

# Bijection code in C++

Andrés Mateo Otálvaro, Santiago Suárez Pérez, Daniel Ermilson Velásquez

September 9, 2015

Listing 1: Bijection algorithm in C++

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 long double f(long double x);
6
7 int main(){
8     cout << "Give me a range you know there's a root for sin(x),
9 then tolerance, and then iterations" << endl;
10    long double xi=0,xs=0,tol=0,iter=0,yi=0,ys=0;
11    yi = f(xi);
12    ys = f(xs);
13    cin >> xi >> xs >> tol >> iter;
14    if(yi*ys==0){
15        cout << "Roots are equals" << endl;
16        return 0;
17    }else if(yi==0){
18        cout << xi << " is a root" << endl;
19        return 0;
20    }else if(ys==0){
21        cout << xs << " is a root" << endl;
22        return 0;
23    }else{
24        long double xm = (xi+xs) / 2;
25        long double ym = f(xm);
26        long double error = tol * 2;
27        long double cont = 1;
28        while(ym!=0 and error>tol and cont <= iter){
29            if(ym*yi == 0){
30                xs = xm;
31                ys = ym;
32            }else{
33                xi = xm;
```

```

34     yi = ym;
35     }
36     long double aux = xm;
37     xm = (xi + xs) / 2;
38     error = abs(xm-aux);
39     cont++;
40 }
41 if(ym==0){
42     cout << xm << " is a root" << endl;
43     return 0;
44 }else if(error<tol){
45     cout << xm << " is a root. Error=" << error << endl;
46 }else if(cont>iter){
47     cout << "iterations over. root not found" << endl;
48 }
49 }
50 return 0;
51 }
52
53 long double f(long double x){
54     return sin(x);
55 }

```