

Chapter 1

Didactic applications

This chapter is an aid to study DR. GEO from didactic examples. Although in the previous chapters several examples were presented, here the approach is little more concrete, while still very original. This chapter was filled with contributions from various cultural sources.

1.1 Perimeter and area

One possible didactic use of DR. GEO is through its Smalltalk script system to resolve classic geometry exercises.

As an example, we will show the solution of the following classic problem – involving the Pythagorean theorem:

A right trapezoid ABCD where the bases and height are known. Calculate its the perimeter and area.

Solution:

First we construct the DR. GEO sketch as follow:

aire = 9.00

périmètre = 12.61

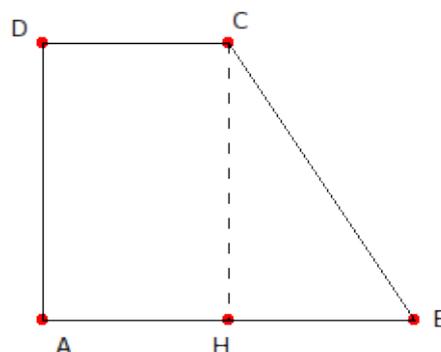


Figure 1.1: Right trapezoid

The sketch contains the data to use to resolve the problem. First we can answer to the question of the area, to do so we write a Smalltalk script with arguments the two bases and the height:

```

areaTrapezoidBase1: b1 base2: b2 height: h
"Calculate the area of a right trapezoid given
its bases and one height"
  ^ h length * (b1 length + b2 length) / 2

```

To calculate the perimeter, we write a script where we calculate the length of BC with the Pythagorean theorem:

```

perimeterTrapezoidBase1: b1 base2: b2 height: h
"Calculate the area of a right trapezoid given
its two bases and one height"
| hb bc |
  hb := (b2 length - b1 length) abs.
  bc := (hb squared + h length squared) sqrt.
  ^ b1 length + b2 length + h length + bc

```

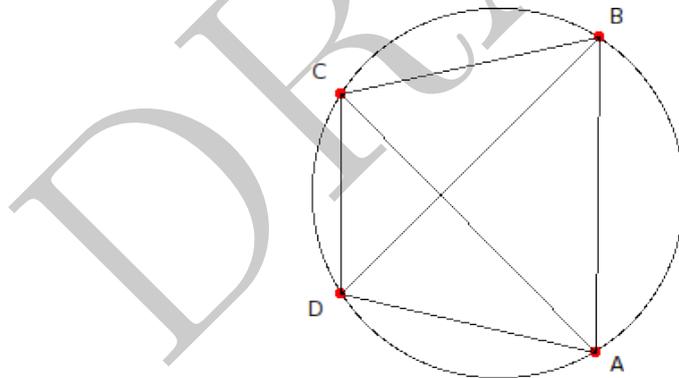
It is not difficult, if you follow the same model, to develop other similar examples.

1.2 Theorem and conjectures

With Smalltalk script one can resolve exercise but can also understand more deeply theorem and verify conjectures. In this section we start to analyse the Tolomeo theorem:

Given an inscribed quadrilateral, the sum of the products of its opposite sides is equal to the product of its diagonals.

We construct the sketch as below where we implemented two scripts to calculate respectively the sum of the products of its opposite sides and the product of its diagonals.



$$(AD * BC) + (BC * AD) = 31.92$$

$$AC * BD = 31.92$$

Figure 1.2: Theorem of Tolomeo: inscribed quadrilateral

The first script:

```

tolomeoSommeS1: ab s2: bc s3: cd s4: ad
"Select the four consecutive sides of the quadrilateral"
  ^ (ad length * bc length) + (ab length * cd length)

```

The second script:

```

tolomeoProduitD1: ac d2: bd
"Select the two diagonals of the quadrilateral"
^ ac length * bd length

```

As we can see, the returned values by the scripts, as stated by the theorem of Tolomeo, are equals¹ When we dynamically modify the sketch, the script values are always equals, except in the following situation where the quadrilateral is not convex:

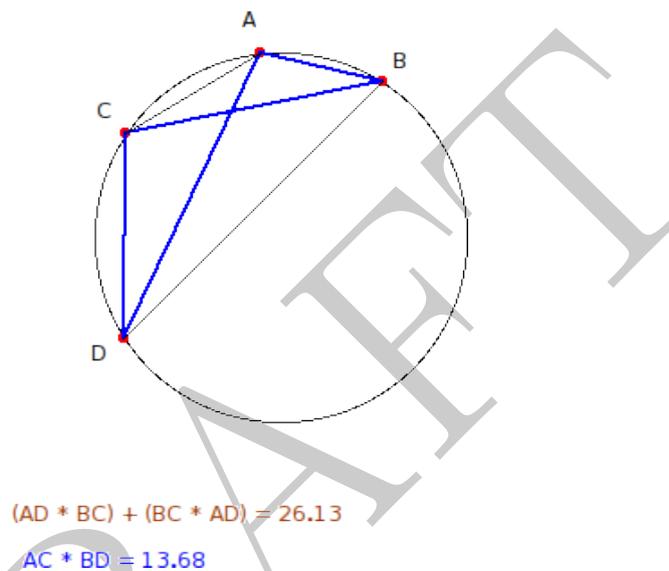


Figure 1.3: Theorem of Tolomeo: non convex inscribed quadrilateral

In this case the theorem is not true and the previous text of the theorem is inaccurate, it should be reformulated as follow:

Given an inscribed CONVEX quadrilateral, the sum of the products of its opposite sides is equal to the product of its diagonals.

Now a additional conjecture can be stated: is the theorem still valid when the convex quadrilateral is **non inscribed**?

With DR. GEO we verify this conjecture is false with the following counterexample bellow. To build this counterexample, we have just detached the point B from the circle by dragging it with the touch SHIFT pressed at the same time.

The reader will easily use DR. GEO to construct additional didactic examples, may be more famous, relatively to the Pythagorean and Euclidean theorems.

¹It is only a numerical verification.

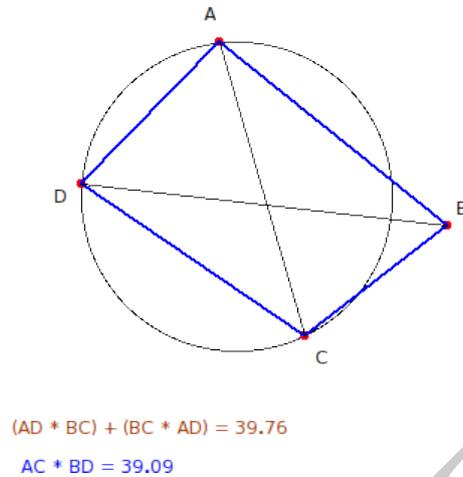


Figure 1.4: Counterexample of the conjecture

1.3 Irrational number

A classic construction, relative to irrational number, known as the Teodoro spiral gives the geometric construction of the integers square root. It is done with isosceles right triangle.

Let's start with the triangle OAB where $OA = 1$:

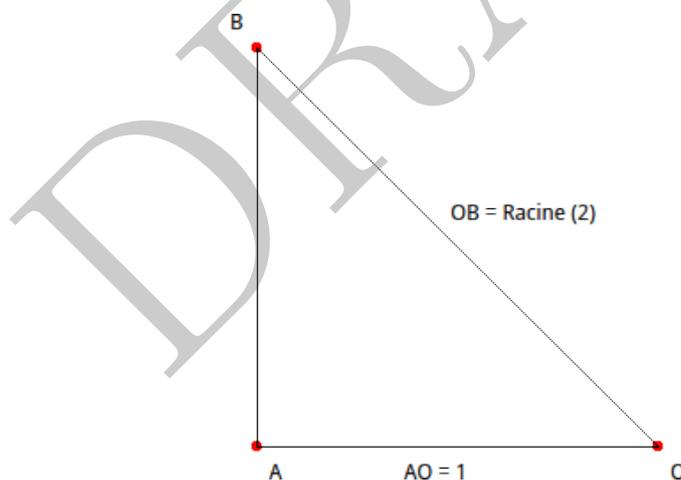


Figure 1.5: Construction of the square root of 2

With the Pythagorean theorem we have OB equal to the square root of 2. Now, with the sketch, we construct another right triangle in B with the sides OB and BC so that $BC = 1$.

Still with the Pythagorean theorem, it is clear the hypotenuse OC of OBC has a length equal to the square root of 3. The process can be repeated at the infinite to get the square root of all integers.

The iterative nature of this construction fully matches Smalltalk sketch.

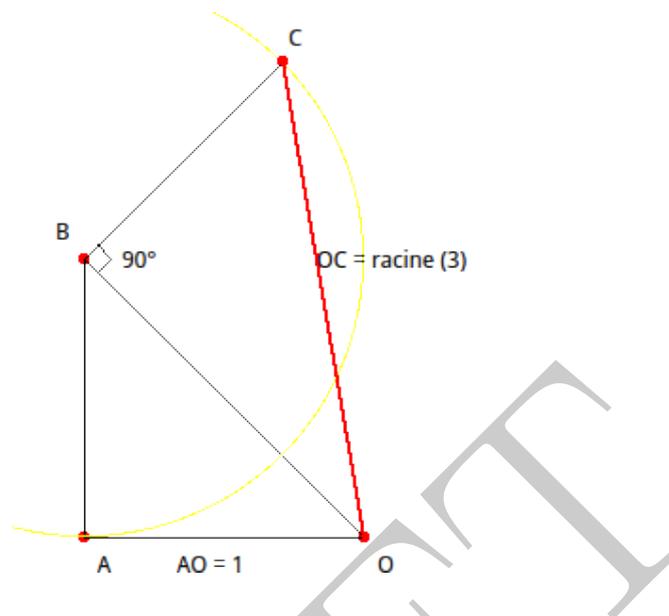


Figure 1.6: Construction of the square root of 3

Let's consider the following Smalltalk sketch source code:

```
| sketch triangle |
sketch := DrGeoCanvas new fullscreen.
triangle := [:p1 :p2 :p3 :n |
  ' |s1 s2 s3 perp circle p4 |
  s1 := sketch segment: p1 to: p2.
  s2 := (sketch segment: p2 to: p3) color: Color red.
  s3 := sketch segment: p3 to: p1.
  perp := (sketch perpendicular: s3 at: p3) hide.
  circle := (sketch circleCenter: p3 to: p2) hide.
  p4 := (sketch altIntersectionOf: circle and: perp) hide.
  n > 0 ifTrue: [triangle value: p1 value: p3 value: p4 value: n -1]].

triangle
value: 0@0
value: -1@0
value: -1@1
value: 50
```

The triangle at the beginning is defined with its coordinates summits. The source code is a direct transcription of the logic of the construction plus the integration of its iterative nature with a recursive loop. We use several time the `#hide` message to mask intermediate constructions, to not overload the user perception with underground constructions. Once executed, DR. GEO gives the following sketch.

The hypotenuse length of each triangle is the square root of integer in the interval $[2 ; 52]$.

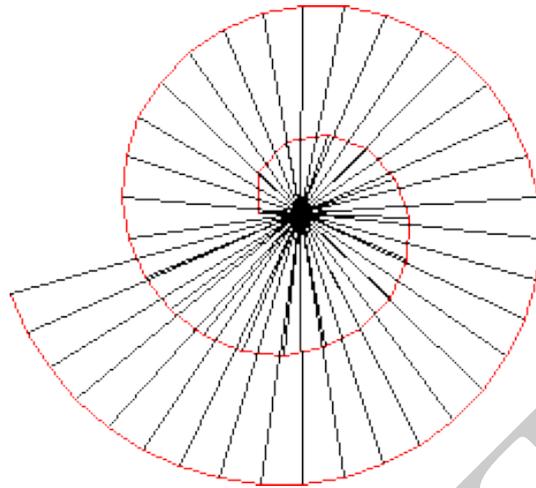


Figure 1.7: Teodoro spiral

The same spiral with shown intermediate items exposes how difficult it will be to *hand construct* such a sketch:

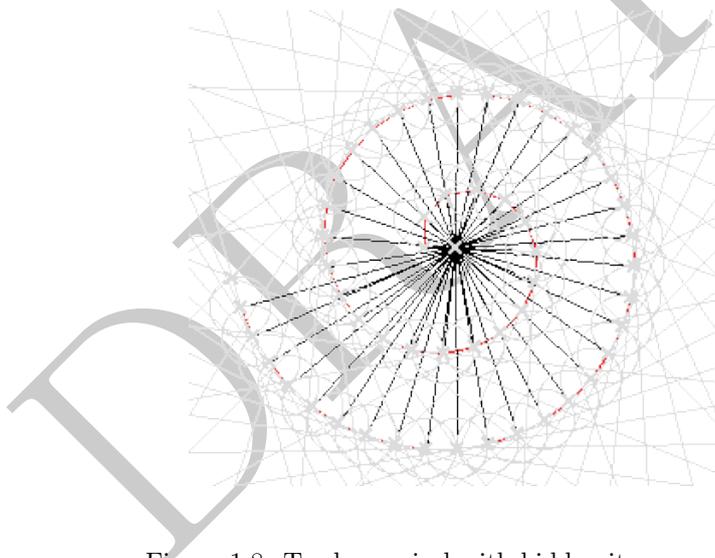


Figure 1.8: Teodoro spiral with hidden item revealed

As a bonus, we propose you bellow an animated version of the previous Smalltalk sketch. To do so we send to the canvas the messages `#do:` and `#update`. It is documented in the chapter ??, p. ??. Note the use of the class `Delay` to slow down the construction:

```
| sketch triangle delay|
sketch := DrGeoCanvas new fullscreen.
triangle := [:p1 :p2 :p3 :n |
  |s1 s2 s3 perp circle p4 |
  s1 := sketch segment: p1 to: p2.
  s2 := (sketch segment: p2 to: p3) color: Color red.
  s3 := sketch segment: p3 to: p1.
  perp := sketch perpendicular: s3 at: p3.
  circle := sketch circleCenter: p3 to: p2.
  p4 := sketch altIntersectionOf: circle and: perp.
```

```

sketch update.
(Delay forMilliseconds: 200) wait.
perp hide. circle hide. p4 hide.
n > 0 ifTrue: [triangle value: p1 value: p3 value: p4 value: n -1]].

sketch do: [triangle value: 0@0 value: -1@0 value: -1@1 value: 50]

```

1.4 Baravelle spiral

As we saw previously, with a Smalltalk sketch it is easy to construct intuitively and simply sketches *to visualise* recursive situations in programming – or cyclic.

We can go one step further by modifying the previous Smalltalk code – used to construct irrational number – to get a famous sketch of the mathematics literature: the Baravelle spiral.

The code to construct this spiral is as follow:

```

| sketch triangle |
sketch := DrGeoCanvas new fullscreen.
triangle := [:p1 :p2 :p3 :n | |s1 s2 s3 m perp cercle p4 |
  s1 := sketch segment: p1 to: p2.
  s2 := sketch segment: p2 to: p3.
  s3 := sketch segment: p3 to: p1.
  m := (sketch middleOf: p1 and: p3) hide.
  perp := (sketch perpendicular: s3 at: p3) hide.
  cercle := (sketch circleCenter: p3 radius: (sketch distance: m to: p3) hide) hide.
  p4 := (sketch altIntersectionOf: cercle and: perp) hide.
  n > 0 ifTrue: [triangle value: m value: p3 value: p4 value: n -1]].

triangle
  value: (sketch point: 0@5)
  value: (sketch point: 5@5)
  value: (sketch point: 5@0)
  value: 9.

triangle
  value: (sketch point: 0@ -5)
  value: (sketch point: -5@ -5)
  value: (sketch point: -5@0)
  value: 9

```

With this sketch and the corresponding Smalltalk code we clearly perceive the iterative nature of the construction mechanism. An interesting problem for the reader is to establish when the two branches of the spiral converge.

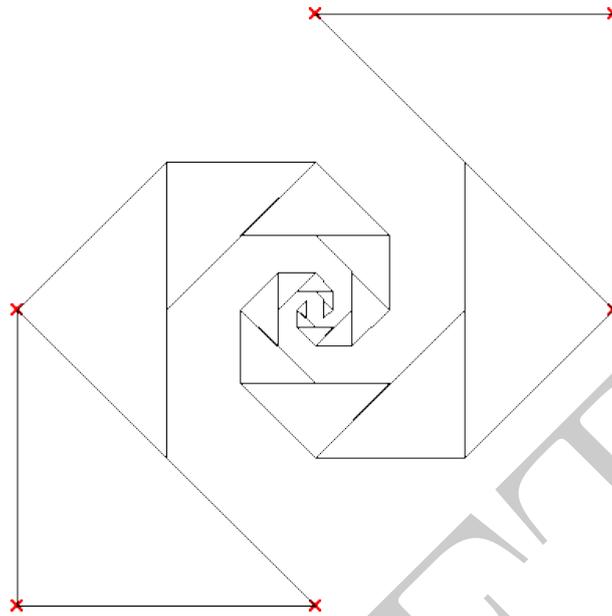


Figure 1.9: The Baravelle spiral after executing the Smalltalk code

1.5 Catena di Pappo

Another classic use of Smalltalk programmed sketch is based on its numeric ability to reproduce a geometric sketch knowing its analytic characteristics.

The example we propose is the famous “Catena di Pappo”.

The successive circles’ centres and radius are analytically known, it is therefore easy to reproduce this sketch by programming.

```
| sketch circle a o m |
sketch := DrGeoCanvas new fullscreen.
circle := [:n | |r c p |
  r := (sketch freeValue: 15 / (n squared + 6)) hide.
  c := sketch point: (15 / (n squared + 6) * 5) @ (15 / (n squared + 6) * n * 2).
  c small; round.
  p := sketch circleCenter: c radius: r.
  n > 0 ifTrue: [circle value: n - 1]].

circle value: 10 .
a := (sketch point: 5@0) name: 'A'.
o := (sketch point: 0@0) name: 'O'.
m := sketch
  middleOf: o
  and: ((sketch point: 15@0) name: 'B').
m name: 'M'.
sketch
  circleCenter: m to: o;
  circleCenter: a to: o;
  line: a to: o.
```

The source code is relatively intuitive and it does not deserve any comment.

A non trivial exercise for the reader consists to determine its ruler-compass construction.

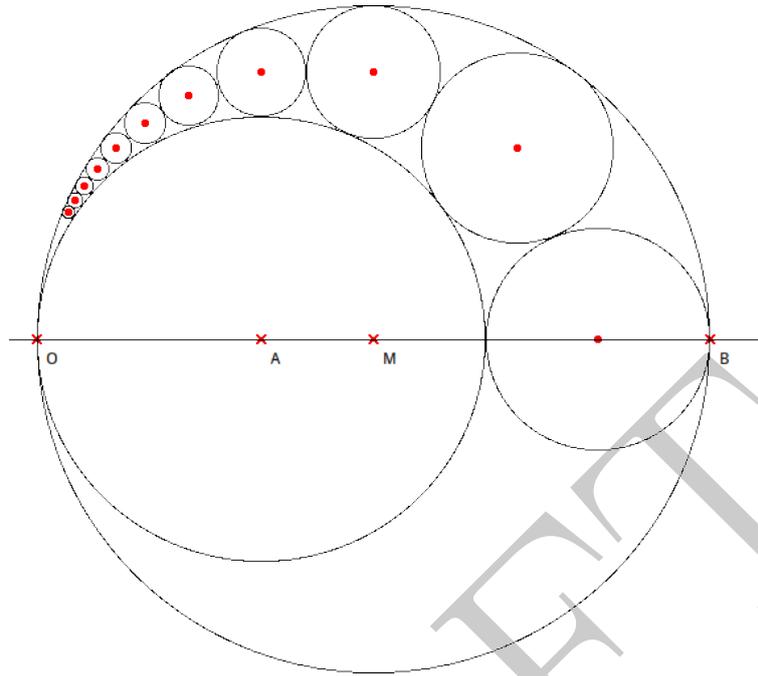


Figure 1.10: Catena di Pappo

1.6 π calculus

The approximation of π played an important role in the history of mathematics. The methods are numerous and contain very often improvements from one version to another one. We propose an approach of the problem very simplified we call – although it is not precisely exact – **exhaustion method**. This approach has however the advantage to expose the essence of the methodology.

We start by the construction of an inscribed regular hexagon.

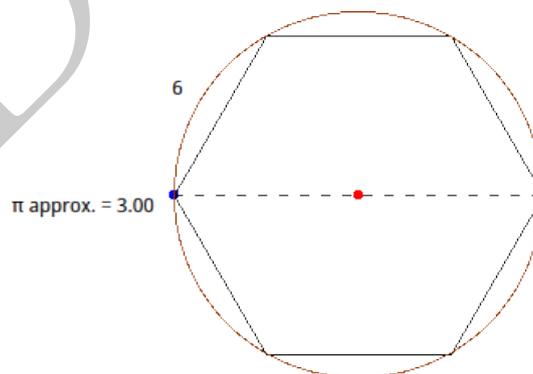


Figure 1.11: Inscribed regular hexagon

The idea of the exhaustion method consists in a first step to approximate the length of the circle with the length of the P_0 hexagon; then to calculate an approximation of π as the quotient of P_0 perimeter by the circle diameter. Clearly the obtained approximation of π is 3.

In a second step, we construct an inscribed dodecagon². We calculate the P_1 perimeter, then a π approximation as the quotient of P_1 perimeter by the circle diameter.

A small script is written to calculate this quotient; its arguments are the polygon and the circle diameter:

```
approxPIpolygon: poly diameter: d
"PI approximation given an inscribed regular polygon.
Arguments: the polygon and the circle diameter"
  ^ poly length / d length
```

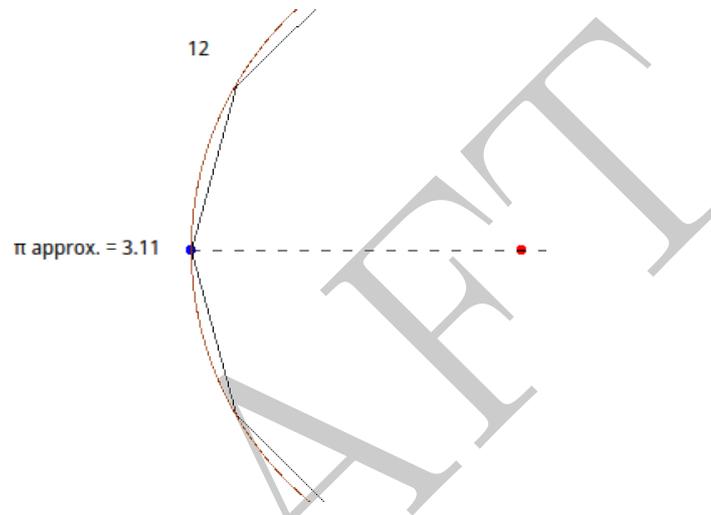


Figure 1.12: π approximation

When increasing the polygon submits we improve the approximation.

To display both the approximation of π and its precision, we can modify the script to display both:

```
approxPIpolygon: poly diameter: d
"PI approximation given an inscribed regular polygon.
Arguments: the polygon and the circle diameter"
  ^ Array
    with: poly length / d length
    with: (poly length / d length - Float pi) abs
```

²To construct the previous regular hexagon, we used the DR. GEO tool to build regular polygon with a number of submit of 6. To construct the dodecagon, we just need to change this free value item to 12.