```
x = ones(100,1);
A = zeros(100,100);
for i = 1:100
    for j = 1:100
        if i == j
            A(i,j) = 5;
        end
        if j == i+2
            A(i,j)=1;
        end
        if j == i-3
            A(i,j)=-2;
        end
    end
end
A
```

```
A =
     5     0     1     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0…
```

```
%create the A x and b that is needed.
w = linspace(0.0001,1.9999,20)
```

```
w =
    0.0001    0.1054    0.2106    0.3159    0.4211    0.5264    0.6316    0.7369
0.8421    0.9474    1.0526    1.1579    1.2631    1.3684    1.4736    1.5789    1.6841
1.7894    1.8946    1.9999
```

```
n= 0;
Relative_Error = norm(x);
iterative_x = zeros(100,50);
r = zeros(100,50);
i = 1;
while Relative_Error > (10^-4)
    r(:,i) = exact_b - A*iterative_x(:,i);
    iterative_x(:,i+1) = iterative_x(:,i) + w(1)*r(:,i);
    error =  iterative_x(:,i+1) - x;
    i = i + 1 ;
    Relative_Error = norm(error);
    n=n+1;
end
n
```

```
n = 29307
```

```
Relative_Error
```

```
Relative_Error = 9.9972e-05
```

I try the other value of w.
For w =0.1054, I get n = 25.
For w = 0.2106, I get n = 19.
And so on.
But somehow, when w is bigger than 0.3159, I found that the relative error turns out to be NaN when the loop ends. So I check the x I get from the loop, x just turns out to be bigger and bigger after serval loops and finally turns to be inf, that is the reason why I get NaN of the relative error.
Later, with the hint from professor, I check the max eigenvalue of the matrix, it is around 7.6, that means w should be smaller than 2.7, which match my result.
I am still curious of why the loop could stop at some large n, with the relative error turns out to be Nan. I mean, why stop at that n, rather than the other.