Введение в численные методы. Точные методы решения СЛАУ

Баев А.Ж.

Казахстанский филиал МГУ

11 февраля 2020

План на семестр

1. СЛАУ (точные методы)
2. СЛАУ (итерационные методы)
3. решение нелинейных уравнений
4. интерполяция
5. аппроксимация
6. интегрирование
7. дифференцирование

Линейная алгебра

Решение системы линейных алгебраических уравнений:

$$Ax = f$$

где $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$, $f \in \mathbb{R}^n$.
Дано A, f. Найти x.

Метод прогонки

Метод прогонки (англ. tridiagonal matrix algorithm)
Алгоритм Томаса (англ. Thomas algorithm)
Дана квадратная трехдиагональная матрица $A \in \mathbb{R}^{n \times n}$.
Решить СЛАУ $Ax = f$.

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & \dots & 0 & 0 & 0 \\ 0 & 0 & a_4 & b_4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & b_{n-2} & c_{n-2} & 0 \\ 0 & 0 & 0 & 0 & \dots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ \vdots \\ f_{n-2} \\ f_{n-1} \\ f_n \end{pmatrix}$$

Метод прогонки

В виде строчных соотношений:

$$\begin{cases} b_1x_1 + c_1x_2 & = f_1 \\ a_ix_{i-1} + b_ix_i + c_ix_{i+1} & = f_i, i = 2, ..., n-1 \\ a_nx_{n-1} + b_nx_n & = f_n \end{cases}$$

Метод прогонки

1) Добавим фиктивные элементы: $a_1 = c_n = 0$.
Добавим фиктивные решения: $x_0 = x_{n+1} = 0$.

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = f_i, i = 1, ..., n \tag{1}$$

Метод прогонки

2) Допустим x_i выражаются линейно друг через друга :

$$x_{i-1} = \alpha_i x_i + \beta_i, i = 1, ..., n+1 \tag{2}$$

Подставим (2) в соотношение (1) при $i = 1, ..., n$:

$$a_i(\alpha_i x_i + \beta_i) + b_i x_i + c_i x_{i+1} = f_i$$

$$(a_i\alpha_i + b_i)x_i + c_i x_{i+1} = f_i - a_i\beta_i$$

$$x_i = \frac{-c_i}{a_i\alpha_i + b_i}x_{i+1} + \frac{f_i - a_i\beta_i}{a_i\alpha_i + b_i}$$

Метод прогонки

3) Из предположения, что $x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1}, i = 0, \dots, n$:

$$\begin{cases} \alpha_{i+1} = \frac{-c_i}{a_i \alpha_i + b_i} \\ \beta_{i+1} = \frac{f_i - a_i \beta_i}{a_i \alpha_i + b_i} \end{cases} \quad i = 1, \dots, n$$

С учетом того, что $x_0 = 0$ получаем $\alpha_1 = \beta_1 = 0$.

Метод прогонки

Прямой ход прогонки:

$$\begin{cases} \alpha_1 = 0, \beta_1 = 0, \\ \alpha_{i+1} = \frac{-c_i}{a_i \alpha_i + b_i} \\ \beta_{i+1} = \frac{f_i - a_i \beta_i}{a_i \alpha_i + b_i} \end{cases} \qquad i = 1, \dots, n \tag{3}$$

Схема вычисления:

$$\begin{tikzcd}
\alpha_1 = 0 \arrow[r] \arrow[dr] & \alpha_2 \arrow[r] \arrow[dr] & \alpha_3 \arrow[r] \arrow[dr] & \dots \arrow[r] \arrow[dr] & \alpha_n \arrow[r] \arrow[dr] & \alpha_{n+1} \\
\beta_1 = 0 \arrow[r] & \beta_2 \arrow[r] & \beta_3 \arrow[r] & \dots \arrow[r] & \beta_n \arrow[r] & \beta_{n+1}
\end{tikzcd}$$

Метод прогонки

Обратный ход прогонки:

$$\begin{cases} x_{n+1} & = 0 \\ x_i & = \alpha_{i+1} x_{i+1} + \beta_{i+1},\, i = \overline{n, 1} \end{cases} \tag{4}$$

Схема вычисления:

$$\begin{tikzcd}
\alpha_1 & \alpha_2 \arrow[dl] & \alpha_3 \arrow[dl] & \dots \arrow[dl] & \alpha_{n-1} \arrow[dl] & \alpha_n \arrow[dl] & \alpha_{n+1} \arrow[dl] \\
x_1 & x_2 \arrow[l] & x_3 \arrow[l] & \dots \arrow[l] & x_{n-1} \arrow[l] & x_n \arrow[l] & x_{n+1} = 0 \arrow[l] \\
\beta_1 & \beta_2 \arrow[ul] & \beta_3 \arrow[ul] & \dots \arrow[ul] & \beta_{n-1} \arrow[ul] & \beta_n \arrow[ul] & \beta_{n+1} \arrow[ul]
\end{tikzcd}$$

Метод прогонки (пример)

$$\begin{pmatrix} 4 & 3 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 8 \end{pmatrix}$$

Запишем вектора a, b, c и f:

$$a = (0, 1, 1)$$

$$b = (4, 3, 2)$$

$$c = (3, 1, 0)$$

$$f = (10, 10, 8)$$

Метод прогонки (пример)

1) Вычислим коэффициенты α и β прямым ходом прогонки (3):

$$\alpha_1 = 0 \qquad \beta_1 = 0$$

$$\alpha_2 = \frac{-c_1}{a_1\alpha_1+b_1} = \frac{-3}{0\cdot 0+4} = -\frac{3}{4} \qquad \beta_2 = \frac{f_1-a_1\beta_1}{a_1\alpha_1+b_1} = \frac{10-0\cdot 0}{0\cdot 0+4} = \frac{5}{2}$$

$$\alpha_3 = \frac{-c_2}{a_2\alpha_2+b_2} = \frac{-1}{1\cdot(-\frac{3}{4})+3} = -\frac{4}{9} \qquad \beta_3 = \frac{f_2-a_2\beta_2}{a_2\alpha_2+b_2} = \frac{10-1\cdot\frac{5}{2}}{1\cdot(-\frac{3}{4})+3} = \frac{10}{3}$$

$$\alpha_4 = \frac{-c_3}{a_3\alpha_3+b_3} = \frac{0}{1\cdot(-\frac{4}{9})+2} = 0 \qquad \beta_4 = \frac{f_3-a_3\beta_3}{a_3\alpha_3+b_3} = \frac{8-1\cdot\frac{10}{3}}{1\cdot(-\frac{4}{9})+2} = 3$$

2) Вычислим решение x обратным ходом прогонки (4):

$$x_4 = 0$$

$$x_3 = \alpha_4 x_4 + \beta_4 = 0 * 0 + 3 = 3$$

$$x_2 = \alpha_3 x_3 + \beta_3 = -\frac{4}{9} * 3 + \frac{10}{3} = 2$$

$$x_1 = \alpha_2 x_2 + \beta_2 = -\frac{3}{4} * 2 + \frac{5}{2} = 1$$

Ответ: $x = (1, 2, 3)$.

Метод прогонки (реализация)

```
sweep(a[1:n], b[1:n], c[1:n], f[1:n]) -> x[1:n]
    alpha[1:n+1]
    beta[1:n+1]

    a[1] := 0
    c[n] := 0
    alpha[1] := 0
    beta[1] := 0
    for i := 1 .. n
        d := a[i] * alpha[i] + b[i]
        alpha[i+1] := - c[i] / d
        beta[i+1] := (f[i] - a[i] * beta[i]) / d

    x[n+1] := 0
    for i := n .. 1
        x[i] := alpha[i+1] * x[i+1] + beta[i+1]

    return x[1:n]
```

Метод прогонки (реализация)

```
sweep(a[1:n], b[1:n], c[1:n], f[1:n]) -> x[1:n]
    alpha[1:n+1]
    beta[1:n+1]

    a[1] := 0
    c[n] := 0
    alpha[1] := 0
    beta[1] := 0
    for i := 1 .. n
        d := a[i] * alpha[i] + b[i]
        alpha[i+1] := - c[i] / d
        beta[i+1] := (f[i] - a[i] * beta[i]) / d

    x[n] := beta[n+1]
    for i := n-1 .. 1
        x[i] := alpha[i+1] * x[i+1] + beta[i+1]

    return x
```

Метод прогонки (алгоритмическая сложность)

1. Прямой ход — $f(n) = 4n$;
2. Обратный ход — $f(n) = n$;
3. Итог — $f(n) = 5n$.

Асимптотика прямого и обратного хода: $\Theta(n)$.

Метод прогонки (существование решения)

Будем говорить, что матрица имеет **диагональное преобладание**, если

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}|$$

Для матриц с диагональным преобладанием справедливо утверждение, что все главные миноры отличны от нуля и система уравнений имеет единственное решение.

Метод прогонки (устойчивость решения)

Если матрица обладает диагональным преобладанием, то

$$|\alpha_i| < 1$$

Доказательство:

$$|\alpha_{i+1}| = \left|\frac{c_i}{a_i\alpha_i + b_i}\right| \leqslant \frac{|c_i|}{|b_i| - |a_i|} < 1$$

Это означает, что ошибка $x_{i+1} = \alpha_i x_i + \beta_i$ не превышает ошибки в x_i.

Метод Холецкого

Пусть $A = A^T$ и $A > 0$. Тогда

$$A = S^T S$$

где S — верхнетреугольная матрица.

Решение $Ax = f$ сводится к решению $S^T y = f$ и $Sx = y$.

Метод Холецкого

$$a_{ij} = \sum_{k=1}^{i} s_{ki}s_{kj}, j \geqslant i$$

.
Получаем:

$$s_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} s_{ki}^2}, j = i + 1 \dots n$$

$$s_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} s_{ki}s_{kj}}{s_{ii}}, j = i + 1 \dots n$$

Зачем?

Обусловленность систем (норма вектора)

Введем норму вектора $||x||$:

1. $||x|| = 0 \Leftrightarrow x = 0$.
2. $||x + y|| \leqslant ||x|| + ||y||$
3. $\alpha \in \mathbb{R}$ $||\alpha x|| = |\alpha| ||x||$.

Например:

1. $||x||_1 = |x_1| + |x_2| + \ldots + |x_n|$.
2. $||x||_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$.
3. $||x||_\infty = \max_i |x_i|$.

Обусловленность систем (норма матрицы)

Матричная норма $||A||$ согласуется нормой вектора $||x||$:

$$||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$$

Например:

1. $||A||_1 = \max_i \left(\sum_j |a_{ij}| \right)$.
2. $||A||_2 = \max_i |\lambda_i|$.
3. $||A||_\infty = \max_j \left(\sum_i |a_{ij}| \right)$.

Обусловленность систем (основное неравенство)

$$||Ax|| \leqslant ||A|| ||x||$$

Обусловленность систем

Решаем $Ax = f$.
Как сильно может измениться решение x в результате изменения правой части f?

Обусловленность систем

$$\delta x = x - \tilde{x}$$

$$\delta f = f - \tilde{f}$$

Получаем

$$A(\delta x) = \delta f$$

Выпишем два неравенства

$$||\delta x|| = ||A^{-1}(\delta f)|| \leqslant ||A^{-1}|| \cdot ||\delta f||$$

$$||f|| = ||Ax|| \leqslant ||A|| \cdot ||x||$$

Перемножим

$$||\delta x|| \cdot ||f|| \leqslant ||A|| \cdot ||A^{-1}|| \cdot ||\delta f|| \cdot ||x||$$

$$\frac{||\delta x||}{||x||} \leqslant ||A|| \cdot ||A^{-1}|| \cdot \frac{||\delta f||}{||f||}$$

Обусловленность систем

Чем меньше число обусловленности, тем лучше:

$$\mu = ||A|| \cdot ||A^{-1}||$$

Пример матрицы с хорошим число обусловленности

$$\begin{pmatrix} 10 & 0 \\ 0 & 11 \end{pmatrix}$$

Пример матрицы с плохим числом обусловленности

$$\begin{pmatrix} 10 & 10 \\ 10 & 11 \end{pmatrix}$$

Обусловленность систем

```python
import numpy as np

a = np.array([[10, 10], [10, 11]])
b = np.linalg.inv(a)

mu = np.linalg.norm(a) * np.linalg.norm(b)
print(mu)
```

Вывод: 42.1000

Обусловленность систем

```python
import numpy as np

a = np.array([[10, 10], [10, 11]])

mu = np.linalg.cond(a)
print(mu)
```

Вывод: 42.0762

Метод прогонки

```python
import scipy.linalg as sl

A = [ [0, 3, 1],
      [4, 3, 2],
      [1, 1, 0] ]
f = [10, 10, 8]
x = np.linalg.solve_banded((1, 1), A, f)
print(x)
```

Вывод: [1, 2, 3]

Разложение Холецкого

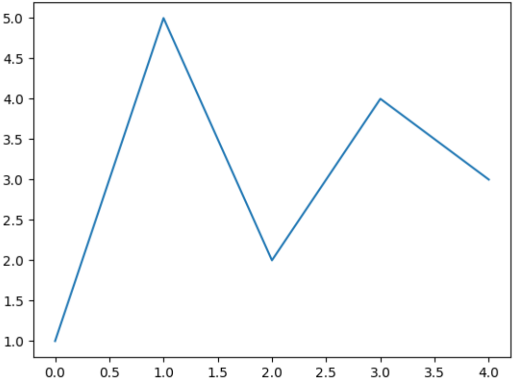
```python
import numpy as np

L = np.array([ [1, 0, 0],
               [2, 1, 0],
               [3, 2, 1] ])
A = L.dot(np.transpose(L))
print(A)

L = np.linalg.cholesky(A, lower=True)
print(L)
```

Упражнения

1. Описать как связаны коэффициенты α_i и β_i с матрицами в LU—разложении.
2. Докажите, что у матрицы с диагональным преобладанием все угловые миноры ненулевые.
3. Описать метод прогонки для блочно трёхдиагональная матрицы, в которой вместо элементов a_i, b_i, c_i находятся подматрицы размера $k \times k$, а вместо f_i — вектор столбец размера k. Основная идея: $x_{i-1} = \alpha_i x_i + \beta_i$, где $\alpha_i \in \mathbb{R}^{k \times k}$, $\beta_i \in \mathbb{R}^k$.

Графики

```python
import matplotlib.pyplot as plt
y = [1, 5, 2, 4, 3]
plt.plot(y)
plt.show()
```



Домашнее задание №1

1. Сравнить решение СЛАУ методом Гаусса библиотечной функцией `np.linalg.solve` со своей реализацией на случайной матрицей с диагональным преобладанием размером 100×100, 200×200 и т.д. Провести несколько экспериментов, пока время счёта меньше 1 сек. Построить графики зависимостей.
2. Сравнить решение СЛАУ методом Холецкого библиотечной функцией `np.linalg.cholesky` со своей реализацией на случайной положительно определённой матрицей с диагональным преобладанием размером 100×100, 200×200 и т.д. Провести несколько экспериментов, пока время счёта меньше 1 сек. Построить графики зависимостей.
3. Сравнить решение СЛАУ методом прогонки библиотечной функцией `np.linalg.solve_banded` со своей реализацией на случайной трехдиагональной матрице с диагональным преобладанием размером 1000×1000, 2000×2000 и т.д. Провести несколько экспериментов, пока время счёта меньше 1 сек. Построить графики зависимостей.