Họ và tên: Trần Đức Long

Lớp: 23T\_Nhat1

MSSV: 102230027

Đồ thị

A graph on a graph paper

AI-generated content may be incorrect.

1. Gradient descent

A black screen with white dots

AI-generated content may be incorrect.

|  |
| --- |
| #include <iostream>  #include <math.h>  using namespace std;  const double LEARNING\_RATE = 0.01f;  const double TOLERANCE = 1e-6f;  const int ITERATIONS = 10000;  const double EPS = 1e-9f;  const double E = exp(1);  double ln(double x){      return log(x)/log(E);  }  double f(double x){      double tu = (3\*x\*x\*x + 3\*x\*x - 4\*x)\*(pow(E,x) + x)\*ln(abs(x));      double mau = 5\*pow(E,x);      return tu/mau;  }  double grad(double x){      return (f(x+EPS) - f(x-EPS))/(2\*EPS);  }  double gradient\_descent(double initial\_x){      double x = initial\_x;      for (int i=1;i<=ITERATIONS;++i){          double df = grad(x);          if (fabs(df) < TOLERANCE){              cout << "Converged at iteration " << i << "\n";              break;          }          if(i%10==0)cout << "i = " << i << " => x = " << x << " => f(x) = " << f(x) << '\n';          x = x - LEARNING\_RATE\*df;      }      return x;  }  int main(){      ios::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);      double x\_result = gradient\_descent(1);      cout << "Result:\n";      cout << "x = "  << x\_result << "\nf(x) = " << f(x\_result) << '\n';      return 0;  } |

1. Gradient descent with momentum

A screenshot of a computer

AI-generated content may be incorrect.

|  |
| --- |
| #include <iostream>  #include <math.h>  using namespace std;  const double LEARNING\_RATE = 0.01;  const double TOLERANCE = 1e-6;  const int ITERATIONS = 10000;  const double EPS = 1e-8;  const double E = exp(1);  const double MOMENTUM = 0.9;  const int INF = 1e9;  double ln(double x){      return log(x)/log(E);  }  double f(double x){      double tu = (3\*x\*x\*x + 3\*x\*x - 4\*x)\*(pow(E,x) + x)\*ln(abs(x));      double mau = 5\*pow(E,x);      return tu/mau;  }  double grad(double x){      return (f(x+EPS) - f(x-EPS))/(2\*EPS);  }  double gd\_momentum(double initial\_x){      double x = initial\_x;      double v = 0;      for (int i=1;i<=ITERATIONS;++i){          double df = grad(x);          if (fabs(df) < TOLERANCE){              cout << "Converged at iteration " << i << "\n";              break;          }          if(i%10==0)cout << "i = " << i << " => x = " << x << " => f(x) = " << f(x) << '\n';          v = MOMENTUM\*v + LEARNING\_RATE \* df;            double x\_new = x - v;          if (fabs(x-x\_new) < TOLERANCE){              cout << "Converged at iteration " << i << "\n";              break;          }          x = x\_new;      }      return x;  }  int main(){      ios::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);      double x\_result = gd\_momentum(2);      cout << "x = " << x\_result << '\n';      cout << "f(x) = " << f(x\_result) << '\n';      return 0;  } |