

# report

March 30, 2024

## 1 Naravni zlepek

Gregor Kovač

### 1.1 Matematično ozadje

Danih je  $n$  interpolacijskih točk  $(x_i, f(x_i)), i = 1, \dots, n$ . *Naravni interpolacijski kubični zlepek*  $S$  je funkcija, za katero velja: -  $S(x_i) = f_i$ , -  $S$  je polinom stopnje 3 ali manj na intervalih  $[x_i, x_{i+1}]$ , -  $S$  je dvakrat zvezno odvedljiva funkcija na intervalu  $[x_1, x_n]$ , -  $S''(x_1) = S''(x_n) = 0$ .

Vsak polinom  $S_i$  zleпка  $S$  ima torej obliko  $S_i(x) = a_i + b_i \cdot x + c_i \cdot x^2 + d_i \cdot x^3$ . Za izračun koeficientov  $a_i, b_i, c_i$  in  $d_i$  najprej nastavimo naslednji sistem enačb:

$$S_i(x_i) = f(x_i), \quad i = 1, \dots, n$$

Uporabimo lahko tudi prvi odvod polinoma:  $S'_i(x) = b_i + 2 \cdot c_i \cdot x + 3 \cdot d_i \cdot x^2$ . Ker ima funkcija zvezen odvod vemo, da bo v neki interpolacijski točki odvod levega in desnega polinoma enak, torej lahko dodamo naslednje enačbe:

$$S'_{i-1}(x_i) = S'_i(x_i), \quad i = 2, \dots, n$$

Ker je funkcija dvakrat zvezno odvedljiva, upoštevamo še drugi odvod:  $S''_i(x) = b_i + 2 \cdot c_i + 6 \cdot d_i \cdot x$ . Podobno kot prej zapišemo še enačbe z drugimi odvodi:  $S''_{i-1}(x_i) = S''_i(x_i), \quad i = 2, \dots, n$

Iz zadnje točke definicije direktno dobimo še dve enačbi:

$$S''(x_1) = 0$$

$$S''(x_n) = 0$$

Dobili smo sistem s  $4n$  enačbami in  $4n$  neznankami. Sistem zapišemo v matrični obliki in ga rešimo. Kot rezultat torej dobimo koeficiente polinomov, ki sestavljajo zlepek.

### 1.2 Primeri

Sedaj si pogledajmo nekaj primerov zlepkov. Za nekatere primere bomo interpolacijske točke generirali iz funkcije. Pri teh primerih bomo izrisali originalno funkcijo s črtkano črto poleg zleпка.

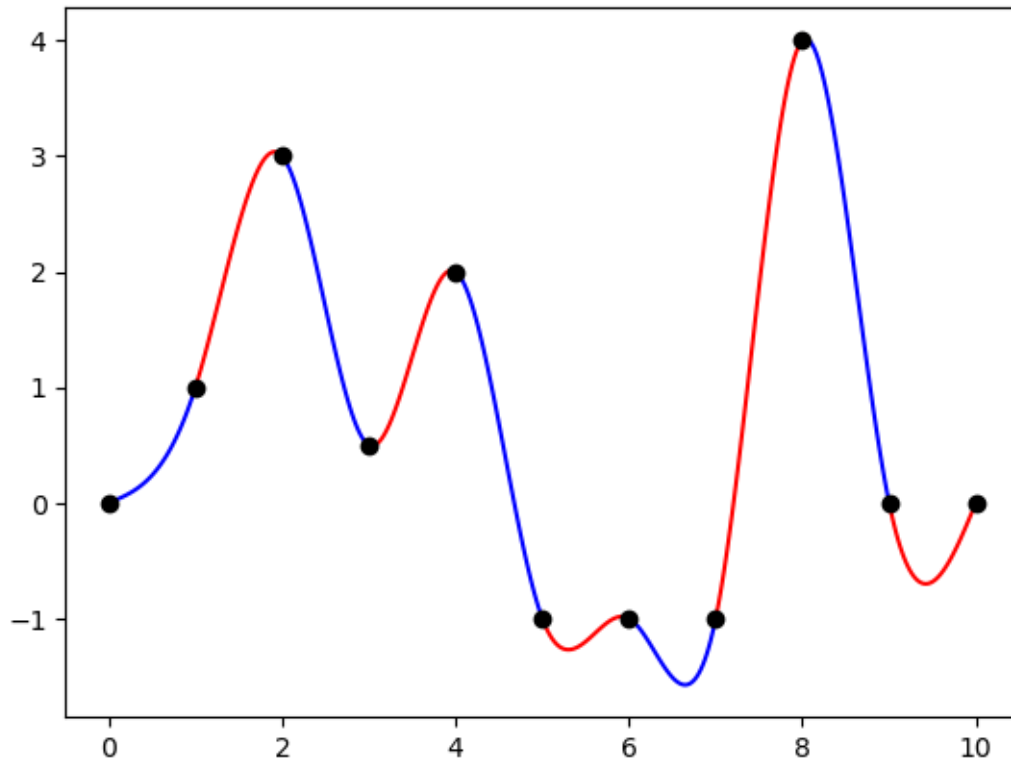
```
[ ]: # Uvoz knjižnic
import zlepek
import numpy as np
```

### 1.2.1 Poljubne točke

```
[ ]: x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
y = np.array([0, 1, 3, 0.5, 2, -1, -1, -1, 4, 0, 0])

z = zlepek.interpoliraj(x, y)

zlepek.plot(z)
```

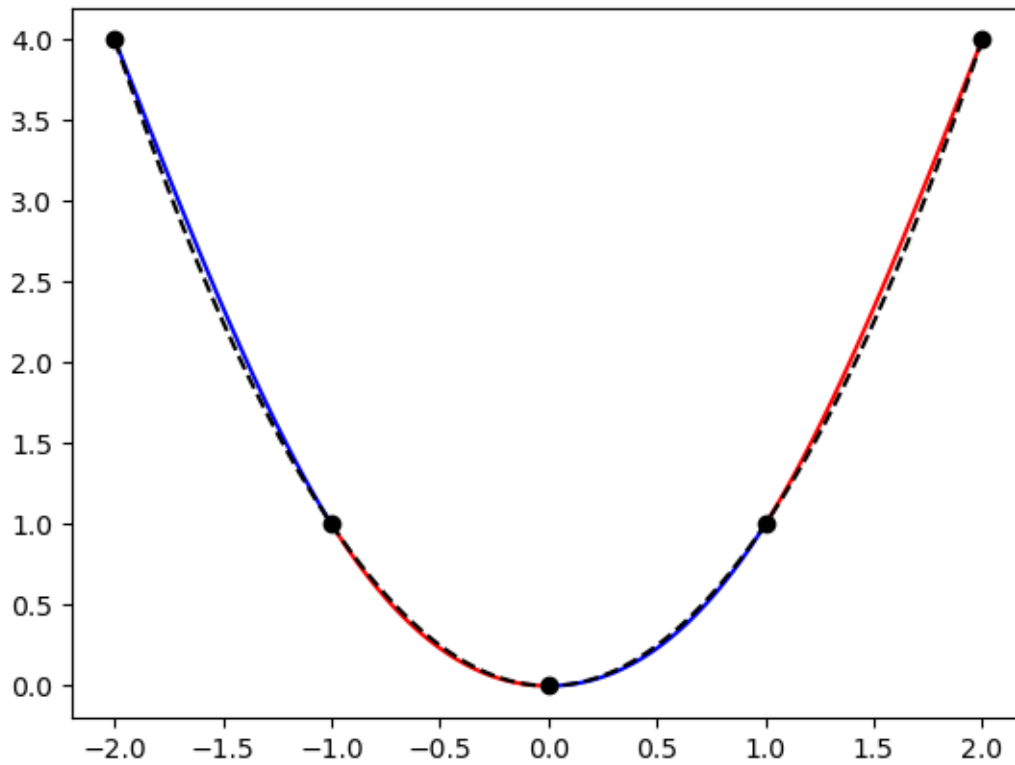


### 1.2.2 Parabola

```
[ ]: x = np.array([-2, -1, 0, 1, 2])
y = x ** 2

z = zlepek.interpoliraj(x, y)

zlepek.plot(z, lambda x: x ** 2)
```

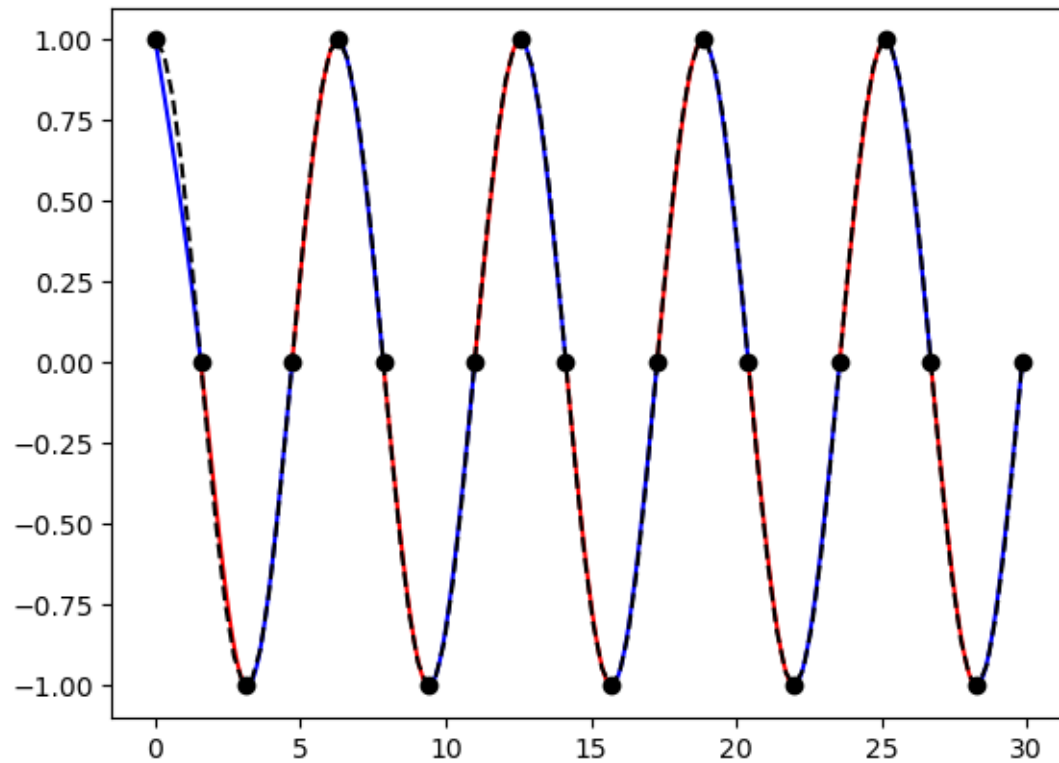


### 1.2.3 Kosinus

```
[ ]: x = np.array([k * np.pi/2 for k in range(20)])
      y = np.cos(x)

      z = zlepek.interpoliraj(x, y)

      zlepek.plot(z, lambda x: np.cos(x))
```



#### 1.2.4 Naključne točke

```
[ ]: x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

np.random.seed(1001)
y = np.random.rand(11)

z = zlepek.interpoliraj(x, y)

zlepek.plot(z)
```

