#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include "Evolution.h"

// Structure definition that can hold the data

struct XYdata

{

int n;

double\* x;

double\* y;

// constructor

XYdata(int, double\*, double\*);

};

// definition of constructor

XYdata::XYdata(int n, double\* x, double\* y)

{

this->n = n;

this->x = new double[n];

this->y = new double[n];

memcpy(this->x, x, n\*sizeof(double));

memcpy(this->y, y, n\*sizeof(double));

}

// the objective function that defines the function to be minimized

double Evolution::objective(double\* theta, int np, void\* data)

{

for(int i=0; i<np; i++)

if(theta[i] <= 0.0)

return HUGE\_VAL;

XYdata\* dat = (XYdata\*) data;

// we are minimizing the sum of sq of the residuals for the given data and model

// ssr = sum of sq for residuals

double ssr = 0.0;

for(int i =0; i<dat->n; i++)

{

double n = dat->x[i];

double y = dat->y[i];

// model is y = f(t,theta) = theta1 - theta1\* exp(-theta2\*x)

// so residual sq is (datay-modely)^2

// theta[0] is theta1 and theta[1] is theta2 here

ssr += pow(y - (theta[0] - theta[0]\*exp(-theta[1]\*n)), 2);

}

return ssr;

}

int main()

{

// Time (days) independent variable

double days[] = {1,2,3,4,5,7};

// BOD (mg/l) dependent variable

double bod[] = {8.3,10.3,19,16,16.6,19.8};

// create a data structure

XYdata\* data = new XYdata(6, days, bod);

// number of parameters (theta1 and theta2)

int np = 2;

// create the Evolution class

Evolution\* ev = new Evolution(np, 10\*np, data);

ev->set\_wt\_factor(0.8);

ev->set\_seed(9911);

ev->toggle\_display();

// define lower and upper bound for theta1 and theta2

double lower[] = {9, 9};

double upper[] = {18, 18};

// create a pointer to hold the estimated parameter values

double\* est = new double[np];

// run the differential evolution

ev->evolve(lower, upper, est);

// print the estimated values of the parameters

fprintf(stdout, "Final parameter values\n");

for(int i=0; i<np; i++)

fprintf(stdout, "Estimation of theta[%d]: %12.6g\n", i+1, est[i]);

return 0;

}

**OUTPUT**

**Iteration Member Value**

**0 6 110.043**

**1 15 109.864**

**2 5 109.6**

**3 15 108.719**

**4 13 69.1864**

**5 9 33.2517**

**6 7 29.6075**

**7 7 29.6075**

**8 7 29.6075**

**9 7 29.6075**

**10 7 29.6075**

**11 7 29.6075**

**12 13 29.1143**

**13 7 23.8736**

**14 11 22.5735**

**15 11 22.5735**

**16 11 22.5735**

**17 11 22.5735**

**18 11 22.5735**

**19 11 22.5735**

**20 11 22.5735**

**21 10 22.4719**

**22 17 22.3648**

**23 6 22.3169**

**24 6 22.3169**

**25 3 22.3119**

**26 3 22.3119**

**27 6 22.3119**

**28 6 22.3119**

**29 16 22.3117**

**30 8 22.3116**

**31 8 22.3116**

**32 8 22.3116**

**33 8 22.3116**

**34 8 22.3116**

**35 8 22.3115**

**36 6 22.3114**

**37 6 22.3114**

**38 4 22.3114**

**39 17 22.3114**

**40 10 22.3114**

**41 18 22.3114**

**42 9 22.3114**

**43 9 22.3114**

**44 9 22.3114**

**45 9 22.3114**

**46 4 22.3114**

**47 17 22.3114**

**48 17 22.3114**

**49 15 22.3114**

**Final parameter values**

**Estimation of theta[1]: 19.5922**

**Estimation of theta[2]: 0.512447**