

Curs 06

(plan de curs)

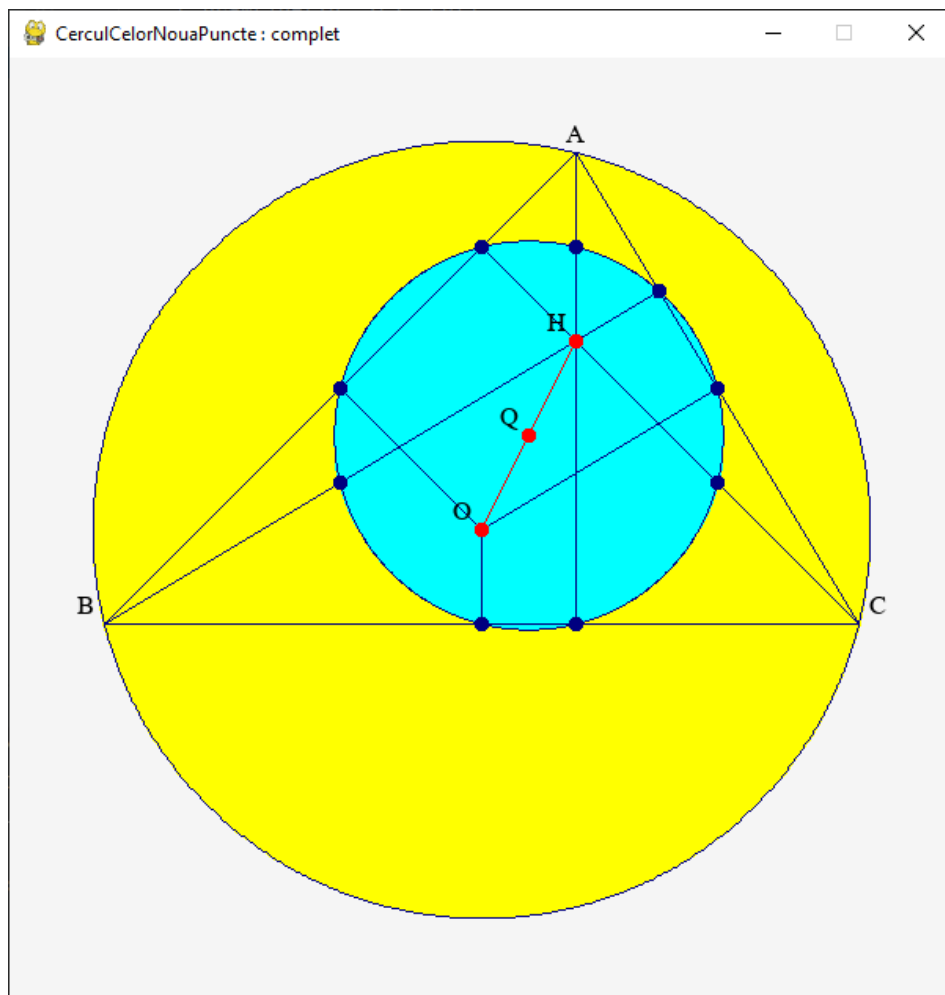
1. **Proiecția lui a pe dreapta bc .** Fie $\omega = \frac{c-b}{|c-b|}$ versorul segmentului bc și fie a' proiecția lui a pe bc . Avem $a' = b + \lambda \omega$ unde coeficientul real λ se determină din condiția de perpendicularitate

$$0 = \langle a - a', \omega \rangle = \langle a - b, \omega \rangle - \langle \lambda \omega, \omega \rangle = \langle a - b, \omega \rangle - \lambda$$

de unde rezultă $\lambda = \langle a - b, \omega \rangle$.

2. **Cercul celor nouă puncte:** https://en.wikipedia.org/wiki/Nine-point_circle

3. **Integrala complexă.**



```

import ComplexPygame as C
import Color
import math

def CerculCelorNouaPuncte():
    def ps(u, v): # produsul scalar
        return (u * v.conjugate()).real

    def pro1pe23(z1, z2, z3): # proiectia lui z1 pe z2z3
        omega = (z3 - z2) / C.rho(z3 - z2)
        return z2 + ps(z1 - z2, omega) * omega

    def unCercQR(q, r, N):
        fi = 2 * math.pi / N
        return [q + C.fromRhoTheta(r, k * fi) for k in range(N)]

    def drawQR(q, r, col):
        C.drawNgon(unCercQR(q, r, 1000), col)

    def fillQR(q, r, col):
        C.fillNgon(unCercQR(q, r, 100), col)

    def bazaApex(zB, zC, A):
        # returneaza apexul zA al triunghiului isoscel cu baza zBzC
        omegaA = C.fromRhoTheta(1, A)
        return (zC - omegaA * zB) / (1 - omegaA)

C.setXminXmaxYminYmax(0, 10, 0, 10)
zA, zB, zC = 6 + 9j, 1 + 4j, 9 + 4j

zA1 = pro1pe23(zA, zB, zC)
zB1 = pro1pe23(zB, zC, zA)
zC1 = pro1pe23(zC, zB, zA)

zA2, zB2, zC2 = (zB + zC) / 2, (zC + zA) / 2, (zA + zB) / 2

A = abs(C.theta((zC - zA) / (zB - zA)))
z0 = bazaApex(zB, zC, 2 * A)
zH = zA + zB + zC - 2 * z0 # zQ = (zA + zB + zC - z0) / 2
zQ = (zH + z0) / 2

zA3, zB3, zC3 = (zA + zH) / 2, (zB + zH) / 2, (zC + zH) / 2

blue = Color.Midnightblue
fillQR(z0, C.rho(zA - z0), Color.Yellow)
drawQR(z0, C.rho(zA - z0), blue)
fillQR(zQ, C.rho(zA1 - zQ), Color.Cyan)
drawQR(zQ, C.rho(zA1 - zQ), blue)

C.drawNgon([zA, zB, zC], blue)

```

```

for z1, z2 in zip([zA, zB, zC, zO, zO, zO], [zA1, zB1, zC1, zA2, zB2, zC2]):
    C.drawLine(z1, z2, blue)
ro = 0.07
for q in [zA1, zB1, zC1, zA2, zB2, zC2, zA3, zB3, zC3]:
    fillQR(q, ro, blue)

C.drawLine(zO, zH, Color.Red)
for q in [zO, zQ, zH]:
    fillQR(q, ro, Color.Red)

C.setText("A", zA)
delta = 0.2
C.setText("B", zB - delta)
C.setText("C", zC + delta)
C.setText("O", zO - delta)
C.setText("H", zH - delta)
C.setText("Q", zQ - delta)

if __name__ == '__main__':
    C.initPygame()
    C.run(CerculCelorNouaPuncte)

```