

Comparison of code with and without functions

In this handout we compare computational fragments of the Quadratic Equation program (Lab 6) with equivalent fragments that make use of functions. There are two functions to be discussed.

The first one, `linRoot`, computes the solution of the linear equation

$$bx + c = 0,$$

which is of course

$$x = -\frac{c}{b}$$

(We are interested in the regular case $b \neq 0$ only.)

The second one, `QuadRoots`, computes two roots of the quadratic equation

$$ax^2 + bx + c = 0$$

at once:

$$x_1 = \frac{-b - \sqrt{D}}{2a}, \quad x_2 = \frac{-b + \sqrt{D}}{2a},$$

assuming, again, the regular case: $a \neq 0$, $D = b^2 - 4ac \geq 0$. The trivial circumstance that we wish to receive two values back from the function rather than just one makes the technical details significantly more difficult (hopefully, not too difficult on an absolute scale). We discuss two approaches: a strict ANSI C and another, C++ish. The latter is syntactically simpler, with only drawback of violating “purity” of the ANSI C standard.

The C version with functions is the file `lab6_func.c`, the C++ version is `lab6_func.cxx`. The former compiles as usually with `gcc` compiler:

```
gcc -lm lab6_func.cxx
```

To compile the latter, one needs the C++ compiler called `g++`, so the command line should be

```
g++ lab6_func.cxx
```

(The `g++` compiler automatically finds math library; no `-lm` option is required.)

1. Root of linear equation

No functions (file lab6.c):

```
int main()
{
    .....
    /* Case of linear equation a*x+b=c */
    x1=-c/b;
    .....
}
```

With functions (files lab6_func.c, lab6_func.cxx):

```
/* Declaration (prototype) of the function */
double linRoot(double u, double v);

int main()
{
    .....
    /* Case of linear equation a*x+b=c */
    x1=linRoot(b,c);
    .....
}

/* Implementation (definition) of the function */
double linRoot(double u, double v)
{
    return (-v/u);
}
```

2. Roots of quadratic equation

No functions (file lab6.c):

```
int main()
{
    .....
    /* Main case (a!=0), main subcase (D>=0) */
    x1=(-b-sqrt(D))/(2.0*a);
    x2=(-b+sqrt(D))/(2.0*a);
    .....
}
```

With functions

Declaration (prototype) of the function, placed before main()

ANSI C style (file lab6_func.c):

```
void quadRoots(double a, double b, double D, double *X1, double *X2);
```

C++ style (file lab6_func.cxx):

```
void quadRoots(double a, double b, double D, double& x1, double& x2);
```

Body of main()

```
int main()
{
    .....
    /* Main case (a!=0), main subcase (D>=0) */
    quadRoots(a,b,D, &x1, &x2);    /* ANSI C style */
```

or

```
    quadRoots(a,b,D, x1, x2);    /* C++ style */
    .....
} /* end of main() */
```

Implementation (definition) of the function, placed after main()

ANSI C style

```
void quadRoots(double a, double b, double D, double* X1, double* X2)
{
    double sqrtD; /* to avoid calculating sqrt(D) twice */
    sqrtD=sqrt(D);
    (*X1)=(-b-sqrtD)/(2.0*a);
    (*X2)=(-b+sqrtD)/(2.0*a);
}
```

C++ style

```
void quadRoots(double a, double b, double D, double& x1, double& x2)
{
    double sqrtD; /* to avoid calculating sqrt(D) twice */
    sqrtD=sqrt(D);
    x1=(-b-sqrtD)/(2.0*a);
    x2=(-b+sqrtD)/(2.0*a);
}
```