Introduction to Python and pyPLaSM

Computational Visual Design Laboratory
(https://github.com/cvlab) “Roma Tre” University, Italy

Computational Graphics – Lecture 10 – March 19, 2013

Examples of MAP

Examples of MAP

Mapping a function over the vertices of a domain

function of point returning a list of coordinate functions

```
def circle(p):
    alpha = p[0]
    return [COS(alpha), SIN(alpha)]
```

primitive constructor `INTERVALS(x)(n)` of a simplicial decomposition of the $[0, x]$ interval into n subintervals

```
obj = MAP(circle)(INTERVALS(2*PI)(32))
VIEW(obj)
```

Mapping a function over the vertices of a domain

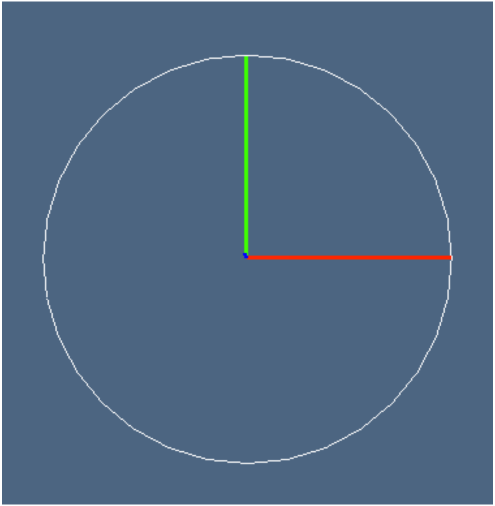


Figure : Unit circle centered in the origin

Mapping an higher-level function

`circle(r)(p)` is now parameterized by the r value

```
def circle(r):
    def circle0(p):
        alpha = p[0]
        return [r*COS(alpha), r*SIN(alpha)]
    return circle0

obj = MAP(circle(2))(INTERVALS(2*PI)(32))
VIEW(obj)
```

Mapping an higher-level function

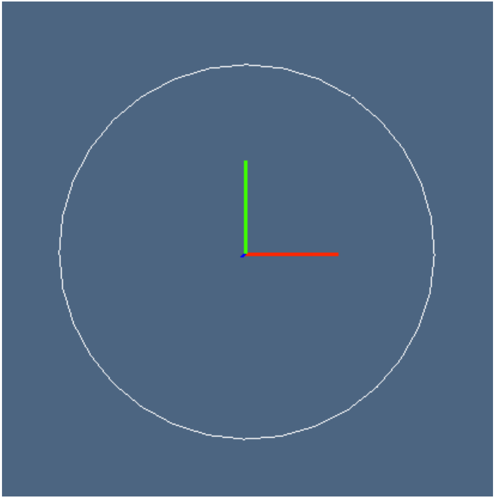


Figure : Circle of radius $r = 2$ centered in the origin

Mapping an higher-level function

`dom(n)` is now parameterized by the *n* values

```
def dom(n):
    return INTERVALS(2*PI*n)(24*n)
```

`spiral(pitch,n)(p)` is now parameterized by the *pitch*, *n* values

```
def spiral(pitch,n):
    def spiral0(p):
        alpha = p[0]
        return [COS(alpha), SIN(alpha), alpha*pitch*n/(2*PI*n)]
    return spiral0

obj = MAP(spiral(0.2,5))(dom(5))
VIEW(obj)
```

Mapping an higher-level function

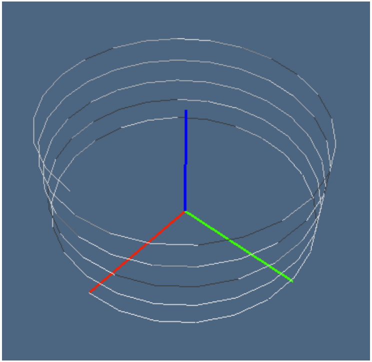


Figure : Spiral curve in 3D (3 coordinate functions)

Mapping a 2D domain

The domain dom2D $= [0, 2\pi] \times [0, 1]$ is the Cartesian product of two 1D intervals

```
dom2D = PROD([INTERVALS(2*PI)(24), INTERVALS(1)(1)])
VIEW(dom2D)
```

It is useful to look at its 1-skeleton

```
VIEW(SKELETON(1)(dom2D))
```

Mapping a 2D domain

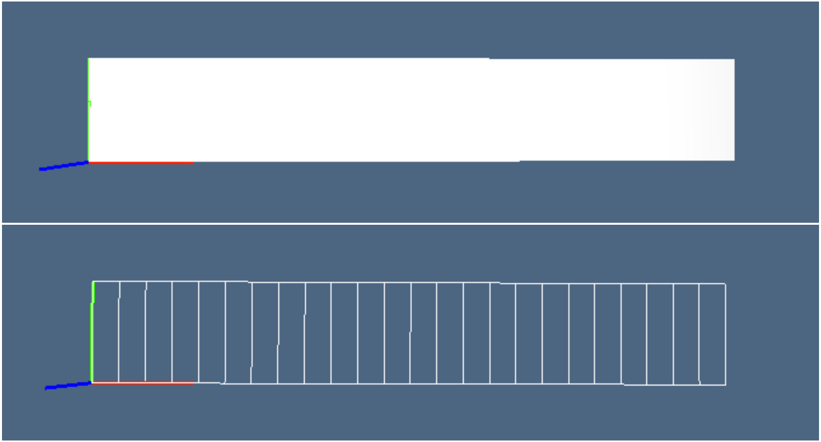


Figure : $\texttt{dom2D} = [0, 2\pi] \times [0, 1]$ and its 1-skeleton

2D/3D spiral surface/solid

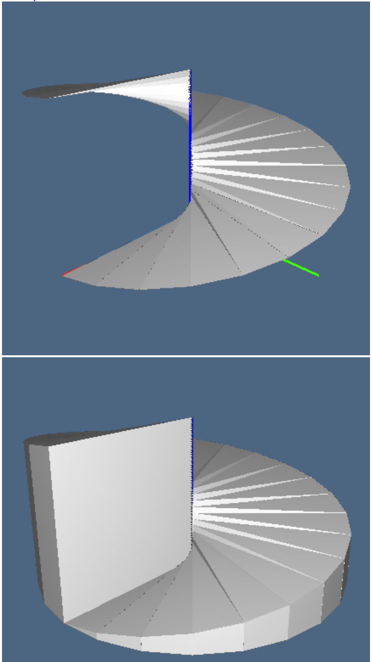
p$\in \mathbb{E}^2$ contains two coordinates

```
def spiral(p):
    alpha,r = p
    return [r*COS(alpha), r*SIN(alpha), alpha/(2*PI)]

obj = MAP(spiral)(dom2D)
VIEW(obj)
```

p$\in \mathbb{E}^3$ contains two coordinates

```
dom1D = INTERVALS(1)(1)
dom3D = INSR(PROD)([INTERVALS(2*PI)(24), dom1D, dom1D])
def spiral(p):
    alpha,r,h = p
    return [r*COS(alpha), r*SIN(alpha), h*alpha/(2*PI)]

obj = MAP(spiral)(dom3D)
VIEW(obj)
```

D/3D spiral surface/solid



3D solid spiraloid

Two surface functions $\mathbb{E}^3 \to \mathbb{E}^2$ are given

```
dom3D = INSR(PROD)([INTERVALS(2*PI)(24), dom1D, dom1D])
def spiral1(p):
    alpha,r,h = p
    return [r*COS(alpha), r*SIN(alpha), alpha/(2*PI)]
def spiral2(p):
    alpha,r,h = p
    return [r*COS(alpha), r*SIN(alpha), alpha/(2*PI)+0.1]
```

The mapping function is a transfinite interpolation of two surface functions

```
obj = STRUCT([MAP(spiral1)(dom3D), MAP(spiral2)(dom3D)])
VIEW(obj)

obj = MAP(BEZIER(S3)([spiral1,spiral2]))(dom3D)
VIEW(obj)
```

3D solid spiraloid