrm{d}f_x}{\mathrm{d}t} \\ \frac{\mathrm{d}f_y}{\mathrm{d}t} \\ \frac{\mathrm{d}f_z}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial x}\frac{\partial x}{\partial t} \\ \frac{\partial f_y}{\partial y}\frac{\partial y}{\partial t} \\ \frac{\partial f_z}{\partial z}\frac{\partial z}{\partial t} \end{bmatrix}$$

also figured out parts of the implementation, but he is missing the derivation step. Therefore he left a gap in the `Derivatives`-section, please fill this gap for him.
Hint: look up the `diff` command. $\nabla\mathbf{f}$ can be computed with a single `MATLAB`-command (first difference, dimension 2).
Great, now Max knows his speed per direction (x, y, z)! But he is actually interested in his total speed and he heard that this can be computed using norms.

Norms

In linear algebra norms are functions that compute the length or the size of a vector. For this class only two norms will be used:

Euclidean norm
Most common norm definition, i.e. straight-line distance between two points. (Here the point $\mathbf{x}$ and the origin)

$$\begin{aligned}
\|\mathbf{x}\|_2 &= \sqrt{x_1^2 + \cdots + x_n^2} \\
\|\mathbf{x}\|_2^2 &= x_1^2 + \cdots + x_n^2 = \mathbf{x}^{\mathrm{T}}\mathbf{x}
\end{aligned}$$

1-norm

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

Which norm would you recommend Max to use and why?

Implement the norm you chose in the `Norms`-section of the script and plot the speed over time. *Hint:* recall what the `.`-operator does and look up the following `MATLAB`-commands: `sqrt`, `abs`

Optional: Can you calculate the length of Max's bike trip and his average speed?

To monitor his speed during the ride Max bought a bike computer that measures the covered distance by counting the full rotations of the front wheel. For accurate speed measurements Max needs to input the radius of his front wheel. To figure it out he makes another tour with front wheel radius in the bike computer set to 27.5cm. After his trip the bike computer detected 5342 full rotations of the front wheel and the distance measured by on the bike computer was $2\,\text{km}$ less than the GPS track.
Max wants to formulate his problem as a linear equation system of the form $Ax = b$, please fill in the gaps for him:

$$\underbrace{\begin{bmatrix} \dots\dots\dots\dots & \dots\dots\dots\dots \\ \dots\dots\dots\dots & \dots\dots\dots\dots \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} r_{\text{actual}} \\ r_{\text{initial}} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} \dots\dots \\ \dots\dots \end{bmatrix}}_{b} \tag{18}$$

Max is wondering if this equation system is solvable (uniquely). Read the following section to find out how this can be determined.

Rank of a Matrix

The rank of a matrix is the number of linear independent rows. This is equivalent to saying the rank of a matrix is the number of independent columns. A matrix is said to have full rank if all rows or columns are linearly independent, that is the rank matches the dimension of that matrix. For linear equation systems this means that a unique solution exists.
The rank can be computed with the `MATLAB`-command `rank()`.

Examples

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{does not have full rank, since it contains a zero row.}$$

$$\begin{bmatrix} 3 & 4 & 1 \\ 5 & 7 & 9 \\ 6 & 8 & 2 \end{bmatrix} \quad \dots \tag{19}$$

$$\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \quad \dots \tag{20}$$

Determine if the matrix for Max's problem has full rank.

$$\text{rank}(A) = \dots \tag{21}$$

Inverse

A square matrix $A \in \mathbb{R}^{n \times n}$ is called invertible if there exists a $B \in \mathbb{R}^{n \times n}$ such that

$$AB = BA = \mathbb{I}_n$$

where $\mathbb{I}_n$ is a n-by-n identity matrix. If B exists it is unique and called the inverse of A, denoted by A^{-1}.
Square matrices that fulfill at least one of the following properties have an inverse:

- full rank (*non-degenerate*)
- determinant is not zero (*non-singular*)
- only non-zero eigenvalues

There are several other criteria. However, they are not relevant for this course and are therefore not listed here. Non square matrices do not have an inverse.
The `MATLAB`-command to calculate the inverse is `inv()`.

With the above Max can solve his problem. He did the following calculation

$$Ax = b \Leftrightarrow A^{-1}Ax = A^{-1}b \Leftrightarrow \mathbb{I}x = A^{-1}b \Leftrightarrow x = A^{-1}b$$

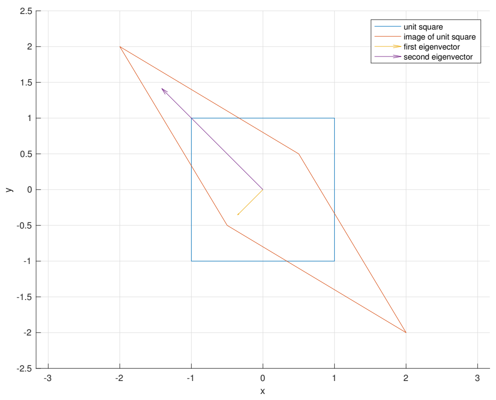

Figure 1: Deformation of a unit square through example transformation.

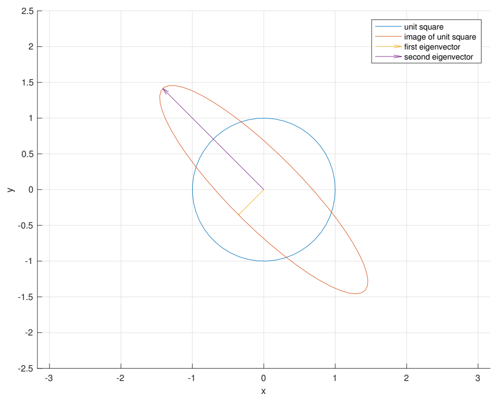

Figure 2: Deformation of a unit circle through example transformation.

Please calculate x (in `MATLAB`). What is the actual radius of Max's front wheel?

$$r_{\text{actual}} = \dots \tag{22}$$

Max is happy and takes off for a trip to a near by Casino. (Learn about his business there in the next tutorial.) But before we can follow him there we have to deal with some more stuff.

Eigenvalues and Eigenvectors

Vectors that do not change the direction when multiplied with A are called *eigenvectors* here denoted as $\mathbf{v}$. When A is multiplied with one of its eigenvectors the result is just a scalar multiple of that eigenvector. This can be formulated in a formula as

$$A\mathbf{v} = \lambda\mathbf{v}$$

where $\mathbf{v}$ is an eigenvector and λ is the corresponding *eigenvalue*.[1]

Ok, great! - And now what does all this mean?

Consider the following equation $Ax = b$ where A is defined as $A = \begin{bmatrix} 1.25 & -0.75 \\ -0.75 & 1.25 \end{bmatrix}$. This is a linear map from $\mathbb{R}^2$ to $\mathbb{R}^2$. Its eigenvalues are $\lambda_1 = 0.5$ and $\lambda_2 = 2$ the corresponding eigenvectors are $v_1 = -\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \end{bmatrix}^{\mathrm{T}}$ and $v_2 = -\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \end{bmatrix}^{\mathrm{T}}$. In the 2-D case this can be visualized by the deformation of a unit circle (figure 2) and a unit square (figure 1).

From math class you may remember that the eigenvalues are the roots of the characteristic polynomial. For this class you do not need to compute them by hand and you can rely on `MATLAB` to find them for you in the exercises. The `MATLAB`-command to calculate eigenvalues and eigenvectors is called `eig()`. Please find out how the results are returned.

Fill in the gaps in the `Eigenvalues and Eigenvectors`-section of the `MATLAB` script.

Optional: Come up with your own matrix and plot the deformation of the unit square and the unit circle, similar to the figures above.

Outlook

Throughout this course you will be dealing with some random variables that follow a certain distribution. The spread of this distribution and the correlation between different variables is contained in a covariance matrix. To visualize the spread of distributions and the relation between different variables you will use deformed unit circles (similar to what you have seen above). These ellipses are called confidence ellipses. [Don't panic! You have seen this already.] Take a look at the figure 3 below. The thing I am talking about is the light blue circle around the blue dot that marks the (most likely) position.

Matrix exponential

The analogous to the exp-function is the *matrix exponential*. For this class it is not important to calculate the matrix exponential by hand, however you will need the `MATLAB`-command which is `expm()`.

[1] `http://math.mit.edu/~gs/linearalgebra/ila0601.pdf`

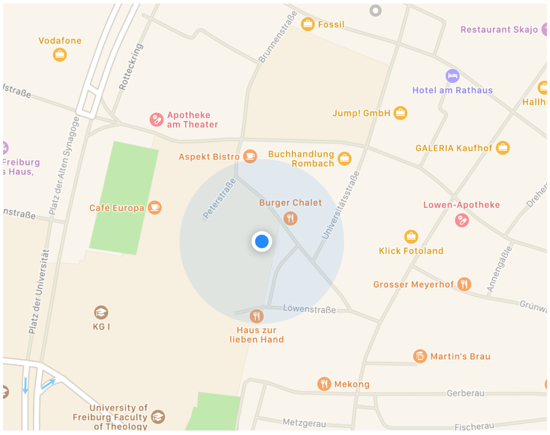

Figure 3: Example Confidence Ellipse: location estimate and area in which the actual position is most likely to lie in.

2.2 Special Matrices

Symmetric Matrices

A matrix is called *symmetric* if it is equal to its transpose.

$$A = A^{\mathrm{T}}$$

Please note that only square matrices can be symmetric and that the product of a matrix with its transpose is symmetric. Thus for any $B \in \mathbb{R}^{m \times n}$ holds:

$$B^{\mathrm{T}}B = B^{\mathrm{T}}(B^{\mathrm{T}})^{\mathrm{T}} \overset{\text{with } (AB)^{\mathrm{T}} = B^{\mathrm{T}}A^{\mathrm{T}}}{=} (B^{\mathrm{T}}B)^{T}$$

Examples

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

$$\begin{bmatrix} a & b & \dots \\ \dots & d & e \\ c & \dots & f \end{bmatrix} \text{(fill in the gaps)} \tag{23}$$

$$\begin{bmatrix} v \\ w \end{bmatrix} \begin{bmatrix} v & w \end{bmatrix} = \dots \tag{24}$$

$$\begin{bmatrix} v & w \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \dots \tag{25}$$

Optional: Calculate the determinant of the matrices above and the eigenvalues for the first two matrices. You can use the `MATLAB`-symbolic-toolbox to check your results. The `MATLAB`-command to compute the determinant is called `det()`.

Positive / Negative (Semi-)Definite Matrices

If a symmetric matrix has no negative eigenvalue (all are positive or zero) it is called *positive semi-definite* (PSD). The same holds for *positive definite* matrices only that the zero is not allowed as eigenvalue.
Similarly a *negative definite* matrix has only strictly negative eigenvalues and a *negative semi-definite* has no positive eigenvalue (all negative or zero).
An alternative definition of positive/ negative (semi-)definiteness is, for any symmetric matrix $M \in \mathbb{R}^{n \times n}$ and all non-zero vector $x \in \mathbb{R}^n$ holds that it is

Negative-definite if $x^{\mathrm{T}}Mx < 0$

Negative-semi-definite if $x^{\mathrm{T}}Mx \leq 0$

Positive-definite if $x^{\mathrm{T}}Mx > 0$

Positive-semi-definite if $x^{\mathrm{T}}Mx \geq 0$

Indefinite if none of the above is true.

Determine if the matrices below are positive semi-definite and give a short reason:

$$\begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} \tag{26}$$

$$\begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix} \tag{27}$$

$$\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \end{bmatrix} \tag{28}$$

$$\begin{bmatrix} 8 & 3 \\ 1 & 6 \\ 6 & 7 \end{bmatrix} \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \end{bmatrix} \tag{29}$$

$$\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \end{bmatrix} \begin{bmatrix} 8 & 3 \\ 1 & 6 \\ 6 & 7 \end{bmatrix} \tag{30}$$

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \tag{31}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{32}$$

Some important properties of positive semi-definite matrices:

- For any matrix $A \in \mathbb{R}^{m \times n}$ holds that $A^{\mathrm{T}}A$ is positive semi-definite (PSD).
- PSD matrices are always invertible and the inverse is also PSD.
- for M PSD holds that for all $r > 0$ that rM is PSD.
- if M is PSD then $A^{\mathrm{T}}MA$ is also PSD.

Orthogonal Matrices

Square matrices that if multiplied with their own transpose equal the identity matrix are called *orthogonal matrix*. In mathematical terms this is expressed as: If $AA^{\mathrm{T}} = A^{\mathrm{T}}A = \mathbb{I}$ A is called an orthogonal matrix. This is equivalent to:

$$A^{\mathrm{T}} = A^{-1}$$

Orthogonal matrix have interesting properties:

- an orthogonal matrix is always invertible
- the determinant of an orthogonal matrix is always ± 1

Examples

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{bmatrix} \qquad \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix} \qquad \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

Please check if the following matrices are orthogonal

$$\begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \tag{33}$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{34}$$

$$\frac{1}{3} \begin{bmatrix} 2 & -2 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & -2 \end{bmatrix} \tag{35}$$

Upper/Lower Triangular Matrices

If all entries of a square matrix above the main diagonal are zero this matrix is called a *lower triangular* matrix. Similarly if all entries of a square matrix below the main diagonal are zero this matrix is called a *upper triangular* matrix.

Examples

$$\begin{bmatrix} 1 & 99 & 5 \\ 0 & 3 & 6 \\ 0 & 0 & 1 \end{bmatrix} \text{(upper triangular matrix)} \tag{36}$$

$$\begin{bmatrix} a & 0 & 0 \\ b & c & 0 \\ d & e & f \end{bmatrix} \text{(lower triangular matrix)} \tag{37}$$

Please calculate the determinant of the matrices above. What do you notice?

Upper and lower triangular matrices play an important role when solving linear equation systems.

Example (backward substitution): solve the following equation system for x_1, x_2, and x_3

$$\begin{bmatrix} 1 & 99 & 5 \\ 0 & 3 & 6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 120 \\ 13 \\ 2 \end{bmatrix}$$

$$x_1 = \dots \qquad x_2 = \dots \qquad x_3 = \dots \tag{38}$$

Important properties:

- the transpose of an upper triangular matrix is a lower triangular matrix and vice versa.
- the determinant of a triangular matrix equals the product of the diagonal entries.

Diagonal Matrices

Matrices in which the off diagonal entries are zero are called *diagonal matrix*. That is any matrix entry $d_{i,j}$ with $i \neq j$ is 0.

Example

$$\begin{bmatrix} 1 & \dots & \dots \\ \dots & 2 & \dots \\ \dots & \dots & 3 \end{bmatrix} \tag{39}$$

This definition also applies for non-square matrices. To be a little bit more specific: only matrix entries with twice the same index (d_i, i) may be non-zero.

Examples

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \end{bmatrix}$$

In `MATLAB` a vector can be turned into a diagonal matrix with the `diag()` command. If `diag()` is called with a matrix the diagonal of the matrix is returned.

Important properties:

- The sum of diagonal matrices is again diagonal
- Multiplication of diagonal matrices equals multiplication of diagonal entries
- inverse of a diagonal square matrix is defined if all diagonal entries are non zero. The inverse is then given by a diagonal matrix with inverse of the diagonal entries.

Please do the following calculations (without `MATLAB`):

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{bmatrix} + \begin{bmatrix} 4 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 5 \end{bmatrix} = \tag{40}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 5 \end{bmatrix} = \tag{41}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{bmatrix}^{-1} = \tag{42}$$

2.3 Special Function Classes

In linear algebra functions are not defined for scalars but for vectors or matrices. This works similarly but might be a bit unintuitive at first.

Linear and Affine Functions

Any function that can be written as $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$ is called a *linear function*. If a linear function is extended by a constant term it becomes an *affine function* and has the form

$$\mathbf{f}(\mathbf{x}) = A\mathbf{x} + b$$

Please reformulate the following functions in the form $\mathbf{f}(\mathbf{x}) = A\mathbf{x} + b$, where $\mathbf{f} = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix}$

$$f_1(\mathbf{x}) = 5x_1 + 7x_2 + 9 \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad A = \qquad b = \tag{43}$$

$$\begin{aligned} g_1(\mathbf{x}) &= f_1(\mathbf{x}) \\ g_2(\mathbf{x}) &= 24x_1 + 23x_3 - 42 \end{aligned} \qquad \mathbf{x} = \qquad A = \qquad b = \tag{44}$$

$$\begin{aligned} h_1(\mathbf{x}) &= f_1(\mathbf{x}) \\ h_2(\mathbf{x}) &= x_2 + \frac{1}{2} \\ h_3(\mathbf{x}) &= 25x_1 - 49x_2 + 81 \end{aligned} \qquad \mathbf{x} = \qquad A = \qquad b = \tag{45}$$

Quadratic Functions

Quadratic functions have a a slightly different structure than their scalar complements.

$$\mathbf{f}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} A\mathbf{x} + B\mathbf{x} + c$$

Please reformulate the following functions in the form $\mathbf{f}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} A\mathbf{x} + B\mathbf{x} + c$

$$f_1(\mathbf{x}) = 7x_1^2 + 4x_1x_2 + 2x_2^2 \qquad \mathbf{x} = \qquad A = \qquad B = \qquad c = \tag{46}$$

$$g_1(\mathbf{x}) = f_1(\mathbf{x}) + 5x_1 + 7x_2 + 9 \qquad \mathbf{x} = \qquad A = \qquad B = \qquad c = \tag{47}$$

The exercises on this sheet give no points that would count towards the exercise points.