MATH 310, REVIEW SHEET

These notes are a summary of the key topics in the book (and follow the book pretty closely). You should be familiar with everything on here, but it's not comprehensive, so please be sure to look at the book and the lecture notes as well. Let me know if you notice any particularly egregious omissions and I'll add more.

These notes also don't contain any examples at all. If you run into something unfamiliar, look at the notes or the book for an example! The goal of this sheet is just to remind you of what the major topics are.

I've also indicated some of the important "problem types" we've encountered so far and that you should definitely be able to do. There will inevitably be a problem on the exam not of one of the types listed.

1. Linear equations

1.1. Systems of linear equations.

1.2. Row reduction and echelon forms.

A linear equation is one of the form $a_1x_1 + \cdots a_nx_n = b$, where the a_i's and b's are a bunch of numbers. A system of linear equations is a bunch of equations of this form, involving the same variables x_i. The goal is to find all the possible x_i that make the equations hold. In general, the number of solutions will be 0, 1, or infinite.

Given a linear system, the first step in solving it is usually to form the *augmented matrix* of the system. We then perform "elementary row operations" on the matrix to find the general solution. These row operations just correspond to simple manipulations of the equations.

There is basically a two-part process for finding all solutions to a linear system : first, use row operations to put the matrix in a standard form, "reduced row echelon form"; second, read off the general solution from the reduced row echelon form of the matrix.

There are three legal row operations:

(1) Replace one row by the sum of itself and a multiple of another row;
(2) Interchange two rows;
(3) Multiply all entries in a row by a nonzero constant.

A matrix is in echelon form if:

(1) All nonzero rows are above any rows of all zeros.
(2) Each leading entry of a row is in a column to the right of the leading entry of the row above it.
(3) All entries in a column below a leading entry are zeros.

If the matrix additional satisfies the following two requirements, it is in reduced echelon form:

(1) The leading entry in each nonzero row is 1;
(2) Each leading 1 is the only nonzero entry in its column.

Revision (None), jdl, (None)

These conditions may sound totally arbitrary; the point is that once a matrix is in reduced echelon form, there is no additional simplification to be done using algebraic operations. It's time to read off the solutions.

There is a five-step process for putting a matrix in reduced echelon form:

(1) Begin with the leftmost nonzero column.
(2) Select a nonzero entry in this column as a pivot. Usually you can just use the top entry; if not, swap to rows to move something nonzero to the top.
(3) Create zeros below the pivot by adding multiples of the top row to subsequent rows.
(4) Cover up the top nonzero row, and repeat the process on what remains.

Please, please, please practice some examples of row reduction. This technique is a little painful, but it is the basis for almost everything else in this chapter.

Once you reach reduced echelon form (or in fact echelon form), you should identify the pivots of your matrix: these are the entries in the matrix where which are the leading 1s in the reduced echelon form.

Once you've reached echelon form, you want to write down the general solution to the system of equations. The first thing to check is whether the system is consistent, i.e. whether there are any solutions at all. This is easy: if there's a row $\left[\begin{array}{ccc|c} 0 & 0 & 0 & b \end{array}\right]$, where b is not 0, then there are no solutions (that's because a row of this kind corresponds to the linear equation $0 = b$).

If there's no row like this, then there are solutions: the system is consistent. Every variable that's not a basic variable is a "free variable", and can take any value. The basic variables are then determined by the values of the free variables, using the equations determined by the rows of the row reduced echelon form. The number of solutions is 1 if there are no free variables, and infinite if there are free variables (because you can plug in any number you want for the free variables.)

Here's another tidbit that can save time on exam: if you want to know whether there are free variables or not, it's enough to put the matrix into echelon form and see what the pivots are: you don't have to go the full way into reduced echelon form. If a question asks whether such-and-such has infinitely many solutions or only one (and doesn't ask you to actually find the solution), this is all you need to know. However, if you actually need to find the general solution, you're probably going to want to put it into rref anyway.

- Use row reduction to matrix a matrix in reduced echelon form
- Write down the general solution to a system of equations
- Determine if a system of equations is consistent
- Variant: for what values of a parameter h is a system consistent

1.3. **Vector equations.** This section introduces vectors and the rules for working with them. You should know what a vector is, how to add vectors, and how to draw vectors in $\mathbb{R}^2$.

The most important topic here is linear combinations. If $\mathbf{v}_1, \ldots, \mathbf{v}_p$ are a bunch of vectors, then $\mathbf{y} = c_1\mathbf{v}_1 + \cdots + c_p\mathbf{v}_p$ is called a *linear combination* of the $\mathbf{v}_i$.

The set of all linear combinations of a bunch of vectors is called the *span*. It's important to think of the span geometrically: if you have two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ in $\mathbb{R}^3$, the span is the set of all the vectors of the form $c_1\mathbf{v}_1 + c_2\mathbf{v}_2$. This span is a 2D plane in 3D space.

We introduced the notion of a *vector equation*: if $\mathbf{a}_1, \ldots, \mathbf{a}_n$ and $\mathbf{b}$ are a bunch of vectors, we can look at the vector equation $x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$: we are asking whether there exist

x_i's such that $\mathbf{b}$ is a linear combination of the $\mathbf{a}_i$'s, with weights x_i. In other words, the vector equation asks whether $\mathbf{b}$ is a linear combination of the $\mathbf{a}_i$.

The vector equation is actually equivalent to a linear system with augmented matrix $\left[\begin{array}{ccc|c} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & b \end{array}\right]$. So to solve it, you use exactly the same methods as before: the difference is that we're now thinking of the equation in terms of linear combinations.

- Do basic operations on vectors
- Convert a system of linear equations into a vector equation, and vice versa
- Find the general solution of a system of vector equations
- Determine if a vector is a linear combination of other given vectors

1.4. **Matrix equations.** Now we introduce matrices. If A is an $m \times n$ matrix and $\mathbf{x}$ is a size-n vector, we can form the product $A\mathbf{x}$, which is a size-m vector. To get the vector you take a linear combination of the columns of A, with weights the entries in the vector $\mathbf{x}$: if the columns of A are called $\mathbf{a}_1, \ldots, \mathbf{a}_n$, the product is $x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n$. There is also another way to compute the product, which is a little faster when doing it by hand, but amounts to basically the same thing. This is the "row–vector rule", which you can read about on page 38 of the book.

Given a matrix A and a vector $\mathbf{b}$, one can ask whether $A\mathbf{x} = \mathbf{b}$ has any solutions. Since $A\mathbf{x}$ is a combination of the columns of A, this really amounts to asking whether $\mathbf{b}$ can be written as as a combination of the columns of A: solving for the vector $\mathbf{x}$ just means that we are trying to find weights to make this work. So $A\mathbf{x} = \mathbf{b}$ has a solution if and only if $\mathbf{b}$ is a linear combination of the columns of A.

The important thing here is that we now how three different ways to write a system of linear equations: as a bunch of equations involving the individual variables, as a vector equation, and as a matrix equation. They all amount to the same thing, and you solve them all the same way: using row reduction. You can see this worked out in an example in the lecture notes from 9/2.

Another useful fact is "Theorem 4" on page 37 of the book: given a matrix A, you can ask whether $A\mathbf{x} = \mathbf{b}$ has a solution for *every* $\mathbf{b}$. For some matrices A this is the case, while for other matrices A there are some $\mathbf{b}$'s for which there are no solutions. The answer is that $A\mathbf{x} = \mathbf{b}$ has a solution for every $\mathbf{b}$ if the columns of A span $\mathbb{R}^m$, or equivalently, if A has a pivot position in every row.

- Multiply matrix $\times$ vector
- Convert a system of linear equations or vector equation into a matrix equation
- Find the general solution of a matrix equation
- Check whether $A\mathbf{x} = \mathbf{b}$ has a solution for every $\mathbf{b}$

1.5. **Solution sets of linear systems.** The idea of this section is to think a bit about what the solution set of a system of linear equations is really like: although there can be infinitely many solutions, they can easily be parametrized.

A system is called *homogeneous* if it looks like $A\mathbf{x} = \mathbf{0}$. This corresponds to a system of linear equations where we are asking that various combinations of the variables are all 0. A homogeneous linear system is always consistent, since $\mathbf{x} = \mathbf{0}$ is a solution. It has a solution other than $\mathbf{x} = \mathbf{0}$ if and only if there is at least one free variable when you do row reduction to echelon form.

In general, given a linear system $A\mathbf{x} = \mathbf{0}$, you can write the solution in *parametric vector form*, which looks like $\mathbf{x} = s\mathbf{u} + t\mathbf{v}$, where $\mathbf{u}$ and $\mathbf{v}$ are two vectors. Here s and t are parameters; in general the number of parameters you need is equal to the number of free variables. Make sure you know how to figure out the parametric vector form of the general solution.

This only handles homogeneous equations, but it can be adapted to inhomogeneous ones too. If you're looking at $A\mathbf{x} = \mathbf{b}$, you can find the general solution by finding the general solution to $A\mathbf{x} = \mathbf{0}$, in parametric vector form, and finding a single solution to $A\mathbf{x} = \mathbf{b}$ (e.g. by plugging in 0 for all the free variables, or just guessing). Then the general solution is the sum of these two things.

- Write solutions to a system in parametric vector form
- Describe geometrically the solution set of a system, using parametric vector form
- Find the general solution to an homogeneous linear system, given the general solution of the corresponding homogeneous linear system

1.6. **Applications of linear systems.** We only talked about two of the types of problems in this section, and only those are fair game: balancing a chemical reaction, and dealing with networks.

Balancing a chemical reaction is pretty straightforward: you are given a reaction with reactants on one side, and products on the other. The challenge is to find how many molecules of each of the reactants and products are needed to balance the equation. To do this, you translate it to a linear system: give a variable name x_i to the coefficient in front of each of the reactants and products. For each of the elements that appears in the reaction, you get a linear equation: the number of atoms of the element appearing on the left is equal to the number of atoms on the right. Then you solve the linear system.

The other thing we talked about was network flow. Here's the set up: you're given a network, which consists of a bunch of nodes, and a bunch of branches connecting the nodes. You can think of it as a network of streets, where the nodes are intersections, and the branches are the roads between the intersections. Given the flow along some of the branches, the goal is to figure out what the flow must be along the others, where it is not known. The strategy is similar: for each of the branches where you don't know the flow, give a variable name to it. Now you get a bunch of linear equations: at each node, the total incoming flow must equal the total outgoing traffic flow. Then you get one extra equation: the total flow into the network must equal the total flow out of the network. This is a bunch of linear equations: now set up the augmented matrix and solve by row reduction.

- Balance a chemical equation by setting up a linear system
- Find the flows in a network by setting up a linear system

1.7. **Linear independence.** Next up is linear independence. Say you have a bunch of vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$. They're *linearly independent* if the only way to get $x_1\mathbf{v}_1 + \cdots + x_n\mathbf{v}_n = 0$ is if all the x_i are 0.

Checking is easy: we want to know if the vector equation $x_1\mathbf{v}_1 + \cdots + x_n\mathbf{v}_n = 0$ has any solutions other than 0. This is something that you can check by row reduction: it's a homogenous linear system, so set up an augmented matrix with a column of 0's, go through row reduction, and check.

For two vectors to be linearly independent means that they are not collinear (this you can check just by looking at them: is one a multiple of the other?); for three vectors to be linearly independent means that they are not all in the same plane.

- Check whether a set of vectors is linearly independent

- Find values of a parameter h for which a set of vectors is linearly independent

1.8. **Introduction to linear transformations.**

1.9. **The matrix of a linear transformation.** If you have a $m \times n$ matrix A, you can think of it as defining a transformation $T : \mathbb{R}^n \to \mathbb{R}^m$: it's a rule that takes as input a size m vector, and gives as output a size n vector.

You can often think about linear transformations graphically: draw a few vectors, and then draw the new vectors you get after applying T. This can help you work figure out a simple geometric description of what a linear transformation does.

Officially, a transformation T is linear if $T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$ and $T(c\mathbf{v}) = cT(\mathbf{v})$. This roughly means that it sends straight lines to other straight lines, and sends $\mathbf{0}$ to $\mathbf{0}$. Any transformation that's given by a matrix is linear, and it turns out that the opposite is true as well: any transformation that's linear is determined by some matrix.

An important task is to be able to figure out the matrix that gives a linear transformation. For example, the transformation might be described to you geometrically or by some equations, and your task is to write down the matrix that does the transformation in question. There is a recipe for this. First, start with the vectors $\mathbf{e}_1 = (1, 0)$ and $\mathbf{e}_2 = (0, 1)$ (you'll need more vectors if your transformation is between vectors with more entries). Figure out what $T(\mathbf{e}_1)$ and $T(\mathbf{e}_2)$ are: to do this, use the description of T that's given to you to work out where T sends $\mathbf{e}_1$ and $\mathbf{e}_2$. This could involve doing some geometry, or plugging in $(1, 0)$ to some equation.

Once you've done that, form a matrix A whose columns are $T(\mathbf{e}_1)$, $T(\mathbf{e}_2)$, etc. That's the matrix for your transformation.

There are also two new definitions introduced in this chapter: if $T : \mathbb{R}^n \to \mathbb{R}^m$ is a linear transformation, given by a matrix A, then:

(1) T is onto if each $\mathbf{b}$ in $\mathbb{R}^m$ is the image of at least one $\mathbf{x}$. To check this, put the matrix A into rref, and check if there is a pivot in every row (if yes, then T is onto).
(2) T is one-to-one if each $\mathbf{b}$ in $\mathbb{R}^m$ is the image of at most one $\mathbf{x}$ in $\mathbb{R}^n$. This is equivalent to T having linearly independent columns: you can check that by our strategy for checking linear independence given above.

- Find the image of a vector under a linear transformation
- Check whether a vector is in the range of a linear transformation
- Describe a linear transformation geometrically
- Find the matrix for a linear transformation, given a description in some other form
- Check whether a transformation is one-to-one and/or onto

1.10. **More linear models.** Electrical networks: given a diagram of a simple circuit containing resistors and batteries, we can find the current through each branch of the circuit. To do this, think of the circuit as a bunch of loops, with some current I_j flowing through it. Our goal is to solve for the I_j's. From each loop, you get a linear equation, using Kirchhoff's voltage law: if you add up the voltage drops RI for each resistor, it's equal to the sum of the batteries in the loop. This gives a system of linear equations that you can solve by row reduction. (This explanation might not make so much sense without an example; take a look at Example 2 on page 83.)

2. Matrix algebra

2.1. **Matrix operations.** You should know how to perform the following operations on a matrix: sum, scalar multiple, multiplication, transpose, powers. This is mostly a matter of practice: make sure you do some of each.

The interesting one is multiplying matrices; the rest work about like you'd expect. Here's the official definition: given two matrices A and B, first check if the number of columns of A is equal to the number of rows of B; if not, then AB isn't defined. If yes, then AB is given by $A\begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}$, where these are the columns of B.

There's another way to multiply matrices that's a little faster when you're doing it by hand: this is the "row–column method" which you can read about in the notes or the book.

Important to remember is that multiplication of matrices doesn't work as nicely as multiplication of numbers. For example, AB and BA aren't always the same (even when both are defined). It's also not true that if $AB = AC$, then $B = C$: you can't "divide both sides of the equation by A".

- Perform basic matrix operations
- Understand the basic properties of these operations

2.2. **The inverse of a matrix.** If A is an $n \times n$ matrix (note: it must be square!), we say A is invertible if there is another $n \times n$ matrix C such that $AC = CA = I_n$, where I_n is the identity matrix. In this case we write $C = A^{-1}$.

One use of this is that if you're trying to solve $A\mathbf{x} = \mathbf{b}$, and A is invertible, there is a unique solution, and it's given by $\mathbf{x} = A^{-1}\mathbf{b}$. This method is slow to actual solve equations, but it's helpful when you're trying to derive formulas for things.

For a 2×2 matrix, there's a formula for the inverse:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Longrightarrow A^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

(Unless $ad - bc = 0$: then the matrix isn't invertible.) For bigger matrices, the best way to find the inverse is this (assuming Matlab is not available to you...): write down the $n \times 2n$ augmented matrix $\begin{bmatrix} A & I_n \end{bmatrix}$, run row reduction until you reach $\begin{bmatrix} I_n & B \end{bmatrix}$. The matrix B that appears on the right is the inverse.

- Find the inverse of a 2×2 matrix
- Find the inverse of a bigger matrix
- Solve $A\mathbf{x} = \mathbf{b}$ when A is invertible, using matrix inversion

2.3. **Characterizations of invertible matrices.** This is a weird chapter, in that there is only one thing in it. However, it's an important thing: if A is a square matrix, the following statements are either all true or all false.

(1) A is invertible
(2) Row reduction on A ends up with I_n
(3) A has n pivots (which must be on the diagonal)
(4) $A\mathbf{x} = 0$ has only the trivial solution
(5) The columns of A are linearly independent
(6) The transformation $\mathbf{x} \mapsto A\mathbf{x}$ is one-to-one
(7) The transformation $\mathbf{x} \mapsto A\mathbf{x}$ is onto
(8) The equation $A\mathbf{x} = \mathbf{b}$ has a solution for every $\mathbf{b}$

(9) The columns of A span $\mathbb{R}^n$
(10) There is a matrix C with $AC = I_n$
(11) There is a matrix D with $DA = I_n$
(12) A^T is invertible
(13) (For 2×2 only, for now) $\det A = ad - bc \neq 0$

The upshot is that to check if A is invertible, you can check any of these other things using the method of your choice. Note that C and D in (10) and (11) end up being the same: the interesting thing is that if you can find a C with $AC = I_n$, then it's automatically true that $CA = I_n$ too.

- Given a matrix, figure out whether it's invertible

2.4. **Matrix factorizations and LU decomposition.** If A is an $m \times n$ matrix, one can (usually) find an LU decomposition $A = LU$. Here L is an $m \times m$ lower triangular matrix (note: the size is different from that of A, and L is always square). The way we arrange things, L is always going to have 1s on the diagonal. The matrix U is upper triangular, and the same size as A.

Here's how to find it: start with your matrix A, and do row reduction until you reach echelon form (NB: not rref. However, you're only allowed to use certain row operations: you can subtract a multiply of a row from a row below it (or equivalently, add a multiple of a row to a row below it). But you can't multiply a row by a number, and you can't swap rows. (Well, really you can, but you need to use a slightly more complicated form of LU decomposition if you want to, and we didn't cover it.)

The matrix U is just going to be the echelon form that you reached. To get L, start off with an $m \times m$ square matrix. Put 1s along the main diagonal and 0s above it. Now, in position (i, j), put the number of Row j's that you subtracted from Row i in the course of doing row reduction. For example, L_{32} is the number of row 2's you subtracted from row 3.

LU decomposition essentially "remembers" how to do row reduction on a matrix A. This has a number of computational advantages. If you want to solve $A\mathbf{x} = \mathbf{b}$, you can instead solve $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$, and then $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$. It may not sound like it, but if you are going to need to solve $A\mathbf{x} = \mathbf{b}$ for many different $\mathbf{b}$'s for a fixed matrix A, this is going to be faster. The point is that $L\mathbf{x} = \mathbf{y}$ only involves doing row reduction on a triangular matrix, which is fast.

- Find LU decomposition of a given matrix.
- Use this to solve $A\mathbf{x} = \mathbf{b}$ by solving $L\mathbf{y} = \mathbf{b}$ and then $U\mathbf{x} = \mathbf{y}$.

3. Determinants

3.1. **Introduction to determinants.**

3.2. **Properties of determinants.** If A is an $n \times n$ matrix (NB: it has to be square), then $\det A$ (also written $|A|$) is a number. Be sure you know the following useful properties of this operation:

(1) Determinant is unchanged when adding one row/column to another row/column.
(2) Determinant changes sign when swapping two rows/columns.
(3) $\det A = 0$ if and only if A has a nonzero nullspace (i.e. it's not invertible).

We have three main methods for computing determinants. I'm not going to explain how each goes, but make sure you know them all. Here they are, with some suggestions one when each one is useful.

(1) Cofactor expansion. Especially suitable when the matrix has a row or column with only a couple nonzero entries. You can always start off with cofactor expansion and then switch to another method for the smaller determinants that pop out.
(2) Product of pivots. If a matrix doesn't fit any of the above (i.e. it's at least 4×4 and doesn't have many zeroes), this is probably the way to go. Do row reduction, without ever multiplying a row by a number. When you get to echelon form, the determinant is just the product of the pivots, with an extra factor of -1 for every time you had to swap rows.
(3) The specific formulas for 2×2 and 3×3 matrices. Good if you are dealing with a 2×2 and 3×3 matrix; obviously not so helpful otherwise.

Perhaps the following should qualify as a basic methods:

(1) Triangular matrices: just multiply the diagonal entries.
(2) Reduce to simpler matrix by row/column operations. These don't change the determinant, so if you can do something sneaky like reduce to an triangular matrix or a matrix with lots of 0's using just a couple row/column operations (including row swaps), you're in business.

- Find the determinant of a given matrix by whatever method is appropriate.

3.3. **Cramer's rule, volume, and linear transformations.** The determinant of A is equal to the volume of the parallelpiped defined by the columns of A; this shows up as the Jacobian in change of coordinates for 2- or 3-dimensional integrals. A consequence of this is that if you look at the linear transformation $T : \mathbb{R}^3 \to \mathbb{R}^3$ defined by the matrix A, it rescales area/volume by a factor of the determinant.

Cramer's rule is a way to solve linear systems $A\mathbf{x} = \mathbf{b}$, where A is square. Here's what you do. Let $A_i(\mathbf{b})$ be the matrix obtained by replacing the i^{th} column of A with the vector $\mathbf{b}$. The $x_i = \det A_i(\mathbf{b}) / \det A$.

As a consequence, we get a formula for A^{-1}: we have $A^{-1} = C^T / \det A$, where C is the matrix of cofactors of A. There are examples of this in the lecture notes, book, and homework solutions, but there hasn't been one on a quiz. You might want to take a look.

These are not efficient ways to solve $A\mathbf{x} = \mathbf{b}$ in general, since you have to take a lot of determinants, which isn't fun. Nevertheless it can sometimes be very helpful to have an explicit formula like this; for example, when your matrix A involves a parameter s, and you want to understand how the solution depends on the parameter s, Cramer's rule gives you a good way to go about it.

- Use Cramer's rule to solve $A\mathbf{x} = \mathbf{b}$, especially when A includes parameters.
- Find the area of a parallelograph with given vertices, using determinants.
- Find the entries of the inverse of a 3×3 matrix using Cramer's rule.

4. Vector spaces

4.1. **Vector spaces and subspaces.** A vector space is any bunch of things that you can add and multiply by scalars in a way that satisfies the usual rules of arithmetic. This includes both normal old vectors $\mathbb{R}^n$ as well as other examples. The one we talked about the most was $\mathbb{P}^n$, the polynomials of degree at most n.

If you have a vector space, a subspace of that vector space is a collection of vectors such that if you add any two, you get another, and if you multiply one by a scalar, you get another. We talked about quite a few examples and non-examples; check the notes to see some.

One basic example of a subspace is the following: if you have a bunch of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ in $\mathbb{R}^n$, then their span (i.e. the set of all linear combinations of the vectors) is a subspace.

- Tell if a given collection of vectors is a subspace or not.

4.2. **Null spaces, column spaces, and linear transformations.**

4.3. **Linearly independent sets; bases.** Suppose you have an $m \times n$ matrix A. It determines two very important subspaces: the null space $\operatorname{Nul} A$ and the column space $\operatorname{Col} A$.

The column space of A is just the span of the columns of A. All the columns are vectors in $\mathbb{R}^m$, so the column space is a subspace of $\mathbb{R}^m$. The null space of A is the set of all solutions to $A\mathbf{x} = \mathbf{0}$; this is just the set of solutions to a homogeneous linear system. This is a subspace of $\mathbb{R}^n$.

A bunch of vectors inside a subspace (or a vector space) is called a *basis* if the vectors are linearly independent and they span the subspace. This means that every vector in the subspace is a combination of vectors in the basis. But the basis can't have too many vectors in it: then they wouldn't be linearly independent.

You should know how to find a basis for the nullspace of a matrix, and a basis for the column space of a matrix. This isn't so hard: for the nullspace, what you need to do is find the general solution of $A\mathbf{x} = \mathbf{0}$, in parametric vector form. Your basis for the nullspace is then given by the vectors that appear in parametric vector form; you'll end up with one for each free variable. To find a basis for the column space of A, run row reduction on A until you reach an echelon matrix, U. Figure out which columns of U are the pivot columns. Your basis for the column space is then given by the corresponding columns of the original matrix A (not of U!)

Don't lose sight of what all this means. A basis for the column space is a collection of vectors such that everything in the nullspace (i.e. every solution of $A\mathbf{x} = \mathbf{0}$ is a linear combination of those vectors in a unique way). A basis for the column space is a collection of linearly independent vectors that span the column space – essentially what our procedure does is get rid of all the columns of A that are linearly dependent on the preceding columns, leaving us with a linearly independent set.

- Know what a basis is, and be able to check whether a given set of vectors is a basis for $\mathbb{R}^n$.
- Find a basis for the null space and column space of a matrix A.
- Find a basis for the span of a given set of vectors.

4.4. **Coordinate systems.** The important thing about a basis for a subspace is that every vector is a combination of the vectors in a unique way. What this means is that every x in the subspace can be written as

$$\mathbf{x} = c_1\mathbf{b}_1 + \cdots + c_n\mathbf{b}_n$$

in only one way. The numbers c_i are called the coordinates of x with respect to the basis $\mathcal{B}$, and we write $[\mathbf{x}]_{\mathcal{B}}$ for the vector whose entries are the c_i's.

If you know $[\mathbf{x}]_{\mathcal{B}}$ for some basis $\mathcal{B}$, you can easily figure out what $\mathbf{x}$ is: it's just $\mathbf{x} = c_1\mathbf{b}_1 + c_2\mathbf{b}_2 + c_3\mathbf{b}_3$. In matrix form, we can write this as $\mathbf{x} = \mathcal{P}_{\mathcal{B}}([\mathbf{x}]_{\mathcal{B}})$. Here $\mathcal{P}_{\mathcal{B}}$ is the "coordinate matrix": it's just the matrix you get when you write the vectors in your basis as columns.

The other thing to do is to be able to find $[\mathbf{x}]_{\mathcal{B}}$ if you're given $\mathbf{x}$. This means you want to write $c_1\mathbf{b}_1 + c_2\mathbf{b}_2 = \mathbf{x}$, which is just a vector equation in c_1 and c_2. You can find them by writing down the augmented matrix and doing row reduction. An alternative is to use the formula above to obtain $\mathbf{x} = \mathcal{P}_{\mathcal{B}}^{-1}([\mathbf{x}]_{\mathcal{B}})$ (the downside to this is that you have to invert a matrix to use it).

Another good genre of problem here is questions about polynomials. "$\mathbb{P}^n$" is the vector space of all polynomials of degree at most n. The strategy for most of these problems is the same: take each of the polynomials, stick its coefficients in a vector (this is the coordinate vector) and then check whatever you're supposed to check for the corresponding coordinate vectors instead.

- Find $\mathbf{x}$ given $[\mathbf{x}]_{\mathcal{B}}$.
- Find $[\mathbf{x}]_{\mathcal{B}}$ given $\mathbf{x}$.
- Work with coordinates for vectors in $\mathbb{P}_n$, the set of polynomials of degree $\leq n$.

4.5. **The dimension of a vector space.** The dimension of a vector space (or subspace of a vector space) is the number of vectors in a basis. The reason this makes sense is that any two bases have the same number of vectors – otherwise this wouldn't be a very useful definition.

To find the dimension of a vector space, you just have to find a basis (using one of the methods discussed above), and then count how many vectors you ended up with. That's all there is to it.

- Find the dimension of $\operatorname{Nul} A$ for a matrix A.
- Find the dimension of $\operatorname{Col} A$ for a matrix A.

4.6. **Rank.** The rank of a matrix is the dimension of its column space. The main fact about it is *the rank theorem*, which says that if A is an $m \times n$ matrix,

$$\operatorname{rank} A + \dim \operatorname{Nul} A = n.$$

This lets you find the dimension of the null space if you know the rank, and vice versa. This is surprisingly handy; there were a few questions on the homework about this.

Another topic introduced in this section was the row space of a matrix. As you can probably guess by now, the row space of an $m \times n$ matrix is the span of its rows. Since the rows have n entries, this is a subspace of $\mathbb{R}^n$. There's a recipe to compute it, as usual: do row reduction on A until you reach an echelon form U. A basis for the row space is given by the nonzero vectors in U (remember that your echelon form is likely to have a bunch of rows of 0s at the bottom; we don't want those). The other important fact about rank is that the dimension of the row space is equal to the dimension of the column space. (This is not obvious at all!)

A note: given a matrix A, you can find bases for the row space and column space just by doing row reduction until you get to echelon form. Then you see what are the pivot columns (which tells you the column space) and what are the nonzero rows (which tells you the row space).

- Compute the rank of a given matrix.
- Find the rank of a matrix given the dimension of the null space, or the other way a round.

- Find a basis for the row space of a matrix.

4.7. **Change of basis.** This is possibly the most annoying chapter, but it is important to know. Sometimes you are dealing with some vector space, and you have two different bases $\mathcal{B}$ and $\mathcal{C}$. It's important to know how to find the coordinates of a vector in $\mathcal{C}$ fi you already know the coordinates in $\mathcal{B}$, and vice versa. It turns out that all you need to do is apply a certain matrix to the coordinates. More precisely,

$$[\mathbf{x}]_{\mathcal{C}} = \underset{\mathcal{C}\leftarrow\mathcal{B}}{\mathcal{P}}([\mathbf{x}]_{\mathcal{B}}).$$

The question is how to figure out what the matrix $\underset{\mathcal{C}\leftarrow\mathcal{B}}{\mathcal{P}}$ is. There are two versions of this, depending on what exactly we're dealing with.

Possibly the main case of this is when you have two bases $\mathcal{B}$ and $\mathcal{C}$ for $\mathbb{R}^n$. In this case, we the formula is pretty easy: you want $\underset{\mathcal{C}\leftarrow\mathcal{B}}{\mathcal{P}} = \mathcal{P}_{\mathcal{C}}^{-1}\mathcal{P}_{\mathcal{B}}$. You can either calculate this directly (pretty painless in the 2×2 case), or use a row reduction trick. Start with $[\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{b}_1\ \mathbf{b}_2]$, do row reduction, and you end up with $[I\ A]$, and A is the matrix you want to use $\underset{\mathcal{C}\leftarrow\mathcal{B}}{\mathcal{P}} = [[\mathbf{b}_1]_{\mathcal{C}}\ [\mathbf{b}_2]_{\mathcal{C}}\ [\mathbf{b}_3]_{\mathcal{C}}]$: express each of your $\mathcal{B}$-basis vectors in $\mathcal{C}$-coordinates, and use that as the change of basis matrix.

The other version of this is when you have two bases for a vector space that isn't $\mathbb{R}^n$; maybe it's the column space of a matrix, or maybe it's a set of polynomials,.... In that case you want

Another thing to know: if you want to go from $\mathcal{C}$ to $\mathcal{B}$ instead of the other way around, you can use the fact that $\underset{\mathcal{B}\leftarrow\mathcal{C}}{\mathcal{P}} = \underset{\mathcal{C}\leftarrow\mathcal{B}}{\mathcal{P}}^{-1}$.

- Find the change of basis matrix (in either situation above).
- Use it to convert coordinates of vectors between different coordinate systems.

4.9. **Application to Markov chains.** A matrix A is called a *stochastic* matrix if all its entries are non-negative and the columns all add up to 1. A vector is called a probability vector if its entries add up to 1. If you apply a stochastic matrix to a probability vector, you get another probability vector. A *Markov chain* is a sequence of probability vectors obtained by taking $\mathbf{x}_{k+1} = A\mathbf{x}_k$.

We ran into some examples earlier in the course, in Section 1.9 when we discussed migration matrices. There are lots more examples in the book and the homework. make you you understand them, and you can translate a word problem into the formula for a Markov process.

The part that's new here was about "steady-state vectors". The deal is this: if you apply a stochastic matrix A to a vector many, many, many times, the vectors $\mathbf{x}_k$ you get are going to get very close to a certain vector $\mathbf{q}$, called the *steady state*. This is a kind of equilibrium: if the system is describing migration, it will eventually level off. The amounts that it levels off at are the steady states.

Finding the steady state isn't so hard. We want a vector $\mathbf{q}$ with $A\mathbf{q} = \mathbf{q}$. That means that $(A - I)\mathbf{q} = \mathbf{0}$. So write down the matrix $A - I$, and find $\mathbf{q}$ that's in its nullspace. That will be the steady state you're after.

- Translate a word problem into a stochastic matrix.
- Find the steady state of a Markov chain.

5. Eigenvalues and eigenvectors

5.1. Eigenvectors and eigenvalues.

5.2. The characteristic equation. A vector $\mathbf{x}$ is called an eigenvector of the matrix A if $A\mathbf{x} = \lambda\mathbf{x}$, where λ is a number (called the eigenvalue): this means that $A\mathbf{x}$ points in the same direction as $\mathbf{x}$ (or the opposite direction, in case $\lambda < 0$).

This is equivalent to saying that $(A - \lambda I)\mathbf{x} = \vec{0}$, which is to say that $\mathbf{x}$ is in the nullspace of $A - \lambda I$. This is the observation that lets us find the eigenvectors.

There are a couple cases where this has particular geometric significance: $\mathbf{x}$ is an eigenvector with $\lambda = 1$ means that $\mathbf{x}$ doesn't change when you apply A; with $\lambda = 0$ means that $A\mathbf{x} = \vec{0}$, i.e. $\mathbf{x}$ is in the nullspace of A; with $\lambda = -1$ means that the direction of $\mathbf{x}$ is reversed when we apply A to $\mathbf{x}$.

For most values of λ, the matrix $A - \lambda I$ won't have a nullspace at all. The only times it does is when $\det(A - \lambda I) = 0$, and so the eigenvalues are precisely the solutions of $\det(A - \lambda I) = 0$. Once you've find an eigenvalue, the way to find the corresponding eigenvector is to write down the matrix $A - \lambda I$ for that value of λ, and then find something in the nullspace using the usual procedure (elimination, special solutions).

- Find the eigenvalues of a matrix.
- Find the eigenvectors corresponding to all the eigenvalues.

5.3. Diagonalization. If we have a bunch of eigenvectors for a matrix A, we can put all of them as the columns of a matrix P, and the eigenvalues as the diagonal entries of a diagonal matrix D. These will satisfy $A = PDP^{-1}$. Part of doing diagonalization is knowing how to invert the matrix P, something we covered earlier in the course, so be ready for it. Make sure you remember the quick way to invert a 2×2 matrix. This will let you find P^{-1} without having to think too hard.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

One thing you can do with this is compute powers of a matrix with ease. If we have $A = PDP^{-1}$, then $A^n = (PDP^{-1})^n = PD^nP^{-1}$. To find D^n you just raise the diagonal entries of D to the nth power. Then to find A^n, just multiply out PD^nP^{-1}, a product of three matrices.

Not every matrix is diagonalizable. If A has n *distinct* eigenvalues, then for each eigenvalue we can find an eigenvector. Eigenvectors with different eigenvalues are automatically independent, so that gives us n independent eigenvectors. Put those into a matrix P as above, and tada, it's diagonalized. Let me stress: if the eigenvalues are all distinct, diagonalization is automatic. If there's a repeated eigenvalue, things can go either way: maybe it's diagonalizable, maybe it isn't. You have to check.

Problems can arise when A has a repeated eigenvalue. It's only guaranteed that we can find a single eigenvector for that eigenvalue, which isn't enough to make a square matrix P. It's still possible that A can be diagonalized, but you actually need to check for eigenvectors. The typical example of this is something like $\left(\begin{smallmatrix} 2 & 1 \\ 0 & 2 \end{smallmatrix}\right)$, which has only a single eigenvector.

- Compute the diagonalization of a matrix A.
- Compute powers A^n of a matrix A.

5.4. **Eigenvectors and linear transformations.** Earlier in the class, we talked about finding the matrix for a transformation. This is the same general deal. You have two vector spaces, let's say V and W. You have a basis $\mathcal{B}$ for V and a basis $\mathcal{C}$ for W. V and W might just be $\mathbb{R}^m$ and $\mathbb{R}^n$, but they could also be the vector vector spaces we've talked about, like vector spaces of polynomials of some degree.

The point is that you can find a matrix M for any transformation $T : V \to W$. Given the $\mathcal{B}$-coordinates of a vector $\mathbf{v}$ in V, you apply the matrix M, and it outputs the $\mathcal{C}$ coordinates of the vector $T(\mathbf{v})$. The rule for finding it is

$$M = \big[[T(\mathbf{b}_1)]_{\mathcal{C}}\; T(\mathbf{b}_2)_{\mathcal{C}} \cdots T(\mathbf{b}_n)]_{\mathcal{C}}\big].$$

When we're just dealing with regular bases for regular vectors, what this means is you take the vectors $\mathbf{b}_i$ in your basis. For each one of them, compute $T(\mathbf{b}_i)$ the find the $\mathcal{C}$-coordinates for that vector, and stick those together as the clomuns of a matrix. This gives the M you're looking for. (This also applies to maps between spaces of polynomials.)

This is especially useful to do when V and W are the same, so you have a map from a vector space to itself (when your transformation is just given by a matrix, this corresponds to the case that you're dealing with a square matrix). In this case, if you have a matrix A giving the transformation, you can take an eigenbasis $\mathcal{B}$ for A. Then the matrix for $\mathcal{B}$ with respect to the eigenbasis is given by the matrix D you get when you diagonalize.

- Find the matrix of a transformation with respect to given bases.
- Find a basis with respect to which a transformation is given by a diagonal matrix.

5.5. **Complex eigenvalues.** It's a fact of life that sometimes when you diagonalize a square matrix with real entries, you're forced to confront complex eigenvalues: the roots of the characteristic polynomial could be complex eigenvalues. This isn't such a big deal, but certain things need to be handled a little differently in this case.

To find the eigenvalues, proceed as usual. To find the eigenvectors, you write down $A - \lambda I$ and find a vector in the nullspace. This can be done using row reduction in the usual way, but things are a little messier because of the complex numbers involved. I think the best way to do complex row reduction is just to think to yourself about what row reduction move you would do if they matrix were full of real numbers instead. For example, if you have a row that started with a 7, you'd want to divide that row by 7. If you have a row that starts with $2 + 3i$, just do the same thing: divide every number in th erow by $2 + 3i$.

Probably the quicker way to find the eigenvectors of a matrix with complex eigevalues, at least if it's 2×2, is just to use the shortcut we discussed: write down $A - \lambda I$, take the two entries in the first row, swap them, and add a $-$ sign to one of them. This gives an eigenvector and saves you from having to do any kind of row reduction with complex numbers.

There's another trick to keep in mind. Once you've found one of the eigenvalue/eigenvector pairs, you don't have to do any more work to find the other (if you're dealing with 2×2 matrices, anyway). The second eigenvalue is the complex conjugate of the first, and the second eigenvector is the complex conjugate of the first. So just go through the eigenvector you found for the first eigenvalue, and change all the $a + bi$ to $a - bi$.

It would also be nice to diagonalize a matrix with complex eigenvalues, and you can do this in the normal way, but sometimes this isn't desirable (you might be dealing with some kind of physical problem where this doesn't really make sense, for example.) Here's the best you can do: you can get $A = PCP^{-1}$. If $a - bi$ is an eigenvalue with eigenvector $\mathbf{v}$, you want

P to have first column the real part of $\mathbf{v}$ and second column the imaginary part of $\mathbf{v}$, and $C = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}$.

- Find the complex eigenvalues and eigenvectors of a matrix.
- Write $A = PCP^{-1}$.
- Understand linear transformations given by matrices of the form $\begin{bmatrix} a & -b \\ b & a \end{bmatrix}$.

5.7. **Applications to differential equations.** We studied differential equations of the form $\mathbf{x}' = A\mathbf{x}$, where $\mathbf{x} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$ is a vector whose entries are two functions. The derivative of x_1 depends on both x_1 and x_2, and similarly for x_2. Such things can arise in a variety of applications. The challenge to is find functions that solve the equation.

The basic goal is to find two independent solutions to the equation. Once you manage that, every other solution is a linear combination of those two. How to get the two basic solutions depends on whether there are two real eigenvalues, or two (conjugate) complex eigenvalues.

If there are two real eigenvalues λ_1 and λ_2, with eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$, then the general solution is

$$\mathbf{x}(t) = c_1\mathbf{v}_1 e^{\lambda_1 t} + c_2\mathbf{v}_2 e^{\lambda_2 t}.$$

Often an initial condition $\mathbf{x}(0)$ is given, and you can use this to solve for the constants c_1 and c_2.

If there are two conjugate eigenvalues $a + bi$ and $a - bi$ instead, here's what you need to do. It's still true that $\mathbf{v}_1 e^{\lambda_1 t}$ is a solution, but it has complex entries, which we don't want. To get two real solutions, you need to take the real and complex parts of this thing. That's a bit of a hassle: you need to expand $e^{(a+bi)t}$ as $e^{at}(\cos bt + i \sin bt)$, and then multiply that by the eigenvector $\mathbf{v}_1$. To actually find the real and imaginary parts, you need to multiply it out the whole way (see the homework sols for some examples).

- Write down the general solution of a couple differtial equation with real eigenvalues.
- Write down the general solution of a couple differtial equation with complex eigenvalues.
- Find the solution of a differential equation with given initial conditions.

6. Orthogonality

6.1. **Inner product, length and orthogonality.**

6.2. **Orthogonal sets.** To find the dot product of two vectors, you just need to multiply the corresponding entries of the two vectors and add them all up. The length of a vector $\mathbf{v}$ is given by $\sqrt{\mathbf{v} \cdot \mathbf{v}}$, and the distance between $\mathbf{u}$ and $\mathbf{v}$ is the length of $\mathbf{u} - \mathbf{v}$. Two vectors are orthogonal if their dot product is 0.

If $V \subset \mathbb{R}^n$ is a subspace, then the set of all vectors orthogonal to V is also a subspace, called the *orthogonal complement* of V and written $V^\perp$. You should keep in mind the examples where $n = 3$. If $V \subset R^3$ is a plane (i.e. a 2-dimensional subspace), then $V^\perp$ is the line through the origin that's perpendicular to that plane.

A bunch of vectors is called an orthogonal set if all of the vectors are orthogonal to each other. Any set of orthogonal vectors is automatically linearly independent. An orthogonal basis for a subspace $V \subset \mathbb{R}^n$ is a basis for V that's given by a bunch of orthogonal vectors.

Working with an orthogonal basis is convenient in many ways. For example, if $\mathbf{u}_1, \dots, \mathbf{u}_m$ are a basis for a subspace V, it's easy to find the coordinates of a vector $\mathbf{x}$ with respect to this basis: $\mathbf{x} = c_1\mathbf{u}_1 + \cdots + c_m\mathbf{u}_m$, where $c_i = \frac{\mathbf{x}\cdot\mathbf{u}_i}{\mathbf{u}_i\cdot\mathbf{u}_i}$.

A bunch of vectors are said to be orthonormal if they are an orthogonal set and all of the vectors have length 1. A useful fact is that a matrix U has orthonormal columns.

6.3. **Orthogonal projections.** If you have a subspace $V \subset \mathbb{R}^n$ and a vector $\mathbf{x}$, you can write $bx = \mathbf{x}_V + \mathbf{x}_{V^\perp}$, where $\mathbf{x}_V$ is a vector in V and $\mathbf{x}_{V^\perp}$. If you have an orthonormal basis for V, it's pretty easy to do this:

$$\mathbf{x}_V = \frac{\mathbf{x}\cdot\mathbf{u}_1}{\mathbf{u}_1\cdot\mathbf{u}_1}\mathbf{u}_1 + \frac{\mathbf{x}\cdot\mathbf{u}_2}{\mathbf{u}_2\cdot\mathbf{u}_2}\mathbf{u}_2 + \frac{\mathbf{x}\cdot\mathbf{u}_3}{\mathbf{u}_3\cdot\mathbf{u}_3}\mathbf{u}_3.$$

Then $\mathbf{x}_{V^\perp} = \mathbf{x} - \mathbf{x}_V$. Usually we write $\operatorname{proj}_V \mathbf{x}$ for what I'm calling $\mathbf{x}_V$. The vector $\operatorname{proj}_V \mathbf{x}$ has additional useful property that it's the point in V that's as close as possible to the vector $\mathbf{x}$.

6.4. **The Gram–Schmidt process.** The previous two sections hopefully convinced you that orthogonal bases can be very convenient for making computations. The problem is that usually when you find a basis for something, the one you get isn't orthogonal: the vectors that show up won't dot to 0. That's the problem solved by the Gram–Schmidt process: you input a basis (not necessarily orthogonal), and it gives you a basis that is orthogonal.

The rule is this. Say you're handed a basis $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$.

$$\begin{aligned}
\mathbf{v}_1 &= \mathbf{x}_1 \\
\mathbf{v}_2 &= \mathbf{x}_2 - \frac{\mathbf{x}\cdot\mathbf{v}_1}{\mathbf{v}_1\cdot\mathbf{v}_1}\mathbf{v}_1 \\
\mathbf{v}_3 &= \mathbf{x}_3 - \frac{\mathbf{x}\cdot\mathbf{v}_1}{\mathbf{v}_1\cdot\mathbf{v}_1}\mathbf{v}_1 - \frac{\mathbf{x}\cdot\mathbf{v}_2}{\mathbf{v}_2\cdot\mathbf{v}_2}\mathbf{v}_2
\end{aligned}$$

You can probably figure out how to adjust this is the basis you're given has more than 3 vectors.

This also gives another important matrix decomposition, the QR factorization. Suppose you have an $m\times n$ matrix A with linearly independent columns. You can write it as $A = QR$, where Q is an $m \times n$ matrix with orthonormal columns (in fact, an orthonormal basis for the column space of A), and R is an $n \times n$ invertible matrix with positive entries on the diagonal (NB: Q is the same size as A, and R is a square matrix.)

Here's how to find it: let Q be a matrix whose columns are the output of Gram–Schmidt on the columns of A (be sure to use the columns *in order* as your original vectors $\mathbf{x}_i$). Then $R = Q^TA$. (There's a slightly more efficient way to do this analogous to what we did to find the LU decomposition, but this is easier to remember.)

6.5. **Least-squares problems.**

6.6. **Application to linear models.** The situation where least-squares problems arise is this: you have some system of linear equations $A\mathbf{x} = \mathbf{b}$, but there are no solutions because $\mathbf{b}$ is not in the column space $\operatorname{Col} A$. What you do instead is solve $A\mathbf{x} = \hat{\mathbf{b}}$, where $\hat{\mathbf{b}} = \operatorname{proj}_{\operatorname{Col} A}(\mathbf{b})$ is the projection of $\mathbf{b}$ onto the column space of A.

There's a shortcut for doing this: just solve $A^TA\mathbf{x} = A^T\mathbf{b}$, which has the same effect. These are called the "normal equations" for $A\mathbf{x} = \mathbf{b}$.

A typical setting where this method comes in handy is in trying to fit a curve of a specific type through a bunch of data points: say we have a bunch of pairs (x_1, y_1), ..., (x_n, y_n) and want to find the best-fit parabola $y = at^2 + bt + c$. To find a parabola going through all the points, we'd want a, b, and c to satisfy $y_i = ax_i^2 + bx_i + c$ for every i. This is a linear equation ("$A\mathbf{x} = \mathbf{b}$") in a, b, c:

$$\begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ & \cdots & \\ x_n^2 & x_n & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}$$

In general this is lots of equations in three variables, and there are no solutions (the right side isn't in the column space of the matrix). That means there aren't any parabolas through all the points. But you can solve $A^T A\mathbf{x} = A^T\mathbf{b}$ instead, and that will be the best fit. The same strategy works for curves other than parabolas as well.

7. Symmetric matrices and quadratic forms

7.1. **Diagonalization of symmetric matrices.** A square matrix A is symmetric if $A^T = A$, so that when you flip it over the main diagonal you get the same matrix. The most interesting thing about symmetric matrices is that they can always be diagonalized, and not only that, the columns of P (i.e. the eigenvectors) are all orthogonal. This is the content of the *spectral theorem*: If A is an $n \times n$ symmetric matrix, then

- A has n real eigenvalues (maybe with multiplicities);
- There is a linearly independent set of eigenvectors;
- The eigenspaces are orthogonal to each other;
- A can be diagonalized PDP^{-1} where P has orthonormal columns.