

Matematičke metode fizike III

Petar Stipanović
Larisa Zoranić

2023/24

Bilješke s predavanja

5	Aproksimacije (interpolacija i ekstrapolacija)	5-1
5.1	Interpolacija polinomima	5-2
5.1.1	Lagrangeov interpolacijski polinom	5-2
5.1.2	Nevillov algoritam	5-7
5.2	Spline interpolacija	5-14
5.2.1	Linearni interpolacijski splajn	5-14
5.2.2	Kubični interpolacijski splajn	5-15
A	Interpolacija	A-18
A.1	Dokaz jedinstvenosti prirodnog kubičnog splajna i primjer	A-18

Nastavni materijal nije za objavu, već samo služi za internu upotrebu kako bi se studentima olakšalo praćenje nastave.

Literatura

- [1] Morten Hjorth-Jensen: *Computational Physics*, University of Oslo, 2007, 2010, 2015.
- [2] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery: *Numerical Recipes in C: The Art of Scientific Computing*, SE, Cambridge University Press, USA, 2002.
- [3] Podsjetnici iz programiranja: *C++ Language*, URL: <https://cplusplus.com/doc/tutorial/>;
Programiz: *Learn C Programming*, URL: <https://cplusplus.com/doc/tutorial/>;
Programiz: *Learn Python Programming*, URL: <https://www.programiz.com/python-programming/>;
Tutorials Point: *Python Tutorial*, URL: <https://www.tutorialspoint.com/python/index.htm>;
W3Schools: *Python Tutorial*, URL: <https://www.w3schools.com/python/>.
- [4] NumPy: *The fundamental package for scientific computing with Python*,
NumPy documentation, URL: <https://numpy.org/doc/stable/index.html>;
Linear algebra numpy.linalg.solve URL: <https://numpy.org/doc/stable/reference/generated/numpy.linalg.solve.html>.
- [5] SciPy: *Fundamental algorithms for scientific computing in Python*, URL: <https://scipy.org/>.
- [6] V. Hari i dr: *Numerička analiza*, skripta, Sveučilište u Zagrebu, PMF, 2004.
- [7] Leandra Vranješ Markić: *Matematičke metode fizike I*, skripta, Sveučilište u Splitu, PMFST, 2009.

- [8] Saša Krešić-Jurić: *Parcijalne diferencijalne jednačbe*, skripta, Sveučilište u Splitu, PMFST, 2014.
- [9] Ivica Smolić: *Matematičke metode fizike*, skripta, Sveučilište u Zagrebu, PMF 2020.
- [10] W. Kolos i L. Wolniewicz, J. Chem. Phys. **43**, 2429 (1965); Chem. Phys. Lett. **24**, 457 (1974).
- [11] M. J. Jamieson, A. Dalgarno i L. Wolniewicz, Phys. Rev. A **61**, 042705 (2000).
- [12] I. Bešlić, L. Vranješ Markić i J. Boronat, Phys. Rev. B **80**, 134506 (2009).
- [13] Zong-Chao Yan, James F. Babb, A. Dalgarno i G. W. F. Drake, Phys. Rev A **54**, 2824 (1996).
- [14] Angela B. Shiflet & George W. Shiflet: *Introduction to Computational Science - Modeling and Simulation for the Sciences*, 2. izdanje, Princeton University Press, New Jersey, USA, 2014.
- [15] *Hrvatska enciklopedija*, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2020.
URL: <http://www.enciklopedija.hr/Natuknica.aspx?ID=18092> te ID=46397, 30418 i 17886 (pristupljeno 3. 5. 2020)
- [16] J. N. Hays: *Epidemics and pandemics: their impacts on human history*, ABC-CLIO, 2005.
- [17] P. van den Driessche: *Reproduction numbers of infectious disease models*, Infectious Disease Modelling, sv. 2, izd. 3, str. 288–303, kol. 2017. DOI: <https://doi.org/10.1016/j.idm.2017.06.002>
- [18] Worldometers.info *COVID-19 Coronavirus Pandemic*, Dover, Delaware, USA.
URL: <https://www.worldometers.info/coronavirus/> (pristupljeno 17. 5. 2020)
- [19] World Health Organization: *WHO Director-General's opening remarks at the media briefing on COVID-19 - 3 March 2020*.
URL: <https://www.who.int/dg/speeches/detail/who-director...on-covid-19—3-march-2020> (pristupljeno 17. 5. 2020)

5 Aproximacije (interpolacija i ekstrapolacija)

Problem aproksimacije predstavlja postupak konstrukcije funkcije φ koja je u nekom smislu slična funkciji $f : X \rightarrow \mathbb{R}$, $X \subseteq \mathbb{R}$ koja je poprilično složena za promatrani problem ili poznata samo u nekim točkama. Postupak aproksimacije provodimo na temelju informacija koje znamo o f .

Jednostavniji primjer aproksimacija su linearne kombinacije

$$\varphi(x) = \sum_{i=0}^m a_i \varphi_i(x)$$

funkcija $\varphi_i(x)$ koje biramo iz baza, koje najbolje odražavaju ponašanje promatranog problema, primjerice:

- algebarski polinomi $\varphi_i(x) \in \{1, x, x^2, \dots, x^m\}$, ili
 $\varphi_i(x) \in \{1, (x - x_0), (x - x_0)^2, \dots, (x - x_0)^m\}$, ili
 $\varphi_i(x) \in \{P_l(x) : l \in \mathbb{N}_0\}$ gdje su P_l Legendreovi polinomi;
- trigonometrijski polinomi $\varphi_i(x) \in \{1, \cos x, \sin x, \cos(2x), \dots\}$;
- spline (splajn) funkcije $\varphi|_{[x_{k-1}, x_k]} = p_k(x)$, tj. po dijelovima polinomi p_k željenog stupnja.

Složeniji primjer čine nelinearne aproksimacijske funkcije, primjerice:

- eksponencijalne aproksimacije ($2r + 2$ nezavisnih parametara)

$$\varphi(x) = C_0 e^{b_0 x} + C_1 e^{b_1 x} + \dots + C_r e^{b_r x};$$

- racionalne aproksimacije ($r + s + 1$ nezavisni podatak)

$$\varphi(x) = \frac{b_0 + b_1 x + \dots + b_r x^r}{c_0 + c_1 x + \dots + c_s x^s}.$$

Aproksimacije upotrebljavamo kada je:

- f prekomplikirana, primjerice za mali x

$$f(x) = \sin x = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \approx x \equiv \varphi(x)$$

- f zadana tablično (točkama)

$$\mathbb{T}_m = \{ (x_k, y_k) \mid k \in [0, m] \subset \mathbb{N}_0; x_i < x_j \text{ za } i < j; y_k = f(x_k) \}, \quad (5.1)$$

a zanima nas $f(x \neq x_k)$ koji aproksimiramo s $\varphi(x)$, primjerice kada zavisnost eksperimentalno izmjerenih veličina želimo opisati funkcijom.

Najčešći zahtjevi konstrukcije funkcije φ :

- ✓ $\varphi(x)$ **minimalno odstupa** od \mathbb{T}_m , odnosno (5.1), primjerice kod linearne regresije zahtijevamo da kvadratno odstupanje bude minimalno;
- ✓ $\varphi(x)$ **prolazi danim točkama** \mathbb{T}_m . Ako je
 - ▶ $x \in I \equiv [x_0, x_m]$, postupak procjene $\varphi(x)$ zovemo **interpolacija**;
 - ▶ $x \notin I$, postupak procjene $\varphi(x)$ zovemo **ekstrapolacija**.

Opširnije se može pronaći u literaturi [6], uvodni dio iz 7.1 te podnaslovi 7.1.1, 7.1.2 i 7.1.3 do minimizacije pogreške, a ovdje ćemo se u nastavku uglavnom posvetiti polinomnoj i spline interpolaciji.

5.1 Interpolacija polinomima

Osnovna zadaća interpolacije je provući dovoljno glatku krivulju kroz skup točaka (x_k, y_k) poput (5.1). Numerički to postizemo pomoću funkcije $\varphi(x)$ zahtijevajući da u tzv. **čvorovima** x_k poprima vrijednosti $\varphi(x_k) = y_k$. Ako se koristi $m + 1$ čvorova, onda se govori o interpolaciji reda m , a stupanj polinoma određen je sljedećim teoremom.

Teorem 5.1 *Teorem o egzistenciji i jedinstvenosti.*

Neka je $m \in \mathbb{N}_0$. Tada za $(x_k, y_k = f(x_k))$, $k = 0, \dots, m$, gdje je $x_i \neq x_{j \neq i}$, postoji jedinstveni (interpolacijski) polinom

$$\varphi(x) \equiv P_m(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (5.2)$$

stupnja $n \leq m$ za koji vrijedi

$$P_m(x_k) = y_k, \quad k = 0, 1, \dots, m. \quad (5.3)$$

Dokaz koristi činjenicu da linearni sustav s regularnom matricom ima jedinstveno rješenje. [6]

5.1.1 Lagrangeov interpolacijski polinom

Jedan od načina da pronađemo (5.2), koji zadovoljava (5.3), jest da ga napišemo kao sumu članova $y_i \cdot L_i(x)$ gdje su $L_i(x)$ takvi polinom da vrijedi

$$L_i(x_j) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}; \quad i, j \in \{0, 1, \dots, m\}, \quad (5.4)$$

odnosno kojima su svi $x_{j \neq i}$ nul-točke. Općenito taj polinom možemo napisati kao produkt

$$L_i(x) = C_i(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_m), \quad (5.5)$$

a konstantu C_i lako podesimo da vrijedi $L_i(x_i) = 1$,

$$C_i = \frac{1}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_m)}. \quad (5.6)$$

Opisanim postupkom dobili smo

Lagrangeov oblik interpolacijskog polinoma

$$\varphi(x) = P_m(x) = \sum_{i=0}^m y_i \cdot L_i(x); \quad L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^m \frac{x - x_j}{x_i - x_j}. \quad (5.7)$$

Mane ovakvog pristupa su:

- slaba efikasnost;
- moguće greške zaokruživanja;
- nema procjene greške ako ne znamo f .

Teorem 5.2 Teorem o grešci.

Ako f ima $f^{(n+1)}$ derivaciju na $[a, b]$ tada $(\forall x' \in [a, b]) (\exists \xi \in \langle a, b \rangle)$ takav da vrijedi

$$f(x') - P_n(x') = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x'); \quad \omega(x') = \prod_{j=0}^n (x - x_j). \quad (5.8)$$

Ako je $f^{(n+1)}$ ograničena na $[a, b]$ ili $f \in C_{[a,b]}^{n+1}$ (veći uvjet), onda se može dobiti sljedeća ocjena greške interpolacijskog polinoma za funkciju f u točki $x \in [a, b]$

$$|f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |\omega(x)|, \quad M_{n+1} = \max_{x \in [a,b]} |f^{(n+1)}(x)|. \quad (5.9)$$

Više se detalja može pronaći u literaturi [6] (uvodni dio iz 7.2 te podnaslovi 7.2.1 i 7.2.2 do zadnjeg paragrafa na 296. str. pa podnaslovi 7.2.3 i 7.2.4), a ovdje ćemo se, nakon primjera 5.1 i 5.2, više koncentrirati na numeričko računanje kada nam nisu poznati analitički izrazi u području gdje obavljamo interpolaciju.

Primjer 5.1

Mjerenjem ovisnosti veličine $y = f(x)$ o veličini x dobiven je skup uređenih parova

$$\mathbb{Y} = \{ (x_i, y_i) \mid f(x_i) = y_i, i = 0, 1, 2 \} = \{ (0, 1), (3, 2), (1, 3) \}.$$

Sljedeći korake navedene u nastavku zadatka odredite linearnu funkciju

$$\varphi(x) = \sum_{i=0}^m a_i \varphi_i(x)$$

koja aproksimira ponašanje veličine $f(x)$ uz zahtjev da prolazi točkama skupa \mathbb{Y} .

- Prikažite grafički skup točaka \mathbb{Y} .
- Koje $\varphi_i(x)$ ima smisla odabrati i koliki m uzeti?
- Koliko polinoma 2. stupnja, $P_2(x)$, prolazi kroz \mathbb{Y} ? Kako možemo odrediti $P_2(x)$?
- Odredite $\varphi(x)$ koristeći Lagrangeov oblik interpolacijskog polinoma, $\varphi(x) = P_2(x)$.
- Usporedite na grafičkom prikazu dobiveno rješenje za $P_2(x)$ s podacima \mathbb{Y} .
- Izračunajte $P_2(1.5)$. Što možete reći o grešci procjene $P_2(1.5)$? ◀

Rješenje:

- Skup \mathbb{Y} prikazan je kružnicama na slici 5.1.
- Iz grafičkog prikaza skupa \mathbb{Y} na slici 5.1 očito je da se radi o nekolinearnim točkama. Međutim, zbog vrlo malog uzorka imamo malo informacija o njihovoj ovisnosti. Ovisnost im donekle slična ponašanju polinoma ili čak trigonometrijske funkcije. Dakle, mogli bismo koristiti bazu algebarskih polinoma $\varphi_i \in \{1, x, x^2, \dots\}$ ili trigonometrijskih polinoma $\varphi_i \in \{1, \sin(x), \sin(2x), \cos(x), \cos(2x), \dots\}$. Kako je prva spomenuta baza puno jednostavnija fokusirat ćemo se na nju. Ako je dan $m+1$ par točaka (x_i, y_i) takav da su čvorovi (bazne točke ili interpolacijske točke) $x_i \neq x_{j \neq i}$, tada postoji jedinstveni polinom $P_m(x)$ stupnja $m \leq n$ takav da vrijedi $P_m(x_i) = f(x_i)$. Stoga ima smisla uzeti $m = 2$.
- Kao što smo već spomenuli, zbog 3 bazne točke dane u \mathbb{Y} , postoji jedinstveni interpolacijski $P_m(x)$ stupnja $m \leq 2$, dok je sa slike 5.1 očito nelinearno ponašanje od $f(x)$ pa je $m \geq 2$. Prema tome, postoji jedinstveni $P_2(x)$ koji prolazi danim točkama. Polinom $P_2(x) = a_2 x^2 + a_1 x + a_0$ možemo odrediti na više načina: rješavajući sustav od 3 jednačbe $P(x_i) = y_i, i = 0, 1, 2$ ili koristeći Lagrangeov oblik interpolacijskog polinoma ili neki drugi oblik povezan s podacima u

\mathbb{Y} poput „Hornerove sheme” za interpolacijski polinom (poglavlje 6.1.6 u [6]) ili Newtonov oblik interpolacijskog polinoma (poglavlje 7.2.5 u [6]).

d) Odaberimo $(x_i, y_i) \in \mathbb{Y}$ tako da je $(\forall i < j) x_i < x_j$, odnosno biramo

$$x_0 = 0, x_1 = 1, x_2 = 3;$$

$$y_0 = 1, y_1 = 3, y_2 = 2.$$

Polinom, koji prolazi kroz dani skup točaka, odredit ćemo koristeći Lagrangeovu metodu konstrukcije interpolacijskog polinoma za $m = 2$

$$P_2(x) = \sum_{i=0}^2 L_i(x) \cdot y_i = L_0(x) \cdot y_0 + L_1(x) \cdot y_1 + L_2(x) \cdot y_2 \quad (5.10)$$

gdje su polinomi

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^2 \frac{x - x_j}{x_i - x_j}. \quad (5.11)$$

redom

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 1)(x - 3)}{(0 - 1)(0 - 3)} = \frac{1}{3}(x^2 - 4x + 3), \quad (5.12a)$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x - 0)(x - 3)}{(1 - 0)(1 - 3)} = -\frac{1}{2}(x^2 - 3x), \quad (5.12b)$$

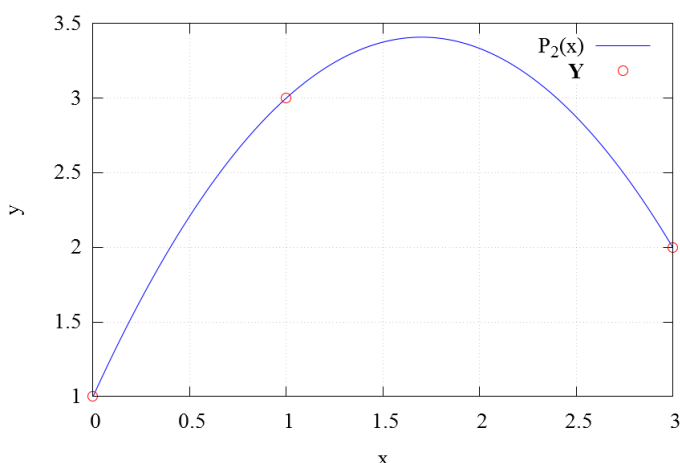
$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x - 0)(x - 1)}{(3 - 0)(3 - 1)} = \frac{1}{6}(x^2 - x). \quad (5.12c)$$

Uvrštavanjem dobivenih izraza (5.12) u (5.10) dobivamo konačan oblik interpolacijskog polinoma

$$P_2(x) = -\frac{5}{6}x^2 + \frac{17}{6}x + 1 \quad (5.13)$$

koji je prikazan zajedno s točkama iz \mathbb{Y} na slici 5.1.

e)



Slika 5.1: Polinom $P_2(x)$ iterpoliran kroz skup točaka \mathbb{Y} .

f) Koristeći (5.13), dobivamo $P_2(1.5) = 27/8 = 3.375$. Međutim, zbog nepoznavanja funkcije $f(x)$ ne možemo ništa reći o grešci napravljene procjene. Također, zbog vrlo malog uzorka nije moguće ni iz grafičkog slijeda podataka predvidjeti točno ponašanje funkcije $f(x)$. Npr. mogli bismo zamisliti funkciju $f(x)$ kojoj je točka $x = 3/2$ pol i tada bi dobivena procjena bila značajno pogrešna.

Primjer 5.2

Na tijelo od početnog trenutka djeluje ukupna sila $\vec{F}(T) = -8\hat{i} \text{ cN} + 6T\hat{i} \text{ cN s}^{-1}$. Detektor u početnom trenutku izmjeri brzinu \vec{V} i položaja tijela kojeg postavlja za referentnu točku (ishodište) te nastavlja svakih $\Delta T = 1 \text{ s}$ pohranjivati informacije o položaju tijela $\vec{R} = X\hat{i}$.

Radi numeričke analize umjesto općeg znaka veličine $B = \{B\} [B]$ uvodimo oznake odgovarajućim malim simbolom $b \equiv \{B\}$ koji sadrži samo vrijednost veličine B bez mjerne jedinice $[B]$. Dakle, $T = t \text{ s}$, $M = m \text{ kg}$, $X = x \text{ cm}$, $V = v \text{ cm s}^{-1}$, $A = a \text{ cm s}^{-2}$ i sl. Ako je za tijelo mase $m = 1$ izmjerena početna brzina $\vec{v} = -2\hat{i}$, odredite skupove uređenih parova koje je pohranio detektor

$$\mathbb{X}_n = \{ (t_k, x_k) \mid t_k = (k+1)\Delta T/\text{s}; x_k = x(t_k); k \in [0, n] \subset \mathbb{N} \}.$$

za svaki $n \in \{2, 3, 4\}$. Dodatno napišite izraz za $X(T)$ i procijenite vrijednost koordinata $x(t)$ u trenucima $t = 2.5$ i $t = 4.5$.

- Svaki \mathbb{X}_n aproksimirajte polinomom koristeći Lagrangeov oblik interpolacijskog polinoma $P_n(t)$. Postoji li veza među tim polinomima? Prikažite na istom grafu \mathbb{Y}_2 , $P_2(t)$ i članove od $P_2(t)$ prije sumiranja.
- Koristeći dobivene $P_n(t)$ procijenite položaje tijela u trenucima $t = 2.5$ i $t = 4.5$ te usporedite s prethodno izračunatim vrijednostima $x(t)$. Kolike su relativne greške? Što možete reći o kvaliteti aproksimacija $P_n(t)$? Slijede li iste tvrdnje iz grafičke usporedbe spomenutih veličina?

Rješenje: Rješavajući diferencijalnu jednadžbu danu 2. Newtonovim zakonom, dobivamo ovisnost položaja o vremenu

$$X(T) = -2T \text{ cm s}^{-1} - 4T^2 \text{ cm s}^{-2} + T^3 \text{ cm s}^{-3}, \quad (5.14a)$$

$$x(t) = -2t - 4t^2 + t^3; y(t) = 0; z(t) = 0. \quad (5.14b)$$

Koristeći (5.14b) lako odredimo tražene skupove

$$\mathbb{X}_2 = \{ (1, -5), (2, -12), (3, -15) \} \quad (5.15a)$$

$$\mathbb{X}_3 = \{ (1, -5), (2, -12), (3, -15), (4, -8) \} \quad (5.15b)$$

$$\mathbb{X}_4 = \{ (1, -5), (2, -12), (3, -15), (4, -8), (5, 15) \} \quad (5.15c)$$

- Odaberimo čvorove t_i , redom kako su navedeni u skupovima \mathbb{X}_n . Koristeći Lagrangeov oblik interpolacijskog polinoma

$$P_n(t) = \sum_{i=0}^n L_{n,i}(t) \cdot x_i; \quad L_{n,i}(t) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t - t_j}{t_i - t_j}. \quad (5.16)$$

dobivamo redom za $n = 2, 3, 4$:

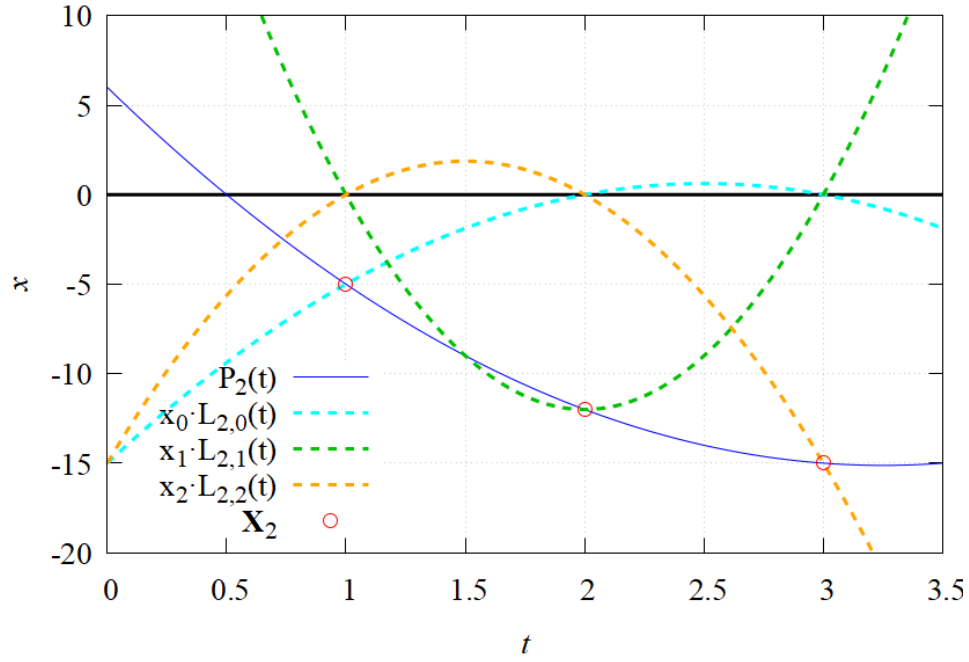
$$P_2(t) = 2t^2 - 13t + 6; \quad (5.17a)$$

$$P_3(t) = t^3 - 4t^2 - 2t; \quad (5.17b)$$

$$P_4(t) = t^3 - 4t^2 - 2t. \quad (5.17c)$$

Usporedbom dobivenih rješenja možemo uočiti jednakost $P_3(t) = P_4(t)$ jer među danim točkama \mathbb{X}_n postoji korelacija opisana izrazom (5.14b). Stoga bi svaki dodatni par točaka ponovno rezultirao polinomom 3. stupnja $P_3(t)$.

Sumiranjem polinoma prikazanih isprekidanim linijama na slici 5.2 dobivamo $P_2(t)$ prikazan punom linijom koji prolazi kroz dani skup točaka \mathbb{X}_2 . Svaki polinom $L_{2,i}$ ima nultočke u svim $t_{j \neq i}$ iz skupa \mathbb{X}_2 , dok u t_i pomnožen sa x_i vraća vrijednost x_i . Time je osigurano da $P_2(t)$ prolazi točno točkama skupa \mathbb{X}_2 . Analogno slijedi poopćenje kada $2 \rightarrow n$.



Slika 5.2: Polinom (5.17a) interpoliran kroz skup točaka \mathbb{X}_2 i njegovi članovi (umnošci polinoma $L_{n,i}$ iz (5.16) s odgovarajućim vrijednostima x_i).

- b) Dobiveni izrazi $P_n(t)$ omogućuju nam procjenu x -koordinata u bilo kojim međutrenutcima. Za $t = 2.5$ i $t = 4.5$ imamo

$$P_3(2.5) = P_4(2.5) = -14.375 \quad (5.18a)$$

$$P_3(4.5) = P_4(4.5) = +1.125 \quad (5.18b)$$

$$P_2(2.5) = -14.0 \quad (5.18c)$$

$$P_2(4.5) = -12.0 \quad (5.18d)$$

pa su prema (5.14b) pripadne relativne greške

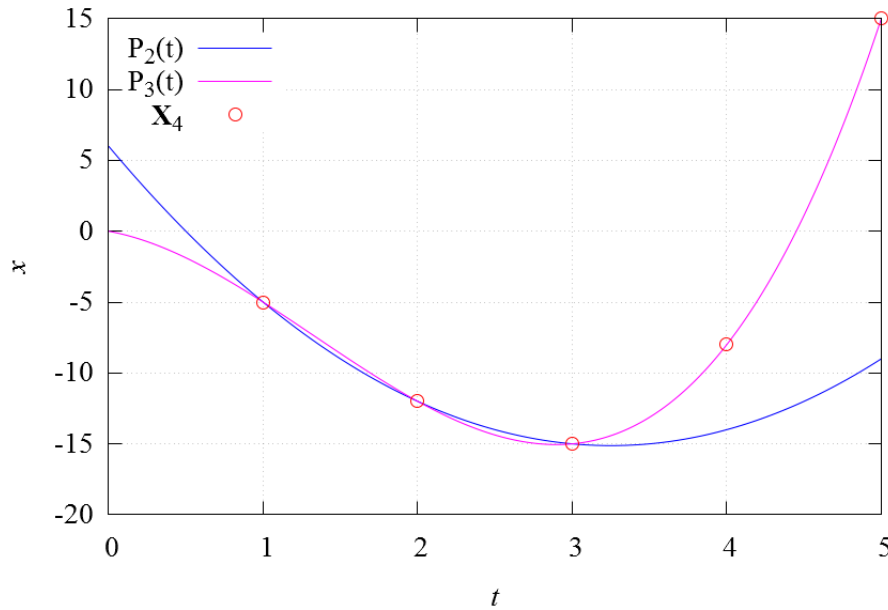
$$\Delta_{\text{rel}} P_{3,4}(2.5) = \frac{|P_3(2.5) - x(2.5)|}{|x(2.5)|} = 0 \quad (5.19a)$$

$$\Delta_{\text{rel}} P_{3,4}(4.5) = \frac{|P_3(4.5) - x(4.5)|}{|x(4.5)|} = 0 \quad (5.19b)$$

$$\Delta_{\text{rel}} P_2(2.5) = \frac{|P_2(2.5) - x(2.5)|}{|x(2.5)|} \approx 0.026 = 2.6\% \quad (5.19c)$$

$$\Delta_{\text{rel}} P_2(4.5) = \frac{|P_2(4.5) - x(4.5)|}{|x(4.5)|} = 11.6 \approx 1167\% \quad (5.19d)$$

Za bilo koji, čak i mali, skup točaka $\mathbb{X}_{n>2}$ interpolacijom dobivamo polinom $P_3(t)$ koji je jednak pravoj funkciji (5.14b) kojom smo generirali taj skup. Zbog jednostavne ovisnosti $x(t)$ u obliku polinoma malog stupnja, polinomijalnom interpolacijom brzo smo uspjeli doći do pravog rješenja čija je relativna greška 0 za bilo koji t . Ako nam nije potreban veoma precizan rezultat, čak i aproksimacija polinomom manjeg stupnja $P_2(t)$ daje prilikom interpolacije nelošu procjenu, npr. u $t = 2.5$ odstupa tek 2.6%. Problem nastaje ako dobiveni oblik $P_2(t)$ iskoristimo za ekstraplaciju, odnosno za $t > 3$, npr. u $t = 4.5$ odstupa za 1167%, dok $P_3(t)$ očekivano daje egzaktnu procjenu. Slični zaključci mogu se izvući i iz grafičkog prikaza ponašanja dobivenih polinoma na slici 5.3. Ekstrapolacije su vrlo riskantne. Ne treba im vjerovati na udaljenosti od zadnjeg t_i koja je veća od tipičnog razmaka između t_i i t_{i+1} .



Slika 5.3: Usporedba aproksimacija P_2 i P_3 .

Zadatak 5.1 Osmislite algoritam za numerički proračun koristeći Lagrangeov oblik interpolacijskog polinoma, napišite kod i provjerite ispravnost na podacima iz primjera 5.2. U kodu definirajte funkciju koja uzima zadani skup podataka \mathbb{X}_n , argument x i izračuna vrijednost $P(x)$. Testirajte točnost prikazujući rezultate kao na slici 5.3.

5.1.2 Nevillov algoritam

Postoji više metoda koje kroz dani skup točaka

$$\mathbb{T}_n = \{ (x_k, y_k) \mid k \in [1, n+1] \subset \mathbb{N}; x_i < x_j; y_k = f(x_k) \} \quad (5.20)$$

provlače jedinstveni polinom $P(x)$ stupnja $m \leq n$. Nevillov algoritam (NA) kao polaznu točku uzima polinom stupnja 0 i dodaje korekcije susjednih točaka povećavajući red polinoma do $P(x)$. Izračun $P(x)$, pomoću NA-a, objasniti ćemo na primjeru kada je dan \mathbb{T}_3 , odnosno niz vrijednosti

$$\{ f(x_1) = y_1, f(x_2) = y_2, f(x_3) = y_3, f(x_4) = y_4 \} \quad (5.21)$$

na temelju kojih želimo izračunati $f(x \neq x_i) \approx P(x)$. Ako bismo imali danu samo i -tu točku iz skupa \mathbb{T}_3 , mogli bismo $f(x)$ aproksimirati polinomima stupnja 0,

$$f(x) \approx P_i(x) = y_i \quad (5.22)$$

gdje samo s P_i označili vrijednost u točki x_i jedinstvenog polinoma reda 0 koji prolazi točkom (x_i, y_i) . Naravno, ovakva aproksimacija nije dobra pa nastavljamo aproksimaciju Lagrangeovim interpolacijskim polinomima 1. reda između dviju točaka u nizu x_i i x_{i+1}

$$f(x) \approx P_{1,2}(x) = \frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_1}{x_2 - x_1} f(x_2), \quad (5.23a)$$

$$f(x) \approx P_{2,3}(x) = \frac{x - x_3}{x_2 - x_3} f(x_2) + \frac{x - x_2}{x_3 - x_2} f(x_3). \quad (5.23b)$$

Dakle, s $P_{i,j}$ označili smo vrijednost u točki x jedinstvenog polinoma 1. reda koji prolazi kroz dvije točke: (x_i, y_i) i (x_j, y_j) . Mogao bi se konstruirati i $P_{1,3}$ koristeći čvorišta $x_{1,3}$ ali se njime sada nećemo zamarati. Koristeći pokrate (5.22), izraze (5.23) možemo zapisati u obliku

$$P_{1,2}(x) = \frac{x - x_2}{x_1 - x_2} P_1(x) + \frac{x - x_1}{x_2 - x_1} P_2(x) = \frac{(x - x_2)P_1(x) - (x - x_1)P_2(x)}{x_1 - x_2}, \quad (5.24a)$$

$$P_{2,3}(x) = \frac{x - x_3}{x_2 - x_3} P_2(x) + \frac{x - x_2}{x_3 - x_2} P_3(x) = \frac{(x - x_3)P_2(x) - (x - x_2)P_3(x)}{x_2 - x_3}. \quad (5.24b)$$

Općenito, množimo $P_i(x)$ sa $(x - x_{j \neq i})$. Izračunamo razliku ovakva 2 produkta „roditelja” i podijelimo razlikom korištenih čvorova x_k . Rezultat je „potomak” polinom $P_{i,j}$ za 1 većeg stupnja nego njegovi „roditelji” koji su istog stupnja. Ujedno polinom „potomak” prolazi točkama korištenim pri prethodnoj konstrukciji njegovih roditelja.

Ideju možemo proširiti kako bismo konstruirali polinom 2. stupnja koji prolazi kroz 3 zadane točke

$$P_{1,2,3}(x) = \frac{(x - x_3)P_{1,2}(x) - (x - x_1)P_{2,3}(x)}{x_1 - x_3}, \quad (5.25)$$

a zamjenom uvedenih definicija za $P_{i,j}$ i P_i dobili bismo izraz za Lagrangeov oblik interpolacijskog polinoma kroz čvorove x_1, x_2 i x_3 ,

$$P_{1,2,3}(x) = \sum_{i=1}^3 f(x_i) \prod_{\substack{j=1 \\ j \neq i}}^3 \frac{x - x_j}{x_i - x_j}. \quad (5.26)$$

Na analogan način [2] definiramo polinome višeg reda sve do $P_{1,2,3,\dots,n+1}$, gdje je $n + 1$ ukupan broj točaka. Shematski možemo prikazati različite P -predke u stupcima s lijeve strane koji vode na jednoznačnog „potomka” u međuretku susjednog desnog stupca. Npr. za $n = 3$

$$\begin{array}{ccccccc} x_1 : & y_1 = & P_1 & & & & \\ & & & P_{1,2} & & & \\ x_2 : & y_2 = & P_2 & & P_{1,2,3} & & \\ & & & P_{2,3} & & P_{1,2,3,4} & \\ x_3 : & y_3 = & P_3 & & P_{2,3,4} & & \\ & & & P_{3,4} & & & \\ x_4 : & y_4 = & P_4 & & & & \end{array} \quad (5.27)$$

Dakle, NA je rekursivni način popunjavanja shematskog prikaza (5.27), stupac po stupac, krećući s lijeve strane. Zasniva se na odnosu između „potomka” P i njegova dva „roditelja” u lijevom stupcu u susjednim redcima,

$$P_{i,i+1,\dots,i+m} = \frac{(x - x_{i+m})P_{i,i+1,\dots,i+m-1} + (x_i - x)P_{i+1,i+2,\dots,i+m}}{x_i - x_{i+m}}. \quad (5.28)$$

što je poopćenje izraza (5.25), a može se dokazati matematičkom indukcijom. Pošto je izraze poput prethodnog teško pratiti zbog velikog broja indeksa, uvest ćemo jednostavniji zapis u kojem ćemo $m - 1$ zajedničkih središnjih indeksa zamijeniti simbolom $*$. Dakle, uz pokratu $* = i + 1, \dots, i + m - 1$ i zamjenu $a = i, b = i + m$, izraz (5.28) biva reduciran na oblik

$$P_{a*b} = \frac{(x - x_b)P_{a*} + (x_a - x)P_{*b}}{x_a - x_b}. \quad (5.29)$$

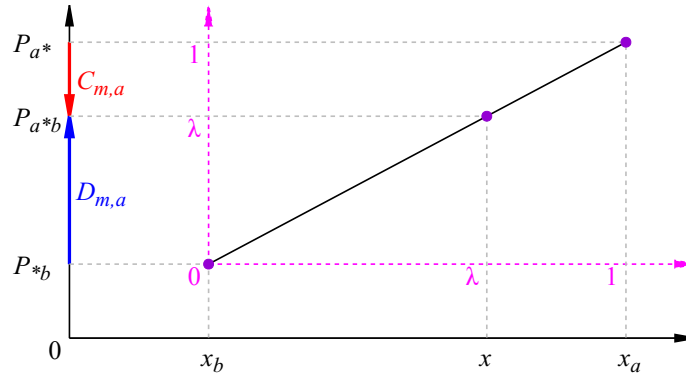
Uvedemo li pokratu

$$\lambda \equiv \frac{x - x_b}{x_a - x_b} \Rightarrow x = x_b + \lambda(x_a - x_b), \quad (5.30)$$

reducirani izraz možemo napisati u obliku još jednostavnije rekurzivne relacije

$$P_{a*b} = \lambda P_{a*} + (1 - \lambda)P_{*b} \Rightarrow P_{a*b} = P_{*b} + \lambda(P_{a*} - P_{*b}). \quad (5.31)$$

Dakle, relativan položaj vrijednosti P_{a*b} između P_{*b} i P_{a*} jednaka je relativnom položaju x između x_b i x_a što je vidljivo iz geometrijske interpretacije na slici 5.4.



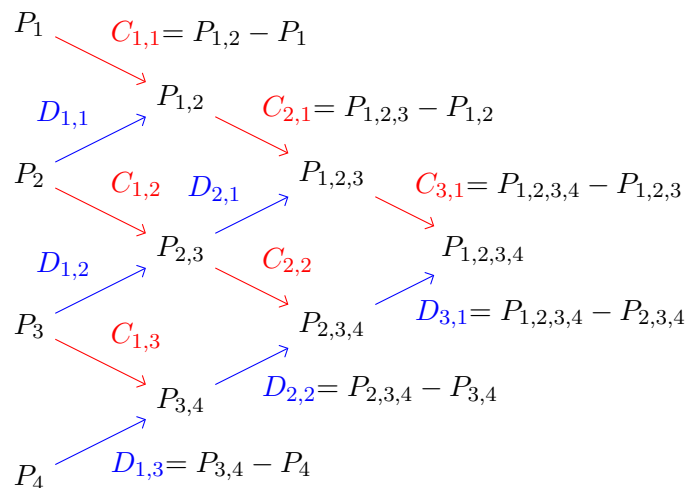
Slika 5.4: Skica korelacija između P_{a*b} , P_{a*} , P_{*b} , C , D i λ .

Proračun možemo učiniti još efikasnijim ako pratimo samo male razlike između „roditelja” i „potomaka”. U tu svrhu definiramo $m = 1, 2, \dots, n$ razlika¹

$$C_{m,a} \equiv P_{a*b} - P_{a*} \quad (5.32a)$$

$$D_{m,a} \equiv P_{a*b} - P_{*b} \quad (5.32b)$$

čije je geometrijsko značenje predloženo na slici 5.4, a na slici 5.5 dan je shematski prikaz za čvorove iz \mathbb{T}_4 . Iz (5.31) slijede rekurzivne relacije za razlike



Slika 5.5: Razlike C i D korištene u Nevillovom algoritmu. [7]

$$D_{m+1,a} = \frac{(x_{b+1} - x)(C_{m,a+1} - D_{m,a})}{x_a - x_{b+1}}, \quad (5.33a)$$

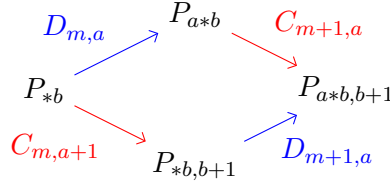
$$C_{m+1,a} = \frac{(x_a - x)(C_{m,a+1} - D_{m,a})}{x_a - x_{b+1}}. \quad (5.33b)$$

Provjeru možemo jednostavno izvesti promotrimo li neki od paralelograma shematskog prikaza na

¹U * se nalazi niz od $m - 1$ indeksa.

slici 5.5 i 5.6, pišući pri tom opće indekse, npr.

$$\begin{aligned}
C_{m+1,a} &= P_{a*b,b+1} - P_{a*b} = \lambda P_{a*b} + (1 - \lambda)P_{*b,b+1} - P_{a*b} \\
&= (1 - \lambda)(P_{*b,b+1} - P_{a*b}) \\
&= (1 - \lambda)[(P_{*b,b+1} - P_{*b} - (P_{a*b} - P_{*b}))] \\
&= (1 - \lambda)(C_{m,a+1} - D_{m,a}).
\end{aligned} \tag{5.34}$$



Slika 5.6: Skica razlika korištenih u dokazu (5.34).

Razlike C i D zapravo su korekcije koje čine interpolaciju jedan red većom. Dakle, NA, koji uzima \mathbb{T}_n kao ulazne podatke za interpolaciju, vraća u točki x konačnu vrijednost

$$P(x) = P_{1,2,\dots,n,n+1} \tag{5.35}$$

koja je dobivena kao suma nekog y_i te korekcija C i/ili D koje se nalaze na putanji kroz „obiteljsko stablo”, od y_i do krajnjeg „potomka” na desnoj strani $P_{1,2,\dots,n,n+1}$ (slično nekoj putanji sa slike 5.5 koja spaja neki y_i i $P_{1,2,3,4}$). Optimalno je birati najkraću putanju od y_i do konačne procjene $P(x)$. Sukcesivne korekcije trebale bi se smanjivati pa se veličina zadnje korekcije dy može koristiti kao procjena greške. To je, uz efikasnost, također još jedna prednost NA-a u odnosu na direktnu primjenu Lagrangeovog interpolacijskog polinoma.

Radi jednostavnijeg kodiranja umjesto uvedenih pokrata $a = i$, $b = i + m$ vratit ćemo u koeficijente (5.33) originalne indekse (stupac, redak) te trenutni stupac izraziti preko prethodnih ($m \rightarrow m - 1$):

$$D_{m,i} = \frac{(x_{i+m} - x)(C_{m-1,i+1} - D_{m-1,i})}{x_i - x_{i+m}}, \tag{5.36a}$$

$$C_{m,i} = \frac{(x_i - x)(C_{m-1,i+1} - D_{m-1,i})}{x_i - x_{i+m}}. \tag{5.36b}$$

Analizom dobivenih rekursivnih relacija (5.36) možemo primijetiti da koeficijenti C i D iz prethodnog stupca ($m - 1$) služe samo za izračunati razlike u brojnicima, a one u i -tom retku imaju indekse $\geq i$. Znači, ako u m -tom stupcu krenemo računati od prvog retka $i = 1$ prema $i = N_P - m$, gdje je

$$N_P = n + 1 \tag{5.37}$$

broj zadanih čvorova, te pohranimo privremenu razliku

$$\mathbf{CD} \equiv C_{m-1,i+1} - D_{m-1,i}, \tag{5.38}$$

možemo iskoristiti ista (dovoljno je jednodimenzionalna) polja za pohranu koeficijenata C i D koja su služila za pohranu njihovih vrijednosti u prethodnome stupcu. Kako bi optimizirati upotrebu memorije, primijetimo da su nazivnici u C i D jednaki razlici

$$\mathbf{odnos'} \equiv \mathbf{bCx} - \mathbf{bDx} = x_i - x_{i+m} \tag{5.39}$$

razlika \mathbf{x} -eva u brojnicima od \mathbf{C} ($\mathbf{bCx} \equiv x_i - x$) i \mathbf{D} ($\mathbf{bDx} \equiv x_{i+m} - x$). Također, kako bismo reducirali broj operacija, u „odnos” možemo dodati i zajednički faktor \mathbf{CD}

$$\mathbf{odnos} \equiv \mathbf{CD} / \mathbf{odnos'} \tag{5.40}$$

a u kodu **odnos** i **odnos** mogu biti iste varijable. Dakle, imamo $N_p - 1$ stupaca za C i D, a u m -tom stupcu $i = 1, \dots, N_p - m$ redaka s rekurzivno povezanim elementima

$$C_i = \mathbf{bCx} \cdot \mathbf{odnos}, \quad D_i = \mathbf{bDx} \cdot \mathbf{odnos}, \quad (5.41)$$

čije su početne vrijednosti (nulti stupac radi rekurzivnih veza (5.36)) jednake zadanim vrijednostima funkcije, $C_i = D_i = y_i$ za $i = 1, \dots, N_p$.

Funkcija, koja obavlja polinomnu interpolaciju koristeći NA,

```
void polint(float xa[], float ya[], int NP, float x, float *y, float *dy)
```

preuzeta je iz *Numerical Recipes* (NR) [2] i dana u datoteci `polint`. Ista pretpostavlja da prvi element polja ima indeks 1. Uzima $N_p = n + 1$ ulaznih parova točaka ($xa[i] = x_i$, $ya[i] = y_i$) i točku x u kojoj želimo procijeniti $P(x)$. Pozivom

```
polint(xa, ya, NP, x, &y, &dy);
```

mijenjanju se vrijednosti varijabli:

$y \rightarrow$ procjena $P(x)$;

$dy \rightarrow$ zadnja primijenjena korekcija koja se može koristiti za procjenu greške.

U NR [2], za one koji koriste C, dane su i funkcije koje omogućavaju definiciju polja čiji se elementi ne indeksiraju nužno od 0 već po želji od MIN do MAX.

Kompletna kod funkcije *polint* preved u py-kod dan je u nastavku:

```
# polint radi polinomnu interpolaciju na temelju Nevillea algoritma
# ulaz: NP parova podataka (xi,yi) i argument x
# izlaz: vrijednost polinoma yN u točki x i greska procjene dy
def polint(xi, yi, NP, x):
    ns=1; C=[0]; D=[0]; xa=[0]; ya=[0];
    for i in range(NP):
        xa.append(xi[i])
        ya.append(yi[i])
    # trazimo najblizeg susjeda (ns) od x
    mdx=abs(x-xa[1]) # udaljenost tocke x od 1. cvora
    for i in range(1,NP+1):
        dx=abs(x-xa[i])
        if (dx < mdx): # udaljenost od ostalih cvorova
            ns=i
            mdx=dx # minimala udaljenost
    # pocetne vrijednosti (nulti stupac)
    C.append(ya[i])
    D.append(ya[i])
    yN=ya[ns] # prva aproksimacija (prvi stupac)
    ns=ns-1
    for m in range(1,NP): # za svaki stupac
        for i in range(1,NP-m+1): # za svaki redak
            bCx=xa[i]-x # brojnik u C: razlika x-eva
            bDx=xa[i+m]-x # brojnik u D: razlika x-eva
            CD=C[i+1]-D[i] # razlika iz prethodnog stupca
            # stop ako postoje zaokruženo-isti xi
            odnos=bCx-bDx
            if (odnos == 0.0): sys.exit("STOP: dijeljenje s 0!")
            odnos=CD/odnos
            D[i]=bDx*odnos # koeficijenti C i D
            C[i]=bCx*odnos
    ''' Nakon svakog izracuna stupaca C-ova i D-ova slijedi odluka:
    koju korekciju dy = (C ili D) dodati (oduzeti) aproksimaciji y.
    Biramo sto ravniju (kracu) putanju kroz tablicu (efikasnije).
    Parcijalne aproksimacije ostaju centrirane obzirom na x-->P(x).
```

```

Pri tom naravno treba paziti gdje smo (umanjiti ns po potrebi).
Korekcije dy se smanjuju pa zadnja predstavlja gresku procjene. '''
if (2*ns < (NP-m)):
    dy=C[ns+1]
else:
    dy = D[ns]
    ns = ns - 1
yN += dy
return yN, dy

```

Analogno rezultate dobivamo pozivom py-funkcije:

```
y, dy = polint(xa, ya, NP, x)
```

Zadatak 5.2 JDW H↓-H↓ potencijal^a, dobiven poprilično složenim vremenski zahtjevnim proračunom^b, dan je u tablici 1 za nekoliko H↓-H↓ udaljenosti r . Na JDW podatke glatko se nastavlja dugodosežni van der Waalsov asimptotski oblik [13]

$$V_{HH}(r \gg) = -C_6 r^{-6}, \quad C_6 \approx 45064 \text{ K } \text{\AA}^{-6}. \quad (5.42)$$

U mapi `5 iPol` u datoteci `V(H-H).txt` upisani su JDW podaci iz tablice 1.

- Primijetite da su podaci dani u različitim jedinicama. Učitajte podatke i obavite pretvorbu podataka $a_0 \rightarrow \text{\AA}$ i hartree $\rightarrow \text{K}$ ako je $1 a_0 = 0.52917721092 \text{\AA}$ te $1 \text{ hartree} = 315775.04 \text{ K}$. Pohranite $(r/\text{\AA}, V/\text{K})$ u `V(H-H)_AK.txt`.
- Skicirajte algoritam za polinomnu interpolaciju pomoću Lagrangeovog oblika. Definirajte funkciju **lagrange** koja uzima N_P parova podataka (x_a, y_a) u kojima su pohranjene vrijednosti $(r_i/\text{\AA}, V_i/\text{K})$, udaljenost $x = r$ te Lagrangeovim postupkom procjenjuje $V(x) \approx y_L$.
- Koristeći poznate podatke $V(r_i)$ procijenite vrijednosti funkcije $V(x)$ u 71 ekvidistantnoj točki iz intervala $2.81 \text{\AA} \leq x \leq 9.81 \text{\AA}$. Radi lakše usporedbe pohranite u `V(H-H)_inter.txt` redom po stupcima

$$\{ x = r, y_L \}.$$

- Dopunite procjenu koristeći Nevillev algoritam, odnosno priloženu funkciju `polint`. Radi lakše usporedbe pohranite u `V(H-H)_inter.txt` redom po stupcima udaljenost, procjenu Lagrangeovim polinomom te procjenu i grešku dobivene Nevillevim algoritmom koristeći `polint`, odnosno

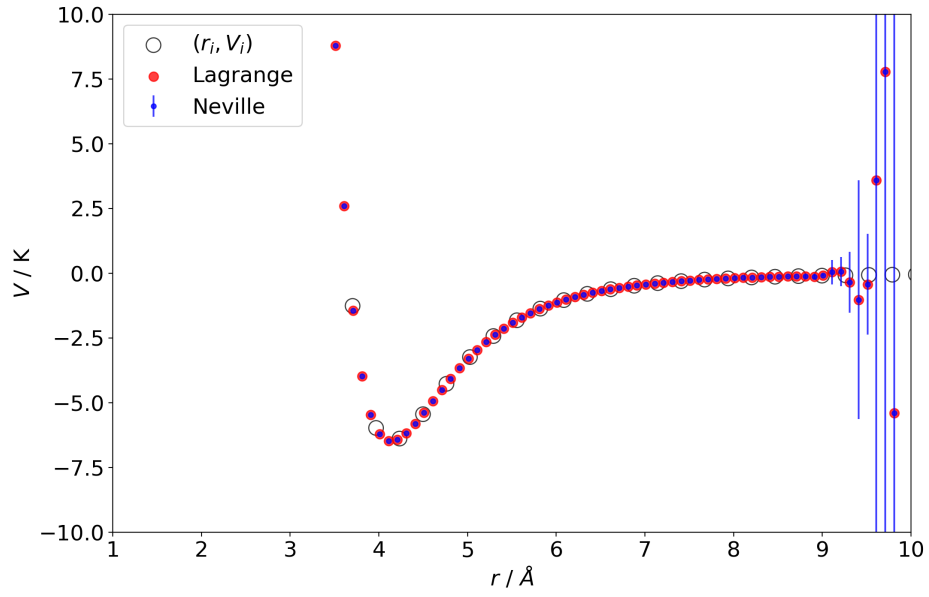
$$\{ x = r, y_L, y_P, dy_P \}.$$

- Usporedite ulazne i izračunate podatke na grafu kao na slici 5.7. U tu svrhu doradite priložene ili napravite nove plt/py-skripte za crtanje grafova. Diskutirajte točnost rezultata.



^aČesto se za potencijalnu energiju koristi skraćeni izraz, odnosno samo „potencijal”.

^bInterakciju atoma spin-polariziranog vodika H↓-H↓ odredili su na egzaktn način Kolos i Wolniewicz [10] još 1964. Objavili su vrijednosti spin-neovisnog centralnog potencijala tripletnog para $b^3\Sigma_u^+$ za odabrane međuatomske udaljenosti $a_0 \leq r \leq 10a_0$, a 1974. poboljšali su njegov oblik u području $6a_0 \leq r \leq 12a_0$. Proširenje na područje $a_0 \leq r \leq 20a_0$ i precizniji proračun u točkama razmaknutima za $0.5a_0$ napravili su tek 2000. godine Jamieson, Dalgarno i Wolniewicz (JDW) [11]. Razlika između KW i JDW potencijala mala je u području gdje se mogu usporediti, a među raznim fitovima kubična *spline* interpolacija pokazala se idealnom [12]. Tako se u složenijim proračunima najčešće koriste JDW podaci dani u tablici 1, dok se u ostalim točkama vrijednosti procjenjuju kubičnom *spline* interpolacijom JDW podataka.



Slika 5.7: Rješenje zadatka 5.2.

Tablica 1: Vrijednosti JDW potencijala [11] s oduzetim vrijednostima za $r = +\infty$.

r/a_0	$V_{\text{JDW}}/\text{hartree}$	r/a_0	$V_{\text{JDW}}/\text{hartree}$
1.0	0.377735694	11.0	-0.000004323
1.5	0.190333563	11.5	-0.000003281
2.0	0.102923717	12.0	-0.000002515
2.5	0.054546281	12.5	-0.000001948
3.0	0.027984965	13.0	-0.000001524
3.5	0.013869115	13.5	-0.000001204
4.0	0.006619941	14.0	-0.000000960
4.5	0.003027120	14.5	-0.000000772
5.0	0.001312747	15.0	-0.000000625
5.5	0.000528252	15.5	-0.000000510
6.0	0.000186553	16.0	-0.000000420
6.5	0.000047167	16.5	-0.000000347
7.0	-0.000003987	17.0	-0.000000289
7.5	-0.000018912	17.5	-0.000000242
8.0	-0.000020214	18.0	-0.000000203
8.5	-0.000017232	18.5	-0.000000172
9.0	-0.000013517	19.0	-0.000000146
9.5	-0.000010246	19.5	-0.000000125
10.0	-0.000007674	20.0	-0.000000107
10.5	-0.000005744	$+\infty$	0

5.2 Spline interpolacija

Analiza rezultata u prethodnom poglavlju pokazala je kako polinomna interpolacija može rezultirati raznovrsnom kvalitetom rezultata, od savršenog slaganja (slika 5.2) do ogromnih odstupanja kada čak i greška metode nekad može postati apsolutno veća od rezultata (slika 5.7). Odstupanja se obično dogode blizu ruba, a nekad ih nije lako izbjeći ni korištenjem metoda koje omogućavaju njihovu redukciju. Obično za manji uzorak podataka dobiveni interpolacijski polinom jednostavno slijedi čvorne točke, dok preveliki uzorci znaju rezultirati pojavom neočekivanih oscilacija između čvorova. K tome polinomi velikog stupnja nisu zgodni za korištenje. Stoga se prirodno nameće ideja „izbjegavanja polinoma velikog stupnja”. Jedan način realizacije jest podjela zadanih čvorova na manje segmente i provlačenjem polinoma manjeg stupnja na svakom pojedinom segmentu zahtijevajući željeni stupanj glatkoće među segmentima. Takve po dijelovima polinomne funkcije, koje zadovoljavaju zadane uvjete glatkoće, zovemo polinomne *spline* (izg. splajn) funkcije.

Neka poznajemo $n + 1$ vrijednost funkcije f , $y_i = f(x_i)$ za $i = 0, \dots, n$. Za veliki n , umjesto aproksimacije interpolacijskim polinomom $P_n(x)$, zgodnije je definirati restrikcije na svakom podintervalu

$$\varphi_i : [x_{i-1}, x_i] \rightarrow \mathbb{R}$$

gdje za φ_i odabiremo jednostavnu funkciju dovoljnog stupnja glatкости (obično neki polinom) pa ćemo u poglavlju 5.2.1 analizirati najjednostavniji slučaj - polinom prvog stupnja koji je zgodan kada je gust skup ulaznih podataka.

Često se u diferencijalnim jednadžbama koje opisuju fizikalne probleme javljaju druge derivacije, primjerice u Newtonovim zakonima te Poissonovoj, Schrödingerovoj, valnoj i difuzijskoj jednadžbi... Stoga nam je ovdje cilj dobiti rješenje koje će imati neprekidne derivacije drugog reda. To naravno možemo postići birajući minimalno po dijelovima kubične polinome pa ćemo se u nastavku (pgl. 5.2.2) analizirati algoritme za konstrukciju kubičnog interpolacijskog splajna.

5.2.1 Linearni interpolacijski splajn

Pretpostavimo da je $f : [a, b] \rightarrow \mathbb{R}$ neprekidna i da znamo vrijednosti u $n + 1$ čvorova

$$a \equiv x_0 < x_1 < \dots < x_n \equiv b, \quad y_i = f(x_i). \quad (5.43)$$

Linearne funkcije na svakom podintervalu $[x_{i-1}, x_i]$ možemo dobiti provlačeći pravce kroz čvorne točke (x_{i-1}, y_{i-1}) i (x_i, y_i) ,

$$\varphi_i(x) = y_{i-1} + \frac{y_i - y_{i-1}}{x_i - x_{i-1}}(x - x_{i-1}), \quad x \in [x_{i-1}, x_i]. \quad (5.44)$$

Tako dobivenu po dijelovima linearnu funkciju $\varphi(x)$, za koji vrijedi $\varphi|_{[x_{i-1}, x_i]} = \varphi_i$, nazivamo linearni interpolacijski splajn. Nedostaci jedinstvenog po dijelovima linearnog interpolacijskog splajna su što nije derivabilan u čvorovima te rezultira većim greškama ako se derivacija funkcije f značajno mijenja između čvorova.

5.2.2 Kubični interpolacijski splajn

Dva puta derivabilnu funkciju $C : [a, b] \rightarrow \mathbb{R}$ možemo dobiti provlačeći minimalno polinom za jedan red veći između susjednih čvorova, odnosno polinom trećeg stupnja

$$C(x) = C_i(x), \quad x \in [x_{i-1}, x_i], \quad i = 1, \dots, n. \quad (5.45)$$

Kako bi $C(x)$ bio klase $C_{[a,b]}^2$, dijelovi C_i moraju zadovoljavati uvjete

$$\begin{aligned} \text{(U1)} \quad & C_i(x_{i-1}) = y_{i-1} & i = 1, \dots, n; \\ \text{(U2)} \quad & C_i(x_i) = y_i & i = 1, \dots, n; \\ \text{(U3)} \quad & C'_i(x_i) = C'_{i+1}(x_i) & i = 1, \dots, n-1; \\ \text{(U4)} \quad & C''_i(x_i) = C''_{i+1}(x_i) & i = 1, \dots, n-1; \end{aligned} \quad (5.46)$$

čime dobivamo skup on $4n - 2$ jednadžbi, a imamo $4n$ neodređenih koeficijenata jer provlačimo n kubičnih polinoma. Rubni uvjeti daju preostale 2 jednadžbe koje su potrebne za određivanje jedinstvene funkcije $\varphi(x) = C(x) : [a, b] \rightarrow \mathbb{R}$. Postoji više izbora što uzeti za rubni uvjet:

a) prve derivacije na rubu (**potpuni splajn**),

$$C'_1(x_0) = f'(x_0), \quad C'_n(x_n) = f'(x_n);$$

b) druge derivacije na rubu

$$C''_1(x_0) = f''(x_0), \quad C''_n(x_n) = f''(x_n);$$

c) druge derivacije postaviti na nulu, odnosno ostaviti slobodne rubove (**prirodni splajn**)

$$C''_1(x_0) = 0, \quad C''_n(x_n) = 0;$$

d) numerički aproksimirati derivacije na rubu.

Prva dva izbora su česta kada poznajemo f na rubu, a ako je nepoznat preostaju nam zadnja dva izbora. Greške procjene uglavnom su reda veličine $\mathcal{O}(h^4)$, gdje je h razmak između susjednih čvorova. U slučaju prirodnog splajna greška može porasti na $\mathcal{O}(h^2)$ kada f'' na rubu nije približno nula, a kod numeričkog računanja ruba uvelike ovisi o grešci procjene derivacije.

Teorem 5.3 *Teorem o jedinstvenosti kubičnog splajna.*

Neka je skup poznatih parova podataka $\{ (x_i, y_i = f(x_i)) \mid i = 0, \dots, (n \in \mathbb{N}_0) \}$ uređen tako da vrijedi $x_i < x_j$ za $i < j$. Tada postoji jedinstveni prirodni kubični interpolacijski splajn $C(x) = C_i(x)$ za $x_{i-1} \leq x \leq x_i$,

$$C_i(x) = y_{i-1} - \frac{s_{i-1}}{6} h_i^2 + b_i(x - x_{i-1}) + \frac{s_{i-1}}{6h_i}(x_i - x)^3 + \frac{s_i}{6h_i}(x - x_{i-1})^3 \quad (5.47)$$

gdje su korištene pokrate

$$b_i = d_i - (s_i - s_{i-1}) \frac{h_i}{6}, \quad d_i = \frac{y_i - y_{i-1}}{h_i}, \quad h_i = x_i - x_{i-1}, \quad (5.48)$$

uz druge derivacije $\{ s_i \equiv C''(x_i) \mid i = 1, \dots, (n-1) \}$ dobivene rješavajući tridijagonalni sustav od $n - 2$ jednadžbe

$$h_i \cdot s_i + 2(h_i + h_{i+1}) \cdot s_{i+1} + h_{i+1} \cdot s_{i+2} = 6(d_{i+1} - d_i), \quad i = 1, \dots, n-2 \quad (5.49)$$

uz uvjet

$$s_0 = s_n = 0 \quad (5.50)$$

Dokaz i primjer dani su u dodatku A.1.

U pythonu možemo koristiti kubični splajn interpolator is biblioteke SciPy [5]:

```
scipy.interpolate.CubicSpline(xa, ya, axis=0, bc_type='not-a-knot', extrapolate=None)
```

gdje su kao i prije (x_a, y_a) ulazni podaci, **axis** (zadano 0) jest os duže koje se mijenja y , **bc_type** (eng. *boundary condition type*) tip je rubnih uvjeta te **extrapolate** opcionalni parametar kojim podešavamo ekstrapolaciju (zadano **None**). Ako je **bc_type=**

- 'not-a-knot' - polinomi su isti na 2 rubna segmenta (zadano);
- 'periodic' - funkcija je periodična $y[0]==y[-1]$ pa je $y'[0]==y'[-1]$ i $y''[0]==y''[-1]$;
- $((n, y^{(n)}[0]), (n, y^{(n)}[-1]))$ - funkcija na rubovima ima n -te derivacije $y^{(n)}[0]$ i $y^{(n)}[-1]$;
- 'clamped' - prve derivacije iščezavaju na rubovima, ekvivalentno $bc_type=((1, 0.0), (1, 0.0))$;
- 'natural' - druge derivacije iščezavaju na rubovima, ekvivalentno $bc_type=((2, 0.0), (2, 0.0))$.

Interpolator konstruira **PPoly**² instancu

```
scipy.interpolate.PPoly(c, x, extrapolate=None, axis=0)
```

s prijelomnim točkama koje odgovaraju danim podacima, odnosno razlomljeni polinom zadan koeficijentima c i prijelomnim točkama x , primjerice polinom k -tog stupnja između $x[i]$ i $x[i+1]$ glasi

```
S = sum(c[m, i] * (xp - x[i])** (k-m) for m in range(k+1))
```

Ako rezultat interpolacije pohranimo u instancu **cs**, trivijano je izračunati njenu n -tu derivaciju u nizu točaka **xs** pozivom:

cs(xs, n)

Primjerice neka ulazne podatke čine vrijednosti $y_a = \sin(x_a)$ u 10 ekvivistantnih točaka x_a od 0 do 9. Analizirat ćemo usporedbu splajna i njegovih derivacija na intervalu $[-0.5, 9.6]$ ³:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import CubicSpline

# generiranje 10 parova ulaznih podataka
x = np.arange(10)
y = np.sin(x)

# konstrukcija splajn polinoma
cs = CubicSpline(x, y)

# kreiranje intervala za provjeru
xs = np.arange(-0.5, 9.6, 0.1)

# graficka usporedba
fig, ax = plt.subplots(figsize=(6.5, 4))
ax.plot(x, y, 'o', label='data')          # ulazni podaci
ax.plot(xs, np.sin(xs), label='true')     # egzakta funkcija
ax.plot(xs, cs(xs), label='S')           # splajn
ax.plot(xs, cs(xs, 1), label='S\'')      # 1. derivacija splajna
ax.plot(xs, cs(xs, 2), label='S\'\'')    # 2. derivacija splajna
ax.plot(xs, cs(xs, 3), label='S\'\'\'')   # 3. derivacija splajna
ax.set_xlim(-0.5, 9.5)
ax.legend(loc='lower left', ncol=2)
plt.show()
```

²<https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.PPoly.html>

³<https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.CubicSpline.html>

Zadatak 5.3 Dopunite procjene iz zadatka 5.2 tako da u `V(H-H)_inter.txt` pohranite

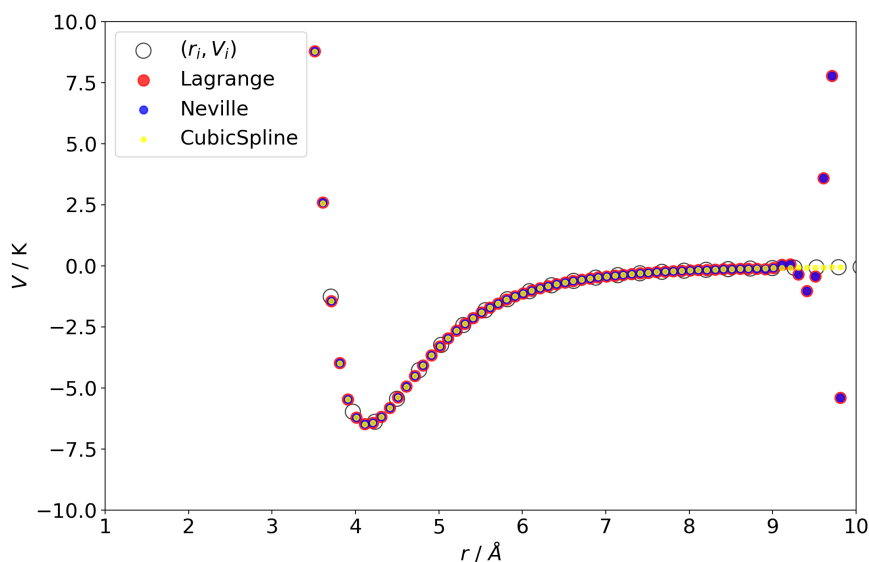
$$\{ x = r, y_L, y_p, dy_p, y_s, y_p - y_s \}$$

gdje je $V(x) \approx y_s$ procjena dobivena potpunim splajnom. Doradite plt/py-skripte kako biste usporedili podatke na grafu, slično usporedbi na slici 5.8. Provjerite rješenja uspoređujući vlastita s odgovarajućim priloženim datotekama `* - Copy.*`.

Zadatak 5.4 Analizirajte rezultate dobivene u zadatku 5.3:

- Ovisi li točnost o odabiru vrijednosti x ? Je li bolja točnost s više ulaznih točaka? Provjerite odbacivanjem pojedinih.
- Usporedite spline metodu i polinomnu interpolaciju. Kakvo je njihovo ponašanje van danog intervala?
- Kako biste odredili koeficijente polinoma za analitički zapis istih?

Pitanja su dana kao smjernice za razmišljanje i testiranje. ◀



Slika 5.8: Rješenje zadatka 5.3.

A Interpolacija

A.1 Dokaz jedinstvenosti prirodnog kubičnog splajna i primjer

➤ **Dokaz** $s_{i-1} := C_i''(x_{i-1}), \quad s_i := C_i''(x_i) \quad i = 1, \dots, n$

$$C_1''(x_0) = C_n''(x_n) = 0$$

$$s_0 = s_n = 0$$

$x \rightarrow C_i''(x)$ linearna

$$C_i''(x) = \frac{s_{i-1}}{h_i}(x_i - x) + \frac{s_i}{h_i}(x - x_{i-1})$$

$$C_i'(x) = b_i - \frac{s_{i-1}}{2h_i}(x_i - x)^2 + \frac{s_i}{2h_i}(x - x_{i-1})^2$$

$$C_i(x) = a_i + b_i x + \frac{s_{i-1}}{6h_i}(x_i - x)^3 + \frac{s_i}{6h_i}(x - x_{i-1})^3$$

$$(i)-(ii): \quad C_i(x_{i-1}) = y_{i-1}, \quad C_i(x_i) = y_i, \quad i = 1, \dots, n$$

$$\begin{array}{l} a_i + b_i x_{i-1} + s_{i-1} \frac{h_i^2}{6} = y_{i-1} \\ a_i + b_i x_i + s_i \frac{h_i^2}{6} = y_i \end{array} \quad \left| \begin{array}{c} - \\ + \end{array} \right. \rightarrow \begin{array}{l} b_i = d_i - (s_i - s_{i-1}) \frac{h_i}{6}, \quad d_i = \frac{y_i - y_{i-1}}{h_i} \\ a_i = -b_i x_{i-1} + y_{i-1} - s_{i-1} \frac{h_i^2}{6} \end{array}$$

$$C_i'(x) = b_i - \frac{s_{i-1}}{2h_i}(x_i - x)^2 + \frac{s_i}{2h_i}(x - x_{i-1})^2$$

$$C_i'(x_i) = b_i + \frac{s_i}{2}h_i, \quad C_{i+1}'(x_i) = b_{i+1} - \frac{s_i}{2}h_{i+1}$$

$$(iii) \quad C_i'(x_i) = C_{i+1}'(x_i), \quad i = 1, \dots, n-1$$

$$d_i - (s_i - s_{i-1}) \frac{h_i}{6} + \frac{s_i}{2}h_i = d_{i+1} - (s_{i+1} - s_i) \frac{h_{i+1}}{6} - s_i \frac{h_{i+1}}{2}$$

➤ sustav od (n-1) jednačbe s (n-1) nepoznanica

$$s_{i-1}h_i + 2s_i(h_i + h_{i+1}) + s_{i+1}h_{i+1} = 6(d_{i+1} - d_i), \quad i = 1, \dots, n-1$$

$$s_0 = s_n = 0$$

$$\mathbf{H}\mathbf{s} = \mathbf{b}$$

$$\mathbf{H} = \begin{bmatrix} 2(h_1 + h_2) & h_2 & \cdots & 0 & 0 \\ h_2 & 2(h_2 + h_3) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & 0 & \cdots & h_{n-1} & 2(h_{n-1} + h_n) \end{bmatrix}$$

$$\mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_{n-1} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 6(d_2 - d_1) \\ \vdots \\ 6(d_n - d_{n-1}) \end{bmatrix}$$

➤ H je **tridijagonalna**, dijag. dominantna, simetrična, pozitivno definitna
 ==>> ∃! rješenje ==>> ∃! prirodni kubični interpolacijski splajn

Primjer: Ako je

x_i	0	0.25	0.5	0.75	1
y_i	1	2	1	0	1

 odredite prirodni kubični interpolacijski splajn $C(x)$ i izračunajte $C(0.7)$.

Rješenje:

$$\mathbf{H}\mathbf{s} = \mathbf{b} \quad \mathbf{H} = \begin{bmatrix} 2(h_1 + h_2) & h_2 & \cdots & 0 & 0 \\ h_2 & 2(h_2 + h_3) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & 0 & \cdots & h_{n-1} & 2(h_{n-1} + h_n) \end{bmatrix}$$

$$s_0 = s_4 = 0, \quad \mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_{n-1} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 6(d_2 - d_1) \\ \vdots \\ 6(d_n - d_{n-1}) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0.25 & 0 \\ 0.25 & 1 & 0.25 \\ 0 & 0.25 & 1 \end{bmatrix} \mathbf{s} = \begin{bmatrix} -48 \\ 0 \\ 48 \end{bmatrix} \longrightarrow s_1 = -48, s_2 = 0, s_3 = 48.$$

$$C(x) = \begin{cases} C_1(x) = 1 + 6x - 32x^3, & 0 \leq x \leq 0.25 \\ C_2(x) = 18x - 48x^2 + 32x^3, & 0.25 \leq x \leq 0.5 \\ C_3(x) = 18x - 48x^2 + 32x^3, & 0.5 \leq x \leq 0.75 \\ C_4(x) = 27 - 90x + 96x^2 - 32x^3, & 0.75 \leq x \leq 1 \end{cases}$$

$$0.7 \in [0.5, 0.75] \quad C(0.7) = C_3(0.7) = 0.056.$$