Matrix Methods in Signal Processing ...
(Lecture notes for EECS 551)

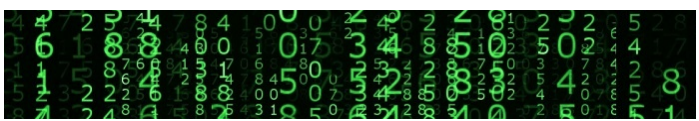
Jeff Fessler
University of Michigan

August 24, 2021

Chapter 0

EECS 551 Course introduction

Contents (class version)



These lecture notes initially were based extensively on Prof. Raj Nadakuditi's hand-written notes. I am grateful to him for sharing his course materials. I also thank former GSIs David Hong, Steven Whitaker, and Caroline Crockett, and numerous EECS 551 students for many corrections and suggestions.

These notes were typeset using LaTeX. One way to learn LaTeX is to use `http://overleaf.com`.
All times (hours, due dates etc.) for this course are in **Eastern time**.

0.1 Course logistics

EECS 551: Matrix Methods For Signal Processing, Data Analysis & Machine Learning 4 credits

Lecture	001	Tue,Thu 9-10:30AM	FMCRB 1060
Discussion	012	Fri. 9:30-10:30 AM	1311 EECS
Discussion	011	Fri. 10:30-11:30 AM	1571 GGBL

Instructor: Prof. Jeff Fessler fessler@umich.edu https://web.eecs.umich.edu/~fessler/

Office hours: Wed 4:00-5:30 PM (Zoom available)
Thu 3:00-3:50 PM (SIPML advising has priority!)

Include [eecs551-f21] in email subject for less slow response. Use Canvas/Piazza when possible!

GSI (office hours held in 3312 EECS):

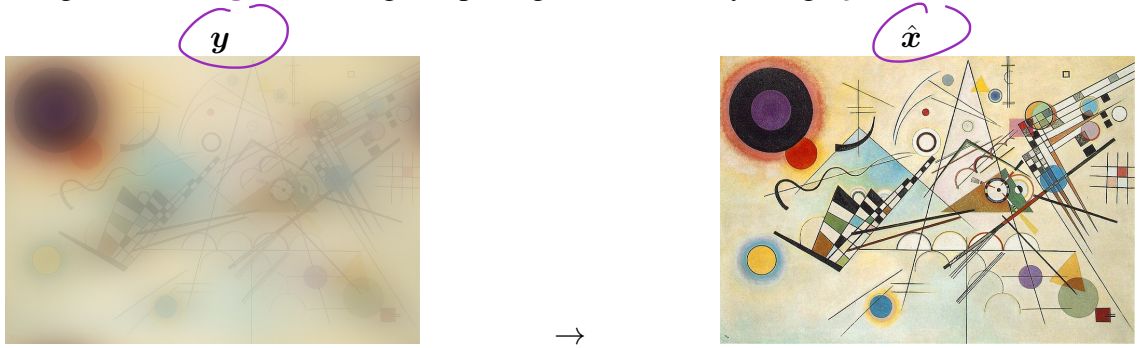
- Eric Cheek echeek@umich.edu https://www.linkedin.com/in/eric-cheek-a799b0a2
  Mon 11AM-Noon Tue 2:50-3:50PM (Zoom) Thu 11:00AM-Noon
- Zongyu Li zonyul@umich.edu https://zongyuli.com/umich
  Tue 10:30-11:30AM Wed 10-11AM (Zoom) Thu 2:00-3:00PM

Course materials: **Action**: bookmark these links.

- Primary site is Canvas: https://umich.instructure.com/courses/461718
  (homework, solutions, lecture notes, announcements, etc.) **Action**: enroll in Piazza on Canvas
- Annotated versions of class notes: https://tinyurl.com/f21-551-lecture
- Secondary site (demos, back-ups): http://web.eecs.umich.edu/~fessler/course/551

Course goal: provide *mathematical foundations* for subsequent signal processing and machine learning courses, while also introducing matrix-based SP/ML *methods/applications* that are useful in their own right.

Example. Image **deblurring** (recovering sharp image $\hat{\boldsymbol{x}}$ from blurry image $\boldsymbol{y}$):



https://commons.wikimedia.org/wiki/File:Vassily_Kandinsky,_1923_-_Composition_8,_huile_sur_toile,_140_cm_x_201_cm,_Mus%C3%A9e_Guggenheim,_New_York.jpg

Model: $\underbrace{\boldsymbol{y}}_{\mathbb{R}^{800 \cdot 554}} = \boldsymbol{A} \underbrace{\boldsymbol{x}}_{\mathbb{R}^{800 \cdot 554}} + \boldsymbol{\varepsilon}$ where $\boldsymbol{A}$ is a matrix of size 443200×443200 describing the optical blur.

Solution (implemented efficiently using FFT operations):

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{R}^{800 \cdot 554}}{\arg\min} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 = \boldsymbol{A}^+ \boldsymbol{y} = \boldsymbol{V} \boldsymbol{\Sigma}^+ \boldsymbol{U}' \boldsymbol{y}.$$

Prerequisites

DSP (*i.e.*, EECS 351, formerly EECS 451) or graduate standing.
Prof. Nadakuditi's EECS 505 perhaps relies less on DSP background. See 505-551 FAQ.

Exam / assessment (other ECE exams)

Midterm (Ch1-5, HW1-6): Mon. Oct. 25 6–8PM, Chrysler 220
Final (cumulative): Mon. Dec. 20 10:30 AM – 12:30 PM, Room TBA

Notes.

- Exams may also include a take-home portion - TBD.
- No HW the week of Midterm.
- Exams may include a mix of formats
  - hand-written problems (scanned and graded on gradescope)
  - multiple-choice questions on paper and/or on Canvas (like weekly quizzes),
  - autograded computing problems (like HW, but possibly limited tries and all-or-nothing scoring)
- No exceptions to this assessment schedule, except for documented medical emergencies.

Grades

Homework and task sheets	15%	(drop lowest HW, but not autograder problems!)
Autograder	5%	
Clicker	5%	DROP 2 LOWEST
Canvas practice quizzes	5%	(drop 2 lowest) (must attempt by deadline to view later!)
Midterm	30%	
Final exam	40%	
Total	100%	

Final grade cutoffs for A/B/C will be at most 90/80/70%, but often end up lower.
Exam scores may be standardized. Grade history.

Religious/Cultural Observance

Students who have religious or cultural observances that coincide with this class or any deadlines should let the instructor know by email by September 24. I encourage you to honor your cultural and religious holidays. If I do not hear from you by September 24, I will assume that you have no conflicts with any of the class activities.

Homework

Typically due on Thursday at 11PM. Typically an automatic extension to Friday at 11PM. No further.
Submit scans of solutions to `https://gradescope.com`. (HW1 on Canvas!)
Hopefully will be graded and "returned" via gradescope within a week.
Written regrade requests via gradescope within 3 days of return date.

Actions:

- Check for your name on gradescope (should be there thanks to Canvas integration)
- Review gradescope scan/pdf submission process. There are also video instructions.

Collaboration policy: Homework assignments are to be completed on your own. You are allowed to consult with other students (and instructors) during the conceptualization of a solution, but all written work, whether in scrap or final form, is to be generated by you, working alone. Also, you are not allowed to use, or in any way derive advantage from, the existence of solutions prepared in prior years. Violation of this policy is an honor code violation. If you have questions about this policy, please contact me. While collaboration can sometimes be helpful to learning, if overused, it can inhibit the development of your problem solving skills.

Ethics

Sharing any materials from this class with other individuals not in the class without written instructor permission will be treated as an Honor Code violation. Posting your own solutions (including code) on public sites like github.com is also prohibited. Keep your materials private! In particular, uploading any materials from this class to web sites akin to coursehero.com will be reported to the Honor Council.

Homework grading

Homework grading is constrained by GEO union policies. See:
http://web.eecs.umich.edu/~fessler/course/551/r/grading,geo.txt
http://web.eecs.umich.edu/~fessler/course/551/r/grader-duties.pdf

Most manually graded problems will be on a scale of 0-3:

0 No solution was attempted
1 A solution was attempted but the approach used did not recognizably conform to any in the solution set
2 The approach used recognizably conformed to one in the solution set, but the answer was incorrect.
3 A solution approach recognizably conformed to one in the solution set, and the answer was correct.

JULIA-based autograder problems (details on HW1) typically will be 10 points each (10 or 0).
Your code must pass *all* autograder tests to earn the 10 points.

Honor code

The UM College of Engineering Honor Code applies. See collaboration policies above.
See https://ecas.engin.umich.edu/honor-council/ for details.

Covid

Policies around academic and public health are subject to change as this pandemic evolves. This course will follow all policies issued by the University, as documented on the Campus Blueprint's FAQ. These policies may change over the course of the term.

Canvas practice quizzes

Starting in the second week of the course, there will be short quizzes (typically 4-6 questions) on Canvas that are due by 9AM on Tue. and Thu. The quizzes will become available 24 hours before each class. These are designed to be quick checks of your understanding of the material covered up to that point. You have 10 minutes to complete the quiz. With about 21 quizzes of about 5 points each, each quiz question counts for a minuscule $5\%/21/5 \approx 0.048\%$ of your final grade, so their main purpose is learning, not assessment.
Thus, Canvas shows you the answers right after you take it.
Post concerns about any quiz question to Piazza *after* the Quiz due date.
Apparently Canvas only allows a student to view a past quiz (*e.g.*, for exam review) that they attempted!

Missing class

- Classes are captured/recorded and viewable on Canvas.
  URL: `https://leccap.engin.umich.edu/leccap/site/s2rwr7oipf3nicudb2u`
  - *Class time will be audio/video recorded and made available to other students in this course. When participating in this course, you may be recorded. If you do not wish to be recorded, please use Piazza to post questions instead of asking live during class.*
  - *Students may not record or distribute any class activity without written permission from the instructor, except as necessary as part of approved accommodations for students with disabilities. Any approved recordings may only be used for the student's own private use.*
- Missed clicker questions and Canvas practice quizzes cannot be made up (but lowest two ~~quizzes~~ dropped).
- Annotated notes are available online; see p. 0.2.
- Daily topic list: `http://web.eecs.umich.edu/~fessler/course/551` (topics)

Class climate

This class is a place where you will be treated with respect, and I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability – and other visible and nonvisible differences. I expect all class members to contribute to a respectful, welcoming and inclusive environment for everyone. I am dedicated to helping each of you achieve all that you can in this class. I may, either in lecture or smaller interactions, accidentally use language that creates offense or discomfort. Should I do this, I invite you to contact me and help me understand and avoid making the same mistake again. Anonymous feedback is also welcome. Please contact me if other members of the teaching staff or fellow students detract from our class climate.

I will also make appropriate accommodations for students with disabilities;
see http://ssd.umich.edu.

Student well-being

Students may experience stressors that can impact both their academic experience and their personal well-being. These may include academic pressure and challenges associated with relationships, mental health, alcohol or other drugs, identities, finances, etc.

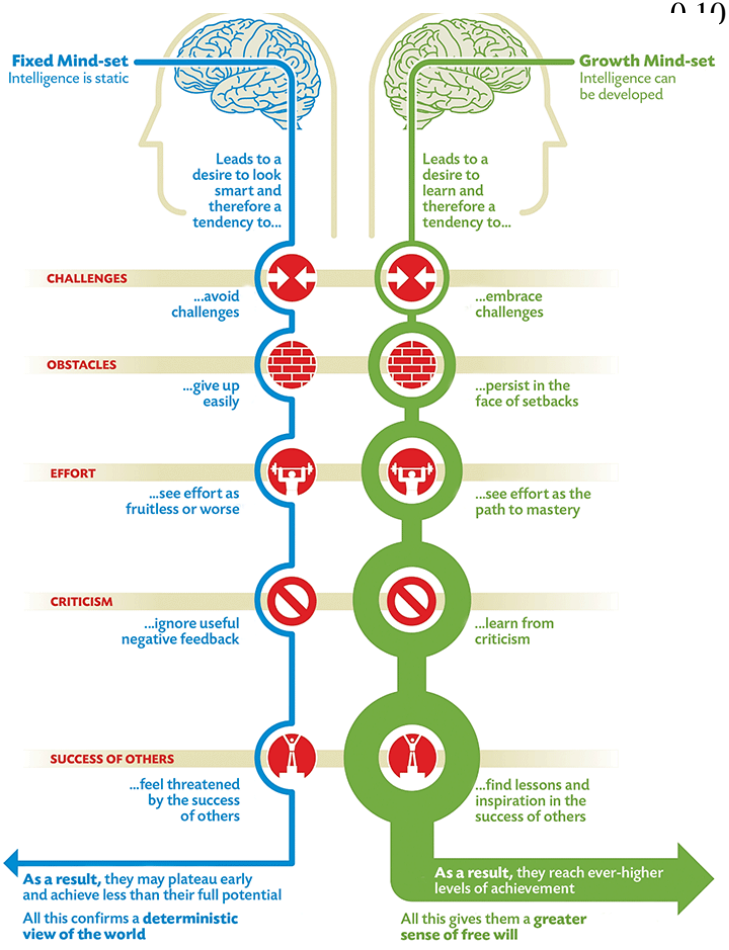
If you are experiencing concerns, seeking help is a courageous thing to do for yourself and those who care about you. If the source of your stressors is academic, please contact me to discuss possible solutions together. For personal concerns, U-M offers many resources, some of which are listed at Resources for Student Well-being on the Well-being for U-M Students website. See additional resources on that website.

Aim for a growth mindset

Figure by Carol S. Dweck, Ph.D.
http://www.nigelholmes.com/gallery

Key attributes:

- Embrace challenges
- Persist in face of setbacks
- See effort as the path to mastery
- Learn from criticism
- Find lessons and inspiration in the success of others



Books and other resources

Action: Decide whether to buy reference textbook [1]:
Laub, 2005; *Matrix analysis for scientists and engineers.*
Probably not worth buying!

About $50 at http://bookstore.siam.org/ot91 - 30% member discount.
Student membership is free: https://siam.org/students/memberships.php
(Select "University of Michigan" not "UM Ann Arbor" as the Academic Member Institution.")

Books that are useful references: [2] [3] [4] [5].

Particularly relevant *online* books:

- Lars Eldén, "Matrix methods in data mining and pattern recognition" [6]
- K B Petersen & M S Pedersen, "The matrix cookbook" [7]
- Free online book about JULIA (version 1.0, generally sufficient):
  https://benlauwens.github.io/ThinkJulia.jl/latest/book.html

Other online books: [8, 9]. The online book [9] has a very useful JULIA companion.

MOOC on advanced linear algebra:
http://www.cs.utexas.edu/users/flame/laff/alaff/ALAFF.html

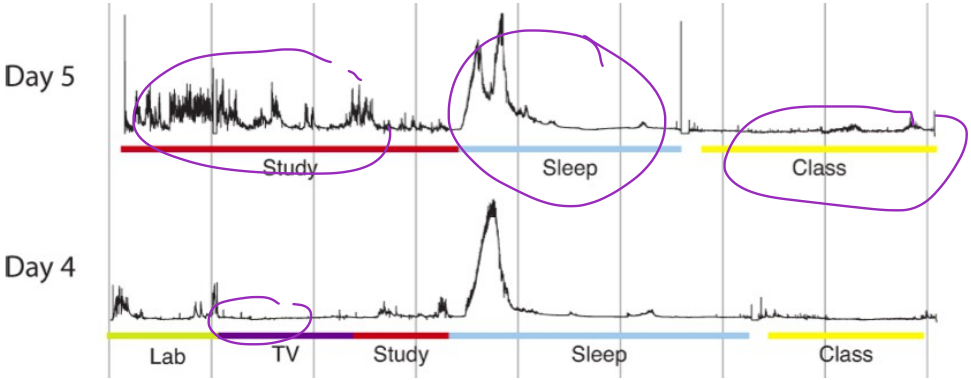
Khan Academy: http://www.khanacademy.org/math/linear-algebra

Clickers

Action: Use the "iClicker Sync" link on Canvas sidebar to sign up to use iClicker Cloud during the class. You will need to bring a device (phone, tablet, laptop) with (free) iClicker app on it to every class.

Clicker question scoring: 2 points for answering, 3 points for correct answer. (Learning, not assessment.)

Why? From M Poh, M Swenson, R Picard: "A wearable sensor for unobtrusive, long-term assessment of electrodermal activity." IEEE Tr. on Biomed. Engin., 57(5):1243-52, May 2010. [10]



Research shows that **active learning** is more effective even though students may not realize it.

1. Have you use something like clickers in a class before?
A: Never B: A bit C: A lot

PDF lecture note features (Read)

These notes highlight some important **term**s in red.
Many important terms have links in the pdf documents to **Wikipedia** in violet. Some links look like: [wiki]
Those links are clickable in the pdf and should cause your browser to open at the appropriate url.

Define. Key definitions are shaded like this.

Particularly important topics are shaded like this.

JULIA `code is shaded like this.` **References to JULIA manual look like this.**

Boxes with this color need completion during class.

A road hazard or **"dangerous bend" symbol** in the margin warns of tricky material.

A double diamond symbol is "experts only" material included for reference that is likely beyond the scope of EECS 551 exams. ♦♦

These notes are not a textbook; they are designed for classroom use. My goal is that every other page or so (except in this part) should have something more interactive than just text, such as a figure or a clicker question or a JULIA code snippet or some incomplete equation(s) that students must complete during class.

These notes are formatted with 16:10 aspect ratio to match the projector in the lecture room; that format is well-suited for printing two slides per paper side. If you print, please save paper by using that option.
Action: Print the next chapter (not this one) or bring pdf to class on a suitable device for annotating.

0.2 Julia language

All code examples and homework code templates will use the **Julia programming language**.

Why?

- It is free; you can download from `https://julialang.org`
- It is a real programming language, developed for numerical computing [11].
- Prof. Nadakuditi and I have used it since W17 for multiple courses at UM and MIT (505, 551, 559...)
- Used by Freshman in Robotics 101 at UM, in ENGR108 at Stanford, and in similar UCLA course.
- Interactive (like Python and MATLAB), yet fast execution because it is compiled.
- Much of its syntax is like MATLAB, so much easier for me to learn and use than python.
  Here is a **list of noteworthy differences from MATLAB**
- python has known downsides like speed, scope, ...
- DSP / data-science / machine learning are all done with many software languages...
- Jupyter notebooks (based on IPython) are educational, integrating math with documented code and figures.

Some documentation / books:

- Online documentation: (stable version) **(latest version)**
- Wikibook Intro to JULIA: https://en.wikibooks.org/wiki/Introducing_Julia
- Textbook: https://benlauwens.github.io/ThinkJulia.jl/latest/book.html
- Introductory book written by a 15-year old: Tanmay Teaches Julia for Kids and Beginners
- Julia for Pythonistas jupyter notebook at google's colab
- Fundamentals of Numerical Computation by Driscoll & Braun, now in JULIA
- Udemy courses https://www.udemy.com/topic/julia-programming-language/
- Online "Getting started with JULIA" book: https://search.lib.umich.edu/catalog/record/013714926
- Cheatsheet: https://juliadocs.github.io/Julia-Cheat-Sheet/
- Cheatsheet: https://cheatsheets.quantecon.org/julia-cheatsheet.html
- Cheatsheet: http://math.mit.edu/~stevenj/Julia-cheatsheet.pdf
- YouTube intro video: https://www.youtube.com/watch?v=puMIBYthsjQ

News articles about business uses:

- Forbes magazine article
- InfoWorld comparison of JULIA and Python
- Nature article about JULIA [12]
- Medium article: "Bye-bye Python. Hello Julia!"

Sponsors of Juliacon 2018 and Juliacon 2019:

- Using JULIA on Google's Colab with GPU support
- Python's mutable default argument gotcha

A brief comparison of three interactive languages

Operation	MATLAB	JULIA	Python `import numpy as np`
Dot product	`dot(x,y)`	`dot(x,y)`	`np.dot(x,y)`
Matrix mult.	`A * B`	`A * B`	`A @ B`
Element-wise	`A .* B`	`A .* B`	`A * B`
Scaling	`3 * A`	`3A or 3*A`	`3 * A`
Matrix power	`A^2`	`A^2`	`np.linalg.matrix_power(A,2)`
Element-wise	`A.^2`	`A.^2`	`A**2`
Inverse	`inv(A)`	`inv(A)`	`np.linalg.inv(A)`
Inverse	`A^(-1)`	`A^(-1)`	`np.linalg.inv(A)`
Indexing	`A(i,j)`	`A[i,j]`	`A[i,j]`
Range	`1:9`	`1:9`	`np.arange(1,9,1)`
Range	`linspace(0,4,9)`	`LinRange(0,4,9)`	`np.arange(0,4.01,0.5)`
Strings	`'text'`	`"text"`	`(either)`
Inline func.	`f = @(x,y) x+y`	`f = (x,y) -> x+y`	`f = lambda x,y : x+y`
Increment	`A = A + B`	`A += B`	`A += B`
Herm. transp.	`A'`	`A'`	`A.conj().T`
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	`[1 2; 3 4]`	`[1 2; 3 4]`	`np.array([[1, 2], [3, 4]])`

The JULIA column assumes you have typed: `using LinearAlgebra` for using the `dot` function.
See https://cheatsheets.quantecon.org for more.

JULIA logistics

- Software parts of homework solutions (JULIA code) will be autograded, with *unlimited* tries.
- More details in Discussion section for HW
- Some tutorials (and cautionary notes for MATLAB users):
  https://web.eecs.umich.edu/~fessler/course/551/julia/tutor/
  - 02-vector : vector/matrix operations and "**call by reference**" aspect of JULIA
  - 04-slice : matrix indexing (slicing)
  - sum-simd.html : acceleration using `@simd`
- You will use JULIA in many discussion sections (starting the 1st week!) so you will need a computer with JULIA installed.
- Editors (see list at JULIA web site: https://julialang.org)
  - VScode with JULIA integration - recommended!
  - Juno: JULIA IDE http://junolab.org
  - Atom: https://atom.io
  - vim: https://www.vim.org
- You must use JULIA version $\geq$ 1.6.2 for F21!
  Many HW/Exam problems use random numbers and to match the solutions and earn full credit you must use this version. The latest version also is faster than older versions.
  Beware of online Q/A for older versions of JULIA!
- **Actions:** Bring laptop to Discussion section Friday.
  Install Chrome browser shortcut for fast access to the manual

One professor's software language history

year	name	still using?
1977	BASIC	
1980	FORTRAN	
1981	Z80 assembly	
1982	APL	
1983	PASCAL	
1983	C	Y?
1985	LISP	
1986	CSH	Y
1988	Matlab	Y?
2000	Perl	
2004	Python	
2010	CUDA	Y?
2017	Julia	Y

JULIA has a **machine-learning** library called Flux [13].
It also has an interface to Tensorflow if you prefer that.
Another "ML in JULIA" package is MLJ.
There is also a CUDA interface for GPU programming [14].

MATLAB is "free" (*i.e.*, included) for UM students.

JULIA: getting started (Read)

For F21, we recommend (but do not require) that you use the JULIA extension with VSCode: https://www.julia-vscode.org, because of its excellent integration with the Julia Debugger. There is a video about it here: https://www.youtube.com/watch?v=rQ7D1lXt3GM. Alternatively, you may use your own favorite editor (though you may find debugging more challenging).

Actions:

- Follow installation instructions at https://www.julia-vscode.org/docs/dev/gettingstarted
  - Install JULIA: at least version v1.6.2.
  - Optional power-user tip: set up your shell to make sure the `julia` executable is in your path. On my Mac, I made a link in `/usr/local/bin` that points to the executable: `[path-to-julia]/Contents/Resources/julia/bin/julia`
  - Install VSCode
  - Install its JULIA extension ("Julia Language Support") per instructions linked above
  - Peruse VSCode tutorials: https://code.visualstudio.com/docs
  - Customize (if you want), *e.g.*, by installing your favorite keymap. (I use vim – retro...)
- Launch JULIA, either within VSCode or separately.
- You should see JULIA's **REPL** prompt: `julia>`
- Type `1+2` to verify it works.

- Press the `?` key and then type a command like `LinRange` to get documentation about it.
- Experiment with some basic commands like `x=3` and `x+2` and `x^2`
- Press the `]` key to enter the **package manager** that has a prompt like `(v1.6) pkg>`
  Add some useful packages using the package manager:
  - `add Plots`
  - `add Polynomials`
  - `add FFTW`
  - `add IJulia`
  - `add SparseArrays`
  - Press the delete key to exit the package manager and return to JULIA's main REPL.
- At the JULIA prompt, try launching a Jupyter notebook:
  `using IJulia; notebook()` (Could be slow the first time as it gets compiled.)
  For help with Juypter, see https://github.com/JuliaLang/IJulia.jl
  e.g., you might prefer `notebook(detached=true)`
  or `notebook(detached=true, dir="/some/path")`
- Experiment with the Jupyter notebook, and peruse some online resources.
- For documentation of the `Plots.jl` plotting package, see:
  http://docs.juliaplots.org/latest/
  https://github.com/sswatson/cheatsheets/blob/master/plotsjl-cheatsheet.pdf
- Even if none of the above works for you, try the "Julia101" tutorial with link on Canvas.

Clicker survey questions

2. How are you feeling about JULIA? (Pick the answer that is your strongest feeling.)
   - A: Anticipate it will be useful
   - B: Bothered about learning something new
   - C: Concerned about my software skills
   - D: Indifferent (or none of the others apply)
   - E: Excited to be on the cutting edge of numerical computing

3. Prior software experience?
   - A: Not Matlab, but any of C or C++ or Python
   - B: Matlab only
   - C: Matlab and (Python | C | C++)
   - D: JULIA and (Matlab | Python | C | C++)
   - E: None of the above

4. Office hours (do not answer if your schedule is still uncertain)
   - A: Wed 8:30-9:30AM works for me, but not Thu.
   - B: Thu 10:30-11:30 works for me, but not Wed.
   - C: Both Wed and Thu work
   - D: Neither Wed nor Thu work for me, but some GSI hour(s) work
   - E: None of the Prof. or GSI office hours work for me

0.3 Course topics

Here are some likely course topics in approximate chronological order, along with sections from [1]. (Read)

1. Introduction to matrices

Topic (Laub §)
Julia
vector (1.1)
matrix (1.1)
linear operation
DFT
system of linear equations
convolution
LTI
term-document matrix
diagonal
upper triangular
Gaussian elimination
lower triangular
Cholesky decomposition
tridiagonal
Finite difference
pentadiagonal
Gram matrix
upper Hessenberg
eigenvalue algorithms
lower Hessenberg
rectangular diagonal
dense matrix
sparse matrix
Toeplitz
Circulant
block matrix
block diagonal matrix
transpose
Hermitian transpose
symmetric
Hermitian
linear algebra
dot product
inner product
outer product
rank 1 matrix
matrix multiplication (1.2)
orthogonal (1.3)
orthonormal
norm
orthogonal matrix
unitary matrix
Parseval's theorem
determinant (1.4)
Gram matrix
invertible matrix
Laplace's formula
eigenvalues (9.1)
characteristic polynomial
fundamental theorem of algebra
Gram matrix
covariance matrix
singular matrix
scatter matrix
trace (1.1)
field (2.1)
vector space (2.1)
linear transform (3.1)

2. Matrix factorizations / SVD

eigenvalues (10.1)
spectral theorem
eigenvectors
orthogonal
orthonormal
eigendecomposition
diagonalizable
SVD (5.1)
spectral norm (7.4)
MIMO channels
beamforming
positive definite (10.2)
positive semidefinite

3. Subspaces and rank

dimensionality reduction
subspace (2.2)
periodic functions
span (2.3)
linear combinations
linearly independent (2.3)
linearly dependent
monomials
basis (2.3)
discrete cosine transform
wavelet transform
coordinate system
additive synthesis
basis (2.3)
subspace sum (2.4)
intersection
direct sum
orth. complement (3.4)
linear map (3.1)
range (3.4)
column space
row space
rank (3.5)
null space (3.4)
kernel
nullity
fundamental theorem of linear algebra (3.5)
orthogonal basis

4. Linear equations and least-squares

linear equations (6.1)
linear least-squares (8.1)
residual
orthogonal polynomials
convex function
normal equations (8.1)
SVD (8.4)
over-determined system
Moore-Penrose
pseudoinverse (4.1)
left inverse (4.3)
right inverse
rectangular diagonal
SVD (5.2)
QR decomposition (8.5)
under-determined
orthogonality principle
error
linear variety
flat
idempotent matrix
minimum norm sol. (8.1)
compressed sensing
truncated SVD
floating-point precision
condition number
low-rank approx. (8.1)
Tikhonov regularization
regularization parameter
conjugate gradient

5. Norms

vector norm (7.3)
Euclidean norm
triangle inequality
Parseval's theorem
inner product spaces (7.2)
inner product
Frobenius inner product
parallelogram law
Cauchy-Schwarz ineq.
angle between subspaces
correlation coefficient
matrix norms (7.4)
sub-multiplicative
Frobenius norm
induced norm
Schatten p-norm
norm equivalence
unitarily invariant norms
Procrustes problem
polar decomposition
idempotent matrix

6. Low-rank approximation

dimensionality reduction
low-rank approximation
factor analysis
scree plot
Eckart-Young-Mirsky theorem
Unitary invariance
PCA
multidimensional scaling
geodesic distances
Heaviside step function
Stein's unbiased risk
estimate
weakly differentiable
Divergence
OptShrink
matrix completion
Netflix problem
ISTA
proximal gradient method
proximity operation

7. Special matrices

monic polynomial
companion matrix
minimum polynomial
Vandermonde matrix
Kronecker sum
circulant matrix
DFT
fast Fourier transform
power iteration
positive matrix
nonnegative matrix
primitive matrix
Geršgorin disk theorem
Perron-Frobenius theorem
Gelfand's formula
multiplicity one
algebraic multiplicity
geometric multiplicity
irreducible matrices
Markov chain
directed graph
transition matrix
stochastic eigenvector
law of total probability
irreducible matrix
simple eigenvalue
strongly connected graph
Google's PageRank

8. Optimization basics

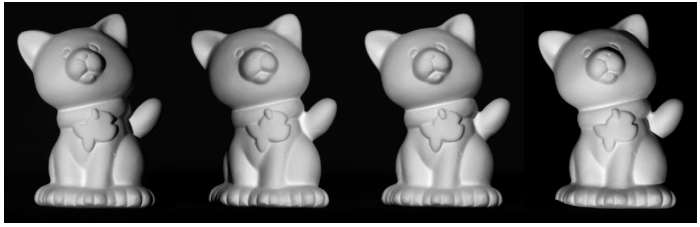
optimization
convergence
(matrix) square root
principal square root
matrix powers
positive semidefinite
positive definite
gradient descent
Lipschitz continuity
Taylor's theorem
logistic regression
Hessian matrix

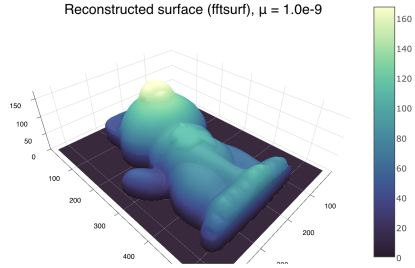
9. Matrix completion

matrix completion
ill posed
Netflix problem
latent variable
degrees of freedom
sampling bounds
polylog
Hadamard product
NP hard
projection onto convex set
majorize-minimize (MM)
ISTA
proximal gradient method
proximity operation

A coarse goal is for students to be familiar with *all* of these topics and be able to formulate and solve related problems. These topics are relevant to *many* applications in signal processing and machine learning, some of which are explored in HW and discussion section.

Example. **photometric stereo** (shape from shading):





Example. Hand-written digit recognition

Bibliography

[1] A. J. Laub. *Matrix analysis for scientists and engineers*. Soc. Indust. Appl. Math., 2005 (cit. on pp. 0.11, 0.23).

[2] T. K. Moon and W. C. Stirling. *Mathematical methods and algorithms for signal processing*. Prentice-Hall, 2000 (cit. on p. 0.11).

[3] G. H. Golub and C. F. Van Loan. *Matrix computations*. 2nd ed. Johns Hopkins Univ. Press, 1989 (cit. on p. 0.11).

[4] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge: Cambridge Univ. Press, 1985 (cit. on p. 0.11).

[5] G. Strang. *Linear algebra and learning from data*. Cambridge, 2019 (cit. on p. 0.11).

[6] L. Eldén. *Matrix methods in data mining and pattern recognition*. Errata: http://users.mai.liu.se/larel04/matrix-methods/index.html http://users.mai.liu.se/larel04/matrix-methods/. Soc. Indust. Appl. Math., 2007 (cit. on p. 0.11).

[7] K. B. Petersen and M. S. Pedersen. *The matrix cookbook*. ., 2012 (cit. on p. 0.11).

[8] T. Denton and A. Waldron. *Linear algebra in twenty five lectures*. 2012 (cit. on p. 0.11).

[9] S. Boyd and L. Vandenberghe. *Introduction to applied linear algebra: Vectors, matrices, and least squares*. ., 2017 (cit. on p. 0.11).

[10] M-Z. Poh, N. C. Swenson, and R. W. Picard. "A wearable sensor for unobtrusive, long-term assessment of electrodermal activity". In: *IEEE Trans. Biomed. Engin.* 57.5 (May 2010), 1243–52 (cit. on p. 0.12).

[11] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A fresh approach to numerical computing". In: *SIAM Review* 59.1 (2017), 65–98 (cit. on p. 0.14).

[12] J. M. Perkel. "Julia: come for the syntax, stay for the speed". In: *Nature* 572 (2019), 141–2 (cit. on p. 0.16).

[13] M. Innes. "Flux: Elegant machine learning with Julia". In: *J. of Open Source Software* 3.25 (2018), p. 602 (cit. on p. 0.19).

[14] T. Besard, C. Foket, and B. De Sutter. "Effective extensible programming: unleashing Julia on GPUs". In: *IEEE Trans. Parallel. Dist. Sys.* 30.4 (Apr. 2019), 827–41 (cit. on p. 0.19).