STAT 771: My notes

Ralph Møller Trane

Fall 2018 (compiled 2018-09-20)

Contents

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:

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

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.

1.3 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 $\beta = 2$.

1.3.1 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

1.3.2 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

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

1.3.3 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: $2^1 1 - 2 = 2046$
    - 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. $\beta_0 = 1$.

1.4 Errors

1.4.1 Units in the Last Place (ULP)

1.4.2 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 **??**, 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`.

1.4.2.1 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 definted 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 $s_1 = fl(x_1y_y) = x_1y_1(1+\delta_1)$, where $|\delta_1| \leq u$. We can similarly find s_p as

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

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

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

So,

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

We now use **??** to get:

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

□

1.5 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. $||F||_p < 1$ 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 **??** 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.

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

□

1.6 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

1.6.1 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.

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

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.

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

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.

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

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 $C \in \mathbb{R}^{p\times m}, \text{C} = p$, and $d \in \mathbb{R}^p$. Find $x \in \mathbb{R}^m$ s.t.

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

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 $\left|\left|Q_{i*}\right|\right|_2 = 1$ for all $i = 1, \ldots, n$, and $\left|\left|Q_{*j}\right|\right|_2 = 1$ 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

1.6.2 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,

[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.

2) an $n \times n$ orthogonal matrix 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 **??** 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 **??**(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 **??**, 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 **??**). So, we get that (**??**) 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

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

Solution (Example ref(exm:und-linear-system)). In this scenario, $\operatorname{rank}(A) = r < m$. We are looking for $x \in \operatorname{argmin}_y \{||y||_2 \, |Ay = b\}$. Using theorem **??**, we can rewrite this as $\operatorname{argmin}_y \left\{||y||_2 \, \middle| Q \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} y = b\right\}$, and multiplying by Q^T, $\operatorname{argmin}_y \left\{||y||_2 \, \middle| \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} y = Q^T b\right\}$. 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

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

where the last equality is a consequence of the result proved in homework @ref{exr:q403}. Now, let $d = R^{-1}c$ and $p = R^{-1}Sz_2$. Then we can find the minimum of the above expression by differentiating and setting equal to zero:

$$0 = -P^T d + (P^T P + I)z_2 \to z_2 \qquad = (P^T P + I)^{-1} P^T d. \tag{1.2}$$

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

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

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

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

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

1.6.3 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 (**??**))

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 $R_r r = ||\tilde{q}_r||_2$. 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 (**??**))

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 **??***).*

Proof of theorem ref(thm:qr-decomposition). Since $\text{rank}(A) = r$, 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 **??**), 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 $\text{span}\{\tilde{q}_1, \dots, \tilde{q}_r\}$ (columns of $\tilde{Q}$) is equal to $\text{span}\{a_1, \dots, a_r\}$, 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***

1.6.4 Householder

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. HOMEWORK □

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

Proof. Let $A\Pi = \begin{bmatrix} a_1 \dots a_r \end{bmatrix}$. Choose H_1 s.t. $H_1 a_1 = R_{11} e_1 = a_1 - 2 v_1 v_1' a_1$ (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

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

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

□

1.6.5 Givens Rotations

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.

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
```

```
## #11 (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
```

```
## #13 (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

$$\left(1 + 1 \cdot 2^{-1} + \ldots + 1 \cdot 2^{-52}\right) \cdot 2^{-1022} \approx 1.80 \cdot 10^{308}.$$

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

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

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

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

```
## 1.7976931348623157e308
```

Exercise 2.14. Proof lemma (**??**).