

Дисциплина: Численные методы
Лабораторное задание №2

Отчет

Тема: Применение точных методов решения
систем линейных алгебраических уравнений.

Метод Халецкого решения СЛАУ с ленточными матрицами

Выполнила:
студентка 3 курса 62 группы
Пахомова П.В.

Проверила:
старший преподаватель Фролова О.А.

1. Постановка задачи

Решение системы линейных уравнений с разреженными матрицами специального вида

Вариант 1. Метод Халецкого решения СЛАУ с ленточными матрицами.

Входные параметры основной процедуры:

N , L – размерность системы и половина ширины ленты матрицы;

A – массив размерности $N(2L-1)$ – содержащий ленту матрицы исходной системы уравнений;

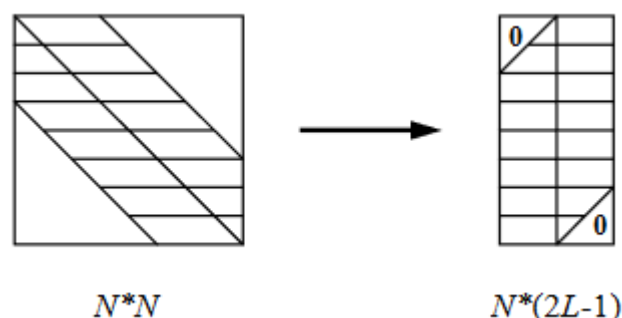
f – вектор правой части системы размерности N .

Выходные параметры основной процедуры:

IER – код завершения;

x – вектор решения размерности N .

Символическое изображение схемы хранения ленточной матрицы:



Требуется написать алгоритм решения системы со стандартной ленточной матрицей.

При численной реализации недопустимо использование матриц размерности $N*N$

Для решения использовать метод Халецкого

Необходимо написать 3 теста для средней относительной погрешности системы.

При записи погрешностей используются 2–3 значащие цифры, не более.

2. Теоретическая часть

Получение ВС разложения матрицы методом Халецкого

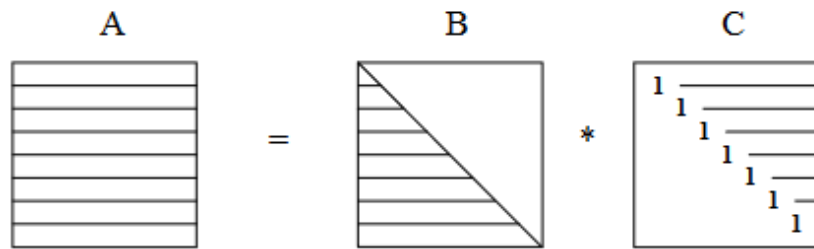


Рис. 1.1.1

Нетрудно видеть, что справедливы формулы:

$$b_{ij} = a_{ij} - \sum_{k=1}^{j-1} b_{ik} c_{kj}, \quad j = 1 \div N, \quad i = j \div N, \quad (1.1.2)$$

$$c_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} b_{ik} c_{kj} \right) / b_{ii}, \quad i = 1 \div N, \quad j = i+1 \div N. \quad (1.1.3)$$

Нахождение решения системы линейных уравнений

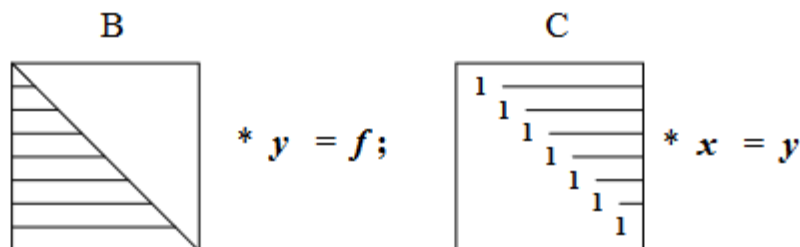


Рис. 1.1.4

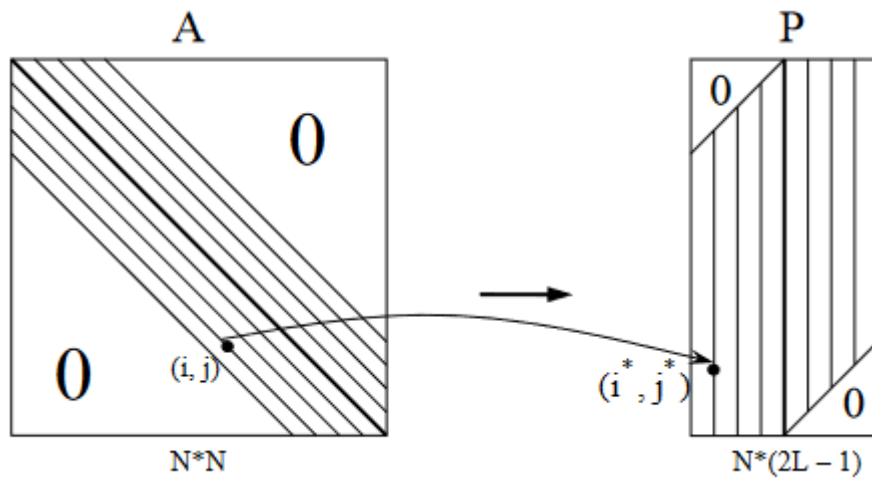
Компоненты векторов x и y определяются по формулам

$$y_i = \left(f_i - \sum_{k=1}^{i-1} b_{ik} y_k \right) / b_{ii}, \quad i = 1 \div N, \quad (1.1.8)$$

$$x_i = y_i - \sum_{k=i+1}^N c_{ik} x_k, \quad i = N \div 1. \quad (1.1.9)$$

Вычисление элементов матриц B , C по формулам (1.1.2), (1.1.3) называется **прямым ходом метода Халецкого**, а определение y , x по формулам (1.1.8), (1.1.9) – **обратным ходом метода Халецкого**.

Преобразование индексов



$$\begin{cases} i^* = i, \\ j^* = j - i + L. \end{cases}$$

Матрица со случайно сгенерированными элементами с очень большой вероятностью *хорошо обусловлена*.

Пусть L – случайная нижнетреугольная матрица с малыми ненулевыми диагональными элементами и поддиагональными элементами умеренной величины, v – аналогичная верхнетреугольная матрица. Тогда $A = LV$ – *плохо обусловленная* матрица.

Матрица Гильберта H с элементами $H_{ij} = 1/(i+j-1)$ очень *плохо обусловлена*:

Размерность H	2	3	4	5	6	7	9	10
$\mu(H)$	$2 \cdot 10^1$	$5 \cdot 10^2$	$2 \cdot 10^4$	$5 \cdot 10^5$	$2 \cdot 10^7$	$5 \cdot 10^8$	$5 \cdot 10^{11}$	$2 \cdot 10^{13}$

1. В отчете должны быть приведены данные о решении систем уравнений с ленточными матрицами порядка 10^1 , 10^2 с диапазоном элементов матриц $-10^1 \div 10^1$ и отношением $L/N \cong 1/10$, $1/L$. Например, если тестируется матрица размерности $N = 40$ (400), то значение L можно взять 4 и 10 (38 и 90). Результаты тестирования помещаются в таблицу.

№ теста	Размерность системы	Отношение L/N	Средняя относительная погрешность решения
1

Минимальное количество строк таблицы равно 4.

О вычислении средней относительной погрешности решения см. замечание о тестировании в задании № 1.

2. В отчете должны быть приведены данные о решении систем уравнений с хорошо обусловленными квадратными матрицами. Хорошо обусловленная система уравнений (см. п. 3 раздела «О составлении численных примеров...») тестируется для двух размерностей порядка 10^1 и двух размерностей порядка 10^2 . Результаты тестирования заносятся в таблицу.

№ теста	Размерность системы	Средняя относительная погрешность решения
1

Минимальное количество строк таблицы равно 4.

3. В отчете должны быть приведены данные о решении систем уравнений с плохо обусловленными матрицами. Плохо обусловленные системы уравнений тестируется для двух размерностей порядка 10^1 . При построении тестовых матриц (см. п. 4 раздела «О составлении численных примеров...») малые диагональные элементы матриц L , U получаются следующим образом. Матрицы L , U заполняются случайно сгенерированными элементами в диапазоне $-10^1 \div 10^1$, а затем диагональные элементы умножаются на 10^{-k} . В отчете должны быть данные для $k = 2, 4, 6$. Результаты вычислительных экспериментов помещаются в таблицу.

№ теста	Порядок k	Размерность системы	Средняя относительная погрешность решения
1

$$\delta_x = \max_i \delta x_i,$$

$$\delta x_i = \begin{cases} \left| \frac{x_i - x_i^*}{x_i^*} \right|, & \text{если } |x_i^*| > q \\ |x_i - x_i^*|, & \text{если } |x_i^*| \leq q, \end{cases}$$

3. Алгоритм

Условия для работы алгоритма:

- При решении в матрице BC диагональный элемент не может быть 0

Данный алгоритм является методом решения системы линейных уравнений.

Переменные

matrix- ленточная матрица, при решении становится матрицей BC

matrixCopy – оригинальная лента

accuracyMatrixLU-BC матрица генерируемая для оценки погрешности плохо обусловленной матрицы

N - Размер обычной матрицы NxN

L - Половина ширины ленты

solved – флаг решена ли матрица

illConditionedMatrices - флаг нужна ли оценка погрешности для плохо обусловленных матриц

x – массив решения

f – правый вектор системы уравнений

q – маленькое число для погрешности

accuracyX – случайно сгенерированный массив решения системы для расчёта погрешности

solutionForAccuracyX – решение сгенерированной системы для расчёта погрешности

accuracyF – полученный через accuracyX вектор системы уравнений

meanRatioRelativeAccuracy - средняя относительная погрешность

accuracyLUF - полученный через accuracyMatrixLU и accuracyX вектор системы уравнений

solutionForAccuracyLUX – решение сгенерированной матрицы BC

meanRatioRelativeAccuracyIllConditionedMatrices - средняя относительная погрешность через accuracyMatrixLU

Шаг 1

Получаем разложение начальной матрицы на верхне и нижнетреугольную

Пусть дана матрица matrix размером $N \times 2L-1$, где N - количество строк, а $2L-1$ - количество столбцов.

1. Проверяем условие: если $matrix[0][L-1]$ равно 0, возвращаем false (противоречие условию)
2. Для каждого i от L до $2L-1$ выполняем: $matrix[0][i] = matrix[0][i] / matrix[0][L-1]$
3. Для каждого i от 1 до N выполняем:
 - Создаем переменные newUpLine, newUpCol, newLeftLine, newLeftCol и sum со значениями $i-1$, L , i , $L-2$ и 0 соответственно.
 - Создаем переменную new_v со значением $L-2$.
 - Для каждого k от 0 до L выполняем:
 - Обновляем newUpLine, newUpCol, newLeftLine и newLeftCol на следующие значения.
 - Если newUpLine находится в пределах от 0 до N , newUpCol находится в пределах от 0 до $2L-2$, newLeftLine находится в пределах от 0 до N и newLeftCol находится в пределах от 0 до $2L-2$, выполняем следующее:
 - Добавляем к sum произведение $matrix[newUpLine][newUpCol]$ и $matrix[newLeftLine][newLeftCol]$.
 - Вычитаем sum из $matrix[i+k][new_v+1]$, если $i+k$ находится в пределах от 0 до N и new_v+1 находится в пределах от 0 до $2L-1$.
 - Уменьшаем new_v на 1.

- Создаем переменную `new_v` со значением `L`.
- Для каждого `k` от 0 до `L-1` выполняем:
 - Обновляем `newUpLine`, `newUpCol`, `newLeftLine` и `newLeftCol` на следующие значения.
 - Если `newUpLine` находится в пределах от 0 до `N`, `newUpCol` находится в пределах от 0 до `2L-2`, `newLeftLine` находится в пределах от 0 до `N` и `newLeftCol` находится в пределах от 0 до `2L-2`, выполняем следующее:
 - Добавляем к `sum` произведение `matrix[newUpLine][newUpCol]` и `matrix[newLeftLine][newLeftCol]`.
 - Если `matrix[i][L-1]` не равно 0, выполняем следующее:
 - Присваиваем `matrix[i][new_v]` значение $(matrix[i][new_v] - sum) / matrix[i][L-1]$.
 - Иначе возвращаем `false` (противоречие условию)
 - Увеличиваем `new_v` на 1.

4. Возвращаем `true`, говоря что удалось построить разложение.

Шаг 2

Проверяем, если удалось построить разложение, переходим к Шагу 3, иначе алгоритм прекращает свою работу из-за противоречия условию

Шаг 3

Находим решение системы уравнений как $Ly=f$, $Ux=y$

1. Решение $Ly=f$:

Пусть дана матрица `matrix` размером $N \times 2L-1$, вектор `y` размером N и вектор `f` размером N .

- Для каждого `i` от 0 до `N-1` выполняем:
 - Создаем переменные `sum` и `accuracySum` со значениями 0.
 - Для каждого `j` от 0 до `L-2` выполняем:
 - Если `i-j-1` находится в пределах от 0 до `N-1`, выполняем следующее:

- Добавляем к sum произведение $y[i-j-1]$ и $matrix[i][L-j-2]$.
- Добавляем к accuracySum произведение $accuracyY[i-j-1]$ и $matrix[i][L-j-2]$.
- Присваиваем $y[i]$ значение $(f[i] - sum) / matrix[i][L-1]$.
- Присваиваем $accuracyY[i]$ значение $(accuracyF[i] - accuracySum) / matrix[i][L-1]$.

2. Решение $Ux=y$:

Пусть дана матрица $matrix$ размером $N \times 2L-1$, вектор x размером N и вектор y размером N .

- Для каждого i от $N-1$ до 0 выполняем:
 - Создаем переменные sum и accuracySum со значениями 0.
 - Для каждого j от 0 до $L-2$ выполняем:
 - Если $i+j+1$ находится в пределах от 0 до $N-1$, выполняем следующее:
 - Добавляем к sum произведение $x[i+j+1]$ и $matrix[i][L+j]$.
 - Добавляем к accuracySum произведение $solutionForAccuracyX[i+j+1]$ и $matrix[i][L+j]$.
 - Присваиваем $x[i]$ значение $y[i] - sum$.
 - Присваиваем $solutionForAccuracyX[i]$ значение $accuracyY[i] - accuracySum$.

Шаг 4

Проверяем что найденное решение корректно, подставим полученные x в $matrixCory$ и проверим что погрешность не превышает допустимую

Пусть даны:

- Матрица $matrixCory$ размером $N \times 2L-1$ - исходная матрица из системы уравнений
- Вектор x размером N - найденное решение системы уравнений $Ux=y$.
- Вектор f размером N - правая часть системы уравнений $Ly=f$.
- Число q - маленькое число, задающее максимально допустимую погрешность при проверке.

Алгоритм проверки выглядит следующим образом:

1. Устанавливаем флаг `check` в значение `true`.
2. Для каждого i от 0 до $N-1$ выполняем:
 - Создаем переменную `sum` со значением 0.
 - Если i находится в диапазоне от 0 до $L-1$ включительно, то:
 - Задаем переменную `count` равной $L+i-1$.
 - Если i находится в диапазоне от L до $N-L-1$ включительно, то:
 - Задаем переменную `count` равной $2L-1$.
 - Если i находится в диапазоне от $N-L$ до $N-1$ включительно, то:
 - Задаем переменную `count` равной $2L-N+i$.
 - Для каждого j от 0 до $\min(\text{count}, N-1)$ выполняем:
 - Если i находится в диапазоне от 0 до $L-1$ включительно, то:
 - Добавляем к `sum` произведение $x[j]$ и `matrixCopy[i][2L-1-count+j]`.
 - Если i находится в диапазоне от L до $N-1$ включительно, то:
 - Добавляем к `sum` произведение $x[i-L+1+j]$ и `matrixCopy[i][j]`.
 - Если $f[i] - \text{sum}$ больше q или меньше $q*(-1)$, то:
 - Задаем флаг `check` в значение `false`.
3. Возвращаем значение флага `check`.

Шаг 5

Если проверка прошла успешно устанавливаем `solved` в `true` и находим среднюю относительную погрешность заданную пользователем
Иначе завершаем алгоритм, говоря что решение найти не удалось

Шаг 6

находим среднюю относительную погрешность (`meanRatioRelativeAccuracy`) для `accuracyF`
и если задано k , то ещё среднюю относительную погрешность (`meanRatioRelativeAccuracyIllConditionedMatrices`) для `accuracyLUF`

Инициализируем переменную `Er2` значением 11.

1. Для каждого i от 0 до $N-1$ выполняем:

- Создаем переменную $er2$ со значением погрешности $solutionForAccuracyX[i] - accuracyX[i]$. (положительное)
 - Если $|accuracyX[i]|$ больше q и $accuracyX[i]$ не равно 0, то:
 - Делим $er2$ на $|accuracyX[i]|$.
 - Находим максимум из $er2$ и записываем в $Er2$
2. Значение $meanRatioRelativeAccuracy$ равно $Er2$.
3. Если указано, что нужно найти погрешность когда матрица плохо обусловлена ($illConditionedMatrices = true$), то:
- Присваиваем $Er2$ значение 11.
 - Для каждого i от 0 до $N-1$ выполняем:
 - Создаем переменную $erLU$ со значением погрешности $(solutionForAccuracyLUX[i] - accuracyX[i]) < 0$. (положительное)
 - Если $|accuracyX[i]|$ больше q и $accuracyX[i]$ не равно 0, то:
 - Делим $erLU$ на $|accuracyX[i]|$.
 - Находим максимум из $erLU$ и записываем в $Er2$
4. Значение $meanRatioRelativeAccuracyIllConditionedMatrices$ равно $Er2$.

Конец алгоритма

Другие функции

построение $accuracyF$

1. Инициализация переменных sum и $count$
 - установка значения sum в 0.
 - если переменная $count$ меньше $2L-1$ и i меньше L , то увеличиваем $count$ на 1.
 - если i больше $N-L$ и N не равно L , то уменьшаем $count$ на 1.
2. Цикл по j от 0 до $count-1$ (или до $N-1$, если $count$ больше N), в котором вычисляем скалярное произведение строки матрицы и вектора-решения:

- если i меньше L , то получаем j -ый элемент вектора-решения $accuracyX$ и $(2L-1-count+j)$ -ый элемент строки матрицы $matrixCory[i]$. Добавляем произведение этих элементов к переменной sum .
 - если i больше или равно L , то получаем $(i-L+1+j)$ -ый элемент вектора-решения $accuracyX$ и j -ый элемент строки матрицы $matrixCory[i]$. Добавляем произведение этих элементов к переменной sum .
3. Присваиваем полученное значение переменной sum элементу $accuracyF[i]$ - i -ому элементу правого вектора системы уравнений.

построение $accuracyLUF$

1. Инициализация переменных

- создание векторов y и $accuracyY$ размерности N .
- заполнение вектора y значениями из вектора $accuracyX$.
- заполнение вектора $accuracyY$ значениями из вектора $accuracyX$.

2. Решение системы $Ux=y$ для вектора x (то есть нахождение вектора y)

- для каждого индекса i вектора x , начиная с последнего элемента, выполняем следующее:
 - инициализируем переменную $accuracySum$ значением 0.
 - для каждого значения j от 0 до $L-2$:
 - если $i+j+1$ меньше N , то добавляем к переменной $accuracySum$ произведение $accuracyX[i+j+1]$ на элемент матрицы LU с индексами i и $L+j$.
 - вычисляем значение элемента $y[i]$ как сумму $accuracyX[i]$ и $accuracySum$.

3. Нахождение правого вектора f , решая систему $Ly=f$

- для каждого индекса i вектора f , начиная с первого элемента, выполняем следующее:
 - инициализируем переменную $accuracySum$ значением 0.
 - для каждого значения j от 0 до $L-2$:

- если $i-j-1$ больше или равно 0, то добавляем к переменной `accuracySum` произведение `accuracyY[i-j-1]` на элемент матрицы LU с индексами i и $L-j-2$.
- вычисляем значение элемента `f[i]` как произведение `accuracyY[i]` на элемент матрицы LU с индексами i и $L-1$, прибавленное к `accuracySum`. Если элемент матрицы LU с индексами i и $L-1$ равен 0, то просто присваиваем `f[i]` значение `accuracySum`.

Тестирование

№Теста	Размерность	Отношение L/N	Средняя относительная погрешность
1	10	1/10	1.37e-16
2	100	1/10	1.06e-12

№Теста	Размерность	Средняя относительная погрешность
1	10	1.23e-14
2	10	3.36e-13
3	100	3.39e-11
4	100	1.02e-12

№Теста	Размерность	Степень k	Средняя относительная погрешность
1	10	2	4.39e+00
2	10	3	1.26e+12
3	10	4	9.57e+18