

- Инфо
- Лекция 1
 - Корень числа
 - Общие правила численного метода
 - Floating numbers
 - Погрешности
 - Неустойчивость
- Семинар 1
 - Векторные и матричные нормы
- Лекция 2
 - Численное решение СЛАУ
 - Оценки погрешности
 - Нормы
 - Обусловленность матрицы
 - Прямые методы решения СЛАУ
- Лекция 3
 - Продолжаем про линейные методы
 - Метод Гаусса и LU-разложение
 - Уточнение решения
 - Ещё немного об LU-разложении
 - Метод квадратного корня (Холецкого)
 - Итерационные методы решения СЛАУ
 - Метод простой итерации
 - Критерий сходимости МПИ
 - Метод Якоби
- Лекция 4
 - Продолжаем про итерационные методы
 - Метод Эйделя
 - Квадратичный функционал
 - Задача минимизации
 - Безусловный экстремум функции
 - Минимизация функции одной переменной
- Лекция 5
 - Продолжаем про минимизацию функций одной переменной
 - Итерационные методы минимизации функций
 - Метод покоординатного спуска (МПС)
 - Градиентный спуск
 - Выпуклые функции и множества
 - Минимизация линейных функций
 - Интерполяция функций
 - Кусочно-линейная интерполяция
- Лекция 6
 - Продолжаем про интерполяцию
 - Интерполяция через обобщённый полином
 - Интерполяционный полином Лагранжа
 - Конечные и разделённые разности
 - Полином Ньютона
 - Обусловленность интерполяции
- Лекция 7
 - Численное интегрирование
 - Квадратурные формулы

- Формула прямоугольников
 - Формула трапеции
 - Формула Симпсона
 - Составные квадратуры
 - Интерполяционная квадратура
 - Полином Лежандра для выбора сетки и формула Гаусса
- Лекция 8
 - Численное интегрирование
 - Усложнённая формула Гаусса
 - Метод Монте-Карло
 - Задача Коши для ОДУ
 - Формула Эйлера
- Лекция 9
 - Продолжение задачи Коши для ОДУ
 - Метод Рунге-Кутты второго порядка точности (предиктор-корректор)
 - Усовершенствованный метод Эйлера
 - Метод Рунге-Кутты четвёртого порядка точности
 - Метод Адамса
 - Процедура уточнения (правило Рунге)
 - Приближённое значение по Ричардсону
 - ОДУ второго порядка
- Лекция 10
 - Продолжаем ОДУ второго порядка
 - Разностный метод
 - Основная теорема теории разностных схем
- Лекция 11
 - Продолжаем ОДУ второго порядка
 - Метод Ритца
 - Метод Галёркина
 - Методы конечных элементов
- Лекция 12
 - Сглаживание функции
 - Метод наименьших квадратов
- Лекция 13
 - Дискретное преобразование Фурье
- Лекция 14
 - Вычисление собственных значений
 - Теорема Гершгорина
 - Отношение Рэля

Инфо

Лектор - Васкевич <3

Семинарист - Левыкин Александр Иванович

Лекция 1

Классификация вычислительных алгоритмов:

- Матричное вычисление (вычислительные методы линала)
- Задачи оптимизации, решение нелинейных уравнений и их систем
- Интерполирование и численное дифференцирование
- Численное приближение не-полиномов
- Численное интегрирование

- Численные и разностные методы для обыкновенных диффузов

Численный метод - числа и арифметические действия, расположенные в определённом порядке

Корень числа

Простейший численный метод - нахождение положительного квадратного корня числа a . Используем рекуррентную формулу:

$$x_n = \frac{1}{2} \left(x_{n-1} + \frac{a}{x_{n-1}} \right)$$

Доказывается через выведение погрешности эpsilon (с условием $\frac{x_n}{\sqrt{a}} = 1 + \epsilon_n$), которая будет убывать быстрее геометрической прогрессии с шагом $1/2$, благодаря чему $\lim_{n \rightarrow \infty} x_n = \sqrt{a}$

И... Реально работает! Вот код на Питоне:

```
a = 2
x = 1.0
for _ in range(1000):
    x = (x + a / x) / 2.0
print(x)
```

Общие правила численного метода

1. Исходная непрерывная задача замещается дискретной задачей (см. пример выше, где мы избавились от непрерывной функции корня)
2. Добавляется количество шагов (параметр N , также называется **дискретным временем**)
3. Увеличивая дискретное время, мы приблизим результат численного метода к результату непрерывного сколь угодно близко (на третий принцип порой забывают и даже очень часто - в зависимости от решаемой задачи)

Floating numbers

Описывается множеством p, t, L, U :

- p - основание системы счисления
- t - разрядность числа
- L - нижняя граница порядка
- U - верхняя граница порядка

Нормализованные числа (точка после первого значимого числа) с плавающей точкой обозначаются как множество F_1 , оно обладает следующими свойствами:

- Числа в нём распределены неравномерно
- В частности, чем больше модуль числа, тем больше расстояние до соседей
- Между нулём и минимальным числом имеется просвет, который в k раз больше следующего числа

По стандарту IEEE 754 тип данных float имеет такое распределение:

- 1 бит - знак
- 8 бит - экспонента
- 23 бита - мантисса

На double мы тут, видимо, забили... А, нет, слегка упомянули. ну в общем, это всё мы итак знаем

Очень часто числа округляются. Есть несколько способов:

- Вниз - $R_d(x)$ - отбрасываем избыточную для формата часть мантиссы
- Вверх - $R_u(x)$ - стандартные правила округления
- Чётное - $R_e(x)$ - отличается от округления вверх только в случае, если x находится ровно между двумя ЧПТ, тогда из них берётся то число, мантисса которого заканчивается на чётное число

Погрешности

Для вещественного числа a и его приближения a^*

Абсолютная погрешность - положительная величина $\Delta(a^*) \geq |a^* - a|$

- Значащую цифру в приближении числа называют **верной**, если $\Delta(a^*)$ не превосходит половины единицы этого разряда (читай как $0.5 * 10^{-t}$)

Относительная погрешность: $\delta(a^*) \geq \frac{|a^* - a|}{|a^*|}$

Неустойчивость

Вычислительный алгоритм называется **неустойчивым**, если в ходе него происходит одно из событий:

- Потеря значащих цифр
- Переполнение ЧПТ

Обусловленность - чувствительность задачи к начальным условиям. Если погрешность результата задачи существенно больше погрешности входных данных, то задача будет **плохо обусловленной**

Семинар 1

Векторные и матричные нормы

Нормы $\|\cdot\|_a, \|\cdot\|_b$ эквивалентны, если

$$\forall X : \exists C_1, C_2 : C_1 \|X\|_b \leq \|X\|_a \leq C_2 \|X\|_b$$

Лекция 2

Численное решение СЛАУ

Разумеется, СЛАУ можно решить, в случае невырожденной матрицы (а иначе она и не решится) методом Крамера, однако считать определитель матрицы - задача не самая лёгкая

При этом в целом на компе при $n \leq 10^6$ определитель считается за разумное время, однако есть и более эффективные методы: прямой и итерационный

Однако для этих численных методов возникает погрешность. Особенно высока она будет в плохо обусловленных системах. Таким образом становится актуальной оценка обусловленности матрицы

Оценки погрешности

Нормы

Все нормы линейны и удовлетворяют неравенству треугольника

Стандартные нормы:

- $\|\vec{u}\|_\infty = \max_{1 \leq i \leq n} |u_i|$ - кубическая
- $\|\vec{u}\|_1 = \sum_{i=1}^n |u_i|$ - октаэдрическая
- $\|\vec{u}\|_2 = \sqrt{\sum_{i=1}^n |u_i|^2}$ - евклидова

Пространство \mathbb{R}^n снабжённое одной из норм будем обозначать как L^n

Линейное преобразование:

$$\vec{v} = A\vec{u}$$

Теперь определим норму для матрицы A :

$$\forall \|\cdot\| : \|A\| = \sup_{\vec{u} \neq 0} \frac{\|A\vec{u}\|}{\|\vec{u}\|}$$

Эта норма также удовлетворяет:

- неравенству треугольника
- однородна ($\|\lambda A\| = |\lambda| \|A\|$)
- неравенству треугольника 2.0 ($\|AB\| \leq \|A\| \|B\|$) (а также $\|A\vec{u}\| \leq \|A\| \|\vec{u}\|$)

Таким образом, из стандартных норм векторов получаем такие нормы для матриц:

- $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$ - ищем максимальную по сумме строку матрицы
- $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$ - почти то же, что и первое, но теперь ищем максимальную сумму по столбцам
- $\|A\|_2 = \sqrt{\max_{1 \leq i \leq n} \lambda_i(A^*A)}$
 - $\lambda_i(A^*A)$ - это с.з. матрицы A^*A , при этом $A^* = A^T$
 - Если A симметричная, тогда $\lambda_i(A^*A) = \lambda_i(A^2) = |\lambda_i(A)|^2$

Напоминка: скалярное произведение $(\vec{u}, \vec{v}) = \sum_{i=1}^n u_i v_i$

Из определения нормы матрицы следует, что если норма матрицы согласована с нормой векторов, то для единичной матрицы норма будет 1

Обусловленность матрицы

O. $\mu(A) = \|A\| \|A^{-1}\|$ называется **числом обусловленности** матрицы. Несложно заметить, что выбор нормы влияет на числа обусловленности. Также обозначается как $cond(A)$

T. Если возмущение матрицы ΔA таково, что:

$$\mu(A) \frac{\|\Delta A\|}{\|A\|} < 1$$

Тогда возмущение вектора-решения $\Delta \vec{u}$ удовлетворяет оценке:

$$\frac{\Delta \vec{u}}{\vec{u}} \leq \frac{\mu(A)}{1 - \mu(A) \frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta \vec{f}\|}{\|\vec{f}\|} + \frac{\|\Delta A\|}{\|A\|} \right)$$

Доказывается через преобразования последней оценки при помощи $A\vec{u} = \vec{f}$ и неравенства треугольника

C. 1 При $\Delta A \approx 0$, получаем оценку погрешности только правой части:

$$\frac{\Delta \vec{u}}{\vec{u}} \leq \mu(A) \frac{\|\Delta \vec{f}\|}{\|\vec{f}\|}$$

C. 2 Если $\Delta A \Delta \vec{u} \approx 0$, то имеет место такая оценка:

$$\frac{\|\Delta \vec{u}\|}{\|\vec{u}\|} \leq \frac{\|\Delta \vec{f}\|}{\|\vec{f}\|} + \frac{\|\Delta A\|}{\|A\|}$$

В силу неравенства треугольника 2.0 и $\|AA^{-1}\| = \|E\| = 1$ получаем, что $\mu(A) \geq 1$

Если число обусловленности не превышает 10, мы говорим о слабом влиянии ошибок входных данных. При $\mu \gg 100$ система плохо обусловлена

См. пример во второй лекции на стр. 29

Замечание 1 - определитель матрицы может быть большим, а число обусловленности - малым (пример - диагональная матрица с $\epsilon > 0$: $\det D = \epsilon^n$, $\mu(D) = 1$) и наоборот (верхнетреугольная матрица с 1 на диагонали и -1 над диагональную имеет $\det A = 1$, но $\mu(A) = n2^{n-1}$)

Замечание 2 - для невырожденной матрицы A и любой нормы будет иметь место оценка снизу:

$$\mu(A) \geq \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|}$$

Прямые методы решения СЛАУ

Для диагональной невырожденной матрицы СЛАУ распадается на n простейших уравнений и решается за n делений

Для верхнетреугольной матрицы идём от конца к началу, вычисляя один компонент за другим. Число операций - $O(n^2)$ (**обратный ход метода Гаусса**)

Для неструктурированной матрицы сначала делаем приведение к треугольному виду, а затем как раньше. Называется алгоритм - **метод исключения Гаусса** (реализовывали на первом курсе на императивке, так что описывать его считаю занятием не самым интересным (при этом там мы даже не только квадратные рассматривали))

- Прямой ход (превращение матрицы в верхнетреугольную) занимает $\approx \frac{3}{2}n^3$ операций

Лекция 3

Продолжаем про линейные методы

Метод Гаусса и LU-разложение

LU-разложение матрицы A - разложение, получаемое в ходе прямого хода метода Гаусса, где:

- A - исходная матрица
- U - итоговая верхнетреугольная матрица
- L - нижнетреугольная матрица коэффициентов, используемых для приведения матрицы A к верхнетреугольному виду (по факту представима через произведение $n - 1$ отдельных матриц-столбцов)
 - $k < j : l_{kj} = 0$
 - $l_{jj} = 1$
 - $k > j : l_{kj} = \frac{a_{kj}^{(j-1)}}{a_{jj}^{(j-1)}}$

На практике в прямом ходе метода Гаусса на каждом шаге делается выбор строки с максимальным по модулю элементом для данного столбца. Потом эта строка меняется местами с текущей строкой, а столбец ниже зануляется

Проблем с методом не будет, если у матрицы есть **диагональное преобладание**:

$$\forall i : |a_{ii}| \geq \sum_{j=1, j \neq i}^N |a_{ij}|$$

Уточнение решения

- Находим **вектор невязки** $\vec{r}_1 = \vec{f} - A\vec{u}_1$
 - Далее решаем систему $A\vec{e}_1 = \vec{r}_1$
 - Уточнённое решение $\vec{u}_2 = \vec{u}_1 + \vec{e}_1$
 - Можно продолжать итеративно
-

Ещё немного об LU-разложении

Зная LU разложение, можно провести такие модификации:

$$\begin{aligned} A\vec{u} &= \vec{f} \\ LU\vec{u} &= \vec{f} \\ U\vec{u} = \vec{v} &\Rightarrow L\vec{v} = \vec{f} \end{aligned}$$

Решение СЛАУ $L\vec{v} = \vec{f}$ с нижнетреугольной матрицей может быть в некоторых случаях найдено проще, чем по классическому алгоритму Гаусса, однако... по сути мы не получаем в таком случае никакой экономии вычислительных мощностей, а будто бы даже наоборот

Разложение на верхне и нижнетреугольную матрицу достаточно просто делается из уравнения $A = LU$. По следующим формулам:

- $u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$ (от элемента A в строке отнимаем произведение в столбце матрицы U над ним на часть строки такой же длины в матрице L)
- $l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}}{u_{jj}}$ (от элемента из A в столбце отнимаем произведение строки из матрицы L слева от него на столбец над ним из матрицы U такой же длины. Результат делим диагональный элемент из U)
- Сначала пишем первую строку матрицы U (идентична строке матрицы A)
- Затем вычисляем столбец матрицы L
- И далее поочерёдно считаем строку и столбец U и L соответственно

Метод квадратного корня (Холецкого)

Если матрица **симметричная и положительно определённая**, то найдётся такое LU-разложение, что $LL^T = A$, причём:

- $l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{j-1} l_{ik}^2}$
- $l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{ii}}$

Опасность тут представляет то, что под корнем в результате ошибок округления могут возникнуть отрицательные числа, хотя в по условию быть такого не должно

Далее решаем СЛАУ через LU-разложение по классике

Итерационные методы решения СЛАУ

Метод простой итерации

Умножим части СЛАУ $A\vec{u} = \vec{f}$ на некий скаляр t и затем прибавим к обеим частям \vec{u} и проведём преобразования:

$$\begin{aligned}\vec{u} + tA\vec{u} &= \vec{u} + t\vec{f} \\ \vec{u} &= (E - tA)\vec{u} + t\vec{f} \\ B &= (E - tA), \vec{F} = t\vec{f} \\ \vec{u} &= B\vec{u} + \vec{F}\end{aligned}$$

Получили главный оператор итерационного метода:

$$\vec{u}_{k+1} = B\vec{u}_k + \vec{F}$$

Работает он так:

- Задаём начальный вектор \vec{u}_0 (можно даже нулевой)
- Считаем невязку $\vec{r}_0 = \vec{f} - A\vec{u}_0$
- Корректируем решение $\vec{u}_1 = \vec{u}_0 + t\vec{r}_0$
- Понимаем, что отбросив невязку, мы получаем в точности $\vec{u}_{k+1} = B\vec{u}_k + \vec{F}$
- Запускаем решение на следующие итерации и идём пить матча-латте

Критерий сходимости МПИ

Т. Если $\|B\| = q < 1$, то МПИ сходится к решению СЛАУ со скоростью геометрической прогрессии с шагом q^{-1}

Доказывается через формулу МПИ и неравенство треугольника 2.0

Для сходимости с точностью ϵ нужно шагов:

$$k \geq \frac{1}{\ln q} \ln \frac{\epsilon}{\|\vec{u} - \vec{u}_0\|}$$

Т. Пусть СЛАУ имеет единственное решение, тогда МПИ будет сходиться тогда и только тогда, когда все с.з. матрицы B по абсолютным значениям меньше единицы

Путём не самых красивых преобразований делается оценка погрешности решения МПИ $\leq \frac{1-q^k}{1-q} d \leq \frac{d}{1-q}$, где d - максимальная ошибка за все итерации. Из формулы следует, что итоговая погрешность не зависит от количества итераций

Метод Якоби

- Разобьём матрицу на нижнетреугольную без диагонали, диагональную и верхнетреугольную без диагонали $A = L + D + U$
- Подставим разбиение в СЛАУ: $L\vec{u} + D\vec{u} + U\vec{u} = \vec{f}$
- Расставляем индексы у векторов u : $L\vec{u}_k + D\vec{u}_{k+1} + U\vec{u}_k = \vec{f}$
- Предполагая, что D не вырождена, задаём \vec{u}_0 и получаем итерационный процесс
- $\vec{u}_{k+1} = -D^{-1}(L + U)\vec{u}_k + D^{-1}\vec{f}$
 - Можно записать как $\vec{u}_{k+1} = B\vec{u}_k + \vec{F}$ с заменами
 - $\vec{F} = D^{-1}\vec{f}$
 - $B = -D^{-1}(L + U)$ - матрица с нулями на диагонали и остальными элементами считающимися как $b_{ij} = -\frac{a_{ij}}{a_{ii}}$

Т о достаточном условии сходимости метода Якоби. Если матрица A имеет строгое диагональное преобладание, тогда её решение методом Якоби сходится к решению этой СЛАУ (Доказывается через определение матрицы B из которого следует, что при условии теоремы $\|B\|_\infty = q < 1$)

Т. о критерии сходимости метода Якоби. Метод Якоби сходится тогда и только тогда, когда корни λ_{ii} уравнения $\det A' = 0$, где $a'_{ii} = \lambda_{ii}$, $a'_{ij} = a_{ij}$ по модулю не превосходят единицу (доказывается через тот факт, что все λ_{ii} из этого уравнения - это с.з. матрицы B , за счёт чего мы переходим к критерию сходимости МПИ)

Лекция 4

Продолжаем про итерационные методы

Метод Эйделя

- Разобьём матрицу на LDU, но расставим индексы иначе: $L\vec{u}_{k+1} + D\vec{u}_{k+1} + U\vec{u}_k = \vec{f}$
- Если матрица $L + D$ невырождена, тогда получаем итерационную формулу: $\vec{u}_{k+1} = -(L + D)^{-1}U\vec{u}_k + (L + D)^{-1}\vec{f}$
 - Можно записать как $\vec{u}_{k+1} = B\vec{u}_k + \vec{F}$ с заменами
 - $\vec{F} = (L + D)^{-1}\vec{f}$
 - $B = -(L + D)^{-1}U$
- Можно упростить расчёт через формулу, строящую новый вектор покомпонентно: $\vec{u}_{k+1} = -D^{-1}L\vec{u}_{k+1} - D^{-1}L\vec{u}_k + D^{-1}\vec{F}$

Т. о достаточном условии сходимости. Если A - вещественная, симметричная и положительно определённая, то метод Эйделя будет сходиться к решению СЛАУ.

Если не хватает симметричности и положительной определённости, то можно рассматривать вместо A матрицу $A^T A$ - то есть провести **симметризацию матрицы** (при этом число обусловленности **будет квадратом от исходного**)

Задав некоторое число t и решая итерационную задачу $(tL + D)\vec{u}_{k+1} + (t - 1)D\vec{u}_k + tU\vec{u}_k = t\vec{f}$, мы можем регулировать скорость схождения алгоритма

- При $t = 1$ получаем метод Эйделя
- При $1 < t < 2$ называется **методом верхней релаксации**
- При $0 < t < 1$ - **методом последовательной нижней релаксации**

Квадратичный функционал

Квадратичный функционал от \vec{u} - это $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$\Phi(\vec{u}) = (A\vec{u}, \vec{u}) - 2(\vec{f}, \vec{u}) + c$$

Чаще всего функционал рассматривается с симметричной матрицей

Если A положительно определено, то $\Phi(\vec{u})$ называется **функционалом энергии**

Т. Если A - вещ., сим. и положительно опр., тогда существует единственное решение \vec{v} , доставляющее минимум $\min_{\vec{u} \in \mathbb{R}^n} \Phi(\vec{u})$, при этом \vec{v} также является решением СЛАУ $A\vec{u} = \vec{f}$ (Доказывается через предположение о том, что \vec{v} - решение СЛАУ и оценку $\Phi(\vec{v})$ снизу функционалом $\Phi(\vec{v} + \vec{w})$ с использованием аддитивности скалярного умножения)

Задача минимизации

Т. Вектор, минимизирующий Φ единственен (Доказывается через представление двух минимизирующих различных векторов, поиск градиента ($\nabla\Phi = 2A\vec{v} - 2\vec{f} = 0$) и установление того факта, что связанная с функционалом СЛАУ будет иметь единственное решение)

Минимум Φ ищут по следующей итерационной формуле: $\vec{u}_{k+1} = \vec{u}_k - \alpha_k \nabla\Phi(\vec{u}_k)$ - **метод наискорейшего спуска**

- α_k - число, определяемое из условия минимума функции $\Phi(\vec{u}_k - \alpha \nabla\Phi(\vec{u}_k))$
- Раскрываем градиент и получаем $\vec{u}_{k+1} = \vec{u}_k - 2\alpha_k(A\vec{v} - \vec{f})$
- Из чего опять приходим к привычной формуле: $\vec{u}_{k+1} = B\vec{u}_k + \vec{F}_k$
 - $t_k = 2\alpha_k$ (а дальше пишут, что он также определяется $t_k = \frac{(\vec{r}_k, \vec{r}_k)}{(A\vec{r}_k, \vec{r}_k)}$)
 - $B = (E - t_k A)$
 - $\vec{F} = t_k \vec{f}$

Можно выбирать t_k таким образом, чтобы минимизировалась евклидова норма соответствующего шагу вектора невязки. Этому условию будет удовлетворять:

$$t_k = \frac{(A\vec{r}_k, \vec{r}_k)}{(A\vec{r}_k, A\vec{r}_k)}$$

Итерационный процесс с использованием этого действия будет называться **методом минимальных невязок**

Безусловный экстремум функции

Выделим в нормированном пространстве U некоторый вектор u^* и функцию, отображающую это пространство в вещественное

О. $\exists \epsilon > 0 : \forall u \in U : \|u - u^*\| < \epsilon : \Phi(u) \geq \Phi(u^*) \Rightarrow$ функция Φ имеет в u^* **локальный минимум**

О. $\forall u \in U : \Phi(u) \geq \Phi(u^*) \Rightarrow$ функция Φ имеет в u^* **глобальный минимум** (inf)

Задача минимизации значений функции тесно связана с решением **системы нелинейных алгебраических уравнений (СНАУ)**:

$$\begin{cases} f_1(u_1, \dots, u_n) = 0 \\ \dots f_n(u_1, \dots, u_n) = 0 \end{cases}$$

$$\Phi(u_1, \dots, u_n) = \sum_{k=1}^n f_k^2(u_1, \dots, u_n)$$

$$\Rightarrow \Phi(\vec{u}) \geq 0$$

$\Phi(\vec{u}) = 0$ только в точках-решениях СНАУ

Если у Φ есть первые непрерывные производные, то её минимум следует искать среди её **стационарных точек** (точек, в которых все частные производные равны нулю). При этом стационарная точка будет минимум в случае, если в этой точке положительно определена матрица Гессе (элементы матрицы - вторые частные производные, где индексы элементов - это индексы переменных, по которым идёт дифференцирование)

Минимизация функции одной переменной

Перебор - делим отрезок на n точек и проверяем их. Работает, если функция унимодальна (имеет единственную точку минимума). При больших n даёт неплохую точность, но при переходе на случай многих переменных становится очень неэффективным

Лекция 5

Продолжаем про минимизацию функций одной переменной

Модификацией перебора будет **исключение отрезков** (По своей сути и скорости похоже на бинарный поиск)

- Берём точки $u_1, u_2 : a < u_1 < u_2 < b$
- Если $\Phi(u_1) \leq \Phi(u_2)$, далее рассматриваем отрезок $[a, u_1]$
- Иначе рассматриваем отрезок $[u_1, b]$
- Взяв близкие точки $u_1 = \frac{a+b-\delta}{2} < u_2 = \frac{a+b+\delta}{2}$, получим **метод дихотомии**
 - $\delta_n = \frac{b-a}{2} + (1 - \frac{1}{2^n})\delta$
 - Количество итераций $n \approx \log_2 \frac{b-a-\delta}{2\epsilon-\delta}$
 - $\delta \rightarrow 0 \Rightarrow \epsilon_n \rightarrow \frac{b-1}{2^{n+1}}$
 - Останавливаемся, когда $\epsilon_n < \epsilon$, то есть достигаем желаемого порога точности
- Можно взять точки таким образом, чтобы одна из них оставалась и в отрезке меньшего размера уже как пробная.
Делается это при помощи **золотого сечения**
 - $u_1 = a + \frac{3-\sqrt{5}}{2}(b-a)$
 - $u_2 = a + \frac{\sqrt{5}-1}{2}(b-a)$

Итерационные методы минимизации функций

Суть та же, что и у всех итерационных методов - хотим от изначального приближения перейти к новому, более близкому к искомому минимуму функции

Метод покоординатного спуска (МПС)

Сводится к последовательной минимизации функции одной переменной:

- Берём начальное приближение \vec{u}_0
- Фиксируем все компоненты вектора, кроме первой
- Находим минимум по первой компоненте и фиксируем её
- "Отпускаем" вторую компоненту и минимизируем по ней
- Продолжаем до тех пор, пока не будет минимизированы все компоненты
- Повторяем с начала с уже новым вектором \vec{u}_1
- Условие остановки: $|\Phi(\vec{u}_{n+1}) - \Phi(\vec{u}_n)| \leq \epsilon$

Т. о достаточном условии сходимости МПС. Если Φ имеет вторые производные и

$$\begin{aligned} \frac{\delta^2 \Phi}{\delta u_1^2} &\geq a_1 > 0 \\ \frac{\delta^2 \Phi}{\delta u_2^2} &\geq a_2 > 0 \\ \frac{\delta^2 \Phi}{\delta u_1 \delta u_2} &\leq a_3 \\ a_1 a_2 &> a_3^2 \Rightarrow \end{aligned}$$

Последовательность приближений по методу МПС сходится к минимуму функции

Градиентный спуск

Градиент в точке \vec{u}_0 , если $\nabla \Phi(\vec{u}_0) \neq 0$, представляет собой вектор, направленный в сторону максимального возрастания функции Φ

Таким образом, можем определить метод **градиентного спуска**: $\vec{u}_{k+1} = \vec{u}_k - t \nabla \Phi(\vec{u}_k)$

- t - скалярный итерационный параметр
- Условие остановки: $\|\nabla \Phi(\vec{u}_{k+1})\| \leq \epsilon$
- На практике итерационный параметр чаще берётся не фиксированным, а считается на каждом этапе как $t_{k+1} = \min_t (\vec{u}_k - t \nabla \Phi(\vec{u}_k))$

Выпуклые функции и множества

О. множество выпуклое, если для любых его векторов сумма любых долей этих векторов (доли дают в сумме 100%) также даст отрезок, целиком лежащий в множестве: $\forall \vec{u}, \vec{v} \in U : \forall 0 \leq l \leq 1 : \forall \vec{0} \leq \vec{x} \leq l\vec{u} + (1-l)\vec{v} : \vec{x} \in U$

О. Функция называется выпуклой, если $\Phi(l\vec{u} + (1-l)\vec{v}) \leq l\Phi(\vec{u}) + (1-l)\Phi(\vec{v})$. Если знак меняется на строгое $<$, то функция будет **строго выпуклой**

Т. Если матрица Гессе дважды непрерывно дифференцируемой функции положительно определена, то функция строго выпуклая

Т. Если функция выпукла на выпуклом множестве, тогда любой её локальный минимум является глобальным на множестве (доказывается через формулу для выпуклой функции и определения локального и глобального минимума, через которые придём к противоречию, если предположим, что локальный минимум и глобальный - разные точки).

- Для строго выпуклой функции её глобальный минимум на выпуклом множестве достигается в единственной точке (доказывается также от противного через формулы из определений)

Минимизация линейных функций

Общий вид линейной функции описывать смысла не вижу

Интерес в данном случае представляют дополнительные ограничения двух видов:

$$\sum_{i=1}^n a_{ij} u_i = b_j$$
$$\sum_{i=1}^n a_{ij} u_i \leq b_j$$

Расположенные на плоскости, эти ограничения дадут нам многомерный многогранник М, в котором область значений минимизируемой функции образует многогранник G.

Если G выпуклы и неограниченный, то минимума с такими ограничениями нет

Если же минимум есть, то им будет одна из точек-вершин G

Интерполяция функций

Сеточная проекция - набор известных значений функции

Дискретизация - замена функции её сеточной проекцией

Интерполяция - восстановление функции по известной последовательности её сеточных проекций

Кусочно-линейная интерполяция

Проста как пять копеек: берём точки из сетки и соединяем соседние линиями. Получаем ломаную со следующей формулой:

$$f_k(x) = \frac{f_{k+1}(x - x_k) + f_k(x_{k+1} - x)}{x_{k+1} - x_k}$$

Т. о погрешности КЛИ. Если функция на отрезке удовлетворяет условию Липшеца

$$\forall x^*, x^{**} \in [a, b] : |f(x^*) - f(x^{**})| \leq L|x^* - x^{**}|$$

- тогда $|f(x) - f^*(x)| \leq \frac{tL}{2}$
- $t = \max_{0 \leq k \leq N-1} (x_{k+1} - x_k)$ (максимальная длина отрезка в аргументах сетки)
- доказывается через формулу выпуклости и опору на известные значения исходной функции, но как-то душно...

С. При равномерном разбиении $t = \frac{b-a}{N} \Rightarrow |f(x) - f^*(x)| \leq \frac{L(b-a)}{2N}$

С этой оценкой говорят, что КЛИ имеет **первый** порядок сходимости и **точно на полиномах первой степени**

Лекция 6

Продолжаем про интерполяцию

Интерполяция через обобщённый полином

Обобщённый полином - линейная комбинация системы функций: $f_n(x) = \sum_{k=1}^N u_k \phi_k(x)$

Этот полином будет интерполянтом функции f , если $\forall k : f_N(x_k) = f(x_k)$ или в виде системы:

$$\begin{cases} u_1 \phi_1(x_1) + \dots + u_N \phi_N(x_1) = f(x_1) \\ \dots \\ u_1 \phi_1(x_N) + \dots + u_N \phi_N(x_N) = f(x_N) \end{cases}$$

Либо же $A \vec{u}_N = \vec{f}_N$, где A - матрица значений функций ϕ от точек сетки

Иногда удобнее решать сопряжённую СЛАУ: $C \vec{u}_N = A^T \vec{f}_N$

- $C = A^T A$ - матрица Грама, элементы которой - скалярное произведение двух - $(\phi_i, \phi_j) = \sum_{k=1}^N \phi_i(x_k) \phi_j(x_k)$
- Такой переход особенно удобен в случае ортогональных сеточных функций, потому что тогда матрица Грама становится диагональной

Интерполяционный полином Лагранжа

Базисный полином Лагранжа $\phi_k^N(x)$ принимает значение 0 во всех $x_i, i \neq k$ и 1, если $i = k$

Интерполяционный полином Лагранжа:

$$L_N(x) = \sum_{k=1}^N f_k \phi_k^N(x)$$

- Она зависит кроме x ещё и от узлов, и от значений в этих узлах
- Является точной для полиномов N -го порядка

Если f , на отрезке $[a, b]$, $N + 1$ раз дифференцируема и $f^{(N+1)}$ ограничена на $[a, b]$, то погрешность будет считаться как

$$R_N(x) = \frac{1}{(N+1)!} \prod_{i=1}^N (x - x_i) f^{(N+1)}(\xi)$$

- ξ - некоторая точка из $[a, b]$
- Доказывается через тот факт, что $R_N(x) \equiv f(x) - L_N(x)$ и ещё какую-то хуйню

Погрешность с равномерной сеткой:

$$|f(x) - L_N(x)| \leq \frac{t^{N+1}}{N+1} \max_{x \in [a, b]} |f^{(N+1)}(x)|$$

доказывается выводом из общей оценки погрешности

Конечные и разделённые разности

Для равномерной сетки с шагом h :

- **Конечная разность нулевого порядка:** $\Delta^0 f_k = f_k$
- **Конечная разность первого порядка:** $\Delta f_k = f(x_{k+1}) - f(x_k) = f(x_k + h) - f(x_k)$
- **Конечная разность второго порядка:** $\Delta^2 f_k = \Delta(\Delta f_k) = \Delta f_{k+1} - \Delta f_k = (f_{k+2} - f_{k+1}) - (f_{k+1} - f_k) = f_{k+2} - 2f_{k+1} + f_k$
- **Конечная разность n -го порядка:** $\Delta^n f_k = \Delta^{n-1} f_{k+1} - \Delta^{n-1} f_k$

Л. Если $f(x)$ принадлежит классу $C^{(n)}[x_k, x_{k+n}]$ (вроде бы это значит, что она дифференцируема n раз на нём), тогда найдётся такая точка $\eta \in (x_k, x_{k+n})$, что

$$\Delta^n f_k = h^n f^{(n)}(\eta)$$

Доказывается индукцией напрямую из формул для конечных разностей

- **С. 1** Конечная разность степени n алгебраического полинома степени n тождественно постоянна, то есть не зависит от k
- **С. 2** Конечная разность степени $n > l$ алгебраического полинома степени l равна нулю для любого k

Для произвольных различных точек x_1, \dots, x_n **разделённые разности функции f** :

- **Нулевого порядка** - значения функций в этих точках
- **Первого порядка**: $f(x_1; x_2) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = f(x_2; x_1)$
- **Порядка n** : $f(x_1; \dots; x_n) = \frac{f(x_2; \dots; x_n) - f(x_1; \dots; x_{n-1})}{x_n - x_1}$
 - Также представима в виде $\sum_{i=1}^n \frac{f(x_i)}{(x_i - x_0) \dots ((x_i - x_{i-1}))(x_i - x_{i+1}) \dots (x_i - x_n)}$ (сумма частного значения в точке и произведения разности других точек с этой. *доказывается по индукции из определения*)

Значение разделённой разности не зависит от нумерации узлов

Л. $f(x_0; \dots; x_n) = \frac{\Delta^n f_0}{n! h^n}$

- **С. 1**: для некоторого (a, b) , содержащего все узлы разделённой разности, найдётся η такое, что $f(x_0; \dots; x_n) = \frac{f^{(n)}(\eta)}{n!}$
- **С. 2** - аналогично второму следствию для конечной разности

Полином Ньютона

Полином степени n вида

$$N_n(x) = f(x_0) + (x - x_0)f(x_0; x_1) + (x - x_0)(x - x_1)f(x_0; x_1; x_2) + \dots + (x - x_0) \dots (x - x_{n-1})f(x_0; \dots; x_n)$$

называется **полиномом Ньютона**

Л. Полином Ньютона во всех узлах равен исходной функции (*доказывается индуктивно по формуле для разделённой разности*)

$N_n \equiv L_n$, но при этом полином Ньютона оказывается удобнее в случаях, когда нам нужно добавить к полиному несколько слагаемых: в Лагранже будет необходимо пересчитать все компоненты, в Ньютоне - просто добавить новые слагаемые

Выразив $x = x_0 + qh$, можем записать полином Ньютона для равномерной сетки через конечные разности. Такой **полином Ньютона для интерполяции вперёд** будет для интерполяции в начале таблицы либо экстраполяции слева от x_0

Используя полином Ньютона, целесообразно брать узлы по очереди с разных сторон от искомой точки, что позволит получить постепенно убывающий ряд

Обусловленность интерполяции

Тут что-то совсем неприятно, так что будет невероятно краткая выжимка

Оценить обусловленность интерполяции позволяет **функция Либегга λ_N** (формулы не будет)

Для равномерной сетки $\lambda_N \approx 2^N$, что говорит о плохой обусловленности уже даже для средних N

Чебышевское распределение узлов:

$$x_m = \frac{a+b}{2} + \frac{b-a}{2} \cos \frac{(2m-1)\pi}{n}$$

даёт $\lambda_N \approx \ln N$

Лекция 7

Численное интегрирование

Квадратурные формулы

Квадратурная формула - численное приближение определённого на $[a, b]$ интеграла некоторой непрерывной функции:

$$\int_a^b f(x) \approx \sum_{j=1}^N c_j f(x_j)$$

Квадратурная формула **точно** для полинома степени $\leq m$, если при замене f в квадратурной формуле на такой полином, приближённое равенство становится точным

На погрешность квадратурной формулы влияет количество узлов, сами значения узлов и их веса

Т. (обобщённая теорема о среднем). Если функции f, g непрерывны на $[a, b]$, причём $\forall x \in [a, b] : g(x) \geq 0$, тогда:

$$\exists \xi \in [a, b] : \int_a^b f(x)g(x)dx = f(\xi) \int_a^b g(x)dx$$

Доказательство: Определим M, m как максимум и минимум функции f на $[a, b]$, отсюда в силу $g(x) \geq 0$ получаем

$$mg(x) \leq f(x)g(x) \leq Mg(x) \Rightarrow m \int_a^b g(x) \leq \int_a^b f(x)g(x) \leq M \int_a^b g(x) \Rightarrow \exists c \in [m, M] : \int_a^b f(x)g(x)dx = c \int_a^b g(x)dx$$

а в силу непрерывности f заключаем, что найдётся такое $\xi \in [a, b]$, что $f(\xi) = c$

Формула прямоугольников

Пусть f непрерывна на $[-\frac{h}{2}, \frac{h}{2}]$, тогда:

$$\int_{-\frac{h}{2}}^{+\frac{h}{2}} f(x)dx \approx hf(0)$$

Называется **формула прямоугольников** (f дважды дифференцируемая) (*Охуеть... Просто прямоугольником покрыли примерно функцию. Точна для тождественно постоянной функции... Логично*)

Погрешность составит $O(h^3)$ (в полноценном виде выводится из формулы Тейлора с остаточным членом Лагранжа и составляет $\frac{h^3}{24}f''(\xi), \xi \in [-h/2, h/2]$)

Формула трапеции

$$\int_0^h f(x)dx \approx h \frac{f(0) + f(h)}{2}$$

Точна для линейных функций (f дважды дифференцируемая)

Остаточный член: $-\frac{h^3}{12}f''(\xi)$

Формула Симпсона

$$\int_{-h}^h f(x)dx \approx \frac{h}{3}(f(-h) + 4f(0) + f(h))$$

Точна для квадратичных функций (f дважды дифференцируемая)

Остаточный член: $-\frac{h^5}{90}f^{(4)}(\xi)$

Составные квадратуры

Суть построения составных квадратур достаточно проста: берём отрезок $[a, b]$ и делим на N отрезков, для которых применяем простейшие квадратурные функции, значения которых потом суммируем

Для формул прямоугольника и трапеции отрезок интегрирования удобнее брать $h = \frac{b-a}{N}$, для Симпсона - $2h$ ($h = \frac{b-a}{2N}$)

В составных формулах погрешность понижается на порядок (как я понял, за счёт дробления интервала на более мелкие), то есть мы имеем для прямоугольной и трапеции $O(h^2)$ и $O(h^4)$ для парабол

Между максимальной степенью полиномов для которых точна квадратура и порядком точности квадратуры по отношению к $1/N$ есть прямая связь

Интерполяционная квадратура

Основывается на полиноме Лагранжа, то есть в качестве весов мы применяем интеграл от -1 до 1 над базисной функцией Лагранжа, а значения берём из известной нам сетки.

Будет точным для полиномов степени $\leq N - 1$

Полином Лежандра для выбора сетки и формула Гаусса

Полином Лежандра:

$$X_N(x) = \frac{1}{N!2^N} \frac{d^N}{dx^N} [(x^2 - 1)^N]$$

Последовательность Полиномов Лежандра обладает ортогональностью на $[-1, 1]$

Рекуррентная формула для полиномов Лежандра:

- $X_0(x) \equiv 1$
- $X_1(x) = x$
- $X_2(x) = \frac{3}{2}x^2 - \frac{1}{2}$
- $(N + 1)X_{N+1}(x) - (2N + 1)xX_N(x) = NX_{N-1}(x) = 0$

Л. Все корни полинома Лежандра вещественны просты (?) и расположены в $(-1, 1)$

Т. Если для интерполяционной квадратуры взяты точки, являющиеся корнями полинома Лежандра степени N , и весами как интегралами от базисных полиномов Лагранжа, то квадратура будет точна для полиномов степени $\leq 2N - 1$

- Доказывается через переход к интерполяционной квадратуре при после деления любого $P_{2N-1}(x)$ на $X_N(x)$ и того факта, что полином Лежандра ортогонален любому полиному меньшей степени

Интерполяционный полином с узлами-корнями полинома Лежандра степени N называется **формулой Гаусса**

Лекция 8

Численное интегрирование

Усложнённая формула Гаусса

Формулу Гаусса с узлами сетки на интервале $[-1, 1]$ называют иногда **канонической**

Усложнённая формула Гаусса предназначена для произвольных отрезков $[a, b]$:

- Делим $[a, b]$ на M равных отрезков (точки X_0^*, \dots, X_M^*)
- Каждый отрезок делим на N , считая точки по формуле:

$$x_{kj} = \frac{X_k^* + X_{k+1}^*}{2} + x_j \frac{b - a}{2M}$$

- x_j точки сетки размера N для канонической формулы
- Получаем типа "каноническую" формулу Гаусса для каждого отрезка из $[a, b]$, после чего суммируем их:

$$\int_a^b f(x) dx \approx \frac{b - a}{2M} \sum_{j=1}^M c_j \sum_{k=0}^N f(x_{kj})$$

- Погрешность усложнённой формулы составит (*даже думать не хочу, какое тут о-большое):

$$R_N(f) = \frac{(b - a)^{2N+1}}{M^{2N}} \frac{(N!)^4}{((2N)!)^3 (2N + 1)} f^{(2N)}(\xi)$$

Метод Монте-Карло

Вероятностный метод, применимый для интегралов на $[0, 1]$, которые даже могут быть несобственными, но лишь в ограниченном числе особых точек (значения в них при этом равны нулю) и должны принадлежать классу L^2 (интеграл от квадрата модуля меньше бесконечности)

Заключается в том, что:

- мы получаем узлы x_1, \dots, x_N как значения случайной величины с равномерным распределением
- считаем значения $y_1 = f(x_1), \dots, y_N = f(x_N)$
- считаем матожидание случайной величины Y как среднее от y_i
- найденное матожидание будет стремиться к $\int_0^1 f(x)dx$ с точностью 0.997 (доказывается через нахождение дисперсии, ЦПТ и закон трёх сигм)

Задача Коши для ОДУ

Вспомним формулировку задачи Коши:

$$\begin{cases} u' = f(x, u) \\ u(x_0) = u_0 \end{cases}$$

По условию полагается, что решение существует и единственно. Интервал $[x_0, x_0 + l]$

Далее вводятся какие-то странные обозначения и особые пространства...:

- Сетка-вектор: $(x_0, \dots, x_N) = \vec{w}_h$ ($h = \frac{l}{N}$)
- В пространстве $U_h \subset \mathbb{R}^{N+1}$ этих векторов определены сложение, умножение на скаляр и норма $\|\vec{w}_h\|_h = \max |w_j|$
- Решение задачи Коши в сетке тогда тоже будет представимо вектором в пространстве U_h

л. Пусть $a > 0, b \geq 0, \epsilon_0 = 0$ и справедливо $\epsilon_{j+1} \leq (1 + a)\epsilon_j + b$, тогда:

$$|\epsilon_k| \leq b \frac{e^{ka} - 1}{a}$$

Доказывается по индукции

Формула Эйлера

Строим точки так:

$$\begin{cases} y_0 = u_0 \\ y_{j+1} = y_j + hf(x_j, y_j) \end{cases}$$

Движение от y_j к y_{j+1} будет представлять собой перемещение по интегральной кривой $u_j(x)$

- Локальная погрешность на шаге h составляет $O(h^2)$
- Глобальная погрешность - $O(h)$

Лекция 9

Продолжение задачи Коши для ОДУ

Метод Рунге-Кутты второго порядка точности (предиктор-корректор)

Заключается в корректировке грубого предположения следующего значения из метода Эйлера:

$$\begin{cases} y_0 = u_0 \\ y_{j+1}^* = y_j + hf(x_j, y_j) \\ y_{j+1} = y_j + h \frac{f(x_j, y_j) + f(x_j, y_{j+1}^*)}{2} \end{cases}$$

- Локальная погрешность - $O(h^3)$
- Глобальная погрешность - $O(h^2)$

Усовершенствованный метод Эйлера

$$\begin{cases} y_0 = u_0 \\ y_{j+\frac{1}{2}} = y_j + \frac{h}{2} f(x_j, y_j) \\ y_{j+1} = y_j + h f(x_j + \frac{h}{2}, y_{j+\frac{1}{2}}) \end{cases}$$

Погрешности такие же, как и Рунге-Кутты второго порядка

Метод Рунге-Кутты четвёртого порядка точности

$$\begin{cases} y_0 = u_0 \\ y_{j+1} = y_j + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ \begin{cases} k_1 = hf(x_j, y_j) \\ k_2 = hf(x_j + \frac{h}{2}, y_j + \frac{k_1}{2}) \\ k_3 = hf(x_j + \frac{h}{2}, y_j + \frac{k_2}{2}) \\ k_4 = hf(x_j + h, y_j + k_3) \end{cases} \end{cases}$$

- Локальная погрешность - $O(h^5)$
- Глобальная погрешность - $O(h^4)$

Метод Адамса

Душнейшая херня с овер-неприятными формулами, которые можно увидеть, начиная со стр. 17. Здесь приведу лишь общую суть

Реализуется с использованием интерполяционного полинома Лагранжа и требует заданных приближённых решений в m начальных точках, которые находятся чаще всего методом Рунге-Кутты

Глобальная погрешность метода Адамса при этом составит $O(h^{m+1})$

Сравнение с Рунге-Куттой:

- Адамс точнее и требует на новом шаге вычисления лишь некоторых значений
- Рунге-Кутт позволяет нам на ходу изменять h и не требует начальных точек кроме y_0

Процедура уточнения (правило Рунге)

Ебала ещё мутнее Адамса... Предположим, у нас есть одно приближение, второе, третье... Ну и хули?! Ладно, переписывать это желания ноль, так что стр. 37 - и в путь

Приближённое значение по Ричардсону

Без комментариев. Стр. 50. Сразу за этим определением идёт пример использования правила Рунге и ПЗР. Наверное его тоже можно написать

ОДУ второго порядка

Общий вид (интервал $[0, 1]$):

$$\begin{cases} u'' + p(x)u' + q(x)u + f(x) \\ u(0) = q_0 \\ u(1) = q_1 \end{cases}$$

Если функции $p, q, f \in C^{(2)}$, то есть единственное решение $\in C^{(4)}$

Сетка w_h будет делиться на w'_h и w_h^* , где первое подмножество включает все точки, кроме крайних, а второе - только крайние. На их основании определим 3 нормы:

- $\|y\|_h = \max_{0 \leq j \leq N} |y_j|$
- $\|y\|'_h = \max_{1 \leq j \leq N-1} |y_j|$
- $\|y\|_h^* = \max\{y_0, y_N\}$

Лекция 10

Продолжаем ОДУ второго порядка

Разностный метод

Разностный оператор: $L^h u$ сопоставляет каждой непрерывной функции какую-нибудь сеточную функцию

Дальше идёт жуть по его применению. Читай с начала и до стр. 17

Разностный оператор аппроксимирует диффур под разностным оператором со вторым порядком точности

Потом со стр. 17 и до 23 про какой-то метод прогонки...

Про порядок аппроксимации разностным методом на стр. 23

Сразу после этого идёт про устойчивость разностной схемы

При измельчении сетки решение разностной задачи сходится к решению исходной краевой задачи

Основная теорема теории разностных схем

Краткая версия: **аппроксимация + устойчивость = сходимость** (а дальше и не пишется об этом больше ничего, лол.

Только пример приводится. Всегда бы так...)

Дальше полностью дублируется содержимое следующей лекции

Лекция 11

Продолжаем ОДУ второго порядка

Тут рассматривается более хитровыебанная краевая задача и вариационная задача. Понятность изучаемого падает в геометрической прогрессии, так что просто катаем со стр. 1

Метод Рунге

стр. 11-23. Возможно, надо также немного раньше 11-й взять

Ну... если немного пытаться описать суть, то мы заменяем исходную краевую задачу на вариационную, которая решается проще и предъявляет меньше требований к функциям

Метод Галёркина

Тут мы переходим от краевой задачи к проекционной, которую также потом решаем как СЛАУ

- Вводная часть - стр. 23
- Основная - стр. 35

Методы конечных элементов

Суть в задании кусочно-линейных функций на отдельных участках области вычисления (конечных элементов), а дальше применяем метод Рунге. За счёт таких функций подсчёт интегралов станет сильно проще. Стр. 41-57

Лекция 12

Сглаживание функции

Метод наименьших квадратов

В случае ошибок в исходных данных сетки, полученных, например, в ходе реальных измерений, проводить простую интерполяцию нецелесообразно, так как мы сохраним ошибку. Надо её как-то сгладить

Обобщённый полином: $\Phi_m(x) = a_0\phi_0(x) + \dots + a_m\phi_m(x)$ - должен приближать значения функции (теперь, в отличие от задачи интерполяции, мы не требуем их точного соответствия в узлах). Часто базисные функции ϕ_i - это просто x_i , поэтому можно использовать полином $P_m(x)$

Формула квадратичного уклонения:

$$\delta(\Phi_m, \vec{y}) = \sqrt{\frac{1}{n+1} \sum_{i=0}^n (\Phi_m(x_i) - y_i)^2}$$

- \vec{y} - заданные табличные значения

Есть альтернативная форма записи через матрицы:

$$S(\vec{a}, \vec{y}) = \sum_{i=0}^n \left\{ \sum_{j=0}^m a_j \phi_j(x_i) \right\}^2 = \|P\vec{a} - \vec{y}\|_2^2$$

- Матрица значений базисных функций с разными индексами

Связаны эти формулы следующим образом:

$$\delta^2(\Phi_m, \vec{y}) = \frac{1}{n+1} S(\vec{a}, \vec{y})$$

Таким образом, **линейная задача МНК** сводится к тому, чтобы найти такие $a_j(\vec{y})$, которые минимизируют среднеквадратичное уклонение

Решить эту задачу можно через поиск экстремума $S(\vec{a}, \vec{y})$ при помощи частных производных, которые дадут нам СЛАУ:

$$P^T P \vec{a} = P^T \vec{y}$$

Т. Пусть порождающие функции ϕ_j независимы, тогда полином наилучшего среднеквадратичного приближения существует и единственен

- При этом, если $m = n$, МНК даст нам в точности интерполяционный полином, поэтому сглаживающий эффект будет наблюдаться при $m \ll n$

Дальше идут ещё несколько формул, подробно описывающих используемые СЛАУ, а на стр. 30 есть неплохой пример

Использование в качестве ϕ_i функций x^i достаточно приятно, но даёт нам плохую обусловленность уже при $m = 5$, так как в совокупности функции могут быть близки к линейной зависимости. Хорошим вариантом будут полиному Чебышева (для интервала $[-1, 1]$):

- $T_0(x) = 1$
- $T_1(x) = x$
- $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$

Со стр. 45 дубликат следующей лекции

Лекция 13

Дискретное преобразование Фурье

Не скажу, что это слишком сложно и непонятно, но формулы слишком душные, поэтому просто пишем из лекции. Здесь оставляю только финальные формулы

$$w = e^{\frac{2\pi i}{N}}$$
$$x_j = \frac{j}{N}$$

Дискретное преобразование Фурье:

$$a_k = \frac{1}{N} \sum_{l=0}^{N-1} f(x_l) w^{-kl}$$

Обратное дискретное преобразование Фурье:

$$f(x_j) = \sum_{k=0}^{N-1} a_k w^{kj}$$

В общем случае вычисление занимает порядка N^2 операций, однако в случае непростого $N \geq 4$ можно существенно ускорить процесс, переиспользуя посчитанные ранее значения. Делается это за счёт разложения $N = N_1 * N_2$ (а подробнее смотри со стр. 23)

Развитием этой идеи ускорения является разложение N на s множителей, которое даёт асимптотику преобразования в $O(N \log N)$

Про расчёт погрешности смотри со стр. 40

Лекция 14

Вычисление собственных значений

Немного рекапа:

Собственные значения - скаляры, удовлетворяющие уравнению $Ax = lx$ (либо более привычно $(A - lE)x = 0$), где x - ненулевой вектор (**собственный вектор** соответствующего СЗ l)

Сами СЗ удобно искать по уравнению $\det(A - lE) = 0$

- $\det(A - lE) = (-1)^m l^m + p_1 l^{m-1} + \dots + p_{m-1} l + p_m$ - характеристический полином

Прямые численные методы сводились к решению уравнения с характеристическим полиномом, однако такой подход даёт очень плохую обусловленность

Матрицы A, B называются **подобными**, если существует невырожденная матрица P такая, что:

$$AP = PB \Leftrightarrow B = P^{-1}AP$$

Подобные матрицы имеют одинаковый спектр (набор СЗ), однако могут иметь разные СВ (но имеет место равенство $\vec{v}_A = P\vec{v}_B$)

Задача поиска СЗ для подобных матриц при этом сведётся к преобразованию матрицы A к верхнетреугольному виду (СЗ будут на диагонали), а можно и ещё проще

Т. Любая квадратная матрица подобна матрице, в которой на диагонали лежат её СЗ, а над диагональю идут нули или единицы, причём в случае единицы элементы слева и под ней будут равны (такая матрица называется **Жордановой формой**)

Матрица будет иметь **простую структуру**, если её жорданова форма - диагональная матрица => все СЗ различны => все СВ различны (СВ при этом образуют базис в \mathbb{R}^m)

Теорема Гершгорина

Будем называть кругами Гершгорина S_i для матрицы A области в комплексной плоскости в центрами $a_i i$ и радиусом r_i , равным сумме элементов i -й строки, исключая диагональный

Т. Все СЗ матрицы лежат в объединении её кругов Гершгорина

Отношение Рэля

Играет важную роль для вычисления СЗ и СВ:

$$\rho(x) = \frac{(Ax, x)}{(x, x)}$$
$$(Ax, x) = \sum_{i,j=1}^m a_{ij} x_i x_j$$

Т. Для симметричной вещественной матрицы её минимальное и максимальное СЗ - это минимум и максимум отношения Рэля

Стационарные точки функция Рэля - это СВ матрицы (градиент равен нулю)

Если вдобавок $\|x\| = 1$, то функция Рэля от этого x хорошо приближает СЗ

Задача поиска СЗ для симметричных матриц хорошо обусловлена, а для несимметричных - достаточно плохо (*на 43 странице есть теорема, из которой это следует, но мне она показалась не сильно интересной*)