Numerical Analysis

established about 4'000 years ago

5. Алгебраїчна проблема власних значень

Нехай задано матрицю $A \in \mathbb{R}^{n \times n}$. Тоді задача на власні значення ставиться так: знайти число λ та вектор $x \neq 0$, що задовольняють рівнянню

$$Ax - \lambda x. \tag{1}$$

> **Означення**: λ називається *власним значенням* A, а x — *власним вектором*.

З (1)

$$\begin{aligned} \det(A - \lambda E) = P_n(\lambda) = \\ = (-1)^n \lambda^n + a_n \lambda^{n-1} + \ldots + a_0 = 0. \end{aligned} \tag{2}$$

Тут $P_n(\lambda)$ — характеристичний багаточлен.

Для розв'язання багатьох задач механіки, фізики, хімії потрібне знаходження всіх власних значень λ_i, $i = \overline{1, n}$, а іноді й всіх власних векторів x_i, що відповідають λ_i.

> **Означення**: Цю задачу називають *повною проблемою власних значень*.

В багатьох випадках потрібно знайти лише максимальне або мінімальне за модулем власне значення матриці. При дослідженні стійкості коливальних процесів іноді потрібно знайти два максимальних за модулем власних значення матриці.

> **Означення**: Останні дві задачі називають *частковими проблемами власних значень*.

5.1. Степеневий метод

Література:

- БЖК, стор. 309–314;

- КБМ, стор. 149–157.

1. Знаходження $\lambda_{\max}$: $\lambda_{\max} \equiv |\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \ldots$.

   Нехай $x^{(0)}$ — заданий вектор, будемо послідовно обчислювати вектори

$$x^{(k+1)} = Ax^{(k)}, \quad k = 0, 1, \ldots \tag{3}$$

   Тоді $x^{(k)} = A^k x^{(0)}$. Нехай $\{e_i\}_{i=1}^{n}$ — система власних векторів. Представимо $x^{(0)}$ у вигляді:

$$x^{(0)} = \sum_{i=1}^{n} c_i e_i. \tag{4}$$

   Оскільки $Ae_i = \lambda_i e_i$, то $x^{(k)} = \sum\limits_{i=1}^{n} c_i \lambda_i^k e_i$. При великах k: $x^{(k)} \approx c_1 \lambda_1^k e_1$. Тому

$$\mu_1^{(k)} = \frac{x_m^{(k+1)}}{x_m^{(k)}} = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right). \tag{5}$$

   Значить $\mu_1^{(k)} \xrightarrow[k\to\infty]{} \lambda_1$.

   Якщо матриця $A = A^\intercal$ симетрична, то існує ортонормована система векторів $\langle e_i, e_j \rangle = \delta_{ij}$. Тому

$$\begin{aligned}
\mu_1^{(k)} &= \frac{\left\langle x^{(k+1)}, x^{(k)} \right\rangle}{\left\langle x^{(k)}, x^{(k)} \right\rangle} = \\
&= \frac{\left\langle \sum\limits_i c_i \lambda_i^{k+1} e_i, \sum\limits_j c_j \lambda_j^k e_j \right\rangle}{\left\langle \sum\limits_i c_i \lambda_i^k e_i, \sum\limits_j c_j \lambda_j^k e_j \right\rangle} = \\
&= \frac{\sum\limits_i c_i^2 \lambda_i^{2k+1}}{\sum\limits_i c_i^2 \lambda_i^{2k}} = \\
&= \frac{c_1^2 \lambda_1^{2k+1} + c_2^2 \lambda_2^{2k+1} + \ldots}{c_1^2 \lambda_1^{2k} + c_2^2 \lambda_2^{2k} + \ldots} = \\
&= \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right) \xrightarrow[k\to\infty]{} \lambda_1.
\end{aligned} \tag{6}$$

   Це означає збіжність до максимального за модулем власного значення з квадратичною швидкістю.

Якщо $|\lambda_1| > 1$, то при проведенні ітерацій відбувається зріст компонент вектора $x^{(k)}$, що приводить до «переповнення» (overflow). Якщо ж $|\lambda_1| < 1$, то це приводить до зменшення компонент (underflow). Позбутися негативу такого явища можна нормуючи вектори $x^{(k)}$ на кожній ітерації.

Алгоритм степеневого методу знаходження максимального за модулем власного значення з точністю ε виглядає так:

```
e[0] = x[0] / norm(x[0])

k = 0
while True:
    k += 1

    x[k + 1] = A * x[k]
    μ[k][1] = scalar_product(x[k + 1], e[k])
    e[k + 1] = x[k + 1] / norm(x[k + 1])

    if abs(μ[k + 1][1] - μ[k][1]) < ε:
        break

λ[1] = μ[k + 1][1]
```

За цим алгоритмом для симетричної матриці $A^\intercal = A$ швидкість прямування $\mu_1^{(k)}$ до $\lambda_{\max}$ — квадратична.

2. Знаходження λ_2: $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \ldots$. Нехай λ_1, e_1 відомі.

Задача 10: Довести, що якщо $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \ldots$ то

$$\mu_2^{(k)} = \frac{x_m^{(k+1)} - \lambda_1 x_m^{(k)}}{x_m^{(k)} - \lambda_1 x_m^{(k-1)}} \xrightarrow[k\to\infty]{} \lambda_2, \tag{7}$$

де $x^{(k+1)} = Ax^{(k)}$, $x_m^{(k)}$ — m-та компонента $x^{(k)}$.

Задача 11: Побудувати алгоритм обчислення λ_2, e_2, використовуючи нормування векторів та скалярні добутки для обчислення $\mu_2^{(k)}$.

3. Знаходження мінімального власного числа $\lambda_{\min}(A) = \min_i |\lambda_i(A)|$.

Припустимо , що $\lambda_i(a) > 0$ то відоме $\lambda_{\max}$. Розглянемо матрицю $B = \lambda_{\max} E - A$. Маємо

$$\forall i: \quad \lambda_i(B) = \lambda_{\max} - \lambda_i(A). \tag{8}$$

Тому $\lambda_{\max}(B) = \lambda_{\max}(A) - \lambda_{\min}(A)$. Звідси $\lambda_{\min}(A) = \lambda_{\max}(A) - \lambda_{\max}(B)$.

Якщо $\exists i$: $\lambda_i(A) < 0$, то будуємо матрицю $\overline{A} = \sigma E + A$, $\sigma > 0$: $\overline{A} > 0$ і для неї попередній розгляд дає необхідний результат. Замість λ_max в матриці B можна використовувати $\|A\|$.

Ще один спосіб обчислення мінімального власного значення полягає в використання обернених ітерацій:

$$Ax^{(k+1)} = x^{(k)}, \quad k = 0, 1, \ldots \tag{9}$$

Але цей метод вимагає більшої кількості арифметичних операцій: складність методу на основі формули (3): $Q = O(n^2)$, а на основі (9) — $Q = O(n^3)$, оскільки треба розв'язувати СЛАР, але збігається метод (9) швидше.

5.2. Ітераційний метод обертання

Література:

- КБМ, стор. 157–161.

Це метод розв'язання повної проблеми власних значень для симетричних матриць $A^\intercal = A$. Існує матриця U, що приводить матрицю А до діагонального виду:

$$A = U\Lambda U^\intercal, \tag{10}$$

де Λ — діагональна матриця, по діагоналі якої стоять власні значення λ_i; U — унітарна матриця, тобто: $U^{-1} = U^\intercal$.

З (1) маємо

$$\Lambda = U^\intercal A U. \tag{11}$$

Нехай $\exists \tilde{U}$ — матриця, така що $\tilde{\Lambda} = \tilde{U}^\intercal A \tilde{U}$ і $\tilde{\Lambda} = \left(\tilde{\lambda}_{ij}\right)_{i,j=1}^{n}$, $\left|\tilde{\lambda}_{ij}\right| < \delta \ll 1, i \neq j$.

Тоді діагональні елементи мало відрізняються від власних значень

$$\left|\tilde{\lambda}_{ij} - \lambda_i(A)\right| < \varepsilon = \varepsilon(\delta). \tag{12}$$

Введемо

$$t(A) = \sum_{\substack{i,j=1 \\ i \neq j}}^{n} a_{ij}^2. \tag{13}$$

З малості величини $t(A)$ випливає, що діагональні елементи малі. По $A = A_0$ за допомогою матриць обертання U_k:

$$U_k = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cos\phi & \cdots & -\sin\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sin\phi & \cdots & \cos\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}. \tag{14}$$

що повертають систему векторів на кут φ, побудуємо послідовність $\{A_k\}$ таку, що $A_k \xrightarrow[k\to\infty]{} \Lambda$.

> **Задача 12**: Показати, що матриця обертання U_k є унітарною: $U_k^{-1} = U_k^{\intercal}$.

Послідовно будуємо:

$$A_{k+1} = U_K^T A_k U_k, \tag{15}$$

> **Означення**: Процес (15) називається *монотонним*, якщо: $t(A_{k+1}) < t(A_k)$.

> **Задача 13**: Довести, що для матриці (15) виконується:
>
> $$a_{i,j}^{(k+1)} = a_{i,j}^{(k)} \cos(2\varphi) + \frac{1}{2}\left(a_{j,j}^{(k)} - a_{i,i}^{(k)}\right)\sin(2\varphi), \tag{16}$$
>
> Показати, що
>
> $$t(A_{k+1}) = t(A_k) - 2\left(a_{i,j}^{(k)}\right)^2 \tag{17}$$
>
> якщо вибирати φ з умови $a_{i,j}^{(k+1)} = 0$.

Звідси

$$\varphi = \varphi_k = \frac{1}{2}\arctan\left(p^{(k)}\right), \tag{18}$$

тобто

$$p^{(k)} = \frac{2a_{i,j}^{(k)}}{a_{i,i}^{(k)} - a_{j,j}^{(k)}}, \tag{19}$$

де

$$\left|a_{i,j}^{(k)}\right| = \max_{\substack{m,l \\ m \neq l}} \left|a_{m,l}^{(k)}\right|. \tag{20}$$

Тоді $t(A_k) \xrightarrow[k\to\infty]{} 0$. Чим більше n тим більше ітерацій необхідно для зведення A до Λ.

Якщо матриця несиметрична, то застосовують QR- або QL-методи.

Назад до лекцій

Назад на головну
