Numerične metode – izpitna vprašanja

1. Kako sta definirani absolutna in relativna napaka? Kako se širi relativna napaka pri seštevanju, odštevanju, množenju in potenciranju?

$A=|V_T-V_P|$; $R=A/V_P$

$$relativna\_napaka=\delta y=\sum_{i=1}^{n}\frac{x_i}{f(x_1,\dots,x_n)}\cdot\frac{\partial f(x_1,\dots,x_n)}{\partial x_i}\delta x_i$$

2. Kako ocenimo začetni približek nelinearne enačbe?

- Funkcijo narišemo in iz slike vidimo kjer seka realno os.
- Lahko pa tudi pogledamo odsek, dovolj majhen (če sta si dva korena blizu), v katerem funkcija menja predznak. Algoritem poteka takole:
  3. Postavimo index i=0, izberemo začetno vrednost x_i, izberemo korak h in izračunamo $f_i=f(x_i)$. Če je $f_i=0$, je $x_i=\alpha$ koren. Gremo na peti korak
  4. Izračunamo $x_{i+1}=x_i+h$ in $f_{i+1}=f(x_{i+1})$. Če je $f_{i+1}=0$, je $x_{i+1}=\alpha$ koren. Gremo na 5 korak.
  5. Če je $f_i f_{i+1}>0$, v intervalu (x_i,x_{i+1}) ni korena. Gremo na 5 korak.
  6. Če je $f_i f_{i+1}<0$, je koren v intervalu (x_i,x_{i+1}). Gremo na 5 korak.
  7. Če smo dosegli zgornjo mejo intervala, kjer iščemo koren, iteracijo zaključimo. Sicer povečamo index i in se vrnemo na drugi korak.

Korak h je ocena napake, s katero je določen koren, saj velja $|x_i-\alpha|<h$ in $|x_{i+1}-\alpha|<h$.
Predznak koraka h določa smer iskanja korena.

8. Opišite navadno iterativno metodo za reševanje nelinearne enačbe! Kako ocenimo napako korena? Ali je možno navadno iterativno metodo izboljšati?

Prevedemo na $x=g(x)$
$\alpha=g(\alpha)\Rightarrow f(\alpha)=0$
- Postavimo i=0, začetni približek x_0, ε, max število iteracij
- Izračunamo nov približek x_{i+1} po formuli $x_{i+1}=g(x_i)$
- Po predpisanem številu iteracij računanje prekinemo če x_{i+1} ni v okviru napake.
- Če $\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right|>\varepsilon$ povečamo i in gremo nazaj na 2 korak sicer je x_{i+1} dober približek.

9. Opišite Newtnowo ali tangentno metodo za reševanje nelinearne enačbe! Kako ocenimo napako korena? ~~Kaj veste o konvergenci metode?~~

Pri Newtnovi metodi izberemo začetni približek x_0 in nadomestimo funkcijo s tangento v točki $(x_0,f(x_0))$. Nato poiščemo mesto, kjer tangenta seka os x. Tako dobimo nov približek korena x_1. Zopet postavimo tangento v točki $(x_1,f(x_1))$ in postopek ponovimo.

$$f'(x_0)=tg\varphi=\frac{f(x_0)}{x_0-x_1}$$
$$x_1=x_0-\frac{f(x_0)}{f'(x_0)}$$
$$x_{i+1}=x_i-\frac{f(x_i)}{f'(x_i)}$$

algoritem:
- Izračunamo f'(x)
- Postavimo index i, izberemo začetni približek x_0, dopustno relativno napako in maksimalno število iteracij.
- Izračunamo nov približek x_{i+1}= po formuli $x_{i+1}=x_i-\frac{f(x_i)}{f'(x_i)}$
- Če je izvedeno predpisano število iteracij računanje prekinemo. x_{i+1} ni primeren približek korena.
- Če je $|(x_{i+1}-x_i)/x_{i+1}|>\varepsilon$, povečamo indeks i in se vrnemo na tretji korak. Sicer je x_{i+1} dober približek korena. Računanje lahko prekinemo.

Pri Newtnovi metodi se z vsakim korakom število točnih cifer podvoji. Zato pravimo, da ima ta metoda kvadratično konvergenco. $\varepsilon_{i+1}?\varepsilon_i^p$ p?2
Ocena napake:

$$\varepsilon_{i+1}=y_{i+1}-\xi=x_i-\frac{f(x_i)}{f'(x_i)}-\xi=\frac{f''(x_i)}{2}(x_i-\xi)^2=f''\frac{x_i}{2}\varepsilon_i^2\qquad \varepsilon_i=k\cdot\varepsilon_i^2$$

10. Opišite sekantno metodo za reševanje nelinearne enačbe! Kako ocenimo napako korena? ~~Kaj veste o konvergenci metode?~~

x_0,x_1 začetna približka

$$tg\varphi=\frac{f(x_0)-f(x_1)}{x_0-x_1}$$
$$tg=\frac{f(x_1)}{x_1-x_2}$$
$$x_2=x_1-\frac{f(x_1)(x_0-x_1)}{f(x_0)-f(x_1)}$$ (vstavi splošne indekse)

- i=1, začetni približek x_0 in x_1,ε, maksimalno število iteracij
- x_{i+1} po formuli zgoraj
- po predpisanem številu iteracij prekineš, x_{i+1} ni dober približek
- $\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right|>\varepsilon$ če je povečaš i in se vrneš na 2, sicer je x_{i+1} dober približek

Konvergenco je težko oceniti, je večja od navadne in manjša od Newtnove metode.

11. Opišite bisekcijsko metodo za reševanje nelinearne enačbe! Kako ocenimo napako korena? ~~Kaj veste o konvergenci metode?~~

Pri bisekcijski metodi moramo poiskati takšen interval (x_L,x_D), da v njem funkcija f(x) natanko enkrat menja predznak. Tak interval na vsakem koraku iteracije razpolovimo in obdržimo pri nadaljnem računanju tisto polovico intervala, ki ima na svojih krajiščih nasprotni predznak. Tako se z zmanjševanjem meji intervala bližamo korenu enačbe.

Algoritem:
- Določimo meji x_L in x_D tako, da velja $f(x_L)*f(x_D)<0$. Določimo še dopustno relativno napako ε in izberemo maksimalno število korakov iteracij. Za i postavimo 1.
- Izračunamo razpolovišče $x_i=(x_L+x_D)/2$ in funkcijski vrednosti $f(x_D)$ in $f(x_i)$. Če je $f(x_i)=0$ dobimo koren in lahko zaključimo z iteracijo.
- Če imata funkcijski vrednosti $f(x_L)$ in $f(x_i)$ nasprotni znak, $f(x_L)*f(x_i)<0$ se koren nahaja v intervalu (x_L,x_i) in lahko postavimo $x_D=x_i$. V nasprotnem primeru pa je $x_L=x_i$. Če je i=1, povečamo indeks in se vrnemo na drugi korak.
- Če je izvedeno predpisano število iteracij računanje prekinemo. x_i ni primeren približek korena.
- Če je $|(x_i-x_{i-1})/x_i|>\varepsilon$ povečamo indeks i in se vrnemo na drugi korak. Drugače je približek pravilen in z iteracijo lahko zaključimo.

Pot konvergence je le 1.
Ocena relativne napake:

$$x_i-x_{i-1}=\frac{x_D-x_L}{2}\qquad x_i=\frac{x_D-x_L}{2}\qquad \varepsilon=\frac{x_D-x_L}{x_L-x_D}$$

12. Opišite metodo regula falsi za reševanje nelinearne enačbe! Kako ocenimo napako korena? Kaj veste o konvergenci metode?

Pri metodi regula falsi moramo poiskati interval (x_L,x_D) tako, da v njem funkcija f(x) natančno enkrat menja predznak. Nato nelinearno funkcijo f(x). Zamenjamo s sekanto skozi točki $(x_L,f(x_L))$ in $(x_D,f(x_D))$ ter ugotovimo kje ta sekanta os x. Tako dobimo nov približek korena x_1.

$$tg\varphi=\frac{f(x_D)}{x_D-x_1}$$
$$tg\varphi=\frac{f(x_L)}{x_L-x_1}$$
$$x_1=\frac{x_L\cdot f(x_D)-x_D\cdot f(x_L)}{f(x_D)-f(x_L)}$$

Algoritem:
- Določimo meji intervala v katerem se nahaja koren, določimo dopustno relativno napako ε in maksimalno število iteracij. Postavimo i=1.
- Izračunamo približek $x_i=x_L-\frac{f(x_L)(x_L-x_D)}{f(x_L)-f(x_D)}$ in funkcijsko vrednost $f(x_i)$, če je le ta enaka 0 potem je x_i koren enačbe.
- Če sta funkcijski vrednosti $f(x_L)$ in $f(x_i)$ nasprotnega predznaka se koren nahaja v intervalu (x_L,x_i) in lahko postavimo $x_D=x_i \Rightarrow f(x_D)=f(x_i)$. V nasprotnem primeru je $x_L=x_i$ ter $f(x_L)=f(x_i)$. Če je indeks i=1 ga povečamo in se vrnemo na drugi korak.
- Če je izvedeno predpisano število iteracij, računanje prekinemo, x_i ni primeren približek korena.
- Če je $|(x_i-x_{i-1})/x_i|>\varepsilon$ povečamo indeks i in se vrnemo na drugi korak sicer pa lahko zaključimo iteracijo.

Metoda regula falsi hitreje konvergira h korenu kot bisekcijska metoda. Ocena napake ni vedno zanesljiva.

13. Opišite Hornerjev algoritem za računanje vrednosti polinoma! V katerih metodah za računanje korenov polinomov ga lahko koristno uporabimo?

Splošna oblika polinoma:
$P_n(x)=a_nx^n+a_{n-1}x^{n-1}+\dots+a_2x^2+a_1x+a_0$
Lahko ga zapišemo tudi:
$P_n(x)=(\dots((a_nx+a_{n-1})x+a_{n-2})x+\dots a_1)x+a_0$

P_n	$a_n^{(0)}$	$a_{n-1}^{(0)}$	$a_{n-2}^{(0)}$	...	$a_1^{(0)}$	$a_0^{(0)}$
$x=x_0$		$a_n^{(0)}x_0$	$a_{n-1}^{(1)}x_0$	...	$a_2x_0^{(1)}$	a_1x_0
	$a_n^{(1)}$	$a_{n-1}^{(1)}$	a_{n-2}	...	$a_1^{(1)}$	$R=P_n(x_0)$

$P_{n-1}^{(1)}=a_n^{(1)}x^{n-1}+a_{n-1}^{(1)}x^{n-2}+\dots+a_2^{(1)}x+a_1^{(1)}$
$P_n(x)=P_{n-1}^{(1)}(x)(x-x_0)+R$
$P_n(x)=P_{n-1}^{(1)}x_0(0)+R$
$P'(x)=P_{n-1}^{(1)}(x-x_0)+P_{n-1}^{(1)}(x)$
$P'(x_0)=P_{n-1}^{(1)}x_0$
$P_n(x_0)=0$
$x_{i+1}=x_i-P_n(x_i)/P_n'(x_i)$

P_{n-1}				
$x=x_0$		$a_n^{(2)}x_0$	$a_{n-1}^{(2)}x_0$	$a_2^{(2)}x_0$
	$a_n^{(2)}$	$a_{n-1}^{(2)}$	$a_{n-2}^{(2)}$	$a_1^{(2)}=P_n(x_0)$

P_1 $x=x_0$	$a_n^{(n-1)}$	$a_{n-1}^{(n-1)}$ $a_n^{(n)}x_0$	
$P_n^{(n)}(x_0)$	$a_n^{(n)}$	$a_{n-1}^{(n)}=1/(n-1)!$	$P_n^{(n-1)}(x)$

$P(x)=P(x_0)+P'(x_0)(x-x_0)+P''(x_0)(x-x_0)^2/2!+\dots$

1. Opišite princip direktnih metod za reševanje sistemov linearnih enačb!

$A=l^*R$ $\quad l_{ii}=1$ $\quad i=1,\dots,n$
$A_x=b$
$LR_x=L^{-1}*b=C$
Zgornji trikotnik $R_x=C$
Spodnji trikotnik $LC=b$
$l_{11}*C_1=b_1\Rightarrow C_1=b_1/l_{11}$
$l_{21}*C_1+l_{22}C_2=b_2\Rightarrow C_2=(b_2-l_{21}C_1)/l_{22}$
$x_n=C_n/r_{nn}$

2. Opišite Gaussov algoritem za reševanje sistemov linearnih enačb! Na katere težave lahko naletimo pri izvajanju tega postopka?

- Tvorimo razširjeno matriko Ã. Sestavljena je iz koeficientov sistema enačb in desne strani. Število vrstic je n in število stolpcev n+1.
- Po potrebi jo normiramo, k=1
- V k-tem stolpcu med elementi a, l=k,k+1,…,n poiščemo največjega in zamenjamo vrstici, da le ta postane pivot.
- Izvršiti moramo n-k eliminacij, tako da tvorimo ničle pod pivotom. Ničel ni potrebno računati. Na novo pa izračunamo vrstice od k+1 dalje. $a_{ij}^*=a_{ij}-\frac{a_ik}{a_*k}$
  *-elementi, ki so na novo izračunani.
- Če je k enak n-1 gremo na naslednji korak , sicer pa k povečamo in se vrnemo na tretji korak.
- Dobimo zgornjo matriko in izračunamo rešitve po formulah

$$x_n=\frac{a_{n,n-1}^*}{a_{n,n}^*}\qquad x_*=\frac{a_{k,n+1}^*-\sum_{j=k+1}^{n}a_j^*x_j}{a_{**}^*}\qquad k=n-1,n-2,\dots 1$$

3. Kako izvedemo LU razcep kvadratne matrike?

LU razcep je razcep permutirane matrike A na produkt spodnje trikotne matrike L, ki ima enice v glavni diagonali in zgornje trikotne matrike U.

$$U=\begin{bmatrix}u_{11}&u_{12}&\dots&u_{1n}\\0&u_{22}&\dots&\\\vdots&&\ddots&\\0&\cdots&0&u_{nn}\end{bmatrix}\quad L=\begin{bmatrix}1&&&\\l_{21}&\ddots&&\\\vdots&&\ddots&\\l_{n1}&\cdots&l_{n,n-1}&1\end{bmatrix}\quad A=\begin{bmatrix}a_{11}&a_{12}&\dots&a_{1n}\\\vdots&&&\\a_{n1}&\dots&\dots&a_{nn}\end{bmatrix}\quad B=\begin{bmatrix}b_1\\b_n\end{bmatrix}$$

Algoritem LU razcepa
- Izvedemo Gaussov eliminacijski postopek in dobimo matriki L in U.
- Izračunamo pomožni faktor $b_1'=b_1$ $\quad b_i'=\Sigma_{k=1}^{i-1}l_{ik}b_k'$ (1=2,3,…,n) kjer so l_{ik} elementi matrike L.
- Izračunamo vektor x po formulah.

$$x_n=\frac{b_n'}{u_{nn}}\qquad x_i=\frac{b_i'-\sum_{k=i+1}^{n}u_{ik}x_k}{u_{ii}}\qquad i=n-1,n-2,\dots,1$$

4. Kako izračunamo determinanto kvadratne matrike?

Matriko, ki ni pravokotna prevedemo v trikotno determinanto. Upoštevati moramo lastnosti determinante:
- Z zamenjavo vrstic le ta spremeni svoj znak
- S poljubnim faktorjem pomnožene vrstice in stolpce lahko seštevamo
- Če pomnožimo s konstantno vrstico, množimo s tem celo determinanto

Trikotno determinanto razvijemo po prvem stolpcu tako, da ostanejo elementi nad glavno diagonalo in elementi glavne diagonale. Ostali elementi so 0. Če nadaljujemo z razvijanjem po k-faktorjih, se izračuna trikotna determinanta na naslednji način.

$$|A|=\prod_{i=1}^{n}a_{ii}$$

5. Kako izračunamo inverzno matriko kvadrantne matrike?

Imamo kvadratno matriko A reda n x n. Če je matrika nesingularna, obstaja takšna matrika B, da je AB=BA=I. Matriko B označimo z A^{-1} in ji rečemo inverzna matrika. Računanje inverzne matrike AB=I sde prevede na reševanje sistema n^2 enačb z n^2 neznankami. Te so elementi matrike B (b_{ij}, i=1,2,…,n; j=1,2,…,n)

6. Opišite Gauss-Jordanov algoritem za reševanje sistemov linearnih enačb! Na ~~katere težave lahko naletimo pri izvajanju tega postopka~~?

Gauss-Jordanov algoritem za reševanje sistemov linearnih enačb je varijanta Gaussove eliminavijske metode. Na vsakem koraku eliminacije eliminiramo po eno neznanko iz enačb pod in nad glavno diagonalo. Sistem enačb prevedemo v ekvivalentni diagonalni sistem oz. razširjeno matriko koeficientov prevedemo v ekvivalentno diagonalno matriko koeficientov z 1 na diagonali. (Delimo oziroma množimo posamične vrstice poleg eliminacije)

7. Opišite metodo Choleskega za reševanje sistemov linearnih enačb! Kakšne sisteme linearnih enačb lahko rešujemo po tej metodi?

Matriki A priredimo transponirano matriko A^T. Elementi obeh matrik so povezani v zvezo $(a^T)_{ji}=a_{ij}$; i,j=1,2,…,n. Simetrična je tako kvadratna matrika za katero velja $A^T=A$ $\quad a_{ji}=a_{ij}$ i,j=1,2,…,n. Vsaka simetrična matrika se da razcepiti v produkt dveh matrik. $A=U^T*U$. U je zgornja trikotna matrika, U^T pa spodnja trikotna matrika.

$$\begin{bmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\a_{31}&a_{32}&q_{33}\end{bmatrix}=\begin{bmatrix}u_{11}&0&0\\u_{21}&u_{22}&0\\u_{31}&u_{32}&u_{33}\end{bmatrix}\begin{bmatrix}u_{11}&u_{12}&u_{13}\\0&u_{22}&u_{23}\\0&0&u_{33}\end{bmatrix}$$

$$a_{11}=u_{11}^2\qquad a_{12}=u_{11}\cdot u_{12}\qquad a_{13}=u_{11}\cdot u_{13}$$
$$a_{22}=u_{12}^2+u_{22}^2\qquad a_{23}=u_{13}\cdot u_{12}+u_{23}u_{22}\qquad a_{33}=u_{13}^2+u_{23}^2+u_{33}^2$$

Če vstavljamo že izračunane neznanke v enačbe lahko rešimo zgornji sitem enačb.

$$u_{11}=\sqrt{a_{11}}\quad u_{1i}=\frac{a_{1i}}{u_{11}}\quad u_{22}=\sqrt{a_{22}-u_{12}^2}\quad u_{23}=\frac{u_{23}-u_{13}\cdot u_{12}}{u_{22}}\quad u_{33}=\sqrt{a_{33}-u_{13}^2-u_{23}^2}$$

Po tej metodi lahko razcepimo matriko A na produkt dveh transponiranih trikotnih matrik samo kadar je matrika koeficientov simetrična in pozitivno definirana.

8. Opišite gradientno metodo za reševanje sistemov linearnih enačb! Kakšne sisteme linearnih enačb lahko rešujemo po tej metodi? Kdaj je metoda hitrejša od drugih metod?

Pri gradientni metodi iščemo namesto rešitev sistema enačb, minimum funkcije več spremenljivk, ki ima lastnost, da doseže absolutni minimum pri tistih vrednostih, ki rrešijo sisteme linearnih enačb. $A_X=b$ – minimiziramo, $F(x)=x^TA_X/2-x^TB$
Algoritem:
- Izberemo začetni približek, zračunamo ostanek $v_0=A_{X0}-b$, postavimo $a_0=v_0$. Števec postavimo i=0.
- Izračunamo pomožni vektor $q_i=A*u_i$ in skalar $c_i=v_iTu_i/q_iTu_i$
- Izračunamo nov približek $x_{i+1}=x_i+c_i*u_i$
- Izračunamo ostanek $v_{i+1}=v_i+c_i*q_i$
- Izračunamo faktor $d_i=-(v_i+q_i)/u_i^T*Au_i$
- Izračunamo vektor $u_{i+1}=v_i+d_i*u_i$
- Števec i povečamo in se vrnemo na prvi korak
- Če se zaporedni približki ločijo za manj kot je dopustna napaka, po 3. koraku prekinemo iteracijo.

Ostanek v_i je po n-iteracijah enak 0. Tako dobimo točno rešitev sistema enačb.

9. Opišite Jacobijevo metodo za reševanje sistemov linearnih enačb. Kakšne sisteme linearnih enačb lahko rešujemo po tej metodi?

Sistem enačb izrazimo v obliki $\sum_{j=1}^{n}a_{ij}x_j=b_i$ (i=1,2,…,n). Pri iterativni Jacobijevi metodi izberemo začetni približek, vektor $x^{(0)}$. Z indeksom v oklepaju štejemo zaporedno število iteracij. Komponento vektorja $x^{(0)}$ vstavimo v enačbo: $x_i^1=\frac{1}{a_{ii}}(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^0-\sum_{j=i+1}^{n}a_{ij}x_j^0)$ (i=1,2,…,n). Tako dobimo nov približek. Ta postopek ponavljamo dokler ne dosežemo želene natančnosti. Na k-tem koraku iteracije pa lahko izračunamo želeni približek po formuli: $x_i^{k+1}=\frac{1}{a_{ii}}(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^k-\sum_{j=i+1}^{n}a_{ij}x_j^k)$ (i=1,2,…,n) Pri tej metodi izračunano novo komponento rešitve uporabimo za računanje nadaljnih komponent približkov. Konvergenca je zagotovljena povsod tam, kjer je v vsaki vrstici matrike koeficientov absolutna vrednost diagonalnega elementa večja od vsote absolutnih vrednosti izven diagonalnih elementov.

Formule: $$x_i^{k+1}=\frac{1}{a_{ii}}(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^k-\sum_{j=i+1}^{n}a_{ij}x_j^k)\qquad x_i^{k+1}=x_i^k+\frac{R_i^k}{a_{ii}}$$

10. Opišite Gauss-Seidlovo metodo za reševanje sistemov linearnih enačb. Kakšne sisteme linearnih enačb lahko rešujemo po tej metodi?

Je podobna metoda jakobijevi, le da tu izračunano novo komponento rešitve takoj uporabimo za računanje nadaljnih komponent približkov. V vsaki vrstici v matriki koeficientov mora biti absolutna vrednost diagonal elementa večja od vsote absolutnih vrednosti nediagonal elementov.

$$x_i^{k+1}=\frac{1}{a_{ii}}(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{k+1}-\sum_{j=i+1}^{n}a_{ij}x_j^k)\qquad i=1,\dots,n$$

predelamo v $$x_i^{k+1}=x_i^k+\frac{R_i^k}{a_{ii}}\qquad R_i^k=b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{k+1}-\sum_{j=i+1}^{n}a_{ij}x_j^k$$

11. Opišite navadno iteracijsko metodo za reševanje sistemov linearnih enačb! Kakšne sisteme linearnih enačb lahko rešujemo po tej metodi?

Rešujemo: $f_1(x,y)=0$ $\quad f_2(x,y)=0$
Ta sistem prevedemo v ekvivalentnega: $x=g_1(x,y)$ $\quad y=g_2(x,y)$
Algoritem:
- Postavimo indeks i=0, izberemo začetni približek (x_0,y_0), max število iteracij in relativno napako ε.
- Izračunamo nov približek (x_{i+1},y_{i+1})po formulah $x_{i+1}=g_1(x_i,y_i)$ $y_{i+1}=g_2(x_i,y_i)$
- Če je izvedeno predpisano število iteracij računanje prekinemo. (x_{i+1},y_{i+1}) ni primeren približek.
- Če je $\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right|+\left|\frac{y_{i+1}-y_i}{y_{i+1}}\right|>\varepsilon$ povečamo indeks i in se vrnemo na drugi korak. Drugače smo dobili rešitev in računanje zaključimo.

Napaka:
$\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right|+\left|\frac{y_{i+1}-y_i}{y_{i+1}}\right|$ izračun mora biti manjši od dopustne napake ε.

12. Opišite Newtnovo metodo za reševanje sistemov nelinearnih enačb! Kako ocenimo napako rešitve?

Po Newtnowi metodi najprej ocenimo začetni približek. Nato pa popravimo le tega in dobimo boljši približek. $x_{i+1}=x_i+?x_i$ $\quad y_{i+1}=y_i+?y_i$
Povpravka $(?x_i, ?y_i)$:

$$\frac{\partial f_1}{\partial x}(x_i,y_i)\Delta x_i+\frac{\partial f_1}{\partial y}(x_i,y_i)\Delta y_i=-f_1(x_i,y_i)$$
$$\frac{\partial f_2}{\partial x}(x_i,y_i)\Delta x_i+\frac{\partial f_2}{\partial y}(x_i,y_i)\Delta y_i=-f_2(x_i,y_i)$$

Alogritem:
- Izračunamo parcialne odvode
- Postavimo indeks i=0, izberemo začetni približek (x_0,y_0), ε in max število iteracij.
- Izračunamo vrednosti parcialnih odvodov in obeh funkcij pri tekočem približku.
- Izračunamo nov približek (x_{i+1},y_{i+1}). Če je i=0 se vrnemo na tretji korak.
- Če jue izvedeno predpisano število iteracij, računanje prekinemo (x_{i+1},y_{i+1}) ni primeren približek.
- Če je $\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right|+\left|\frac{y_{i+1}-y_i}{y_{i+1}}\right|>\varepsilon$ povečamo indeks i in se vrnemo na tretji korak. Sicer pa lahko zaključimo z iteracijo.

Ocena napake:
$\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right|+\left|\frac{y_{i+1}-y_i}{y_{i+1}}\right|<\varepsilon$.

13. Kako numerično izračunamo odvod tabelirane funkcije? ~~Napišite vsaj eno diferenčno formulo za prvi in drugi odvod tabelirane funkcije in podajte za vsako red napake!~~

$$x_{i+1}-x_i=h$$
$$y'=\lim_{x\to 0}\frac{y(x+\Delta x)-y(x)}{\Delta x}$$

Potegnemo neko funkcijo skozi tabelirane točke in naredimo napako. Omejimo se na izračun odvodov samo v tabeliranih točkah.

$$f(x+y)=f(x)+f'(x)h+\frac{f''(x)h^2}{2!}+\frac{f'''(x)}{2!}+\frac{f'''(x)h^3}{3!}$$
$$y_2=y_1+y_1'h+\frac{y''h^2}{2}+\frac{y'''h^3}{6}$$
$$y_0=y_1-y_1'h+\frac{y''h^2}{2}+\frac{y'''h^3}{6}$$

Prvi odvod: $$y_2-y_0=2hy'+\frac{2y'''h^3}{6}$$

Napaka: $$y_1'=\frac{y_2-y_1}{2h}-\frac{y'''(\xi)h^2}{6}$$

Drugi odvod $$y_1''=\frac{y_2-2y_1-y_0}{h^2}-\frac{h^2}{12}y^{(4)}(\xi)$$

14. Kako numerično izračunamo prvi odvod tabelirane funkcije? Kako zmanjšamo napako metode?

Funkcijo potegnemo skozitabelirane točke. Omejimo se na računanje odvodov v tabeliranih točkah. $x_{i+1}-x_i=h$ $\qquad y'=\lim_{\Delta x\to 0}\frac{y(x+\Delta x)-y(x)}{\Delta x}$

$$y_0'=\frac{h}{2}(-3y_0+4y_1-y_2)+\frac{1}{3}h^2y'''(\xi)$$

Bolj natančne formule: $$y_0'=\frac{h}{2}(y_0-4y_1+3y_2)+\frac{1}{3}h^2y'''(\xi)$$

15. Opišite trapezno metodo za računanje določenega integrala! Kako ocenimo napako?

Ploskev razdelimo na n trapezov.

$$\int_a^b f(x)dx\approx\sum_{i=0}^{n-1}\underbrace{\int_{x_i}^{x_{i+1}}f(x)dx}_{(x_1-x_0)\frac{y_0+y_1}{2}}=\sum_{i=0}^{n-1}\frac{x_{i+1}-xi}{2}(y_i+y_{i+1})=$$
$$=\frac{h}{2}(y_0+2y_1+2y_2+2y_3+\dots+2y_{n-2}+2y_{n-1}+y_n)-\underbrace{\frac{(b-a)h^2}{12}f(\xi)}_{napaka\_oh^2}$$

ocena napake

$$I=\underbrace{T_{\frac{n}{2}}+\frac{kh^4}{4}}_{T+kh^2}=T_{\frac{n}{2}+n}$$
$$n=\frac{kh^2}{4}=\frac{T_{\frac{n}{2}}-T_n}{3}$$

16. Opišimo Simpsonovo metodo za računanje določenega integrala! Kako ocenimo napako?

Simpsonova metoda je primerna za polinome do tretje stopnje.
Skozi tri točke položimi kvadratno funkcijo:
$Y=ax^2+bx+c$
$X_1=x_0+h$ $\quad x_2=x_0+2h$
Tudi za polinom tretje stopnje

$$\int_{x_0}^{x_0+2h}f(x)dx\approx\frac{h}{3}(f(x_0)+4f(x_1)+f(x_2))-\frac{h^5}{90}t^{(4)}(\xi)=\int_{x_0}^{x_0+2h}(kvadratna\_f)dx$$
$$\int_a^b f(x)dx=\sum_{i=0,2,4}^{n-2}\int_{x_i}^{x_{i+2}}f(x)dx=\frac{h}{3}(y_0+4y_1+2y_2+4y_3+\dots+4y_{n-1}+y_n)-\underbrace{\frac{b-a}{180}n^4f^{(4)}(\xi)}_{oh^4}$$

ocena napake

$$I=S_h+kh^4=S_{\frac{n}{2}}+h(\frac{h}{2})^4=S_{\frac{n}{2}}+\frac{kh^4}{16}=S_{\frac{n}{2}+n}$$
$$S_h+kh^4=S_{\frac{n}{2}}+\frac{kh^4}{16}$$
$$\frac{kh^4}{16}=\frac{S_{\frac{n}{2}}-S_n}{15}$$
$$I=S_{\frac{n}{2}}+\frac{S_{\frac{n}{2}}-S_n}{15}=\frac{16S_{\frac{n}{2}}-S_n}{15}=oh^6$$

17. Opišite Rombregovo metodo za računanje določenega integrala! Kako ocenimo napako?

R_{11}			
R_{21}	R_{22}		
R_{31}	R_{32}	R_{33}	
R_{41}	R_{42}	R_{43}	R_{44}

$$T_n^0=R_{11}$$
$$T_{\frac{n}{2}}^0=R_{21}$$
$$T_{\frac{n}{3}}^0=R_{31}$$

h=2,3,…,n $\qquad$ j=2,3,…,k

$$R_{k,j}=R_{k,j-1}+\frac{R_{k,j-1}-R_{k-1,j-1}}{4^{j-1}-1}$$

1. **Opišite Gaussovo metodo za računanje določenega integrala! ~~Kako ocenimo napako?~~**

$$\int_a^b f(x)dx = \cong \sum_{i=1}^{n} c_1 f(x_i)$$

$$t = \frac{2x-a-b}{b-a} \Rightarrow x = \frac{1}{2}((b-a)t+a+b)$$

$$\int_{-1}^{1} f(t)dt = \int_{-1}^{1} f(\frac{(b-a)t+a+b}{2}) \cdot \frac{b-a}{2}$$

$$dx = \frac{b-a}{2}dt$$

Dvotočkovna funkcija:

$$\int_{-1}^{1} f(x)dx = c_1 f(x_1) + c_2 f(x_2)$$

$$n = \frac{I_{\frac{n}{2}} - I_n}{2^{n+1}-1}$$

2. **Opišite Eulerjevo metodo za reševanje začetnega problema *y'(x)=f(x,y), a?x?b, y(a)=α* ! Koliko je red lokalne in globalne napake!**

X_0=a, y_0= α

$$y(x_1) = y_0 + \int_{x_0}^{x_0+h} f(x,y(x))dx \approx y_0 + hf(x_0,y_0) + \sigma h^2 = y_1$$

h=x_{n+1}-x_n (ekvidistančnost ni potrebna)

$$y(x_{n+1}) = y(x_n+h) = y(x_n) + hy'(x_n) + \sigma h^2$$

lokalna napaka: -h^2

Za vse začetne probleme je značilno, da je globalna napaka za en red manjša od lokalne. Se pravi h.

3. **Opišite Taylorjevo metodo za reševanje začetnega problema *y'(x)=f(x,y), a?x?b, y(a)=α* ! Koliko je red lokalne in globalne napake!**

y_0= α $\quad y_{i+1} = y_i + hT^{(n)}(x_i,y_i) \quad$ i=0,1,2,…,n-1

$$T^{(n)}(x_i,y_i) = f(x_i,y_i) + \frac{h}{2}f(x_i,y_i) + \ldots + \frac{h^{n-1}}{n!}f^{(n-1)}(x_i,y_i)$$

lokalna napaka: $\frac{1}{(n+1)!}y^{n+1}(\xi_i)h^{n+1} = \sigma(n^{n+1})$

globalna napaka je za en red nižja kot lokalna = σh^n

4. **Opišite metodo Runge Kutta četrtega reda za reševanje začetnega problema *y'(x)=f(x,y), a?x?b, y(a)=α* ! Koliko je red lokalne in globalne napake!**

y_0= α

$$h = x_{i+1} - x_i$$

$$k_1 = hf(x_i,y_i)$$

$$k_2 = hf(x_i + \frac{h}{2}, y_i + \frac{k_1}{2})$$

$$k_3 = hf(x_i + \frac{h}{2}, y_i + \frac{k_2}{2})$$

$$k_4 = hf(x_i + \frac{h}{2}, y_i + k_3)$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \underbrace{\sigma h^5}_{lokalna\_napaka}$$

5. **Opišite strelsko metodo za reševanje robnega problema *y''(x)=f(x,y,y'), a?x?b, y(a)=α, y(b)=β*!**

S to metodo prevedemo reševanje robnega problema na reševanje začetnega.

y'(a)=t

$$\lim_{h\to\infty} y(b,t) = y(b) = \beta$$

$$y(b,t) - \beta = 0$$

$$t_{i+1} = t_i(y(b,t_i) - \beta) - \frac{t_{i-1} - t_i}{y(b,t_{i-1}) - y(b,t_i)}$$

6. Opišite diferenčno metodo za reševanje robnega problema *y''(x)=f(x,y,y'), a?x?b, y(a)=α, y(b)=β*!
7. Opišite diferenčno metodo za reševanje linearnega robnega problema *y''(x)+p(x)y'+q(x)y=r(x), a?x?b, y(a)=α, y(b)=β*!
8. Opišite splošni princip metode najmanših kvadratov in zapišite sistem enačb za določitev koeficientov v primeru linearne regresije *y=a+bx*!
9. Opišite splošni princip metode najmanših kvadratov in zapišite sistem enačb za določitev koeficientov v primeru linearne kombinacije dveh funkcij *y=ah₁(x)+bh₂(x)*!
10. V čem se razlikujejo metode interpolacije od metod aproksimacije!

11. **Naj bo dana tabela funkcije. Kako konstruiramo Lagrangeov interpolacijski polinom? ~~Kakšna je napaka pri interpolaciji!~~**

Lagrangeov interpolacijski polinom :

$$P_n(x) = \sum_{k=0}^{n} y_k L_k^{(n)}(x)$$

$$L_k^{(n)}(x) = \frac{(x-x_0)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_n)} = \frac{\prod_k^n(x)}{\prod_k^n(x_k)} = lagrangeov\_koeficient$$

$L_k^{(n)}(x_i) = 1$ za i=k

$L_k^{(n)}(x_i) = 0$ za i?k

ocena napake:

$$f(x) - P_n(x) = \frac{1}{(n+1)!}f^{(n+1)}(\xi)\Pi_n$$

12. **Naj bo dana tabela funkcije. Kako konstruiramo Newtnow interpolacijski polinom? ~~Kakšna je napaka pri interpolaciji? Kdaj uporabljamo Newtnow interpolacijski polinom?~~**

$$P_n(x) = a_0 + a_1(x-x_1) + a_1(x-x_0)(x-x_1) + \ldots + a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1})$$

Določimo koeficiente a_i, i=0,1,…,n tako, da bo v vseh interpolacijskih točkah.

$P_n(x)=y_i$, i=0,1,…,n $\quad P_n(x)$_interpolacijski polinom

Koeficienti morajo ustrezati pogojem

$$y_0 = a_0$$

$$y_1 = a_0 + a_1(x_1 - x_0)$$

$$y_2 = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$$

$$\vdots$$

$$y_n = a_0 + a_1(x_n - x_0) + a_2(x_n - x_0)(x_n - x_1) + \ldots + a_n(x_n - x_0)(x_n - x_1) + \ldots(x_n - x_{n-1})$$

Dobimo trikotni sistem enačb, ki ga rešimo.

Računska shema

Ocena napake:

$$f(x) - P_m(x) = \binom{s}{m+1}k^{m+1}f^{m+1}(\xi)$$

$$f_{ij} = \frac{f_{i+1,j-1} - f_{i,j-1}}{x_{j+1} - x_i} \quad i=0,1,\ldots,m-j$$

$$P_m(x) = f_{00} + f_{01}(x-x_0) + f_{02}(x-x_0)(x-x_1) + f_{03}(x-x_0)(x-x_1)(x-x_2)$$

13. **Naj bo dana tabela funkcije. Opišite Aitkenovo metodo za računanje interpolacijskega polinoma pri dani vrednosti neodvisne spremenljivke! ~~Kdaj uporabimo Aitkenovo metodo?~~**

Po tej metodi računamo neposredno vrednost interpolacijskega polinoma in na vsakem naslednjem koraku višjega za eno stopnjo. Z računanjem prekinemo, ko se zaporedni interpolirani vrednosti ločita za manj, kot je dopustna napaka.

$P_{i,0}=y_i$ $\quad$ i=0,1,2,…,n

Elemente j-tega stolpca računamo po formuli:

$$P_{ij} = \frac{(x_i - x)P_{i-1,j-1} - (x_{j-1} - x)P_{i,j-1}}{x_i - x_{j-1}} \quad j=1,2,\ldots,m \quad i=j,j+1,\ldots,m-j$$

Z aitkenovo metodo računamo vrednosti int. Polinomov skozi m+1 točki in m ? n

Računska shema:

14. **Kaj računamo pri inverzni interpolaciji?**

Pri inverzni interpolaciji določamo vrednost spremenljivke x, ki ustreza vrednosti odvisne spremenljivke y, če je funkcijska vrednost y=f(x) dana s tabelo. Inverzna interpolacija je računanje vrednosti inverzne funkcije x=x(y)

Način reševanja:

- Podamo tabelo inverzne funkcije x=x(y). Le to dobimo tako, da zamenjamo pomen neodvisne in odvisne spremenljivke v tabeli funkcije y=f(x)
- Rešujemo enačbo f(x)=y
- Odvisnost (y=f(x)) je dana tabelarično in y konstanta
- Rešitev enačbe: f(x)-y=0
- Rešitev izračunamo po iterativni metodi, f(x) pa nadomestimo z interpolacijskim polinomom.

15. **Kaj so zlepki? Katerim pogojem morajo ustrezati kubični zlepki in kako jih izačunamo? ~~Zakaj imamo različne tipe kubičnih zlepkov?~~**

IDEJA: Potegniti krivuljo skozi vse točke. Primer: napnemo žico okoli žebljičkov, ki predstavljajo točke.

$S_i(x)=a_i+b_i(x-x_i)+c_i(x-x_i)^2+d_i(x-x_i)^3$

$S_i(x_i)=y_i$ => $a_i=y_i$

$S_i(x_{i+1})=y_{i+1}$ n-enačb

Kubični zlepki:

- S(x) je kubični polinom Si(x) na vsakem podintervalu $[x_i,x_{i+1}]$ (i=1,…,n)
- S(x) je interpolacijska funkcija, to pomeni, da mora imeti $S(x_i)=y_i$ (i=1,…,n)
- S(x) je zvezna funkcija z zveznim prvim in drugim odvodom

Vrste kubičnih zlepkov:

- Naravni kubični zlepki
- Kubični zlepki s predpisanimi prvimi odvodi na robovih
- Periodični kubični zlepki
- Zlepki brez dodatnih robnih pogojev
- Posplošeni naravni zlepki
- Kubični zlepki s predpisanim tretjim odvodom

Različne tipe kubičnih zlepkov imamo zato ker moramo sami izbrati še 2 pogoja, da dobimo še 2 enačbi za rešitev sistema enačb. Zlepka S predpišemo posebno…