Rešitev 2. projektne naloge MM

Aljaž Verlič, Lina Lumborovska, Blažka Blatnik, Luka Tavčer
Mentor: Damir Franetič

5. junij, 2017

1 Presek dveh implicitno danih ploskev

V $\mathbb{R}^3$ imamo podani dve poljubni implicitno podani ploskvi, opisanimi z enačbama
$f_1(x) = C_1$ in
$f_2(x) = C_2$, presek pa je množica rešitev tega nelinearnega sistema enačb.
Naša naloga je poiskati krivuljo K, ki predstavlja presek teh dveh ploskev.

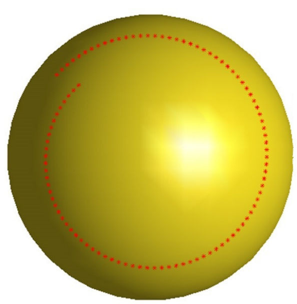
Nalogo bomo rešili na 4 načine z uporabo metod za numerično reševanje diferencialnih enačb. Uporabili bomo:

- Eulerjevo/Runge-Kutta s fiksno dolžino koraka
- Eulerjevo/Runge-Kutta z adaptivno dolžino koraka

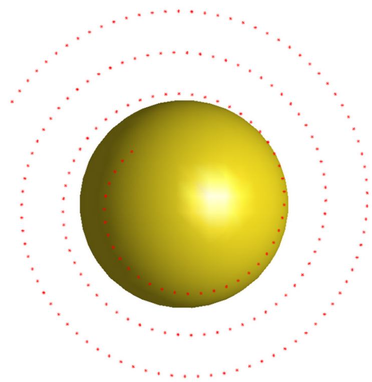
1.1 Delovanje metode

nek text spredi še....

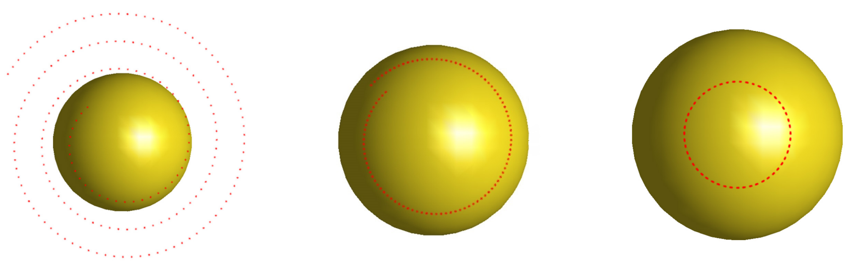
Opazimo, da je samo Eulerjeva metoda "blizu" pravilni rešitvi, ampak ni vredu.



Napake se seštevajo in so na večjem intervalu bolj opazne.



Ko za popravljanje napake uporabimo Newtonovo metodo, dobimo pravilno "krivuljo".



1.2 Potrebni pogoj in Jacobijeva matrika

Potreben pogoj za delovanje metod je, da sta funkciji f_1 in f_2 parcialno odvedljivi in da ima Jacobijeva matrika parcialnih odvodov poln rang 2. Za uspešno delovanje Newtonove metode moramo poiskati Jacobijevo matriko leve strani sistema nelinearnih enačb.

$$\mathrm{JG} = \begin{bmatrix} grad(f_1) \\ grad(f_2) \\ grad(\vec{v} \cdot \vec{x}) \end{bmatrix} \text{ oziroma } \mathrm{JG} = \begin{bmatrix} grad(f_1) \\ grad(f_2) \\ grad(\vec{v}^{\mathsf{T}}) \end{bmatrix}$$

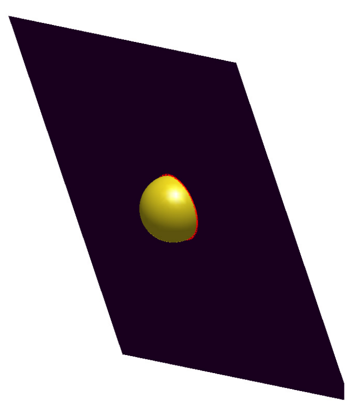
```
function [x, k] = newton(G, JG, x0, tol, maxit)
  for k = 1:maxit
    %Izvedemo en korak Newtonove metode...
    x = x0 - feval(JG, x0)\feval(G, x0);
```

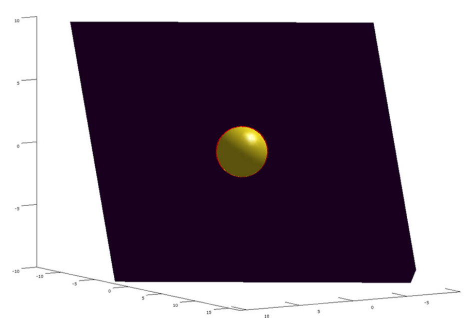
```
    if(norm(x - x0) < tol)  % Je metoda ze konvergirala?
      break;
    end
    x0 = x;
  endfor
  % Izpis opozorila, ce zadnji priblizek ni znotraj tolerance.
  if(k == maxit)
    disp("Warning: The method did not converge after maxit iterations.")
  end
endfunction
```

1.3 Implementacija, testiranje in ugotovitve

Delovanje našega programa lahko preverimo s programom, ki smo ga napisali v Octave-u. Kot vhodne parametre mu podamo obe implicitno podani funkciji f_1, f_2, $C1$, $C2$, $grad(f_1)$, $grad(f_2)$. Določimo tudi začetni približek x_0, začetno dolžino koraka in pa parameter, ki določa metodo delovanja (Euler/Runge-Kutta).
Program poženemo na različnih primerih in štejemo povprečno dolžino koraka ter število porabljenih korakov.
Primer 1:

- $f_1(x, y, z) = x^2 + y^2 + z^2 = 4$
- $f_2(x, y, z) = 3x + 2y + z = 1$

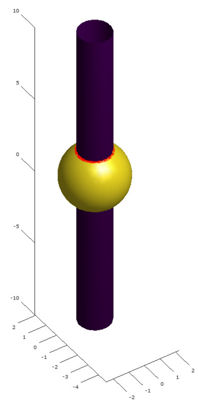
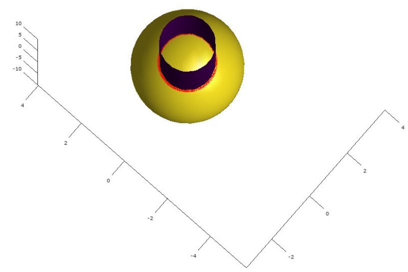




MERITEV	EULER	RK4
povprečje število korakov	3.0303	1.0505

Primer 2:

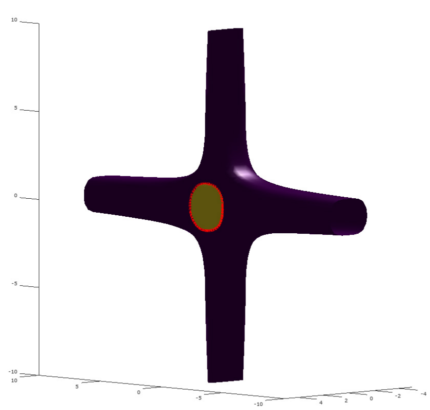
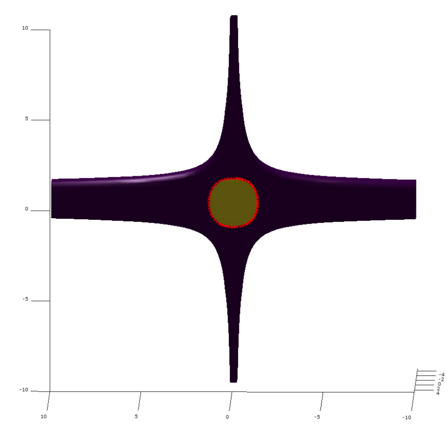
- $f_1(x, y, z) = x^2 + y^2 + z^2 = 4$
- $f_2(x, y, z) = x^2 + y^2 = 1$

	EULER	RK4
povprečno število korakov	3.0202	1.0404

Primer 3:

- $f_1(x, y, z) = x^2 + y^2 + z^2 = 4$
- $f_2(x, y, z) = y^4 + log(x^2 + 1)z^2 - 4 = 1$

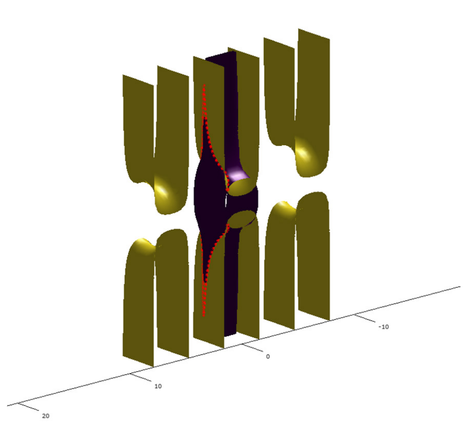
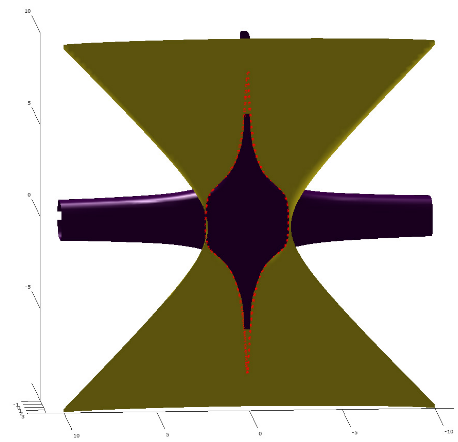
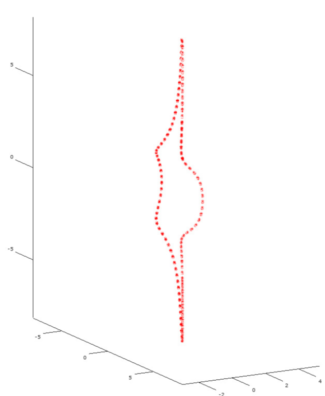
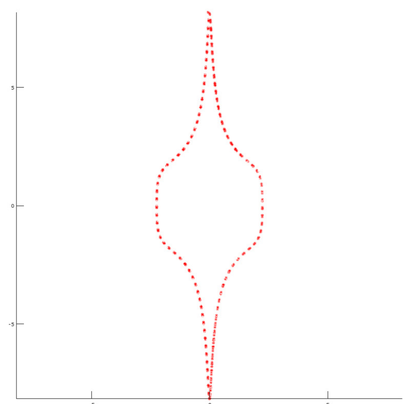



	EULER	RK4
povprečno število korakov	3.0303	1.7273

Primer 4:

- $f_1(x, y, z) = x^2 + cos(y)z^2 - 12 = 4$

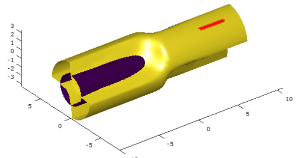
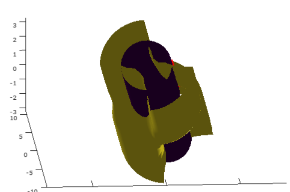
- $f_2(x, y, z) = y^4 + log(x^2 + 1)z^2 - 4 = 1$

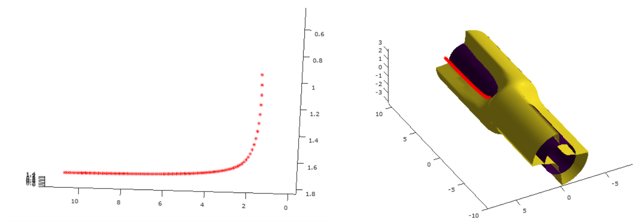





	EULER	RK4
povprečno število korakov	75.856	16.843

Primer 5:

- $f_1(x, y, z) = e^{(-x^2+1)} + y^2 + z^2 = 3$
- $f_2(x, y, z) = e^{(xyz)} + y^2 + z^2 = 10$

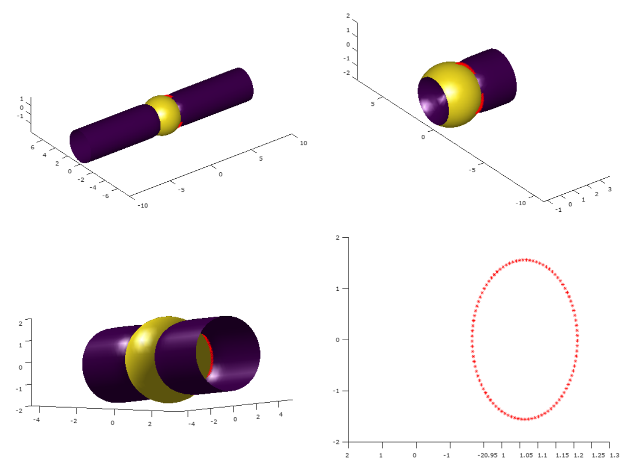





	EULER	RK4
povprečno število korakov	1.2323	2.6566

Primer 6:

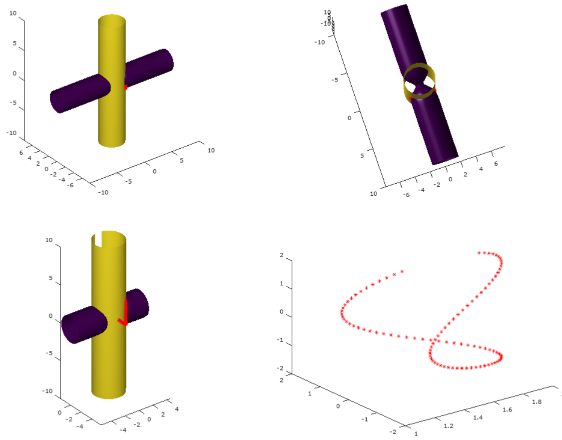
- $f_1(x, y, z) = e^{(-x^2+1)} + y^2 + z^2 = 3$
- $f_2(x, y, z) = x^2 + y^2 + z^2 = 4$



	EULER	RK4
povprečno število korakov	1.0404	3.0202

Primer 7:

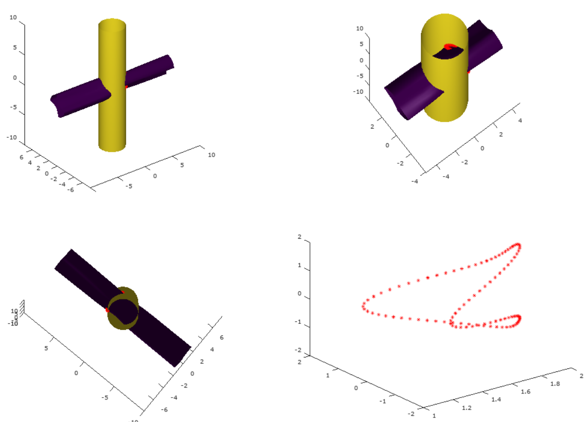
- $f_1(x, y, z) = e^{(-x^2+1)} + y^2 + z^2 = 3$
- $f_2(x, y, z) = x^2 + y^2 = 1$



	EULER	RK4
povprečno število korakov	1.9495	3.0202

Primer 8:

- $f_2(x, y, z) = e^{(xyz)} + y^2 + z^2 = 10$
- $f_2(x, y, z) = x^2 + y^2 = 1$



	EULER	RK4
povprečno število korakov	2.111	3.1212