Last (family) name:_____

First (given) name:_____

I have neither given nor received aid on this examination, nor concealed any violation of the Honor Code.
The only online resources I have used during this exam are those listed below as well as the following sites (list URL and exam problem #):

Signature: _____

ID Number: _____

EECS 551 Final, 2020-12-15 EST (24 hour online)

- There are 6 problems (on this part) for a total of 72 points.
  This part of the exam has 8 pages. Make sure your copy is complete.
- This is a 24-hour "online" exam. Some exam questions will be on Canvas; this pdf has additional questions that you should answer by submitting to gradescope unless instructed otherwise in each problem.
- You can start the Canvas part any time during the 24 hour period, and you must finish it within **2 hours** of when you started. You can answer each Canvas question only once. When you first start the quiz in Canvas, you will get an alert like: "Once you have submitted an answer, you will not be able to change it later. You will not be able to view the previous question." After you select one or more options, you click the "Next" button and that will finalize your selections for that question.

  If you have a temporary internet connection loss, you should be able to reconnect and continue where you left off, during the 2 hour time window. We set this time window long enough to allow for this possibility. It should take less than an hour to finish the Canvas part, so there is a 1-hour buffer to handle any connection issues.

  To be fair to students taking the Canvas part at different times, we will *not* reply to *any* questions about the Canvas part on Piazza during the exam period, so do not waste your time posting there.

  We have vetted the Canvas questions carefully and there is nothing intentionally ambiguous there. If you think there is some ambiguity in a Canvas question, then answer based on your best interpretation about what was meant. If you think there is an ambiguity on a Canvas question/answer, then upload an explanation along with your Honor Code statement as the last entry to gradescope.
- This is an "open book" exam. During the exam, you may use all of the course materials on the Canvas EECS 551 site including the course notes on google drive, as well as wikipedia, the built-in JULIA help, and the online JULIA manual. If you use any other resources for solving the Canvas questions, then you must cite the source along with your Honor Code statement. *If you use any other resources for solving the gradescope questions, then cite the source as part of your solution submitted to gradescope.* Be sure to *sign* the honor code above and scan (e.g., photograph) the top part of this page and submit to gradescope.
- You may use without rederiving any of the results presented in the course notes.
- You must complete the exam entirely on your own.
  Obviously you must not put your code or solutions anywhere that would be accessible to any other student.

Failure to adequately keep your work private would itself be an Honor Code violation.
Do not discuss the exam with anyone until the entire exam period is over.

- Read and think about all questions before asking any questions on Piazza.
  Posting questions about how to *solve* a problem is inappropriate, and may lead to point deductions.
  Only post a (private) question if you are sure that an exam question needs to be clarified.

- Clearly box your final answers. For full credit, show your complete work clearly and legibly.
  Answers must be submitted properly to gradescope to earn credit.

- For multiple-choice questions, select *all* correct answers.

- To "disprove" any statement, provide a concrete counter-example. For maximum credit, make that counter-example as small and simple as possible, *e.g.*, having the smallest possible matrix dimensions and using the simplest numbers like "0" and "1" as much as possible. For example, to disprove the statement "any square matrix $\boldsymbol{A}$ is invertible," the smallest and simplest counter-example is the 1×1 matrix $\boldsymbol{A} = [0]$.

autograder instructions

For any problem involving writing a JULIA function, carefully follow these instructions.

- Submit a mathematical explanation of your solution to gradescope.
- Submit a readable screenshot of your code to gradescope for grading *efficiency*.
  Solutions must be as efficient as possible to earn full credit for any problem involving JULIA.
- Test your code thoroughly *before* submitting, to maximize credit earned.
  (Passing on the 1st or 2nd submission will earn full credit; partial credit will decrease rapidly after that.)
- Submit your tested code by email attachment to `eecs551@autograder.eecs.umich.edu` as usual.
  Incorporate any necessary `using` statements.
  Name your function correctly.
  Name the file attachment like `thefunction.jl` where `thefunction` is the function name.
  The very first line of your `.jl` file must be a comment line that has *your* unique name in it like this:
  `# fessler`
  That first comment line must be visible in any code that you submit to the autograder and to gradescope.

For any prohibited function, you may not use the "in place" version either. For example, if `sort` is not allowed, then the corresponding in-place version `sort!` is also not allowed.

Unless you are told otherwise, the autograder is set up only with the packages that you have used on autograded homework problems. You cannot use `Pkg` to add other packages. Unlike in Midterm3, submissions with misspelled package names, uninstalled package names, etc., will count towards your attempts. Test carefully!

The following packages are built-in and available with any Julia installation:

```
LinearAlgebra
SparseArrays
Statistics
```

The following packages are installed on the auto-grader:

```
AbstractAlgebra v0.11.1
FFTW v1.2.4
Polynomials v1.1.5
```

Submitting code that tries `using` with other packages will produce an error that may prevent you from earning full credit. If you think this list overlooks something you have used, then check on Piazza well in advance of submitting.

[8] **1**.

(a) Draw a Markov chain whose 2×2 transition matrix is rank-1, primitive, and asymmetric.

(b) Specify the corresponding transition matrix.

(c) Specify the equilibrium distribution of this Markov chain.

[8] **2**. Define $f : \mathbb{F}^{M\times N} \mapsto \mathbb{R}$ by $f(\boldsymbol{X}) = \|\boldsymbol{AX}\|_*$, where $\boldsymbol{A}$ is an invertible $M \times M$ matrix. This function is Lipschitz continuous; determine a (not necessarily best) Lipschitz constant for it. (You may pick any norms you want for defining Lipschitz continuity, just specify your choice.) Explain the Lipschitz constant you determine.
Optional challenge (not for points): discuss whether it is the best (smallest) Lipschitz constant.

[8] **3**. Complete the following JULIA function so that it computes $\alpha_* = \underset{\alpha\in\mathbb{R}}{\arg\min} \left\| \boldsymbol{I} - \alpha \boldsymbol{A}'\boldsymbol{A} \right\|_{\mathrm{F}}$ when $\boldsymbol{A} \in \mathbb{F}^{M\times N}$ is a nonzero matrix. Your code should be as efficient as possible. Show your work to explain your math.

```
function alphastar(A::Matrix)
```

```
end
```

This is not an autograder problem! Just submit a screenshot to gradescope.

[15] **4**. The compact SVD of a matrix $\boldsymbol{Y} \in \mathbb{F}^{M \times N}$ is $\boldsymbol{Y} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r'$, where $r = \mathsf{rank}(\boldsymbol{Y})$ and $\boldsymbol{U}_r$ consists of the first r left singular vectors of an SVD of $\boldsymbol{Y}$. Complete the following JULIA function so that it returns

$$\hat{\boldsymbol{X}}_K \triangleq \underset{\boldsymbol{X} \in \mathbb{F}^{M \times N} \,:\, \mathsf{rank}(\boldsymbol{X}) \le K}{\arg\min} \|\boldsymbol{Y} - \boldsymbol{X}\|_{\mathrm{F}},$$

when given $\boldsymbol{Y}$, its corresponding $\boldsymbol{U}_r$, and $K \in \{0, 1, 2, \ldots\}$, *without* calling any matrix decomposition functions like `svd`, `eigen`, `qr`, etc., or their relatives. Hint. Be sure to consider *all* nonnegative K values.
Template:

```
"""
    Xh = lr_ur(Y, Ur, K::Int)

Compute the rank-at-most-K best approximation of `Y = U_r Σ_r V_r'`

In:
- `Y` : `M × N` matrix
- `Ur` : `M × r` matrix with first `r` left singular vectors of `Y`
- `K` maximum rank of `Xh` (any nonnegative integer)

Out:
- `Xh` : `M × N` best approximation, having rank <= `K`
"""
function lr_ur(Y::AbstractMatrix, Ur::AbstractMatrix, K::Int)
```

See autograder submission instructions near front of exam.
Remember to put your uniquename in a comment on the first line!

[15] **5**. Consider the set of $N \times M$ tight frames: $\mathcal{T} = \left\{ \boldsymbol{T} \in \mathbb{F}^{N \times M} \ : \ \boldsymbol{T} \text{ is a tight frame} \right\}$.
Write a JULIA function that, given a nonzero $N \times M$ matrix $\boldsymbol{X}$, returns the closest (in the Frobenius norm sense) tight frame to $\boldsymbol{X}$, *i.e.*, $\hat{\boldsymbol{T}} = \underset{\boldsymbol{T} \in \mathcal{T}}{\arg\min} \left\| \boldsymbol{T} - \boldsymbol{X} \right\|_{\mathrm{F}}$. Assume $N \leq M$.
Optional (not graded): think about why the problem statement assumes $\boldsymbol{X}$ is nonzero.
Template:

```
"""
    T = neartf(X)

Find nearest (in Frobenius norm sense) tight frame to matrix `X`.

In:
* `X` : `N × M` nonzero matrix with `N ≤ M`

Out:
* `T` `N × M` matrix that is the nearest tight frame to `X`
"""
function neartf(X::AbstractMatrix)
```

See autograder submission instructions near front of exam.
Remember to put your uniquename in a comment on the first line!

[18] **6**. Let $\boldsymbol{X}$, $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{Y}$ all denote matrices in $\mathbb{F}^{N\times N}$, and define the following optimization problem

$$\hat{\boldsymbol{X}} = \underset{\boldsymbol{X}\in\mathbb{F}^{N\times N}}{\arg\min} \|\boldsymbol{U}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{V} - \boldsymbol{Y}\|_{\mathrm{F}}.$$

(a) Derive an analytical solution for $\hat{\boldsymbol{X}}$ in terms of the other matrices given in problem. Explain.
Hint: trying to differentiate would be unwise; we never used that strategy for problems with this norm in 551.

(b) Specify and explain necessary and sufficient conditions on the problem matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{Y}$ for ensuring that the above optimization problem has a unique minimizer.

(c) Complete the following JULIA function so that it returns $\hat{\boldsymbol{X}}$ given $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{Y}$. You may assume that the input matrices are square and have the same size and satisfy appropriate sufficient conditions for uniqueness of $\hat{\boldsymbol{X}}$. The target matrix size is small enough that you can use your analytical solution; do not use any iterative method.

To earn full credit, the code must be as efficient as possible, and should work for both real and complex inputs. Passing the autograder with inefficient code will not earn full credit.

Template:

```
"""
    Xh = ux_xv(U, V, Y)
Compute `\\hat{X} = \\argmin_X \\| UX + XV - Y \\|_F`
assuming `U` and `V` are such that the minimizer is unique.

In:
* `U, V, Y` : `N × N` matrices

Out:
* `Xh` `N × N` matrix; minimizer of the optimization problem
"""
function ux_xv(U::AbstractMatrix, V::AbstractMatrix, Y::AbstractMatrix)
```

See autograder submission instructions near front of exam.
Remember to put your uniquename in a comment on the first line!

end