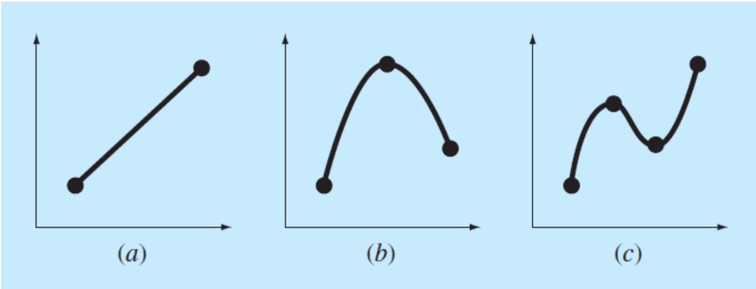
1 Интерполација

Претпоставимо да нам је вредност функције позната у одређеном броју дискретних тачака, односно да нам вредност функције није позната између тих тачака. Често се јавља потреба за проценом вредности функције између тачака у којима су познате тачне вредности функције. Најпознатији метод који се користи у ове сврхе је полиномска интерполација. Полином n-тог реда може се написати у облику

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \; . \tag{1.1}$$

Ако нам је на располагању вредност функције у $n+1$-ој тачки (интерполације), постоји само један полином реда n који пролази кроз све те тачке. Примера ради, постоји само једна права линија (односно полином првог реда) који пролази кроз две задате тачке, као што је приказано на слици 1.1(а). Слично томе, постоји само једна парабола (полином другог реда) која пролази кроз три тачке, као што је приказано на слици 1.1(б). Полиномска интерполација представља одређивање јединственог полинома n-тог рада који пролази кроз $n+1$ тачака интерполације. Помоћу ових полинома лако се процењују непознате вредности функције између тачака интерполације.



Слика 1.1.

Иако постоји само један полином n-тог реда који пролази кроз унапред задату $n+1$ тачку, овај полином може се записати на више начина, односно до тих полинома може се доћи различитим математичким поступцима. У оквиру овог поглавља приказаћемо два начина одређивања ових полинома који су погодни за имплементацију на рачунарима. Први је Њутнов интерполациони полином, а други је Лагранжов интерполациони полином.

1.1 Њутнов интерполациони полином са подељеним разликама

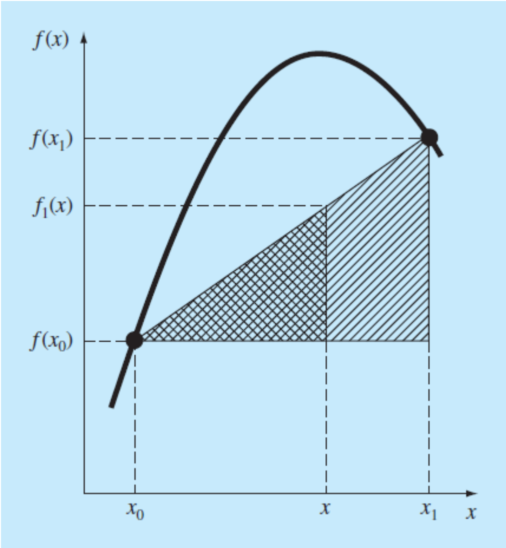
Као што је већ наведено, постоји варијетет алтернативних облика представљања интерполационих полинома. Њутнов интерполациони полином са подељеним разликама један је од најпопуларнијих облика интерполационих полинома. Пре него прикажемо општи облик ових полинома, приказаћемо Њутнов интерполациони полином првог и Њутнов интерполациони полином другог реда због њихове једноставне визуелне интерпретације. Полазећи од тих полинома, доћи ћемо и до општег облика Њутновог интерполационог полинома.

1.1.1 Линеарна интерполација

Најједноставнији облик интерполације је повезивање две интерполационе тачке правом линијом, што представља линеарну интерполацију која је приказана на слици 1.2. Користећи се сличношћу троуглова са слике 1.2 добијамо

$$\frac{f_1(x) - f(x_0)}{x - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \; , \tag{1.2}$$

што представља линеарну интерполациону формулу. У претходној формули са $f_1(x)$ смо означили да је у питању интерполациони полином првог реда. Згодно је приметити да осим што члан са десне стране знака једнакости у (1.2) представља коефицијент нагиба линије која повезује интерполационе тачке, тај члан представља и количник коначних разлика који је апроксимација првог извода функције. У општем случају, (линеарна) интерполација је боља што је краћи интервал између интерполационих тачака. Ово је директна последица тога што је са скраћењем интервала између интерполационих тачака права линија боља апроксимација континуалне функције $f(x)$. Ово својство приказано је на следећем примеру.

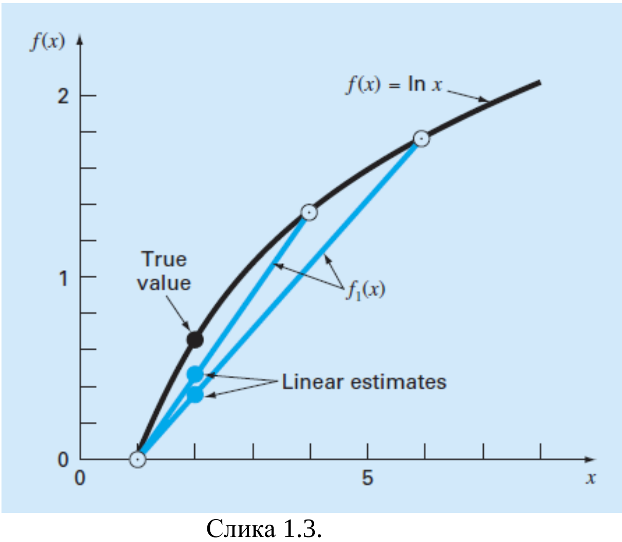


Слика 1.2.

Пример 1.1: Потребно је проценити вредност $\ln 2$ помоћу линеарне интерполације. Нека су нам, у првом случају, познате вредности функције $\ln x$ у тачкама $x_0 = 1$ и $x_1 = 6$ ($\ln 1 = 0$, $\ln 6 = 1{,}79176$), односно нека су нам познате вредности функције $\ln x$ у тачкама $x_0 = 1$ и $x_1 = 4$ ($\ln 1 = 0$ и $\ln 4 = 1{,}38629$) у другом случају. Тражена тачна вредност је $\ln 2 = 0{,}693147$.

Користећи се једначином (1.2) у првом случају добијамо $f_1(2) = \dfrac{2-1}{6-1}(1{,}79176 - 0) + 0 = 0{,}358352$, а грешка овог решења је око $48{,}3\%$.

Користећи једначину (1.2) у другом случају добијамо $f_1(2) = \dfrac{2-1}{4-1}(1{,}38629 - 0) + 0 = 0{,}462097$, а грешка овог решења је око $33{,}3\%$. Очигледно је да се грешка интерполације смањује како се интерполационе тачке приближавају тачки у којој одређујемо вредност функције, а ово је илустровано и на слици 1.3.



Слика 1.3.

1.1.2 Квадратна интерполација

Грешка у примеру 1.1 последица је апроксимације криве правом линијом. На основу тога може се закључити да је један од начина побољшања тачности интерполације додавање закривљености интерполационој линији која повезује интерполационе тачке. Ако су нам на располагању вредности функције у три (интерполационе) тачке, ове тачке могу се повезати полиномом другог реда (који се још назива и квадратни полином односно парабола). Полином другог реда може се написати на разне начине, а облик посебно погодан за конструкцију интерполационог полинома је

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) . \tag{1.3}$$

Приметити да иако једначина (1.3) на први поглед изгледа другачије од облика (1.1) у коме смо обично навикли да записујемо полиноме, за $n = 2$ у једначини (1.1) ова два облика су еквивалентна. Ово се лако може показати множењем чланова из (1.3) што, након сређивања, даје $f_2(x) = b_0 - b_1 x_0 + b_2 x_0 x_1 + (b_1 - x_0 b_2 - x_1 b_2)x + b_2 x^2$. Поређењем овог израза са изразом (1.1) за $n = 2$ добијамо да су два полиноме идентична ако је задовољено $a_0 = b_0 - b_1 x_0 + b_2 x_0 x_1$, $a_1 = b_1 - x_0 b_2 - x_1 b_2$ и $a_2 = b_2$. Према томе једначине (1.1) и (1.3) су еквивалентне (алтернативне) формулације јединственог интерполационог полинома другог реда који повезује три тачке.

Једноставна процедура може се спровести како би се одредили коефицијенти из једначине (1.3). Уврштавањем $x = x_0$ у једначину (1.3), за коефицијент b_0 добијамо

$$b_0 = f(x_0) . \tag{1.4}$$

Сада се једначина (1.4) може заменити у (1.3), а након замене $x = x_1$ у једначину (1.3) добијамо коефицијент b_1

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} . \tag{1.5}$$

Финално, једначине (1.4) и (1.5) могу се заменити у једначину (1.3), а након замене $x = x_2$ у једначину (1.3) добијамо коефицијент b_2

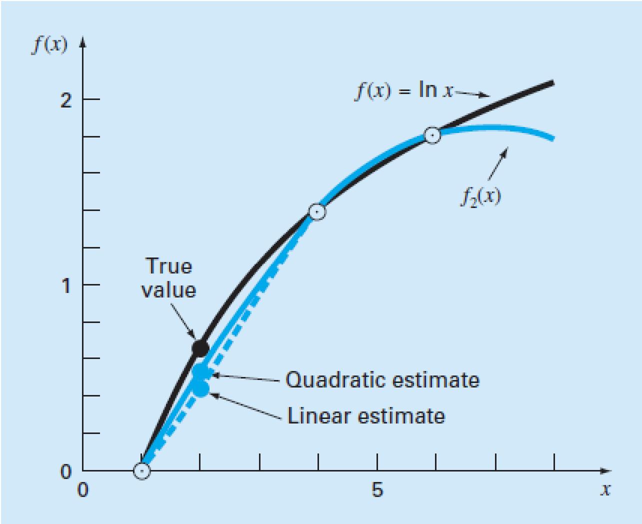
$$b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} . \tag{1.6}$$

Приметити да, као и у случају линеарне интерполације, коефицијент b_1 одговара коефицијенту нагиба праве линије која пролази кроз интерполационе тачке са координатама x_0 и x_1, а коефицијент b_0 одговара вредности функције у интерполационој тачки са координатом x_0. Према томе, прва два члана у (1.3) (који садрже b_0 и b_1) одговарају линеарној интерполацији (1.2). Последњи члан, $b_2(x - x_0)(x - x_1)$, додаје закривљеност (полином другог реда) у интерполациону формулу (1.3).

Пре него наставимо даље, испитајмо облик формуле за прорачун b_2. На основу (1.6) видимо да се коефицијент b_2 прорачунава количником коначних разлика који апроксимира други извод функције. Према томе, формула (1.3) почиње да открива структуру врло сличну Тејлоровом развоју. Ово ће бити додатно разјашњено када се Њутнови интерполациони полиноми буду упоредили са Тејлоровим развојем у одељку 1.1.4. Пре свега тога, кроз пример 1.2, који је сличан примеру 1.1, приказаћемо примену формуле (1.3).

Пример 1.2: Потребно је проценити вредност функције $\ln x$ у тачки $x = 2$ користећи интерполациони полином другог реда који пролази кроз интерполационе тачке са координатама $x_0 = 1$, $x_1 = 4$ и $x_2 = 6$ ($f(x_0) = 0$, $f(x_1) = 1{,}38629$ и $f(x_2) = 1{,}79176$).

Користећи се формулама (1.4)-(1.6) одређујемо коефицијенте $b_0 = 0$, $b_1 = 0{,}462098$ и $b_2 = -0{,}0518731$, па је процењена вредност функције $\ln x$ за $x = 2$ $f_2(x)\big|_{x=2} = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)\big|_{x=2} = 0{,}565844$, што резултује апсолутном грешком од око $18{,}4\%$. Према томе, закривљење које је последњи члан у (1.3) увео у интерполациони полином резултује смањењем грешке интерполације у односу на пример 1.1 када тога члана није било. Како овај члан доприноси тачности интерполације може се закључити и на основу слике 1.4.



Слика 1.4.

1.1.3 Општи облик Њутнових интерполационих полинома

Претходна анализа може се уопштити на полиноме n-тог реда који повезују $n+1$ интерполациону тачку. Општи облик Њутновог интерполационог полинома n-тог реда је

$$f_n(x)=b_0+b_1(x-x_0)+b_2(x-x_0)(x-x_1)+\ldots+b_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}). \tag{1.7}$$

Слично као и у претходним случајевима линеарне и квадратне интерполације, координате тачака интерполације користе се за одређивање коефицијената интерполационог полинома $b_0, b_1, \ldots, b_n$. За одређивање интерполационог полинома n-тог реда потребна је $n+1$ интерполациона тачака које чине парове $(x_0, f(x_0))$, $(x_1, f(x_1))$, ..., $(x_n, f(x_n))$ помоћу којих се одређују коефицијенти Њутновог интерполационог полинома као

$$b_0 = f(x_0), \tag{1.8}$$

$$b_1 = f[x_1, x_0], \tag{1.9}$$

$$b_2 = f[x_2, x_1, x_0], \tag{1.10}$$

$$\vdots$$

$$b_n = f[x_n, x_{n-1}, \ldots, x_1, x_0], \tag{1.11}$$

при чему угласте заграде представљају оператор количника коначних разлика. На пример, количник коначних разлика првог реда је

$$f[x_i, x_j]=\frac{f(x_i)-f(x_j)}{x_i-x_j}. \tag{1.12}$$

Количник коначних разлика другог реда, који представља количник коначних разлика количника коначних разлика првог реда је

$$f[x_i, x_j, x_k]=\frac{f[x_i, x_j]-f[x_j, x_k]}{x_i-x_k}. \tag{1.13}$$

Финално, количник коначних разлика n-тог реда је

$$f[x_n, x_{n-1}, \ldots, x_1, x_0]=\frac{f[x_n, x_{n-1}, \ldots, x_2, x_1]-f[x_{n-1}, x_{n-2}, \ldots, x_1, x_0]}{x_n-x_0}. \tag{1.14}$$

Ове разлике могу се искористити како би се прорачунали коефицијенти из једначина (1.8)-(1.11), који се након тога могу заменити у формулу (1.7) како би се добио интерполациони полином

$$\begin{aligned} f_n(x)= {} & f(x_0)+f[x_1, x_0](x-x_0)+f[x_2, x_1, x_0](x-x_0)(x-x_1)+ \\ & +\ldots+f[x_n, x_{n-1}, \ldots, x_1, x_0](x-x_0)(x-x_1)\cdots(x-x_{n-2})(x-x_{n-1}), \end{aligned} \tag{1.15}$$

који се назива Њутнов интерполациони полином са подељеним разликама. Згодно је приметити да није неопходно да интерполационе тачке у формули (1.15) буду еквидистантне, нити да су вредности

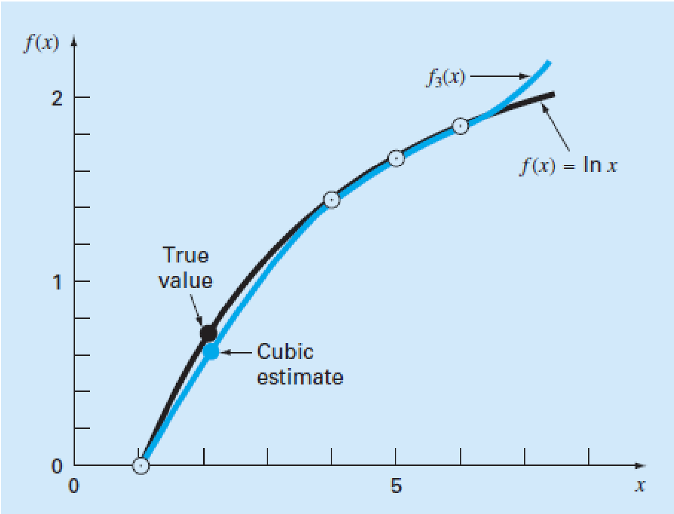
$x_0, x_1, ..., x_n$ у растућем редоследу, као што ће бити приказано у наредном примеру. Такође, згодно је приметити да су формуле (1.12)-(1.14) рекурзивне, то јесте коначне разлике вишег реда рачунају се као коначне разлике коначних разлика нижег реда. Овај процес (рекурзија) илустрована је на слици 1.5. Ово својство биће искоришћено када будемо конструисали ефикасан програм за имплементацију Њутнових интерполационих полинома, приказан у одељку 1.1.5.

i	x_i	$f(x_i)$	First	Second	Third
0	x_0	$f(x_0)$	$f[x_1, x_0]$	$f[x_2, x_1, x_0]$	$f[x_3, x_2, x_1, x_0]$
1	x_1	$f(x_1)$	$f[x_2, x_1]$	$f[x_3, x_2, x_1]$	
2	x_2	$f(x_2)$	$f[x_3, x_2]$		
3	x_3	$f(x_3)$			

Слика 1.5.

Пример 1.3: У односу на пример 1.2, на располагању је још једна интерполациона тачка функције $\ln x$ ($x_3 = 5$, $\ln 5 = 1{,}60944$). На основу ових података конструисати интерполациони полином трећег реда и проценити вредност $\ln 2$.

Количници коначних разлика првог реда су $f[x_3, x_2] = 0{,}182322$, $f[x_2, x_1] = 0{,}202733$ и $f[x_1, x_0] = 0{,}462098$. Количници коначних разлика другог реда су $f[x_3, x_2, x_1] = \dfrac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1} = -0{,}020411$ и $f[x_2, x_1, x_0] = \dfrac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = -0{,}0518731$, а количник коначних разлика трећег реда је $f[x_3, x_2, x_1, x_0] = \dfrac{f[x_3, x_2, x_1] - f[x_2, x_1, x_0]}{x_3 - x_0} = 0{,}00786553$. На основу (1.15) Њутнов интерполациони полином трећег реда је $f_3(x) = 0 + 0{,}462098\,(x - x_0) - 0{,}0518731\,(x - x_0)(x - x_1) + 0{,}00786553\,(x - x_0)(x - x_1)(x - x_2)$ чија вредност за $x = 2$ износи $f_3(2) = 0{,}628769$. Апсолутна вредност грешке је око $9{,}3\%$. Изглед интерполационог полинома $f_3(x)$ приказан је на слици 1.6.



Слика 1.6.

1.1.4 Грешка Њутнових интерполационих полинома

Приметимо да је структура формуле (1.15) слична Тејлоровом развоју по томе што се сукцесивно додају чланови развоја који би требало да опишу понашање функције вишег реда. У случају формуле (1.15) ови чланови су виши редови количника коначних разлика који представљају апроксимацију извода вишег реда. Последично, слично као и у случају Тејлоровог развоја, ако је функција која се интерполира полином n-тог реда, интерполациони полином n-тог реда који пролази кроз $n+1$ тачака функције која се интерполира резултоваће тачним решењем.

Грешка Тејлоровог развоја реда n функције $f(x)$ може се записати у облику $e_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x_{i+1} - x_i)^{n+1}$, при чему је $x_i \le \xi \le x_{i+1}$. За интерполациони полином n-тог реда, аналогни израз за грешку је

$$e_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n), \tag{1.16}$$

при чему је ξ негде између две интерполационе тачке најближе тачки у којој се одређује непозната вредност функције. Како би се искористила претходна формула за прорачун грешке, функција која се интерполира мора бити позната и диференцијабилна довољан број пута, што обично није случај. Свом срећом постоји алтернативна формула за прорачун грешке интерполације која не захтева познавање функције која се интерполира. Као апроксимација извода реда $n+1$ користе се количници коначних разлика реда $n+1$, а израз за грешку постаје

$$e_n = f[x, x_n, x_{n-1}, ..., x_1, x_0](x - x_0)(x - x_1)\cdots(x - x_n), \tag{1.17}$$

при чему је $f[x, x_n, x_{n-1}, ..., x_1, x_0]$ количник коначних разлика реда $n+1$. Због тога што једначина (1.17) садржи непознату вредност функције $f(x)$, грешка интерполације не може се добити директном применом формуле (1.17). Са друге стране, ако је позната вредност функције $f(x_{n+1})$ у додатној тачки x_{n+1} (мимо тачака интерполације), једначина (1.17) може се искористити за процену грешке као

$$e_n \approx f[x_{n+1}, x_n, x_{n-1}, ..., x_1, x_0](x - x_0)(x - x_1)\cdots(x - x_n). \tag{1.18}$$

Пример 1.4: Користећи се једначином (1.18) проценити грешку интерполационог полинома другог реда из примера 1.2. У односу на тај пример, на располагању је још једна интерполациона тачка $x_3 = 5$, $f(x_3) = 1{,}60944$.

Слично као у примеру 1.3, прорачунавамо количник коначних разлика трећег реда $f[x_3, x_2, x_1, x_0] = 0{,}00786553$, па је на основу (1.18) процена грешке $e_2 \approx 0{,}0629242$. На основу решења примера 1.2 добијамо да је грешка једнака $e_2 = 0{,}565844 - \ln 2 = -0{,}127303$, што је истог реда величине као и процена грешке.

На основу примера 1.4 као и једначина (1.18) и (1.15) може се лако закључити да апроксимација грешке за интерполациони полином n-тог реда може да се израчуна као разлика интерполације реда $n+1$ и реда n, односно као

$$e_n \approx f_{n+1}(x) - f_n(x). \tag{1.19}$$

Другим речима додатни члан који се додаје интерполацији n-тог реда како би се добила интерполација реда $n+1$, што је у ствари израз (1.18), може се тумачити као процена грешке. Ово се јасно види ако се једначина (1.19) напише у облику $f_{n+1}(x) \approx f_n(x) + e_n$.

Валидност овог приступа заснива се на претпоставци да посматрани развој (брзо) конвергира. У том случају предикција (интерполација) реда $n+1$ требало би да буде значајно ближе тачној вредности од предикције (интерполације) реда n. Самим тим једначина (1.19) одговара стандардној процени за грешку као разлика између тачног и апроксимативног решења, при чему за тачно решење сматрамо апроксимативно решење првог следећег реда. Међутим, за разлику од већине итеративних процедура, код којих се грешка процењује на основу текућег и претходног решења, у једначини (1.19) грешка се процењује на основу будућег и текућег решења. Додатно, за развоје који споро конвергирају, процена грешке (1.19) може бити значајно мања од стварне грешке, што може бити лоше својство ако се процена грешке користи као критеријум за заустављање итеративног метода.

Финално, као што ће бити показано у наставку, интерполациони полиноми високог реда врло су осетљиви на мале грешке улазних података, односно врло су лоше условљени. Самим тим, када се користе за интерполацију, обично дају резултате који значајно одступају од тачних вредности. „Гледајући унапред“, формула (1.19) је осетљивија на такве случајеве дивергенције. Самим тим формула (1.19) корисна је и као критеријум за утврђивање који подаци се могу успешно интерполирати интерполационим полиномима одређеног реда.

1.1.5. Рачунарски алгоритми за Њутнове интерполационе полиноме

Три својства чине Њутнове интерполационе полиноме изузетно атрактивним за рачунарску имплементацију:

1. Као што је приказано у формули (1.7), интерполација вишег реда може се добити додавањем једног члана претходној интерполацији нижег реда. Ово значајно олакшава прорачун неколико интерполација различитих редова у оквиру једне програмске рутине, а што је посебно погодно када се унапред не зна ред (односно облик) функције која се интерполира. Додавањем члана по члан и сукцесивним повећавањем реда интерполације, можемо да одредимо када је задовољен критеријум конвергенције, односно када додавање следећег члана вишег реда не побољшава значајно тачност решења, или када у одређеним случајевима решења почињу да дивергирају. Једначине за процену грешке, о којима ће бити више речи у тачки 3, погодне су за објективну процену критеријума конвергенције.

2. Количници коначних разлика који представљају коефицијенте полинома интерполације, приказани у једначинама (1.8)-(1.11), могу се прорачунати ефикасно. Као што је приказано у формули (1.14) и на слици 1.5, коначне разлике вишег реда прорачунавају се на основу коначних разлика нижег реда. Коришћењем информација из претходне итерације, коефицијенти наредне интерполације могу се ефикасно прорачунати. Алгоритам за спровођење овог поступка приказан је на слици 1.7.

3. Предикција грешке, приказана једначином (1.18), односно (1.19) може се врло лако инкорпорирати у постојеће алгоритме у којима се коефицијенти интерполације рачунају секвенцијално.

```
SUBROUTINE NewtInt (x, y, n, xi, yint, ea)
  LOCAL fdd_{n,n}
  DOFOR i = 0, n
    fdd_{i,0} = y_i
  END DO
  DOFOR j = 1, n
    DOFOR i = 0, n − j
      fdd_{i,j} = (fdd_{i+1,j−1} − fdd_{i,j−1})/(x_{i+j} − x_i)
    END DO
  END DO
  xterm = 1
  yint_0 = fdd_{0,0}
  DOFOR order = 1, n
    xterm = xterm * (xi − x_{order−1})
    yint2 = yint_{order−1} + fdd_{0,order} * xterm
    ea_{order−1} = yint2 − yint_{order−1}
    yint_{order} = yint2
  END order
END NewtInt
```

Слика 1.7.

Све ове карактеристике могу се искористити и инкорпорирати у општи алгоритам за имплементацију прорачуна Њутнових интерполационих полинома, као што је приказано на слици 1.7. Приметити да се алгоритам састоји из два дела: У првом делу одређују се коефицијенти развоја у формули (1.7), а у другом се врши процена вредности функције и процена грешке интерполације.

Анализа грешке интерполационих полинома открива следећу чињеницу. Како бисмо смањили грешку интерполације, интерполационе тачке требало би да буду центриране око и што је ближе могуће тачкама у којима желимо да апроксимирамо вредност функције. Ово је и у складу са формулом (1.17) за процену грешке. Ако претпоставимо да количници коначних разлика не варирају нагло са променом позиције координате x, процена грешке је пропорционална са производом $(x-x_0)(x-x_1)\cdots(x-x_{n-1})(x-x_n)$. Очигледно, што су интерполационе тачке $x_0, x_1, ..., x_{n-1}, x_n$ ближе тачки x (у којој се процењује функција), мањи ће бити и производ из претходне реченице, а самим тим и процена грешке ће бити мања.

1.2 Лагранжови интерполациони полиноми

Лагранжови интерполациони полиноми су још једна од формулација интерполационих полинома, а у односу на Њутнове интерполационе полиноме за њихов прорачун није потребно рачунати количнике коначних разлика. Интерполација Лагранжовим полиномима може се концизно записати као

$$f_n(x)=\sum_{i=0}^{n} L_i^n(x) f(x_i), \tag{1.20}$$

при чему су Лагранжови полиноми реда n дефинисани као

$$L_i^n(x)=\prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}, \tag{1.21}$$

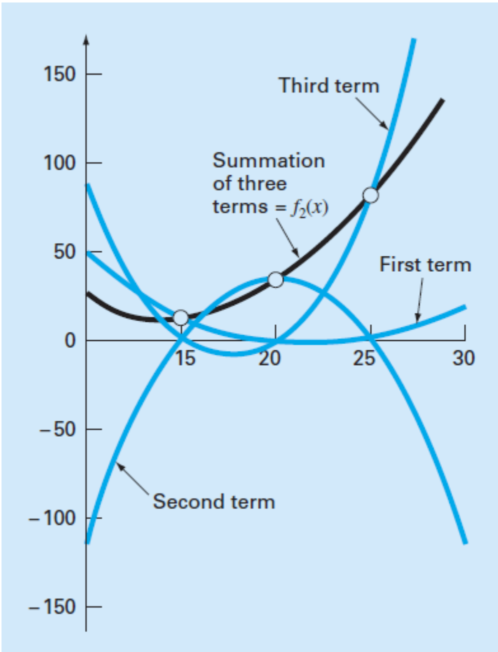
при чему симбол Π означава производе чланова под тим симболом у границама од $j=0$ до $j=n$, изузев члана за који је $j=i$. Примера ради, у случају линеарне интерполације ($n=1$) формула (1.20) постаје

$$f_1(x)=\frac{x-x_1}{x_0-x_1} f(x_0)+\frac{x-x_0}{x_1-x_0} f(x_1), \tag{1.22}$$

а у случају интерполације другог реда ($n=2$) добијамо

$$f_2(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0)+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1)+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2). \tag{1.23}$$

Једначина (1.20) може се директно извести полазећи од Њутнових интеграционих полинома. Са друге стране, једначина (1.20) може се директно извести полазећи од чињенице да је Лагранжов полином $L_i^n(x)$ једнак 1 за $x=x_i$, односно да је једнак нули у свим осталим интеграционим тачкама $x=x_j$, $0\leq j\leq n$, $j\neq i$, као што је приказано на слици 1.10 за $n=2$. Према томе производ $L_i^n(x)f(x_i)$ узима вредности $f(x_i)$ у тачкама интерполације x_i и нулту вредност у свим осталим интерполационим тачкама. Последично, сумирање свих производа из израза (1.20) представља јединствени полином реда n који пролази кроз $n+1$ интерполациону тачку.



Слика 1.10.

Пример 1.6: Користећи се Лагранжовим интерполационим полиномом првог и другог реда проценити вредност $\ln 2$. Сматрати да је вредност функције $\ln x$ позната у истим тачкама као и у претходним примерима: $x_0=1$, $x_1=4$ и $x_2=6$ ($f(x_0)=0$, $f(x_1)=1{,}38629$ и $f(x_2)=1{,}79176$).

Користећи се формулом (1.20), односно (1.22) процена вредности $\ln 2$ помоћу Лагранжовог интерполационог полинома првог реда је $f_1(x)\big|_{x=2}=\frac{x-x_1}{x_0-x_1} f(x_0)+\frac{x-x_0}{x_1-x_0} f(x_1)\Big|_{x=2}=0{,}462098$. Слично овоме, користећи се формулом (1.20), односно (1.23) добијамо

$$f_2(x)\Big|_{x=2} = \frac{x-x_1}{x_0-x_1}\frac{x-x_2}{x_0-x_2}f(x_0) + \frac{x-x_0}{x_1-x_0}\frac{x-x_2}{x_1-x_2}f(x_1) + \frac{x-x_0}{x_2-x_0}\frac{x-x_1}{x_2-x_1}f(x_2)\Bigg|_{x=2} = 0{,}565844$$

, што су, као што је и очекивано, исти резултати који су добијени и коришћењем Њутнових интерполационих полинома.

Подсетимо се да је грешка интерполационог полинома реда n дата изразом (1.17), без обзира на то да ли је се до тог интерполационог полинома дошло користећи се количником коначних разлика (Њутнови интерполациони полиноми) или користећи се Лагранжовим полиномима (Лагранжови интерполациони полиноми). Грешка из (1.17) може се скраћено записати као $e_n = f_n[x, x_n, x_{n-1}, ..., x_1, x_0]\prod_{i=0}^{n}(x-x_i)$. Даље, ако је на располагању вредност функције у додатној тачки $x = x_{n+1}$, грешка се може проценити на основу формуле (1.18). Међутим, пошто се количници коначних разлика не користе у оквиру алгоритма конструкције Лагранжових полинома, грешка се ретко процењује на овај начин, а доста чешће помоћу формуле (1.19).

Једначине (1.20) и (1.21) могу се релативно једноставно имплементирати у оквиру рачунарског програма, а на слици 1.11 приказан је псеудопрограм који се може искористити за ове сврхе.

```
FUNCTION Lagrng(x, y, n, xx)
  sum = 0
  DOFOR i = 0, n
    product = y_i
    DOFOR j = 0, n
      IF i ≠ j THEN
        product = product*(xx − x_j)/(x_i − x_j)
      ENDIF
    END DO
    sum = sum + product
  END DO
  Lagrng = sum
END Lagrng
```

Слика 1.11.

На крају, упоредимо Њутнове и Лагранжове интерполационе полиноме са становишта практичне примене. Укратко, у случајевима када ред интерполационих полинома није познат унапред, Њутнов метод има предност због тога што пружа увид у понашање и квалитет интерполационих полинома различитих редова. Додатно, процена грешке из формуле (1.18) може се једноставно интегрисати у алгоритам прорачун Њутнових интерполационих полинома пошто та процена грешке користи количник коначних разлика. Према томе, за одређивање потребног реда интерполације одређене функције обично је погоднији Њутнов метод.

Даље, ако је унапред познат жељени ред интерполационог полинома, Лагранжова и Њутнова формула имају приближно сличну сложеност нумеричког прорачуна. Са друге стране, Лагранжова верзија нешто је једноставнија за имплементацију (програмирање) због тога што не захтева прорачун и складиштење количника коначних разлика. Због тога се Лагранжови интерполациони полиноми чешће користе када је унапред познат жељени ред интерполационог полинома.

Интерполациони полиноми вишег реда могу бити лоше условљени, односно могу бити врло осетљиви на грешке заокруживања[1] (коначне тачности представљања бројева и коначне тачности прорачуна функција помоћу рачунара). Представљање бројева у двострукој тачности понекад ублажава овај проблем. Са друге стране, како степен интерполационих полинома расте, у једном тренутку доћи ће се до тренутка када ће грешка заокруживања онемогућити интерполацију на једноставан начин како је то приказано до сад.

1.3 Коефицијенти интерполационих полинома

Иако су и Њутнови и Лагранжови интерполациони полиноми подједнако примерени за одређивање вредности функције између тачака интерполације, они не резултују стандардним обликом записа полинома који је

[1] Слично важи и за регресију вишег реда.

$$f(x)=a_0+a_1x+a_2x^2+...+a_{n-1}x^{n-1}+a_nx^n\,. \tag{1.24}$$

Коефицијенти из претходне формуле могу се, барем теоријски, једноставно израчунати на основу чињенице да је потребно $n+1$ тачака у којима су познате вредности функције да би се одредило $n+1$ коефицијената полинома реда n. Према томе, може се конструисати систем линеарних једначина како би се одредили коефицијенти $a_0, a_1, ..., a_n$. На пример, претпоставимо да желимо да одредимо коефицијенте интерполационог полинома другог реда (интерполацију параболом)

$$f_2(x)=a_0+a_1x+a_2x^2\,. \tag{1.25}$$

За то нам је потребна позната вредност функције коју желимо да интерполирамо, $f(x)$, у три тачке: $(x_0,\, f(x_0))$, $(x_1,\, f(x_1))$ и $(x_2,\, f(x_2))$. Ако сваку од ове три вредности функције изједначимо са полиномом из (1.25) у одговарајућој тачки интерполације добијамо следећи систем једначина

$$\begin{aligned} f(x_0) &= a_0+x_0a_1+x_0^2a_2 \\ f(x_1) &= a_0+x_1a_1+x_1^2a_2 \\ f(x_2) &= a_0+x_2a_1+x_2^2a_2 \end{aligned}\,. \tag{1.26}$$

У претходном систему једначина x_0, x_1 и x_2 су познате вредности, а a_0, a_1 и a_2 непознате вредности по којима се систем решава. Пошто је број непознатих величина једнак броју једначина, једначине (1.26) могу се решити на неки од стандардних начина, укључујући и метод елиминације непознатих.

Изузетно је важно напоменути да овај алгоритам не представља ефикасно решење за одређивање коефицијената интерполационих полинома. Једначине у систему једначина попут оног у (1.26) су изузетно лоше условљене. Без обзира да ли се овај систем решава методом елиминације или неким другим ефикаснијим методом, прорачунати коефицијенти могу бити врло нетачни, посебно за велике редове n. Када се ти коефицијенти искористе за накнадну интерполацију, обично резултују значајним грешкама.

Укратко, ако је циљ да се одреде вредности функције између тачака интерполације препоручљиво је користити Њутнове или Лагранжове интерполационе полиноме. Уколико је из неког разлога неопходно конструисати интерполациони полином у облику (1.24), то се може урадити само за ниже редове полинома, уз посебну опрезност.

1.4 Инверзна интерполација

Према стандардној номенклатури, $f(x)$ и x представљају зависну и независну променљиву, респективно. Као последица тога, вредности променљиве x обично су униформно распоређене. Једноставан пример овакве ситуације је табела са вредностима функције $f(x)=1/x$ у еквидистантним тачкама,

x	1	2	3	4	5	6	7
$f(x)$	1	0.5	0.3333	0.25	0.2	0.1667	0.1429

Претпоставимо сада да имамо другачији проблем у односу на досадашње, односно да нам је позната вредност функције $f(x)$ за коју је потребно одредити одговарајућу вредности x. Примера ради, за вредности из горње табеле, претпоставимо да је задатак да се одреди вредност x за коју ће бити $f(x)=0{,}3$. Пошто је нама функција на основу које су добијене табелиране вредности позната, ми унапред знамо да је тачно решење $x=1/f(x)=1/0{,}3=3{,}3333$. Међутим, решимо овај проблем под претпоставком да нам облик функције на основу које су добијене табелиране вредности није познат, што често одговара ситуацији у пракси.

Проблеми оваквог типа називају се инверзни проблеми. За мало компликованије проблеме, односно када нам функција за коју су добијени табелирани подаци није позната, једна од могућих идеја је да се замене улоге које имају $f(x)$ и x, односно да се нацрта график x у функцији $f(x)$, а да се онда помоћу Лагранжових или Њутнови интерполациони полиноми одреди тражена вредност x. Нажалост, када се улога променљивих (x и $f(x)$) замени, не постоји гаранција да ће вредности на новој апсциси (односно вредности $f(x)$) бити униформно распоређене, што је обично добар предуслов за тачну и ефикасну интерполацију. У ствари, у већини случајева, вредности ће бити згуснуте на појединим местима. На неким деловима постојаће велика густина познатих вредности, а на неким другим велико растојање између две интерполационе тачке. Примера ради, за функцију $f(x)=1/x$ након замене улога промењљивих добијамо следећу табелу

$f(x)$	0.1429	0.1667	0.2	0.25	0.3333	0.5	1
x	7	6	5	4	3	2	1

Таква неуниформна расподела на апсциси обично води ка великим осцилацијама у резултујућим интерполационим полиномима. До овога може доћи чак и у случају интерполационих полинома ниског реда.

Алтернативно решење је да се за оригиналне податке ($f(x)$ у зависности од x) одреди интерполациони полином $f_n(x)$. У већини случајева, због тога што су интерполационе тачке униформно распоређене дуж x осе, овај интерполациони полином неће бити лоше условљен. Решење проблема онда се своди на проналажење вредности x за коју је интерполациони полином једнак датом $f(x)$. Самим тим проблем инверзне интерполације своди се на проналажење нуле функције, а то се може решити разним стандардним алгоритмима, од којих је метод сечице само један у низу.

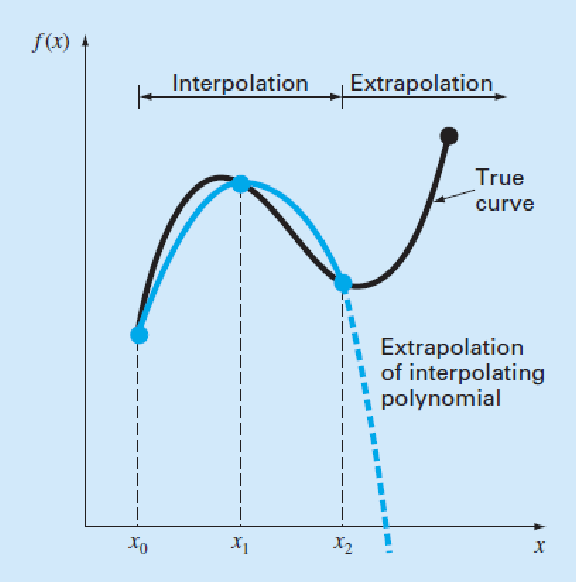
1.5 Додатни коментари

Пре него наставимо даље, згодно је поменути још две теме: интерполација са еквидистантним тачкама и екстраполација.

Због тога што ни Њутнови ни Лагранжови интерполациони полиноми не захтевају еквидистантно распоређене интерполационе тачке, неко се може запитати зашто би се тој теми посветила посебна пажња. Пре омасовљења дигиталних рачунара, поменуте технике интерполације имале су широку примену над униформно табелираним подацима. У ствари, концепт прорачуна количника коначних разлика и алгоритам приказан у табели 1.5 осмишљен је како би се олакшао прорачун и имплементација интерполационих полинома пре појаве рачунара.

Упркос томе што приликом прорачуна интерполационих полинома помоћу рачунара нема потребе за еквидистантним тачкама, тај случај важан је за остале алгоритме који се помињу у оквиру метода коначних елемената. Један од таквих примера је и нумеричка интеграција, односно процена грешке нумеричке интеграције. Када су интерполационе тачке еквидистантно распоређене, Њутнова формула за интерполацију позната је и под називом Њунт-Грегорова интерполациона формула унапред (*Newton-Gregory forward interpolation formula*).

Друга тема је екстраполација. Екстраполација подразумева процес процене вредности функције $f(x)$ изван опсега тачака интерполације $x_0, x_1, ..., x_n$, као што је приказано на слици 1.13. У претходном одељку напоменули смо да се најтачнија интерполација обично добија у случајевима када је непозната вредност у близини интерполационих тачака. Очигледно, ово није задовољено када се непозната вредност одређује (далеко) изван оквира интерполационих тачака, а самим тим грешка екстраполације може бити велика. Као што је приказано на слици 1.13, процес екстраполације представља корак „у непознато“, пошто екстраполација проширује криву изван познатог региона. Због свега тога права крива лако може да се значајно разиђе са предикционом кривом. Због свега тога када год настане неопходност за екстраполацијом, читаву процедуру треба спровести са изузетном опрезношћу.



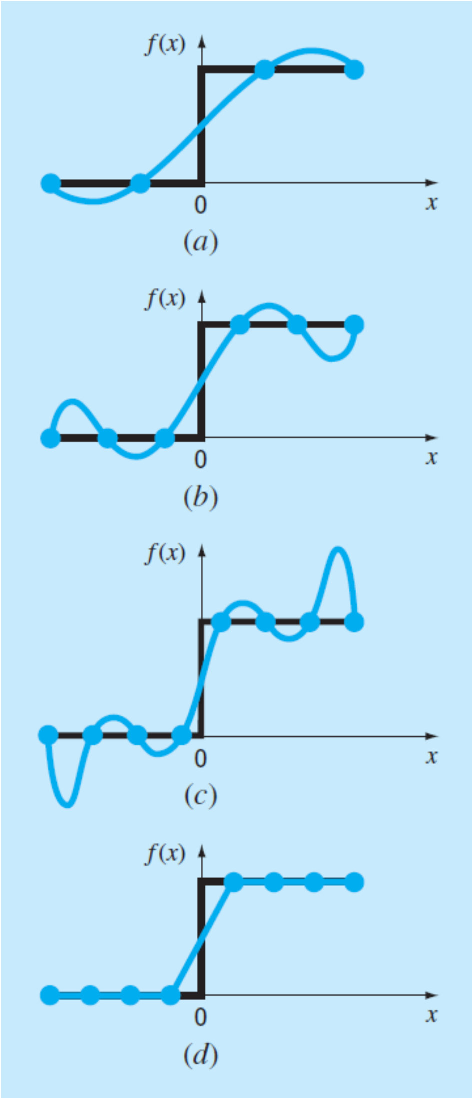
Слика 1.13.

1.6 Интерполација сплајновима

До сада смо користили полиноме n -тог реда за интерполацију између $n+1$ тачке. На пример, за осам интерполационих тачака, приказали смо поступак за одређивање полинома седмог реда који ће пролазити кроз те тачке. Ова крива (интерполациони полином) осликаће сва закривљења (барем закључно са седмим редом) диктирана тачкама интерполације. Међутим, постоје случајеви када овакви интерполациони полиноми могу водити ка великим грешкама услед грешке заокруживања и ефекта премашења. Алтернативан приступ је примена полинома нижег реда на подскуп интерполационих тачака, а такви међусобно повезани полиноми називају се сплајнови (*spline functions*).

На пример, криве трећег реда којима се повезују парови тачака називају се кубни сплајнови (*cubic splines*). Овакви полиноми могу се конструисати тако да спојеви између два суседна сплајна буду визуелно глатки (спојеви). На први поглед може се учинити да апроксимација сплајновима трећег реда мора бити инфериорна у односу на оригиналну интерполацију полиномима седмог реда (Њутновим односно Лагранжовим интерполационим полиномима). Читалац се може запитати зашто би ико користио сплајнове за интерполацију.

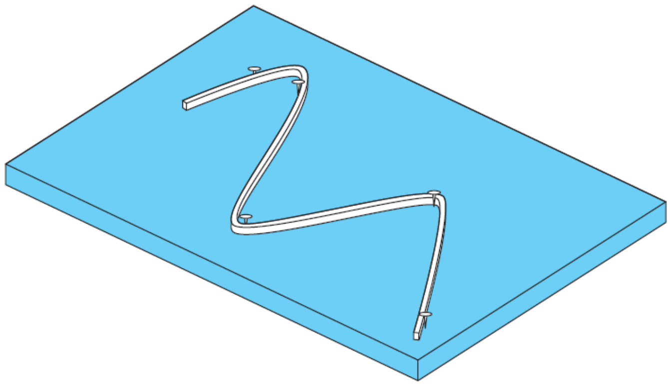
На слици 1.14 приказана је једна од ситуација у којој сплајнови резултују бољом апроксимацијом у односу на интерполационе полиноме вишег реда. У питању је ситуација где је у највећем делу функција која се интерполира глатка са благом променом, а у одређеном броју делова има нагле промене. Хевисајдова функција приказана на слици 1.14 представља екстреман пример таквих функција и овде је искоришћена у илустративне сврхе.



Слика 1.14.

Слике од 1.14(a) до 1.14(c) показују да интерполациони полиноми вишег реда имају тенденцију да неконтролисано осцилују пролазећи кроз интерполационе тачке у близини брзе промене функције. Са друге стране, сплајн такође повезује интерполационе тачке (пролази кроз њих), али због тога што је ограничен на полиноме нижег реда, осцилације су сведене на минималне, као што је приказано на слици 1.14(d).

Концепт сплајнова потекао је из технике која се користила у производњи, када се користила танка флексибилна (метална) трака (названа сплајн) за цртање глатких крива које пролазе кроз задати скуп тачака. Овај процес је илустрован на слици 1.15 за скуп од пет пинова (ексера), који овде представљају тачке интерполације. Према овој техници, моделар постави папир преко равне дрвене подлоге, а затим закуца ексере (који пролазе кроз папир и завршавају у дрвеној подлози) у позицијама тачака интерполације. Након савијања сплајна између закуцаних ексера добија се глатка крива која се оловком преноси на папир. Без обзира на ред апроксимације, овакав начин интерполације одомаћен је под називом „кубни сплајнови“ (cubic spline).



Слика 1.15.

У оквиру овог одељка искористиће се једноставна линеарна функција како би се увели неки основни концепти и проблеми приликом интерполације сплајновима. Након тога развиће се алгоритам за интерполацију задатих података квадратним сплајновима. На крају ће бити приказан материјал о кубним сплајновима, који су једни од најкориснијих и најчешће коришћених типова сплајнова у инжењерској пракси.

1.6.1 Линеарни сплајнови

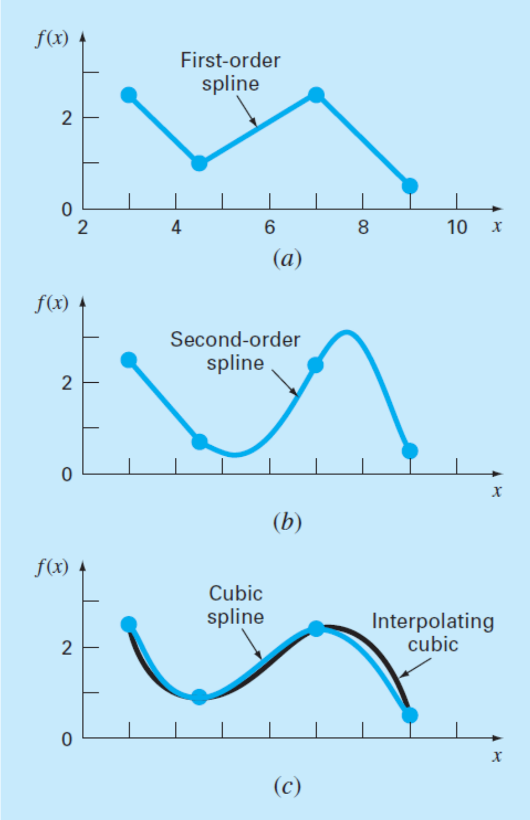
Најједноставнији начин да се повежу две тачке је помоћу правих линија. Сплајнови првог реда који повезују интерполационе тачке могу се дефинисати као скуп линеарних функција облика:

$$\begin{aligned} f(x) &= f(x_0) + m_0(x - x_0), \quad x_0 \le x \le x_1 \\ f(x) &= f(x_1) + m_1(x - x_1), \quad x_1 \le x \le x_2 \\ &\vdots \\ f(x) &= f(x_{n-1}) + m_{n-1}(x - x_{n-1}), \quad x_{n-1} \le x \le x_n \end{aligned} \tag{1.27а}$$

при чему су m_i коефицијенти нагиба правих линија које повезују интерполационе тачке и који се рачунају као

$$m_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}. \tag{1.27б}$$

На основу претходних једначина може се одредити вредност функције у било којој тачки између x_0 и x_n тако што се прво одреди интервал коме припада та тачка, а онда се искористи одговарајућа линеарна једначина. Приказани метод очигледно је идентичан са методом линеарне интерполације на део по део интерполационих тачака. Један од примера употребе сплајнова првог реда приказан је на слици 1.16(а). На основу ове слике лако се може уочити да линеарни сплајнови резултују кривом која није глатка, што представља и један од њихових главних недостатака. Укратко, у тачки споја два сплајна (која се назива и чвор, *knot*), нагиб криве мења се брзо, а формално математички први извод интерполације има прекид у овим тачкама. Овај недостатак отклања се коришћењем сплајнова вишег реда који гарантују глаткоћу у чворовима тако што изједначавају изводе два сплајна у тачки споја. Овај поступак биће приказан у наставку.



Слика 1.16.

1.6.2 Квадратни сплајнови

Како би се обезбедило да је m извода интерполационе функције континуално у чвору, мора се користити сплајн реда барем $m+1$. Полиноми трећег реда, односно кубни сплајнови обезбеђују континуалност првог и другог извода у чвору, а они уједно представљају и сплајнове најчешће коришћене у пракси. Иако трећи и виши изводи могу имати прекиде када се користе кубни сплајнови, ови прекиди обично се не могу детектовати визуелно, њихова континуалност није од значаја у већини инжењерских примена, па се због тога сплајнови реда вишег од трећег ретко користе.

Због тога што извођење кубних сплајнова може бити релативно сложено, то извођење остављено је за наредни одељак, а у оквиру овог одељка прво ћемо илустровати концепт интерполације сплајновима другог реда. Ови „квадратни сплајнови" имају континуалан први извод у чворовима. Иако квадратни сплајнови не гарантују континуалност другог извода у чворовима, њиховом анализом може се јасно демонстрирати процедура за развој сплајнова вишег реда.

Циљ квадратних сплајнова је конструкција полинома другог реда за сваки интервал између интерполационих тачака. Квадратни сплајнови за сваки интервал могу се представити у облику

$$f_i(x) = a_i x^2 + b_i x + c_i \, . \tag{1.28}$$

Слика 1.17 уведена је како би се додатно разјаснила нотација која ће бити коришћена. За $n+1$ тачку интерполације ($i = 0, 1, 2, ..., n$), постоји n интервала и, последично, $3n$ непознатих коефицијената (a, b и c у формулама облика (1.28)) које треба прорачунати. Према томе, потребно је поставити $3n$ једначина односно услова како би се одредили непознати коефицијенти a, b и c. Ови услови су:

1. Вредности суседних сплајнова (који имају заједнички чвор) морају бити међусобно једнаке у заједничком чвору, односно једнаке вредности функције која се интерполира. Ово се математички може записати као

$$a_{i-1} x_{i-1}^2 + b_{i-1} x_{i-1} + c_{i-1} = f(x_{i-1}), \tag{1.29}$$

$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = f(x_{i-1}), \tag{1.30}$$

за $2 \le i \le n$. Због тога што се у једначинама (1.29) и (1.30) користе сви чворови изузев почетног и последњег, свака од ових једначина доприноси са $n-1$ условом, а заједно резултују са $2n-2$ услова.

2. Први и последњи сплајн морају проћи кроз интерполациону тачку. Овај услов резултује са две додатне једначине

$$a_1x_0^2 + b_1x_0 + c_1 = f(x_0) \text{ и} \tag{1.31}$$

$$a_nx_n^2 + b_nx_n + c_n = f(x_n), \tag{1.32}$$

што заједно са претходним условима укупно води ка $2n-2+2=2n$ услова.

3. Први изводи у унутрашњим чворовима морају бити једнаки. Први извод једначине (1.28) је

$$f'(x) = 2ax + b,$$

према томе овај услов се математички може записати као

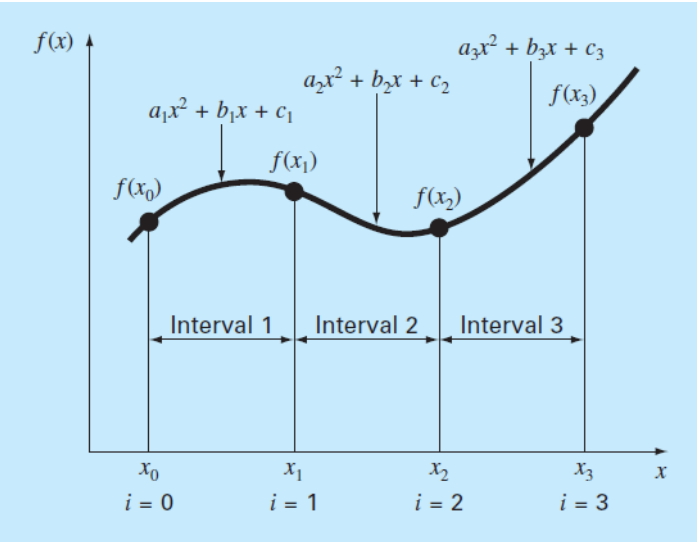
$$2a_{i-1}x_{i-1} + b_{i-1} = 2a_ix_{i-1} + b_i \tag{1.33}$$

за $2 \le i \le n$. Ово доприноси са додатних $n-1$ једначина, што укупно са претходним условима води ка $2n+n-1=3n-1$ једначини. Због тога што имамо $3n$ непознатих, у овом тренутку недостаје нам још један услов. Осим ако немамо неку додатну информацију у вези са функцијом коју желимо да интерполирамо или у вези са њеним изводом, потребан нам је још један независан услов како бисмо могли да спроведемо процедуру одређивања коефицијената a, b и c. Иако постоји више различитих опција које је могуће изабрати, додатни услов изабраћемо на следећи начин.

4. Претпоставићемо да је други извод једнак нули у почетној интерполационој тачки. Због тога што је други извод једначине (1.28) једнак $2a_i$, овај услов може се математички записати као

$$a_1 = 0. \tag{1.34}$$

Овај услов визуелно одговара ситуацији где су прве две интерполационе тачке повезане правом линијом.



Слика 1.17.

18.6.3 Кубни сплајнови

Циљ кубних сплајнова је да се конструише полином трећег реда за сваки интервал између интерполационих чворова, односно да се одреде коефицијенти полинома облика

$$f_i(x) = a_ix^3 + b_ix^2 + c_ix + d_i. \tag{1.35}$$

Према томе, за $n+1$ интерполациону тачку ($i = 0, 1, ..., n$), постоји n интерполационих интервала и, последично, $4n$ коефицијента које је потребно одредити. Слично као и код квадратних сплајнова, сада је потребно $4n$ услова како би се могли одредити коефицијенти a, b, c и d. Ови услови су:

1. Вредности сплајнова морају бити једнаке и једнаке вредности функције у свим интерполационим тачкама изузев почетне и крајње ($2n-2$ услова).

2. Први и последњи сплајн морају да пролазе кроз прву, односно последњу интерполациону тачку (2 услова).

3. Први изводи у унутрашњим интерполационим тачкама морају бити једнаки ($n-1$ услов).

4. Други извод у унутрашњим интерполационим тачкама морају бити једнаки ($n-1$ услов).

5. Други изводи морају бити једнаки нули у крајњим интерполационим тачкама (почетној и последњој) (2 услова).

Визуелна интерпретација услова 5 је да функција пролази кроз крајње чворове као права линија. Овакав услов води ка сплајновима који се називају „природни“. Назив су добили по томе што се трака за моделовање (илустрована на слици 1.15) понаша на овакав начин. Ако вредност другог извода у крајњим чворовима није једнака нули, односно ако у крајњим чворовима сплајнови имају закривљење, ова информација може се искористити као алтернатива услову 5 како би се добила друга два услова.

Претходно приказаних пет услова резултују са $4n$ једначина које је потребно решити како би се одредило $4n$ непознатих коефицијената (a , b , c и d). Иако је могуће конструисати кубне сплајнове на овај начин, у наставку ће бити приказан алтернативан начин који захтева решавање само $n-1$ једначине. Иако је извођење овог алгоритма (приказаног у додатку 1.3) мање очигледно него у случају квадратних сплајнова, добитак у ефикасности оваквог начина вредан је додатног труда.

Додатак 1.3: Први корак у овом извођењу (Cheney and Kincaid, 2008) заснован је на запажању да због тога што је сваки пар интерполационих тачака повезан полиномима трећег реда, други извод дуж сваког интерполационог интервала је права линија. Једначина (1.35) може се диференцирати два пута како би се потврдило ово запажање. На основу овога, други извод сплајна може се представити помоћу Лагранжовог интерполационог полинома првог реда, као у једначини (1.22):

$$f_i''(x)=\frac{x-x_i}{x_{i-1}-x_i}f''(x_{i-1})+\frac{x-x_{i-1}}{x_i-x_{i-1}}f''(x_i),\ x_{i-1}\le x\le x_i, \tag{д1.3.1}$$

при чему је $f_i''(x)$ вредност другог извода сплајна у било којој тачки x на i -том интервалу. Према томе, претходна једначина представља једначину праве линије која повезује вредност другог извода $f''(x_{i-1})$ у чвору x_{i-1} са вредношћу другог извода $f''(x_i)$ у чвору x_i .

Даље, једначину (д1.3.1) можемо интегралити два пута како бисмо добили израз за $f_i(x)$. Обратити пажњу да ћемо приликом те интеграције у израз за $f_i(x)$ увести две нове непознате (интеграционе) константе. Ове константе могу се одредити на основу услова да једначина $f_i(x)$ мора бити једнака $f_i(x_i)$ у тачки x_i, односно да функција $f_i(x)$ мора бити једнака $f_i(x_{i-1})$ у тачки x_{i-1}. Након спровођења те процедуре финално добијамо израз за кубни сплајн

$$\begin{aligned} f_i(x)=&\frac{f''(x_{i-1})}{6(x_i-x_{i-1})}(x_i-x)^3+\frac{f''(x_i)}{6(x_i-x_{i-1})}(x-x_{i-1})^3 \\ &+\left[\frac{f(x_{i-1})}{x_i-x_{i-1}}-\frac{f''(x_{i-1})(x_i-x_{i-1})}{6}\right](x_i-x) \qquad ,\ x_{i-1}\le x\le x_i . \\ &+\left[\frac{f(x_i)}{x_i-x_{i-1}}-\frac{f''(x_i)(x_i-x_{i-1})}{6}\right](x-x_{i-1}) \end{aligned} \tag{д1.3.2}$$

Строго говорећи овај израз јесте компликованији од оригиналног израза за кубни сплајн на интервалу i који је дат у једначини (1.35). Са друге стране, израз (д1.3.2) садржи само два непозната „коефицијента“, друге изводе на почетку и крају сегмента, $f''(x_{i-1})$ и $f''(x_i)$. Према томе ако успемо да одредимо други извод сплајнова у сваком чвору, једначина (д1.3.2) представља сплајн трећег реда.

Други извод сплајнова у интерполационим чворовима може се одредити полазећи од услова да први извод сплајнова у чворовима мора бити континуалан, односно да мора да важи

$$f_i'(x_i)=f_{i+1}'(x_i) \tag{д1.3.3}$$

Једначина (д1.3.2) може се диференцирати како би се добио израз за први извод сплајна. Ако се ово спроведе за интервал i и интервал $i+1$, а та два израза се изједначе према једначини (д1.3.3), добијамо следећу једначину

$$\begin{aligned} &(x_i-x_{i-1})f''(x_{i-1})+2(x_{i+1}-x_{i-1})f''(x_i)+(x_{i+1}-x_i)f''(x_{i+1})= \\ &=\frac{6}{x_{i+1}-x_i}\left(f(x_{i+1})-f(x_i)\right)+\frac{6}{x_i-x_{i-1}}\left(f(x_{i-1})-f(x_i)\right) \end{aligned} . \tag{д1.3.4}$$

Ако се једначина (д1.3.4) напише за све унутрашње чворове, добија се систем од $n-1$ једначине по $n+1$ непознатом другом изводу. Међутим, пошто овде разматрамо природне сплајнове, други извод сплајнова у крајњим тачкама једнак не нули, па се проблем своди на систем од $n-1$ једначине по $n-1$ непознатом другом изводу. Додатно, може се приметити да једначина (д1.3.4) води ка тридијагоналном систему једначина. Према томе, поступком описаном у овом додатку, не само да је смањен ред система једначина, већ је он учињен и тридијагоналним, за чије решавање постоје посебно ефикасни алгоритми.

На основу извођења из додатка 1.3 добијамо следећу кубну једначину за сваки од интервала:

$$\begin{aligned} f_i(x) = {} & \frac{f''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f''(x_i)}{6(x_i - x_{i-1})}(x - x_{i-1})^3 + \\ & + \left[\frac{f(x_{i-1})}{x_i - x_{i-1}} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6}\right](x_i - x) + \\ & + \left[\frac{f(x_i)}{x_i - x_{i-1}} - \frac{f''(x_i)(x_i - x_{i-1})}{6}\right](x - x_{i-1}) \end{aligned} \quad , \; x_{i-1} \le x \le x_i \, , \tag{1.36}$$

Ова једначина садржи само два непозната коефицијента – други извод на крају сваког интерполационог интервала. Ови коефицијенти могу се израчунати из следеће једначине:

$$\begin{aligned} & (x_i - x_{i-1}) f''(x_{i-1}) + 2(x_{i+1} - x_{i-1}) f''(x_i) + (x_{i+1} - x_i) f''(x_{i+1}) = \\ & = \frac{6}{x_{i+1} - x_i}\left(f(x_{i+1}) - f(x_i)\right) + \frac{6}{x_i - x_{i-1}}\left(f(x_{i-1}) - f(x_i)\right) \end{aligned} \quad . \tag{1.37}$$

Ако се ова једначина напише за све унутрашње чворове, добија се $n-1$ једначина по $n-1$ непознатом другом изводу функције. (Подсетимо се да су други изводи на крајњим чворовима постављени на нулу.)

Резултати интерполације једног скупа података сплајновима првог, другог и трећег реда сумирани су на слици 1.16. Са те слике јасно се може уочити јасно побољшање интерполације како напредујемо од линеарних, преко квадратних ка кубним сплајновима. На тој слици (слика 1.16(c)) приказан је и кубни интерполациони полином. Иако се кубни сплајн састоји од серије кривих трећег реда, резултујућа интерполација разликује се у односу на ону добијену интерполационим полиномом трећег реда. Ово је директна последица тога што природни сплајнови захтевају нулти други извод у крајњим чворовима, што није случају за кубне интерполационе полиноме.

1.6.4 Рачунарски алгоритам за кубне сплајнове

Алгоритам за прорачун кубних сплајнова приказан у претходном одељку идеалан је за имплементацију на рачунару. Подсетимо се да је, уз помоћ елегантних манипулација, метод сведен на решавање линеарног система једначина димензија $n-1$. Додатна предност тог алгоритма је, као што је приказано у формули (1.37), то што је систем једначина тридијагоналан, а постоје алгоритми да се такав систем једначина реши изузетно ефикасно. На слици 1.18 приказан је рачунарски алгоритам који инкорпорира ове карактеристике.

```
SUBROUTINE Spline (x,y,n,xu,yu,dy,d2y)
LOCAL e_n, f_n, g_n, r_n, d2x_n
CALL Tridiag(x,y,n,e,f,g,r)
CALL Decomp(e,f,g,n−1)
CALL Subst(e,f,g,r,n−1,d2x)
CALL Interpol(x,y,n,d2x,xu,yu,dy,d2y)
END Spline

SUBROUTINE Tridiag (x,y,n,e,f,g,r)
f_1 = 2 * (x_2−x_0)
g_1 = (x_2−x_1)
r_1 = 6/(x_2−x_1) * (y_2−y_1)
r_1 = r_1+6/(x_1−x_0) * (y_0−y_1)
DOFOR i = 2, n−2
  e_i = (x_i−x_i−1)
  f_i = 2 * (x_i+1 − x_i−1)
  g_i = (x_i+1 − x_i)
  r_i = 6/(x_i+1 − x_i) * (y_i+1 − y_i)
  r_i = r_i+6/(x_i − x_i−1) * (y_i−1 − y_i)
END DO
e_n−1 = (x_n−1 − x_n−2)
f_n−1 = 2 * (x_n − x_n−2)
r_n−1 = 6/(x_n − x_n−1) * (y_n − y_n−1)
r_n−1 = r_n−1 + 6/(x_n−1 − x_n−2) * (y_n−2 − y_n−1)
END Tridiag
```

```
SUBROUTINE Interpol (x,y,n,d2x,xu,yu,dy,d2y)
flag = 0
i = 1
DO
  IF xu ≥ x_i−1 AND xu ≤ x_i THEN
     c1 = d2x_i−1/6/(x_i − x_i−1)
     c2 = d2x_i/6/(x_i − x_i−1)
     c3 = y_i−1/(x_i − x_i−1) − d2x_i−1 * (x_i−x_i−1)/6
     c4 = y_i/(x_i − x_i−1) − d2x_i * (x_i−x_i−1)/6
     t1 = c1 * (x_i − xu)^3
     t2 = c2 * (xu − x_i−1)^3
     t3 = c3 * (x_i − xu)
     t4 = c4 * (xu − x_i−1)
     yu = t1 + t2 + t3 + t4
     t1 = −3 * c1 * (x_i − xu)^2
     t2 = 3 * c2 * (xu − x_i−1)^2
     t3 = −c3
     t4 = c4
     dy = t1 + t2 + t3 + t4
     t1 = 6 * c1 * (x_i − xu)
     t2 = 6 * c2 * (xu − x_i−1)
     d2y = t1 + t2
     flag = 1
  ELSE
     i = i + 1
  END IF
  IF i = n + 1 OR flag = 1 EXIT
END DO
IF flag = 0 THEN
  PRINT “outside range”
  pause
END IF
END Interpol
```

Слика 1.18.

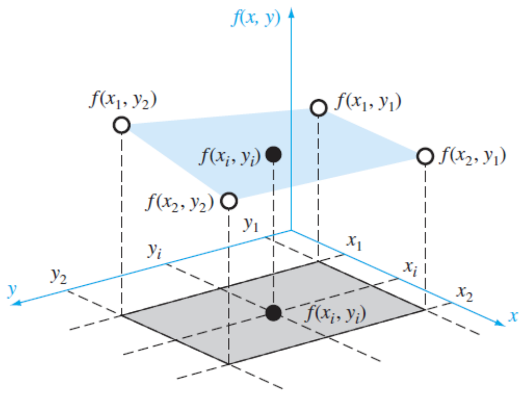
Приметимо да рутина приказана на слици 1.18 враћа једну интерполирану вредност, yu, за задату вредност зависне променљиве, xu. Ово је само један од начина на који се може спровести интерполација сплајновима. На пример, могуће је и прво одредити коефицијенте, а онда спровести произвољан број интерполација. Додатно, рутина враћа први (dy) и други ($dy2$) извод интерполационог сплајна у тачки xu. Иако није неопходно прорачунати ове изводе, они могу бити корисни у многим случајевима примене сплајнова за интерполацију.

1.7 Вишедимензиона интерполација

Поступци приказани до сад за једнодимензиону интерполацију лако се могу проширити на вишедимензионе проблеме. У оквиру овог одељка приказаћемо најједноставнији случај дводимензионе интерполације у Декартовом координатном систему.

1.7.1 Билинеарна интерполација

Дводимензиона интерполација бави се одређивањем вредностима функције између тачака интерполације за функције две променљиве које се могу записати као $z = f(x, y)$. Као што је приказано на слици 1.19, претпоставимо да имамо вредности функције у четири (интерполационе) тачке: $f(x_1, y_1)$, $f(x_2, y_1)$, $f(x_1, y_2)$ и $f(x_2, y_2)$ и да желимо да спроведемо интерполацију између ових тачака како бисмо проценили вредност функције за произвољне аргументе x и y ($x_1 \le x \le x_2$ и $y_1 \le y \le y_2$). Ако за интерполацију користимо линеарне функције, резултујућа површ повезује интерполационе тачке помоћу две линеарне функције. Таква функција назива се билинеарна.



Слика 1.19.

Једноставан начин за развој билинеарне функције приказан је на слици 1.20. У првом кораку, сматрамо да је y-вредност фиксна и применимо једнодимензиону линеарну интерполацију у x-смеру. Користећи се Лагранжовим интерполационим полиномом првог реда, резултујућа интерполација у тачки (x_i, y_1) је

$$f(x_i, y_1) = \frac{x_i - x_2}{x_1 - x_2} f(x_1, y_1) + \frac{x_i - x_1}{x_2 - x_1} f(x_2, y_1), \tag{1.38}$$

док је слична интерполација у тачки (x_i, y_2) једнака

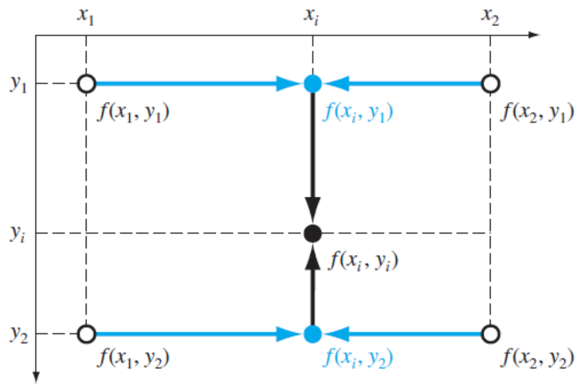
$$f(x_i, y_2) = \frac{x_i - x_2}{x_1 - x_2} f(x_1, y_2) + \frac{x_i - x_1}{x_2 - x_1} f(x_2, y_2). \tag{1.39}$$

Вредности у овим тачкама могу се онда искористити како би се извршила линеарна интерполација у y-смеру, што води ка финалном решењу

$$f(x_i, y_i) = \frac{y_i - y_2}{y_1 - y_2} f(x_i, y_1) + \frac{y_i - y_1}{y_2 - y_1} f(x_i, y_2). \tag{1.40}$$

Након замене једначина (1.38) и (1.39) у једначину (1.40) добијамо

$$\begin{aligned} f(x_i, y_i) = {} & \frac{x_i - x_2}{x_1 - x_2} \frac{y_i - y_2}{y_1 - y_2} f(x_1, y_1) + \frac{x_i - x_1}{x_2 - x_1} \frac{y_i - y_2}{y_1 - y_2} f(x_2, y_1) + \\ & + \frac{x_i - x_2}{x_1 - x_2} \frac{y_i - y_1}{y_2 - y_1} f(x_1, y_2) + \frac{x_i - x_1}{x_2 - x_1} \frac{y_i - y_1}{y_2 - y_1} f(x_2, y_2) \end{aligned}. \tag{1.41}$$



Слика 1.20.