MATH 3200: OUTLINE OF CHAPTER 1
MATRICES AND MATRIX OPERATIONS

WINFRIED JUST, OHIO UNIVERSITY

This chapter covers the basic terminology for talking about matrices and vectors, and the definitions of sums, differences, and products of matrices. It also introduces several special kinds of matrices, such as symmetric, zero, identity, triangular, and diagonal matrices. Applications of matrices as spreadsheets, adjacency matrices of graphs, and transition probability matrices of Markov chains are introduced. Our protagonists discuss how to prove some properties of matrix operations.

1. Concepts and Facts

1.1. Matrices (L1, L2, L3).

- A *matrix* is a rectangular array of *elements* arranged in horizontal *rows* and vertical *columns* (L1).

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = [a_{ij}]_{m\times n}$$

  - The elements aka *entries* of a matrix are usually numbers, but we also allow them to be other objects. It will usually be implied by the context whether or not the elements of a matrix are assumed to be numbers.
  - A *diagonal element* of a matrix is an element a_{ii} at the intersection of the row and the column with the same number i. These elements a_{ii} form the *(main) diagonal* of $\mathbf{A}$.
  - The *order* aka *size* aka *shape* of a matrix $\mathbf{A}$ is an expression of the form $m \times n$, where m is the number of rows of $\mathbf{A}$ and n is the number of columns of $\mathbf{A}$.
  - The numbers m and n are sometimes called the *dimensions* of the matrix.
  - For example, the matrix $\mathbf{A} = \begin{bmatrix} 1 & 0 & -1 \\ 6 & \pi & 4 \end{bmatrix}$ is a 2×3 matrix; it has $m = 2$ rows and $n = 3$ columns.
    The elements of this matrix are $a_{11} = 1, a_{12} = 0, a_{13} = -1, a_{21} = 6, a_{22} = \pi, a_{23} = 4$.
- A matrix $\mathbf{A}$ of order $m \times n$ is a *square matrix* if $m = n$.
- Two matrices $[a_{ij}]_{m\times n}$ and $[b_{ij}]_{m'\times n'}$ *are equal* if they have the same order (that is, $m = m'$ and $n = n'$), and $a_{ij} = b_{ij}$ for all $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$.
  - For example, the matrices $\mathbf{A} = \begin{bmatrix} 1 & 0 & -1 \\ 6 & \pi & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 1 & 6 \\ 0 & \pi \\ -1 & 4 \end{bmatrix}$

    are not equal, as they do not have the same order.

1.2. **Vectors (L2).**

- A $1 \times n$ matrix is called a *row vector* and and $m \times 1$ matrix is also called a *column vector*.
  - For example, the matrix $\vec{\mathbf{x}} = \begin{bmatrix} 1 & 0 & -1 & 0.23 \end{bmatrix}$ is a row vector, and $\vec{\mathbf{y}} = \begin{bmatrix} 6 \\ \pi \\ -1 \end{bmatrix}$ is a column vector.
  - Vectors will be denoted by symbols like: $\vec{\mathbf{x}}, \vec{\mathbf{y}}, \vec{\mathbf{z}}, \vec{\mathbf{u}}, \vec{\mathbf{v}}, \vec{\mathbf{w}}$.
  - The number of entries in a vector is called its *dimension* or its *length*.
  - Linear arrays of data can be represented as either row vectors or column vectors; it depends on the particular calculation that we want to perform which representation to use.

1.3. **Simple matrix operations (L4).**

- Addition and subtraction of matrices:
  - Let $\mathbf{A} = [a_{ij}]_{m\times n}$ and $\mathbf{B} = [b_{ij}]_{m\times n}$ be two matrices of the same order. We define:
    * $\mathbf{A} + \mathbf{B} = [a_{ij} + b_{ij}]_{m\times n}$
    * $\mathbf{A} - \mathbf{B} = [a_{ij} - b_{ij}]_{m\times n}$
  - When $\mathbf{A}$ and $\mathbf{B}$ have *different orders,* then $\mathbf{A} + \mathbf{B}$ and $\mathbf{A} - \mathbf{B}$ are *undefined.*
  - The $m \times n$ matrix that has all entries equal to zero will be denoted by $\mathbf{O}_{m\times n}$ or simply $\mathbf{O}$ if the dimensions are implied by the context. It is called the *zero matrix (of order $m \times n$).*
  - Properties of matrix addition:
    * $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$ (commutativity),
    * $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$ (associativity),
    * $\mathbf{A} + \mathbf{O} = \mathbf{O} + \mathbf{A} = \mathbf{A}$.
- The transpose of a matrix:
  - The *transpose* $\mathbf{A}^T$ of an $m \times n$ matrix $\mathbf{A} = [a_{ij}]_{m\times n}$ is the $n \times m$ matrix $\mathbf{B} = [b_{ji}]_{n\times m}$ such that $b_{ji} = a_{ij}$ for all $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.
  - Every matrix $\mathbf{A}$ has a transpose $\mathbf{A}^T$.
  - For every matrix $\mathbf{A}$ we have $(\mathbf{A}^T)^T = \mathbf{A}$.
  - The transpose of an $m \times 1$ column vector is a $1 \times m$ row vector.
  - The transpose of a $1 \times n$ row vector is an $n \times 1$ column vector.
- Multiplication of a matrix by a scalar:
  - In linear algebra, the word "scalar" simply means "number".
  - For any matrix $\mathbf{A} = [a_{ij}]_{m\times n}$ and scalar λ we define
    * $\lambda\mathbf{A} = [\lambda a_{ij}]_{m\times n} = [a_{ij}\lambda]_{m\times n} = \mathbf{A}\lambda$.
  - For any matrices $\mathbf{A}$, $\mathbf{B}$ of the same order and scalars λ, κ we have:
    * $\lambda\mathbf{A}^T = (\lambda\mathbf{A})^T$.
    * $\lambda\mathbf{A} + \lambda\mathbf{B} = \lambda(\mathbf{A} + \mathbf{B})$ (distributivity).
    * $\mathbf{A} + (-1)\mathbf{B} = \mathbf{A} - \mathbf{B}$.
    * $\kappa(\lambda\mathbf{A}) = (\kappa\lambda)\mathbf{A}$.

1.4. **Matrix multiplication (L5).**

- The *product* $\mathbf{AB}$ of two matrices $\mathbf{A} = [a_{ij}]_{k\times n}, \mathbf{B} = [b_{ij}]_{m\times p}$ is defined if, and only if, $n = m$, that is, the number of columns of $\mathbf{A}$ is equal to the number of rows of $\mathbf{B}$.
  If $\mathbf{AB}$ is defined, it has order $k \times p$.
- If $\mathbf{AB} = [c_{ij}]_{k\times p}$ is defined, then $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{\ell=1}^{n} a_{i\ell}b_{\ell j}$.
- The matrix product $\vec{\mathbf{x}}\vec{\mathbf{y}}$ of a row vector $\vec{\mathbf{x}}$ and a column vector $\vec{\mathbf{y}}$ of the same length is a 1×1 matrix whose single element is called *the inner product* or *dot product* of these vectors.
- Matrix products obey the following laws:
  - $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$ (associativity).
  - $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB} + \mathbf{AC}$ (left distributivity).
  - $(\mathbf{B}+\mathbf{C})\mathbf{A} = \mathbf{BA} + \mathbf{CA}$ (right distributivity).
  - $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$. Note that the order of multiplication changes here.
- *Commutativity may fail:* Even when $\mathbf{AB}$ and $\mathbf{BA}$ both exist and have the same order, we *may* have $\mathbf{AB} \neq \mathbf{BA}$.
- The sums of the rows of $\mathbf{A}$ are given by the matrix product $\mathbf{A}\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}$.
- The sums of the columns of $\mathbf{A}$ are given by the matrix product $[1\dots1]\mathbf{A}$.

1.5. **Submatrices (L6).**

- A *submatrix* $\mathbf{B}$ of a given matrix $\mathbf{A}$ is any matrix that can be obtained by removing some rows and/or columns from $\mathbf{A}$.
- Consecutive rows and columns of $\mathbf{A}$ that remain in $\mathbf{B}$ may or may not be adjacent in $\mathbf{A}$. However, when forming a submatrix of $\mathbf{A}$, the order of rows and of columns must be preserved.

1.6. **Matrices with special properties (L3, L7).**

- A *square matrix* is a matrix of order $n \times n$ for some positive integer n.
- A square matrix $\mathbf{A}$ is *symmetric* when $a_{ij} = a_{ji}$ for all i, j.
  This is the case if, and only if, $\mathbf{A} = \mathbf{A}^T$.
  For every square matrix $\mathbf{A}$ the sum $\mathbf{A} + \mathbf{A}^T$ is symmetric.
- $\mathbf{O}_{m\times n}$ denotes the *zero matrix* of order $m \times n$. All of its elements are equal to 0.
  When the order is implied by the context, we simply write $\mathbf{O}$.
  Zero matrices that are vectors are called *zero vectors* and denoted by $\vec{\mathbf{0}}$.
- The *identity* matrix $\mathbf{I}_n$ is the $n \times n$ matrix whose diagonal elements are all ones and whose off-diagonal elements are all zeros.
  When the order is implied by the context, we write $\mathbf{I}$ instead of $\mathbf{I}_n$.
  $\mathbf{IA} = \mathbf{A}$ and $\mathbf{AI} = \mathbf{A}$ whenever these products are defined.
- A square matrix $\mathbf{A} = [a_{ij}]_{n\times n}$ is:
  - *diagonal* if $a_{ij} = 0$ whenever $i \neq j$.
  - *upper-triangular* if $a_{ij} = 0$ whenever $i > j$,
  - *lower-triangular* if $a_{ij} = 0$ whenever $i < j$.
  - *triangular* if it is either upper- or lower-triangular.
- Identity matrices are examples of diagonal matrices.

- Upper triangular matrices are exactly those matrices that have only zeros below their main diagonals; lower triangular matrices are exactly those matrices that have only zeros above their main diagonals.
- Products of two diagonal matrices of the same order are again diagonal matrices. They can be computed by multiplying the corresponding diagonal elements.
- The product of two upper-triangular matrices of the same order is again an upper-triangular matrix; the product of two lower-triangular matrices of the same order is again a lower-triangular matrix.

2. Skills

Afer working through this chapter, students will:

- Be able to interpret and use *summation notation,* as defined in M2: $a_1 + a_2 + \cdots + a_n = \sum_{j=1}^{n} a_j$.
- Be able to perform the following matrix operations by hand and using MatLab:
  - Addition, subtraction, transpose, and multiplication by a scalar (L4, M5).
  - Matrix multiplication (L5, M7, M8).
  - Computing the inner and the outer product of two vectors (L5, M7).
  - Finding powers $\mathbf{A}^n$ for integers $n \geq 0$ of a square matrix $\mathbf{A}$ (M9).
  - Creating submatrices of a given matrix by removing selected columns and/or rows (L6, M9).

3. Applications

3.1. Spreadsheets (L1, C2).

- Every spreadsheet can be treated as a matrix $\mathbf{A}$.
- We looked at the particular example of the instructor's gradebook.

3.2. Adjacency matrices of undirected and directed graphs (L3, C3).

- An *graph* G is a mathematical construct that shows the connections between objects called *nodes* or *vertices.*
- An *undirected graph* can be constructed as follows:
  - Draw a little circle for each node i of the graph.
  - Then connect nodes i and j with one or more line segments that represent the relevant connections between i and j.
  - These line segments represent the *edges* of G.
- A *directed graph*, or *digraph* for short, can be constructed as follows:
  - Draw a little circle for each node i of the graph.
  - Then connect nodes i and j with one or more arrows that represent the relevant directed connections between i and j.
  - These arrows represent the *directed edges* or *arcs* of G. When an arrow points from i to j, then i is called the *source* of this arrow or arc, and j is called its *target.*
- A graph is *loop-free* if there are no edges or arcs from any node i to itself.
- A graph is *simple* if between any give two nodes there is at most one edge or arc.
- We consider the following examples of graphs:

- Graphs of friendships. Here the nodes represent people in a group, and two nodes i and j are connected by an edge if, and only if, i is a friend of j. We assume that friendship is always reciprocated, so that when j is a friend of i, then also i is a friend of j. This makes graphs of friendships undirected. We also assume that these graphs are loop-free and simple.
- Graphs of transmission of an infectious disease. Here there is an arc from node i to node j if, and only if, j caught the infection from i. These graphs are directed, loop-free, and simple.
- Graphs of the connectivity of the internet. Here nodes represent web pages, and there is an arc from i to j for each link to page j that is embedded in page i. These graphs are directed and loop-free, but not simple, as there may be multiple links at page i to the same page j.

* A graph can be represented by its *adjacency matrix.* If G has n nodes, then the adjacency matrix $\mathbf{A}$ of G has order $n \times n$, and the element a_{ij} in row i and column j of $\mathbf{A}$ shows the number of connections from node i to node j.
* The adjacency matrix $\mathbf{A}$ of a graph G has the following properties:
  - $\mathbf{A}$ is always a square matrix.
  - When G is undirected, then $\mathbf{A}$ is symmetric.
  - G is simple if, and only if, each element a_{ij} of A is either 0 or 1.
  - G is loop-free if, and only if, each diagonal element a_{ii} of A is equal to 0.

3.3. **Markov chains C7, C8, C9, C10.**

* *Markov chains* are mathematical models for prediciting the probability that certain chance events will occur. These predictions rely on matrix multiplication. We present in detail a very simple example of a Markov chain for predicting the weather (C7, C8, C9), and a second example that features a quirky character named Waldo (C10). Both examples illustrate—in a greatly simplified form—important real-world applications of Markov chains.
* *Probabilities* are covered in C7 and M11.
  - Probabilities are numbers p such that $0 \leq p \leq 1$.
  - A probability of $p = 0$ for an event or outcome signifies that the event will *never* occur.
  - A probability of $p = 1$ for an event signifies that the event *will occur with certainty.*
  - A probability of $p = 0.5$ for an event signifies that there is a fifty-fifty chance that the event will occur.
  - The probability of an event can be thought of as the proportion of times this event would occur if we conduct a large number of observations of very similar situations. For example, if we flip a fair coin, then the event that it will come up heads will be $p = 0.5$. Similarly, if we roll a fair die, then the event that it will come up "3" will be $p = 1/6$.
* Construction of Markov chains are covered in C8 and C10:
  - In Markov chains, time is assumed to proceed in discrete *time steps,* and the real-world situations of interest are categorized into nonoverlapping *states.* When constructing a Markov chain model of a real-world situation, we need to start by specifying the meaning of each state and the meaning of one time step.
  - In Markov chains, we are interested in the probabilities p_{ij} that when the system is in state i at time t, it will be in state j at time $t + 1$.

- These probabilities p_{ij} are called the *transition probabilities.* In Markov chains, they do not depend on t, and also do not depend on how prior history, that is, how the system reached state i.
- The *transition probability matrix* $\mathbf{P} = [p_{ij}]$ has order $n \times n$, where n is the number of states, and is a *stochastic matrix,* which means that each of its rows sums up to 1. M12 illustrates how to construct transition probability matrices $\mathbf{P}$ of Markov chains.
- In Markov chains, we are interested in the probabilities p_{ij} that when the system is in state i at time t, it will be in state j at time $t+1$.

- Making predictions using a Markov chain with n states and transition probability matrix $\mathbf{P}$ is covered in C9, M13, M14:
  - The *probability distribution* at time t is a row vector $\vec{\mathbf{x}}(t) = [x_1(t), \ldots, x_n(t)]$ that gives the probabilities $x_i(t)$ that the system is in state i at time t.
  - The probability distribution for the next step is given by $\vec{\mathbf{x}}(t+1) = \vec{\mathbf{x}}(t)\mathbf{P}$.
  - More generally, for any $k \geq 1$ and state $\vec{\mathbf{x}}(t)$, the distribution $\vec{\mathbf{x}}(t+k)$ after k time steps is given by $\vec{\mathbf{x}}(t+k) = \vec{\mathbf{x}}(t)\mathbf{P}^k$, where $\mathbf{P}^k$ is the *k-step transition probability matrix.* For long-range predictions, that is, for relatively large values of k, one would always use software like MatLab to compute $\mathbf{P}^k$.

4. Proofs (C2, C4, C5, C6)

In this course you will learn how to evaluate correctness of mathematical proofs and construct some simple ones yourself. Conversations 2, 4, 5, and 6 are essentially demos of how one should go about these tasks. Issues that are covered in these conversations include:

- Proofs are tools for verifying that mathematical methods, such as recipes for performing certain calculations, always give correct results.
- Writing proofs is a lot easier than most people believe. The kind of proofs that we will do in this course are essentially calculations with symbols.
- First make sure you fully understand the meaning of the theorem. You may want to look up the relevant notions in the lectures.
- A single numerical example does not constitute a proof. However, Numerical examples can guide us in finding proofs. Sometimes it suffices to redo the same calculations as in a numerical example with symbols.
- In order to prove that a given calculation always works as claimed, we need to set up a notation that covers all possibilities. Using a convenient notation may simplify your work.
- When translating numerical calculations into symbolic ones, make sure you use symbols for every quantity the theorem talks about.
- At the beginning of a proof, you need to express the assumptions of a theorem in terms of your symbolic notation.
- When some of the assumptions imply that certain quantities are equal, you may be able to use fewer symbols than you would need otherwise.
- You also need to translate the conclusion of the theorem into symbols. Here "conclusion" refers the property that the theorem asserts and that you want to prove.
- The best strategy is to defer any symbolic calculations until after you have translated the assumptions and the conclusion.

- Your first attempt may only be partially successful. This is normal. Take a close look at your first version of a proof and fix it if need be. If your calculations don't work out as expected, double-check whether you have correctly translated the assumptions of the theorem.
- Make sure you have used every assumption of the theorem somewhere in your proof.