(1)

$$I = (I + uv^*)(I + \alpha uv^*)$$
$$= I + \alpha uv^* + uv^* + (uv^* \cancel{)} \alpha uv^*)$$
$$0 = \alpha uv^* + uv^* + \alpha (v^* u)(uv^*)$$
$$= (\alpha + 1 + \alpha v^* u)(uv^*) \Rightarrow$$
$$0 = \alpha + \alpha v^* u + 1 = \alpha(1 + v^* u) + 1, \text{ or}$$

$$\boxed{\frac{-1}{(1 + v^* u)} = \alpha}$$

Note that α is undefined if $v^* u = -1$. In this case, A is singular.

To find Nul A, we seek $\vec{k}$ s.t.

$$A\vec{k} = \vec{0}$$
$$(I + uv^*)\vec{k} = \vec{0}$$
$$\vec{k} + u\underbrace{(v^* \vec{k})}_{\text{scalar}} = \vec{0}$$
$$\vec{k} + (v^* \vec{k})u = \vec{0}$$

So $\boxed{\text{Nul } A = \{ \vec{k} \mid (v^* \vec{k})\vec{u} = -\vec{k} \}}$

2. Show: $\rho(A) \leq \|A\|_2$

pf/ (by contradiction)

Let λ be an eigenvalue of a square matrix A, such that $\rho(A) = |\lambda|$, and let x_1 be its associated eigenvector. Assume $\rho(A) > \|A\|_2$. From the eigenvalue equation,

$$A x_1 = \lambda x_1$$

In other words, the action of A has scaled x_1 by a factor of λ. The greatest possible scaling of any vector by A, however, is defined to be $\|A\|_2$. This contradicts the assumption.

$$\therefore \lambda \leq \|A\|_2$$

$\blacksquare$

③ a. $\kappa(A) = \dfrac{\sigma_1}{\sigma_m}$

We can find a lower bound for the condition number by obtaining an <u>upper</u> bound for σ_m, and a lower bound for σ_1.

The upper bound for σ_m is found by examining A^{-1}, as in the in-class worksheet. From Gaussian elimination we see that A^{-1} has the form

$$A^{-1} = \begin{bmatrix} 1 & & & (-2)^{m-1} \\ & 1 & & \vdots \\ & & 1 & (-2)^2 \\ & & \ddots & -2 \\ & & & 1 \end{bmatrix}.$$

Choosing a vector $x = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$ ← (m rows) we see that

$\|A^{-1}\|_2 \geq 2^{m-1}$

To relate this to σ_m, we consider the SVD of A, $A = U \Sigma V^*$, and invert both sides. This yields, $A^{-1} = V \Sigma^{-1} U^*$. The action of a unitary matrix leaves singular values unchanged, and U and V^* (and thus also V and U^*) are unitary, so the singular values of $A^{-1} = SV's(V \Sigma^{-1} U^*) = SV's(\Sigma^{-1})$.

(3) cont'd

Σ is a diagonal matrix with the singular values of A as its entries,

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_m \end{bmatrix} \qquad \sigma_1 \geq \sigma_2 \geq \dots \sigma_m$$

which makes its inverse easy to compute:

$$\Sigma^{-1} = \begin{bmatrix} \sigma_1^{-1} & & & \\ & \sigma_2^{-1} & & \\ & & \ddots & \\ & & & \sigma_m^{-1} \end{bmatrix}.$$

σ_m is the smallest of all the σ_i's, thus σ_m^{-1} is the greatest of all the singular values of A^{-1}. This gives the equality,

$\|A^{-1}\|_2 = \sigma_m^{-1}$, yielding $\sigma_m^{-1} > 2^{m-1}$, the desired upper-bound.

Finding the lower bound for σ_1 is a somewhat simpler process.

$$A = \begin{bmatrix} 1 & 2 & & & \\ & 1 & 2 & & \\ & & 1 & 2 & \\ & & & \ddots & \ddots \\ & & & & \ddots \end{bmatrix}.$$ choosing our x to be a $m \times 1$ vector of ones,

We see that $Ax = \begin{bmatrix} 3 \\ 3 \\ \vdots \\ 1 \end{bmatrix}$, so $\frac{\|Ax\|_2}{\|x\|_2} = \frac{\sqrt{9(m-1)+1}}{\sqrt{m \cdot 1}}$

$= \sqrt{\frac{9m-8}{m}} = \left(9 - \frac{8}{m}\right)^{1/2}$. This let's us say

$\|A\|_2 \geq \left(9 - \frac{8}{m}\right)^{1/2}$. Thus $\sigma_1 \geq \left(9 - \frac{8}{m}\right)^{1/2}$.

Finally, we see that

$$\boxed{\kappa(A) \geq \left(9 - \frac{8}{m}\right)^{1/2} \cdot 2^{m-1}}$$

b. To estimate digit accuracy we examine the product $\varepsilon_{mach} \cdot \kappa(A)$. Taking the $\log_{10}$, (recall $\varepsilon_{mach} \approx 10^{-16}$) and setting it equal to zero,

$0 = \log(10^{-16} \cdot \kappa(x)) = -16 + \log_{10} \kappa(x)$, we see that for all digits to be lost $\kappa(x) \gtrsim 10^{16}$.

Using the result from a., $10^{16} = \left(9 - \frac{8}{m}\right)^{1/2} 2^{(m-1)}$. Logging,

$16 = \frac{1}{2}\log_{10}\left(9 - \frac{8}{m}\right) + (m-1)\log_{10} 2$. For reasonably large m,

$\frac{16 - \frac{1}{2}\log_{10}(9)}{\log_{10} 2} + 1 \approx m$, so $\boxed{m \approx 52}$

4 A = rand(100, 100);

Sample code:

```
int i,j; float num;
float A[100,100];
i=1; j=1;

if i ≤ 100 {
    if j ≤ 100 {
        num = gen_ran();
        A[i,j] = num
        j++;
    }
    i++;
    j=1;
}
```

So $\boxed{\text{2 nested loops}}$

x = 1:.1:100;

sample code:

```
float x[991];
float j;
int i; i=1; j=1;
if i ≤ 991 {
    x[i] = j;
    j = j+.1;
    i++;
    }
```

So $\boxed{\text{1 loop}}$

$b = A \cdot x$ sample code:

```
float A[m,n]; float b[n]; float x[n];
b[n] = [0,0, ... ,0]     (* n zeros *)
int i,j;
i=1; j=1;
 if  i ≤ m {
      if j ≤ n {
          b[i] = b[i] + A[i,j]·x[j]
          j++;
          }
      i++;
      j=1;
 }
```

$\boxed{\text{So 2 loops}}$

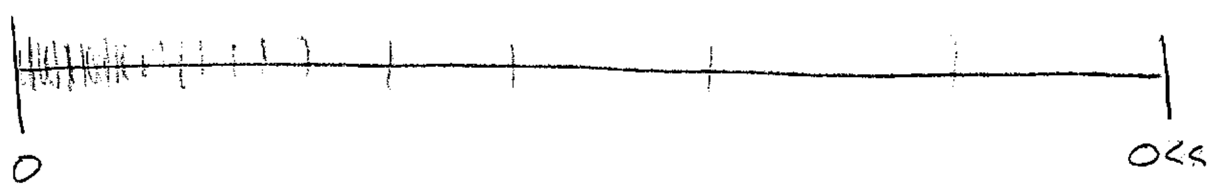
4 cont'd

$$B = A \cdot A$$

This code is the exact same as the previous question, but looped through every column of A.

So there are $\boxed{\text{3 nested loops}}$

(5) All elements in F are of the form $\frac{m}{\beta^t}\beta^e$ where $\beta^{t-1} \le m \le \beta^t - 1$. In the floating point system, the gaps between adjacent elements in F scale proportionate to the magnitude of these elements. Schematically, this means the set is denser close to zero, with gaps increasing as the distance from



zero grows. From this view, we can think of the size of a single "step" as $\frac{\beta^e}{\beta^t} = \beta^{e-t}$. To find the threshold where steps become greater than 1 in size, we choose e and t such that $1 < \beta^{e-t}$.

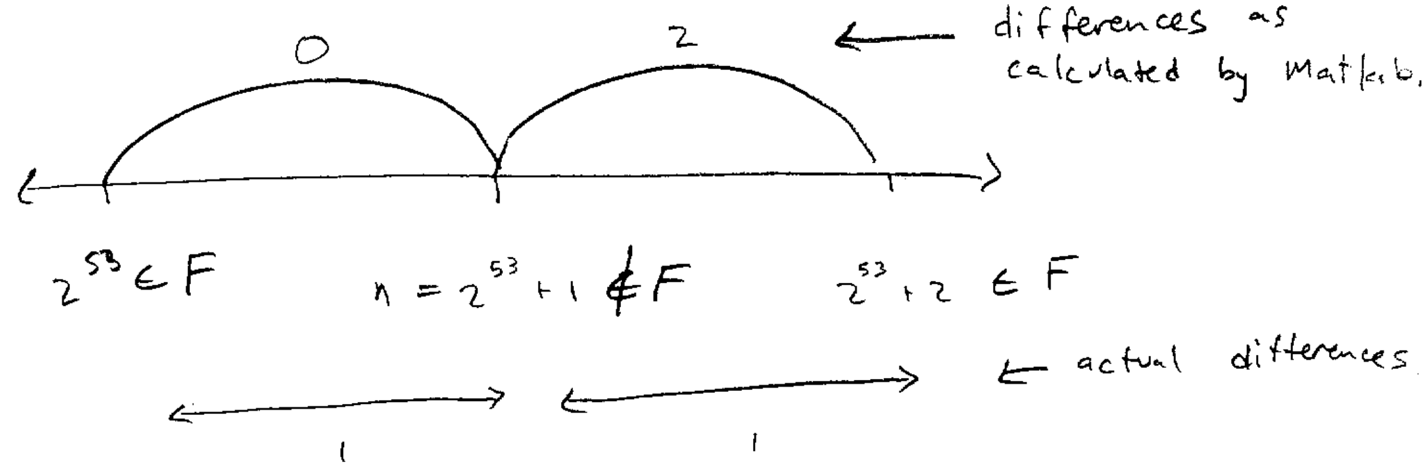
For double precision floating points, $t = 53$, and $\beta = 2$, so we choose $e = 54$. For maximum accuracy, we adopt the lower bound value for m, and we can write

$$n = \frac{\beta^{t-1}}{\beta^t}\beta^e = \frac{\beta^{t-1}\beta^{t+1}}{\beta^t} = \beta^t$$

For the DP case discussed above,

$n = 2^{53}+1$, and for the single precision case,

$n = 2^{24}+1$, ($t = 24$ for SP floating points).



$n \notin F$, so $2^{53}+1$ is approximated by 2^{53}, and the difference, $2^{53}+1 - 2^{53}$, according to Matlab, is 0.

For the same reason, Matlab will tell you that $(2^{53}+2) - (2^{53}+1) = 2$.