Math 307 Learning Goals

March 23, 2010

Course Description

The course presents core concepts of linear algebra by focusing on applications in Science and Engineering. Examples of applications from recent years include: interpolation, finite difference approximations, least squares, Fourier series, graphs and networks, FFT, JPEG compression, power method, recursion relations, the Anderson tight binding model, Markov chains, Google PageRank, principal component analysis.
Students in the course are required to use a computer algebra system (Matlab or Octave) to perform computations related to these applications.

Course Learning Objectives

After taking this course, students should be able to apply basic ideas about linear systems, bases and subspaces associated to matrices, orthogonality and projections, eigenvalues and eigenvectors to study problems arising from real-world applications, and be able to construct computer programs related to these applications.
There could be considerable variation in the application content and the order in which it is dealt with, depending on the instructor. However, all students who receive a passing grade in the course should be able to demonstrate achievement of the following minimal goals:

- Define, compare, and compute the 1-, 2-, and infinity norms of a vector; define and, in specific cases, compute the matrix norm and condition number of a matrix.

- Discuss the sensitivity of a linear equation $A\mathbf{x} = \mathbf{b}$ to small changes (errors) in $\mathbf{b}$, and make estimates of the relative error in the solution $\mathbf{x}$.

- Translate real-world problems involving linear systems of equations into linear algebra problems and solve them (examples of such problems are interpolation problems and finite difference approximations): students should be able to define the problem in mathematical terms and solve it by deriving a matrix equation, computing solutions when they exist, and discussing the sensitivity of the equation to small errors.

- Give a definition of and explain the meaning of some basic linear algebra concepts such as vector space, subspace, basis of vectors, dimension of a subspace, linear (in)dependence of vectors, nullspace (or kernel) $N(A)$ range (or image) $R(A)$ of a matrix A, rank of a matrix, orthogonal vectors, orthogonal complement.

- Write down relations between the size of a matrix A, the rank and dimensions of $N(A)$, $R(A)$, $N(A^T)$, and $R(A^T)$ and explain why they are true; explain why $N(A)$ and $R(A^T)$ are orthogonal subspaces.

- Apply basic ideas about subspaces, bases and dimension to directed graphs; an example of graphs applied to real-world problems are the directed graphs that describe resistor networks and electric circuits: in this case, given a graph students should be able to find quantitative information about currents, voltages, and effective resistances for the circuit.

- Define the inner product and norm for the n-dimensional vector spaces $\mathbb{R}^n$ and $\mathbb{C}^n$ and vector spaces of functions, list and apply their properties, compute inner products and norms in given examples.

- Define orthogonal and orthonormal bases, determine when a collection of vectors is an orthogonal/orthonormal basis, and determine how to expand vectors in such bases.

- Identify the real and imaginary parts, modulus, argument, and complex conjugate of a complex number, perform arithmetic with complex numbers and basic matrix operation with complex vectors and matrices; define and perform basic computations (addition, differentiation, integration) with the complex exponential function.

- Derive the least square equation, show that it has always a solution, and apply the least squares technique to curve-fitting problems and other applications where overdetermined systems arise.

- Explain how a function defined on a given interval and with finite norm can be expanded in series using a Fourier basis, and find such series in specific examples.

- Define the discrete Fourier transform and perform basic signal analysis using the Fast Fourier Transform algorithm.

- Define and recognize when a matrix is stochastic, list its basic properties and show that they are true.

- Determine when a matrix is diagonalizable and use eigen-analysis to perform matrix diagonalization and to study the behaviour of powers of a diagonalizable matrix.

- Use the results of eigen-analysis to find eigenvalues of symmetric and Hermitian matrices using the power method, to determine the asymptotic behaviour of the solutions to a recursion relation, to find the limiting distribution of a Markov chain (when it exists) and solve practical Markov chain problems.

In addition to the above goals, students should be able to create computer programs in Matlab/Octave to test mathematical conjectures involving the concepts listed above and to perform all relevant computations arising in the applications discussed in the course. In particular, in the Matlab/Octave environment students should demonstrate achievement of the following minimum goals:

- Create matrices and vectors by either entering matrix entries by hand, generating larger matrices from smaller blocks, extracting submatrices from larger matrices, or executing specific commands such as `diag`, `rand`, `linspace`, `zeros`, `ones`, `vander`, `eye`, etc.
- Perform basic matrix manipulations and operations: this includes recognizing when a variable is a scalar or a matrix, finding its dimension, performing basic scalar operations (addition, multiplication, division, etc.) and matrix operations (addition `+`, multiplication `*`, powers `^`) and constrasting these with componentwise operations implemented with `.+`, `.*`, and `.^`, also computing the dot product, finding the inverse and transpose, extracting or modifying single elements, rows, columns or submatrices from larger matrices, and computing norms, condition numbers, and eigenvalues/eigenvectors by executing specific commands.
- Compute the value of mathematical functions using specific commands such as `sin`, `cos`, `exp`, `abs`, etc.
- Write code including `for` loops.
- Use relational operators (e.g. `>`, `=<`, `==`); write code including `if` statements.
- Execute and interpret the output of the `rref` command and the \ operator; describe the limitations of the \ operator for finding solution(s) to a linear system.
- Plot data using the `plot` command; this includes determining adequate ranges for the independent axis, adjusting axis scale as necessary, creating overlapping plots using different plotting styles, inserting a legend.
- Compute the Fast Fourier Transform of a vector using the `fft` command.
- Create, execute, and interpret the output of m files (both Matlab/Octave scripts and functions).
- Search for information with the `help` command.

1 Linear Equations

1.1 Solving Linear Equations

Prerequisites and Learning Goals

From your work in previous courses, you should be able to

- Write a system of linear equations using matrix notation.
- Use Gaussian elimination to bring a system of linear equations into upper triangular form and reduced row echelon form (rref).
- Determine whether a system of equations has a unique solution, infinitely many solutions or no solutions, and compute all solutions if they exist; express the set of all solutions in parametric form.
- Compute the inverse of a matrix when it exists, use the inverse to solve a system of equations, describe for what systems of equations this is possible.
- Find the transpose of a matrix.
- Interpret a matrix as a linear transformation acting on vectors.

After completing this section, you should be able to

- Write down the definition and explain the meaning of the standard Euclidean norm, the 1-norm and the infinity norm of a vector; compute norms by hand or using the MATLAB/Octave command `norm`.
- Calculate the Hilbert-Schmidt norm of a matrix.
- Define the matrix norm of a matrix; describe the connection between the matrix norm and how a matrix stretches the length of vectors and express it in a mathematical form; explain what range of values the matrix norm can take; compute the matrix norm of a diagonal matrix and, given sufficient information, of other types of matrices.
- Find solutions to linear equations of the form $A\mathbf{x} = \mathbf{b}$ by executing and interpreting the output of the MATLAB/Octave commands `rref` and `\`.
- Define the condition number of a matrix and its relation to the matrix norm; compute it by hand when possible or using the MATLAB/Octave command `cond`.

- Discuss the sensitivity of a linear equation $A\mathbf{x} = \mathbf{b}$ by relating changes in the solution $\mathbf{x}$ to small changes in $\mathbf{b}$; define the relative error, explain why the condition number is useful in making estimates of relative errors in the solution $\mathbf{x}$, and use it to make such estimates.

1.2 Interpolation

Prerequisites and Learning Goals

From your work in previous courses, you should be able to

- compute the determinant of a square matrix; apply the basic linearity properties of the determinant, and explain what its value means about existence and uniqueness of solutions.

After completing this section, you should be able to

- Give a definition of an interpolating function and show how a given interpolation problem (i.e. the problem of finding an interpolating function of given shape/form) translates into finding the solutions to a linear system; explain the idea of getting a unique interpolating function by restricting the class of functions under consideration.

- Define the problem of Lagrange interpolation and express it in terms of a system of equations where the unknowns are the coefficients of a polynomial of given degree; set up the system in matrix form using the Vandermonde matrix, derive the formula for the determinant of the Vandermonde matrix; explain why a solution to the Lagrange interpolation problem always exists.

- Define the mathematical problem of interpolation using splines, this includes listing the conditions that the splines must satisfy and translating them into mathematical equations; express the spline-interpolation problem in terms of a system of equations where the unknowns are related to the coefficients of the splines, derive the corresponding matrix equation.

- Compare and constrast interpolation with splines with Lagrange interpolation.

- Explain how minimizing the bending energy leads to a description of the shape of the spline as a piecewise polynomial function.

- Given a set of points, use MATLAB/Octave to calculate and plot the interpolating functions, including the interpolating polynomial in Lagrange interpolation and the piecewise cubic function for splines; this requires you be able to execute and interpret the output of specific commands such as `vander`, `polyval`, `plot`.

- Discuss how the condition numbers arising in the above interpolation problems vary; explain why Lagrange interpolation is not a practical method for large numbers of points.

1.3 Finite difference approximations

Prerequisites and Learning Goals

From your work in previous courses, you should be able to

- explain what it is meant by a boundary value problem.

After completing this section, you should be able to

- compute an approximate solution to a second order linear boundary value problem by deriving the corresponding finite difference equation and computing its solution with MATLAB/Octave; model different types of boundary conditions, including conditions on the value of the solution at the boundary and on the value of the first derivative at the boundary.

2 Subspaces, Bases and Dimension

2.1 Subspaces, basis and dimension

Prerequisites and Learning Goals

From your work in previous courses, you should be able to

- Write down a vector as a linear combination of a set of vectors, and express the linear combination in matrix form.
- Define linear independence for a collection of vectors.
- Define a basis for a vector subspace.

After completing this section, you should be able to

- Explain what a vector space is, give examples of vector spaces.
- Define vector addition and scalar multiplication for vector spaces of functions.
- Give a definition of subspace and decide whether a given collection of vectors forms a subspace.
- Recast the dependence or independence of a collection of vectors in $\mathbb{R}^n$ or $\mathbb{C}^n$ as a statement about existence of solutions to a system of linear equations, and use it to decide if a collection of vectors are dependent or independent.
- Define the span of a collection of vectors; show that given a set of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ the span $\mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ is a subspace; determine when a given vector is in the span.
- Describe the significance of the two parts (independence and span) of the definition of a basis.
- Check if a collection of vectors is a basis.
- Show that any basis for a subspace has the same number of elements.
- Show that any set of k linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ in a k dimensional subspace S is a basis of S.
- Define the dimension of a subspace.

2.2 The four fundamental subspaces for a matrix

From your work in previous courses, you should be able to

- Recognize and use the property of transposes for which $(AB)^T = B^T A^T$ for any matrices A and B.

- Define the inner (dot) product of two vectors, and its properties (symmetry, linearity), and explain its geometrical meaning.

- Use the inner product to decide if two vectors are orthogonal, and to compute the angle between two vectors.

- State the Cauchy-Schwarz inequality and know for which vectors the inequality is an equality.

- For a linear system classify variables as basic and free.

After completing this section, you should be able to

- Define the four fundamental subspaces $N(A)$, $R(A)$, $N(A^T)$, and $R(A^T)$, associated to a matrix A and its transpose A^T, and show that they are subspaces.

- Express the Gaussian elimination process performed to reduce a matrix A to its row reduced echelon form matrix U as a matrix factorization, $A = EU$, using elementary matrices, and perform the steps using MATLAB/Octave.

- Compute bases for each of the four fundamental subspaces $N(A)$, $R(A)$, $N(A^T)$ and $R(A^T)$ of a matrix A; infer information on a matrix given the bases of its fundamental subspaces.

- Define and compute the rank of a matrix.

- State the formulas for the dimension of each of the four subspaces and explain why they are true; when possible, use such formulas to find the dimension of the subspaces.

- Explain what it means for two subspaces to be orthogonal ($V \perp W$) and for one subspace to be the orthogonal complement of another ($V = W^{\perp}$).

- State which of the fundamental subspaces are orthogonal to each other and explain why, verify the orthogonality relations in examples, and use the orthogonality relation for $R(A)$ to test whether the equation $A\mathbf{x} = \mathbf{b}$ has a solution.

2.3 Graphs and Networks

Prerequisites and Learning Goals

From your work in previous courses you should be able to

- State Ohm's law for a resistor.
- State Kirchhoff's laws for a resistor network.

After completing this section, you should be able to

- Write down the incidence matrix of a directed graph, and draw the graph given the incidence matrix.
- Define the Laplace operator L, or Laplacian, for a graph; given a graph, determine the entries of L and describe how they are related to the nodes and edges of the graph.
- When the edges of a graph represent resistors or batteries in a circuit, you should be able to
  - define voltage, voltage difference, current, and loop vectors;
  - interpret each of the four subspaces associated with the incidence matrix D and their dimensions in terms of voltage, current, and loop vectors; relate the dimension of $N(D)$ to the number of disconnected pieces in a graph; find bases for each subspace of D and verify the orthogonality relations between such subspaces;
  - interpret the range and the nullspace of the Laplacian in terms of voltage and current vectors, and give a physical justification of such interpretation;
  - explain what happens to L if two nodes in the graph are renamed and why and when it is useful to do so;
  - construct the voltage-to-current map for a pair of nodes and use it to calculate voltages, currents, or effective resistances; perform all necessary computations in MATLAB/Octave.

3 Orthogonality

3.1 Projections

Prerequisites and Learning Goals

After completing this section, you should be able to

- Write down the definition of an orthogonal projection matrix and determine when a matrix is an orthogonal projection.

- Identify the range of a projection matrix as the subspace onto which it projects and use properties of the projection matrix to derive the orthogonality relation between the range and nullspace of the matrix.

- Given the projection matrix onto a subspace S, compute the projection matrix onto $S^{\perp}$; describe the relations between the nullspace and range of the two matrices.

- Use orthogonal projection matrices to decompose a vector into components parallel to and perpendicular to a given subspace.

- Explain how the problem of finding the projection of a vector $\mathbf{b}$ onto a certain subspace spanned by the columns of a matrix A translates into finding a vector $\mathbf{x}$ that minimizes the length $||A\mathbf{x} - \mathbf{b}||$; show that $\mathbf{x}$ always exists and satisfies the least squares equation; discuss how the results of the minimization problem can vary depending on the type of norm used; discuss the sensitivity of a least squares fit to outliers.

- Compute the orthogonal projection matrix whose range is the span of a given collection of vectors.

- Perform least squares calculations to find polynomial fits to a given set of points, or in other applications where overdetermined systems arise. You should be able to perform all necessary computations and plot your results using MATLAB/Octave.

- Interpret the output of the MATLAB/Octave \ command when applied to systems that have no solutions.

3.2 Orthonormal bases and Orthogonal Matrices

Prerequisites and Learning Goals

After completing this section, you should be able to

- Write down the definition of an orthonormal basis, and determine when a given set of vectors is an orthonormal basis.
- Compute the coefficients in the expansion of a vector in an orthonormal basis.
- Compute the norm of a vector from its coefficients in its expansion in an orthonormal basis.
- Write down the definition of an orthogonal matrix; recognize when a matrix is orthogonal; describe the action of an orthogonal matrix on vectors; describe the properties of the rows and columns of an orthogonal matrix.

3.3 Complex vector spaces and inner product

Prerequisites and Learning Goals

From your work in previous courses, you should be able to

- Perform arithmetic with complex numbers.
- Write down the definition of and compute the complex conjugate, modulus and argument of a complex number.

After completing this section, you should be able to

- Define and perform basic matrix calculations with complex vectors and complex matrices.
- Define and compute the complex inner product and the norm of complex vectors, state basic properties of the complex inner product.
- Define and compute the matrix adjoint for a complex matrix; explain its relation to the complex inner product; compare its properties to the properties of the transpose of a real matrix.
- Define an orthonormal basis for $\mathbb{C}^n$ and determine whether a set of complex vectors is an orthonormal basis; determine the coefficients in the expansion of a complex vector in an orthonormal basis.
- Write down the definition of a unitary matrix and list its properties; recognize when a matrix is unitary.

- Define and compute the inner product and the norm for complex- (or real-) valued functions that are defined on a given interval; define what it means for two functions to be orthonormal, and verify it in specific examples.
- Define the complex exponential function, compute its value at given points and perform basic computations (addition, differentiation, integration) involving complex exponential functions.
- Explain what are the elements of the vector space $L^2([a, b])$ for an interval $[a, b]$.
- Use complex numbers in MATLAB/Octave computations, specifically `real(z)`, `imag(z)`, `conj(z)`, `abs(z)`, `exp(z)` and `A'` for complex matrices.

3.4 Fourier series

Prerequisites and Learning Goals

After completing this section, you should be able to

- Show that the functions $e_n(x) = e^{2\pi inx/L}$ for $n = 0, \pm1, \pm2, ...,$ $a < x < b$ and $L = b - a$ form an orthonormal (scaled by $\sqrt{L}$) set in $L^2([a, b])$.
- Use the fact that the functions $e_n(x)$ form an infinite orthonormal basis to expand a L^2 function in a Fourier series; explain how this leads to a formula for the coefficients of the series, and compute the coefficients (in real and complex form).
- State and derive Parseval's formula and use it to sum certain infinite series.
- Use MATLAB/Octave to compute and plot the partial sums of Fourier series.
- Explain what an amplitude-frequency plot is and generate it for a given function using MATLAB/Octave; describe the physical interpretation of the plot when the function is a sound wave.

3.5 The Discrete Fourier Transform

Prerequisites and Learning Goals

After completing this section, you should be able to

- Explain why the vectors in $\mathbb{C}^n$ obtained by sampling the exponential Fourier bases $e_n(t)$ form an orthogonal basis for $\mathbb{C}^n$ (discrete Fourier bases).
- Use the discrete Fourier basis to expand a vector in $\mathbb{C}^n$ obtaining the discrete Fourier transform of the vector; recognize the matrix that implements the discrete Fourier transform as an unitary matrix.

- Use the Fast Fourier transform (fft) algorithm to compute the discrete Fourier transform, and explain why the Fast Fourier transform algorithm is a faster method. You should be able to perform Fourier transform computations by executing and interpreting the output of the MATLAB/Octave `fft` command.

- Explain the relation between the coefficients in the Fourier series of a function f defined on $[0, L]$ and the coefficients in the discrete Fourier transform of the corresponding sampled values of f, and discuss its limitations.

- Construct a frequency-amplitude plot for a sampled signal using MATLAB/Octave; give a physical interpretation of the resulting plot; explain the relation between this plot and the infinite frequency-amplitude plot.

4 Eigenvalues and Eigenvectors

4.1 Eigenvalues and Eigenvectors

Prerequisites and Learning Goals

After completing this section, you should be able to

- Write down the definition of eigenvalues and eigenvectors and compute them using the standard procedure involving finding the roots of the characteristic polynomial. You should be able to perform relevant calculations by hand or using specific MATLAB/Octave commands such as `poly`, `roots`, and `eig`.
- Define algebraic and geometric multiplicities of eigenvalues and eigenvectors; discuss when it is possible to find a set of eigenvectors that form a basis.
- Determine when a matrix is diagonalizable and use eigenvalues and eigenvectors to perform matrix diagonalization.
- Recognize the form of the Jordan Canonical Form for non-diagonalizable matrices.
- Explain the relationship between eigenvalues and the determinant and trace of a matrix.
- Use eigenvalues to compute powers of a diagonalizable matrix.

4.2 Power Method for Computing Eigenvalues

Prerequisites and Learning Goals

After completing this section, you should be able to

- Define and compare real symmetric and Hermitian matrices.
- Show that the eigenvalues of Hermitian matrices are real and that eigenvectors corresponding to distinct eigenvalues are orthogonal.
- Explain why a Hermitian matrix can be diagonalized with a unitary matrix.

- Explain how to implement the power method in order to compute certain eigenvalues/eigenvectors, write out the steps of the algorithm and discuss its output; implement the algorithm in MATLAB/Octave and use it to compute the largest and smallest eigenvalues and the eigenvalue closest to a given number (and corresponding eigenvectors) of a Hermitian matrix; discuss what it means when the algorithm does not convergence.

4.3 Recursion Relations

Prerequisites and Learning Goals

After completing this section, you should be able to

- Derive a matrix equation from a recursion relation and use it to find the n-th term in the sequence given some initial conditions. You should be able to perform these computations by hand or with MATLAB/Octave.
- Discuss how the relative magnitude of the eigenvalues of the matrix obtained from a recursion relation determines the behaviour of the high power terms of the recursion.
- Determine initial values for which the solution of a recursion relation will become large or small (depending on the eigenvalues of the associated matrix).

4.4 The Anderson Tight Binding Model

Prerequisites and Learning Goals

After completing this section, you should be able to

- describe a bound state with energy E for the discrete Schrodinger equation for a single electron moving in a one dimensional semi-infinite crystal.
- describe a scattering state with energy E.
- compute the energies for which a bound state exists and identify the conduction band, for a potential that has only one non-zero value.
- compute the conduction bands for a one dimensional crystal.

4.5 Markov Chains

Prerequisites and Learning Goals

After completing this section, you should be able to

- Write down the definition of a stochastic matrix, list its properties and explain why they are true; recognize when a matrix is stochastic.
- Write down the stochastic matrix and any relevant state vectors for a random walk, use them to describe the long-time behaviour of the random walk, explain why the probabilities in a random walk approach limiting values; use stochastic matrices to solve practical Markov chain problem. You should be able to perform these computations by hand or with MATLAB/Octave.
- For an internet specified by a directed graph, write down the stochastic matrix associated with the Google Page rank algorithm for a given damping factor α (and related algorithms), and compute the ranking of the sites (by hand or with MATLAB/Octave).
- Use the Metropolis algorithm to produce a stochastic matrix with a predetermined limiting probability distribution; use the algorithm to perform a random walk that converges with high probability to the maximum value of a given positive function.

4.6 The Singular Value Decomposition

Prerequisites and Learning Goals

After completing this section, you should be able to

- explain what the singular value decomposition of a matrix is
- explain why a Hermitian matrix always has real eigenvalues and an orthonormal basis of eigenvectors.
- write down the relationship between the singular values of a matrix and its matrix norm