

Class Notes 1: PRELIMINARIES

Math 639d

Due Date: Jan. 24

(updated: September 26, 2020)

Background: Linear algebra.

- (1) Review matrix multiplication. For example, compute AB , BC and ABC where

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & -1 & 0 \\ 1 & 4 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 4 & 4 \end{pmatrix} \quad C = \begin{pmatrix} 4 & 1 & 2 & 3 \\ 3 & 1 & -1 & 0 \end{pmatrix}.$$

- (2) Review matrix vector multiplication, e.g., compute Av and Cw where

$$v = \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix} \quad w = \begin{pmatrix} 11 \\ 4 \\ 3 \\ 2 \end{pmatrix}.$$

- (3) Review linear independence, e.g., are the vectors

$$\begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 2 \\ 1 \\ -1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 5 \\ 3 \\ 1 \end{pmatrix}$$

linearly independent?

- (4) Review span, basis and dimension. Do the above vectors span \mathbb{R}^4 , if not, then what is the dimension of the linear space which they span?
- (5) Suppose A is an $n \times n$ matrix. What is the range of A ? What is the kernel (or null space) of A ? Suppose A is the 4×4 matrix made by using the 4 vectors above as columns. What is the dimension of the range of A ? What is the dimension of the kernel of A ?
- (6) Suppose A is an $n \times n$ matrix. Given $b \in \mathbb{R}^n$, which of the following conditions imply that the linear system $Ax = b$ has a unique solution $x \in \mathbb{R}^n$?
- (a) The determinant of A is nonzero.
 - (b) The determinant of A is zero.
 - (c) The kernel of A is trivial, i.e., $\text{Ker}(A) = \{\mathbf{0}\}$ where $\mathbf{0}$ denotes the zero vector in \mathbb{R}^n .
 - (d) The dimension of the kernel of A is greater than zero.
 - (e) The range of A has dimension n .
 - (f) The dimension of the range of A is less than n .
 - (g) The rows of A are linearly independent.

- (h) The rows of A are linearly dependent.
- (i) The columns of A are linearly independent.
- (j) The columns of A are linearly dependent.
- (7) Review eigenvectors and eigenvalues: An eigenvector for an $n \times n$ matrix A is a nonzero vector $x \in \mathbb{R}^n$ satisfying

$$Ax = \lambda x$$

for some real or complex number λ . In this case, λ is the eigenvalue corresponding to x . Eigenvectors and eigenvalues satisfy the following:

- (a) Eigenvalues are roots of the characteristic equation $P(\lambda) = \det(A - \lambda I)$.
- (b) Eigenvectors with distinct eigenvalues are linearly independent.
- (c) The eigenvectors of A span a space of dimension d with $k \leq d \leq n$ where k is the number of distinct roots of $P(\lambda)$.
- (d) When $d = n$, we say that A is diagonalizable, i.e., there are n linearly independent eigenvectors whose span is \mathbb{R}^n .
- (e) When A is diagonalizable, we can form a matrix M with n independent eigenvectors of A as columns. In this case we get

$$D = M^{-1}AM$$

where D is a diagonal matrix whose diagonal entries are the corresponding eigenvalues.

Discussion: One of the main goals of this class is to study iterative solution of linear systems, i.e., to compute $x \in \mathbb{R}^n$ solving

$$(1.1) \quad Ax = b$$

where A is an $n \times n$ matrix and $b \in \mathbb{R}^n$ is given. This system has a unique solution if and only if any of the conditions (a), (c), (e), (g), (i) hold (they are all equivalent). In this case, we say that the matrix is non-singular. When any of the conditions (b), (d), (f), (h), or (j) of (6) hold then the system either does not have any solution or it has infinitely many of them. In this case, we say that the system is singular. We shall assume that A is such that we have a unique solution, i.e., A is non-singular.

Although we shall start our investigation of iterative methods with simple examples, it should be kept in mind that our goal is to solve systems with a huge number of unknowns which arise from large scale scientific computation. These problems, involving millions of unknowns, are well beyond what can be solved by direct methods (e.g., Gaussian Elimination) even on today's fastest computers.

An iterative method for computing the solution of (1.1) is a process which generates a sequence of iterates x_1, x_2, \dots in \mathbb{R}^n given an initial iterate $x_0 \in \mathbb{R}^n$. Here is a simple example:

$$(1.2) \quad x_{i+1} = x_i + (b - Ax_i).$$

Let $b = (1, 2)^t$, $x_0 = (0, 0)^t$ (for a row vector v^t denotes the corresponding column vector) and

$$(1.3) \quad A = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}.$$

Using MATLAB (or your favorite programming language) run the above iteration for, say, 20 steps. This involves setting up a simple loop, e.g., in MATLAB:

```
x=[0,0]';    % same as x=[0;0];
b=[1;2];
a=[3,0;0,1]
for i=1:20
x=x+(b-a*x)
end
```

The above illustrates a simple programming example in MATLAB. Note that the first and second line both create a column vector of dimension 2. In matrix definitions, the comma separates entries in a row while the semicolon separates rows. The “for” statement sets up a loop with counter “i” varying from 1 to 20. The statement after the for loop implements the iteration (1.2). Note that each time it is executed, the new iterate replaces (overwrites) the previous. This is a common practice in implementation of iterative methods: intermediate iterates are discarded in the iteration process.

What happens when you run the iteration. Is it converging? Do the same experiment with

$$(1.4) \quad A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

and

$$(1.5) \quad A = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}.$$

(Make sure that you reset your initial iterate before rewriting the “for” loop).

Some of you may have run the three examples by essentially typing the six lines above three times (with appropriate modification to the definition “a=...” line). To avoid such duplication, MATLAB provides the so-called “m-file” capability. To illustrate this, create a file in your MATLAB directory

$$b = Ax$$

containing the five lines above excluding the “a=...” line and call it “iter.m”
We can now run the three examples by:

```
a=[3,0;0,1]
iter
a=[2,1;1,2]
iter
a=[1,.5;.5,1]
iter
```

$$\begin{aligned} x - x_{i+1} &= x - x_i - (b - Ax_i) \\ &= x - x_i - Ax + Ax_i \\ &= e_i - Ae_i = (I - A)e_i \end{aligned}$$

In all but the last example, the iterations were DIVERGING. To investigate the convergence or divergence of an iterative method it is useful to derive a recurrence for the error. We first note that the solution x is a fixed point of the iteration, i.e.,

$$x_{i+1} = x_i + (b - Ax_i)$$

$$(1.6) \quad x = x + (b - Ax).$$

If the solution is a fixed point for the iteration then we say that the iteration is “**consistent**.” Subtracting the (1.6) and (1.2), we find that the error $e_i = x - x_i$ satisfies

$$(1.7) \quad e_{i+1} = (I - A)e_i.$$

Here I denotes the identity matrix (2×2 in this example). We see that the new error is related to the old by a simple matrix multiplication. Of course, multiplication by a matrix is a linear mapping. An iterative method is called “**linear**” if the new error results from applying a linear map (matrix multiplication) to the old error.

Note that the sequence $\{x_i\}$ converges to x if and only if the sequence of errors $\{e_i\}$ converges to the zero vector.

Let us consider the first example above. We have

$$e_{i+1} = \begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix} e_i$$

from which it follows that

$$e_i = \begin{pmatrix} (-2)^i & 0 \\ 0 & 0 \end{pmatrix} e_0$$

and is clearly divergent (unless the first component of the initial error is zero). This example illustrates that an iteration may converge for some initial errors and not others. In general, we shall be interested in methods which converge for any right hand side and any starting iterate!

Let us generalize this problem and consider A to be an $n \times n$ diagonal matrix with entries $A_{jj} = d_j$, $j = 1, \dots, n$. Applying the above analysis, we again have (1.7) and so the j 'th component of the error e_i is given by

$$(e_i)_j = (1 - d_j)^i (e_0)_j.$$

This method will converge for any initial error (and hence for any b and any choice of starting iterate x_0) if and only if

$$|1 - d_j| < 1, \quad \text{for } j = 1, \dots, n.$$

Of course, we would like to have methods which converge on a more general class of problems. We start by generalizing the iteration methods by introducing an iteration parameter τ and consider the iteration

$$(1.8) \quad x_{i+1} = x_i + \tau(b - Ax_i).$$

The above method is called **Richardson's method**. This iteration is consistent and a simple manipulation (do it!) gives

$$(1.9) \quad e_{i+1} = (I - \tau A)e_i$$

and so (1.8) is a linear iterative method. When A is diagonal, we have that the j 'th component of the error is given by

$$(e_i)_j = (1 - \tau d_j)^i (e_0)_j, \quad j = 1, \dots, n.$$

Thus, this method is convergent for any right hand side and starting iterate if and only if

$$(1.10) \quad \max_{j=1, \dots, n} |1 - \tau d_j| < 1.$$

Now suppose that for two numbers λ_0 and λ_1 ,

$$\lambda_0 \leq d_j \leq \lambda_1, \quad \text{for } j = 1, \dots, n$$

with $\lambda_0 > 0$. Then if we take $\tau = \lambda_1^{-1}$,

$$\max_{j=1, \dots, n} |1 - \tau d_j| = \max_{j=1, \dots, n} (1 - d_j/\lambda_1) \leq 1 - \frac{\lambda_0}{\lambda_1}.$$

For our first example (1.3), we have $\lambda_1 = 3$. Run this example (with $\tau = 1/3$) for 20 iterations and observe the convergence behavior.

Remark 1. This is not the best choice of iteration parameter. Actually, $\tau = 1/2$ is optimal. Can you see why?

We note that the derivation of (1.9) did not assume anything special about A and so could be used for non-diagonal A . Suppose that M is an invertible $n \times n$ matrix. Set $\tilde{e}_i = Me_i$ then from (1.9),

$$Me_{i+1} = M(I - \tau A)M^{-1}Me_i,$$

$$\begin{aligned} x - x_{i+1} &= x - x_i - \tau(b - Ax_i) \\ &= x - x_i - \tau(Ax - Ax_i) \\ &= e_i - \tau Ae_i \\ &= -(I - \tau A)e_i \end{aligned}$$

we want $\max_{j=1, \dots, n} |1 - \tau d_j|$ as small as possible.

Since we know d_j , $|1 - 3\tau|$ and $|1 - \tau|$

so plot $\max(|1 - 3\tau|, |1 - \tau|)$ as τ changes. so what's the smallest value?

i.e.,

$$(1.11) \quad \tilde{e}_{i+1} = (I - \tau \tilde{A})\tilde{e}_i \text{ where } \tilde{A} = MAM^{-1}.$$

Now, $\{e_i\}$ converges to the zero vector if and only if $\{\tilde{e}_i\}$ converges to the zero vector (why?).

The transformation MAM^{-1} is called a similarity transformation. We will be in great shape if we can find such a transformation which results in a diagonal matrix \tilde{A} since, in that case, we can simply apply our earlier analysis for the diagonal case. A matrix is called diagonalizable if there exists an invertible matrix M with MAM^{-1} diagonal. The matrix (1.4) is diagonalizable since

$$\tilde{A} = MAM^{-1} = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

We see from (1.11) that $\{\tilde{e}_i\}$ will converge to the zero vector for any initial error \tilde{e}_i if we choose $\tau = 1/3$. Note that the matrix M is used only for the analysis. Run the iteration (1.8) using A given by (1.4) and $\tau = 1/3$.

Remark 2. *Not all matrices are diagonalizable. There are, however, fairly general classes of matrices which are known to be diagonalizable. This will be the subject of discussion in a later class.*

Our method with parameter is, unfortunately, not general enough to deal with all problems. Consider the matrix given by

$$(1.12) \quad A = \begin{pmatrix} 3 & 0 \\ 0 & -1 \end{pmatrix}.$$

If you try the above method, you will find that it diverges (or at least fails to converge for some starting iterate) no matter what choice of τ . The analysis leading to (1.10) is still valid so for convergence we would need

$$|1 - 3\tau| < 1 \quad \text{and} \quad |1 + \tau| < 1.$$

This is impossible (even if we used complex τ). To deal with this situation, we need to change our strategy. We note that since A is non-singular so is A^t (the transpose of A). Thus, multiplying the equation $Ax = b$ by A^t does not change the solution set. We consider iterative solution of

$$(1.13) \quad A^t Ax = A^t b.$$

The above equation is called the “normal” equation corresponding to the system $Ax = b$. In our simple example, $A^t = A$ so we could have just multiplied by A . As we shall see in later classes, multiplication by A^t will work for more general matrices.

We can now apply our method (1.8) to (1.13). In this case, we use the matrix $A^t A$ and $A^t b$ as the right hand side. Now $A^t A$ has the positive diagonal entries $\{1, 9\}$ and we can choose $\tau = 1/9$. Run this iteration 20 times. Note that you get convergence but it is not as fast as the analogous iteration applied to the case when A was given by (1.3).

Problem 1. Consider A given by

$$(1.14) \quad A = \begin{pmatrix} 3 & -4 \\ -4 & 3 \end{pmatrix}.$$

Use (1.8) applied to the normal equations (1.13) to iteratively compute the solution to $Ax = b$ (b and x_0 as above). Experiment with different values of τ and try to determine when you get a convergent iteration.

Remark 3. We shall also be interested in solving systems involving matrices with complex coefficients. In this case, the transpose in the normal equations is replaced by the conjugate transpose, A^* where $(A^*)_{ij} = \bar{A}_{ji}$ and the bar denotes the complex conjugate.

In summary. Richardson's Method.

- Introduce iteration parameter τ to control the convergence
- In some special cases, we may transfer $Ax = b$ to

$$A^T A x = A^T b.$$

Class Notes 2: NORMS AND THE JACOBI METHOD

Math 639d

Due Date: Jan. 24

(updated: September 26, 2020)

We shall be primarily interested in solving matrix equations with real coefficients. However, the analysis of iterative methods sometimes requires the use of complex matrices for reasons which shall become clear in the subsequent classes.

We shall consider first vector norms on \mathbb{C}^n , the set of vectors with n components, each one a complex number. We start the Euclidean norm (standard vector length) defined by

$$\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2}$$

for $x = (x_1, x_2, \dots, x_n)^t \in \mathbb{C}^n$. This norm satisfies the usual norm axioms:

$$\begin{aligned} (2.1) \quad & \|x\|_2 \geq 0, \text{ for all } x \in \mathbb{C}^n, \text{ (non-negativity)} \\ & \|x\|_2 = 0, \text{ only if } x = \mathbf{0}, \text{ (definiteness)} \\ & \|\alpha x\|_2 = |\alpha| \|x\|_2, \text{ for } \alpha \in \mathbb{C}, x \in \mathbb{C}^n, \\ & \|x + y\|_2 \leq \|x\|_2 + \|y\|_2, \text{ for } x, y \in \mathbb{C}^n, \text{ (triangle inequality).} \end{aligned}$$

We shall sometimes refer to this norm as the ℓ^2 norm. This norm obviously can be restricted to real vectors in \mathbb{R}^n .

Norms give a notion of length to vectors. Consider the case of \mathbb{R}^2 . You can think of vectors in \mathbb{R}^2 as points (the vector being from the origin to the point). The set of vectors having Euclidean norm less than or equal to one consists of points which are within the circle (including the boundary) of radius one centered about the origin.

The conditions in (2.1) represent the set of axioms which a norm is required to satisfy. Many other norms are possible. For example, the “ ℓ^∞ ” norm is defined by

$$\|x\|_\infty = \max_{j=1, \dots, n} |x_j|.$$

It is not difficult to check that $\|\cdot\|_\infty$ satisfies (2.1). The set of vectors in \mathbb{R}^2 of length at most one using this norm is no longer a circle as was the case with the Euclidean norm. Find this set by tracing out its boundary; the set of vectors in \mathbb{R}^2 whose ℓ^∞ norm equals one.

Another useful norm is the “ ℓ^1 ” norm. It is given by

$$\|x\|_1 = \sum_{j=1}^n |x_j|.$$

Again, it is not hard to see that it satisfies all of the norm axioms (2.1). Sketch the set of vectors of \mathbb{R}^2 of length less than or equal to one when measured in the $\|\cdot\|_\infty$ norm.

These norms can be generalized into a whole scale of norms, i.e.,

$$(2.2) \quad \|x\|_p = \left(\sum_{j=1}^n |x_j|^p \right)^{1/p}.$$

These are norms for $1 \leq p < \infty$. When $0 < p < 1$, these expressions fail to satisfy the triangle inequality.

Problem 1. For $n = 2$ and $p = 1/2$, find two vectors $x, y \in \mathbb{R}^2$ satisfying

$$\|x + y\|_p > \|x\|_p + \|y\|_p. \quad x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

We shall use norms to more precisely measure convergence of iterative methods. Our discussion of an error vector e_i converging to the zero vector in Class 1 was somewhat intuitive. In contrast, we can be more rigorous if we show that for some norm $\|\cdot\|$ on \mathbb{R}^n , $\|e_i\| \rightarrow 0$ as $i \rightarrow \infty$.

Exercise 1. Let M be an $n \times n$ nonsingular matrix with complex coefficients and $\|\cdot\|$ be a norm on \mathbb{C}^n . Show that $\|\cdot\|_M$ defined by $\|x\|_M = \|Mx\|$ is also a norm on \mathbb{C}^n , i.e., show that this norm satisfies the axioms (2.1).

Let G be an $n \times n$ complex matrix. A vector norm $\|\cdot\|$ on \mathbb{C}^n induces a matrix norm on G by

$$(2.3) \quad \|G\| = \sup_{x \in \mathbb{C}^n, x \neq 0} \frac{\|Gx\|}{\|x\|}.$$

The set of $n \times n$ matrices is a vector space V (under matrix addition and the usual definition of scalar-matrix multiplication) and $\|G\|$ satisfies the norm axioms (2.1) on V . This is sometimes called the operator norm or the matrix norm induced by a vector norm and, in general, depends on the choice of vector norm.

If G is an $n \times n$ real matrix, we can define an alternative matrix norm by

$$(2.4) \quad \|G\|_r = \sup_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Gx\|}{\|x\|}.$$

It is clear that

$$\|G\|_r \leq \|G\|.$$

Most of our results below hold for this matrix norm as well and we shall often use the same notation $\|G\|$ for both. We shall sometimes refer to this as the “real” matrix norm.

Remark 1. Because the “operator norm” is a norm, we can make conclusions of the form

$$\|A + \epsilon B\| \leq \|A\| + |\epsilon| \|B\|.$$

Here A and B are $n \times n$ matrices and ϵ is a scalar. We used two of the norm axioms for the above inequality.

Remark 2. The spectral radius of a real or complex matrix A is defined by

$$\rho(A) = \max |\lambda_i|$$

where the maximum is over all eigenvalues λ_i of A . We have that $\|A\| \geq \rho(A)$ no matter what norm $\|\cdot\|$ we choose to use on \mathbb{C}^n . Indeed, if λ_i is an eigenvalue of A and x_i is a corresponding eigenvector, then $Ax_i = \lambda x_i$ so

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \geq \max_i \frac{\|Ax_i\|}{\|x_i\|} = \frac{\|\lambda_i x_i\|}{\|x_i\|} = \max_i |\lambda_i| = \rho(A)$$

Now, $\|A\|$ is the supremum of the quantity on the left hand side above over all non-zero vectors and so $\|A\| \geq \rho(A)$. *This is true for all norm.*

The opposite inequality does not hold in general. The matrix

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

has only the eigenvalue $\lambda = 1$ and so $\rho(A) = 1$. However, $\|A\|_\infty = 2$ as we shall see below (see, Proposition 1).

Remark 3. For an $n \times n$ real matrix, the real matrix norm given by (2.4) also satisfies

$$\|G\|_r \geq \rho(G).$$

This is no where near as obvious as in the complex case (Remark 2). We shall prove this in Class 4.

As well as satisfying the norm hypothesis, the operator norm satisfies two important additional inequalities. We state and prove these for the real operator norm and real matrices. The proofs for the complex case are essentially identical. Specifically, if A, B are $n \times n$ real matrices and $y \in \mathbb{R}^n$,

$$(2.5) \quad \|AB\| \leq \|A\| \|B\| \quad \text{and} \quad \|Ay\| \leq \|A\| \|y\|.$$

We shall prove the second inequality first. It is obviously true if $y = \mathbf{0}$ as both the left and right hand side are zero. If $y \neq \mathbf{0}$ then by the definiteness property of the norm, $\|y\| \neq 0$ and

$$\frac{\|Ay\|}{\|y\|} \leq \sup_{x \in \mathbb{R}^n, x \neq \mathbf{0}} \frac{\|Ax\|}{\|x\|} = \|A\|.$$

This is just the second inequality of (2.5) in disguise.

For the first inequality in (2.5), if $x \in \mathbb{R}^n$ with $x \neq \mathbf{0}$,

$$\frac{\|ABx\|}{\|x\|} = \frac{\|A(Bx)\|}{\|x\|} \leq \frac{\|A\| \|Bx\|}{\|x\|} \leq \frac{\|A\| \|B\| \|x\|}{\|x\|} = \|A\| \|B\|$$

where we used the second inequality of (2.5) twice above. The first inequality of (2.5) follows by taking the supremum over $x \in \mathbb{R}^n, x \neq \mathbf{0}$.

Now, consider a linear¹ iterative method for solving $Ax = b$ with A an $n \times n$ matrix. Let x_1, x_2, \dots be the sequence of iterates generated by the method and x_0 be the starting iterate. Since the method is linear, the errors $e_i = x - x_i$ are related by

$$e_{i+1} = Ge_i$$

for some $n \times n$ matrix G . **When the iteration is linear, the corresponding matrix G will be referred to as the reduction matrix associated with the iteration.** Let $\|\cdot\|$ be a vector norm and $\|G\|$ be the corresponding matrix norm. Applying (2.5) gives

$$\|e_{i+1}\| \leq \|G\| \|e_i\|.$$

Repetitively applying this gives

$$\|e_i\| \leq (\|G\|)^i \|e_0\|.$$

Now if $\gamma = \|G\|$ is less than one, then the $\|\cdot\|$ -norm of the error converges to zero as $i \rightarrow \infty$. Moreover, each step of the iteration reduces this norm of the error by at least a factor of γ . Thus, we have shown:

A linear iteration method with reduction matrix G converges for any starting iterate and any right hand side provided that there is a vector norm $\|\cdot\|$ which satisfies $\|G\| < 1$.

Thus, it is convenient (when possible) to characterize the matrix norm corresponding to a given vector norm. Such a characterization is illustrated in the following proposition.

¹Defined in Class Notes 1.

Proposition 1. *Let G be an $n \times n$ matrix. Then*

$$\|G\|_\infty = \max_{i=1}^n \left\{ \sum_{j=1}^n |G_{ij}| \right\}.$$

Proof. Set

$$\gamma = \max_{i=1}^n \left\{ \sum_{j=1}^n |G_{ij}| \right\}.$$

Let x be in \mathbb{C}^n with $x \neq 0$ and define $y = Gx$. Then,

$$\begin{aligned} |y_i| &= \left| \sum_{j=1}^n G_{ij} x_j \right| \leq \sum_{j=1}^n |G_{ij}| |x_j| \\ &\leq \sum_{j=1}^n |G_{ij}| \|x\|_\infty \leq \gamma \|x\|_\infty. \end{aligned}$$

Taking the maximum over $i = 1, \dots, n$ gives

$$\|Gx\|_\infty = \|y\|_\infty \leq \gamma \|x\|_\infty.$$

Dividing by $\|x\|_\infty$ and taking the supremum over all such x gives $\|G\|_\infty \leq \gamma$.

For the other direction, we first note that if $G = \mathbf{0}$, the identity is obvious. Otherwise, we let i be an index for which

$$\gamma = \sum_{j=1}^n |G_{ij}|$$

and set $x \in \mathbb{C}^n$ by

$$x_j = \begin{cases} 0 & : \text{ if } G_{ij} = 0 \\ \bar{G}_{ij}/|G_{ij}| & : \text{ otherwise.} \end{cases}$$

Here \bar{G}_{ij} denotes the complex conjugate of G_{ij} . Note that

$$\|x\|_\infty = 1 \quad \text{and} \quad \|Gx\|_\infty \geq |(Gx)_i| = \sum_{j=1}^n |G_{ij}| = \gamma.$$

Thus

$$\gamma \leq \frac{\|Gx\|_\infty}{\|x\|_\infty} \leq \|G\|_\infty.$$

This completes the proof of the proposition. \square

Example 1. (The Classical Jacobi Method). Let A be an $n \times n$ matrix with nonzero diagonal entries and D be the diagonal² matrix with entries $D_{jj} = A_{jj}$. The Jacobi Method for solving $Ax = b$ is given by

$$(2.6) \quad Dx_{i+1} = Dx_i + (b - Ax_i).$$

Note that this method is cheap to implement as it only requires simple linear operations on vectors and the inversion of a diagonal matrix times a vector.

Remark 4. The whole point of using iterative methods to solve matrix equations is to get a sufficiently accurate solution without doing an enormous amount of computation. Accordingly, each step of the iterative method should be relatively cheap (at least much cheaper than the amount of computational work required to invert the matrix using direct methods such as Gaussian Elimination). Thus, whenever we propose an iterative method, we will always consider the computational effort required per step. Of course, we shall also be interested in convergence/divergence and the rate of convergence.

Exercise 2. Show that the Jacobi Method is consistent and that the error for the Jacobi Method satisfies

$$(2.7) \quad De_{i+1} = De_i - Ae_i,$$

i.e.,

$$e_{i+1} = (I - D^{-1}A)e_i.$$

Definition 1. A matrix A is called (strictly) diagonally dominant if

$$|A_{jj}| > \sum_{i=1, i \neq j}^n |A_{ji}|$$

for $j = 1, \dots, n$.

Theorem 1. If A is (strictly) diagonally dominant then the Jacobi method converges for any right hand side and initial iterate.

Proof. A simple computation shows that the reduction matrix $G = (I - D^{-1}A)$ is given by

$$G_{ji} = \begin{cases} 0 : & \text{if } i = j, \\ -A_{j,i}/A_{j,j} : & \text{otherwise.} \end{cases}$$

Applying the previous proposition gives

$$\|G\|_\infty = \max_{j=1}^n \frac{\sum_{i=1, i \neq j}^n |A_{ji}|}{|A_{jj}|}$$

which is less than one when A is diagonally dominant. \square

²A matrix is diagonal if all of its off diagonal entries are zero, i.e., $A_{ij} = 0$ when $i \neq j$.

Remark 5. *The above proof illustrates how it may be more convenient to work with a specific norm to obtain a specific result. The ℓ^∞ norm fits very well with the diagonally dominant assumption.*

We shall be applying iterative methods to sparse matrices. A matrix is sparse if it contains relatively few non-zero elements. An example is the tridiagonal $n \times n$ matrix A_3 defined by

$$(A_3)_{ij} = \begin{cases} 2 & \text{if } i = j, \\ -1 & \text{if } |i - j| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

A full $n \times n$ matrix has n^2 non-zero entries. In contrast, A_3 only has $3n - 2$ nonzero entries.

Dealing with sparse matrices efficiently involves avoiding computations involving the zero entries. To do this, the matrix must be stored in a scheme which only involves the nonzero entries. We shall use a modified “Compressed Sparse Row” (CSR) structure (also called compressed row storage (CRS) or the Yale format). A nice discussion of compressed sparse row structure can be found at

<http://www.cs.utk.edu/~dongarra/etemplates/node373.html>.

This structure is designed so that it is easy to access the entries in a row. Our modification is made so that it is also easy to access the diagonal entry in any row.

The CRS structure involves three arrays, VAL, CIND and RIND. VAL is an array of real numbers and stores the actual (nonzero) entries of A . CIND is an integer array which contains the column indices for nonzero entries in A . The length of VAL and CIND are equal to the number of nonzero entries in A . Finally, RIND is an integer array of dimension $n + 1$ and contains the row offsets (into the arrays VAL and CIND). By convention, RIND($n+1$) is set to the total number of nonzeros plus one. This structure should become clear by examining the following example (see also the URL above). Consider

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1/3 & 3 & -2/3 & 0 \\ 0 & -1/4 & 4 & -3/4 \\ 0 & 0 & -1/5 & 5 \end{pmatrix}.$$

The modified CRS structure is as follows:

VAL	2	-1	3	-1/3	-2/3	4	-1/4	-3/4	5
	-1/5								
CIND	1	2	2	1	3	3	2	4	4
	3								

and

RIND 1 3 6 9 11.

Note that the i 'th entry of RIND points to the start of the nonzero values (in VAL) for the i 'th row. It also points to the start of the column indices for that row. The modification is that we always put the diagonal entry at that location, i.e. $\text{VAL}(\text{RIND}(i)) = A_{i,i}$. The general CRS structure does not do this. In fact, the general CRS storage scheme does not even have a diagonal entry whenever the diagonal entry is zero.

Class Notes 3: GAUSS-SEIDEL

Math 639d

Due Date: Jan. 31

(updated: September 29, 2020)

The Jacobi method in the last class was the first example of the definition of an iterative method by splitting. Specifically, we started with $Ax = b$ and split $A = D + (A - D)$. Moving the two terms on different sides of the equation, we obtain

$$Dx = -(A - D)x + b.$$

The resulting iterative method involved replacing x on the left by x_{i+1} while replacing x on the right by x_i . The method is automatically consistent (why?).

The Gauss-Seidel method is developed in a similar fashion but uses the splitting $A = (L + D) + U$ resulting in the (consistent) iterative method

$$(3.1) \quad (D + L)x_{i+1} = -Ux_i + b.$$

Here D is the diagonal matrix with entries $D_{i,i} = A_{i,i}$, $i = 1, \dots, n$, L is strictly lower triangular (i.e., $L_{i,j} = 0$ when $j \geq i$) and U is strictly upper triangular (i.e., $U_{i,j} = 0$ when $i \geq j$).

Let $e_i = x - x_i$ then a simple computation (do it!) shows that

$$e_{i+1} = Ge_i \quad (D+L)(x-x_{i+1}) = (D+L)x + Ux_i - b$$

with

$$(3.2) \quad G = -(D + L)^{-1}U. \quad = (D+L)x + Ux_i - (D+L)x - Ux$$

Thus, the Gauss-Seidel method is a linear iterative method.

As in the Jacobi method, for this method to make sense, we need to assume that $A_{i,i} \neq 0$ for $i = 1, \dots, n$. Indeed, each step in the Gauss-Seidel method requires inversion of the lower triangular matrix $(D + L)$. As

$$\det(D + L) = \prod_{i=1}^n D_{i,i} \quad \Rightarrow \quad e_{i+1} = -(D+L)^{-1}Ue_i$$

(why?), $D + L$ is nonsingular if and only if D has nonzero diagonal entries.

The Gauss-Seidel method can be implemented as a “sweep.” To illustrate this we consider the following pseudo-code:

FUNCTION $gs(X, B, A, n)$

FOR $j = 1, 2, \dots, n$ *DO* {

$$X_j = (B_j - \sum_{\{k \neq j, A_{j,k} \neq 0\}} A_{j,k} X_k) A_{j,j}^{-1} \}$$

END

RETURN

The arguments justifying the above routine are as follows. X and B are n -dimensional vectors. On input, X contains the vector x_i . On return, X contains x_{i+1} (x_i is overwritten and lost). B contains the right hand vector b . Often the matrix A is sparse (contains relatively few nonzero entries per row). Note that the sum above only includes terms when $A_{j,k}$ differs from zero. It is important that the entries when $A_{j,k} = 0$ are not included. To do this, one needs to use a sparse storage structure for A . The judicious use of sparse matrix structure such as CSR can result in significant savings in both execution time and memory.

Theorem 1. *If A is diagonally dominant, then the Gauss-Seidel method converges for any starting iterate and any right hand side.*

Proof. Set

$$\gamma = \max_{j=1}^n \frac{\sum_{k>j} |A_{j,k}|}{|A_{j,j}| - \sum_{k<j} |A_{j,k}|}.$$

Now, by diagonal dominance, for each j ,

$$|A_{j,j}| - \sum_{k<j} |A_{j,k}| > \sum_{k>j} |A_{j,k}|$$

so γ is less than one. We will show that

$$(3.3) \quad \|G\|_{\infty} \leq \gamma$$

where G is the reduction matrix associated with the Gauss-Seidel method.

Suppose that $x \in \mathbb{R}^n$ with $x \neq 0$. Let $y = Gx$ (or $(D + L)y = -Ux$). For any j ,

$$(3.4) \quad A_{j,j}y_j + \sum_{k<j} A_{j,k}y_k = - \sum_{k>j} A_{j,k}x_k$$

Let j be such that $\|y\|_{\infty} = |y_j|$. Then absolute value of the left hand side of (3.4) is bounded from below by

$$\begin{aligned} |A_{j,j}y_j + \dots| &\geq |A_{j,j}y_j| - \sum_{k<j} |A_{j,k}y_k| \geq |A_{j,j}||y_j| - \sum_{k<j} |A_{j,k}| |y_k| \\ &\quad \uparrow \\ &\text{triangle inequality} \\ &\geq |A_{j,j}||y_j| - \sum_{k<j} |A_{j,k}| |y_j| \\ &= (|A_{j,j}| - \sum_{k<j} |A_{j,k}|) \|y\|_{\infty}. \end{aligned}$$

Analogously, the absolute value of the right hand side of (3.4) is bounded from above by

$$\sum_{k>j} |A_{j,k}| |x_k| \leq \sum_{k>j} |A_{j,k}| \|x\|_\infty.$$

Combining the above gives,

$$(|A_{j,j}| - \sum_{k<j} |A_{j,k}|) \|y\|_\infty \leq \sum_{k>j} |A_{j,k}| \|x\|_\infty$$

from which (3.3) immediately follows. \square

Our goal will be to formulate a somewhat more general theorem concerning the convergence of iterative methods. To this end, we consider the following definitions.

Definition 1. *The spectrum of an $n \times n$ matrix A (denoted by $\sigma(A)$) is the set of eigenvalues of A , i.e.,*

$$\sigma(A) = \{\lambda : \lambda \text{ an eigenvalue of } A\}$$

Definition 2. *The spectral radius $\rho(A)$ is defined by*

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|.$$

Our goal will be to prove the following theorem:

Theorem 2. *Let A be an $n \times n$ matrix with complex entries. Given $\epsilon > 0$ there exists a norm $\|\cdot\|_*$ with*

$$\|A\|_* < \rho(A) + \epsilon.$$

In some sense, this theorem reduces the question of convergence of linear iterative methods to the computation of the spectral radius of the error reduction matrix. This is stated more precisely in the following corollary.

Corollary 1. *Let G be the reduction matrix for a linear iterative method. Then the iterative method converges for any (complex) starting iterate and any (complex) right hand side if and only if*

$$\rho(G) < 1.$$

Remark 1. *The proof (below) of the corollary involves using the (proof of the) theorem to construct a norm $\|\cdot\|_*$ for which*

$$\|G\|_* < \gamma = \rho(G) + (1 - \rho(G))/2 < 1.$$

It then follows that

$$(3.5) \quad \|e_j\|_* \leq \gamma^j \|e_0\|_*.$$

Any interesting application for iterative methods involves very large matrices. If the matrix is sparse, then the number of nonzeros might be $O(n)$. (Here $O(n)$ denotes a number which is bounded by a constant times n , for example A may have at most 5 nonzero entries per row in which case the constant would be 5). This means that each matrix evaluation involves computer work time proportional to n . We would ideally like to design iterative methods which converge and have asymptotic work as close to $O(n)$ as possible. This is not always possible but gives us something to strive for.

Example 1. Consider applying Gauss-Siedel to the matrix

$$A_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -1 & \text{if } j = i + 1, \\ 0 & \text{otherwise.} \end{cases}$$

The error reduction matrix and is given by

$$G = -(D + L)^{-1}U = -D^{-1}U = -U.$$

The matrix G has only the eigenvalue 0 (why?) and so $\rho(G) = 0$. The above corollary shows that Gauss-Seidel converges (for this example) for any starting iterate and any right hand side.

Consider iterating for the solution of $Ax = \mathbf{0}$ with initial iterate given by

$$x_0 = (0, 0, 0, \dots, 0, -1)^t.$$

We clearly have

$$e_0 = (0, 0, 0, \dots, 0, 1)^t$$

and a simple computation gives

$$e_i = (0, 0, \dots, 0, 1, 0, \dots, 0)^t$$

where the 1 appears in the $(n-i)$ 'th entry (when $0 \leq i < n$). Finally, $e_n = \mathbf{0}$. If we measure the error in the ℓ^∞ norm, then

$$\|e_0\|_\infty = \|e_1\|_\infty = \dots = \|e_{n-1}\|_\infty = 1.$$

Even though the corollary guarantees eventual convergence, we do not obtain any convergence in ℓ^∞ norm until the n 'th step. The work required to solve this problem using Gauss-Seidel is $O(n^2)$ since each iteration requires $O(n)$ operations.

This is a pedagogical example. Note that $\|G\|_\infty = 1$ so

$$\|e_i\|_\infty \leq \|G\|_\infty^i \|e_0\|_\infty$$

is of little use. On the other hand, the norm $\|e_i\|_*$ drastically scales the cartesian coordinates of e_i enabling (3.5) with any $0 < \gamma < 1$.

The above example illustrates that the corollary and remark may not tell the whole story. The trouble in this example stems from the fact that G has a large Jordan Block (an eigenvalue of high multiplicity with a low dimensional eigenspace). Jordan Blocks and their relation to the proof of the above theorem will be discussed in the next reading assignment.

We will prove the above theorem in the next class but here we prove the corollary assuming that the theorem has already been verified.

Proof of the Corollary. Suppose that $\rho(G) \geq 1$. Then there is an eigenvalue λ with $|\lambda| \geq 1$. Let ϕ be the corresponding eigenvector (notice that ϕ may have to be complex even if A has real valued entries). Consider solving a problem with initial error $e_0 = \phi$. Then

$$e_i = G^i \phi = \lambda^i \phi$$

does not converge to zero.

If, on the other hand, $\rho(G) < 1$ set $\epsilon = (1 - \rho(G))/2$. Then, by the theorem, there is a norm $\|\cdot\|_*$ with

$$\|G\|_* < \rho(G) + \epsilon = \frac{1 + \rho(G)}{2} < 1.$$

It follows that

$$\|e_i\|_* \leq \left(\frac{1 + \rho(G)}{2} \right)^i \|e_0\|_*$$

and so $\|e_i\|_*$ converges to zero as i goes to infinity. □

Class Notes 4: THE SPECTRAL RADIUS, NORM CONVERGENCE AND SOR.

Math 639d

Due Date: Feb. 7

(updated: October 7, 2020)

In the first part of this week's reading, we will prove Theorem 2 of the previous class. We will use the Jordan Decomposition Theorem. (Those of you who took my 640 class last semester saw an alternative proof using Schur's Theorem). To this end, we make the following definition:

Definition 1. A Jordan Block is a square matrix of the form

$$(4.1) \quad J_{ij} = \begin{cases} \lambda & \text{if } i = j, \\ 1 & \text{if } j = i + 1, \\ 0 & \text{otherwise.} \end{cases}$$

The following theorem can be found in any reasonably good text on linear algebra. For more details, see also,

<http://mathworld.wolfram.com/JordanCanonicalForm.html>.

Theorem 1. (Jordan Canonical Form Theorem) Any square (complex) matrix A is similar to a block diagonal matrix with Jordan Blocks on the "block" diagonal, i.e., $A = NJN^{-1}$ with N an $n \times n$ complex valued non-singular matrix and J block diagonal with Jordan blocks on the diagonal.

Here is an example of a block diagonal matrix with Jordan Blocks on the diagonal:

$$\begin{pmatrix} J_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & J_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & J_3 \end{pmatrix}.$$

Here J_i are Jordan Blocks of dimension $k_i \times k_i$ with λ_i on the diagonal. The remaining zero blocks get their sizes from the diagonal blocks, e.g., the 1, 3 block has dimension $k_1 \times k_3$.

Remark 1. The above theorem states that given a square complex matrix A , we can write $J = N^{-1}AN$ where J is a block diagonal matrix with Jordan Blocks on the diagonal. The similarity matrix N is, in general, complex even when A has all real entries. Since similarity transformations preserve eigenvalues and the eigenvalues of an upper triangular matrix (a matrix with zeroes below the diagonal) are the values on the diagonal, the values of λ appearing in the Jordan Blocks are the eigenvalues of the matrix A . As the eigenvalues of matrices with real coefficients are often complex, the similarity matrices even for matrices with real coefficients end up being complex as well.

We shall use Proposition 1 of Class Notes 2 which we restate here as a reminder.

Proposition 1. *Let B be an $m \times m$ matrix with possibly complex entries then*

$$\|B\|_\infty = \max_{j=1}^m \left(\sum_{k=1}^m |B_{jk}| \right)$$

Proof of Theorem 2 of Class Notes 3. We start by applying the Jordan Canonical Form Theorem and conclude that $J = N^{-1}AN$ with J being a block diagonal matrix with K Jordan Blocks on the diagonal. The l 'th diagonal block of J has the following form:

$$J_{ij}^l = \begin{cases} \lambda_l & \text{if } i = j, \\ 1 & \text{if } j = i + 1, \\ 0 & \text{otherwise.} \end{cases}$$

As $\{\lambda_l\}$ for $l = 1, \dots, K$ are the eigenvalues of A ,

$$\rho(A) = \max_{l=1}^K |\lambda_l|.$$

Let $\epsilon > 0$ be given and set M to be the diagonal matrix with entries $M_{ii} = \epsilon^i$. Further, set

$$\|x\|_* = \|M^{-1}N^{-1}x\|_\infty.$$

That this is a norm follows from Problem 2 of Class 2. Now

$$\begin{aligned} \|A\|_* &= \sup_{x \in \mathbb{C}^n, x \neq 0} \frac{\|Ax\|_*}{\|x\|_*} \\ &= \sup_{x \in \mathbb{C}^n, x \neq 0} \frac{\|M^{-1}N^{-1}Ax\|_\infty}{\|M^{-1}N^{-1}x\|_\infty} \\ &= \sup_{y \in \mathbb{C}^n, y \neq 0} \frac{\|M^{-1}N^{-1}ANMy\|_\infty}{\|y\|_\infty} = \|M^{-1}JM\|_\infty. \end{aligned}$$

The second to last equality above followed from substituting $y = M^{-1}N^{-1}x$ and noting that as x goes over all nonzero vectors, so does y . A direct computation gives

$$(M^{-1}JM)_{ij} = \begin{cases} \lambda_l & \text{if } i = j, \\ 0 \text{ or } \epsilon & \text{if } j = i + 1, \\ 0 & \text{otherwise.} \end{cases}$$

Applying the proposition gives,

$$\|A\|_* = \|M^{-1}JM\|_\infty \leq \max_{i=1}^K |\lambda_i| + \epsilon = \rho(A) + \epsilon.$$

This completes the proof. \square

The Successive Over Relaxation Method.

In the remainder of this class, we consider the “Successive Over Relaxation Method” (SOR). This is another example of a splitting method. In this case we split $A = (\omega^{-1}D + L) + ((1 - \omega^{-1})D + U)$, i.e.,

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

and obtain the iterative method

$$(4.2) \quad (\omega^{-1}D + L)x_{i+1} = ((\omega^{-1} - 1)D - U)x_i + b.$$

As in the Gauss-Seidel and Jacobi iterations, we can only apply this method to matrices with non-vanishing diagonal entries.

In the above method, ω is an iteration parameter which we shall have to choose. When $\omega = 1$, the method reduces to Gauss-Seidel. Like Gauss-Seidel, this method can be implemented as a sweep.

We first develop a necessary condition on ω required for convergence of the SOR method. A simple computation shows that the reduction matrix for the SOR method is

$$G_{SOR} = (\omega^{-1}D + L)^{-1}((\omega^{-1} - 1)D - U).$$

From the corollary of the previous class, a necessary condition for SOR to converge for all starting iterates and right hand sides is that $\rho(G_{SOR}) < 1$. The eigenvalues of G_{SOR} are the roots of the characteristic polynomial,

$$P(\lambda) = \det(G_{SOR} - \lambda I) = (-1)^n \lambda^n + C_{n-1} \lambda^{n-1} + \cdots + C_0.$$

Note that $C_0 = P(0) = \det(G_{SOR})$ and that

$$P(\lambda) = (-1)^n \prod_{i=1}^n (\lambda - \lambda_i)$$

where $\lambda_1, \lambda_2, \dots, \lambda_n$ are the roots of P . Expanding the above expression shows that

$$C_0 = \prod_{i=1}^n \lambda_i = \det(G_{SOR}).$$

Note that if $|C_0| \geq 1$, there has to be at least one root whose absolute value is at least one, i.e., $\rho(G_{SOR}) \geq 1$. Consequently, a necessary condition for $\rho(G_{SOR}) < 1$ and the convergence of the SOR iteration is that

$$|C_0| = |\det(G_{SOR})| < 1.$$

Now, the determinant of an upper or lower triangular matrix is the product of the entries on the diagonal. Using this and other simple properties of the determinant gives

$$\det(G_{SOR}) = \frac{\det((\omega^{-1} - 1)D - U)}{\det(\omega^{-1}D + L)} = \frac{\prod_{i=1}^n (\omega^{-1} - 1)D_{i,i}}{\prod_{i=1}^n \omega^{-1}D_{i,i}} = (1 - \omega)^n.$$

For this product to be less than one in absolute value, it is necessary that $|1 - \omega| < 1$, i.e., $0 < \omega < 2$. We restate this as a proposition.

Proposition 2. *A necessary condition for the SOR method to converge for any starting vector and right hand side is that $0 < \omega < 2$.*

We shall provide a convergence theorem for the SOR method. Note that the reduction matrix G_{SOR} is generally not symmetric even when A is symmetric. Accordingly, even if A has real entries, any analysis of G_{SOR} will have to involve complex numbers as, in general, G_{SOR} will have complex eigenvalues. Accordingly, we shall provide an analysis for complex Hermitian matrices A .

Definition 2. *The conjugate transpose N^* of a general $n \times m$ matrix N with complex entries is the $m \times n$ matrix with entries*

$$(N^*)_{i,j} = \bar{N}_{j,i}.$$

Here the bar denotes complex conjugate. An $n \times n$ matrix A with complex entries is called Hermitian (or conjugate symmetric) if $A^ = A$.*

We shall use the Hermitian inner product (\cdot, \cdot) on $\mathbb{C}^n \times \mathbb{C}^n$ defined by

$$(x, y) = \sum_{i=1}^n x_i \bar{y}_i.$$

We shall discuss more general Hermitian inner products in later classes. We note that

$$(x, y) = \overline{(y, x)}, \quad (\alpha x, y) = \alpha(x, y), \quad \text{and} \quad (x, \alpha y) = \bar{\alpha}(x, y)$$

for all $x, y \in \mathbb{C}^n$ and $\alpha \in \mathbb{C}$. In addition, it is easy to see that when N is an $n \times m$ matrix, N^* is the unique matrix satisfying (check it by applying it to the standard basis vectors!)

$$(Nx, y) = (x, N^*y) \quad \text{for all } x \in \mathbb{C}^m, y \in \mathbb{C}^n.$$

We also note the following properties of a Hermitian $n \times n$ matrix A .

- (Ax, x) is real since

$$(Ax, x) = (x, Ax) = \overline{(Ax, x)}.$$

- If A is positive definite¹ then $A_{i,i} > 0$ since $A_{i,i} = (A\mathbf{e}_i, \mathbf{e}_i) > 0$ where \mathbf{e}_i denotes the i 'th standard basis vector for \mathbb{C}^n .

We finish this class by stating a convergence theorem for SOR. Its proof will be given in the next class.

Theorem 2. *Let A be an $n \times n$ Hermitian positive definite matrix and ω be in $(0, 2)$. Then the SOR method for iteratively solving $Ax = b$ converges for any starting vector and right hand side.*

¹A complex $n \times n$ matrix A is positive definite if $(Ax, x) > 0$ for all non zero $x \in \mathbb{C}^n$.

Class Notes 5: MORE ON THE SUCCESSIVE OVER RELAXATION METHOD

Math 639d

Due Date: Feb. 14

(updated: October 7, 2020)

The first task of this class is to prove the convergence theorem for successive over relaxation (SOR) introduced last week. This proof is not very intuitive and was probably discovered by playing with the equations until something worked.

Theorem 1. *Let A be an $n \times n$ Hermitian positive definite matrix and ω be in $(0, 2)$. Then the SOR method for iteratively solving $Ax = b$ converges for any starting vector and right hand side.*

Proof. The reduction matrix for SOR is $G_{SOR} = (\omega^{-1}D + L)^{-1}((\omega^{-1} - 1)D - U)$ and we set $Q = (\omega^{-1}D + L)$. We shall show that $\rho(G_{SOR}) < 1$.

We note that

$$(5.1) \quad (I - G_{SOR}) = (\omega^{-1}D + L)^{-1}((\omega^{-1}D + L) - ((\omega^{-1} - 1)D - U)) = Q^{-1}A.$$

Let $x \in \mathbb{C}^n$ be an eigenvector of G_{SOR} with eigenvalue λ and set $y = (I - G_{SOR})x = (1 - \lambda)x$. Then, by (5.1),

$$(5.2) \quad Qy = Ax$$

and

$$(5.3) \quad \begin{aligned} (Q - A)y &= (Q - A)Q^{-1}Ax = (A - AQ^{-1}A)x \\ &= A(I - Q^{-1}A)x = AG_{SOR}x = \lambda Ax. \end{aligned}$$

The next to the last equality above followed from (5.1).

Let $(x, y) = x \cdot \bar{y}$ for $x, y \in \mathbb{C}^n$ denote the Hermitian inner product on \mathbb{C}^n . Taking the inner product of (5.2) with y in the second place and (5.3) with y in the first place gives

$$\begin{aligned} (Qy, y) &= (Ax, y), \quad \text{and} \\ (y, (Q - A)y) &= (y, \lambda Ax). \end{aligned}$$

We note that $d_{ii} = (A\mathbf{e}_i, \mathbf{e}_i)$ where \mathbf{e}_i is the i 'th standard basis vector. Since A is Hermitian positive definite, it follows that the diagonal matrix D has real positive diagonal entries and hence is also Hermitian positive definite.

The above equations can thus be rewritten

$$\begin{aligned} \omega^{-1}(Dy, y) + (Ly, y) &= (1 - \bar{\lambda})(Ax, x), \quad \text{and} \\ (\omega^{-1} - 1)(Dy, y) - (y, Uy) &= (1 - \lambda)\bar{\lambda}(x, Ax). \end{aligned}$$

Now since A is Hermitian, $L^* = U$ so $(Ly, y) = (y, Uy)$. Adding the two equations above gives

$$(5.4) \quad (2\omega^{-1} - 1)(Dy, y) = (1 - |\lambda|^2)(Ax, x).$$

Now, $x \neq \mathbf{0}$ implies $(Ax, x) > 0$ so $Ax \neq \mathbf{0}$. As Q is nonsingular, (5.2) implies $y \neq \mathbf{0}$. In addition, the assumption on ω implies that $2\omega^{-1} - 1 > 0$ so that the left hand side of (5.4) is positive (from above we know that D is diagonal with positive numbers on the diagonal). For the right hand side of (5.4) to be positive, it is necessary that $|\lambda| < 1$. As this holds for every eigenvalue of G , $\rho(G_{SOR}) < 1$. \square

Remark 1. *The above proof works for other splittings of A , e.g. $A = D + L + U$ where D is Hermitian positive definite and $L^* = U$ (D , L and U need not be diagonal, strictly lower triangular, and strictly upper triangular, respectively).*

Remark 2. *Unlike the analysis for Gauss-Seidel and Jacobi given earlier, the above proof for SOR does not lead to any explicit bound for the spectral radius. All it shows is that the spectral radius has to be less than one.*

One step of an iterative method for solving $Ax = b$ can be used to provide an “approximate” inverse for A . Specifically, we define $B : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by defining By to be the result of one iterative step applied to solving $Ax = y$ with initial iterate $x_0 = \mathbf{0}$. All of the iterative methods that we have considered up to now have been of the form

$$Qx_1 = (Q - A)x_0 + b \text{ and hence } x_1 = Q^{-1}b \approx A^{-1}b,$$

i.e., the approximation to A^{-1} is $B = Q^{-1}$. With this notation, the reduction matrix G is given by $G = I - Q^{-1}A = I - BA$.

When A is symmetric positive definite (SPD) real (or Hermitian positive definite complex) it is advantageous to have approximate inverses which are of the same type. We shall consider the case when A is SPD real. The approximate inverse corresponding to the Jacobi method is $B = D^{-1}$ and is SPD when A is SPD. The approximate inverse in the case of Gauss-Seidel is $B = (D + L)^{-1}$ and is generally not symmetric. We can develop a symmetric approximate inverse from Gauss-Seidel by introducing the “transpose” iteration,

$$(D + U)x_{i+1} = -Lx_i + b.$$

Note that the implementation of this method as a sweep is similar to the original Gauss-Seidel method except that one goes through the vector in reverse order. The pseudo-code is as follows:

FUNCTION $gst(X, B, A, n)$

```

FOR  $j = n, n-1, \dots, 1$  DO {
     $X_j = (B_j - \sum_{k \neq j, A_{j,k} \neq 0} A_{j,k} X_k) A_{j,j}^{-1}$ .}
RETURN
END

```

We get a symmetric approximate inverse by defining By by first applying one step of Gauss-Seidel with zero initial iterate and right hand side y and using the result x_1 as an initial iterate for one step of the transpose iteration, i.e., $By = x_2$ where

$$(5.5) \quad (D + L)x_1 = y, \quad \text{and} \quad (D + U)x_2 = -Lx_1 + y.$$

We can explicitly compute $B = Q^{-1}$ by computing the reduction matrix $G = I - Q^{-1}A$ for the two step procedure and identifying Q^{-1} . Let x be the solution of $Ax = y$ and, as usual, set $e_i = x - x_i$ with $x_0 = \mathbf{0}$. The errors e_1 and e_0 are related by $e_1 = -(D + L)^{-1}Ue_0 = (I - (D + L)^{-1}A)e_0$ (the usual Gauss-Seidel reduction matrix) while the errors e_2 and e_1 are related by $e_2 = -(D + U)^{-1}Le_1 = (I - (D + U)^{-1}A)e_1$. Thus the error reduction matrix for the two steps is given by

$$\begin{aligned}
 G &= (I - (D + U)^{-1}A)(I - (D + L)^{-1}A) \\
 &= I - [(D + U)^{-1} + (D + L)^{-1} - (D + U)^{-1}A(D + L)^{-1}]A \\
 &= I - (D + U)^{-1}[(D + L) + (D + U) - A](D + L)^{-1}A \\
 &= I - (D + U)^{-1}D(D + L)^{-1}A.
 \end{aligned}$$

Thus, the approximate inverse associated with the two steps is given by

$$(5.6) \quad B = (D + U)^{-1}D(D + L)^{-1}$$

and is obviously SPD when A is SPD (why?).

Exercise 1. *The symmetric successive over relaxation method (SSOR) is defined in an analogous fashion, i.e., by taking one step of SOR with zero initial iterate followed by onestep of the transpose iteration*

$$(\omega^{-1}D + U)x_{i+1} = ((\omega^{-1} - 1)D - L)x_i + b.$$

Compute the approximate inverse associated with SSOR. Your result should reduce to (5.6) when $\omega = 1$.

Optimization of SOR The whole point of introducing a parameter into the SOR iteration is so that by judicious choice, one can get a significantly faster iteration. To illustrate that it is indeed possible to do this, we consider a special class of matrices, specifically, those satisfying the so-called “Property A” condition. The definition of Property A will be given shortly but

first we shall simplify the convergence study to the case when the diagonal of A coincides with that of the identity.

Recall the SOR iteration:

$$(5.7) \quad (\omega^{-1}D + L)x_{i+1} = ((\omega^{-1} - 1)D - U)x_i + b$$

where $A = D + L + U$. Assume that $D_{ii} > 0$ and set $D^{1/2}$ to be the diagonal matrix with entries $D_{ii}^{1/2}$. We set

$$\hat{x}_i = D^{1/2}x_i.$$

Then substituting this into (5.7) and multiplying by $D^{-1/2}$ gives

$$(5.8) \quad (\omega^{-1}I + \hat{L})\hat{x}_{i+1} = ((\omega^{-1} - 1)I - \hat{U})\hat{x}_i + \hat{b}$$

where

$$\begin{aligned} \hat{L} &= D^{-1/2}LD^{-1/2}, & \hat{U} &= D^{-1/2}UD^{-1/2} \\ \hat{b} &= D^{-1/2}b, & \hat{A} &= D^{-1/2}AD^{-1/2} = I + \hat{L} + \hat{U}. \end{aligned}$$

Note that \hat{L} and \hat{U} are also lower and upper triangular. Now $\hat{x} = D^{1/2}x$ is the solution of $\hat{A}\hat{x} = \hat{b}$ and (5.8) is the corresponding SOR method. Set $e_i = x - x_i$ and $\hat{e}_i = \hat{x} - \hat{x}_i = D^{1/2}e_i$. Then e_i converges to zero if and only if \hat{e}_i converges to zero and their norms are related in the obvious way. In this way, we can reduce the study of SOR to the case when $\hat{A} = I + \hat{L} + \hat{U}$. This is an example of rescaling A to obtain some desired property. Thus, at least in the case when $D_{ii} > 0$, it suffices to analyze the simpler case when $D = I$.

Definition 1. (Property A) Suppose that A is an $n \times n$ matrix with 1's on the diagonal. Writing $A = I + L + U$ with L and U strictly lower and upper triangular, we say that A satisfies Property A if all of the eigenvalues of

$$J_z = \frac{1}{z}L + zU$$

are independent of z ($z \neq 0$). Note that we allow z to be complex.

Example 1. Any tridiagonal matrix satisfies Property A. Indeed, let A be a tridiagonal matrix of dimension $n \times n$. To check Property A, we show that J_z , for any nonzero z , is similar to J_1 . This suffices since similar matrices share the same spectrum. To do this, we introduce the diagonal similarity transformation matrix M_z which has $(M_z)_{ii} = z^i$, $i = 1, \dots, n$, as diagonal entries. A simple computation gives

$$(M_z^{-1}J_1M_z)_{i,j} = z^{j-i}(J_1)_{i,j}$$

from which it immediately follows that for tridiagonal A ,

$$M_z^{-1}J_1M_z = J_z$$

i.e., J_z is similar to J_1 .

Example 2. In this example, we consider a finite difference matrix obtained from lexicographical ordering of the unknowns in a finite difference approximation to a boundary value problem associated with a partial differential equation. This is explained below.

Let Ω be an open connected bounded subset of \mathbb{R}^2 (we refer to Ω as our domain). We want to approximate the function U defined on Ω satisfying

$$(5.9) \quad \begin{aligned} -\Delta U(x) &= f(x) \quad \text{for all } x = (x_1, x_2) \in \Omega, \\ U(x) &= 0 \quad \text{for } x \in \partial\Omega. \end{aligned}$$

Here $\partial\Omega$ denotes the boundary of Ω and

$$\Delta u := \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}$$

denotes the Laplacian. We think of covering \mathbb{R}^2 with a uniform grid consisting of lines parallel to the x -axis (at $y = jh$, j an integer) and lines parallel to the y -axis (at $x = ih$, i an integer). Here h is a (small) positive number and determines the accuracy of the approximation. The nodes of the grid are the points $x_{i,j} = (ih, jh)$ and are where the lines intersect. We seek a nodal function $u_{i,j}$ where $u_{i,j}$ approximates $U(x_{i,j})$. We get equations for $\{u_{i,j}\}$ by replacing the derivatives on the left hand side of (5.9) by finite differences. Specifically,

$$-\frac{\partial^2 u(x_{i,j})}{\partial x_1^2} \approx \frac{2u_{i,j} - u_{i-1,j} - u_{i+1,j}}{h^2}$$

and

$$-\frac{\partial^2 u(x_{i,j})}{\partial x_2^2} \approx \frac{2u_{i,j} - u_{i,j-1} - u_{i,j+1}}{h^2}$$

which gives

$$(5.10) \quad 4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = h^2 f(x_{i,j}).$$

We require that (5.10) holds for each $x_{i,j} \in \Omega$ and we set $u_{i,j} = 0$ when $x_{i,j}$ is outside of Ω (from the boundary condition). The unknowns are the values of $u_{i,j}$ for $x_{i,j}$ in Ω . Thus, there are n unknowns where n is the number of grid nodes inside Ω . There are also n equations.

To get a matrix problem, we have to order the unknowns. The grid node ordering is not useful for this as there are two parameters in the grid ordering and we have only one parameter (index) for a vector. An ordering of the unknowns is a one to one (and onto) mapping k from the set of (i, j) values corresponding to points $x_{i,j} \in \Omega$ to the indices $1, 2, \dots, n$. In this way, the (i, j) grid unknown becomes the $k(i, j)$ 'th vector unknown. We order the

equations the same way so that the equation (5.10) corresponding to (i, j) becomes the $k(i, j)$ 'th equation (row of the matrix which we shall denote by A_5).

We examine the structure of A_5 in more detail. It is clear that the $k(i, j)$ row of the matrix has at most 5 non-zeroes (note that, for example, when $x_{i-1,j}$ is outside of Ω , $u_{i-1,j} = 0$ and so there is no corresponding entry in the matrix). As $u_{i,j}$ is the $k(i, j)$ 'th unknown and the equation corresponding to (i, j) is the $k(i, j)$ 'th equation, the term $4u_{i,j}$ in (5.10) leads to a 4 on the $k(i, j)$ 'th entry of the diagonal. Similarly, the term $-u_{i-1,j}$ gives the entry -1 in $(A_5)_{k(i,j),k(i-1,j)}$ (when $x_{i-1,j} \in \Omega$). The remaining terms of (5.10) produce analogous entries in A_5 .

Lexicographical ordering involves defining k following the text reading direction, i.e., left to right on the first line followed by left to right on the second line, etc. For example a simple triangular grid of nodes would be ordered

$$(5.11) \quad \begin{array}{cccccc} 1 & & & & & \\ 2 & 3 & & & & \\ 4 & 5 & 6 & & & \\ 7 & 8 & 9 & 10 & & \\ 11 & 12 & 13 & 14 & 15 & \\ 16 & 17 & 18 & 19 & 20 & 21 \end{array} \quad .$$

Similar to the previous example, we define a diagonal similarity matrix by

$$(M_z)_{k,k} = z^{i-j} \quad \text{where } k = k(i, j).$$

There is no ambiguity with this definition as there is a unique index pair (i, j) corresponding to each index in $\{1, \dots, n\}$ since $k(\cdot, \cdot)$ is invertible.

We claim that $(M_z)^{-1}J_1M_z = J_z$. Indeed,

- If k_1 corresponds to a grid point directly above $k = k(i, j)$ then $k_1 = k(i, j + 1)$ and

$$((M_z)^{-1}(J_1)M_z)_{k,k_1} = z^{-i+j}(J_1)_{k,k_1}z^{i-(j+1)} = z^{-1}L_{k,k_1}.$$

The last equality came from the observation that k_1 appears before k in the ordering so $J_{k,k_1} = L_{k,k_1}$ (see (5.11)).

- If k_1 corresponds to a grid point directly below $k = k(i, j)$ then $k_1 = k(i, j - 1)$ and

$$((M_z)^{-1}J_1M_z)_{k,k_1} = z^{-i+j}(J_1)_{k,k_1}z^{i-(j-1)} = zU_{k,k_1}.$$

- If k_1 corresponds to a grid point one to the right of $k = k(i, j)$ then $k_1 = k(i + 1, j)$ and

$$((M_z)^{-1}J_1M_z)_{k,k_1} = z^{-i+j}(J_1)_{k,k_1}z^{i+1-j} = zU_{k,k_1}.$$

- If k_1 corresponds to a grid point one to the left of $k = k(i, j)$ then $k_1 = k(i - 1, j)$ and

$$((M_z)^{-1}J_1M_z)_{k,k_1} = z^{-i+j}(J_1)_{k,k_1}z^{i-1-j} = z^{-1}L_{k,k_1}.$$

As these are the only nonzero entries of L and U , $(M_z)^{-1}J_1M_z = J_z$ and so the finite difference matrix satisfies Property A.

Class Notes 5: THE ACCELERATION OF SOR

Math 639d

Due Date: Feb. 14

(updated: February 12, 2018)

We shall see that the judicious choice of parameter ω can lead to a significantly faster converging algorithm. Let us first set the stage. Suppose that we have a linear iterative method with reduction matrix G satisfying $\rho(G) < 1$. Then we know there is a norm $\|\cdot\|_*$ such that the induced matrix norm satisfies

$$\|G\|_* = \gamma,$$

with $\gamma < 1$. Actually, γ can be taken arbitrarily close to $\rho(G)$. We then have

$$\|e_k\|_* \leq \|G\|_*^k \|e_0\|_* = \gamma^k \|e_0\|_*.$$

Now, to reduce the $\|\cdot\|_*$ -error by a factor of ϵ , we need

$$\gamma^k \leq \epsilon, \text{ i.e., } k \geq \frac{\ln(\epsilon^{-1})}{\ln(\gamma^{-1})}.$$

Setting $\delta = 1 - \gamma$, we find that

$$\ln(\gamma^{-1}) = -\ln(\gamma) = -\ln(1 - \delta) \approx \delta$$

where we used Taylor's series for the approximation. Thus, the number of iterations for a fixed reduction should grow proportionally with $\delta^{-1} = (1 - \gamma)^{-1} \approx (1 - \rho(G))^{-1}$, i.e., $k \cdot (1 - \gamma)$ should behave like a constant.

Recall that for A_3 we computed the spectral radius of G_J (the Jacobi method) and found that $\rho(G_J) = \cos(\pi/(n+1)) \approx 1 - \pi/(2(n+1)^2)$. We again used Taylor's series for the approximation. Thus, one should expect that k grows like a constant times n^2 for Jacobi (compare this with your homework results). Our goal is to show that $\rho(G_{SOR}) = 1 - O(n^{-1})$ for an appropriate choice of ω .

To analyze the SOR method we start by setting

$$\beta = \rho(L + U).$$

Note that the Jacobi method in the case when $A = I + L + U$ is

$$x_{i+1} = -(L + U)x_i + b$$

and the corresponding reduction matrix is $G_J = -(L + U)$ so β is nothing more than the spectral radius of G_J .

We introduce the following hypothesis:

(A.1) $0 < \omega < 2$.


(A.2) G_J only has real eigenvalues and $\rho(G_J) = \beta$. with $0 < \beta < 1$.

(A.3) The matrix $A = I + L + U$ satisfies Property A.

Remark 1. *The case when $\beta = 0$ is not interesting since this means that $\rho(G_J) = 0$ and we should use the Jacobi method.*

Remark 2. *The assumption of real eigenvalues is automatically satisfied when A is symmetric.*

Under the above hypothesis, we have the following theorem. This theorem was proved by David Young in his PH.D. thesis (1950) from Harvard University. At that time, computers were in their infancy and it was far from clear whether iterative methods would be an attractive technique for use on the “automatic computing machines” under development. He spent the last 50 years of his career at the University of Texas at Austin.

Theorem 1. *(D. Young ) Assume that (A.1)-(A.3) hold and let G_{SOR}^ω be the reduction matrix corresponding to SOR with iteration parameter ω . Then,*

$$\rho(G_{SOR}^\omega) = \begin{cases} \omega - 1 : & \text{if } \omega \in [\omega_{opt}, 2), \\ 1 - \omega + \frac{1}{2}\omega^2\beta^2 + \omega\beta\sqrt{1 - \omega + \frac{\omega^2\beta^2}{4}} : & \text{if } \omega \in (0, \omega_{opt}]. \end{cases}$$

Here

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \beta^2}}.$$

Remark 3. *We shall see in the proof of the theorem that both of the expressions for $\rho(G_{SOR}^\omega)$ give the same result when $\omega = \omega_{opt}$. In addition, the second is always greater than or equal to $\omega_{opt} - 1$. As $\omega - 1$ is an increasing function of ω , it follows that the choice of $\omega = \omega_{opt}$ gives the smallest spectral radius over all $\omega \in (0, 2)$, namely,*

$$\rho(G_{SOR}^{\omega_{opt}}) = \omega_{opt} - 1 = \frac{1 - \sqrt{1 - \beta^2}}{1 + \sqrt{1 - \beta^2}} \approx 1 - 2\sqrt{1 - \beta^2}.$$

Before proving the theorem, we further investigate the above estimate. We start by assuming that the Jacobi method requires a large number of iterations, i.e., β is positive and close to one, i.e., $\beta = 1 - \gamma$ with γ small. In this case,

$$\sqrt{1 - \beta^2} = \sqrt{2\gamma - \gamma^2} \approx 2^{1/2}\gamma^{1/2}$$

so

$$\rho(G_{SOR}^{\omega_{opt}}) \approx 1 - 2^{3/2}\gamma^{1/2}.$$

Thus, the number of iterations for SOR with the optimal parameter choice should grow like $\gamma^{-1/2}$ instead of the γ^{-1} growth for Jacobi and Gauss Seidel

iteration. This slower growth in the number of iterations will be illustrated in Homework 4. Even though we will only use a ball park estimate for ω_{opt} , we will see considerable acceleration.

Proof. Let λ be a nonzero eigenvalue for G_{SOR}^ω with eigenvector e . Then

$$(5.1) \quad \lambda(\omega^{-1}I + L)e = ((\omega^{-1} - 1)I - U)e$$

or

$$(\lambda\omega L + \omega U)e = (1 - \lambda - \omega)e.$$

Dividing by $\pm\omega\sqrt{\lambda}$ gives

$$\left(\frac{1}{z}L + zU\right)e = \frac{1 - \lambda - \omega}{\pm\omega\sqrt{\lambda}}e$$

where $z = \pm 1/\sqrt{\lambda}$. Note that λ and hence z may be a complex number. It follows by Property A that

$$(5.2) \quad \mu = \frac{1 - \lambda - \omega}{\pm\omega\sqrt{\lambda}}$$

is an eigenvalue of $-G_J = (L + U)$. By Property A, the eigenvalues of $J_{-1} = -L - U$ are the same as $J_1 = L + U$ so that if μ is an eigenvalue of G_J , so is $-\mu$. It follows that equation (5.2) has the same solutions (we think of λ as being the unknown here) as

$$(5.3) \quad (\lambda + \omega - 1)^2 = \omega^2\lambda\mu^2.$$

Note that if $\lambda \neq 0$ satisfies (5.3) for some eigenvalue μ of G_J , then, going through the equations in reverse order implies that there is an eigenvector e satisfying (5.1) with this value of λ . Thus, if λ and μ solve (5.2) or (5.3), then λ is an eigenvalue of G_{SOR} if and only if μ is an eigenvalue of G_J . Examining (5.2) and (5.3), we may assume, without loss of generality, that $\mu \geq 0$.

As seen in the proof of Proposition 2 of Class 4, $\det(G_{SOR}^\omega) = (1 - \omega)^n$ and hence is nonzero except for $\omega = 1$. When $\omega = 1$, any eigenvector e of G_J with eigenvalue $\mu \neq 0$ leads to an eigenvalue $\lambda = \mu^2$ for G_{SOR}^1 . It follows in this case that $\rho(G_{SOR}^1) = \beta^2$ and is in agreement with the theorem¹ (since $\omega_{opt} > 1$ and the second expression of the theorem equals β^2 when $\omega = 1$). For all other ω , G_{SOR}^ω has only nonzero eigenvalues and every one comes from (5.3).

¹If run Jacobi and Gauss-Seidel you will see that Jacobi takes roughly twice as many iterations to converge to the same accuracy.

The equation (5.3) can be rewritten

$$(5.4) \quad \lambda^2 + 2 \left[(\omega - 1) - \frac{\omega^2 \mu^2}{2} \right] \lambda + (\omega - 1)^2 = 0.$$

Its roots are

$$(5.5) \quad \begin{aligned} \lambda^\pm(\mu, \omega) &= (1 - \omega) + \frac{\omega^2 \mu^2}{2} \pm \sqrt{\left(1 - \omega + \frac{\omega^2 \mu^2}{2}\right)^2 - (\omega - 1)^2} \\ &= (1 - \omega) + \frac{\omega^2 \mu^2}{2} \pm \omega \mu \sqrt{1 - \omega + \frac{\omega^2 \mu^2}{4}}. \end{aligned}$$

First, the only way that we can get a complex root is if

$$(5.6) \quad \omega - 1 > \frac{\omega^2 \mu^2}{4}.$$

In this case, the roots are complex conjugates of each other and by (5.4), their product equals $(\omega - 1)^2$, i.e., they each have absolute value $\omega - 1$.

Now, for

$$(5.7) \quad \omega - 1 \leq \frac{\omega^2 \mu^2}{4},$$

the coefficient of λ in (5.4) is negative and hence we have two positive roots. The larger one, $\lambda^+(\mu, \omega)$, is clearly an increasing function of μ (for fixed ω).

When (5.7) holds, $\lambda^+(\mu, \omega)$ is a decreasing function of ω (for fixed μ). To see this we use a geometric argument. Multiplying (5.2) by $\pm\sqrt{\lambda}$, we see that the solutions λ of (5.2) occur at the intersection of the straight line $f_1^\omega(\lambda) = \frac{\lambda + \omega - 1}{\omega}$ and the double valued function $f_2^\mu(\lambda) = \mu\sqrt{\lambda}$. This situation is illustrated in Figure 1 where $f_2^\mu(\lambda)$ for $\mu = .8$ and $f_1^\omega(\lambda)$ for $\omega = 1.25$ and $\omega = .7$ are plotted (you should be able to identify each of these in the picture). Clearly, $f_1^\omega(\lambda)$ is a straight line passing through (1,1) with slope $1/\omega$. As ω increases the line rotates upwards (at least for $\lambda \in [0, 1]$). The rightmost value of λ at the point of intersection of the line and the upper branch of F_2 is $\lambda^+(\mu, \omega)$ and moves to the left as ω increases, i.e., $\lambda^+(\mu, \omega)$ is a decreasing function of ω for fixed μ .

It is not hard to check that the value of ω where f_1^ω is tangent to f_2^μ coincides with the value of ω resulting in the double root (this is also clear from the figure). Thus, ω is the solution of

$$(5.8) \quad \omega - 1 = \frac{\omega^2 \mu^2}{4}.$$

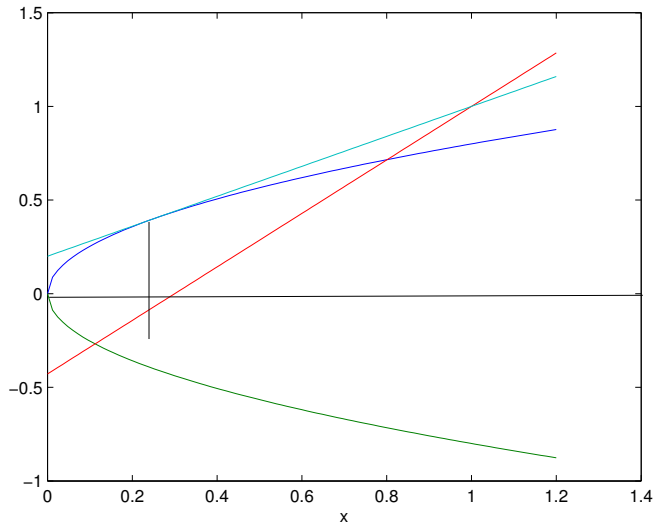


FIGURE 1. f_1 corresponding to $\omega = .7$ and $\omega = \omega_{opt}(.8) = 1.25$ and f_2 with $\mu = .8$ (The axis labeled x should be labeled λ).

We denote the root in $(0, 2)$ by $\omega_{opt}(\mu)$ and calculate

$$\omega_{opt}(\mu) = \frac{2}{1 + \sqrt{1 - \mu^2}}.$$

It easily follows from (5.8) that

$$\lambda^{\pm}(\mu, \omega_{opt}(\mu)) = \omega_{opt}(\mu) - 1.$$

From the above discussion, we see that:

- (a) If $\omega \geq \omega_{opt}(\beta)$, $|\lambda^{\pm}(\mu, \omega)| = \omega - 1$ as this is the complex or double root case for every μ .
- (b) If $\omega \leq \omega_{opt}(\beta)$, then (5.4) with $\mu = \beta$ has positive real roots, the largest given by $\lambda^+(\beta, \omega)$. This is the second expression appearing in the theorem. The remaining $\mu \in \sigma(G_J)$ with $\mu \geq 0$ satisfy $\mu < \beta$. We can only have complex roots if $\omega > 1$ and in this case, as $\lambda^+(\beta, \omega)$ is the larger positive root of (5.4) with $\mu = \beta$, $\lambda^+(\beta, \omega) \geq \omega - 1$. For all values of ω , if μ leads to real roots then they are smaller than $\lambda^+(\beta, \omega)$ as $\lambda^+(\mu, \omega)$ is an increasing function of μ for ω fixed. Thus, all eigenvalues have absolute value at most $\lambda^+(\beta, \omega)$ when $\omega \in (0, \omega_{opt}(\beta)]$.

This completes the proof of the theorem.



Class Notes 7: CONVERGENCE IN NATURAL NORMS FOR THE SPD CASE

Math 639d

Due Date: Feb. 21

(updated: February 28, 2018)

In earlier classes, we saw that provided that the spectral radius of the reduction matrix G was less than one, there was an abstractly defined norm $\|\cdot\|_*$ for which

$$(7.1) \quad \|G\|_* < 1.$$

This, in turn, implied that the method ultimately converged for any initial iterate and right hand side. We also saw that there were examples where even though $\rho(G)$ was zero, no convergence in the L^∞ norm was observed until the n 'th step where n was the number of unknowns (see, Example 1 of Class Notes 3). This, somewhat annoying behavior was a result of a Jordan block with a very large order. Even when large dimensional Jordan blocks are not present, the abstract norm $\|\cdot\|_*$ may be far from some natural norm.

For this class, we shall assume that A is a (real) symmetric and positive definite (SPD) $n \times n$ matrix. Our goal will be to study the convergence of iterative methods in two natural norms, the first being the ℓ^2 norm and the second being the so-called “operator norm.” The operator norm or A -norm (for $x \in \mathbb{R}^n$) is defined by

$$(7.2) \quad \|x\|_A = (Ax, x)^{1/2}$$

where (\cdot, \cdot) is defined by the dot product,

$$(7.3) \quad (x, y) = \sum_{j=1}^n x_j y_j.$$

In your linear algebra course, you should have studied inner products. We shall start with inner products defined on \mathbb{R}^n . An inner product is a mapping from $\mathbb{R}^n \times \mathbb{R}^n$ into \mathbb{R} . One often uses the notation $\langle \cdot, \cdot \rangle$ so that $\langle x, y \rangle$ is a real number when x, y are vectors in \mathbb{R}^n . The inner product is required to satisfy the following axioms:

- (1) $\langle x, x \rangle \geq 0$ for all $x \in \mathbb{R}^n$ and equals 0 only if $x = \mathbf{0}$ (positive definite).
- (2) $\langle x, y \rangle = \langle y, x \rangle$ for all $x, y \in \mathbb{R}^n$ (symmetry).
- (3) For $\alpha, \beta \in \mathbb{R}$ and $x, y, z \in \mathbb{R}^n$,

$$\langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$$

and

$$\langle x, \alpha y + \beta z \rangle = \alpha \langle x, y \rangle + \beta \langle x, z \rangle$$

(bilinearity).

The simplest inner product in \mathbb{R}^n is the ℓ^2 -inner product defined by (7.3).

Exercise 1. Show that (7.3) satisfies the inner product axioms. Show that $(\cdot, \cdot)_A$ defined by

$$(7.4) \quad (x, y)_A = (Ax, y) \text{ for all } x, y \in \mathbb{R}^n$$

satisfies the inner product axioms (this only holds for symmetric and positive definite A).

Inner products have a number of interesting properties. The first is that for any inner product $\langle \cdot, \cdot \rangle$, there is a corresponding norm given by

$$(7.5) \quad \|x\| = \langle x, x \rangle^{1/2} \text{ for all } x \in \mathbb{R}^n.$$

In the case of the dot product, $\|x\|$ is just the length of x . It is obvious that the first two axioms for being a norm are satisfied by (7.5). That the triangle inequality holds is a consequence of the Cauchy-Schwarz (or Schwarz) inequality:

$$(7.6) \quad |\langle x, y \rangle| \leq \|x\| \|y\| \text{ for all } x, y \in \mathbb{R}^n.$$

Indeed,

$$\begin{aligned} \|x + y\|^2 &= \langle x + y, x + y \rangle = \langle x, x \rangle + 2\langle x, y \rangle + \langle y, y \rangle \\ &\leq \|x\|^2 + 2\|x\|\|y\| + \|y\|^2 = (\|x\| + \|y\|)^2 \end{aligned}$$

and the triangle inequality follows by taking square roots.

The inequality (7.6) is obvious for the dot product from the identity

$$(7.7) \quad (x, y) = \|x\| \|y\| \cos(\theta)$$

where θ is the angle between x and y . Although we shall not prove this inequality in general, the Cauchy-Schwarz inequality holds for any inner product, e.g .,

$$|(x, y)_A| \leq \|x\|_A \|y\|_A.$$

Whenever we deal with an inner product, we shall almost always be using the corresponding norm. Usually, we shall indicate that the norm corresponds to the inner product. In any event, we shall always explicitly point out if we are using a norm which does not correspond to the inner product norm.

Remark 1. Note that the A -norm is the norm corresponding to the $(\cdot, \cdot)_A$ inner product (the A -inner product). Consequently, the expression given by the right hand side of (7.2) is indeed a norm.

As we have already noticed, different norms assign different “lengths” to the same vector. The norms coming from different inner products have the same property. Besides defining a length (norm), inner products give rise to a notion of angle by generalizing (7.7), i.e., the angle between nonzero vectors x and y with respect to the inner product $\langle \cdot, \cdot \rangle$ is θ where

$$\cos(\theta) = \frac{\langle x, y \rangle}{\|x\| \|y\|}.$$

Two vectors x and y are orthogonal with respect to the inner product $\langle \cdot, \cdot \rangle$ if $\theta = \pi/2$, i.e., $\langle x, y \rangle = 0$. The angles between two vectors corresponding to different inner products are often not the same as seen in the following example.

Example 1. *Let*

$$x = (1, 1)^t \quad \text{and} \quad y = (1, -1)^t$$

then x and y are orthogonal in (\cdot, \cdot) , i.e., $(x, y) = 0$. The matrix

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

is symmetric and positive definite and $(x, y)_A = -1$ (compute it!). Thus, x and y are not orthogonal in $(\cdot, \cdot)_A$.

You should recall from your linear algebra class, that a set of vectors $\{v_1, v_2, \dots, v_k\}$ are (mutually) orthogonal if $\langle v_i, v_j \rangle = 0$ when $i \neq j$. This set of vectors is called orthonormal if in addition, $\|v_i\| = 1$ for $i = 1, \dots, k$. Of course, $\|\cdot\|$ is the norm corresponding to $\langle \cdot, \cdot \rangle$.

Definition 1. *We say that an $n \times n$ matrix B is “self-adjoint” with respect to the inner product $\langle \cdot, \cdot \rangle$ on \mathbb{R}^n if*

$$\langle Bx, y \rangle = \langle x, By \rangle \quad \text{for all } x, y \in \mathbb{R}^n.$$

It is easy to check (do it!) that an $n \times n$ matrix A is self adjoint with respect to (\cdot, \cdot) if and only if A is symmetric. We also generalize the notion of positive definite and say that A is positive definite with respect to $\langle \cdot, \cdot \rangle$ if $\langle Ax, x \rangle \geq 0$ for all $x \in \mathbb{R}^n$ and $\langle Ax, x \rangle = 0$ only if $x = \mathbf{0}$.

Example 2. *Let A and B be $n \times n$ matrices with A symmetric and positive definite and B symmetric. In general, BA is not symmetric (unless B and A commute). However, BA is always self adjoint with respect to the A -inner product since*

$$(BAx, y)_A = (ABAx, y) = (BAx, Ay) = (Ax, BAy) = (x, BAy)_A.$$

Note that if B is positive definite then BA is positive definite with respect to the A -inner product since

$$(BAx, x)_A = (BAx, Ax)$$

which is non-negative (and zero only if $x = \mathbf{0}$ since A is non-singular).

Exercise 2. Assume that A and B are as in the previous example with B SPD. Show that B^{-1} is also symmetric and positive definite. Show that BA is self adjoint with respect to the B^{-1} -inner product.

The following theorem will be important for the analysis of iterative methods. Its proof is available in any reasonably good book on linear algebra.

Theorem 1. (Spectral Theorem). Let A be an $n \times n$ symmetric real matrix. Then there is an orthonormal basis of (real) eigenvectors for A , i.e. there is a set n vectors $\{\phi_1, \dots, \phi_n\}$ satisfying

$$A\phi_i = \lambda_i\phi_i, \quad \text{for } i = 1, \dots, n$$

and

$$(\phi_i, \phi_j) = \delta_{ij} \equiv \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

We shall also use a more general version given in the following theorem. The result in the generality stated can be easily derived from the simpler case and is an exercise (below).

Theorem 2. (A More General Spectral Theorem). Let A and B be $n \times n$ matrices with B symmetric and positive definite and A self adjoint with respect to the B -inner product. Then there is a B -orthonormal basis of (real) eigenvectors with real eigenvalues for A , i.e. there is a set of n vectors $\{\phi_1, \dots, \phi_n\}$ satisfying

$$A\phi_i = \lambda_i\phi_i, \quad \text{for } i = 1, \dots, n$$

and

$$(\phi_i, \phi_j)_B = \delta_{ij} \equiv \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

As an example of the application of the first theorem, we consider the definition of the square root of a symmetric and positive $n \times n$ matrix A . Let $\{\phi_1, \dots, \phi_n\}$ be the orthonormal basis of eigenvectors. We expand $x \in \mathbb{R}^n$ as

$$x = \sum_{i=1}^n c_i \phi_i$$

and define

$$(7.8) \quad A^{1/2}x = \sum_{i=1}^n \lambda_i^{1/2} c_i \phi_i \quad (\text{symmetry})$$

(symmetry)

Theorem: If A is positive definite matrix, we must have $A^{\frac{1}{2}}$.

we don't say

$$A^T = A$$

In other words.

doesn't mean $A^T = A$.

all eigenvalues of A are positive.

5

We already observed in an earlier class that the eigenvalues of a positive definite matrix A are all positive and we set $\lambda_i^{1/2}$ to be the positive square root. The above definition is more of a transformation representation of the matrix $A^{1/2}$. A more explicit (equivalent) representation is given by

$A^{-1/2}$ means,
the inverse of $A^{1/2}$

$$(7.9) \quad A^{1/2} = \sum_{i=1}^n \lambda_i^{1/2} \phi_i [\phi_i]^t.$$

This is a sum of $n \times n$ matrices since the matrix product $\phi_i [\phi_i]^t$ is an $n \times n$ matrix for each i . It is not hard to check that:

- (1) The matrix product $A^{1/2} A^{1/2}$ equals A .
- (2) $A^{1/2}$ is invertible and its inverse is given by $A^{-1/2}$ (defined as in (7.8) or (7.9) with $\lambda_i^{1/2}$ replaced by $\lambda_i^{-1/2}$).
- (3) The matrix $A^{1/2}$ is symmetric and positive definite.

Exercise 3. *Hint:* Use Theorem 1 to prove the general version of the spectral theorem. First show that $C = B^{1/2} A B^{-1/2}$ is a symmetric matrix. Apply Theorem 1 to define an orthogonal basis $\{\tilde{\phi}_1, \dots, \tilde{\phi}_n\}$ of eigenvectors for C . Then show that $\{\phi_1, \phi_2, \dots, \phi_n\}$ where $\phi_j = B^{-1/2} \tilde{\phi}_j$ are B^{-1} -orthonormal eigenvectors for A . Hint: You will need to use properties (1), (2), (3) above with B replacing A .

Corollary 1. Let A and B be as in the above theorem. Then

$$\|A\|_B = \rho(A).$$

Here $\|A\|_B$ denotes the operator norm of A induced from the vector norm $\|\cdot\|_B$.

Proof. Let λ_i and ϕ_i , $i = 1, \dots, n$ be as in the general spectral theorem. Let $x \in \mathbb{R}^n$ be nonzero with expansion

$$x = \sum_{i=1}^n c_i \phi_i.$$

Then, by bilinearity,

$$\begin{aligned} (x, x)_B &= \left(\sum_{i=1}^n c_i \phi_i, \sum_{j=1}^n c_j \phi_j \right)_B = \sum_{i,j=1}^n c_i c_j (\phi_i, \phi_j)_B \\ &= \sum_{i=1}^n c_i^2. \end{aligned}$$

A similar computation using the identity

$$Ax = \sum_{i=1}^n c_i A\phi_i = \sum_{i=1}^n c_i \lambda_i \phi_i$$

gives

$$(Ax, Ax)_B = \sum_{i=1}^n c_i^2 \lambda_i^2 \leq \rho(A)^2 \sum_{i=1}^n c_i^2 = \rho(A)^2 (x, x)_B.$$

Thus,

$$\frac{\|Ax\|_B}{\|x\|_B} = \frac{(Ax, Ax)_B^{1/2}}{(x, x)_B^{1/2}} \leq \rho(A).$$

Taking the supremum over x above gives $\|A\|_B \leq \rho(A)$. The opposite inequality was proved in an earlier class, i.e., $\|A\| \geq \rho(A)$ holds for any norm $\|\cdot\|$. \square

Application: Let A be a symmetric and positive definite $n \times n$ matrix and consider the linear iterative method (the Richardson Method)

$$x_{i+1} = x_i + \tau(b - Ax_i).$$

Its reduction matrix is given by $G = (I - \tau A)$. We clearly have that

$$\rho(G) = \max_{i=1}^n |(1 - \tau \lambda_i)|$$

where $\lambda_i, i = 1, \dots, n$ are the eigenvalues of A . Here we used the fact that the eigenvalues of $G = (I - \tau A)$ are $(1 - \tau \lambda_i), i = 1, \dots, n$. Since A is symmetric with respect to the usual inner product, G is also. It is also obvious that G is self adjoint with respect to the A -inner product (check it!). It follows from the corollary that $\|G\| = \|G\|_A = \rho(G)$. This immediately implies that

$$\|e_{i+1}\| \leq \rho(G) \|e_i\| \quad \text{and} \quad \|e_{i+1}\|_A \leq \rho(G) \|e_i\|_A.$$

This means that each step of the iteration is guaranteed to reduce the error by a factor of $\rho(G)$ in either of the natural norms $\|\cdot\|$ or $\|\cdot\|_A$.

If $\tilde{\lambda}$ is any bound on $\rho(A)$ then taking $\tau = 1/\tilde{\lambda}$ leads to

$$\rho(G) = 1 - \lambda_1/\tilde{\lambda}.$$

Here we have assumed, w.l.o.g, that the eigenvalues of A are ordered so that $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. It is natural to look for the value of τ which results in the smallest value of $\rho(G)$, i.e., the fastest reduction rate.

To find this τ , we consider the graph of $f(\lambda) = 1 - \tau\lambda$ and its behavior as we vary τ . It is clear that the spectral radius of G for this value of τ is bounded from above by the maximum of the absolute value of $f(\lambda)$ over the interval $[\lambda_1, \lambda_n]$. Figure 1 shows two values of τ . The one on the left

corresponds to a smaller τ . For the left hand plot, it is clear that the spectral radius of G is taken on at λ_1 and is given by $1 - \tau\lambda_1$. The second picture is more interesting. From the picture, it is clear that the spectral radius again comes from λ_1 . As we make τ larger, the line will rotate in a clockwise direction (it always passes through $(0, 1)$). It is clear that the value at λ_1 will decrease as τ becomes larger. Moreover, the absolute value of $f(\lambda_n)$ will be increasing (once τ is large enough so that $f(\lambda_n) \leq 0$). It is clear that we get the smallest spectral radius when $f(\lambda_1) = -f(\lambda_n)$, i.e.,

$$1 - \tau\lambda_1 = \tau\lambda_n - 1$$

or

$$(7.10) \quad \tau = \frac{2}{\lambda_1 + \lambda_n}.$$

For this choice of τ , we get (check it!)

$$\rho(G) = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}.$$

This is the optimal choice of τ and results in the smallest value of $\rho(G_\tau)$, i.e., the fastest convergence. Unfortunately, though, the use of this parameter in practice requires the knowledge of λ_1 and λ_n , quantities which may not be readily available.

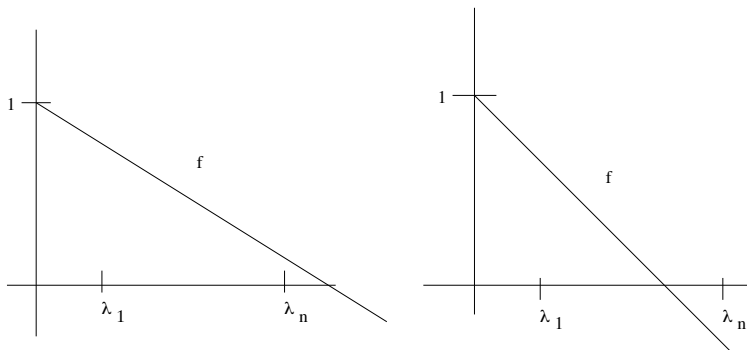


FIGURE 1. $f(\lambda)$ for two different values of τ .

The errors obtained using the Richardson method can be further characterized by expansion in the eigenvectors. Indeed, consider expanding

$$e_i = \sum_{j=1}^n c_j^i \phi_j.$$

Then

$$e_{i+1} = \sum_{j=1}^n c_j^{i+1} \phi_j = (I - \tau A) \sum_{j=1}^n c_j^i \phi_j = \sum_{j=1}^n (1 - \tau \lambda_j) c_j^i \phi_j.$$

The above gives two different expansions for the same vector in a basis. It follows that the coefficients must be equal, i.e.,

$$c_j^{i+1} = (1 - \tau \lambda_j) c_j^i.$$

Thus, each step of the iterative method reduces absolute value of the j 'th coefficient in the expansion by a factor of $|1 - \tau \lambda_j|$. We gather some of these results into the following proposition.

Proposition 1. *Let A be a symmetric and positive definite matrix with maximum and minimum eigenvalues λ_n and λ_1 , respectively. Then, for τ given by (7.10),*

$$\|I - \tau A\|_{\ell^2} = \|I - \tau A\|_A = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}.$$

The results of this class go over without change when A is a complex matrix provided that it is conjugate symmetric and positive definite. We also refer to conjugate symmetric matrices as Hermitian.

The changes to the development are as follows.

- (1) We use Hermitian inner products $\langle \cdot, \cdot \rangle$ defined on $\mathbb{C}^n \times \mathbb{C}^n$ which produce complex numbers satisfying:
 - (a) $\langle x, y \rangle = \overline{\langle y, x \rangle}$ for all $x, y \in \mathbb{C}^n$.
 - (b) $\langle x, x \rangle \geq 0$ for all $x \in \mathbb{C}^n$ and equals 0 only if $x = \mathbf{0}$ (positive definite).
 - (c) For $\alpha, \beta \in \mathbb{C}$ and $x, y, z \in \mathbb{C}^n$,

$$\langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$$

and

$$\langle x, \alpha y + \beta z \rangle = \bar{\alpha} \langle x, y \rangle + \bar{\beta} \langle x, z \rangle.$$

- (2) The ℓ^2 -Hermitian inner product is given by

$$(x, y) := \sum_{i=1}^n x_i \bar{y}_i.$$

- (3) For a Hermitian, positive definite matrix A , the A -inner product is also a Hermitian inner product,

$$(x, y)_A := (Ax, y) \text{ for all } x, y \in \mathbb{C}^n$$

where (\cdot, \cdot) is the ℓ^2 Hermitian inner product above.

- (4) A matrix A is self adjoint with respect to a Hermitian inner product $\langle \cdot, \cdot \rangle$ if

$$\langle Ax, y \rangle = \langle x, Ay \rangle \quad \text{for all } x, y \in \mathbb{C}^n.$$

- (5) Matrices which are self adjoint with respect to a Hermitian inner product have real eigenvalues and the spectral theorems above hold provided that one replaces inner products with Hermitian inner products, symmetric matrices with conjugate symmetric matrices and \mathbb{R}^n with \mathbb{C}^n . Moreover, the eigenvectors can be taken to be real. The corollary also holds, i.e.,

$$\|A\|_B = \rho(A).$$

when A is self adjoint with respect to the B -inner product.

If A is a complex Hermitian positive definite matrix then the analysis above for the Richardson method carries over without change. This includes the choice of iterative parameter ($\tau = 2/(\lambda_1 + \lambda_n)$ is the best choice) and behavior of the coefficients in the eigenvector expansion of the error, i.e.,

$$c_j^{i+1} = (1 - \tau\lambda_j)c_j^i.$$

Class Notes 8: PRECONDITIONING

Math 639d

Due Date: March 7

(updated: October 20, 2020)

We now consider preconditioning. First, we consider a motivational example (involving A_3 discussed earlier).

Consider the tridiagonal matrix $n \times n$ matrix A_3 defined by

$$(A_3)_{ij} = \begin{cases} 2 & \text{if } i = j, \\ -1 & \text{if } |i - j| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

As we have already seen, the eigenvalues and eigenvectors of A_3 are

$$(8.1) \quad \lambda_i = 2 - 2 \cos \left(\frac{i\pi}{n+1} \right) = 4 \sin^2(i\pi/2(n+1)).$$

and

$$\phi_i = \left(\sin \left(\frac{i\pi}{n+1} \right), \sin \left(\frac{2i\pi}{n+1} \right), \dots, \sin \left(\frac{ni\pi}{n+1} \right) \right)^t,$$

for $i = 1, \dots, n$. Note that $\lambda_1 < \lambda_2 < \dots < \lambda_n$.

Suppose we take $\tau = 1/\lambda_n$ in Richardson's method,

$$x_{i+1} = x_i + \tau(b - Ax_i).$$

Then $G = (I - \tau A)$ is the corresponding reducer and

$$\sigma(G) = \{1 - \lambda/\lambda_n : \lambda \in \sigma(A)\}.$$

This implies that $\rho := \rho(G) = 1 - \lambda_1/\lambda_n := 1 - 1/K$ where K denotes the largest eigenvalue of A divided by the smallest, i.e., $K = \lambda_n/\lambda_1$.

Definition 1. For a symmetric and positive definite real valued $n \times n$ matrix A , the spectral condition number of A (often denoted by K) is defined to be the largest eigenvalue of A divided by the smallest eigenvalue of A .

Remark 1. Let A be a symmetric and positive definite real $n \times n$ matrix and $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ be its eigenvalues. Taking B to be the identity in the corollary of the previous lecture gives

$$\lambda_n = \|A\|_{\ell^2}.$$

Clearly, the eigenvalues of A^{-1} are $\lambda_n^{-1} \leq \dots \leq \lambda_2^{-1} \leq \lambda_1^{-1}$ so $\|A^{-1}\|_{\ell^2} = \lambda_1^{-1}$. Thus, the spectral condition number K of A is given by

$$K = \lambda_n/\lambda_1 = \|A\|_{\ell^2} \|A^{-1}\|_{\ell^2}.$$

The norms appearing above are the operator norms induced by the $\|\cdot\|_{\ell^2}$ norm on \mathbb{R}^n . In general, the condition number of a matrix A with respect to a norm $\|\cdot\|$ on \mathbb{R}^n is defined by

$$\text{Cond}(A) = \|A\| \|A^{-1}\|$$

and depends on the choice of norm.

In general, the condition number provides an indication of the behavior of iterative methods. To achieve a fixed reduction, the direct application of a simple iterative method to a matrix with a large condition number requires a number of steps proportional to the condition number. The idea of “preconditioning” is to transform the problem with large condition number to one which has a significantly smaller one.

Preconditioning: Suppose that we want to iteratively solve the system

$$(8.2) \quad Ax = b$$

involving an $n \times n$ nonsingular matrix A which is poorly conditioned (i.e., has large condition number). A preconditioner B is another nonsingular $n \times n$ matrix. Multiplying (8.2) by B does not change the solution (why?) and we consider the iterative solution of

$$(8.3) \quad BAx = Bb.$$

A good preconditioner will satisfy the following properties:

- (1) The application B to a vector $v \in \mathbb{R}^n$ should be relatively cheap, i.e., it should not take much more computer effort than the cost of applying A to a vector. So, for example, if A is sparse and has $O(n)$ non-zeroes, then the application of an ideal B (to a vector in \mathbb{R}^n) should only involve $O(n)$ operations.
- (2) The condition number of BA should be significantly smaller than that of A .

We first consider two trivial examples for B . Clearly, $B = I$ does nothing at all. Alternatively, we could consider $B = A^{-1}$ in which case the system becomes $x = A^{-1}b$ and we have solved the problem. Of course, the whole point of iterative methods is to get an accurate approximation to the solution x in much less time (computational effort) than it would have taken to solve the problem by direct methods. Neither of the extremes, $B = I$ or $B = A^{-1}$ are useful choices for a preconditioner in a preconditioned iterative method.

The construction and analysis of preconditioners has been the subject of intensive research in the last forty years. As we proceed with this course, we shall examine some basic ideas for the construction of preconditioners. As

for now, we shall assume that B has been given and see what properties are required for effective preconditioning.

We consider the case when A and B are symmetric and positive definite (real) matrices. The Richardson method applied to (8.3) becomes

$$x_{i+1} = x_i + \tau B(b - Ax_i)$$

and the error $e_i = x - x_i$ satisfies

$$e_{i+1} = Ge_i \text{ where } G = I - \tau BA.$$

In general, BA is no longer symmetric but, as we observed in Class Notes 7, BA is self adjoint with respect to either the A -inner product or the B^{-1} -inner product. This means that G is also self adjoint with respect to either inner product. The corollary of the Class Notes 7 (with B replacing A of the corollary and A or B^{-1} replacing B of the corollary) implies that

$$\|G\|_{B^{-1}} = \|G\|_A = \rho(G) = \max_{i=1}^n |1 - \tau \lambda_i|.$$

Here $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ are the (real) eigenvalues of BA (see, Theorem 2 of Class Notes 7). Note that it follows that

$$\|e_j\|_A = \|G^j e_0\|_A \leq \rho(G)^j \|e_0\|_A \leq (1 - 1/K)^j \|e_0\|_A$$

when $\tau = \lambda_n$. The analysis of Homework/Programming 5 concerning the number of iterative steps applies here as well and we find that we can get a fixed reduction in the A -norm with a number of iterations proportional to $K = \lambda_n/\lambda_1$. The difference is that the eigenvalues now appearing are those for BA and not for A .

The following proposition is a useful characterization of the eigenvalues of BA when B and A are symmetric and positive definite.

Proposition 1. *Let B and A be symmetric and positive definite (real) $n \times n$ matrices and λ_1 and λ_n be, respectively, the smallest and largest eigenvalues for BA . Then λ_1 is the maximum value for c_0 and λ_n is the minimal value of c_1 satisfying any of the following inequalities:*

$$(8.4) \quad c_0(Ax, x) \leq (ABAx, x) \leq c_1(Ax, x), \quad \text{for all } x \in \mathbb{R}^n,$$

$$(8.5) \quad c_0(B^{-1}x, x) \leq (Ax, x) \leq c_1(B^{-1}x, x), \quad \text{for all } x \in \mathbb{R}^n,$$

$$(8.6) \quad c_0(A^{-1}x, x) \leq (Bx, x) \leq c_1(A^{-1}x, x), \quad \text{for all } x \in \mathbb{R}^n.$$

Before proving the above proposition, we note that if A and B are symmetric and positive definite real matrices then the largest and smallest eigenvalues of BA are respectively the maximum and minimum of the Rayleigh

quotient, i.e.,

$$(8.7) \quad \lambda_n = \max_{x \in \mathbb{R}^n, x \neq \mathbf{0}} \frac{(BAx, x)_A}{(x, x)_A} \quad \text{and} \quad \lambda_1 = \min_{x \in \mathbb{R}^n, x \neq \mathbf{0}} \frac{(BAx, x)_A}{(x, x)_A}.$$

(The quantity

$$\lambda(x) = \frac{(BAx, x)_A}{(x, x)_A}$$

is often referred to as the Rayleigh quotient at x .)

Exercise 1. Prove (8.7). *Hint: Use the general version of the spectral theorem given in the previous class.*

(Proof of the proposition). As already observed, BA is self adjoint and positive definite with respect to the A -inner product. By (8.7), λ_n is the minimal number satisfying

$$\frac{(BAx, x)_A}{(x, x)_A} \leq \lambda_n \quad \text{for all } x \in \mathbb{R}^n, x \neq \mathbf{0}.$$

This is the right hand inequality in (8.4), the left follows analogously. The two inequalities in (8.5) follow from similar arguments and the fact that BA is self adjoint in the B^{-1} -inner product. Finally, (8.6) follows from substituting $x = A^{-1}y$ in (8.4). \square

Remark 2. The above proposition enables us to get bounds on the spectral condition number corresponding to the preconditioning system by deriving inequalities appearing (8.4)-(8.6). Thus, if any one of the left hand inequalities hold with c_0 and any one of the right hand inequalities hold with c_1 , then the spectral condition number K satisfies

$$K \equiv K(BA) \leq \frac{c_1}{c_0}.$$

Example 1. The matrix A_5 comes from a finite difference approximation to the two dimensional boundary value problem

$$(8.8) \quad \begin{aligned} -\Delta u(x) &= f, \quad \text{for } x = (x_1, x_2) \text{ in } (0, 1)^2 \\ u(x) &= 0, \quad \text{for } x_1 = 0 \text{ or } 1 \text{ or } x_2 = 0 \text{ or } 1. \end{aligned}$$

Here Δ denotes the Laplacian and is defined by

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}.$$

Let $h = 1/(n+1)$ (for n a large integer) and $x_{i,j} = (ih, jh)$, $i, j = 0, \dots, n+1$. Note that if i or j is 0 or $n+1$, then $x_{i,j}$ is on the boundary of the square

$(0, 1)^2$. The finite difference approximation to (8.9) is a mesh vector $\{u_{i,j}^h\}$ satisfying

$$(8.9) \quad 4u_{i,j}^h - u_{i-1,j}^h - u_{i+1,j}^h - u_{i,j+1}^h - u_{i,j-1}^h = h^2 f(x_{i,j}),$$

for $i, j = 1, 2, \dots, n$ with

$$(8.10) \quad u_{i,j}^h = 0 \text{ when } i \text{ or } j = 0 \text{ or } n+1.$$

We have seen that finite difference problems can be written as linear systems when we choose an ordering of the unknowns. The above finite difference equations lead to a linear system with n^2 unknowns, corresponding to $i, j = 1, \dots, n$. Homework 3 and 4 used CSR files for the five point operator on a triangle.

Using lexicographical ordering, the finite difference problem is converted to a matrix problem

$$A_5 U = F$$

on \mathbb{R}^{n^2} as discussed in an earlier class. For example, we set $k(i, j) = i + (j - 1) * n$ so that $U_{k(i,j)} = u_{i,j}^h$ and $F_{k(i,j)} = h^2 f(x_{i,j})$. The matrix A_5 is given by

$$(A_5)_{k(i_1,j_1),k(i_2,j_2)} = \begin{cases} 4 & \text{if } i_1 = i_2 \text{ and } j_1 = j_2 \\ -1 & \text{if } i_1 = i_2 \text{ and } j_1 = j_2 + 1 \text{ and } j_2 \neq n \\ -1 & \text{if } i_1 = i_2 \text{ and } j_1 = j_2 - 1 \text{ and } j_2 \neq 1 \\ -1 & \text{if } i_1 = i_2 + 1 \text{ and } j_1 = j_2 \text{ and } i_2 \neq n \\ -1 & \text{if } i_1 = i_2 - 1 \text{ and } j_1 = j_2 \text{ and } i_2 \neq 1 \\ 0 & \text{otherwise.} \end{cases}$$

As we vary i_1, i_2, j_1, j_2 in the set $\{1, 2, \dots, n\}$, we get all possible pairs of indices, $i = k(i_1, j_1), j = k(i_2, j_2) \in \{1, \dots, n^2\}$. It is not hard to see that A_5 is symmetric.

Examining the matrix A_5 and its relation to the finite difference equation we have that

$$(A_5 V, V) = \sum_{i,j=1}^n (4v_{i,j}^h - v_{i-1,j}^h - v_{i+1,j}^h - v_{i,j+1}^h - v_{i,j-1}^h) v_{i,j}^h$$

where $V(k(i, j)) = v_{i,j}^h$. Now

$$\begin{aligned} \sum_{i=1}^n (2v_{i,j}^h - v_{i-1,j}^h - v_{i+1,j}^h) v_{i,j}^h &= \sum_{i=1}^n (v_{i,j}^h - v_{i-1,j}^h) v_{i,j}^h - \sum_{l=2}^{n+1} (v_{l,j}^h - v_{l-1,j}^h) v_{l-1,j}^h \\ &= (v_{1,j}^h)^2 + \sum_{i=2}^n (v_{i,j}^h - v_{i-1,j}^h)^2 + (v_{n,j}^h)^2. \end{aligned}$$

For the first equality above, we split each term on the left into two and changed the index $l = i + 1$ for the second. Using this and the analogous identity

$$\sum_{j=1}^n (2v_{i,j}^h - v_{i,j-1}^h - v_{i,j+1}^h) v_{i,j}^h = (v_{i,1}^h)^2 + \sum_{j=2}^n (v_{i,j}^h - v_{i,j-1}^h)^2 + (v_{i,n}^h)^2$$

gives

$$(8.11) \quad \begin{aligned} (A_5 V, V) = & \sum_{j=1}^n \left((v_{1,j}^h)^2 + \sum_{i=2}^n (v_{i,j}^h - v_{i-1,j}^h)^2 + (v_{n,j}^h)^2 \right) \\ & + \sum_{i=1}^n \left((v_{i,1}^h)^2 + \sum_{j=2}^n (v_{i,j}^h - v_{i,j-1}^h)^2 + (v_{i,n}^h)^2 \right). \end{aligned}$$

Note that it immediately follows from the above identity that A_5 is positive definite (why?).

We now consider solving a variable coefficient problem on the same domain, i.e.,

$$\begin{aligned} -\frac{\partial}{\partial x_1} a_1(x) \frac{\partial u(x)}{\partial x_1} - \frac{\partial}{\partial x_2} a_2(x) \frac{\partial u(x)}{\partial x_2} &= f, \quad \text{for } x = (x_1, x_2) \text{ in } (0, 1)^2 \\ u(x) &= 0, \quad \text{for } x_1 = 0 \text{ or } 1 \text{ or } x_2 = 0 \text{ or } 1. \end{aligned}$$

Its finite difference approximation is of the form

$$(8.12) \quad \begin{aligned} & \left[a_1(x_{i+1/2,j})(u_{i,j}^h - u_{i+1,j}^h) + a_1(x_{i-1/2,j})(u_{i,j}^h - u_{i-1,j}^h) \right. \\ & \quad \left. + a_2(x_{i,j+1/2})(u_{i,j}^h - u_{i,j+1}^h) + a_2(x_{i,j-1/2})(u_{i,j}^h - u_{i,j-1}^h) \right] = h^2 f(x_{i,j}), \end{aligned}$$

for $i, j = 1, 2, \dots, n$ with (8.10). It leads to the matrix \tilde{A}_5 (using the same ordering)

$$(\tilde{A}_5)_{k(i_1,j_1),k(i_2,j_2)} = \begin{cases} (a_1(x_{i_1+1/2,j_1}) + a_1(x_{i_1-1/2,j_1}) + a_2(x_{i_1,j_1+1/2}) + a_2(x_{i_1,j_1-1/2})) & \text{if } i_1 = i_2 \text{ and } j_1 = j_2 \\ -a_2(x_{i_1,j_1+1/2}) & \text{if } i_1 = i_2 \text{ and } j_1 = j_2 + 1 \text{ and } j_2 \neq n \\ -a_2(x_{i_1,j_1-1/2}) & \text{if } i_1 = i_2 \text{ and } j_1 = j_2 - 1 \text{ and } j_2 \neq 1 \\ -a_1(x_{i_1+1/2,j_1}) & \text{if } i_1 = i_2 + 1 \text{ and } j_1 = j_2 \text{ and } i_2 \neq n \\ -a_1(x_{i_1-1/2,j_1}) & \text{if } i_1 = i_2 - 1 \text{ and } j_1 = j_2 \text{ and } i_2 \neq 1 \\ 0 & \text{otherwise.} \end{cases}$$

This matrix is also symmetric. Moreover, a computation similar to that leading to (8.11) gives

$$\begin{aligned}
 (\tilde{A}_5 V, V) = & \sum_{j=1}^n \left(a_1(x_{1/2,j})(v_{1,j}^h)^2 + \sum_{i=2}^n a_1(x_{i-1/2,j})(v_{i,j}^h - v_{i-1,j}^h)^2 \right. \\
 (8.13) \quad & \left. + a_1(x_{n+1/2,j})(v_{n,j}^h)^2 \right) + \sum_{i=1}^n \left(a_2(x_{i,1/2})(v_{i,1}^h)^2 + \right. \\
 & \left. \sum_{j=2}^n a_2(x_{i,j-1/2})(v_{i,j}^h - v_{i,j-1}^h)^2 + a_2(x_{i,n+1/2})(v_{i,n}^h)^2 \right).
 \end{aligned}$$

Suppose that the coefficients a_1 and a_2 are positive, bounded from above and bounded away from zero below. Specifically, suppose that there are constants $0 < \mu_0 \leq \mu_1$ satisfying

$$\mu_0 \leq a_1(x) \leq \mu_1 \quad \text{and} \quad \mu_0 \leq a_2(x) \leq \mu_1 \quad \text{for all } x \in (0, 1)^2.$$

It follows that \tilde{A}_5 is also positive definite. It can be efficiently solved by preconditioned iteration, using $B = A_5^{-1}$ as a preconditioner. Examining the identities (8.11) and (8.13), we find that

$$\mu_1^{-1}(\tilde{A}_5 V, V) \leq (A_5 V, V) \leq \mu_0^{-1}(\tilde{A}_5 V, V) \quad \text{for all } V \in \mathbb{R}^{n^2}.$$

This can be rewritten (with $B^{-1} = A_5$)

$$\mu_0(B^{-1}V, V) \leq (\tilde{A}_5 V, V) \leq \mu_1(B^{-1}V, V) \quad \text{for all } V \in \mathbb{R}^{n^2}$$

and so $K(B\tilde{A}_5) \leq \mu_1/\mu_0$ follows by applying the proposition. The preconditioned iteration converges very fast when μ_1/μ_0 is not large and the convergence rate is independent of h and the number of unknowns. This will be illustrated in the next programming exercise.

Class Notes 10: MULTI-PARAMETER ITERATION

Math 639d

Due Date: March 7

(updated: October 22, 2020)

In this class, we shall consider using multiple parameters in a Richardson like iteration. Specifically, we introduce parameters $\tau_1, \tau_2, \dots, \tau_M$ and consider the multi-parameter iteration

$$(10.1) \quad \begin{aligned} x_{l+1} &= x_{l,M} \text{ where } x_{l,0} = x_l \text{ and} \\ x_{l,j} &= x_{l,j-1} + \tau_j(b - Ax_{l,j-1}), \text{ for } j = 1, \dots, M. \end{aligned}$$

As usual, we start with an initial iterate x_0 . Each iterative step $x_l \rightarrow x_{l+1}$ involves M simple iterations using each of the iteration parameters.

Remark 1. *The multi-parameter iteration is important in that its analysis will be used to analyze a much more practical iterative algorithm, namely, the conjugate gradient (CG) method. The CG method will be discussed in subsequent lectures.*

Remark 2. *The multi-parameter iteration poses a significant problem, “How should the iteration parameters be chosen.” The optimal choice of iteration parameters may, in fact, be an intractable problem. In these notes, we shall develop a good choice of parameters in the positive definite case.*

We next try to get some insight into the behavior of the multi-parameter method. Let $e_{l,j} = x - x_{l,j}$. It is immediate the second line of (10.1) that

$$e_{l,j} = (I - \tau_j A)e_{l,j-1}$$

so

$$e_{l+1} = \left(\prod_{j=1}^M (I - \tau_j A) \right) e_l$$

where, as usual, $e_l = x - x_l$. The iteration matrix for this process is thus

$$(10.2) \quad G_M = \left(\prod_{j=1}^M (I - \tau_j A) \right).$$

There is clearly no point in using $\tau_j = 0$ so we will assume that all iteration parameters are nonzero.

All of the terms in (10.2) commute since A and I commute so we need not worry about the order of the multiplications. Moreover, the above expression represents a polynomial of degree M in A . Expanding G_M we find that

$$(10.3) \quad G_M = I + c_1 A + \dots + c_M A^M$$

for appropriately defined coefficients c_i , $i = 1, \dots, M$.

Now if λ_i is an eigenvalue of A with eigenvector ϕ_i , then

$$(10.4) \quad \left(\prod_{j=1}^M (I - \tau_j A) \right) \phi_i = \left(\prod_{j=1}^M (1 - \tau_j \lambda_i) \right) \phi_i.$$

Moreover, by the Jordan Decomposition Theorem, $A = NJN^{-1}$ with J a block diagonal matrix with Jordan blocks on the diagonal (See, Class Notes 4). It follows that

$$G_M = N \left(\prod_{j=1}^M (I - \tau_j J) \right) N^{-1}.$$

Because of the structure of the Jordan Blocks, the matrix

$$\prod_{j=1}^M (I - \tau_j J)$$

is upper triangular with diagonal entries

$$\prod_{j=1}^M (1 - \tau_j \lambda_i).$$

Thus, all of the eigenvalues of G_M are given in the form of (10.4) and

$$\sigma(G_M) = \left\{ \left(\prod_{j=1}^M (1 - \tau_j \lambda_i) \right) : \lambda_i \in \sigma(A) \right\}.$$

It immediately follows that

$$\rho(G_M) = \max \left\{ \left| \prod_{j=1}^M (1 - \tau_j \lambda_i) \right| : \lambda_i \in \sigma(A) \right\}.$$

Remark 3. Suppose a general nonsingular matrix A has only k distinct eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_k$. Then, as above,

$$\rho(G_k) = \max \left\{ \left| \prod_{j=1}^k (1 - \tau_j \lambda_i) \right| : \lambda_i \in \sigma(A) \right\}.$$

It follows that if $\tau_i = (\lambda_i)^{-1}$ for $i = 1, \dots, k$ then $\rho(G_k) = 0$. Thus, from an earlier theorem, we know that given $\epsilon > 0$, there is a norm $\|\cdot\|_*$ for which $\|G_k\|_* < \epsilon$ and hence the resulting multi-parameter iterative method converges for any starting vector and right hand side.

A somewhat stronger result can be developed using the following theorem:

Theorem 1. (*Cayley-Hamilton*) Let $P(\lambda) = \det(A - \lambda I)$ be the characteristic polynomial of a general $n \times n$ matrix A . Then $P(A) = \mathbf{0}$.

Proof. By the Jordan Decomposition Theorem, $A = N^{-1}JN$ and it is easy to check that

$$P(A) = N^{-1}P(J)N$$

so it suffices to show that $P(J) = \mathbf{0}$. Now J is block diagonal with (Jordan) diagonal blocks J_1, J_2, \dots, J_k . The matrix $P(J)$ is also block diagonal with diagonal blocks $P(J_1), P(J_2), \dots, P(J_k)$. Now if λ_i is on the diagonal of J_i and if J_i has dimension k_i then λ_i is a root of P of multiplicity at least k_i . Thus, $P(x) = Q(x)(x - \lambda_i)^{k_i}$ for some polynomial $Q(x)$ and

$$P(J_i) = Q(J_i)(J_i - \lambda_i I)^{k_i} = Q(J_i)(N_{k_i})^{k_i}.$$

Here N_{k_i} is the $k_i \times k_i$ matrix given by

$$(N_{k_i})_{l,m} = \begin{cases} 1 & \text{if } m = l + 1, \\ 0 & \text{otherwise.} \end{cases}$$

A simple computation shows $(N_{k_i})^{k_i} = \mathbf{0}$. This means that $P(J_i) = \mathbf{0}$ for all i and hence $P(J) = \mathbf{0}$. \square

Suppose that A is a non-singular $n \times n$ matrix. Then $\lambda = 0$ is not a root of the characteristic polynomial P . Moreover, the leading term of $P(x) = \det(A - xI)$ is $(-1)^n x^n$ thus

$$P(x) = (-1)^n \prod_{i=1}^n (x - \lambda_i)$$

where $\lambda_i, i = 1, \dots, n$ are the eigenvalues of A (roots of P). It follows from the above theorem that

$$\mathbf{0} = P(A) = (-1)^n \prod_{i=1}^n (A - \lambda_i I)$$

and so

$$\mathbf{0} = \prod_{i=1}^n (I - \lambda_i^{-1} A).$$

Thus, the multi-parameter iteration with parameters $\tau_i = \lambda_i^{-1}$, for $i = 1, \dots, n$ produces the exact solution using one sweep through the parameters (since $G_M = \mathbf{0}$ in this case). By the above remark, we see that with the reciprocal of each eigenvalue appearing only once, we only obtain $\rho(G_M) = 0$. In contrast, we get $G_M = \mathbf{0}$ when $\tau_i = \lambda_i^{-1}$ appears m times where m is the multiplicity of λ_i .

The situation is somewhat different in the case when A is symmetric. It is not difficult to prove that the product of two commuting symmetric matrices is symmetric and so A^k is symmetric for any k . Clearly, any linear combination of symmetric matrices is also symmetric so G_M is symmetric. The proposition below follows from an earlier result.

Proposition 1. *Let A be a symmetric $n \times n$ matrix. If $\rho = \rho(G_M)$ is less than one, then the multi-parameter iteration (10.1) converges for any initial vector x_0 and right hand side b . Moreover, each group of iterations reduces the error by a factor ρ in the ℓ^2 norm, i.e.,*

$$\|e_{i+1}\|_{\ell^2} \leq \rho \|e_i\|_{\ell^2}.$$

Remark 4. *In the case when A is symmetric and $\rho(G_M) = 0$, then*

$$\|e_1\|_{\ell^2} = 0,$$

i.e., we solve the iteration in one multi-step. As observed above, we can make $\rho(G_M)$ zero by using $\tau_i = 1/\lambda_i$ and varying over all distinct eigenvalues. Thus, in the case of symmetric A , it is not necessary to repeat the terms corresponding to eigenvalues of multiplicity greater than one to get $G_M = \mathbf{0}$.

Remark 5. *When A is symmetric and positive definite then G_M is also self adjoint with respect to the A -inner product. In this case, we also have*

$$\|e_{i+1}\|_A \leq \rho(G_M) \|e_i\|_A.$$

The above discussion was meant to provide some understanding on how a multi-parameter iteration can lead to a direct solution procedure in theory. However, use of the multi-parameter iteration as a direct solver is not practical in applications because:

- The matrices A that are of interest have a large number of distinct eigenvalues and so the multi-parameter iteration as a direct solve would require M to be large.
- The problem of computing the eigenvalues of A is difficult and requires much more computational time than the direct solution of $Ax = b$.

We shall continue investigating the multi-level method as an iterative method in the case when A is symmetric and positive definite. Our goal will be to use several parameters to develop a faster iterative method (but not a direct solver). We shall assume that the smallest and largest eigenvalues (λ_1 and λ_n) are given.

We are left with the problem of choosing the iteration parameters.¹ In general, we want to minimize $\rho(G_M)$ over all possible choices of parameters.

This problem is a little too specific to solve in that its general solution would depend on knowing the entire spectrum of A . This is more information than we have in applications and much too difficult of a problem to solve in general. Instead, we will find parameters which minimize

$$(10.5) \quad \max_{\lambda \in [\lambda_1, \lambda_n]} \left| \left(\prod_{j=1}^M (1 - \tilde{\tau}_j \lambda) \right) \right|$$

over all real parameters $\{\tilde{\tau}_1, \dots, \tilde{\tau}_M\}$. The max using the optimal parameters (denoted by $\{\tau_1, \dots, \tau_M\}$) will be denoted by $\rho_{opt,M}$. This will be good enough, in general, since

$$\rho(G_M) = \max_{\lambda \in \sigma(A)} \left(\prod_{j=1}^M (1 - \tau_j \lambda) \right) \leq \rho_{opt,M}.$$

Note that G_M is the unique polynomial in \mathbb{P}^M satisfying $G_M(0) = 1$ with roots $\{\tau_1^{-1}, \tau_2^{-1}, \dots, \tau_M^{-1}\}$.

Without loss of generality, we assume that $\lambda_1 < \lambda_n$ for, if they were equal then

$$A = \lambda_1 I$$

and the solution of $Ax = b$ is trivial.

To solve the above problem, we shall use Chebyshev polynomials. Recall, from Theorem 5.1 of the 609d lecture notes, the scaled Chebyshev polynomial $2^{-M+1}T_M(x)$ was the monic polynomial with minimum absolute value over the interval $[-1, 1]$. We obtain the minimal polynomial G_M on $[\lambda_1, \lambda]$ by using the linear mapping of the interval $[\lambda_1, \lambda_n] \rightarrow [-1, 1]$:

$$x(\lambda) = -1 + \frac{2}{\lambda_n - \lambda_1}(\lambda - \lambda_1)$$

and setting

$$G_M(\lambda) = T_M(x(\lambda)) / T_M(x(0))$$

We shall see in the next lecture that $T_M(x(0)) \neq 0$ and that G_M solves the minimization problem. In addition, we shall provide estimates for $\rho_{opt,M}$.

¹ Note, that we have already solved the problem in the case of $M = 1$ in earlier classes, i.e.,

$$\tau_1 = \frac{2}{\lambda_1 + \lambda_n}.$$

5. fix $\{\tau_1, \dots, \tau_M\}$
 we will have
 (10.5) which is a
 function over λ and
 we could choose
 the max over all
 $\lambda \in [\lambda_1, \lambda_n]$.
 over all choice of τ
 and its corresponding
 λ , there is a
 largest one (λ).
 we may find its
 $\{\tau_1, \dots, \tau_M\}$.
 Under this set of
 $\{\tau_1, \dots, \tau_M\}$.
 $\rho(G_M) \leq \rho_{opt,M}$.
 The reason is
 $\lambda \in \sigma(A)$ discrete
 and hence
 the maximum
 not achieved.
 Try to minimize
 $\rho_{opt,M}$.

Class Notes 10: MULTI-PARAMETER ITERATION(CONTINUED) AND THE STEEPEST DESCENT METHOD.

Math 639d

Due Date: March 21
(updated: October 26, 2020)

We continue the analysis of the multi-parameter iteration in the case when A is symmetric and positive definite and we have bounds on the spectrum, $0 < \lambda_1 \leq \lambda \leq \lambda_n$, for all $\lambda \in \sigma(A)$. Without loss of generality, we assume that $\lambda_n > \lambda_1$ as $A = \lambda_1 I$ when $\lambda_1 = \lambda_n$ and the solution of $Ax = b$ is trivial in that case.

In the last class, we used the Chebyshev polynomials to develop a polynomial for multi-parameter iteration. Specifically, we set

$$(10.1) \quad G_M(\lambda) = \frac{T_M(x(\lambda))}{T_M(x(0))}$$

where

$$(10.2) \quad x(\lambda) = -1 + \frac{2}{\lambda_n - \lambda_1}(\lambda - \lambda_1).$$

We note that T_M has $M+1$ extreme points on $[-1, 1]$ where $T_M(y_j) = \pm 1$ with $-1 = y_1 < y_2 < \dots < y_{M+1} = 1$ with oscillating signs. It follows that T_M has M distinct zeros in $(-1, 1)$ and by Rolle's Theorem, T'_M has $M-1$ zeros in $(-1, 1)$. Thus, $T_M(y)$ is an increasing function for $y \geq 1$, i.e., $T_M(y) > T_M(1) = 1$ for $y > 1$. A simple computation shows that

$$(10.3) \quad x(0) = -1 - \frac{2}{\lambda_n - \lambda_1} \lambda_1 = -\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1} =: -\gamma.$$

Note that $\gamma > 1$ and since $T_M(y)$ is either an even or odd function of y ,

$$|T_M(x(0))| = T_M(\gamma) > 1,$$

i.e., the denominator in (10.1) is non-zero. As $G_M(0) = 1$,

$$G_M(y) = \prod_{j=1}^M \left(1 - \tau_j y\right)$$

with $\tau_j = r_j^{-1}$ where r_j is the j 'th (distinct) root of $P(y) = T_M(x(y))$. Thus, $G_M(A)$ is the reducer for the M step multi-parameter iteration associated with parameters $\{\tau_1, \tau_2, \dots, \tau_M\}$. The following theorem shows that this is the optimal choice.

$$|1 - \frac{2}{\lambda_n - \lambda_1} \lambda|$$

$$= | \frac{1}{\gamma} - \lambda |$$

\therefore let r_j be the root of

$$T_M(x(x))$$

$$\frac{1}{\tau_j} = r_j$$

$$\therefore \tau_j = r_j^{-1}$$

Theorem 1. Let G_M be as above and set

$$(10.4) \quad \rho := \rho_{opt,M} = \max_{\lambda \in [\lambda_1, \lambda_n]} |G_M(\lambda)|.$$

If Q is any other real polynomial of degree M satisfying $Q(0) = 1$, then

$$\eta = \max_{\lambda \in [\lambda_1, \lambda_n]} |Q(\lambda)| \geq \rho_{opt,M}.$$

Proof. Suppose to the contrary that η is less than ρ . Consider the polynomial $P(\lambda) = G_M(\lambda) - Q(\lambda)$. Now $G_M(\lambda)$ has $M+1$ extreme points in the interval $[\lambda_1, \lambda_n]$ with oscillating signs. As η is less than ρ , the sign of $P(\lambda) = G_M(\lambda) - Q(\lambda)$ is the same as $G_M(\lambda)$ at each of these points. This implies that $P(\lambda)$ has at least M distinct zeroes in the interval (λ_1, λ_n) . It also vanishes at $\lambda = 0$, i.e., P_M has at least $M+1$ distinct zeroes. This implies that P_M is the zero polynomial and is a contradiction to the assumption that η is strictly less than ρ . \square

We are left to bound $\rho(G_M)$. Note that since $|T_M(y)| \leq 1$ on $[-1, 1]$,

$$|G_M(\lambda)| \leq \frac{1}{|T_M(x(0))|} = \frac{1}{T_M(\gamma)}, \quad \text{for all } \lambda \in [\lambda_1, \lambda_n],$$

with γ given by (10.3). We need a more convenient expression for $T_M(\gamma)$ when $\gamma > 1$. Note that the formula $T_M(y) = \cos(M \cos^{-1}(y))$ is not valid for $y > 1$. Instead, we consider the formula

$$(10.5) \quad T_M(\gamma) = \cosh(M \cosh^{-1}(\gamma)).$$

That this is a valid expression for $T_M(\gamma)$ when $\gamma > 1$ is shown by showing that it gives the right polynomials for $M = 0$ and $M = 1$ and satisfies the recurrence relation derived for the Chebyshev polynomials in 639d lecture 5, i.e., one checks that for $y \geq 1$,

$$\begin{aligned} \cosh(0 \cosh^{-1}(y)) &= 1 & (\text{obvious}), \\ \cosh(1 \cosh^{-1}(y)) &= y & (\text{obvious}), \\ \cosh((j+1) \cosh^{-1}(y)) &= 2y \cosh(j \cosh^{-1}(y)) \\ &\quad - \cosh((j-1) \cosh^{-1}(y)). \end{aligned}$$

The derivation of the recurrence relation above is similar to the argument used in showing that the formula in 639d lecture 5 but uses the analogous identities involving hyperbolic functions.

$$\begin{aligned} \cosh(x) &= \frac{e^x + e^{-x}}{2} \\ \sinh(x) &= \frac{e^x - e^{-x}}{2} \end{aligned}$$

Using (10.5) gives for $\theta = \cosh^{-1}(\gamma)$

$$\begin{aligned} T_M(\gamma) &= \cosh(M\theta) = \frac{e^{M\theta} + e^{-M\theta}}{2} \geq \frac{1}{2}e^{M\theta} = \frac{1}{2}(e^\theta)^M \\ &= \frac{1}{2}(\cosh(\theta) + \sinh(\theta))^M = \frac{1}{2}\left(\gamma + \sqrt{\gamma^2 - 1}\right)^M, \end{aligned}$$

where we used the identity

$$\sinh(\theta) = \sqrt{\cosh^2(\theta) - 1}.$$

Let $K = \lambda_n/\lambda_1$ then $\gamma = (K + 1)/(K - 1)$ so

$$\begin{aligned} \gamma + \sqrt{\gamma^2 - 1} &= \frac{K + 1 + 2\sqrt{K}}{K - 1} = \frac{(\sqrt{K} + 1)^2}{(\sqrt{K} + 1)(\sqrt{K} - 1)} \\ &= \frac{\sqrt{K} + 1}{\sqrt{K} - 1}. \end{aligned}$$

Combining the above inequalities gives

$$\rho(G_M) = \frac{1}{T_M(\gamma)} \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^M.$$

We have proved the following theorem:

Theorem 2. (Multi-Parameter) *Let A be a symmetric and positive definite $n \times n$ real matrix with spectral condition number K . Let G_M be the reducer for the M parameter method with parameters $\{\tau_1, \tau_2, \dots, \tau_M\}$. Then,*

$$\rho(G_M) \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^M.$$

Moreover,

$$(10.6) \quad \|G_M\| \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^M$$

where $\|\cdot\|$ is the operator norm induced by either of the norms $\|\cdot\|_{\ell^2}$ or $\|\cdot\|_A$ on \mathbb{R}^n .

Remark 1. *The optimal one parameter iteration has a reduction bound given by*

$$\rho = \frac{K - 1}{K + 1} \approx 1 - \frac{2}{K}.$$

We saw in Homework 5 that this required $O(K)$ iterative steps to obtain a fixed error reduction ϵ . In contrast, the reduction per iteration for the

multi-parameter iteration limits to

$$\rho = \frac{\sqrt{K} - 1}{\sqrt{K} + 1} \approx 1 - \frac{2}{\sqrt{K}}.$$

Thus, only $O(\sqrt{K})$ iterations are required in this case.

We saw that the matrices A_3 and A_5 had condition numbers which were on the order of h^{-2} . Accordingly, a one parameter iteration would require on the order of h^{-2} iterations for a fixed reduction while a multi-parameter iteration would only require on the order of h^{-1} iteration. This convergence acceleration results in a significant reduction in the number of iterations when h is small.

Remark 2. We shall see in subsequent lectures, that the CG method with M iterations obtains at least as small of an error as the multi-parameter iteration. Although, the convergence rate of the CG method depends on K , its implementation does not require knowledge of the largest and smallest eigenvalues of A .

The above theorem carries over to the preconditioned case. Let B be a symmetric and positive definite matrix. We consider the preconditioned equations

$$(10.7) \quad BAx = Bb$$

and apply the multi-parameter method (9.1) to it. The reduction matrix is now of the form

$$(10.8) \quad G_M = \left(\prod_{j=1}^M (I - \tau_j BA) \right).$$

The spectrum of G_M is related to that of BA as in the simpler case, i.e.,

$$\sigma(G_M) = \left\{ \left(\prod_{j=1}^M (I - \tau_j \lambda_i) \right) : \lambda_i \in \sigma(BA) \right\}.$$

The eigenvalues of BA are all positive and can be reordered $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. The multi-parameter optimization proceeds exactly as in the simpler case. We have the following theorem:

Theorem 3. (Multi-Parameter-Preconditioned) Let A and B be symmetric and positive definite $n \times n$ real matrices and assume that $K = \lambda_n/\lambda_1$ with λ_1 and λ_n being the smallest and largest eigenvalue of BA . Let G_M be defined by (10.8). Then the multi-parameter iteration for (10.7) has an

reduction matrix G_M defined by (10.8) and satisfies

$$\rho(G_M) \leq \min \left\{ 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^M, \left(\frac{K - 1}{K + 1} \right)^M \right\}.$$

Moreover,

$$(10.9) \quad \|G_M\| = \left\| \left(\prod_{j=1}^M (I - \tau_j BA) \right) \right\| \leq \min \left\{ 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^M, \left(\frac{K - 1}{K + 1} \right)^M \right\}$$

where $\|\cdot\|$ is the operator norm induced by either of the norms $\|\cdot\|_A$ or $\|\cdot\|_{B^{-1}}$ on \mathbb{R}^n .

Remark 3. We used the fact that G_M is self adjoint in either the A -inner product or the B^{-1} inner product and Corollary 1 of Class Notes 7 to conclude the estimates in (10.9).

Remark 4. If B is a good preconditioner for the system $Ax = b$, then the condition number K of BA is significantly smaller than that of A and far fewer preconditioned iterations are required.

STEEPEST DESCENT:

Suppose that A is a SPD $n \times n$ real matrix and, as usual, we consider iteratively solving $Ax = b$. By now, you should understand that the goal of any iterative method is to drive down (a norm of) the error $e_i = x - x_i$ as rapidly as possible. We consider a method of the following form:

$$x_{i+1} = x_i + \alpha_i p_i.$$

Here $p_i \in \mathbb{R}^n$ is a “search direction” while α_i is a real number which we are free to choose. It is immediate (subtract this equation from $x = x$) that

$$(10.10) \quad e_{i+1} = e_i - \alpha_i p_i.$$

We also introduce the “residual” $r_i = b - Ax_i$. A simple manipulation shows that

$$r_i = Ae_i \quad \text{and} \quad r_{i+1} = r_i - \alpha_i Ap_i.$$

The first method that we shall develop takes p_i to be the residual r_i . The idea is then to try to find the best possible choice for α_i . Ideally, we should choose α_i so that it results in the maximum error reduction, i.e., $\|e_{i+1}\|$ should be as small as possible. For arbitrary norms, this goal is not a viable one. The problem is that we cannot assume that we know e_i at any step of the iteration. Indeed, since we always have x_i available, knowing e_i is tantamount to knowing the solution since $x = x_i + e_i$.

It is instructive to see what happens with the wrong choice of norm. Suppose that we attempt to choose α_i so that $\|e_{i+1}\|_{\ell^2}$ is minimal, i.e.,

$$\|e_{i+1}\|_{\ell^2} = \min_{\alpha \in \mathbb{R}} \|e_i - \alpha r_i\|_{\ell^2}.$$

The above problem can be solved geometrically and its solution is illustrated in Figure 1. Clearly, α_i should be chosen so that the error e_{i+1} is orthogonal to r_i , i.e.,

$$(e_{i+1}, r_i) = 0.$$

A simple algebraic manipulation using the properties of the inner product and (10.10) gives

$$(10.11) \quad (e_i - \alpha_i r_i, r_i) = (e_i, r_i) - \alpha_i (r_i, r_i) = 0 \quad \text{or} \quad \alpha_i = \frac{(e_i, r_i)}{(r_i, r_i)}.$$

Of course, this method is not computable as we do not know e_i during the iteration so the numerator in the definition of α_i in (10.11) is not available.

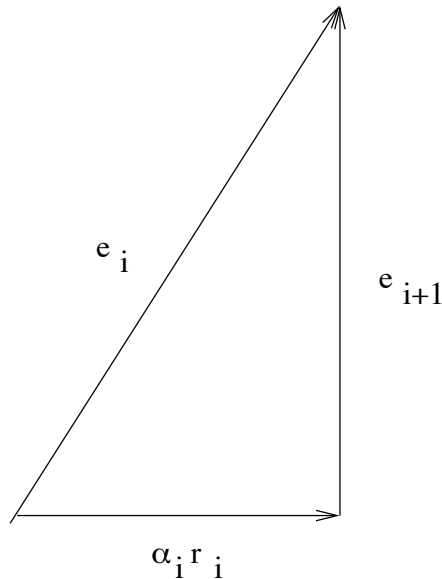


FIGURE 1. Minimal error

We can fix up the above method by introducing a different norm, actually, we introduce a different inner product. Recall, that from earlier classes, inner products not only provide norms but they give rise to a (different) notion of angle. We shall get a computable algorithm by replacing the ℓ^2 inner product above with the A -inner product, i.e., we define

$$(10.12) \quad \|e_{i+1}\|_A = \min_{\alpha \in \mathbb{R}} \|e_i - \alpha r_i\|_A.$$

The solution of this problem is to make e_{i+1} A -orthogonal to r_i , i.e.,

$$(e_{i+1}, r_i)_A = 0.$$

Repeating the above computations (but with the A -inner product) gives

$$(10.13) \quad (e_i - \alpha_i r_i, r_i)_A = 0 \quad \text{or} \quad \alpha_i = \frac{(e_i, r_i)_A}{(r_i, r_i)_A} = \frac{(Ae_i, r_i)}{(Ar_i, r_i)}, \quad \text{i.e.,}$$

$$\alpha_i = \frac{(r_i, r_i)}{(Ar_i, r_i)}.$$

We have now obtained a computable method. Clearly, the residual $r_i = b - Ax_i$ and α_i are computable without using x or e_i . We can easily check that this choice of α_i solves (10.12). Indeed, by A -orthogonality and the Schwarz inequality,

$$(10.14) \quad \|e_{i+1}\|_A^2 = (e_{i+1}, e_{i+1})_A = (e_{i+1}, e_i - \alpha r_i + (\alpha_i - \alpha)r_i)_A$$

$$= (e_{i+1}, e_i - \alpha r_i)_A \leq \|e_{i+1}\|_A \|e_i - \alpha r_i\|_A$$

holds for any $\alpha \in \mathbb{R}$. Clearly if $\|e_{i+1}\|_A = 0$ then (10.12) holds. Otherwise, (10.12) follows by dividing (10.14) by $\|e_{i+1}\|_A$.

The algorithm which we have just derived is known as the steepest descent method and is summarized in the following:

Algorithm 1. (*Steepest Descent*). Let A be a SPD $n \times n$ matrix. Given an initial iterate x_0 , define for $i = 0, 1, \dots$,

$$x_{i+1} = x_i + \alpha_i r_i, \quad r_i = b - Ax_i,$$

and

$$(10.15) \quad \alpha_i = \frac{(r_i, r_i)}{(Ar_i, r_i)}.$$

Proposition 1. Let A be a SPD $n \times n$ matrix and $\{e_i\}$ be the sequence of errors corresponding to the steepest descent algorithm. Then

$$\|e_{i+1}\|_A \leq \left(\frac{K-1}{K+1} \right) \|e_i\|_A$$

where K is the spectral condition number of A .

Proof. Since e_{i+1} is the minimizer

$$\|e_{i+1}\|_A \leq \|(I - \tau A)e_i\|_A$$

for any real τ . Taking $\tau = 2/(\lambda_1 + \lambda_n)$ as in the proposition of Class Notes 7 and applying that proposition completes the proof. \square

Remark 5. Note that λ_1 and λ_n only appear in the analysis. We do not need any eigenvalue estimates for implementation of the steepest descent method.

Remark 6. *As already mentioned, it is not practical to attempt to make the optimal choice with respect to other norms as the error is not explicitly known. Alternatively, at least one application of A can eliminate this drawback, for example, one could design a method which minimized*

$$\|Ae_i\|_{\ell^2}.$$

One could also propose to minimize some other norm, i.e.,

$$\|Ae_i\|_{\ell^\infty}.$$

Although this is feasible, since the ℓ^∞ norm does not come from an inner product, the computation of the parameter α_i ends up being a difficult non-linear problem.

It is interesting to note that the Steepest Descent Method is the first example (in this course) of an iterative method that is not a linear iterative method. Note that e_{i+1} can be theoretically expressed directly from e_i (without knowing x_i or b) since one simply substitutes $r_i = Ae_i$ to compute α_i and uses

$$e_{i+1} = e_i - \alpha_i r_i.$$

Thus, there is a mapping $e_i \rightarrow e_{i+1}$ however it is not given by matrix multiplication. This can be illustrated by considering the 2×2 matrix

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

For either $e_0^1 = (1, 0)^t$ or $e_0^2 = (0, 1)^t$, a direct computation gives $e_1^j = (0, 0)^t$, for $j = 1, 2$ where e_1^j is the error after one step of steepest descent is applied to e_0^j . For example, for $e_0 = e_0^1$, $r_0 = Ar_0 = (1, 0)^t$, $\alpha_0 = 1$ and $e_1 = e_0 - r_0 = (0, 0)^t$. In contrast, for $e_0 = e_0^1 + e_0^2 = (1, 1)^t$, we find

$$\begin{aligned} Ae_0 = r_0 &= \begin{pmatrix} 1 \\ 2 \end{pmatrix}, & Ar_0 &= \begin{pmatrix} 1 \\ 4 \end{pmatrix}, & \alpha_0 &= \frac{5}{9} \\ e_1 &= \begin{pmatrix} 4/9 \\ -1/9 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = e_1^1 + e_1^2. \end{aligned}$$

Class Notes 11: THE CONJUGATE GRADIENT METHOD

Math 639d

Due Date: March 28

(updated: October 30, 2020)

In this set of notes, A will be an symmetric and positive definite real $n \times n$ matrix. Everything that is to be done also works for positive definite Hermitian complex $n \times n$ matrices.

The steepest descent method makes the error e_{i+1} A -orthogonal r_i . Unfortunately, if this step results in little change, then r_{i+1} lies in almost the same direction as r_i so the step to e_{i+2} is not very effective either since e_{i+1} is already A -orthogonal to r_i and almost A -orthogonal to r_{i+1} . This is a shortcoming of the steepest descent method.

The fix is simple. We generalize the algorithm and let p_i be the direction which we use to compute e_{i+1} . We make e_{i+1} A -orthogonal to p_i . The idea is to preserve this orthogonality when going to e_{i+2} . Since e_{i+1} is already A -orthogonal to p_i , e_{i+2} will remain A -orthogonal to p_i only if our new search direction p_{i+1} is also A -orthogonal to p_i . Thus, instead of using r_{i+1} as our next search direction, we use the component of r_{i+1} which is A -orthogonal to p_i , i.e.,

$$p_{i+1} = r_{i+1} - \beta_i p_i$$

where β_i is chosen so that

$$(p_{i+1}, p_i)_A = 0.$$

A simple computation gives

$$\beta_i = \frac{(r_{i+1}, p_i)_A}{(p_i, p_i)_A}.$$

We continue by making e_{i+2} A -orthogonal to p_{i+1} , i.e.,

$$x_{i+2} = x_{i+1} + \alpha_{i+1} p_{i+1}$$

with α_{i+1} satisfying

$$(e_{i+2} - \alpha_{i+1} p_{i+1}, p_{i+1})_A = 0 \quad \text{or} \quad \alpha_{i+1} = \frac{(r_{i+2}, p_{i+1})}{(A p_{i+1}, p_{i+1})}.$$

As both e_{i+1} and p_{i+1} are A -orthogonal to p_i and $e_{i+2} = e_{i+1} - \alpha_{i+1} p_{i+1}$, e_{i+2} is A -orthogonal to both p_i and p_{i+1} . The above discussion leads to the following algorithm.

Algorithm 1. (*Conjugate Gradient*). Let A be a SPD $n \times n$ real matrix and $x_0 \in \mathbb{R}^n$ (the initial iterate) and $b \in \mathbb{R}^n$ (the right hand side) be given. Start by setting $p_0 = r_0 = b - Ax_0$. Then for $i = 0, 1, \dots$, define

$$(11.1) \quad \begin{aligned} x_{i+1} &= x_i + \alpha_i p_i, & \text{where } \alpha_i &= \frac{(r_i, p_i)}{(Ap_i, p_i)}, \\ r_{i+1} &= r_i - \alpha_i Ap_i, \\ p_{i+1} &= r_{i+1} - \beta_i p_i, & \text{where } \beta_i &= \frac{(r_{i+1}, Ap_i)}{(Ap_i, p_i)}. \end{aligned}$$

Remarks:

- The above algorithm works for complex valued Hermitian positive definite matrices A provided that one uses the sesquilinear inner product $(u, v) := u \cdot \bar{v}$ with \bar{v} denoting the complex conjugate of v .
- Notice that we have moved the matrix-vector evaluation in the above inner products so that it is clear that only one matrix-vector evaluation, namely Ap_i , is required per iterative step after start up.
- The first step in the conjugate gradient method coincides with the steepest descent algorithm. It follows that the CG algorithm is not a linear iterative method.

We illustrate pseudo code for the conjugate gradient algorithm below: We have implicitly assumed that $A(X)$ is a routine which returns the result of A applied to X and $IP(\cdot, \cdot)$ returns the result of the inner product. Here k is the number of iterations, X is x_0 on input and X is x_k on return.

FUNCTION CG(X, B, A, k, IP)

$R = P = B - A(X);$

FOR $j = 1, 2, \dots, k$ *DO* {

$AP = A(P); al = IP(R, P)/IP(P, AP);$

$X = X + al * P; R = R - al * AP;$

$be = IP(R, AP)/IP(P, AP);$

$P = R - be * P;$

}

RETURN

END

The above code is somewhat terse but was included to illustrate the fact that one can implement CG with exactly 3 extra vectors, R, P , and AP . An actual code would include extra checks for consistency and convergence. For example, a tolerance might be passed and the residual might be tested against it causing the routine to return when the desired tolerance was achieved. Also, for consistency, one would check to see that $(Ap, p) > 0$

for when this is negative or zero, it is a sure sign that the matrix is either not good (not SPD) or that you have iterated to convergence (we shall see below that the solution has been reached when $(Ap, p) = 0$).

Below, HW indicates that the result is part of the next homework assignment.

To analyze the conjugate gradient method, we start by introducing the so-called Krylov subspace,

$$K_m := K_m(A, r_0) := \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\}.$$

It is clear that the dimension of K_m is at most m . Sometimes, K_m has dimension less than m , for example, if r_0 was an eigenvector, then $\dim(K_m)=1$, for $m = 1, 2, \dots$, since $A^j r_0 = \lambda^j r_0$.

A simple argument by mathematical induction (HW), shows that the search direction p_i is in K_{i+1} , for $i = 0, 1, \dots$. It follows that $x_i = x_0 + \theta$ for some $\theta \in K_i$ and $e_i = e_0 - \theta$ (for the same θ).

We note that the dimension of the Krylov space keeps growing until $A^{l_0}r_0 \in K_{l_0}$ in which case $K_l = K_{l+1} = K_{l+2} \dots$ (HW). Let l_0 be the minimal such value, i.e., $A^{l_0}r_0 \in K_{l_0}$ but $A^{l_0-1}r_0 \notin K_{l_0-1}$. Then there are coefficients, c_0, \dots, c_{l_0-1} satisfying

$$(11.2) \quad c_0 r_0 + c_1 A r_0 + \dots c_{l_0-1} A^{l_0-1} r_0 = A^{l_0} r_0.$$

If $c_0 = 0$, multiplying (11.2) by A^{-1} would imply that $A^{l_0-1}r_0 \in K_{l_0-1}$. As we have assumed that this is not the case, we must have $c_0 \neq 0$ and so

$$e_0 = A^{-1}r_0 = c_0^{-1}(A^{l_0-1}r_0 - c_1 r_0 - c_2 A r_0 \dots - c_{l_0-1} A^{l_0-2} r_0) \in K_{l_0}.$$

Similar arguments show that l_0 is, in fact, the smallest index such that $e_0 \in K_{l_0}$ (HW).

It is natural to consider the best approximation with respect to the A -norm to x over vectors of the form $\tilde{x}_i = x_0 + \theta$ for $\theta \in K_i$ i.e.,

$$(11.3) \quad \|x - \tilde{x}_i\|_A := \|x - (x_0 + \theta)\|_A := \min_{\zeta \in K_i} \|x - (x_0 + \zeta)\|_A.$$

This can be rewritten

$$(11.4) \quad \|\tilde{e}_i\|_A = \min_{\zeta \in K_i} \|e_0 - \zeta\|_A$$

with $\tilde{e}_i = x - \tilde{x}_i$. When $i = l_0$, we may take $\zeta = e_0 \in K_{l_0}$ to conclude that $\tilde{e}_{l_0} = \mathbf{0}$ and hence $x_{l_0} = x$, i.e., we have the solution at the l_0 'th step.

The minimization problem with respect to a finite dimensional space associated with a norm which comes from an inner product can always be

reduced to a matrix problem. We illustrate this for the above minimization problem in the following lemma:

Lemma 1. *There is a unique solution $\theta \in K_i$ solving the minimization problem (11.3). It is characterized as the unique function of this form satisfying*

$$(11.5) \quad (x - x_0 - \theta, \zeta)_A = (x - \tilde{x}_i, \zeta)_A = 0 \quad \text{for all } \zeta \in K_i.$$

Proof. We first show that there is a unique function $\theta \in K_i$ for which (11.5) holds. Given a basis $\{v_1, \dots, v_l\}$ for the Krylov space K_i , we expand

$$\theta = \sum_{j=1}^l c_j v_j.$$

The condition (11.5) is equivalent to

$$\left(x - x_0 - \sum_{j=1}^l c_j v_j, v_m \right)_A = 0 \quad \text{for } m = 1, \dots, l$$

or, using the bilinearity of $(\cdot, \cdot)_A$,

$$\sum_{j=1}^l c_j (v_j, v_m)_A = (b - Ax_0, v_m) \quad \text{for } m = 1, \dots, l.$$

This is the same as the matrix problem $Nc = F$ with $N \in \mathbb{R}^{l \times l}$ and $F, c \in \mathbb{R}^l$ and

$$N_{m,j} = (v_j, v_m)_A \quad \text{and} \quad F_m = (b - Ax_0, v_m), \quad j, m = 1, \dots, l.$$

This problem has a unique solution if N is nonsingular. To check this, we let $d \in \mathbb{R}^l$ be arbitrary and compute

$$\begin{aligned} (Nd, d) &= \sum_{j,k=1}^l (N_{j,k} d_k) d_j = \sum_{j,k=1}^l (v_k, v_j)_A d_k d_j \\ &= \left(\sum_{k=1}^l d_k v_k, \sum_{j=1}^l d_j v_j \right)_A = (w, w)_A \end{aligned}$$

where $w = \sum_{j=1}^l d_j v_j$. Since $\{v_1, \dots, v_l\}$ is a basis, w is nonzero whenever d is nonzero. It follows from the definiteness property of the inner product that $0 \neq (w, w)_A = (Nd, d)$ whenever $d \neq \mathbf{0}$. Thus, $\mathbf{0}$ is the only vector for which $Nd = \mathbf{0}$, i.e., the matrix N is nonsingular and there is a unique $\theta \in K_i$ satisfying (11.5).

We next show that the unique solution θ of (11.5) is a solution to the minimization problem. Indeed, if ζ is in K_i then

$$\begin{aligned}\|x - \tilde{x}_i\|_A^2 &= (x - \tilde{x}_i, x - [(x_0 + \zeta) + (\theta - \zeta)])_A \\ &= (x - \tilde{x}_i, x - (x_0 + \zeta))_A \leq \|x - \tilde{x}_i\|_A \|x - (x_0 + \zeta)\|_A\end{aligned}$$

where we used the Cauchy-Schwarz inequality for the inequality above. It follows that

$$\|x - \tilde{x}_i\|_A \leq \|x - (x_0 + \zeta)\|_A.$$

As this holds for all ζ in K_i , θ is a minimizer.

The proof will be complete once we show that anything which is a minimizer satisfies (11.5) and hence (11.5) gives rise to the unique minimizer. Suppose that θ solves the minimization problem and $\tilde{x}_i = x_0 + \theta$. Given $\zeta \in K_i$, we consider for $t \in \mathbb{R}$,

$$f(t) = \|x - \tilde{x}_i + t\zeta\|_A^2 = (x - \tilde{x}_i + t\zeta, x - \tilde{x}_i + t\zeta)_A.$$

By using bilinearity of the inner product, we see that

$$f(t) = (x - \tilde{x}_i, x - \tilde{x}_i)_A + 2t(x - \tilde{x}_i, \zeta)_A + t^2(\zeta, \zeta)_A.$$

Since \tilde{x}_i is the minimizer, $f'(0) = 2(x - \tilde{x}_i, \zeta)_A = 0$, i.e., (11.5) holds. \square

The proof of the above lemma shows one way of solving the minimization problem, i.e., constructing the matrix N , solving $Nc = F$ and setting

$$\tilde{x}_i = x_0 + \sum_{j=0}^l c_j v_j.$$

The following theorem shows that the conjugate gradient algorithm provides a much more elegant solution.

Theorem 1. (*CG-equivalence*) *Let l_0 be as above. Then, for $i = 1, 2, \dots, l_0$, x_i defined by the conjugate gradient (CG) algorithm coincides with the solution \tilde{x}_i of the minimization problem (11.3).*

The above theorem shows that the conjugate gradient algorithm provides a very efficient implementation of the Krylov minimization problem (11.3). As discussed above, at least mathematically, we get the exact solution on the l_0 'th step ($x_{l_0} = x$). At this point the CG algorithm breaks down as $r_{l_0} = p_{l_0} = 0$ and so the denominators are zero.

Historically, when the conjugate gradient method was discovered, it was proposed as a direct solver. Since, $K_l \subseteq \mathbb{R}^n$, l_0 can at most be n . The above theorem shows that we get convergence in at most l_0 iterations. However, it was found that, because of round-off errors, implementations of the CG

method often failed to converge in n iterations. Nevertheless, CG is very effective when used as an iterative method on problems with reasonable condition numbers. The following theorem gives a bound for its convergence rate.

Theorem 2. (*CG-error*) Let A be a SPD $n \times n$ matrix and e_i be the sequence of errors generated by the conjugate gradient algorithm. Then,

$$\|e_i\|_A \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^i \|e_0\|_A.$$

Here K is the spectral condition number of A .

Proof. Let

$$G_i = \prod_{j=1}^i (I - \tau_j A)$$

be the reduction matrix for the optimal multi-parameter iteration with i steps. Then

$$G_i e_0 = \prod_{j=1}^i (I - \tau_j A) e_0 = e_0 - Q_{i-1}(A) A e_0 = e_0 - Q_{i-1}(A) r_0$$

for some polynomial Q_{i-1} of degree $i - 1$, i.e.,

$$G_i e_0 = x - x_0 - \zeta \quad \text{for some } \zeta \in K_i.$$

It follows from

Lemma 2. 1, Theorem 2 of Class Notes 10 and Theorem 1 that

$$\|e_i\|_A \leq \|G_i e_0\|_A \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^i \|e_0\|_A.$$

□

The above theorem guarantees an accelerated convergence rate ($O(\sqrt{K})$ iterations for a fixed reduction) when compared to Gauss-Seidel or Jacobi iterations. In addition, the implementation of the CG method does require bounds on the spectrum.

Proof of Theorem (CG-equivalence). The proof is by induction. We shall prove that for $i = 1, \dots, l_0$,

(I.1) $\{p_0, p_1, \dots, p_{i-1}\}$ forms an A -orthogonal basis for K_i .

(I.2) $(e_i, \theta)_A = 0$ for all $\theta \in K_i$.

By definition, $p_0 = r_0$ and forms a basis for K_1 . Moreover, $(e_1, r_0)_A = 0$ by the definition of α_0 so (I.1) and (I.2) hold for $i = 1$.

We inductively assume that (I.1) and (I.2) hold for $i = k$ with $k < l_0$. By the definition of β_{k-1} , p_k is A -orthogonal to p_{k-1} . For $j < k - 1$,

$$(11.6) \quad (p_k, p_j)_A = (r_k - \beta_{k-1}p_{k-1}, p_j)_A = (e_k, Ap_j)_A - \beta_{k-1}(p_{k-1}, p_j)_A.$$

Of the two terms on the right, the first vanishes by Assumption (I.2) with k since $Ap_j \in K_k$ while the second vanishes by assumption (I.1) with k . Thus, p_k satisfies the desired orthogonality properties.

The validity of (I.1) for $k + 1$ will follow if we show that $p_k \neq \mathbf{0}$. If $p_k = \mathbf{0}$ then $r_k = \beta_{k-1}p_{k-1} \in K_k$ and by (I.2) at k ,

$$0 = (e_k, p_{k-1})_A = (r_k, p_{k-1}) = \beta_{k-1}(p_{k-1}, p_{k-1}).$$

This implies that $r_k = \mathbf{0}$ and hence $e_k = \mathbf{0}$. This means that $e_0 \in K_k$ contradicting the assumption that $k < l_0$.

We next verify (I.2) at $k+1$. As observed earlier, p_k and α_k are constructed so that e_{k+1} is A -orthogonal to both p_k and p_{k-1} . To complete the proof, we need only check its orthogonality to p_j for $j < k - 1$. In this case,

$$(11.7) \quad (e_{k+1}, p_j)_A = (e_k - \alpha_k p_k, p_j)_A = (e_k, p_j)_A - \alpha_k (p_k, p_j)_A.$$

The first term is zero by the induction assumption (I.2) while the second vanishes by (I.1) for $k + 1$ (which we proved above).

Thus, (I.1) and (I.2) hold for $i = 1, 2, \dots, l_0$. We already observed that $e_i = e_0 + \theta$ for some $\theta \in K_i$. Thus, (I.2) and Lemma 1 implies that $x_i = \tilde{x}_i$. \square

Class Notes 12: PRECONDITIONED CONJUGATE GRADIENT, CG-EIGENVALUE AND MINRES.

Math 639d

Due Date: April 4
(updated: October 30, 2020)

Preconditioned conjugate gradient iteration. In the development and analysis of the conjugate gradient algorithm, we see the interaction of the matrix A and the inner product. In fact, if you look carefully, you see that the entire development goes through replacing the ℓ^2 inner product (\cdot, \cdot) by any other inner product provided that the matrix A is positive definite and self adjoint with respect to new inner product. This observation immediately leads to a preconditioned conjugate gradient algorithm. Specifically, we assume that we are given two SPD matrices A and B (with B a preconditioner). As usual, we consider the preconditioned system

$$(12.1) \quad BAx = Bb$$

and use the B^{-1} -inner product as our base inner product (which replaces the ℓ^2 inner product). We now apply CG to the preconditioned system (12.1) with the B^{-1} -inner product to obtain:

Algorithm 1. (*Preconditioned Conjugate Gradient, Version 1*). Let A and B be SPD $n \times n$ matrices and $x_0 \in \mathbb{R}^n$ (the initial iterate) and $b \in \mathbb{R}^n$ (the right hand side) be given. Start by setting $p_0 = r_0 = Bb - BAx_0$. Then for $i = 0, 1, \dots$, define

$$(12.2) \quad \begin{aligned} x_{i+1} &= x_i + \alpha_i p_i, & \text{where } \alpha_i &= \frac{(r_i, p_i)_{B^{-1}}}{(BAp_i, p_i)_{B^{-1}}} \\ r_{i+1} &= r_i - \alpha_i BAp_i, \\ p_{i+1} &= r_{i+1} - \beta_i p_i, & \text{where } \beta_i &= \frac{(r_{i+1}, BAp_i)_{B^{-1}}}{(BAp_i, p_i)_{B^{-1}}}. \end{aligned}$$

The above algorithm is not completely satisfactory. The reason being is that the preconditioner may be defined by a fairly complicated process and so its inverse may not be computationally available. Although, the denominator and the numerator for β_i poses no trouble as B^{-1} and B cancel, the numerator for α_i could be troublesome. As in the preconditioned steepest descent algorithm, the troublesome term can be dealt with by introducing an auxiliary variable, \tilde{r}_i defined $\tilde{r}_i = b - Ax_i$. Then $r_i = B\tilde{r}_i$ and the algorithm becomes:

Algorithm 2. (*Preconditioned Conjugate Gradient*). Let A and B be SPD $n \times n$ matrices and $x_0 \in \mathbb{R}^n$ (the initial iterate) and $b \in \mathbb{R}^n$ (the right hand

side) be given. Start by setting $\tilde{r}_0 = b - Ax_0$ and $p_0 = r_0 = B\tilde{r}_0$. Then for $i = 0, 1, \dots$, define

$$(12.3) \quad \begin{aligned} x_{i+1} &= x_i + \alpha_i p_i, & \text{where } \alpha_i &= \frac{(\tilde{r}_i, p_i)}{(Ap_i, p_i)}, \\ \tilde{r}_{i+1} &= \tilde{r}_i - \alpha_i Ap_i, & r_{i+1} &= B\tilde{r}_{i+1}, \\ p_{i+1} &= r_{i+1} - \beta_i p_i, & \text{where } \beta_i &= \frac{(r_{i+1}, Ap_i)}{(Ap_i, p_i)}. \end{aligned}$$

We note that in the above algorithm, there is exactly one evaluation of B and one evaluation of A per iterative step after start up. The original CG method minimized the error in the A -norm which was defined using the base inner product, (\cdot, \cdot) , i.e. $(\cdot, \cdot)_A = (A\cdot, \cdot)$. The preconditioned conjugate gradient method minimizes the error in the operator inner product defined in terms of the base inner product $(\cdot, \cdot)_{B^{-1}}$, i.e.,

$$((v, w))_{BA} = (BAv, w)_{B^{-1}} = (Av, w) = (v, w)_A.$$

In fact, the use of the B^{-1} -inner product was motivated by this observation, i.e., we get the best approximation in the Krylov space in the usual A -inner product (See the remark below).

The analysis of the preconditioned version is identical to the original CG algorithm except the minimization is over the preconditioned Krylov space,

$$K_i = K_i(BA, r_0).$$

We again prove (I.1) and (I.2) of Class Notes 11 by induction. The critical property which makes everything work is that BA is self adjoint with respect to the A inner product so (11.6) is replaced by

$$(p_k, p_j)_A = (r_k - \beta_{k-1}p_{k-1}, p_j)_A = (e_k, BA p_j)_A - \beta_{k-1}(p_{k-1}, p_j)_A.$$

This shows that for the preconditioned algorithm,

$$\|x - x_i\|_A = \min_{\zeta \in K_i(BA, r_0)} \|x - (x_0 - \zeta)\|_A.$$

Applying the multi-parameter preconditioned theorem (Theorem 3 of Class Notes 10) and an argument similar to that used in the proof of Theorem 2 of Class Notes 11 then gives the following theorem.

Theorem 1. (PCG-error) *Let A and B be SPD $n \times n$ matrices and e_i be the sequence of errors generated by the preconditioned conjugate gradient algorithm. Then,*

$$\|e_i\|_A \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^i \|e_0\|_A.$$

Here K is the spectral condition number of BA .

Remark 1. *Alternative preconditioned conjugate gradient algorithms can be defined. For example, one can use the A -inner product as the base inner product. The Krylov space remains the same, i.e., $K_l(BA, r_0)$. The only difference is that the error minimization is with respect to the ABA -inner product, i.e.*

$$\|x - x_i\|_{ABA} = \min_{\zeta \in K_i(BA, r_0)} \|x - (x_0 - \zeta)\|_{ABA}.$$

This results in the corresponding error bound

$$\|e_i\|_{ABA} \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^i \|e_0\|_{ABA}.$$

The CG-Eigenvalue approximation:

We next investigate the use of CG to get bounds on the spectrum for A . As before, A is a SPD $n \times n$ real matrix. We start by considering the Rayleigh-Ritz eigenvalue approximation. For a symmetric matrix A ,

$$(12.4) \quad \lambda_n = \max_{x \in \mathbb{R}^n, x \neq 0} \frac{(Ax, x)}{(x, x)} \quad \text{and} \quad \lambda_1 = \min_{x \in \mathbb{R}^n, x \neq 0} \frac{(Ax, x)}{(x, x)}.$$

Given a subspace $K \subseteq \mathbb{R}^n$, we can approximate λ_n and λ_1 by replacing \mathbb{R}^n above by K , i.e.,

$$(12.5) \quad \tilde{\lambda}_n = \max_{x \in K, x \neq 0} \frac{(Ax, x)}{(x, x)} \quad \text{and} \quad \tilde{\lambda}_1 = \min_{x \in K, x \neq 0} \frac{(Ax, x)}{(x, x)}.$$

This is the so-called Rayleigh-Ritz approximation.

We can get eigenvalue approximations from our CG algorithm by computing $\tilde{\lambda}_n$ and $\tilde{\lambda}_1$ satisfying (12.5) using $K = K_m$ (the m 'th Krylov space). The numbers $\tilde{\lambda}_n$ and $\tilde{\lambda}_1$ are, respectively, the largest and smallest eigenvalues of the generalized eigenvalue problem: Find pairs $\{\lambda, \phi\}$ with $\phi \in K_m$ and $\lambda \in \mathbb{R}$ satisfying

$$(12.6) \quad (A\phi, \theta) = \lambda(\phi, \theta), \quad \text{for all } \theta \in K_m.$$

It is always possible to solve eigenvalue problems of the form of (12.6) by introducing a basis $\{v_1, \dots, v_l\}$. Here l denotes the dimension of K_m which may be less than m . The obvious basis is $r_0, Ar_0, \dots, A^{l-1}r_0$ but this turns out not to be the most computationally effective. Given a basis $\{v_1, v_2, \dots, v_{l-1}\}$, we then solve the (generalized) matrix eigenvalue problem

$$(12.7) \quad Mx = \lambda Nx \quad \text{for } x \in \mathbb{R}^l.$$

Here

$$M_{i,j} = (Av_i, v_j) \quad \text{and} \quad N_{i,j} = (v_i, v_j), \quad i, j = 1, \dots, l.$$

It is easy to see that (ϕ, λ) solves (12.6) if and only if (x, λ) solves (12.7). Here x is the vector of coefficients in the expansion of ϕ in the basis, i.e.,

$$\phi = \sum_{i=1}^l x_i v_i.$$

Note that both M and N are SPD $l \times l$ matrices. Although there are routines available for solving this problem, we can use some of the properties of the CG algorithm to develop a much more efficient approach.

Specifically, we use $\{r_0, r_1, \dots, r_{l-1}\}$ for our basis. Note that by (I.1) in the proof of Theorem (CG-equivalence), $\{p_0, p_1, \dots, p_{l-1}\}$ is a basis for K_l and so it follows from $p_i = r_i - \beta_{i-1}p_{i-1}$ that $\{r_0, r_1, \dots, r_{l-1}\}$ is also a basis for K_l . We first note that if $j > i$, (I.2) (again, the proof of Theorem (CG-equivalence)) implies

$$(r_j, r_i) = (e_j, r_i)_A = 0$$

so the N matrix is diagonal. Its diagonal entries are given by

$$N_{i,i} = (r_i, r_i) = (r_i, p_i) + \beta_{i-1}(e_i, p_{i-1})_A = (r_i, p_i).$$

Note that the quantity (r_i, p_i) appears as the numerator in α_i .

We now consider the matrix M . First we observe that it is tridiagonal since if $j > i + 1$, (I.2) implies

$$(Ar_j, r_i) = (e_j, Ar_i)_A = 0.$$

We next compute its diagonal entries. Clearly, $M_{1,1} = (Ap_0, p_0)$. This is the denominator in α_0 . In addition, $r_i = p_i + \beta_{i-1}p_{i-1}$ is an A -orthogonal decomposition. The Pythagorean Theorem holds for arbitrary inner products so

$$\|r_i\|_A^2 = \|p_i\|_A^2 + \beta_{i-1}^2 \|p_{i-1}\|_A^2.$$

This can be rewritten

$$M_{i,i} = (Ar_i, r_i) = (Ap_i, p_i) + \beta_{i-1}^2 (Ap_{i-1}, p_{i-1}).$$

Thus, for $i > 1$, the value of $M_{i,i}$ can be computed from β_{i-1} and the denominators for α_i and α_{i-1} . Finally, applying (I.1) gives

$$\begin{aligned} M_{j,j-1} &= (Ar_j, r_{j-1}) = (p_j, r_{j-1})_A + \beta_{j-1}(p_{j-1}, r_{j-1})_A \\ &= \beta_{j-1}(p_{j-1}, r_{j-1})_A = \beta_{j-1}(Ap_{j-1}, p_{j-1}). \end{aligned}$$

We see the denominator of α_{i-1} appearing here as well.

Finally, if x satisfies (12.7) with eigenvalue λ if and only if $y = N^{1/2}x$ satisfies

$$\widetilde{M}y = \lambda y \quad \text{where } \widetilde{M} = N^{-1/2}MN^{-1/2}.$$

This is a standard eigenvalue problem involving a symmetric tridiagonal matrix. The nonzero entries of M and N can be gathered during the CG iteration and the matrix \widetilde{M} and its eigenvalues can be computed as part of a post processing step. Efficient procedures for computing the eigenvalues for symmetric tridiagonal matrices are available in standard software libraries such as LAPACK, see www.netlib.org/lapack/.

This technique goes over to the preconditioned case as well and produces estimates for the condition number for the preconditioned system. These estimates are useful in that they provide information on the quality of the preconditioner. Specifically, a small condition number indicates a well designed preconditioner.

CG with outlying eigenvalues:

Another interesting property of the conjugate gradient method is that it performs well when there are a few outlying eigenvalues with most of the spectrum restricted to a small interval. We illustrate this by a simple example where there is only one outlying eigenvalue. We consider a SPD matrix A with eigenvalues $0 < \lambda_0 \ll \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. Here we assume that $K_1 = \lambda_n/\lambda_1$ is much smaller than $K_0 = \lambda_n/\lambda_0$. Applying the convergence theorem for CG gives

$$(12.8) \quad \|e_i\|_A \leq \left(\frac{\sqrt{K_0} - 1}{\sqrt{K_0} + 1} \right)^i \|e_0\|_A.$$

This estimate assumes nothing about the distribution of the spectrum of A .

Alternatively, as conjugate gradient is the minimizer, we can bound the error by approximating it by any element in the Krylov space. In this case, we use

$$\|e_i\|_A \leq \left\| (I - \lambda_0^{-1}A) \prod_{j=1}^{i-1} (I - \tau_j A) e_0 \right\|_A.$$

Here the parameters $\tau_1, \dots, \tau_{i-1}$ are chosen using the Chebyshev technique applied to the interval $[\lambda_1, \lambda_n]$. We estimate the operator norm on the right

hand side above by

$$\begin{aligned}
& \left\| (I - \lambda_0^{-1}A) \prod_{j=1}^{i-1} (I - \tau_j A) \right\|_A \\
& \leq \max_{k=1}^n \left\{ (1 - \lambda_0^{-1}\lambda_k) \prod_{j=1}^{i-1} (I - \tau_j \lambda_k) \right\} \\
& \leq \max_{\lambda \in [\lambda_1, \lambda_n]} \left\{ (1 - \lambda_0^{-1}\lambda) \prod_{j=1}^{i-1} (I - \tau_j \lambda) \right\}
\end{aligned}$$

Note that the factor $I - \lambda_0^{-1}\lambda$ produces 0 when $\lambda = \lambda_0$ and hence we can exclude λ_0 from the middle maximum above. Applying the multi-parameter theorem and the bound

$$|1 - \lambda_0^{-1}\lambda| \leq K_0 \quad \text{for } \lambda \in [\lambda_1, \lambda_n]$$

gives

$$(12.9) \quad \|e_i\|_A \leq 2K_0 \left(\frac{\sqrt{K_1} - 1}{\sqrt{K_1} + 1} \right)^{i-1} \|e_0\|_A.$$

Comparing (12.8) and (12.9), we see that although (12.9) has a larger constant ($2K_0$), it produces a much better asymptotic (as i becomes large) reduction assuming that $K_1 < K_0$.

MINRES:

There are many other Krylov based methods around. For example, we consider the case when A is nonsingular and symmetric (but not positive definite). In this case, A^2 is SPD and we consider the minimization problem: $x_i = x_0 + \theta$ for $\theta \in K_i \equiv K_i(A, r_0)$ satisfying

$$(12.10) \quad \|e_i\|_{A^2} = \min_{\zeta \in K_i} \|x - (x_0 + \zeta)\|_{A^2}.$$

Defining the residual, $r(\zeta) = b - A(x_0 + \zeta)$, (12.10) can be rewritten

$$(12.11) \quad \|r_i\|_{\ell^2} = \|r(\theta)\|_{\ell^2} = \min_{\zeta \in K_i} \|r(\zeta)\|_{\ell^2}$$

i.e., x_i is the vector in $x_0 + K_i$ which minimizes the residual over all vectors in $x_0 + K_i$.

This problem could be solved by setting up an $i \times i$ matrix as discussed in the lemma of the previous class but can be much more efficiently solved by the following conjugate gradient like algorithm.

Algorithm 3. (*MINRES*). Let A be a symmetric nonsingular $n \times n$ matrix and $x_0 \in \mathbb{R}^n$ (the initial iterate) and $b \in \mathbb{R}^n$ (the right hand side) be given. Start by setting $p_0 = r_0 = b - Ax_0$. Then for $i = 0, 1, \dots$, define

$$\begin{aligned}
 (12.12) \quad & x_{i+1} = x_i + \alpha_i p_i, & \text{where } \alpha_i &= \frac{(r_i, Ap_i)}{(Ap_i, Ap_i)} \\
 & r_{i+1} = r_i - \alpha_i Ap_i, \\
 & p_{i+1} = r_{i+1} - \beta_i p_i, & \text{where } \beta_i &= \frac{(Ar_{i+1}, Ap_i)}{(Ap_i, Ap_i)}.
 \end{aligned}$$

Exercise 1. The above algorithm requires two evaluations of A per step, namely Ap_i and Ar_{i+1} . By possibly introducing extra vectors, find an implementation which only requires one A evaluation per step after start up.

That the MINRES algorithm solves the minimization problem (12.11) can be seen by repeating the proof of Theorem (CG-equivalence) replacing the A inner product by the A^2 -inner product.

Application: We consider a finite difference approximation to a Helmholtz equation:

$$\begin{aligned}
 -\Delta u(x) - k^2 u(x) &= f(x), \quad \text{in } \Omega, \\
 u(x) &= 0 \quad \text{on } \partial\Omega.
 \end{aligned}$$

This equation models, for example, time harmonic acoustic waves in bounded waveguide (Ω contained in \mathbb{R}^2 or \mathbb{R}^3). The finite difference equations have two contributions, one from the Laplacian and the other from the lower order term (the k^2 -term), i.e. $A = A_1 + A_2$. We considered the finite difference approximation to the Laplacian earlier in this course and saw that it resulted in a symmetric and positive definite matrix A_1 . In contrast, the second term results in $A_2 = -k^2 I$ where I is the identity matrix. Let $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ be the eigenvalues of A_1 . We then have

$$\sigma(A) = \{\lambda_i - k^2 : i = 1, \dots, n\}.$$

When k^2 is greater than λ_0 , A is symmetric but not positive definite. For this problem, A is nonsingular only if k^2 is not an eigenvalue of A_1 . The resulting matrix A , when nonsingular, is a natural candidate for the MINRES algorithm.

We next provide an analysis of the above algorithm. As MINRES produces the solution of the minimization (12.10), its analysis consists of coming up with a good approximation in the Krylov space. Recall that for the analysis of CG, we obtained a good approximation in the Krylov space by using the approximation based on the Chebyshev polynomials. We shall use the Chebyshev polynomials again but on an interval which results from the SPD

operator A^2 . Our assumptions on A imply that there are positive numbers a, b, c, d satisfying

$$\sigma(A) \subset [-a, -b] \cap [c, d].$$

The spectrum of A^2 is contained in the interval $[\lambda_1, \lambda_n]$ where $\lambda_1 = \min(c^2, b^2)$ and $\lambda_n = \max(a^2, d^2)$. After $2m + 1$ iterations,

$$\|e_{2m+1}\|_{A^2} \leq \left\| \prod_{i=1}^m (I - \tau_i A^2) \right\|_{A^2} \|e_0\|_{A^2}$$

where $\{\tau_1, \tau_2, \dots, \tau_m\}$ are the optimal Chebyshev iteration parameters based on the interval $[\lambda_1, \lambda_n]$. Note that λ_1 and λ_n are, respectively, the smallest and largest eigenvalue of A^2 . Applying the multi-parameter theorem gives the following result.

Theorem 2. *Suppose that A is a symmetric and nonsingular real matrix. Then, the errors corresponding to the MINRES iteration satisfy*

$$(12.13) \quad \|e_{2m+1}\|_{A^2} \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^m \|e_0\|_{A^2}.$$

Here K is the spectral condition number of A^2 .

Remark 2. *Alternatively, one could apply the conjugate gradient method directly to the system $A^2x = Ab$. These are the normal equations in this case. The resulting iteration requires 2 matrix vector evaluations per step (so m steps involve more or less the same number of matrix evaluations as $2m$ steps of MINRES). The resulting cg errors satisfy*

$$\|e_m\|_{A^2} \leq 2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^m \|e_0\|_{A^2}$$

which is essentially the same as (12.13). Thus, at least theoretically, MINRES is no better than the application of conjugate gradient to the normal equations. Note that we only used the terms in the Krylov space having even powers of A in the analysis of MINRES. The odd powers might give improved convergence in some examples. However, it is possible to construct problems for which MINRES and the normal equation approach gives rise to exactly the same sequence of iterates, i.e., the result at $2i + 1$ for MINRES equals the result at i for CG-normal.