**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Уфимский государственный нефтяной технический университет**

**Кафедра вычислительной техники и инженерной кибернетики**

**Лабораторная работа №5**

**“Компьютерное МОДЕЛИРОВАНИЕ”**

*Тема :* Задачи скалярной оптимизации

Выполнил: студент гр. БПО-18-01 Р.Р. Исангулов

Вариант 8

Проверил: профессор кафедры ВТИК Ф.У. Еникеев

Уфа 2022

**Задание**

Дана функция *.* Требуется найти минимум абсолютного значения функции с заданной точностью e=0.000001 следующими численными методами:

* Оптимальный пассивный поиск
* Метод половинного деления
* Метод золотого сечения
* Метод бисекции
* Метод Ньютона
* Метод итераций
* Метод квадратичной интерполяции
* Метод Фибоначчи

Полученные результаты следует проверить с помощью встроенных средств программы MS Excel. По результатам работы составить отчет.

**Постановка задачи**

В лабораторной работе №5 по варианту №8 требуется найти минимум абсолютного значения функции с заданной точностью ε=0.000001. Требуется найти минимум модуля функции ||→min или же минимум ее квадрата ()2→min. В данном отчете представлены результаты, полученные в ходе выполнения лабораторной работы студентом группы БПО-18-01 Исангуловым Ринатом.

1. **Оптимальный пассивный поиск**

С целью повышения точности и сокращения вычислений метод пассивного поиска может быть модифицирован следующим образом. В первой серии пробных точек находится отрезок [xk-1,xk], на котором имеется минимум функции f(x) и затем процедура оптимального пассивного поиска проводится уже не для отрезка [a,b], а только для его части [xk-1,xk].

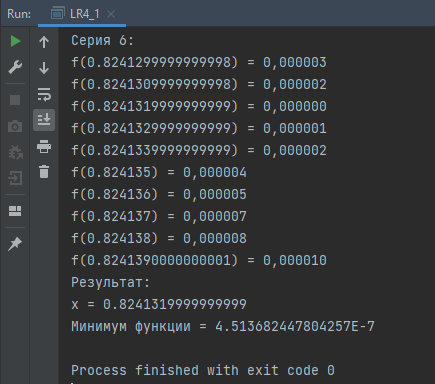
**Таблица 1.1.**Список идентификаторов для программы LR4\_1.java

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Переменная** | **Примечание** |
| x | x | Независимая переменная x |
| y | y | Значение функции y |
| a, b (xk-1, xk) | a, b | Отрезок |
| h | h | Шаг |

Листинг программы:



Результаты расчетов:



1. **Метод половинного деления**
2. Вычисляют α(k)=[a(k)+b(k)]/2−δ и β(k)=[a(k)+b(k)]/2+δ
3. Находят f(α(k)) и f(β(k))
4. Определяют новый отрезок локализации по следующему правилу:
5. Если f(α(k))≤f(βk), то [a(k+1),b(k+1)]=[a(k),β(k)] и x(k+1)=α(k)
6. Если f(α(k))≥f(β(k)), то [a(k+1),b(k+1)]=[α(k),b(k)] и x(k+1)=β(k)

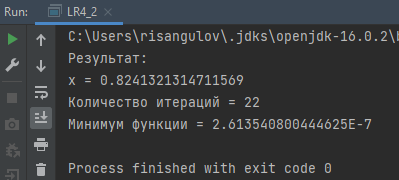
**Таблица 2.1.**Список идентификаторов для программы LR4\_2.java

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Переменная** | **Примечание** |
| x | x | Независимая переменная |
| a | a | Левый конец заданного отрезка [a,b] |
| b | b | Правый конец заданного отрезка [a,b] |
| f(x) | f(x) | Заданная функция |
| N | n | Максимальное количество итераций |
| alpha | α | Параметр метода (α=(a+b)/2−δ) |
| beta | β | Параметр метода (β=(a+b)/2+δ) |
| delta | δ=ε/3 | Параметр метода (0<δ<(b−a)/2) |
| e | ε | Заданная точность |

Листинг программы:



Результаты расчетов:



1. **Метод золотого сечения**
2. Полагают a(0) = a, b(0)=b, k=0.
3. k=k+1
4. Вычисляют αk, βk по формулам:

αk = ak + 2·(bk−ak)/(3+√5)

βk = ak + 2·(bk−ak)/(1+√5)

1. Находят f(αk) и f(βk)
2. Определяют новый отрезок локализации по следующему правилу:

Если f(αk)≤f(βk), то [a(k+1),b(k+1)]=[a(k),β(k)] и x(k+1)=α(k)

Если f(α(k))≥f(β(k)), то [a(k+1),b(k+1)]=[α(k),b(k)] и x(k+1)=β(k)

1. Проверяют выполнение условия |b(k)−a(k)|<ε. Если оно выполняется, конец вычислений, если нет, переход к следующей итерации (шаг 2).

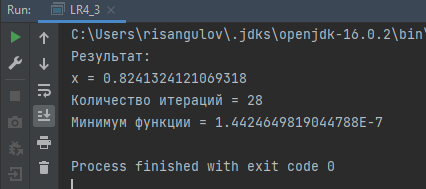
**Таблица 3.1.**Список идентификаторов для программы LR4\_3.java

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Переменная** | **Примечание** |
| x | x | Независимая переменная |
| a | a | Левый конец заданного отрезка [a,b] |
| b | b | Правый конец заданного отрезка [a,b] |
| f(x) | f(x) | Заданная функция |
| alpha | α | Параметр метода (α=a + 2·(b−a)/(3+√5)) |
| beta | β | Параметр метода (β=a + a + 2·(b−a)/(1+√5)) |
| e | ε | Заданная точность |

Листинг программы:



Результаты расчетов:



1. **Метод бисекции**

Пусть отрезок локализации =[a(k),b(k)] известен и найдено значение x(k)=(a(k)+b(k))/2.

Алгоритм метода бисекции:

1. Вычисляют значение f′(x(k))
2. Если f′(x(k))<0, то [a(k+1),b(k+!)]=[x(k),b(k)], в противном случае [a(k),b(k)]=[a(k),x(k)]
3. Вычисляют x(k+1)=(a(k+1)+b(k+1))/2.

Сходимость метода: |x(k)–x\*|≤(b–a)/2(k+1)

Поскольку в методе бисекции требуется вычислять первую производную функции f(x) представляется целесообразным вместо поиска минимума модуля функции ||→min искать минимум ее квадрата →min. Это связано с тем обстоятельством, что первая производная функции y=|| терпит разрыв первого рода в точке экстремума. В то же время, первая производная функции y= непрерывна в точке экстремума.

Имеем:

Функция f(x) =

Первая производная

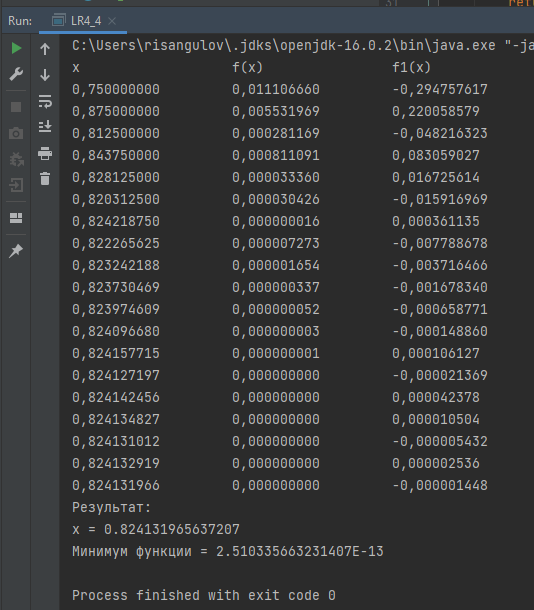
**Таблица 4.1.**Список идентификаторов для программы LR4\_4.java

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Переменная** | **Примечание** |
| x | x | Независимая переменная |
| a | a | Левый конец заданного отрезка [a,b] |
| b | b | Правый конец заданного отрезка [a,b] |
| f(x) | f(x) | Заданная функция f(x) |
| f1(x) | f(x) | Первая производная заданной функции f′(x) |
| e | ε | Заданная точность |

Листинг программы:



Результаты расчетов:



1. **Метод Ньютона**

Поскольку в методе Ньютона требуется вычислять первую производную функции f(x) представляется целесообразным вместо поиска минимума модуля функции ||→min искать минимум ее квадрата →min. Это связано с тем обстоятельством, что первая производная функции y=|5x–6x–3| терпит разрыв первого рода в точке экстремума. В то же время, первая производная функции y=непрерывна в точке экстремума.

Имеем:

Функция f(x) =

Первая производная:

Вторая производная:

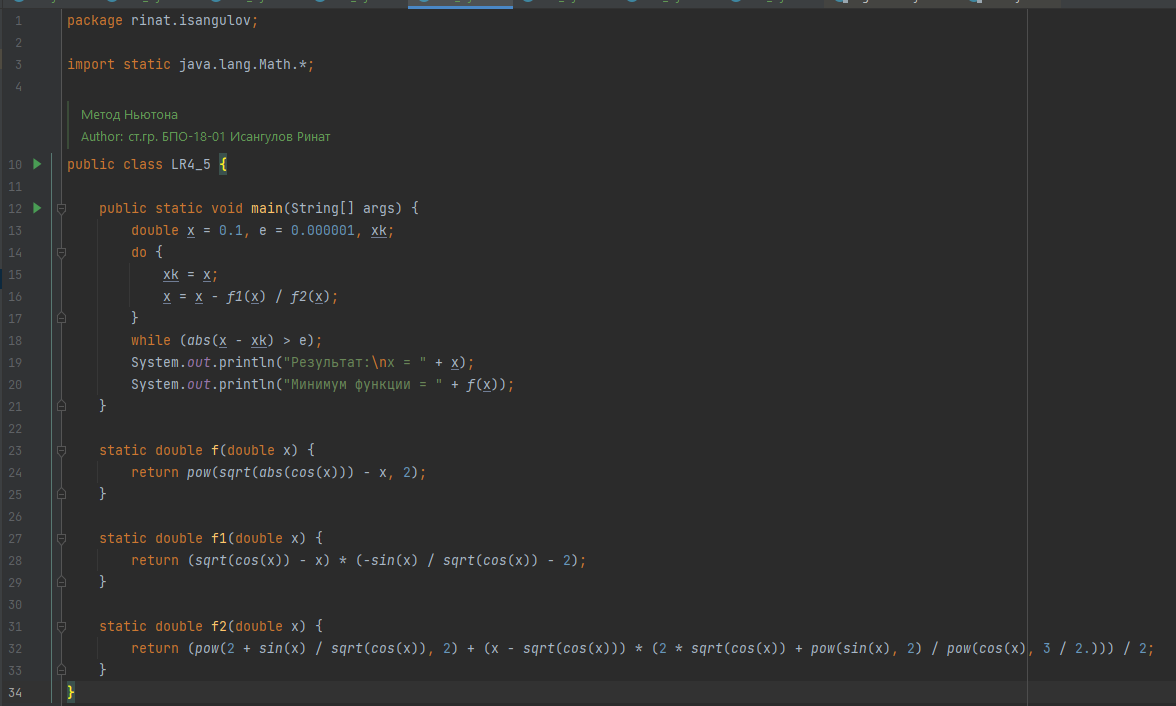
Итерационная формула метода Ньютона:

x(k+1)=x(k) − f'(x(k))/f''(x(k))

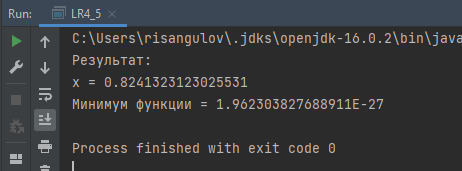
**Таблица 5.1.**Список идентификаторов для программы LR4\_5.java

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Переменная** | **Примечание** |
| x | x | Независимая переменная |
| f(x) | f(x) | Заданная функция f(x) |
| f1(x) | f(x) | Первая производная заданной функции |
| f2(x) | f(x) | Вторая производная от заданной функции |
| e | ε | Заданная точность |

Листинг программы:



Результаты расчетов:



1. **Метод итераций**

В программе LR4\_6.java реализована следующая итерационная процедура. Для заданной функции f(x) = ее первая производная равна . Перепишем уравнение

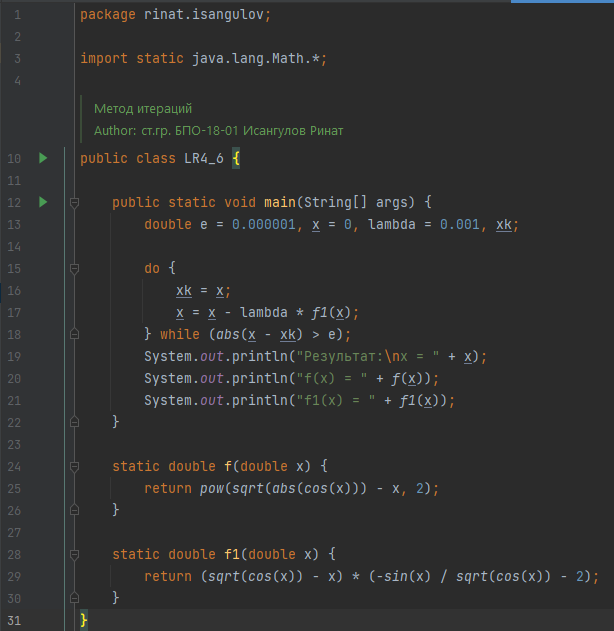
x = x − λ·f′(x) и на этом основании составим следующую итерационную процедуру

x(k+1) = x(k) − λ·f′(x)

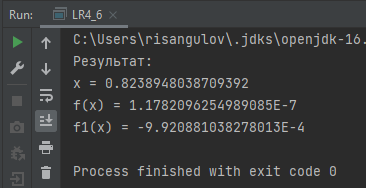
**Таблица 6.1.**Список идентификаторов для программы LR4\_6.java

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Переменная** | **Примечание** |
| x | x | Независимая переменная |
| f(x) | f(x) | Заданная функция f(x) |
| f1(x) | f(x) | Первая производная заданной функции |
| e | ε | Заданная точность |

Листинг программы:



Результаты расчетов:



1. **Метод квадратичной интерполяции**

В программе LR4\_7.java, листинг которой приведен ниже, используется следующая итерационная формула x\* = −b/2a = x0 − (h/2)·(f2 − f1) / (f2 − 2f0 + f1)

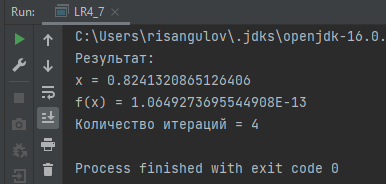
**Таблица 7.1.**Список идентификаторов для программы LR4\_7.java

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Переменная** | **Примечание** |
| x | x | Независимая переменная |
| f(x) | f(x) | Заданная функция f(x) |
| h | h | Параметр метода |
| e | ε | Заданная точность |

Листинг программы:



Результаты расчетов:

****

1. **Метод Фибоначчи**

По заданной точности вычислений Δ определяем число вычислений значений функции f(x) из условия F(n)>(b−a)/(2Δ)

Метод Фибоначчи состоит из (n−1) шагов. На (k+1) итерации находят точки:

α(k) = a(k) + F(n−2)/F(n)·(b(k) − a(k))

β(k) = a(k) + F(n−1)/F(n)·(b(k) − a(k))

Новый отрезок локализации определяют по следующему правилу:

если f(α(k)) ≤ f(β(k)), то [a(k+1),b(k+1)] = [a(k+1),β(k+1)]

если f(α(k)) > f(β(k)), то [a(k+1),b(k+1)] = [α(k+1),b(k+1)]

В первом случае за очередное приближение к точке минимума принимают x(k+1)α(k), а во втором случае x(k+1)β(k). Важно то, что в любом случае точка x(k+1) совпадает с одной из точек

α(k+1) = a(k+1) + F(n−k−2)/F(n−k)·(b(k+1) − a(k+1))

β(k+1) = a(k+1) + F(n−k+1)/F(n−k)·(b(k+1) − a(k+1))

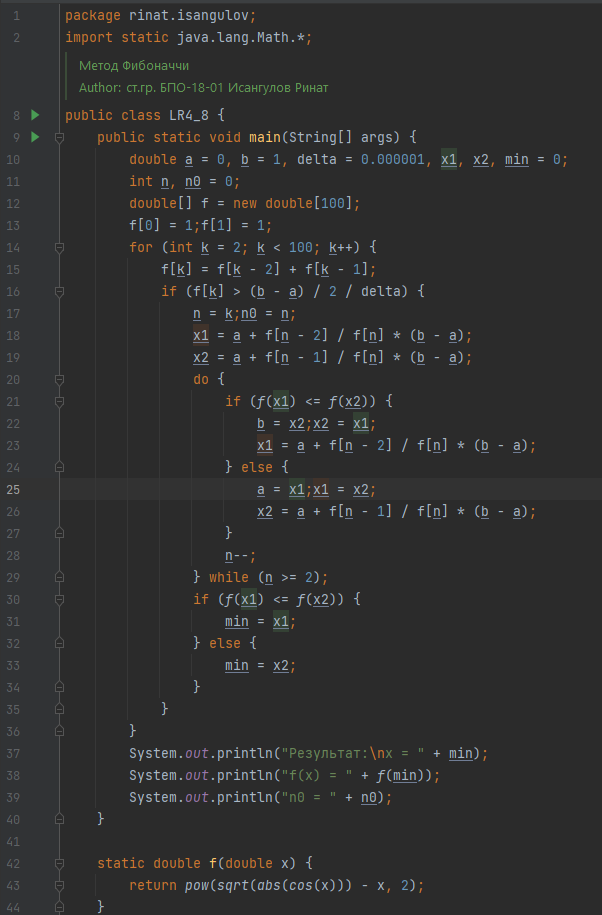
Поэтому на очередном шаге достаточно вычислить значение функции лишь в одной недостающей точке.

В результате выполнения (n−1) шагов отрезок локализации уменьшается в Fn+1/2 раз, а точка x(n−1) оказывается центральной для последнего отрезка локализации (a(n−1),b(n−1)]. Поэтому для x(n−1) справедлива следующая оценка погрешности |x\* − x(n−1)| ≤ Δ / Fn+1

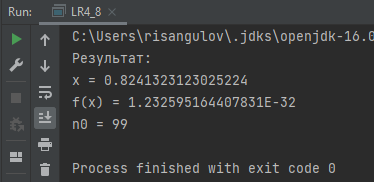
**Таблица 8.1.**Список идентификаторов для программы LR4\_8.java

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Переменная** | **Примечание** |
| min | x | Независимая переменная |
| a | a | Левый конец заданного отрезка [a,b] |
| b | b | Правый конец заданного отрезка [a,b] |
| f(x) | f(x) | Заданная функция (y = f(x) = 4x−2x−2) |
| x1 | α | Параметр метода |
| x2 | β | Параметр метода |
| delta | Δ | Заданная точность |

Листинг программы:



Результаты расчетов:

****

1. **Решение задачи c помощью встроенных средств программы MS Excel**

Строим график функции y= (не абсолютного значения, а самой функции).

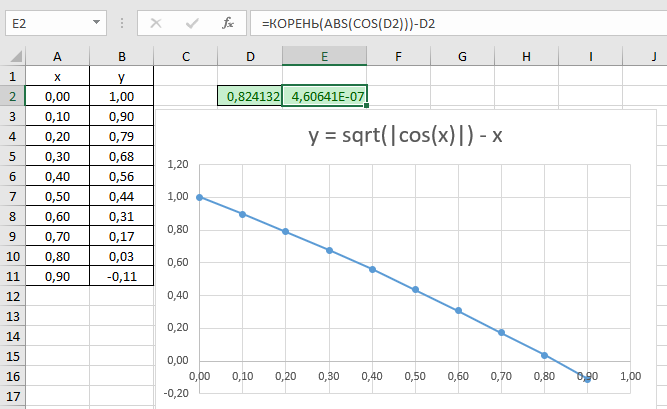
Выделим на рабочем листе любые две (желательно соседние) свободные ячейки, например, D2 и E2.

В ячейку D2 введем найденное приближенное значение корня уравнения f(x)=0;

В ячейку E2 вводим формулу =КОРЕНЬ(ABS(COS(D2)))-D2.

Устанавливаем точность вычислений: Сервис – Параметры – Вычисления для чего в окне "Относительная погрешность" устанавливаем заданное значение 0,000001.

Решаем уравнение f(x)=0, для чего выбираем команду Сервис – Подбор параметра

****

1. **Сводная таблица полученных результатов**

|  |  |  |
| --- | --- | --- |
| Метод | Значение аргумента x\* | Значение функции f(x\*) |
| Независимый метод – MS Excel | 0,82413199358389 | 4,60641E-07 |
| Оптимальный пассивный поиск | 0.8241319999999999 | 4.513682447804257E-7 |
| Метод половинного деления | 0.8241321314711569 | 2.613540800444625E-7 |
| Метод золотого сечения | 0.8241324121069318 | 1.4424649819044788E-7 |
| Метод бисекции | 0.824131965637207 | 2.510335663231407E-13 |
| Метод Ньютона | 0.8241323123025531 | 1.962303827688911E-27 |
| Метод итераций | 0.8238948038709392 | 1.1782096254989085E-7 |
| Метод квадратичной интерполяции | 0.8241320865126406 | 1.0649273695544908E-13 |
| Метод Фибоначчи | 0.8241323123025224 | 1.232595164407831E-32 |

**Выводы**

В лабораторной работе №5 студентом группы БПО-18-01 Исангуловым Ринатом реализованы 8 численных методов на языке программирования высокого уровня java. Как следует из результатов работы всех восьми программ, во всех случаях удалось добиться приемлемого совпадения результатов расчетов с решением, полученным в MS Excel. Из полученных результатов следует, что все восемь программ работают правильно, причем наибольшей точностью обладает метод бисекции.