1 DN1 - Naravni zlepek, Matija Ojo, 63200205

1.1 Uvod

Imamo n interpolacijskih točk $(x_i, f_i), \quad i = 1, 2 \ \ldots \ n$. Naravni interpolacijski kubični zlepek S je funkcija, ki izpolnjuje naslednje pogoje:

1. graf zlepka gre skozi interpolacijske točke: $S(x_i) = f_i, \quad i = 1, 2 \ \ldots n$,
2. je polinom stopnje 3 ali manj na vsakem podintervalu $[x_i, x_{i+1}], \quad i = 1, 2 \ \ldots n-1$,
3. je dvakrat zvezno odvedljiva funkcija na interpolacijskem intervalu $[x_1, x_n]$,
4. drugi odvod v začetni in končni točki je enak 0: $S''(x_1) = S''(x_n) = 0$.

Zadnji pogoj je potreben, da je sistem enačb zaprt in dobimo t.i. naravni zlepek. Na ta način dobimo zlepek, ki je dvakrat zvezno odvedljiv (C^2). V primeru le 2 točk, dobimo linearni zlepek, oziroma premico.

1.2 Izračun zlepka

Zlepek S je sestavljen iz $n-1$ kubičnih polinomov, ki jih označimo z $S_i(x)$, kjer je i indeks polinoma. Vsak polinom $S_i(x)$ je oblike: $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$, kjer so a_i, b_i, c_i, d_i koeficienti polinoma, x_i je x-koordinata i-te interpolacijske točke, x pa spremenljivka.

Koeificente lahko izračunamo da izpeljemo tridiagonalni sistem. Označimo $t = x - x_i$ in brez izgube na splošnosti za naslednje izračune predpostavimo $t \in [0, 1]$. Potem je $S_i(t) = a_i + b_i t + c_i t^2 + d_i t^3$. Pri tem upoštevamo pogoje:

$$S_i(0) = f_i = a_i$$
$$S_i(1) = f_{i+1} = a_i + b_i + c_i + d_i$$
$$S_i'(0) = 0 + 1 * b_i + 2 * c_i * 0 + 3 * d_i * 0^2 = b_i =: D_i$$
$$S_i'(1) = 0 + 1 * b_i + 2 * c_i * 1 + 3 * d_i * 1^2 = b_i + 2c_i + 3d_i =: D_{i+1}$$

Z nekaj računanja dobimo:

$$a_i = f_i$$
$$b_i = D_i$$
$$c_i = 3(f_{i+1} - f_i) - 2D_i - D_{i+1}$$
$$d_i = 2(f_i - f_{i+1}) + D_i + D_{i+1}$$

Za notranje točke vemo, da se mora zlepek ujemati v soslednjih notranjih točkah ter v prvem in drugem odvodu. Tako dobimo 4 enačbe:

$$S_{i-1}(1) = f_i$$

$$S_i'(0) = f_i$$
$$S_{i-1}'(1) = S_i'(0)$$
$$S_{i-1}''(1) = S_i''(0)$$

Zlepek se mora tudi ujemati v robnih točkah, in dobimo se 2 enačbi:

$$S_0(0) = f_0$$
$$S_{n-1}(1) = f_n$$

Tako imamo $4 * (n-1)$ enačb za notranje točke in 2 enačbi za robni točki, kar znese $4n-2$ enačb za $4n$ spremenljivk.

Tak sistem v splošnem ni rešljiv, zato moramo dodati še dve enačbi, ki bosta tvorili zaprt sistem:

$$S_0''(0) = 0$$
$$S_{n-1}''(1) = 0$$

Za reševanje zaprtega tridiagonalnega sistema bomo uporabili Thomasov algoritem. Thomasov algoritem se uporablja za reševanje linearnih sistemov oblike: $a_i * x_{i-1} + b_i * x_i + c_i * x_{i+1} = d_i$. Ideja algoritma je zelo preprosta, v vsakem koraku se izračuna nove vrednosti koeficientov:

$$c_1' = \frac{c_1}{b_1}$$
$$c_i' = \frac{c_i}{b_i - a_i c_{i-1}'}, \quad i = 2, 3 \ldots n-1$$

$$d_1' = \frac{d_1}{b_1}$$
$$d_i' = \frac{d_i - a_i d_{i-1}'}{b_i - a_i c_{i-1}'}, \quad i = 2, 3 \ldots n-1$$

Nato pa se z obratno substitucijo izračuna resitev:

$$x_n = d_n'$$
$$x_i = d_i' - c_i' * x_{i+1}, \quad i = n-1, n-2 \ldots 1$$

Časovna zahtevnost tega algoritma je $O(n)$, torej enako kot učinkovito reševanje tridiagonalnih sistemov.

V nalogi je uporabljen posebej prilagojen algoritem, ki temelji na Thomasovem algoritmu.

V implementaciji smo definirali podatkovni tip `Zlepek`, ki vsebuje koeficiente za vsak posamezen zlepek ($n \times 4$ matrika) ter x koodrinate interpolacijskih točk. Za izračun vrednosti zlepka pri neki točki x uporabimo bisekcijo za iskanje ustreznega intervala in nato izračunamo vrednost polinoma v tem intervalu. Časovna zahtevnost izračuna vrednosti zlepka je $O(\lg n)$, saj je iskanje intervala z bisekcijo logaritemsko, seštevanje in množenje realnih stevil pa je konstantno.

1.3 Rezultati

```
using DN1
using Plots
```

Definirajmo nekaj poljubnih točk

```
x = [0.0, 1.0, 2.0, 3.0, 4.0, 5.0]
y = [1.0, 3.0, 1.0, 2.0, 0.0, 6.0]
s = interpoliraj(x, y)
p1 = vrednost(s, 5)
println("S(5)=", p1)
```

```
S(5)=6.0
```

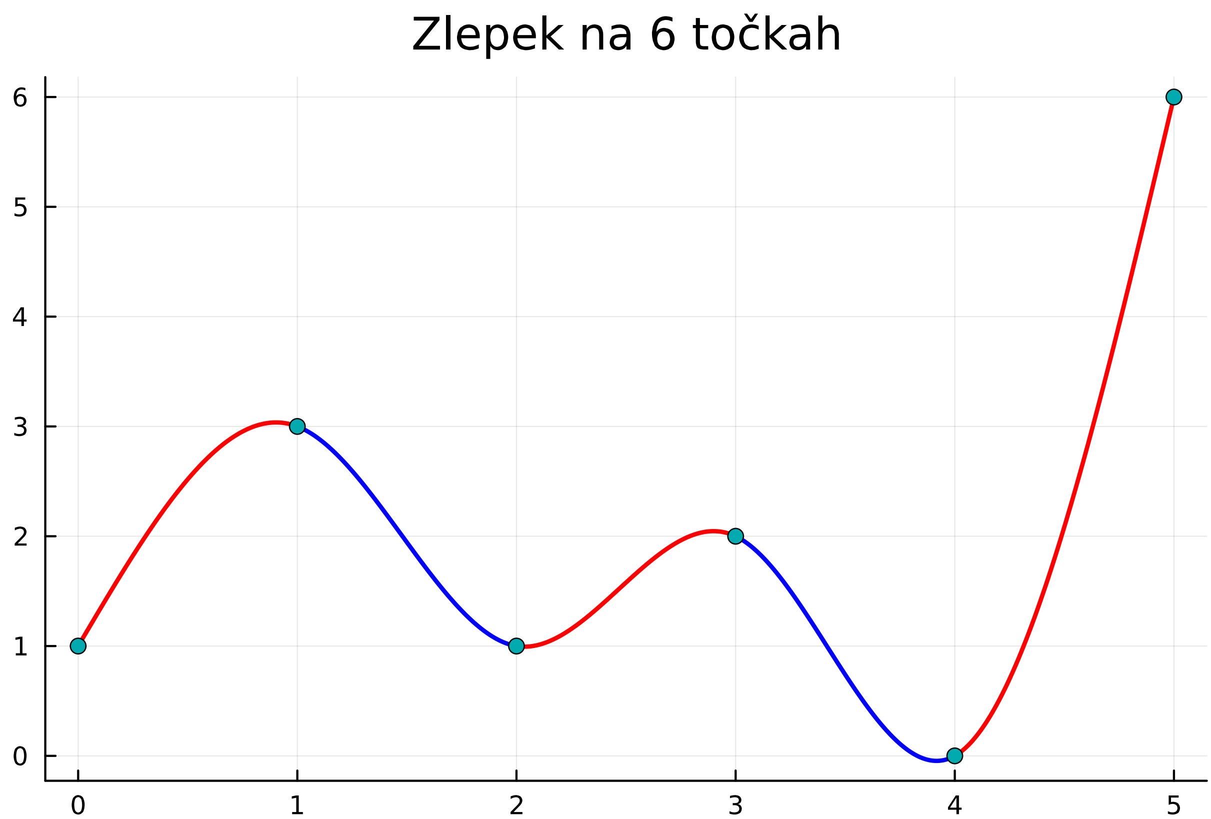
Veljati mora $S(5) = 6.0$, saj je bila točka $(5.0, 6.0)$ podana kot interpolacijska točka.

```
p2 = vrednost(s, 4.99)
println("S(4.99)=", p2)
```

```
S(4.99)=5.915648368421055
```

Veljati mora $S(4.99) \approx 6.0$, saj je točka z $x = 4.99$ blizu točki $(5.0, 6.0)$.

```
izirsi_zlepek(s)
```

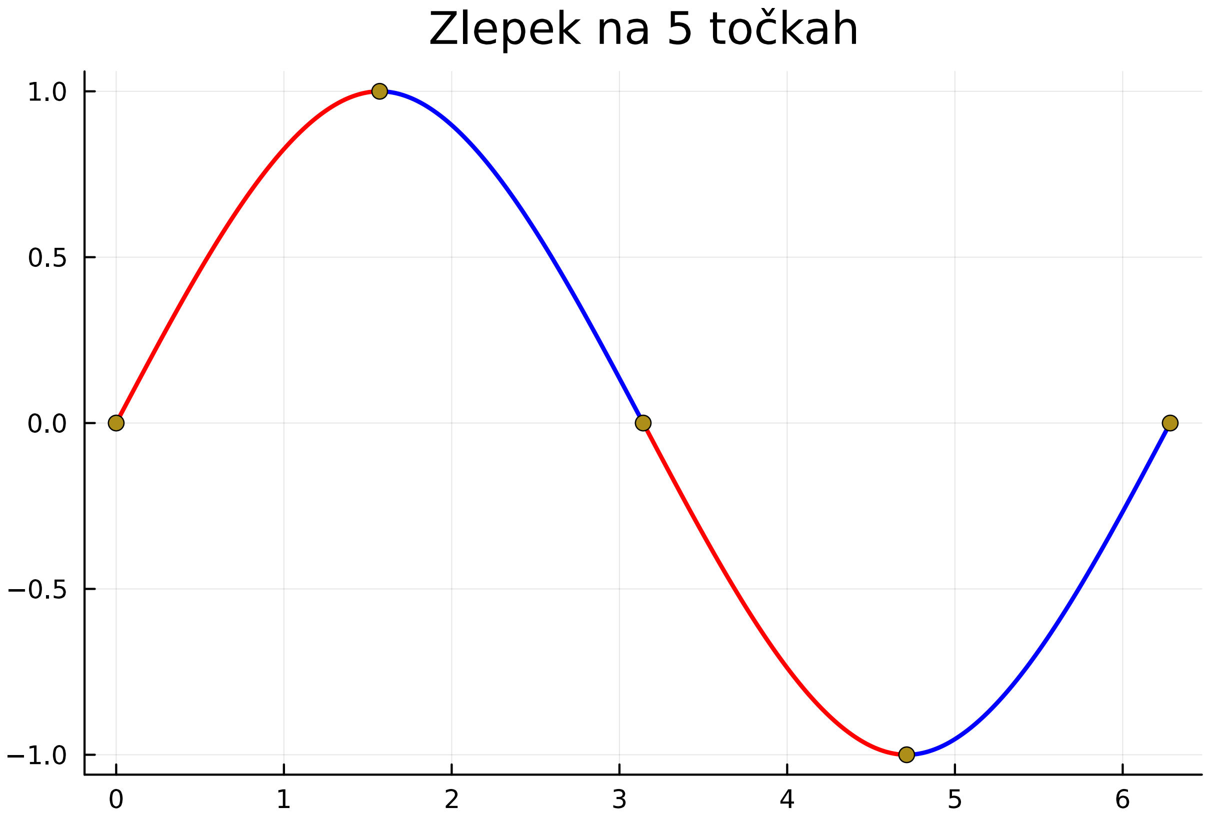


Kot vidimo iz slike, se zlepek prilega interpolacijskim točkam.

Poglejmo si še en primer.

```
x = range(0, stop=2 * pi, length=5)
y = sin.(x)
s = interpoliraj(x, y)
```

```
izirsi_zlepek(s)
```

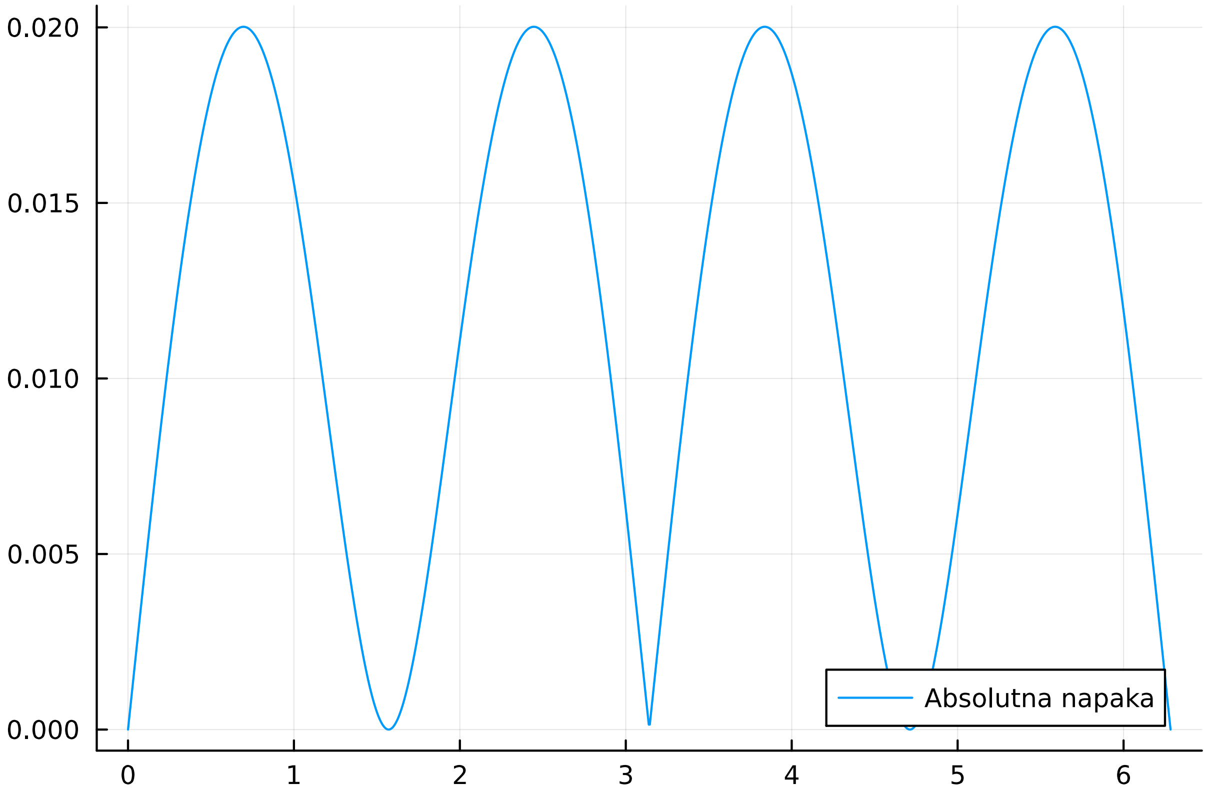


Kot vidimo iz slike, je zlepek uspešno interpoliral funkcijo sinus. Poglejmo si še graf napake za interpolacijo funkcije sinus na le 5 točkah.

```
function graf_napake(x, y)
  s = interpoliraj(x, y)
  xs = range(x[1], stop=x[end], length=1000)
  diffs = [abs(vrednost(s, xi) - sin(xi)) for xi in xs]

  return plot(xs, diffs, label="Absolutna napaka")
end

graf_napake(x, y)
```

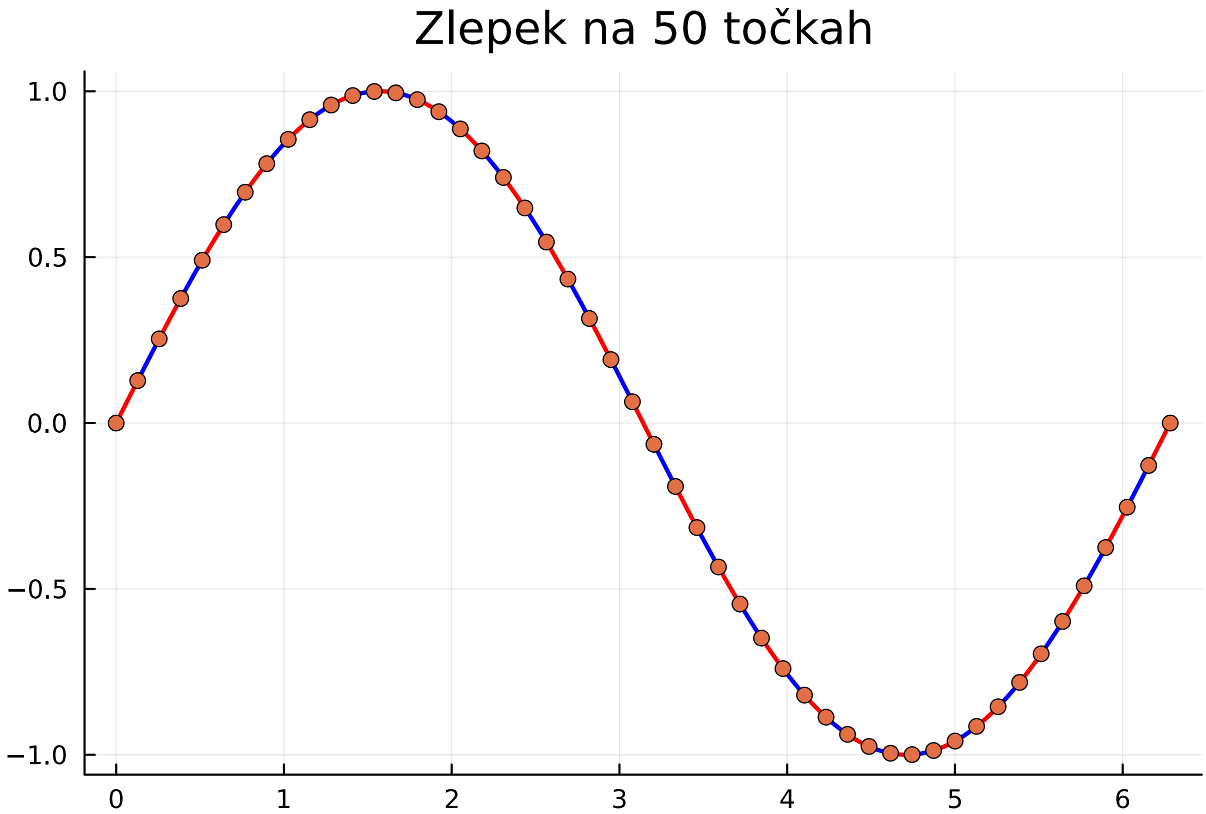


Kot vidimo iz grafa, je napaka interpolacije funkcije v vsaki interpolacijski točki enaka 0, Med točkami pa je največja napaka na sredini intervala, kjer je točka najbolj oddaljena od interpolacijskih točk po x koordinati. Največja absolutna napaka ima vrednost 0.02, kar ni popolnoma zanemarljivo. Zavedati pa se moramo, da smo interpolirali funkcijo sinus le na 5 točkah.

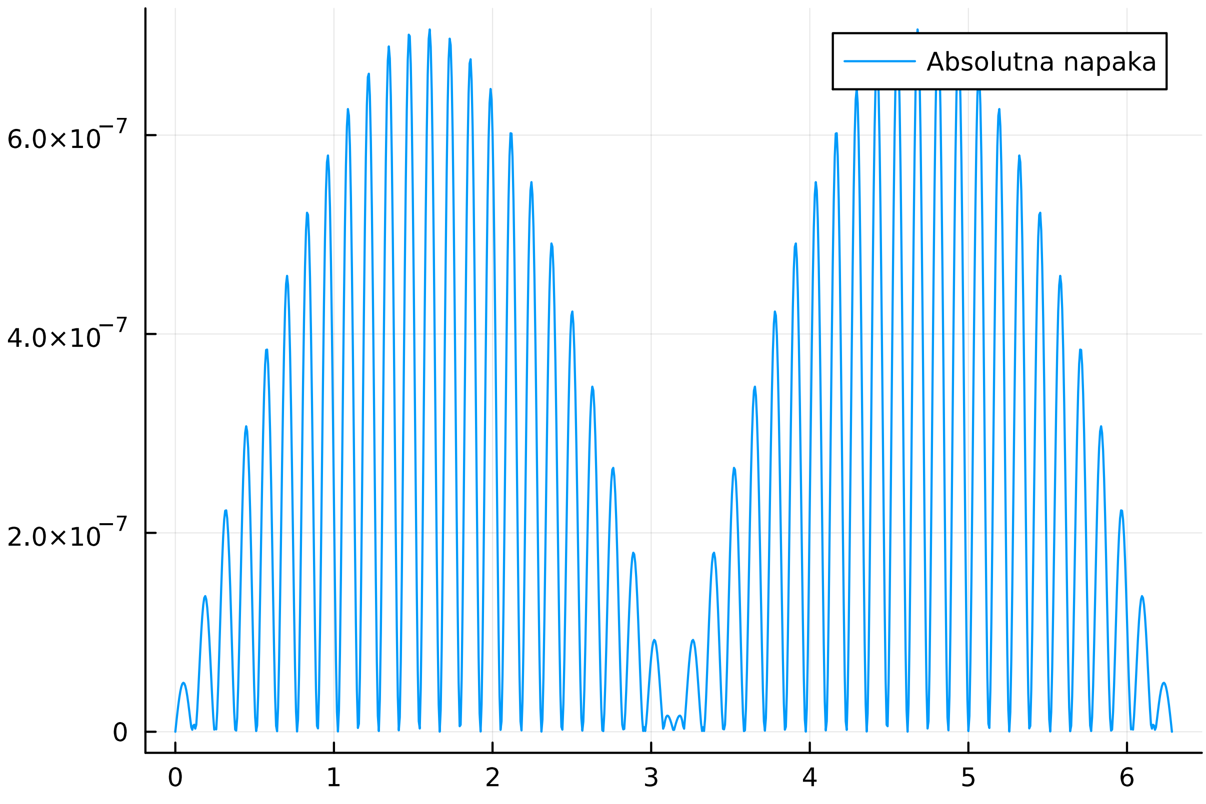
Poglejmo si še napako pri interpolaciji funkcije sinus na 50 točkah na istem intervalu.

```
x = range(0, stop=2 * pi, length=50)
y = sin.(x)
s = interpoliraj(x, y)

izirsi_zlepek(s)
```




```
graf_napake(x, y)
```



Absolutna napaka je sedaj bistveno manjša do te mere da je zanemarljiva.