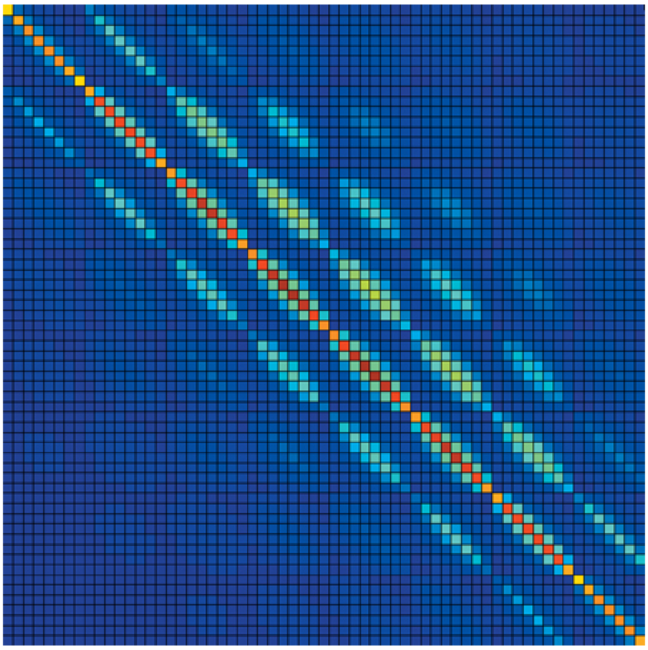
COSC 70/170 Numerical and Computational Tools for Applied Science

Winter 2012, 10A: Tuesday, Thursday 10:00 AM - 11:50 AM
Sudikoff 115
Instructor: **Kimo Johnson**



Course Description

This course provides a practical and principled coverage of useful numerical and computational tools of use in many disciplines. The first half of this course provides the mathematical (linear algebra) and computing (MATLAB) framework upon which data analysis tools are presented. These tools include data fitting, Fourier analysis, dimensionality reduction, estimation, clustering, and pattern recognition. This course is designed for undergraduate and graduate students across the Sciences and Social Sciences.

Prerequisites

This course requires programming experience as well as linear algebra and basic calculus. The following courses (or equivalent courses at other institutions) are helpful prerequisites: Computer Science 1, Engineering Sciences 20, Mathematics 8, Mathematics 22 or Mathematics 24. It is strongly recommended that all projects and assignments be completed in MATLAB. All demo code and test harnesses will be provided for MATLAB.

Textbook

There is no required textbook for this class. We will rely on papers and free book chapters. In particular, the course will use material from these sources:

- Math Refresher for Engineers and Scientists by J. Fanchi. (e-book available through library).
- Numerical Computing with MATLAB, by C. Moler (e-book available through library). This book has a lot of examples worked out in MATLAB.
- Elementary Linear Algebra by R. Larson. **Chapter 10** on Numerical Methods is available online.
- A Multigrid Tutorial, by W. Briggs (e-book available through library). This book gives a good overview of iterative methods and provides intuition for how multigrid can speed them up.
- **An Introduction to the Conjugate Gradient Method without the Agonizing Pain**, by J. Shewchuk. This paper provides a thorough overview of the conjugate gradient method, as well as basic linear algebra and optimization.
- **Least-Squares Fitting of Circles and Ellipses**, by Gander et al. This paper provides a

good introduction to nonlinear least-squares through the seemingly simple problem of fitting circles and ellipses to data.

- **Methods for Non-linear Least Squares Problems**, by Madsen et al. This book covers the Steepest Descent, Gauss-Newton, and Levenberg-Marquardt methods.
- Numerical Methods for Engineers and Scientists, by Gilat and Subramaniam. This book is suggested as a reference for students needing additional background. It is not required.
- Linear Algebra and Its Applications by G. Strang. This is a classic text on linear algebra.
- **Least Squares Optimization**, by Eero Simoncelli. Good overview of least-squares and PCA
- **Matrix Calculus**, by Carlos Felippa. Has some basic formulas for matrix and vector calculus.
- **The Spectral Modeling Toolbox**. My master's thesis explored spectral analysis, so it and the references in the back should be helpful for understanding the Fourier transform and the STFT.
- **A Tutorial on Fourier Theory** , by Y. Yoo. This seems to be a good overview of Fourier Theory.
- **A Tutorial on Principal Component Analysis**, by J. Shlens. This tutorial provides good context for PCA and also connects it to the SVD.

The recommended readings associated with each lecture are listed in the syllabus below.

Assignments and grading

The course will consist of eight problem sets, one midterm exam, and a self-chosen final project. Each assignment is worth at least the number of points specified below:

- Homework 0: 10 points
- 8 problem sets: 50 points each
- Midterm exam: 200 points
- Final project: 200 points

On each assignment, you are required to attempt at least the number of points listed above, though the assignments may have additional points available. Your final grade will be computed as the total number of points earned divided by the total number of points attempted. Letter grades will be assigned as follows: A = [90, 100], B = [80, 90[, C = [70, 80[, D = [60 - 70[. The '+' part of each range is [6.67, 10.0[, and the '-' part of each range is [0, 3.33[. For graduate grades, HP = [93.33, 100], P = [70, 93.33[, and LP = [60, 70[.

Extra credit

Some problem sets may have more than 50 points worth of problems. You are required to attempt at least 50 points worth of problems, however, you may choose to complete additional problems for extra credit. The extra credit problems will be graded using the following logic. Let X be the number of points earned and T be the total number of points for the problem.

- if $X / T < 0.5$, then $E_n = 0$ and $E_d = 0$
- if $X / T \geq 0.5$, then $E_n = 2X$ and $E_d = 2X$

The value E_n is added to the numerator of the points earned for the assignment and the value E_d is added to the points attempted (the denominator). Therefore, extra credit can only improve your score on each assignment.

Late assignments

Unless otherwise specified, assignments are due at 3:00 PM on the due day. You have two "late days" in which you can turn in an assignment 24 hours late without penalty. If you use up your late days, your grade will be decreased as follows. For every hour the assignment is late, you will lose 1 percent off of the number of points earned. For example, if your assignment is 8 hours late your points earned will be multiplied by 0.92. Late assignments must be turned in via email or in person (i.e., not left in a mailbox). Late assignments will not be accepted after the solutions have been posted.

Submitting assignments
Please follow these guidelines when submitting homeworks:

- Print the assignment, write your name at the top and write the letter 'E' next to problems that you complete in excess of 50 points. Your should complete exactly 50 points to be graded as regular credit and anything beyond that should be extra credit. If you do not mark the extra credit problems, I will choose them at random which may not work out in your favor.
- Provide a hard copy of your code with the problem number and your name in the comments section. If you only submit a digital copy, I will print a hard copy for you and charge you 2 points for printing services. Prices subject to change.
- Show that the code functions properly by submitting a hard copy of the MATLAB output. This can be done by submitting a MATLAB diary or by copying the output into comments within each function.
- For any question that asks you to plot something, provide a hard copy of the plot and be sure the axes are labeled.
- Submit code on blackboard by placing the files in a folder with your last name and zipping it. I will run the code if I question the correctness of it when reading the hard copy.

Links:

- **MATLAB Tutorial from Brown**
- **MATLAB Getting Started Guide**

Contact Info and Office Hours:

Kimo's office for the term is Sudikoff 208 and you can contact him via email at kimo[at]cs.dartmouth.edu.
Kimo's office hours will be during the x-period, Wednesday 3:00 - 3:50, unless otherwise specified. However, students may feel free to ask a question if the office door is open.

Tentative Syllabus

Class Date	Topic	Reading	Notes	Homeworks
Th, Jan 5th	Course introduction, MATLAB basics		**MATLAB code**	**HW 0**, **hw1.pdf**, **hw1.zip**
Tu, Jan 10th	Linear algebra basics	Fanchi 4	**MATLAB code**	HW 0 due
Th, Jan 12th	Linear systems, direct solutions, conditioning, iterative techniques	**Larson 10.1, 10.2**, Fanchi 5	**MATLAB code**	HW 1 due, **hw2.pdf**, **hw2.zip**
Tu,	Singular value decomposition,	Shewchuk 5,	**MATLAB**	

Jan 17th	eigenvectors, convergence of Jacobi	Fanchi 5	**code**	**HW1 solution**
Th, Jan 19th	Least-squares, PCA	Simoncelli paper	**MATLAB code**	HW 2 due, **hw3.pdf**, **hw3.zip**
Tu, Jan 24th	Tikhonov regularization, Steepest descent	Shewchuk 6, 7, 8	**MATLAB code**	**HW2 solution**
Th, Jan 26th	Conjugate gradient	Shewchuk 6, 7, 8	**MATLAB code**	HW 3 due, **hw4.pdf**, **hw4.zip**
Tu, Jan 31st	Applications of PCA and SVD	**A Tutorial on PCA**	**MATLAB code**	**HW3 solution**
Th, Feb 2nd	Fourier transforms	**Fourier transform notes**	**MATLAB code**	HW 4 due, **hw5.pdf**, **hw5.zip**
Tu, Feb 7th	Fourier transforms, contd.	**Tutorial on Fourier Theory**	**MATLAB code**	**HW4 solution**
Th, Feb 9th	Time-frequency transforms, MP3	**Spectral Modeling Toolbox, Ch. 2**	**MATLAB code**	HW 5 due, **hw6.pdf**, **hw6.zip**
Tu, Feb 14th	Taylor series, nonlinear least squares	**Madsen et al. paper**	**MATLAB code**	**HW5 solution**, **midterm.pdf**, **midterm.zip**
Th, Feb 16th	Gauss-Newton and Levenbert Marquardt	**Gander et al.**	**MATLAB code**	HW 6 due
Tu, Feb 21st	Applications of nonlinear least squares, Numerical solutions to ODEs	**C. Moler, Chapter 7**	**MATLAB code**	**HW6 solution**
Th, Feb 23rd	PDEs	A Multigrid Tutorial, Briggs, Ch. 1	**MATLAB code**	Midterm due, **hw7.pdf**, **hw7.zip**
Tu, Feb 28th	Multigrid	A Multigrid Tutorial, Briggs, Ch.	**MATLAB code**	**Midterm solution**

		2,3		
Th, Mar 1st	Fisher's linear discriminant, Cross validation		**MATLAB code**	HW 7 due, **hw8.pdf**, **hw8.zip**
Tu, Mar 6th	Robust fitting: RANSAC and IRLS		**MATLAB code**	**HW7 solution**, HW 8 due
Tu, Mar 13th	No class			Final projects due

Projects

The final project is to implement a research paper that uses a scientific computing technique, preferably a technique we have covered in class. I have compiled a list of papers and added descriptions of how they are related to the different topics in the class. However, these papers are heavily biased towards my background in computer vision, audio processing and graphics. If you want to find a project more closely related to your own interests, I am happy to help you find an appropriate paper.

- **Poisson Image Editing** Seamless cloning was introduced to the graphics community in this paper. This technique can be used to blend regions from different images without visible seams. The core idea is that a smooth interpolant between the image regions can be found by solving a PDE known as Poisson's equation. For the project, you can assume a square domain rather than an arbitrary domain.
- **Eigenfaces vs. Fisherfaces** The Eigenfaces paper showed that a low-dimensional representation of faces images could be used for identity recognition. For this project you should implement face recognition algorithms, one based on PCA (Eigenfaces) and one based on Fisher's linear discriminant (Fisherfaces). Compare the two approaches.
- **Flexible Camera Calibration** Camera calibration is the first step towards recovering accurate 3D measurements from an imaging system. This paper derives a closed-form solution and then uses this solution as an initial condition to a nonlinear least-squares optimization. For this project, you should implement both parts of the algorithm, but you do not need to handle lens distortion or feature point detection.
- **Independent Component Analysis by Minimization of Mutual Information** We saw that principal component analysis could decorrelate a set of signals through a linear change of basis. Independent component analysis seeks to make the signals statistically independent. One application of this algorithm is to the cocktail party problem: N sounds are linearly combined and there are N distinct recordings of this combination. The goal is to unmix the sounds.
- **Separating Reflections from Images using Independent Component Analysis** Independent component analysis can also be used to separate mixed images, such as the view through a window and the reflection of the viewer off of the glass. This paper describes a straightforward solution to ICA in this context.
- **Improved Phase Vocoder Time-Scale Modification of Audio** The phase vocoder can stretch or shrink the duration of a sound without changing the frequency. A basic phase vocoder can be implemented using a short-time Fourier transform. For this project, you should write the forward and inverse short-time Fourier transforms and then resample the spectrogram to modify time without changing frequency. It is not necessary to implement the method in this paper; a simpler, but working, algorithm will be sufficient.
- **Analytic PCA construction for theoretical analysis of lighting variability** This

paper considered the problem of finding a low-dimensional representation of lighting variability on a diffuse object. They used PCA and found that even for arbitrarily complex lighting environments, the observed shading could be well-approximated by a low-dimensional model. For this project, you should recreate their result by generating a large set of shaded images of an object, perform PCA and plot the relationship between captured variance and dimensionality.

- **Orthogonal Distance Fitting of Implicit Curves and Surfaces** This paper describes a least-squares approach to fitting implicit models to point clouds. For this project it is sufficient to fit a sphere and an ellipsoid.
- **A Multigrid Tutorial** (e-book on blackboard) Multigrid is a method for solving boundary value problems through a multi-scale iterative approach. The convergence is significantly faster than standard iterative approaches on large domains. For this project, you should implement a multigrid Poission solver on a square domain and compare the convergence of your multigrid implementation to a standard Gauss-Seidel implementation.
- **Shape and Motion from Image Streams under Orthography** This paper shows how matrix factorization, in particular the SVD, can be used to recover camera view directions as well as object shape from an image sequence. I can provide simulated data for this project.
- **The Anatomy of a Large-Scale Hypertextual Web Search Engine** The PageRank algorithm is based on the idea that all web pages can be represented as nodes in a graph with hyperlinks between web pages as edges in the graph. If the graph is represented as a matrix, with appropriate assumptions, PageRank ends up being the principal eigenvector of the matrix. For this project, you should work with simulated data to learn how PageRank works.
- **Document Clustering Based on Non-negative Matrix Factorization** Non-negative matrix factorization is a popular technique for factorizing data into non-negative components. This factorization often leads to more intuitive components since natural data is often constrained to only have positive values (e.g., light). This paper considers NMF applied to document clustering.

Academic integrity

You may discuss the assignments with other current COSC 70/170 students, but your homework submission must be entirely your own work. That is, your code and any other solutions you submit must be created, written/typed, and documented by you alone. You may not copy anything directly from another student's work. For example, memorizing or copying onto paper a portion of someone else's solution would violate the honor policy, even if you eventually turn in a different answer. Similarly, e-mailing a portion of your code to another student, or posting it on-line for them to see would violate the honor code. I do encourage discussion of assignments between students, subject to these rules. You cannot make use of any code taken from outside references for your homework. You should treat any external code as software written by another COSC 70/170 student: you are not allowed to copy it or to use it as a template to implement your solution. You cannot collaborate or copy in any way on exams.

Acknowledgements

The materials from this class rely significantly on materials prepared by other instructors, especially Hany Farid and Lorenzo Torresani.

Previous Versions of Course

Lorenzo Torresani's most recent offering of COSC 170 (formerly COSC 136) can be found **here**.

Comments, questions to **Kimo Johnson**.