1. (a) There are two places that can cause potential subtractive cancellation: $\beta - \sqrt{\beta^2 - x^2}$, the two minus sign hightlighted in function.

① For the first '$-$' sign $\beta - \sqrt{\beta^2 - x^2}$, if $x \approx 0$, then $f(x) \approx 0$

② For the second '$-$' sign, only need to look at $\beta^2 - x^2$, which is if $x \approx \beta$, then $f(x) \approx \beta$, but this case is insignificant, because $\beta > 0$ and $|x| < \beta$, so $\beta \neq 0 \Longrightarrow f(x) \neq 0$, which means this case is insignificant.

(b) ① $x \approx 0$, $f'(x) = -\frac{-2x}{2\sqrt{\beta^2 - x^2}} = \frac{x}{\sqrt{\beta^2 - x^2}}$

$$\lim_{x \to 0} \operatorname{cond}(f) = \lim_{x \to 0} \left| \frac{x \cdot f'(x)}{f(x)} \right|$$

$$= \lim_{x \to 0} \left| \frac{x \cdot \left(\frac{x}{\sqrt{\beta^2 - x^2}} \right)}{\beta - \sqrt{\beta^2 - x^2}} \right|$$

$$= \lim_{x \to 0} \left| \frac{x^2}{\beta\sqrt{\beta^2 - x^2} - \beta^2 + x^2} \right| \quad (\beta > 0)$$

$$= 0 \quad \text{well-conditioned}$$

Since when $x \approx 0$, $f(x)$ is well-conditioned, it doesn't the cancellation error

(c)
$$\beta - \sqrt{\beta^2 - x^2} \left(\frac{\beta + \sqrt{\beta^2 - x^2}}{\beta + \sqrt{\beta^2 - x^2}} \right)$$

$$= \frac{\beta^2 - (\beta^2 - x^2)}{\beta + \sqrt{\beta^2 - x^2}}$$

$$= \frac{\beta^2 - \beta^2 + x^2}{\beta + \sqrt{\beta^2 - x^2}}$$

$$= \frac{x^2}{\beta + \sqrt{\beta^2 - x^2}}$$

(d) From (b) and (c), I find that if the condition number of an expression which suffering from subtractive cancellation is small (expression is well-conditioned), then there exists a alternate form of that expression which doesn't have subtrative cacellation issue.

2. Let $\delta_{x/z}$ denote te δ for $fl(fl(x) \div fl(z))$

$$fl((fl(fl(x) \div fl(z))) \cdot fl(y))$$

$$= \left[\left(\frac{x(1-\delta_x)}{z(1-\delta_z)}(1-\delta_{x/z})\right) \cdot y(1-\delta_y)\right](1-\delta_{(x/z)\cdot y})$$

$$= \left[\left(\frac{x}{z} \cdot \frac{1-\delta_x}{1-\delta_z} \cdot \frac{1+\delta_z}{1+\delta_z}(1-\delta_{x/z})\right) \cdot y(1-\delta_y)\right](1-\delta_{(x/z)\cdot y})$$

$$= \left[\left(\frac{x}{z} \cdot \frac{1-\delta_x+\delta_z-\delta_x\delta_z}{1-\delta_z^2}(1-\delta_{x/z})\right) \cdot y(1-\delta_y)\right](1-\delta_{(x/z)\cdot y})$$

$$= \left(\frac{x}{z}\right) \cdot y(1-\delta_x+\delta_z-\delta_{x/z}-\delta_y-\delta_{(x/z)\cdot y})$$ Note: $\delta < eps$

$$= \left(\frac{x}{z}\right) \cdot y(1-\delta_{(\,)\cdot})$$ $|\delta^2| \ll \delta$

$$\left(\frac{x}{z}\right) \cdot y(1-\delta_{(\,)\cdot}) = \left(\frac{x}{z}\right) \cdot y(1-\delta_x+\delta_z-\delta_{x/z}-\delta_y-\delta_{(x/z)\cdot y})$$ $|\delta^3| \lll \delta$

$$1-\delta_{(\,)\cdot} = 1-\delta_x+\delta_z-\delta_{x/z}-\delta_y-\delta_{(x/z)\cdot y}$$ $|\delta^4| \llll \delta$

$$\delta_{(\,)\cdot} = \delta_x-\delta_z+\delta_{x/z}+\delta_y+\delta_{(x/z)\cdot y}$$ $|\delta^5| \lllll \delta$

$$|\delta_{(\,)\cdot}| \le |\delta_x|+|\delta_z|+|\delta_{x/z}|+|\delta_y|+|\delta_{(x/z)\cdot y}|$$ by triangle equality

$$|\delta_{(\,)\cdot}| \le 5\,eps$$

3. $\|x\|_p = \frac{\|x\|_\infty}{\|x\|_\infty}\|x\|_p$

$= \|x\|_\infty \left(\frac{\|x\|_p}{\|x\|_\infty}\right)$

$= \|x\|_\infty \left(\frac{(|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}}}{\|x\|_\infty}\right)$

$= \|x\|_\infty \left(\frac{(\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}}{\|x\|_\infty}\right)$

$= \|x\|_\infty \left(\frac{\sum_{i=1}^n |x_i|^p}{\|x\|_\infty^p}\right)^{\frac{1}{p}}$

$= \|x\|_\infty \left(\sum_{i=1}^n \frac{|x_i|^p}{\|x\|_\infty^p}\right)^{\frac{1}{p}}$

$= \|x\|_\infty \left(\sum_{i=1}^n \left(\frac{|x_i|}{\|x\|_\infty}\right)^p\right)^{\frac{1}{p}}$

$\leq \|x\|_\infty \cdot n^{\frac{1}{p}}$ (b/c $\|x\|_\infty = \max_{1\leq i\leq n}|x_i|$ by definition, so $\forall i\ |x_i| \leq \|x\|_\infty$ therefore $\frac{|x_i|}{\|x\|_\infty} \leq 1 \Leftrightarrow \left(\frac{|x_i|}{\|x\|_\infty}\right)^p \leq 1^p = 1 \Leftrightarrow \sum_{i=1}^n \left(\frac{|x_i|}{\|x\|_\infty}\right)^p \leq n$)

$\because \|x\|_\infty = \max_{1\leq i\leq n}|x_i|$

$\therefore \|x\|_\infty = \left(\max_{1\leq i\leq n}|x_i|^p\right)^{\frac{1}{p}}$

$\leq \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}}$ (b/c it is clear that $\max_{1\leq i\leq n}|x_i|^p \leq \sum_{i=1}^n |x_i|^p$)

$= \|x\|_p$

$\therefore \|x\|_\infty \leq \|x\|_p \leq \|x\|_\infty \cdot n^{\frac{1}{p}}$

$\therefore$ Take limit to the above inequality

$\lim_{p\to\infty}\|x\|_\infty \leq \lim_{p\to\infty}\|x\|_p \leq \lim_{p\to\infty}\left(\|x\|_\infty \cdot n^{\frac{1}{p}}\right)$

$\Rightarrow \|x\|_\infty \leq \lim_{p\to\infty}\|x\|_p \leq \|x\|_\infty \cdot n^{\frac{1}{\infty}}$

$\Rightarrow \|x\|_\infty \leq \lim_{p\to\infty}\|x\|_p \leq \|x\|_\infty \cdot n^0$ (b/c $\frac{1}{\infty} = 0$)

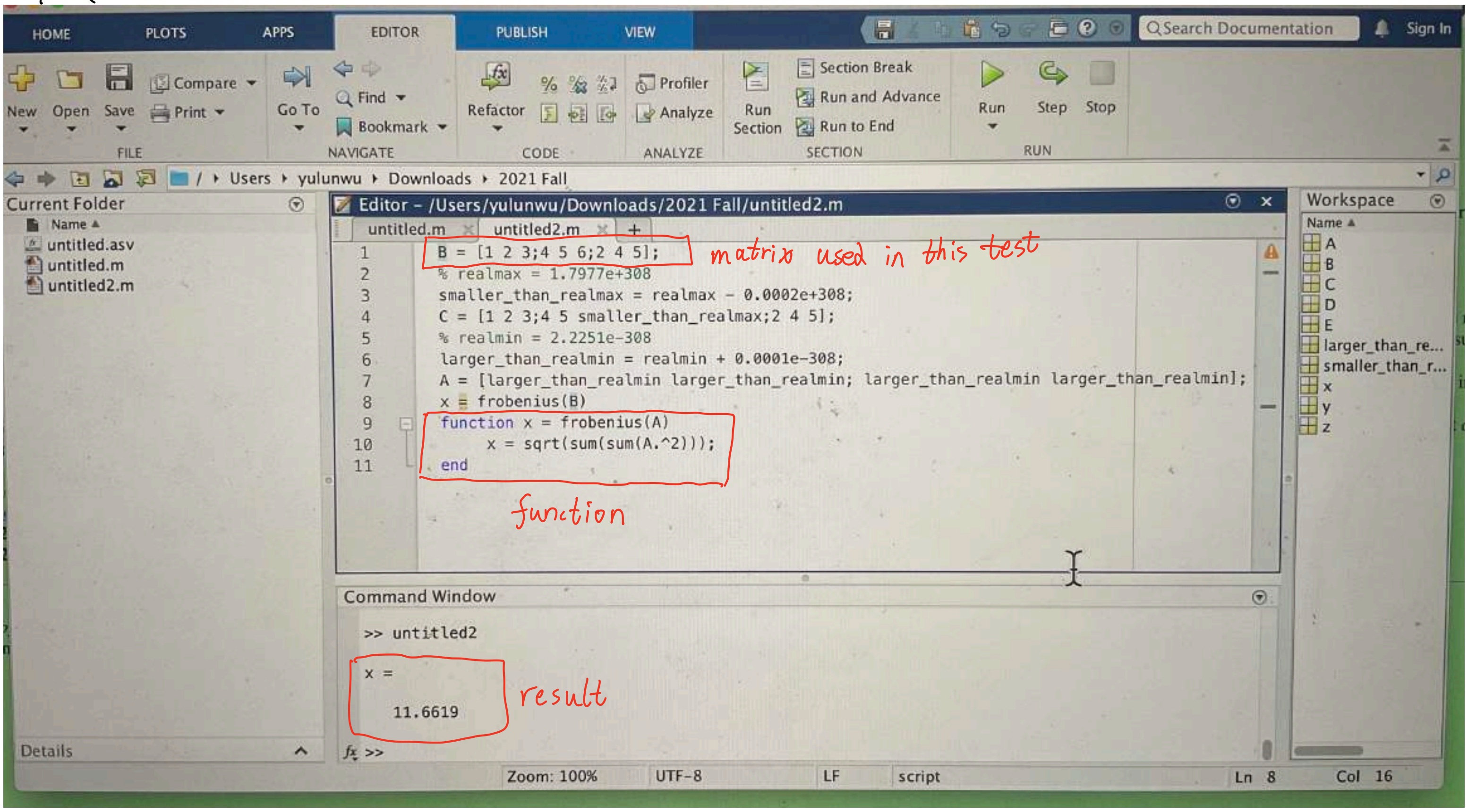
$\Rightarrow \|x\|_\infty \leq \lim_{p\to\infty}\|x\|_p \leq \|x\|_\infty \cdot 1$

$\Rightarrow \|x\|_\infty \leq \lim_{p\to\infty}\|x\|_p \leq \|x\|_\infty$

by squeeze theorem,

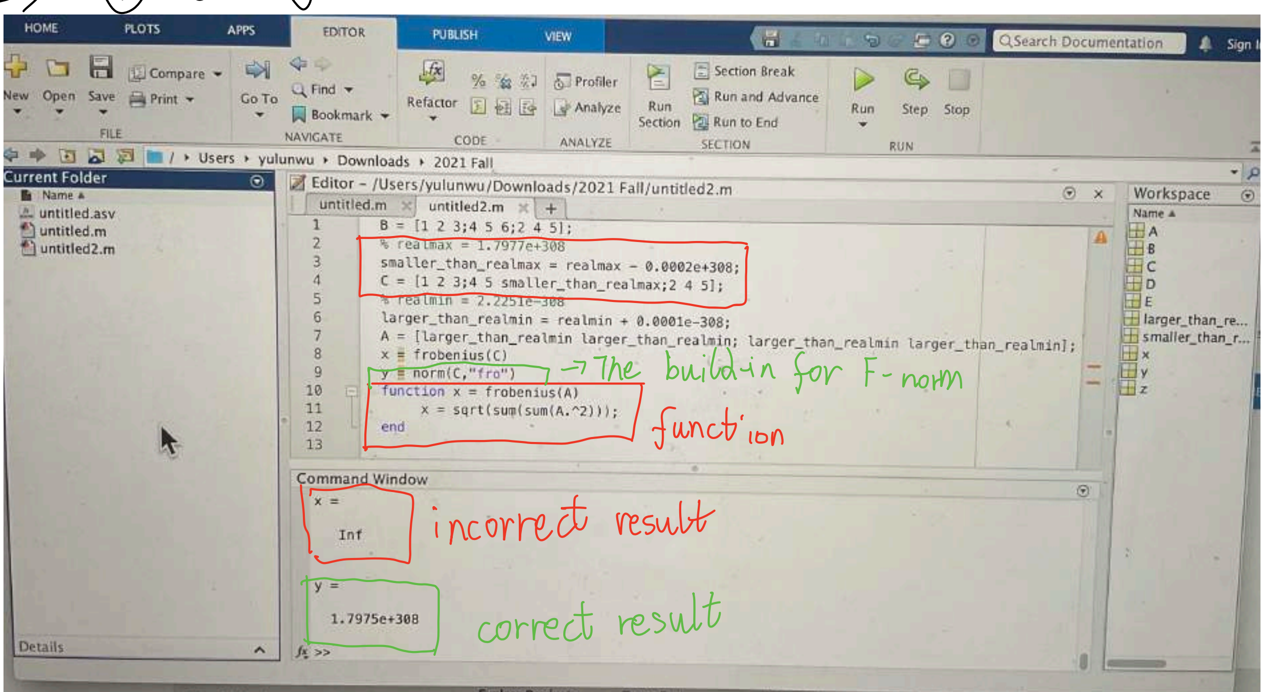
$\|x\|_\infty = \lim_{p\to\infty}\|x\|$ QED

4.(a)



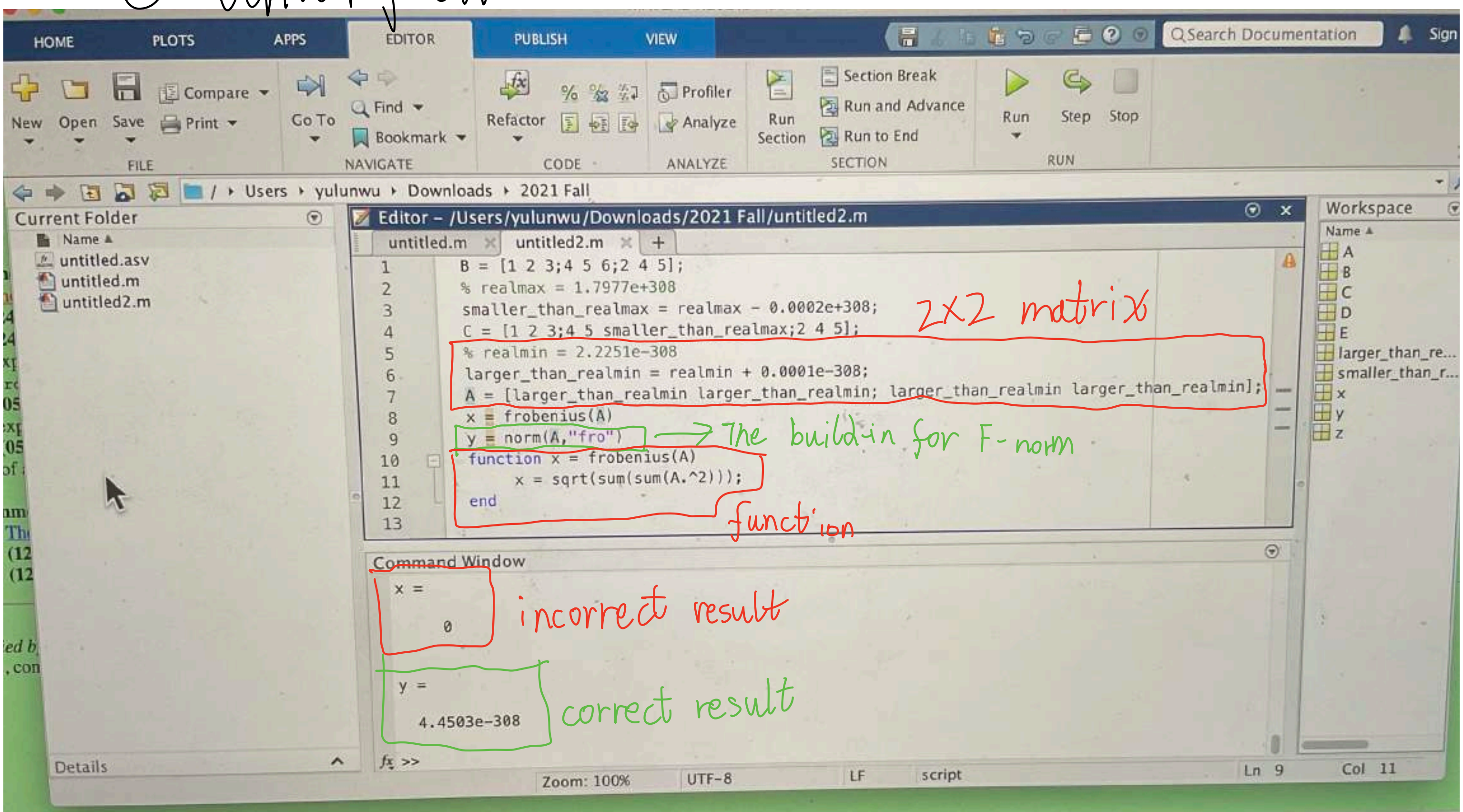
$$\|B\|_F = (1^2+2^2+3^2+4^2+5^2+6^2+2^2+4^2+5^2)^{\frac{1}{2}}$$
$$= (1+4+9+16+25+36+4+16+25)^{\frac{1}{2}}$$
$$= \sqrt{136}$$
$$= 11.6619 \text{ (rounded)}$$

(b) ① overflow case



Frobenius (C) gives inf because when the function squres smaller_tan_realmax, the squared number overflows, but correct result shouldn't overflow because there is a square root at the end of computation.

② underflow



Frobenius (A) gives 0 because when the function squres larger_tan_realmin, all squared number underflow, but correct result shouldn't underflow because there is a square root at the end of computation.

(c)

ulunwu ▸ Downloads ▸ 2021 Fall

Editor – /Users/yulunwu/Downloads/2021 Fall/frobenius1.m *

frobenius1.m * | untitled2.m | +

Overflow case

```
C = [1 2 3;4 5 smaller_than_realmax;2 4 5];
% realmin = 2.2251e-308
larger_than_realmin = realmin + 0.0001e-308;
A = [larger_than_realmin larger_than_realmin; larger_than_realmin larger_than_realmin]
x = frobenius(C)
function x = frobenius(A)
    x = sqrt(sum(sum(A.^2)));
```

explanation

```
    % If x equal to inf, then this means there is an overflow that
    % may cause by squaring entries of A, so in order to avoid the overflow caused
    % by squaring, divide every entry of A by 10^200 before squaring (so the big
    % number that is less than realmax is no more overflow after squaring),
    % multiply 10^200 back after taking square root to cancel the division out.
    if x == inf
        x = (10^200)*sqrt(sum(sum((A./(10^200)).^2)));
```

code added for dealing overflow

```

    % If x equal to 0, then this means there is an underflow that
    % may cause by squaring entries of A, so in order to avoid the underflow caused
    % by squaring, multiply every entry of A by 10^200 before squaring (so the small
```

Command Window

```
>> frobenius1

x =

   1.7975e+308

>>
```

result

NAVIGATE | CODE | ANALYZE | SECTION | RUN

ulunwu ▸ Downloads ▸ 2021 Fall

Editor – /Users/yulunwu/Downloads/2021 Fall/frobenius1.m

frobenius1.m | untitled2.m | +

Underflow case

```
    x = sqrt(sum(sum(A.^2)));

    % If x equal to inf, then this means there is an overflow that
    % may cause by squaring entries of A, so in order to avoid the overflow caused
    % by squaring, divide every entry of A by 10^200 before squaring (so the big
    % number that is less than realmax is no more overflow after squaring),
    % multiply 10^200 back after taking square root to cancel the division out.
    if x == inf
        x = (10^200)*sqrt(sum(sum((A./(10^200)).^2)));
```

explanation

```
    % If x equal to 0, then this means there is an underflow that
    % may cause by squaring entries of A, so in order to avoid the underflow caused
    % by squaring, multiply every entry of A by 10^200 before squaring (so the small
    % number that is bigger than realmin is no more underflow after squaring),
    % divide 10^200 back after taking square root to cancel the multiplcation out.
    elseif x == 0
        x = (sqrt(sum(sum((A.*(10^200)).^2))))/(10^200);
    end
end
```

code added for dealing underflow

Command Window

```
>> frobenius1

x =

   4.4503e-308

>>
```

result

Zoom: 100% | UTF-8 | LF | script / frobenius | Ln 10

Compare to the old version of frobenius(A), the new version add if statements to re-calculate the F-norm if the original cacnlation result in inf or 0
In the worst case of this function, A is originally calculated to be inf or 0 in line 10, then it goes to line 18 (v == inf) or line 26 (v==0).
Line 18 and line 26 are repeat the calculation in line 10 with extra multiplication and division.
The operation cost for the old function and the best case for the new function is:

$$\underbrace{(m\times n)}_{\substack{\text{squaring} \\ .\wedge 2}}+\underbrace{n(m-1)}_{\substack{\text{inner sum} \\ \text{sum( )}}}+\underbrace{(n-1)}_{\substack{\text{outer sum} \\ \text{sum( )}}}+\underbrace{1}_{\substack{\text{square root} \\ \text{sqrt( )}}} = 2mn$$

The operation cost for the new version of function at the worst case is:

$$2(2mn)+\underbrace{mn}_{\substack{\text{operation} \\ \text{before squaring}}}+\underbrace{1}_{\text{operation after square root}} = 5mn+1$$

Therefore, the extra cost needed by the new version of function is $5mn+1-2mn=3mn+1$ more than the old version of function in the worst case.