МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни
«Оптимізація інженерних та проектних рішень в електромеханіці»
для студентів спеціальності 141«Електроенергетика, електротехніка та електромеханіка»
(освітня програма «Електричні та електронні апарати» та «Електромеханічне обладнання енергоємних виробництв»)
усіх форм навчання

2024

Методичні вказівки до лабораторних робіт з дисципліни “Оптимізація інженерних та проектних рішень електричних в електромеханіці” для студентів спеціальності 141«Електроенергетика, електротехніка та електромеханіка» (освітня програма «Електричні та електронні апарати» та «Електромеханічне обладнання енергоємних виробництв») усіх форм навчання / Укл.: М. І. Коцур. – Запоріжжя: НУ "Запорізька політехніка", 2024. – 34 с.

Укладач: М. І. Коцур, доцент, канд. техн. наук

Рецензент: О. В. Близняков, доцент, канд. техн. наук

Відповідальний
за випуск: М. І. Коцур, доцент, канд. техн. наук

Затверджено
на засіданні кафедри
“Електричні та електронні апарати”
Протокол № 1
від “06“ серпня 2024р.

Рекомендовано до видання НМК
Електротехнічного факультету
Протокол № 3
від 24 жовтня 2024р.

ЗМІСТ

ВСТУП

При експериментальних дослідженнях часто зустрічається завдання представлення якоїсь залежності, заданої окремими точками, у вигляді гладкої функції. Завдання оптимізації - завдання, що дозволяють вибрати на множині можливих напрямків ті з них, які забезпечують найбільш ефективне (з погляду певного критерію) досягнення до поставленої мети.

У розв'язку будь-якої практичної оптимізаційного завдання існує кілька етапів.

На першому етапі визначають границі досліджуваної системи, що дозволяє сформулювати деяке завдання виду $f(x) \rightarrow min$, яке необхідно розв'язати.

Наступним етапом є вибір математичного методу, який би забезпечував одержання кінцевих результатів з найменшими витратами на обчислення або ж давав можливість одержати найбільший обсяг інформації про шуканий розв'язок. Вибір того або іншого методу в значній мірі визначається постановкою оптимального завдання, а також використанням математичних моделей об'єкта оптимізації.

У цей час розроблене безліч чисельних методів для завдань як безумовної, так і умовної оптимізації. Природнім є прагнення вибрати для розв'язку конкретного завдання найкращий метод, що дозволяє за найменший час використання ПК одержати розв'язок із заданою точністю.

Якість чисельного методу характеризується багатьма факторами: швидкістю збіжності, часом виконання однієї ітерації, обсягом пам'яті ПК, необхідним для реалізації методу, класом розв'язуваних завдань і т.д. Розв'язувані завдання також досить різноманітні: вони можуть мати високу й малу розмірність, бути унімодальними і багато екстремальними і т.д. Той самий метод, ефективний для розв'язку завдань одного типу, може виявитися зовсім неприйнятним для завдань іншого типу. Тому для розв'язку поставлених завдань дуже корисно знати основні властивості, специфіку методів оптимізації. Це забезпечує здатність правильно орієнтуватися в різних ситуаціях розрахунків, що виникають у їх процесі, і щонайкраще розв'язати завдання.

ЛАБОРАТОРНА РОБОТА 1

ЗНАХОДЖЕННЯ ЕКСТРЕМАЛЬНИХ ЗНАЧЕНЬ ФУНКЦІЇ ВІД ОДНІЄЇ ЗМІННОЇ

Мета роботи: Вивчення та надбання навичок щодо знаходження екстремальних значень функції $f(x)$.

1.1. **Короткі теоретичні відомості**

Функція $f(x)$ має ***локальний мінімум*** у точці $x = p$, якщо існує такий відкритий інтервал I, що містить p, при умові $f(p) \le f(x)$ для всіх $x \in I$.

Функція $f(x)$ має ***локальний максимум*** у точці $x = p$, при умові, що $f(x) \le f(p)$ для всіх $x \in I$. Якщо функція $f(x)$ має або локальний мінімум, або локальний максимум у точці $x = p$, то говорять, що вона має ***локальний екстремум*** у точці $x = p$.

Критерій першої похідної. Припустимо, що функція $f(x)$ безперервна на відрізку $I = [a;\ b]$. Крім того, припустимо, що $f'(x)$ визначена для всіх $x \in (a, b)$ за винятком, можливо, точки $x = p$.

Якщо $f'(x) < 0$ на інтервалі (a, p) і $f'(x) > 0$ на (p, b), то $f(p)$ — локальний мінімум.

Якщо $f'(x) > 0$ на інтервалі (a,p) і $f(x) < 0$ на (p, b), то $f(p)$ — локальний максимум.

Критерій другої похідної. Припустимо, що функція $f(x)$ безперервна на відрізку $[a;\ b]$ і $f'(x)$ і $f''(x)$ визначені на (a, b). Також припустимо, що $p \in (a, b)$ — критична крапка, у якій $f'(p) = 0$.

Якщо $f''(p) > 0$, то значення $f(p)$ є локальним мінімумом $f(x)$.

Якщо $f''(p) < 0$, значення $f(p)$ є локальним максимумом $f(x)$.

Якщо $f''(p) = 0$, то цей критерій не є остаточним.

1.2 **Хід роботи**

1.2.1 Визначити інтервали функції $f(x)$, де вона зростає (зменшується) згідно заданого варіанта (табл. 1.1).

1.2.2 Доведіть або спростуйте, що функція $f(x)$ є унимодальною у межах заданого інтервалу (табл. 1.1).

1.2.3. Оформити звіт з лабораторної роботи

Таблиця 1.1 – Варіанти завдань

Номер завдання	Номер варіанта	
1.2.1	1	$f(x) = 2x^3 – 9x^2 + 12x – 5;$
	2	$f(x)=x^2/(x + l);$
	3	$f(x) = (x + l)/x^2;$
	4	$f(x)=x^2-2;$
	5	$f(x) = 25x^3 – 8x^2 + 14x – 2;$
	6	$f(x) = 14x^3 +2x^2 - 4x +52;$
	7	$f(x)=x^2-x-2;$
	8	$f(x)=x^2-4x+6;$
	9	$f(x)=26x^2-4x+1;$
	10	$f(x)=18x^2+87x+6;$
2.2.2	1	$f(x) =x^2- 2x +1; [0;4]$
	2	$f(x) =cos(x); [0;3]$
	3	$f(x) = x^2-4\ \ [1; 10]$
	4	$f(x) = -x(3-x)^{5/3}; [0;3]$
	5	$f(x)=x^2-x-2; [0;78]$
	6	$f(x)=x^2-4x+6; [-5;14]$
	7	$f(x)=26x^2-4x+1; [0;35]$
	8	$f(x)=18x^2+87x+6; [1; 10]$
	9	$f(x) = 2x^3 – 9x^2 + 12x – 5; [0;78]$
	10	$f(x) = x^2-4\ \ [-8; 25]$

1.3 Приклад алгоритму розрахунку

Програма розрахунку виконується у середовищі MatLAB за допомогою створення m-файлу.

Для завдання 1.2.1. текст програми має вигляд:

```
dfun=inline('3.*x.^2+4*x+84');
x=[-1:0.03:1];
% the sign for the first interval and store it into “znak”. The value 0 for
% "-" and 1 for "+"
k=1;
if dfun(x(k))> 0
   znak(k)=1;
else znak(k)=0;
end
% store the first interval value at which the function corresponds to “znak”
interval=x(k);
% iteration for the derivative value, the change in function sign is stored
for i=2:length(x)
   y=dfun(x(i));
   if dfun(x(1))> 0
      z=1;
   else z=0;
   end
   if znak(k) ~= z
      k=k+1;
      interval(k)=x(i);
   end
end
if length(interval)== 1
   interval(k+1)=x(i);
end
```

Для завдання 1.2.2. текст програми має вигляд:

```
a=0;
b=4;
dfun=inline('2*x-2');
x=[a:0.1:b];
```

```
y=dfun(x);
k=1;
m(1,1)=0;
flag=0;
for i=1:length(x)
    if y(i)==0
        m(k)=i;
        k=k+1;
        flag=1;
    end
end
plot(x,y);
if flag==1
for i=1:length(m)
    if (y(m(i)-1) < y(m(i))) && (y(m(i)+1) > y(m(i)))
        unim=1;
    else
        unim=0;
        break;
    end
end
else
    unim=0;
end
```

1.4 **Контрольні запитання**

1. Надайте визначення локального мінімуму та локального максимуму функції *f(x)*;

2 Надайте визначення зростаючої та спадаючої функції;

3. Надайте визначення унімодальної функції;

ЛАБОРАТОРНА РОБОТА 2

ЗНАХОДЖЕННЯ ОПТИМАЛЬНОГО ЗНАЧЕННЯ ФУНКЦІЇ МЕТОДОМ ЗОЛОТОГО ПЕРЕТИНУ

Мета роботи: Вивчення та надбання навичок щодо знаходження оптимального значення функції методом золотого перетину.

2.1. **Короткі теоретичні відомості**

Нехай r Є [0; 1] — початковий інтервал. Якщо $0,5 < r < 1$, то $0 < 1 – r < 0,5$ і інтервал ділиться на три під інтервалів: $[0; 1 - r]$, $[1 - r; r]$ і $[r; 1]$. У процесі розв'язку використовується або стиск вправо й одержання нового інтервалу [0; r], або стиснення уліво й одержання інтервалу $[1- r; 1]$. Потім отримані під інтервали далі діляться на три під інтервалів в такому ж співвідношенні, як у випадку з інтервалом [0; 1].

Таким чином, необхідно так вибрати r, щоб одна зі старих точок була в правильному положенні щодо нового інтервалу. Із цього випливає, що відношення $(1 - r) : r$ таке ж, як і $r : 1$. Отже, r задовольняє рівнянню $1 - r = r^2$, яке можна записати у вигляді квадратного рівняння виду $r^2 + r - 1 = 0$.

Розв'язок r, що задовольняє нерівності $0,5 < r < 1$, дорівнює $r = (\sqrt{5} - 1) /2$.

Функція $f(x)$ повинна задовольняти особливим умовам, які гарантують існування дійсного мінімуму на інтервалі, щоб можна було використовувати пошук мінімуму функції $f(x)$ методом золотого перетину.

2.2 **Хід роботи**

2.2.1. Визначити локальний мінімум методом золотого перетину функції виду *f(x)* з точністю до третього знаку з кроком 0,1. Номер варіанта згідно табл.1.1.

2.2.1. Визначити локальний мінімум методом золотого перетину функції виду *f(x)* з точністю до восьми десятинних знаків з кроком 0,01.

2.2.3. Оформити звіт з лабораторної роботи.

2.3 **Приклад алгоритму розрахунку**

Програма розрахунку виконується у середовищі MatLAB за допомогою створення m-файлу.

```
function Maximum(a,b,eps)
function [S,E,G] =golden(f,a,b,delta,epsilon)
r1=(sqrt(5)-1)/2;
r2=r1^2;
h=b-a;
ya=feval(f,a);
yb=feval(f,b);
c=a+r2*h;
d=a+r1*h;
yc=feval(f,c);
yd=feval(f,d);
k=1;
A(k)=a;
B(k)=b;
C(k)=c;
D(k)=d;
while(abs(yb-ya)>epsilon)|(h>delta)
    k=k+1;
    if(yc<yd)
        b=d;
        yb=yd;
         d=c;
```

```
        yd=yc;
       h=b-a;
        c=a+r2*h;
        yc=feval(f,c);
    else
        a=c;
        ya=yc;
        c=d;
        yc=yd;
        h=b-a;
        d=a+r1*h;
        yd=feval(f,d);
    end
    A(k)=a;
    B(k)=b;
    C(k)=c;
    D(k)=d;
end
dp=abs(b-a);
dy=abs(yb-ya);
p=a;
yp=ya;
if(yb<ya)
    p=b;
    yp=yb;
end
G=[A' C' D' B'];
S=[p yp];
E=[dp dy];
```

2.4 **Контрольні запитання**

1. У чому полягає метод золотого перетину?
2. Чим відрізняється метод золотого перетину від методу інтервалів?
3. Чим відрізняється метод золотого перетину від методу Фібоначчі?

ЛАБОРАТОРНА РОБОТА 3

ЗНАХОДЖЕННЯ ОПТИМАЛЬНОГО ЗНАЧЕННЯ ФУНКЦІЇ МЕТОДОМ *P* - КВАДРАТИЧНОГО НАБЛИЖЕННЯ

Мета роботи: Вивчення та надбання навичок щодо знаходження оптимального значення функції методом *p* - квадратичного наближення.

3.1. **Короткі теоретичні відомості**

Знаходження мінімуму функції за допомогою квадратичного наближення, необхідно знайти таке значення $p_{\min}$, яке є наближенням до *p.* Для цього використовується поліном Лагранжа, який має вигляд:

$$Q(x)=\frac{y_0(x-p_1)(x-p_2)}{2h^2}-\frac{y_1(x-p_0)(x-p_2)}{h^2}+ \\ +\frac{y_2(x-p_0)(x-p_1)}{2h^2} \tag{3.1}$$

Похідна від *Q(x)* дорівнює

$$Q'(x)=\frac{y_0(2x-p_1-p_2)}{2h^2}-\frac{y_1(x-p_0-p_2)}{h^2}+\frac{y_2(2x-p_0-p_1)}{2h^2}; \tag{3.2}$$

Запишемо $Q'(x)=0$ у вигляді $Q'(p_0+h_{min})=0$:

$$0=\frac{y_0(2(p_0+h_{min})-p_1-p_2)}{2h^2}-\frac{y_1(4(p_0+h_{min})-2p_0-2p_2)}{2h^2}+ \\ +\frac{y_2(2(p_0+h_{min})-p_0-p_1)}{2h^2} \tag{3.3}$$

Помножимо кожний член в (3.3) на $2h^2$ та поєднаємо члени, які містять h_{min}:

$$-h_{min}(2y_0 - 4y_1 + 2y_2) = y_0(2p_0 - p_1 - p_2) - y_1(4p_0 - 2p_0 - 2p_2) + y_2(2p_0 - p_0 - p_1) = y_0(-3h) - y_1(-4h) + y_2(-h). \tag{3.4}$$

Останнє рівняння легко одержати відносно h_{min}:

$$h_{min} = \frac{h(4y_1 - 3y_0 - y_2)}{4y_1 - 2y_0 - 2y_2}. \tag{3.5}$$

Значення $p_{min} = p_0 + h_{min}$ є кращим наближенням до p, чому p_0. Тому можна замінити p_0 на p_{min} і повторити схему двох описаних вище процесів, щоб визначити нову довжину кроку h і нове h_{min}. Ітерація триває до необхідної точності.

3.2 **Хід роботи**

3.2.1 Визначити локальний мінімум, використавши квадратичне інтерполювання функції виду *f(x)* з точністю до восьми десятинних знаків. (номер варіанту за табл.1.1);

3.2.1. Визначити локальний мінімум, використавши квадратичне інтерполювання функції виду *f(x)* з точністю до десяти десятинних знаків.

3.2.3. Оформити звіт з лабораторної роботи.

3.3 **Приклад алгоритму розрахунку**

Програма розрахунку виконується у середовищі MatLAB за допомогою створення m-файлу.

```
function [p, yp, dp, dy] =quadmin (f1,a,b,delta, eps)
% - f - function
% - a, b - inteval extremal points
% - delta – admissible value for abciss
% - eps - admissible value for ordinate
```

```
% - p - min of abciss
% - yp - min ordinate
% - dp - p-error
% - dy - yp-error
% - P - iteration vector
p0=a; maxj=20;
maxk=30;
big=1e006;
err=0.9;
k=1;
cond=0;
h=1;
if(abs(p0)>1e004),
   h=abs(p0)/1e004;
end
while(k<maxk&&err>eps&&cond~=5)
   f2=(feval(f1,p0+0.00001)-feval(f1,p0-0.00001))/0.00002 ; if(f2>0),h=-
abs(h);end
   pl=p0+h; p2=p0+2*h;
   pmin=p0;
   y0=feval(f1,p0);
   y1=feval(f1,pl);
   y2=feval(f1,p2);
   ymin=y0;
   cond=0;
   j=0;
% h under condition yl<y0&yl<y2
   while(j<maxj&&abs(h)>delta&&cond==0)
      if (y0<=y1),
      p2=pl;
      y2=y1;
      h=h/2;
      pl=p0+h;
      y1=feval(f1,pl);
      else
         if(y2<y1),
         pl=p2;
         y0=y2;
         h=2*h;
         p2=p0+2*h;
         y2=feval(f1,p2);
```

```
        else
        cond=-1;
        end
    end
        j=j+1;
    if (abs(h)>big || abs(p0)>big),cond=5; end
 if(cond==5), pmin=pl;ymin=feval(f1,pl); end
% Quadratic interpolation for yp finding
    d=4*y1-2*y0-2*y2;
    if(d<0), hmin=h*(4*y1-3*y0-y2)/d;
    else
    hmin=h/3;
    cond=4;
  pmin=p0+hmin;
  ymin=feval(f1,pmin);
  h=abs(h);
  h0=abs(hmin);
  hl=abs(hmin-h);
  h2=abs(hmin-2*h);
% Next h value determination
  if(h0<h),
    h=h0;
  end
  if(hl<h),
    h=hl;
  end
  if(h2<h),
    h=h2;
  end
  if(h==0),
    h=hmin;
  end
  if(h<delta),
    cond=l;
  end
  if (abs(h)>big||abs(pmin)>big),
    cond=5;
  end
    end
% Stop criterion
e0=abs(y0-ymin);
```

```
el=abs(y1-ymin) ;
e2=abs(y2-ymin);
   if(e0~=0 || e0<err),
   err=e0;
   end
   if(el~=0 || err),
   err=el;
   end
   if (e2~=0 || 2<err),
   err=e2;
   end
   if (e0~=0 || el==0 || e2==0),
   err=0;
   end
   if(err<eps),
   cond=2;
   end
p0=pmin;
k=k+1;
   end
if(cond==2&&h<delta),
   cond=3;
end
end
p=p0;
dp=h;
yp=feval(f1,p);
dy=err;
```

3.4 **Контрольні запитання**

1. У чому полягає метод *p* - квадратичного наближення?
2. Чим відрізняється метод *p* - квадратичного наближення від методу інтервалів?
3. Чим відрізняється метод золотого перетину від методу *p* - квадратичного наближення?

ЛАБОРАТОРНА РОБОТА 4

ЗНАХОДЖЕННЯ ОПТИМАЛЬНОГО ЗНАЧЕННЯ ФУНКЦІЇ МЕТОДОМ НЕЛДЕРА-МИДА

Мета роботи: Вивчення та надбання навичок щодо знаходження оптимального значення функції методом Нелдера-Мида.

4.1. **Короткі теоретичні відомості**

4.1.1 Знаходження екстремальних значень функції $f(x, y)$

Функція $f(x, y)$ має локальний мінімум у точці (p, q), якщо $f(p, q) < f(x, y)$ для кожної точки $(x, y) \in R$.

Функція $f(x, y)$ має локальний максимум у точці (p,q), якщо $f(x, y) < f(p,q)$ для кожної точки $(x, y) \in R$.

Критерій другої похідної. Припустимо також, що функція $f(x, y)$ і її перша й друга часткові похідні безперервні в області R. Припустимо, що $(p, q) \in R$ — критична точка, у якій і $f_x(p, q) = 0$, і $f_y(p, q) = 0$. Часткові похідні вищого порядку використовуються для визначення природи критичної точки.

Якщо $f_{xx}(p, q) \cdot f_{yy}(p, q) - f^2_{xy}(p, q) > 0$ і $f_{xx}(p, q) > 0$, то $f(p, q)$ – локальний мінімум функції $f(x,y)$.

Якщо $f_{xx}(p, q) \cdot f_{yy}(p, q) - f^2_{xy}(p, q) > 0$ і $f_{xx}(p, q) < 0$, то $f(p, q)$ – локальний максимум функції $f(x,y)$.

Якщо $f_{xx}(p, q) \cdot f_{yy}(p, q) - f^2_{xy}(p, q) < 0$, то функція $f(x, y)$ не має локального екстремуму в точці (p, q).

Якщо $f_{xx}(p, q)\, f_{yy}(p, q) - f^2_{xy}(p, q) = 0$, цей критерій не є остаточним.

4.1.2 Метод Нелдера-Мида

Симплекс-Метод знаходження локального мінімуму функції від декількох змінних винайдений Нелдером і Мидом. Для двох змінних симплексом є трикутник, і метод — це схема пошуку, який порівнює значення функції в трьох вершинах трикутника. Найгірша вершина, у якій функція *f*(*x*,*y*,*z*,) приймає найбільше значення, відкидається й заміняється новою вершиною. Формується новий трикутник, і пошук триває. При цьому будується послідовність трикутників (вони можуть мати різну форму), значення функції у вершинах якої стають усе менше й менше. Зменшується розмір трикутника, і координати крапки мінімуму знайдені.

У формулюванні алгоритму використовується термін "симплекс" (узагальнений N-мірний трикутник). З його допомогою перебуває мінімум функції від *N* змінних. Він ефективний і компактний при обчисленні.

4.2 **Хід роботи**

4.2.1 Визначити локальний мінімум, використавши метод Нелдера-Мида для функцій виду *f(x,y)* з точністю до восьми десятинних знаків. (номер варіанту за табл.4.1);

4.2.2. Визначити локальний мінімум, використавши метод Нелдера-Мида для функцій виду *f(x,y,z)* з точністю до восьми десятинних знаків. (номер варіанту за табл.4.2)

4.2.3. Оформити звіт з лабораторної роботи.

Таблиця 4.1 – Варіанти завдання

Номер варіанта	Вид цільової функції та початкові умови
1	$f(x,y) = x^3 + y^3 - 3\cdot x - 3\cdot y + 5$; (1;2);
2	$f(x,y) = x^2 + y^2 + x - 2\cdot y - x\cdot y + 1$; (2;0);
3	$f(x,y) = x^2 + x\cdot y^2 - 3x\cdot y$; (2;1)
4	$f(x,y) = (x-y)/(x^2 + y + 2)$; (0;2)
5	$f(x,y) = 100(y - x^2)^2 + (1 - x)^2$; (0;2)
6	$f(x,y) = x^2 + 24\cdot x\cdot y^2 - 3x\cdot y$; (2;1)
7	$f(x,y) = 4\cdot x^3 + 7\cdot y^3 - 3\cdot x - 3\cdot y + 5$; (0;2);
8	$f(x,y) = 4(y - x^2) + (1 - x)^2$; (0;2)

Продовження таблиці 4.1

Номер варіанта	Вид цільової функції та початкові умови
9	$f(x,y) = 4x^2+4x\cdot y^2-8\cdot x\cdot y$; (3;10)
10	$f(x,y) = 4\cdot(x-y)/(x^2+y^2 -1)$; (0;2)
11	$f(x,y) = (x^2-y)/(x+y^2 -1)$; (0;0)
12	$f(x,y) = (y^2 - x^2)+ (1 - x)^2$; (0;0)

Таблиця 4.2 – Варіанти завдання

Номер варіанта	Вид цільової функції та початкові умови
1	$f(x,y,z) = x^3 + y^3-3\cdot x-3\cdot y + z$; (1;2;2);
2	$f(x,y,z) =x^2+y^2+ z^2+x-2\cdot z-y + 1$; (2;0;0);
3	$f(x,y,z) =2\cdot x^2+2\cdot y^2+z^2-2\cdot x\cdot y+ z\cdot y-7\cdot y-4\cdot z$; (1;1;1)
4	$f(x,y,z) =2\cdot x^2+2\cdot y^2+z^2-2\cdot x\cdot y+ z\cdot y-7\cdot y-4\cdot z$; (0;1;0)
5	$f(x,y,z) =2\cdot x^2+2\cdot y^2+z^2-2\cdot x\cdot y+ z\cdot y-7\cdot y-4\cdot z$; (0;0;1)
6	$f(x,y,z) =x^2+y^2+z^2-4\cdot x\cdot y+ z\cdot y-7\cdot y-8\cdot x$; (1;1;1)
7	$f(x,y,z) =x^2+y^2+z^2-4\cdot x\cdot y+ z\cdot y-7\cdot y-8\cdot x$; (0;1;0)
8	$f(x,y,z) =x^2+y^2+z^2-4\cdot x\cdot y+ z\cdot y-7\cdot y-8\cdot x$; (0;0;1)
9	$f(x,y,z) =2 x^3 +4 y^3-7\cdot z-3\cdot y + 5$; (1;2;2);
10	$f(x,y,z) =2 x^3 +4 y^3-7\cdot z-3\cdot y + 5$; (0;2;0);
11	$f(x,y,z) =2\cdot x^2+4\cdot y^2- 4\cdot z^2+x-2\cdot z-y^2 + 1$; (0;0;0);
12	$f(x,y,z) =x^2+y^2+ z^2+x-8\cdot x-y + 7$; (0;0;0);

4.5 **Приклад алгоритму розрахунку**

Програма розрахунку виконується у середовищі MatLAB за допомогою створення m-файлу.

Початкові данні: $f(x,y)=\frac{x-y}{x^2+y^2+2}$; (1;2), (2;0) и (2;2).

```
function z=F(V1)
z=0;
x=V1(1);
```

```
y=V1(2);
z=(x-y)./(x.^2+y.^2+2);
```

Функція для трьох змінних визначається наступним чином:

```
function e=F(V1)
z=V1(1);
x=V1(1);
y=V1(1);
e=[2*x.^2+2*y.^2+z.^2-2*x.*y+y.*z-7*y-4*z];
```

```
function [V0,y0,dV,dy]=nedler(V,mini,maxi,epsilon,show)
   if nargin==5
      show=0;
   end
   [mm,n]=size(V);
   for j=1:n+1
      Z=V(j,1:n);
      Y(j)=feval(@F,Z);
   end
   [mm,lo]=min(Y);
   [mm,hi]=max(Y);
   li=hi;
   ho=lo;
   for j=1:n+1
      if(j~=lo) && (j~=hi) && (Y(j)<=Y(li))
         li=j;
      end
      if(j~=hi) && (j~=lo) && (Y(j)>=Y(ho))
         ho=j;
      end
      cnt=0;
      % begin
      while((Y(hi)>Y(lo)+epsilon) && (cnt<maxi)||(cnt<mini))
         S=zeros(1,1:n);
         for j=1:n+1
            S=S+V(j,1:n);
         end
         M=(S-V(hi,1:n))/n;
         R=2*M-V(hi,1:n);
         yR=feval(@F,R);
```

```
if(yR<Y(ho))
   if(Y(li)<yR)
      V(hi,1:n)=R;
      Y(hi)=yR;
   else
      E=2*R-M;
      yE=feval(@F,E);
      if(yE<Y(li))
         V(hi,1:n)=E;
         Y(hi)=yE;
      else
         V(hi,1:n)=R;
         Y(hi)=yR;
      end
   end
else
   if (yR<Y(hi))
      V(hi,1:n)
      Y(hi)=yR;
   end
   C=(V(hi,1:n)+M)/2;
   yC=feval(@F,C);
   C2=(M+R)/2;
   yC2=feval(@F,C2);
   if(yC<Y(hi))
      V(hi,1:n)=C;
      Y(hi)=yC;
   else
      for j=1:n+1
         if (j~=lo)
            V(j,1:n)=(V(j,1:n)+V(lo,1:n))/2;
            Z=V(j,1:n);
            Y(j)=feval(@F,Z);
         end
      end
   end
end
[mm,lo]=min(Y);
[mm,hi]=max(Y);
li=hi;
ho=lo;
```

```
    for j=1:n+1
        if(j~=lo)&&(j~=hi)&&(Y(j)<=Y(li))
            li=j;
        end
        if(j~=hi)&& (j~=lo)&&(Y(j)>=Y(ho))
            ho=j;
        end
    end
    cnt=cnt+1;
    P(cnt,:)=V(lo,:);
    Q(cnt)=Y(lo);
 end
snorm=0;
for j=1:n+1
    s=norm(V(j)-V(lo));
    if(s>=snorm)
        snorm=s;
    end
end
Q=Q';
V0=V(lo,1:n);
y0=Y(lo);
dV=snorm;
dy=abs(Y(hi)-Y(lo));
if(show == 1)
    disp(P);
    disp(Q);
end
```

4.6 **Контрольні запитання**

1. Надайте визначення локального мінімуму та локального максимум функції *f(x;y)*;

2 Критерій другої похідної функції *f(x;y)*;

3. Особливості методу Нелдера-Мида**.**

ЛАБОРАТОРНА РОБОТА 5

ЗНАХОДЖЕННЯ ОПТИМАЛЬНОГО ЗНАЧЕННЯ ФУНКЦІЇ МЕТОДОМ НАЙШВИДШОГО СПУСКУ

Мета роботи: Вивчення та надбання навичок щодо знаходження оптимального значення функції методом найшвидшого спуску.

5.1. **Короткі теоретичні відомості**

Звернемося до мінімізації функції $f(\boldsymbol{X})$ від $\boldsymbol{N}$ змінних, де $\boldsymbol{X} = (x_1, x_2, \ldots, x_N)$. Градієнт $f(\boldsymbol{X})$ — це вектор (векторна функція), визначений як:

$$grad(f_1, f_2, \ldots, f_N), \tag{5.1}$$

де частинна похідні $f_k = \partial f \, / \, \partial x_k$ обчислюються в точці $\boldsymbol{X}$.

Градієнт (5.1) указує напрямок найбільшої швидкості зростання функції $f(\boldsymbol{X})$. Отже, - $grad\ f(\boldsymbol{X})$ указує напрямок найбільшого убування.

Пошук починається із точки $\boldsymbol{P_0}$ уздовж лінії, що проходить через $\boldsymbol{P_0}$ у напрямку

$$\boldsymbol{S_0} = -\,\boldsymbol{G} / \, ||G||,$$

де $\boldsymbol{G} = grad\ /(\boldsymbol{P_0})$.

При знаходженні точки $\boldsymbol{P_1}$, де перебуває локальний мінімум, точка $\boldsymbol{X}$ змушена буде потрапити на лінію $\boldsymbol{X} = \boldsymbol{P_0} + t s_0$. Потім можна обчислити $\boldsymbol{G} = grad\ f(\boldsymbol{P_1})$ і рухатися в напрямку $\boldsymbol{S_1} = -\,\boldsymbol{G} / \, ||G||$. При знаходженні точки $\boldsymbol{P_2}$, крапка $\boldsymbol{X}$ змушена буде потрапити на лінію $\boldsymbol{X}$ =

$P_1 + ts_1$. Ітерація породжує послідовність точок $\{P_k\}$, що володіють властивістю

$$f(P_0) > f(P_1) > \ldots > f(P_k) > \ldots . \qquad (5.2)$$

Якщо $\lim_{k \to \infty} P_k = P$, то $f(P)$ буде локальним мінімумом для $f(X)$.

Схема методу градієнта

Припустимо, що послідовність точок P_k отримана.

Крок 1. Обчислення градієнта $G = grad\ f(P_k)$.

Крок 2. Визначення напрямку пошуку $S = -G/\|G\|$.

Крок 3. Визначається єдиний параметр мінімізації $\Phi(t) = f(P_k + ts)$ на інтервалі $[0, b]$, де $t = h_{min}$. Для $\Phi(t)$ перебуває локальний мінімум. Співвідношення $\Phi(h_{min}) = f(P_k + h_{min} s)$ визначає мінімум для $f(X)$ уздовж обраної лінії $X = P_k + h_{min} s$.

Крок 4. Побудова наступних крапок $P_{k+1} = P_k + h_{min} s$.

Крок 5. Визначається критерій достатньої мінімізації, тобто чи досить близькі значення функції $f(P_k)$ і $f(P_{k+1})$ і чи досить мала відстань $\|P_{k+1} - P_k\|$.

5.2 **Хід роботи**

5.2.1. Визначити максимальне значення цільової функції резонансного контуру за своїм варіантом, використовуючи метод найшвидшого спуску. Завдання виконується для двох та трьох змінних за табл. 4.1., та табл.4.2.

5.2.2. Оформити звіт з лабораторної роботи.

5.3 **Приклад алгоритму розрахунку**

Програма розрахунку виконується у середовищі MatLAB за допомогою створення m-файлу.

Початкові данні: f(x,y)= $x^3 \frac{x-y}{x^2+y^2+2}$;

Часткові похідні: dz/dx= $\frac{-x^2+2*x*y+y^2+2}{(x^2+y^2+2)^2}$ dz/dy= $\frac{-x^2-2*x*y+y^2-2}{(x^2+y^2+2)^2}$,

Po= (1; 2).

Функція для двох змінних визначається наступним чином

```
function z=G(V)
z=zeros(1,2);
x=V(1); y=V(1);
g=[(-x.^2+2*x*y+y.^2+2)/(x.^2+y.^2+2).^2  (-x.^2-2*x*y+y.^2-2)/
(x.^2+y.^2+2).^2];
z=-(1/norm(g))*g;
```

Алгоритм програми:

```
function [P0,y0,err]=grads(P0,maxi,delta,epsilon,show)
   if nargin == 5
      show=0;
   end
   [mm n]=size(P0);
   maxj=10;
   big=1e8;
   h=1;
   P=zeros(maxj,n+1);
   len=norm(P0);
   y0=feval(@F,P0);
   if len>1e4
      h=len;
   end
   err=1;
   cnt=0;
   cond=0;
   P(cnt+1,:)=[P0 y0];
   while(cnt<maxi) && (cond~=5) && ((h>delta) || (err>epsilon))
      % direct search begining
      S=feval(@G,P0);
      P1=P0+h*S;
      P2=P0+2*h*S;
      y1=feval(@F,P1);
      y2=feval(@F,P2);
```

```
cond=0;
j=0;
while (j<maxi) && (cond == 0)
   len=norm(P0);
   if(y0<y1)
      P2=P1;
      y2=y1;
      h=h/2;
      P1=P0+h*S;
      y1=feval(@F,P1);
   else
      if (y2<y1)
         P1=P2;
         y1=y2;
         h=2*h;
         P2=P0+2*h*S;
         y2=feval(@F,P2);
      else
         cond=-1;
      end
   end
   j=j+1;
   if(h<delta)
      cond=1;
   end
   if(abs(h)>big) || (len>big)
      cond=5;
   end
end
if (cond==5)
   Pmin=P1;
   ymin=y1;
else
   d=4*y1-2*y0-2*y2;
   if(d<0)
      hmin=h*(4*y1-3*y0-y2)/d;
   else
      cond=4;
      hmin=h/3;
   end
   Pmin=P0+hmin*S;
```

```
    ymin=feval(@F,Pmin);
    h0=abs(hmin);
    h1=abs(hmin-h);
    h2=abs(hmin-2*h);
    if(h0<h)
       h=h0;
    end
    if(h1<h)
       h=h1;
    end
    if(h2<h)
       h=h2;
    end
    if(h==0)
       h=hmin;
    end
    if(h<delta)
       cond=1;
    end
    e0=abs(y0-ymin);
    e1=abs(y1-ymin);
    e2=abs(y2-ymin);
    if (e0~=0) && (e0<err)
       err=e0;
    end
    if (e0 && e1 < err)
       err=e1;
    end
    if (e2~=0) && (e2<err)
       err=e2;
    end
    if (e0==0) && (e1==0) && (e2==0)
       err=0;
    end
    if (err<epsilon)
       cond=2;
    end
    if(cond == 2) && (h < delta)
       cond=3;
    end
end
```

```
        cnt=cnt+1;
        P(cnt+1,:)=[Pmin ymin];
        P0=Pmin;
        Y0=ymin;
    end
    if show == 1
        disp(P);
    end
end
```

5.4 **Контрольні запитання**

1. Надайте визначення локального мінімуму та локального максимум функції *f(x;y)*;

2 Критерій другої похідної функції *f(x;y)*;

3. Особливості методу найшвидшого спуску**.**

ЛАБОРАТОРНА РОБОТА 6

ВИВЧЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ ЗА ДОПОМОГОЮ ЗАСОБІВ OPTIMIZATION TOOLBOX MATLAB

Мета роботи: вивчити та освоїти можливості пакету optimization toolbox програмного засобу MatLAB.

6.1 **Короткі теоретичні відомості**

В Matlab існує багато методів мінімізації як одномірних *(fminbnd*), так і багатомірних *(fminsearch, lsqnonlin, fminmax, fminuns, fmincon)* функцій, які реалізують різні чисельні методи.

У функції *fminsearch* використовується метод Нелдера-Мида. Перевагою цієї функції є можливість її використання для негладких і розривних цільових функцій.

Форма звертання до цієї функції має вигляд:

$$\mathbf{x = fminsearch(fun,\ x_0,\ options,\ p_1,\ p_2,...).} \tag{6.1}$$

У тому випадку, коли функція є досить гладкої, для пошуку її мінімуму можна скористатися процедурою *fminunc*. Дана функція реалізує метод Ньютона.

Функція *lsqnonlin* застосовується в тих випадках, коли цільова функція має вигляд:

$$\boldsymbol{F} = 1/2 \cdot \sum f_i^2. \tag{6.2}$$

У цьому випадку градієнт ***g*** і гессіан ***H*** функції ***F*** виражаються через якобіан ***J вектор***-функції $f|f_1, f_2, \ldots, f_m|'$:

$$\boldsymbol{g} = \boldsymbol{J}' \times \boldsymbol{f}; \tag{6.3}$$

$$\mathbf{H} = \boldsymbol{J}' \times \boldsymbol{J} + \boldsymbol{R}. \tag{6.4}$$

Залишковий член $\boldsymbol{R}$ включає другі похідні від $\boldsymbol{F}$, але в околиці мінімуму їм звичайно зневажають у порівнянні з $\boldsymbol{J'} \times \boldsymbol{J}$. Це дає можливість не обчислювати другі похідні, що значно прискорює роботу з порівнянням із загальним випадком.

Система рівнянь для відшукання вектора зрушення $\boldsymbol{H} \times \boldsymbol{p} = -\boldsymbol{g}$ при заміні $\boldsymbol{H}$ на $\boldsymbol{J'} \times \boldsymbol{J}$ перетворюється в $\boldsymbol{J'} \times \boldsymbol{J} \times \boldsymbol{p} = -\boldsymbol{J'} \times \boldsymbol{f}$. Цей спосіб називається методом Гаусса-Ньютона.

У методі Левенберга-Марквардта матриця $\boldsymbol{J'} \times \boldsymbol{J}$ у лівій частині системи рівнянь заміняється на $\boldsymbol{J'} \times \boldsymbol{J} + \boldsymbol{\lambda} \times \boldsymbol{I}$, де $\boldsymbol{I}$ - одинична матриця, *а* $\boldsymbol{\lambda}$ – деяке від'ємне число. Для вектора зсуву задається обмеження $|p|<\Delta$, де Δ і λ взаємозалежні.

В функції *Isqnonlin* застосовуються обидва метода: метод Гаусса-Ньютона й Левенберга-Марквардта.

У зверненні до **x=lsqnonlin(fun,$\mathbf{x_0}$)** мінімізується сума квадратів компонент вектора - стовпця, який виробляє функція fun (це може бути також матриця). Другий аргумент x_0 - стартова точка для пошуку.

При використанні функції *Isqnonlin,* у якій цільова функція є згорток вектора, можна задавати обмеження. Для функції *Isqnonlin* доступні тільки найпростіші обмеження типу $lb \le x \le ub$. Звертання до функції *Isqnonlin* у цьому випадку має вигляд:

$$\mathbf{x=lsqnonlin(fun,\ x_0,\ lb,\ ub,\ options,\ p_1,p_2,...)} \qquad (6.5)$$

Функція *Isqnonlin* може повернути досить багато вихідних параметрів:

[x, resnorm, residual, e_flag, inform, lambda, jacobian]=lsqnonlin(...)

Вихідних аргументів:

x – вектор-розв'язок;

resnorm – значення цільової функції в знайденій точці;

residual - компоненти функції fun(x) у знайденій точці;

e_flag - e_flag=l - розв'язок системи знайдений, e_flag=0 - розв'язок системи не знайдений, e_flag=-l - досягнутий мінімум не є розв'язком системи;

inform – містить три поля:

iterations - кількість ітерацій, виконаних при пошуку кореня;

funccount - кількість звертань до функції fun;

algorithm - найменування алгоритму, використаного для знаходження кореня;

lambda - вектор множників Лагранжа;

jacobian – якобіан функції fun у знайденій точці.

До функції *Isqnonlin* ідейно близька функція *fminimax,* в обох скалярна цільова функція не задається безпосередньо у зверненні, а формується шляхом згортки з компонентів вектора (або матриці), переданого функції. В *Isqnonlin* мінімізується сума квадратів компонент, а в *fminimax* - максимальний компонент. Алгоритм, реалізований в *fminimax,* багаторазово використовує квадратичне програмування, а також пошук за допомогою градієнта й гесіана.

Усі функції оптимізації включають у списку своїх вхідних параметрів перелік властивостей, що впливають на хід ітераційних процесів. Ці властивості представлені структурою options, поля якої формуються за допомогою функції *optimset*. Завдання значення будь-якої властивості проводиться парою параметрів функції *optimset,* перший з яких представляє найменування властивості, а другий - його значення:

options=optimset('name1',vall,'name2',val2,...); (6.6)

x=fxxx(fun,$x_{0,...,}$options,p_1,p_2,...) (6.7)

Перед вхідним параметром options може розташовуватися деяка кількість вхідних параметрів, перелік яких для кожної функції оптимізації індивідуальний.

Параметри p_1, p_2,..., розташовувані слідом за структурою options, передаються оптимізованій функції разом з незалежним аргументом x – fun(x,pl,p2,...). У табл. 6.1 наведений список параметрів, керуючих процесом знаходження мінімуму функцій.

6.2 **Хід роботи**

6.2.1. Вивчити особливості користування пакету optimization toolbox програмного засобу MatLAB.

Таблиця 6.1. Перелік параметрів, які керують процесом пошуку мінімуму функції

MaxFunEvals	максимальна кількість звернень до функції fun
Maxlter	максимальна кількість ітерацій
TolFun	припинення ітерацій при досягненні точності по значенню функції
TolX	припинення ітерацій при досягненні мінімального кроку по X

6.2.2 Відповідно свого індивідуального завдання створити цільові функції для застосування методів оптимізації.

6.2.3 Визначити оптимальні значення цільової функції методом Нелдера-Мида застосовуючи функцію *fminsearch*.

6.2.4. Визначити оптимальні значення цільової функції метод Гаусса-Ньютона и Левенберга-Марквардта, застосовуючи функцію lsqnonlin.

6.2.5. Провести порівняльний аналіз двох методів оптимізації при заданої точності 1e-3, 1e-5 та 1e-10 за критеріями: кількість ітерацій та час розрахунку.

6.3 Контрольні запитання

1. Чим, та в яких умовах застосовуються методи оптимізації Нелдера-Мида та Гаусса-Ньютона и Левенберга-Марквардта?
2. У якій формі має вигляд цільова функція при застосуванні метода Нелдера-Мида?
3. У якій формі має вигляд цільова функція при застосуванні метода Гаусса-Ньютона та Левенберга-Марквардта.
4. Який метод має найменшу кількість ітерації та час розрахунку?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна література

1. Жалдак, М.І. Основи теорії і методів оптимізації: навч. посібник [Текст] / М. І. Жалдак, Ю.В. Триус – Черкаси: Брама-Україна, 2005. – 608с. ISBN 966-8756-04-5

2. Ладієва, Л.Р. Методи оптимізації та пошуку оптимальних рішень: навч. посібник [Текст] / Л.Р. Ладієва– Київ: Національний технічний університет "Київський політехнічний інститут імені Ігоря Сікорського, 2023. – 73с.

3. Глушик, М.М.. Математичне програмування: навч. посібник / М. М. Глушик, І. М. Копич, В. М. - Львів: Новий Світ, 2014. – 280 с. ISBN 978-966-418-103-4

4. Зайченко Ю.П. Дослідження операцій: навч. посібник / Ю. П. Зайченко.-Київ: Видавничий дім «Слово», 2006.- -816с.

5. Ладієва Л.Р. Оптимізація технологічних процесів: навчал. посібник / Л. Р. Ладієва– Київ: ІВЦ видавництво «Політехніка»», 2004. – 192с.

6. Sage, E.P. Optimal systems control / E.P. Sage, C.S. White– Prentice-Hall, Englewood Cliffs, N.J, 2006. - 392p.

7. Ravindran A. Engineering Optimization Methods and Application / A. Ravindran, Ragsdell K.M., Reklaitis G.V. – Publication John Willy and sons, Inc, NJ, 2006.- 688p.

Додаткова література

8. Yarymbash, D. A New Simulation Approach of the Electromagnetic Fields in Electrical Machines [Text] / D. Yarymbash, M. Kotsur, S. Subbotin, A. Oliinyk //. IEEE: The International Conference on Information and Digital Technologies, July 5th - 7th, 2017: Catalog Number CFP17CDT-USB. - Slovakia, 2017. - p. 452-457.

9. Kotsur, M.I. Converter for frequency-current slip-power recovery scheme [Text] / M.I. Kotsur, P.D. Andrienko, I. M. Kotsur, O.V Bliznyakov // Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu. – 2017. - №4. – P. 49-54.

10. Kotsur, M. A New Approach of the Induction Motor Parameters Determination in Short-Circuit Mode by 3D Electromagnetic Field Simulation [Text] / M. Kotsur, D. Yarymbash, S. Yarymbash, I. Kotsur // International Young Scientists Forum on Applied Physics and Engineering

(YSF), October 17th - 20th, Lviv, Ukraine, 2017. - P. 207-210.

11. Kotsur, M. I. Increasing of Thermal Reliability of a Regulated Induction Motor in Non-Standard Cycle Time Conditions [Text] / M. I. Kotsur, I.M. Kotsur, Yu. Bezverkhnia, D. Andrienko // IEEE: International Conference on Modern Electrical and Energy Systems (MEES), November 15th - 17th 2017. - Kremenchuk, Ukraine, 2017.- P. 88-91.

12. Kotsur M. Comparative analysis of a different geometric shapes of a busbar’s trolley parameters in the higher harmonic current condition [Electronic Resource] / M. Kotsur, D. Yarymbash, Yu. Bezverkhnia, I. Kotsur // IEEE: 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), February 22-26, Lviv Polytechnic Week, Ukraine, 2022, pp. 87-92. DOI:10.1109/TCSET55632.2022