Linear Algebra and Differential Equations

Math 21b

Harvard University

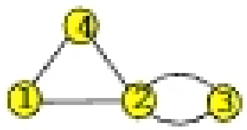
Fall 2004

Oliver Knill

- Instructor: Oliver Knill (http://www.math.harvard.edu/~knill)
- Office: 434, Science Center
- Office hours: MWF 15:00-16:00
- E-mail: knill@math.harvard.edu
- Phone: 5-5549
- Classroom: 309
- Classtime: MWF 11-12
- CA: Tien Anh Nguyen, e-mail tanguyen@fas
- Course page: http://www.courses.fas.harvard.edu/~math21b/
- Midterms Wed Oct 27 6:30pm, Wed, Dec 1, 6:00pm
- Textbook: Linear Algebra and its applications by Otto Bretscher (third edition)
- Grade: Midterms 20% each, homework: 20 %, Final: 40 %.
- Homework: Due at beginning of each class.

9/20/2004, USE OF LINEAR ALGEBRA I — Math 21b, O. Knill

This is not a list of topics covered in the course. It is rather a loose selection of subjects for which linear algebra is useful or relevant. The aim is to convince you that it is worth learning this subject. Most of this handout does not make much sense yet to you because the objects are not defined yet. You can look at this page at the end of the course again, when some of the content will become more interesting.



GRAPHS, NETWORKS. Linear algebra can be used to understand networks. A network is a collection of nodes connected by edges and are also called graphs. The adjacency matrix of a graph is defined by an array of numbers. One defines $A_{ij} = 1$ if there is an edge from node i to node j in the graph. Otherwise the entry is zero. A problem using such matrices appeared on a blackboard at MIT in the movie "Good will hunting".

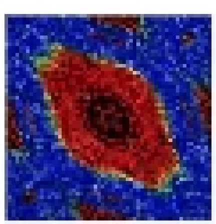
How does the array of numbers help to understand the network. One application is that one can read off the number of n-step walks in the graph which start at the vertex i and end at the vertex j. It is given by A^n_{ij}, where A^n is the n-th power of the matrix A. You will learn to compute with matrices as with numbers.

CHEMISTRY, MECHANICS Complicated objects like a bridge (the picture shows Storrow Drive connection bridge which is part of the "big dig"), or a molecule (i.e. a protein) can be modeled by finitely many parts (bridge elements or atoms) coupled with attractive and repelling forces. The vibrations of the system are described by a differential equation $\dot{x} = Ax$, where $x(t)$ is a vector which depends on time. Differential equations are an important part of this course.

The solution $x(t) = \exp(At)$ of the differential equation $\dot{x} = Ax$ can be understood and computed by finding the eigenvalues of the matrix A. Knowing these frequencies is important for the design of a mechanical object because the engineer can damp dangerous frequencies. In chemistry or medicine, the knowledge of the vibration resonances allows to determine the shape of a molecule.

QUANTUM COMPUTING A quantum computer is a quantum mechanical system which is used to perform computations. The state x of a machine is no more a sequence of bits like in a classical computer but a sequence of qubits, where each qubit is a vector. The memory of the computer can be represented as a vector. Each computation step is a multiplication $x \mapsto Ax$ with a suitable matrix A.

Theoretically, quantum computations could speed up conventional computations significantly. They could be used for example for cryptological purposes. Freely available quantum computer language (QCL) interpreters can simulate quantum computers with an arbitrary number of qubits.



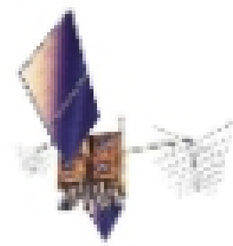
CHAOS THEORY. Dynamical systems theory deals with the iteration of maps or the analysis of solutions of differential equations. At each time t, one has a map $T(t)$ on the vector space. The linear approximation $DT(t)$ is called Jacobean is a matrix. If the largest eigenvalue of $DT(t)$ grows exponentially in t, then the system shows "sensitive dependence on initial conditions" which is also called "chaos".

Examples of dynamical systems are our solar system or the stars in a galaxy, electrons in a plasma or particles in a fluid. The theoretical study is intrinsically linked to linear algebra because stability properties often depends on linear approximations.

USE OF LINEAR ALGEBRA II — Math 21b, O. Knill

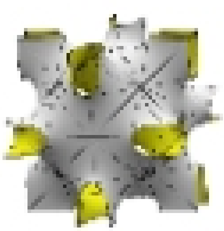
CODING, ERROR CORRECTION Coding theory is used for encryption or error correction. For encryption, data x are maped by a map T into code y=Tx. T usually is a "trapdoor function": it is hard to get x back when y is known. In the second case, a code is a linear subspace X of a vector space and T is a map describing the transmission with errors. The projection onto the subspace X corrects the error.

Linear algebra enters in different ways, often directly because the objects are vectors but also indirectly like for example in algorithms which aim at cracking encryption schemes.



DATA COMPRESSION Image- (i.e. JPG), video- (MPG4) and sound compression algorithms (i.e. MP3) make use of linear transformations like the Fourier transform. In all cases, the compression makes use of the fact that in the Fourier space, information can be cut away without disturbing the main information.

Typically, a picture, a sound or a movie is cut into smaller junks. These parts are represented by vectors. If U denotes the Fourier transform and P is a cutoff function, then $y = PUx$ is transferred or stored on a CD or DVD. The receiver obtains back $U^T y$ which is close to x in the sense that the human eye or ear does not notice a big difference.



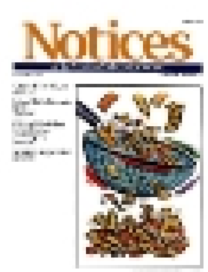
SOLVING SYSTEMS OR EQUATIONS When extremizing a function f on data which satisfy a constraint $g(x) = 0$, the method of Lagrange multipliers asks to solve a nonlinear system of equations $\nabla f(x) = \lambda \nabla g(x), g(x) = 0$ for the $(n+1)$ unknowns (x, l), where ∇f is the gradient of f.

Solving systems of nonlinear equations can be tricky. Already for systems of polynomial equations, one has to work with linear spaces of polynomials. Even if the Lagrange system is a linear system, the task of solving it can be done more efficiently using a solid foundation of linear algebra.

GAMES Moving around in a world described in a computer game requires rotations and translations to be implemented efficiently. Hardware acceleration can help to handle this.

Rotations are represented by matrices which are called orthogonal. For example, if an object located at $(0,0,0)$, turning around the y-axes by an angle ϕ, every point in the object gets transformed by the matrix

$$\begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

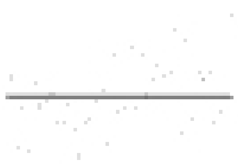


CRYPTOLOGY. Much of current cryptological security is based on the difficulty to factor large integers n. One of the basic ideas going back to Fermat is to find integers x such that x^2 mod n is a small square y^2. Then $x^2 - y^2 = 0$ mod n which provides a factor $x - y$ of n. There are different methods to find x such that x^2 mod n is small but since we need squares people use sieving methods. Linear algebra plays an important role there.

Some of the best factorization algorithms use Gaussian elimination. One is the quadratic sieve. The ongoing factorization challenge "RSA Challenge Numbers". The smallest nonfactored problem is currently the 193 digit number [illegible] called RSA-640. If you factor this number you win 20'000 dollars.

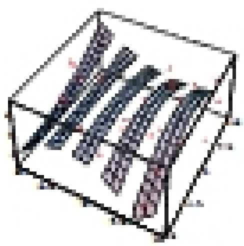
USE OF LINEAR ALGEBRA (III)

Math 21b, Oliver Knill



STATISTICS When analyzing data statistically, one often is interested in the correlation matrix $A_{ij} = E[Y_iY_j]$ of a random vector $X = (X_1, \dots, X_m)$ with $Y_i = X_i - E[X_i]$. This matrix is derived from the data and determines often the random variables when the type of the distribution is fixed.

For example, if the random variables have a Gaussian (=Bell shaped) distribution, the correlation matrix together with the expectation $E[X_i]$ determines the random variables.

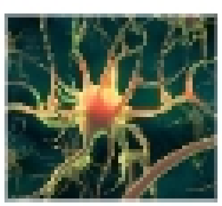


DATA FITTING. Given a bunch of data points, we often want to see, whether there are any trends which allow predictions. Linear algebra allows to solve this problem elegantly and very generally. For example, to approximate some data points using certain type of functions, we can do that. It even would work in higher dimensions, where we wanted to see how a certain datapoint depends on two data sets.

We will see explicit examples in this course. The most used datafitting problem is probably the linear fitting, where one wants to see how certain data depend on others.

GAME THEORY Abstract Games are often represented by pay-off matrices. These matrices tell the outcome when the decisions of each player are known.

A famous example is the prisoner dilemma. Each player has the choice to corporate or to cheat.. The game is described by a 2x2 matrix like for example $\begin{pmatrix} 3 & 0 \\ 5 & 1 \end{pmatrix}$. If a player cooperates and his partner also, both get 3 points. If his partner cheats and he cooperates, he gets 5 points. If both cheat, both get 1 point. More generally, in a game with two players where each player can chose from n strategies, the pay-off matrix is a n times n matrix A. A Nash equilibrium is a vector $p \in S = \{\sum_i p_i = 1, p_i \geq 0\}$ for which $qAp \leq pAp$ for all $q \in S$.



NEURAL NETWORK In part of neural network theory, for example Hopfield networks, the state space is a $2n$-dimensional vector space. Every state of the network is given by a vector x, where each component takes the values -1 or 1. If W is a symmetric nxn matrix, one can define a "learning map" $T: x \mapsto \mathrm{sign} Wx$, where the sign is taken component wise. The energy of the state is the dot product $-(x, Wx)/2$. One is interested in fixed points of the map.

For example, if $W_{ij} = x_i y_j$, then x is a fixed point of the learning map.

USE OF LINEAR ALGEBRA (IV)

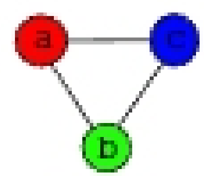
Math 21b, Oliver Knill

MARKOV. Suppose we have three bags with 10 balls each. Every time we throw a dice and a 5 shows up, we move a ball from bag 1 to bag 2, if the dice shows 1 or 2, we move a ball from bag 2 to bag 3, if 3 or 4 turns up, we move a ball from bag 3 to bag 1 and a ball from bag 3 to bag 2. What distribution of balls will we see in average?

The problem defines a Markov chain described by a matrix $\begin{bmatrix} 5/6 & 1/6 & 0 \\ 0 & 2/3 & 1/3 \\ 1/6 & 1/6 & 2/3 \end{bmatrix}$. From this matrix, the equilibrium distribution can be read off as an eigenvector of a matrix. Eigenvectors will play an important role throughout the course.

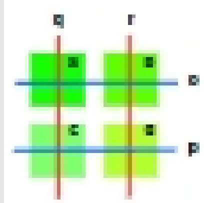
SPLINES In computer aided design (CAD) used for example to construct cars, one wants to interpolate points with smooth curves. One example: assume you want to find a curve connecting two points P and Q and the direction is given at each point. Find a cubic function $f(x, y) = ax^3 + bx^2y + cxy^2 + dy^3$ which interpolates.

If we write down the conditions, we will have to solve a system of 4 equations for four unknowns. Graphic artists (i.e. at the company "Pixar") need to have linear algebra skills also at many other topics in computer graphics.



SYMBOLIC DYNAMICS Assume that a system can be in three different states a, b, c and that transitions $a \mapsto b$, $b \mapsto a$, $b \mapsto c$, $c \mapsto c$, $c \mapsto a$ are allowed. A possible evolution of the system is then $a, b, a, b, a, c, c, c, a, b, c, a$... One calls this a description of the system with symbolic dynamics. This language is used in information theory or in dynamical systems theory.

The dynamics of the system is coded with a symbolic dynamical system. The transition matrix is $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$. Information theoretical quantities like the "entropy" can be read off from this matrix.



INVERSE PROBLEMS The reconstruction of a density function from projections along lines reduces to the solution of the Radon transform. Studied first in 1917, it is today a basic tool in applications like medical diagnosis, tokamak monitoring, in plasma physics or for astrophysical applications. The reconstruction is also called tomography. Mathematical tools developed for the solution of this problem lead to the construction of sophisticated scanners. It is important that the inversion $h = R(f) \mapsto f$ is fast, accurate, robust and requires as few datas as possible.

Toy problem: We have 4 containers with density a, b, c, d arranged in a square. We are able and measure the light absorption by by sending light through it. Like this, we get $o = a + b$, $p = c + d$, $q = a + c$ and $r = b + d$. The problem is to recover a, b, c, d. The system of equations is equivalent to $Ax = b$, with $x = (a, b, c, d)$ and $b = (o, p, q, r)$ and $A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$.