STAT 771: My notes

Ralph Møller Trane

Fall 2018 (compiled 2018-10-18)

Contents

Intro **5**

1 Lecture Notes **7**
1.1 Positional numeral system 7
1.2 Floating Point Format 7
1.3 Orthogonalization 14
1.4 Singular Value Decomposition (SVD) 21
1.5 Iterative Methods 29

2 Homework Assignments **37**
2.1 Homework 1 37
2.2 Homework 2 39
2.3 Homework 3 41
2.4 Homework 4 45
2.5 Homework 5 47
2.6 Homework 6 48
2.7 Homework 7 49
2.8 Homework 8 50
2.9 Homework 9 51
2.10 Homework 10 51
2.11 Homework 11 51

Intro

Chapter 1

Lecture Notes

Lecture 1: 9/6

Goals for the first few lectures:

1. Develop basic understanding of floating point numbers (*fp* numbers)
2. Develop some basic notions of errors and their consequences

References:

i. David Goldberg (1991)
ii. John D. Cook (2009)
iii. Hingham (2002)

1.1 Positional numeral system

We assume we have a decimal representation of numbers. I.e. that it exists. It is not within the scope of this class to prove this.

Now, this is NOT the optimal way for a computer to represent numbers. For various reasons, there are more desirable ways to store numbers. So we need a different way of representing the numbers.

Ingredients for different representation:

i. A base, refered to as β. It holds that $\beta \in 2, 3, 4, ...$
ii. A significand: a sequence of digits: $d_0.d_1d_2d_3d_4...$, where $d_j \in \{0, 1, ..., \beta - 1\}$
iii. An exponent: $e \in \mathbb{Z}$.

The representation $d_0.d_1d_2... \times \beta^e$ means $\left(d_0 + d_1 \cdot \beta^{-1} + ... + d_{p-1}\beta^{-(p-1)}\right) \cdot \beta^e$.

1.2 Floating Point Format

Definition 1.1. A fp is one that can be represented in a base β with a fixed digit p (precision), and whose exponent is between e_{min} and e_{max}.
Example 1.1. Let $\beta = 10, p = 3, e_{min} = -1, e_{max} = 1$. Want to represent 0.1. Several options:

i. Let d_0=0, d_1 = 0, d_2 = 1, e = 1.
ii. Let d_0=0, d_1 = 1, d_2 = 0, e = 0.
iii. d_0 = 1, d_1 = 0, d_2 = 0, e = -1.

If we fill into the equation above, we get 0.1:

$$i : \left(0 + 0 \cdot 10^{-1} + 1 \cdot 10^{-2}\right) \cdot 10^{1}$$
$$ii : \left(0 + 1 \cdot 10^{-1} + 1 \cdot 10^{-2}\right) \cdot 10^{0}$$
$$iii : \left(1 + 0 \cdot 10^{-1} + 1 \cdot 10^{-2}\right) \cdot 10^{-1}$$

Definition 1.2. A fp number is said to be *normalized* if $d_0 \neq 0$.
Exercise 1.1. What is the total number of values that can be represented in the normalized fp format with base $\beta, p, e_{min}, e_{max}$?

We count the different values each of the elements of a *fp* can take:

- d_0 can be from 1 to $\beta - 1$, so $\beta - 1$ different values.
- $d_1, ..., d_{p-1}$ each takes a value in $\{0, 1, ..., \beta - 1\}$. Hence, we can choose the digits $d_1, ..., d_{p-1}$ in β^{p-1} different ways.
- e can take $e_{max} - e_{min} + 1$ different values (all integers from e_{min} to e_{max}, both included, hence the $+1$).

So, in total, there are $(\beta - 1) \cdot \beta^{p-1} \cdot (e_{max} - e_{min} + 1)$ different values that can be represented in the normalized fp format with base β, precision p, and e_{min}, e_{max} given.

1.2.1 IEEE Standards

IEEE have standards for how to deal with approximations and errors.

For our purposes, a bit is a single unit of storage on a computer, which can either be 0 or 1. Hence, we'll be focusing on fp formats where $\beta = 2$.

1.2.1.1 The 16 bit standard (half precision standard).

The 16 bits of storage are used in the following way, when following the 16 bit standard:

- 1 bit for the sign
  - 0 = positive
  - 1 = negative
- 5 bits for the exponent
  - 00000 is reserved for 0
  - 11111 is reserved for ∞
  - 30 exponents left: $2^5 - 2 = 30$
  - the 16 bit standard dictates that the used exponents are $-14, ..., 15$.
    - **Note**: 0 is also included in this list of 30 exponents. This is because the 00000 representation is reserved for integers, while 01111 is used with non-integers.
- 11 bit for the significand.
  - 10 are actually stored – we always work with normalized FP numbers, i.e. $\beta_0 = 1$.

Question: What are smallest and largest positive numbers that can be represented?

Answer: Smallest non-normalized number would be the one with the smallest possible exponent, and all digits of the significand are 0 except the very last one. So, the smallest non-normalized FP number in the 16 bit standard would be

$$\left(0 + 0 \cdot 2^{-1} + ... + 0 \cdot 2^{-9} + 1 \cdot 2^{-10}\right) \cdot 2^{-14} = 2^{-24} \approx 5.96 \cdot 10^{-8}$$

The smallest normalized number is the one with all digits 0 (except for the leading digit, of course, which has to be 1 for it to be normalized), and $e = -14$. So the smallest normalized FP number:

$$\left(1 + 0 \cdot 2^{-1} + ... + 0 \cdot 2^{-10}\right) \cdot 2^{-14} = 2^{-14} \approx 6.10 \cdot 10^{-5}$$

Finally, the largest (finite) FP number in the 16 bit standard is the one where the exponent is as large as possible ($e = 15$), and all digits are 1. So

$$\left(1 + 1 \cdot 2^{-1} + ... + 1 \cdot 2^{-10}\right) \cdot 2^{15} = 65504$$

Lecture 2: 9/11

1.2.1.2 The 32 bit standard (single precision)

The 32 bits of storage are used in the following way, when following the 32 bit standard:

- 1 bit for the sign
  - 0 = positive
  - 1 = negative
- 8 bits for the exponent
  - 00000000 is reserved for 0
  - 11111111 is reserved for ∞
  - exponents left: $2^8 - 2 = 254$
  - the 32 bit standard dictates that the used exponents are $-126, ..., 127$.
    - **Note**: 0 is also included in this list of the 254 exponents. This is because the 00000000 representation is reserved for integers, while 01111111 (I think this is the representation for 0 here...) is used with non-integers.
- 24 bit for the significand.
  - 23 are actually stored – we always work with normalized FP numbers, i.e. $\beta_0 = 1$.

Question: What are smallest and largest positive numbers that can be represented in the 32 bit standard?

Answer: Smallest non-normalized number would be the one with the smallest possible exponent, and all digits of the significand are 0 except the very last one. So, the smallest non-normalized FP number in the 32 bit standard would be

$$\left(0 + 0 \cdot 2^{-1} + ... + 0 \cdot 2^{-22} + 1 \cdot 2^{-23}\right) \cdot 2^{-126} = 2^{-149} \approx 1.40 \cdot 10^{-45}$$

The smallest normalized number is the one with all digits 0 (except for the leading digit, of course, which has to be 1 for it to be normalized), and $e = -126$. So the smallest normalized FP number:

$$\left(1 + 0 \cdot 2^{-1} + ... + 0 \cdot 2^{-23}\right) \cdot 2^{-126} = 2^{-126} \approx 1.18 \cdot 10^{-38}$$

Finally, the largest (finite) FP number in the 32 bit standard is the one where the exponent is as large as possible ($e = 127$), and all digits are 1. So

$$\left(1 + 1 \cdot 2^{-1} + ... + 1 \cdot 2^{-126}\right) \cdot 2^{127} = 3.40 \cdot 10^{38}$$

1.2.1.3 The 64 bit standard (double precision)

The 64 bits of storage are used in the following way, when following the 64 bit standard:

- 1 bit for the sign

  - 0 = positive
  - 1 = negative
- 11 bits for the exponent
  - 00000000 is reserved for 0
  - 11111111 is reserved for ∞
  - exponents left: $2^11 - 2 = 2046$
  - the 64 bit standard dictates that the used exponents are $-1024, ..., 1023$.
    - **Note**: 0 is also included in this list of the 254 exponents. This is because the 00000000 representation is reserved for integers, while 01111111 (I think this is the representation for 0 here...) is used with non-integers.
- 53 bit for the significand.
  - 52 are actually stored – we always work with normalized FP numbers, i.e. $\beta_0 = 1$.

1.2.2 Errors

1.2.2.1 Units in the Last Place (ULP)

1.2.2.2 Absolute and Relative Error

Let $fl : \mathbb{R}_{\geq 0} \to \mathcal{S}$ be a function that takes a real value and return a FP number. Then we define the absolute and relative error as follows:
Definition 1.3. Let $z \in \mathbb{R}_{\geq 0}$. The *absolute error* is defined as

$$|fl(z) - z|.$$

The *relative error* is defined as

$$\left|\frac{fl(z) - z}{z}\right|$$

Lemma 1.1. *If z has exponent e, then the maximum absolute error is $\frac{\beta^{e-p+1}}{2}$.*

Proof. □

Lemma 1.2. *If z has exponent e, then the maximum relative error is $\frac{\beta^{1-p}}{2}$.*

Proof. If z has exponent e, then $\beta^e \leq z$. Using this with 1.1, we get that

$$\left|\frac{fl(z) - z}{z}\right| \leq \frac{\beta^{e-p+1}}{2\beta^e} = \frac{\beta^{1-p}}{2}.$$

□

Note: the upper bound of the relative error is called the *machine epsilon*. This can be obtained in Julia using the function `eps`.

1.2.2.3 The Fundamental Axiom

... is that for any of the four arithmetic operations $(+, -, \cdot, /)$, we have the following error bound:

$$fl(x \text{op} y) = (x \text{op} y)(1 + \delta),$$

with $|\delta| \leq u$, where u is commonly $2 \cdot \epsilon$. (**NOTE: NEED TO CLARIFY IF THE ABOVE IS CORRECT!**)

Example: Matrix storage. Let $A \in \mathbb{R}^{m \times n}$. Then:

$$|fl(A) - A| \leq u\,|A|$$

Example: Dot product. Let $x, y \in \mathbb{R}^n$. Recall that the dot product of x and y is defintied as $x'y = \sum_{i=1}^{n} x_i \cdot y_i$. This can be calculated in the following way:

```
fl = function(x,y)
  # Get length of x
  n = length(x)
  # Check that length of y is equal to length of x. If not, throw error.
  if(length(y) != n)
    return "ERROR: y does not have same dimension as x"
  end

  # s will be the result of the dot product calculation
  s = 0

  for i = 1:n
    s += x[i]*y[i]
  end

  return(s)

end
```

Next we want to prove the following lemma:
Lemma 1.3. *Let $x, y \in \mathbb{R}^n$, and $n \cdot u \leq 0.01$. Then*

$$|fl(x'y) - x'y| \leq 1.01 \cdot n \cdot u \cdot |x|'\,|y|$$

Lecture 3: 9/13

To prove the lemma above, we will need another lemma...
Lemma 1.4. *If $|\delta_i| \leq u, \forall i = 1, \ldots, n$ s.t. $n \cdot u < 2$. Let $1 + \eta = \prod_{i=1}^{n}(1 + \delta_i)$. Then*

$$|\eta| \leq \frac{n \cdot u}{1 - \frac{n \cdot u}{2}}$$

Proof. Using the definition of ν, we can rewrite it to get

$$|\eta| = \left| \prod_{i=1}^{n}(1 + \delta_i) - 1 \right|.$$

By induction, we will show that the expression above is less than or equal to $(1+u)^n - 1$. [TO BE COMPLETED!]

Since $1 + u \leq e^u$ for all $u \in \mathbb{R}$, we have that

$$\begin{aligned}
|\eta| &\leq e^{n\cdot u} - 1 \\
&\leq n \cdot u + \frac{(n\cdot u)^2}{2!} + \frac{(n\cdot u)^3}{3!} + \ldots \text{(used the Taylor expansion)} \\
&\leq n \cdot u + \frac{(n\cdot u)^2}{2^1} + \frac{(n\cdot u)^3}{2^2} + \frac{(n\cdot u)^4}{2^3} + \ldots \text{(used that } x! > 2^{x-1} \text{ for } x > 1) \\
&= \sum_{k=0}^{\infty} n \cdot u \left(\frac{n\cdot u}{2}\right)^k \text{ (identify this as a geometric series with } r = \tfrac{n\cdot u}{2}\text{, which is less than 1 by assumption)} \\
&= \frac{n\cdot u}{1 - \frac{n\cdot u}{2}},
\end{aligned}$$

which is exactly what we wanted. □

With this in hand, we will prove the previously stated lemma.

Proof. Let s_p denote the value of s after the p'th iteration of the algorithm described above. Then, since we're assuming the Fundamental Axiom, we have that $s_1 = fl(x_1 y_y) = x_1 y_1(1+\delta_1)$, where $|\delta_1| \leq u$. We can similarly find s_p as

$$\begin{aligned}
s_p &= fl(s_{p-1} + fl(x_p y_p)) \\
&= (s_{p-1} + fl(x_p y_p))(1+\epsilon_p)(\text{where } |\epsilon_p| \leq u) \\
&= (s_{p-1} + x_p y_p(1+\delta_p))(1+\epsilon_p)(\text{where } |\delta_p| \leq u).
\end{aligned}$$

Let $\epsilon_1 = 0$. s_p is a recursive formula, and can be rewritten as follows:

$$s_p = \sum_{i=1}^{p} x_i y_i (1+\delta_i) \prod_{j=1}^{p} (1+\epsilon_j).$$

So,

$$\begin{aligned}
|s_n - x'y| &= \left| \sum_{i=1}^{n} (x_i y_i)(1+\delta_i) \prod_{j=1}^{p} (1+\epsilon_j) - \sum_{i=1}^{n} x_i y_i \right| \\
&= \left| \sum_{i=1}^{n} (x_i y_i) \left((1+\delta_i) \prod_{j=1}^{p} (1+\epsilon_j) - 1 \right) \right| \\
&\leq \sum_{i=1}^{n} |x_i y_i| \left| (1+\delta_i) \prod_{j=1}^{p} (1+\epsilon_j) - 1 \right|.
\end{aligned}$$

We now use 1.4 to get:

$$\sum_{i=1}^{n} |x_i y_i| \left| (1+\delta_i) \prod_{j=1}^{p} (1+\epsilon_j) - 1 \right| \leq \frac{nu}{1-\frac{nu}{2}} \sum_{i=1}^{n} |x_i y_i|$$
$$\leq \frac{nu}{0.995} \sum_{i=1}^{n} |x_i||y_i|$$
$$\leq 1.01 \cdot nu \cdot |x|'|y|$$

□

1.2.3 Square Linear Systems

In the following, let $A \in \mathbb{R}^{n\times m}$ be an invertible matrix, and assume $Ax = b$ for a $b \neq 0$. This implies that $x = A^{-1}b$.

Theorem 1.1. *Let $\kappa_\infty = ||A||_\infty \left|\left|A^{-1}\right|\right|_\infty$. Assume we can store A with precision E (i.e. as $A+E$), where $||E||_\infty \leq u\,||A||_\infty$, and b with precision e (i.e. as $b+e$), where $||e||_\infty \leq u\,||b||_\infty$.*

If $||A+E||\,\hat{x} = b+e$ and $u \cdot \kappa_\infty < 1$, then

$$\frac{||x-\hat{x}||_\infty}{||x||} \leq \frac{2\cdot u\cdot \kappa_\infty}{1-u\cdot\kappa_\infty}$$

Lemma 1.5. *Let $I \in \mathbb{R}^{n\times n}$ be the identity matrix, and $F \in \mathbb{R}^{n\times n}$ s.t. $\left|\left|F\right|\right|_p < 1$ for some $p \in [1,\infty]$. Then $I-F$ is invertible, and*

$$\left|\left|(I-F)^{-1}\right|\right|_p \leq \frac{1}{1-||F||_p}$$

Proof. **HOMEWORK** □

Lemma 1.6. *Suppose $\exists \epsilon > 0$ s.t. $||\Delta A|| \leq \epsilon\,||A||$ and $||\Delta b|| \leq \epsilon\,||b||$, and y s.t. $(A+\Delta A)y = b + \Delta b$.*

If $\epsilon\,||A||\left|\left|A^{-1}\right|\right| = r < 1$, then $A+\Delta A$ is invertible and

$$\frac{||y||}{||x||} \leq \frac{1+r}{1-r}.$$

Proof. Note that $A+\Delta A = A\left(I + A^{-1}\Delta A\right) = A\left(I - \left(-A^{-1}\Delta A\right)\right)$. Since $\left|\left|-A^{-1}\Delta A\right|\right| = \left|\left|A^{-1}\Delta A\right|\right| \leq \epsilon\left|\left|A^{-1}\right|\right| \cdot ||A|| < 1$ (by assumptions), Lemma 1.5 gives us that $I + A^{-1}\Delta A$ is invertible. Since A is also invertible (again, by assumption), $A+\Delta A$ is invertible (product of two invertible matrices is invertible).

Performing some linear algebra:

$$\begin{aligned}(A+\Delta A) &= b+\Delta b \Leftrightarrow\\ A(I+A^{-1}\Delta A)y &= b+\Delta b \Leftrightarrow\\ (I+A^{-1}\Delta A)y &= A^{-1}b + A^{-1}\Delta b \Leftrightarrow\\ y &= (I+A^{-1}\Delta A)^{-1}A^{-1}b + A^{-1}\Delta b.\end{aligned}$$

Remember that $A^{-1}b = x$. From the definition of r we have that $\left|\left|A^{-1}\right|\right| = \frac{r}{||A||}$. These two identities with the assumption that $||\Delta b|| \leq \epsilon b$ gives us

$$\begin{aligned}
||y|| &\leq \left|\left|(I + A^{-1}\Delta A)^{-1}\right|\right| \left(||x|| + \left|\left|A^{-1}\Delta b\right|\right|\right) \\
&\leq \frac{1}{1 - ||A^{-1}\Delta A||} \left(||x|| + \frac{r}{\epsilon\,||A||} \cdot ||\Delta b||\right) \\
&\leq \frac{1}{1-r} \left(||x|| + \frac{r}{\epsilon\,||A||} \cdot \epsilon\,||b||\right) \\
&= \frac{1}{1-r} \left(||x|| + \frac{r \cdot ||b||}{||A||}\right).
\end{aligned}$$

Finally, recall that $Ax = b$, hence $||A|| \cdot ||x|| \geq ||b||$, so $||x|| \geq \frac{||b||}{||A||}$. So,

$$\begin{aligned}
||y|| \leq \frac{1}{1-r} \left(||x|| + r \cdot ||x||\right) &\Leftrightarrow \\
\frac{||y||}{||x||} &\leq \frac{1+r}{1-r}.
\end{aligned}$$

□

Lemma 1.7.

$$\frac{||y - x||}{||x||} \leq \frac{2\epsilon \left|\left|A^{-1}\right|\right| \cdot ||A||}{1-r}.$$

Proof.

$$\begin{aligned}
(A + \Delta A)\, y &= b + \Delta b \Leftrightarrow \\
Ay - b &= \Delta b - \Delta A y \Leftrightarrow \\
y - A^{-1}b &= A^{-1}\Delta b - A^{-1}\Delta A y \Leftrightarrow \\
y - x &= A^{-1}\Delta b - A^{-1}\Delta A y \Leftrightarrow \\
||y - x|| &\leq \left|\left|A^{-1}\right|\right| ||\Delta b|| + \left|\left|A^{-1}\right|\right| ||\Delta A||\, ||y|| \\
&\leq \left|\left|A^{-1}\right|\right| \epsilon\, ||b|| + \left|\left|A^{-1}\right|\right| \epsilon\, ||A||\, ||y|| \\
&\leq \epsilon \left|\left|A^{-1}\right|\right| ||A||\, ||x|| + \epsilon \left|\left|A^{-1}\right|\right| ||A||\, ||y|| \\
&\leq \epsilon \left|\left|A^{-1}\right|\right| ||A|| \left(||x|| + ||y||\right) \\
&= \epsilon \left|\left|A^{-1}\right|\right| ||A|| \left(||x|| + \frac{1+r}{1-r}\, ||x||\right) \Leftrightarrow \\
\frac{||y - x||}{||x||} &\leq \epsilon \left|\left|A^{-1}\right|\right| ||A|| \left(\frac{1-r}{1-r} + \frac{1+r}{1-r}\right) \\
&= 2\epsilon \left|\left|A^{-1}\right|\right| ||A|| \frac{1}{1-r}
\end{aligned}$$

□

1.3 Orthogonalization

Goals

1) Introduce and prove the existence of QR decomposition
2) Overview of the algorithm to perfor QR decomposition
3) Solve least squares problems
4) "Large" data problems

Outline

1) Motivating problems and solutions with QR
2) Gram-Schmidt procedure, existence of QR
3) Householder, Givens
4) "Large" least squares problems datadown

1.3.1 Motivating problems

Example 1.2 (Motivating Problem 1 (Consistent Linear System))**.** Assume $A \in \mathbb{R}^{n\times m}, n \geq m, \text{rank}(A) = m$, and $b \in \text{range}(A) \subset R^m$. Find $x \in \mathbb{R}^m$ s.t. $Ax = b$.
Example 1.3 (Motivating Problem 2 (Least Squares Regression))**.** Assume $A \in \mathbb{R}^{n\times m}, n \geq m, \text{rank}(A) = m$, and $b \in R^n$. Find $x \in \mathbb{R}^m$ s.t.

$$x \in \text{argmin}_{y\in\mathbb{R}^m} ||Ay - b||_2 .$$

Example 1.4 (Motivating Problem 3 (Underdetermined Linear System)**.** Assume $A \in \mathbb{R}^{n\times m}, n \geq m, \text{rank}(A) < m$, and $b \in \text{range}(A)$. Find $x \in \mathbb{R}^m$ s.t.

$$x \in \text{argmin}_{y\in\mathbb{R}^m} \{||y||_2 \,|Ay = b\} .$$

Example 1.5 (Motivating Problem 4 (Underdetermined Least Squares Regression))**.** Assume $A \in \mathbb{R}^{n\times m}, n \geq m, \text{rank}(A) < m$, and $b \in \mathbb{R}^n$. Find $x \in \mathbb{R}^m$ s.t.

$$x \in \text{argmin}_{z\in\mathbb{R}^m} \left\{ ||z||_2 \,\middle|\, ||Ay - b||_2 = \min_{y\in\mathbb{R}^m} ||Ay - b||_2 \right\} .$$

Example 1.6 (Motivating Problem 5 (Constrained Least Squares Regression))**.** Assume $A \in \mathbb{R}^{n\times m}, n \geq m, \text{rank}(A) = m$, and $b \in \mathbb{R}^n$. Let $C \in \mathbb{R}^{p\times m}, \text{rank}(C) = p$, and $d \in \mathbb{R}^p$. Find $x \in \mathbb{R}^m$ s.t.

$$x = \text{argmin}_{y\in\mathbb{R}^m} ||Ay - b||_2 \quad \text{s.t.} \quad Cy = d.$$

Before we take a crack at solving these problems, we will need to get some definitions down.
Definition 1.4 (Permutation Matrix)**.** A permutation matrix is a square matrix such that each column has exactly one element that is 1, the rest are 0.
Example 1.7. The following is a permutation matrix:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Definition 1.5 (Orthogonal Matrix)**.** A matrix Q is said to be an *orthogonal matrix* if $Q^TQ = QQ^T = I$.

Note: for an orthogonal matrix $Q \in \mathbb{R}^{n\times m}$, it holds that $||Q_{i*}||_2 = 1$ for all $i = 1, \ldots, n$, and $||Q_{*j}||_2 = 1$ for all $j = 1, \ldots, m$.[1]
Definition 1.6 (Upper Triangular Matrix)**.** A matrix R is an *upper triangular matrix* if $R_{ij} = 0$ for all $i > j$.

Lecture 4: 9/18

1.3.2 QR Decomposition

In order to actually solve the problems listed above, we need the QR Decomposition:
Theorem 1.2 (Existence of QR Decomposition)**.** *Let $A \in \mathbb{R}^{n\times m}$ and let $r = rank(A)$. Then there exists:*

1) an $m \times m$ permutation matrix Π,
2) an $n \times n$ orthogonal matrix Q,

[1]Here we use the notion Q_{i*} to mean the i'th row, and Q_{*j} to mean the j'th column of Q.

3) an $r \times r$ upper triangular matrix R, with non-zero diagonal elements (i.e. invertible)
4) an $r \times (m-r)$ matrix S (if $m > r$),

such that

$$A = Q \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} \Pi^T.$$

With this in hand, we can solve the motivating problems stated above.
Solution (Example ref(exm:linear-system)). We want to find x such that $Ax = b$.

We use theorem 1.2 to rewrite this as $Q \begin{bmatrix} R \\ 0 \end{bmatrix} \Pi^T x = b$. Note that since $\text{rank}(A) = m$, there is no S matrix.

Now, since Q is an orthogonal matrix, we know that $Q^{-1} = Q^T$, so

$$\begin{bmatrix} R \\ 0 \end{bmatrix} \Pi^T x = Q^T b = c = \begin{bmatrix} c_1 \\ 0 \end{bmatrix}. \tag{1.1}$$

So now the equation we are trying to solve becomes

$$R\Pi^T x = c_1.$$

Since R is an upper triangular matrix with non-zero diagonal elements, it is invertible. Since Π is a permutation matrix, $\Pi^{-1} = \Pi^T$. Using this we can find the solution:

$$x = \Pi R^{-1} c_1.$$

Solution (Example ref(exm:least-squares)). We want to find x such that $x \in \text{argmin}_{y \in \mathbb{R}^m} ||Ay - b||_2$.

Once again, $\text{rank}(A) = m$, so using theorem 1.2, we can rewrite the expression we are trying to minimize as

$$\min \left|\left| Q \begin{bmatrix} R \\ 0 \end{bmatrix} \Pi^T y - b \right|\right|_2.$$

Since $Q^T = Q^{-1}$ is orthogonal, $\left|\left|Q^T x\right|\right|_2 = ||x||_2$ for all x (homework exercise 2.29). So, we get that (1.3.2) is the same as

$$\min \left|\left| \begin{bmatrix} R \\ 0 \end{bmatrix} \Pi^T y - Q^T b \right|\right|_2.$$

Now let $c = Q^T b$. Then, c is of the form $\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$, where c_2 is the last $n - r$ rows (i.e. corresponding to the 0 rows of $\begin{bmatrix} R \\ 0 \end{bmatrix}$). Then

$$\min \left|\left| \begin{bmatrix} R\Pi^T y - c_1 \\ -c_2 \end{bmatrix} \right|\right|_2 = \min \sqrt{||R\Pi^T y - c_1||_2^2 + ||c_2||_2^2}.$$

Now this is minimized by $\text{argmin}_y \left|\left|R\Pi^T y - c_1\right|\right|_2^2$. As before, R^{-1} exists since R is upper triangular with non-zero diagonal elements, $\Pi^T = \Pi^{-1}$ since Π is a permutation matrix, so

$$\begin{aligned} x &= \text{argmin}_y \left|\left|R\Pi^T y - c_1\right|\right|_2^2 \Leftrightarrow \\ R\Pi^T x &= c_1 \Leftrightarrow \\ x &= \Pi R^{-1} c_1. \end{aligned}$$

Solution (Example ref(exm:und-linear-system)). In this scenario, $\text{rank}(A) = r < m$. We are looking for $x \in \text{argmin}_y \{||y||_2 \,|Ay = b\}$. Using theorem 1.2, we can rewrite this as $\text{argmin}_y \left\{||y||_2 \,|Q \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} y = b\right\}$, and multiplying by Q^T, $\text{argmin}_y \left\{||y||_2 \left| \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} y = Q^T b\right.\right\}$. We introduce the vector c such that $Q^T b = \begin{bmatrix} c & 0 \end{bmatrix}^T$ (0 entries correspond to 0 rows in $\begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix}$). If we furthermore write $\Pi^T y$ as $\begin{bmatrix} z_1 & z_2 \end{bmatrix}^T$.

Then, since $||y||_2 = ||z||_2$, our problem becomes

$$\begin{aligned} &x \in \text{argmin}_z \{||z||_2 \,|Rz_1 + Sz_2 = c\} \\ &x \in \text{argmin}_z \left\{||z||_2 \left| z_1 = R^{-1}c - R^{-1}Sz_2\right.\right\} \\ &x \in \text{argmin}_z \sqrt{||R^{-1}c - R^{-1}Sz_2||_2^2 + ||z_2||_2^2} \\ &x \in \text{argmin}_z \left\{\left|\left|R^{-1}c - R^{-1}Sz_2\right|\right|_2^2 + ||z_2||_2^2\right\}, \end{aligned}$$

where the last equality is a consequence of the result proved in homework 2.30. Now, let $d = R^{-1}c$ and $P = R^{-1}S$. Then we can find the minimum of the above expression by differentiating and setting equal to zero:

$$0 = -P^T d + (P^T P + I) z_2 \rightarrow \tag{1.2}$$

$$z_2 = (P^T P + I)^{-1} P^T d. \tag{1.3}$$

Solution (Example ref(exm:und-least-squares)). We want to find $\min_z \left\{||z||_2 \left| z \in \text{argmin}_y ||Ay - b||_2\right.\right\}$ Use theorem 1.2:

$$\begin{aligned} \min_z \left\{||z||_2 \left| z \in \text{argmin}_y ||Ay - b||_2\right.\right\} &= \left\{||z||_2 \left| z \in \text{argmin}_y \left|\left| \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} \Pi^T y - Q^T b \right|\right|_2 \right.\right\} \\ &= \left\{||w||_2 \left| w \in \text{argmin}_y \left|\left| \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} y - Q^T b \right|\right|_2 \right.\right\}, \end{aligned}$$

since $||y||_2 = \left|\left|\Pi^T y\right|\right|_2$. This is exactly the problem solved in example 1.4. In conclusion,

$$w = \begin{bmatrix} R^{-1}(c_1 - Sy_y) \\ y_2 \end{bmatrix}.$$

Solution (Example ref(exm:constrained-least-squares)).

1.3.3 Existence of QR-decomposition.

To prove the existence of the QR-decomposition, we need the Gram-Schmidt process.

Lemma 1.8 (The Gram-Schmidt Process)**.** *Let $r \in \mathbb{N}$. Given a set of linearly independent vectors $\{a_1, \ldots, a_r\}$, there exists a set of orthonormal vectors $\{q_1, \ldots, q_r\}$ such that span$\{q_1, \ldots, q_r\}$ = span$\{a_1, \ldots, a_r\}$.*

The q_i's are given by...

Proof. We will prove this by induction. For $i = 1$: let $R_{11} = ||a_1||_2$, $q_1 = \frac{1}{R_{11}} a_1$. Notice that $||q_1|| = 1$.

(At this point, it might be beneficial to check out the intuitive side note (1.1))

Define q^r in the following way: let $R_{ir} = q_i' a_r$, $\tilde{q}_r = a_r - \sum_{i=1}^{r-1} R_{ir} q_i$, and $R_{rr} = ||\tilde{q}_r||_2$. Then $q_r = \frac{\tilde{q}_r}{R_{rr}}$. (Note: $\tilde{q}_r \neq 0$ since the a_is are linearly independent, and q_i is given as a linear combination of $a_1, \ldots, a_i$.)

Assume the result holds for $i \leq r-1$. I.e. we have vectors $q_1, \ldots, q_{r-1}$ given as above, and that

i) $\text{span}\{q_1, \ldots, q_{r-1}\} = \text{span}\{a_1, \ldots, a_{r-1}\}$,
ii) $q_i \cdot q_j$ for all $i, j = 1, \ldots, r-1$ with $i \neq j$,
iii) $q_i' \cdot q_i = 1$ for all $i = 1, \ldots, r-1$.

Now, we want to show that we can construct a q_r such that

a) $\text{span}\{q_1, \ldots, q_r\} = \text{span}\{a_1, \ldots, a_r\}$,
b) $q_r \cdot q_j = 0$ for all $j = 1, \ldots, r-1$,
c) $q_r' \cdot q_r = 1$.

We start from below.

c) By definition of q_r: $q_r' q_r = \frac{\tilde{q}_r' \tilde{q}_r}{R_{rr}^2} = \frac{||\tilde{q}_r||^2}{R_{rr}^2} = 1$.
d) Let $i < r$. Then

$$\begin{aligned} q_i' \tilde{q}_r &= q_i' a_r - \sum_{j=1}^{r-1} R_{jr} q_i' q_j \\ &= q_i' a_r - R_{ir} q_i' q_i \\ &= q_i' a_r - R_{ir} = 0 \text{ (by definition of } R_{ir}). \end{aligned}$$

a) We need to show that a_r can be written as a linear combination of q_is.

$$\begin{aligned} \sum_{i=1}^{r} R_{ir} q_i &= \sum_{i=1}^{r-1} R_{ir} q_i + R_{rr} q_r \\ &= \sum_{i=1}^{r-1} R_{ir} q_i + R_{rr} \frac{1}{R_{rr}} \tilde{q}_r \\ &= \sum_{i=1}^{r-1} R_{ir} q_i + R_{rr} \frac{1}{R_{rr}} \left(a_r - \sum_{i=1}^{r-1} R_{ir} q_i \right) \\ &= \sum_{i=1}^{r-1} R_{ir} q_i + a_r - \sum_{i=1}^{r-1} R_{ir} q_i \\ &= a_r. \end{aligned}$$

□

Remark 1.1 (Intuitive side note)**.** *It is fairly easy to find q_2. We want to find it such that $a_2 = R_{12} q_1 + R_{22} q_2$, and $||q_2||_2 = 1$ and $q_1 \perp q_2$, i.e. $q_1 \cdot q_2 = 0$. So, if we multiply the equation by q_1, we get that $q_1 a_2 = R_{12}$. Substituting this into the first equation, $q_2 = \frac{a_2 - R_{12} q_1}{R_{22}}$.*

Note that this is a circular argument, and hence not a formal way of doing this.

Lecture 5: 9/20

(Finished up proof of The Gram-Schmidt Process (1.8))

Remark 1.2 (Gram-Schmidt in Matrix Form)**.** *If we write up $a_1, \dots, a_r$ in a matrix, we see that*

$$\begin{bmatrix} a_1 & \dots & a_r \end{bmatrix} = \begin{bmatrix} q_1 & \dots & q_r \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1r} \\ 0 & R_{22} & \dots & R_{2r} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & R_{rr} \end{bmatrix}$$

This is quite similar to the result we are after (the QR-decomposition 1.2).

Proof of theorem ref(thm:qr-decomposition). Since $\operatorname{rank}(A) = r$, A has r linearly independent columns. Hence, there exists a permutation matrix Π such that

$$A\Pi = \begin{bmatrix} a_1 & \dots & a_r & a_{r+1} \dots a_m \end{bmatrix},$$

where $a_1, \dots, a_r$ are linearly independent, and $a_{r+1}, \dots, a_m$ are linearly dependent on the first r columns.

Using Gram-Schmidt (lemma 1.8), we know that there exists $\tilde{Q} \in \mathbb{R}^{n \times r}, R \in \mathbb{R}^{r \times r}$ such that $A\Pi = \tilde{Q}R$. Since $\operatorname{span}\{\tilde{q_1}, \dots, \tilde{q_r}\}$ (columns of $\tilde{Q}$) is equal to $\operatorname{span}\{a_1, \dots, a_r\}$, there exists an $s_{k(j-r+2)}$ for any $j \in \{r+1, \dots, m\}$ and $k \in \{1, \dots, r\}$ such that $a_j = \sum_{k=1}^{r} s_{k(j-r+2)} q_k$. So,

$$A\Pi = \tilde{Q} \begin{bmatrix} R & S \end{bmatrix}.$$

This is almost the form we want, BUT $\tilde{Q}$ is not orthonormal (it is not square). However, we know that we can pick $n - r$ vectors from $\mathbb{R}^n$ such that adding these as columns to $\tilde{Q}$ we get a set of n linearly independent columns. Now, use Gram-Schmidt to normalize. Since the first r columns are already normalized, these will stay the same. The result is a matrix Q, where the columns are all length 1, and they are all linearly independent. I.e. $Q^T Q = I$. So, $A\Pi = Q \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix}$, hence

$$A = Q \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} \Pi^T.$$

□

Basically, this gives us a way to perform QR decomposition. However, using the Gram-Schmidt procedure is NOT numerical stable. I.e. we might end up with matrices Q, R, and S from which we CANNOT recover A. To overcome this, there is a different method called the *Modified Gram-Schmidt Procedure.*

Lemma 1.9 (The Modified Gram-Schmidt Procedure)**.** ***HOMEWORK***

Definition 1.7 (Householder Reflections)**.** A matrix $H = I - 2vv'$, where $||v||_2 = 1$, is called a *Householder Reflection.*

A Householder reflection takes any vector and reflects it over $\{tv : t \in \mathbb{R}\}$.

Lemma 1.10. *Householder reflections are orthogonal matrices.*

Proof. By definition of a Householder matrix (1.7), $H = I - 2vv'$ for a v with $||v||_2 = 1$.

So,

$$\begin{aligned} H'H &= (I - 2vv')'(I - 2vv') \\ &= (I' - 2(vv')')(I - 2vv') \\ &= (I - 2vv')(I - 2vv') \\ &= I - 2vv' - 2vv' + 4vv'vv' \quad (\text{ recall: } v'v = ||v||_2 = 1) \\ &= I - 2vv' - 2vv' + 4vv' \\ &= I. \end{aligned}$$

So by definition (1.5), H is an orthogonal matrix. □

Lemma 1.11. *There exists Householder refelctions $H_1, \cdots, H_r$ such that $H_r \cdots H_1 \cdot A \cdot \Pi = R$.*

Proof. Let $A\Pi = \begin{bmatrix} a_1 & \cdots & a_r \end{bmatrix}$. Choose H_1 s.t. $H_1 a_1 = R_{11} e_1 = a_1 - 2v_1 v_1' a_1$ (last equality due to definition of Householder reflections). This is equivalent to $v_1(2v_1' a_1) = a_1 - R_{11} e_1$.

Now, let $v_1 = \frac{a_1 - R_{11} e_1}{||a_1 - R_{11} e_2||_2}$. Plug this into the equation for $R_{11} e_1$ above to get

$$R_{11} e_1 = a_1 - \frac{(a_1 - R_{11} e_1)}{||a_1 - R_{11} e_1||_2} \frac{a_1' a_1 - R_{11} a_1' e_1}{||a_1 - R_{11} e_1||_2}.$$

If we multiply this by e_1' from the right, we get

$$R_{11} = \pm ||a_1||_2, v_1 = \frac{a_1 - ||a_1|| e_1}{||a_1 - ||a_1||_2 e_1||_2}.$$

$$H_1 = I - \frac{a_1 - ||a_1||_2 e_1)((a_1 - ||a_1||_2 e_1)}{||a_1 - ||a_1||_2 e_1||_2^2}$$

□

Definition 1.8 (Givens Rotations)**.** A *Givens Rotation* is a matrix $G^{(i,j)}$ with entries (g_{ij}) such that

i) $g_{ii} = g_{jj} = \lambda$ (the ith and jth elements of the diagonal are λ).
ii) $g_{kk} = 1$ for all $k \notin \{i, j\}$. (all other diagonal elements are 1)
iii) $g_{ij} = -g_{ji} = \sigma$
iv) $g_{ij} = 0$ for all other pairs of i, j.

In words: $G^{(i,j)}$ is the identity matrix with the ith and jth diagonal elements made λ, and the entries at (i, j) and (j, i) are σ.

Example 1.8 (2x2 Givens rotation to create upper triangular matrix.)**.** The general $G^{(1,2)}$ is $\begin{bmatrix} \lambda & \sigma \\ -\sigma & \lambda \end{bmatrix}$. Let us consider a general $M \in \mathbb{R}^{2 \times m}$:

$$M = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1m} \\ M_{21} & M_{22} & \dots & M_{2m} \end{bmatrix}.$$

We want this Givens matrix to be orthogonal. By homework exercise 2.44, we know this is the case when $\lambda^2 + \delta^2 = 1$. We also want $G^{(1,2)} M$ to be an upper triangular matrix. Since

$$G^{(i,j)} M = \begin{bmatrix} \lambda M_{11} + \sigma M_{12} & \dots & \lambda M_{1m} + \sigma M_{2m} \\ -\sigma M_{11} + \lambda M_{12} & \dots & -\sigma M_{1m} + \lambda M_{2m} \end{bmatrix},$$

we need $\lambda M_{12} = \sigma M_{11}$. So, solving these two equations, we find that $\lambda = \frac{M_{11}}{\sqrt{M_{11}^2 + M_{21}^2}}$ and $\sigma = \frac{M_{21}}{\sqrt{M_{11}^2 + M_{21}^2}}$. Hence, given the matrix M, we can find a Givens rotation which when multiplied by M returns an upper triangular matrix.

This also means that given a vector $a \in \mathbb{R}^n$, we can find a Givens rotation such that $G^{(i,j)} a$ gives back a except for one entry, which has been changed to 0.

1.3.4 "Large" Data Problem

Finally, we will take a look at how to solve a "large" data problem using QR decomposition. To do so, let $A \in \mathbb{R}^{n\times m}$ be a matrix with n "big". By "big", we mean so large that A won't fit in memory, but the first $m+1$ rows of A will.

Gentleman published a few papers in 1973/1974 describing a method for incremental QR decomposition. The key ideas are as follows:

1) Remember that the solution to the linear model is $(A'A)^{-1}A'b = R^{-1}(Q'b)$.
2) The QR decomposition of $\begin{bmatrix} A & b \end{bmatrix}$ gives $\begin{bmatrix} R & Q'b \\ 0 & S \end{bmatrix}$ (ex. 2.34 and 2.35)

Now, if we do QR decomposition on the first $m+1$ rows of A we get $\begin{bmatrix} \tilde{R}_{m+1} & \tilde{Q}'_{m+1}b_{m+1} \\ 0 & S_{m+1} \end{bmatrix}$. Now, add the next row of A to get $\begin{bmatrix} \tilde{R}_{m+1} & \tilde{Q}'_{m+1}b_{m+1} \\ 0 & S_{m+1} \\ a_{m+2} & b_{m+2} \end{bmatrix}$. Hit this with the right Givens rotation to change entry $(m+1,1)$ to 0. This will give us something of the form $\begin{bmatrix} \tilde{R}_{m+2} & \tilde{Q}'_{m+2}b_{m+2} \\ 0 & S_{m+2} \end{bmatrix}$. Here, $\tilde{R}_{m+2}$ is still an upper triangular matrix with less than m rows. Repeat the procedure until we've added all rows of A/elements of b.

1.4 Singular Value Decomposition (SVD)

Outline:

I) Motivating problems
II) SVD and solutions
III) Existence and properties
IV) Random projections (a modern application of SVD)

1.4.1 Motivating Problems

Example 1.9. Let $A \in \mathbb{R}^{n\times m}$ with $\text{rank}(A) < m \leq n$, and let $B \in \text{range}(A)$. Find x s.t.

$$x \in \text{argmin}_{y\in\mathbb{R}^m} \{||y||_2 : Ay = b\}$$

Example 1.10. Let $A \in \mathbb{R}^{n\times m}$. Find

$$||A||_2 = \sup_{v\in\mathbb{R}^m\setminus\{0\}} \frac{||Av||_2}{||v||_2}.$$

Example 1.11. Let $A \in \mathbb{R}^{n\times m}$. Find x s.t.

$$x = \text{argmin}_{rank(Y)\leq k} ||A - Y||_F .$$

Note: $||A||_F$ is the *Frobenius norm* and is defined as $||A||_F = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2\right)^{1/2}$. So it is sort of a Euclideaon norm extended to matrices.

1.4.2 SVD

1.4.2.1 SVD definition

Theorem 1.3. *Suppose $A \in \mathbb{R}^{n\times m}$ and $n \geq m$. Then there exists $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{m\times m}$ that are both orthogonal, and a diagonal matrix $\Sigma \in \mathbb{R}^{n\times m}$ with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$ such that*

$$A = U\Sigma V'$$

Note:

- The diagonal elements of Σ are called the singular values of A.
- The columns of U are called the left singular vectors
- The columns of V are called the right singular vectors

Corollary 1.1. *rank(A) is the number of non-zero singular values of A.*

1.4.2.2 Solutions to motivating problems

Solution (Solution to example ref(exm:svd-problem-1)). Since $b \in$ range(A), there exists a y such that $Ay = b$. Now, by theorem 1.3 there exist U,V, and Σ such that $A = U\Sigma V'$. Since U is orthogonal, $U^{-1} = U'$. So,

$$\begin{aligned} U\Sigma V'y = b \Leftrightarrow \\ \Sigma V'y = U'b. \end{aligned}$$

Let $z = V'y$ and $c = U'b$. By corollary 1.1 we now that there are exactly rank(A) $= r$ non-zero singular values. So

$$\begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ 0 \end{bmatrix}.$$

So, $z_1 = \begin{bmatrix} \sigma_1^{-1} & & \\ & \ddots & \\ & & \sigma_r^{-1} \end{bmatrix} c_1$. We want to minimize $||y||_2 = ||V'z||_2 = ||z||_2 = \sqrt{||z_1||_2^2 + ||z_2||_2^2}$. This is done by setting $z_2 = 0$, which we can do since z_2 does not affect the equation above. (It is multiplied by all the 0 rows of Σ.) So, the minimum norm solution is

$$\begin{aligned} x &= V \begin{bmatrix} \sigma_1^{-1} & & & \\ & \ddots & & \\ & & \sigma_r^{-1} & \\ & & 0 & \end{bmatrix} c_1 \\ &= V \cdot V'y. \end{aligned}$$

For the solution to example 1.10, we'll need the following result:

Lemma 1.12. *Let D be a non-zero diagonal matrix of size $n \times m$, $n \geq m$. Then,*

$$||D||_2 = \max_i |D_{ii}|$$

.

Proof. Note that $||A||_2 = \max_{||v||_2=1} ||Av||_2$. Also, if we let $D_{ii} = \max_j |D_{jj}|$ and v be a vector with norm 1,

$$||Dv||_2^2 = \sum_{j=1}^{m} (D_{jj}^2 v_j)^2 \leq D_{ii}^2 \sum_{j=1}^{m} v_j^2 = D_{ii}^2 ||v||_2^2 = D_{ii}^2.$$

Now, let $z \in \mathbb{R}^m$ such that $z_{ij} = \begin{cases} 0, & j \neq i \\ \frac{D_{ii}}{|D_{ii}|}, & j = i \end{cases}$. Then $||z||_2 = \sqrt{\frac{D_{ii}^2}{|D_{ii}|^2}} = 1$, and

$$||Dz||_2^2 = \sum_{j=1}^{m} (D_{jj} z_j)^2 = D_{ii}^2 \frac{D_{ii}^2}{|D_{ii}|^2} = D_{ii}^2.$$

Since $||z||_2 = 1$,

$$D_{ii}^2 ||Dz||_2^2 \leq (\max_{||v||_2=1} |D_{ii}|)^2 \leq D_{ii}^2$$

,

so $||D||_2 = \max_{||v||_2=1} |D_{ii}| = |D_{ii}| = max_j |D_{jj}|$. □

Solution (Solution to example ref(exm:svd-problem-2)). First, use the SVD of A to write $A = U\Sigma V'$. Now, since the 2-norm is invariant under multiplication of orthogonal matrices, we have that

$$\begin{aligned} ||A||_2 &= \max_{||v||=2} ||U\Sigma V' v||_2 \\ &= \max_{||v||=2} ||\Sigma V' v||_2 . \end{aligned}$$

If we let $z = V'v$, we see that $||z||_2 = ||v||_2 = 1$, since V' is an orthogonal matrix. Hence,

$$||A||_2 = \max_{||z||_2=1} ||\Sigma z||_2 = ||\Sigma||_2 = \max_i |\sigma_i| = \sigma_1,$$

where we used the lemma above to obtain the second to last equality.
Solution (Solution to example ref(exm:svd-problem-3)). Recall, $A = \sum_{i=1}^{n} \sigma_i u_i v_i' = U\Sigma V'$. We want to find $\text{argmin}_{\text{rank}(Y) \leq k} ||A - Y||_F$.

Case 1: If $\text{rank}(A) \leq k$, then $Y = A$ is the solution.

Case 2: $\text{rank}(A) > k$. Since the Frobenius norm is orthogonally invariant, we can obtain that

$$\begin{aligned} ||A - Y||_F^2 &= ||U\Sigma V' - Y||_F^2 \\ &= ||\Sigma - U'YV||_F^2 . \end{aligned}$$

If we let $X = U'YV$, we get that

$$
\begin{aligned}
||A - Y||_F^2 &= \sum_{i=1}^{n}\sum_{j=1}^{m}(\sigma_{ij} - x_{ij})^2 \\
&= \sum_{i=1}^{n}\sum_{j=1,i\neq j}^{m}(-x_{ij})^2 + \sum_{i=1}^{n}(\sigma_i - x_{ij})^2,
\end{aligned}
$$

since $\sigma_{ij} = 0$ for all $i \neq j$, and at most k of the σ_{ii} are non-zero. To minimize the expression above, we choose Y such that $x_{ij} = 0$ for $i \neq j$, and $x_{ii} = \sigma_i$. Hence,

$$
X = \begin{bmatrix}
\sigma_1 & & & & & & \\
& \ddots & & & & & \\
& & \sigma_k & & & & \\
& & & 0 & & & \\
& & & & \ddots & & \\
& & & & & 0
\end{bmatrix},
$$

and $Y = UXV'$.

1.4.3 Existence and Properties

SVD (ref(thm:svd)). Recall that $||A||_2 = \sup_{v\neq 0} \frac{||Av||_2}{||v||_2} = \max_{||v||_2=1} ||Av||_2$. I.e. there exists a v_1 such that $||v_1||_2 = 1$ and $||A||_2 = ||Av_1||_2$.

Let u_1 be a vector such that $||A||_2 u_1 = Av_1$. This implies that $||u_1||_2 = 1$, since $||||A||_2 u_1||_2 = ||A||_2 ||u_1||_2 = ||Av_1||_2 = ||A||_2$.

Let $\sigma_1 = ||A||_2$. So, $\sigma_1 u_1 = Av_1$. Using the Gram-Schmidt procedure, we can create a matrix $V_1 \in \mathbb{R}^{n\times(n-1)}$ and $\tilde{U}_1 = \begin{bmatrix} u_1 V_1 \end{bmatrix} \in \mathbb{R}^{n\times n}$ being an orthogonal matrix. Similarly, we can create $\tilde{V}_1 \in \mathbb{R}^{m\times m}$ such that $\tilde{V}_1 = \begin{bmatrix} v_1 V_1 \end{bmatrix} \in \mathbb{R}^{m\times m}$ is orthogonal.

Now, $A\tilde{V}_1 = \begin{bmatrix} Av_1 & AV_1 \end{bmatrix} = \begin{bmatrix} \sigma_1 u_1 & AV_1 \end{bmatrix}$. Hence, $\tilde{U}_1' A \tilde{V}_1 = \begin{bmatrix} u_1 \\ U_1' \end{bmatrix} \begin{bmatrix} \sigma_1 u_1 & AV_1 \end{bmatrix} = \begin{bmatrix} \sigma_1 & w' \\ 0 & \tilde{A} \end{bmatrix}$, since u_1 is orthogonal to all columns of U_1'. We need to show that $w = 0$. Since the 2-norm is orthogonally invariant, it holds for any $z \neq 0$,

$$
\begin{aligned}
\sigma_1^2 &= ||A||_2^2 \\
&= \left|\left|\tilde{U}_1' A \tilde{V}_1\right|\right|_2^2 \\
&= \left|\left|\begin{bmatrix} \sigma_1 & w' \\ 0 & \tilde{A} \end{bmatrix}\right|\right|_2^2 \\
&\geq \frac{\left|\left|\begin{bmatrix} \sigma_1 & w' \\ 0 & \tilde{A} \end{bmatrix} z\right|\right|_2^2}{||z||_2^2}.
\end{aligned}
$$

Let $z = \begin{bmatrix} \sigma_1 \\ w \end{bmatrix}$. Then,

$$\sigma_1^2 \geq \frac{\left|\left|\begin{bmatrix}\sigma_1 & w'\\ 0 & \tilde{A}\end{bmatrix}\begin{bmatrix}\sigma_1\\ w\end{bmatrix}\right|\right|_2^2}{\left|\left|\begin{bmatrix}\sigma_1\\ w\end{bmatrix}\right|\right|_2^2} = \frac{(\sigma_1^2+||w||_2^2)^2 + \left|\left|\tilde{A}w\right|\right|_2^2}{\sigma_1^2+||w||_2^2} \geq \sigma_1^2+||w||_2^2 \geq \sigma_1^2,$$

i.e. $||w||_2^2 = 0$. So,

$$\tilde{U}_1 A V_1 = \begin{bmatrix}\sigma_1 & 0\\ 0 & \tilde{A}\end{bmatrix}.$$

Repeat the same procedure to get U, V such that $A = U'\Sigma V$. □

Corollary 1.2. *rank(A) is exactly the number of non-zero singular values of A.*

Proof. $\text{rank}(A) = \text{rank}(U\Sigma V') = \text{rank}(\Sigma) =$ number of non-zero diagonal elements. □

Corollary 1.3. *Let $A, E \in \mathbb{R}^{n\times m}$, σ_{max} (σ_{min}) denote the largest (smallest) singular value of A. Then*

$$\sigma_{max}(A+E) \leq \sigma_{max}(A) + ||E||_2$$

and

$$\sigma_{min}(A+E) \geq \sigma_{min}(A) - ||E||_2 .$$

Corollary 1.4 (Hoffman-Wielandt Inequality)**.** *Let $A, E \in \mathbb{R}^{n\times m}$, $\sigma_k(\dot{})$ denote the k'th largest singular value. Let $p = \min(m,n) \leq ||E||_F^2$. Then*

$$\sum_{k=1}^{p} (\sigma_k(A+E) - \sigma_k(A))^2 \leq ||E||_F^2 .$$

Corollary 1.5. *Let $r = rank(A)$.*

1) *range(A) = span$(u_1, \ldots, u_r)$*
2) *row space of A = span$(v_1, \ldots, v_r)$*
3) *null(A) = span$(v_{r+1}, \ldots, v_m)$*
4) *null(A′) = span$(u_{r+1}, \ldots, u_m)$*

Moreover,

1) $U_r = \begin{bmatrix}u_1 & \cdots & u_r\end{bmatrix}$, $U_r U_r' = P_{range(A)}$
2) $U_{n-r} = \begin{bmatrix}u_{r+1} & \cdots & u_n\end{bmatrix}$, $U_{n-r}U_{n-r}' = P_{null(A)}$
3) $V_r = \begin{bmatrix}v_1 & \cdots & v_r\end{bmatrix}$, $V_r V_r' = P_{row(A)}$
4) $V_{n-r} = \begin{bmatrix}v_{r+1} & \cdots & v_n\end{bmatrix}$, $V_{n-r}V_{n-r}' = P_{null(A)}$

where $P_{\mathcal{B}}$ is the projection onto the space $\mathcal{B}$.

Definition 1.9 (Pseudo-inverse)**.** For a matrix $A \in \mathbb{R}^{n\times m}$, we call $A^+ \in \mathbb{R}^{m\times n}$ a pseudo-inverse to A if

1) $AA^+A = A$
2) $A^+AA^+ = A^+$
3) $(AA^+)' = AA^+$
4) $(A^+A)' = A^+A$

Theorem 1.4. *For any matrix $A \in \mathbb{R}^{n\times m}$, there exists a unique pseudo-inverse A^+.*

Proof. Suppose B and C are both pseudo-inverses of A. Then

$$\begin{aligned} BA &= B(ACA) \\ &= (BA)(CA) \\ &= (BA)'(CA)' \\ &= A'B'A'C' \\ &= (ABA)'C' \\ &= A'C' \\ &= (CA)' \\ &= CA. \end{aligned}$$

Similarly,

$$\begin{aligned} AB &= (ACA)B \\ &= (AC)(AB) \\ &= (AC)'(AB)' \\ &= C'A'B'A' \\ &= C'(ABA)' \\ &= C'A' \\ &= (AC)' \\ &= AC. \end{aligned}$$

So, $B = BAB = CAB = CAC = C$. This shows that if there exists a pseudo-inverse, it is unique.

Now, let $D \in \mathbb{R}^{n\times m}$ be a diagonal matrix with d_{ii} its diagonal elements. Then the diagonal matrix $E \in \mathbb{R}^{m\times n}$ with diagonal elements d_{ii}^{-1} for $d_{ii} \neq 0$ and 0 otherwise is the pseudo-inverse of D.

So,

$$[DE]_{ij} = \sum_{k=1}^{m} D_{ik}E_{kj} = \begin{cases} 0, & i \neq j \\ 1, & i = j, d_{ii} \neq 0 \\ 0, & i = j, d_{ii} = 0 \end{cases}$$

and

$$[ED]_{ij} = \sum_{k=1}^{n} D_{ik}E_{kj} = \begin{cases} 0, & i \neq j \\ 1, & i = j, d_{ii} \neq 0 \\ 0, & i = j, d_{ii} = 0 \end{cases}$$

Since ED and DE are both diagonal matrices, $(ED)' = ED$ and $(DE)' = DE$.

Furthermore,

$$[DED]_{ij} = \sum_{k=1}^{n} D_{ik}[ED]_{kj} = \begin{cases} 0, & i \neq j \\ D_{ij}, & i = j \end{cases}$$

and

$$[EDE]_{ij} = \sum_{k=1}^{n} E_{ik}[DE]_{kj} = \begin{cases} 0, & i \neq j \\ E_{ij}, & i = j \end{cases}$$

So E satisfies the four conditions of a pseudo-inverse. I.e. for any diagonal matrix, there exists a pseudo-inverse.

Now, we know that there exist U, Σ, V such that $A = U\Sigma V'$. Let $B = V\Sigma^+U'$. Recall that $U'U = I$ and $V'V = I$.

1) $ABA = U\Sigma V'V\Sigma^+U'U\Sigma V' = U\Sigma\Sigma^+\Sigma V' = U\Sigma V' = A$

2) $BAB = V\Sigma^+U'U\Sigma V'V\Sigma^+U = V\Sigma^+\Sigma\Sigma^+U' = V\Sigma^+U' = B$

3)

$$\begin{aligned}(AB)' = B'A' &= (V\Sigma^+U')'(U\Sigma V')' \\ &= U(\Sigma^+)'V'V\Sigma'U' \\ &= U(\Sigma\Sigma^+)'U' \\ &= U\Sigma\Sigma^+U' \\ &= U\Sigma V'V\Sigma^+U' \\ &= AB\end{aligned}$$

4)

$$\begin{aligned}(BA)' = A'B' &= (U\Sigma V')'(V\Sigma^+U')' \\ &= V\Sigma'U'U(\Sigma^+)'V' \\ &= V(\Sigma^+\Sigma)'V' \\ &= V\Sigma^+\Sigma V' \\ &= V\Sigma^+U'U\Sigma V' \\ &= BA.\end{aligned}$$

So, B is a pseudo-inverse of A.

Hence, any matrix has a unique pseudo-inverse. □

1.4.4 Random Projections

Let $A \in \mathbb{R}^{n\times m}$, $m >> n$. When this is the case, it is very hard to compute the SVD. So instead we try to find a matrix C with range$(C) \approx$ range(A). In other words, $A \approx P_{\text{range}(C)}A$. One way to find such a C is to simply sample columns from A.

Theorem 1.5. *Suppose $A \in \mathbb{R}^{n\times m}$, C is as in the algorithm `inexactRankK` (see here), and H is the k left singular vectors of C. Then*

$$||A - HH'A||_F^2 \leq ||A - A_k||_F^2 + 2\sqrt{k}\,||AA' - CC'||_F\,,$$

where A_k is the rank k approximation of A.

Proof. Recall that $||A||_F^2 = \text{tr}(A'A)$, and $H'H = I$, since the columns of H are singular vectors, i.e. are orthogonal to each other.

$$\begin{aligned}
||A - HH'A||_F^2 &= \text{tr}((A - HH'A)'(A - HH'A)) \\
&= \text{tr}((A' - A'HH')(A - HH'A)) \\
&= \text{tr}(A'A - A'HH'A - A'HH'A + A'HH'HH'A) \\
&= \text{tr}(A'A - A'HH'A) \\
&= \text{tr}(A'A) - \text{tr}(A'HH'A) \\
&= ||A||_F^2 - ||A'H||\,.
\end{aligned}$$

Since

$$||A||_F^2 = ||A - A_k||_F^2 + \sum_{j=1}^{k} \sigma_j^2(A),$$

we have that

$$\begin{aligned}
||A - HH'A||_F^2 &= ||A - A_k||_F^2 + \sum_{j=1}^{k} \sigma_j^2(A) - ||A'H||_F^2 \\
&= ||A - A_k||_F^2 + \sum_{j=1}^{k} \left[\sigma_j^2(A) - \sigma_j^2(C)\right] + \sum_{j=1}^{k} \sigma_j^2 C - ||A'H||_F^2\,.
\end{aligned}$$

Note that, using the Cauchy-Schwarts inequality, we get

$$\begin{aligned}
\sum_{j=1}^{k} \left[\sigma_j^2(A) - \sigma_j^2(C)\right] &= \left| \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \sigma_1^2(A) - \sigma_1^2(C) & \cdots & \sigma_k^2(A) - \sigma_k^2(C) \end{bmatrix} \right| \\
&\leq \sqrt{\left|\left| \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right|\right| \left|\left| \begin{bmatrix} \sigma_1^2(A) - \sigma_1^2(C) & \cdots & \sigma_k^2(A) - \sigma_k^2(C) \end{bmatrix} \right|\right|} \\
&= \sqrt{k} \sqrt{\sum_{j=1}^{k} \left(\sigma_j^2(A) - \sigma_j^2(C)\right)^2} \\
&\leq \sqrt{k} \sqrt{\sum_{j=1}^{k} \left(\sigma_j(AA') - \sigma_j(CC')\right)^2} \\
&= \sqrt{k} \sqrt{\sum_{j=1}^{k} \left(\sigma_j(CC' + AA' - CC') - \sigma_j(CC')\right)^2} \\
&\leq \sqrt{k}\, ||AA' - CC'||_F\,.
\end{aligned}$$

If $H_1, \ldots, H_k$ denote columns of H, then $A'H = \begin{bmatrix} A'H_1 & \cdots & A'H_k \end{bmatrix}$. Since

$$||A'H||_F^2 = \sum_{j=1}^{k} ||A'H_j||_2^2\,,$$

we can use Cauchy-Schwartz as above to obtain that

$$\begin{aligned}\sum_{j=1}^{k}\sigma_j^2(C) - ||A'H||_F^2 &\leq \sqrt{k}\sqrt{\sum_{j=1}^{k}\left[\sigma_j CC' - H_j'AA'H_j\right]^2} \\ &= \sqrt{k}\sqrt{\sum_{j=1}^{k}\left[H_j'CC'H_j - H_j'AA'H_j\right]^2} \\ &\leq \sqrt{k}\,||AA' - CC'||_F\,,\end{aligned}$$

where we in the last step use the Hoffman-Wielandt inequality (corollary 1.4).

Combining all of this, we get

$$||A - HH'A||_F^2 \leq ||A - A_k||_F^2 + 2\sqrt{k}\,||AA' - CC'||_F\,.$$

□

1.5 Iterative Methods

1.5.1 Overview

First, some references:

a) Golub and Van Loan
b) Saad (2000): *Iterative methods for sparse linear systems* link
c) Shewchuk (1994): *Introduction to Conjugated Gradients without the Agonizing Pain* link
d) Gower and Richtarik (2015): *Randomized Iterative Methods for Linear Systems* link

1.5.2 Outline

We will be going throuh the following:

i) Why iterative methods?
ii) Splitting Methods
iii) Randomized Kaezmarz Method
iv) Gradient Descent
v) Conjugated Gradients
vi) GMRES (optional)

1.5.3 Motivation

These methods have had a great impact historically speaking. They take advantage of any sparsity found in a matrix, making them "easier" computational.

When working with huge systems (such as weather prediction, computational chemistry, genomics, etc), this is the way to deal with it.

1.5.4 Splitting Methods

Let A be a matrix, and b a vector. Let

- x^c denote the current iterate

- x^+ denote the next iterate
- r^c denote the current residual
- r^+ denote the next residual

where $r^c = Ax^c - b$ and $r^+ = Ax^+ - b$. The goal is to find methods to minimize the residual by iteratively defining x^+ based on x^c.

A splitting method takes a matrix A and splits it. One way of doing so is by splitting A into three parts $A = D - E - F$, where

- D is the diagonal
- E is the negative lower triangular part of A excluding the diagonal
- F is the negative upper triangular part of A excluding the diagonal

Based on this split, there are a few different ways to iteratively update r^+.
Definition 1.10 (The Jacobi Method)**.** *The Jacobi Method* finds x^+ from x^c using the following rule:

$$x_i^+ = \frac{b_i - \sum_{k \neq i} A_{ik} x_k^c}{A_{ii}}.$$

If we do this for all i, we can write this in matrix form:

$$x^+ = D^{-1}(b + (E + F)x^c).$$

For the Jacobi Method, we can see that $r_i^+ = 0$:

$$\begin{aligned} x_i^+ = \frac{b_i - \sum_{k \neq i} A_{ik} x_k^c}{A_{ii}} &\iff \\ A_{ii} x_i^+ = b_i - \sum_{k \neq i} A_{ik} x_k^c &\iff \\ 0 = b_i - \sum_{k \neq i} A_{ik} x_k^c - A_{ii} x_i^+ = b_i - \sum_{k} A_{ik} x_k^c = r_i^+. \end{aligned}$$

Definition 1.11 (The Gauss-Seidel Method)**.** *The Gauss-Seidel Method* finds x^+ from x^c using the following rule:

$$x_i^+ = \frac{1}{A_i i} \left(b_i - \sum_{k=1}^{i-1} A_{ik} x_k^+ - \sum_{k=i+1}^{d} A_{ik} x_k^c \right),$$

which in matrix formulation is

$$x^+ = D^{-1}(b + Ex^+ + Fx^c)$$

,

or equivalently

$$x^+ = (D - E)^{-1}(b + Fx^c).$$

As for the Jacobi method, it can be seen that $r_i^+ = 0$.

Definition 1.12 (Successive Over Relaxation (SOR))**.** When we update x^c to x^+ using a rule of the form

$$(D - \omega E)x^+ = (\omega F + (1 - \omega)D)x^c + \omega b,$$

it is called a *Successive Over Relaxation* method.

Note: if we pick $\omega = 1$, we get back the Gauss-Seidel method.
Definition 1.13 (Backward SOR)**.** The following is called the *Backward SOR*:

$$(D - \omega E)x^+ = (\omega F + (1 - \omega)D)x^c + \omega b,$$

Definition 1.14 (Symmetric SOR)**.** The *Symmetric SOR* is a method where we first find z based on the rule

$$(D - \omega F)z = (\omega F + (1 - \omega)D)x^c + \omega b$$

before using the rule

$$(D - \omega F)x^+ = (\omega F + (1 - \omega)D)z + \omega b$$

to find x^+.

1.5.5 Convergence

All of the schemes mentioned above are of the form $x^+ = Gx^c + f$:

- **Jacobi**: $G = D^{-1}(E + F)$, $f = D^{-1}b$.
- **Gauss-Seidel**: $G = (D - E)^{-1}F$, $f = (D - E)^{-1}b$.
- **SOR**: $G = (D - \omega E)^{-1}(\omega F + (1 - \omega)D)$, $f = \omega(D - \omega E)^{-1}b$.

Now, if we assume that x^* is a vector such that $Ax^* = b$, then $(I - G)x^* = f$: (for the following, keep in mind that $A = D - E - F$)

- **Jacobi**:

$$\begin{aligned} f &= D^{-1}b \\ &= D^{-1}Ax^* \\ &= D^{-1}(D - E - F)x^* \\ &= (DD^{-1} - D^{-1}(E + F))x^* \\ &= (I - G)x^*. \end{aligned}$$

- **Gauss-Seidel**:

$$\begin{aligned} f &= (D - E)^{-1}b \\ &= (D - E)^{-1}Ax^* \\ &= (D - E)^{-1}(D - E - F)x^* \\ &= ((D - E)^{-1}(D - E) - (D - E)^{-1}F)x^* \\ &= (I - G)x^*. \end{aligned}$$

- **SOR**:

$$\begin{aligned}
f &= \omega(D-\omega E)^{-1}b \\
&= \omega(D-\omega E)^{-1}Ax^* \\
&= \omega(D-\omega E)^{-1}(D-E-F)x^* \\
&= (D-\omega E)^{-1}(\omega D-\omega E-\omega F)x^* \\
&= (D-\omega E)^{-1}(D-D+\omega D-\omega E-\omega F)x^* \\
&= (D-\omega E)^{-1}(D-\omega E-[(1-\omega)D+\omega F])x^* \\
&= ((D-\omega E)^{-1}(D-\omega E)-(D-\omega E)^{-1}[D(1-\omega)+\omega F])x^* \\
&= (I-G)x^*.
\end{aligned}$$

Lemma 1.13. *For the Jacobi, Gauss-Seidel, and SOR methods, if there exists a x^* s.t. $Ax^* = b$, then $x^+ - x^* = G(x^c - x^*)$.*

Proof. We just saw that $Ax^* = b$ implies $(I-G)x^* = f$. We also saw that $x^+ = Gx^c + f$. Hence

$$\begin{aligned}
x^+ &= Gx^c + f \\
&= Gx^c + (I-G)x^* \\
&= Gx^c + x^* - Gx^* \iff \\
x^+ - x^* &= G(x^c - x^*).
\end{aligned}$$

□

Theorem 1.6. *Suppose there exists a x^* s.t. $Ax^* = b$. Let x_0 be arbitrary and define a sequence $\{x_k\}_{k\in\mathbb{N}}$ by*

$$x_k = Gx_{k-1} + f.$$

If $\rho(G) < 1$, then x^* is unique, and $x_k \to x^*$ as $k \to \infty$.

In the theorem above, $\rho(G)$ is the spectral radius of the matrix G. This is defined as the largest eigenvalue of the matrix G.

Definition 1.15 (Jordan Canonical Form)**.** Let $A \in \mathbb{R}^{n\times n}$. There exists an X which is invertible, and a block-diagonal matrix J, whose blocks are of the form $\lambda I + E$, where λ is an eigenvalue of A, and

$$E = \begin{cases} 1, & j = i+1 \\ 0, & \text{otherwise} \end{cases}$$

such that $A = XJX^{-1}$.

1.5.6 Randomized Kaczmarz Method

Consider $Ax = b$. Let the rows of A be denoted as $a_1', a_2', \ldots, a_n'$, and $b = (b_1, \ldots, b_n)'$. Now, the goal is to choose the next iteration of x such that $r^+ = b_i - a_i'x^+ = 0$. Let's say we try to set $x^+ = x^c + v$ for some appropriate v. What should v be then? We want $b_i = a_i'(x^c + v)$, which would imply that $a_i'v = b_i - a_i'x^c$. This has many possible solutions. So let's look for a particular one, namely one that is proportional to a_i: $v = \alpha a_i$. Then $\alpha\,||a_i||_2^2 = b_i - a_i'x^c$ which implies $\alpha = \frac{b_i - a_i'x^c}{||a_i||_2^2}$.

The above approach gives us Kaczmarz Method: the next iterate of x is $x^+ = x^c + \frac{b_i - a_i'x^c}{||a_i||_2^2}a_i$.

Now, the *Randomized* Kaczmarz Method is the one where we randomly choose which column to use for the next iteration based on the probability distribution over all columns of A given by $P(i=l) = \frac{||a_l||_2^2}{||A||_F^2}$. The following theorem guarantees us that this approach actually works, i.e. we converge towards the solution of $Ax = b$.

Theorem 1.7 (Randomized Kaczmarz Method)**.** *Suppose $A \in \mathbb{R}^{n\times m}$ is invertible and $x^* = A^{-1}b$.*

Given a sequence of i.i.d. random variables $i_1, i_2, \ldots$ ($P(i_k = l) = \frac{||a_l||_2^2}{||A||_F^2}$), and x_0 an arbitrary initial value, define

$$x_k = x_{k-1} + \frac{b_{i_{k-1}} - a'_{i_{k-1}} x_{k-1}}{||a_{i_{k-1}}||_2^2} a_{i_{k-1}}$$

for all $k > 1$. Then

$$E\left[||x_k - x^*||_2^2\right] \leq \left(1 - \frac{\sigma_{min}(A)^2}{||A||_F^2}\right)^k ||x_0 - x^*||_2^2 .$$

Note that the above result (by Chebyshev) implies

$$P\left[||x_k - x^*||_2 > \epsilon\right] \leq \frac{1}{\epsilon^2} E\left[||x_k - x^*||_2^2\right]$$

which in turn implies $x_k \to_p x^*$ as $k \to \infty$.

Proof. First of all, note that since i_k is independent of i_l for all $l < k$, i_k is independent of all x_l for all $l < k$.
Observation 1: $x_{k+1} - x_k$ is orthogonal to $x_{k+1} - x^*$. Therefore

$$||x_{k+1} - x^*||_2^2 = ||x_k - x^*||_2^2 - ||x_{k+1} - x_k||_2^2 .$$

Since $b_{i_k} = a'_{i_k} x^*$, we have that

$$\begin{aligned} x_{k+1} - x_k &= \frac{b_{i_k} - a'_{i_k} x_k}{||a_{i_k}||_2^2} a_{i_k} \\ &= \frac{a'_{i_k} x^* - a'_{i_k} x_k}{||a_{i_k}||_2^2} a_{i_k} \\ &= \frac{a'_{i_k}(x^* - x_k)}{||a_{i_k}||_2^2} a_{i_k} . \end{aligned}$$

So

$$\begin{aligned} ||x_{k+1} - x_k||_2^2 &= ||a_{i_k}||_2^2 \frac{[a'_{i_k}(x_k - x^*)]^2}{||a_{i_k}||_2^4} \\ &= \frac{[a'_{i_k}(x_k - x^*)]^2}{||a_{i_k}||_2^2} . \end{aligned}$$

Observation 2: For any vector $y \in \mathbb{R}^n$ it holds that

$$E\left(\frac{(a'_{i_0} y)^2}{||a_{i_0}||_2^2}\right) \geq \frac{\sigma_{min}^2(A)}{||A||_F^2} ||y||_2^2 .$$

"Proof:"
Recall that $||A||_2^2 = \max_{y\neq 0} \frac{||Ay||_2^2}{||y||_2^2} = \sigma_{max}(A)^2 >= \sigma_{min}(A)^2$. With this in mind

$$\sum_{i=1}^{n}(a_i'y)^2 = ||Ay||_2^2 \geq \sigma_{min}(A)^2\,||y||_2^2\,.$$

Divide through by $||A||_F^2$ to get $\sum_{i=1}^{n}\frac{1}{||A||_F^2}(a_i'y)^2 \geq \frac{\sigma_{min}(A)^2}{||A||_F^2}\,||y||_2^2$, before we simply multiply each term in the sum by $1 = \frac{||a_i||_2^2}{||a_i||_2^2}$. So

$$\begin{aligned}\frac{\sigma_{min}(A)^2}{||A||_F^2}\,||y||_2^2 &\leq \sum_{i=1}^{n}\frac{||a_i||_2^2}{||A||_F^2}\frac{(a_i'y)^2}{||a_i||_2^2}\\ &= \sum_{i=1}^{n}P(i_0=i)\frac{(a_i'y)^2}{||a_i||_2^2}\\ &= E\left(\frac{(a_{i_0}'y)^2}{||a_{i_0}||_2^2}\right).\end{aligned}$$

Now, using these two observations together:

$$\begin{aligned}E\left[||x_{k+1}-x^*||_2^2\,|x_k\right] &= E\left[||x_k-x^*||_2^2\,|x_k\right] - E\left[||x_{k+1}-x_k||_2^2\,|x_k\right]\\ &= ||x_k-x^*||_2^2 - E\left[\frac{(a_{i_k}'(x_k-x^*))^2}{||a_{i_k}||_2^2}\right]\\ &\leq ||x_k-x^*||_2^2 - \frac{\sigma_{min}(A)^2}{||A||_F^2}\,||x_k-x^*||_2^2\\ &= (1-\frac{\sigma_{min}(A)^2}{||A||_F^2})\,||x_k-x^*||_2^2\,.\end{aligned}$$

Finally, taking expectation on both sides gives us that $E\left[||x_{k+1}-x^*||_2^2\right] \leq \left(1-\frac{\sigma_{min}(A)^2}{||A||_F^2}\right)E\left[||x_k-x^*||_2^2\right]$. This upper bound holds for any k, so we can repeatedly use this to get

$$E\left[||x_{k+1}-x^*||_2^2\right] \leq \left(1-\frac{\sigma_{min}(A)^2}{||A||_F^2}\right)^k ||x_0-x^*||_2^2$$

since $E[||x_0-x^*||_2^2] = ||x_0-x^*||_2^2$. □

1.5.7 Gradient Descent

The Gradient Descent iterative method updates the current x^c to x^+ by

$$x^+ = x^c + \alpha\sum_{i=1}^{n}a_i(b_i - a_ix^c) \tag{1.4}$$

$$= x^c + \alpha A'(b-Ax^c) \tag{1.5}$$

The question now is: how do we choose α?

Strategy 1

Choose the α that minimizes $||r^+||_2^2 = ||Ax^+-b||_2^2$.

Lemma 1.14.

$$argmin_\alpha \left|\left|r^+\right|\right|_2^2 = \frac{||A'r^c||_2^2}{||AA'r^c||_2^2}$$

Proof. First, note that by homework exercise 2.66, $A'r^c \neq 0$, since if this were the case, the problem would be solved already.

$$\left|\left|r^+\right|\right|_2^2 = ||r^c||_2^2 - 2\alpha \, ||A'r^c||_2^2 + \alpha^2 \, ||AA'r^c||_2^2,$$

since $r^+ = Ax^+ - b = A(x^c + \alpha A'(b - Ax^c)) - b = Ax^c - b - \alpha AA'(Ax^c - b) = r^c - \alpha AA'r^c$, and $||r^+||_2^2 = (r^+)'r^+$. Differentiate this equation with respect to α, equate to 0, and solve for α will yield the result. □

Theorem 1.8. *Suppose A is invertible. Given $x_0\mathbb{R}^n$, let x_k be defined as*

$$x_k = x_{k-1} + \alpha_{k-1}A'(b - Ax_{k-1}),$$

where

$$\alpha_k = \frac{||A'r_k||_2^2}{||AA'r_k||_2^2}.$$

Then

$$||r_k||_2^2 \leq \left(1 - \frac{4\sigma_1^2\sigma_n^2}{\sigma_1^2 + \sigma_n^2}\right)^k ||r_0||_2^2 .$$

Proof.

$$\begin{aligned} ||r_{k+1}||_2^2 &= ||r_k||_2^2 - \alpha_k 2 \, ||A'r_k||_2^2 + \alpha_k^2 \, ||AA'r_k||_2^2 \\ &= ||r_k||_2^2 - \frac{||A'r_k||_2^4}{||AA'r_k||_2^2} \\ &= ||r_k||_2^2 \left(1 - \frac{||A'r_k||_2^4}{||r_k||_2^2 \, ||AA'r_k||_2^2}\right). \end{aligned}$$

If we let $A = U\Sigma V'$ be the SVD of A, then

$$\begin{aligned} 1 - \frac{||A'r_k||_2^4}{||r_k||_2^2 \, ||AA'r_k||_2^2} &= 1 - \frac{||\Sigma U'r_k||_2^4}{||U'r_k||_2^2 \, ||\Sigma\Sigma'U'r_k||_2^2} \\ &= 1 - \frac{||w||_2^4}{||\Sigma^{-1}w||_2^2 \, ||\Sigma w||_2^2}. \end{aligned}$$

We want to find an upper bound on $\left|\left|\Sigma^{-1}u\right|\right|_2^2 ||\Sigma u||_2^2$ for any unit vector w.

$$\begin{aligned} \left|\left|\Sigma^{-1}u\right|\right|_2^2 ||\Sigma u||_2^2 &= \left(\sum_{i=1}^k \frac{u_i^2}{\sigma_i^2}\right)\left(\sum_{i=1}^n \sigma_i^2 u_i^2\right) \\ &\leq \frac{(\sigma_1^2 + \sigma_n^2)^2}{4\sigma_1^2\sigma_n^2}, \end{aligned}$$

where the last inequality is the Kontorovich inequality (see homework 2.68). □

Strategy 2

Choose $\alpha = \text{argmin}_\alpha \, ||x_{k+1} - x^*||_2^2$.

Strategy 3

Note that $||r^+||_2 = ||(I - \alpha AA')r^c||_2 \leq ||I - \alpha AA'||_2 \, ||r^c||_2$.

Choose $\alpha = \text{argmin}_\alpha \, ||I - \alpha AA'||_2$.

Strategy 4

Note that $||x_{k+1} - x^*||_2 \leq ||I - \alpha AA'||_2 \, ||x_k - x^*||_2$.

Choose $\alpha = \text{argmin}_\alpha \, ||I - \alpha AA'||_2 \, ||x_k - x^*||_2$.

Chapter 2

Homework Assignments

2.1 Homework 1

Exercise 2.1. Can all nonnegative real numbers be represented in such a manner (i.e. as a fp number) for an arbitrary base $\beta \in \{2, 3, ...\}$?
Solution. No. For any given β and a largest exponent e_{max}, any decimal larger than $\beta \cdot \beta^{e_{max}}$ is larger than the largest number possibly representated.
Exercise 2.2. Suppose $e = -1$, what are the range of numbers that can be represented for an arbitrary base $\beta \in \{2, 3, ...\}$?
Solution. The smallest number that can be represented for an arbitrary base must be $(0 + 0 \cdot \beta^{-1} + ... + 0 \cdot \beta^{-(p-1)}) \cdot \beta^{-1}$.

Since $0 \leq d_i < \beta, \forall i$, the largest value must be attained when $d_i = \beta - 1$ for all i. I.e. the largest value must be

$$\begin{aligned} MAX &= (\beta - 1 + (\beta - 1)\beta^{-1} + ... + (\beta - 1)\beta^{-(p-1)}) \cdot \beta^{-1} \\ &= (1 + \beta^{-1} + ... + \beta^{-(p-1)})(\beta - 1) \cdot \beta^{-1} \\ &= (1 + \beta^{-1} + ... + \beta^{-(p-1)}) \cdot (1 - \beta^{-1}) \\ &= (1 + \beta^{-1} + ... + \beta^{-(p-1)}) \cdot (1 - \beta^{-1}) \end{aligned}$$

Exercise 2.3. Characterize the numbers that have a unique representation in a base $\beta \in \{2, 3, ...\}$.
Solution. Let

$$f = \left(d_1 \cdot \beta^{-1} + \ldots + d_{p-1} \cdot \beta^{-(p-1)}\right) \cdot \beta^e,$$

i.e. f is not normarlized. Then,

$$f = \left(d_1 + d_2\beta^{-1} + \ldots + d_{p-1} \cdot \beta^{-p} + 0 \cdot \beta^{-(p-1)}\right) \cdot \beta^{e-1}.$$

So, non-normalized fp numbers are NOT unique.

Now, let f be a normalized fp number. I.e.

$$f = \left(d_0 + d_1 \cdot \beta^{-1} + \ldots + d_{p-1} \cdot \beta^{-(p-1)}\right) \cdot \beta^e,$$

where $d_0 \neq 0$. If we let $e_n < e$, then

$$f > \left(d_0 + d_1 \cdot \beta^{-1} + \ldots + d_{p-1} \cdot \beta^{-(p-1)}\right) \cdot \beta^{e_n},$$

and if $e_n > e$, then

$$f < \left(d_0 + d_1 \cdot \beta^{-1} + \ldots + d_{p-1} \cdot \beta^{-(p-1)}\right) \cdot \beta^{e_n}$$

If we let

$$d_i' \neq d_i$$

for some number of i's, then

$$f \neq \left(d_0' + d_1\prime \cdot \beta^{-1} + \ldots + d_{p-1}\prime \cdot \beta^{-(p-1)}\right) \cdot \beta^{e}.$$

Hence, normalized FP numbers are unique.

Exercise 2.4. Write a function that takes a decimal number, base, and precision, and returns the closest normalized FP representation. I.e. a vector of digits and the exponent.

Solution. The function provided in class is actually the solution (?). This is guarenteed to give a normalized FP representation. Using this algorithm gives $d_0 = \lfloor \frac{N}{\beta^{\lfloor log_\beta(N) \rfloor}} \rfloor$. It holds that $\lfloor log_\beta(N) \rfloor \leq log_\beta(N)$, which implies that $\beta^{\lfloor log_\beta(N) \rfloor} \leq \beta^{log_\beta(N)} = N$ (remember, $\beta \geq 2$). Hence, $d_0 > 0$.

```
get_normalized_FP = function(number::Float64, base::Int64, prec::Int64)
    #number = 4; base = float(10); prec = 2
    si=sign(number)
    base = float(base)
    e = floor(Int64,log(base,abs(number)))
    d = zeros(Int64,prec)
    num = abs(number)/(base^e)

    for j = 1:prec
        d[j] = floor(Int64,num)
        num = (num - d[j])*base
    end

    return "The sign is $si, the exponent is $e, and the vector with d is $d"
end
```

```
## #53 (generic function with 1 method)
```

Exercise 2.5. List all normalized fp numbers that can be representated given base, precision, e_{min}, and e_{max}.

```
all_normalized_fp = function(base::Int64, prec::Int64, emin::Int64, emax::Int64)
    ## Number of possible values for each e:
    N = (base-1)*base^(prec-1)#*(emax-emin+1)

    out=zeros(Int64, N, prec, emax-emin+1)

    es = emin:emax

    for e=1:length(es)
        for b0=1:(base-1)
            for i=1:(base^(prec-1))
```

```
                out[(b0-1)*(base^(prec-1))+i,1,e] = b0
                for j=1:(prec-1)
                    out[(b0-1)*(base^(prec-1))+i,prec-j+1,e] = floor((i-1)/base^(j-1))%base
                end
            end
        end
    end

    return(out)
end
```

```
## #55 (generic function with 1 method)
```

2.2 Homework 2

Exercise 2.6. Lookup the 64 bit standard to find allowed exponents.
Solution. According to Wikipedia, the allowed exponents for the 64 bit standard are $-1022, \ldots, 1023$.
Exercise 2.7. What is the smallest non-normalized positive value for the 64 bit standard?
Solution. The smallest non-normalized positive value is

$$\left(0 + 0 \cdot 2^{-1} + \ldots + 0 \cdot 2^{-51} + 1 \cdot 2^{-52}\right) \cdot 2^{-1022} = 2^{-1074} \approx 4.94 \cdot 10^{-324}$$

Exercise 2.8. What is the smallest normalized positive value?
Solution. The smallest normalized positive value is

$$\left(1 + 0 \cdot 2^{-1} + \ldots + 0 \cdot 2^{-52}\right) \cdot 2^{-1022} = 2^{-1022} \approx 2.23 \cdot 10^{-308}$$

Exercise 2.9. What is the largest normalized positive value?
Solution. The largest normalized finite value is

$$\left(1 + 1 \cdot 2^{-1} + \ldots + 1 \cdot 2^{-52}\right) \cdot 2^{1023} \approx 1.80 \cdot 10^{308}.$$

Exercise 2.10. Is there a general formula for determining the largest positive value for a given base β, precision p, and largest exponent e_{max}?
Solution. The largest positive, finite value is

$$\left(\sum_{i=0}^{p-1} (\beta - 1)\beta^{-i}\right) \cdot \beta^{e_{max}} = \ldots = \frac{\beta^p - 1}{\beta^{p-1}} \beta^{e_max}.$$

Exercise 2.11. Verify the smallest non-normalized, positive number that can be represented.
Solution. See the `Julia` chunk below.

```
nextfloat(Float64(0)) == 2^(-1074)
```

```
## true
```

Exercise 2.12. Verify the smallest normalized, positive number that can be represented.

```
nextfloat(Float64(0))*2^(52)
```

```
## 2.2250738585072014e-308
```

Exercise 2.13. Verify the largest, finite number that can be represented.

```
prevfloat(Float64(Inf))
```

```
## 1.7976931348623157e308
```

Exercise 2.14. Proof lemma (1.1).
Solution. Let $z = (d_0 + d_1\beta^{-1} + \dots)\beta^e$ be a number. Let $fl(z) = (d_0' + d_1'\beta^{-1} + \cdots + d_{p-1}'\beta^{-(p-1)})\beta^e$ be its fp representation with precision p.

We know that $(d_0 + d_1\beta^{-1} + \cdots + d_{p-1}\beta^{-(p-1)})\beta^e \leq z \leq (d_0 + d_1\beta^{-1} + \cdots + (d_{p-1}+1)\beta^{-(p-1)})\beta^e$. We know that $fl(z)$ is equal to the one of these two bounds that is closest to z. Hence, $|fl(z) - z|$ must be at most half the distance between these two. Subtract the upper bound from the lower bound, and you get $\beta^{-(p-1)+e}$, i.e.

$$|fl(z) - z| \leq \frac{\beta^{e-p+1}}{2}.$$

Exercise 2.15. What happens with lemmas (bounds of absolute and relative error) if we consider negative numbers?
Solution. They still hold. Let $z' = -z$. Then $fl(z') = -fl(z)$. Hence,

$$|fl(z') - z'| = |-fl(z) + z| = |fl(z) - z|,$$

hence the bounds still hold.
Exercise 2.16. Show that $||A||_1 =$ max of the l^1 norms of the columns of A.
Solution. By definition, $||A||_1 = \max_{x\neq 0} \frac{||Ax||_1}{||x||_1} = \max_{||x||_1=1} ||Ax||_1$. Let $A \in \mathbb{R}^{m\times n}$ and $x \in \mathbb{R}^n$ s.t. $||x||_1 = 1$.

Recall that

$$Ax = \sum_{j=1}^{n} x_j A_{*,j},$$

where $A_{*,j}$ is the j'th column of A. So

$$\begin{aligned}
||Ax||_1 &= \sum_{i=1}^{m} \left| \sum_{j=1}^{n} x_j A_{i,j} \right| \\
&\leq \sum_{i=1}^{m} \sum_{j=1}^{n} |x_j| \cdot |A_{i,j}| \\
&= \sum_{j=1}^{n} |x_j| \left\{ \sum_{i=1}^{m} |A_{i,j}| \right\} \\
&= \sum_{j=1}^{n} |x_j| \, ||A_{*,j}|| \\
&\leq \sum_{j=1}^{n} |x_j| \max_{j\in\{1,\dots,n\}} ||A_{*,j}|| \\
&= \max_{j\in\{1,\dots,n\}} ||A_{*,j}||
\end{aligned}$$

I.e. $\max_{||x||_1=1} ||Ax||_1 \leq \max_{j\in\{1,\dots,n\}} ||A_{*,j}||$.

Now, let $1_i = (x_j)_{j=1}^n$ be defined as

$$x_j = \begin{cases} 1, & \text{if } j = i \\ 0, & \text{otherwise} \end{cases}$$

Then $||1_i||_1 = 1$. Since $\max_{j\in\{1,\dots,n\}} ||A_{*,j}|| = ||A \cdot \mathbf{1}_i||$ for $i = \operatorname{argmax} ||A_{*,j}||_1$, we have that $\max_{j\in\{1,\dots,n\}} ||A_{*,j}||_1 \leq \max_{||x||_1=1} ||Ax||_1$.

Exercise 2.17. Let $A \in \mathbb{R}^{n\times m}$. Show that $||A||_\infty$ = max of the l^1 norms of the rows of A.

Solution. Let $j \in \{1,\dots,n\}$ such that $|a_{j1}| + \cdots + |a_{jm}| \geq |a_{i1}| + \cdots + |a_{im}|$ for all $i \in \{1,\dots,n\}$, and let $\tilde{x} \in \mathbb{R}^m$ such that $\tilde{x}_i = \operatorname{sign}(a_{ji})$. Note that for all $x \in \mathbb{R}^m$ with $||x||_\infty = 1$ it holds that $|x_i| \leq 1$. So,

$$\begin{aligned} |a_{i1}x_1 + \cdots + a_{im}x_m| &\leq |a_{i1}| + \cdots + |a_{im}| \\ &\leq |a_{j1}| + \cdots + |a_{jm}| \\ &= |a_{j1}\tilde{x}_1 + \cdots + a_{jm}\tilde{x}_1|, \end{aligned}$$

I.e. for any $x \in \mathbb{R}^m$ with $||x||_\infty = 1$,

$$\begin{aligned} |a_{j1}\tilde{x}_1 + \cdots + a_{jm}\tilde{x}_1| &\geq \max_{1\leq i\leq n} \{|a_{i1}x_1 + \cdots + a_{im}x_m|\} \\ &= ||Ax||_\infty . \end{aligned}$$

Since $||\tilde{x}||_\infty = 1$, we have that

$$\begin{aligned} \max_{||x||=1} ||Ax||_\infty &= |a_{j1}\tilde{x}_1 + \cdots + a_{jm}\tilde{x}_1| \\ &= |a_{j1}| + \cdots + |a_{jm}| \\ &= \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ji}|, \end{aligned}$$

which is exactly the maximum over the $\updownarrow^1$-norms of the rows in A.

Exercise 2.18. Assume the Fundamental Axiom. Show the following:

$$||fl(A) - A||_p \leq u\, ||A||_p$$

Solution.

$$\begin{aligned} ||fl(A) - A||_p &= |||[fl(a_{ij}) - a_{ij}]|||_p \\ &\leq |||[u \cdot a_{ij}]|||_p \\ &= ||u \cdot A||_p = u\, ||A|| \end{aligned}$$

2.3 Homework 3

Exercise 2.19. Prove lemma 1.5.

Solution. Recall that a matrix A is invertible if and only if $Ax = 0$ implies that $x = 0$. So to check that $I - F$ is invertible, we check this:

$$(I - F)x = x - Fx = 0 \Rightarrow x = Fx \Rightarrow ||x||_p \leq ||F||_p\, ||x||_p .$$

Since $||F||_p < 1$ by assumption, the only solution to the inequality above is $x = 0$. So, $I - F$ is invertible.

Note that $\sum_{k=0}^{N} F^k(I-F) = I - F^{N+1}$. Since $\left|\left|F^k\right|\right|_p < ||F||_p^k < 1$, we know that $F^N \to 0$ as $N \to \infty$. So, $\sum_{k=0}^{\infty} F^k(I-F) = I$, and using that $I - F$ is invertible, we get $(I-F)^{-1} = \sum_{k=0}^{\infty} F^k$. Finally,

$$\left|\left|(I-F)^{-1}\right|\right|_p \leq \sum_{k=1}^{\infty} ||F||_p^k = \frac{1}{1-||F||_p}$$

.

Exercise 2.20. Consider Theorem and Lemmas under "Square Linear Systems" (1.2.3). What happens if we use l^1-norm instead?

Solution. Since we never use any properties of the infinity norm to prove these theorems and lemmas, we could replace it with the $\updownarrow^1$-norm.

Exercise 2.21. Generate examples that show the bound in theorem 1.1 is too conservative.

```
using LinearAlgebra

p = 53 # precision for float-64
```

```
## 53
```

```
u = 2.0^(-p+1)
```

```
## 2.220446049250313e-16
```

```
A = [
    0. 0 0.000001;
    200000 0 0;
    0 1 20000;
]
```

```
## 3×3 Array{Float64,2}:
##       0.0  0.0      1.0e-6
##  200000.0  0.0      0.0
##       0.0  1.0  20000.0
```

```
x = [1.; 1; 1;]
```

```
## 3-element Array{Float64,1}:
##  1.0
##  1.0
##  1.0
```

```
b = A*x
```

```
## 3-element Array{Float64,1}:
##       1.0e-6
##  200000.0
##   20001.0
```

```
kappa = norm(A, Inf) * norm(inv(A), Inf)
```

```
## 4.0e15
```

```
bound = 2.0*u*kappa/(1.0 - u*kappa)
```

```
## 15.885635265004744
```

```
println("Diff: $(norm(x - inv(A)*b, Inf)/norm(x,Inf))")
println("Bound: $(bound)")
```

Exercise 2.22. Generate examples that show the bound is nearly achieved

```
using LinearAlgebra

A = [
    0. 0 1.;
    1 0 0;
    0 1 0;
]
```

```
## 3×3 Array{Float64,2}:
##  0.0  0.0  1.0
##  1.0  0.0  0.0
##  0.0  1.0  0.0
```

```
x = [1.; 1; 1;]
```

```
## 3-element Array{Float64,1}:
##  1.0
##  1.0
##  1.0
```

```
b = A*x
```

```
## 3-element Array{Float64,1}:
##  1.0
##  1.0
##  1.0
```

```
kappa = norm(A, Inf) * norm(inv(A), Inf)
```

```
## 1.0
```

```
bound = 2.0*2.0^(-p+1)*kappa/(1.0 - 2.0^(-p+1)*kappa)
```

```
## 4.440892098500627e-16
```

```
println("Diff: $(norm(x - inv(A)*b, Inf)/norm(x,Inf))")
println("Bound: $(bound)")
```

Exercise 2.23. For motivating problems 1-5, when is x unique?
Solution. - Motivating Problem 1: * when A is invertible - Motivating Problem 2: Always. * Since $F(y) = ||Ay - b||_2 = (Ay - b)'(Ay - b)$ is convex (twice differentiated is positive definite because $\nabla^2 f = A'A$ and A has full rank). * (since objective function is $A'A$, and it is positive definite (because A is full rank), then the function is convex, and you always have a unique solution) - Motivating Problem 3: Always. - Motivating Problem 4: Always. * We know how to characterize all z that satisfy objective: $Q \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} \Pi' = A$.

- Motivating Problem 5: always

Exercise 2.24. For motivating problem 5, what happens if $p \geq m$? Explore the case where $m >> n$.
Solution. * For m >> n, you have an undertermined system with more unknowns than equations. Since the null space of A has a dimension larger than zero, for any particular solution xp for the system, xp+h with $h \in$ null(A) is also a solution, and there are infinitely many choices for h. The constraint system might help narrow down the solutions from the null space. * For p >= m: if C becomes inconsistent, we cannot narrow down the solutions in the null space. If C is consistent and rank(C)=m, then we can pick a unique solution. If the rank of C is less than m, then C constrains some of the possible solutions for y.
Exercise 2.25. What do you get if you multiply a matrix by a permutation matrix from the left? From the right? A permutation with itself?
Solution. From the left: permute rows.

From the right: permute columns.

Permutation squared gives you the identity.

Exercise 2.26. Suppose $R \in \mathbb{R}^{m\times m}$ is an upper triangular matrix with $R_{ii} \neq 0$ for all $i = 1, \ldots, n$. Is R invertible?

Solution. Since R is an upper triangular matrix, $\det(R) = \prod_{i=1}^{m} R_{ii} > 0$. Hence, R is invertible.

Exercise 2.27. Assume R is an invertible upper triangular matrix. Implement a solution to invert R.

Solution. First, we will show that the inverse of R is also an upper triangular matrix. So, let $B = R^{-1}$. The $RB = I$. We will show this using induction. Let $i = n - 0, j < i$. Then, since $r_{ij} = 0$ for all $i > j$,

$$\begin{aligned} 0 &= \sum_{k=1}^{n} r_{n,k} b_{k,j} \\ &= r_{n,n} b_{n,j}. \end{aligned}$$

Since R is invertible, $r_{n,n} \neq 0$, hence $b_{n,j} = 0$ for all $j < n$.

Now assume that $b_{i,j} = 0$ for all $i = n - 0, n - 1, \ldots, n - m, j < i$. We then want to show it holds for $i = n - (m + 1) = n - m - 1$. Let $j < n - (m + 1)$. Then

$$\begin{aligned} 0 &= \sum_{k=1}^{n} r_{n-(m+1),k} b_{k,j} \\ &= \sum_{k=n-(m+1)}^{n} r_{n-(m+1),k} b_{k,j} \\ &= r_{n-(m+1),n-(m+1)} b_{n-(m+1),j}, \end{aligned}$$

where the first equality is due to the fact that $r_{ij} = 0$ for all $i > j$, and the last equality holds since $b_{k,j} = 0$ for all $j < k$ when $k \geq n - m$ (per the induction hypothesis). Since $r_{ii} \neq 0$ for all i, $b_{n-(m+1),j} = 0$.

So, $b_{ij} = 0$ for all $i > j$.

Now, lets look at the case where $i = j$, i.e. diagonal elements of the inverse matrix. Then

$$1 = \sum_{k=1}^{n} r_{ik} b_{ki} = \sum_{k=i}^{n} r_{ik} b_{ki} = r_{ii} b_{ii}.$$

The second equality holds since $r_{ik} = 0$ for all $i > k$, the last equality holds because $b_{ki} = 0$ for all $k > i$. We see that $b_{ii} = r_{ii}^{-1}$.

Finally, consider the case where $i < j$. Then, using that $r_{ik} = 0$ for all $i > k$ and $b_{kj} = 0$ for all $k > j$, we see that

$$0 = \sum_{k=1}^{n} r_{ik} b_{kj} = \sum_{k=i}^{j} r_{ik} b_{kj},$$

which implies

$$b_{ij} = \frac{-\sum_{k=i+1}^{j} r_{ik} b_{kj}}{a_{ii}}.$$

So, in other words, given i, j, we can find b_{ij} if we know b_{kj} for all $k > i$ (i.e. all entries in the same column below the entry we are considering). Since we already know all entries below and on the diagonal, this is true for all columns. Hence, we can construct B this way.

See the `Julia` chunk below for an implementation of this.

```
function invertUpperTri(A)
    ## Get dimensions of A
    n, m = size(A)

    ## Setup empty array to hold result
    B = zeros(n,m)

    ## Fill out diagonal with inverse diagonal from A
    for k = 1:n
        B[k, k] = 1/A[k,k]
    end

    ## Starting in the lower right corner, fill out the rest of the matrix.
    for i = (n-1):-1:1
        for j = (i+1):n
            B[i,j] = -sum(A[i,(i+1):j].*B[(i+1):j,j])/A[i,i]
        end
    end

    return(B)
end
```

```
## invertUpperTri (generic function with 1 method)
```

```
## Create an upper triangular matrix to check
A = rand(4,4);
A[2:4,1] = [0 0 0];
A[3:4,2] = [0 0];
A[4,3] = 0;

## Check function
B = invertUpperTri(A);
B - inv(A)  # should be 0 matrix
```

```
## 4×4 Array{Float64,2}:
##  0.0  0.0  0.0  0.0
##  0.0  0.0  0.0  0.0
##  0.0  0.0  0.0  0.0
##  0.0  0.0  0.0  0.0
```

```
A*B         # should be identity
```

```
## 4×4 Array{Float64,2}:
##  1.0  -6.78909e-17  -2.06953e-17  -3.17238e-17
##  0.0   1.0           3.67183e-17   3.01139e-17
##  0.0   0.0           1.0           6.41489e-18
##  0.0   0.0           0.0           1.0
```

2.4 Homework 4

Exercise 2.28. In the solution to 1.2, why do 0 rows on the left-hand side of (1.1) correspond to 0 entries of the c vector on the right-hand side.

Solution. By definition of matrix multiplication, if the ith row is a zero row of A, then $c_{ij} = [AB]_{ij} = \sum j = 1^n a_{ij} b_{ij} = 0$, no matter what B is.

Exercise 2.29. Let Q be an orthogonal matrix. Show that $||Qx||_2 = ||x||_2$.
Solution. Since $||x||_2^2 = x'x$ and $Q'Q = 1$ (Q is orthogonal),

$$||Qx||_2^2 = (Qx)'Qx = x'Q'Qx = x'x = ||x||_2^2 .$$

Exercise 2.30. Let f be a vector-valued function over $\mathbb{R}^d$. When is

$$\text{argmin}_x \, ||f(x)||_2 = \text{argmin}_x \, ||f(x)||_2^2 .$$

Solution. Most of the time...?
Exercise 2.31. Prove that $P^TP + I$ from solution to example 1.4 is invertible.
Solution. For all non-zero x,

$$x'(P^TP + I)x = x'P'Px + x'x = ||Px||_2 + ||x||_2 > 0.$$

This means $P'P + I$ is positive definite, which in turne implies that it is invertible.
Exercise 2.32. Write out the solution to example 1.6. Also consider the case where $p \geq m$.
Solution. First we do QR decomposition on C':

$$C' = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$$

Then we have:

$$AQ = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$$
$$Q'y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

And we can update the objective:

$$\begin{aligned} O &= \min \|Ay - b\|_2 : Cy = d \\ &= \min \|AQQ'y - b\|_2 : Cy = d \\ &= \min \left\| \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - b \right\|_2 : Cy = d \\ &= \min \|A_1y_1 + A_2y_2 - b\|_2 : Cy = d \end{aligned}$$

We can also update the constraint:

$$\begin{aligned} O &= \min \|A_1y_1 + A_2y_2 - b\|_2 : \begin{bmatrix} R' & 0 \end{bmatrix} Q'y = d \\ &= \min \|A_1y_1 + A_2y_2 - b\|_2 : \begin{bmatrix} R' & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = d \\ &= \min \|A_1y_1 + A_2y_2 - b\|_2 : R'y_1 = d \\ &= \min \|A_1y_1 + A_2y_2 - b\|_2 : y_1 = (R')^{-1}d \\ &= \min \left\|A_1(R')^{-1}d + A_2y_2 - b\right\|_2 \\ &= \min \left\|A_2y_2 - (b - A_1(R')^{-1}d)\right\|_2 \end{aligned}$$

So y_1 can be calculated using a consistent linear system solver, and now the objective is the same as that of a least squares solver, which can be used to calculate y_2. We can finally recover y:

$$y = Q \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Exercise 2.33. For all motivating problems, implement solutions.

For the following, assume $A \in \mathbb{R}^{n\times m}$ with $\text{rank}(A) = m$, and $b \in \mathbb{R}^n$. Let $C = \begin{bmatrix} A & b \end{bmatrix}$.
Exercise 2.34. What does the last column of R (from the QR decomposition of C) represent?
Solution. Generally, $Q'b$, which is the projection of b onto the column space of A. If b is in the column space of A, then it is specifically Rx where x is the solution to $Ax = b$.
Exercise 2.35. What does the last entry of last column of R (from the QR decomposition of C) represent?
Solution. The square of the last entry of the last column of R is the sum of squares of the residuals.
Exercise 2.36. How can this be used in computation?
Exercise 2.37. We can use it for incremental QR for large datasets.

2.5 Homework 5

Exercise 2.38. Implement the Gram-Schmidt procedure for matrices $A \in \mathbb{R}^{n\times m}$ assuming A has full column rank.

Create examples to show that the function works (well enough).
Solution. See here.
Exercise 2.39. Find examples where Gram-Schmidt fails, i.e. where either $QR \neq A$ or $Q^TQ \neq I$.
Solution. Let $A = \begin{bmatrix} 1 & 1 & 1 \\ 10^{-8} & 0 & 0 \\ 0 & 10^{-8} & 0 \end{bmatrix}$. Then $Q'Q \neq I$.
Exercise 2.40. Look up the modified Gram-Schmidt Procedure and implement it (again assuming A has full column rank).
Solution. See here.
Exercise 2.41 (Pivoting (*OPTIONAL*))**.** References:

1. Businger, Galub: Linear Least Squares by Householder Transformation (1965)
2. Engler: The Behavior of QR-factorization algorithm with column pivoting (1997)

Implement modified Gram-Schmidt with column pivoting.

Find example where the modified Gram-Schmidt fails, but the modified Gram-Schmidt with column pivoting does not.
Exercise 2.42. Show that Householder reflections are orthogonal matrices.
Solution. Show that $H'H = I$. By definition of a Householder matrix (1.7), $H = I - 2vv'$ for a v with $||v||_2 = 1$. Note that $||v||_2 = \sqrt{v' \cdot v}$.

So,

$$\begin{aligned} H'H &= (I - 2vv')'(I - 2vv') \\ &= (I' - 2(vv')')(I - 2vv') \\ &= (I - 2vv')(I - 2vv') \\ &= I - 2vv' - 2vv' + 4vv'vv' \\ &= I - 2vv' - 2vv' + 4vIv' \\ &= I. \end{aligned}$$

So by definition (1.5), H is an orthogonal matrix.
Exercise 2.43. Show that if $H_1, \ldots, H_r$ are Householder reflections, then $H_r \cdots H_1$ (i.e. the product) is orthogonal.
Solution.

$$(H_r \cdots H_1)'(H_r \cdots H_1) = H_1' \cdots H_r' H_r \cdots H_1 = I,$$

since all H_i are Householder reflections, which implies they are orthogonal, which implies $H_i'H_i = I$.

Exercise 2.44. Show that a Givens rotation is an orthonormal matrix when $\sigma^2 + \lambda^2 = 1$.
Solution. Let $G^{(a,b)}$ be a Givens rotation. I.e. the elements $g_{i,j}$ are

$$g_{i,j} = \begin{cases} 1, & i = j \notin \{a, b\} \\ \lambda, & i = j \in \{a, b\} \\ \sigma, & i = a, j = b \\ -\sigma, & i = b, j = a \\ 0, & \text{otherwise} \end{cases}$$

Transposing this matrix gives us a new matrix K where

$$k_{i,j} = \begin{cases} 1, & i = j \notin \{a, b\} \\ \lambda, & i = j \in \{a, b\} \\ -\sigma, & i = a, j = b \\ \sigma, & i = b, j = a \\ 0, & \text{otherwise} \end{cases}$$

Now, let $L = K \cdot G$. Then the elements of L are $l_{i,j} = \sum_{s=1}^{n} k_{i,s} g_{s,j}$.

If $i \neq a,\ j \neq b$, and $i \neq j$, then $l_{i,j} = k_{i,i} g_{i,j} = 0$ (since $g_{i,j} = 0$).

If $i \notin \{a, b\}$, then $l_{i,i} = k_{i,i} g_{i,i} = 1$ (since $i \notin \{a, b\}$).

If $j \notin \{a, b\}$, then $l_{a,j} = k_{a,b} g_{b,j} + k_{a,a} g_{a,j} = 0$ (since $j \notin \{a, b\}$ implies $g_{b,j} = 0$ and $g_{a,j} = 0$).

Furthermore,

- $l_{a,b} = k_{a,b} g_{b,b} + k_{a,a} g_{a,b} = -\sigma\lambda + \lambda\sigma = 0$,
- $l_{a,a} = k_{a,b} g_{b,a} + k_{a,a} g_{a,a} = (-\sigma)(-\sigma) + \lambda\lambda = \sigma^2 + \lambda^2$,
- $l_{b,b} = k_{a,b} g_{b,a} + k_{a,a} g_{a,a} = (-\sigma)(-\sigma) + \lambda\lambda = \sigma^2 + \lambda^2$.

So,

$$l_{i,j} = \begin{cases} 1, & i = j \notin \{a, b\} \\ \sigma^2 + \lambda^2, & i = j \in \{a, b\} \\ 0, & \text{otherwise} \end{cases}$$

Similar calculations can be performed for $G \cdot K$. So $G^{(a,b)}$ is orthogonal if and only if $\sigma^2 + \lambda^2 = 1$.

2.6 Homework 6

Exercise 2.45. Determine the computational complexity of the QR decomposition using

a) Gram-Schmidt
b) Modified Gram-Schmidt
c) Householder
d) Givens rotations

for any arbitrary, dense $n \times m$ matrix. (dense = don't know how many entries are 0.)
Solution. a) Gram-Schmidt requires $O(m^2 n)$ computations. b) Same as Gram-Schmidt. c) $O(m^2 n)$. d) $O(m^2 n)$.
Exercise 2.46. Compare the computational complexity of Householder and Givens for a sparse matrix (i.e. a matrix where a substantial number of entries are 0).
Solution. Householder still have the same complexity even for sparse matrices, whereas for Givens we do not have to complete as many multiplications with Givens rotations. (Everytime we encounter a 0 in the lower triangular matrix that can be paired with a zero above it, we can pair them up, and skip the computation.)

Exercise 2.47. Implement QR decomposition using Householder. Write it as a function that takes a matrix $A \in \mathbb{R}^{n\times m}$ with $\text{rank}(A) = m$ as its input, and gives back Q and R.
Solution. See here.
Exercise 2.48. Implement QR decomposition using Givens. (As above.)
Solution. See here
Exercise 2.49. What happens in theorem 1.3 if $m > n$.
Solution. If $m > n$, the diagonal matrix of the SVD of A will be of the form $\begin{bmatrix}\Sigma & 0\end{bmatrix}$, where $\Sigma = \begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_n\end{bmatrix}$.
Exercise 2.50. If u, v are respectively left and right singular vectors corresponding to the singular value σ, show that $Av = \sigma u$, and $A'u = \sigma v$.
Solution. Using the SVD of A, we see that $AV = U\Sigma$, which implies

$$A\begin{bmatrix}v_1 & \cdots & v_m\end{bmatrix} = \begin{bmatrix}\sigma_1 u_1 & \cdots & \sigma_m u_m\end{bmatrix}.$$

where $v_1, \ldots, v_m$ are the right singular vectors of A, and $u_1, \ldots, u_m$ are the left singular vectors of A.

So, $Av_i = \sigma_i u_i$ for all i.

Similarly, $A' = V\Sigma U'$ implies $A'U = V\Sigma$. As above, this gives us $A'u_i = \sigma_i v_i$.

2.7 Homework 7

Exercise 2.51. Show that $||A||_F^2 = \sum_{i=1}^n \sum_{j=1}^n a_{ij}^2$ is orthogonally invariant, i.e. if Q, P are orthogonal matrices, then $||QA||_F^2 = ||A||_F^2$ and $||AP||_F^2 = ||A||_F^2$.
Solution. Recall, that $||A||_F^2 = \text{trace}(A'A)$. So,

$$||QA||_F^2 = \text{trace}(A'Q'QA) = \text{trace}(A'A) = ||A||_F^2,$$

and

$$||AP||_F^2 = \text{trace}(P'A'AP) = \text{trace}(A'APP') = \text{trace}(A'A) = ||A||_F^2.$$

Exercise 2.52. Proof corollary 1.3.
Solution. Recall that $\sigma_{\max}(A) = ||A||_2$. Hence, $\sigma_{\max}(A+E) = ||A+E||_2 \leq ||A||_2 + ||E||_2 = \sigma_{\max}(A) + ||E||_2$.

For the second inequality, recall that $\sigma_{\min}(A) = \min_{x\neq 0} \frac{||Ax||}{||x||}$. First, consider the case where $\sigma_{\min}(A) = 0$. Then the result holds. Second, consider the case where $\sigma_{\min}(A) > 0$. Then

$$\begin{aligned}\sigma_{\min}(A+E) &= \min_{v\neq 0} \frac{||(A+E)v||_2}{||v||_2} \geq \min_{v\neq 0} \frac{||Av||_2}{||v||_2} - \max_{v\neq 0} \frac{||Ev||_2}{||v||_2} \\ &= \sigma_{\min}(A) - ||E||_2,\end{aligned}$$

where we use the inverse triangle equality.
Exercise 2.53. Proof corollary 1.5.
Solution (1). Let $b \in \text{range}(A)$. Then there exists some z such that $Az = b$. Use the SVD of A to get

$$b = U\Sigma V' z.$$

Since $\text{rank}(A) = r$, the last $n - r$ rows of Σ are all 0, so $U\Sigma V' = U_r\Sigma_r(V_r)'$, where U_r is the first r columns of U, Σ_r the first r columns and r rows of Σ, and V_r the first r columns of V. Hence, $\Sigma_r(V_r)'z = y \in \mathbb{R}^r$. So, $b = U_r y$, which means it is in the span of the first r columns of U. Hence, $\text{range}(A) \subset \text{span}\{u_1, \ldots, u_r\}$.

Now, let $b \in \text{span}\{u_1, \ldots, u_r\}$. Then $b = U_r y$ for some $y \in \mathbb{R}^r$. Recall that $U = AV\Sigma^{-1}$, hence $U_r = AV_r\Sigma_r^{-1}$. So, $b = AV\Sigma_r^{-1}y$. Since $V\Sigma_r^{-1}y = z \in \mathbb{R}^m$, b is in the range of A. Hence, $\text{span}\{u_1, \ldots, u_r\} \subset \text{range}(A)$.
Solution (2). Note that the row space of A is the range of A'. Solve as above.
Solution (3). Let $b \in \text{span}\{v_{r+1}, \ldots, v_m\}$. Then $b = x_{r+1}v_{r+1} + \cdots + x_m v_{r+1}$ for some coefficients $x_j \in \mathbb{R}$. Then

$$\begin{aligned} Ab &= U\Sigma V'b \\ &= U\Sigma(x_{r+1}V'v_{r+1} + \cdots + x_m V'v_m) \\ &= U\Sigma \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_{r+1} \\ \vdots \\ x_m \end{bmatrix}, \end{aligned}$$

since all columns of V are orthogonal.

Since the last $n-r$ rows of Σ are all 0 rows, $\Sigma V'b = 0$, so $Ab = 0$, so $b \in \text{null}(A)$. Hence, $\text{span}\{v_{r+1}, \ldots, v_m\} \subset \text{null}(A)$.

Now, Assume $b \in \text{null}(A)$, then $Ab = 0$. Because $b \in R^m$ and $v_1, \cdots, v_m$ is a base for R^m, there exists a sequence $\{x_i\}$ s.t. $b = x_1v_1 + \cdots + x_r v_r + x_{r+1}v_{r+1} + \cdots + x_m v_m$.

Assume for contradiction that $b \notin \text{span}(v_{r+1}, \cdots, v_m)$, i.e. that $x_1, \cdots, x_r$ are not all zero. Then

$$Ab = \sum_{i=1}^{r} \sigma_i u_i v_i^T b = \sum_{i=1}^{r} [\sigma_i u_i v_i^T (\sum_{j=1}^{m} x_j v_j)] = \sum_{i=1}^{r} [\sigma_i u_i (\sum_{j=1}^{m} x_j v_i^T v_j)] = \sum_{i=1}^{r} \sigma_i u_i x_i \neq 0$$

But this contradicts the assumption that $b \in \text{null}(A)$. So, b must be in $\text{span}\{v_{r+1}, \cdots, v_m\}$. Hence, $\text{null}(A) \subset \text{span}\{v_{r+1}, \cdots, v_m\}$.

2.8 Homework 8

Exercise 2.54. What is the solution to $\min_x ||Ax - b||_2^2$ in terms of the Moore-Penrose inverse? (pseudo-inverse)
Solution. To find the solution, we differentiate and set equal to 0. So, differentiate $g(x) = x'A'Ax - x'Ab - b'Ax + b'b$ and set to 0:

$$\frac{dg}{dx} = x'(A'A + A'A) - b'A - (A'b)' = 2x'(A'A) - 2b'A = 0$$

which implies $(AA')x = bA'$. Check that $x = A^+b$ satisfies this.
Exercise 2.55. Let Q be an orthogonal matrix with columns $q_1, \ldots, q_n$. Let $Z \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Show that $\sum_{i=1}^{n}(q_i'Zq_i)^2 \leq ||Z||_F^2$.
Solution. Since the frobenius norm is invariant to mulitplication by orthogonal matrices: $||Z||_F^2 = ||Q'ZQ||_F^2$. Since

$$\begin{bmatrix} q_1' & \cdots q_n' \end{bmatrix} Z \begin{bmatrix} q_1 \\ \vdots q_n \end{bmatrix} = \begin{bmatrix} q_1' & \cdots q_n' \end{bmatrix} \begin{bmatrix} Zq_1 \\ \vdots Zq_n \end{bmatrix}$$
$$= \begin{bmatrix} q_1'Zq_1 & \cdots & q_1'Zq_n \\ \vdots & \ddots & \vdots \\ q_n'Zq_1 & \cdots & q_n'Zq_n \end{bmatrix},$$

we have that $||Z||_F^2 = \sum_{i=1}^n \sum_{j=1}^n (q_i'Zq_j)^2 \geq \sum_{i=1}^n (q_i'Zq_i)^2$.

2.9 Homework 9

Exercise 2.56. What do we need from A to ensure that the different iterative method schemes are well-defined?
Solution. A must be square and diagonal entries must be non-zero.
Exercise 2.57. Does lemma 1.13 hold for the symmetric SOR?
Exercise 2.58. Compute J^k where $J = I\lambda + E$ (the Jordan Canonical Form from definition 1.15)

Show that if $G = XJX^{-1}$ and $\rho(G) < 1$, then $J^k \to 0$ as $k \to \infty$.

2.10 Homework 10

Exercise 2.59. Implement Jacobi, Gauss-Seidel without using the \ operator.
Solution. See Jacobi and Gauss-Seidel.
Exercise 2.60. Implement SOR, SSOR without using the \ operator.
Solution. See SOR.
Exercise 2.61. Randomly generate problems and

a) at each iteration, record residual norm and absolute error
b) compare rate of convergence against spectral radious
c) put all this information into a narrative using graphics

Exercise 2.62. Prove that observation 1 from the proof of 1.7 is true.
Exercise 2.63. Implement Random Kaczmarz for random permutations.
Exercise 2.64. Give a detailed comparison of cycle, randomized, and random permutation Kaczmarz.

2.11 Homework 11

Exercise 2.65. Show that $\alpha \sum_{i=1}^n a_i(b_i - a_i x^c) = \alpha A'(b - Ax^c)$.
Exercise 2.66. Show that if $A'r^c = 0$, then $x^c = x^*$, where $Ax^* = b$.
Exercise 2.67. Why is it enough to find an upper bound on $\left|\left|\Sigma^{-1}u\right|\right|_2^2 ||\Sigma u||_2^2$ for any unit vector w in the proof of theorem 1.8?
Exercise 2.68 (Kontorovich's Inequality)**.** Let $0 < u_n \leq u_{n-1} \leq \cdots \leq u_1$. Then

$$\left(\sum_{i=1}^n p_i u_i\right)\left(\sum_{i=1}^n \frac{p_i}{u_i}\right) \leq \frac{(u_1 + u_n)^2}{4u_1 u_n}$$

Exercise 2.69. For strategy 2, answer the following questions:

a) What is α? Is it practical?
b) With this α, will we converge? If so, what is the rate of convergence?

Exercise 2.70. For strategy 3, what is α?
Exercise 2.71. For strategy 4, show the inequality holds.
Exercise 2.72. For strategy 4, what is α?
Exercise 2.73. Implement a single update of the gradient descent method for a user supplied alpha, assuming that A is invertible.
Exercise 2.74. Now, implement four algorithms one for each of the four strategies of alpha discussed in class.
Exercise 2.75. Generate several test problems for your four algorithms, and compare their performance.
Exercise 2.76. In a narrative, explain which algorithm you would use and when you would use this algorithm.

Understanding Gradient Descent's Dependence on the Condition Number
Exercise 2.77. Write a function to generate 2 equations with 2 unknowns such that the coefficient matrix is dense and symmetric with user specified nonzero eigenvalues.
Exercise 2.78. How are the eigenvalues related to the singular values?
Exercise 2.79. How are the eigenvectors related to the left and right singular vectors?
Exercise 2.80. Write a function to draw the level sets of an arbitrary residual function $f(x) = ||Ax - b||_2^2$.
Exercise 2.81. Run Gradient Descent using the step size (alpha) from Strategy 2 to solve a sequence of problems where the difference between the singular values of your problem increases in size. Plot the points that Gradient Descent visits (on your level set plot) as it finds its way to the solution.
Exercise 2.82. What do you observe? What is the impact of the singular values on gradient descent? Why does this make sense based on your graphs?