**Министерство образования Республики Беларусь**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

Фурс Дмитрий Сергеевич

**Методы вычислений**

Отчет по лабораторной работе № 2,

Вариант 8

студента 2-го курса 12-ой группы

|  |  |  |
| --- | --- | --- |
|  |  | **Преподаватель** |
|  | **Мойса А.В.** |
|  | | |

**Минск 2019**

Оглавление

[Задание 1. Генерация матрицы 3](#_Toc9793921)

[Задание 2. Степенной метод 3](#_Toc9793922)

[Задание 3. QR-алгоритм 4](#_Toc9793923)

[Задание 4. Анализ графика функции 5](#_Toc9793924)

[Задание 5. Метод бисекции 6](#_Toc9793925)

[Задание 6. Дискретный метод Ньютона 6](#_Toc9793926)

[Задание 7. Улучшенный метод Ньютона 7](#_Toc9793927)

# Задание 1. Генерация матрицы

Для получения необходимого количества значащих цифр использовал функцию из стандартного модуля random – random.uniform(a,b), которая возвращает число с плавающей точкой x, a <= x <= b. Заполнение матрицы происходит поэлементно.

# Задание 2. Степенной метод

Для нахождения собственных значений, можно рассмотреть характеристический многочлен (удобно для матриц 2х2, 3х3). В общем случае точных методов нахождения собственных значений нет, так как если степень многочлена >= 5, то нельзя выразить корни в радикалах. Поэтому часто для поиска собственных значений используют итерационные методы.

Степенной метод позволяет найти максимальное по модулю собственное значение и соответствующий ему собственный вектор. Суть метода заключается в следующем: выберем произвольный вектор y0 и рассмотрим его разложение по базису собственных векторов матрицы A

Y0 = a1v1 + a2v2 + … + anvn, где vi – собственные векторы.

Тогда если рассмотреть последовательность {yk}: yk+1 = Ayk при k стремящемся к бесконечности вектор yk будет всё сильнее приближаться к собственному подпространству, соответствующему vmax (собственный вектор, соответствующий максимальному по модулю собственному значению). Дополнительно на каждой итерации необходимо нормировать вектор yk. Если расписать yk по базису собственных векторов, можно увидеть, что степенной метод будет сходиться со скоростью геометрической прогрессии со знаменателем l2/l1, где |l1| > |l2| (l1 – максимальное по модулю собственное значение).

Для поиска второго собственного значения использовал следующий алгоритм:

1. Находил матрицу B = A – l1E, где l1 – уже найденное собственное значение
2. Находил максимально по модулю собственное значение матрицы B – lb
3. l2 = lb + l1, v2 = vb

При тестировании метод при точности порядка 10-14 очень часто не сходится.

Норма разности: 8.149003440634096e-15

8.011234740023206e-15

Как уже отмечалось выше, сходимость зависит от близости собственных значений.

# Задание 3. QR-алгоритм

QR-алгоритм – один из наиболее известных методов нахождения всех собственных значений произвольной квадратной матрицы. Позволяет найти все (в том числе и комплексные) собственные значения. Простейший алгоритм состоит в следующем:

1. Строим QR-разложение матрицы Ak (A0 = A): Ak = QkRk
2. Вычисляем Ak+1 = RkQk

Преобразование Ak -> Ak+1 является преобразованием подобия (т.к. Ak+1 = RkQk = Qk-1AkQk), а значит сохраняет спектр матрицы.

На практике этот алгоритм как правило не применяют, поскольку он очень трудоёмкий. Поэтому матрицу A вначале преобразованием подобия приводят к форме Хессенберга, а затем уже начинают итерации QR-алгоритма.

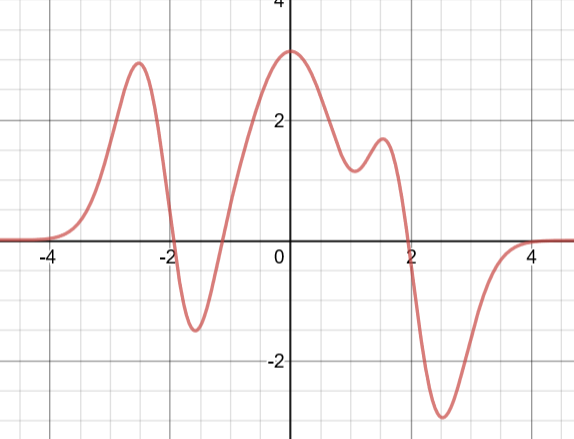
Для того, что определить различие в трудоёмкости и сопоставить его с теоретическим, в лабораторной работе были реализованы два алгоритма: QR-алгоритм на неподготовленной матрице A и QR-алгоритм с предварительным приведением матрицы А к форме Хессенберга алгоритмом Арнольди. Для каждого из алгоритмов задавалось максимально количество итераций (для первого 1000, для второго – 100). Полученные собственные значения совпадали с точностью 10-12. Теоретически приведение матрицы к форме Хессенберга должно снизить трудоёмкость одной итерации алгоритма с O(n3) до O(n2).

Получив почти верхнетреугольную матрицу, обнуляю близкие к нулю элементы (порядка 10-10) и прохожусь по диагонали. Если под диагональю стоит нулевой элемент, то сразу получаю вещественное собственное значение, если нет, то рассматриваю матрицу 2х2, у которой на диагонали расположены два диагональных элемента общей матрицы, и нахожу её собственные значения, решая квадратное уравнение.

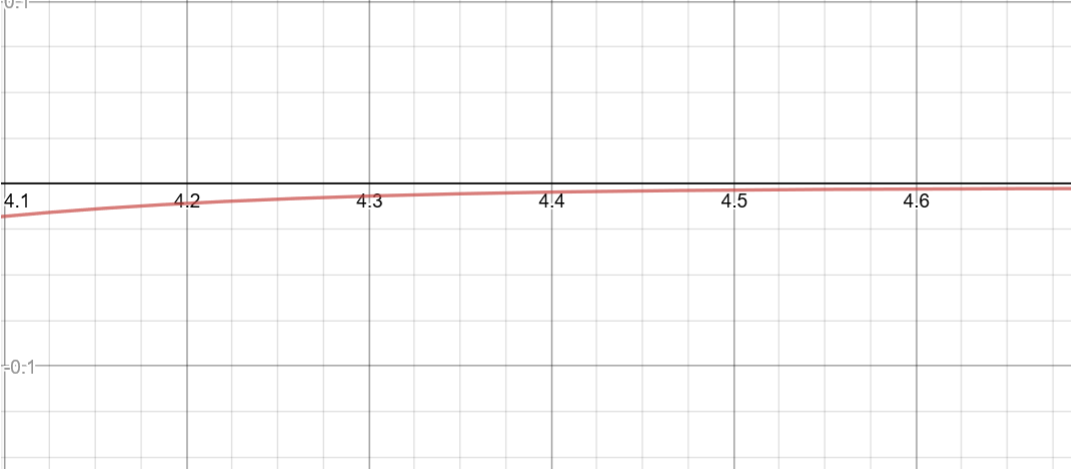
Среднее время нахождения собственных значений: 0,0829184 c.

В отличие от степенного метода позволяет достаточно быстро найти все собственные значения матрицы. (для нахождения собственных векторов необходимо дополнительно сохранять матрицы Qk).

# Задание 4. Анализ графика функции



Более детально рассмотрим правую/левую ветви



Заметим, что ось X они не пересекают, а лишь стремятся к ней.

Определим отрезки корней:

* x1 - (-2, -1.6)
* x2 - (-1.4, -1)
* x3 – (1.6, 2)

На промежутке [-4.5, 4.5] функция непрерывна => других корней на этом промежутке нет. Если рассмотреть функцию на промежутке , заметим, что первое слагаемое функции будет стремиться к нулю, а само значение функции будет стремиться к -x/2018, а значит других корней мы не получим.

# Задание 5. Метод бисекции

Суть метода заключается в последовательном делении пополам первоначально предполагаемых отрезков, которым принадлежит корень. На концах отрезка функция должна быть противоположных знаков. Поэтому мы поделим отрезок пополам и возьмём в качестве нового отрезка тот, на концах которого функция имеет различные знаки.

1. x1
   1. первоначальное приближение:

(-2, -1,6)

* 1. приближение, полученное методом бисекции:

(-1.9519531250000002, -1.9518554687500003)

* 1. количество итераций

12

1. x2
   1. первоначальное приближение:

(-1.4, -1)

* 1. приближение, полученное методом бисекции:

(-1.13583984375, -1.1357421874999998)

* 1. количество итераций

12

1. x3
   1. первоначальное приближение:

(1.6, 2)

* 1. приближение, полученное методом бисекции:

(1.9518554687500003, 1.9519531250000002)

* 1. количество итераций

12

Таким образом получили отрезки, ширина которых меньше 10-4. Это приближение можно использовать для нахождения корня, например, по методу Ньютона.

# Задание 6. Дискретный метод Ньютона

xn+1 = xn – (h\*f(xn))/(f(xn+h)-f(xn))

В качестве h брал 10-5, eps = 10-15, где eps – условие остановки |xn-xn-1| <eps.

В качестве начального приближения выбиралась правая граница отрезка, полученного по методу бисекции.

1. x1
   1. приближение, полученное методом бисекции:

-1.9518554687500003

* 1. приближение, полученное дискретным методом Ньютона

-1.9519125752999187

* 1. количество итераций

3

1. x2
   1. приближение, полученное методом бисекции:

-1.1357421874999998

1. приближение, полученное дискретным методом Ньютона

-1.135756704469539

1. количество итераций

2

1. x3
   1. приближение, полученное методом бисекции:

1.9519531250000002

1. приближение, полученное дискретным методом Ньютона

1.9519145740760813

1. количество итераций

3

# Задание 7. Улучшенный метод Ньютона

xn+1 = xn – f(xn)/f ,(xn)

В качестве eps брал 10-16

1. x1
   1. приближение, полученное дискретным методом Ньютона

-1.9519125752999187

* 1. улучшенное приближение

-1.9519125752999191

* 1. количество итераций

1

1. x2
   1. приближение, полученное дискретным методом Ньютона

-1.135756704469539

* 1. улучшенное приближение

-1.1357567044695389

1. количество итераций

1

1. x3
   1. приближение, полученное дискретным методом Ньютона

1.9519145740760813

* 1. улучшенное приближение

1.9519145740760815

1. количество итераций

1

Удалось немного улучшить корни, полученные дискретным методом Ньютона.