СОДЕРЖАНИЕ

Введение.............................................................................................................3

Глава 1. Теоретическая часть............................................................................4

Глава 2. Практическая часть..............................................................................8

Заключение........................................................................................................12

Список использованной литературы................................................................13

Приложение 1. Код программы........................................................................14

Введение

Актуальность данной работы обусловлена важностью задачи поиска максимума или минимума функции для различных областей математики и физики.

Целью данной работы является нахождение функции в приближенном виде с помощью разложения в ряд Тейлора и интерполяции многочленом Лагранжа решения задачи Коши для обыкновенного дифференциального уравнения, поиск максимального значения полученной функции.

Задачами работы являются написание программа на языке С++, построение графиков найденных функций.

Глава 1. Теоретическая часть

Необходимость поиска максимума или минимума функции часто возникает при решении задач из различных областей математики и физики.

Экстремумами функции *F*(*x*) на заданном множестве X являются точки этого множества, в которых значения функции принимают максимальные или минимальные значения.

Точка называется локальным минимумом (максимумом) функции *F*(*x*), если существует число такое, что для любой точки x из сколь угодно малой окрестности точки :

Наибольший из всех локальных максимумов называется глобальным максимумом. Аналогично определяется глобальный минимум.

Примеры экстремумов показаны на рисунке 1. Здесь имеются два минимума (*Хэ*1 и *Хэ*3), причём первый из них локальный, а второй – глобальный. Если предположить, что дальше экстремумов больше не будет, *Хэ*2 является глобальным максимумом функции.

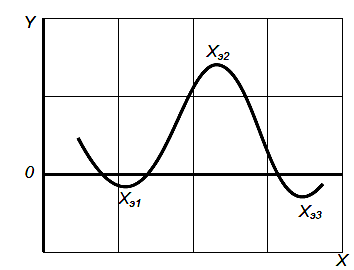


Рисунок 1 - Примеры экстремумов

Для нахождения экстремумов существуют аналитические и приближенные методы. Первые из них требуют знания производной данного уравнения с последующим нахождением её корня – точки, в которой производная превращается в нуль, является экстремумом функции. Исключениями могут быть функции с перегибами, но для них имеет место сохранение знака производной до и после точки с нулевым значением.

Численные методы позволяют находить экстремумы для любых уравнений с заданной точностью, они не требуют вычисления производных и позволяют находить экстремумы, которые локализованы в определённом интервале, но они требуют большого объёма вычислений. Развитие вычислительной техники позволило в настоящее время широко использовать последние методы.

Уравнения могут содержать несколько экстремальных точек, которые должны быть локализованы интервалами поиска. Ясно, что в точке экстремума производная превращается в нуль, а слева и справа от неё производные имеют различные знаки. Пользуясь этими условиями, мы может локализовать все возможные экстремумы функции.

Как обычно реализуются численные методы поиска экстремума, рассмотрим на графике, показанном на рисунке 2. Задаём интервал поиска от *Хлев* до *Хпр*, содержащий искомую точку экстремума, и начинаем его уменьшать. Но нам надо знать, в каком из новых интервалов находится экстремум. Для этого мы вычисляем не одно, а два значения функции и принимаем, что нужный нам интервал находится вокруг большей из этих точек (точка *Х*1 на рисунке 2), т.е. он должен быть взят от границы (*Хлев*) через максимальную точку (*Х*1) до второй вычисленной точки (*Х*2). Интервал от *Х*2 до *Хпр* отбрасываем.

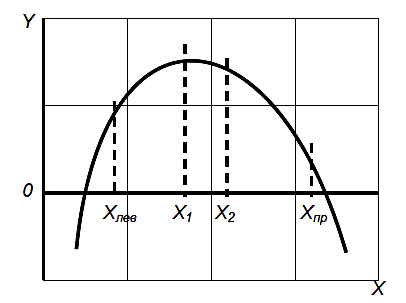


Рисунок 2 - Метод золотого сечения

Данную процедуру повторяем до тех пор, пока расстояние между *Хпр* и *Хлев* не станет меньше заданной погрешности вычисления. Ответом будет одна из двух средних точек, имеющая максимальное значение. Остаётся вопрос: как выбирать точки X1, X2 для вычисления новых значений? Это можно сделать методом золотого сечения.

По методу золотого сечения рассматриваемый отрезок делится в пропорции золотого сечения в обоих направлениях, то есть выбираются две точки x1 и x2 такие, что:

где - пропорция золотого сечения.

Таким образом:

То есть точка *x*1 делит отрезок [a,*x*2] в отношении золотого сечения. Аналогично *x*2 делит отрезок [*x*1,b] в той же пропорции. Это свойство и используется для построения итеративного процесса:

На первой итерации заданный отрезок делится двумя симметричными относительно его центра точками, и рассчитываются значения в этих точках.

После чего тот из концов отрезка, к которому среди двух вновь поставленных точек ближе оказалась та, значение в которой максимально (для случая поиска минимума), отбрасывают.

На следующей итерации в силу показанного выше свойства золотого сечения уже надо искать всего одну новую точку.

Процедура продолжается до тех пор, пока не будет достигнута заданная точность.

Алгоритм метода можно описать так:

Шаг 1. Задаются начальные границы отрезка [a, b] и точность .

Шаг 2. Рассчитывают начальные точки деления: , и значения в них целевой функции:, ,.

Если (для поиска max изменить неравенство на ), то

Иначе .

Шаг 3.

Если < , то и останов.

Иначе возврат к шагу 2.

Глава 2. Практическая часть

Найти максимальное значение функции

где

- интерполяционный многочлен, построенный по таблице решений дифференциального уравнения

, с шагом *h* (6)

Исходные данные:

*a* = 0, *b* = 1, *h* = 0.5 (7)

Решение:

1. Запишем функцию приближённо в виде многочлена с помощью разложения в ряд:

Разложим :

Для функции ряд Тейлора имеет вид:

отсюда

Отбрасываем члены при - , и окончательно получаем

2. Решаем диф. уравнение методом Рунге-Кутты 4-го порядка:

Найти функцию Y = Y(x), удовлетворяющую уравнению (для x>x0)

Y' = PravCh(x) , Y(x0) = Y0

Метод Рунге-Кутты четвёртого порядка для данной задачи имеет вид:

3. По полученным значениям получаем коэффициенты интерполяционного многочлена P2(x) .

Интерполяционный многочлен Лагранжа имеет вид:

4. Максимум функции на отрезке [a,b] найдём методом золотого сечения.

По результату вычислений искомый максимум функции F(x) равен 0.6465, максимум достигается в точке x = 0.999913.

Результат выполнения программы показан на рисунке 3.



Рисунок 3 - Максимум функции на отрезке [0,1]

В таблице 1 представлен результат вычислений.

Код программы на языке С++ описан в приложении 1.

Таблица 1. Вычисленные значения функций на заданном отрезке

|  |  |  |  |
| --- | --- | --- | --- |
| x | F(x)=|P2(x)-f(x)| | P2(x) | f(x) |
| 0 | 0 | 0 | 0 |
| 0,05 | 0,009077 | 0,010347 | 0,00127 |
| 0,1 | 0,013245 | 0,018394 | 0,005149 |
| 0,15 | 0,012421 | 0,024143 | 0,011722 |
| 0,2 | 0,006552 | 0,027592 | 0,02104 |
| 0,25 | 0,00438 | 0,028741 | 0,033122 |
| 0,3 | 0,020356 | 0,027592 | 0,047948 |
| 0,35 | 0,041314 | 0,024143 | 0,065457 |
| 0,4 | 0,067152 | 0,018394 | 0,085547 |
| 0,45 | 0,097718 | 0,010347 | 0,108065 |
| 0,5 | 0,132812 | 0 | 0,132812 |
| 0,55 | 0,17218 | -0,012646 | 0,159534 |
| 0,6 | 0,215512 | -0,027592 | 0,18792 |
| 0,65 | 0,262437 | -0,044837 | 0,217601 |
| 0,7 | 0,312525 | -0,064381 | 0,248144 |
| 0,75 | 0,365277 | -0,086224 | 0,279053 |
| 0,8 | 0,420127 | -0,110367 | 0,30976 |
| 0,85 | 0,476437 | -0,136809 | 0,339628 |
| 0,9 | 0,533493 | -0,16555 | 0,367942 |
| 0,95 | 0,590504 | -0,196591 | 0,393913 |
| 1 | 0,646598 | -0,229931 | 0,416667 |

Графики функций представлены на рисунке 4.

Рисунок 4 - Графики полученных функций.

Заключение.

В данной работе выполнен поиск максимального значения функции, полученной в приближенном виде с помощью разложения в ряд Тейлора и интерполяции многочленом Лагранжа решения задачи Коши для обыкновенного дифференциального уравнения. Программа выполнена на языке С++, построены графики.

Цель работы достигнута.

Список использованной литературы

1. Калиткин, Н.Н. Численные методы: [учебное пособие для студентов университетов и высших технических учебных заведений] / Н. Н. Калиткин; под ред. А. А. Самарского. – 2-е изд. – СПб. : БХВ-Петербург, 2011. – 586 с.

1. Вержбицкий, В.М. Основы численных методов: учебник для вузов / В. М. Вержбицкий. – М. : Высш. шк., 2009. – 840 с.
2. Волков, Е.А. Численные методы : учебное пособие / Е. А. Волков. – 5-е изд., стер. – СПб. [и др.] : Лань, 2008. – 248 с.
3. Демидович, Б.П. Основы вычислительной математики : учебное пособие / Б. П. Демидович, И.А. Марон. – 8-е изд., стер. –СПб. : Лань, 2011. – 672 с.
4. Лапчик, М. П. Элементы численных методов / М. П. Лапчик, М. И. Рагулина, Е. К. Хеннер. – М. : Академия, 2007. – 224 с.
5. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М: БИНОМ, 2004. – 636с.
6. Рябенький, В. С. Введение в вычислительную математику / В. С. Рябенький. – М. : Физмагиз, 2000. – 294 с.
7. Самарский, А. А. Введение в численные методы : учебноепособие для вузов / А. А. Самарский; МГУ им. М. В. Ломоносова. – СПб. : Лань, 2005. – 288 с.

9. Тарасов В. Н. Математическое программирование. Теория, алгоритмы, программы: учеб. пособие / В. Н. Тарасов, Н. Ф. Бахарева; Поволжский гос. ун-т телекоммуникаций и информатики. - Изд. 3-е, перераб. - Самара : ПГУТИ, 2017. - 222 с.

10. Белева Л. Ф. Программирование на языке С++ [Электронный ресурс]: учеб. пособие / Л. Ф. Белева. - Саратов : Ай Пи Эр Медиа, 2018. - 81 с.

Приложение 1. код программы

#include <iostream>

#include <stdio.h>

#include "math.h"

float A=0.0, B=1.0, h=0.5;

float f(float x){

float f0=-1.0/12.0, f1=-1.0/6.0, f2=1.0/6.0, f3=0.5, f4=0, f5=0; //koef. f(x)

return f0\*pow(x,5)+f1\*pow(x,4)+f2\*pow(x,3)+f3\*pow(x,2)+f4\*x+f5;

}

//Runge-Kutta

int k;

const int n2=2; // =(b-a)/h

float x2[n2], y2[n2]; //razbienie otr.

float K1, K2, K3, K4;

float PravCh(float x){

return sin(x)\*cos(x);

}

//interpol. polinom

float P2(float x, float\* x2, float\* y2, int n2){

float \_P2 = 0; int k,i;

float Proizv = 1;

for(i=0;i<n2;i++){

for(k=0;k<n2;k++) {

if(k!=i)Proizv \*= (x-x2[k])/(x2[i]-x2[k]);

}

\_P2 += y2[i]\*Proizv;

}

return \_P2;

}

//maximum

float eps=0.0001; //pogreshn

float x, max, aa, bb;

float fi, X1, X2, Y1, Y2;

float F(float x, float\* x2, float\* y2, int n2){

return fabs(P2(x,x2,y2,n2)-f(x));

}

//

float xRez[20], FRez[20];//для записи результата

int main(int argc, char \*argv[])

{

FILE\* file1; int i=0;

//Runge-Kutta

for(k=0;k<n2;k++) x2[k]=A+k\*h;

y2[0]=A;

for(k=0;k<n2;k++){

K1=h\*PravCh(x2[k]); K2=h\*PravCh(x2[k]+0.5\*h);

K3=h\*PravCh(x2[k]+0.5\*h); K4=h\*PravCh(x2[k]+h);

y2[k+1]=y2[k]+(1.0/6.0)\*(K1+2\*K2+2\*K3+K4);

}

//

max=0.0, x=0.0, aa=A, bb=B, fi = 1.618;

X1=aa, X2=bb;

while(X2-X1 > eps) {

X1 = bb-(bb-aa)/fi; X2 = aa+(bb-aa)/fi;

Y1 = F(X1,x2,y2,n2); Y2 = F(X2,x2,y2,n2);

if(Y1<=Y2) aa = X1;

else bb = X2;

}

x = (aa+bb)/2.0;

max = F(x,x2,y2,n2);

std::cout << "\t x="<<x<<" max="<<max<<"\n";

//

for(i=0;i<=20;i++){

xRez[i] = A+i\*(B-A)/20.0; FRez[i] = F(xRez[i],x2,y2,n2);

}

fopen\_s(&file1, "result\_x.txt", "w");//Открывается файл для записи результатов

for(i=0;i<=20;i++)fprintf(file1,"%f\n",xRez[i]); fclose(file1);

fopen\_s(&file1, "result\_F.txt", "w");//Открывается файл для записи результатов

for(i=0;i<=20;i++)fprintf(file1,"%f\n",FRez[i]); fclose(file1);

fopen\_s(&file1, "result\_P2.txt", "w");//Открывается файл для записи результатов

for(i=0;i<=20;i++)fprintf(file1,"%f\n",P2(xRez[i],x2,y2,n2)); fclose(file1);

fopen\_s(&file1, "result\_ff.txt", "w");//Открывается файл для записи результатов

for(i=0;i<=20;i++)fprintf(file1,"%f\n",f(xRez[i])); fclose(file1);

std::cout << "files ready"<<"\n";

}