Методы одномерной оптимизации.

Дана некоторая функция f(x) от одной переменной x, надо определить такое значение x*, при котором функция f(x) принимает экстремальное значение. Под ним обычно понимают минимальное или максимальное значения.

В общем случае функция может иметь одну или несколько экстремальных точек. Нахождение этих точек с заданной точностью можно разбить на два этапа. Сначала экстремальные точки отделяют, т.е. определяются отрезки, которые содержат по одной экстремальной точке, а затем уточняют до требуемой точности ε.

Отделение можно осуществить, как графически, так и табулированием. Все методы уточнения точек экстремумов будем рассматривать относительно уточнения минимума на заданном

Особенности исследуемых экспериментальных областей и их ограничения

Методы оптимизации позволяют достигать *локальной оптимизации,* но НЕ глобальной.

Исследуемая область:

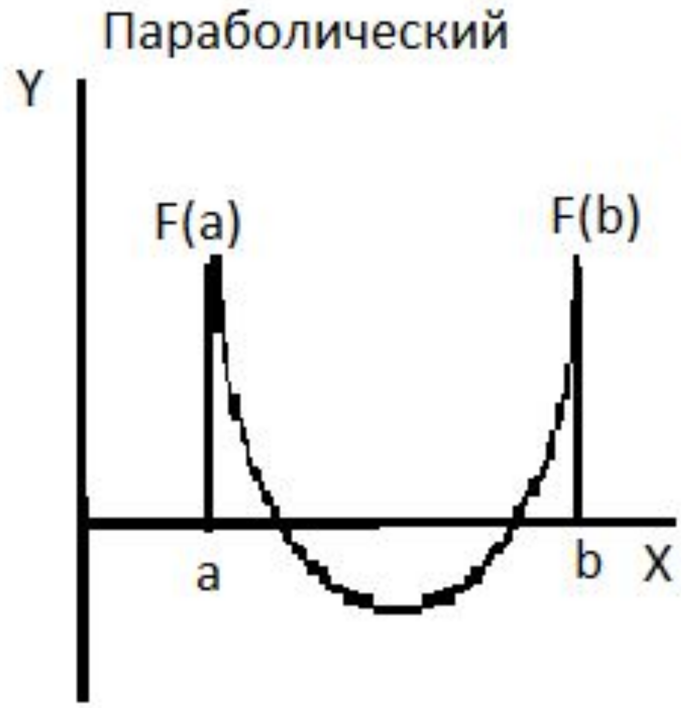
Монотонность (нет точек перегиба, нет производных равных нулю, нет экстремумов).

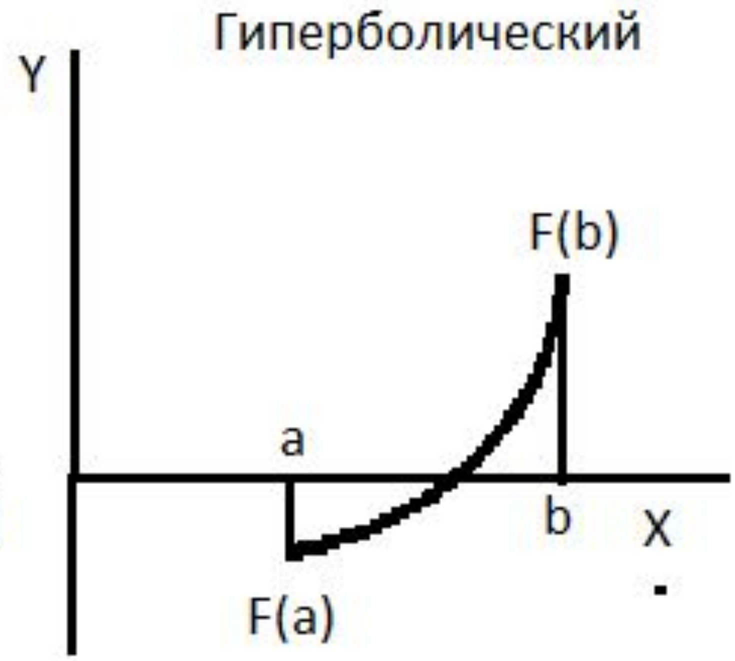
Время оптимизации (на выбор варианта).

Ограничения:

- *Выпуклые области* – области, в которых нельзя найти такого отрезка, концы которого принадлежат области, а сам он пресекает ее границы;
- *Вогнутые области* – те, в которых можно найти отрезок, концы которого принадлежат области, а сам он пересекает ее границы.

Виды функций





Метод деления отрезка пополам.

Метод деления отрезка пополам

1. Дан отрезок [a;b] на котором определена функция $f(x)$ и точность ε. Надо уточнить точку

минимума с заданной точностью. Введём новое обозначение точек $x1=a$ и $x4=b$.

2. Делим отрезок пополам и определяем точку середины $x2=(x4+x1)/2$ и точку $x3$, отстоящую на

незначительное расстояние от середины $x3=x2+\varepsilon/100$. Вычисляем значения функции в этих

точках $F2=f(x2)$ $F3=f(x3)$.

3. Определяем новый отрезок, содержащий точку экстремума, сравнив значения функций $F2$

и $F3$. Если $F2 < F3$, то границы нового отрезка определим как $x1=x1$, а $x4=x3$, иначе $x1=x2$, а

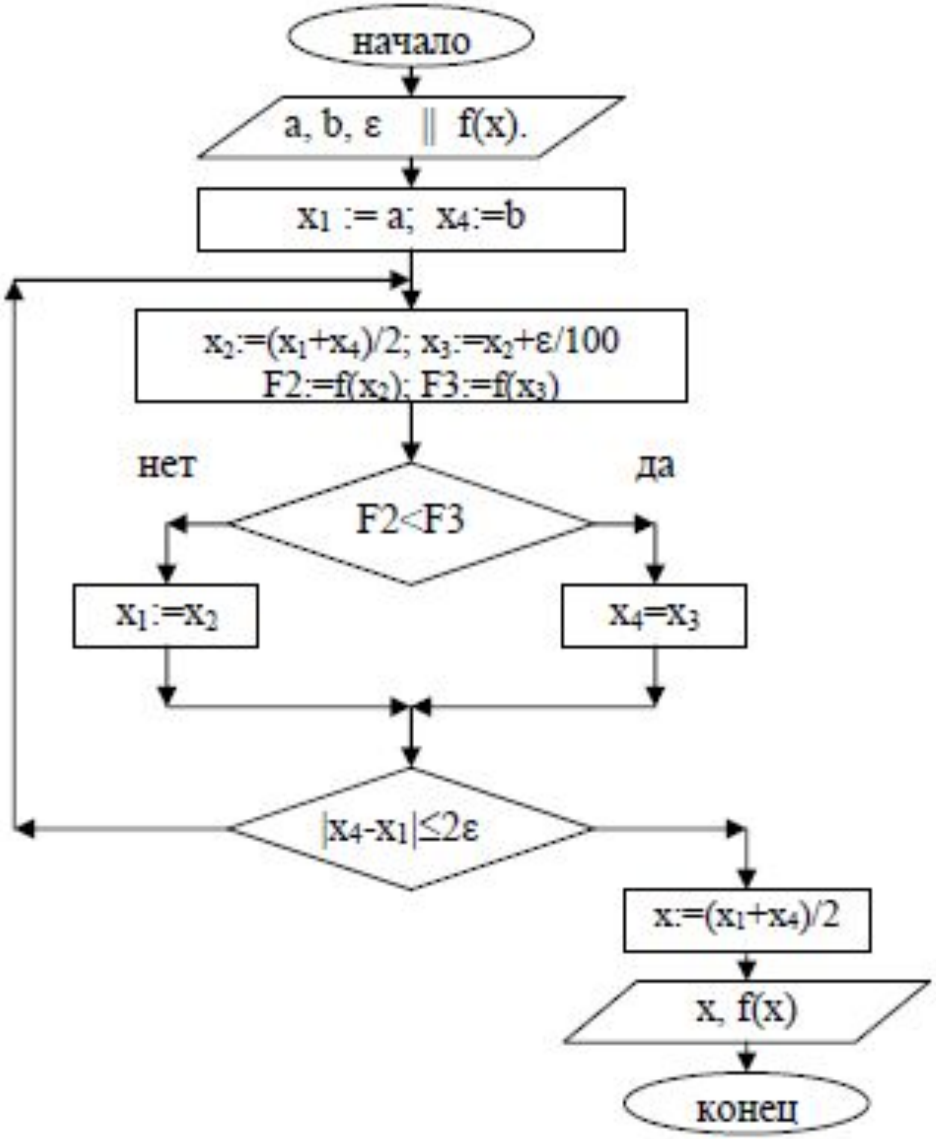
$x4=x4$.

4. Проверяем условие окончания итерационного процесса $| x4-x1 | \le 2\varepsilon$. Если оно выполняется,

то определим решение, как $x=(x4+x1)/2$ и значение функции в этой точке $f(x)$. Иначе перейдем

на пункт 2.

Метод деления отрезка пополам



Эффективность метода

Введем понятие эффективности, как отношение доли сокращения отрезка к количеству вычисления функции на одной итерации

- Эффективность метода Q≈0,5/2=0,25

Метод деления отрезка пополам

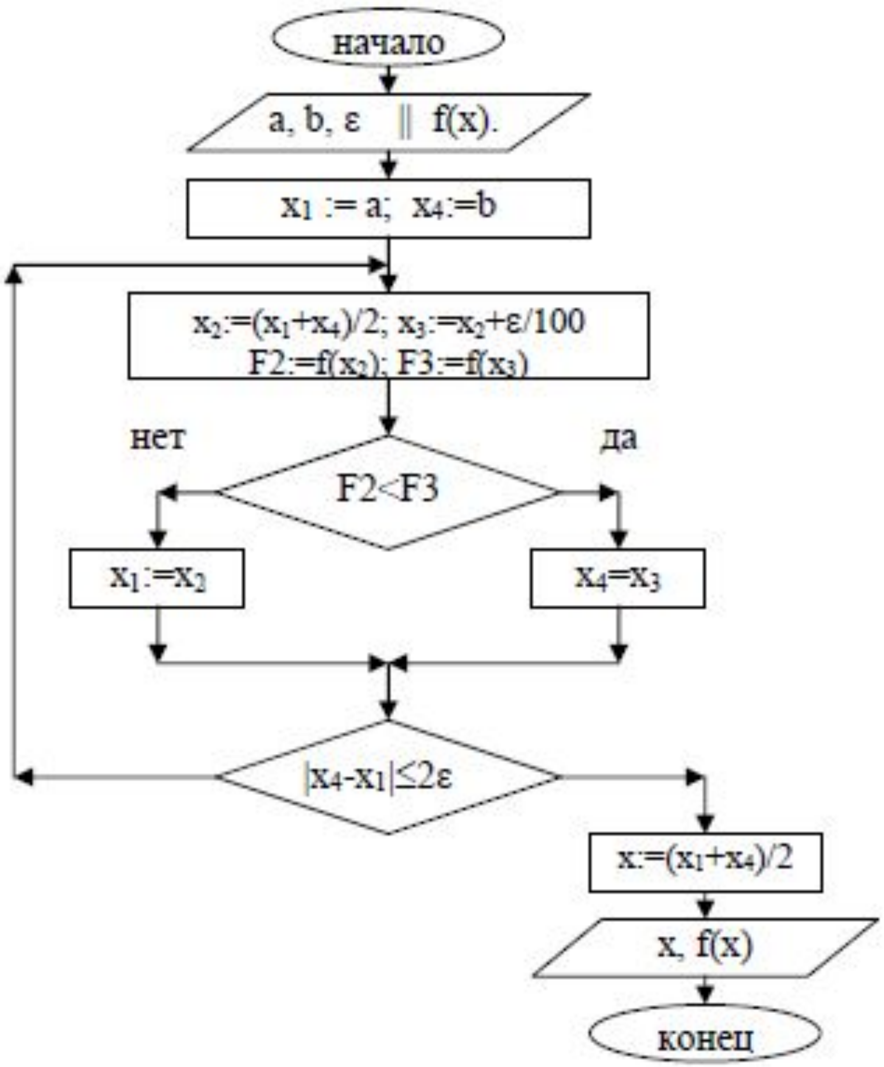
- function [c] = bisec(f,a,b,e)
- while abs(b-a)>e
-     c=a+(a+b)/2;
-     if f(a)*f(c)>0
-         a=c;
-     else
-         b=c;
-     end
- end
- disp(['Ответ x=' num2str(c,5)]);
- end

Метод деления на три равных отрезка

Метод деления на три равных отрезка

- 1. Дан отрезок [a;b] на котором определена функция $f(x)$ и точность ε. Надо уточнить точку
- минимума с заданной точностью. Введём новое обозначение точек $x1=a$ и $x4=b$. И вычислим
- $Z=1/3$.
- 2. Делим отрезок на три равные части и определяем точку $x2=x1+Z(x4-x1)$ и точку $x3=x4-Z(x4-x1)$.
- Вычисляем значения функции в этих точках $F2=f(x2)$ $F3=f(x3)$.
- 3. Определяем новый отрезок, содержащий точку экстремума, сравнив значения функций $F2$
- и $F3$. Если $F2 < F3$, то границы нового отрезка определим как $x1=x1$, а $x4=x3$, иначе $x1=x2$, а
- $x4=x4$.
- 4. Проверяем условие окончания итерационного процесса $| x4-x1 | \leq 2\varepsilon$. Если оно выполняется,
- то определим решение, как $x=(x4+x1)/2$ и значение функции в этой точке $f(x)$. Иначе перейдем
- на пункт 2.

Метод деления на три равных отрезка



Эффективность метода

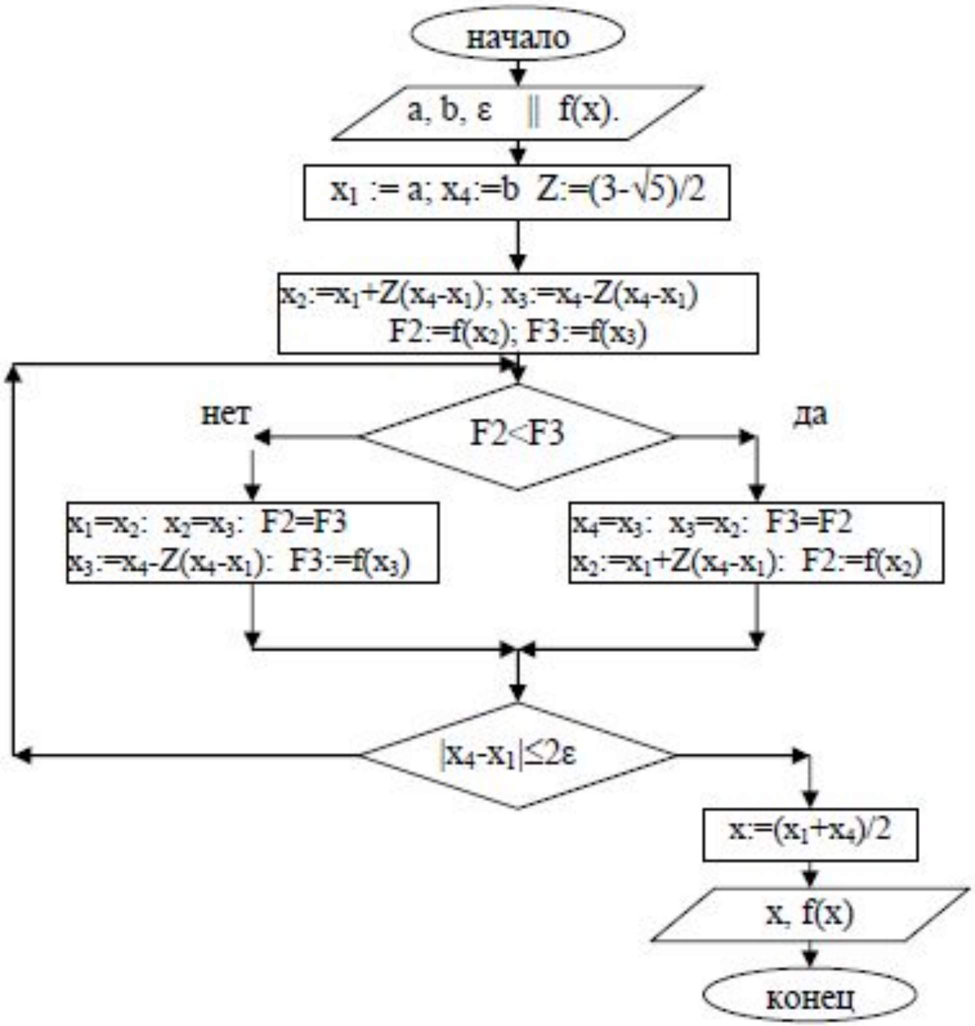
вычисления функции на одной итерации тогда: $Q=0{,}33/2\approx 0{,}17$.

Метод золотого сечения.

Метод золотого сечения.

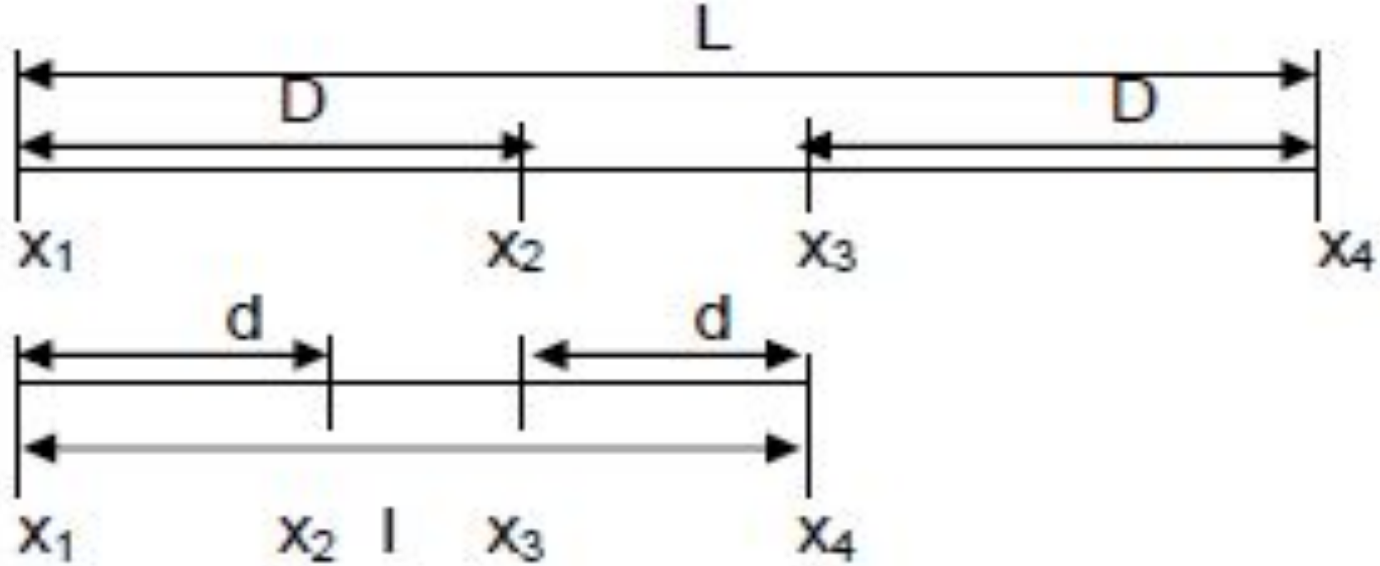
1. Дан отрезок [a;b] на котором определена функция $f(x)$ и точность ε. Надо уточнить точку минимума с заданной точностью. Вычислим $Z=(3-(5)^{-2})/2$ *и введём новое обозначение точек* $x_1=a$ и $x_4=b$
2. Делим отрезок на три части и определяем точку $x_2=x_1+Z(x_4-x_1)$ и точку $x_3=x_4-Z(x_4-x_1)$. Вычисляем значения функции в этих точках $F2=f(x_2)$ $F3=f(x_3)$.
3. Определяем новый отрезок, содержащий точку экстремума, сравнив значения функций F2 и F3. Если $F2 < F3$, то пункт 4 иначе пункт 5
4. границы нового отрезка определим как $x_1=x_1$, $x_4=x_3$, а $x_3=x_2$, $F3=F2$, $x_2=x_1+Z(x_4-x_1)$ и $F2=f(x_2)$ пункт 6
5. границы нового отрезка определим как $x_1=x_2$, $x_4=x_4$, а $x_2=x_3$, $F2=F3$, $x_3=x_4-Z(x_4-x_1)$ и $F3=f(x_3)$ пункт 6
6. Проверяем условие окончания итерационного процесса $| x_4-x_1 | \le 2\varepsilon$. Если оно выполняется, то определим решение, как $x=(x_4+x_1)/2$ и значение функции в этой точке $f(x)$. Иначе перейдем на пункт 3.

Метод золотого сечения.



Эффективность метода

Попробуем разбивать отрезок на такие части, чтобы одну из двух точек и соответствующее значение функции мы могли использовать на следующей итерации



Метод золотого сечения.

$$\frac{D}{L}=\frac{d}{l};\, d=L-2D;\, l=L-D;\, \frac{D}{L}=\frac{(L-2D)}{(L-D)};\, DL-D^2=L^2-2DL$$

делим на L^2 $\quad \frac{D}{L}-\left(\frac{D}{L}\right)^2-1+2\cdot\frac{D}{L}=0$

Заменяем $Z=\frac{D}{L}$ $\quad Z^2-3Z+1=0$

Решая получим $Z=\frac{(3-\sqrt{5})}{2}\approx 0.3819$

Эффективность метода

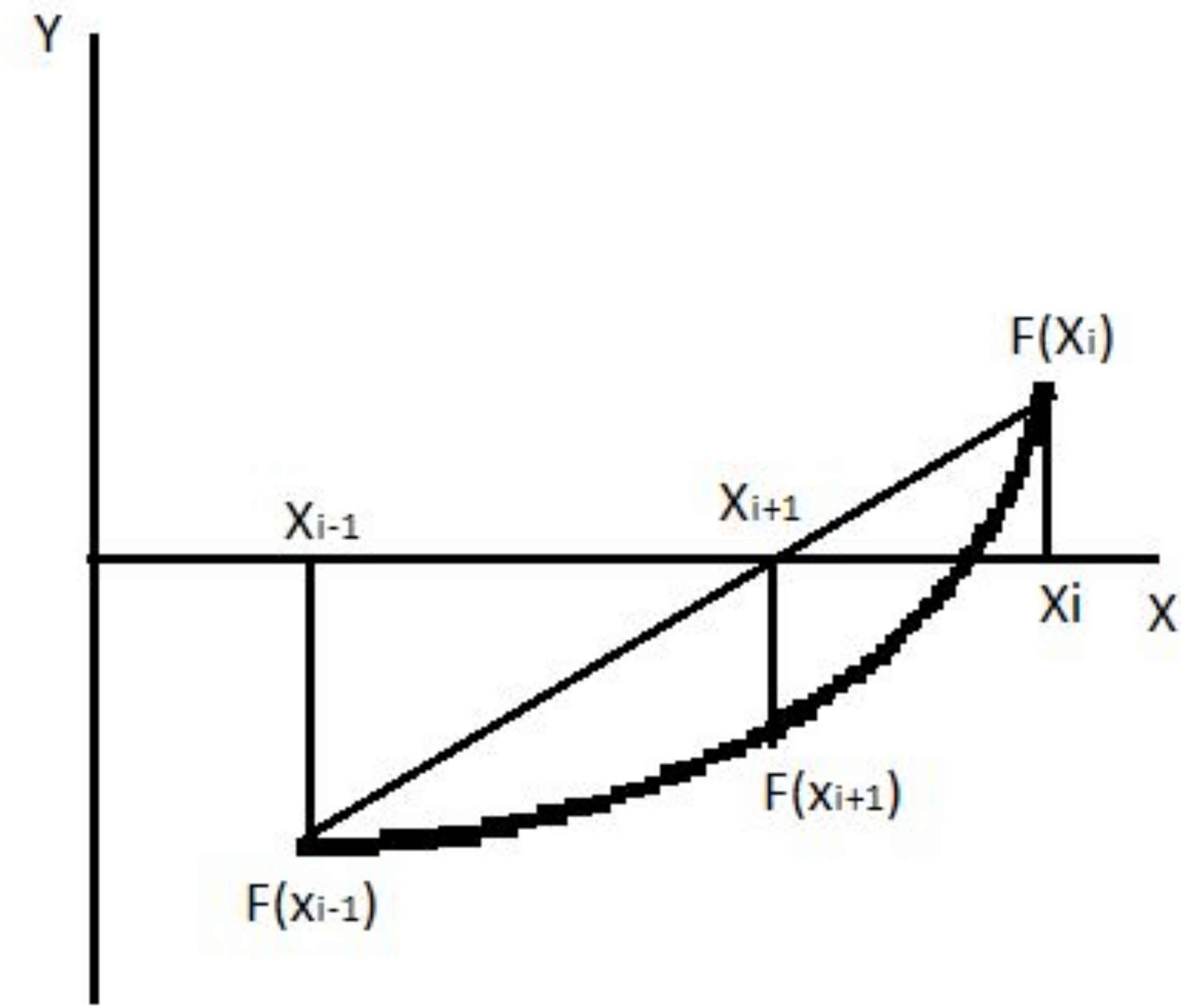
$$Q=0,38/1 \approx 0,38$$

Метод золотого сечения.

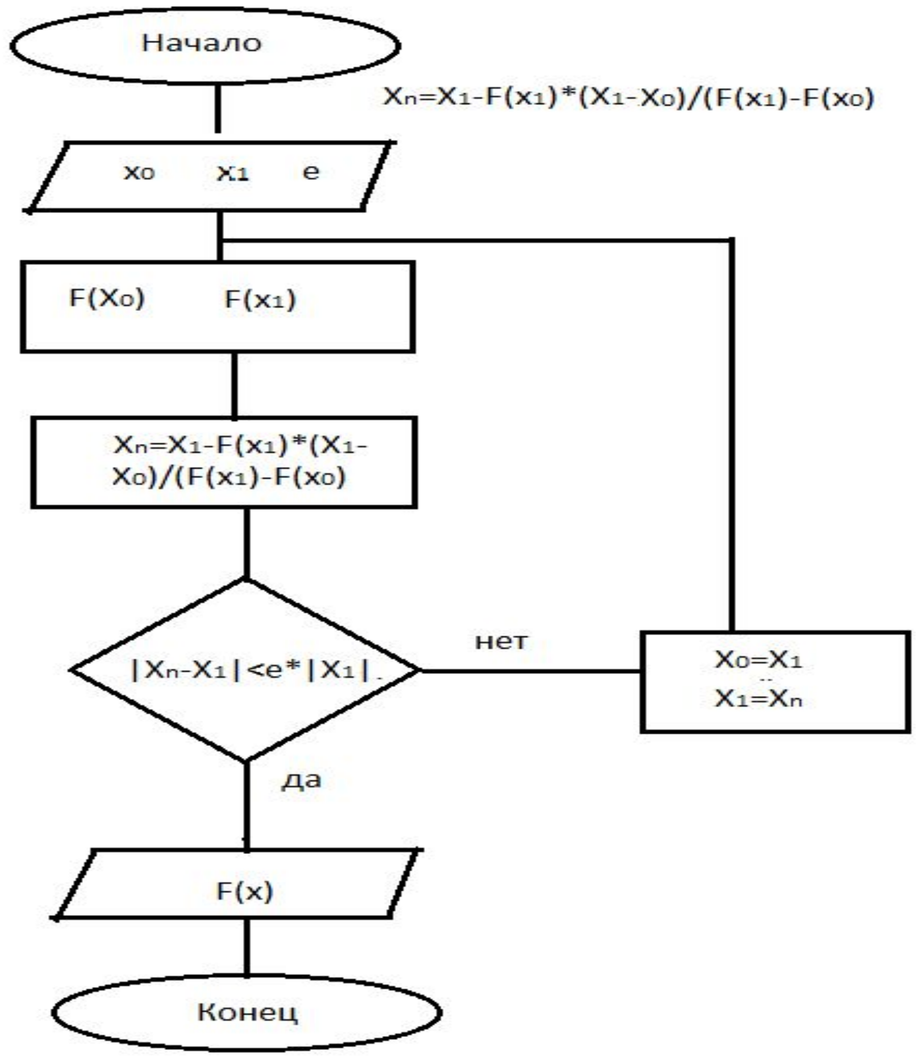
```
function [ x] = extremum( f,a,b,e )
x1=a+0.382*(b-a);
x2=b-0.382*(b-a);
while abs(b-a)>e
    if f(x1)<f(x2)
        b=x2;
        x2=x1;
        x1=a+0.382*(b-a);
    else
        a=x1;
        x1=x2;
        x2=b-0.382*(b-a);
    end
end
x=(a+b)/2;
y=f(x);
disp(['Ответ x=' num2str(x,15)]);
disp(['ответ y=' num2str(y,16)]);
end
```

Метод хорд

Метод хорд



Метод хорд



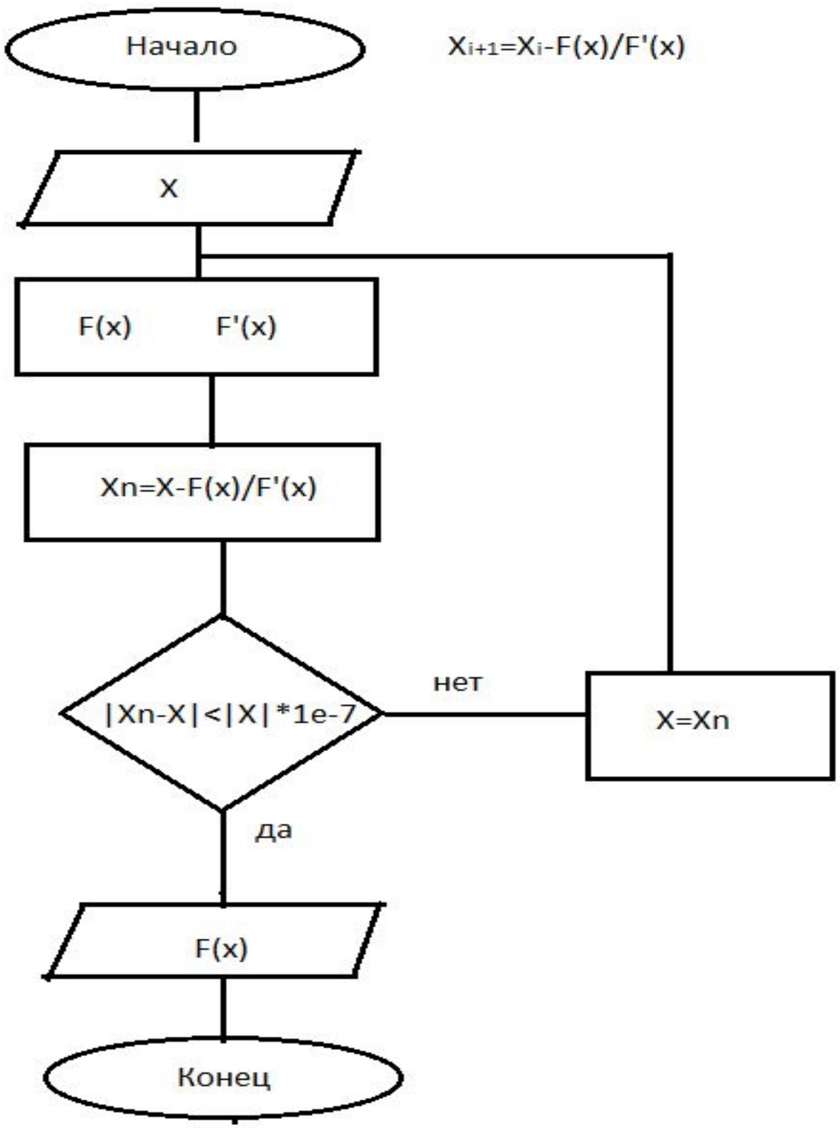
Начало
Xn=X1-F(x1)*(X1-X0)/(F(x1)-F(x0)
x0 x1 e
F(X0) F(x1)
Xn=X1-F(x1)*(X1-X0)/(F(x1)-F(x0)
|Xn-X1|<e*|X1|
нет
X0=X1
X1=Xn
да
F(x)
Конец

Метод хорд

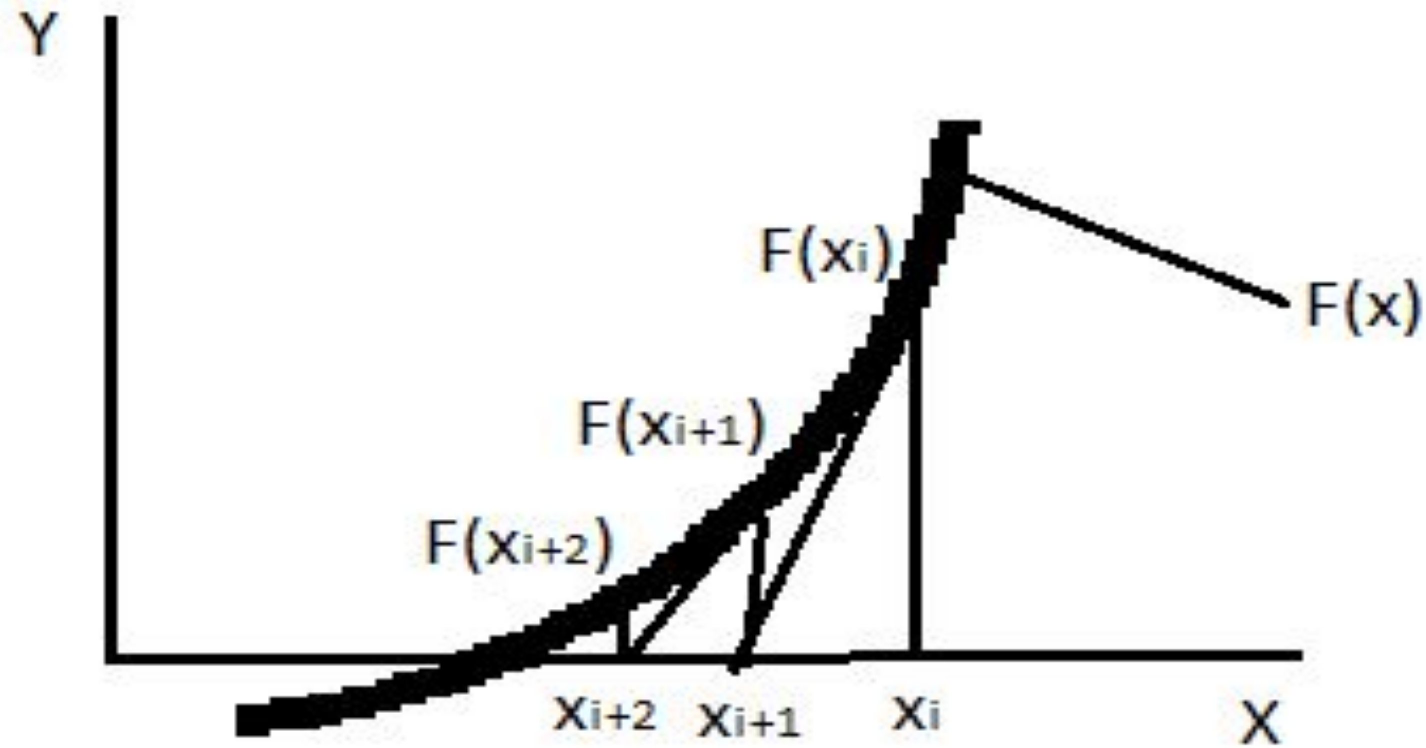
- function [x] = hord(f ,a , b, e)
- while abs(b-a)>e
-     x=(a*f(b)-b*f(a))/(f(b)-f(a));
-     if f(a)*f(x)>0
-         a=x;
-     else
-         b=x;
-     end
- end
- disp(['Ответ x=' num2str(x,3)]);
- end

Метод Ньютона (Метод касательных)

Метод Ньютона (Метод касательных)



Метод Ньютона (Метод касательных)



Метод Ньютона (Метод касательных)

- function [x] = kas(f,p,x1,e)
- x=x1-f(x1)/p(x1);
- while abs(x-x1)>e
-     x1=x;
-     x=x1-f(x1)/p(x1);
- end
- disp(['Ответ x=' num2str(x,5)]);
- end

Метод Фибоначчи

- %метод поиска минимума с использованием чисел фибоначчи (1-мерный)
- function [x,y]=Fib1()
- %Ввод исходных данных
- eps=input('точность=');
- xleft=input('левый край поиска=');
- xright=input('правый край поиска=');
- F(1)=1;
- F(2)=2;
- s=2;
- N=abs(xright-xleft)/eps;
- % определяются числа Фибоначчи
- while N>F(s)
-     s=s+1;
-     F(s)=F(s-1)+F(s-2);
- end %конец while
- h=(xright-xleft)/F(s);
- x1=xleft+h*F(s-2);
- R1=f(x1); % вызов функции y=f(x)
- x2=x1+h*F(s-3);
- R2=f(x2); % вызов функции y=f(x)
- k=s-3;

Метод Фибоначчи

```
x3=0;
while k>1 %основной цикл
   k=k-1
   if R2<R1
      x3=x2+h*F(k)
   else
      x3=x1-h*F(k)
   end %конец if
   x1=x2;
   R1=R2;
   x2=x3;
   R2=f(x3)
end %конец while
% Вывод текстом
Sx=strcat('при x=',num2str(x3));
Sy=strcat('функция минимальна и равна ',num2str(R2));
disp(Sx)
disp(Sy)
% Построение графика
x1=xleft:h:xright;
y1=(x1+2).*(x1-4);
plot(x1,y1,'k-');
grid on
title('y=(x+2)(x-4)')
xlabel('X');
ylabel('Y');
end
```

метод встроенной функцией в пакет

- function[x,y]=VF1
- %метод встроенной функцией 1-мерный
- xleft=input('введите левую границу диапазона поиска!');
- xright=input('введите правую границу диапазона поиска!');
- [x,y]=fminbnd(@f,xleft,xright);
- sx=strcat('при x=',num2str(x));
- sy=strcat('функция минимальна и равна',num2str(y));
- disp(sx)
- disp(sy)
- h=0.1;
- x1=xleft:h:xright;
- y1=f(x1);
- plot(x1,y1,'k-');
- grid on
- title('y=(x+2)(x-4)')
- xlabel('X');
- ylabel('Y');
- end

- function [x] = iterac(f,a,e)
- x0=a;
- x=f(x0);
- while abs(x-x0)>e
-     x0=x;
-     x=f(x0);
- end
- disp(['Ответ x=' num2str(x,6)]);
- end