Caleb Logemann
MATH 562 Numerical Analysis II
Final Exam

1. Let $A \in \mathbb{R}^{m \times m}$ be written in the form $A = L + D + U$, where L is strictly lower triangular, D is the diagonal of A, and U is the strictly upper triangular part of A. Assuming D is invertible, $A\mathbf{x} = \mathbf{b}$ is equivalent to $\mathbf{x} = -D^{-1}(L+U)\mathbf{x} + D^{-1}\mathbf{b}$. The Jacobi iteration method for solving $A\mathbf{x} = \mathbf{b}$ is defined by

$$\mathbf{x}^{(n+1)} = -D^{-1}(L+U)\mathbf{x}^{(n)} + D^{-1}\mathbf{b}$$

Show that if A is nonsingular and strictly row diagonally dominant:

$$0 < \sum_{j \neq i} (|a_{ij}|) < |a_{ii}|$$

then the Jacobi iteration converges to $\mathbf{x}_* = A^{-1}\mathbf{b}$ for each fixed $\mathbf{b} \in \mathbb{R}^m$.

Proof. Let $\mathbf{e}_n$ be the error of the nth iteration of the Jacobi iteration from the actual solution, that is let

$$\mathbf{e}_n = \mathbf{x}^{(n)} - \mathbf{x}_*.$$

The Jacobi iteration converges to the real solution if

$$\lim_{n \to \infty} (\|\mathbf{e}_n\|_\infty) = 0$$

The error vector can be expressed recursively by noting that $\mathbf{x}^{(n)}$ is the Jacobi iteration evaluated on $\mathbf{x}^{(n-1)}$ and that $\mathbf{x}_*$ is a fixed point of the Jacobi iteration as it is the true solution to the linear system. This means that

$$\begin{aligned}
\mathbf{x}^{(n)} &= -D^{-1}(L+U)\mathbf{x}^{(n-1)} + D^{-1}\mathbf{b} \\
\mathbf{x}_* &= -D^{-1}(L+U)\mathbf{x}_* + D^{-1}\mathbf{b}.
\end{aligned}$$

Therefore we can express the error recursively as

$$\begin{aligned}
\mathbf{e}_n &= \mathbf{x}^{(n)} - \mathbf{x}_* \\
\mathbf{e}_n &= \left(-D^{-1}(L+U)\mathbf{x}^{(n-1)} + D^{-1}\mathbf{b}\right) - \left(-D^{-1}(L+U)\mathbf{x}_* + D^{-1}\mathbf{b}\right) \\
\mathbf{e}_n &= -D^{-1}(L+U)\mathbf{x}^{(n-1)} + D^{-1}(L+U)\mathbf{x}_* \\
\mathbf{e}_n &= -D^{-1}(L+U)\left(\mathbf{x}^{(n-1)} - \mathbf{x}_*\right) \\
\mathbf{e}_n &= -D^{-1}(L+U)\mathbf{e}_{n-1}.
\end{aligned}$$

Extrapolating this backwards we see that $\mathbf{e}_n$ can be expressed in terms of $\mathbf{e}_0$

$$\mathbf{e}_n = \left(-D^{-1}(L+U)\right)^n \mathbf{e}_0.$$

Now we can consider the limit of $\|\mathbf{e}_n\|_\infty$ as n goes to infinity.

$$\lim_{n\to\infty}(\|\mathbf{e}_n\|_\infty) = \lim_{n\to\infty}\left(\left\|\left(-D^{-1}(L+U)\right)^n \mathbf{e}_0\right\|_\infty\right)$$
$$\lim_{n\to\infty}(\|\mathbf{e}_n\|_\infty) \leq \|\mathbf{e}_0\|_\infty \lim_{n\to\infty}\left(\left\|D^{-1}(L+U)\right\|_\infty^n\right)$$

Now consider $\|D^{-1}(L+U)\|_\infty$. The infinity norm is the max row sum of the matrix, that is

$$\left\|D^{-1}(L+U)\right\|_\infty = \max *_{1\leq i\leq m} \sum_{j=1}^{m}\left(\left|\left(D^{-1}(L+U)\right)_{ij}\right|\right)$$

Because $L+U = A-D$, $(L+U)_{ij} = a_{ij}$ if $i \neq j$ and $(L+U)_{ii} = 0$. Also D^{-1} is diagonal with $(D^{-1})_{ii} = \frac{1}{D_{ii}} = \frac{1}{a_{ii}}$. Therefore the matrix product $D^{-1}(L+U)$ has entries $(D^{-1}(L+U))_{ij} = \frac{a_{ij}}{a_{ii}}$ if $i \neq j$ or if $i = j$, then $(D^{-1}(L+U))_{ii} = 0$. We can now say that

$$\left\|D^{-1}(L+U)\right\|_\infty = \max *_{1\leq i\leq m} \sum_{j\neq k}\left(\left|\frac{a_{ij}}{a_{ii}}\right|\right)$$
$$\left\|D^{-1}(L+U)\right\|_\infty = \max *_{1\leq i\leq m} \frac{1}{|a_{ii}|}\sum_{j\neq k}(|a_{ij}|)$$

However since A is strictly row diagonally dominant $|a_{ii}| > \sum_{j\neq k}(|a_{ij}|)$, we can conclude that $\frac{1}{|a_{ii}|}\sum_{j\neq k}(|a_{ij}|) < 1$. Therefore

$$\left\|D^{-1}(L+U)\right\|_\infty < 1$$

Since $\|D^{-1}(L+U)\|_\infty < 1$, it is true that $\lim_{n\to\infty}\left(\|D^{-1}(L+U)\|_\infty^n\right) = 0$. Thus

$$\lim_{n\to\infty}(\|\mathbf{e}_n\|_\infty) \leq \|\mathbf{e}_0\|_\infty \lim_{n\to\infty}\left(\left\|D^{-1}(L+U)\right\|_\infty^n\right)$$
$$\lim_{n\to\infty}(\|\mathbf{e}_n\|_\infty) \leq 0$$

This shows that the error converges to zero, and this proves that the Jacobi iteration does converge to the true solution if A is strictly row diagonally dominant. □

2. Let $A \in \mathbb{R}^{m\times m}$ be symmetric positive definite (SPD), $\mathbf{b} \in \mathbb{R}^m$ and define $\phi : \mathbb{R}^m \to \mathbb{R}$ by

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$$

Suppose K is a subspace of $\mathbb{R}^m$. Show that $\hat{\mathbf{x}} \in K$ minimizes $\phi(\mathbf{x})$ over K if and only if $\nabla\phi(\hat{\mathbf{x}}) \perp K$.

Proof. $\square$

3.

4. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Let Gaussian elimination be carried out on A without pivoting. After k steps, A will be reduced to the form

$$A^{(k)} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix}$$

where $A_{22}^{(k)}$ is an $(n-k) \times (n-k)$ matrix. Show by induction

(a) $A_{22}^{(k)}$ is symmetric positive definite.

Proof. $\square$

(b) $a_{ii}^{(k)} \leq a_{ii}^{(k-1)}$ for all $k \leq i \leq n$, $k = 1, \cdots, n-1$.

Proof. $\square$

5. Let $A \in \mathbb{R}^{m\times n}$ with $m > n$ and

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

where A_1 is a nonsingular $n \times n$ matrix, and A_2 is an $(m-n) \times n$ arbitrary matrix.

6. Let $A \in \mathbb{C}^{m\times m}$ with $\operatorname{rank}(A) = r$. Suppose an SVD of A is given by $A = U\Sigma V^*$, where $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$ denote the columns of U and $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ denote the columns of V. Prove that $\langle \mathbf{v}_{r+1}, \ldots, \mathbf{v}_m \rangle = (A)$.

Proof. $\square$

7. Problem 33.2 (Page 255)

8. Problem 36.1 (Page 283)

9.

10. MATLAB project

Below are the function for performing the Jacobi method, the Gauss-Seidel Method, and the Conjugate Gradient method.

```
function [x0, k, r] = Jacobi(A, b, tol, maxIter)
    M = diag(diag(A));
    N = M - A;

    x0 = zeros(size(b));
    k = 0;
    r = norm(b - A*x0, inf);
    while(r(end) > tol && k < maxIter)
        k = k + 1;
        x = M\(N*x0) + M\b;
        x0 = x;
        r = [r, norm(b - A*x0, inf)];
    end
end
```

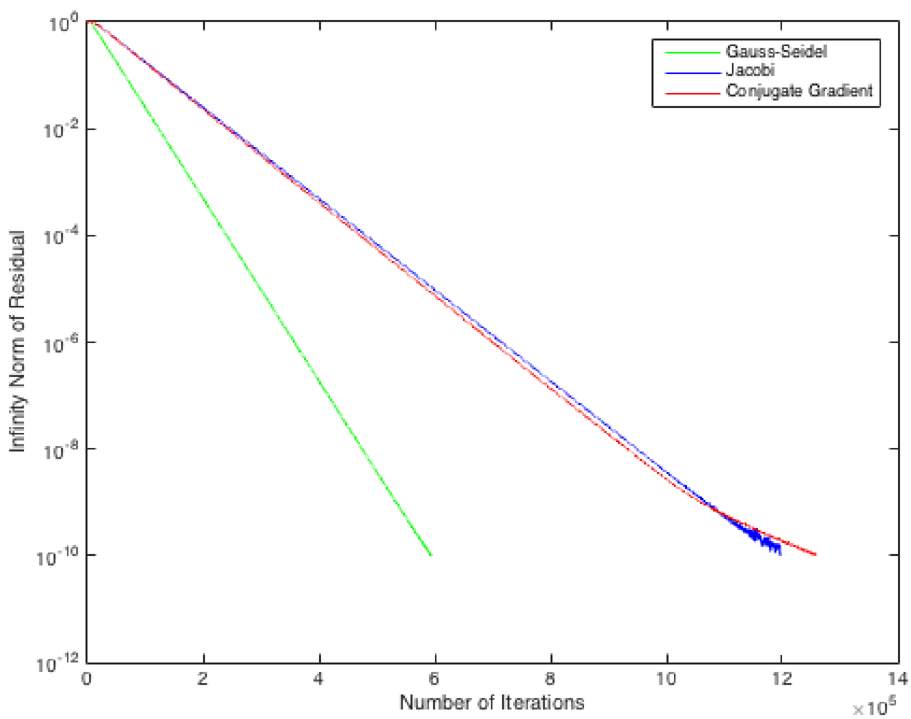
```
function [x0, k, r] = GaussSeidel(A, b, tol, maxIter)
    M = tril(A);
    N = M - A;

    x0 = zeros(size(b));
    k = 0;
    r = norm(b - A*x0, inf);
    while(r(end) > tol && k < maxIter)
        k = k + 1;
        x = M\(N*x0) + M\b;
        x0 = x;
        r = [r, norm(b - A*x0, inf)];
    end
end
```

```
function [x0, k, r] = ConjugateGradient(A, b, tol, maxIter)
    x0 = zeros(size(b));
    k = 0;
    r0 = b - A*x0;
    r = norm(r0, inf);
    while(r(end) > tol && k < maxIter)
        k = k+1;
        Ar0 = A*r0;
        a = (r0'*Ar0)/(Ar0'*Ar0);
        x = x0 + a*r0;
        x0 = x;
        r0 = b - A*x0;
        r = [r, norm(r0,inf)];
    end
end
```

The following script uses these three methods to solve the diffusion equation $-u_{xx} = 1$ on $x \in (0, 1)$. It also plots the residual against the number of iterations.

```
%% Problem 10
% Initial matrix
tol = 1e-10;
maxIter = 2e6;
m = 500;
h = 1/(m+1);
e = ones(m, 1);
A = (1/h^2)*spdiags([-e, 2*e, -e], -1:1, m,m);
[xGS, kGS, rGS] = GaussSeidel(A, e, tol, maxIter);
[xJ, kJ, rJ] = Jacobi(A, e, tol, maxIter);
[xCG, kCG, rCG] = ConjugateGradient(A, e, tol, maxIter);
figure;
semilogy(0:kGS, rGS, 'g', 0:kJ, rJ, 'b', 0:kCG, rCG, 'r');
xlabel('Number of Iterations');
ylabel('Infinity Norm of Residual');
```



We see in this plot that the Gauss-Seidel method converges much faster than either the Jacobi method or the Conjugate Gradient method. Looking at the slopes of these lines on the semilog plot, we can see that the Gauss-Seidel method converges quadractically while the Jacobi and Conjugate Gradient methods are converges linearly, this is visible because the slopes of these lines are -2 and -1 respectively.