Biyection code in C++

Andrés Mateo Otálvaro, Santiago Suárez Pérez, Daniel Ermilson Velásquez

September 16, 2015

Listing 1: Biyection algorithm in C++

```
#include <iostream>
#include <cmath>
using namespace std;

long double f(long double x);

int main(){
  cout << "Give me a range you know there's a root for sin(x),
then tolerance, and then iterations" << endl;
  long double xi=0,xs=0,tol=0,iter=0,yi=0,ys=0;
  yi = f(xi);
  ys = f(xs);
  cin >> xi >> xs >> tol >> iter;
  if(yi*ys==0){
    cout << "Roots are equals" << endl;
    return 0;
  }else if(yi==0){
    cout << xi << " is a root" << endl;
    return 0;
  }else if(ys==0){
    cout << xs << " is a root" << endl;
    return 0;
  }else{
    long double xm = (xi+xs) / 2;
    long double ym = f(xm);
    long double error = tol * 2;
    long double cont = 1;
    while(ym!=0 and error>tol and cont <= iter){
      if(ym*yi == 0){
    xs = xm;
    ys = ym;
      }else{
    xi = xm;
    yi = ym;
      }
      long double aux = xm;
      xm = (xi + xs) / 2;
      error = abs(xm-aux);
      cont++;
    }
    if(ym==0){
      cout << xm << " is a root" << endl;
      return 0;
    }else if(error<tol){
      cout << xm << " is a root. Error=" << error << endl;
    }else if(cont>iter){
      cout << "iterations over. root not found" << endl;
    }
  }
  return 0;
}

long double f(long double x){
  return sin(x);
}
```