3. Методи розв’язання систем лінійних алгебраїчних рівнянь (СЛАР)

Методи розв’язування СЛАР поділяються на *прямі* та *ітераційні*. При умові точного виконання обчислень прямі методи за скінчену кількість операцій в результаті дають точний розв’язок. Використовуються вони для невеликих та середніх СЛАР $n = 10^2 - 10^4$. Ітераційні методи використовуються для великих СЛАР $n > 10^5$, як правило розріджених. В результаті отримуємо послідовність наближень, яка збігається до розв’язку.

3.1. Метод Гауса

Розглянемо задачу розв’язання СЛАР

$$A\vec{x} = \vec{b}, \tag{1}$$

причому $A = (a_{ij})_{i,j=1}^n$, $\det A \neq 0$, $\vec{x} = (x_i)_{i=1}^n$, $\vec{b} = (b_j)_{j=1}^n$. Метод Крамера з обчисленням визначників для такої системи має складність $Q = O(n! \cdot n)$.

Запишемо СЛАР у вигляді

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n = b_1 \equiv a_{1,n+1}, \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n = b_2 \equiv a_{2,n+1}, \\ \ldots \\ a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n = b_n \equiv a_{n,n+1}. \end{cases} \tag{2}$$

Якщо $a_{1,1} \neq 0$, то ділимо перше рівняння на нього і виключаємо x_1 з інших рівнянь:

$$\begin{cases} x_1 + a_{1,2}^{(1)}x_2 + \ldots + a_{1,n}^{(1)}x_n = a_{1,n+1}^{(1)}, \\ \qquad a_{2,2}^{(1)}x_2 + \ldots + a_{2,n}^{(1)}x_n = a_{2,n+1}^{(1)}, \\ \qquad \ldots \\ \qquad a_{n,2}x_2^{(1)} + \ldots + a_{n,n}^{(1)}x_n = a_{n,n+1}^{(1)}. \end{cases} \tag{3}$$

Процес повторюємо для $x_2, \ldots, x_n$. В результаті отримуємо систему з трикутною матрицею

$$\begin{cases} x_1 + a_{1,2}^{(1)}x_2 + \ldots + a_{1,n}^{(1)}x_n = a_{1,n+1}^{(1)}, \\ \qquad x_2 + \ldots + a_{2,n}^{(2)}x_n = a_{2,n+1}^{(2)}, \\ \qquad \ldots \\ \qquad x_n = a_{n,n+1}^{(n)}. \end{cases} \tag{4}$$

Тобто

$$A^{(n)}\vec{x} = \vec{a}^{(n)}. \tag{5}$$

Це прямий хід методу Гауса. Формули прямого ходу

```
for k in range(1, n):
  for j in range(k + 1, n + 2):
    a[k, j][k] = a[k, j][k - 1] / a[k, k][k - 1]
    for i in range(k + 1, n + 1):
      a[i, j][k] = a[i, j][k - 1] - \
        a[i, j][k - 1] * a[k, j][k]
```

Звідси

$$x_n = a_{n,n+1}^{(n)}, \quad x_i = a_{i,n+1}^{(i)} - \sum_{j=i+1}^{n} a_{i,j}^{(n)} x_j, \tag{6}$$

для $i = \overline{n-1, 1}$. Це формули оберненого ходу.

Складність, тобто кількість операцій, яку необхідно виконати для реалізації методу: $Q_f = 2/3n^2 + O(n^2)$ для прямого ходу, $Q_b = n^2 + O(n)$ для оберненого ходу.

Умова

$$a_{k,k}^{(k-1)} \neq 0 \tag{7}$$

не суттєва, оскільки знайдеться m, для якого

$$\left|a_{m,k}^{(k-1)}\right| = \max_i \left|a_{i,k}^{(k-1)}\right| \neq 0 \tag{8}$$

(оскільки $\det A \neq 0$). Тоді міняємо місцями рядки номерів k і m.

Означення: Елемент

$$a_{k,k}^{(k-1)} \neq 0 \tag{9}$$

називається *ведучим*.

Введемо матриці

$$M_k = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & m_{k,k} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & m_{n,k} & \cdots & 1 \end{pmatrix} \tag{10}$$

елементи якої обчислюється так:

$$m_{k,k} = \frac{1}{a_{k,k}^{(k-1)}}, \quad m_{k,k} = -\frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}. \tag{11}$$

Нехай на k-му кроці $A_{k-1}\vec{x} = \vec{b}_{k-1}$. Множимо цю СЛАР зліва на M_k: $M_k A_{k-1}\vec{x} = M_K \vec{b}_{k-1}$. Позначимо $A_k = M_k A_{k-1}$; $A_0 = A$. Тоді прямий хід методу Гауса можна записати у вигляді

$$M_n M_{n-1} \dots M_1 A\vec{x} = M_n M_{n-1} \dots M_1 \vec{b}. \tag{12}$$

Позначимо останню систему, яка співпадає з (5), так

$$U\vec{x} = \vec{c}, \quad U = (u_{i,j})_{i,j=1}^{n}, \tag{13}$$

причому

$$\begin{cases} u_{i,i} = 1, \\ u_{i,j} = 0, \quad i > j. \end{cases} \tag{14}$$

Таким чином $U = M_n M_{n-1} \dots M_1 A$. Введемо матриці

$$L_k = M_k^{-1} = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{k,k}^{(k-1)} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{n,k}^{(k-1)} & \cdots & 1 \end{pmatrix} \tag{15}$$

Тоді

$$A = L_1 \dots L_n U = LU; \quad L = L_1 \dots L_n, \tag{16}$$

де L — нижня трикутня матриця, U — верхня трикутня матриця. Таким чином метод Гауса можна трактувати, як розклад матриці A в добуток двох трикутних матриць — LU-розклад.

Введемо матрицю перестановок на k-му кроці (це матриця, отримана з одиничної матриці перестановкою k-того і m-того рядка). Тоді при множені на неї матриці A_{k-1} робимо ведучим елементом максимальний за модулем.

$$P_k = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \tag{17}$$

За допомогою цих матриць перехід до трикутної системи (13) тепер має вигляд:

$$M_n M_{n-1} P_{n-1} \dots M_1 P_1 A\vec{x} = M_n M_{n-1} P_{n-1} \dots M_1 P_1 \vec{b}. \tag{18}$$

> **Твердження**: Знайдеться така матриця P перестановок, що $PA = LU$ — розклад матриці на нижню трикутну з ненульовими діагональними елементами і верхню трикутну матрицю з одиницями на діагоналі.

Висновки про **переваги** трикутного розкладу:

- Розділення прямого і оберненого ходів дає змогу економно розв'язувати декілька систем з одноковою матрицею та різними правими частинами.
- Зберігання M, або L та U на місці A.
- Обчислюючи ℓ — кількість перестановок, можна встановити знак визначника.

3.2. Метод квадратних коренів

Цей метод призначений для розв'язання систем рівнянь із симетричною матрицею

$$A\vec{x} = \vec{b}, \quad A^\intercal = A. \tag{19}$$

Він оснований на розкладі матриці A в добуток:

$$A = S^\intercal D S, \tag{20}$$

де S — верхня трикутна матриця, $S^\intercal$ — нижня трикутна матриця, D — діагональна матриця.

Виникає питання: як обчислити S, D по матриці A? Маємо

$$DS_{i,j} = \begin{cases} d_{i,i} s_{i,j}, & i \le j, \\ 0, & i > j. \end{cases} \tag{21}$$

Далі

$$S^\intercal DS_{i,j} = \sum_{l=1}^{n} s^\intercal_{i,l} d_{l,l} s_{l,j} = \sum_{l=1}^{i-1} s^\intercal_{l,i} s_{l,j} d_{l,l} + s_{i,i} s_{i,j} d_{i,i} + \underbrace{s^\intercal_{l,i} \sum_{l=i+1}^{n} s^\intercal_{l,i} s_{l,j} d_{l,l}}_{=0} = a_{i,j}, \tag{22}$$

для $i, j = \overline{1, n}$.

Якщо $i = j$, то

$$|s^2_{i,i}| d_{i,i} = a_{i,i} - \sum_{l=1}^{i-1} |s^2_{l,i}| d_{l,l} \equiv p_i. \tag{23}$$

Тому

$$d_{i,i} = \operatorname{sign}(p_i), \quad s_{i,i} = \sqrt{|p_i|}. \tag{24}$$

Якщо $i < j$, то

$$s_{i,j} = \left(a_{i,j} - \sum_{l=1}^{i-1} s_{l,i}^{\intercal} d_{l,l} s_{l,j} \right) / (s_{i,i} d_{i,i}), \tag{25}$$

де $i = \overline{1, n}$, а $j = \overline{i+1, n}$.

Якщо $A > 0$ (тобто головні мінори матриці A додатні), то всі $d_{i,i} = +1$.

Знайдемо розв’язок рівняння (19). Враховуючи (20), маємо:

$$S^{\intercal} D \vec{y} = \vec{b} \tag{26}$$

і

$$S \vec{x} = \vec{y} \tag{27}$$

Оскільки S — верхня трикутна матриця, а $S^{\intercal} D$ — нижня трикутна матриця, то

$$y_i = \frac{b_i - \sum_{j=1}^{i-1} s_{j,i} d_{j,j} y_j}{s_{i,i} d_{i,i}}, \tag{28}$$

для $i = \overline{1, n}$ і

$$x_i = \frac{y_i - \sum_{j=1}^{i-1} s_{i,j} x_j}{s_{i,i}}, \tag{29}$$

для $i = \overline{n-1, 1}$, де $x_n = y_n / s_{n,n}$.

Метод застосовується лише для симетричних матриць. Його складність $Q = n^3/3 + O(n^2)$.

Переваги цього методу:

- він витрачає в 2 рази менше пам’яті ніж метод Гауса для зберігання $A^{\intercal} = A$ (необхідний об’єм пам’яті $n(n+1)/2 \sim n^2/2$;
- метод однорідний, без перестановок;
- якщо матриця A має багато нульових елементів, то і матриця S також.

Остання властивість дає економію в пам’яті та кількості арифметичних операцій. Наприклад, якщо A має m ненульових стрічок по діагоналі (m-діагональна), то $Q = O(m^2 n)$.

3.3. Обчислення визначника та оберненої матриці

Кількість операцій обчислення детермінанту за означенням — $Q_{\text{det}} = n!$. В методі Гауса — $PA = LU$. Тому

$$\det P \det A = \det L \det U \tag{30}$$

звідки

$$\det A = (-1)^{\ell} \det L \det U = (-1)^{\ell} \prod_{k=1}^{n} a_{k,k}^{(k)}, \tag{31}$$

де ℓ — кількість перестановок. Ясно, що за методом Гауса

$$Q_{\det} = \frac{2}{3} \cdot n^3 + O(n^2) \tag{32}$$

В методі квадратного кореня $A = S^{\intercal} D S$. Тому

$$\det A = \det S^{\intercal} \det D \det S = \prod_{k=1}^{n} d_{k,k} \prod_{k=1}^{n} s_{k,k}^2. \tag{33}$$

Тепер $Q_{\det} = n^3/3 + O(n^2)$.

За означенням

$$AA^{-1} = E, \tag{34}$$

де A^{-1} обернена до матриці A. Позначимо

$$A^{-1} = (\alpha_{i,j})_{i,j=1}^{n}. \tag{35}$$

Тоді $\vec{\alpha}_j = (\alpha_{i,j})_{i=1}^{n}$ — вектор-стовпчик оберненої матриці. З (34) маємо

$$A\vec{\alpha}_j = \vec{e}_j, \quad j = \overline{1,n}. \tag{36}$$

де $\vec{e}_j$ — стовпчики одиничної матриці: $\vec{e}_j = (\delta_{i,j})_{i=1}^{n}$,

$$\delta_{i,j} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{37}$$

Для знаходження A^{-1} необхідно розв’язати n систем. Для знаходження A^{-1} методом Гауса необхідна кількість операцій $Q = 2n^3 + O(n^2)$.

3.4. Метод прогонки

Це економний метод для розв’язання СЛАР з три діагональною матрицею:

$$\begin{cases} -c_0 y_0 + b_0 y_1 = -f_0, \\ a_i y_{i-1} - c_i y_i + b_i y_{i+1} = -f_i, \\ a_N y_{N-1} - c_N y_N = -f_N. \end{cases} \tag{38}$$

Матриця системи

$$A = \begin{pmatrix} -c_0 & b_1 & & 0 \\ a_0 & \ddots & \ddots & \\ & \ddots & \ddots & b_N \\ 0 & & a_N & -c_N \end{pmatrix} \tag{39}$$

тридіагональна.

Розв'язок представимо у вигляді

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = \overline{0, N-1}. \tag{40}$$

Замінимо в (40) і $i \mapsto i-1$ і підставимо в (33), тоді

$$(a_i \alpha_i - c_i) \cdot y_i + b_i y_{i+1} = -f_i - a_i \beta_i \tag{41}$$

Звідси

$$y_i = \frac{b_i}{c_i - a_i \alpha_i} \cdot y_{i+1} + \frac{f_i + a_i \beta_i}{c_i - a_i \alpha_i}. \tag{42}$$

Тому з (36)

$$\alpha_{i+1} = \frac{b_i}{c_i - a_i \alpha_i}, \quad \beta_{i+1} = \frac{f_i + a_i \beta_i}{c_i - a_i \alpha_i}, \quad i = \overline{1, N-1}. \tag{43}$$

Умова розв'язності (38) — $c_i - a_i \alpha_i \neq 0$.

Щоб знайти всі α_i, β_i, треба задати перші значення. З (38):

$$\alpha_1 = \frac{b_0}{c_0}, \quad \beta_1 = \frac{f_0}{c_0}. \tag{44}$$

Після знаходження всіх α_i, β_i обчислюємо y_N з системи

$$\begin{cases} a_N y_N - c_N y_N = -f_N, \\ y_{N-1} = \alpha_N y_N + \beta_N. \end{cases} \tag{45}$$

Звідси

$$y_N = \frac{f_N + a_N \beta_N}{c_N - a_N \alpha_N}. \tag{46}$$

Алгоритм:

```
alpha[1], beta[1] = b[0] / c[0], f[0] / c[0]

for i in range(1, N):
  z[i] = c[i] - a[i] * alpha[i]
  alpha[i + 1], beta[i + 1] = b[i] / z[i], \
    (f[i] + a[i] * beta[i]) / z[i]

y[N] = (f[N] + a[N] * beta[N]) / \
  (c[N] - a[N] * alpha[N])

for i in range(N - 1, -1, -1):
  y[i] = alpha[i + 1] * y[i + 1] + beta[i + 1]
```

Складність алгоритму $Q = 8N - 2$.

Метод можна застосовувати, коли $c_i - a_i\alpha_i \neq 0, \forall i: |\alpha_i| \leq 1$. Якщо $|\alpha_i| \geq q > 1$ то $\Delta y_0 \geq q^N \Delta y_N$ (тут Δy_i абсолютна похибка обчислення y_i), а це приводить до експоненціального накопичення похибок заокруглення, тобто нестійкості алгоритму прогонки.

Теорема (*про достатні умови стійкості метода прогонки*): Нехай $a_i, b_i \neq 0$, та

$$|c_i| \geq |a_i| + |b_i|, \quad \forall i, \quad a_0 = b_N = 0, \tag{47}$$

та хоча би одна нерівність строга. Тоді $|\alpha_i| \leq 1$ та

$$z_i = c_i - a_i\alpha_i \neq 0, \quad i = \overline{1, N}. \tag{48}$$

Задача 8: Довести теорему про стійкість методу прогонки.

3.5. Обумовленість систем лінійних алгебраїчних рівнянь

Нехай задано СЛАР

$$A\vec{x} = \vec{b}. \tag{49}$$

Припустимо, що матриця і права частина системи задані неточно і фактично розв'язуємо систему

$$B\vec{y} = \vec{h}, \tag{50}$$

де $B = A + C$, $\vec{h} = \vec{b} + \vec{\eta}$, $\vec{y} = \vec{x} + \vec{z}$.

Малість детермінанту $\det A \ll 1$ не є необхідною умовою різкого збільшення похибки. Це ілюструє наступний приклад:

$$A = \operatorname{diag}(\varepsilon), \quad a_{i,j} = \varepsilon\delta_{i,j}. \tag{51}$$

Тоді $\det A = \varepsilon^n \ll 1$, але $x_i = b_i/\varepsilon$. Тому $\Delta x_i = \Delta b_i/\varepsilon \gg 1$.

Оцінимо похибку розв'язку. Підставивши значення B, $\vec{h}$, та $\vec{z} = \vec{y} - \vec{x}$, отримаємо:

$$(A + C)(\vec{x} + \vec{z}) = \vec{b} + \vec{\eta}. \tag{52}$$

Віднімемо від цієї рівності (49) у вигляді $A\vec{z} + C\vec{x} + C\vec{z} = \vec{\eta}$. Тоді

$$A\vec{z} = \vec{\eta} - C\vec{x} - C\vec{z}, \quad \vec{z} = A^{-1}(\vec{\eta} - C\vec{x} - C\vec{z}). \tag{53}$$

Означення: Введемо *норми векторів*: $\|\vec{z}\|$:

$$|\vec{z}|_1 = \sum_{i=1}^{n} |z_i|, \tag{54}$$

$$|\vec{z}|_2 = \left(\sum_{i=1}^{n} |z_i|^2\right)^{1/2}, \tag{55}$$

$$|\vec{z}|_\infty = \max_i |z_i|. \tag{56}$$

Означення:_ Норми матриці_, що відповідають нормам вектора, тобто

$$|A|_m = \sup_{|\vec{x}|_m \neq 0} \frac{|A\vec{x}|_m}{|\vec{x}|_m}, \quad m = 1, 2, \infty. \tag{57}$$

такі:

$$|A|_1 = \max_j \sum_{i=1}^{n} |a_{i,j}|, \tag{58}$$

$$|A|_2 = \max_i \sqrt{\lambda_i(A^\intercal A)}, \tag{59}$$

$$|A|_\infty = \max_i \sum_{j=1}^{n} |a_{i,j}|, \tag{60}$$

де $\lambda_i(B)$ — власні значення матриці B.

Позначимо $\delta(\vec{x}) = \|\vec{z}\|/\|\vec{x}\|$, $\delta(\vec{b}) = \|\vec{\eta}\|/\|\vec{b}\|$, $\delta(A) = \|C\|/\|A\|$ — відносні похибки $\vec{x}$, $\vec{b}$, A, де $\|\cdot\|_k$ — одна з введених вище норм.

Для характеристики зв’язку між похибками правої частини та розв’язку вводять поняття обумовленості матриці системи.

Означення: *Число обумовленості матриці* A — $\mathrm{cond}\,(A) = \|A\| \cdot \|A^{-1}\|$.

Теорема: Якщо $\exists A^{-1}$ та $\|A^{-1}\| \cdot \|C\| < 1$, то

$$\delta(\vec{x}) \leq \frac{\mathrm{cond}\,(A)}{1 - \mathrm{cond}\,(A) \cdot \delta(A)} (\delta(A) + \delta(\vec{b})), \tag{61}$$

де $\mathrm{cond}\,(A)$ — число обумовленості.

Доведення:

$$A\vec{z} = \vec{\eta} - C\vec{x} - C\vec{z}, \quad \vec{z} = A^{-1}\vec{\eta} - A^{-1}C\vec{x} - A^{-1}C\vec{z} \tag{62}$$

$$|\vec{z}| \leq |A^{-1}\vec{\eta}| + |A^{-1}C\vec{x}| + |A^{-1}C\vec{z}| \leq |A^{-1}| \cdot |\vec{\eta}| + |A^{-1}| \cdot |C| \cdot |\vec{x}| + |A^{-1}| \cdot |C| \cdot |\vec{z}|. \tag{63}$$

$$|\vec{z}| \leq \frac{|A^{-1}| \cdot (|\vec{\eta}| + |C| \cdot |\vec{x}|)}{1 - |A^{-1}| \cdot |C|} \tag{64}$$

Оцінка похибки

$$\begin{aligned} \delta(\vec{x}) \leq \frac{|A^{-1}|}{1 - |A^{-1}| \cdot |C|} \left(\frac{|\vec{\eta}|}{|\vec{x}|} + |C| \right) = \frac{|A^{-1}| \cdot |A|}{1 - |A^{-1}| \cdot |A| \cdot \dfrac{|C|}{|A|}} \left(\frac{|\vec{\eta}|}{|A| \cdot |\vec{x}|} + \delta(A) \right) \leq \\ \leq \frac{\mathrm{cond}\,(A)}{1 - \mathrm{cond}\,(A) \cdot \delta(A)} \left(\frac{|\vec{\eta}|}{|\vec{x}|} + \delta(A) \right). \quad \square \end{aligned} \tag{65}$$

Наслідок: Якщо $C \equiv 0$, то $\delta(\vec{x}) \le \operatorname{cond}(A) \cdot \delta(\vec{b})$.

Властивості $\operatorname{cond}(A)$:

- $\operatorname{cond}(A) \ge 1$;
- $\operatorname{cond}(A) \ge \max_i |\lambda_i(A)| / \min_i |\lambda_i(A)|$;
- $\operatorname{cond}(AB) \le \operatorname{cond}(A) \cdot \operatorname{cond}(B)$;
- $A^\intercal = A^{-1} \implies \operatorname{cond}(A) = 1$.

Друга властивість має місце оскільки довільна норма матриці не менше її найбільшого за модулем власного значення. Значить $\|A\| \ge \max|\lambda_A|$. Оскільки власні значення матриць A^{-1} та A взаємно обернені, то

$$|A^{-1}| \ge \max \frac{1}{|\lambda_A|} = \frac{1}{\min|\lambda_A|}. \tag{66}$$

Якщо $1 \ll \operatorname{cond}(A)$, то система називається *погано обумовленою*.

Оцінка впливу похибок заокруглення при обчисленні розв’язку СЛАР така (Дж. Уілкінсон): $\delta(A) = O(n\beta^{-t})$, $\delta(\vec{b}) = O(\beta^{-t})$, де β — розрядність ЕОМ, t — кількість розрядів, що відводиться під мантису числа. З оцінки (61) витікає: $\delta(\vec{x}) = \operatorname{cond}(A) \cdot O(n\beta^{-t})$. Висновок: найпростіший спосіб підвищити точність обчислення розв’язку погано обумовленої СЛАР — збільшити розрядність ЕОМ при обчисленнях. Інші способи пов’язані з розглядом цієї СЛАР як некоректної задачі із застосуванням відповідних методів її розв’язання.

Приклад погано обумовленої системи — системи з матрицею Гільберта

$$H_n = \left(\frac{1}{i+j-1} \right)_{i,j=1}^{n}, \tag{67}$$

наприклад $\operatorname{cond}(H_8) \approx 10^9$.