STAT 309: MATHEMATICAL COMPUTATIONS I
FALL 2019
LECTURE 14

1. Jacobi method

- the simplest splitting is to take M to be the diagonal part of A and $-N$ to be the off-diagonal part — this works as long as the diagonal elements of A is nonzero (but the iterates may not converge)
- if we write $A\mathbf{x} = \mathbf{b}$ in coordinate form,

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1, \dots, n,$$

then

$$a_{ii}x_i = b_i - \sum_{i \neq j} a_{ij}x_j,$$

or

$$x_i = \frac{1}{a_{ii}}\Big[b_i - \sum_{j \neq i} a_{ij}x_j\Big] \tag{1.1}$$

- in other words,

$$M = \begin{bmatrix} a_{11} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}, \quad N = -\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix}$$

- our iteration is therefore

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\Big[b_i - \sum_{j \neq i} a_{ij}x_j^{(k)}\Big],$$

known as the *Jacobi method*

- if we write $A = L + D + U$ where

$$L = \begin{bmatrix} 0 & & & \\ a_{21} & \ddots & & \\ \vdots & & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix}, \quad D = \begin{bmatrix} a_{11} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{bmatrix}$$

the the Jacobi mathod can be written in matrix form as

$$D\mathbf{x}^{(k+1)} = -(L + U)\mathbf{x}^{(k)} + \mathbf{b} \tag{1.2}$$

Date: November 13, 2019, version 1.0. Comments, bug reports: lekheng@galton.uchicago.edu.

- the iteration matrix is

$$M^{-1}N = I - D^{-1}A = -\begin{bmatrix} 0 & \frac{a_{12}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{a_{n1}}{a_{nn}} & \cdots & \frac{a_{n,n-1}}{a_{nn}} & 0 \end{bmatrix} =: B_J$$

- so if

$$\|B_J\|_\infty = \max_{1\le i\le n} \sum_{j\ne i} \left|\frac{a_{ij}}{a_{ii}}\right| < 1,$$

  i.e., if A is *strictly diagonally dominant*, then the iteration converges
- therefore, a sufficient condition for convergence of the Jacobi method is $\|B_J\|_\infty < 1$ where

$$b_{ij} = \begin{cases} -\dfrac{a_{ij}}{a_{ii}} & i \ne j, \\ 0 & i = j \end{cases}$$

- for example, suppose

$$A = \begin{bmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix},$$

  then $\|B_J\|_\infty = \frac{1}{2}$ and so the Jacobi method converges rapidly
- on the other hand, if

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix},$$

  which arises from discretizing the one-dimensional Laplacian, then $\|B_J\|_\infty = 1$
- a more subtle analysis can be used to show convergence in this case, but convergence is slow

2. Gauss–Seidel method

- in the Jacobi method, we compute $x_i^{(k+1)}$ using the elements of $\mathbf{x}^{(k)}$, even though $x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}$ are already known

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}\right]$$

- a general adage in numerical computations is: *use the latest information available*
- the *Gauss–Seidel* method is designed to take advantage of the latest information available about $\mathbf{x}$:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}\right] \tag{2.1}$$

- if we write $A = L + D + U$ where

$$L = \begin{bmatrix} 0 & & & \\ a_{21} & \ddots & & \\ \vdots & & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix}, \quad D = \begin{bmatrix} a_{11} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{bmatrix}$$

  then the Gauss–Seidel iteration can be written in matrix form as

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

  or

$$(D + L)\mathbf{x}^{(k+1)} = -U\mathbf{x}^{(k)} + \mathbf{b} \tag{2.2}$$

  which yields

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

- thus the iteration matrix for the Gauss–Seidel method is

$$B_{GS} = -(D + L)^{-1}U$$

  as opposed to the iteration matrix for the Jacobi method

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

- in some cases (cf. last line of the section on optimal SOR parameter in the next lecture)

$$\rho(B_{GS}) = \rho(B_J)^2$$

  so the Gauss–Seidel method converges twice as fast
- on the other hand, note that Gauss–Seidel is very sequential, i.e., it does not lend itself to parallelism
- note that the matrix forms for Jacobi and Gauss–Seidel (1.2) and (2.2) are only convenient representations useful in mathematical analysis of the methods, one should *never* implement these algorithms in such forms, instead use (1.1) and (2.1)
- we saw earlier that a sufficient condition for convergence of the Jacobi method is $\|B_J\|_\infty < 1$ where

$$b_{ij} = \begin{cases} -\dfrac{a_{ij}}{a_{ii}} & i \neq j, \\ 0 & i = j \end{cases}$$

- since

$$\|B_J\|_\infty = \max_i \sum_{j \neq i} \left| \frac{a_{ij}}{a_{ii}} \right| < 1,$$

  this is equivalent to saying that A is strictly diagonally dominant
- we will see that this is also enough to guarantee the convergence of Gauss–Seidel, i.e., if A is strictly diagonally dominant, then Gauss–Seidel is convergent
- define

$$r_i = \sum_{j \neq i} \left| \frac{a_{ij}}{a_{ii}} \right|, \quad r = \max_i r_i$$

Theorem 1. *If $r < 1$, then $\rho(B_{GS}) < 1$, i.e., the Gauss–Seidel iteration converges if A is strictly diagonally dominant.*

Proof. The proof proceeds using induction on the elements of $\mathbf{e}^{(k)}$. We have

$$(D+L)\mathbf{e}^{(k+1)} = -U\mathbf{e}^{(k)},$$

which can be written as

$$\sum_{j=1}^{i} a_{ij} e_j^{(k+1)} = -\sum_{j=i+1}^{n} a_{ij} e_j^{(k)}, \quad i = 1, \ldots, n.$$

Thus

$$e_i^{(k+1)} = -\sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} e_j^{(k)} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} e_j^{(k+1)}, \quad i = 1, \ldots, n.$$

For $i = 1$, we have

$$|e_1^{(k+1)}| \leq \sum_{j=2}^{n} \left| \frac{a_{ij}}{a_{11}} \right| |e_j^{(k)}| \leq r_1 \|\mathbf{e}^{(k)}\|_\infty \leq r \|\mathbf{e}^{(k)}\|_\infty.$$

Assume that for $p = 1, \ldots, i-1$,

$$|e_p^{(k+1)}| \leq \|\mathbf{e}^{(k)}\|_\infty r_p \leq r \|\mathbf{e}^{(k)}\|_\infty.$$

Then,

$$\begin{aligned}
|e_i^{(k+1)}| &\leq \sum_{j=1}^{i-1} \left| \frac{a_{ij}}{a_{ii}} \right| |e_j^{(k+1)}| + \sum_{j=i+1}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| |e_j^{(k)}| \\
&\leq r \|\mathbf{e}^{(k)}\|_\infty \sum_{j=1}^{i-1} \left| \frac{a_{ij}}{a_{ii}} \right| + \|\mathbf{e}^{(k)}\|_\infty \sum_{j=i+1}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| \\
&\leq \|\mathbf{e}^{(k)}\|_\infty \sum_{j \neq i} \left| \frac{a_{ij}}{a_{ii}} \right| \\
&= r_i \|\mathbf{e}^{(k)}\|_\infty \\
&\leq r \|\mathbf{e}^{(k)}\|_\infty.
\end{aligned}$$

Therefore

$$\|\mathbf{e}^{(k+1)}\|_\infty \leq r \|\mathbf{e}^{(k)}\|_\infty \leq r^{k+1} \|\mathbf{e}^{(0)}\|_\infty,$$

from which it follows that

$$\lim_{k \to \infty} \|\mathbf{e}^{(k)}\|_\infty = 0$$

since $r < 1$. □

- while both the Jacobi method and the Gauss–Seidel method both converge if A is diagonally dominant, convergence can be slow in some cases
- for example, for

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

we have

$$-D^{-1}(L+U) = \begin{bmatrix} 0 & 1/2 & & \\ 1/2 & \ddots & \ddots & \\ & \ddots & \ddots & 1/2 \\ & & 1/2 & 0 \end{bmatrix}$$

and therefore

$$\rho(B_J) = \cos\frac{\pi}{n+1} = \cos\pi h \approx 1 - \frac{\pi^2 h^2}{2} + \cdots$$

which is approximately 1 for small $h = 1/(n+1)$

- suppose $B_J = B_J^{\mathsf{T}}$, then

$$\frac{\|\mathbf{e}^{(k)}\|_2}{\|\mathbf{e}^{(0)}\|_2} \leq \|B_J\|_2^k = \rho(B_J)^k$$

- if we want $\|\mathbf{e}^{(k)}\|_2/\|\mathbf{e}^{(0)}\|_2 \leq \varepsilon$, then setting $\rho^k = \varepsilon$, we get that

$$k = \frac{-\log\varepsilon}{-\log\rho}$$

is the number of iterations necessary for convergence

- so $\rho = \rho(A)$ controls the rate of convergence

3. SOR METHOD

- reminder: in coordinatewise form, Jacobi method is

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right]$$

and Gauss–Seidel method is

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right]$$

- reminder: in matrix form, Jacobi method is

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

and Gauss–Seidel is

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

- another general adage in numerical computations is: *don't discard previous information, try to use it too*
- applying this to Gauss–Seidel, we could try to use both $x_j^{(k+1)}$ and $x_j^{(k)}$ for $j = 1, \ldots, i-1$ to obtain $x_i^{(k+1)}$ — this yields the method of *successive over relaxation* (SOR)
- this is given by the iteration

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right] + (1-\omega)x_i^{(k)} \tag{3.1}$$

- the quantity ω is called the *relaxation parameter*
- if $\omega = 1$, then the SOR method reduces to the Gauss–Seidel method, i.e.,

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right]$$

- the name 'over relaxation' comes from choosing $\omega > 1$
- in matrix form, the iteration can be written as

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

which can be rearranged to obtain

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

or

$$\mathbf{x}^{(k+1)} = \left(\frac{1}{\omega}D + L\right)^{-1}\left[\left(\frac{1}{\omega} - 1\right)D - U\right]\mathbf{x}^{(k)} + \left(\frac{1}{\omega}D + L\right)^{-1}\mathbf{b} \tag{3.2}$$

- the iteration matrix is

$$B_\omega = \left(\frac{1}{\omega}D + L\right)^{-1}\left[\left(\frac{1}{\omega} - 1\right)D - U\right]$$

- since $B_1 = B_{GS}$, if we pick some $\omega \neq 1$ such that

$$\rho(B_\omega) < \rho(B_{GS}),$$

  we would improve the convergence of Gauss–Seidel
- so SOR is at least as fast as Gauss–Seidel and often faster
- in fact, for certain types of matrices, one can pick ω so that $\rho(B_\omega)$ is minimized
- for example, if A is (i) a nonsingular matrix, (ii) its Jacobi iteration matrix B_J has only real eigenvalues, and (iii) A may be permuted into the form

$$A = \Pi_1 \begin{bmatrix} D_1 & B_{12} \\ B_{21} & D_2 \end{bmatrix} \Pi_2$$

  where Π_1, Π_2 are permutation matrices and D_1, D_2 are diagonal matrices, then the optimal elaxation parameter is given by

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 + \rho(B_J)^2}}$$

  and

$$\rho(B_{\omega_{\text{opt}}}) = \frac{1 + \sqrt{1 - \rho(B_J)^2}}{1 + \sqrt{1 + \rho(B_J)^2}}$$

- note that if $A\mathbf{x} = \mathbf{b}$, then

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

  and so

$$\mathbf{x} = \left(\frac{1}{\omega}D + L\right)^{-1}\left[\left(\frac{1}{\omega} - 1\right)D - U\right]\mathbf{x}^* + \left(\frac{1}{\omega}D + L\right)^{-1}\mathbf{b} \tag{3.3}$$

- subtracting (3.3) from (3.2), we get

$$\mathbf{e}^{(k+1)} = B_\omega \mathbf{e}^{(k)}$$

- note that

$$\begin{aligned} \det B_\omega &= \det\left(\frac{1}{\omega}D + L\right)^{-1} \det\left[\left(\frac{1}{\omega} - 1\right)D - U\right] \\ &= \frac{1}{\det\left(\frac{1}{\omega}D + L\right)} \det\left[\left(\frac{1}{\omega} - 1\right)D - U\right] \\ &= \frac{\omega^n}{\prod_{i=1}^n a_{ii}} \frac{(1-\omega)^n \prod_{i=1}^n a_{ii}}{\omega^n} \\ &= (1 - \omega)^n \end{aligned}$$

- therefore $\prod_{i=1}^n \lambda_i = (1 - \omega)^n$ where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of B_ω, with $|\lambda_1| \geq \cdots \geq |\lambda_n|$
- hence we get

$$|\lambda_1|^n \geq (1 - \omega)^n$$

- since we must also have
$$|\lambda_1| = \rho(B_\omega) < 1$$
for convergence
- it follows that a necessary condition for convergence of SOR is
$$0 < \omega < 2$$
- if A is symmetric positive definite, then the condition $0 < \omega < 2$ is also sufficient — a result of Ostrowski implies that for such an A, $\rho(B_\omega) < 1$ iff $0 < \omega < 2$
- suppose $A \in \mathbb{R}^{n\times n}$ is a symmetric matrix, then $U = L^\mathsf{T}$ and if we set
$$M_\omega = \frac{\omega}{2-\omega}\left(\frac{1}{\omega}D + L\right)D^{-1}\left(\frac{1}{\omega}D + L^\mathsf{T}\right)$$
and define our iteration as
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k-1)} - M_\omega^{-1}(A\mathbf{x}^{(k)} - \mathbf{b})$$
- this is called the method of *symmetric successive over relaxation* (SSOR)
- there are yet other variants of SOR such as:
  - block SOR or block SSOR when the matrix has a block structure; likewise, we may also introduce block Jacobi or block Gauss–Seidel
  - applying the SOR extrapolation to Jacobi method to get
$$x_i^{(k+1)} = \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right] + (1-\omega)x_i^{(k)} \tag{3.4}$$
to preserve parallelism; this is often called JOR
  - note the difference between (3.4) and (3.1) and note that when $\omega = 1$ in (3.4), then the JOR method reduces to Jacobi method
  - one may even define a nonlinear version of SOR for iterations of the form $\mathbf{x}^{(k+1)} = f(\mathbf{x}^{(k)})$ where f is some nonlinear function $f : \mathbb{R}^n \to \mathbb{R}^n$:
$$\mathbf{x}_{\text{SOR}}^{(k+1)} = (1-\omega)\mathbf{x}_{\text{SOR}}^{(k)} + \omega f(\mathbf{x}_{\text{SOR}}^{(k)})$$

4. Richardson method

- unlike the splitting methods in the previous lecture, the iterative methods here do not require splitting A into a sum of two matrices but they are a bit like SOR in that there is a scalar parameter involved at each step
- this scalar parameter can either be fixed throughout (e.g., Richardson) or can vary from one interation to the next (e.g., steepest descent, Chebyshev) or there can even be two scalar parameters at each step (e.g., conjugate gradient)
- the simplest one is known as the *Richardson method*, where the iteration is simply
$$\begin{aligned}\mathbf{x}^{(k+1)} &= (I - \alpha A)\mathbf{x}^{(k)} + \alpha\mathbf{b} \\ &= \mathbf{x}^{(k)} + \alpha(\mathbf{b} - A\mathbf{x}^{(k)}) \\ &= \mathbf{x}^{(k)} + \alpha\mathbf{r}^{(k)}\end{aligned} \tag{4.1}$$
where $\mathbf{r}^{(k)} := \mathbf{b} - A\mathbf{x}^{(k)}$ is the *residual* at the kth step
- note that if $\mathbf{x} = A^{-1}\mathbf{b}$, then we trivially have
$$\mathbf{x} = (I - \alpha A)\mathbf{x} + \alpha\mathbf{b} \tag{4.2}$$

- as usual we define the error $\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$, then subtracting (4.1) from (4.2) yields
$$\mathbf{e}^{(k+1)} = B_\alpha \mathbf{e}^{(k)}$$
where the iteration matrix is $B_\alpha = I - \alpha A$
- we want to choose the parameter $\alpha > 0$ a priori so as to minimize $\rho(B_\alpha)$
- suppose A is symmetric positive definite, with eigenvalues
$$\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n > 0$$
- since $B_\alpha = I - \alpha A$, we have $\lambda_i = 1 - \alpha\mu_i$ for $i = 1, \ldots, n$
- if we want α so that $\rho(B_\alpha)$ is minimized, i.e.,
$$\min_\alpha \max_{1\leq i\leq n} |\lambda_i(\alpha)| = \min_\alpha \max_{1\leq i\leq n} |1 - \alpha\mu_i| = \min_\alpha \max(|1 - \alpha\mu_1|, |1 - \alpha\mu_n|),$$
the optimal parameter α_* is attained when
$$|1 - \alpha_*\mu_1| = |1 - \alpha_*\mu_n|$$
and since these must differ by a sign,
$$1 - \alpha_*\mu_n = -(1 - \alpha_*\mu_1),$$
which yields
$$\alpha_* = \frac{2}{\mu_1 + \mu_n}$$
- note that when $1 - \alpha\mu_1 = -1$, the iteration diverges for some choice of $\mathbf{x}^{(0)}$
- hence the method converges for
$$0 < \alpha < \frac{2}{\mu_1}$$
- however this iteration is sensitive to perturbation and therefore bad numerically
- for example, if $\mu_1 = 10$ and $\mu_n = 10^{-4}$, then the optimal α is $2/(10 + 10^{-4})$, but this is close to a value of α for which the iteration diverges, $\alpha = 2/10$
- also, note that
$$\lambda_1(\alpha_*) = 1 - \frac{2}{\mu_1 + \mu_n}\mu_1 = \frac{\mu_n - \mu_1}{\mu_1 + \mu_n} = \frac{1 - \kappa(A)}{1 + \kappa(A)} \leq 0,$$
and similarly,
$$\lambda_n(\alpha_*) = \frac{\mu_1 - \mu_n}{\mu_1 + \mu_n} = \frac{\mu_1/\mu_n - 1}{\mu_1/\mu_n + 1} = \frac{\kappa(A) - 1}{\kappa(A) + 1} \geq 0$$
- therefore
$$\rho(B_{\alpha_*}) = \frac{\kappa(A) - 1}{\kappa(A) + 1}$$
and we see that the convergence rate depends on $\kappa(A)$