# Numerical analysis Hw7

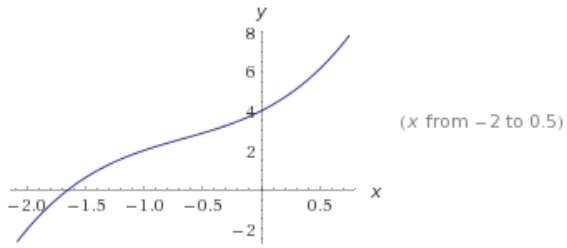
## 1. Implementation:

I implement this homework in C++ code. It supports finding one of the roots of double-format-parameters, polynomial functions.

## 2. Experiment:

i. $f(x) = x^3 + 2x^2 + 3x + 4$

Plots:

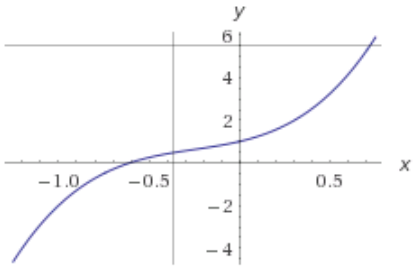



(x from −2 to 0.5)

```
C:\Users\user\Dropbox\Courses\Junior_2\NumericalAnalysis\hw7>hw7.exe 4 3 2 1
Root of bisection -1.650629
Root of newton -1.650629
```

ii. $f(x) = 4x^3 + 3x^2 + 2x + 1$

Plots:




(x from −1 to 0.5)

```
C:\Users\user\Dropbox\Courses\Junior_2\NumericalAnalysis\hw7>hw7.exe 1 2 3 4
Root of bisection -0.605830
Root of newton -0.605830
```

## 3. Code implementation :

```cpp
#include <cstdio>
#include <cstdlib>
#include <cfloat>
#include <vector>

using namespace std;
const double eps = 1e-10;

vector<double> vct;
vector<double> dvct;

void parseArg(int argc, char *argv[]) {
    /* function parameter in reverse order */
    if (argc <= 1) {
        fprintf(stderr, "Please insert function parameters in reverse order.\n");
        exit(EXIT_FAILURE);
    }

    for (int i = 1; i < argc; i++) {
        double d = strtod(argv[i], NULL);
        vct.push_back(d);
        if (i > 1) dvct.push_back((double)d * (i - 1));
    }
}

double cal(double x, vector<double> &v) {
    double ret = 0;
    double mul = 1;
    for (vector<double>::iterator it = v.begin(); it != v.end(); it++) {
        ret += *it * mul;
        mul *= x;
    }
    return ret;
}

double bisection(double a = -DBL_MAX, double b = DBL_MAX) {
    while (true) {
        double p = (a + b) / 2;
        if (cal(p, vct) <= eps && cal(p, vct) >= -eps) return p;
        else if (cal(a, vct) * cal(p, vct) < 0) b = p;
        else a = p;
    }
}

double newton(double x = 0) {
    while (true) {
        if (cal(x, vct) <= eps && cal(x, vct) >= -eps) return x;
        x += -cal(x, vct) / cal(x, dvct);
    }
}

int main(int argc, char *argv[]) {
    /* read in f(x) parameter */
    parseArg(argc, argv);

    /* bisection */
    printf("Root of bisection %f\n", bisection());

    /* newton */
    printf("Root of newton %f\n", newton());
}
```