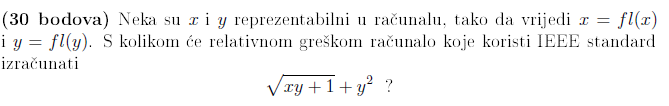
# IEEE standard – greške:

Napomena:

Pri rješavanju se koriste slijedeći pomoćni rezultati:



Rješenje:

Napomene:

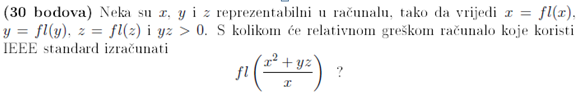
- greška množenja, greška zbrajanja, greška korijenovanja, greška množenja, greška zbrajanja.

- 1 se može napisati kao 1(1+0). Znači greška je 0, pa se može primjeniti 'max' pravilo.

- Ako su ispod korjena faktori onda je najjednostavnije faktore greške izbaciti u novi korijen.

- Epsiloni se mogu ali ne moraju indeksirati pošto su svi manji od *u*.

- xy u ovom zadatku je pozitivan broj (nije bilo navedeno zadatku greškom).



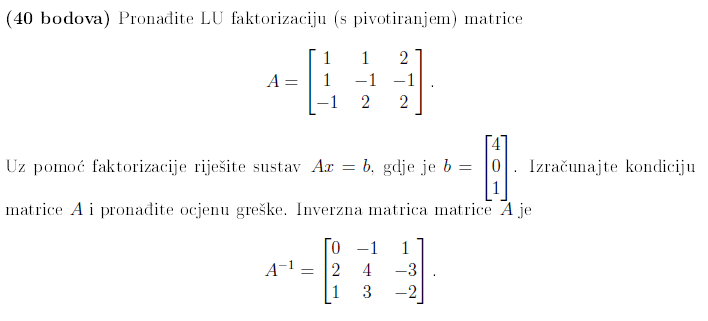
Rješenje:

**Zadatak (sa vježbi):**

Naći grešku za

Rješenje:

# LU faktorizacija



Rješenje:

Sustav se rješava na normalan način. Umjesto da se pišu nule u toku rješavanja sustava, zgodnije je pisati dobivene faktore u istoj matrici.

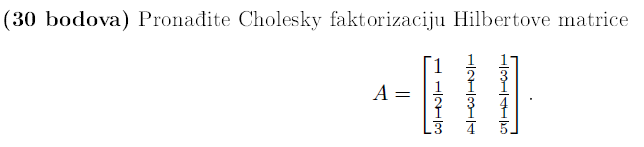
Napomena:

* Dobiveni faktori mijenjaju predznak kada se pišu u matrici L.
* Pivotiranje se radi po apsolutnim vrijednostima. Permutacijska (jedinična) matrica P služi samo za praćenje zamjene redaka te kada se pomnoži sa vektorom b, permutira vektor b tako da bude u skladu za izmjenjenim retcima. Ako nije bilo pivotiranja onda se koristi zadani b.
* U slučaju da se ne zada inverzna matrica dobro je obnoviti računanje inverza.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | } |  |  |  | } |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | } |  |  |  | } |  |
|  |  |  |  |
|  |  |  |  |

# Cholesky faktorizacija

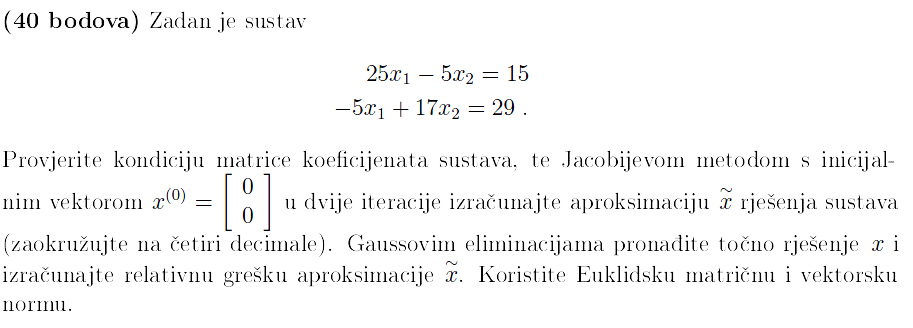


Rješenje:

Napomena:

Vrijedi i (slično kao i kod LU faktorizacije)

# Jacobijeva metoda



Prva iteracija:

Druga iteracija:

Gaussovom eliminacijom zadanog sustava dobije se točno rješenje: .

Relativna greška:

Napomena:

Ako nije zadan broj iteracija može se tražiti preciznost na određeni broj značajnih znamenki. U tom slučaju se radi suma razlika (po apsolutnoj vrijednosti) svih članova vektora iz trenutne i prethodne iteracije. Ako je suma manja od zadane preciznosti onda se iteracije zaustavljaju. U gornjem zadatku bi za drugu iteraciju to izgledalo ovako:

, pa se gleda da li je manji od zadane vrijednosti.

# Gauss-Seidel

Zadan je slijedeći sustav:

Riješiti sustav sa 2 iteracije Gauss-Seidela uz početni vektor

Rješenje:

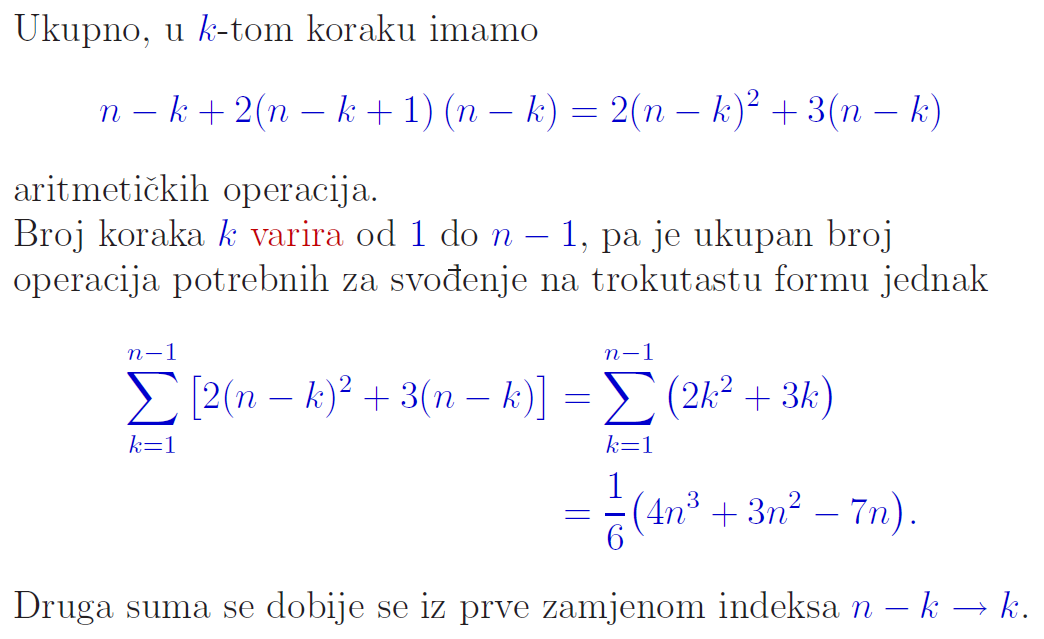
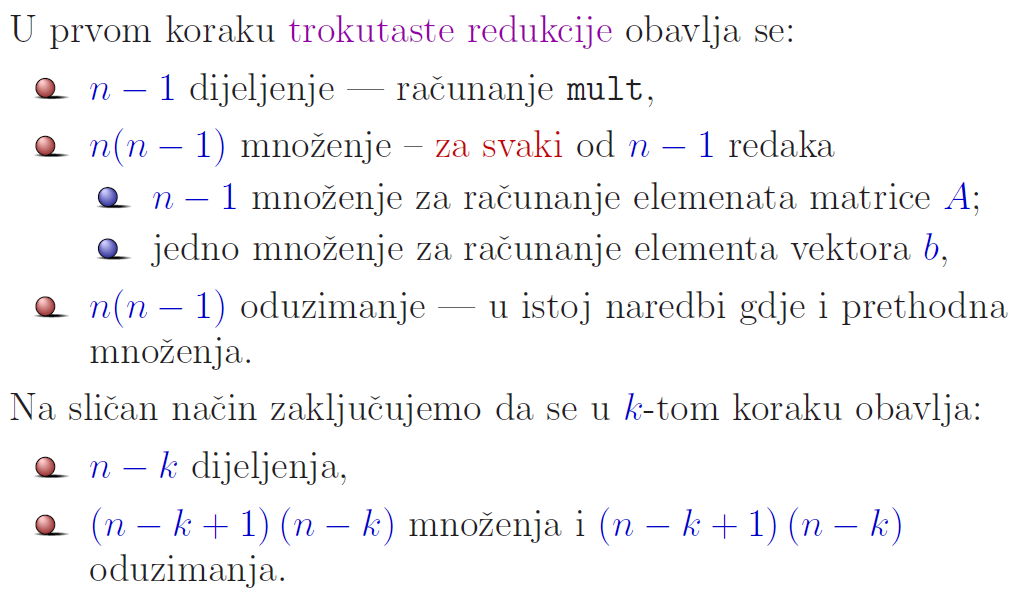
Gauss-Seidelove iteracije su oblika:

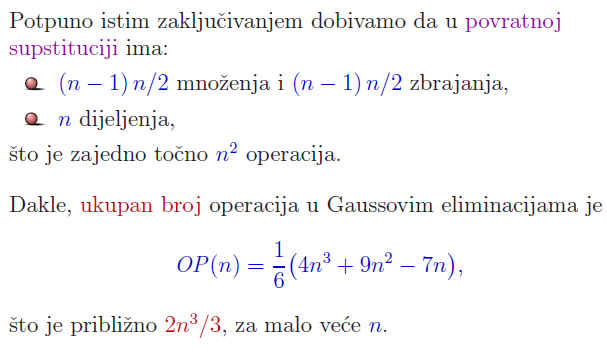
Prva iteracija:

Druga iteracija:

# Teorija

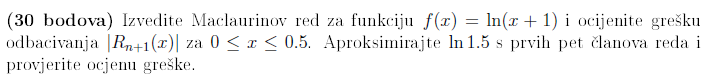






[**http://web.math.hr/~singer/num\_mat/NM\_0809/03.pdf**](http://web.math.hr/~singer/num_mat/NM_0809/03.pdf)

# Maclaurin



Rješenje:

|  |  |  |
| --- | --- | --- |
| 1. Nekoliko prvih derivacija  (dok se ne nađe opći član): | 2. Vrijednost u x=0: | 3. Maclaurinov red:  4. Prvih 5 članova reda su: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

5. Traži se ocjena greške nastale odbacivanjem svih članova nakon 5-og:

6. Pri procjeni napravljene greške tražimo najnepovoljniji slučaj u intervalu . Greška će biti maksimalna kada je u gornjem izrazu Uvrstimo nulu te dobijemo procjenu greške:

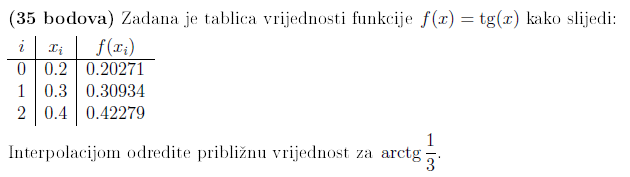
7. Za zadani interval greška odbacivanja najveća je za x=0.5, pa vrijedi:

Dakle, u zadanom intervalu greška ne može biti veća od 0.0026.

8. Aproksimacija za *ln 1.5*:

9. Ocjena greške:

# Interpolacija



Rješenje:

Pošto se traže inverzne vrijednosti od tg(x) onda se u tablici samo zamijene strane pa dobijemo vrijednosti koje se interpoliraju direktno:

|  |  |  |
| --- | --- | --- |
| i | xi | g(xi) |
| 0 | 0.20271 | 0.2 |
| 1 | 0.30934 | 0.3 |
| 2 | 0.42279 | 0.4 |

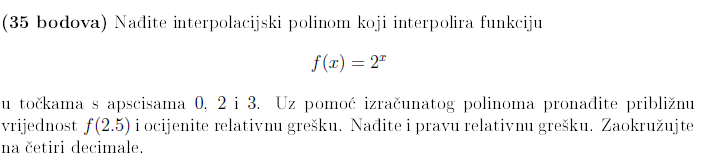
Nakon što se riješi ovaj sustav dobije se:

Pa je:

Uvrštavanjem u polinom dobije se:

Dakle približna vrijednost je:

# Interpolacija



Rješenje:

|  |  |  |
| --- | --- | --- |
| i | xi | f(xi) |
| 0 | 0 | 1 |
| 1 | 2 | 4 |
| 2 | 3 | 8 |

Nakon što se riješi ovaj sustav dobije se:

Pa je:

Uvrštavanjem u polinom dobije se:

Točna vrijednost je:

Relativna greška:

Ocjena relativne greške:

Prvo se nađe , formula je:

Ovisno kojeg je reda polinom, treba se raditi *n+1* derivacija funkcije. U našem slučaju polinom je drugog reda pa je *n=2+1=3*  i rade se 3 derivacije.

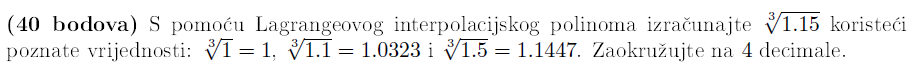
U treću derivaciju se uvrštava vrijednost iz zadanog intervala (intervala [0,3]) za koju funkcija poprima maksimalnu vrijednost, te se izračuna ta vrijednost:

pa imamo

Sada za ocjene relativne greške koristimo izraz:

Napomena: Izraz podrazumijeva da se treba napraviti *n* derivacija a u n-tu derivaciju se uvrštava vrijednost iz zadanog intervala za koju funkcija poprima maksimalnu vrijednost. Isto je za i , samo se na kraju traži minimum a ne maksimum.

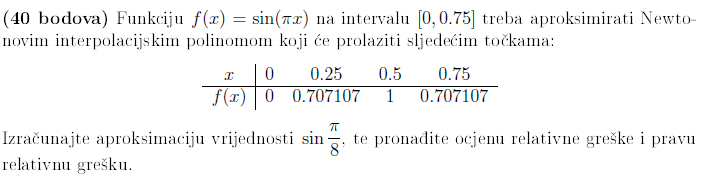
# Lagrangeov interpolacijski polinom



Rješenje:

Provjera:

# Newtonov interpolacijski polinom



Rješenje:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| xi | f(xi) |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Iz gornje tablice podjeljenih razlika dakle dobivamo (po formuli Newtonovog polinoma):

=0.3827

Ocjena relativne greške:

Prvo se nađe , formula je:

Ovisno kojeg je reda polinom, treba se raditi *n+1* derivacija funkcije. U našem slučaju polinom je trećeg reda pa je *n=3+1=4*  i rade se 4 derivacije.

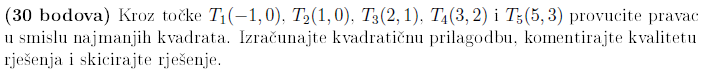
U četvrtu derivaciju se uvrštava vrijednost iz zadanog intervala () za koju funkcija poprima maksimalnu vrijednost (u ovom slučaju to je 0.5 to jest ) , te se izračuna ta vrijednost:

pa imamo

Sada za ocjenu relativne greške koristimo izraz:

Prava relativna greška (ona koja je stvarno napravljena) je:

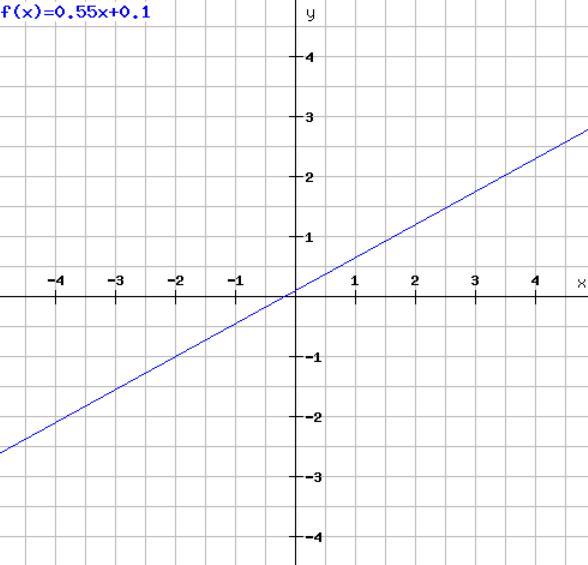
# Metoda najmanjih kvadrata



Rješenje:

=

=



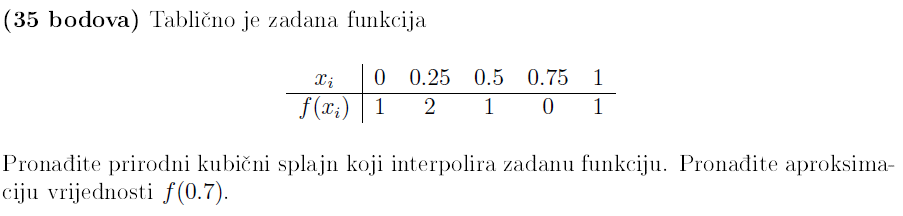
Napomene:

- U drugom stupcu matrice A su uvijek samo jedinice, prvi stupac se izvuče iz zadanih točaka.

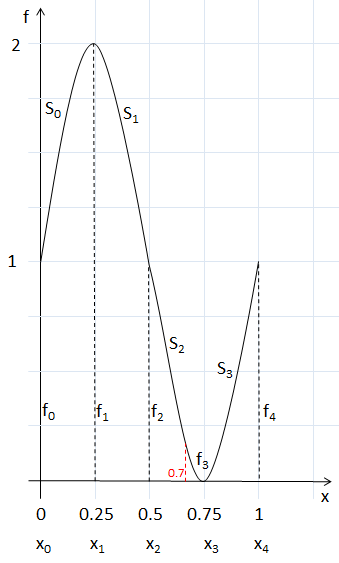
- Prilagodba je dobra ako je blizu nule (manja od 0.5). Ako je veća od 0.5 onda nije dobra.

- Na grafu se još mogu nacrtati zadane točke T, čime se može vidjeti koliko dobro izračunati pravac aproksimira te točke.

# Kubični splajn

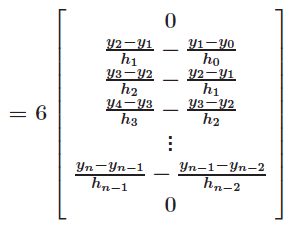
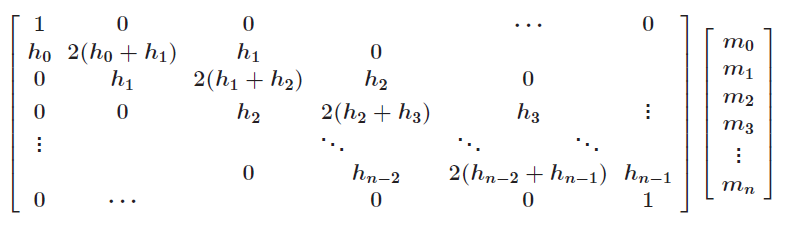


Pomoću kubičnog splajna zadana funkcija se aproksimira tako da se za svaki interval odredi polinom trećeg reda koji aproksimira taj dio funkcije. Grafički to izgleda kao na slijedećoj slici (dakle tražimo polinome S0 do s3):



Prvo se odrede duljine svakog od intervala (duljine mogu biti iste ili različite).

Nakon što se nađu duljine intervala, koristi se slijedeći oblik sustava linearnih jednadžbi.



Dakle na 3 središnje dijagonale su 'koraci' a ostalo su nule, pa imamo:

=

Nakon što se riješi sustav dobijemo da je . Pošto se radi o prirodnom kubičnom splajnu onda su (inače da nije zadan prirodni kubični splajn onda bi trebale biti zadane te rubne vrijednosti). Sada se izvode polinomi za svaki interval pomoću slijedećih izraza:

gdje je

,

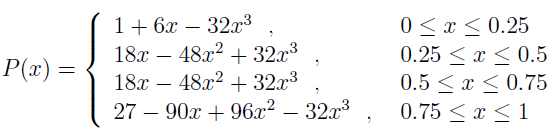
Za prvi interval imamo:

, , ,

Pa je prvi interval aproksimiran sa:

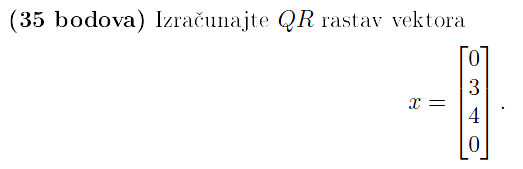
Slično se napravi i za ostale intervale pa dobijemo:

Rezultat je dakle:



Još trebamo naći aproksimaciju vrijednosti f(0.7). Ova točka se nalazi u intervalu [0.5, 0.75] pa onda koristimo polinom izračunat za taj interval, to jest:

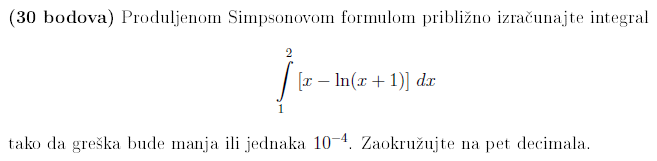
# QR rastav



Rješenje:

U vektoru r je prvi član suprotnog predznaka a ostali su uvijek nule. Sada tražimo vektor v.

# Simpsonova formula



Rješenje:

1. Da bi greška bila manja od zadane vrijednosti treba se napravit *n* koraka; *n* se nalazi pomoću slijedeće nejednakosti:

|  |  |
| --- | --- |
|  | gdje je |

U intervalu četvrta derivacija je maksimalna za *x=1* pa imamo:

Napomena: n mora biti cijeli i paran broj.

2. Korak *h* je:

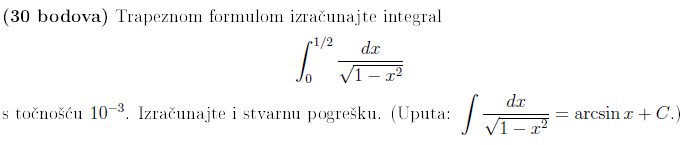
3. Radi se tablica sa *n* koraka duljine *h* i računaju se funkcijske vrijednosti:

|  |  |  |
| --- | --- | --- |
| i | xi | f(xi)= xi-ln(xi+1) |
| 0 | 1 | 0.30685 |
| 1 | 1.25 | 0.43907 |
| 2 | 1.5 | 0.58371 |
| 3 | 1.75 | 0.73840 |
| 4 | 2 | 0.90139 |

4. Prema Simpsonovoj formuli računa se:

Napomena: U nekim zadacima se odmah daju funkcijske vrijednosti (npr. mjerenja itd). U tom slučaju se te vrijednosti jednostavno uvrste u formulu.

# Trapezna formula



Rješenje:

1. Da bi greška bila manja od zadane vrijednosti treba se napravit *n* koraka; *n* se nalazi pomoću slijedeće nejednakosti:

|  |  |
| --- | --- |
|  | gdje je |

U intervalu [0,] druga derivacija je maksimalna za , pa imamo:

Napomena: n mora biti cijeli broj.

2. Korak *h* je:

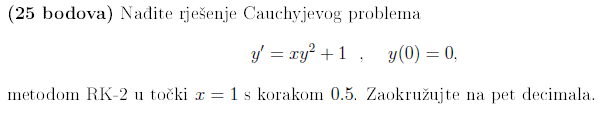
3. Radi se tablica sa *n* koraka duljine *h* i računaju funkcijske vrijednosti:

|  |  |  |
| --- | --- | --- |
| i |  |  |
| 0 | 0 | 1 |
| 1 | 0.083333 | 1.00349 |
| 2 | 0.166667 | 1.014185 |
| 3 | 0.25 | 1.032796 |
| 4 | 0.333333 | 1.06066 |
| 5 | 0.416667 | 1.100038 |
| 6 | 0.5 | 1.154701 |

4. Prema Trapeznoj formuli računa se:

5. Stvarna pogreška (računa se u radijanima):

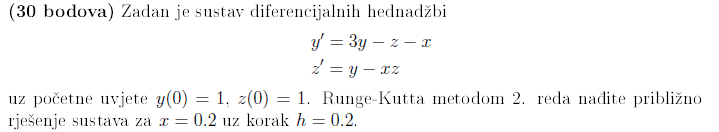
# RK-2 metoda



Rješenje:

Prva iteracija:

Druga iteracija:

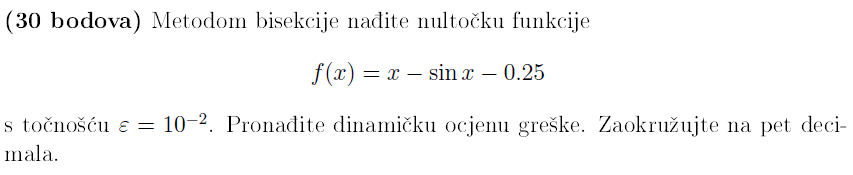


Rješenje:

# RK-4 Metoda

RK-4 metoda je slična RK-2, samo ima više koraka u svakoj iteraciji. Formula je:

# Metoda bisekcije



Rješenje:

Prvo se treba naći interval unutar kojeg se nalazi nultočka (pošto nije već zadan). Najbolje je nacrtati funkciju i vidjeti gdje su presjecišta a može se napraviti i tablica sa vrijednostima funkcija u proizvoljnim točkama, npr:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 0 | 0.25 | 1 | 1.57=π/2 | 2 |
| x-25 | -0.25 | 0 | 0.75 | 1.32 | 1.75 |
| sin(x) | 0 | 0.247 | 0. 84 | 1 | 0.909 |

Vidimo da se u intervalu [1,2] funkcije sijeku pa bi tu trebala biti nultočka. Točnost odabranog intervala se može i lako provjeriti tako da se uvrste rubovi intervala u zadanu funkciju (funkcije bi onda trebale biti različitog predznaka). Provjera:

|  |  |  |
| --- | --- | --- |
| =1-sin(1)-0.25<0  =2-sin(2)-0.25>0 | } | Interval x∊[1,2] je dobar |

Napomena:

Što se izabere bolji interval to će biti manje iteracija/polovljenja poslije.

Nakon što je određen interval traži se broj iteracija koji će zadovoljiti zadanu točnost, pomoću slijedeće formule:

Sada se rade iteracije/polovljenja:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | a | b | f(a) | f(b) | c=(a+b)/2 | f(c) |
| 1 | 1 | 2 | -0,09147 | 0,84070 | 1,5 | 0,25250 |
| 2 | 1 | 1,5 | -0,09147 | 0,25251 | 1,25 | 0,051011 |
| 3 | 1 | 1,25 | -0,09147 | 0,05102 | 1,125 | -0,0272 |
| 4 | 1,125 | 1,25 | -0,02727 | 0,05102 | 1,1875 | 0,01006 |
| 5 | 1,125 | 1,1875 | -0,02727 | 0,01006 | 1,15625 | -0,00904 |
| 6 | 1,15625 | 1,1875 | -0,00905 | 0,01006 | 1,171875 | 0,00039 |

Ako su f(a) i f(c) različitog predznaka onda u je slijedećoj iteraciji a=a, b=c.

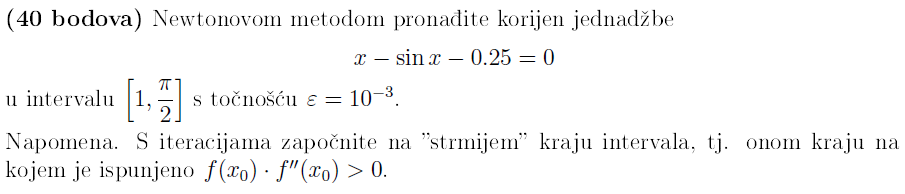
Ako su f(a) i f(c) istog predznaka onda u je slijedećoj iteraciji a=c, b=a.

Ako je f(c)=0 onda je pronađena nultočka te se iteracije prekidaju (rijedak slučaj).

Za rezultat se uzima vrijednost c=(a+b)/2 iz zadnje iteracije. Dakle riješenje je:

Dinamička ocjena greške je:

# Newtonova metoda (metoda tangente)



Rješenje:

|  |  |  |
| --- | --- | --- |
|  | } |  |
|  |

Dinamička ocjena greške / zahtjev točnosti:

|  |  |  |
| --- | --- | --- |
|  | } |  |
|  |

|  |  |  |
| --- | --- | --- |
|  | } |  |
|  |

Iteracije (počinju od , sa strmijeg kraja):

Uvjet je zadovoljen pa se prekidaju iteracije.

Horner

MinMax

Metoda sekante

Ostalo?