# Matrix Methods in Signal Processing ...

## (Lecture notes for EECS 551)

Jeff Fessler
University of Michigan

August 27, 2019

# Chapter 0

# EECS 551 Course introduction: F19

## 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 and Steven Whitaker and 551 students in F17 and F18 for many corrections to earlier versions.

These notes were typeset using LaTeX. One way to learn LaTeX is to use http://overleaf.com.

## 0.1 Course logistics

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

Lecture: Tue,Thu 9-10:30AM, 1500 EECS

| | | | |
|---|---|---|---|
| Discussion: | 012 | Fri. 9:30-10:30 AM, | 1003 EECS |
| | 011 | Fri. 10:30-11:30 AM, | 1500 EECS |

Instructor: Prof. Jeff Fessler fessler@umich.edu `https://web.eecs.umich.edu/~fessler/`
Office hours: Wed,Thu 10:30-11:30AM (often until 11:45AM), 4431 EECS.
Include `[eecs551-f19]` in email subject for less slow response. Use Canvas/Piazza when possible.

GSI (office hours held in 3312 EECS):

- Caroline Crockett cecroc@umich.edu `http://web.eecs.umich.edu/~cecroc`
  Mon 4-5PM Tue 10:30-12PM Thu 2-3PM

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

- Primary site is Canvas: `https://umich.instructure.com/courses/310523`
  (homework, solutions, lecture notes, announcements, etc.)
- Annotated versions of class notes: `https://tinyurl.com/f19-551-lecture`
- Secondary site (demos, back-ups): `http://web.eecs.umich.edu/~fessler/course/551`

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

## Prerequisites

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

## Exams (other ECE exams)

Midterm Exam 1: Mon. Oct. 21, 6-8:00 PM, 1303&1500 EECS (more feedback,
Midterm Exam 2: Mon. Nov. 25, 6-8:00 PM, 1303&1500 EECS less stress per exam)
Final Exam: Wed. Dec 18, 1:30-3:30 PM, Room TBA

## Grades

| | | |
|---|---|---|
| Homework and task sheets | 20% | |
| Clicker | 5% | |
| Canvas quizzes | 0% | (must attempt by deadline to view later!) |
| Midterm exam 1 | 20% | |
| Midterm exam 2 | 25% | |
| Final exam | 30% | |

Final grade cutoffs will be 90/80/70% or lower. Exam scores may be standardized. Grade history.

## Honor code

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

## Homework

Typically due on Thursday at 4PM. Typically an automatic extension to Friday at 4PM. 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

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 auto-graded problems (details on HW1) typically will be 10 points each (10 or 0).

## 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. 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 a Canvas Discussion *after* the Quiz due date. The instructor quiz interface on Canvas is horrid, so quizzes and lectures can get out of sync. That is why quizzes are worth 0%.
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.
- Missed clicker questions and quizzes cannot be made up.
- Annotated notes are available online; see p. 0.2.
- Daily topic list: `http://web.eecs.umich.edu/~fessler/course/551` (topics)

## Books and other resources

**Action:** Decide whether to buy reference textbook [1]:
Laub, 2005; *Matrix analysis for scientists and engineers.*
Should be on reserve at UM Engineering Library

$52 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] Online books: [6–8].
Khan Academy: `http://www.khanacademy.org/math/linear-algebra`

Free online book about JULIA:
`https://benlauwens.github.io/ThinkJulia.jl/latest/book.html`

## Clickers

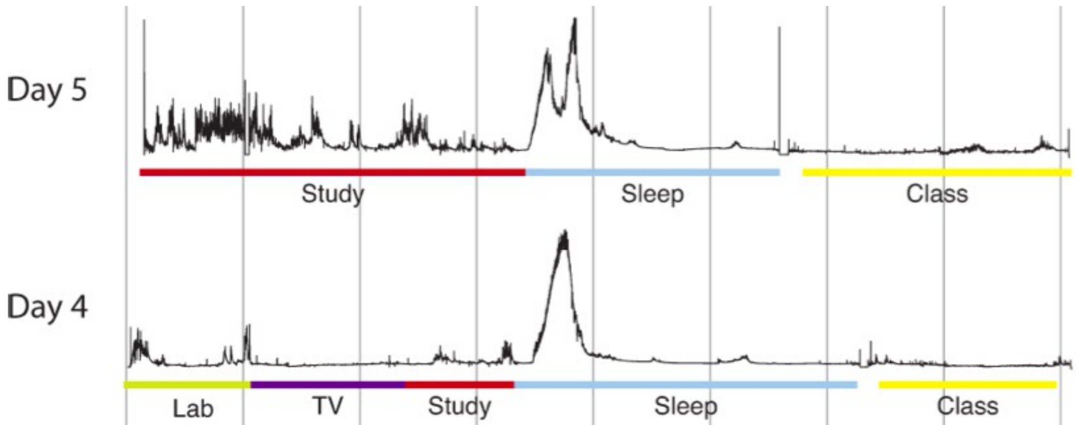
http://caenfaq.engin.umich.edu/10909-clickers/ Bring batteries!

**Action:** Buy at http://computershowcase.umich.edu/remotes/ ($29 used, buy back for $19)

**Action:** Register your clicker at Canvas.

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. [9]



## PDF lecture note features (Read)

These notes highlight some important **terms** 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.`

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 [10].
- Prof. Nadakuditi and I have used it since W17 for multiple courses at UM and MIT (505, 551, 598...)
- 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.
  For differences with MATLAB, see:
  `https://docs.julialang.org/en/v1/manual/noteworthy-differences`
- 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.
- JuliaBox allows within-browser use in the cloud, mostly for limited toy experimenting. Not recommended!

Some documentation / books:

- Online docs: `https://docs.julialang.org/en/stable/manual/documentation/`
- 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: A Springboard to Machine Learning"

- Online "Getting started with JULIA" book: https://search.lib.umich.edu/catalog/record/013714926
- 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

Sponsors of Juliacon 2018 and Juliacon 2019:

## 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 auto-graded, 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/`
  - `julia-tutor-vector` : vector/matrix operations and "**call by reference**" aspect of JULIA
  - `julia-tutor-slice` : matrix indexing (slicing)
  - `julia-tutor-sum-simd.html` : acceleration using `@simd`
- Later we will do in-class activities using JULIA and you will want to bring a laptop with JULIA on it.
- Editors (see list at JULIA web site: `https://julialang.org/`)
  - Juno: JULIA IDE `http://junolab.org/`
  - Atom: `https://atom.io/`
  - vim: `https://www.vim.org/`
- Use JULIA version 1.1 or later for F19; current version is 1.2.
  Beware of online Q/A for older versions of JULIA!
- **Actions:** Bring laptop to Discussion section Friday.

## 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 [11].
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 [12].

MATLAB is "free" for UM students:

http://caenfaq.engin.umich.edu/10378-Free-Software-for-Students/matlab-for-students

## JULIA: getting started

For F19, we recommend (but do not require) that you use the Juno IDE in the powerful Atom editor because of its excellent integration with the Julia Debugger. Alternatively, you may use your own favorite editor (though you may find debugging more challenging) or use the (free) JuliaPRO version from
`https://juliacomputing.com/products/juliapro`

- **Actions:** Follow detailed installation instructions at
  `https://github.com/JeffFessler/MIRT.jl/blob/master/doc/start-juno.md`
- 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`
- Link to video about Juno debugger (20 mins into JuliaCon 2019 talk)
  `https://youtu.be/SU0SmQnnGys?t=1200`

## Clicker survey questions

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

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

Office hours (do not answer if your schedule is still uncertain)

A: Wed 10:30-11:30 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].

### 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 semi-definite**

## 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. 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

## 8. Special matrices

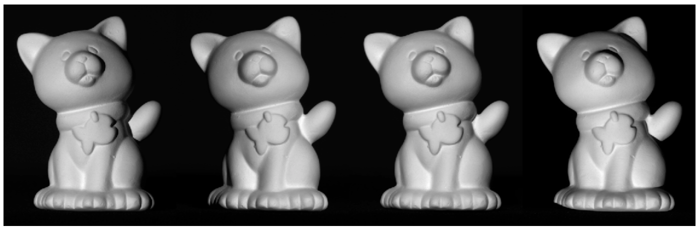
**monic polynomial**
**companion matrix**
**minimum polynomial**
**Vandermonde matrix**
**Kronecker sum**
**circulant matrix**
**DFT**
**fast Fourier transform**
**power iteration**
**positive matrix**
**nonnegative matrix**
**primitive matrix**
**Geršgorin disk theorem**
**Perron-Frobenius theorem**
**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**

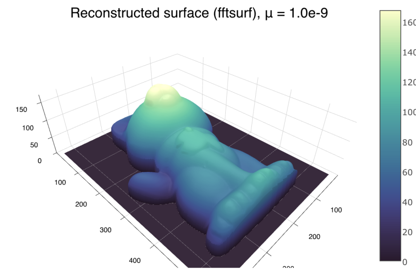
## 9. Matrix completion

**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**

Many applications in signal processing and machine learning, especially 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. 0.6, 0.16).

[2] T. K. Moon and W. C. Stirling. *Mathematical methods and algorithms for signal processing*. Prentice-Hall, 2000 (cit. on p. 0.6).

[3] G. H. Golub and C. F. Van Loan. *Matrix computations*. 2nd ed. Johns Hopkins Univ. Press, 1989 (cit. on p. 0.6).

[4] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge: Cambridge Univ. Press, 1985 (cit. on p. 0.6).

[5] G. Strang. *Linear algebra and learning from data*. Cambridge, 2019 (cit. on p. 0.6).

[6] L. Eldén. *Matrix methods in data mining and pattern recognition*. Errata: http://users.mai.liu.se/larel04/matrix-methods/index.html. Soc. Indust. Appl. Math., 2007 (cit. on p. 0.6).

[7] K. B. Petersen and M. S. Pedersen. *The matrix cookbook*. ., 2012 (cit. on p. 0.6).

[8] S. Boyd and L. Vandenberghe. *Introduction to applied linear algebra: Vectors, matrices, and least squares*. ., 2017 (cit. on p. 0.6).

[9] 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. 0.7).

[10] 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. 0.9).

[11] M. Innes. “Flux: Elegant machine learning with Julia”. In: *J. of Open Source Software* 3.25 (2018), p. 602 (cit. on p. 0.13).

[12] 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. 0.13).