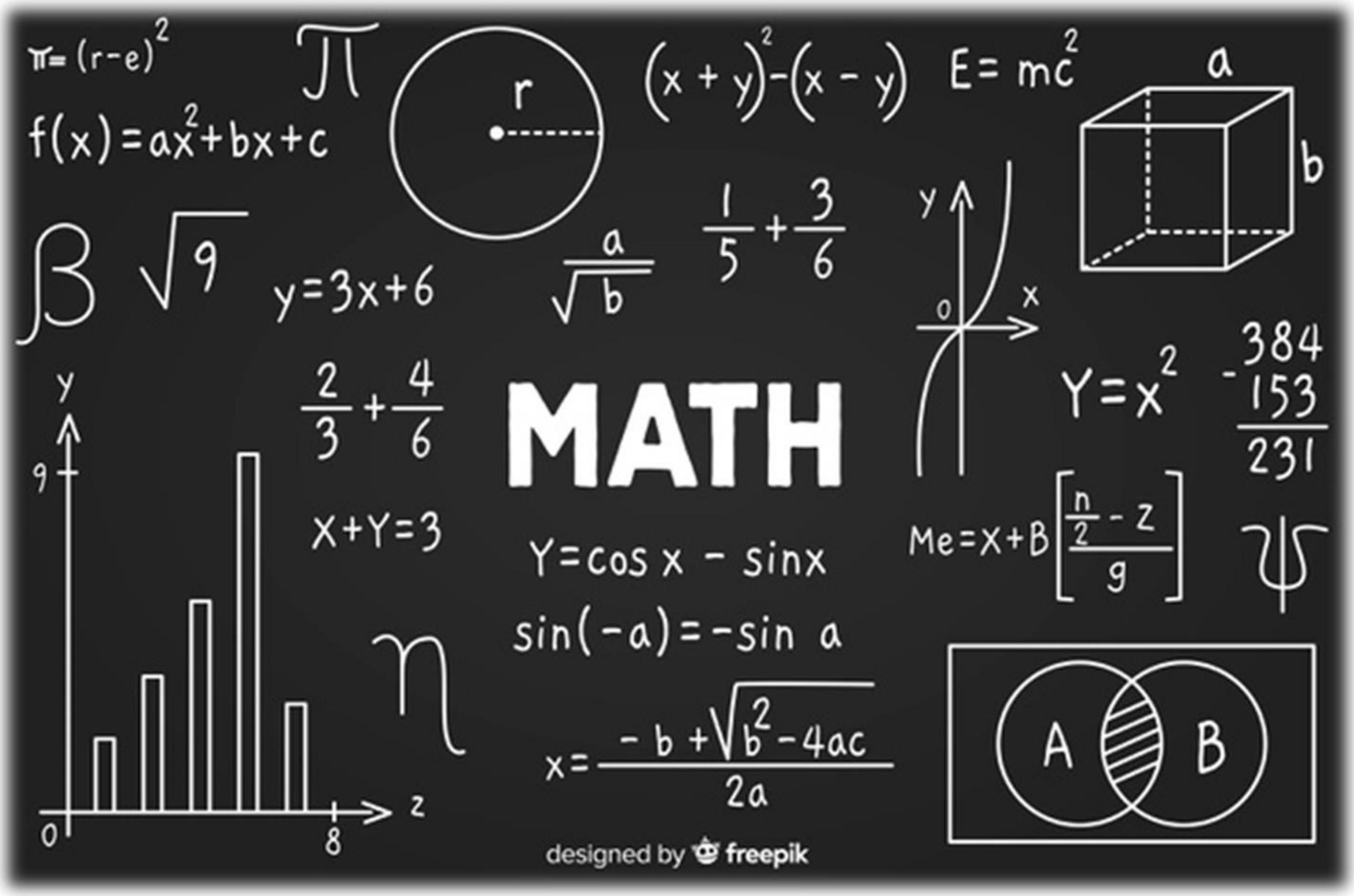
Maths
Projets



101Pong :

The goal of this project is to work on a 3D version of this game (or of the Breakout game. . .). Only one paddle will be considered, located in the (Oxy) plane (which is defined by the equation z = 0).

USAGE

Terminal

```
~/B-MAT-100> ./101pong -h
USAGE
    ./101pong x0 y0 z0 x1 y1 z1 n

DESCRIPTION
    x0   ball abscissa at time t - 1
    y0   ball ordinate at time t - 1
    z0   ball altitude at time t - 1
    x1   ball abscissa at time t
    y1   ball ordinate at time t
    z1   ball altitude at time t
    n    time shift (greater than or equal to zero, integer)
```

Examples:

Terminal

```
~/B-MAT-100> ./101pong 1 3 5 7 9 -2 4
The velocity vector of the ball is:
(6.00, 6.00, -7.00)
At time t + 4, ball coordinates will be:
(31.00, 33.00, -30.00)
The ball won't reach the paddle.
```

Terminal

```
~/B-MAT-100> ./101pong 1.1 3 5 -7 9 2 4
The velocity vector of the ball is:
(-8.10, 6.00, -3.00)
At time t + 4, ball coordinates will be:
(-39.40, 33.00, -10.00)
The incidence angle is:
16.57 degrees
```

102architect:

The goal is to simplify the process of drawing the plan, and to integrate various features such as scale management, changing the point of view, moving doors and windows along walls. O being the origin of both axis, here are the transformations to be implemented:

- Translation.
- Scaling.
- Rotation centered at O.
- Reflection over any axis that passes through O.
- Any combination of the previous transformations.

USAGE

```
~/B-MAT-100> ./102architect -h
USAGE
    ./102architect x y transfo1 arg11 [arg12] [transfo2 arg12 [arg22]] ...

DESCRIPTION
    x   abscissa of the original point
    y   ordinate of the original point

    transfo arg1 [arg2]
    -t i j  translation along vector (i, j)
    -z m n  scaling by factors m (x-axis) and n (y-axis)
    -r d    rotation centered in O by a d degree angle
    -s d    reflection over the axis passing through O with an inclination
            angle of d degrees
```

EXAMPLES

```
~/B-MAT-100> ./102architect 5 0 -t -1 1
Translation along vector (-1, 1)
1.00    0.00    -1.00
0.00    1.00    1.00
0.00    0.00    1.00
(5.00, 0.00) => (4.00, 1.00)
```

```
~/B-MAT-100> ./102architect 2 2 -z -1 1
Scaling by factors -1 and 1
-1.00   0.00    0.00
0.00    1.00    0.00
0.00    0.00    1.00
(2.00, 2.00) => (-2.00, 2.00)
```

```
~/B-MAT-100> ./102architect 1 0 -r 90
Rotation by a 90 degree angle
0.00    -1.00   0.00
1.00    0.00    0.00
0.00    0.00    1.00
(1.00, 0.00) => (0.00, 1.00)
```

```
~/B-MAT-100> ./102architect 3 -1 -s 270
Reflection over an axis with an inclination angle of 270 degrees
-1.00   0.00    0.00
0.00    1.00    0.00
0.00    0.00    1.00
(3.00, -1.00) => (-3.00, -1.00)
```

```
~/B-MAT-100> ./102architect 1 2 -t 2 3 -z 1 -2 -r 45 -s 30
Translation along vector (2, 3)
Scaling by factors 1 and -2
Rotation by a 45 degree angle
Reflection over an axis with an inclination angle of 30 degrees
0.97    -0.52   0.38
0.26    1.93    6.31
0.00    0.00    1.00
(1.00, 2.00) => (0.31, 10.44)
```

103cipher:

The goal of the project is to encrypt a message (such as the Hill cipher).

USAGE

```
~/B-MAT-100> ./103cipher -h
USAGE
    ./103cipher message key flag

DESCRIPTION
    message     a message, made of ASCII characters
    key         the encryption key, made of ASCII characters
    flag        0 for the message to be encrypted, 1 to be decrypted
```

EXAMPLES

```
~/B-MAT-100> ./103cipher "Just because I don't care doesn't mean I don't
understand." "Homer S" 0
Key matrix:
72      111     109
101     114     32
83      0       0

Encrypted message:
26690 21552 11810 19718 16524 13668 25322 22497 14177 28422 26097 16433 12333
11874 5824 27541 23754 14452 17180 17553 7963 26387 22047 13895 18804 14859 12033
27738 23835 15331 21487 16656 13238 21696 15978 6976 20750 23307 14093 16788 11751
8981 22339 24861 15619 21295 16524 13668 26403 23610 15190 29451 25764 16106 26394
23307 14093 3312 5106 5014
```

104intersection:

The goal of the project is to find intersection points coordinates between straight lines.

O being the origin of the coordinate system, and X, Y and Z the axis, the surfaces that must be handled in this project are:

- O-centered spheres.
- Cylinders of revolution around Z axis.
- Cones of revolution around Z axis whose apex is O.

USAGE

```
~/B-MAT-100> ./104intersection -h
USAGE
    ./104intersection opt xp yp zp xv yv zv p

DESCRIPTION
    opt              surface option: 1 for a sphere, 2 for a cylinder, 3 for a cone
    (xp, yp, zp)     coordinates of a point by which the light ray passes through
    (xv, yv, zv)     coordinates of a vector parallel to the light ray
    p                parameter: radius of the sphere, radius of the cylinder, or
                     angle formed by the cone and the Z-axis
```

EXAMPLES

```
~/B-MAT-100> ./104intersection 1 0 0 2 1 1 0 1
Sphere of radius 1
Line passing through the point (0, 0, 2) and parallel to the vector (1, 1, 0)
No intersection point.
```

```
~/B-MAT-100> ./104intersection 1 4 0 3 0 0 -2 4
Sphere of radius 4
Line passing through the point (4, 0, 3) and parallel to the vector (0, 0, -2)
1 intersection point:
(4.000, 0.000, 0.000)
```

```
~/B-MAT-100> ./104intersection 2 0 0 2 1 1 0 1
Cylinder of radius 1
Line passing through the point (0, 0, 2) and parallel to the vector (1, 1, 0)
2 intersection points:
(0.707, 0.707, 2.000)
(-0.707, -0.707, 2.000)
```

```
~/B-MAT-100> ./104intersection 3 -1 -1 -1 1 1 5 30
Cone with a 30 degree angle
Line passing through the point (-1, -1, -1) and parallel to the vector (1, 1, 5)
2 intersection points:
(-1.568, -1.568, -3.842)
(-0.537, -0.537, 1.315)
```

```
~/B-MAT-100> ./104intersection 2 1 0 0 0 0 1 1
Cylinder of radius 1
Line passing through the point (1, 0, 0) and parallel to the vector (0, 0, 1)
There is an infinite number of intersection points.
```

105torus:

The objective of this project is to solve a 4th degree equation: a4x4 +a3x3 +a2x2 +a1x1 +a0 = 0.

A direct resolution method does exist (Ferrari's method), but does not generalize to higher degrees. Thus, we will rather compare 3 iterative algorithms:

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

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 method
                    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)
```

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
```

106bombyx:

The goal of this project is to call at the evolution of some animal species. Butterflies for instance the bombyx.

USAGE

Terminal

```
~/B-MAT-200> ./105torus -h
USAGE
    ./106bombyx n [k | i0 i1]

DESCRIPTION
    n        number of first generation individuals
    k        growth rate from 1 to 4
    i0       initial generation (included)
    i1       final generation (included)
```

EXAMPLES

Your program output has to be strictly identical to the one below

Terminal

```
~/B-MAT-200> ./106bombyx 10 3.3 > data
~/B-MAT-200> head data
1 10.00
2 32.67
3 104.29
4 308.26
5 703.68
6 688.10
7 708.24
8 681.89
9 715.82
10 671.29
~/B-MAT-200> tail data
91 823.60
92 479.43
93 823.60
94 479.43
95 823.60
96 479.43
97 823.60
98 479.43
99 823.60
100 479.43
```

Terminal

```
~/B-MAT-200> cat drawer.gnu
set terminal dumb
set nokey
plot "data"
```

Terminal

```
~/B-MAT-200> ./106bombyx 10 10000 10010 > data
~/B-MAT-200> head -n 30 data
1.00 0.10
1.00 0.10
1.00 0.10
1.00 0.10
1.00 0.10
1.00 0.10
1.00 0.10
1.00 0.10
1.00 0.10
1.00 0.10
1.00 0.10
1.01 9.90
1.01 9.90
1.01 9.90
1.01 9.90
1.01 9.90
1.01 9.90
1.01 9.90
1.01 9.90
1.01 9.90
1.01 9.90
1.01 9.90
1.02 19.61
1.02 19.61
1.02 19.61
1.02 19.61
1.02 19.61
1.02 19.61
1.02 19.61
1.02 19.61
```

107transfer:

The goal of the project is to develop a program to optimize the transfer function computations.

A transfer function is defined by two strings (one for the numerator, one for the denominator), composed by the polynomial coefficients split by the '*' sign.

For instance, "1*4*2*6*0*8" stands for 8x 5 + 6x 3 + 2x 2 + 4x + 1.

USAGE

Terminal

```
~/B-MAT-200> ./107transfer -h
USAGE
    ./107transfer [num den]*

DESCRIPTION
    num     polynomial numerator defined by its coefficients
    den     polynomial denominator defined by its coefficients
```

EXAMPLES

Terminal

```
~/B-MAT-200> ./107transfer "0*1*2*3*4" "1" > file
~/B-MAT-200> head -n 12 file
0.000 -> 0.00000
0.001 -> 0.00100
0.002 -> 0.00201
0.003 -> 0.00302
0.004 -> 0.00403
0.005 -> 0.00505
0.006 -> 0.00607
0.007 -> 0.00710
0.008 -> 0.00813
0.009 -> 0.00916
0.010 -> 0.01020
0.011 -> 0.01125
~/B-MAT-200> tail file
0.991 -> 9.73282
0.992 -> 9.76223
0.993 -> 9.79171
0.994 -> 9.82126
0.995 -> 9.85087
0.996 -> 9.88056
0.997 -> 9.91031
0.998 -> 9.94014
0.999 -> 9.97003
1.000 -> 10.00000
```

108trigo:

The goal of the project is to resolve trigonometric and hyperbolic functions. By given a matrix and the name of a function, the program applies the latter to the former, and print the result.

USAGE

Terminal

```
~/B-MAT-200> ./108trigo -h
USAGE
    ./108trigo fun a0 a1 a2 ...

DESCRIPTION
    fun      function to be applied, among at least "EXP", "COS", "SIN", "COSH"
             and "SINH"
    ai       coeficients of the matrix
```

EXAMPLES

Terminal

```
~/B-MAT-200> ./108trigo COS 4 5 9 3 3 5 0 1 9
0.70    -0.43   -1.94
-0.16   0.67    -1.23
-0.06   -0.15   0.07
```

Terminal

```
~/B-MAT-200> ./108trigo EXP 1 2 3 4
51.97   74.74
112.10  164.07
```

Terminal

```
~/B-MAT-200> ./108trigo SINH 1 0 2 0
1.18    0.00
2.35    0.00
```

109titration:

The goal of the project is to the code derivative method, which consists in calculating the derivative of the curve; the equivalence point matches with the maximum of this derivative.

```
Terminal
~/B-MAT-200> ./109titration values.csv
Derivative:
2.0 ml -> 1.00
3.0 ml -> 0.73
5.0 ml -> 0.20
6.0 ml -> 0.80
7.0 ml -> 1.53
7.5 ml -> 2.00
8.0 ml -> 2.27
9.0 ml -> 1.61
12.0 ml -> 0.22
14.0 ml -> 0.07
16.0 ml -> 0.06

Equivalence point at 8.0 ml

Second derivative:
3.0 ml -> -0.27
5.0 ml -> 0.31
6.0 ml -> 0.67
7.0 ml -> 0.87
7.5 ml -> 0.73
8.0 ml -> 0.14
9.0 ml -> -0.61
12.0 ml -> -0.23
14.0 ml -> -0.04

Second derivative estimated:
7.5 ml -> 0.73
7.6 ml -> 0.61
7.7 ml -> 0.49
7.8 ml -> 0.38
7.9 ml -> 0.26
8.0 ml -> 0.14
8.1 ml -> 0.06
8.2 ml -> -0.01
8.3 ml -> -0.09
8.4 ml -> -0.16
8.5 ml -> -0.24
8.6 ml -> -0.31
8.7 ml -> -0.39
8.8 ml -> -0.46
8.9 ml -> -0.53
9.0 ml -> -0.61

Equivalence point at 8.2 ml
```

110borwein:

The goal of the project is to resolve the Borwein brothers' integrals function.

USAGE

Terminal

```
~/B-MAT-200> ./110borwein -h
USAGE
    ./110borwein n

DESCRIPTION
    n    constant defining the integral to be computed
```

EXAMPLES

Terminal

```
~/B-MAT-200> ./110borwein 0
Midpoint:
I0 = 1.5707651076
diff = 0.0000312192

Trapezoidal:
I0 = 1.5707660806
diff = 0.0000302462

Simpson:
I0 = 1.5707654320
diff = 0.0000308948
```