Hardeep Bassi
09/23/2021

Math 231 Homework #2

1. (a) • Given $x^{(k)} = Bx^{(k-1)} + c$, we see that:

$\hookrightarrow x^{(1)} = Bx_0 + c = Bx_0 + \mathbb{I}c$

$\hookrightarrow x^{(2)} = Bx_1 + c = B(Bx_0 + c) + c = B^2 x_0 + (\mathbb{I} + B)c$

$\hookrightarrow x^{(3)} = Bx_2 + c = B(B^2 x_0 + (\mathbb{I}+B)c) + c$
$= B^3 x_0 + (B + B^2)c + c$
$= B^3 x_0 + (\mathbb{I} + B + B^2)c$

Inductively continuing in this fashion, we see that:

$\hookrightarrow \boxed{x^{(k)} = B^k x_0 + (\mathbb{I} + B + \cdots + B^{k-1})c.}$

• For the sake of contradiction, assume $(\mathbb{I} - B)$ has a 0 eigenvalue. This would the imply, for corresponding eigenvector v,

$\hookrightarrow (\mathbb{I} - B)v = 0v$

$\hookrightarrow v - Bv = 0$

$\hookrightarrow Bv = v.$

Namely, this says 1 is an eigenvalue of B by $Bv = 1 \cdot v$. But, by assumption, we have $\rho(B) < 1$. This means the maximum absolute eigenvalue is less than 1. Hence, it can not be that 0 is an eigenvalue of $(\mathbb{I} - B)$, as this implies 1 is an eigenvalue of B, which contradicts $\rho(B) < 1$. Since $(\mathbb{I} - B)$ has no 0 eigenvalue, $\boxed{(\mathbb{I} - B) \text{ is invertible}}$

• Observe the following:

$\hookrightarrow (\mathbb{I} - B)(\mathbb{I} + B + \cdots + B^{k-1}) = (\mathbb{I} + B + \cdots + B^{k-1}) - (B + B^2 + \cdots + B^k)$

$\hookrightarrow (\mathbb{I} + \cancel{B} - \cancel{B} + \cancel{B^2} - \cancel{B^2} + \cdots + \cancel{B^{k-1}} - \cancel{B^{k-1}} - B^k)$

$\hookrightarrow (\mathbb{I} - B^k)$

$\hookrightarrow (\mathbb{I} - B)(\mathbb{I} + B + \cdots + B^{k-1}) = (\mathbb{I} - B^k)$. Hence, by the previous part, we know $(\mathbb{I} - B)$ is invertible, thus:

$\hookrightarrow \boxed{(\mathbb{I} + B + \cdots + B^{k-1}) = (\mathbb{I} - B)^{-1}(\mathbb{I} - B^k)}$

$\longrightarrow$

1) (a) • By $\rho(B) < 1 \Rightarrow B^k \to 0$, we see that regardless of choice of x_0, $B^k x_0 \to 0$. Hence, observing our iteration form:

$\hookrightarrow x^{(k)} = B^k x_0 + (I + B + \cdots + B^{k-1})c$

$= B^k x_0 + (I - B)^{-1}(I - B^k)c$

$\therefore$ as $k \to \infty$, $x^{(k)} \to 0 + (I - B)^{-1}c$ by $B^k \to 0$.

Hence, regardless of choice of x_0,

$\boxed{x^{(k)} \to (I - B)^{-1}c}$ due to $B^k \to 0$ for $\rho(B) < 1$.

This means, $\rho(B) < 1 \Rightarrow x^{(k)} = Bx^{(k-1)} + c$ converges for any choice of x_0. ✓

(b) • If $e^{(k)} = x^{(k)} - x^*$ for x^* the convergent solution, then by $x^{(k)} = Bx^{(k-1)} + c$, we see:

$\hookrightarrow e^{(k)} = Bx^{(k-1)} + \cancel{c} - Bx^* - \cancel{c}$

$= Bx^{(k-1)} - Bx^*$

$= B(x^{(k-1)} - x^*)$

$= Be^{(k-1)}$

Since $e^{(k)} = Be^{(k-1)}$, iterating downwards implies:

$\hookrightarrow e^{(k)} = Be^{(k-1)} = B(Be^{(k-2)}) = B^2 e^{(k-2)} = \cdots = B^k e^{(0)}$

By definition of $e^{(0)}$, this becomes:

$\hookrightarrow \boxed{e^{(k)} = B^k e^{(0)} = B^k(x_0 - x^*)}$ ✓

• For the sake of contradiction, assume $\rho(B) \geq 1$. Since we know we have convergence for any choice of x_0, let's choose $x_0 = x^* + u$, for u an eigenvector of B. Observe:

$\hookrightarrow e^{(0)} = x_0 - x^* = x^* + u - x^* = u$.

By the previous part, we see $e^{(k)} = B^k e^{(0)}$, thus:

$\hookrightarrow e^{(k)} = B^k u = \lambda^k u$ by u an eigenvector, with $|\lambda| = \rho(B)$.

Hence, as $k \to \infty$, $e^{(k)} = \lambda^k u \not\to 0$ by $\lambda \geq 1$ according to our hypothesis. Observing the norm of the error obtains:

$\hookrightarrow \|e^{(k)}\| = \|\lambda^k u\|$

$= |\lambda|^k \|u\| \neq 0$ by u an eigenvector and $\lambda \geq 1$.

Hence, $\forall k$, $e^{(k)} \neq 0$, which implies our choice of x_0 does not converge, contradiction our assumption $\boxed{\text{Hence, } \rho(B) < 1}$.

② (a) Using the definition of $x^* = (\mathbb{I}-B)^{-1}c$ by $x^* = Bx^* + c$ and $x^{(k)} = B^k x_0 + (\mathbb{I} + B + \cdots + B^{k-1})c$, observe:

↳ $x^{(k)} - x^* = B^k x_0 + (\mathbb{I} + B + \cdots + B^{k-1})c - (\mathbb{I}-B)^{-1}c$

$$= B^k x_0 + ((\mathbb{I}+B+\cdots+B^{k-1}) - (\mathbb{I}-B)^{-1})c$$

$$= B^k x_0 + (\mathbb{I}-B)^{-1}((\cancel{\mathbb{I}}-B^k) - \cancel{\mathbb{I}})c$$

by $\sum_{n=0}^{k-1} B^n = (\mathbb{I}-B)^{-1}(\mathbb{I}-B^k)$

$$= B^k x_0 + (\mathbb{I}-B)^{-1}(-B^k)c$$

$$= -B^k(\mathbb{I}-B)^{-1}(c - (\mathbb{I}-B)x_0)$$

$$= -B^k(\mathbb{I}-B)^{-1}(c - x_0 + Bx_0)$$

$$= -B^k(\mathbb{I}-B)^{-1}(Bx_0 + c - x_0)$$

$$= -B^k(\mathbb{I}-B)^{-1}(x_1 - x_0)$$

$$\therefore \boxed{x^{(k)} - x^* = -B^k(\mathbb{I}-B)^{-1}(x_1 - x_0)}$$

(b) To prove the inequality, observe the following:

↳ $\mathbb{I} = (\mathbb{I}-B) + B$

↳ $(\mathbb{I}-B)^{-1} = ((\mathbb{I}-B) + B)(\mathbb{I}-B)^{-1}$

↳ $(\mathbb{I}-B)^{-1} = \mathbb{I} + B(\mathbb{I}-B)^{-1}$

Now, observe the norm:

↳ $\|(\mathbb{I}-B)^{-1}\| = \|\mathbb{I} + B(\mathbb{I}-B)^{-1}\|$

$\leq \|\mathbb{I}\| + \|B(\mathbb{I}-B)^{-1}\|$ by the triangle inequality

$= 1 + \|B(\mathbb{I}-B)^{-1}\|$

$\leq 1 + \|B\| \cdot \|(\mathbb{I}-B)^{-1}\|$ by $\|AB\| \leq \|A\| \cdot \|B\|$

Hence we get,

↳ $\|(\mathbb{I}-B)^{-1}\| \leq 1 + \|B\| \cdot \|(\mathbb{I}-B)^{-1}\|$

↳ $\|(\mathbb{I}-B)^{-1}\|(1 - \|B\|) \leq 1$.

Since $\|B\| < 1$, we know $(1 - \|B\|) > 0$, hence we can divide to get:

↳ $\boxed{\|(\mathbb{I}-B)^{-1}\| \leq \dfrac{1}{1-\|B\|}}$ ✓

Since (a) tells us $x^{(k)} - x^* = -B^k(\mathbb{I}-B)^{-1}(x_1 - x_0)$

↳ $\|x^{(k)} - x^*\| = \|-B^k(\mathbb{I}-B)^{-1}(x_1 - x_0)\|$

$\leq \dfrac{\|B\|^k}{1-\|B\|}\|x^{(1)} - x_0\|$ by part (b) ✓

Scanned with CamScanner

MATH 231 Homework 2 MATLAB

1. Write the following `MATLAB` function for the SOR iteration method to solve a general nxn system `A * x = b`:

   - `function [final sol,sols] = SOR(A,b,x0,niter, omega)`

   Here, `omega` is the relaxation parameter ω, `final_sol` is the iterative solution after `niter` iterations starting with initial guess `x0` and `sols` is the sequence of iterative solutions $\{x^{(0)}, x^{(1)}, ..., x^{(niter)}\}$ (n x $(niter + 1)$ matrix).

SOLUTION:

```
SOR METHOD

function [final_sol,sols] = SOR(A,b,x0,niter,omega)
x = x0;
sols = [x];
n = size(A);
for k = 1:niter
    for i= 1:n
        x(i) = omega * ((b(i) - A(i, 1:i-1) * x(1:i-1) - A(i, i+1:n) *
            x(i+1:n))/ (A(i,i))) + (1-omega)*x(i);
    end
    sols = [sols x];
end
final_sol = sols(:,end);
```

Using the psuedocode from lecture, the SOR method is implemented in MATLAB as shown above. This method uses a weight parameter, `omega`, and can update its `x` values using previously calculated values, similar to how Gauss-Seidel does. In the case of `omega` $= 1$, we see that the SOR method actually reduces to the Gauss-Seidel method.

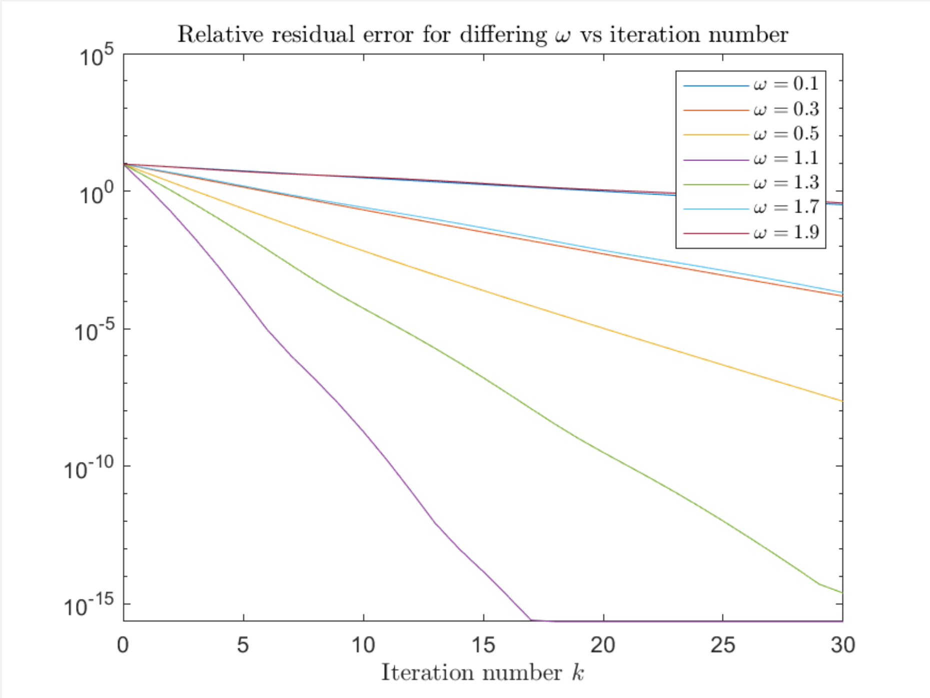
2. Choose a 6 x 6 symmetric positive-definite matrix. For several values of $0 < \omega < 2$, compare convergence rates by plotting the semi-log plot of the relative residual errors versus the iteration number.

SOLUTION:

Let

$$\texttt{A} = \begin{bmatrix} 10 & 1 & 0 & 0 & 0 & 0 \\ 1 & 20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 7 \\ 0 & 0 & 0 & 0 & 7 & 60 \end{bmatrix}, \ \texttt{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, \ \texttt{x0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

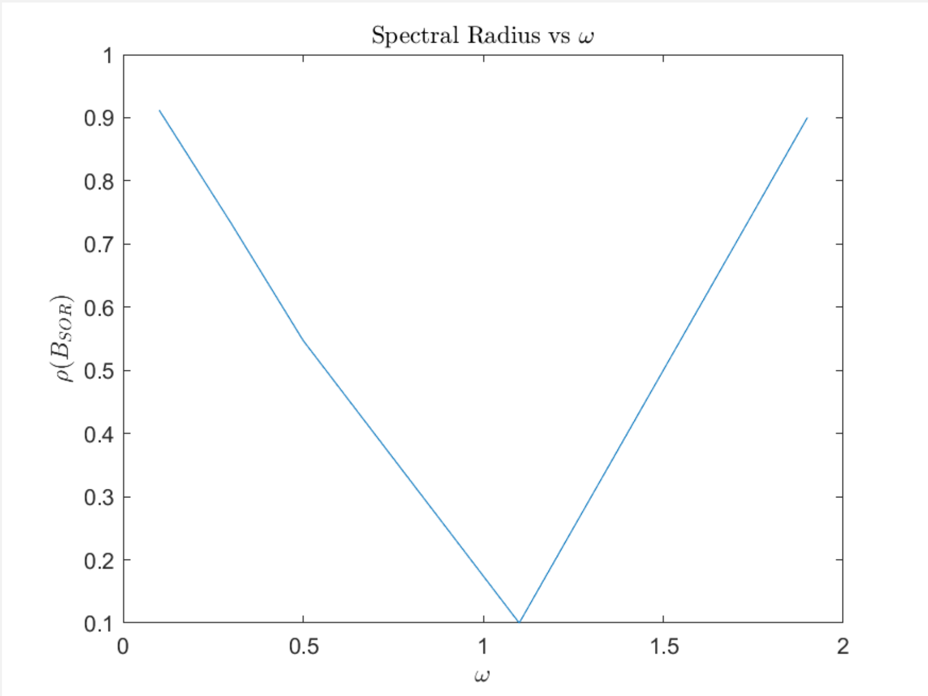
along with `niter` = 30. Using `MATLAB`, we see that the eigenvalues of `A` are [9.9, 20.099, 30, 40, 46.3977, 63.6023]. Since they are all larger than 0, `A` is considered positive definite. By inspection, `A` is also symmetric as $\texttt{A}^T = \texttt{A}$. Considering different values of ω from the array `omegas` $= [0.1, 0.3, 0.5, 1.1, 1.3, 1.7, 1.9]$, the script `plotSPD.m` attached in the appendix generates the image below. We observe the residual error for `niter` iterations using the SOR method with each `omega` value listed previously. The plot shows that too low or high values for `omega`, while still having low residual error tending towards 0, converge rather slowly as compared to a more intermediate value between $0 < \omega < 2$ for the SOR method on our `A*x=b` system. We see that values below 1 have slow convergence behaviors, and values above 1.3 also have slow convergence behavior. Although 1.3 does eventually reach a low relative residual error, we see that $\omega = 1.1$ achieves the lowest computable relative residual error in far less iterations, making this our optimal value from the values considered.



3. For the same matrix, write a script to plot the spectral radius $\rho(B_{SOR})$ versus $0 < \omega < 2$. Locate the optimal value of ω.

SOLUTION:

From lecture, we know that the B_{SOR} matrix can be computed as:
$B_{SOR} = (D+\omega L)^{-1}((1-\omega)D-\omega U)$, where L, D, U are obtained from decomposing A as a lower triangular, diagonal, and upper triangular matrix respectively. From here, we can calculate different $\rho(B_{SOR})$ depending on which ω is chosen, since differing values of ω change the iteration matrix directly. As we can see, all values of ω produce $\rho(B_{SOR}) < 1$, which will indicate convergence for the SOR method. However, as we saw in Homework 1, the lowest spectral radius will produce the fastest convergence. We see that the spectral radius decreases to its minimal value of 0.1 as we progress ω from 0 to 1.1 , and as a result, that the rate of convergence in Question 2 begins tending to the lowest computable relative residual error quicker. Then as we increase ω from 1.1, the spectral radius begins increasing, and thus we see that the rate of convergence for the SOR method is not as quick as it was with $\omega = 1.1$. The figure below shows that $\omega = 1.1$ has the lowest spectral radius value compared to the other values used, hence it should converge the fastest. Looking back to Question 2 on this assignment, we see that $\omega = 1.1$ produces the fastest convergence to the lowest computable relative residual error. Hence, this figure confirms that our optimal value of ω is indeed $\omega = 1.1$.



APPENDIX

plotSPD.m

```
clear all; close all; clc;

% problem setup

%Symmetric positive definite
A = [10 1 0 0 0 0; 1 20 0 0 0 0; 0 0 30 0 0 0; 0 0 0 40 0 0; 0 0 0 0 50 7; 0
    0 0 0 7 60];
b=[1;2;3;4;5;6];
x0=[0;0;0;0;0;0];
niter=30;
omegas = [0.1, 0.3, 0.5, 1.1, 1.3, 1.7, 1.9];



[final_sol1, sols1] = SOR(A,b,x0,niter,0.1);
errs1 = [];
for j=1:niter+1
    err=norm(A*sols1(:,j) -b);
    errs1 = [errs1 err];
end
semilogy(0:niter, errs1)


[final_sol2, sols2] = SOR(A,b,x0,niter,0.3);
errs2 = [];
for j=1:niter+1
    err=norm(A*sols2(:,j) -b);
    errs2 = [errs2 err];
end
hold on
semilogy(0:niter, errs2)
hold off

[final_sol3, sols3] = SOR(A,b,x0,niter,0.5);
errs3 = [];
for j=1:niter+1
    err=norm(A*sols3(:,j) -b);
    errs3 = [errs3 err];
end
hold on
semilogy(0:niter, errs3)
hold off

[final_sol4, sols4] = SOR(A,b,x0,niter,1.1);
errs4 = [];
for j=1:niter+1
    err=norm(A*sols4(:,j) -b);
    errs4 = [errs4 err];
end
hold on
semilogy(0:niter, errs4)
hold off
```

```
[final_sol5, sols5] = SOR(A,b,x0,niter,1.3);
errs5 = [];
for j=1:niter+1
    err=norm(A*sols5(:,j) -b);
    errs5 = [errs5 err];
end
hold on
semilogy(0:niter, errs5)
hold off

[final_sol6, sols6] = SOR(A,b,x0,niter,1.7);
errs6 = [];
for j=1:niter+1
    err=norm(A*sols6(:,j) -b);
    errs6 = [errs6 err];
end
hold on
semilogy(0:niter, errs6)
hold off

[final_sol7, sols7] = SOR(A,b,x0,niter,1.9);
errs7 = [];
for j=1:niter+1
    err=norm(A*sols7(:,j) -b);
    errs7 = [errs7 err];
end
hold on
semilogy(0:niter, errs7)
hold off


xlabel("Iteration number $k$", 'Interpreter', 'latex')
title('Relative residual error for differing $\omega$ vs iteration number',
    'Interpreter', 'latex')
legend('$\omega = 0.1$', '$\omega = 0.3$', '$\omega = 0.5$', '$\omega =
    1.1$', '$\omega = 1.3$', '$\omega = 1.7$','$\omega = 1.9$',
    'Interpreter', 'latex')
saveas(gcf, 'comparing_omega.png')
```

```
plotspec.m
clear all; close all; clc;

% problem setup

%Symmetric positive definite
A = [10 1 0 0 0 0; 1 20 0 0 0 0; 0 0 30 0 0 0; 0 0 0 40 0 0; 0 0 0 0 50 7; 0
    0 0 0 7 60];
b=[1;2;3;4;5;6];
x0=[0;0;0;0;0;0];
niter=30;
omegas = [0.1, 0.3, 0.5, 1.1, 1.3, 1.7, 1.9];

D = diag(diag(A));
L = tril(A, -1);
U = triu(A,1);
spec_radii = [];
for i=1:length(omegas)
    BSOR = inv(D+ omegas(i) *L) * ( (1-omegas(i)) * D - omegas(i)* U);
    radius = max(abs(eig(BSOR)));
    spec_radii = [spec_radii radius];
end
plot(omegas, spec_radii)



xlabel("$\omega$", 'Interpreter', 'latex')
ylabel("$\rho(B_{SOR})$", 'Interpreter', 'latex')
title('Spectral Radius vs $\omega$', 'Interpreter', 'latex')
saveas(gcf, 'radius_vs_omega.png')
```