L3

Linear Equations

Number of operations

Gauss –Jordan Elimination

Camille Jordan

Linear Systems

Gabriel Cramer

Gauss Elimination

Carl Friedric Gauss

LU Factorization

Iterative Methods

Norms and Conditions

Iterative Compensation

For linear system $Ax = b$,

The exact solution x
Your numerical solution z

your error $\delta x = (x - z)$

Recover $\delta b = Az - b$

Solve for δx $A\ \delta x = \delta b$

Improved solution $(z + \delta x)$

- This procedure can be continued iteratively for better accuracy.
- Convergence checked at every iteration
- Use of LU factorization will reduce no. of operations (?).

Accuracy and Convergence

Two subsequent iterations k and $k+1$

x- vector values x^k and x^{k+1}

Error at step $k+1$ $\Delta x^k = x^{k+1} - x^k$

Accuracy – Choose ε to be much smaller than unity.

Absolute accuracy

$$\left|\Delta x_i^k\right|_{max} \le \varepsilon$$

$$\sum_{i=1}^{n}\left|\Delta x_i^k\right| \le \varepsilon$$

$$\left[\sum_{i=1}^{n}\left(\Delta x_i^k\right)^2\right]^{1/2} \le \varepsilon$$

Relative accuracy

$$\left|\frac{\left|\Delta x_i^k\right|_{max}}{x_i}\right| \le \varepsilon$$

$$\sum_{i=1}^{n}\left|\frac{\Delta x_i^k}{x_i}\right| \le \varepsilon$$

$$\left[\sum_{i=1}^{n}\left(\frac{\Delta x_i^k}{x_i}\right)^2\right]^{1/2} \le \varepsilon$$

Iterative Methods

For $Ax = b$ or $\sum_{j=1}^{n} a_{ij} x_j = b_i$

$$a_{11}\, x_1 + a_{12}x_2 + a_{13}x_3 - b_1 = R_1$$
$$a_{21}\, x_1 + a_{22}x_2 + a_{23}x_3 - b_2 = R_2$$
$$a_{31}\, x_1 + a_{32}x_2 + a_{33}x_3 - b_3 = R_3$$

Residual values for row $i = R_i$

Jacoby Method

At iteration k, solve for x_i in row i

$$R_i^k = a_{i1}\, x_1^k + a_{i2}x_2^k + a_{i3}x_3^k - b_i$$

$$x_1^{k+1} = \frac{R_1^k + b_1 - a_{12}x_2^k - a_{13}x_3^k}{a_{11}} = x_1^k - \frac{R_1^k}{a_{11}}$$
$$x_2^{k+1} = \frac{R_2^k + b_2 - a_{21}x_1^k - a_{13}x_3^k}{a_{11}} = x_2^k - \frac{R_2^k}{a_{22}}$$
$$x_3^{k+1} = x_3^k - \frac{R_3^k}{a_{33}}$$

Initial guess x_i^0 as x_1^0, x_2^0, x_3^0

Use recurrence equations above to solve for x_1^1, x_2^1, x_3^1 and beyond, until desired convergence.

Gauss-Seidel Method

Jacoby method with updated values for $k+1$ iteration known as successive iteration

$$x_1^{k+1} = \frac{R_1^k + b_1 - a_{12}x_2^k - a_{13}x_3^k}{a_{11}}$$
$$x_2^{k+1} = \frac{R_2^k + b_2 - a_{21}x_1^{k+1} - a_{13}x_3^k}{a_{11}}$$
$$x_3^{k+1} = \frac{R_3^k + b_3 - a_{31}x_1^{k+1} - a_{32}x_2^{k+1}}{a_{11}}$$

Faster convergence than Jacoby method.

Successive Over Relaxation- SOR

Recurrence equations can be modified for faster convergence. Use Gauss-Seidel method with

$$x_i^{k+1} = x_i^k - \omega \frac{R_i^k}{a_{ii}}$$

- ω is over-relaxation factor and for most systems $1 < \omega < 2$.
- Optimal ω, ω_{opt} is determined by numerical experimentation.
- For marginally stable systems, SOR can diverge the calculations
- For $0 < \omega < 1$, successive under relaxation (SUR). Slows the calculations and can stabilize the calculations for complex systems.

$$\begin{bmatrix} 4 & -1 & 0 & 1 & 0 \\ -1 & 4 & -1 & 0 & 1 \\ 0 & -1 & 4 & -1 & 0 \\ 1 & 0 & -1 & 4 & -1 \\ 0 & 1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}$$

Table 1.3. Solution by the Jacobi Iteration Method

k	x_1	x_2	x_3	x_4	x_5
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	25.000000	25.000000	25.000000	25.000000	25.000000
2	25.000000	31.250000	37.500000	31.250000	25.000000
3	25.000000	34.375000	40.625000	34.375000	25.000000
4	25.000000	35.156250	42.187500	35.156250	25.000000
5	25.000000	35.546875	42.578125	35.546875	25.000000
...	...	...	...	...	...
16	25.000000	35.714284	42.857140	35.714284	25.000000
17	25.000000	35.714285	42.857142	35.714285	25.000000
18	25.000000	35.714285	42.857143	35.714285	25.000000

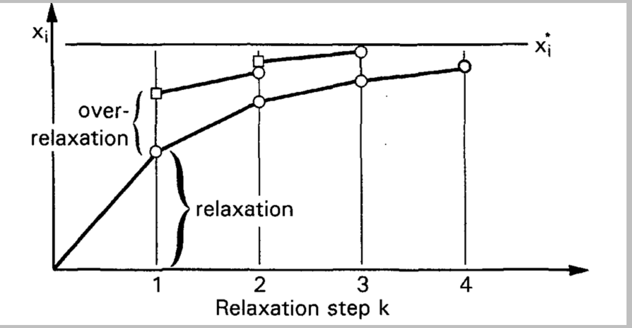

Table 1.4. Solution by the Gauss-Seidel Iteration Method

k	x_1	x_2	x_3	x_4	x_5
0	0.000000	0.000000	0.000000	0.000000	
1	25.000000	31.250000	32.812500	26.953125	23.925781
2	26.074219	33.740234	40.173340	34.506226	25.191498
3	24.808502	34.947586	42.363453	35.686612	25.184757
4	24.815243	35.498485	42.796274	35.791447	25.073240
5	24.926760	35.662448	42.863474	35.752489	25.022510
...	...	...	...	...	...
13	25.000002	35.714287	42.857142	35.714285	25.999999
14	25.000001	35.714286	42.857143	35.714285	25.000000
15	25.000000	35.714286	42.857143	35.714286	25.000000

Table 1.5. Solution by the SOR Method $\omega = 1.1$

k	x_1	x_2	x_3	x_4	x_5
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	27.500000	35.062500	37.142188	30.151602	26.149503
2	26.100497	34.194375	41.480925	35.905571	25.355629
3	24.419371	35.230346	42.914285	35.968342	25.167386
4	24.855114	35.692519	42.915308	35.790750	25.010375
5	24.987475	35.726188	42.875627	35.717992	24.996719
...	...	...	...	...	...
11	24.999996	35.714285	42.857145	35.714287	25.000000
12	25.000000	35.714286	42.857143	35.714286	25.000000
13	25.000000	35.714286	42.857143	35.714286	25.000000

Use Jacobi and Gauss-Seidel to calculate x iteratively

Linear System

$A = [1,2;3,4]$
$b = [2;2]$
$x^{(o)} = [0;0]$