Содержание

Задания по вычислительной математике

7 октября 2018 г.

1 Вычислительная линейная алгебра

1.1 Метод простой итерации

Написать программу для решения линейной системы

$$Ax = b,\ A = A^T > 0$$

методом простой итерации (методом релаксации).

Требования к программе:

- Программа должна принимать на вход размерность системы n, создавать случайную положительно определённую матрицу и правую часть такого размера.
- После этого нужно найти точное решение с помощью функции из стандартной библиотеки `numpy.linalg.solve`.
- Перед использованием итерационного метода нужно найти оценку собственных чисел матрицы с помощью кругов Гершгорина, сравнить с точными собственными числами (функция из стандартной библиотеки `numpy.linalg.eigvals`).
- Программа должна вычислять приближенное решение методом простой итерации с произвольным параметром, в том числе с оптимальным значением вычисленным по оценкам собственных чисел и по точным собственным числам. На входе задаётся требуемая точность, которая используется в критерии остановки итераций.
- Программа должна выводить число итераций, точную ошибку (вычисленную по точному решению), а также график зависимости логарифма ошибки от номера итерации.
- Автор программы должен уметь объяснить полученные результаты на основе изученной теории.

1.2 Метод Зейделя

Написать программу для решения линейной системы

$$Ax = b,\ A = A^T > 0$$

методом Зейделя.

Требования к программе:

- Программа должна принимать на вход размерность системы n, создавать случайную положительно определённую матрицу и правую часть такого размера.
- После этого нужно найти точное решение с помощью функции из стандартной библиотеки `numpy.linalg.solve`.
- Программа должна вычислять приближенное решение методом Зейделя, **причем в итерационном методе нельзя использовать обращение матриц и матричное умножение: нужно реализовать метод поэлементно с помощью цилов**. На входе задаётся требуемая точность.
- Программа должна выводить число итераций, точную ошибку (вычисленную по точному решению), а также график зависимости логарифма ошибки от номера итерации.
- Автор программы должен уметь объяснить полученные результаты на основе изученной теории.

1.3 Метод Гаусса с выбором главного элемента

Написать программу для решения линейной системы

$$Ax = b$$

методом Гаусса с выбором главного элемента (по строке или по столбцу). Требования к программе

- Программа должна принимать на вход матрицу и правую часть
- Сначала нужно вычислить матрицы L, U и матрицу перестановки P
- После этого нужно решить системы с треугольными матрицами
- **Нельзя использовать матричное умножение и обращение матриц готовыми функциями. Метод нужно реализовать с помощью циклов, поэлементно**.
- Программа должна выводить норму разницы между полученным решением и решением из стандартной функции `numpy.linalg.solve`

1.4 Метод Холецкого

Написать программу для решения линейной системы

$$Ax = b, A = A^T > 0$$

методом Холецкого с выбором главного элемента (по строке или по столбцу). Требования к программе

- Программа должна создавать матрицу $A = A^T > 0$ и правую часть
- Сначала нужно вычислить матрицу : $A = CC^T$
- После этого нужно решить 2 системы с треугольными матрицами
- **Нельзя использовать матричное умножение и обращение матриц готовыми функциями. Метод нужно реализовать с помощью циклов, поэлементно**.
- Программа должна выводить норму разницы между полученным и решением из стандартной функции `numpy.linalg.solve`. Для проверки правильности разложения, можно использовать функцию `numpy.linalg.cholesky`

2 Дополнительные задачи на выбор

Задачи могут быть описаны кратко, постановку нужно обсуждать с преподавателем.

2.1 Решение уравнения Пуассона с помощью нейронной сети

Для решения краевой задачи для уравнения Пуассона в круге:

$$u_{xx} + u_{yy} = u^2 + \frac{3}{2}u^3 \quad (1)$$

$$u|_{x^2+y^2=1} = -2 \quad (2)$$

Для выполнения граничного условия можно использовать подстановку:

$$u(x, y) = (1 - x^2 - y^2)n(x, y) - 2 \quad (3)$$

Такая функция удовлетворяет граничному условию для любой ограниченной $n(x, y)$.

Для аппроксимации функции n нужно использовать нейросеть, у которой 2 входа (x и y) и один выход (значение неизвестной функции $n(x, y)$). Функция ошибки, которую нужно минимизировать, это невязка в уравнении:

$$\varepsilon_0 = (u_{xx} + u_{yy} - f(x, y, u))^2 \quad (4)$$

Её можно вычислить дифференцируя нейросеть по входным параметрам (x и y), для этого в каждой реализации есть соответствующая функция, например, в TensorFlow – tf.gradients. Здесь усложнение только в том, что нужно считать 2-е производные.

Обучать нейросеть нужно на точках, внутри области $x^2 + y^2 = 1$, для начала можно попробовать случайно накиданные точки.

Можно использовать статью с такой конфигурацией слоев:

$$2, 64, 64, 64, 64, 64, 64, 1$$

Сигмоиды должны быть гладкими (бесконечно дифференцируемые).

После обучения, нужно сравнить полученную аппроксимацию с аналитическим решением:

$$u = \frac{4}{x^2 + y^2 - 3} \tag{5}$$

Подробная статья с описанием: ссылка

2.2 Решение задачи Коши для системы уравнений теории метеоров

Нужно по формулам запрограммировать вычисление правой части системы ОДУ, и написать программу для решения задачи Коши с помощью готового метода из какой-либо библиотеки, и построения графиков решений.

Система уравнений появится позже...

2.3 Спектроскопия динамического рассеяния света

Любое задание отсюда: https://vk.com/numan_research. Эти описания будут в ближайшее время обновляться.