Описание метода решения

LU-разложение — это представление матрицы A в виде произведения двух матриц $A = LU$, где L - нижняя треугольная матрица, а U - верхняя треугольная матрица.

$$\begin{pmatrix} \alpha_{00} & 0 & 0 & 0 \\ \alpha_{10} & \alpha_{11} & 0 & 0 \\ \alpha_{20} & \alpha_{21} & \alpha_{22} & 0 \\ \alpha_{30} & \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} \begin{pmatrix} \beta_{00} & \beta_{01} & \beta_{02} & \beta_{03} \\ 0 & \beta_{11} & \beta_{12} & \beta_{13} \\ 0 & 0 & \beta_{22} & \beta_{23} \\ 0 & 0 & 0 & \beta_{33} \end{pmatrix}$$
$$= \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \qquad (3)$$

Для решения СЛАУ можно использовать следующее разложение,

$$Ax = (LU)x = L(Ux) = b \qquad (4)$$

сначала решив для вектора y,

$$Ly = b \qquad (5)$$

а затем для вектора x

$$Ux = y \qquad (6)$$

Преимущество разложения матрицы в произведение двух треугольных матриц заключается в тривиальности решения СЛАУ для треугольных матриц. Так СЛАУ (5) может быть решено с помощью прямой подстановки

$$y_0 = \frac{b_0}{\alpha_{00}} \qquad (7)$$

$$y_i = \frac{1}{\alpha_{ii}} \left[b_i - \sum_{j=0}^{i-1} \alpha_{ij} y_j \right] \qquad i = 1,2, \dots, N-1 \qquad (8)$$

Аналогично СЛАУ (6) может быть решено с помощью обратной подстановки

$$x_{N-1} = \frac{y_{N-1}}{\beta_{N-1,N-1}} \qquad (9)$$

$$x_i = \frac{1}{\beta_{N-1,N-1}}\left[y_i - \sum_{j=i+1}^{N-1} \beta_{ij} x_j\right] \qquad i = N-2, N-3, \dots, 0 \qquad (10)$$

Заметим, что получив LU-разложение матрицы A, можно решить сколько угодно СЛАУ с различными правыми частями b. Это является преимуществом метода решения СЛАУ с помощью LU-разложения над другими методами.

Найдем L и U, при данной матрице A. Для этого найдем i,j-й элемент матрицы (3).

$$\alpha_{i0}\beta_{0j} + \dots = a_{ij} \qquad (11)$$

Данная сумма зависит от порядка i и j. Существует три случая:

$$i < j: \alpha_{i0}\beta_{0j} + \alpha_{i1}\beta_{1j} + \dots + \alpha_{ii}\beta_{ij} = a_{ij} \qquad (12)$$

$$i = j: \alpha_{i0}\beta_{0j} + \alpha_{i1}\beta_{1j} + \dots + \alpha_{ii}\beta_{jj} = a_{ij} \qquad (13)$$

$$i < j: \alpha_{i0}\beta_{0j} + \alpha_{i1}\beta_{1j} + \dots + \alpha_{ij}\beta_{jj} = a_{ij} \qquad (14)$$

Так как количество неизвестных больше количества уравнений, определим N неизвестных произвольно, присвоив им арбитрарные значения. Действительно, всегда можно принять, что

$$\alpha_{ii} \equiv 1 \quad i = 0, \dots, N-1 \qquad (15)$$

Чтобы решить уравнения (12) – (14) для всех α и β достаточно проделать следующие операции:

1) $\alpha_{ii} = 1, i = 0, \dots, N-1$

2) Для каждого $j = 0,1,2, \dots, N-1$ проделаем следующие две процедуры:

2.1) Во-первых, для $i = 0,1, \dots, j$ найдем β_{ij} как

$$\beta_{ij} = \alpha_{ij} - \sum_{k=0}^{i-1} \alpha_{ik}\beta_{kj} \qquad (16)$$

2.2) Во-вторых, для $i = j+1, j+2, \dots, N-1$ найдем α_{ij} как

$$a_{ij} = \frac{1}{\beta_{jj}}\left(a_{ij} - \sum_{k=0}^{j-1} \alpha_{ik}\beta_{ki} \right) \qquad (17)$$

Можно заметить, что различные значения α и β, которые появляются только с правой части уравнений (16) и (17), точно определены уже к моменту их использования в вычислениях. Кроме того, каждое значение a_{ij} используется при вычислениях лишь единожды. Следовательно, соответствующие α_{ij} и β_{ij} могут быть сохранены на месте хранения соответственного a_{ij}. (Диагональные элементы α_{ii} равные единицы не сохраняются вообще.) В итоге, результатом алгоритма является матрица вида

$$\begin{pmatrix} \beta_{00} & \beta_{01} & \beta_{02} & \beta_{03} \\ \alpha_{10} & \beta_{11} & \beta_{12} & \beta_{13} \\ \alpha_{20} & \alpha_{21} & \beta_{22} & \beta_{23} \\ \alpha_{30} & \alpha_{31} & \alpha_{32} & \beta_{33} \end{pmatrix} \qquad (18)$$

Выбор опорного элемента (а именно выбор значения β_{jj} в уравнении (17)) крайне важен для устойчивости описываемого метода. Поэтому будем производить LU-разложение не с самой исходной матрицей A, а с построчной перестановкой матрицы A. (Если следить за обрабатываемой перестановкой, ее разложение не менее полезно, чем разложение оригинальной матрицы).

Заметим, что в случае $i = j$ уравнение (16) не отличается от уравнения (17), за исключением деления в уравнении (17). В обоих случаях верхний предел сумм равен $k = j - 1 = i - 1$. Следовательно, необязательно сразу производить деление на элемент β_{jj} (который находится на диагонали на текущей итерации). Возможно позже выбрать более оптимальный вариант, когда обработаны все кандидаты. Будем выбирать максимальные по абсолютной величине варианты, а после произведем деление массово. В нашей реализации алгоритм имеет одну особенность: сначала найдем максимальные значения в каждой строке, а затем в процессе поиска максимального опорного элемента

масштабируем множители так, как если бы мы изначально масштабировали все уравнения, чтобы их максимальный коэффициент был равен единице.

Список использованных источников

1. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical Recipes. The Art of Scientific Computing. Third Edition. — Cambridge University Press, 2007. [Электронный ресурс].
URL: http://e-maxx.ru/bookz/files/numerical_recipes.pdf
(дата обращения: 03.04.2024).