INTERPOLACIJA

Interpolacija – problem

- Problem: odrediti vrednosti nepoznate funkcije $f: x \to y, y \in \mathbb{R}$

- Funkcija f je nepoznata u smislu da ne poznajemo tačnu matematičku formulu zavisnosti y od x

- U mogućnosti smo da izvršimo eksperimentalna merenja pomoću kojih možemo da dobijemo vrednosti y u *određenim tačkama* x
  - Poznat nam je skup podataka (x_1,y_1), (x_2,y_2), (x_3,y_3),..., (x_n,y_n).
  - Želimo da odredimo y za x koje nije ni jedno od x_i

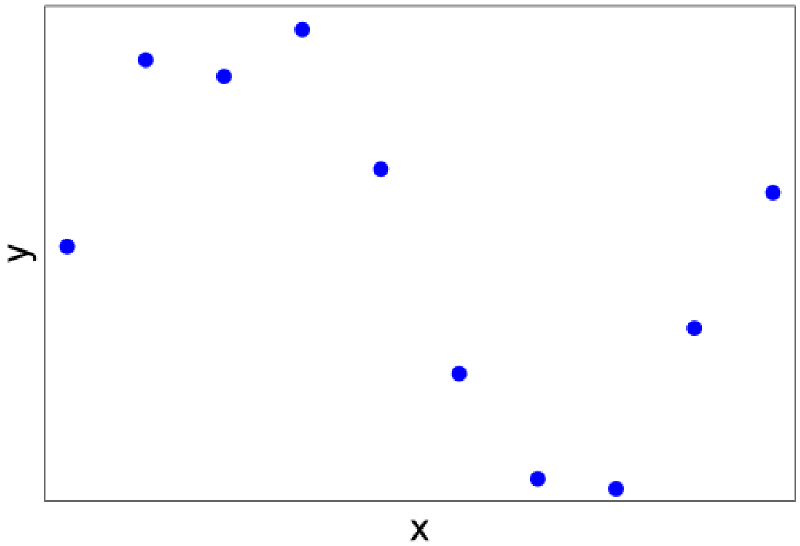
Interpolacija – problem

Tabela 4.1. Tačke (x_i, y_i) u kojima nam je poznata vrednost funkcije $f: x \to y$.

x_i	0.000	0.698	1.400	2.09	2.79	3.49	4.19	4.89	5.59	6.28
y_i	0.0538	0.826	0.759	0.952	0.374	-0.473	-0.909	-0.951	-0.285	0.277

- Kako da odredimo vrednost y u tačkama x koje se ne nalaze među izmerenim vrednostima?
  - Npr., $y = f(4) = ?$

Poznate vrednosti



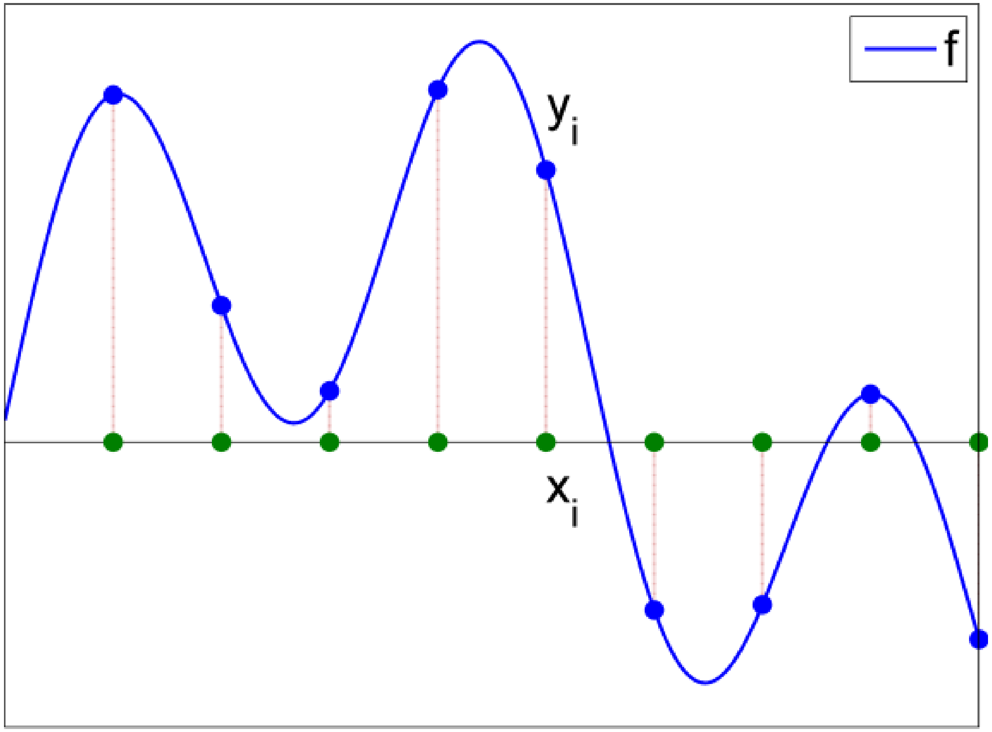
Interpolacija – motivacija: zvuk

- Zvuk je kontinualni (analogni) talas

- x_i - vremenski trenutak
- y_i - vrednost zvučnog signala u x_i

- Digitalni zvuk: pri snimanju uzorkujemo samo određene tačke

- Stvarni oblik funkcije zvučnog signala f nam je nepoznat i zvuk imamo snimljen samo kao niz tačaka (x_i, y_i)



Interpolacija – motivacija: zvuk

- Da bismo preslušali snimljeni zvuk, njegov digitalni zapis moramo da pretvoriti u analogni talas

- Odnosno, moramo odrediti vrednosti y između tačaka u kojima je zvuk snimljen
  - rekonstruisati vrednosti odbačene tokom snimanja

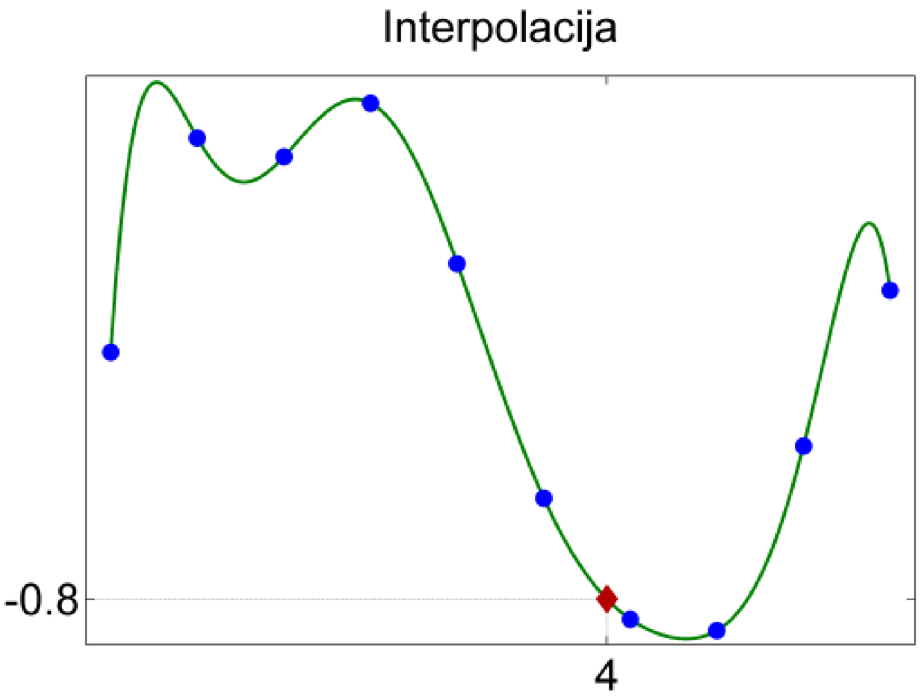
Interpolacija – osnovna ideja

- *Aproksimiraćemo* funkciju f drugom funkcijom g ($g \approx f$)

- Formula funkcije g će nam biti poznata i moći ćemo je koristiti da odredimo nepoznate vrednosti funkcije f

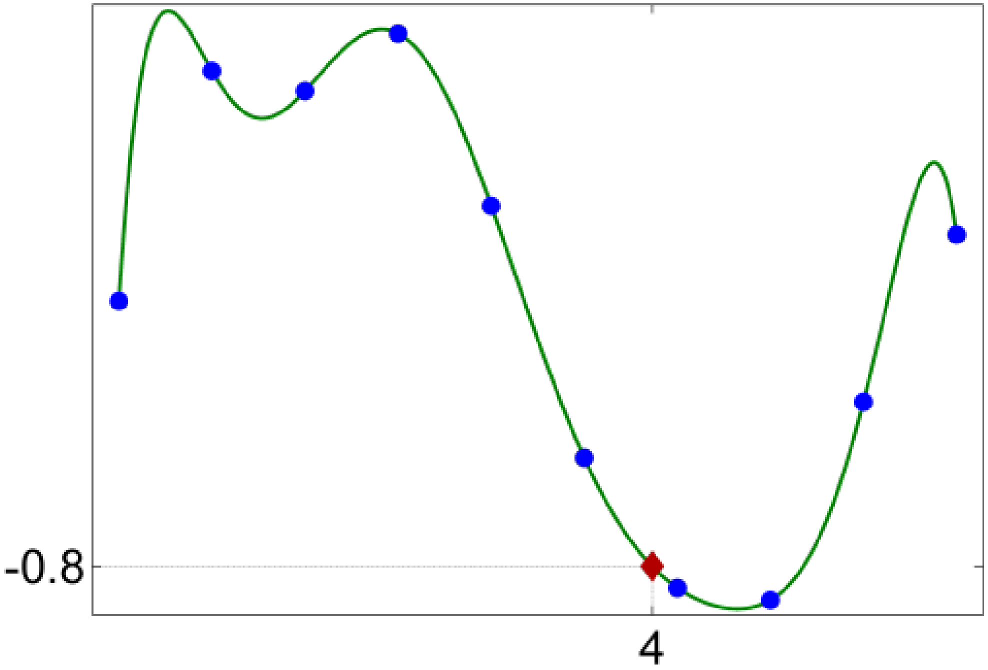
- Na primer, ako znamo formulu funkcije $g \approx f$, možemo odrediti *približnu* vrednost funkcije f u tački x_q kao $g(x_q)$

- f – plave tačke
- g – zelena kriva
- Određujemo $f(4)$



Interpolacija – osnovna ideja

- Iterpolacija: funkcija g *mora* da prođe kroz svaku od poznatih tačaka (x_i, y_i)
- Najčešće, za funkciju g koristimo polinom (polinomijalna interpolacija)



Motivacija: skaliranje slike

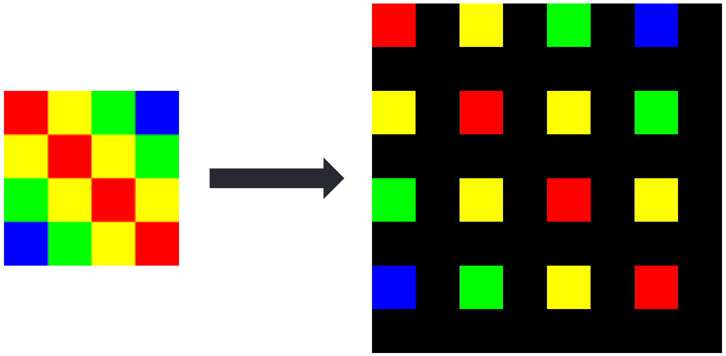
- Promena dimenzija slike: smanjivanje ili povećavanje broja piksela

- Slika rezolucije 800×600 sadrži $800 \cdot 600 = 480\,000$
- Slika rezolucije 1600×1200 sadrži $1600 \cdot 1200 = 1\,920\,000$ piksela
- Dakle, kada uvećamo rezoluciju sa 800×600 na 1600×1200 imamo $1\,920\,000 - 480\,000 = 1\,440\,000$ piksela koje treba da popunimo odgovarajućim vrednostima – kako ih odrediti?

Motivacija: skaliranje slike

- Ako sliku $m \times n$ želimo da uvećamo dva puta, prvi korak je da proširimo originalnu matricu: $(2 \cdot m) \times (2 \cdot n)$
- Nedostajuće vrednosti su 0

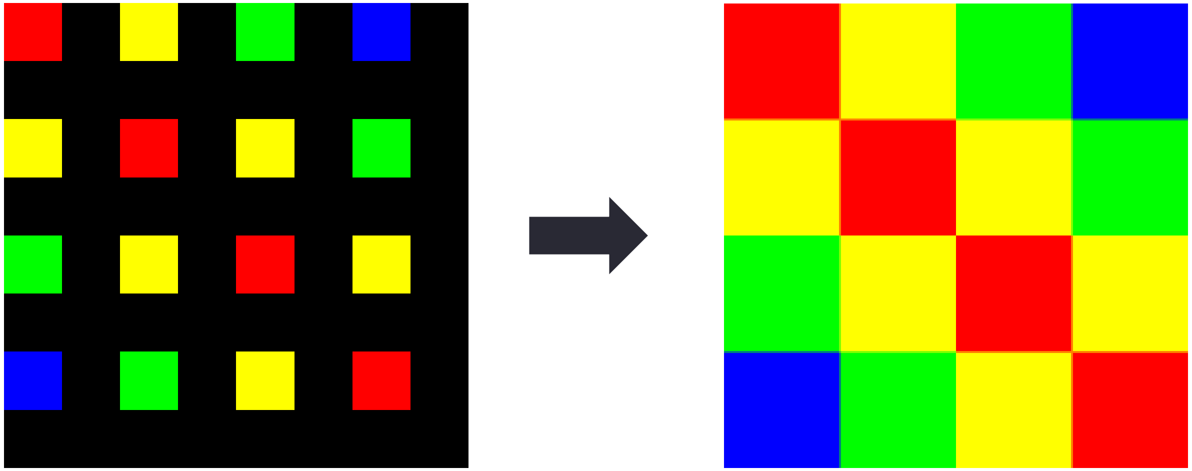
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{1} & 0 & \mathbf{2} & 0 & \mathbf{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{4} & 0 & \mathbf{5} & 0 & \mathbf{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{7} & 0 & \mathbf{8} & 0 & \mathbf{9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Motivacija: skaliranje slike

Skaliranje slike – *Nearest Neighbor*

- Najjednostavniji metod je da svaki nedostajući piksel mapiramo na jedan od (najbližih) piksela originalne slike
- To je interpolacija najbližih komšija



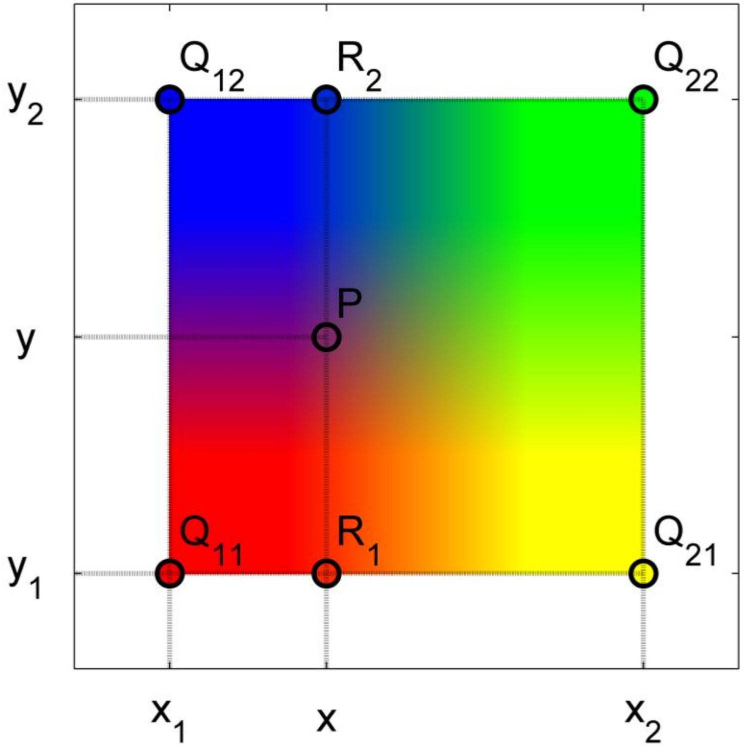
Skaliranje slike – *Nearest Neighbor*

```
imshow(imresize(imread('original.jpg'),2.0,'nearest'))
```

- Nedostatak: „grubi“ prelazi između boja na slikama

Skaliranje slike – bilinearna interpolacija

- Rešenje: bilinearna interpolacija
  - Prvo primenimo linearnu interpolaciju u jednom pravcu, a zatim primenimo linearnu interpolaciju i u drugom pravcu



- Poznati pikseli: $Q_{11}, Q_{12}, Q_{21}, Q_{22}$
- Određujemo: P

1. Linearna interpolacija po x osi da odredimo R_1 i R_2:

$$f(R_1) = f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

(R_1 - težinski prosek Q_{11} i Q_{22})

2. Linearna interpolacija po y osi da odredimo P kao težinski prosek R_1 i R_2

Skaliranje slike – bilinearna interpolacija

```
imshow(imresize(imread('original.jpg'),2.0,'bilinear'))
```

Nearest Neighbor

Bilinearna interpolacija

Skaliranje slike

- Povećanje stepena interpolacionog polinoma će povećati kvalitet slike

Bilinearna interpolacija

Bikubna interpolacija

```
imshow(imresize(imread('original.jpg'),2.0,'bicubic'))
```

Interpolacija polinomom

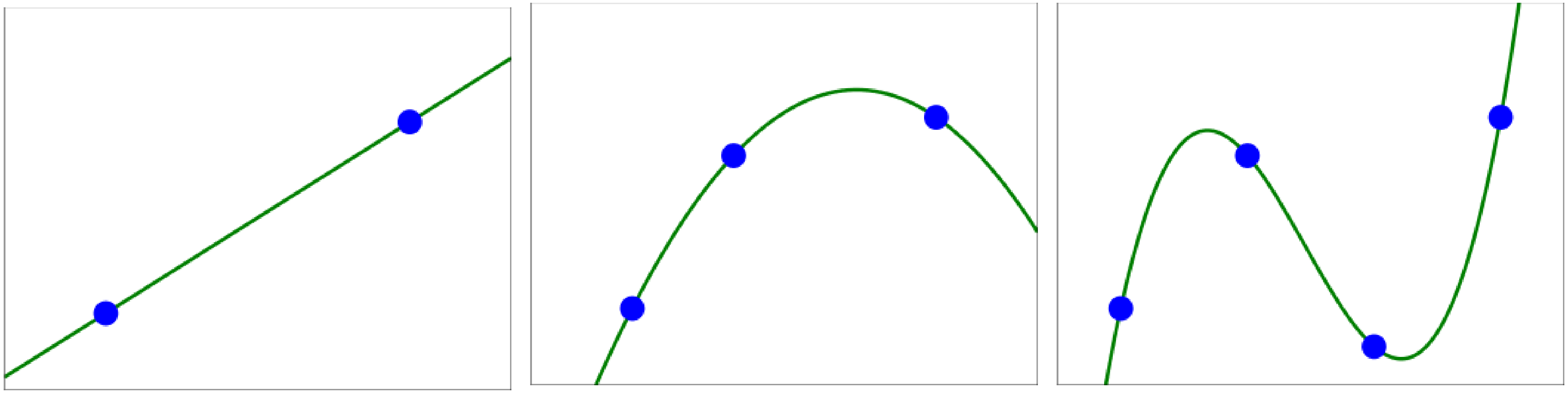
- Zadatak: pronaći jedinstveni polinom stepena $N - 1$
$$g(x) = a_1 + a_2 x + a_3 x^2 + \cdots + a_N x^{N-1}$$
koji prolazi kroz svaku od zadatih N tačaka

- g – interpolacioni polinom

- Svodi se na određivanja vrednosti koeficijenata a_i koji jedinstveno definišu polinom

Jedinstvenost interpolacionog polinoma

- Za datih N tačaka, polinom stepena $N-1$ koji prolazi kroz svaku od datih tačaka je jedinstven



Polinomi u MATLAB-u

- Polinomi se mogu predstaviti kao vektori koeficijenata
- Na primer: $p(x) = 6.6x^3 - 5x^2 - 3x + 0.7$
- MATLAB reprezentacija: $p = [6.6, -5, -3, 0.7]$
- MATLAB određivanje vrednosti $p(1)$: polyval(p, 1)

- Zbog konzistentnosti sa MATLAB-om $g(x)$ ćemo zapisivati:

$$g(x) = p_1 x^{N-1} + p_2 x^{N-2} + \cdots + p_{N-1} x + p_N$$

(umesto $g(x) = a_1 + a_2 x + a_3 x^2 + \cdots + a_N x^{N-1}$)

Koeficijenti interpolacionog polinoma

$$g(x) = p_1 x^{N-1} + p_2 x^{N-2} + \cdots + p_{N-1} x + p_N$$

- Mora da prođe kroz tačke (x_i, y_i): $g(x_i) = y_i$ za svako $i = 1, \ldots, N$

- Ovo definiše sistem od N jednačina sa N nepoznatih

$$\begin{gathered} g(x_1) = y_1 = p_1 x_1^{N-1} + p_2 x_1^{N-2} + \cdots + p_{N-1} x_1 + p_N \\ g(x_2) = y_2 = p_1 x_2^{N-1} + p_2 x_2^{N-2} + \cdots + p_{N-1} x_2 + p_N \\ \vdots \\ g(x_N) = y_N = p_1 x_N^{N-1} + p_2 x_N^{N-2} + \cdots + p_{N-1} x_N + p_N \end{gathered}$$

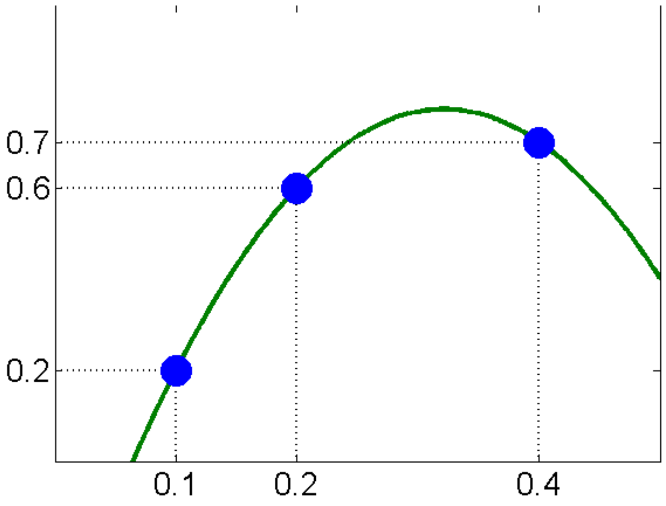
Koeficijenti interpolacionog polinoma

- Matrični oblik:

$$\begin{bmatrix} x_1^{N-1} & x_1^{N-2} & \dots & x_1 & 1 \\ x_2^{N-1} & x_2^{N-2} & \dots & x_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_N^{N-1} & x_N^{N-2} & \dots & x_N & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

- Ali...
  - Rešavanje sistema nije uvek lako, a ni efikasno
  - Problem su loše uslovljeni sistemi

Koeficijenti interpolacionog polinoma



x_i	0.1	0.2	0.4
y_i	0.2	0.6	0.7

Polinom g koji prolazi kroz sve tri tačke mora biti 2. stepena $g(x) = p_1 x^2 + p_2 x + p_3$

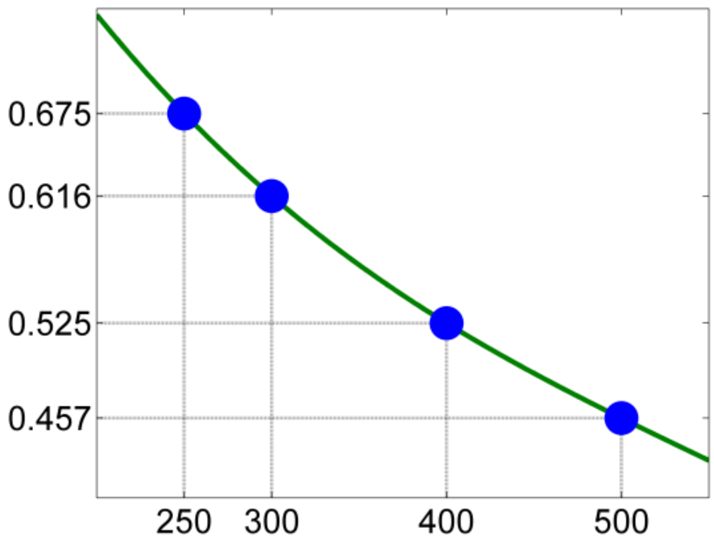
$$A = \begin{bmatrix} 0.1^2 & 0.1 & 1 \\ 0.2^2 & 0.2 & 1 \\ 0.4^2 & 0.4 & 1 \end{bmatrix} \quad p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad b = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.7 \end{bmatrix}$$

```
A = [0.1^2 0.1 1; 0.2^2 0.2 1; 0.4^2 0.4 1];
b = [0.2; 0.6; 0.7];
p = A\b
```

- cond(A) je 125.2586 (nije prevelik – dobro uslovljen sistem)

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} -11.7 \\ 7.50 \\ -0.433 \end{bmatrix}$$, tj. $g(x) = p_1 x^2 + p_2 x + p_3 = -11.7 \cdot x^2 + 7.5 \cdot x - 0.433$

Koeficijenti interpolacionog polinoma



$$A = \begin{bmatrix} 250^3 & 250^2 & 250 & 1 \\ 300^3 & 300^2 & 300 & 1 \\ 400^3 & 400^2 & 400 & 1 \\ 500^3 & 500^2 & 500 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 0.675 \\ 0.616 \\ 0.525 \\ 0.475 \end{bmatrix}$$

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} -0.000000002600000 \\ 0.000004270000000 \\ -0.002937000000000 \\ 1.183000000000007 \end{bmatrix}$$

x_i	250	300	400	500
y_i	0.675	0.616	0.525	0.457

cond(A) je $9.31 \cdot 10^9$ - loše uslovljen sistem

- Tip matrice A se naziva *Vandermondova matrica*

Koeficijenti interpolacionog polinoma

- Čak i za svega četiri tačke možemo dobiti loše uslovljen sistem
  - osetljiv na greške prilikom nastale prilikom zaokruživanja
- Ovaj problem postaje još izraženiji kada imamo više tačaka (veći broj jednačina)
- Generalno, gotovo sve velike ili čak umereno velike Vandermondove matrice imaju veoma veliki kondicioni broj
- Treba nam bolji pristup od rešavanja SLAJ...

Zapis polinoma g

- „Standardni“ oblik:

$$g(x) = p_1 x^{N-1} + p_2 x^{N-2} + \cdots + p_{N-1} x + p_N$$

- Alternativni način zapisa je Njutnov interpolacioni polinom

$$g_N(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2) + \cdots \\ + b_n(x - x_1)(x - x_2) \cdots (x - x_{N-1})$$

- Radi se o istom polinomu zapisanom u drugačijem obliku

Njutnova interpolacija

- Koriste se Njutnove “podeljene (konačne) razlike” da bi se odredili koeficijenti polinoma koji ima oblik:

$$\boxed{\begin{aligned} f_{n-1}(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2) + \cdots \\ + b_n(x - x_1)(x - x_2)\cdots(x - x_{n-1}) \end{aligned}}$$

- Koeficijenti nižeg reda (stepena) ostaju isti kad se poveća stepen polinoma

- Dakle, lako je dodati nove podatke (tačke) i uraditi interpolaciju polinomom većeg stepena

Njutnova linearna interpolacija

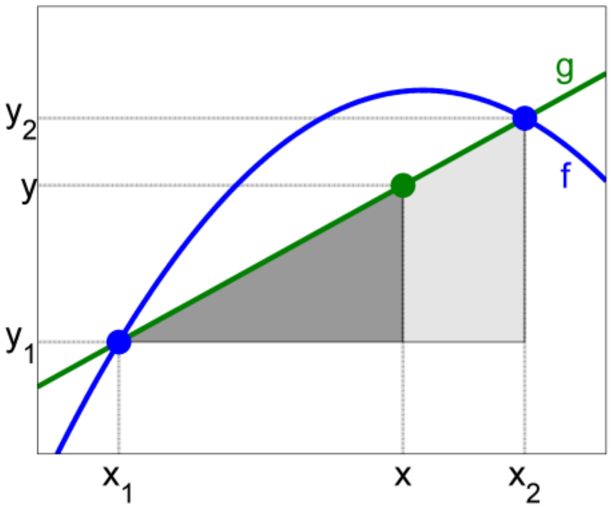
- Interpolacioni polinom je prava: $g(x) = p_1 x + p_2$
- Njutnova forma: $g(x) = b_1 + b_2(x - x_1)$

- Sličnost trouglova (jedan od načina da se odredi jednačina date prave):

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

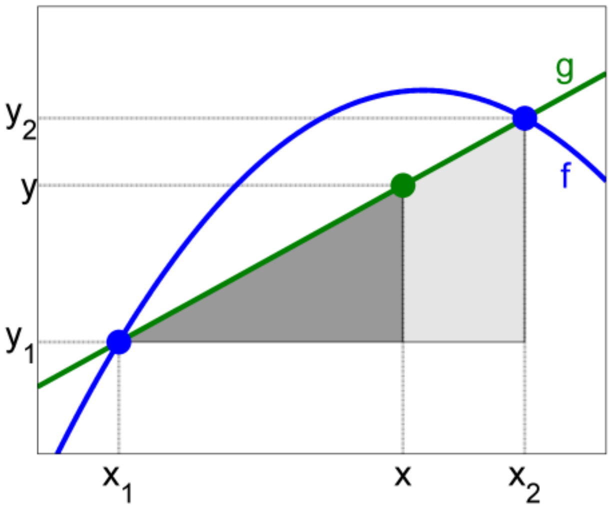
$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

- Tj. $b_1 = y_1$ i $b_2 = (y_2 - y_1)/(x_2 - x_1)$



Njutnova linearna interpolacija

- Alternativno:



$$g(x_1) = y_1 = kx_1 + n$$
$$g(x_2) = y_2 = kx_2 + n$$

(2) – (1):

$$y_2 - y_1 = k(x_2 - x_1)$$

$$k = \frac{y_2 - y_1}{x_2 - x_1}; n = y_1 - kx_1$$

$$y = g(x) = kx + n = kx + y_1 - kx_1 = y_1 + k(x - x_1)$$

$$y = y_1 + k(x - x_1) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

Njutnova linearna interpolacija

- Njutnova linearna interpolacija formula:

$$\boxed{g_1(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1)}$$

- Primer 1: odrediti e^2 pomoću vrednosti za e^1 i e^5.

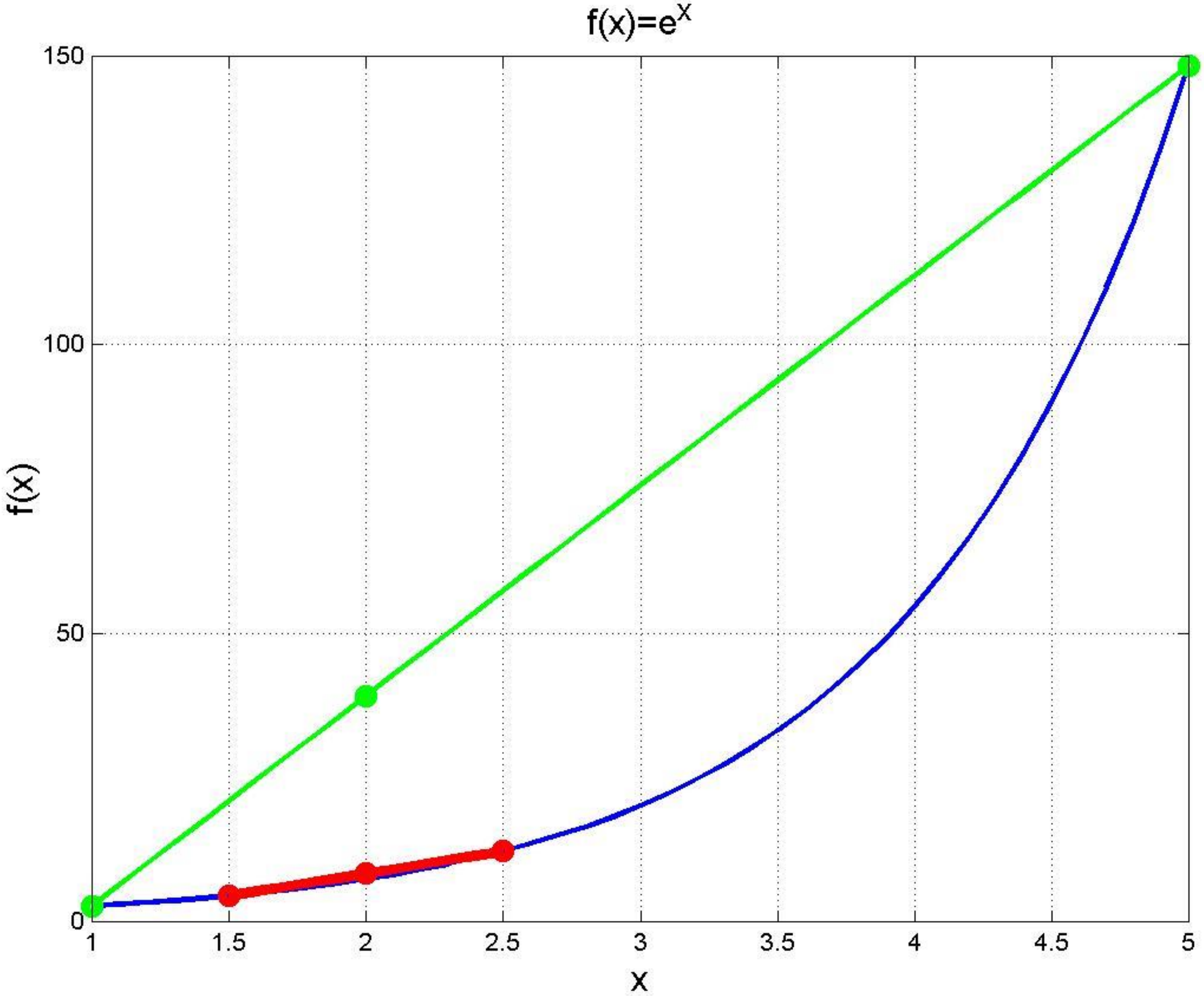
$$\boxed{f_1(2) = e^1 + \frac{e^5 - e^1}{5 - 1}(2 - 1) = 2.7183 + \frac{148.41 - 2.7183}{4}(1) = 36.423}$$

- Primer 2: odrediti e^2 pomoću vrednosti za $e^{1.5}$ i $e^{2.5}$.

$$\boxed{f_1(2) = e^{1.5} + \frac{e^{2.5} - e^{1.5}}{2.5 - 1.5}(2 - 1.5) = 4.4817 + \frac{12.1825 - 4.4817}{1}(0.5) = 8.3321}$$

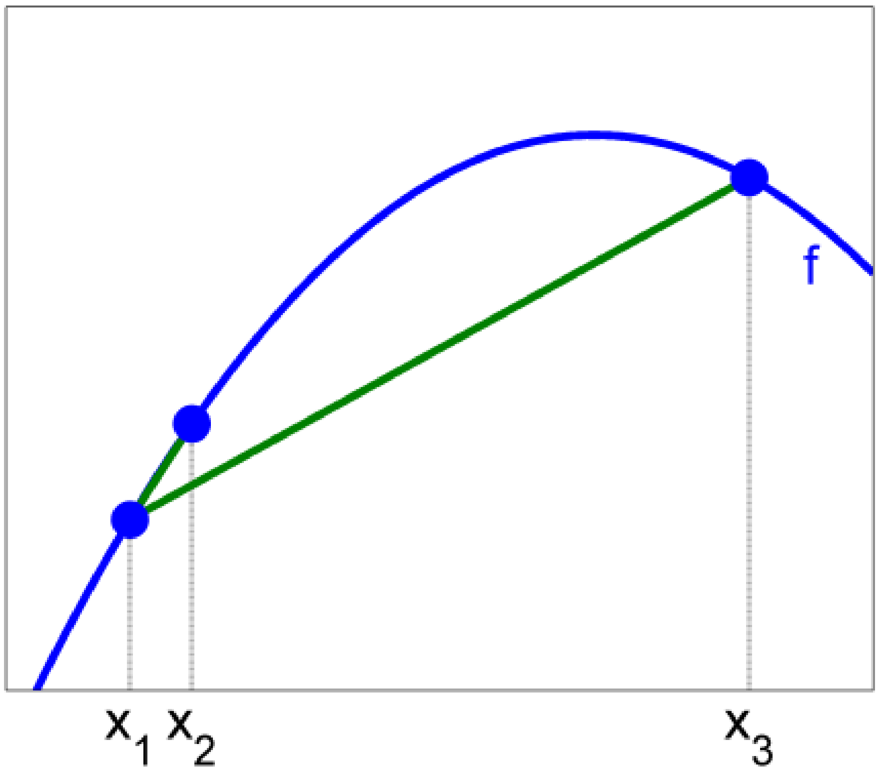
Tačno: $\boxed{e^2 = 7.3891}$

Primer 1 i 2



Tačnost procena

- U opštem slučaju, što je interval između dve tačke $[x_1, x_2]$ manji, funkcija će biti bolje aproksimirana pravom linijom



Njutnova kvadratna interpolacija

- U prethodnim primerima nam je vrednost funkcije bila dostupna u samo dve tačke
- Zbog toga smo aproksimirali krivu liniju pravom linijom
- Iz te aproksimacije je poticala naša greška procene vrednosti funkcije u nepoznatim tačkama
- Ova greška se može smanjiti ukoliko umesto prave linije koristimo krivu

Njutnova kvadratna interpolacija

- Ako su nam dostupne tri tačke, određujemo parabolu:

$$g_2(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2)$$

- Intuitivno, ovo bi morala biti bolja procena f:
  - imamo više informacija o f (dodatnu tačku)
  - parabola se može bolje prilagoditi krivoj u odnosu na pravu

Njutnova kvadratna interpolacija

- Kako odrediti koeficijente b_i?

$$g_2(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2)$$

- Zamenimo x = x_1, da bi dobili $b_1 = f(x_1)$

$$g_2(x_1) = y_1 = b_1 + b_2(x_1 - x_1) + b_3(x_1 - x_1)(x_1 - x_2) = b_1$$

$$\Rightarrow b_1 = y_1$$

- Korisimo b_1 i zamenimo x = x_2 da bi dobili b_2

$$g_2(x_2) = y_2 = b_1 + b_2(x_2 - x_1) + b_3(x_2 - x_1)(x_2 - x_2)$$
$$= b_1 + b_2(x_2 - x_1)$$

$$\Rightarrow b_2 = \frac{y_2 - b_1}{x_2 - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

Njutnova kvadratna interpolacija

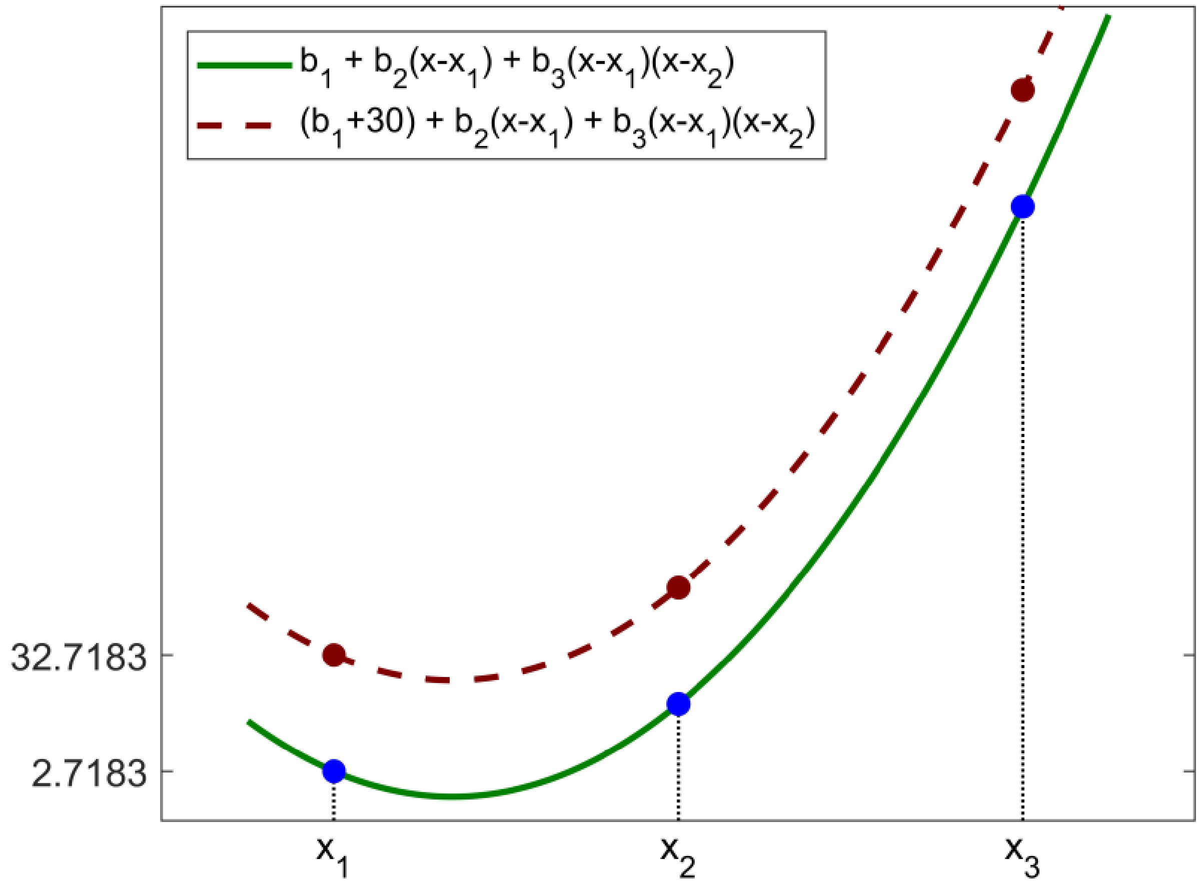
- Konačno, iskoristićemo tačku (x_3, y_3) da odredimo b_3:

$$g_2(x_3) = y_3 = b_1 + b_2(x_3 - x_1) + b_3(x_3 - x_1)(x_3 - x_2)$$

$$\Rightarrow b_3 = \frac{\dfrac{y_3 - y_2}{x_3 - x_2} - \dfrac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$$

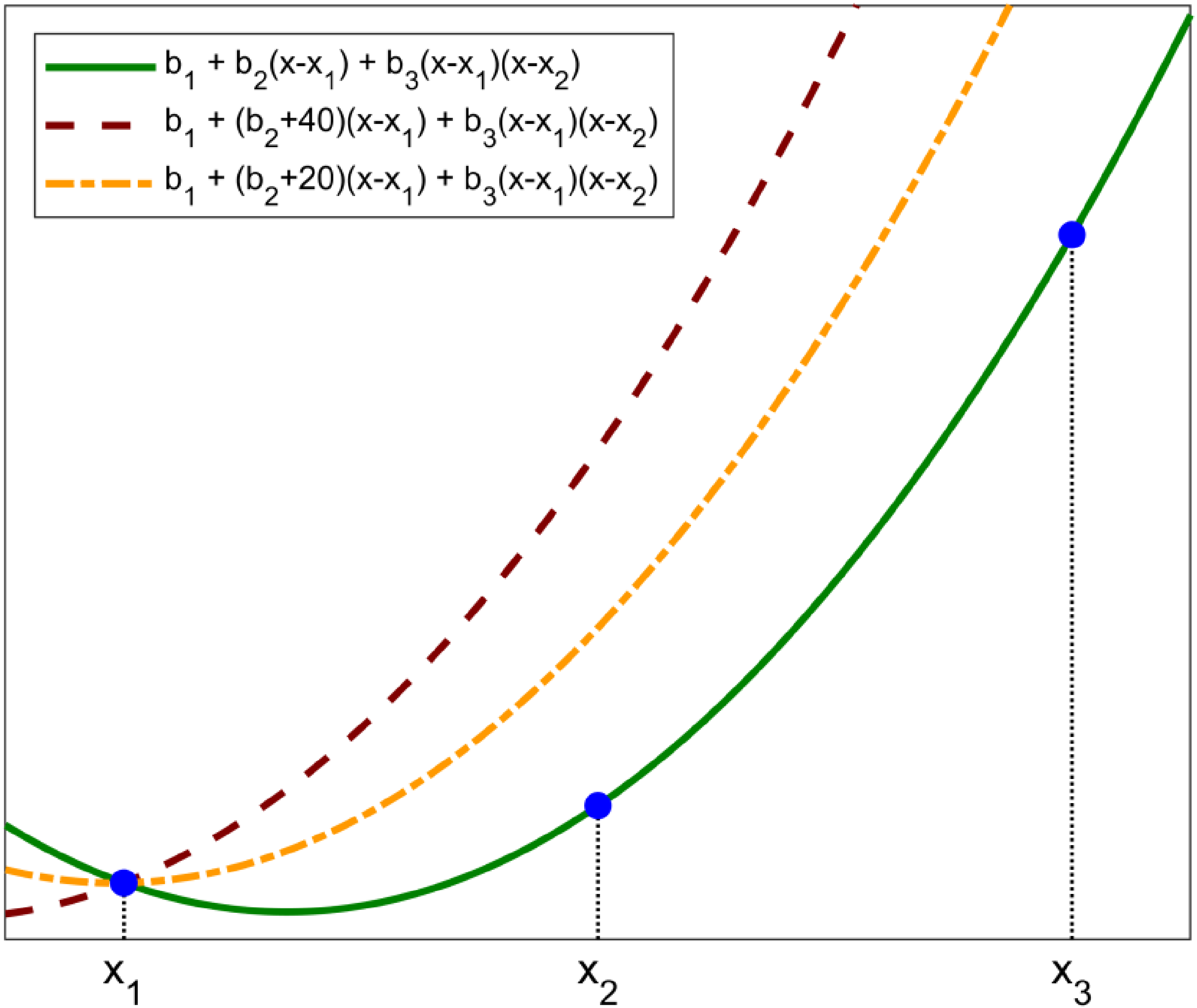
Rezultat – Njutnova kvadratna interpolacija

- $b_1 = y_1$ konstanta
- Definiše translaciju po y
- Ako b_1 uvećali za 30, kriva bi se translirala na gore za 30 po y



Rezultat – Njutnova kvadratna interpolacija

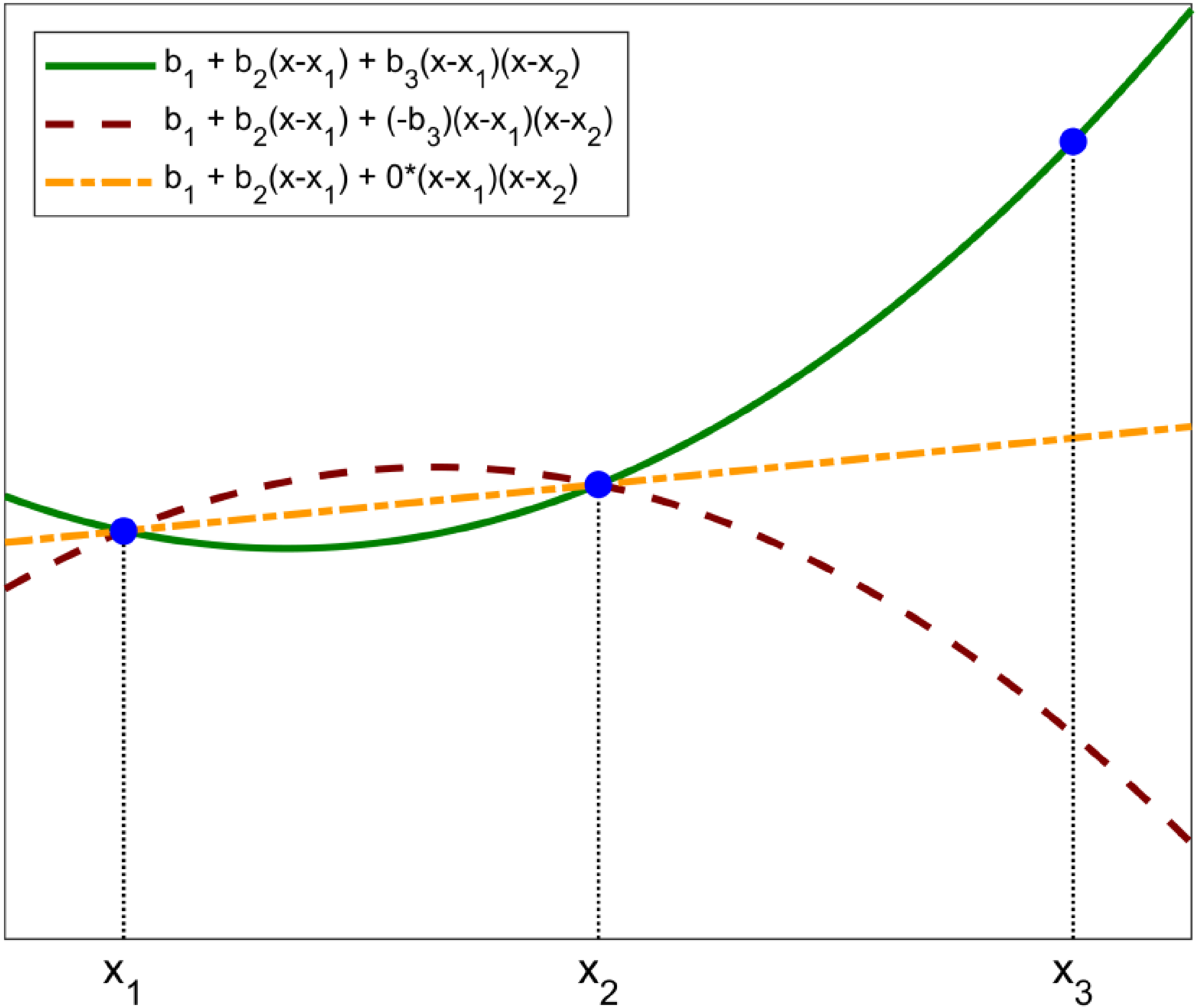
- $b_2 = \frac{y_2 - y_1}{x_2 - x_1}$
- Nagib



Rezultat – Njutnova kvadratna interpolacija

$$b_3 = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$$

- Zakrivljenost
- Ako bismo promenili znak b_3, dobili bismo krivu zakrivljenu u jednakoj meri na suprotnu stranu
- Ako bismo postavili $b_3 = 0$ dobili bismo pravu liniju



Rezultat – Njutnova kvadratna interpolacija

- Njutnova linearna interpolacija:

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

- Njutnova kvadratna interpolacija:

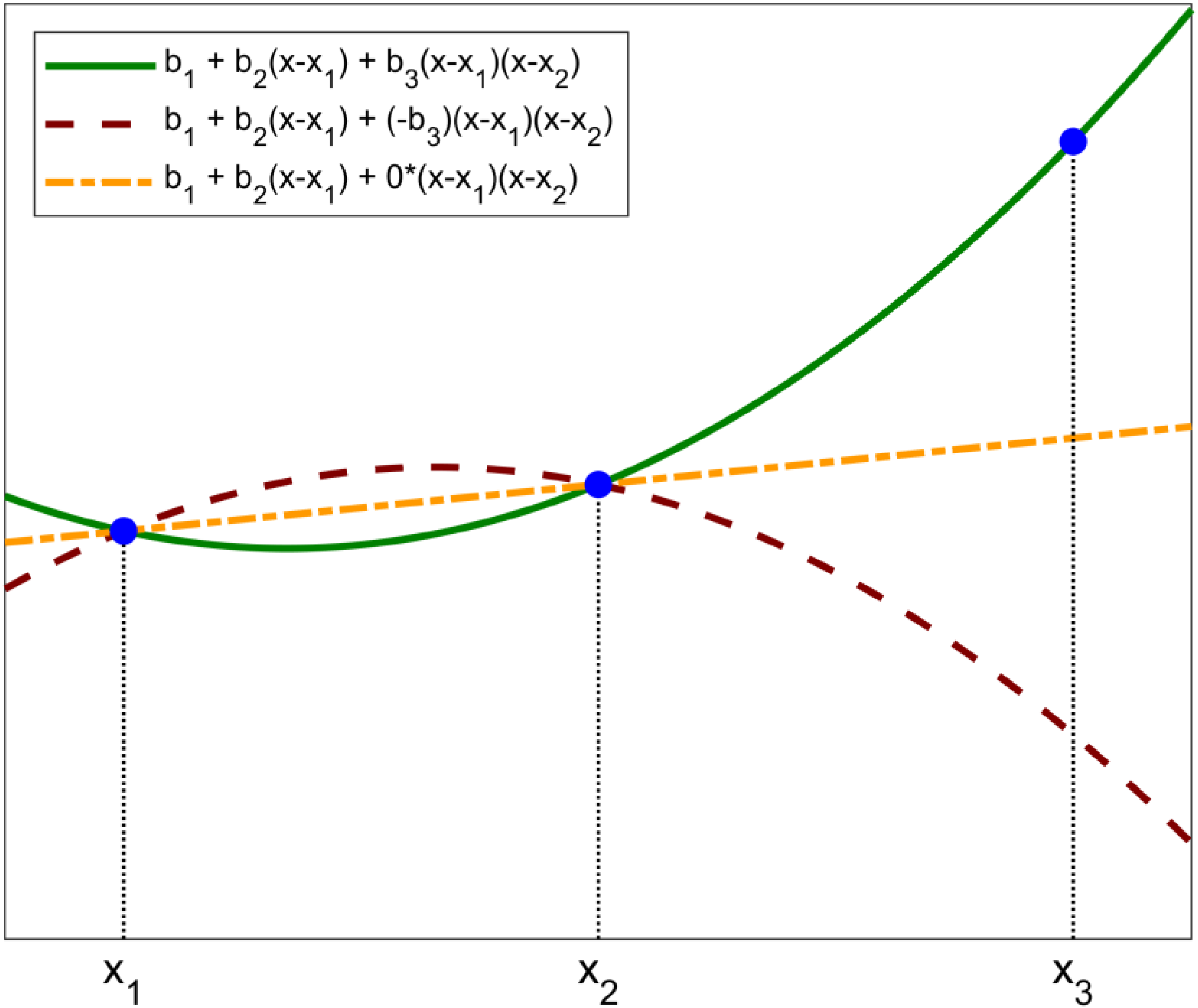
$$b_1 = y_1, b_2 = \frac{y_2 - y_1}{x_2 - x_1}, b_3 = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$$

- U oba slučaja vrednosti koeficijenta b_1 i b_2 su iste!
- Kvadratna interpolacija ne menja koeficijente dobijene linearnom interpolacijom, samo dodaje novi koeficijent b_3 koji definiše zakrivljenost krive

Rezultat – Njutnova kvadratna interpolacija

- Zelenom linijom je prikazan polinom dobijen na osnovu x_1, x_2 i x_3;

- Ako postavimo $b_3 = 0$ dobijamo pravu koju bismo dobili linearnom interpolacijom da su nam dostupne samo x_1 i x_2



Njutnova interpolacija

- Pošto koeficijenti nižeg stepena ostaju nepromenjeni kada povećavamo stepen polinoma, lako možemo dodati nove podatke
  - Nove tačke kroz koje kriva mora da prođe

- Ovo je velika prednost zapisa polinoma u Njutnovoj formi nad zapisom u „standardnoj“ formi

Primer: određivanje e^x u tački 2

- Koristimo dve tačke: $x_1 = 1$ i $x_2 = 3$:
  - $b_1 = y_1 = e^1 = 2.7183,$
  - $b_2 = \frac{y_2 - y_1}{x_2 - x_1} = \frac{e^3 - e^1}{3 - 1} = 8.6836$
  - $g_1(x) = b_1 + b_2(x - x_1)$
  - $g_1(2) = 2.7183 + 8.6836 \cdot (2 - 1) = 11.4019$

- Tačna vrednost: 7.3891. Relativna greška procene je:

$$|11.4019 - 7.3891|/7.3891 = 0.5431 \approx 54\%$$

Primer: određivanje e^x u tački 2

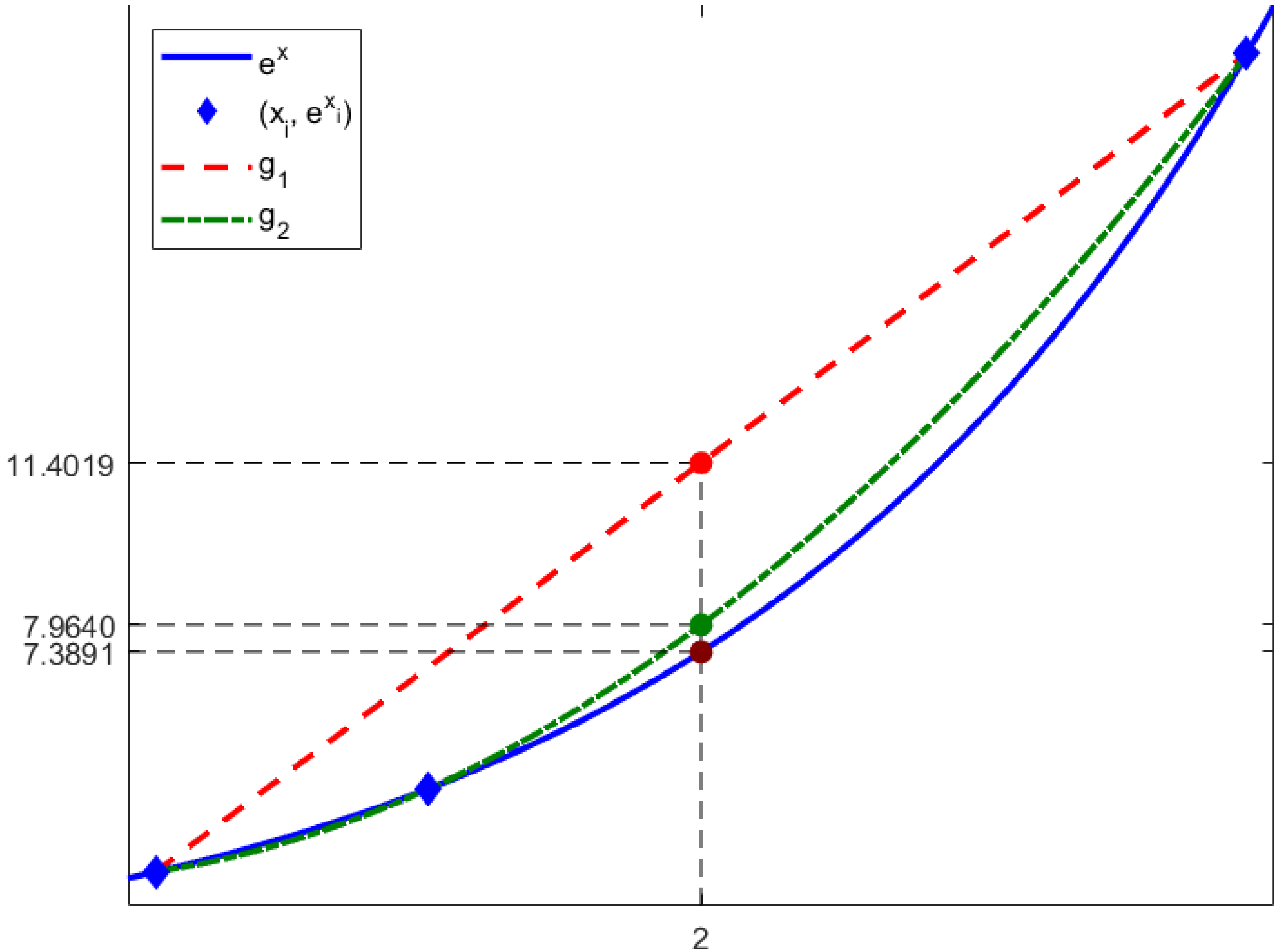
- Pobojšajmo procenu dodavanjem dodatne tačke $x_3 = 1.5$:
  - $b_3 = \frac{\frac{y_3-y_2}{x_3-x_2}-\frac{y_2-y_1}{x_2-x_1}}{x_3-x_1} = \frac{\frac{e^{1.5}-e^3}{1.5-3}-\frac{e^3-e^1}{3-1}}{1.5-1} = 3.4379$
  - $g_2(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2)$
  - $g_2(2) = 2.7183 + 8.6836 \cdot (2 - 1) + 3.4379 \cdot (2 - 1)(2 - 3) = 7.9640$

- Relativna greška u ovom slučaju je $|7.9640 - 7.3891|/7.3891 = 0.0778 \approx 7.8\%$

Tačnost procene

- Povećanjem stepena polinoma raste tačnost procene

- Slika je ilustracija prethodnog primera (određivanje e^x u 2)



Da smo koristili standardan zapis…

- U oba slučaja bismo dobili identične polinome, samo se postupak razlikuje
  - N tačaka jedinstveno određuje polinom stepena $N-1$ koji prolazi kroz njih

- Da bismo dobili polinom prvog stepena moramo rešiti sistem
$$g_1(1) = e^1 = p_1 \cdot 1 + p_2$$
$$g_2(3) = e^3 = p_1 \cdot 3 + p_2$$
- Ako dodamo još jednu tačku moramo rešiti novi sistem:
$$g_2(1) = e^1 = p_1 \cdot 1^2 + p_2 \cdot 1 + p_3$$
$$g_2(3) = e^3 = p_1 \cdot 3^2 + p_2 \cdot 3 + p_3$$
$$g_2(1.5) = e^{1.5} = p_1 \cdot 1.5^2 + p_2 \cdot 1.5 + p_3$$
- Potencijalno možemo imati problem i sa lošom uslovljenošću

Njutnova interpolacija opšta formula

- Opšta formula za Njutnov polinom:

$$\boxed{\begin{aligned} g_{n-1}(x) &= b_1 + b_2(x-x_1) + b_3(x-x_1)(x-x_2) + \cdots \\ &\quad + \cdots + b_n(x-x_1)(x-x_2)(x-x_3)\cdots(x-x_{n-1}) \end{aligned}}$$

$$\begin{aligned} b_1 &= f(x_1) \\ b_2 &= f[x_2, x_1] \\ b_3 &= f[x_3, x_2, x_1] \\ &\vdots \\ b_n &= f[x_n, x_{n-1}, \cdots, x_2, x_1] \end{aligned}$$

- Funkcije u uglastim zagradama su konačne razlike

Konačne razlike

- Konačna razlika nultog reda

$$f[x_i] = f(x_i)$$

- Konačna razlika prvog reda

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$

- Konačna razlika drugog reda

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k}$$

Konačne razlike

- Konačna razlika *n*-tog reda:

$$f[x_n, x_{n-1}, \ldots, x_2, x_1] = \frac{f[x_n, x_{n-1}, \ldots, x_3, x_2] - f[x_{n-1}, x_{n-2}, \ldots, x_2, x_1]}{x_n - x_1}$$

Iterativni algoritam:

1. **Izračunati sve konačne razlike prvog reda pomoću vrednosti funkcije $f(x_i)$.**
2. **Izračunati sve konačne razlike drugog reda koristeći razlike prvog reda.**
3. **Nastaviti proces do razlika n-tog reda.**

Konačne razlike

$$f_{n-1}(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2) + \cdots + b_n(x - x_1)\cdots(x - x_{n-1})$$

$$b_1 = f(x_1) = f[x_1]$$

$$b_2 = f[x_2, x_1] = \frac{f[x_2] - f[x_1]}{x_2 - x_1}$$

$$b_3 = f[x_3, x_2, x_1] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1}$$

$$b_4 = f[x_4, x_3, x_2, x_1] = \frac{f[x_4, x_3, x_2] - f[x_3, x_2, x_1]}{x_4 - x_1}$$

$$b_5 = f[x_5, x_4, x_3, x_2, x_1] = \frac{f[x_5, x_4, x_3, x_2] - f[x_4, x_3, x_2, x_1]}{x_5 - x_1}$$

- **Ne moramo da rešavamo sistem jednačina**
- **Važno je napomenuti da tačke $x_1, x_2, \dots, x_N$ ne moraju biti uniformno raspoređene (na jednakim rastojanjima)**
- **Takođe, ne mora da važi $x_1 < x_2 < \cdots x_N$ – tačke mogu biti raspoređene u bilo kom redosledu**

Konačne razlike

$$g_{n-1}(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2) + \cdots + b_n(x - x_1)\cdots(x - x_{n-1})$$

$$b_n = f\left[x_n, x_{n-1,\ldots}, x_2, x_1\right] = \frac{f\left[x_n, x_{n-1}, \cdots, x_3, x_2\right] - f\left[x_{n-1}, x_{n-2,} \cdots, x_2, x_1\right]}{x_n - x_1}$$

i	x_i	$y_i = f(x_i)$	$f\left[x_{i+1}, x_i\right]$ I red	$f\left[x_{i+2}, x_{i+1}, x_i\right]$ II red	$f\left[x_{i+3}, \cdots, x_i\right]$ III red	$f\left[x_{i+4}, \cdots, x_i\right]$ IV red
1	x_1	$f(x_1)$	$f\left[x_2, x_1\right]$	$f\left[x_3, x_2, x_1\right]$	$f\left[x_4, x_3, x_2, x_1\right]$	$f\left[x_5, x_4, x_3, x_2, x_1\right]$
2	x_2	$f(x_2)$	$f\left[x_3, x_2\right]$	$f\left[x_4, x_3, x_2\right]$	$f\left[x_5, x_4, x_3, x_2\right]$	$f\left[x_6, x_5, x_4, x_3, x_2\right]$
3	x_3	$f(x_3)$	$f\left[x_4, x_3\right]$	$f\left[x_5, x_4, x_3\right]$	$f\left[x_6, x_5, x_4, x_3\right]$	
4	x_4	$f(x_4)$	$f\left[x_5, x_4\right]$	$f\left[x_6, x_5, x_4\right]$		
5	x_5	$f(x_5)$	$f\left[x_6, x_5\right]$			
6	x_6	$f(x_6)$				

Korisimo prvi element svake kolone (= b_i) da bi procenili f(x) za nepoznate vrednosti

Primer e^x – kubna interpolacija

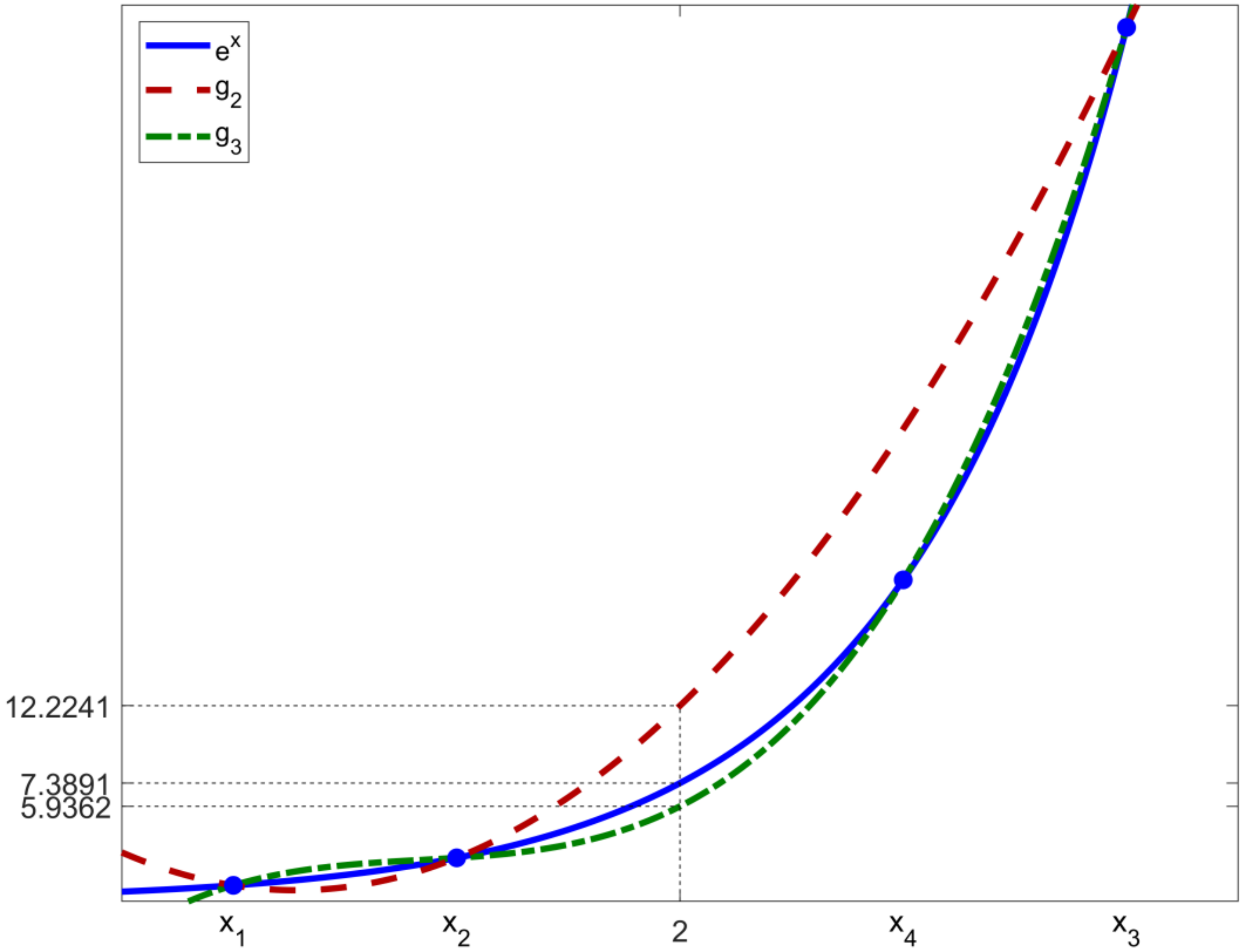
Proceniti $f(x) = e^x$ *u* $x = 2$ za $(x_1, x_2, x_3, x_4) = (0,1,4,3)$

i	x_i	$f(x_i)$	$f[x_{i+1}, x_i]$	$f[x_{i+2}, x_{i+1}, x_i]$	$f[x_{i+3}, x_{i+2}, x_{i+1}, x_i]$
1	0	1.000000	1.718282	3.893752	1.571970
			$\Delta x = 1 - 0$	$\Delta x = 4 - 0$	$\Delta x = 3 - 0$
2	1	2.718282	17.29329	8.609662	
			$\Delta x = 4 - 1$	$\Delta x = 3 - 1$	
3	4	54.59815	34.51261		
			$\Delta x = 3 - 4$		
4	3	20.08554			

$$f_2(2) = 1 + 1.718282x + 3.893752x(x-1) = 12.22407$$

$$f_3(2) = 1 + 1.718282x + 3.893752x(x-1) + 1.571970x(x-1)(x-4) = 5.936187$$

Primer e^x – kubna interpolacija



Polinomi u MATLAB-u

- Polinomi se mogu predstaviti kao vektori koeficijenata

$$p_1 = x - 2$$
$$p_2 = x^2 + 2x + 5$$

- Reprezentacija:
  - `p1 = [1, -2];`
  - `p2 = [1, 2, 5];`

- Množenje: `p = conv(p1, p2)`
  - Dobijamo `p=[1, 0, 1, -10]`, to jest, $p = x^3 + x - 10$

- Sabiranje:
  - `p1 = [0, 1, -2];` - vektori koje sabiramo moraju biti istih dimenzija
  - `p2 = [1, 2, 5];`
  - `p = p1 + p2`

Matlab kod – Njutnova interpolacija

```
function p=newton_interpolation(x,y)
n=length(x);
p=zeros(1,n);
D=zeros(n,n); %D je matrica konacnih razlika kao npr. dole na slajdu
D(:,1)=y';
for j=2:n
    for i=1:n-j+1
        D(i,j)=(D(i+1,j-1)-D(i,j-1))/(x(i+j-1)-x(i));
    end
end
for i=1:n
    pom=1;
    for j=2:i
        pom=conv(pom,[1 -x(j-1)]);
    end
    p = p + [zeros(1,n-i) pom*D(1,i)];
end
```

$$\boxed{f[x_n, x_{n-1}, \ldots, x_2, x_1] = \frac{f[x_n, x_{n-1}, \ldots, x_3, x_2] - f[x_{n-1}, x_{n-2}, \ldots, x_2, x_1]}{x_n - x_1}}$$

$$p_{n-1}(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2) + \cdots + b_n(x - x_1)\cdots(x - x_{n-1})$$

i	x_i	$f(x_i)$	$f[x_{i+1}, x_i]$	$f[x_{i+2}, x_{i+1}, x_i]$	$f[x_{i+3}, x_{i+2}, x_{i+1}, x_i]$
1	0	1.000000	1.718282	3.893752	1.571970
			$\Delta x = 1 - 0$	$\Delta x = 4 - 0$	$\Delta x = 3 - 0$
2	1	2.718282	17.29329	8.609662	
			$\Delta x = 4 - 1$	$\Delta x = 3 - 1$	
3	4	54.59815	34.51261		
			$\Delta x = 3 - 4$		
4	3	20.08554			

Primer polinoma za e^x

```
>> x
x =
     0     1     4     3
>> y=exp(x)
y =
    1.0000    2.7183   54.5982   20.0855
>> p=newton_interpolation(x,y)
p =

    1.5720   -3.9661    4.1124    1.0000
>> a=polyval(p,2)
a =
    5.9362
```

polyval(p,x) – izračunavanje vrednosti polinoma p u tački x tj. p(x) – u ovom konkretnom primeru p(2)

Uporedite rezultat sa ručno dobijenim primerom ranije:

$$f_3(2) = 1 + 1.718282x + 3.893752x(x-1) + 1.571970x(x-1)(x-4) = 5.936187$$

Napomena: polinom dobijen u Matlabu ima grupisane članove po stepenima, dok ručno određeni nema. Zato se na prvi pogled razlikuju iako su u suštini isti polinom.

Lagranžov interpolacioni polinom

- Drugačiji način za formiranje interpolacionog polinoma
- Podsetimo se da za *N* tačaka postoji jednistveni polinon stepena *N*-1 koji prolazi kroz njih
- Ovo znači da, za zadatih N tačaka, govorimo o Njutnovom i Lagranžovom obliku *istog* polinoma

Lagranžov interpolacioni polinom

- Lagranžov polinom ima sledeći oblik:

$$g_{N-1}(x) = L_1(x) f(x_1) + L_2(x) f(x_2) + \cdots + L_N(x) f(x_N) = \sum_{i=1}^{N} L_i(x) f(x_i)$$

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{N} \frac{x - x_j}{x_i - x_j} = \frac{P_i(x)}{P_i(x_i)}$$

$$= \frac{(x - x_1)(x - x_2)\cdots\cdots(x - x_{i-1})(x - x_{i+1})\cdots\cdots(x - x_N)}{(x_i - x_1)(x_i - x_2)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_N)}$$

Lagranžov interpolacioni polinom

- Lagranžov polinom prvog reda

$$f_1(x) = L_1 f(x_1) + L_2(x) f(x_2) = \frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_1}{x_2 - x_1} f(x_2)$$

- Lagranžov polinom drugog reda

$$f_2(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} f(x_1) + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} f(x_2) + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} f(x_3)$$

- Lagranžov polinom trećeg reda

$$f_4(x) = \frac{(x - x_2)(x - x_3)(x - x_4)}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)} f(x_1) + \frac{(x - x_1)(x - x_3)(x - x_4)}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)} f(x_2)$$
$$+ \frac{(x - x_1)(x - x_2)(x - x_4)}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)} f(x_3) + \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)} f(x_4)$$

Lagranžov interpolacioni polinom

$$g_{N-1}(x) = \sum_{i=1}^{N} L_i(x) f(x_i)\,, L_i(x) = \prod_{j=1, j\neq i}^{N} \frac{x - x_j}{x_i - x_j}$$

- Primetite da za svaku tačku x_j iz skupa podataka važi:
  - $L_i(x_j) = \begin{cases} j = i; \quad L_i(x_j) = L_i(x_i) = \frac{(x_i - x_1)(x_i - x_2)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_N)}{(x_i - x_1)(x_i - x_2)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_N)} = 1 \\ j \neq i; \quad L_i(x_j) = \frac{(x_j - x_1)\cdots(\boldsymbol{x_j - x_j})\cdots(x_j - x_N)}{(x_i - x_1)(x_i - x_2)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_N)} = 0 \end{cases}$
  - Tj. $g_{N-1}(x_j) = L_1(x_j) f(x_1) + \cdots + L_j(x_j) f(x_j) + \cdots + L_N(x_j) f(x_N) = 0 \cdot f(x_1) + \cdots + 1 \cdot f(x_j) + \cdots + 0 \cdot f(x_N) = f(x_j) = y_j$
  - Dakle, $g_{N-1}(x_j) = y_j$, tj. možemo videti da interpolacioni polinom prolazi tačno kroz svaku zadatu tačku

Polinom drugog reda pojašnjenje

$$f_2(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} f(x_1) + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} f(x_2) + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} f(x_3)$$

$$g_2(x_1) = \frac{(x_1-x_2)(x_1-x_3)}{(x_1-x_2)(x_1-x_3)} f(x_1) + \cancel{\frac{(x_1-x_1)(x_1-x_3)}{(x_2-x_1)(x_2-x_3)}} f(x_2) + \cancel{\frac{(x_1-x_1)(x_1-x_2)}{(x_3-x_1)(x_3-x_2)}} f(x_3) = f(x_1)$$

$$g_2(x_2) = \cancel{\frac{(x_2-x_2)(x_2-x_3)}{(x_1-x_2)(x_1-x_3)}} f(x_1) + \frac{(x_2-x_1)(x_2-x_3)}{(x_2-x_1)(x_2-x_3)} f(x_2) + \cancel{\frac{(x_2-x_1)(x_2-x_2)}{(x_3-x_1)(x_3-x_2)}} f(x_3) = f(x_2)$$

$$g_2(x_3) = \cancel{\frac{(x_3-x_2)(x_3-x_3)}{(x_1-x_2)(x_1-x_3)}} f(x_1) + \cancel{\frac{(x_3-x_1)(x_3-x_3)}{(x_2-x_1)(x_2-x_3)}} f(x_2) + \frac{(x_3-x_1)(x_3-x_2)}{(x_3-x_1)(x_3-x_2)} f(x_3) = f(x_3)$$

Lagranžov interpolacioni polinom

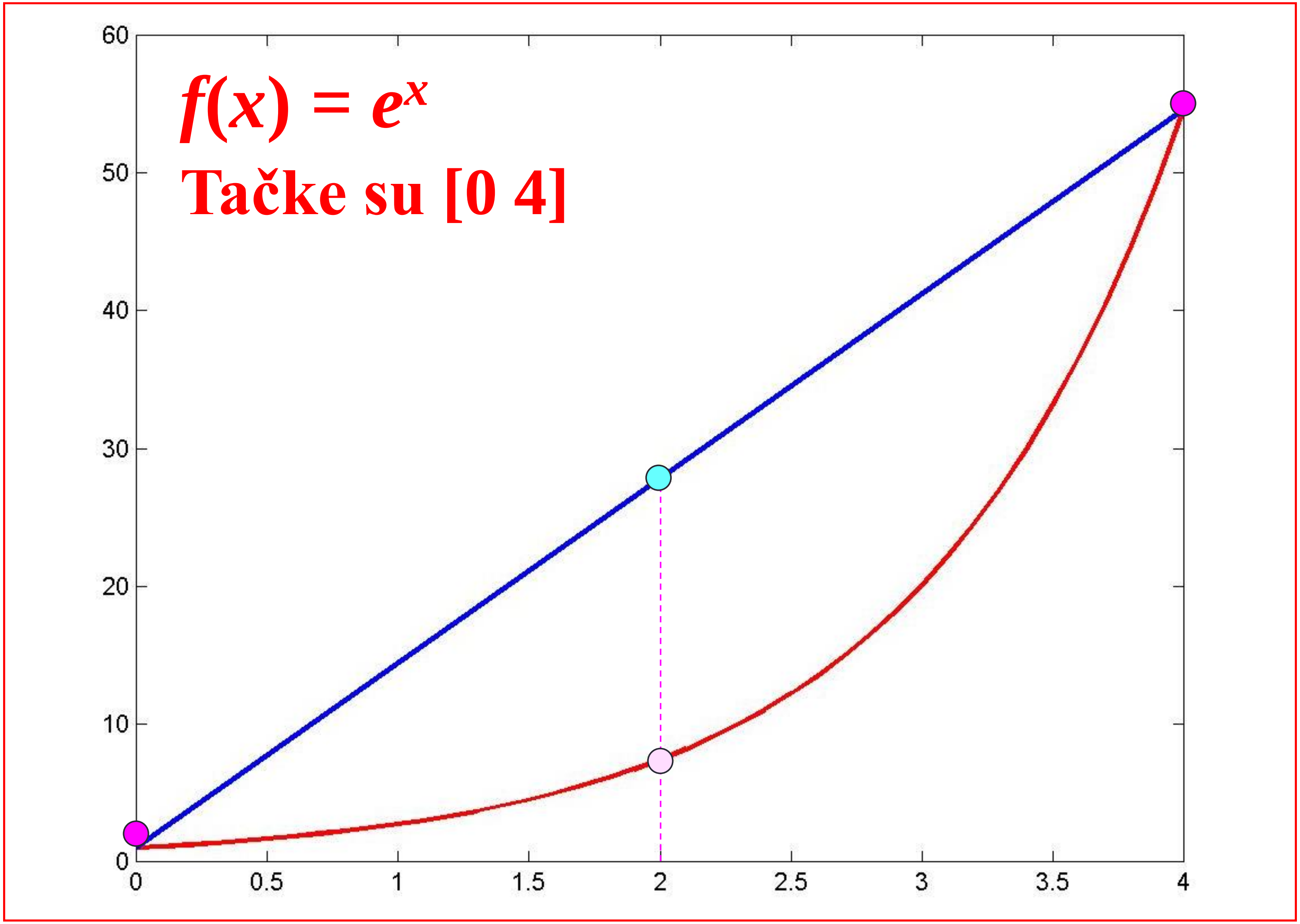
$$L_i(x) = \prod_{j=1, j \neq i}^{N} \frac{x - x_j}{x_i - x_j}$$

- $L_i(x)$ zavisi isključivo od vrednosti x tačaka u kojima se meri funkcija (ne i od vrednosti same funkcije u tim tačkama)

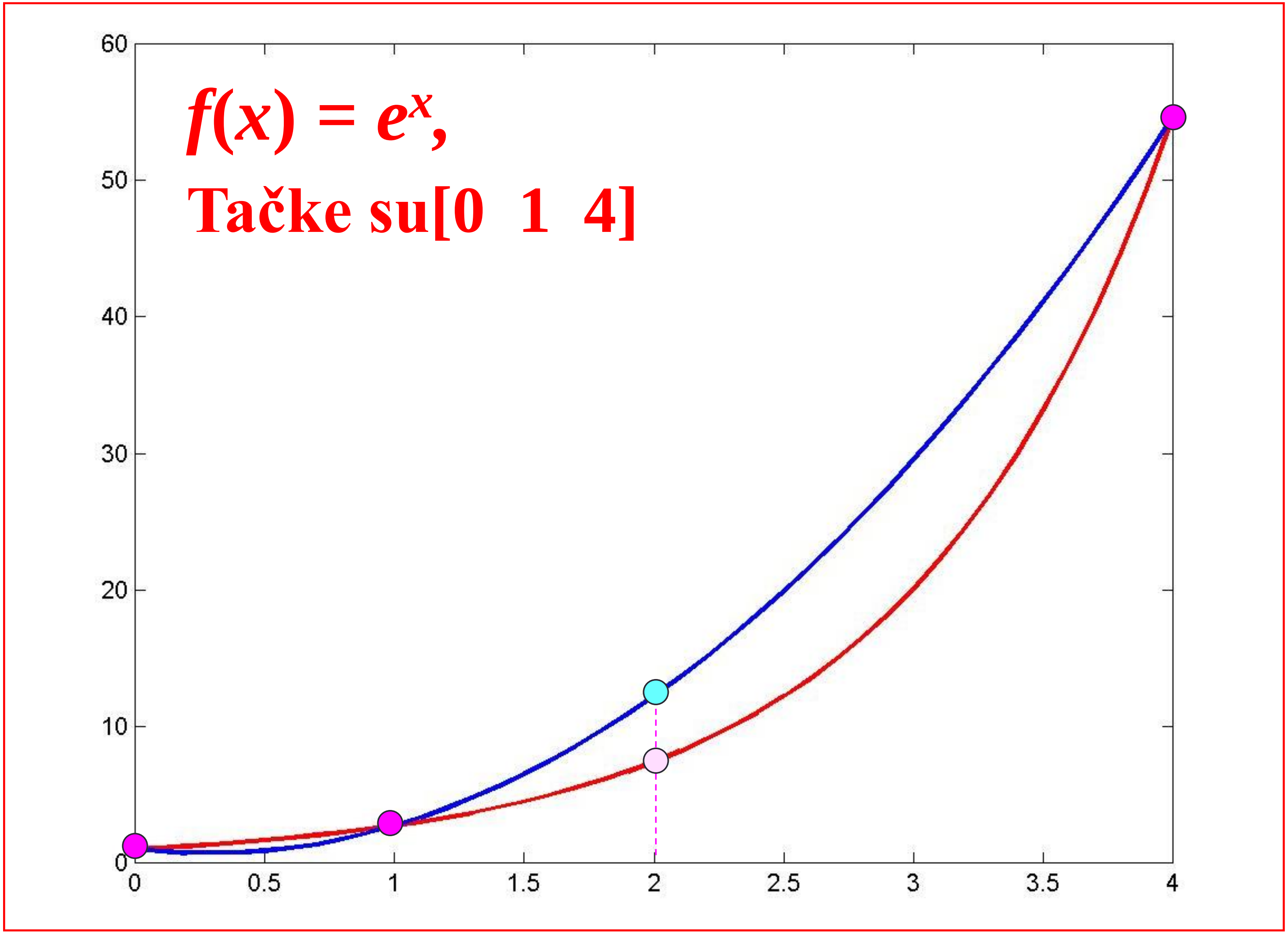
- Odatle sledi da je Lagranžov oblik polinoma pogodan za slučajeve kada imamo više različitih skupova y_i za iste x_i
  - Zgodno kada imamo veličinu koja se meri uvek u istim trenucima x_i

- Za razliku od Njutnovog oblika polinoma, Lagranžov oblik polinoma nije toliko pogodan za dodavanje novih tačaka

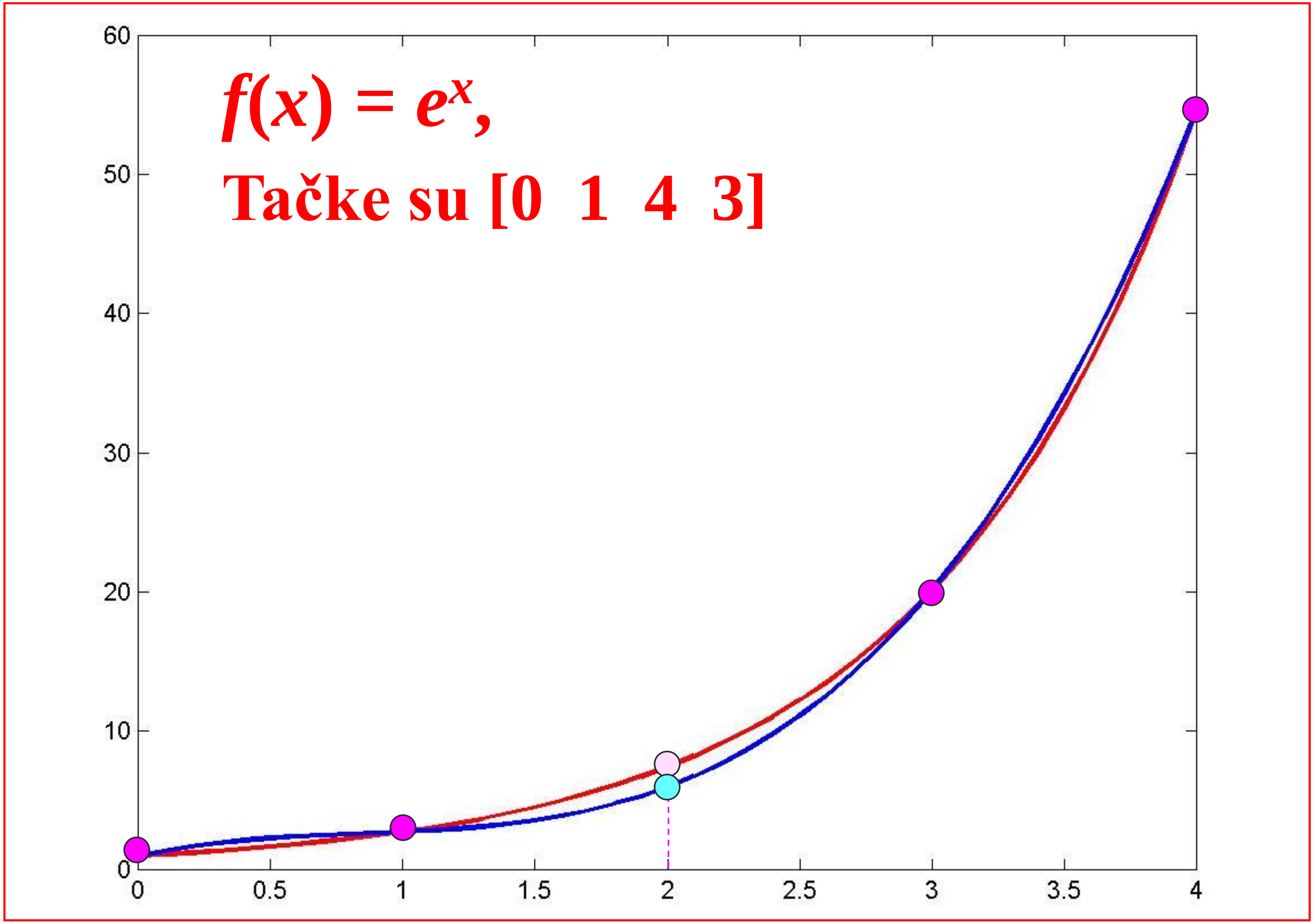
Polinom prvog stepena



Polinom drugog stepena



Polinom trećeg stepena



Primer

- Proceniti e^x u $x = 2$ koristeći $(x_0,x_1,x_2,x_3) = (0,1,4,3)$
- polinom drugog stepena: $(x_1, x_2, x_3) = (0,1,4)$

$$f_2(x) = \frac{(x-1)(x-4)}{(0-1)(0-4)} f(0) + \frac{(x-0)(x-4)}{(1-0)(1-4)} f(1) + \frac{(x-0)(x-1)}{(4-0)(4-1)} f(4)$$

$$f_2(2) = \frac{(2-1)(2-4)}{(0-1)(0-4)}(1.0) + \frac{(2-0)(2-4)}{(1-0)(1-4)}(1.71828) + \frac{(2-0)(2-1)}{(4-0)(4-1)}(54.5982) = 12.224067$$

Primer

- polinom trećeg stepena: $(x_1,x_2,x_3,x_4) = (0,1,4,3)$

$$f_3(x) = \frac{(x-1)(x-4)(x-3)}{(0-1)(0-4)(0-3)} f(0) + \frac{(x-0)(x-4)(x-3)}{(1-0)(1-4)(1-3)} f(1)$$

$$+ \frac{(x-0)(x-1)(x-3)}{(4-0)(4-1)(4-3)} f(4) + \frac{(x-0)(x-1)(x-4)}{(3-0)(3-1)(3-4)} f(3)$$

$$f_3(2) = \frac{2}{-12}(1.0) + \frac{4}{6}(2.71828) + \frac{-2}{12}(54.5982) + \frac{-4}{-6}(20.08554) = 5.936187$$

Matlab kod

```
function p=linterp(x,y)
n=length(x);
p=0;
for i=1:n
    L=1;
    for j=1:n
        if(i~=j)
            L=conv(L,[1 -x(j)])/(x(i)-x(j));
        end
    end
    p = p + y(i)*L;
end
```

$$p_{n-1}(x) = \sum_{i=1}^{n} L_i(x) f(x_i)$$

$$L_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Primer polinoma za e^x

```
>> x
x =
    0     1     4     3
>> y=exp(x)
y =
   1.0000    2.7183   54.5982   20.0855
>> p=linterp(x,y)
p =

   1.5720   -3.9661    4.1124    1.0000
>> a=polyval(p,2)
a =
   5.9362
```

polyval(p,x) – izračunavanje vrednosti polinoma p u tački x tj. p(x) – u ovom konkretnom primeru p(2)

Uporedite ga sa rezultatom ručno dobijenim primerom ranije:

$$f_3(2) = \frac{2}{-12}(1.0) + \frac{4}{6}(2.71828) + \frac{-2}{12}(54.5982) + \frac{-4}{-6}(20.08554) = 5.936187$$

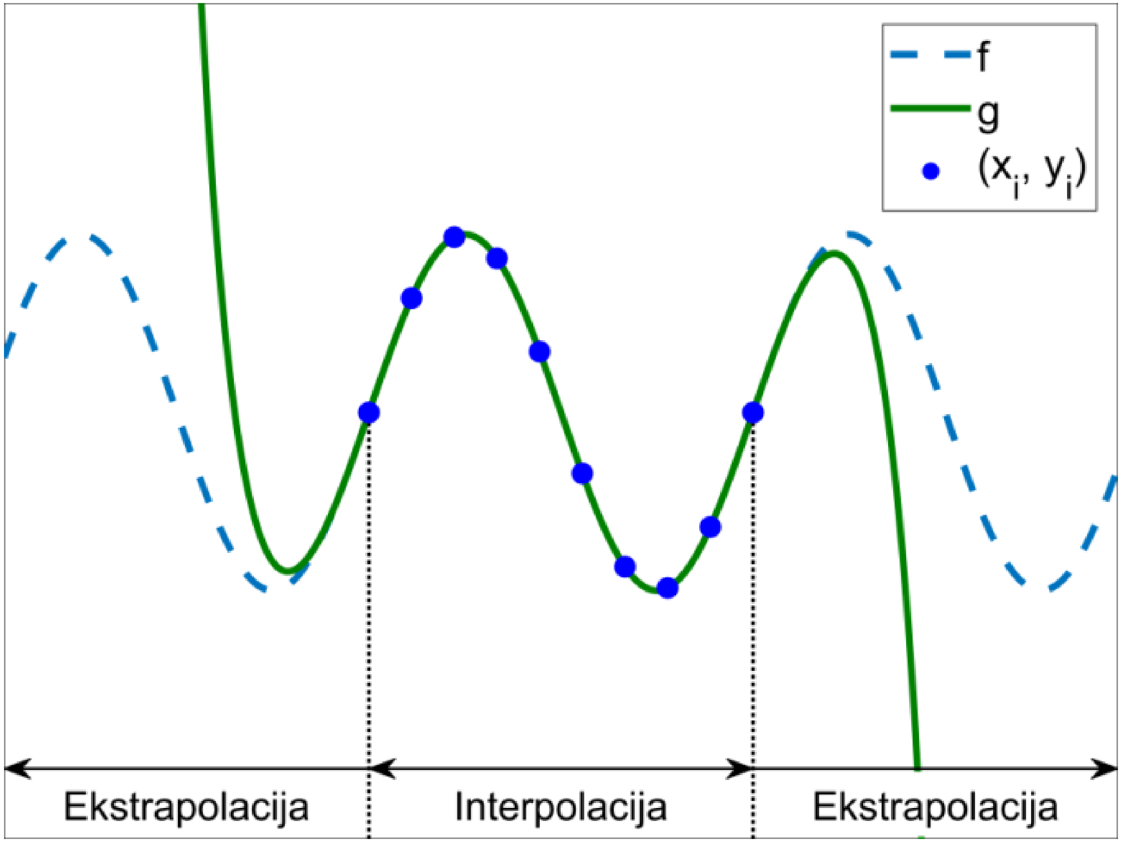
Inverzna interpolacija

- Do sada, za date *x* i *f(x)* interpolacija nam daje mogućnost da izračunamo *f(x)* za novo *x*
- Šta bi bilo ako želimo da odredimo u kojoj tački x funkcija f(x) ima vrednost *k*?

1. Zameniti *x* i *f*(*x*) i uraditi interpolaciju. Međutim, razmak između y je obično veoma ne-uniforman (za razliku od x) što rezultuje u oscilacijama u interpolacionom polinomu

2. Interpolirati f(x) u tačkama x – dobijamo g(x). Upotrebiti metode za određivanje nula funkcija da bi pronašli x takvo da je g(x)=k

Ekstrapolacija

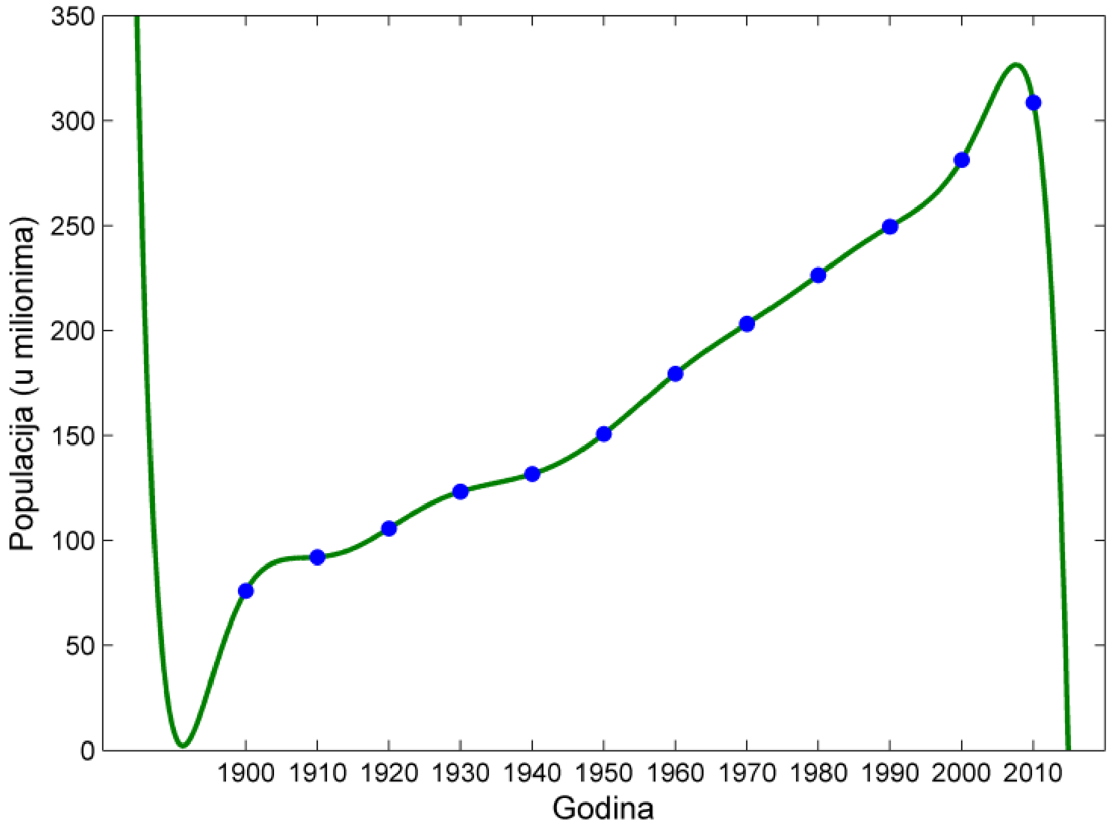


- I kod interpolacije i kod ekstrapolacije funkcija *g* je ista
- Razlikuju se tačke u kojima aproksimiramo *f*

- Trebalo bi izbegavati ekstrapolaciju ukoliko je to moguće

Primer problema ekstrapolacije

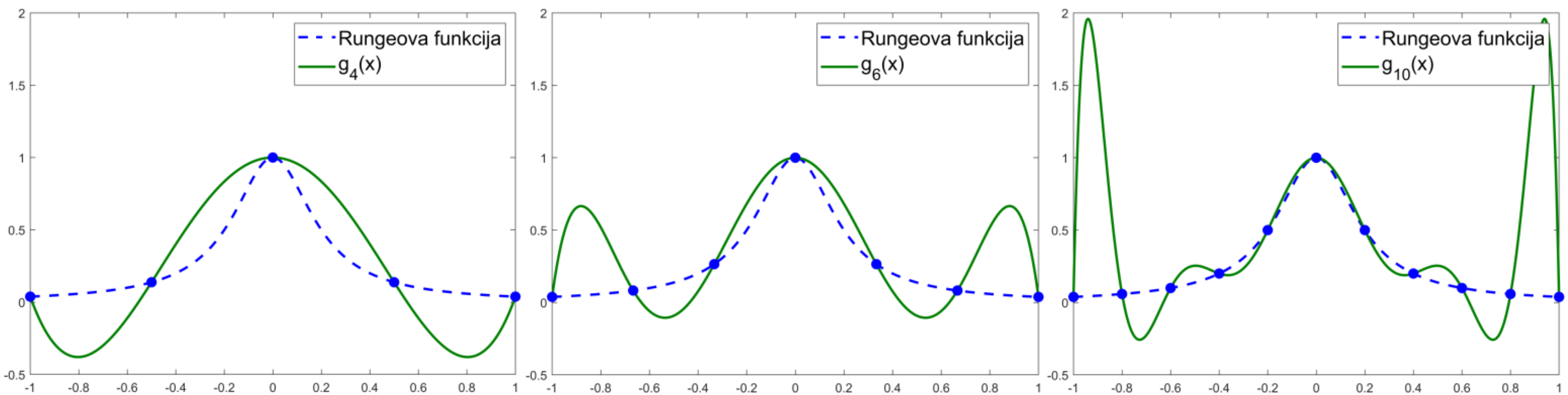
- Ekstrapolacija rasta populacije SAD
  - Podaci https://www.mathworks.com/company/newsletters/articles/fitting-and-extrapolating-u-s-census-data.html



- predikcije dobijene interpolacijom su prilično razumne
- Ekstrapolacija: populacija SAD će pasti na nulu posle 2010. godine...

Problemi sa interpolacijom – oscilacije polinoma velikog stepena

- *Rungeov fenomen* označava problem oscilovanja interpolacionog polinoma na ivicama datog intervala prilikom korišćenja polinoma *velikog* stepena

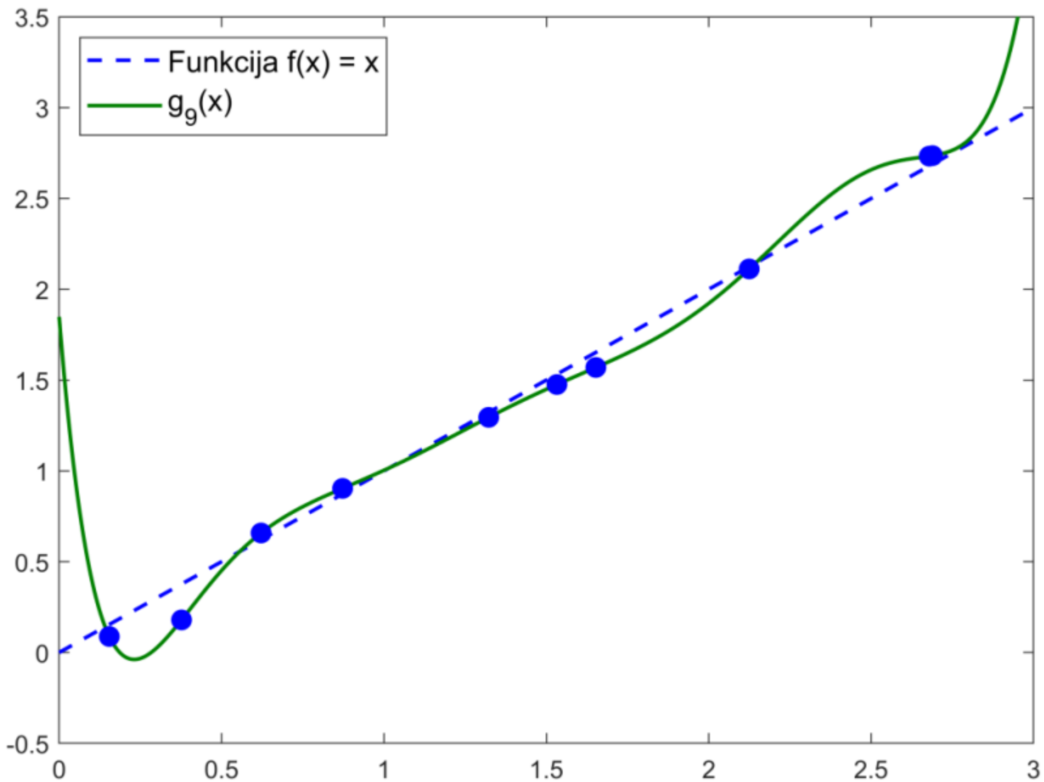


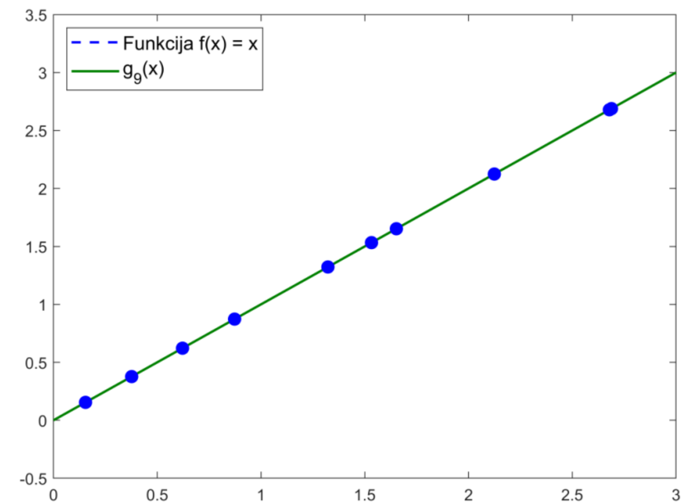
Runge-ova funkcija: $f(x) = \frac{1}{1+25\cdot x^2}$

Korišćenje polinoma većeg stepena nije pobojšalo aproksimacuju

Problemi sa interpolacijom – oscilacije polinoma velikog stepena

- Aproksimacija u prisustvu šuma
  - Uzorkovali smo 10 tačaka i dodali im mali šum





Bez šuma, ovaj problem se ne ispoljava

Problemi sa interpolacijom – oscilacije polinoma velikog stepena

- Iako postoje određene situacije u kojima su nam polinomi velikog stepena od koristi, u većini primena ih izbegavamo

- Polinomi nižeg stepena obično mogu efektivno da opišu trend koji postoji u krivoj, a da ne uvedu problem velikih oscilacija

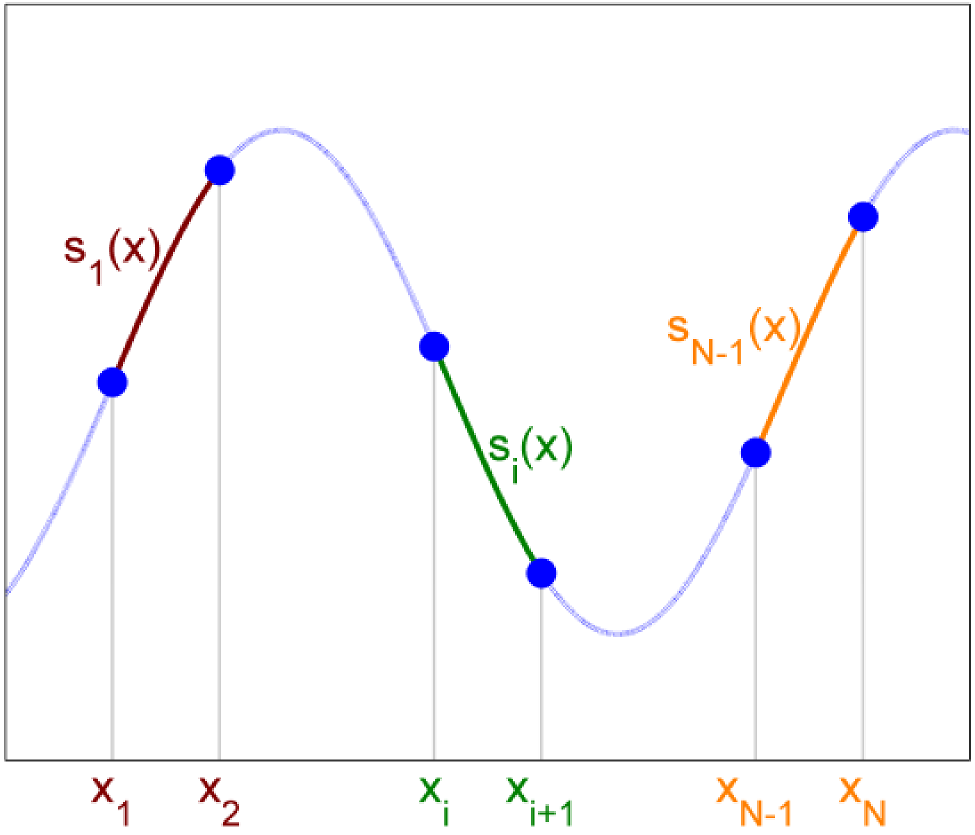
Interpolacija Splajnom

- Videli smo kako možemo iskoristiti N tačaka da odredimo polinom stepena $N-1$ koji prolazi kroz njih

- Ali, ako je *N* veliko rezultujući polinom je velikog stepena

- Uvideli smo da takvi polinomi mogu dovesti do loših rezultata zbog velikih oscilacija i grešaka zaokruživanja

Interpolacija Splajnom



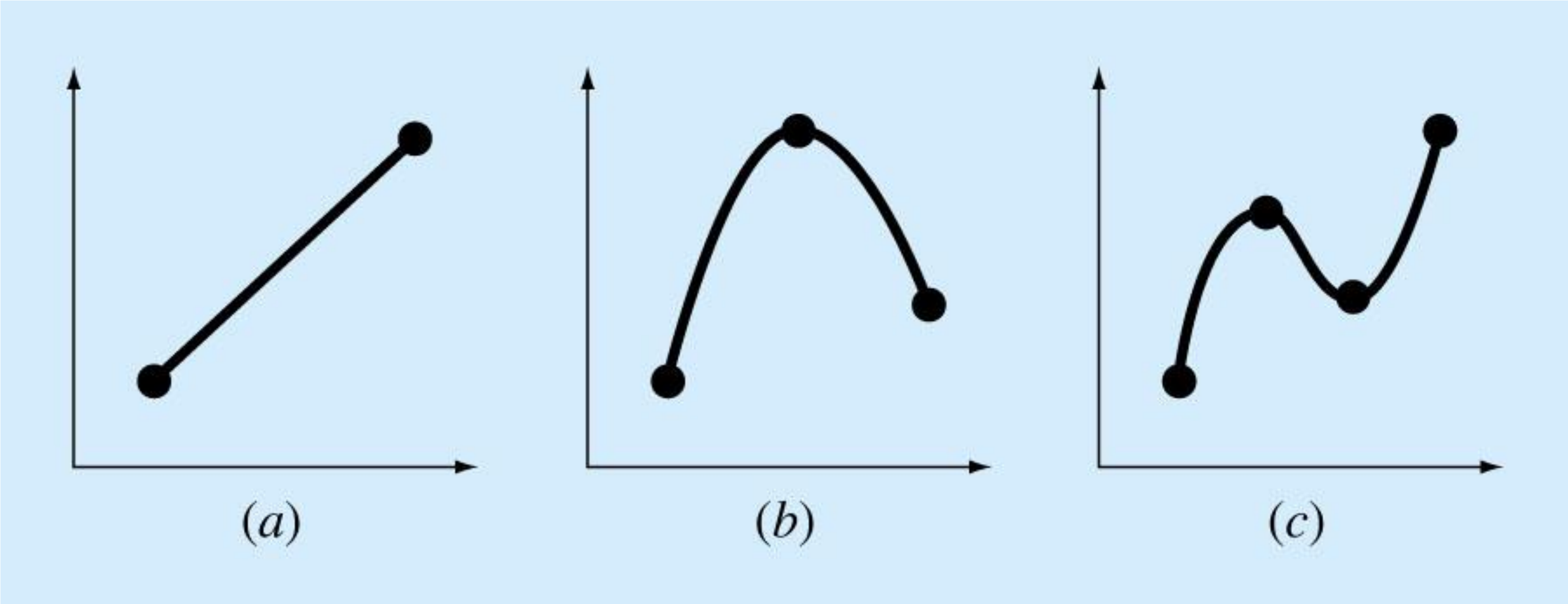
1. Delimo interval $[x_1, x_N]$ na kome su nam zadate vrednosti funkcije na $N-1$ podintervala $[x_i, x_{i+1}], i \in \{1, \dots, N-1\}$

2. Na svakom podintervalu $[x_i, x_{i+1}]$ određujemo zaseban interpolacioni polinom (nižeg stepena)

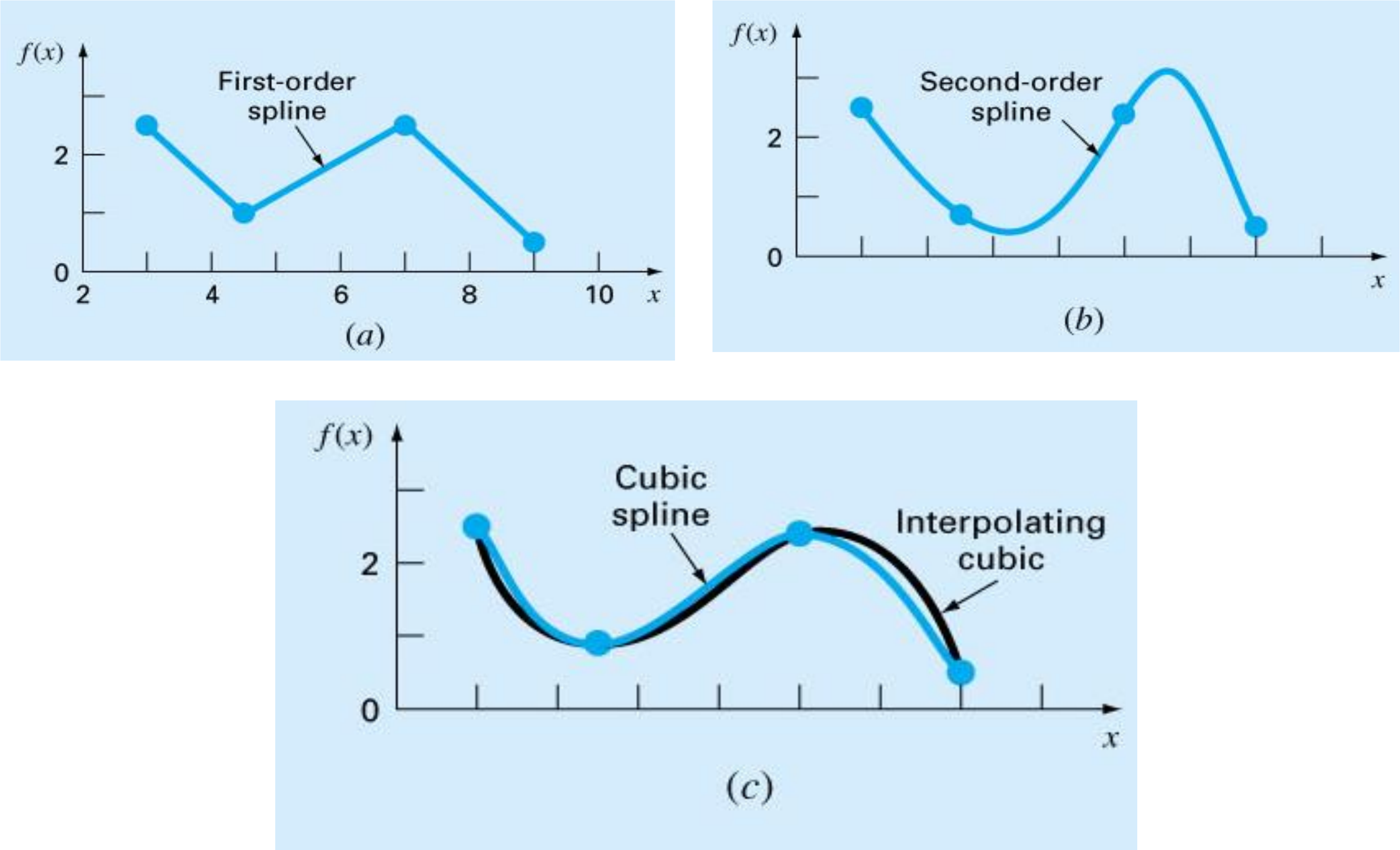
Splajn - ideja

- Koristimo više polinoma nižeg stepena da interpoliramo date tačke
- U zavisnosti od stepena polinoma koji koristimo na podintervalima $[x_i, x_{i+1}]$ imamo:

Linearni splajn (polinom 1. reda) | Kvadratni splajn (polinom 2. reda) | Kubni splajn (polinom 3. reda)



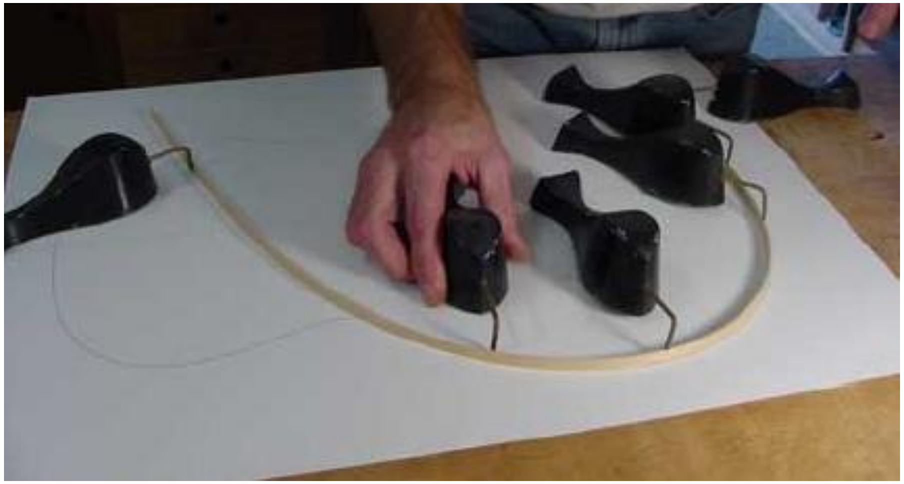
Primeri

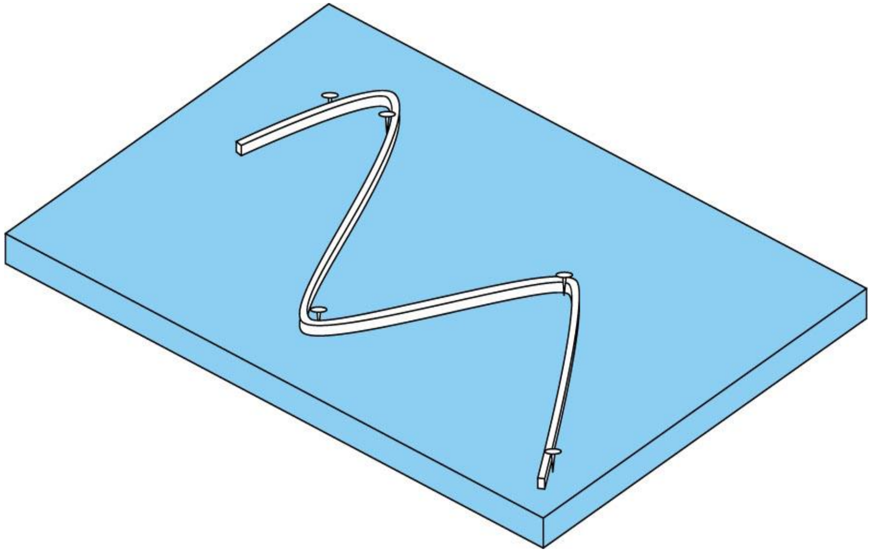


f(x)
First-order
spline
2
0
2
4
6
8
10
x
(a)
f(x)
Second-order
spline
2
0
x
(b)
f(x)
Cubic
spline
Interpolating
cubic
2
0
x
(c)

Poreklo splajna

- Naziv i ideja za splajn dolazi iz metode za crtanje krive na papiru, kod koje se savitljiva traka od plastike ili drveta (splajn) savija u željeni oblik pomoću klinova (tačaka).





Linearni splajn

- Interpolaciju vršimo tako što između svake dve date tačke povlačimo pravu

- Između dve poznate tačke (x_i, y_i) i (x_{i+1}, y_{i+1}) na intervalu $[x_i, x_{i+1}]$ povlačimo pravu liniju $s_i(x)$ oblika:

$$s_i(x) = p_1 \cdot x + p_2$$

- Mora da važi $y_i = p_1 x_i + p_2$ i $y_{i+1} = p_1 x_{i+1} + p_2$
- Odatle sledi:

$$s_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i), x \in [x_i, x_{i+1}]$$

Linearni splajn - primer

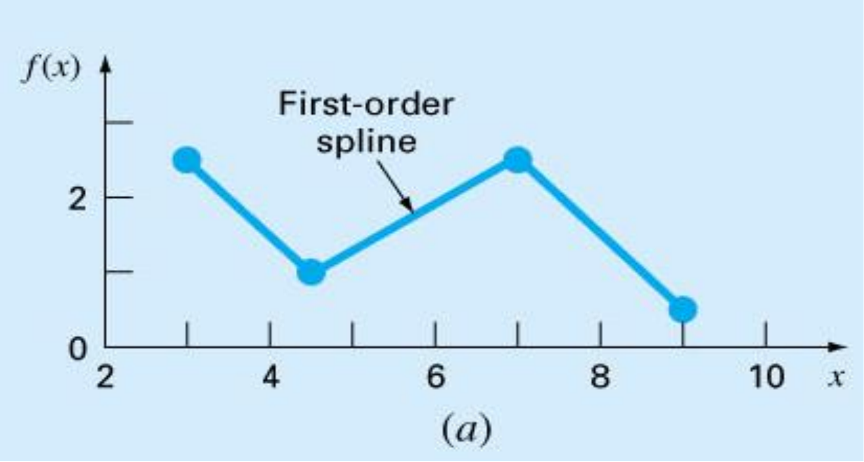
Kreiramo linearni interpolacioni splajn za date tačke. Procenjujemo vrednost u x = 5.

x	$f(x)$
3.0	2.5
4.5	**1.0**
7.0	**2.5**
9.0	0.5

$$m = \frac{2.5-1}{7-4.5} = 0.6$$

$$s_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i)$$

$$f(5) = f(4.5) + m(5-4.5)$$
$$= 1.0 + 0.6 \times 0.5$$
$$= 1.3$$



Linearni splajn mane

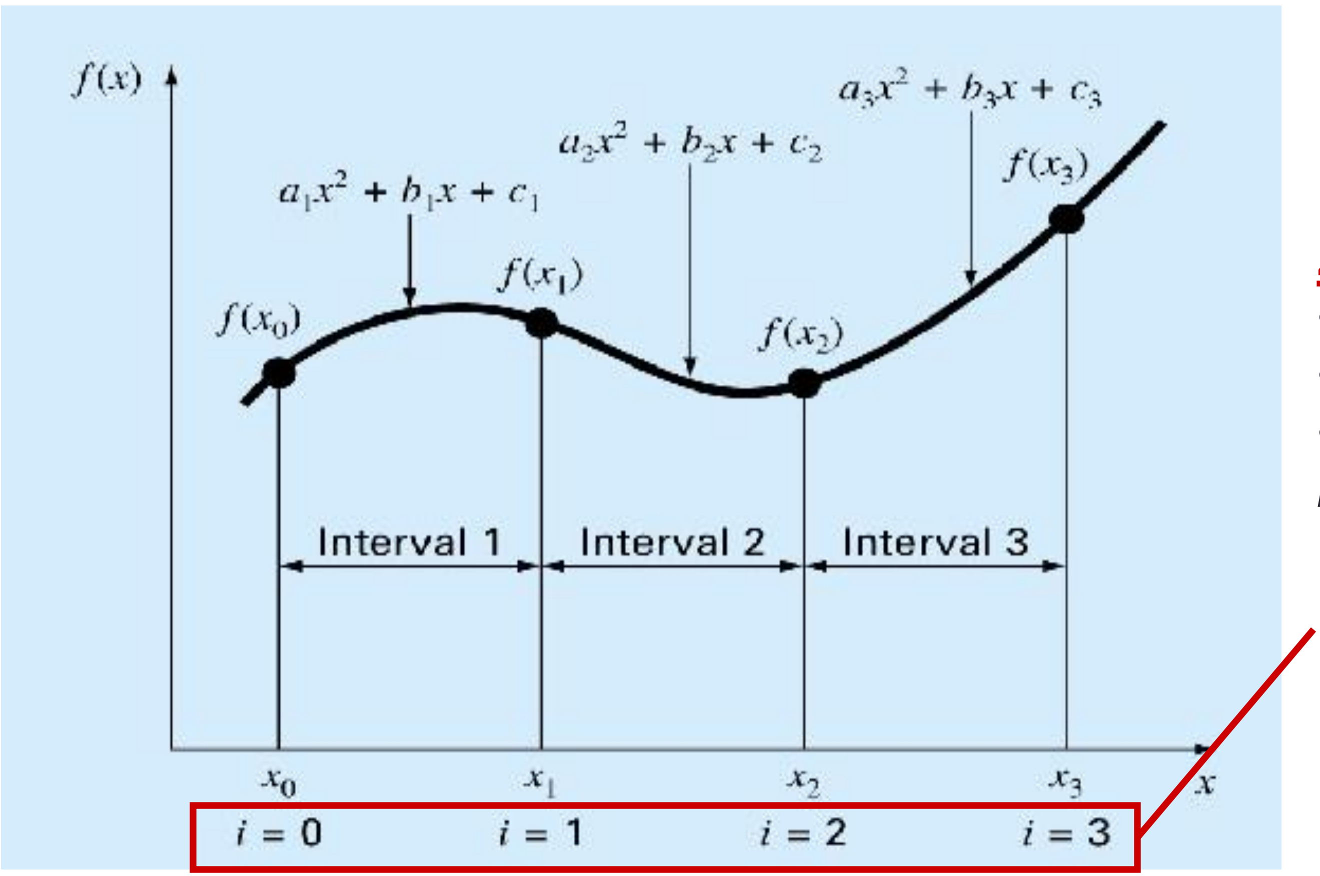
- Nije gladak: imamo nagle promene vrednosti prvog izvoda u tačkama u kojima se dva splajna spajaju (čvorovi)

- Prvi izvod splajna ima prekide u tačkama spoja
  - Nezgodno za mnoge kasnije analize koje bismo možda želeli da sprovedemo nad aproksimiranom funkcijom (na primer, pronalaženje minimuma funkcije)

- Ako koristimo splajnove većeg stepena možemo da obezbedimo neprekidnost prvog izvoda

Kvadratni splajn

- Cilj: odrediti polinom drugog stepena između svake dve date tačke $s_i(x) = a_i x^2 + b_i x + c_i$
- Uvodimo nove termine: “unutrašnji čvor” i “kranji čvor”.



Za *N* tačaka:

- *i = (1, 2, …N),*
- ***N-1** podintervala,*
- ***3(N-1)** nepoznatih (a_i, b_i i c_i)*

Kvadratni splajn

- Vrednosti splajnova u unutrašnjim čvorovima moraju biti jednake. **2**(**n-2**).

$$a_{i-1}x_{i-1}^{\;2}+b_{i-1}x_{i-1}+c_{i-1}=f(x_{i-1})\quad i=2,3,4,\ldots,n$$

$$a_{i}x_{i-1}^{\;2}+b_{i}x_{i-1}+c_{i}=f(x_{i-1})\quad i=2,3,4,\ldots,n$$

- Prvi i poslednji splajn moraju da prođu kroz krajnje tačke (**2**).

$$a_1x_0^{\;2}+b_1x_0+c_1=f(x_0)$$

$$a_nx_n^{\;2}+b_nx_n+c_n=f(x_n)$$

Kvadratni splajn

- Prvi izvodi u unutrašnjim čvorovima moraju biti jednaki (**n-2**).

$$s_i'(x) = 2a_i x + b_i$$

$$2a_{i-1}x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i$$

- Do sada 2(N-2)+2+N-2=3(N-1)-1 jednačina
- Imamo 3(N-1) nepoznatih
- Uvodimo pretpostavku da je drugi izvod u prvoj tački nula. Tako dobijamo još (**1**) jednačinu:

$$a_1 = 0$$

(Prve dve tačke biće povezane pravom linijom) – proizvoljan izbor često korićen u praksi

Kvadratni splajn - primer

Kreiramo **kvadratni splajn za date tačke**.
Procenjujemo vrednost u x = 5.

x	3.0	4.5	7.0	9.0
$f(x)$	2.5	1.0	2.5	0.5

Rešenje:

Imamo **3** splajna (n=3), **9** nepoznatih.

Kvadratni splajn - primer

1. **Unutrašnje tačke imaju iste vrednosti:**

- **Prva unutrašnja tačka (4.5, 1.0)**

Prva jednačina:

$$x_1^2 a_1 + x_1 b_1 + c_1 = f(x_1)$$

$$(4.5)^2 a_1 + 4.5 b_1 + c_1 = f(4.5) \longrightarrow \boxed{20.25 a_1 + 4.5 b_1 + c_1 = 1.0}$$

Druga jednačina:

$$x_1^2 a_2 + x_1 b_2 + c_2 = f(x_1)$$

$$(4.5)^2 a_2 + 4.5 b_2 + c_2 = f(4.5) \longrightarrow \boxed{20.25 a_2 + 4.5 b_2 + c_2 = 1.0}$$

Kvadratni splajn - primer

- **Druga unutrašnja tačka (7.0, 2.5)**

Treća jednačina:

$$x_2^2 a_2 + x_2 b_2 + c_2 = f(x_2)$$

$$(7)^2 a_2 + 7b_2 + c_2 = f(7) \longrightarrow \boxed{49a_2 + 7b_2 + c_2 = 2.5}$$

Četvrta jednačina:

$$x_2^2 a_3 + x_2 b_3 + c_3 = f(x_2)$$

$$(7)^2 a_3 + 7b_3 + c_3 = f(7) \longrightarrow \boxed{49a_3 + 7b_3 + c_3 = 2.5}$$

Kvadratni splajn - primer

- Prvi i poslednji splajn moraju da prođu kroz krajnje tačke.

Početna tačka **(3.0, 2.5)**

$$x_0^2 a_1 + x_0 b_1 + c_1 = f(x_0) \rightarrow \boxed{9a_1 + 3b_1 + c_1 = 2.5}$$

Poslednja tačka **(9, 0.5)**

$$x_3^2 a_1 + x_3 b_3 + c_3 = f(x_3) \rightarrow \boxed{81a_3 + 9b_3 + c_3 = 0.5}$$

Kvadratni splajn - primer

- Jednaki prvi izvodi u unutrašnjim tačkama.

Prva unutrašnja tačka **(4.5, 1.0)**

$$2x_1 a_1 + b_1 = 2x_1 a_2 + b_2 \rightarrow \boxed{9a_1 + b_1 = 9a_2 + b_2}$$

Druga unutrašnja tačka **(7.0, 2.5)**

$$2x_2 a_2 + b_2 = 2x_3 a_3 + b_3 \rightarrow \boxed{14a_2 + b_2 = 14a_3 + b_3}$$

- Drugi izvod u prvoj tački je 0

$$\boxed{f''(x_0) = a_1 = 0}$$

Kvadratni splajn - primer

$$\begin{bmatrix} 4.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20.25 & 4.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 49 & 7 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 49 & 7 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 81 & 9 & 1 \\ 1 & 0 & -9 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 1 & 0 & -14 & -1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2.5 \\ 2.5 \\ 2.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}$$

Kvadratni splajn - primer

Rešavanjem sistema sa 8 jednačina i 8 nepoznatih dobija se:

$$a_1 = 0, \quad b_1 = -1, \quad c_1 = 5.5$$

$$a_2 = 0.64, \; b_2 = -6.76, \; c_2 = 18.46$$

$$a_3 = -1.6, \; b_3 = 24.6, \quad c_3 = -91.3$$

$$f_1(x) = -x + 5.5, \qquad 3.0 \le x \le 4.5$$

$$f_2(x) = 0.64x^2 - 6.76x + 18.46, \qquad 4.5 \le x \le 7.0$$

$$f_3(x) = -1.6x^2 + 24.6x - 91.3, \qquad 7.0 \le x \le 9.0$$

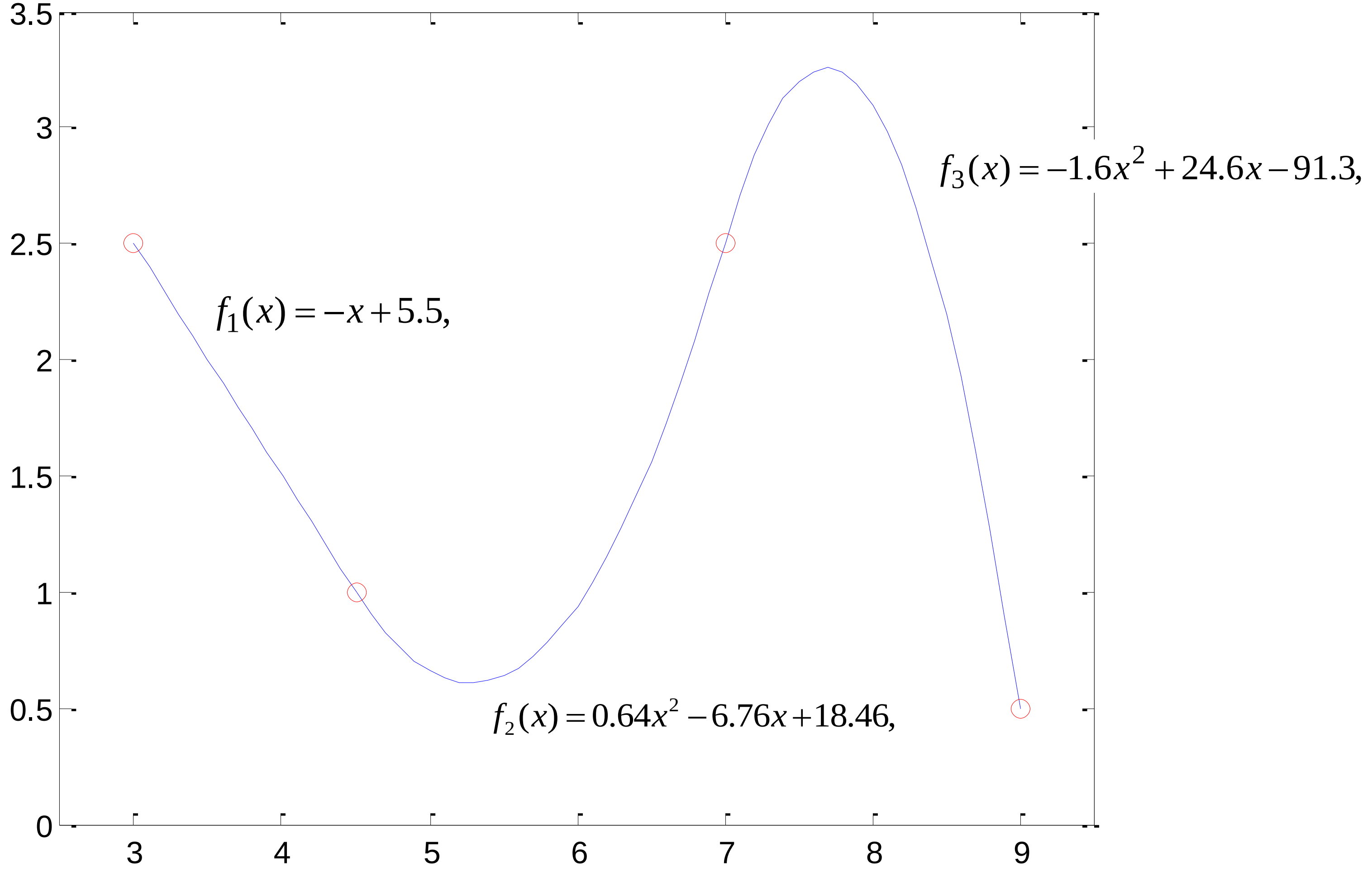
Kvadratni splajn - primer

- Procenjujemo vrednost za x=5.
- Zamenimo ga u odgovarajući splajn.

$$f_2(x) = 0.64x^2 - 6.76x + 18.46, \quad 4.5 \le x \le 7.0$$

$$f_2(x) = 0.64 * 5^2 - 6.76 * 5 + 18.46 = 0.66$$

Kvadratni splajn – primer grafik



Kubni splajn

- Cilj: odrediti polinom trećeg stepena između svake dve date tačke

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

- Za N tačaka i = (1, 2, …N):
  - N-1 splajnova (intervala),
  - 4(N-1) nepoznatih (a_i, b_i ,c_i, d_i)

- Definisaćemo ograničenja za interpolacionu funkciju
  - Idealno: sistem od 4(N-1) jednačina gde su nepoznate koeficijenti (a_i, b_i ,c_i, d_i)

Kubni splajn

- Vrednosti splajnova moraju biti jednake u unutrašnjim tačkama **2(n-2)**
- Prvi i poslenji splajn moraju da prolaze kroz krajnje tačke **(2)**
- Želimo da funkcija bude glatka:
  - prvi izvodi splajnova moraju biti jednaki u unutrašnjim tačkama **(n-2)**
  - Drugi izvodi splajnova moraju biti jednaki u unutrašnjim tačkama **(n-2)**
- Ukupno: 2n-4+2+n-2+n-2=4n-6 =4(N-1)-2
  - Nedostaju nam još 2 jednačine.

Tipovi kubnog splajna

- Postoji više mogućnosti za dve dodatne jednačine

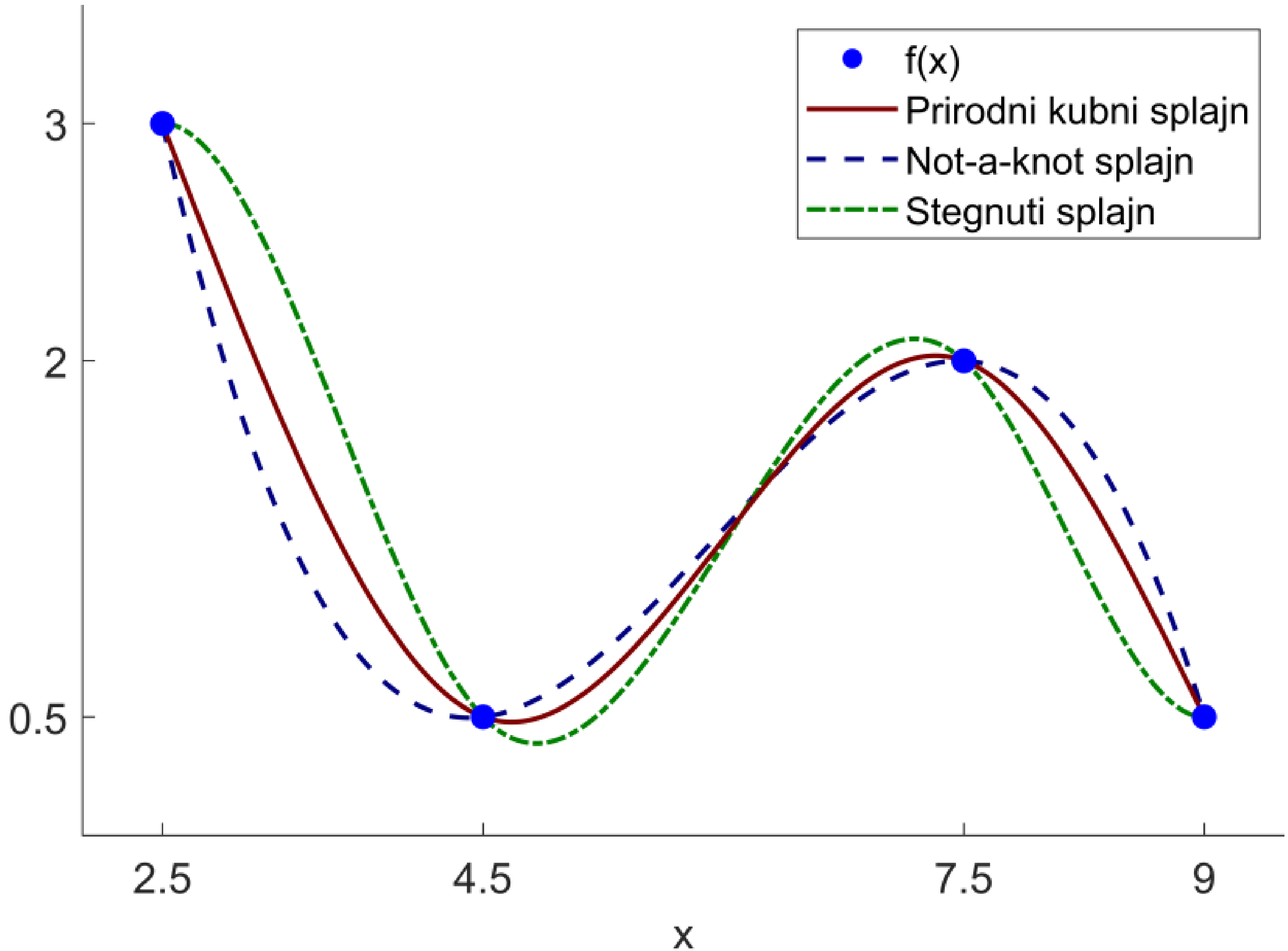
- Prirodni splajn (*natural spline*)
  - Pretpostavimo da su drugi izvodi u kranjim tačkama jednaki nuli
  - Aproksimaciona funkcija postaje prava linija u krajnjim čvorovima
  - Više simetričan u odnosu na kvadratni splajn (gde je to samo za 1. interval)

- Stegnuti splajn (*clamped spline*)
  - Pretpostavimo da su prvi izvodi u prvoj i poslednjoj tački konstante date unapred
  - Npr. ako je je konstanta 0, splajn postaje horizontalan na krajevima

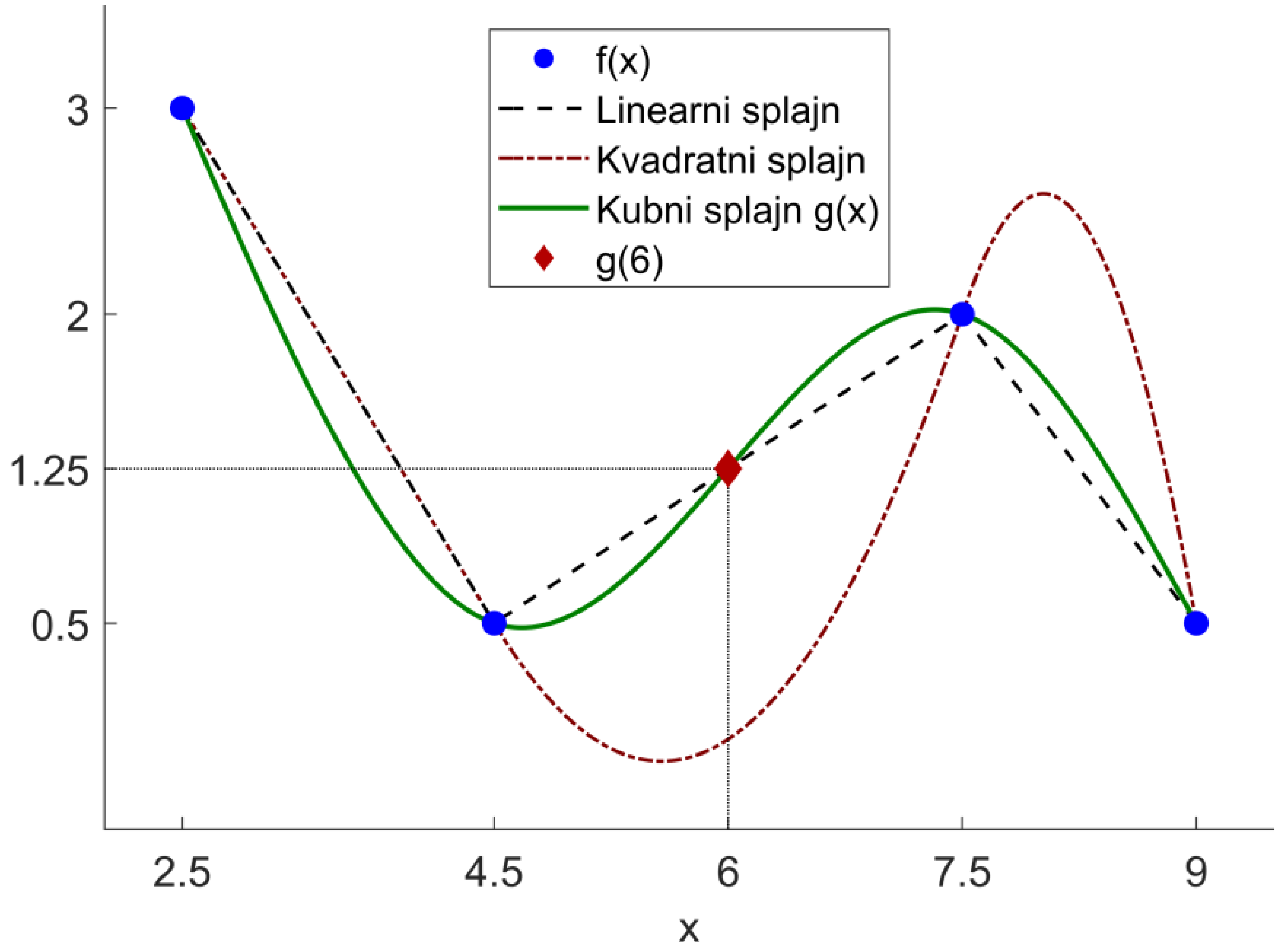
Tipovi kubnog splajna

- *“Not-a-knot”* splajn
  - Pretpostaviti jednakost trećeg izvoda u drugoj i preposlednjoj tački (u prethodnim uslovima smo već specificirali kontinuitet prvog i drugog izvoda)
  - Ovaj uslov će rezultovati time da su susedni splajnovi za prva dva segmenta međusobno jednaki, kao i to da su susedni splajnovi za poslednja dva segmenta međusobno jednaki
  - Ime *“Not-a-knot”* potiče od činjenice da druga i pretposlednja tačka nisu više čvorovi, tj. nisu više spoj *različitih* splajn funkcija

Tipovi kubnog splajna



Poređenje linearnog, kvadratnog i kubnog splajna



Splajn u Matlabu

- Ugrađena funkcija spline, kreira kubni splajn

```
approximated_values = spline(x, y, query_points)
```

- x, y – koordinate poznatih tačaka funkcije *f*
- query_points – tačke u kojima želimo da odredimo nepoznate vrednosti funkcije *f*
- Ako pozovemo funkciju na prethodno prikazani način dobijamo *“Not-a-knot”* splajn
- Ako u vektor y dodamo još dve tačke na prvo i poslednje mesto: vrednosti prvog izvoda u prvoj i poslednjoj tački, dobijamo *clamped* splajn

Not-a-knot splajn

- Generišemo tačke:

```
x = linspace(-1, 1, 9);
y = 1./(1+25*x.^2);
```

- Generišemo 100 xx tačaka

i pozivamo spline funkciju

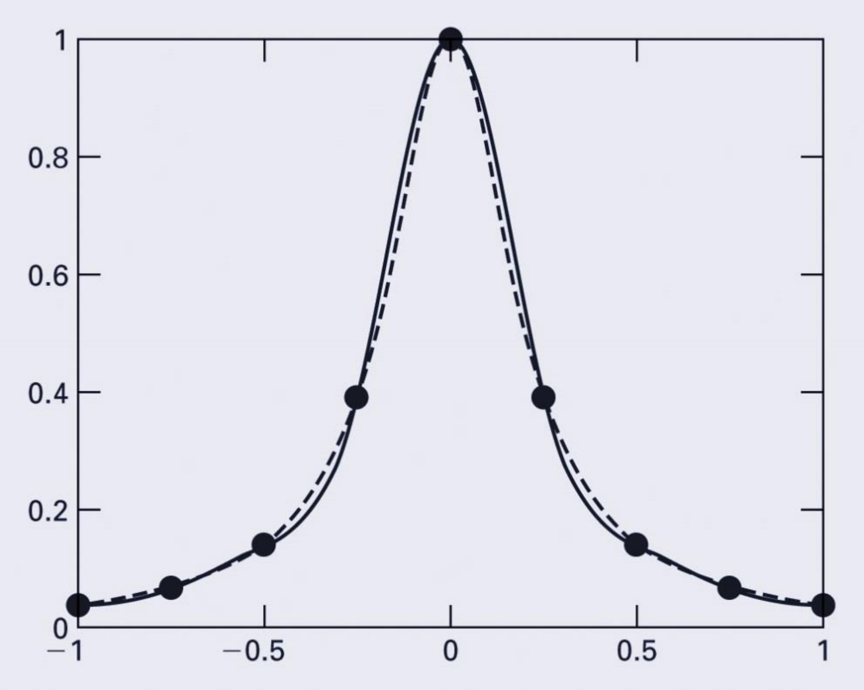
```
xx = linspace(-1, 1);
yy = spline(x, y, xx);
```

- Izračunavamo vrednost funkcije u

9 tačaka iz x i crtamo grafik splajna

(puna linija) i funkcije (isprekidana linija),

```
yr = 1./(1+25*xx.^2)
plot(x, y, 'o', xx, yy, '-', xx, yr, '--');
```



Clamped splajn

- Generišemo tačke:

```
x = linspace(-1, 1, 9);
y = 1./(1+25*x.^2);
```

Fiksiramo prvi izvod u poslednjoj i prvoj tački da je 1 i -4

```
yc = [1 y -4]
```

- Generišemo 100 xx tačaka

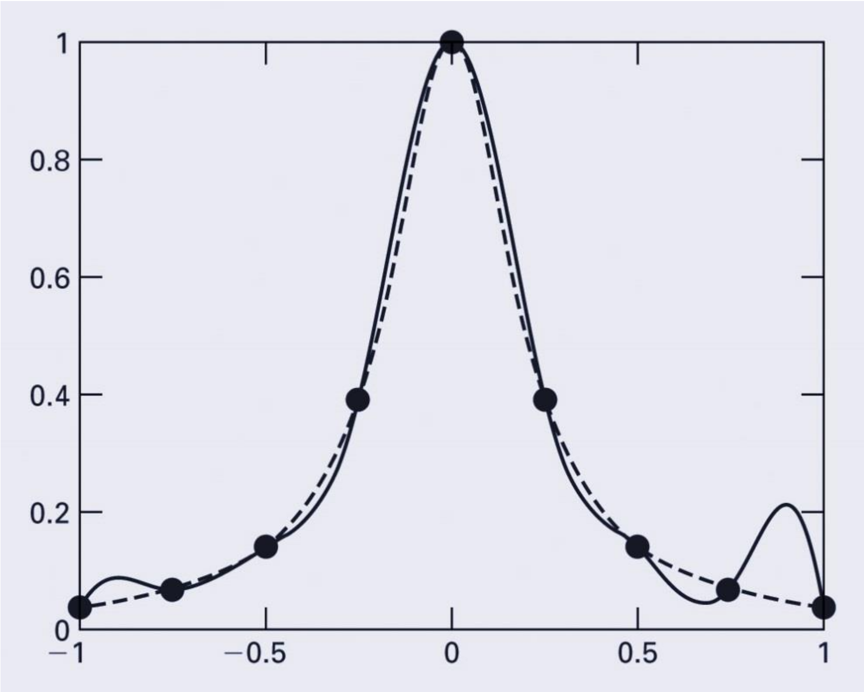
i pozivamo spline funkciju

```
xx = linspace(-1, 1);
yyc = spline(x, yc, xx);
```

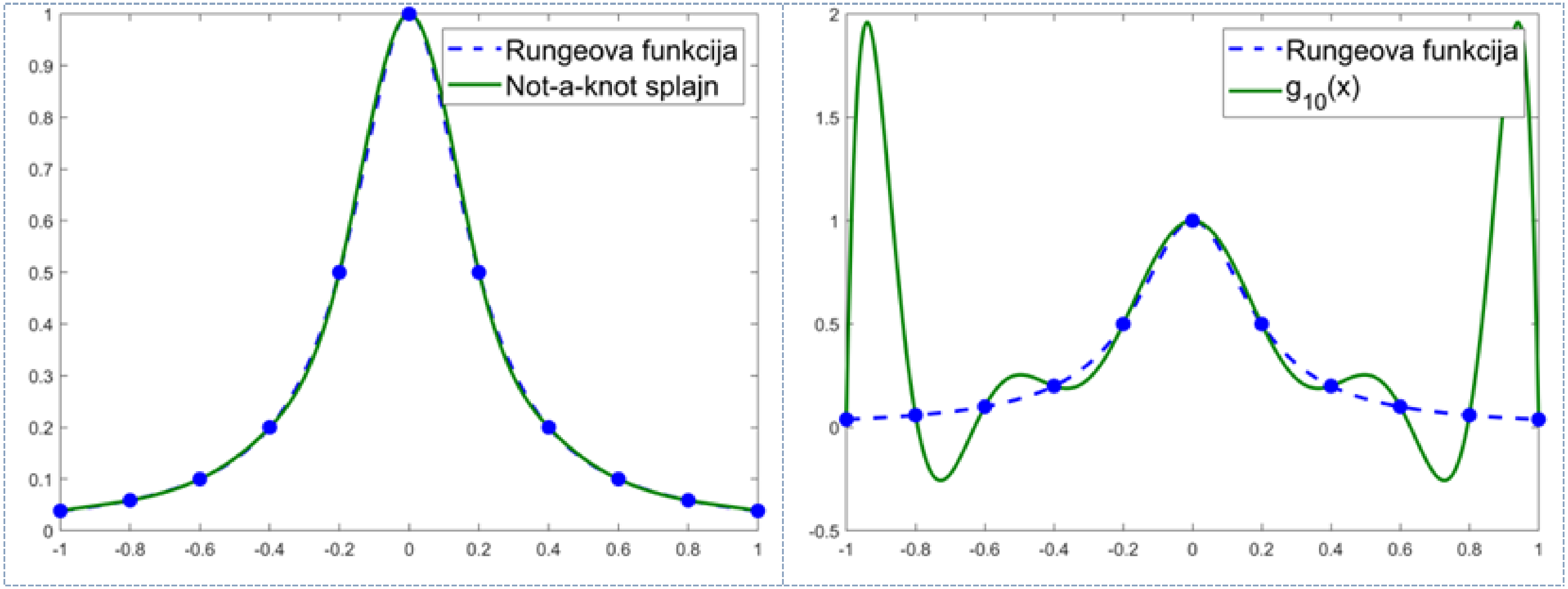
- Izračunavamo vrednost funkcije u

9 tačaka iz x i crtamo grafik splajna (puna linija) i funkcije (isprekidana linija),

```
yr = 1./(1+25*xx.^2)
plot(x, y, 'o', xx, yyc, '-', xx, yr, '--')
```



Splajn i oscilacije interpolacionog polinoma



Grafikon 4.22. Rezultat interpolacije Rungeove funkcije pomoću $N = 11$ ravnomerno raspoređenih tačaka na intervalu $[-1, 1]$. Na slici levo je prikazana interpolacija pomoću „not-a-knot" splajna, a na slici desno interpolacija pomoću jedinstvenog polinoma 10. stepena.