Final Project

1 Problem 1

(a) After k steps Arnoldi process, we got V_k, H_k. Then $AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T$. From above, when $R = r \in \mathbb{C}$, since $r = R$ and $\beta = ||r||_2$, $M^j R = ||R||_2 M^j V_k e_1 = ||r||_2 V_k H_k^j e_1 = \beta V_k H_k^j e_1$
(b) We use Taylor expansion at s_0 and get

$$Z(s) = \sum_{j=k-1}^{\infty} (s - s_0)^j M_j \tag{1}$$

where $M_j = C^T M^j R$, Same, we can get

$$Z_k(s) = \sum_{j=k-1}^{\infty} (s - s_0)^j M_j^{(k)} \tag{2}$$

where $M_j^{(k)} = C_k^H H_k^j \beta e_1$
From (1),(2) and (a), we can get

$$\begin{aligned} Z(s) - Z_k(s) &= \sum_{j=k-1}^{\infty} (s - s_0)^j (M_j - M_j^{(k)}) \\ &= \sum_{j=k-1}^{\infty} (s - s_0)^j (C^T M^j R - C^H V_k H_k^j \beta e_1) \\ &= \sum_{j=k-1}^{\infty} (s - s_0)^j (C^T - C^H) M^j R \\ &= O((s - s_0)^k) \end{aligned}$$

Therefore, $Z(s) = Z_k(s) + O((s - s_0)^k)$

2 Problem 2

```
function [Q,H] = arnoldi(A,q1,m)
n = length(A);
if nargin < 3, m = n; end
q1 = q1/norm(q1);
Q = zeros(n,m);
Q(:,1) = q1;
H = zeros(m+1,m);
```

(a) Lanczos: Zk(1,1) (b) Arnoldi: Zk(1,1) (c) Z(1,1))

```
for k=1:m
        z = A*Q(:,k);
        for i=1:k
                H(i,k) = Q(:,i)'*z;
                z = z - H(i,k)*Q(:,i);
        end
        if k < n
                H(k+1,k) = norm(z);
        if H(k+1,k) == 0, return, end
                Q(:,k+1) = z/H(k+1,k);
        end
end
H=H(1:m,1:m);
Q=Q(1:n,1:m);
end
```

3 Problem 3

(a) When $A = A^H$, we can choose $c = \bar{r}$ in the nonsymmetric lanczos algorithms. And therefore, $w_n = \bar{v_n}$ for all n. Thus, instead doing two matrix-vector multiplication in then nonsymmetric lanczos case, we can only do once. That is to say, we needn't calculate s, γ. Therefore, each matrix vector product with A^T can be replaced by a matrix vector multiplication with J. Therefore, it saved half calculating time and space. When $A^T J = JA$, we can use the same idea as before. We set $c = \frac{Jr}{||r||}$ then $w_1 = \frac{c}{||c||} = \frac{Jv_1}{||Jv_1||}$. In the later iteration, $w_n = \frac{Jv_n}{|||Jv_n||}$. Therefore, as before, it saved half calculating time and space.
(b) When $A^H J = JA = J^H A = (A^H J)^H$, we can use the same idea as before. We set $c = \frac{Jr}{||r||}$ then $w_1 = \frac{c}{||c||} = \frac{Jv_1}{||Jv_1||}$. In the later iteration, $w_n = \frac{Jv_n}{|||Jv_n||}$. Therefore, as before, it saved half calculating time and space.

4 Problem 4

4.1 Plots of Z and Zk

In the "example1" dataset, we choose s0=f$_max, when k = 130, it convergence. The result of$ Z$_k^{(1,1)}(s)$ for both nonsymmetric lanczos and Arnoldi algorithms and the exact result of $Z^{(1,1)}(s)$ are as follows:
The result of $Z_k^{(1,2)}(s)$ for both nonsymmetric lanczos and Arnoldi algorithms and the exact result of $Z^{(1,2)}(s)$ are as follows:
The result of $Z_k^{(2,2)}(s)$ for both nonsymmetric lanczos and Arnoldi algorithms and the exact result of $Z^{(2,2)}(s)$ are as follows:
The result of $Z_k^{(2,3)}(s)$ for both nonsymmetric lanczos and Arnoldi algorithms and the exact result

(a) Lanczos: Zk(1,2) (b) Arnoldi: Zk(1,2) (c) Z(1,2)

(a) Lanczos: Zk(2,2) (b) Arnoldi: Zk(2,2) (c) Z(2,2)

of $Z^{(2,3)}(s)$ are as follows:

4.1.1 Choise of S0

In "example1"&"example2" data, since we can get the exact number of Z, therefore, when the difference between Z_k and Z is less than 1% of Z, it is a good approximation of Z.

(a) Lanczos: Zk(2,3) (b) Arnoldi: Zk(2,3) (c) Z(2,3)