FLOPS Computation

Tiep M. H.

Notations: $\mathbb{R}^{m\times n}$ denotes the real field that includes all real-valued matrices of size $m \times n$; $\mathbb{C}^{m\times n}$ denotes the complex field that includes all complex-valued matrices of size $m \times n$; The operation $\mathsf{diag}\left([z_1,\ldots,z_K]\right)$ diagonalizes a row vector $[z_1,\ldots,z_K]$ into a diagonal matrix; Bold lowercase letters and bold uppercase letters denote vectors and matrices, respectively; $\mathbf{I}_n$ denotes the identity matrix of size $n \times n$; The upperscripts $(\cdot)^\top$, $(\cdot)^*$, and $(\cdot)^\dagger$ represent the transpose, conjugate, and Hermitian operators, respectively; $\Re\{\cdot\}$ denotes the real part of a complex-valued matrix; $\Im\{\cdot\}$ denotes the imaginary part of a complex-valued matrix.

Abbreviations: "Mults" stands for "multiplications", and "summs" stands for "summations".

Conventions: We consider that $(a + jb)$ does *not* include any **real** summation. Indeed, we will only count the number of FLOPS based on **real** multiplications and **real** additions.

I. Multiplications

A. Scalar-Vector Multiplication: $\alpha\mathbf{u}$

Let us denote $\mathbf{u} = [u_1,\ldots,u_M]^\top \in \mathbb{C}^{M\times 1}$ and $u_m = a_m + jb_m$.

1) When α is real-valued: We have

$$\alpha\mathbf{u} = [\alpha u_1,\ldots,\alpha u_M]^\top = [\,\underbrace{\alpha a_1 + j\alpha b_1}_{\text{2 real mults}},\ldots,\alpha a_M + j\alpha b_M\,]^\top \tag{1}$$

The number of FLOPS: $2M$ real mults and no real summ.

2) When α is complex-valued: Rewriting α as $\alpha = x + jy$, We have

$$\begin{aligned}\alpha\mathbf{u} &= [\,(x+jy)(a_1+jb_1),\ldots,(x+jy)(a_M+jb_M)\,]^\top \\ &= [\,\underbrace{(xa_1 - yb_1) + j(xb_1 + ya_1)}_{\text{4 real mults and 2 real summs}},\ldots,(xa_M - yb_M) + j(xb_M + ya_M)\,]^\top\end{aligned} \tag{2}$$

The number of FLOPS: $4M$ real mults and $2M$ real summs.

B. Scalar-Matrix Multiplication: $\alpha\mathbf{A}$

Denote $\mathbf{a}_n = [a_{1n},\ldots,a_{Mn}]^\top \in \mathbb{C}^{M\times 1}$, $1 \le n \le N$, as the n-th column vector of $\mathbf{A}$.

1) When α is real-valued: We have

$$\alpha \mathbf{A} = [\ \underbrace{\alpha \mathbf{a}_1, \ldots, \alpha \mathbf{a}_N}_{N \text{ scalar-vector mults}}\ ] = \begin{bmatrix} \alpha\,\Re\{a_{11}\} + j\alpha\,\Im\{a_{11}\}, \ldots, \alpha\,\Re\{a_{1N}\} + j\alpha\,\Im\{a_{1N}\} \\ \vdots \qquad\qquad ,\ldots, \qquad\qquad \vdots \\ \alpha\,\Re\{a_{M1}\} + j\alpha\,\Im\{a_{M1}\}, \ldots, \alpha\,\Re\{a_{MN}\} + j\alpha\,\Im\{a_{MN}\} \end{bmatrix} \tag{3}$$

The number of FLOPS: $N \times (2M) = 2MN$ real mults and no real summ.

2) When α is complex-valued: With $\alpha = x + jy$, We have

$$\alpha \mathbf{A} = [\ \underbrace{\alpha \mathbf{a}_1, \ldots, \alpha \mathbf{a}_N}_{N \text{ scalar-vector mults}}\ ] \tag{4}$$

Each column of $(\alpha \mathbf{A})$ is a scalar-matrix multiplication, where α is complex-valued, thus each column contains $4M$ real mults and $2M$ real summs (see the previous sub-section).

The number of FLOPS: $N \times (4M) = 4MN$ real mults and $N \times (2M) = 2MN$ real summs.

C. Vector-Vector Multiplication: $\mathbf{x}^\dagger \mathbf{y}$

Let us denote $\mathbf{x} = [x_1, \ldots, x_N]^\top \in \mathbb{C}^{N \times 1}$ and $\mathbf{y} = [y_1, \ldots, y_N]^\top \in \mathbb{C}^{N \times 1}$. We have

$$\begin{aligned} \mathbf{x}^\dagger \mathbf{y} &= \sum_{m=1}^{N} x_n^* y_n = \sum_{m=1}^{N} \left(\Re\{x_n\} - j\,\Im\{x_n\}\right)\left(\Re\{y_n\} + j\,\Im\{y_n\}\right) \\ &= \sum_{m=1}^{N} \left[\left(\Re\{x_n\}\,\Re\{y_n\} + \Im\{x_n\}\,\Im\{y_n\}\right) + j\left(\Re\{x_n\}\,\Im\{y_n\} - \Im\{x_n\}\,\Re\{y_n\}\right)\right]. \end{aligned} \tag{5}$$

Denote $a_n = \Re\{x_n\}\,\Re\{y_n\}$, $b_n = \Im\{x_n\}\,\Im\{y_n\}$, $c_n = \Re\{x_n\}\,\Im\{y_n\}$, and $d_n = -\,\Im\{x_n\}\,\Re\{y_n\}$. We can rewrite

$$\begin{aligned} \mathbf{x}^\dagger \mathbf{y} = &\underbrace{\sum_{m=1}^{N} a_n}_{(N-1) \text{ real summs and } N \text{ real mults}} \underbrace{+}_{1 \text{ real summ}} \underbrace{\sum_{m=1}^{N} b_n}_{(N-1) \text{ real summs and } N \text{ real mults}} \\ &+ j \left[\underbrace{\sum_{m=1}^{N} c_n}_{(N-1) \text{ real summs and } N \text{ real mults}} \underbrace{+}_{1 \text{ real summ}} \underbrace{\sum_{m=1}^{N} d_n}_{(N-1) \text{ real summs and } N \text{ real mults}} \right]. \end{aligned} \tag{6}$$

The number of FLOPS: $4N$ real mults and $(4N - 2)$ real summs.

Complexity We have $f(N) = (4N) + (4N - 2) = 8N - 2 \le Mg(N)$ with $M = 8$ and $g(N) = N$. Thus, we have $f(N) = \mathcal{O}(g(N)) = \mathcal{O}(N)$. To be a little more specific, we can also write $f(N) = \mathcal{O}(8N)$.

D. Matrix-vector Multiplication: $\mathbf{Av}$

Let us denote $\mathbf{v} = [v_1, \ldots, v_N]^\top \in \mathbb{C}^{N\times 1}$, $\mathbf{a}_n = [a_{1n}, \ldots, a_{Mn}]^\top \in \mathbb{C}^{M\times 1}$ and $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \in \mathbb{C}^{M\times N}$. We have

$$\mathbf{Av} = \begin{bmatrix} \sum_{n=1}^{N} a_{1n}v_n \\ \vdots \\ \sum_{n=1}^{N} a_{mn}v_n \\ \vdots \\ \sum_{n=1}^{N} a_{Mn}v_n \end{bmatrix} \in \mathbb{C}^{M\times 1}, \tag{7}$$

where the m-th element of $\mathbf{Av}$ is $\sum_{n=1}^{N} a_{mn}v_n$. Recall that

$$a_{mn}v_n = \underbrace{(\Re\{a_{mn}\}\,\Re\{v_n\} - \Im\{a_{mn}\}\,\Im\{v_n\})}_{=\ \Re\{a_{mn}v_n\}\text{ contains 2 real mults and 1 real summ}} + j\underbrace{(\Re\{a_{mn}\}\,\Im\{v_n\} + \Im\{a_{mn}\}\,\Re\{v_n\})}_{=\ \Im\{a_{mn}v_n\}\text{ contains 2 real mults and 1 real summ}}$$

has 4 real mults and 2 real summs, while the following has $4N$ real mults and $(4N-2)$ real summs

$$\begin{aligned} \sum_{n=1}^{N} a_{mn}v_n &= \sum_{n=1}^{N} \Re\{a_{mn}v_n\} + j\sum_{n=1}^{N} \Im\{a_{mn}v_n\} \\ &= \underbrace{\Re\{a_{m1}v_1\}}_{\text{contains 2 real mults, 1 real summ}} + \underbrace{\Re\{a_{m2}v_2\}}_{\text{contains 2 real mults, 1 real summ}} + \ldots + \underbrace{\Re\{a_{mN}v_N\}}_{\text{contains 2 real mults, 1 real summ}} \\ &+ j\Big[\underbrace{\Im\{a_{m1}v_1\}}_{\text{contains 2 real mults, 1 real summ}} + \underbrace{\Im\{a_{m2}v_2\}}_{\text{contains 2 real mults, 1 real summ}} + \ldots + \underbrace{\Im\{a_{mN}v_N\}}_{\text{contains 2 real mults, 1 real summ}}\Big]. \end{aligned} \tag{8}$$

Thus, the m-th element of $\mathbf{Av}$ has $4N$ real mults and $(4N-2)$ real summs. Since the total elements of $\mathbf{Av}$ is M, we have $M \times (4N)$ real mults and $M \times (4N-2)$ real summs.

The number of FLOPS: $4MN$ real mults and $M(4N-2)$ real summs.

Complexity: Let us define $f(M,N) = 4MN + M(4N-2) = 8MN - 2M$ and $g(M,N) = 8MN$. Obviously, $f(M,N) < g(M,N)$ and $\lim_{N\to\infty} f(M,N) \approx g(M,N)$, hence we can have the complexity of $\mathcal{O}(8MN)$.

E. Matrix-matrix Multiplication: $\mathbf{AB}$

Let $\mathbf{A} \in \mathbb{C}^{M\times N}$ and $\mathbf{B} \in \mathbb{C}^{N\times L}$. We have

$$\mathbf{AB} = \begin{bmatrix} \sum_{n=1}^{N} a_{1n}b_{n1}, & \sum_{n=1}^{N} a_{1n}b_{n2}, & \ldots, & \sum_{n=1}^{N} a_{1n}b_{nL} \\ \vdots, & \vdots, & \ddots, & \vdots \\ \sum_{n=1}^{N} a_{Mn}b_{n1}, & \sum_{n=1}^{N} a_{Mn}b_{n2}, & \ldots, & \sum_{n=1}^{N} a_{Mn}b_{nL} \end{bmatrix}. \tag{9}$$

Each element of $\mathbf{AB}$ requires $4N$ real mults and $(4N-2)$ real summs. With ML elements in total, we will need $ML \times (4N)$ real mults and $ML \times (4N-2)$ real summs.

The number of FLOPS: $4MNL$ real mults and $ML(4N-2)$ real summs.

Complexity: The total number of real mults and summs can be expressed as a function of M, N and L, i.e. $f(M,N,L) = (4MNL) + (ML(4N-2)) = 8MNL - 2ML$. Obviously, $f(M,N,L) < g(M,N,L) = 8MNL$. Thus, when $M,N,L \to \infty$, we can write $f(M,N,L) = \mathcal{O}(8MNL)$.

F. Squared Norm: $\|\mathbf{z}\|^2$

Let $\mathbf{z} = [z_1, \ldots, z_N]^\top \in \mathbb{C}^{N\times 1}$. We have

$$\|\mathbf{z}\|^2 = \sum_{m=1}^{N} |z_n|^2 = \sum_{m=1}^{N} [\ (\Re\{z_n\})^2 + (\Im\{z_n\})^2\ ]$$
$$= \underbrace{\left[(\Re\{z_1\})^2 + (\Im\{z_1\})^2\right] + \left[(\Re\{z_2\})^2 + (\Im\{z_2\})^2\right]}_{(2\times 2) \text{ real mults and } (2\times 2-1) \text{ real summs}} + \ldots + \left[(\Re\{z_N\})^2 + (\Im\{z_N\})^2\right]. \tag{10}$$

With N elements in $\mathbf{z}$, we have $N \times 2$ real mults and $(N \times 2 - 1)$ real summs.

The number of FLOPS: $2N$ real mults and $(2N-1)$ real summs.

Complexity: Writing $f(N) = (2N) + (2N-1) = 4N - 1$. We have $f(N) < g(N) = 4N$. Thus, when $N \to \infty$, we can write $f(N) = \mathcal{O}(4N)$.

NOTE: $\|\mathbf{z}\|^2$ is the special case of $\mathbf{x}^\dagger\mathbf{y}$ with $\mathbf{x} = \mathbf{y} = \mathbf{z}$ in Sub-section I-C. If we operate the algorithm in Sub-section I-C, then we incur the cost of $4N$ real mults and $(4N-2)$ real summs. However, if we follow the algorithm of performing (10), the algorithm can reduce the complexity to $2N$ real mults and $(2N-1)$, implying that the execution time reduces 50%.

G. Squared Norm: $\|\mathbf{Av}\|^2$

Let $\mathbf{A} \in \mathbb{C}^{M\times N}$ and $\mathbf{v} \in \mathbb{C}^{N\times 1}$. As for $\mathbf{Av}$, we need $4MN$ real mults and $M(4N-2)$ real summs. Denote $\mathbf{z} = \mathbf{Av} \in \mathbb{C}^{M\times 1}$. Then, we execute the operation $\|\mathbf{z}\|^2$ and have additional $2\times(\text{length of } \mathbf{z}) = 2M$ real mults and $(2M-1)$ real summs. In total, we have $(4MN)+(2M) = 2M(2N+1)$ real mults and $(4MN-2M)+(2M-1) = (4MN-1)$ real summs.

The number of FLOPS: $2M(2N+1)$ real mults and $(4MN-1)$ real summs.

Complexity: The total number of real mults and summs can be expressed as $f(M,N) = 2M(2N+1)+(4MN-1) = 8MN+2M-1$. We see that $f(M,N) < 8MN+2M = g(M,N)$. When $M,N \to \infty$, we can write $f(M,N) = \mathcal{O}(8MN+2M)$.

H. *The form of* $\mathbf{u}^\dagger\mathbf{A}\mathbf{v}$

Suppose we have $\mathbf{u} \in \mathbb{C}^{M\times 1}$, $\mathbf{A} \in \mathbb{C}^{M\times N}$ and $\mathbf{v} \in \mathbb{C}^{N\times 1}$. Defining $\mathbf{z} = \mathbf{A}\mathbf{v}$, we need $4MN$ real mults and $M(4N-2)$ real summs to obtain $\mathbf{z}$. Then, to find $\mathbf{u}^\dagger\mathbf{A}\mathbf{v} = \mathbf{u}^\dagger\mathbf{z}$, we will need $4M$ additional real mults and $(4M-2)$ additional real summs. In total, it is required to have $4M(N+1)$ real mults and $(4MN+2M-2)$ real summs.

The number of FLOPS: $4M(N+1)$ real mults and $(4MN+2M-2)$ real summs.

Complexity: Define $f(M,N)$ as the function of total number of FLOPS. We have $f(M,N) = 4MN+4M+4MN+2M-2 = 8MN+6M-2$. Moreover, $f(M,N) < 8MN+6M = g(M,N)$, we can then write $f(M,N) = \mathcal{O}(8MN+6M)$.

II. Demonstration of counting FLOPS with Matlab

A. *Scalar-Vector Multiplication:* $\alpha\mathbf{u}$

1) *When* α *is real-valued:*

```
M = 6;
alpha = randn(1);
u = randn(M, 1) + 1j*randn(M, 1);

count_Multiplications = 0;
count_Summations = 0;

au_real = zeros(M, 1);
au_imag = zeros(M, 1);

for m=1:M
    au_real(m) = alpha*real(u(m));
    au_imag(m) = alpha*imag(u(m));
    count_Multiplications = count_Multiplications + 2;
    count_Summations = count_Summations + 0;
end

%%
au_ = au_real + 1j .* au_imag;
```

```
au = alpha * u;
error = round(au_ - au, 1)

%%
count_Multiplications
FLOPS_Multiplications_theory = 2*M
```

2) When α is complex-valued:

```
M = 7;
alpha = randn(1) + 1j*randn(1);
u = randn(M, 1) + 1j*randn(M, 1);

count_Multiplications = 0;
count_Summations = 0;

au_real = zeros(M, 1);
au_imag = zeros(M, 1);

for m=1:M
    au_real(m) = real(alpha)*real(u(m)) - imag(alpha)*imag(u(m));
    au_imag(m) = real(alpha)*imag(u(m)) + imag(alpha)*real(u(m));
    count_Multiplications = count_Multiplications + 4;
    count_Summations = count_Summations + 2;
end

%%
au_ = au_real + 1j .* au_imag;
au = alpha * u;
error = round(au_ - au, 1)

%%
count_Multiplications
```

```
FLOPS_Multiplications_theory = 4*M

count_Summations
FLOPS_Summations = 2*M
```

B. *Scalar-Matrix Multiplication:* $\alpha\mathbf{A}$

1) *When* α *is real-valued:*

```
M = 7; N = 8;
alpha = randn(1);  % alpha is real-valued
A = randn(M, N) + 1j*randn(M, N);

aA_real = zeros(M, N);
aA_imag = zeros(M, N);

count_Multiplications = 0;
count_Summations = 0;

for m=1:M
    for n=1:N
        aA_real(m, n) = alpha * real(A(m, n));
        aA_imag(m, n) = alpha * imag(A(m, n));
        count_Multiplications = count_Multiplications + 2;
        count_Summations = count_Summations + 0;
    end
end

%%
aA_ = aA_real + 1j .* aA_imag
aA = alpha .* A
err = round(aA_ - aA, 1)

%%
```

```
count_Multiplications
FLOPS_Multiplications_theory = 2*M*N
```

2) When α is complex-valued:

```
M = 7; N = 9;
alpha = randn(1) + 1j*randn(1);  % alpha is complex-valued
A = randn(M, N) + 1j*randn(M, N);

aA_real = zeros(M, N);
aA_imag = zeros(M, N);

count_Multiplications = 0;
count_Summations = 0;

for m=1:M
    for n=1:N
        aA_real(m, n) = real(alpha)*real(A(m, n)) - imag(alpha)*imag(A(m, n));
        aA_imag(m, n) = real(alpha)*imag(A(m, n)) + imag(alpha)*real(A(m, n));
        count_Multiplications = count_Multiplications + 4;
        count_Summations = count_Summations + 2;
    end
end

%%
aA_ = aA_real + 1j .* aA_imag
aA = alpha .* A
err = round(aA_ - aA, 1)

%%
count_Multiplications
```

```
FLOPS_Multiplications_theory = 4*M*N

count_Summations
FLOPS_Summations_theory = 2*M*N
```

C. Vector-Vector Multiplication: $\mathbf{x}^{\dagger}\mathbf{y}$

```
N = 17;
x = randn(N, 1) + 1j*randn(N, 1);
y = randn(N, 1) + 1j*randn(N, 1);

xH_y_real = zeros(1);
xH_y_imag = zeros(1);

count_Multiplications = 0;
count_Summations = 0;

for n=1:N
    if n==1
        xH_y_real = real(x(n))*real(y(n)) + imag(x(n))*imag(y(n));
        xH_y_imag = real(x(n))*imag(y(n)) - imag(x(n))*real(y(n));
        count_Multiplications = count_Multiplications + 4;
        count_Summations = count_Summations + 2;
    else
        xH_y_real = xH_y_real + real(x(n))*real(y(n)) + imag(x(n))*imag(y(n)
           );
        xH_y_imag = xH_y_imag + real(x(n))*imag(y(n)) - imag(x(n))*real(y(n)
           );
        count_Multiplications = count_Multiplications + 4;
        count_Summations = count_Summations + 4;
    end
end
```

```
%% We have to check if z_ = z or not.
z_ = x' *y;
z = xH_y_real + 1j .* xH_y_imag;
error = round(sum(z_ - z), 1);
error

%% Check if the number of FLOPS counted through simulation matches up with
    that derived theoretically.

FLOPS_Multiplications_theory = 4*N
FLOPS_Summations_theory = 4*N-2

count_Multiplications
count_Summations
```

D. Matrix-vector Multiplication: $\mathbf{Av}$

```
M = 8; N = 17;
A = randn(M, N) + 1j*randn(M, N);
v = randn(N, 1) + 1j*randn(N, 1);

Av_real = zeros(M, 1);
Av_imag = zeros(M, 1);

count_Multiplications = 0;
count_Summations = 0;

for m=1:M
    for n=1:N
        if n==1
            Av_real(m) = real(A(m, n))*real(v(n)) - imag(A(m,n))*imag(v(n));
```

```
            Av_imag(m) = real(A(m, n))*imag(v(n)) + imag(A(m,n))*real(v(n));
            count_Multiplications = count_Multiplications + 4;
            count_Summations = count_Summations + 2;
        else
            Av_real(m) = Av_real(m) + real(A(m, n))*real(v(n)) - imag(A(m,n))*imag(v(n));
            Av_imag(m) = Av_imag(m) + real(A(m, n))*imag(v(n)) + imag(A(m,n))*real(v(n));
            count_Multiplications = count_Multiplications + 4;
            count_Summations = count_Summations + 4;
        end
    end
end

%% We have to check if z_ = z or not.
z_ = A*v;
z = Av_real + 1j .* Av_imag;
error = round(sum(z_ - z), 1);
error

%% Check if the number of FLOPS counted through simulation matches up with that derived theoretically.

FLOPS_Multiplications_theory = 4*M*N
FLOPS_Summations_theory = M*(4*N-2)

count_Multiplications
count_Summations
```

E. Matrix-vector Multiplication: $\mathbf{Av}$

```
M = 8; N = 17; L = 2;
```

```
A = randn(M, N) + 1j*randn(M, N);
B = randn(N, L) + 1j*randn(N, L);
AB_real = zeros(M, L);
AB_imag = zeros(M, L);

count_Multiplications = 0;
count_Summations = 0;


for m=1:M
    for ell=1:L
        AB_real(m, ell) = 0;
        AB_imag(m, ell) = 0;
        for n=1:N
            if n==1
                %% calculate the real part of the (m, ell)-th of AB
                AB_real(m, ell) = real(A(m, n))*real(B(n,ell)) - imag(A(m, n
                    ))*imag(B(n,ell));
                % the above operation has 2 multiplications and 1 summation
                count_Multiplications = count_Multiplications + 2;
                count_Summations = count_Summations + 1;
                %% calculate the imag. part of the (m, ell)-th of AB
                AB_imag(m, ell) = real(A(m, n))*imag(B(n,ell)) + imag(A(m, n
                    ))*real(B(n,ell));
                % the above operation has 2 multiplications and 1 summation
                count_Multiplications = count_Multiplications + 2;
                count_Summations = count_Summations + 1;
            else
                %% calculate the real part of the (m, ell)-th of AB
                AB_real(m, ell) = AB_real(m, ell) + real(A(m, n))*real(B(n,
                    ell)) - imag(A(m, n))*imag(B(n,ell));
```

```
                % the above operation has 2 multiplications and 2 summation
                count_Multiplications = count_Multiplications + 2;
                count_Summations = count_Summations + 2;
                %% calculate the imag. part of the (m, ell)-th of AB
                AB_imag(m, ell) = AB_imag(m, ell) + real(A(m, n))*imag(B(n, ell)) + imag(A(m, n))*real(B(n,ell));
                % the above operation has 2 multiplications and 2 summation
                count_Multiplications = count_Multiplications + 2;
                count_Summations = count_Summations + 2;
            end
        end
    end
end

%%
AB_ = A*B;
AB = AB_real + 1j .* AB_imag;
error = round(sum(AB_ - AB), 1);
error

%%
FLOPS_Multiplications_theory = M*L*(4*N);
FLOPS_Summations_theory = M*L*(4*N-2);

count_Multiplications
FLOPS_Multiplications_theory

count_Summations
FLOPS_Summations_theory
```

F. Squared Norm: $\|\mathbf{z}\|^2$

```
N = 18;
z = randn(N, 1) + 1j*randn(N, 1);

sqNorm = zeros(1);

count_Multiplications = 0;
count_Summations = 0;

for n=1:N
    if n==1
        sqNorm = real(z(n))*real(z(n)) + imag(z(n))*imag(z(n));
        count_Multiplications = count_Multiplications + 2;
        count_Summations = count_Summations + 1;
    else
        sqNorm = sqNorm + real(z(n))*real(z(n)) + imag(z(n))*imag(z(n));
        count_Multiplications = count_Multiplications + 2;
        count_Summations = count_Summations + 2;
    end
end

%% We have to check if || z_ ||^2 = || z ||^2 or not.
sqNorm_ = z' * z;
error = round(sum(sqNorm_ - sqNorm), 1);
error

%% Check if the number of FLOPS counted through simulation matches up with that derived theoretically.

FLOPS_Multiplications_theory = 2*N
FLOPS_Summations_theory = 2*N-1

```

```
count_Multiplications
count_Summations
```

G. Squared Norm: $\|\mathbf{A}\mathbf{v}\|^2$

```
M = 6; N = 15;
A = randn(M, N) + 1j*randn(M, N);
v = randn(N, 1) + 1j*randn(N, 1);

count_Multiplications = 0;
count_Summations = 0;

%% We calculate z = Av and store the real and imag part of z in 2 arrays
z_real = zeros(M, 1);
z_imag = zeros(M, 1);
for m=1:M
    for n=1:N
        if n == 1
            z_real(m) = real(A(m, n))*real(v(n)) - imag(A(m, n))*imag(v(n));
            z_imag(m) = real(A(m, n))*imag(v(n)) + imag(A(m, n))*real(v(n));
            count_Multiplications = count_Multiplications + 4;
            count_Summations = count_Summations + 2;
        else
            z_real(m) = z_real(m) + real(A(m, n))*real(v(n)) - imag(A(m, n))
                *imag(v(n));
            z_imag(m) = z_imag(m) + real(A(m, n))*imag(v(n)) + imag(A(m, n))
                *real(v(n));
            count_Multiplications = count_Multiplications + 4;
            count_Summations = count_Summations + 4;
        end
    end
end
```

```
%% We now calculate || Av ||^2 = || z ||^2 as follows:
sqNorm = zeros(1);
for m=1:M
    if m == 1
        sqNorm = z_real(1)^2 + z_imag(1)^2;
        count_Multiplications = count_Multiplications + 2;
        count_Summations = count_Summations + 1;
    else
        sqNorm = sqNorm + z_real(m)^2 + z_imag(m)^2;
        count_Multiplications = count_Multiplications + 2;
        count_Summations = count_Summations + 2;
    end
end


%% We have to check if || Av ||^2 = sqNorm or not.
sqNorm_ = norm(A*v, 2)^2
sqNorm
error = round(sqNorm_ - sqNorm, 1);
error

%% Check if the number of FLOPS counted through simulation matches up with that derived theoretically.

FLOPS_Multiplications_theory = 2*M*(2*N + 1)
FLOPS_Summations_theory = 4*M*N - 1

count_Multiplications
count_Summations
```

H. *The form of* $\mathbf{u}^{\dagger}\mathbf{A}\mathbf{v}$

```
M = 5; N = 9;
u = randn(M, 1) + 1j*randn(M, 1);
A = randn(M, N) + 1j*randn(M, N);
v = randn(N, 1) + 1j*randn(N, 1);

count_Multiplications = 0;
count_Summations = 0;

%% We calculate z = Av and store the real and imag part of z in 2 arrays
z_real = zeros(M, 1);
z_imag = zeros(M, 1);
for m=1:M
    for n=1:N
        if n == 1
            z_real(m) = real(A(m, n))*real(v(n)) - imag(A(m, n))*imag(v(n));
            z_imag(m) = real(A(m, n))*imag(v(n)) + imag(A(m, n))*real(v(n));
            count_Multiplications = count_Multiplications + 4;
            count_Summations = count_Summations + 2;
        else
            z_real(m) = z_real(m) + real(A(m, n))*real(v(n)) - imag(A(m, n))
                *imag(v(n));
            z_imag(m) = z_imag(m) + real(A(m, n))*imag(v(n)) + imag(A(m, n))
                *real(v(n));
            count_Multiplications = count_Multiplications + 4;
            count_Summations = count_Summations + 4;
        end
    end
end

z = z_real + 1j .* z_imag;
```

```
%% We now calculate a = uH * (Av) = uH * z as follows:
a_real = zeros(1);
a_imag = zeros(1);
for k=1:M
    if k == 1
        a_real = real(u(k))*z_real(k) + imag(u(k))*z_imag(k);
        a_imag = real(u(k))*z_imag(k) - imag(u(k))*z_real(k);
        count_Multiplications = count_Multiplications + 4;
        count_Summations = count_Summations + 2;
    else
        a_real = a_real + real(u(k))*z_real(k) + imag(u(k))*z_imag(k);
        a_imag = a_imag + real(u(k))*z_imag(k) - imag(u(k))*z_real(k);
        count_Multiplications = count_Multiplications + 4;
        count_Summations = count_Summations + 4;
    end
end

%% Check if a_real + 1j .* a_imag = u' * A * v = u' * z or not
u' * z
u' * A * v
a = a_real + 1j * a_imag

%% Check if the number of FLOPS counted through simulation matches up with
    that derived theoretically.
count_Multiplications
count_Summations

FLOPS_Multiplications_theory = 4*M*(N+1)
FLOPS_Summations_theory = 4*M*N + 2*M - 2
```