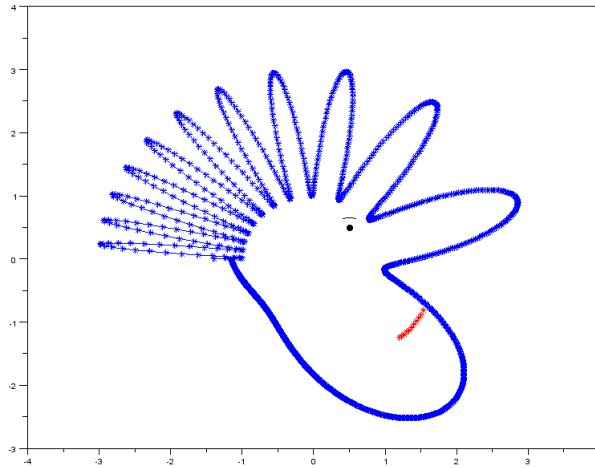


0 SciLab'i testülesanne

1. Joonestame järgmise punkari.



2. Põhijoon on kirjeldatav polaarkoordinaatides valemiga

$$r(t) = \sin(2^t) - 2, \quad t \in [0, 2\pi]. \quad (0.1)$$

Selleks, et antud joont saaks kuvada tavalisel xy -tasandil (x väärtused vastu y väärtusi), tuleks teha koordinaatide teisendus

$$\begin{cases} x = r(t) \cos(t) \\ y = r(t) \sin(t) \end{cases}. \quad (0.2)$$

3. Lahenduse idee ise on järgmine. Moodustame näiteks t väärtuste vektori

$$t = [0, 1, 2, 3] \Rightarrow r(t) = [r(0), r(1), r(2), r(3)],$$

millest

$$\cos(t) = [\cos(0), \cos(1), \cos(2), \cos(3)].$$

4. Polaarkoordinaatidelt üleminek ristkoordinaatidele.

Nüüd peab jälgima, et SciLab'is tehakse vaikumisi kõik tehted maatriksitega (vektoritega). Näiteks ei saa me korrutada omavahel $t * t$, kuna maatriksite korrutamise reegel seda ei luba (dimensioonid on valed). Korrutades $t * t'$ saame 1×1 maatriksi (1×4 ja 4×1), korrutades $t' * t$ saame 4×4 maatriksi (4×1 ja 1×4).

Suureks abiks on punktiviisiline korrutamine “. * “. Siis

$$\begin{aligned}
r(t) .* \cos(t) &= [r(0), r(1), r(2), r(3)] .* [\cos(0), \cos(1), \cos(2), \cos(3)] \\
&= [r(0)\cos(0), r(1)\cos(1), r(2)\cos(2), r(3)\cos(3)].
\end{aligned}$$

5. Nõu osa juures kasutame lihtsalt punktide graafikule kandmist.

0.1 SciLab, lahendus

```

clear // Lihtsalt kasulik käsk, tühjendab mälu
// Sissejuhataav ülesanne. Marek Kolk, 2013–15.
// Joonistame punkari

function [ xx ] = xtelg( t , raadius ) // x-telje väärtused
    xx = raadius( t ) .* cos( t ) // punktiviisiline *
endfunction

function [ yy ] = ytelg( t , raadius ) // y-telje väärtused
    yy = raadius( t ) .* sin( t )
endfunction

function [ yy ] = raadius( t ) // Joon polaarkoordinaatides
    yy = sin( 2^t )-2
endfunction

// Loomenurga t väärtused reavektorina, t muutub nullist 2 piini
t = [ 0 : %pi/720 : 2*%pi ]
x = xtelg( t , raadius ) // x-telje väärtused
y = ytelg( t , raadius ) // y-telje väärtused
// Graafikuga seonduv, saab ka lühemalt ja saab lihtsamini
gcf() // Graafiku järjehoidja
scf(0) // Tee aktiivseks aken nr 0
clf(0) // Puhasta aken nr 0
graafik = gca() // Graafiku telgede järjehoidja
graafik.data_bounds = [ -4 , 4 , -3 , 4 ]
plot( x , y , "b-" ) ; plot( x , y , "b*" )
plot( 0.5 , 0.5 , "k." )
kulm = [ -0.1 : 0.02 : 0.1 ]
plot( kulm+0.5 , cos(kulm).^2 -0.35 , "k-" )
suu = [ 0 : 0.1 : 1 ]
plot( suu/3 + 1.2 , exp(suu)/4 - 1.5 , "r*" )

```