1 Матричные нормы

1.1 Алгоритм вычисления основных матричных норм

Входные данные: Матрица A размером $m \times n$

Выходные данные: Значения различных норм матрицы

Algorithm 1 Норма Фробениуса (F-норма)

```
sum = 0
for i = 1 to m do
    for j = 1 to n do
        sum = sum + |A[i, j]|^2
    end for
end for
||A||_F = sqrt(sum)
```

Algorithm 2 Максимальная норма (∞-норма)

```
max_row_sum = 0
for i = 1 to m do
    row_sum = 0
    for j = 1 to n do
        row_sum = row_sum + |A[i, j]|
    end for
    if row_sum > max_row_sum then
        max_row_sum = row_sum
    end if
end for
||A||_∞ = max_row_sum
```

Algorithm 3 Первая норма (1-норма)

```
max_col_sum = 0
for j = 1 to n do
    col_sum = 0
    for i = 1 to m do
        col_sum = col_sum + |A[i, j]|
    end for
    if col_sum > max_col_sum then
        max_col_sum = col_sum
    end if
end for
||A||_1 = max_col_sum
```

2 LU разложение

2.1 Алгоритм LU разложения без выбора главного элемента

Входные данные: Квадратная матрица A размером $n \times n$
Выходные данные: Нижнетреугольная матрица L и верхнетреугольная матрица U

```
Инициализация
L = I_{n×n} (единичная матрица)
U = A (копия матрицы A)
for k = 1 to n − 1 do
    if U[k, k] = 0 then
        Вывести "Ошибка: деление на ноль"
        Завершить
    end if
    for i = k + 1 to n do
        L[i, k] = U[i, k]/U[k, k]
        for j = k to n do
            U[i, j] = U[i, j] − L[i, k] × U[k, j]
        end for
    end for
end for
return L, U
```

Проверка: $A = L \times U$

3 Метод Гаусса-Зейделя в координатной форме

Для системы уравнений $Ax = b$
Система уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

Проверка диагонального преобладания
for $i = 1$ to n **do**
  if $|a_{ii}| \leq \sum_{j \neq i} |a_{ij}|$ **then**
    Предупреждение о возможной расходимости
  end if
end for
Инициализация
$x^{(0)} = [0, 0, \ldots, 0]$ (начальное приближение)
$\varepsilon =$ заданная точность
$max\_iter =$ максимальное число итераций
$k = 0$ (номер итерации)
while $k < max\_iter$ **do**
  $k = k + 1$
  for $i = 1$ to n **do**
    $sum1 = 0$
    for $j = 1$ to $i - 1$ **do**
      $sum1 = sum1 + a_{ij} \times x_j^{(k)}$
    end for
    $sum2 = 0$
    for $j = i + 1$ to n **do**
      $sum2 = sum2 + a_{ij} \times x_j^{(k-1)}$
    end for
    $x_i^{(k)} = (b_i - sum1 - sum2)/a_{ii}$
  end for
  if $\|x^{(k)} - x^{(k-1)}\| < \varepsilon$ **then**
    Решение найдено: $x = x^{(k)}$
    Завершить

end if
end while
if $k = max\_iter$ **then**
Предупреждение: "Достигнуто максимальное число итераций"
end if

3.1 Координатная форма для системы 3×3:

$$\begin{cases} x_1^{(k)} = (b_1 - a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)})/a_{11} \\ x_2^{(k)} = (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)})/a_{22} \\ x_3^{(k)} = (b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)})/a_{33} \end{cases}$$

4 Метод Ньютона для решения нелинейного уравнения

Для уравнения $f(x) = 0$
Входные данные: Функция $f(x)$, её производная $f'(x)$, начальное приближение x_0
Выходные данные: Приближенное решение уравнения

Инициализация
$x_0 =$ начальное приближение
$\varepsilon =$ заданная точность
$max\_iter =$ максимальное число итераций
$k = 0$
Проверка начального приближения
if $f'(x_0) = 0$ **then**
    Выбрать другое начальное приближение
end if
while $k < max\_iter$ **do**
    $k = k + 1$
    if $f'(x_{k-1}) = 0$ **then**
        Вывести "Ошибка: производная равна нулю"
        Завершить
    end if
    $x_k = x_{k-1} - f(x_{k-1})/f'(x_{k-1})$
    if $|x_k - x_{k-1}| < \varepsilon$ ИЛИ $|f(x_k)| < \varepsilon$ **then**
        Решение найдено: $x = x_k$
        Завершить
    end if
end while
if $k = max\_iter$ **then**
    Предупреждение: "Достигнуто максимальное число итераций"
end if

4.1 Геометрическая интерпретация:

На каждой итерации строится касательная к графику функции в точке $(x_{k-1}, f(x_{k-1}))$, и находится точка пересечения этой касательной с осью x.

4.2 Условия сходимости:

- $f(x)$ и $f'(x)$ непрерывны в окрестности корня
- $f'(x) \neq 0$ в окрестности корня
- Начальное приближение достаточно близко к корню

4.3 Модифицированный метод Ньютона (для кратных корней):

$$x_k = x_{k-1} - m \cdot \frac{f(x_{k-1})}{f'(x_{k-1})}$$

где m - кратность корня.

5 Интерполяционный полином Лагранжа

Для табличных значений $(x_0, f_0), (x_1, f_1), \ldots, (x_n, f_n)$

1: Записать общую формулу полинома Лагранжа:

$$L(x) = \sum_{i=0}^{n} f_i \cdot l_i(x)$$

2: Вычислить базисные полиномы Лагранжа:

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

3: Для каждого узла i:

4: Записать произведение $(x - x_j)$ для всех $j \neq i$ в числителе

5: Записать произведение $(x_i - x_j)$ для всех $j \neq i$ в знаменателе

6: Умножить каждый $l_i(x)$ на соответствующее значение f_i

7: Сложить все слагаемые и упростить полином

5.1 Пример выполнения:

Для 3 узлов $(x_0, f_0), (x_1, f_1), (x_2, f_2)$:

$$L(x) = f_0 \cdot \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f_1 \cdot \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + f_2 \cdot \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

6 Интерполяционный полином Ньютона

1: Составить таблицу разделенных разностей:

- 0-го порядка: $f[x_i] = f_i$
- 1-го порядка: $f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$
- k-го порядка: $f[x_i, \ldots, x_{i+k}] = \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}$

2: Записать полином Ньютона:

$$P(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots$$

$$+ f[x_0, \ldots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1})$$

3: Построить таблицу разделенных разностей:

$$\begin{array}{c|lllll}
x_0 & f_0 & & & \\
 & & f[x_0, x_1] & & \\
x_1 & f_1 & & f[x_0, x_1, x_2] & \\
 & & f[x_1, x_2] & & f[x_0, x_1, x_2, x_3] \\
x_2 & f_2 & & f[x_1, x_2, x_3] & \\
 & & f[x_2, x_3] & & \\
x_3 & f_3 & & &
\end{array}$$

4: Взять коэффициенты из первой строки таблицы и записать итоговый полином

7 Априорная оценка погрешности интерполяции

1: Записать формулу погрешности интерполяции:

$$R(x) = f(x) - P(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot \omega(x)$$

где $\xi \in [\min(x_0, \dots, x_n, x), \max(x_0, \dots, x_n, x)]$

2: Вычислить $\omega(x)$ - произведение отклонений от узлов:

$$\omega(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$$

3: Найти максимум $|f^{(n+1)}(x)|$ на интервале интерполяции:

$$M = \max |f^{(n+1)}(x)| \text{ на } [a, b]$$

4: Записать априорную оценку:

$$|R(x)| \leq \frac{M}{(n+1)!} \cdot |\omega(x)|$$

5: Для конкретной точки x подставить значения и вычислить числовую оценку

7.1 Особые случаи:

- Для равноотстоящих узлов с шагом h: $|\omega(x)| \leq h^{n+1} \cdot n!/4$ (при n четном)
- Для интерполяции на отрезке $[a, b]$: использовать чебышевские узлы для минимизации $\omega(x)$

8 Квадратурная формула методом неопределенных коэффициентов

8.1 Алгоритм для 1 узла:

1: Записать квадратурную формулу:

$$\int_a^b f(x)dx \approx A_0 \cdot f(x_0)$$

2: Потребовать точности для многочленов степени 0:
3: Для $f(x) = 1$: $\int_a^b 1dx = A_0 \cdot 1$
4: Получаем: $b - a = A_0$
5: Следовательно: $A_0 = b - a$
6: Выбрать оптимальный узел $x_0 = \frac{a+b}{2}$
7: Итоговая формула (прямоугольников):

$$\int_a^b f(x)dx \approx (b-a) \cdot f\left(\frac{a+b}{2}\right)$$

8.2 Алгоритм для 2 узлов:

1: Записать квадратурную формулу:

$$\int_a^b f(x)dx \approx A_0 \cdot f(x_0) + A_1 \cdot f(x_1)$$

2: Потребовать точности для многочленов степени 0, 1, 2, 3:

- Для $f(x) = 1$:

$$\int_a^b 1dx = A_0 + A_1 \Rightarrow b - a = A_0 + A_1 \quad (1)$$

- Для $f(x) = x$:

$$\int_a^b xdx = A_0x_0 + A_1x_1 \Rightarrow \frac{b^2 - a^2}{2} = A_0x_0 + A_1x_1 \quad (2)$$

- Для $f(x) = x^2$:

$$\int_a^b x^2dx = A_0x_0^2 + A_1x_1^2 \Rightarrow \frac{b^3 - a^3}{3} = A_0x_0^2 + A_1x_1^2 \quad (3)$$

- Для $f(x) = x^3$:

$$\int_a^b x^3 dx = A_0 x_0^3 + A_1 x_1^3 \Rightarrow \frac{b^4 - a^4}{4} = A_0 x_0^3 + A_1 x_1^3 \quad (4)$$

Решить систему уравнений (1)-(4) относительно A_0, A_1, x_0, x_1

Решение для отрезка $[-1, 1]$ (формула Гаусса):

$$x_0 = -\frac{1}{\sqrt{3}}, \quad x_1 = \frac{1}{\sqrt{3}}, \quad A_0 = A_1 = 1$$

Для произвольного отрезка $[a, b]$:

Выполнить замену переменных: $t = \frac{2x-a-b}{b-a}$

Узлы: $x_0 = \frac{a+b}{2} - \frac{b-a}{2\sqrt{3}}, \quad x_1 = \frac{a+b}{2} + \frac{b-a}{2\sqrt{3}}$

Веса: $A_0 = A_1 = \frac{b-a}{2}$

Итоговая формула Гаусса:

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\left[f\left(\frac{a+b}{2} - \frac{b-a}{2\sqrt{3}}\right) + f\left(\frac{a+b}{2} + \frac{b-a}{2\sqrt{3}}\right)\right]$$

9 Численное интегрирование

9.1 Формула трапеций

Цель: Вычислить определенный интеграл $\int_a^b f(x)dx$

1: Задать границы интегрирования $[a, b]$ и количество разбиений n

2: Вычислить шаг: $h = \frac{b-a}{n}$

3: Вычислить узлы: $x_i = a + i \cdot h$, где $i = 0, 1, \ldots, n$

4: Применить формулу:

$$I \approx \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right]$$

9.2 Формула Симпсона (1/3)

1: Убедиться, что n четное

2: Вычислить шаг: $h = \frac{b-a}{n}$

3: Применить формулу:

$$I \approx \frac{h}{3}\left[f(x_0) + 4\sum_{\text{нечетные } i} f(x_i) + 2\sum_{\text{четные } i} f(x_i) + f(x_n)\right]$$

10 Метод Рунге-Кутты 4-го порядка

Цель: Решить дифференциальное уравнение $y' = f(x, y)$ с начальным условием $y(x_0) = y_0$

Задать начальные условия: $x_0 = 0, y_0 = y(0)$, конечная точка $x = 1$
Выбрать шаг h (например, $h = 0.1$ для 10 шагов)
for каждый шаг **do**
    Вычислить коэффициенты:
    $k_1 = h \cdot f(x, y)$
    $k_2 = h \cdot f(x + h/2, y + k_1/2)$
    $k_3 = h \cdot f(x + h/2, y + k_2/2)$
    $k_4 = h \cdot f(x + h, y + k_3)$
    Обновить значения:
    $y = y + \frac{k_1+2k_2+2k_3+k_4}{6}$
    $x = x + h$
end for

11 QR-разложение матрицы 2×2 через отражение Хаусхолдера

Цель: Разложить матрицу $A = QR$, где Q - ортогональная, R - верхнетреугольная

Для матрицы $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$:

Взять первый столбец: $v_1 = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix}$

Вычислить норму: $\alpha = \pm\|v_1\|_2 = \pm\sqrt{a_{11}^2 + a_{21}^2}$

Выбрать знак α противоположный знаку a_{11} для численной устойчивости

Вычислить вектор отражения: $u = v_1 - \alpha e_1$, где $e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Нормировать: $w = u/\|u\|_2$

Матрица отражения: $H = I - 2ww^T$

Применить отражение:

$R = H \cdot A$

$Q = H^T$ (поскольку H симметричная и ортогональная)

12 Интерполяционный полином Лагранжа с узлами Чебышева

Цель: Построить полином 2-й степени для интерполяции функции на отрезке $[a, b]$

Построение узлов Чебышева на отрезке $[a, b]$:
Для полинома степени $n = 2$ нужно $n + 1 = 3$ узла
Узлы Чебышева на $[-1, 1]$: $x_i = \cos\left(\frac{(2i-1)\pi}{2n+2}\right), i = 1, 2, 3$
Преобразование на $[a, b]$: $t_i = \frac{a+b}{2} + \frac{b-a}{2} \cdot x_i$
Вычисление значений функции в узлах:
$y_i = f(t_i)$ для $i = 1, 2, 3$
Построение полинома Лагранжа:
$P(x) = \sum_{i=1}^{3} y_i \cdot L_i(x)$
где $L_i(x) = \prod_{j \neq i} \frac{x-t_j}{t_i-t_j}$