Linear Algebra

Lecture # 13

20.04.2021

Numerical methods for solving systems of linear algebraic equations - SLAE

SLAE

- Solve the system of n equations with n unknowns:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

- In matrix form: $Ax = b$, A square matrix $(n \times n)$
- The solution: $x_* = A^{-1}b$ exists and is unique

If $\det A \neq 0$

Iterative and Variational Methods

- Iterative methods (methods of successive approximations). Allows to calculate a sequence of vectors $x^{(k)}$ which converges to the exact solution: $x_* = A^{-1}b$ as $k \to \infty$.
- In practice, the number k is defined by the required accuracy.
- Variational Methods use the connection between the variational problem and the problem of solving SLAE: $Ax = b$

Diagonal dominance

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \qquad \forall i = 1, \dots, n$$

If this requirement is satisfied, then there is no problems with finding the solution to the SLAE by any of the known methods!

Simple iteration method

- Solving SLAE:

$$Ax = b$$

- Let's make several equivalent transformations :

$$\tau Ax + x = \tau b + x$$

- The equation system can now be written in the equivalent form convenient for iterations:

$$x = (I - \tau A)x + \tau b$$

Simple iteration method

- Construct the approximation sequence to the solution of the system. Choose an arbitrary vector x_0 as the initial approximation. Usually taken $x_0 = 0$.
- The system solution is found as the limit of the approximation sequence with terms in the form:

$$x_{k+1} = (I - \tau A)x_k + \tau b$$

- If the limit of the sequence exists, then speak of the iterative process convergence to the solution

Simple iteration method

- The iterative process converges to the solution at the geometric rate when the condition is satisfied:

$$||I - \tau A|| < 1$$

- Let x_* be the exact solution of the SLAE: $Ax_* = b$. Subtracting it from the iteration sequence we obtain the error at each of the iterations :

$$x_{k+1} - x_* = (I - \tau A)x_k + \tau A x_* - x_*$$

$$r_{k+1} = x_{k+1} - x_* = (I - \tau A)(x_k - x_*)$$

$$||r_{k+1}|| = ||(I - \tau A)r_k|| \leq ||(I - \tau A)|| \cdot ||r_k||$$

Simple iteration method

- We denote the error $r_k = x_k - x_*$ then

$$||r_k|| \le ||(I - \tau A)||^k \cdot ||r_0||$$

- Hence it follows that for $q = ||(I - \tau A)|| < 1$
- The iterative process converges:

$$\lim_{k\to\infty} x_k = x_* \iff \lim_{k\to\infty} ||r_k|| = 0$$

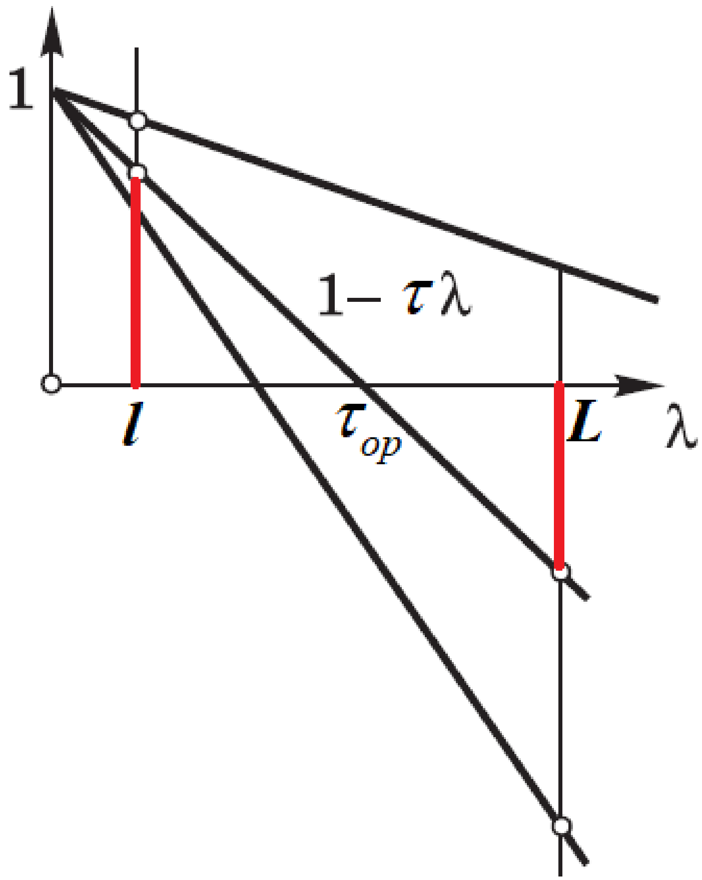
- It is also possible to estimate the iteration number required to achieve accuracy ε:

$$\varepsilon \le q^k \left\| r_0 \right\| \Rightarrow \quad k \ge \ln \frac{\varepsilon}{\left\| r_0 \right\|} \Big/ \ln q$$

Simple iteration method

- The condition for the iterative process convergence $||I - \tau A||<1$ is equivalent that all eigenvalues of the matrix $I - \tau A$ are less than **1** in absolute value.
- Gauss method requires $\sim n^3$ operations. Simple iteration method $\sim kn^2$ where *k* is the iteration number required to achieve given accuracy ε.
- In real tasks, mainly $k << n$.

Optimal parameter τ



- The convergence rate of the iterative process can be characterized by the following value:

$$\max_{\lambda_{\min} \le \lambda(A) \le \lambda_{\max}} \left|1 - \tau\lambda\right|$$

- If A – Positive Definite Matrix when all $\lambda\,(A) > 0$. It is quite typical when the estimate is known for all $\lambda\,(A)$ of the form: $0 < l \le \lambda(\mathrm{A}) \le L < \infty$.

- For some $\tau = \tau_{op}$ exists the moment when

$$1 - \tau_{op} l = -(1 - \tau_{op} L)$$

Optimal parameter τ

- From condition: $1-\tau_{op}l=-(1-\tau_{op}L)$
- We can find optimal parameter:

$$\tau_{op}=2/(L+l)$$

- In this case the convergence condition is:

$$q_{op}=1-\lambda_{min}\tau_{op}=\frac{L-l}{L+l}$$

- Obviously, the closer $l=\lambda_{min}$ to $L=\lambda_{max}$ the closer to zero q_{op} and the faster the convergence. With the removal of λ_{min} from λ_{max} the number $q_{op}\longrightarrow 1$ and the convergence slows down.

Jacobi and Seidel methods

- We represent the matrix of $Ax = b$ in the form:

$$\mathrm{A} = L + D + U$$

- where L and U are lower and upper triangular matrices with zero entries on the diagonal, D is the diagonal matrix. Then $Ax = b$ can be rewritten in the following form:

$$(L + D + U)x = b$$

- Let's construct two iterative processes :

$$Dx_{k+1} + (L + U)x_k = b \;\Rightarrow\; x_{k+1} = -D^{-1}(L + U)x_k + D^{-1}b$$

$$(L + D)x_{k+1} + Ux_k = b \;\Rightarrow\; x_{k+1} = -(L + D)^{-1}Ux_k + (L + D)^{-1}b$$

- These iterative processes are called Jacobi and Seidel methods.

Jacobi and Seidel methods

- Write down these processes in the component notation. Jacobi method will be:

$$\begin{cases} x_1^{k+1} = -(a_{12}x_2^k + a_{13}x_3^k + \cdots + a_{1n}x_n^k - b_1)/a_{11} \\ x_2^{k+1} = -(a_{21}x_1^k + a_{23}x_3^k + \cdots + a_{2n}x_n^k - b_2)/a_{22} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{k+1} = -(a_{n1}x_1^k + a_{n2}x_2^k + \cdots + a_{n,n-1}x_{n-1}^k - b_n)/a_{nn} \end{cases}$$

- Seidel method will be:

$$\begin{cases} x_1^{k+1} = -(a_{12}x_2^k + a_{13}x_3^k + \cdots + a_{1n}x_n^k - b_1)/a_{11} \\ x_2^{k+1} = -(a_{21}x_1^{k+1} + a_{23}x_3^k + \cdots + a_{2n}x_n^k - b_2)/a_{22} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{k+1} = -(a_{n1}x_1^{k+1} + a_{n2}x_2^{k+1} + \cdots + a_{n,n-1}x_{n-1}^{k+1} - b_n)/a_{nn} \end{cases}$$

Variational methods

- Let us construct the connection between the variational problem and the problem of solving SLAE $Ax = b$. Let the vector $y \in R^n$ where R^n is n −dimensional Euclidean space. Consider the quadratic functional of y called the Energy Functional:

$$P(y) = y^T Au - 2y^T b + c$$

- We will also assume that matrix A is positive definite, those $\forall y \neq 0$: $y^T Au > 0$.
- Then the only vector x giving the minimum value to the functional $P(y)$ will be the solution of the SLAE: $Ax = b$.

Variational methods

- $\forall\, y \in R^n$ we can write the difference:

$$P(y) - P(x) = y^T Ay - 2y^T b - x^T Ax + 2x^T b = (Ax = b) =$$

$$= y^T Ay - 2y^T Ax + x^T Ax = y^T Ay - y^T Ax - y^T Ax + x^T Ax =$$

$$= \left(y^T Ax = x^T Ay\right) = y^T Ay - x^T Ay - y^T Ax + x^T Ax =$$

$$= (y-x)^T Ay - (y-x)^T Ax = (y-x)^T A(y-x) > 0, \ \ \forall y \neq x$$

- Those for $Ax = b$ and $\forall y$ is true:

$$\boxed{x = \min_{y \in R^n} P(y)}$$

Variational methods

- All variational methods compose in finding the next approximation by shifting towards the gradient of the functional $\nabla P(x)$:

$$\nabla P(x) = \nabla_x P(x) = \nabla_x \left(x^T A x - 2x^T b + c \right) =$$

$$= \nabla x^T A x + x^T A \nabla x - 2\nabla x^T b = 2\nabla x^T A x - 2\nabla x^T b =$$

$$= 2\nabla x^T \left(Ax - b \right) = 0 \;\Rightarrow\; Ax - b = 0 \;\Rightarrow\; x = x_*$$

$$x_{k+1} = x_k - \tau_k \nabla P(u) = x_k - \tau_k \left(Ax_k - b \right)$$

- Steepest descent method: the iterative parameter τ_k is determined from the condition of the minimum of the functional $P(x_{k+1}, \tau_k)$ by τ_k: $\dfrac{\partial P(\tau_k, x_{k+1})}{\partial \tau_k} = 0$

Steepest descent method

$$\frac{\partial P(\tau_k, x_{k+1})}{\partial \tau_k} = \frac{\partial}{\partial \tau_k}\left(x_{k+1}^T A x_{k+1} - 2x_{k+1}^T b + c\right) = \frac{\partial x_{k+1}^T}{\partial \tau_k} A x_{k+1} + x_{k+1}^T A \frac{\partial x_{k+1}}{\partial \tau_k} -$$

$$-2\frac{\partial x_{k+1}^T}{\partial \tau_k} b = 2\frac{\partial x_{k+1}^T}{\partial \tau_k} A x_{k+1} - 2\frac{\partial x_{k+1}^T}{\partial \tau_k} b = 2\frac{\partial x_{k+1}^T}{\partial \tau_k}\left(A x_{k+1} - b\right) =$$

$$= 2\frac{\partial}{\partial \tau_k}\left(x_k - \tau_k\left(A x_k - b\right)\right)^T \left(A\left(x_k - \tau_k\left(A x_k - b\right)\right) - b\right) =$$

$$= -2\left(A x_k - b\right)^T \left(\left(A x_k - b\right) - \tau_k A\left(A x_k - b\right)\right) = -2R_k^{\ T}\left(R_k - \tau_k A R_k\right) =$$

$$= -2\left(R_k^{\ T} R_k - \tau_k R_k^{\ T} A R_k\right) = 0 \quad \Rightarrow \quad \boxed{\tau_k = \frac{R_k^T R_k}{R_k^T A R_k}}$$

Where $R_k = (Ax_k - b) = Ar_k$ is the Residual

Minimal residual method

- The method of minimal residuals consists in finding the next approximation to minimize the Euclidean norm of the residual R_k:

$$R_k = \left(Ax_k - b\right) \Rightarrow \; x_{k+1} = x_k - \tau_k \left(Ax_k - b\right) = x_k - \tau_k R_k \Rightarrow$$

$$R_{k+1} = Ax_{k+1} - b = Ax_k - \tau_k AR_k - b \;\Leftrightarrow\; R_{k+1} = R_k - \tau_k AR_k$$

- We minimize the squared norm of the residual R_k with respect to τ_k: $R_{k+1}^T R_{k+1} = R_k^T R_k - 2\tau_k R_k^T AR_k + \tau_k^2 R_k^T A^2 R_k \Rightarrow$

$$\frac{\partial}{\partial \tau_k}\left(R_{k+1}^T R_{k+1}\right) = -2R_k^T AR_k + 2\tau_k R_k^T A^2 R_k = 0 \Rightarrow$$

$$\Rightarrow \quad \boxed{\tau_k = \frac{R_k^T AR_k}{R_k^T A^2 R_k}}$$

Conjugate gradient method

- The conjugate gradient method consists in finding the next approximation by simultaneously orthogonalizing and minimizing the residual vectors R_k and R_{k+1}:

$$R_{k+1} = R_k - \tau_k A R_k \Rightarrow R_k^T R_{k+1} = R_k^T R_k - 2\tau_k R_k^T A R_k = 0$$

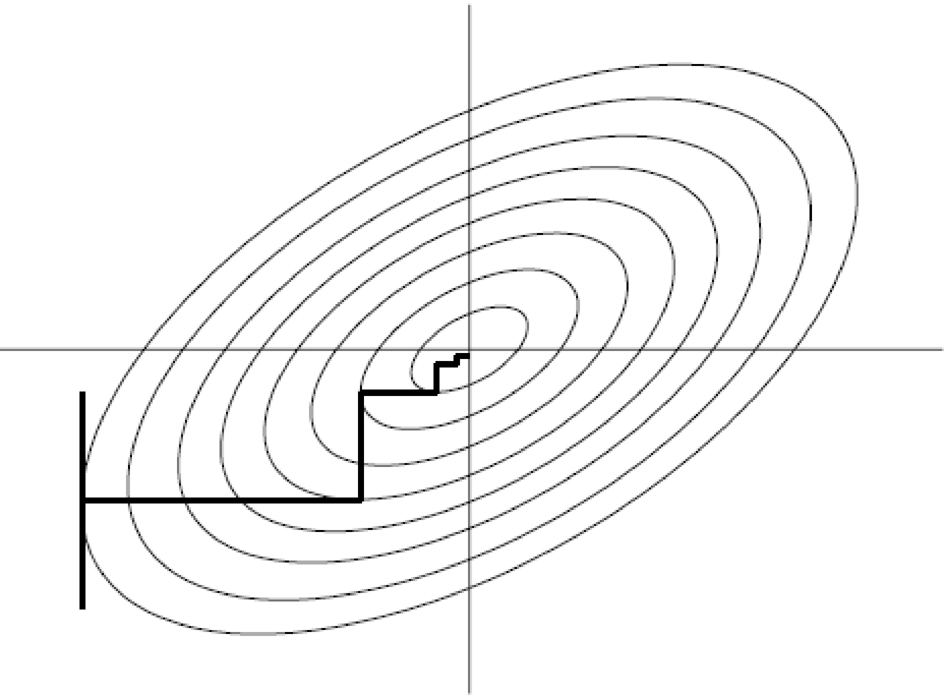
$$\Rightarrow \boxed{\tau_k = \frac{R_k^T R_k}{R_k^T A R_k}}$$

$$x_1 = (I - \tau_0 A)x_0 + \tau_0 b, \qquad R_k = A x_k - b$$

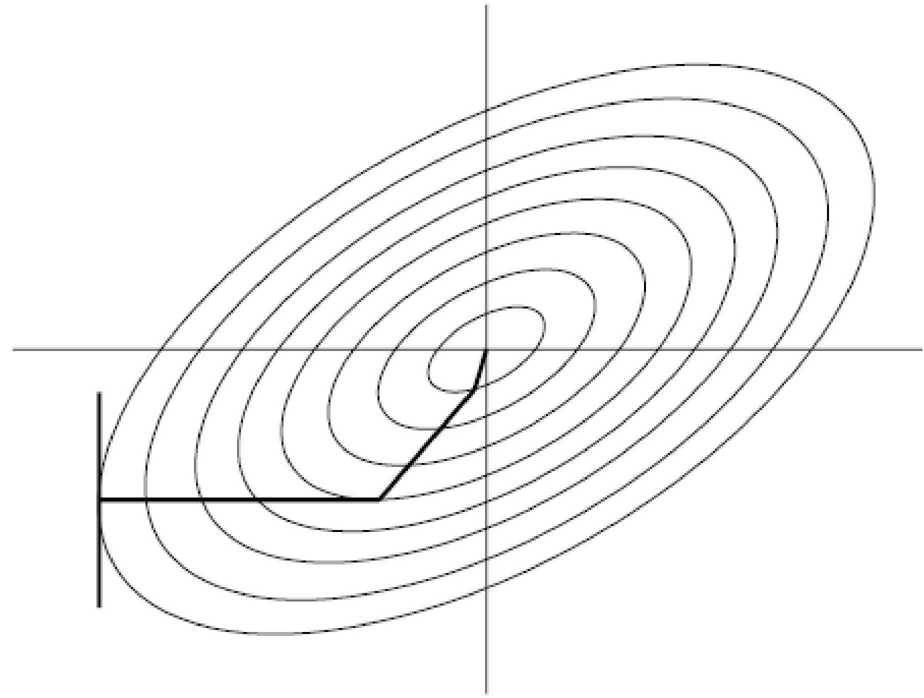
$$x_{k+1} = \alpha_{k+1}\big((I - \tau_k A)x_k + \tau_k b\big) + (1 - \alpha_{k+1})x_{k-1}$$

$$\alpha_1 = 1, \quad \alpha_{k+1} = \left(1 - \frac{\tau_k}{\alpha_k \tau_{k-1}} \frac{R_k^T R_k}{R_{k-1}^T R_{k-1}}\right)^{-1}$$

Conjugate gradient method



Steepest descent



Conjugate gradients

When you have orthogonality, projection and minimizations can be computed one direction at a time

Summary of results by solution methods of SLAE with matrix $A(n \times n)$ and k iterations

Method	Gauss	Simple Iterations	Jacobi	Seidel	Conjugate gradients	Fast Fourier
Knowing the matrix spectrum	no	yes	no	no	no	no
Number of operations	n^3	kn^2	kn^2	$\frac{kn^2}{2}$	kn	$n\log_2 n$