**Name Of The Experiment: Finding the root of a function using bisection method.**

**Name: Showkot Hossain**

**Roll: 42**

**Problem statement:**

1. f(x)=e^x-5\*x^2
2. f(x)=-139.34411 + 1.575701\*(10^5)/x - 6.642308\*(10^7)/(x^2) + 1.2438\*(10^10)/(x^2)- 8.621949\*(10^11)/(x^4) – ln(8);

Find the root of the above functions using bisection methods.

**Solution (c++ code):**

**Program 1:**

**#include <stdio.h>**

**#include <string.h>**

**#include <stdlib.h>**

**#include <ctype.h>**

**#include <math.h>**

**#include <limits.h>**

**#include <set>**

**#include <algorithm>**

**#include <iostream>**

**#include <vector>**

**#include <stack>**

**#include <string>**

**#include <list>**

**#include <map>**

**#include <queue>**

**#include <sstream>**

**#include <utility>**

**#define maxn 1005**

**using namespace std;**

**FILE \*fpw,\*fpr,\*fpw1;**

**int n;**

**int flag;**

**double e=2.71;**

**int osf;**

**double f(double x)**

**{**

**double ret=pow(e,x)-5\*x\*x;**

**return ret;**

**}**

**double bisection(double a,double b,double tol) {**

**flag=true;**

**if (f(a)\*f(b)>=0) {**

**cout << "You have not assumed right a and b\n";**

**flag=false;**

**return 0;**

**}**

**int tf=0;**

**double c,pc;int it=1;**

**while(true) {**

**c=(a+b)/2;**

**//fprintf(fpw, "%.8lf,%.8lf\n",c,f(c));**

**if(tf) {**

**double ftol=fabs(pc-c);**

**if(ftol<0) {**

**ftol\*=(-1.00);**

**}**

**ftol/=fabs(c);**

**fprintf(fpw, "%d,%.8lf\n",it,ftol);**

**if(ftol<=tol) {**

**return c;**

**}**

**}**

**else**

**{**

**fprintf(fpw, "%d,0.00\n",it);**

**}**

**if (f(c) == 0.0)**

**{**

**return c;**

**}**

**else if (f(c)\*f(a)<0.000)**

**b=c;**

**else**

**a=c;**

**tf=1;**

**pc=c;**

**it++;**

**}**

**}**

**int main() {**

**fpw1 = fopen("outpu1.csv", "w+");**

**for(double i=-1;i<=1;i+=0.1)**

**{**

**cout<<"x: "<<i<<" f(x): "<<f(i)<<endl;**

**fprintf(fpw1, "%.8lf,%.8lf\n",i,f(i));**

**}**

**fpw = fopen("output2.csv", "w+");**

**double a=-5000,b=5000,prec=.0000055;**

**cin>>a>>b>>prec;**

**double ret=bisection(a,b,prec);**

**cout<<"Root : "<<ret<<endl;**

**return 0;**

**}**

**Program 2:**

**#include <stdio.h>**

**#include <string.h>**

**#include <stdlib.h>**

**#include <ctype.h>**

**#include <math.h>**

**#include <limits.h>**

**#include <set>**

**#include <algorithm>**

**#include <iostream>**

**#include <vector>**

**#include <stack>**

**#include <string>**

**#include <list>**

**#include <map>**

**#include <queue>**

**#include <sstream>**

**#include <utility>**

**#define maxn 1005**

**using namespace std;**

**FILE \*fpw,\*fpr,\*fpw1;**

**int n;**

**int flag;**

**double e=2.71;**

**int osf;**

**double f(double y)**

**{**

**double x=y+273.15;**

**return -139.34411+1.575701\*pow(10,5)/x-6.642308\*pow(10,7)/(x\*x)+1.2438\*pow(10,10)/(x\*x\*x)-8.621949\*pow(10,11)/(pow(x,4))-log(osf);**

**}**

**double bisection(double a,double b,double tol) {**

**flag=true;**

**if (f(a)\*f(b)>=0) {**

**cout << "You have not assumed right a and b\n";**

**flag=false;**

**return 0;**

**}**

**int tf=0;**

**double c,pc;int it=1;**

**while(true) {**

**c=(a+b)/2;**

**//fprintf(fpw, "%.8lf,%.8lf\n",c,f(c));**

**if(tf) {**

**double ftol=fabs(pc-c);**

**if(ftol<0) {**

**ftol\*=(-1.00);**

**}**

**ftol/=fabs(c);**

**fprintf(fpw, "%d,%.8lf\n",it,ftol);**

**if(ftol<=tol) {**

**return c;**

**}**

**}**

**else**

**{**

**fprintf(fpw, "%d,0.00\n",it);**

**}**

**if (f(c) == 0.0)**

**{**

**return c;**

**}**

**else if (f(c)\*f(a)<0.000)**

**b=c;**

**else**

**a=c;**

**tf=1;**

**pc=c;**

**it++;**

**}**

**}**

**int main() {**

**fpw1 = fopen("output1.csv", "w+");**

**cin>>osf;**

**for(double i=0;i<=40;i+=1)**

**{**

**cout<<"x: "<<i<<" f(x): "<<f(i)<<endl;**

**fprintf(fpw1, "%.8lf,%.8lf\n",i,f(i));**

**}**

**fpw = fopen("outpu2.csv", "w+");**

**double a=-5000,b=5000,prec=.0000055;**

**cin>>a>>b>>prec;**

**double ret=bisection(a,b,prec);**

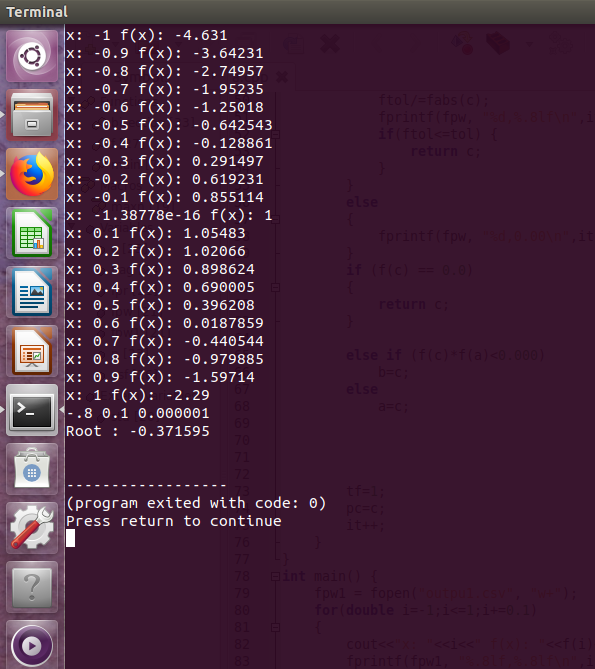
**cout<<"Root : "<<ret<<endl;**

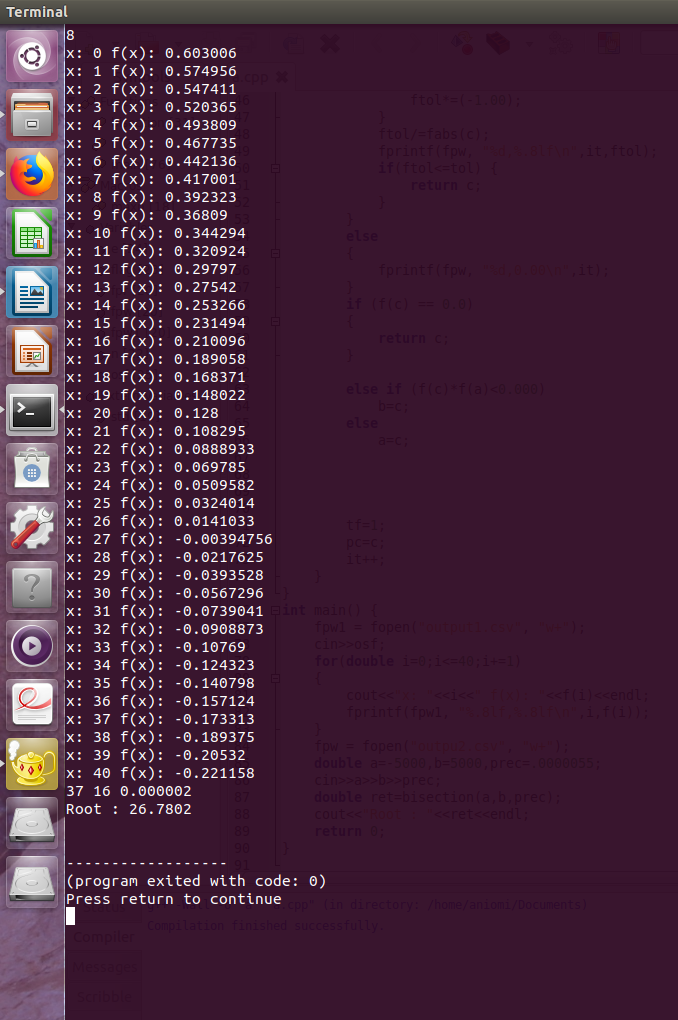
**return 0;**

**}**

**Sample Input/Output:**

**Input / Output of problem 1:**

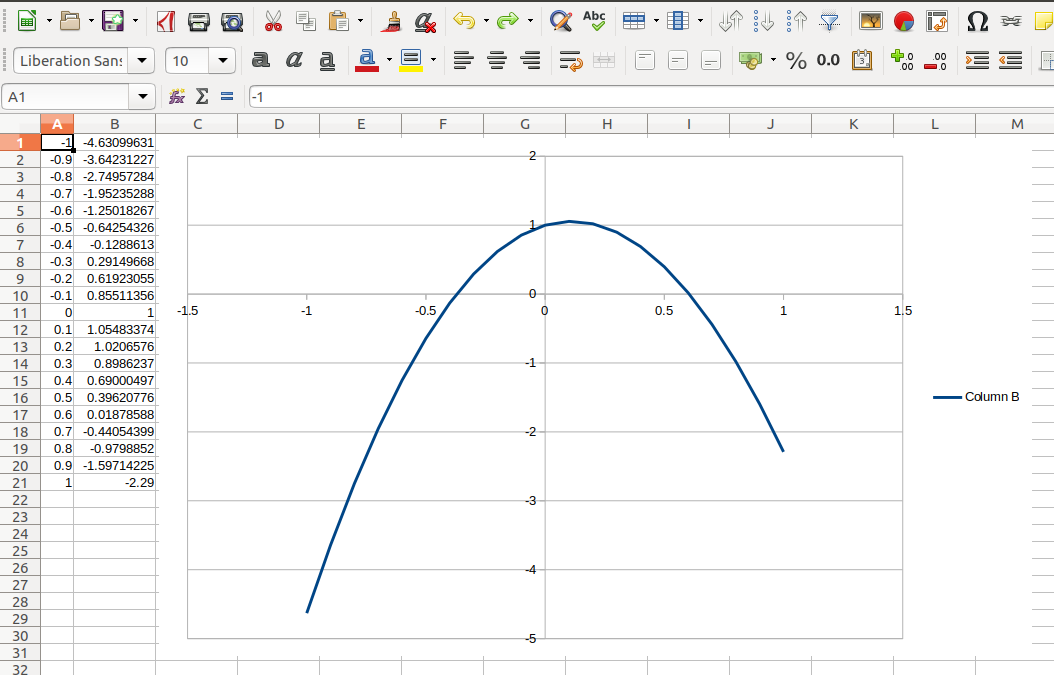
****

**Input / Output for Problem 2:**

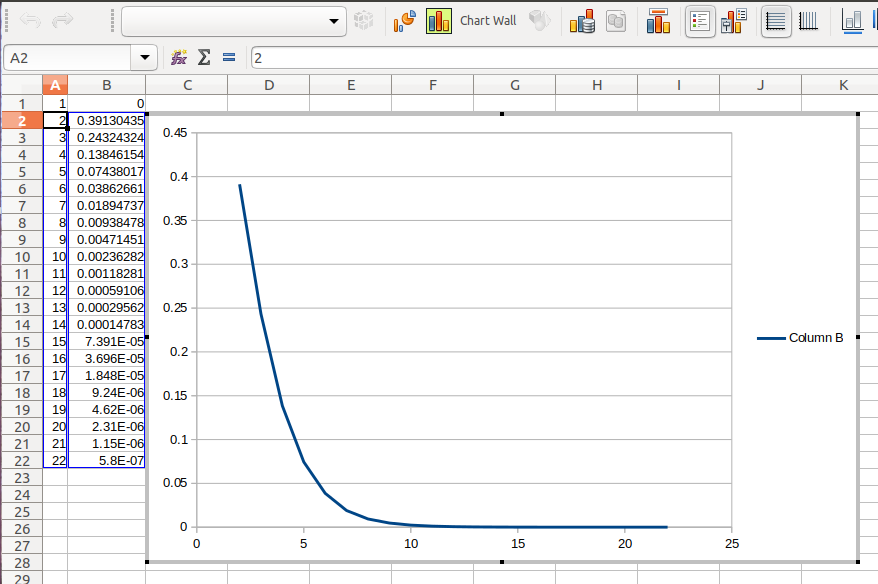
**Graphs:**

**Problem 1:**

**x vs f(x):**

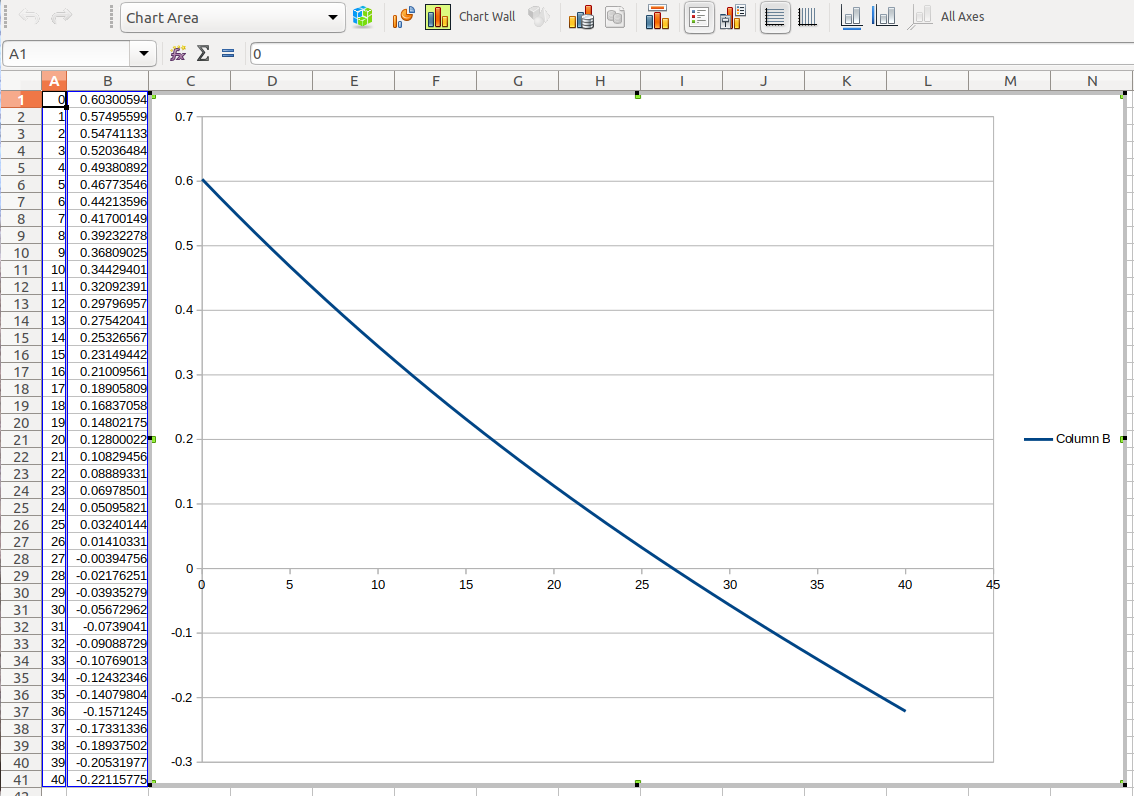
****

**iteration vs relative approximate error:**

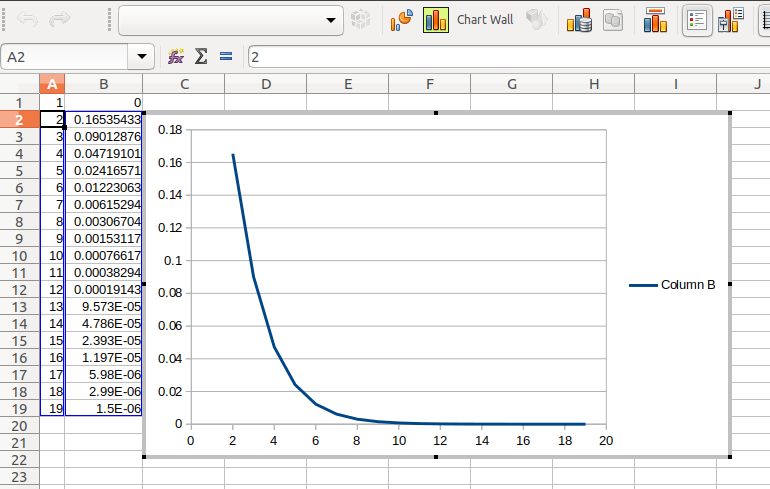
****

**Problem 2:**

**x vs f(x):**

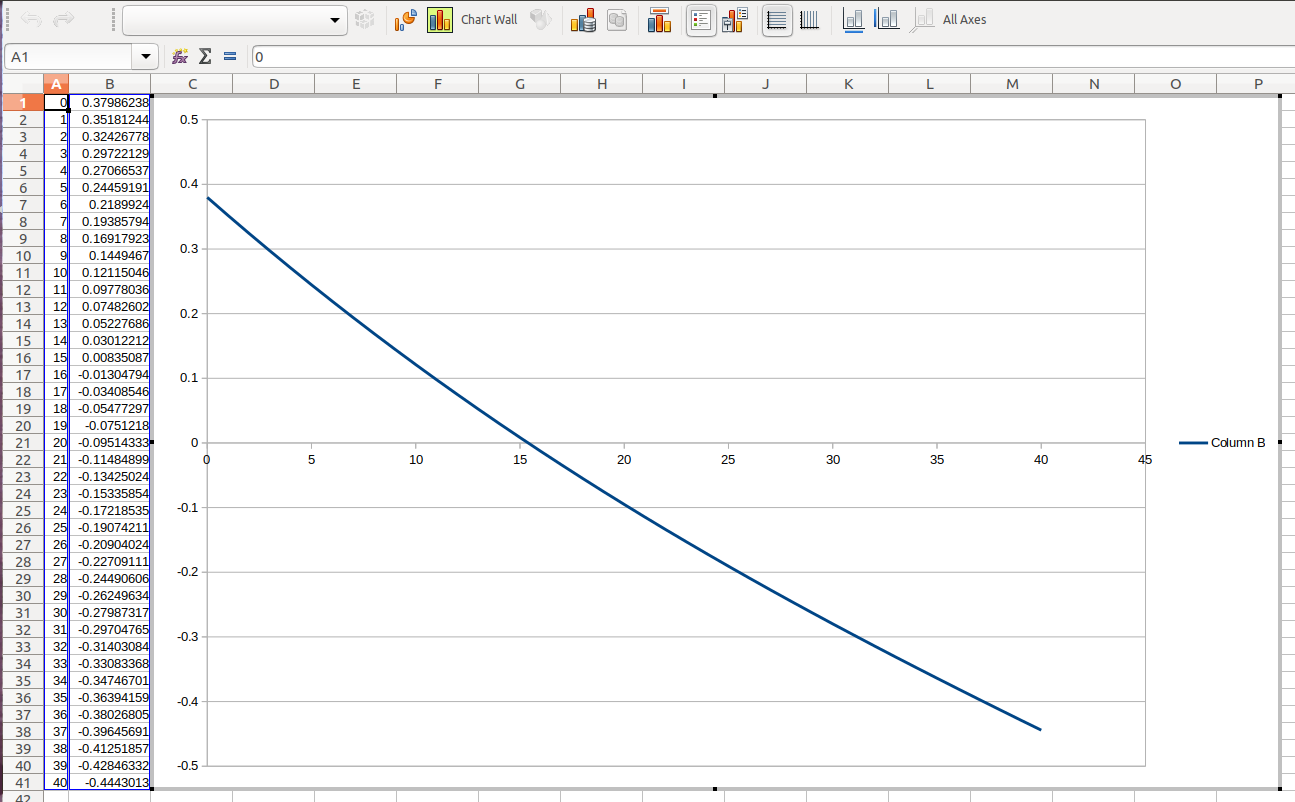
****

**iteration vs relative approximate error:**

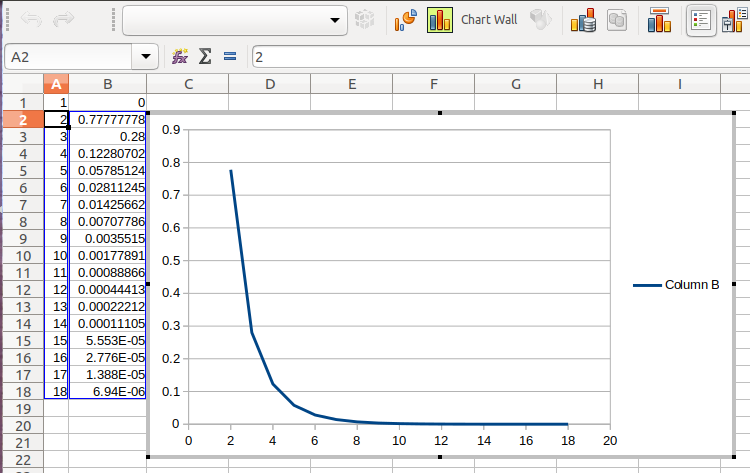
****

**Problem 3:**

**x vs f(x):**

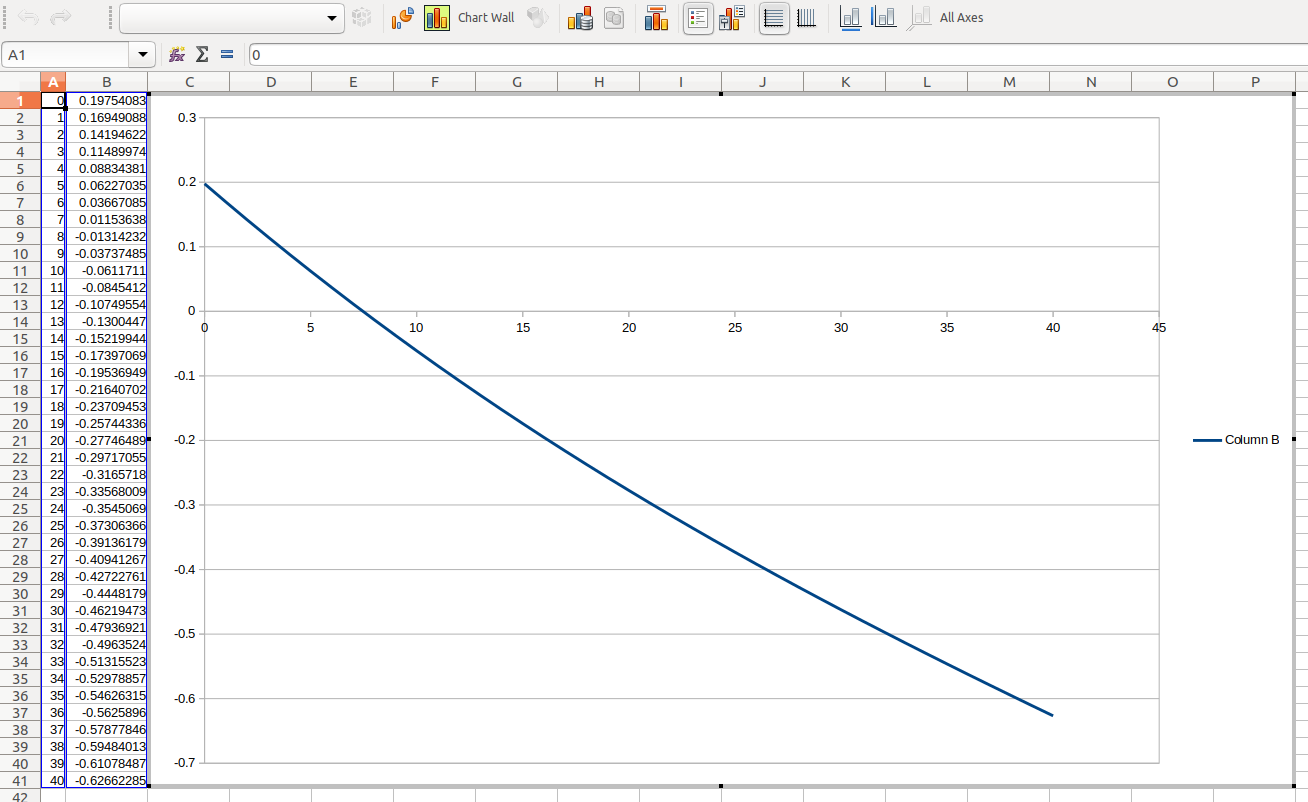
****

**iteration vs relative approximate error:**

****

**Problem 4:**

**x vs f(x):**

****

**iteration vs relative approximate error:**

