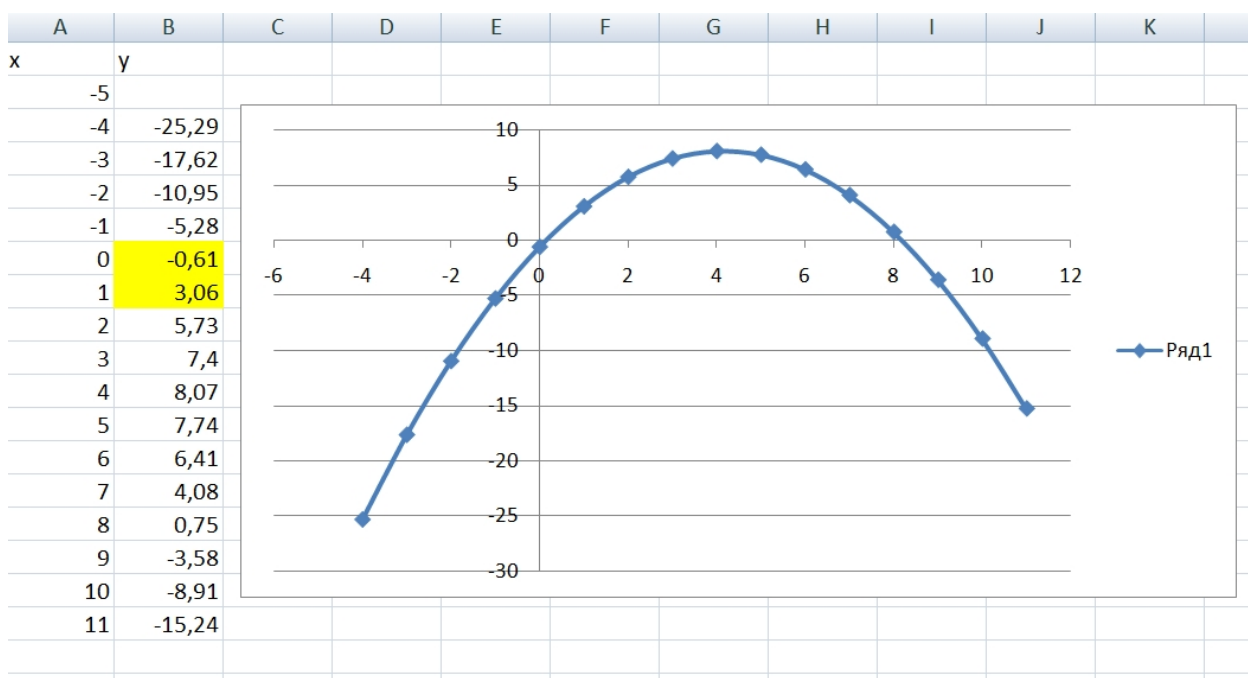


Завдання для вхідної лабораторної роботи з дисципліни «Математичне моделювання і оптимізація в енергетиці»

Досліджуємо суть методів ітерацій, опираючись на матеріал з попередніх дисциплін, де розглядалися числові методи ітерації (метод половинного розподілу, метод «золотого перетину», метод послідовних ітерацій, метод хорд, метод дотичних та інші).

Головна умова більшості ітераційних методів: значення функції на границях інтервалу дослідження повинні бути різних знаків.



Для задач оптимізації за методами розрахунок починається не з самої функції, а з її першої похідної заданої функції: точка, де графік похідної перетинає вісь Ox (тобто похідна дорівнює нулю), і є точкою екстремуму (мінімуму чи максимуму самої функції).

Для виконання вхідної роботи оберіть будь який ітераційний метод і змінюючи саму функцію, спробуйте знайти відповідь за допомогою програм m -файлів, створених у Матлаб (посібник з дисципліни «Числові методи») або тексти програм, наприклад:

Функція $f = (x - 4)^2 - 3x + 8$;
на інтервалі $[-7; 6]$ має точне значення $X_0 = 3.0000$

Сутність методу **половинного розподілу** у послідовному діленні заданого інтервалу навпіл і виключенні інтервалу, на якому значення функції на кінцях інтервалу мають однакові знаки. Процес пошуку продовжується до тих пір, поки значення модуля функції у серединній точці після n -ної ітерації не стане менше заданої погрішності.

Реалізація методу **половинного розподілу**:

```
e= input(['e=']);
a= input(['a=']), b= input(['b=']);
f=@ (x) (x-4)^2-3*x+8;
while f(a)*f(b)>0
    a= input(['a=']), b= input(['b=']);
end
x0=(a+b)/2;
f(x0);
while abs(f(x0))>e
    if f(a)*f(x0)<0
        b=x0;
    else a=x0;
end;
x0=(a+b)/2;
end;
x0, f(x0)
```

В основі методу «**золотого перетину**» покладено принцип ділення у пропорціях золотого перетину. У знайдених точках x_1 і x_2 знаходиться значення заданої функції і перевіряються наступні умови:

- якщо $f(x_1)*f(x_2)<0$ – то нові границі a і b будуть дорівнювати x_1 і x_2 відповідно;
- якщо $f(a)*f(x_2)>0$ – то $a=x_2$;
- якщо $f(x_1)*f(b)>0$ – то $b=x_1$.

Процес пошуку продовжується доки значення модуля функції у серединній точці після n -ної ітерації не стане менше заданої погрішності.

Реалізація методу «**золотого перетину**»:

```
k=0.618;
e= input(['e=']);
a= input(['a=']), b= input(['b=']);
f=@ (x) (x-4)^2-3*x+8;
while f(a)*f(b)>0
    a= input(['a=']), b= input(['b=']);
end;
x1= a+(1-k)*(b-a);
```

```

x2= b-(1-k)*(b-a);
while abs(x2-x1)>e
if f(a)*f(x1)>0
    a=x1;
else f(b)*f(x2)>0
    b=x2;
end;
x1= a+(1-k)*(b-a);
x2= b-(1-k)*(b-a);
end;
x0=(x1+x2)/2, f(x0)

```

Сутність методу **послідовних ітерацій** у послідовному виключенні інтервалів, на кінцях яких задана функція приймає значення з однаковими знаками. Обирається число кроків (інтервалів), на які розділяється заданий відрізок. Процес пошуку продовжується до тих пір, доки значення модуля функції у серединній точці після n-ної ітерації не стане менше заданої погрішності.

Реалізація методу **послідовних ітерацій**:

```

e= input(['e=']);
a= input(['a=']), b= input(['b=']);
f=@ (x) (x-4)^2-3*x+8;
while f(a)*f(b)>0
    a= input(['a=']), b= input(['b=']);
end
x0=a+e
while abs(f(x0))>e
    x0=x0+e;
end;
x0, f(x0)

```

У **методах хорд** та **дотичних** у якості наближень до кореня рівняння приймаються значення координат точок перетину хорди та дотичної з віссю ОХ. Формулу для розрахунку точок перетину отримують з рівняння хорди та дотичної. Ітераційний процес виконується доти, доки модуль значення функції в отриманій точці не буде меншим за погрішність, або доки різниця між значеннями точок у двох послідовних ітераціях не буде меншою за погрішність.

метод дотичних і хорд

Реалізація **методу хорд**:

```

e= input(['e=']);
a= input(['a=']), b= input(['b=']);
f=@ (x) (x-4)^2-3*x+8;
while f(a)*f(b)>0
    a= input(['a=']), b= input(['b=']);

```

```

end
c0=a-((f(a)*(b-a))/(f(b)-f(a)));
while abs(f(c0))>e
    if f(a)*f(c0)<0
        b=c0;
    else a=c0;
end;
c0=a-((f(a)*(b-a))/(f(b)-f(a)));
end;
c0, f(c0)

```

Реалізація методу дотичних:

```

e= input(['e=']);
a= input(['a=']), b= input(['b=']);
f=@ (x) (x-4)^2-3*x+8;
df=@ (x) 2*(x-4)-3;
while f(a)*f(b)>0
    a= input(['a=']), b= input(['b=']);
end
x=a;
x1=x-(f(x)/(df(x)));
while abs(f(x1))>e
    x=x1;
    x1=x-(f(x)/(df(x)));
end;
x, f(x1)

```