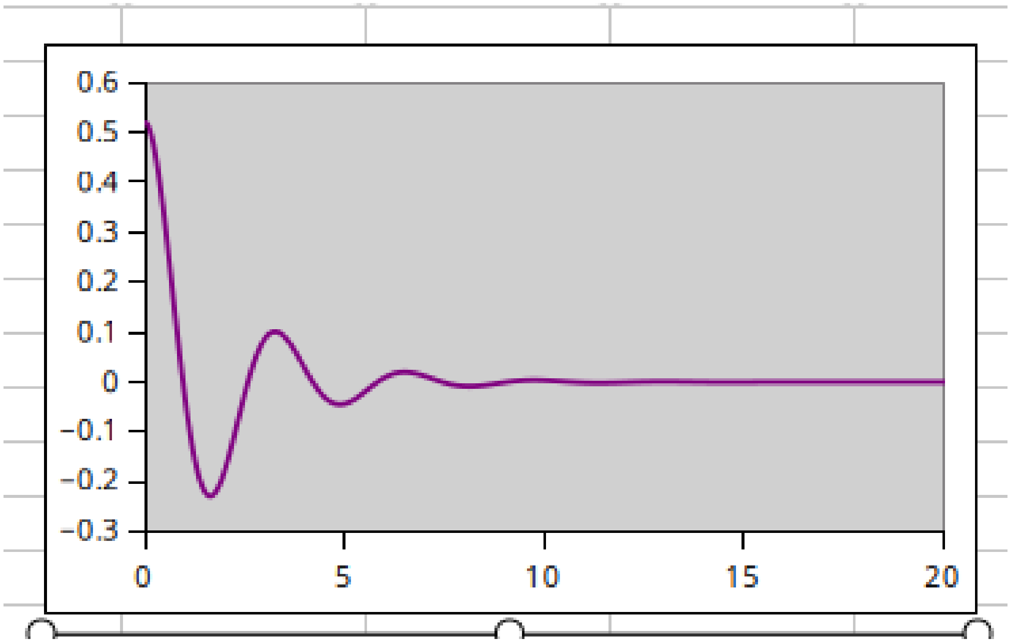
Forced-damped pendulum



```
#include <iostream>
#include <cmath>
#include <fstream>

using namespace std;
const double W0=2.0;
const double A=1.07;
const double b=0.5;
const double Wd=2.0;

double f1(double x1, double x2,double t){
    return x2;
    }
double f2(double x1, double x2, double x3, double t){
    return -W0*W0*sin(x1)-b*x2+A*cos(x3);
}
double f3(double t){
    return Wd;
}

int main(){
    double T=80.0;
    double tau=0.001;
    int nmax=T/tau;
 double x1[nmax+1],x2[nmax+1],x3[nmax+1];
    cout<<"Enter the intital angle (x1[0] in radians): ";
    cin>>x1[0];
    cout<<"Enter the intital angular velocity (x2[0] in rad/s): ";
    cin>>x2[0];


ofstream file("pendulum.txt");

    for(int n=0;n<nmax;n++){
    double t=n*tau;

 double k1[3], k2[3], k3[3], k4[3];

    k1[0]=f1(x1[n],x2[n],t);
    k1[1]=f2(x1[n],x2[n],x3[n],t);
    k1[2]=f3(t);

    k2[0]=f1(x1[n]+0.5*tau*k1[0],x2[n]+0.5*tau*k1[1],t+0.5*tau);

k2[1]=f2(x1[n]+0.5*tau*k1[0],x2[n]+0.5*tau*k1[1],x3[n]+0.5*tau*k1[2],t
+0.5*tau);
    k2[2]=f3(t+0.5*tau);

    k3[0]=f1(x1[n]+0.5*tau*k2[0],x2[n]+0.5*tau*k2[1],t+0.5*tau);

k3[1]=f2(x1[n]+0.5*tau*k2[0],x2[n]+0.5*tau*k2[1],x3[n]+0.5*tau*k2[2
],t+0.5*tau);
    k3[2]=f3(t+0.5*tau);

    k4[0]=f1(x1[n]+tau*k3[0],x2[n]+tau*k3[1],t+tau);
    k4[1]=f2(x1[n]+tau*k3[0],x2[n]+tau*k3[1],x3[n]+tau*k3[2],t+tau);
    k4[2]=f3(t+tau);

    x1[n+1]=x1[n]+tau*(k1[0]+2*k2[0]+2*k3[0]+k4[0])/6.0;
    x2[n+1]=x2[n]+tau*(k1[1]+2*k2[1]+2*k3[1]+k4[1])/6.0;
    x3[n+1]=x3[n]+tau*(k1[2]+2*k2[2]+2*k3[2]+k4[2])/6.0;

    file<<t<<'\t'<<x1[n]<<'\t'<<x2[n]<<'\t'<<x3[n]<<'\n';
    }
    file.close();
    return 0;
    }
```