|  |
| --- |
| Московский авиационный институт  (Национальный исследовательский университет)  Факультет информационных технологий и прикладной математики  Кафедра вычислительной математики и программирования |
|  |

|  |
| --- |
|  |

|  |
| --- |
| Лабораторные работы по курсу «Численные методы» |
| Студент: Устинов А. Э.  Группа: М8О-304Б-20  Преподаватель: Демидова О. Л.  Оценка: |
| Москва 2023 |

**Пункт 1.1**

**Задача:** Решение СЛАУ, нахождение обратной матрицы и определителя матрицы системы

**Метод:** LU-разложение

**Основные применения**: Решение систем алгебраических уравнений, вычисление определителя, вычисление обратной матрицы и др.

LU – разложение матрицы A представляет собой разложение матрицы A в произведение нижней и верхней треугольных матриц (A = LU)

Для экономии памяти, будем записывать обе матрицы в одну, подразумевая, что у матрицы L на главной диагонали единицы.

На первом этапе решается СЛАУ Lz = b . Поскольку матрица системы - нижняя треугольная, решение можно записать в явном виде

На втором этапе решается СЛАУ Ux = z с верхней треугольной матрицей. Здесь, как и на предыдущем этапе, решение представляется в явном виде:

Второй этап эквивалентен обратному ходу методу Гаусса, тогда как первый соответствует преобразованию правой части СЛАУ в процессе прямого хода.

**Особенности LU-разложения**

1. Легко вычисляется определитель матрицы:
2. Однажды найдя LU-разложение для матрицы мы можем очень быстро решать системы линейных алгебраических уравнений с различной правой частью.

Пусть

Тогда

Так как [](http://www.codecogs.com/eqnedit.php?latex=L) — нижнетреугольная матрица, то очень легко находим   
Решаем

Легко находим , так как U — верхнетреугольная матрица

1. Сложность алгоритма:  
   LU-разложение:

Последующее решение систем:

**Пункт 1.2**

**Задача:** Решение СЛАУ с трехдиагональной матрицей

**Метод:** Метод прогонки

**Основные применения**: Эффективный метод решения СЛАУ с трех - диагональными матрицами, возникающих при конечно-разностной аппроксимации задач для обыкновенных дифференциальных уравнений (ОДУ) и уравнений в частных производных второго порядка и является частным случаем метода Гаусса.

Решения ищутся в виде:

Прямой ход предназначен для нахождения прогоночных коэффициентов

Обратный ход предназначен для нахождения вектора

X = () в соответствии с найденными

Для устойчивости метода прогонки (1.4)-(1.7) достаточно выполнение следующих условий:

**Особенности метода прогонки**

Метода можно решать только специфические системы, имеющие не более трех неизвестных в каждой строке.

Сложность O(N)

**Пункт 1.3**

**Задача:** Решение СЛАУ

**Методы:** Метод простых итераций и метод Зейделя

**Основные применения**: Для решения СЛАУ с разреженными матрицами.

1. **Метод простых итераций**

За нулевое приближение принимается вектор правых частей. В следующем приближении, для вычисления X подставляются значение вектора X на предыдущем приближении. Итерации повторяются до достижения заданной точности

Метод простых итераций сходится к единственному решению при любом начальном приближении , если какая-либо норма матрицы эквивалентной системы меньше единицы

Для сходимости итерационного процесса необходимо и достаточно, чтобы спектр матрицы α эквивалентной системы лежал внутри круга с радиусом, равным единице.

При выполнении достаточного условия сходимости оценка погрешности решения на k - ой итерации дается выражением:

Процесс итераций останавливается при выполнении условия , где - задаваемая вычислителем точность.

Поскольку является только достаточным (не необходимым) условием сходимости метода простых итераций, то итерационный процесс может сходиться и в случае, если оно не выполнено. Тогда критерием окончания итераций может служить неравенство

**Особенности метода простых итераций**

Удобен при распараллеливании, так как приближения считаются построчно независимо друг от друга.

Можно задать точность приближения

В вычислительном процессе участвуют только произведения матрицы на вектор, что позволяет работать только с ненулевыми элементами матрицы, значительно упрощая процесс хранения и обработки матриц, по сравнению с методом Гаусса

1. **Метод Зейделя**

Метод простых итераций довольно медленно сходится. Для его ускорения существует метод Зейделя, заключающийся в том, что при вычислении компонента вектора неизвестных на (k+1)-й итерации используются , уже вычисленные на (k+1)-й итерации. Значения остальных компонент берутся из предыдущей итерации. Значение остальных компонент берутся из предыдущей итерации.

Оценка погрешности вычисляется аналогично

Первое приближение – вектор правых частей

**Особенности метода Зейделя**

Сходимость быстрее, чем у метода простых итераций, но невозможно распараллеливаливание.

**Пункт 1.4**

**Задача:** Нахождение собственных значений и собственных векторов симметричной матрицы

**Методы:** Метод вращений

**Основные применения**: Нахождение собственных значений и собственных векторов симметричной матрицы

1. Находим максимальный недиагональный элемент матрицы
2. Находим соответствующую этому элементу матрицу вращения
3. Вычисляем матрицу

В качестве критерия окончания итерационного процесса используется условие малости суммы квадратов внедиагональных элементов:

Если

продолжается. Если , то итерационный процесс останавливается, и в качестве искомых собственных значений принимаются

**Пункт 1.5**

**Задача:** Нахождение собственных значений

**Методы:** Метод QR разложений

**Основные применения**: Нахождение собственных значений

1. Ищем матрицу Хаусхолдера, соответствующую вектору:

по формуле

Изображение выглядит как текст, антенна

Автоматически созданное описание

1. Проделываем аналогичную процедуру для всех векторов матрицы A
2. На каждой итерации производим умножение
3. R =

После проделывания этой операции, считаем

Повторяем вышеназванную последовательность действий, пока для каждого из диагональных значений матрицы не будет выполнено одно из двух условий:

или

Причем, в случае первого условия, проверим диагональную матрицу 2x2, решив квадратное уравнение:

**Особенности метода QR разложений**

Существенным недостатком рассмотренного выше алгоритма является большое число операций необходимое для - факторизации матрицы на каждой итерации.

Преимущество метода в том, что с помощью него можно находить комплекснозначные собственные значения