Check Grades Online

~~Hand in HW 21~~

~~HW 22 on dropbox.~~

~~Quiz Review Ideas online~~ ~~Quiz on Wed.~~

~~Short OpenBook/Notes.~~

Iteration Methods. For many systems $A\vec{x} = b$ the matrix is either too large to be stored or so large that Gaussian Elimination is too expensive.

Iteration methods compute a series of progressively accurate iterates to approximate the soln.

JACOBI ITERATION METHOD:

Ex

$$\begin{aligned} 9x_1 + x_2 + x_3 &= b_1 \\ 2x_1 + 10x_2 + 3x_3 &= b_2 \\ 3x_1 + 4x_2 + 11x_3 &= b_3 \end{aligned}$$

Solve each eqn for one of the x's.

$$\left.\begin{aligned} x_1 &= 1/9[b_1 - x_2 - x_3] \\ x_2 &= 1/10[b_2 - 2x_1 - 3x_3] \\ x_3 &= 1/11[b_3 - 3x_1 - 4x_2] \end{aligned}\right\}$$

make an initial guess and use this to iterate.

$\vec{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, x_3^{(0)}]^T$ then

$$\left.\begin{aligned} x_1^{k+1} &= 1/9[b_1 - x_2^k - x_3^k] \\ x_2^{k+1} &= 1/10[b_2 - 2x_1^k - 3x_3^k] \\ x_3^{k+1} &= 1/11[b_3 - 3x_1^k - 4x_2^k] \end{aligned}\right\}$$

This is Jacobi Iteration or The method of simultaneous replacements.

- we make an Initial guess plug into RHS find the new guess an keep going until our solution converges.

NUMERICAL EXAMPLE...

notice that as we iterate the solution approaches the real answer.

by iteration $k = 30$ our answer is not changing much

NUMERICAL EXAMPLE.

Let $b = [10,\ 19,\ 0]^T$.

The solution is $x = [1,\ 2,\ -1]^T$.

To measure the error, we use

$$Error = \|x - x^{(k)}\| = \max_i \left| x_i - x_i^{(k)} \right|$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	*Error*	*Ratio*
0	0	0	0	2.00E + 0	
1	1.1111	1.9000	0	1.00E + 0	0.500
2	0.9000	1.6778	−0.9939	3.22E − 1	0.322
3	1.0351	2.0182	−0.8556	1.44E − 1	0.448
4	0.9819	1.9496	−1.0162	5.06E − 2	0.349
5	1.0074	2.0085	−0.9768	2.32E − 2	0.462
6	0.9965	1.9915	−1.0051	8.45E − 3	0.364
7	1.0015	2.0022	−0.9960	4.03E − 3	0.477
8	0.9993	1.9985	−1.0012	1.51E − 3	0.375
9	1.0003	2.0005	−0.9993	7.40E − 4	0.489
10	0.9999	1.9997	−1.0003	2.83E − 4	0.382
30	1.0000	2.0000	−1.0000	3.01E − 11	0.447
31	1.0000	2.0000	−1.0000	1.35E − 11	0.447

GAUSS – SEIDEL ITERATION

- instead of calculating all x_1, x_2, x_3 why not calculate one and plug into the next.

$$\left.\begin{aligned} x_1^{k+1} &= 1/9\left[b_1 - x_2^k - x_3^k\right] \\ x_2^{k+1} &= 1/10\left[b_2 - 2x_1^{k+1} - 3x_3^k\right] \\ x_3^{k+1} &= 1/11\left[b_3 - 3x_1^{k+1} - 4x_2^{k+1}\right] \end{aligned}\right\}$$

method of successive replacements.

We are subbing in more accurate estimates into x_2 and x_3. Assuming our method is converging this should be more accurate.

NUMERICAL EXAMPLE

notice how quickly our solution converges.

IN GENERAL... when developing an iterative method we break our A-matrix into pieces:

$$Ax = b \longrightarrow Nx - Px = b \quad \text{or} \quad Nx = b + Px$$

then we solve

$$Nx^{k+1} = b + Px^k$$

N must be non-singular

Usually want N to be simple $Nx = f$ is easy to solve.

From our example:

$$\begin{bmatrix} 9 & 1 & 1 \\ 2 & 10 & 3 \\ 3 & 4 & 11 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 11 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 \\ 2 & 0 & 3 \\ 3 & 4 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

N = diagonal of A then $P = N - A$

$$\begin{bmatrix} 9 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 11 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^{k+1} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} 0 & -1 & -1 \\ -2 & 0 & -3 \\ -3 & -4 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^{k}$$

JACOBI METHOD.

NUMERICAL EXAMPLE.

Let $b = [10,\ 19,\ 0]^T$.

The solution is $x = [1,\ 2,\ -1]^T$.

To measure the error, we use

$$Error = \|x - x^{(k)}\| = \max_i \left| x_i - x_i^{(k)} \right|$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	*Error*	*Ratio*
0	0	0	0	$2.00E+0$	
1	1.1111	1.6778	-0.9131	$3.22E-1$	0.161
2	1.0262	1.9687	-0.9958	$3.13E-2$	0.097
3	1.0030	1.9981	-1.0001	$3.00E-3$	0.096
4	1.0002	2.0000	-1.0001	$2.24E-4$	0.074
5	1.0000	2.0000	-1.0000	$1.65E-5$	0.074
6	1.0000	2.0000	-1.0000	$2.58E-6$	0.155

The values of *Ratio* do not approach a limiting value with larger values of the iteration index k.

$$\begin{bmatrix} 9 & 1 & 1 \\ 2 & 10 & 3 \\ 3 & 4 & 11 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 & 0 & 0 \\ 2 & 10 & 0 \\ 3 & 4 & 11 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$N =$ lower triangular part then $P = N - A$

This is gauss - SEIDEL

$$\begin{bmatrix} 9 & 0 & 0 \\ 2 & 10 & 0 \\ 3 & 4 & 11 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^{k+1} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} 0 & -1 & -1 \\ 0 & 0 & -3 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^{k}.$$

There are lots of ways to choose these types of methods.

Ex. $N =$ the tridiagonal matrix

$$\begin{bmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 0 & 4 & 11 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^{k+1} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -3 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^{k}$$

At each step you then must "solve" the system using Gaussian Elimination.

__BUT__ Because this is tridiagonal you can use shortcuts: LU Factorization (6.4)
__AND__ Many of the multipliers and most elements are zero.

Actualy Very Fast!

Our only Problem: CONVERGENCE.
how do we know the method will converge?

VECTOR and MATRIX NORMS:

- a norm is a generalization of the absolute value and we use it to measure the "size" of a vector.

  __Ex__ size of the error

Different ways to define the norm:

EUCLIDEAN NORM: let $\vec{x}$ be a column vector with n-components

$$\|x\|_2 = \left[\sum_{j=1}^{n} |x_j|^2\right]^{1/2}$$

this is the standard definition for the length of a vector. "straight line" distance between the head and tail of the vetor.

1 - NORM

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

sometimes called the "taxi cab" norm. Corresponds to distance as measured when driving on a rectangular grid.

∞ - NORM

$$\|x\|_\infty = \max_{1 \le j \le n} |x_j|$$

also called the maximum norm or Chebyshev norm. Used in numerical analysis to measure maximum error component in a vector quantity.

We also need to be able to relate these norms to MATRIX NORMS.

in other words relate the size of A and x to Ax.

if $\|\cdot\|_\infty$ is the vector norm then the associated matrix norm is

$$\|A\| = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| \quad \text{"row norm"}$$

This is usually the norm we use for numerical analysis.

We will use this idea to talk about convergence!

Convergence means that successive approximations have less error.

Error = Real − Approximate

Real: $Nx = b + Px$

Approx: $Nx^{k+1} = b + Px^{k}$

subtract these.

$$N(x - x^{(k+1)}) = P(x - x^{(k)})$$
$$x - x^{k+1} = N^{-1}P(x - x^{k})$$
$$e^{(k+1)} = M e^{(k)} \qquad \text{where} \qquad M = N^{-1}P$$

To talk about the size of the error we use the ∞-norm.

$$\|e^{(k+1)}\| = \|Me^{(k)}\| \leq \|M\|\|e^{(k)}\|$$

↳ this is a property of Matrix Norms, not proved here.

true for all $n \times n$ matrices M and $n \times 1$ vectors x

This tells us that the error converges to zero if $\|M\| < 1$

Consider our previous Methods:

JACOBI

$$\begin{bmatrix} 9 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 11 \end{bmatrix} \vec{x}^{k+1} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} 0 & -1 & -1 \\ -2 & 0 & -3 \\ -3 & -4 & 0 \end{bmatrix} \vec{x}^{k}$$

$$M = N^{-1}P = \begin{bmatrix} 1/9 & 0 & 0 \\ 0 & 1/10 & 0 \\ 0 & 0 & 1/11 \end{bmatrix} \begin{bmatrix} 0 & -1 & -1 \\ -2 & 0 & -3 \\ -3 & -4 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1/9 & -1/9 \\ -2/10 & 0 & -3/10 \\ -3/11 & -4/11 & 0 \end{bmatrix}$$

Row Norm:

so $\|M\| = 7/11 \doteq 0.636$ so the method converges.

Note the actual convergence rate was better than this!

GAUSS - SEIDEL

$$M = \begin{bmatrix} 9 & 0 & 0 \\ 2 & 10 & 0 \\ 3 & 4 & 11 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -1 & -1 \\ 0 & 0 & -3 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1/9 & -1/9 \\ 0 & 1/45 & -5/18 \\ 0 & 1/45 & 13/99 \end{bmatrix}$$

$$\|M\| = 1/45 + 5/18 = .3$$

- this method converges faster

For most methods we can test this.

GENERAL CONVERGENCE THEOREM.

$$Nx^{(k+1)} = b + Px^{(k)}$$

will converge for all b and all initial guesses $x^{(0)}$, if and only if all of the eigenvalues, λ, of $M = N^{-1}P$ satisfy

$$|\lambda| < 1.$$

- This is the basis for deriving other splittings $A = N - P$ that lead to convergent methods.