МАТЕМАТИЧНІ МЕТОДИ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЇ ПРОЦЕСІВ

Практичне заняття № 10

Тема. Чисельні методи оптимізації першого порядку.

План проведення заняття

Вступ.
1. Метод градієнтного спуску із постійним кроком.
2. Метод найшвидшого градієнтного спуску.
Заключення.

Завдання на СРС:
Підготуватись до наступного ПЗ.

Вступ

В попередній темі розглядались аналітичні методи пошуку екстремумів функцій багатьох змінних. Ті методи базувались на положенні, що полягає в такому: в точці екстремуму функція має нульові частинні похідні. Ці методи застосовуються для нескладних функцій, що задані аналітично. У випаду, коли функція задається неявно або алгоритмічно, то для знаходження екстремумів треба застосовувати чисельні методи оптимізації.

До методів оптимізації першого порядку відносяться алгоритми, в яких в процесі пошуку екстремуму окрім інформації про саму функцію використовується інформація про похідні першого порядку. До групи таких методів відносяться різні градієнтні методи.

Градієнт функції в будь-якій точці показує напрямок найбільшого локального збільшення функції f(x1, x2, …, xn). Тому при пошуку мінімуму слід рухатися в напрямку, який є протилежним напрямку градієнта в даній точці, тобто в напрямку найшвидшого спуску.

1. Метод градієнтного спуску із постійним кроком.

Постановка задачі

Нехай задана функція *f(x)*, обмежена знизу на множині R^n та має неперервні часткові похідні у всіх точках.

Потрібно знайти локальний мінімум функції *f(x)* на множині допустимих рішень $X = R^n$, тобто знайти таку точку $x^* \epsilon\, R^n$, що

$$f(x^*) = \min_{x \epsilon R^n} f(x).$$

Стратегія пошуку

Стратегія рішення задачі заключається в побудові послідовних точок $\{x^k\}, k = 0,1, ...$, таких, що $f(x^{k+1}) < f(x^k)$, *k=0,1,…*, Точки послідовності $\{x^k\}$ обчислюються за правилом

$$x^{k+1} = x^k - t_k \nabla f(x^k), k = 0,1, ...,$$

де точка x^0 задається користувачем; $\nabla f(x^k)$ – градієнт функції $f(x)$, обчислений в точці x^k; величина кроку t_k задається користувачем і залишається постійною до тих пір, поки функція спадає в точках послідовності, що контролюється шляхом перевірки виконання умови $f(x^{k+1}) - f(x^k) < 0$ або $f(x^{k+1}) - f(x^k) < -\varepsilon\left\|\nabla f(x^k)\right\|^2, 0 < \varepsilon < 1$

Побудова послідовності $\{x^k\}$ закінчується в точці x^k, для якої $\left\|\nabla f(x^k)\right\| < \varepsilon_1$, де ε_1- задане мале позитивне число, або $k \geq M$, де *M* - граничне число ітерацій, або при дворазовому одночасному виконанні двох нерівностей $\left\|x^{k+1} - x^k\right\| < \varepsilon_2$, $\left|f(x^{k+1}) - f(x^k)\right| < \varepsilon_2$, де ε_2 - мале позитивне число.

Питання про те, чи може точка x^k розглядатися як знайдене наближення шуканої точки мінімума, вирішується шляхом проведення додаткового дослідження, яке описано нижче.

Алгоритм

Крок 1. Задати $x^0, 0 < \varepsilon < 1, \varepsilon_1 > 0, \varepsilon_2 > 0, M$ - граничне число ітерацій. Знайти градієнт функції в довільній точці $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n}\right)^T$.

Крок 2. Покласти *k = 0*.

Крок 3. Обчислити $\nabla f(x^k)$.

Крок 4. Перевірити виконання критерію закінчення $\left\|\nabla f(x^k)\right\| < \varepsilon_1$:

а) якщо критерій виконаний, розрахунок закінчений, $x^* = x^k$;

б) якщо критерій не виконано, то перейти до кроку 5.

Крок 5. Перевірити виконання нерівності $k \geq M$:

а) якщо нерівність виконано, то розрахунок закінчено: $x^* = x^k$;

б) якщо ні, то перейти до кроку 6.

Крок 6. Задати величину кроку t_k.

Крок 7. Обчислити $x^{k+1} = x^k - t_k \nabla f(x^k)$.

Крок 8. Перевірити виконання умови

$f(x^{k+1}) - f(x^k) < 0$ (або $f(x^{k+1}) - f(x^k) < -\varepsilon\left\|\nabla f(x^k)\right\|^2$):

а) якщо умова виконана, то перейти до кроку 9;

б) якщо умова не виконана, покласти $t_k = \frac{t_k}{2}$ і перейти до кроку 7.

Крок 9. Перевірити виконання умов

$\left\|x^{k+1} - x^k\right\| < \varepsilon_2$, $\left|f(x^{k+1}) - f(x^k)\right| < \varepsilon_2$:

а) якщо обидві умови виконані при поточному значенні k і $k = k - 1$, то розрахунок закінчено, $x^* = x^{k+1}$;

б) якщо хоча б одна з умов не виконана, покласти $k = k + 1$ і перейти до кроку 3.

Геометрична інтерпретація методу для n = 2 наведена на рис. 6.1.

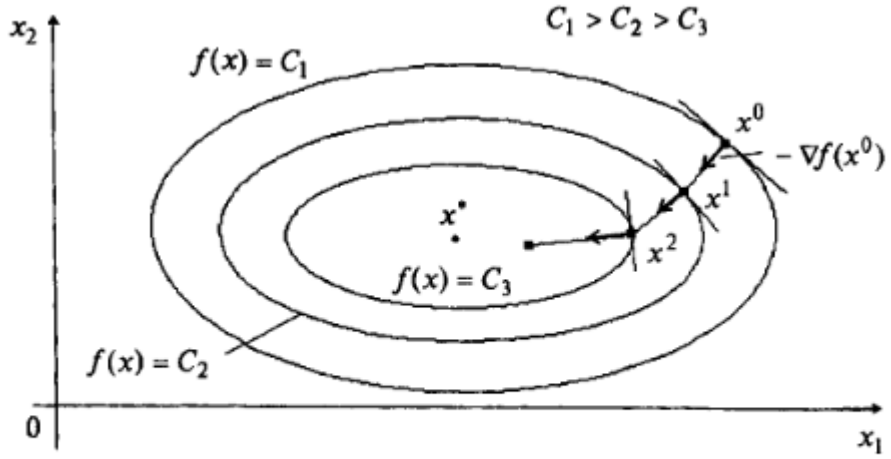


Рис. 6.1

2. Метод найшвидшого градієнтного спуску

Постановки задачі

Нехай дана функція $f(x)$, обмежена знизу на множині R^2 і має безперервні часткові похідні у всіх її точках.

Потрібно знайти локальний мінімум функції $f(x)$ на множині допустимих рішень $X = R^2$, тобто знайти таку точку $x^* \epsilon R^n$, що

$$f(x^*) = \min_{x \epsilon R^n} f(x).$$

Стратегія пошуку

Стратегія рішення задачі заключається в побудові послідовних точок $\{x^k\}, k = 0,1, ...$, таких, що $f(x^{k+1}) < f(x^k)$, $k=0,1,\ldots$, Точки послідовності $\{x^k\}$ обчислюються за правилом

$$x^{k+1} = x^k - t_k \nabla f(x^k),$$

де точка x^0 задається користувачем; величина кроку t_k визначається для кожного значення k з умови

$$\varphi(t_k) = f\left(x^k - t_k \nabla f(x^k)\right) \to \min_{t_k}.$$

Рішення завдання може здійснюватися з використанням необхідної умови мінімуму $\frac{\partial^2 \varphi}{\partial {t_k}^2} = 0$ з подальшою перевіркою достатньої умови мінімуму $\frac{\partial^2 \varphi}{\partial t_k^2} > 0$. Такий шлях може бути використаний або при досить простій функції, що мінімізується $\varphi(t_k)$, або при попередній апроксимації досить складної функції $\varphi(t_k) = f\left(x^k - t_k \nabla f(x^k)\right)$ полиномом $P(t_k)$ (Як правило, другого або третього ступеня), і тоді умова $\frac{\partial \varphi}{\partial t_k} = 0$ замінюється умовою $\frac{\partial P}{\partial t_k}$=0, а умова $\frac{\partial^2 \varphi}{\partial t_k^2} > 0$ - умовою $\frac{\partial^2 P}{\partial t_k^2} > 0$.

Інший шлях вирішення задачі пов'язаний з використанням чисельних методів, коли шукається $\min\limits_{t_k\in[a,b]} \varphi(t_k) = \min\limits_{t_k\in[a,b]} f\left(x^k - t_k\nabla f(x^k)\right)$
Границі інтервалу [a,b] задаються користувачем. При цьому ступінь близькості знайденого значення t_k до оптимального значення ${t_k}^*$. задовольняючому умовам $\frac{\partial\varphi}{\partial t_k} = 0, \frac{\partial^2\varphi}{\partial t_k^2} > 0$ залежить від задання інтервалу [a,b] і точності методів одномірної мінімізації.

Побудова послідовності $\{x^k\}, k = 0,1, ...,$ закінчується в точці x^k, для якої $\left\|\nabla f(x^k)\right\| < \varepsilon_1$. де ε_1 - задане число, або, якщо , $k \geq M, M$ -граничне число ітерацій, або при дворазовому одночасному виконанні нерівностей $\left\|x^{k+1} - x^k\right\| < \varepsilon_2$, $\left|f(x^{k+1}) - f(x^k)\right| < \varepsilon_2$, де ε_2- мале позитивне число. Питання про те, чи може точка x^k розглядатися як знайдене наближення шуканої точки локального мінімуму x^*, вирішується шляхом додаткового дослідження.

Алгоритм

Крок 1. Задати $x^0, \varepsilon_1 > 0, \varepsilon_2 > 0, M$ - граничне число ітерацій. Знайти градієнт функції в довільній точці $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n}\right)^T$.

Крок 2. Покласти *k = 0.*

Крок 3. Обчислити $\nabla f(x^k)$.

Крок 4. Перевірити виконання критерію закінчення $\left\|\nabla f(x^k)\right\| < \varepsilon_1$:

а) якщо критерій виконаний, розрахунок закінчений, $x^* = x^k$;

б) якщо критерій не виконано, то перейти до кроку 5.

Крок 5. Перевірити виконання нерівності $k \geq M$:

а) якщо нерівність виконано, то розрахунок закінчено: $x^* = x^k$;

б) якщо ні, то перейти до кроку 6.

Крок 6. Обчислити величину кроку t_k^* з умови

$$\varphi(t_k) = f\left(x^k - t_k\nabla f(x^k)\right) \to \min_{t_k}.$$

Крок 7. Обчислити $x^{k+1} = x^k - t_k^*\nabla f(x^k)$.

Крок 8. Перевірити виконання умов

$$\left\|x^{k+1} - x^k\right\| < \varepsilon_2, \left|f(x^{k+1}) - f(x^k)\right| < \varepsilon_2$$

а) якщо обидві умови виконані при поточному значенні k і $k = k - 1$, то розрахунок закінчено, $x^* = x^{k+1}$;

б) якщо хоча б одна з умов не виконана, покласти $k = k + 1$ і перейти до кроку 3.

Геометрична інтерпретація методу для n = 2 наведена на рис. 6.1

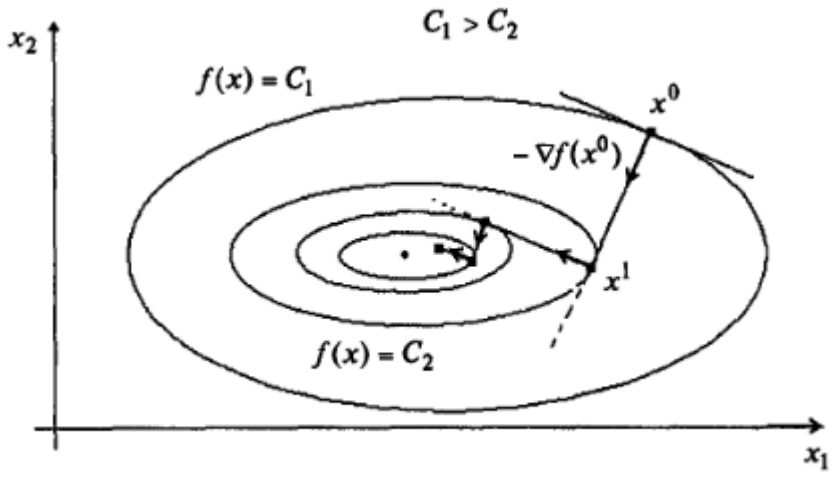


Рис. 6.3

Завдання 1: Знайти мінімум функції
$f(x,y) = 8x^2 - 4xy + 5y^2 + 8\sqrt{5}(x + 2y)$,
при ε=0,001.
а) Методом градієнтного спуску із постійним кроком.
б) Методом найшвидшого градієнтного спуску.
Побудувати графік, результати перевірити аналітично.

Розв'язання:

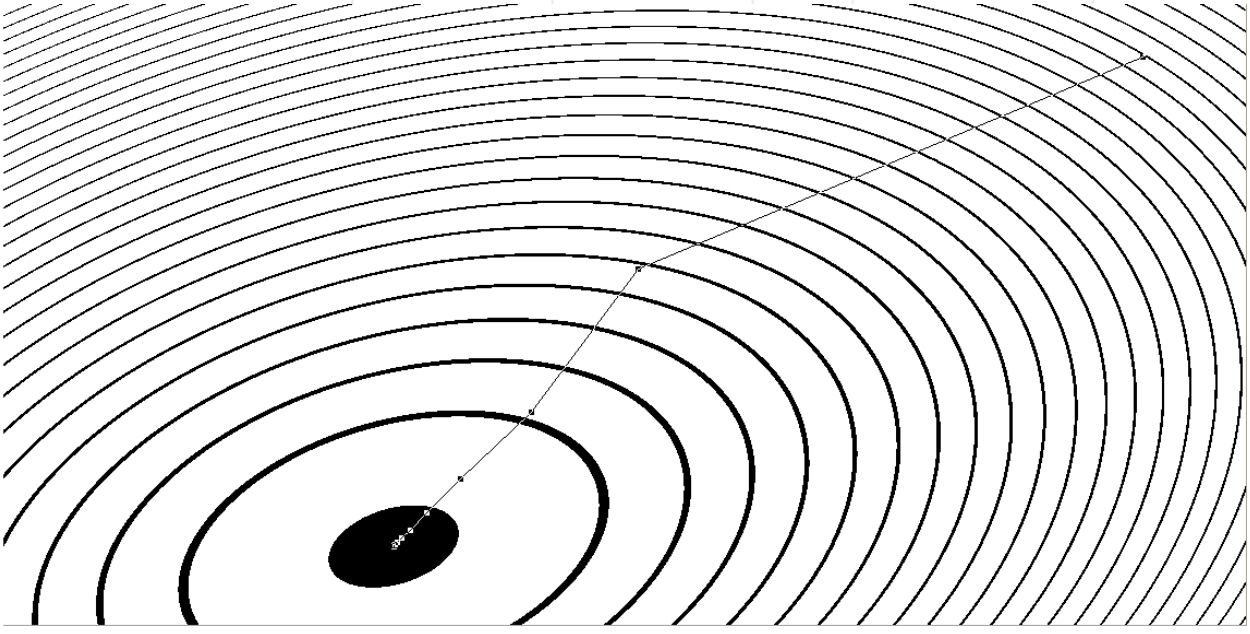
а) Приклад реалізації даного алгоритму мовою Delphi:

```
procedure
TfrmMain.FastFallWithCrackingStep(eps:double;fp:TWorldPoint);
var k:integer;
    gamma,cappa:double;
    lastx,x,grad:TWorldPoint;
    Screen2:TScreenPoint;
const cappa0=1;
begin
k:=1;gamma:=0.5
x:=fp;//fp - початкова точка
lastx:=fp;
GradientFunc(fp.x,fp.y,grad);// антиградієнт функции
while sqrt(sqr(grad.x)+sqr(grad.y))>eps do
begin
     cappa:=cappa0;
     GradientFunc(lastx.x,lastx.y,grad);
     x.x:=lastx.x+cappa*grad.x;
     x.y:=lastx.y+cappa*grad.y;
     while                                    Func(lastx.x,lastx.y)-
Func(x.x,x.y)<0.5*cappa*(sqr(grad.x)+sqr(grad.y)) do
     begin
        cappa:=cappa*gamma;
        x.x:=lastx.x+cappa*grad.x;
        x.y:=lastx.y+cappa*grad.y;
```

```
      end;
      BuiltReport(x,grad,k,cappa);//Вивід звіту про ітерації
      //побудова траєкторії спуску
      World2Screen(Area, copyscr.Canvas.ClipRect,x,screen2);
      SetPoint(x);
      lastx:=x;
      copyscr.Canvas.LineTo(screen2.x,screen2.y);
      inc(k);//лічильник ітерацій
end;{while}
end;
```

Результати роботи програми

N итерации	(x,y)	f(x,y)	cappa	Норма градиента
1.000	(0.132,0.889)	101.785	0.063	(-77.889,-65.777)
2.000	(-0.896,-1.870)	-1.719	0.063	(-16.444,-44.139)
3.000	(-1.585,-3.161)	-27.432	0.063	(-11.035,-20.663)
4.000	(-1.908,-3.818)	-33.858	0.063	(-5.166,-10.507)
5.000	(-2.073,-4.145)	-35.465	0.063	(-2.627,-5.232)
6.000	(-2.154,-4.309)	-35.866	0.063	(-1.308,-2.619)
7.000	(-2.195,-4.390)	-35.967	0.063	(-0.655,-1.309)
8.000	(-2.216,-4.431)	-35.992	0.063	(-0.327,-0.655)
9.000	(-2.226,-4.452)	-35.998	0.063	(-0.164,-0.327)
10.000	(-2.231,-4.462)	-35.999	0.063	(-0.082,-0.164)
11.000	(-2.234,-4.467)	-36.000	0.063	(-0.041,-0.082)
12.000	(-2.235,-4.470)	-36.000	0.063	(-0.020,-0.041)
13.000	(-2.236,-4.472)	-36.000	0.125	(-0.010,-0.020)
14.000	(-2.236,-4.472)	-36.000	0.000	(0.000,-0.000)



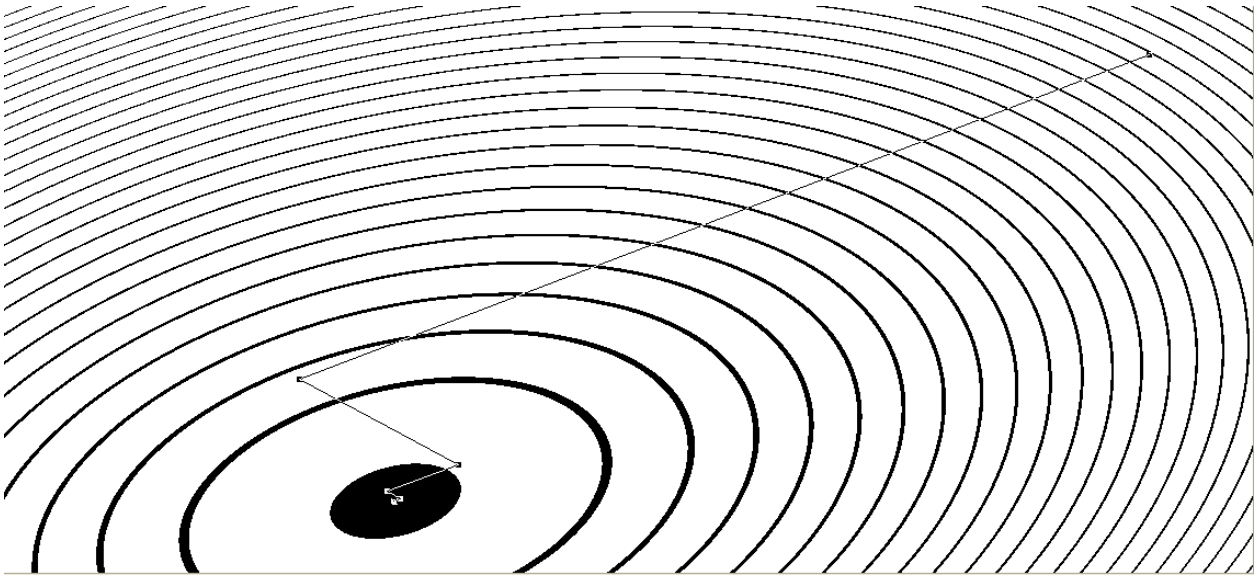
Відповідь: x=-2,236, y=-4,472, k=14

б) Приклад реалізації даного алгоритму мовою Delphi:

```
procedure TfrmMain.QuickestDescent(eps:double;fp:TWorldPoint);
var grad:TWorldPoint;
    curx,lastx:TWorldPoint;
    funcs:T1DFunction;
    screen1:TScreenPoint;
    cappa:double;
    k:integer;
begin
curx:=fp;
lastx:=fp;
lastx.x:=lastx.x+23;
GradientFunc(fp.x,fp.y,grad);
funcs:=@Pseudo1D;
k:=1;
while abs(Func(curx.x,curx.y)-Func(lastx.x,lastx.Y))>eps do
begin
     GradientFunc(curx.x,curx.y,grad);
     xk:=curx;
     uk:=grad;
        // спуск в заданому напрямку
     cappa:=MakeDichotomy(0,1,1e-5,eps/100,funcs);
     lastx:=curx;
     curx.x:=curx.x+cappa*grad.x;
     curx.y:=curx.y+cappa*grad.y;
     BuiltReport(curx,grad,k,cappa);//додавання рядків в таблицю
ітерацій
      // побудова траєкторії спуску
     World2Screen(Area, copyscr.Canvas.ClipRect,curx,screen1);
     SetPoint(curx);
     copyscr.Canvas.LineTo(screen1.x,screen1.y);
     inc(k);
end;
end;
```

Результати роботи програми

N итерации	(x,y)	f(x,y)	cappa	Норма градиента
1.000	(-3.148,-1.881)	13.674	0.105	(-77.889,-65.777)
2.000	(-1.630,-3.679)	-31.841	0.061	(24.961,-29.557)
3.000	(-2.312,-4.255)	-35.652	0.105	(-6.521,-5.507)
4.000	(-2.185,-4.406)	-35.971	0.061	(2.090,-2.475)
5.000	(-2.242,-4.454)	-35.998	0.105	(-0.546,-0.461)
6.000	(-2.232,-4.467)	-36.000	0.061	(0.175,-0.207)
7.000	(-2.237,-4.471)	-36.000	0.105	(-0.046,-0.039)
8.000	(-2.236,-4.472)	-36.000	0.061	(0.015,-0.017)
9.000	(-2.236,-4.472)	-36.000	0.105	(-0.004,-0.003)
10.000	(-2.236,-4.472)	-36.000	0.061	(0.001,-0.001)
11.000	(-2.236,-4.472)	-36.000	0.105	(-0.000,-0.000)



Відповідь: x=-2,236, y=-4,472, k=14.

Перевірити отримані результати самостійно, взявши часткові похідні.

Заключення

На практиці були розглянуті метод градієнтного спуску із постійним кроком та метод найшвидшого градієнтного спуску. Це є найрозповсюдженішими методами оптимізації першого порядку. Суть цих методів полягає в ітераційному русі в напрямку антиградієнта та поступовому наближені до мінімуму функції.

В даній практиці були використані дані програми, розробленої ***Бабаскіним Євгеном Михайловичем,*** які є у вільному доступі на сайті **nsft.narod.ru**

Завідувач кафедри вищої математики,
математичного моделювання та фізики
кандидат фізико-математичних наук, доцент І.В. Замрій