**Bài tập buổi 8**

**(21/04/2025)**

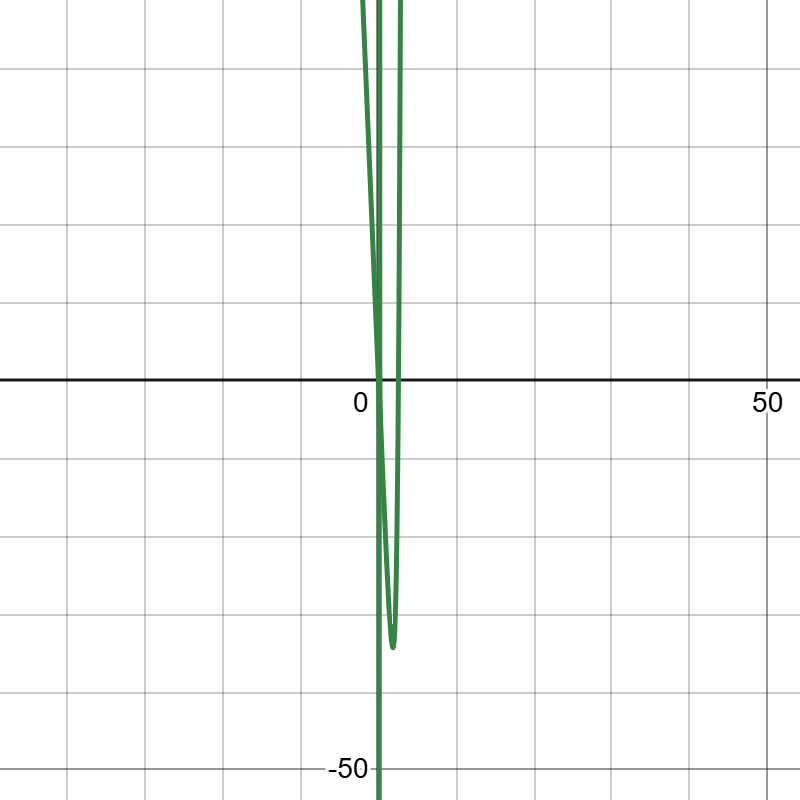
Trường: Đại học Bách khoa - Đại học Đà Nẵng.

Họ và tên sinh viên: Tôn Thất Văn.

Mã sinh viên: 102230055.

Lớp sinh hoạt: 23T\_Nhat1.

1. Đồ thị hàm số mục tiêu:

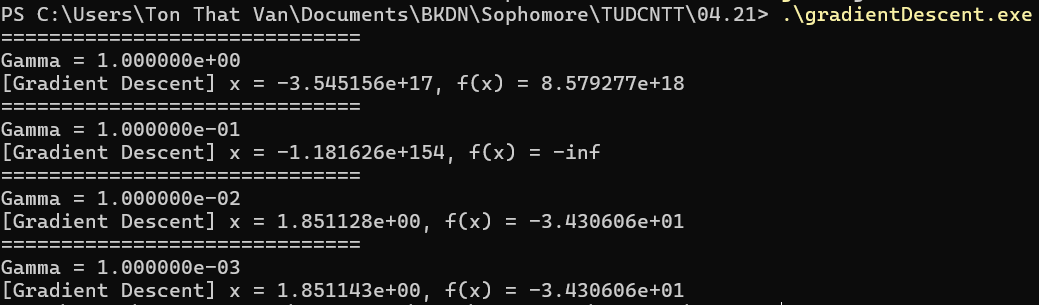


1. Thuật toán Gradient Descent

***Source***

|  |
| --- |
| *#include <iostream>*  *#include <cmath>*  *#include <iomanip>*  *using namespace std;*  *// Hàm mục tiêu*  *double func(double x) {*  *if (fabs(15 \* x - 2) < 1e-8) return INFINITY;*  *return (exp(3 \* x) + 12 \* x \* x + 5 \* x) / (15 \* x - 2) - 25 \* x;*  *}*  *// Đạo hàm*  *double grad(double x) {*  *double v = 15 \* x - 2;*  *if (fabs(v) < 1e-8) return INFINITY;*  *double u = exp(3 \* x) + 12 \* x \* x + 5 \* x;*  *double u\_prime = 3 \* exp(3 \* x) + 24 \* x + 5;*  *double v\_prime = 15;*  *double df = (u\_prime \* v - u \* v\_prime) / (v \* v) - 25;*  *return df;*  *}*  *// Gradient Descent cơ bản*  *double gradientDescent(double x0, double gamma) {*  *double x = x0;*  *int iter = 0, max\_iter = 100000;*  *while (iter++ < max\_iter) {*  *double grad\_x = grad(x);*  *if (!isfinite(grad\_x)) break;*  *double new\_x = x - gamma \* grad\_x;*  *if (fabs(new\_x - x) < 1e-6) break;*  *x = new\_x;*  *}*  *return x;*  *}*  *int main() {*  *double x\_init = 2.0;*  *cout << scientific << setprecision(6);*  *for (int i = 0; i < 4; i++) {*  *double gamma = pow(10, -i);*  *cout << "==============================" << endl;*  *cout << "Gamma = " << gamma << endl;*  *double x\_gd = gradientDescent(x\_init, gamma);*  *cout << "[Gradient Descent] x = " << x\_gd*  *<< ", f(x) = " << func(x\_gd) << endl;*  *}*  *return 0;*  *}* |

***Kết quả thực thi***

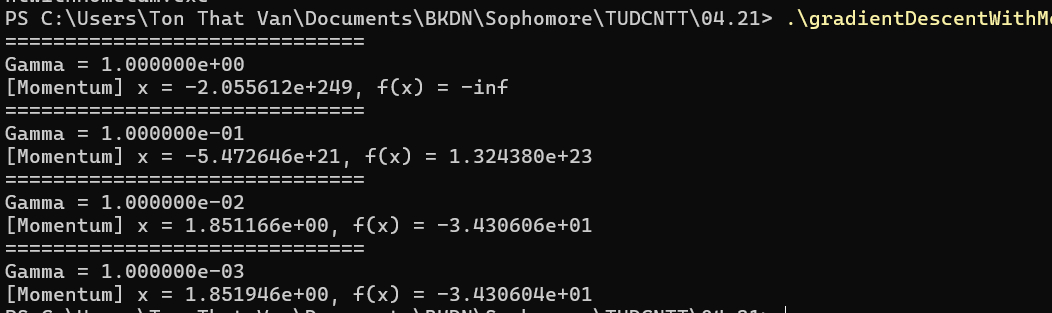
******

1. Thuật toán Gradient Descent với Momentum

***Source***

|  |
| --- |
| *#include <iostream>*  *#include <cmath>*  *#include <iomanip>*  *using namespace std;*  *// Hàm mục tiêu*  *double func(double x) {*  *if (fabs(15 \* x - 2) < 1e-8) return INFINITY;*  *return (exp(3 \* x) + 12 \* x \* x + 5 \* x) / (15 \* x - 2) - 25 \* x;*  *}*  *// Đạo hàm*  *double grad(double x) {*  *double v = 15 \* x - 2;*  *if (fabs(v) < 1e-8) return INFINITY;*  *double u = exp(3 \* x) + 12 \* x \* x + 5 \* x;*  *double u\_prime = 3 \* exp(3 \* x) + 24 \* x + 5;*  *double v\_prime = 15;*  *double df = (u\_prime \* v - u \* v\_prime) / (v \* v) - 25;*  *return df;*  *}*  *// Gradient Descent có Momentum*  *double gradientDescentMomentum(double x0, double gamma, double alpha) {*  *double x = x0;*  *double v = 0.0;*  *int iter = 0, max\_iter = 100000;*  *while (iter++ < max\_iter) {*  *double grad\_x = grad(x);*  *if (!isfinite(grad\_x)) break;*  *v = alpha \* v - gamma \* grad\_x;*  *double new\_x = x + v;*  *if (fabs(new\_x - x) < 1e-6) break;*  *x = new\_x;*  *}*  *return x;*  *}*  *int main() {*  *double x\_init = 2.0;*  *double alpha = 0.9;*  *cout << scientific << setprecision(6);*  *for (int i = 0; i < 4; i++) {*  *double gamma = pow(10, -i);*  *cout << "==============================" << endl;*  *cout << "Gamma = " << gamma << endl;*  *double x\_momentum = gradientDescentMomentum(x\_init, gamma, alpha);*  *cout << "[Momentum] x = " << x\_momentum*  *<< ", f(x) = " << func(x\_momentum) << endl;*  *}*  *return 0;*  *}* |

***Kết quả thực thi***

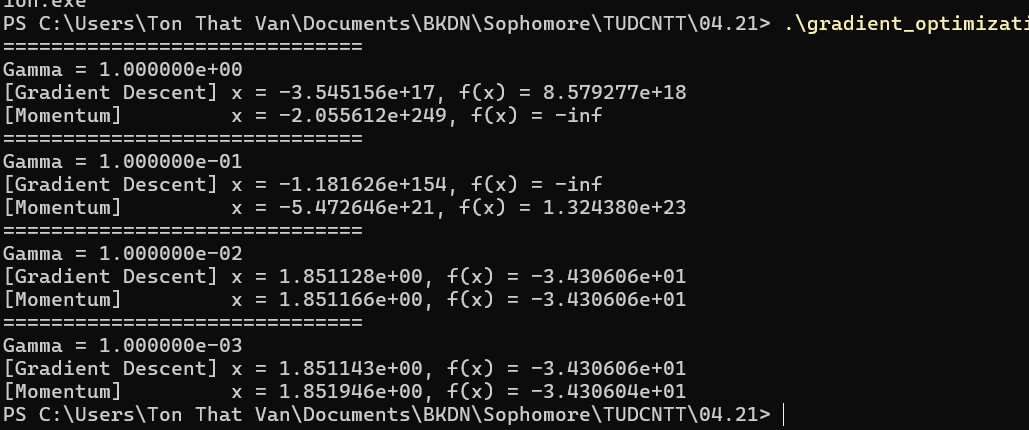
******

1. So sánh 2 thuật toán

***Source***

|  |
| --- |
| *#include <iostream>*  *#include <cmath>*  *#include <iomanip>*  *using namespace std;*  *// Hàm mục tiêu*  *double func(double x) {*  *if (fabs(15 \* x - 2) < 1e-8) return INFINITY; // Tránh chia cho 0*  *return (exp(3 \* x) + 12 \* x \* x + 5 \* x) / (15 \* x - 2) - 25 \* x;*  *}*  *// Đạo hàm chính xác*  *double grad(double x) {*  *double v = 15 \* x - 2;*  *if (fabs(v) < 1e-8) return INFINITY; // Tránh chia cho 0*  *double u = exp(3 \* x) + 12 \* x \* x + 5 \* x;*  *double u\_prime = 3 \* exp(3 \* x) + 24 \* x + 5;*  *double v\_prime = 15;*  *double df = (u\_prime \* v - u \* v\_prime) / (v \* v) - 25;*  *return df;*  *}*  *// Gradient Descent cơ bản*  *double gradientDescent(double x0, double gamma) {*  *double x = x0;*  *int iter = 0, max\_iter = 100000;*  *while (iter++ < max\_iter) {*  *double grad\_x = grad(x);*  *if (!isfinite(grad\_x)) break;*  *double new\_x = x - gamma \* grad\_x;*  *if (fabs(new\_x - x) < 1e-6) break;*  *x = new\_x;*  *}*  *return x;*  *}*  *// Gradient Descent có momentum*  *double gradientDescentMomentum(double x0, double gamma, double alpha) {*  *double x = x0;*  *double v = 0.0;*  *int iter = 0, max\_iter = 100000;*  *while (iter++ < max\_iter) {*  *double grad\_x = grad(x);*  *if (!isfinite(grad\_x)) break;*  *v = alpha \* v - gamma \* grad\_x;*  *double new\_x = x + v;*  *if (fabs(new\_x - x) < 1e-6) break;*  *x = new\_x;*  *}*  *return x;*  *}*  *int main() {*  *double x\_init = 2.0;*  *double alpha = 0.9;*  *cout << scientific << setprecision(6);*  *for (int i = 0; i < 4; i++) {*  *double gamma = pow(10, -i);*  *cout << "==============================" << endl;*  *cout << "Gamma = " << gamma << endl;*  *double x\_gd = gradientDescent(x\_init, gamma);*  *cout << "[Gradient Descent] x = " << x\_gd*  *<< ", f(x) = " << func(x\_gd) << endl;*  *double x\_momentum = gradientDescentMomentum(x\_init, gamma, alpha);*  *cout << "[Momentum] x = " << x\_momentum*  *<< ", f(x) = " << func(x\_momentum) << endl;*  *}*  *return 0;*  *}* |

***Kết quả thực thi***

******