6 Iterative Methods

Methods

The goal is to solve

$$Ax = b.$$

Jacobi Method

You write $A = D + R$ where

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \text{ and } R = A - D = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}$$

The algorithm is

$$Dx^{k+1} + Rx^k = b \implies x^{k+1} = D^{-1}\left(b - Rx^k\right)$$

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

Gauss-Seidel Method

You write $A = L + U$ where

$$L = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \text{ and } U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

The algorithm is

$$Lx^{k+1} + Ux^k = b \implies x^{k+1} = L^{-1}\left(b - Ux^k\right)$$

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

Exercise

Exercise 1: Write a function that solve the system $Ax = b$ using the Jacobi method.Then test your functions with a test case, and explain why it works.

Jacobi converve if $\sum_{j \neq i} |a_{ij}| < |a_{ii}|$. For example you can pick

```
A = np.array([[10, 1, 3,], [2, 30, 1], [3, 5, 25]])
```

Exercise 2: Write a function that solve the system $Ax = b$ using the Gauss-Seidel method. Then test your functions with a test case, and explain why it works.

Exercise 3: Solve the following system using Jacobi and Gauss-Seidel methods. Specify how many iterations were needed.

$$\begin{cases} 2x_1 + x_2 & = 0 \\ -x_1 + 3x_2 + x_3 & = -4 \\ -x_2 + 3x_3 + x_4 & = 12 \\ -x_3 + 4x_4 + x_5 & = 6 \\ -x_4 + 2x_5 & = 9 \end{cases}$$

Exercise 4: Solve the tridiagonal system $Ax = f$ (using Jacobi and Gauss-Seidel methods) with

$$A_{ii} = 4, \quad A_{i,i-1} = A_{i,i+1} = 1$$

for all i. Let the order of the system be $n = 100$, and let

$$f = [1, 1, \dots, 1]^T$$

To see if you got the right answer, you can also solve the system using "scipy.linalg.solve" and compare the solution to yours.

Exercise 5: By using both iterative methods solve the linear system $x = b + Mx$ with

$$M_{ij} = \frac{1}{2n}\left[\frac{t_i^3}{1+t_j} + 1\right], \quad b_j = \frac{1}{4} + t_i - \frac{1}{2}t_i^3,$$

anmd $t_i = (2i - 1)/2n$.

The true solution is

$$x_i = 1 + t_i \quad 1 \leq i \leq n$$

Solve this system for $n = 100$. Then calculate/plot the error $||x - x^{(k)}||$ at each iteration (up to 10 iterations), and the ratios with which they decrease (this means the convergence rate).
In higher dimension the error between two vectors is:

$$||x - x^{(k)}|| = \sum_{i=1}^{n} |x_i - x_i^k|,$$

where x is the exact solution and x^k is your solution at the iteration k.