Численные методы линейной алгебры

Метод прогонки

для решения систем линейных алгебраических уравнений (СЛАУ), характеризуемых 3-х диагональной матрицей

Система уравнений $Ax = F$ равносильна соотношению

$$A_i x_{i-1} + B_i x_i + C_i x_{i+1} = F_i\,,\ \ i = 2, \dots, n-1 \qquad (1)$$

$$B_1 x_1 + C_1 x_2 = F_1, \qquad A_n x_{n-1} + B_n x_n = F_n.$$

$$\begin{array}{ccccc} B_1 & C_1 & \dots & 0 & 0 \\ A_2 & B_2 & C_2 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & A_{n-1} & B_{n-1} & C_{n-1} \\ 0 & 0 & \dots & A_n & B_n \end{array} \quad = \quad \begin{array}{c} F_1 \\ F_2 \\ \dots \\ F_{n-1} \\ F_n \end{array}$$

Метод прогонки основывается на предположении, что искомые неизвестные связаны рекуррентным соотношением:

$$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1},\ \text{где}\ i = n-1, n-2, \dots, 1. \qquad (2)$$

Используя это соотношение, выразим x_{i-1} и x_i через x_{i+1} и подставим в уравнение (1):

$$(A_i \alpha_i \alpha_{i+1} + B_i \alpha_{i+1} + C_i)\, x_{i+1} + A_i \alpha_i \beta_{i+1} + A_i \beta_i + B_i \beta_{i+1} - F_i = 0,$$

где F_i — правая часть i-го уравнения. Это соотношение будет выполняться независимо от решения, если потребовать

$$\begin{cases} A_i \alpha_i \alpha_{i+1} + B_i \alpha_{i+1} + C_i = 0 \\ A_i \alpha_i \beta_{i+1} + A_i \beta_i + B_i \beta_{i+1} - F_i = 0 \end{cases}$$

Отсюда следует:

$$\begin{cases} \alpha_{i+1} = \dfrac{-C_i}{A_i \alpha_i + B_i} \\ \beta_{i+1} = \dfrac{F_i - A_i \beta_i}{A_i \alpha_i + B_i} \end{cases}$$

Прямая прогонка

Из первого уравнения получим:

$$\begin{cases} \alpha_2 = \dfrac{-C_1}{B_1} \\ \beta_2 = \dfrac{F_1}{B_1} \end{cases}$$

После нахождения прогоночных коэффициентов α и β, используя уравнение (2), получим решение системы. При этом,

$$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1}, i = n - 1..1$$

$$x_n = \frac{F_n - A_n \beta_n}{B_n + A_n \alpha_n}$$

Обратная прогонка

Количество вычислений $O(n)$.

Итерационные методы (метод Якоби)

для решения СЛАУ с хранимой матрицей

Можно уменьшить/сэкономить память и, возможно, число вычислений, если использовать метод итераций.

$$Ax = f,$$

Преобразуем к виду:

$$x = Bx + b$$

и строим итерац. процесс:

$$\boxed{x^{k+1} = Bx^k + b} \quad k = 0, 1, \ldots$$

найти с пом. MathCAD

Т. Если $\|B\| < 1$ то система имеет единственное решение и итерац. процесс сходится к решению

<u>Метод Якоби</u> (≈ Зейделя. Отличаются матрицей B и $\vec{b}$)

Преобразуем матрицу $A = A_1 + D + A_2$, где

$A_1 = \begin{pmatrix} 0 & 0 & 0 \\ x & 0 & 0 \\ x & x & 0 \end{pmatrix}$ – нижняя Δ с $\emptyset$ диагон., $A_2 = \begin{pmatrix} 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \end{pmatrix}$ – верхняя Δ с $\emptyset$ диагон.

$D = diag(a_{ij}) = \begin{pmatrix} x & & \emptyset \\ & x & \\ \emptyset & & x \end{pmatrix}$ – диагональная.

Тогда система

$$Ax = A_1 x + Dx + A_2 x = f$$

$$Dx = f - A_1 x - A_2 x$$

$$x = -D^{-1}(A_1 + A_2)x + D^{-1}f.$$

Т.о. $\boxed{B = -D^{-1}(A_1 + A_2), \quad b = D^{-1}f.}$

÷ Итерац. метод Якоби удобно записать в форме:

$$D x^{k+1} = -\underbrace{(A_1 + A_2)}_{= A \text{ без диагон. членов}} x^k + f$$

$$A \cdot x = \sum_i a_{ji} x_i$$

– эта система в развёрнутом виде:

$$a_{jj} x_j^{k+1} = -\sum_{\substack{i=1 \\ i \neq j}}^{m} a_{ji} x_i^k + f_j, \qquad j = 1, \dots, m$$

произведение матрицы A на вектор $\vec{x}$, исключая диагональные элементы a_{jj}

Количество вычислений:

$$= \underset{\text{число итераций}}{K} * \Big[\underset{\text{умнож.}}{(m-2)} + \underset{\text{сложен.}}{(m-2)} + \underset{\text{вычитан}}{1} + \underset{\text{делен.}}{1} \Big] * \underset{\text{число компонент } \vec{x}}{m} \approx K 2m^2 = O(m^2)$$

Итого вычислении в методах решения СЛАУ

- Метод Крамера – $O(n!)$,
- Метод Гаусса – $O(n^3)$,
- Метод обратной матрицы – $O(n^4)$,
- Метод прогонки – $O(n)$ для 3-х диагональной матрицы,
- Итерационный метод Якоби – $O(n^2)$