Ministerul Educaţiei, Culturii și Cercetării  
Universitatea Tehnică a Moldovei

Departamentul Informatică și Ingineria Sistemelor

**RAPORT**

Lucrarea de laborator nr.1  
 *la Metode și Modele de Calcul*

Varianta 2

A efectuat:   
st. gr. TI-206 Borș Nicoleta

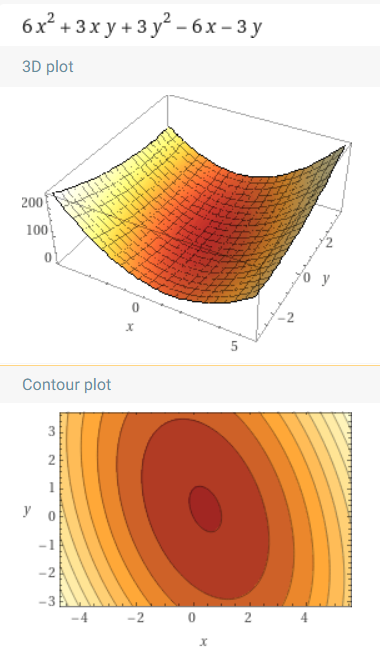
A verificat:   
asistent univ. Oleg Buldumac

Chişinău – 2021

**Tema**: Optimizarea unei funcții neliniare după metoda gradientului cu fracționarea pasului

**Sarcina:** . Să se găsească punctul minim al funcției pornind din =(a, b), ε = (a = 6, b = 3)

Cu ajutorul graficului observăm că avem o funcție neliniară convexă. Din orice punct o să pornim, o să avem același minim.



const A = 6, B = 3, GAMA = 0.5, EPSILON = Math.pow(10, -6), DELTA = 0.06;

const myFunction = (x, y) => A \* Math.pow(x, 2) + 3 \* x \* y + B \* Math.pow(y, 2) - A \* x - B \* y;

const gradientFunction = (x, y) => [2 \* A \* x + 3 \* y - A, 3 \* x + 2 \* B \* y - B];

const magnitudineFunction = (x, y) => Math.pow(x, 2) + Math.pow(y, 2);

*//--------------------Fractionarea pasului-------------------*

const proceduraDeFractionareAPasului = (x, y, lamda, gradX, gradY, xValue, magnitudinePatrat) => {

  let executeFract = true;

  while (executeFract) {

    let z = [x - lamda \* gradX, y - lamda \* gradY];

    let zValue = myFunction(z[0], z[1]);

    if (zValue - xValue <= -DELTA \* lamda \* magnitudinePatrat) {

      executeFract = false;

      return [z[0], z[1], lamda];

    } else {

      lamda = lamda \* GAMA;

    }

  }

};

const minimMetodaGradientului = () => {

  let x = A, y = B, k = 0, lamda = 1;

  let executeMinim = true;

  while (executeMinim) {

    let functionValue = myFunction(x, y);

    let gradientValue = gradientFunction(x, y);

    let magnitudinePatrat = magnitudineFunction(gradientValue[0], gradientValue[1]);

    console.log(`k = ${k}\nf(x) = ${functionValue}\ngradient = ${gradientValue}\nmagnitudine = ${Math.sqrt(magnitudinePatrat)}\nxk=(${x}, ${y}))`,);

    if (Math.sqrt(magnitudinePatrat) <= EPSILON || k === 1000) {

      console.log( `%c----------------------------------------------------------------\nPunctul minim este: (${x}, ${y})\n----------------------------------------------------------------`, 'color: yellow');

      executeMinim = false;

    } else {

      let valuesOfZ = proceduraDeFractionareAPasului(x, y, lamda, gradientValue[0], gradientValue[1], functionValue, magnitudinePatrat,);

      x = valuesOfZ[0];

      y = valuesOfZ[1];

      lamda = valuesOfZ[2];

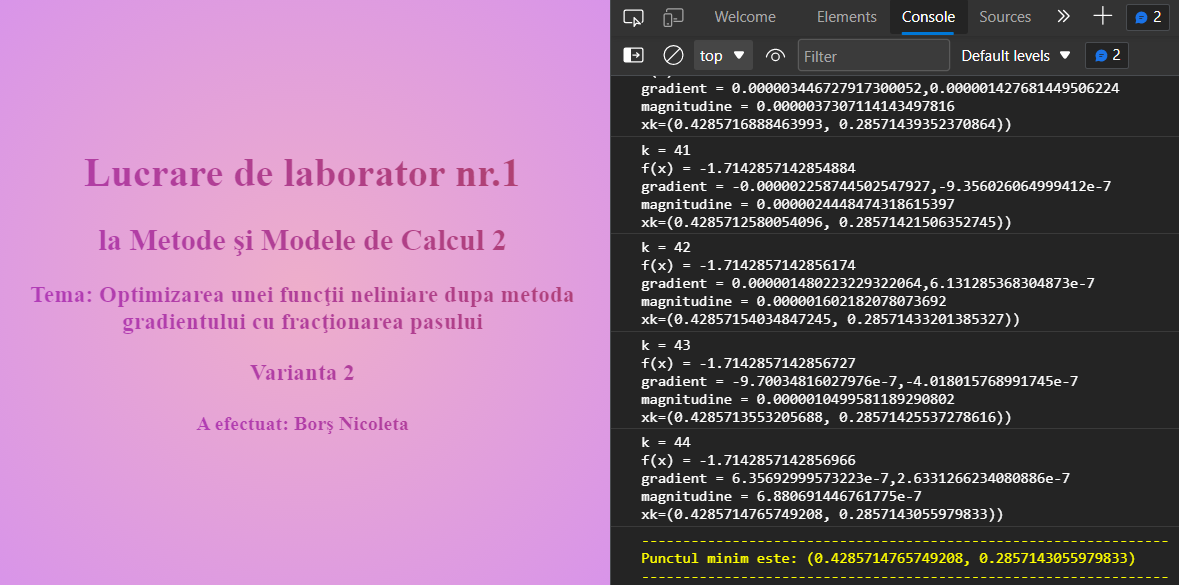
      k++;

    }

  }

};

minimMetodaGradientului();

****

#### **Concluzie:**

#### În cadrul lucrării de laborator la tema “Optimizarea unei funcții neliniare după metoda gradientului cu fracționarea pasului” am avut posibilitatea să elaborez un algortim de căutare a punctului minim. Mai mult decât atât am făcut cunoștință cu avantajele acestei metode, unul fiind faptul că nu trebuie să calculăm derivatele secundare de ordinul doi. Metoda gradient însă poate fi foarte lentă dacă pasul ales este mic. Alt dezavantaj e faptul că algoritmul se oprește la primul minim local.

#### **Bibliografia:**

#### WolframAlpha: [https://www.wolframalpha.com/input/?i=6x%5E2%2B3\*x\*y%2B3\*y%5E2-6\*x-3\*y](https://www.wolframalpha.com/input/?i=6x%5E2%2B3*x*y%2B3*y%5E2-6*x-3*y)

#### Github: <https://github.com/Nicoleta-Bors/MMC2.git>

#### Anexa:

#### 

#### 

#### 