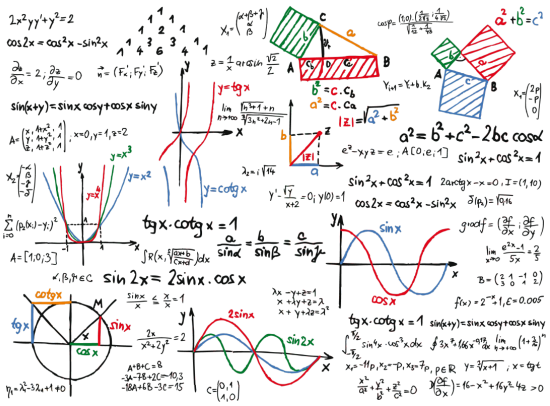
B1 - Mathematics

B-MAT-100

105torus

Mathematics of the donut



2.0

105torus

binary name:	105torus
repository name:	105torus_*$ACADEMIC_YEAR*
repository rights:	ramassage-tek
language:	C, C++, python3, perl, ruby, php or bash
compilation:	when necessary, via Makefile, including re, clean and fclean rules

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Drawing circles, cylinders and cones is a good start for an image synthesis software, but one have to admit it is not fully satisfying… This project is the continuation of the previous one, and should allow you to draw more complex forms, such as tores, which do not emerge from 2nd degree equations, but from superior degree equations (4th degree in the torus case).

The objective of this project is to solve a 4th degree equation: $a_4x^4 + a_3x^3 + a_2x^2 + a_1x_1 + a_0 = 0$. A direct resolution method does exist (Ferrari's method), but does not generalize. Thus, we will rather compare 3 iterative algorithms:

- The bisection method,
- Newton's method,
- The secant method.

Equations to be solved here will all have one and only one solution, in the $[0; 1]$ interval. This is the solution we are looking for. The initial value for Newton's method will be 0.5, those for the 2 other methods will be 0 and 1.

Just in case you would need it, the derivative of the polynomial function $x \mapsto a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ is the function $x \mapsto 4a_4x^3 + 3a_3x^2 + 2a_2x + a_1$

USAGE

Terminal

```
~/B-MAT-100> ./105torus -h
USAGE
    ./105torus opt a0 a1 a2 a3 a4 n

DESCRIPTION
    opt       method option:
                  1 for the bisection method
                  2 for Newton's mehod
                  3 for the secant method
    a[0-4]    coefficients of the equation
    n         precision (the application of the polynomial to the solution should
              be smaller than 10^-n)
```

SUGGESTED BONUSES

- Graphical interface to compare the rates of convergence,
- Solving higher degree equation.

EXAMPLES

```
Terminal
~/B-MAT-100> ./105torus 1 -1 0 6 -5 1 6
x = 0.5
x = 0.75
x = 0.625
x = 0.5625
x = 0.53125
x = 0.515625
x = 0.523438
x = 0.519531
x = 0.521484
x = 0.522461
x = 0.522949
x = 0.522705
x = 0.522827
x = 0.522766
x = 0.522736
x = 0.522751
x = 0.522743
x = 0.522739
x = 0.522741
x = 0.522740
```

```
Terminal
~/B-MAT-100> ./105torus 2 -1 0 6 -5 1 12
x = 0.5
x = 0.522727272727
x = 0.522740003514
x = 0.522740003526
```

```
Terminal
~/B-MAT-100> ./105torus 3 -1 0 6 -5 1 8
x = 0.5
x = 0.52941176
x = 0.52274853
x = 0.52274000
x = 0.52274000
```