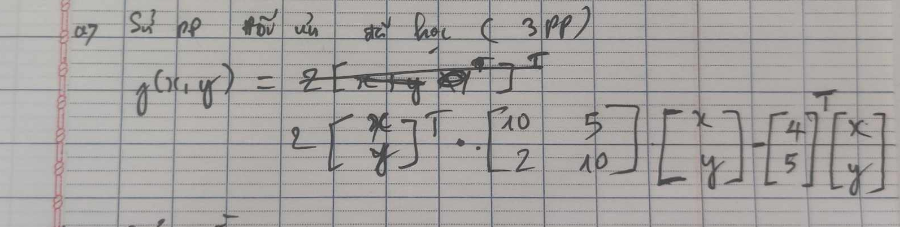
Họ và tên: Nguyễn Tấn Lộc

MSSV: 102220155

Nhóm: 22Nh11

Bài 1.

Đề:



Source code:

#include <iostream>

#include <vector>

#include <cmath>

#include <iomanip>

#define e 10e-5

using namespace std;

double hamMucTieu(double *x*, double *y*)

{

    return 20 \* *x* \* *x* + 14 \* *x* \* *y* + 20 \* *y* \* *y* - 4 \* *x* - 5 \* *y*;

}

vector<double> daoHam(double *x*, double *y*, double *h* = 1e-5)

{

    double dfdx = (hamMucTieu(*x* + *h*, *y*) - hamMucTieu(*x* - *h*, *y*)) / (2 \* *h*);

    double dfdy = (hamMucTieu(*x*, *y* + *h*) - hamMucTieu(*x*, *y* - *h*)) / (2 \* *h*);

    return {dfdx, dfdy};

}

vector<vector<double>> hessian(double *x*, double *y*, double *h* = 1e-5)

{

    double d2fdx2 = (hamMucTieu(*x* + *h*, *y*) - 2 \* hamMucTieu(*x*, *y*) + hamMucTieu(*x* - *h*, *y*)) / (*h* \* *h*);

    double d2fdy2 = (hamMucTieu(*x*, *y* + *h*) - 2 \* hamMucTieu(*x*, *y*) + hamMucTieu(*x*, *y* - *h*)) / (*h* \* *h*);

    double d2fdxdy = (hamMucTieu(*x* + *h*, *y* + *h*) - hamMucTieu(*x* + *h*, *y* - *h*) - hamMucTieu(*x* - *h*, *y* + *h*) + hamMucTieu(*x* - *h*, *y* - *h*)) / (4 \* *h* \* *h*);

    return {

        {d2fdx2, d2fdxdy},

        {d2fdxdy, d2fdy2}};

}

void phuongPhapNewtonRaphson(double *x*, double *y*, double *tol*, int *max\_iter*)

{

    int iter = 0;

    while (iter < *max\_iter*)

    {

        vector<double> grad = daoHam(*x*, *y*);

        vector<vector<double>> hess = hessian(*x*, *y*);

        double det = hess[0][0] \* hess[1][1] - hess[0][1] \* hess[1][0];

        if (fabs(det) < 1e-6)

            break;

        vector<vector<double>> inv\_hess = {

            {hess[1][1] / det, -hess[0][1] / det},

            {-hess[1][0] / det, hess[0][0] / det}};

*x* -= inv\_hess[0][0] \* grad[0] + inv\_hess[0][1] \* grad[1];

*y* -= inv\_hess[1][0] \* grad[0] + inv\_hess[1][1] \* grad[1];

        if (sqrt(grad[0] \* grad[0] + grad[1] \* grad[1]) < *tol*)

            break;

        iter++;

    }

    cout << "Ket qua phuong phap newton: x = " << *x* << ", y = " << *y* << "\n f(x, y) = " << hamMucTieu(*x*, *y*) << endl;

}

void phuongPhapMauGoc(double *x*, double *y*, double *learning\_rate*, double *tol*, int *max\_iter*)

{

    int iter = 0;

    while (iter < *max\_iter*)

    {

        vector<double> grad = daoHam(*x*, *y*);

*x* -= *learning\_rate* \* grad[0];

*y* -= *learning\_rate* \* grad[1];

        if (sqrt(grad[0] \* grad[0] + grad[1] \* grad[1]) < *tol*)

            break;

        iter++;

    }

    cout << "Ket qua voi phuong phap gradient descent:\n x = " << *x* << ", y = " << *y* << "\n f(x, y) = " << hamMucTieu(*x*, *y*) << endl;

}

void phuongPhapMauGocVoiMomen(double *x*, double *y*, double *learning\_rate*, double *momentum*, double *tol*, int *max\_iter*)

{

    double vx = 0, vy = 0;

    int iter = 0;

    while (iter < *max\_iter*)

    {

        vector<double> grad = daoHam(*x*, *y*);

        vx = *momentum* \* vx - *learning\_rate* \* grad[0];

        vy = *momentum* \* vy - *learning\_rate* \* grad[1];

*x* += vx;

*y* += vy;

        if (sqrt(grad[0] \* grad[0] + grad[1] \* grad[1]) < *tol*)

            break;

        iter++;

    }

    cout << "Ket qua phuong phap gradient descent voi momentum:\n x = " << *x* << ", y = " << *y* << "\n f(x, y) = " << hamMucTieu(*x*, *y*) << endl;

}

int main()

{

    double x = 0.0, y = 0.0, rate = 0.01, momentum = 0.9;

    int max = 10000;

    cout << "f(x, y) = 20x^2 + 14xy + 20y^2 - 4x - 5y\n";

    phuongPhapNewtonRaphson(x, y, e, max);

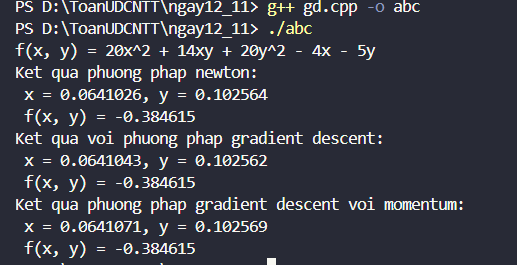
    phuongPhapMauGoc(x, y, rate, e, max);

    phuongPhapMauGocVoiMomen(x, y, rate, momentum, e, max);

    return 0;

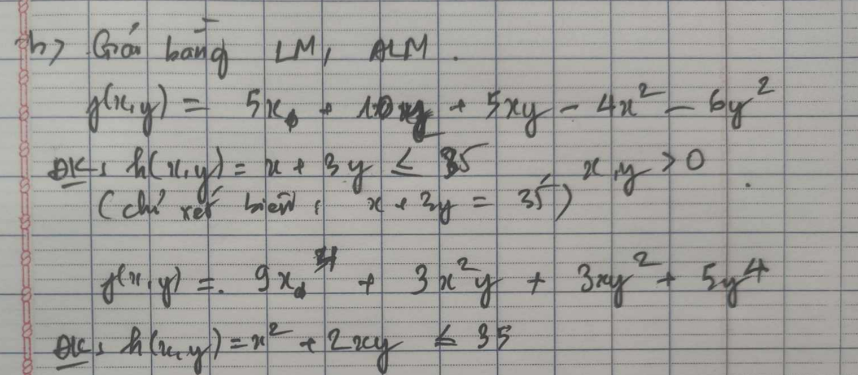
}

Kết quả:



Bài 2:

Đề:



a)

Source code:

#include <iostream>

#include <cmath>

using namespace std;

double f(double *x*, double *y*)

{

    return 5 \* *x* + 10 \* *y* + 5 \* *x* \* *y* - 4 \* *x* \* *x* - 6 \* *y* \* *y*;

}

double g(double *x*, double *y*)

{

    return *x* + 3 \* *y*;

}

void toiUuKKT(double (\**f*)(double, double), double (\**g*)(double, double))

{

    double x = 0, y = 0, lambda = 0;

    double buoc = 0.00001;

    double tol = 0.0001;

    int max\_iter = 1000;

    int iter = 0;

    while (iter < max\_iter)

    {

        double df\_dx = 5 + 5 \* y - 8 \* x;

        double df\_dy = 10 + 5 \* x - 12 \* y;

        double dg\_dx = 1;

        double dg\_dy = 3;

        cout << "Lan lap " << iter << ": g(x, y) = " << *g*(x, y)

             << ", df\_dx = " << df\_dx << ", df\_dy = " << df\_dy << endl;

        if (fabs(*g*(x, y) - 35) < tol)

        {

            double mau = dg\_dx \* dg\_dy - df\_dx \* df\_dy;

            if (fabs(mau) > 1e-8)

            {

                lambda = (df\_dx \* dg\_dy - df\_dy \* dg\_dx) / mau;

            }

            else

            {

                lambda = 0;

            }

        }

        else

        {

            lambda = 0;

        }

        double x\_moi = x - buoc \* (df\_dx - lambda \* dg\_dx);

        double y\_moi = y - buoc \* (df\_dy - lambda \* dg\_dy);

        if (fabs(x\_moi - x) < tol && fabs(y\_moi - y) < tol)

        {

            break;

        }

        x = x\_moi;

        y = y\_moi;

        iter++;

    }

    cout << "f(x) = 5x + 10y + 5xy - 4x^2 - 6y^2" << endl;

    cout << "Dieu kien: x + 3y <= 35" << endl;

    cout << "Ket qua tim duoc sau " << iter << " lan lap:\n";

    cout << "x = " << x << ", y = " << y << ", lambda = " << lambda << endl;

    cout << "Gia tri ham muc tieu f(x, y) = " << *f*(x, y) << endl;

}

int main()

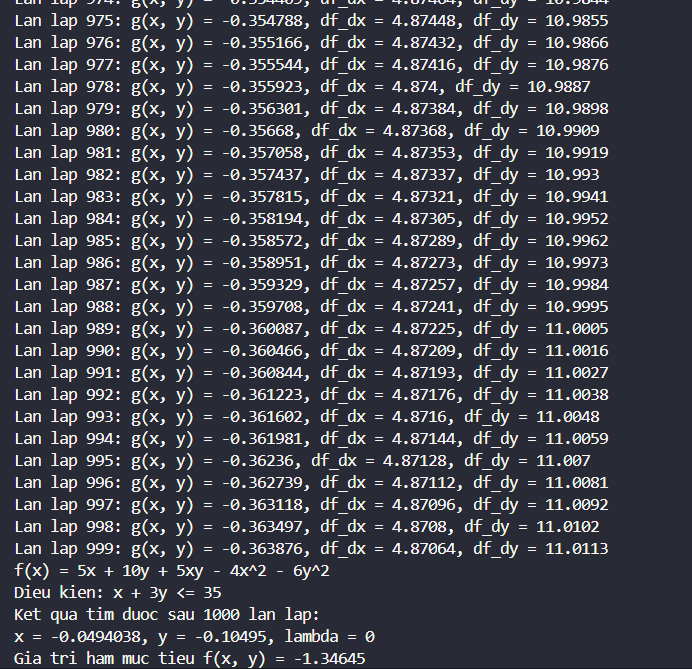
{

    toiUuKKT(f, g);

    return 0;

}

Kết quả:



b)

Source code:

#include <iostream>

#include <cmath>

using namespace std;

// Updated objective function f(x, y)

double f(double *x*, double *y*)

{

    return 9 \* pow(*x*, 4) + 3 \* *y* \* pow(*x*, 2) + 3 \* *x* \* pow(*y*, 2) + 5 \* pow(*y*, 4);

}

// Constraint function g(x, y)

double g(double *x*, double *y*)

{

    return *x* + 3 \* *y*;

}

// Modified optimization function with the new derivatives

void toiUuKKT(double (\**f*)(double, double), double (\**g*)(double, double))

{

    double x = 1, y = 1, lambda = 0;

    double buoc = 0.00001;

    double tol = 0.0001;

    int max\_iter = 1000;

    int iter = 0;

    while (iter < max\_iter)

    {

        // Partial derivatives of the new f(x, y)

        double df\_dx = 36 \* pow(x, 3) + 6 \* y \* x + 3 \* pow(y, 2);

        double df\_dy = 20 \* pow(y, 3) + 3 \* pow(x, 2) + 6 \* x \* y;

        double dg\_dx = 1;

        double dg\_dy = 3;

        cout << "Lan lap " << iter << ": g(x, y) = " << *g*(x, y)

             << ", df\_dx = " << df\_dx << ", df\_dy = " << df\_dy << endl;

        if (fabs(*g*(x, y) - 35) < tol)

        {

            double mau = dg\_dx \* dg\_dy - df\_dx \* df\_dy;

            if (fabs(mau) > 1e-8)

            {

                lambda = (df\_dx \* dg\_dy - df\_dy \* dg\_dx) / mau;

            }

            else

            {

                lambda = 0;

            }

        }

        else

        {

            lambda = 0;

        }

        double x\_moi = x - buoc \* (df\_dx - lambda \* dg\_dx);

        double y\_moi = y - buoc \* (df\_dy - lambda \* dg\_dy);

        if (fabs(x\_moi - x) < tol && fabs(y\_moi - y) < tol)

        {

            break;

        }

        x = x\_moi;

        y = y\_moi;

        iter++;

    }

    cout << "f(x) = 9x^4 + 3x^2y + 3xy^2 + 5y^4" << endl;

    cout << "Dieu kien: x + 3y <= 35" << endl;

    cout << "Ket qua tim duoc sau " << iter << " lan lap:\n";

    cout << "x = " << x << ", y = " << y << ", lambda = " << lambda << endl;

    cout << "Gia tri ham muc tieu f(x, y) = " << *f*(x, y) << endl;

}

int main()

{

    toiUuKKT(f, g);

    return 0;

}

Kết quả:

