numerika

Datum: 21.8.02

---

1. ABSOLUTNA IN RELATIVNA NAPAKA

Razlika med natančno rešitvijo matematičnega modela,pri danih podatkih,in izračunano rešitvijo po numerični metodi im.(imenujemo) ABSOLUTNA NAPAKA.Formalno jo lahko zapišemo:

$r_t = r_p + e_c$ – celotna napaka

r_t – točna rešitev modela; r_p – približna reš. mod.

$$\boxed{e_c = r_t - r_p}$$

Relativna napaka glede na točno vrednost pa je def.(definirana) takole:

$$\varepsilon_c = \frac{e_c}{r_t} = \frac{r_t - r_p}{r_t}$$

Običajno nimamo znane točne rešitve zato skušamo približno rešiti nalogo in oceniti napako.Približna relativna napaka :

$$\varepsilon_p = \frac{e_p}{r_p}$$ – ocena napake; – približna rešitev

Zaradi približkov pri nadaljnem računanju določimo dopustno relativno napako,ki nam ustreza.Ko rel. nap. Pade pod dop. Vrednost je za nas naloga rešena (dovolimo neko min. odstopanje od rezultata).

$$|\varepsilon_p| < \varepsilon_d \qquad \varepsilon_d = 0{,}5 \cdot 10^{-n}$$

n – št. točnih cifer

2. ZAČETNI PRIBLIŽEK NELINEARNIH ENAČB

Je sestavljen iz dveh faz:

1. Poiščemo interval kjer se koren nahaja
2. Izrač.(izračunamo) koren s predpisano natančnostjo

Interval lahko določimo tako da narišemo graf funkc. In določimo kje funkc. seka os x.Ali pa fukc. Tabeliramo in poiščemo zaporedni vrednost x_i in x_i+h tako da imata funkcijski vrednosti $f(x_i)$ in $f(x_i+h)$ nasprotni predznak.

ALGORITEM:

1. Postavimo 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 koren enačbe(zaključimo)
2. Če ne,izračunamo $x_i+h=x_{i+1}$ in $f_{i+1}=f(x_i+1)$.Računanje nadaljujemo dokler f_{i+1} ni enako 0(nič).
3. Če je $f_if_{i+1}>0$,v int.(intervalu) (x_i,x_{i+1}) ni korena (zaključimo).
4. Če je $f_if_{i+1}<0$,v int. (x_i,x_{i+1}) JE koren in nadaljujemo z računanje dokler ne dosežemo dop. rel. nap.

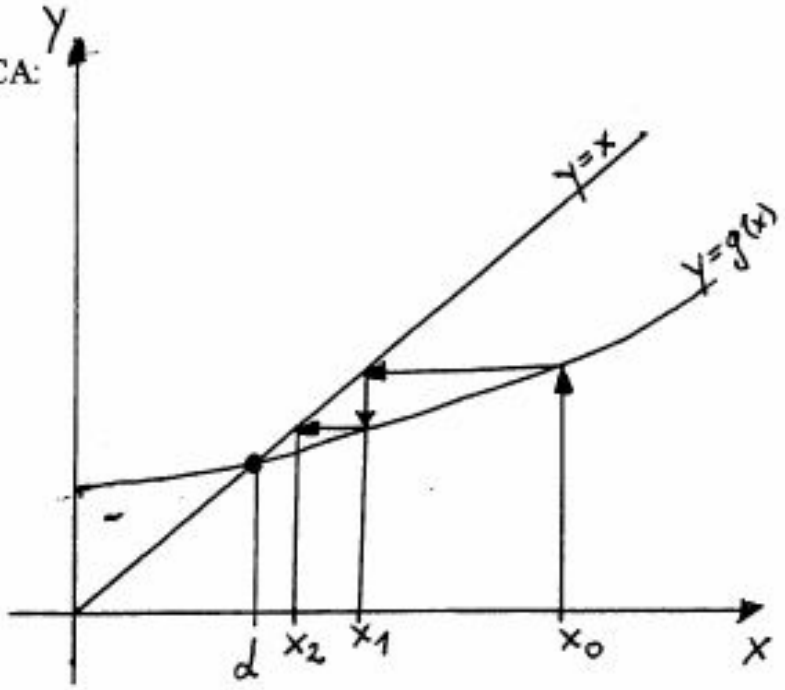
Če sta dva korena preblizu in h (korak) prevelik lahko pride do napake in s tem algoritmom ne najdemo obeh,ampak samo enega.

Metode delimo na odprte in zaprte
zaprte - približki korenov znotraj iskanega int.
odprte - približki se lahko oddaljijo od korena

3. ITERACIJSKA METODA

Navadno enačbo f(x)=0 prevedemo v obliko: x=g(x).Ce najdemotako vrednost korena α da je $\alpha=g(\alpha)$, je hkrati res tudi $f(\alpha)=0$.Enačba nam pove da je koren enačbe f(x)=0 tista vrednost $x=\alpha$ pri kateri se funkciji y=x in y=g(x) sekata.

SKICA:



ALGORITEM:

1. Postavimo indeks i=0,izberemo začetni približek x_0,dop. rel. nap. In predpisemo max št. iteracij.
2. Izračunamo nov približek:

   $$x_{i+1}=g(x_i)$$

3. Če je prekoračeno št. iteracij zaključimo,saj x_{i+1} ni primeren približek.
4. Če je $\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right|>\varepsilon$ povečamo indeks i in nadaljujemo,če pa je $<\varepsilon$ je x_{i+1} dober približek in zaključimo.

Ta metoda ima linearno konvergenco

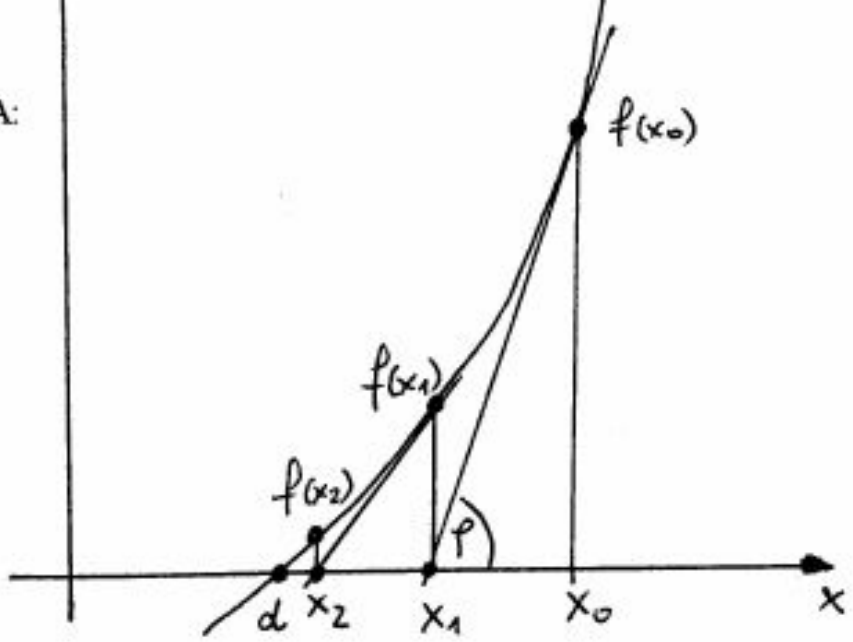
4. NEWTNOVA METODA

Newtnova metoda vedno konvergira če smo dovolj blizu korena.Slaba stran paje da moramo izrač. odvod.Pri tej metodi najprj izberemo začeti približek x_0 in nadomestimo nelinearno funkcijo f(x) z tangento v točki $(x_0,f(x_0))$,poiščemo kje tang. Seka x os in tam je naš novi približek.

$$f'(x_0)=\frac{f(x_0)}{x_0-x_1} \Rightarrow x_1=x_0-\frac{f(x_0)}{f'(x_0)}$$

Splošno: $$x_{i+1}=x_i-\frac{f(x_i)}{f'(x_i)}$$

SKICA:



ALGORITEM:

1. Izračunamo odvod f'(x)
2. Postavimo indeks i=0,izberemo začetni približek x_0,dop. rel. nap. ε in predpišemo max št. iteracij.
3. Izračunamo nov približek x_{i+1}
4. Če je $\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right|>\varepsilon$ povečamo i in nadaljujemoo,če ne je x_{i+1} dober približek.
5. Če je izvedeno predpisano št. iteracij x_{i+1} ni dober približek.

5.BISEKCIJSKA METODA

Je ena najpreprostejših metod za računanje korena enačbe f(x)=0.Najprej moramo poiskati tak interval (x_L,x_D) da funkc. v njem natanko enkrat menja predznak.Če je $f(x_L)f(x_D)<0$ je koren α v tem intervalu.Interval ki vsebuje koren razpolovimo in obdržimo tisti polovico int. V kateri imata krajišči nasprotni predznak.Interval manjšamo glede na to kako natančmo želimo določiti koren.

ALGORITEM:

1. Določimo meji x_L in x_D da je $f(x_L)f(x_D)<0$,dop. rel. nap. ε in max. Št. korakov .Postavimo i=1

2. Izračunamo razpolovišče $x_i = \frac{x_L + x_D}{2}$ in fnkc. Vrednosti $f(x_L)$ in $f(x_i)$ in če je $f(x_i)=0$ je x_i koren enačbe in zaključimo.

3. Če imata funkcijski vrednosti $f(x_L)$ in $f(x_i)$ nasprotni predznak ,$f(X_i)f(x_i)=0$, se koren nahaja v int. (x_L,x_i) v nasprotnem primeru pa v int. (x_i,x_D).Če je i=1 povečamo indeks i in računamo naprej.Prav tako računamo naprej če je:

$$\left|\frac{x_i - x_{i-1}}{x_i}\right| > \varepsilon$$

4. Če presežemo predpisano št. iteracij x_i ni dober približek in zaključimo.

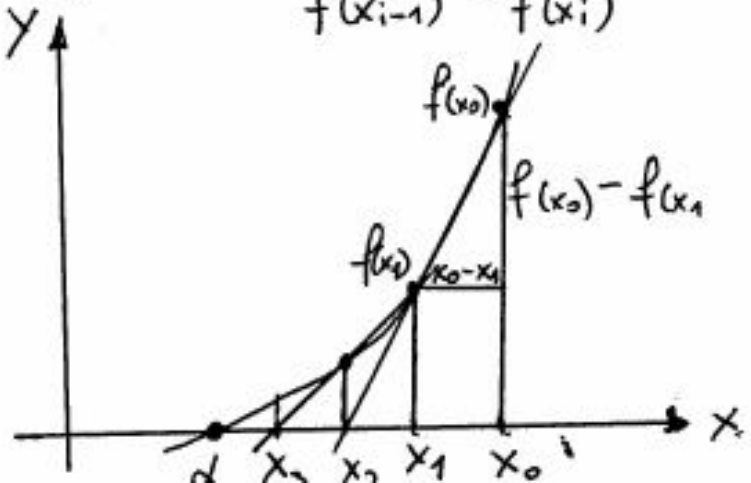
6.SEKANTNA METODA

Sekantno metodo uporaimo kadar je računanje odvoda dolgotrajno. Za določitev funkc. f(x) najprej izberemo približka x_0 in x_1. Nato nadomestimo nelin. funkc. z sekanto skozi točki $(x_0,f(x_0))$ in $(x_1,f(x_1))$ ter izračunamo kje seka x os. To je novi približek x_2.X2 pa lahko izračunamo po enačbi:

$$X_2 = X_1 - \frac{f(x_1)(x_0 - x_1)}{f(x_0) - f(x_1)}$$

SPLOŠNO: $$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$

SKICA:



ALGORITEM:

1. Postavimo indeks i =1,izberemo približka x_0,x_1 in dop. rel. nap. Ter predpisemo max. št. iteracij.

2. Izračunamo nov približek x_{i+1}.
3. Če je $\left| \frac{x_{i+1} - x_i}{x_{i+1}} \right| > \varepsilon$ povečamo indeks i nadaljujemo z rač. Če ne, je x_{i+1} dober približek.

4. Če izvedemo predpisano št. iteracij računanje prekinemo saj x_{i+1} ni dober približek.

8. GAUSSOVA ELININACIJSKA METODA

Imamo sist. treh enačb:

$a_{11}x_1+a_{12}x_2+a_{13}x_3=b_1$

$a_{21}x_1+a_{22}x_2+a_{23}x_3=b_2$

$a_{31}x_1+a_{32}x_2+a_{33}x_3=b_3$

pivot

a_{11} ne sme biti nič.Element s katerim delimo se im. pivot. Delimo a_{21}/a_{11} in pomnožimo rezultat z vsakim el.(elementom) a_{ii} v prvi vrstici in odštejemo dobljeno od druge vrstice,prav tako množimo tudi desno stran (b_i).Nato delimo še a_{31}/a_{11} in pomnožimo tretjo vrstico.V nadaljnem računanju je pivot a_{22}*.Zvezdica pomeni da je na novo izračunano:

$a_{22}^*=a_{22}-(a_{21}/a_{11})a_{12}$

$a_{11}x_1+a_{12}x_2+a_{13}x_3=b_1$

$a_{22}^*x_2+a_{23}^*x_3=b_2^*$

$a_{32}^*x_2+a_{33}^*x_3=b_3^*$

Nato delimo a_{32}^*/a_{22}^* in pomnožimo drugo, ter jo odštejemo od tretje. Dobimo :

$a_{33}^{**}x_3=b_3^{**}$

Iz tega izračunamo x_3 in se z izračunano vrednostjo vrnemo po postopku nazaj ,ter izračunamo še x_2 in x_1.

Kadar se sist. en. Ločijo le po desnih straneh (b),jih lahko združimo v **en** sam sist. Desne strani računamo hkrati z ostalim sistemom,v vsakem koraku sproti..

Pri tej metodi večkrat delimo. Če je kateri od koef.(koeficijentov) enak nič, deljenje ni možno.Ni priporočeno tudi,če je delitelj premajhen in se zaradi tega veča zaokrožitvena napaka.Tudi koef, na diagonali ne smejo biti enaki 0, takrat Gaussov algoritem odpove. Te težave lahko rešimo s pivotiranjem (menjava vrstic). Za pivot ponavadi vzamemo največji koef. V prvem stolpcu.

Normiranje pa pomeni iskanje in urejanje pivotov po velikosti (po absolutni vrednosti). Pravi pivot smo zbrali ko so kef. V sist. približno enaki.

Normiranje in pivotiranje skupaj zavirata rast zaokrožitvenih napak.

ALGORITEM:

1. Tvorimo razširjeno matriko $\bar{A}$, ki je sestavljena iz koef. sisit. enačb in desne strani. Ima n vrstic in n+1 stolpcev.
2. Če je potrebno matriko normiramo. Postavimo k=1
3. Poiščemo v k-tem stolpcu med elementi ,a_{ik} ; i=k,k+1,...n,največjega in zamenjamo vrstici da postane pivot.
4. Izvršimo eliminiranje tako da so pod pivotom ničle. Ničel ne računamo, moramo pa nato vse vrstice od k+1 dalje. Na novo izračunamo el. Vrstic po enačbi:

$$a^*_{ij} = a_{ij} - \frac{a_{ik}}{a_{kk}} \quad (i = k+1, \cdots, n\,;\; j = k+1, \cdots, n+1)$$

*-na novo izračunani el.(elementi)

5. Če je k=n-1 gremo na naslednji korak,sicer povečamo k in se vrnemo na tretji korak.
6. Dobili smo zg. trikotno matriko in iz nje izračunamo neznanke. Najprej

$$x_n = \frac{a^*_{n,n+1}}{a^*_{nn}}$$

nato

$$x_k = \frac{a^*_{k,n+1} - \sum_{j=k+1}^{n} a^*_{kj}\, x_j}{a^*_{kk}}$$

9.DETERMINANTA MATRIKE

Zg. trikotna determinanta:

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{vmatrix}$$

Trikotna determinanta se izraža takole:

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

Determinanto ki ni trikotna prevedemo v trikotno. Det.(determinanta) ima določene lastnosti:

- ne spremeni svoje vrednosti če katerikoli vrstici prištejemo s poljubnim št. pomnoženo drugo vrstico (gauss). Ko to naredimo dobimo zg. trikotno det. in jo izračunamo po zg. en.
- če pa mora biti det. izračunana natančno,uporabimo normiranje in pivotiranje.

Obstajata pa pri tem dve pravili:

- če množimo vrstico det. s konst.(konstanto),množimo s to konst. celo det.
- če zamenjamo dve vrstici det., zamenja ta det. predznak

10.RAČUNANJE INVERZNE MATRIKE

Imamo kvadratno mat. A reda nxn. Če je matrika singularna obstaja takšna mat. B da je AB=BA=I. B običajno pišemo A^{-1} in im.inverzna matrika. Računanje le te pa se lahko glede na njeno def.: AB=I pretvori v sist. n^2 en . z n^2 neznankami. To so el. mat. B. Sist. en. pa lahko rešimo po gaussovi metodi.

11.GAUSS-JORDANOVA METODA

Je varianta Gaussove metode.G-J metoda se razlikuje le po tem,da eliminira koef. **nad** in **pod** glavno diagonalo (Gauss samo spodaj).Razširjeno matriko koef. prevedemo v enako diagonalno matriko z samimi 1 na diagonali, vektor desne strani pa postane vektor rešitve (v desnem stolpcu nam ostanejo rešitve).

ALGORITEM:

1. Tvorimo razširjeno matriko $\overline{A}$ ki je sestavljena iz sist. en. In desne strani. Ima n vrstic in n+1 stolpcev.
2. Po potrebi matriko normiramo. Postavimo k=1.
3. Delimo k-to vrstico z a_{kk} in tako dosežemo da je a_{kk} enak 1.
4. Eliminiramo tako da so ničle pod in nad el. a_{kk}. Na novo izračunamo v vsaki vrstici koef. po enačbi:

$$a_{ij}^{*}=a_{ij} - a_{ik}$$

5. Če je k=n gremo na naslednji korak,če ne povečamo k in gremo na tretji korak.
6. Dobili smo enotsko matriko in stolpec z rešitvami.

12.METODA CHOLESKEGA

Matriki A lahko priredimo transponirano matriko A^T. Ti matriki sta povezani z zvezo:

$$(a^T)_{ji}=a_{ij} \quad i,j=1,2,...n$$

Simetrična pa je tista [illegible] matrika za katero velja $A^T=A$,torej:

$$a_{ji}=a_{ij} \qquad i,j=1,2,....n$$

Simetrična mat. je za računanje ugodna, saj so koef. shranjeni v trikotni mat.,za katero je treba manj računanja kot za nesimetrično. Simetrična se da razcepiti na dve trikotni matriki:

$$A=U^TU$$

V mat. A je pomembnih samo 6 el..,ostali trije so določeni s simetrijo. Sist. en. lahko nastavimo tako da združimo koef. na levi in desni. Ko rešimo sist. en. dobimo:

$$u_{11}=\sqrt{a_{11}} \qquad u_{1i}=\frac{a_{1i}}{u_{11}} \qquad u_{22}=\sqrt{a_{22}-u_{12}^2}$$

$$u_{23}=\frac{a_{23}-u_{13}u_{12}}{u_{22}} \qquad u_{33}=\sqrt{a_{33}-u_{13}^2-u_{23}^2}$$

Ti koreni imajo lahko neg. ali poz. predznak. Razcep po Choleskem da trikotno mat. z real. koef. samo za nekatere kvadratne simetrične mat.

ALGORITEM:

1. Koef. prve vrstice mat. U izračunamo po formulah:

$$u_{11}=\sqrt{a_{11}} \qquad u_{1i}=\frac{a_{1i}}{u_{11}} \qquad i=2,3,\dots n$$

Ostale vrstice pa:

$$u_{ii}=\sqrt{a_{ii}-\sum_{k=1}^{i-1}u_{ki}^2}$$

$$u_{ij}=\frac{a_{ij}-\sum_{k=1}^{i-1}u_{ki}u_{kj}}{u_{ii}} \qquad j=i+1,i+2,\dots n$$

2. Izračunamo pomožni vektor b' iz matrične enačbe $U^Tb'=b$ in sicer po formulah:

$$b_1'=\frac{b_1}{u_{11}} \qquad b_i'=\frac{b_i-\sum_{k=1}^{i-1}u_{ik}b_k'}{u_{ii}}$$

Kjer je uik el. mat. U.

3. Izračunamo vektor x iz matrične en. Ux=b' po formulah:

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

13.JACOBIJEVA METODA

Sistem linearnih enačb(sist. lin. en.)zapišemo v obliki:

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad (i = 1, 2, \ldots n)$$

Ta pa je primeren za izpeljavo te metode.

Iz vsake en. v sist. izračunamo komp.(komponente) vektorje rešitve,ki je po diagonalnem el. To je xi in dobimo:

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} x_j \right) \quad (i = 1, .. n)$$

ter izberemo začetni približek $x^{(0)}$. Št. zg. indeksa v oklepaju nam pove zaporedno iteracijo,z nič pa je označen vektor začetnih približkov. Komp. Vektorja $x^{(0)}$ vstavimo v prejšnjo en. in dobimo nov približek $x_i^{(1)}$. To ponavljamo dokler ne dosežemo predpisane natančnosti .

Na k-tem koraku približek izračunamo:

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

Če označimo razliko med desno stranjo in **približno** levo stranjo i-te enačbe,na k-tem koraku dobimo :

$$R_i^k = b_i - \sum_{j=1}^{n} a_{ij} x_j^k \qquad (i = 1, \ldots n)$$

Tako lahko prejšnje en. (če na desni prištejemo in odštejemo x_i^k) zapišemo:

$$x_i^{k+1} = x_i^k + \frac{R_i^k}{a_{ii}} \qquad (i = 1, \ldots n)$$

Vse komp. x_i^{k+1} novega približka so odvisne samo od vektorja približkov v predhodnem koraku iteracije $x^{(k)}$. Vrstni red približkov ni pomemben.

14.GAUSS-SEIDELOVA METODA

Je podobna Jacobijevi metodi (novi približki iz prejšnjih približkov), le da tu novo komponento rešitve takoj uporabimo za računanje novih komp. približkov. Konvergenca je zagotovljena ,če je v vsaki vrstici v mat.koeficijentov absolutna vrednost diagonalnega el.,večja od vsote abs. vred. izvendiag. elementov.

FORMULE:

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^{k} \right) \qquad i = 1, 2 \ldots n$$

V prvi sumaciji nastopajo približki,ki so bili izračunani v tekoči iteraciji. V tem se ločijo Jacobijeve formule od teh . Prav tako lahko zapišemo:

$$x_i^{k+1} = x_i^{k} + \frac{R_i^{k}}{a_{ii}}$$

Kjer je:

$$R_i^{k} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i}^{n} a_{ij} x_j^{k}$$

15.NAVADNA ITERACIJSKA METODA ZA SIST. NELINEANIH ENAČB

Rešujemo sistem enačb:

$$f_1(x,y)=0$$

$$f_2(x,y)=0$$

ki ga pri tej metodi prevedemo na sist. en.:

$$x=g_1(x,y)$$

$$y=g_2(x,y)$$

Najti moramo tak par št. (α,β),da sta prevedena sist. en. izpolnjena in s tem sta izpolnjena tudi en. začetnih funkc.

ALGORITEM:

1. Postavimo indeks i=0,izberemo začetna približka (x_0,y_0),dop. rel nap.,določimo max št. iteracij.
2. Izrač. nov približek (x_{i+1},y_{i+1}) po formulah $x_{i+1}=g_1(x_i,y_i)$ in $y_{i+1}=g_2(x_i,y_i)$.
3. 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 nadaljujemo,sicer je par št.(x_i,y_i) dober približek in zaključimo.
4. Če izvedemo predpisano št. iteracij,par št. (x_{i+1},y_{i+1}) ni dober približek in končamo.

Zaporedje približkov se približuje rešitvi sist.,če je blizu rešitve:

$$\left|\frac{\partial g_1}{\partial x}\right|+\left|\frac{\partial g_1}{\partial y}\right|<1$$
$$\left|\frac{\partial g_2}{\partial x}\right|+\left|\frac{\partial g_2}{\partial y}\right|<1$$

16.NEWTNOVA METODA ZA <u>SISTEME ENAČB</u> (NELINEARNIH)

Je najbolj znana metoda za reševanje sist. lin. en. Tako kot pri vsaki iteracijski metodi, najprej ocenimo začetni približek (x_0,y_0) rešitve sist. en.,$f_1(x,y)=0$ $f_2(x,y)=0$. Boljši približek izračunamo z računanjem popravka predhodnega pribl.:

$$x_{i+1}=x_i+\Delta x_i$$

$$y_{i+1}=y_i+\Delta y_i$$

Popravka (Δx_i,Δy_i) pa izračunamo iz sist. en.:

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

ALGORITEM:

1. Izračunamo parcijalne odvode: $\frac{\partial f_1}{\partial x}, \frac{\partial f_1}{\partial y}, \frac{\partial f_2}{\partial x}, \frac{\partial f_2}{\partial y}$
2. Postavimo indeks i=0,izberemo začetna približka (x_0,y_0),dop. rel. nap. In predpišemo št. iteracij.
3. Pri določenem približku izračunamo vrednost parcijalnih odvodov in obeh funkcij. Zapišemo sist. en. (str. 13) in dobimo (Δx_i,Δy_i). Ter izrač. nov približek (x_{i+1},y_{i+1}). Če je i=0povečamo indeks i in računanje ponovimo.
4. 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 računanje ponovimo,če ne je (x_{i+1},y_{i+1}) dober približek.
5. Če pa je izvedeno predpisano št. iteracij računanje prekinemo.

Ocena napake pa je običajno pesimistična!

23.PRINCIP NAJMANJŠIH KVADRATOV

Z njim računamo presek Bezierove krivulje s premico.Iščemo najmanjši kvadrat v katerem je presečišče.

Začnemo tako da vzamemo najvišjo Bezierovo točko na desni in najnižjo na levi. Ti točki med seboj tvorita diagonalo najmanjšega pravokotnika, ki zajema vse Bezierove točke neke krivulje. Če premica ne seka tega pravokotnika, ne seka tudi krivulje. Po de Casteljauovem algoritmu (Interpolacija str.100) razdelimo krivuljo na dva dela (Lin D). S tem dobimo dva nova Bezierova poligona. To delitev lahko nadaljujemo in po k korakih dobimo 2^k poligonov. Vsak od teh pa vsebuje majhen del prvotne krivulje in zato za ta del določa minimalni pravokotnik. Od dveh pravokotnikov izberemo tistega, ki ga seka premica (če seka določimo enako kot v bisekcijski metodi) in tistega ponovno razcepimo. Delitev lahko nadaljujemo poljubno korakov,ki pa so odvisni od tega, kako natančno želimo izračunati presečišče.

* - 100

25.APROKSIMACIJA/INTERPOLACIJA(!?!)

Metode aproksimacije iščejo najmanjši interval,oz. približne vrednosti,neke neznanke. Medtem ko metode int. s pomočjo polinomov iščejo idealno linijo skozi dane točke (rezultati meritev,...).

Utemeljitev:

Vse aproksimacijske metode (gauss,bis. met.,sek. met.,...) iščejo neko št. ali koren ki nam ni znan,ali pa iščemo presek premice s krivuljo. Prav tako pa rešujemo sist. en.

Pri int. met. (Newtnov int. pol.,la grange,aitken, inv. int.) pa imamo tabelo danih točk y=f(x). Iz te tabele poskušamo najti polinom ,ali polinome (zlepki), določene stopnje ,ki najbolj popišejo dane točke.

26.LAGRANGEOV INTERPOLACIJSKI POLINOM

Idejo Lagrangeove int. formule najlažje pokažemo z int. skozi dve točki $(x_0,y_0),(x_1,y_1)$. Skozi njiju lahko potegnemo premico (polinom 1. stopnje):

$$P_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}$$

Formulo im. **linearna interpolacija** (lin. int). Skozi tri točke pa lahko potegnemo kvadratno funkcijo (kvadratna int.). Skozi n+1 točk lahko potegnemo polinom n-te stopnje.

Skozi n+1 točk lahko potegnemo en sam polinom,ki pa je enak tistemu ,ki smo ga izračunali z sist. enačb.

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

Zgornja formula se imenuje LIP[1], $L_k^{(n)}(x)$ pa je polinom stopnje n in se im. k-ti Lagrangeov pol.

[1] Lagrangeov int. pol.

27.NEWTNOV INTERPOLACIJKI POLINOM(NIP)

L. int. met. zahteva veliko računanja in če bi hoteli izračunati int. pol. še na eni točki, bi morali začeti znova, saj si z vmesnimi rezultati ne moremo pomagati.

Zapišemo polinom n-te stopnje:

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots$$
$$\dots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1})$$

Nato določimo koef. ai da bo v vseh točkah $P_n(x_i)=y_i$. Ko je pogoj izpolnjen nadaljne koef: računamo po rekruzijski formuli:

$$f_{ij} = \frac{f_{i+1,j-1} - f_{ij-1}}{x_{i+j} - x_i} \qquad j = 0,1,\dots,m-j \qquad f_{0,j} = a_j$$

In jih nato razvrstimo v računsko shemo.(i-stolpec i-vrstica)

PREDNOSTI:

- Preprosto lahko izrač. polinom skozi dodatno točko (dodamo v vsak stolpec še en el.,ter izračunamo koef. $f_{0,m+1}$).
- Je tudi najboljša metoda za hkratno računanje več vrednosti int. pol.

28.AITKENOVA INTERPOLACIJSKA METODA(AIM)

Po AIM računamo neposredno vrednost int. pol. in na vsakem koraku višjega za eno stopnjo. Računanje prekinemo,ko se zaporedni int. vrednosti ne ločita za manj kot je dop. nap. Označimo $P_{i,0}=y_i$ i=1,2,...n. El. j-tega stolpca pa računamo po formuli: $(j = 1,2,\dots m)$

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

Računanje pa organiziramo v računsko shemo.

29.INVERZNA INTERPOLACIJA (INV. INT.)

Interpolacija je proces s katerim določamo vrednost odvisne spremenljivke y=f(x),če je dana vrednost x in tabela funkcije f(x).Inv. int. pa je postopek s katerim določamo vrednost neodvisne spr. x, ki ustreza dani vrednosti odv. spr. y ,če je funkcijska odvisnost dana s tabelo y=f(x). Torej računamo vrednost inv. funkc. x=x(y).

Lahko jo rešujemo na dva načina:

- podamo tabelo funkc. x=x(y). To pomeni ,da tabeli funkc. y=f(x) zamenjamo stolpca in dobimo tabelo funkc. x=x(y).
- rešujemo enačbo f(x)=y ,kjer je odvisnost y=f(x) dana s tabelo in je y konstanta. Rešitev enačbe f(x)-y=0 računamo iterativno,f(x) pa nadomestimo z int. pol.

30.ZLEPKI

Ponavadi skozi int. točke položimo en sam polinom. Lahko pa se zgodi da je pol. visoke stopnje in zaradi tega oscilira, to pa pripelje int. s pol. do velikih napak. Namesto da določimo en sam polinom za celo tabelo, razdelimo tabelo na več podtabel. Za vsako od teh podtabel pa izračunamo pol. nižje stopnje in jih povežemo med seboj. Dobimo int. funkc. za celotno tabelo.

Težava pa je gladkost stikov med int. pol., zato določimo:

1. Posamezni int. pol. (označimo $S_i(x)$) je definiran samo na intervalu $[x_i, x_{i+1}]$ ($i=0,1,2,...n-1$).
2. Sosednja polinoma $S_i(x)$ in $S_{i+1}(x)$ naj se stikata čim bolj gladko.

Posledica druge zahteve je,da sosednja polinoma zgledata kot zlepljena (zlepki; ang.:spline,S(x)). Zlepke ločimo po stopnji pol. in gladkosti stikov. Pri int. pa zahtevamo da so argumenti urejeni strogo po naraščajočem vrstnem redu.

KUBIČNI ZLEPKI:

1. S(x) je kubični pol. $S_i(x)$ na vsakem podintervalu $[x_i, x_{i+1}]$ $i=0,1,...n-1$.
2. S(x) je int. funkc.,to pomeni da mora biti $S(x_i)=y_i$ $i=0,1,...n$
3. S(x) je zvezna funkc. z zveznim 1. In 2. odvodom.

31.METODA REGULA FALSI

Najprej poiščemo tak interval (x_L,x_D), da funkc. f(x) natanko enkrat menja predznak. Koren α mora biti v intervalu (x_L,x_D). Nato nelin. funkc. f(x) na tem intervalu zamenjamo s sekanto skozi točki (x_L,f(x_L)) in (x_D,f(x_D)), ter ugotovimo kje seka x os. To pa je nov približek x_1. Nastaneta dva intervala (x_L ,x_1) in (x_1 , x_D). Izberemo tistega v katerem funkc. f(x) menja predznak.

Splošna formula za računanje približka:

$$X_i = X_L - \frac{f(x_L)(x_L - x_D)}{f(x_L) - f(x_D)}$$

ALGORITEM:

1. določimo meji int.(intervala)x_L in x_D kjer se nahaja koren (x_L <x_D), f(x_L)f(x_D)<0, določimo dop. rel. nap. in izberemo št iteracij. Postavimo i=1.
2. Izračunamo približek, po formuli, in funkcijsko vrednost f(x_i). Če je f(x_i)=0 je x_i koren. Iteracijo zaključimo.
3. Če sta f(x_L) in f(x_i) nasprotnega predznaka, f(x_L) f(x_i)<0, se koren nahaja v int. (x_L, x_i)in zamenjamo x_D= x_i in f(x_D)=f(x_i). V nasprotnem primeru zamenjamo x_L z x_i.
4. Če je $\left|\frac{X_i - X_{i-1}}{X_i}\right| > \varepsilon$, povečamo indeks i in nadaljujemo z računanjem. Če ne, je x_i dober približek. Zaključimo .
5. Ko izvedemo določeno št. iteracij domnevamo da x_i ni dober približek. Zaključimo.

32.CRAMERJEVA METODA

Imamo n lin. en.in n neznank :

$$Ax=b$$

Po Cramerjevem pravilu dobimo rešitev iz enačbe:

$$x_j = \frac{|A^j|}{|A|}$$

A^j –matrika reda nxn.

Dobimo jo tako da v matriki A zamenjamo j-ti stolpec z stolpcem na desni strani (b).

$|A|$- determinanta matrike

$|A^j|$- det. mat. A^j

b – vektor desnih strani

x – vektor neznank

33.SISTEMI LINEARNIH ENAČB (splošno)

Metode za reševanje sist.en. delimo na DIREKTNE in ITERATIVNE. Direktne so tiste, s katerimi pridemo, po natančno določenih algebrajičnih operacijah, do rešitve. Št. operacij pa je odvisno samo od velikosti sist. en. Pri iterativnih met. izbiramo začetni približek in nato izračunamo novega. Takšna metoda pa lahko konvergira (se približuje) ali pa ne.

1. DIREKTNE MET.

Znanih je več. Zelo znana je Cramerjeva metoda. Moramo izračunati veliko determinant in je zato zahtevna (veliko računanja) in neučinkovita. Veliko boljše so tiste met. ,ki uporabljajo eliminacijo(Gauss, G-J, LU razcep,...).

2. ELIMINACIJSKE MET.

Pri teh met. iz ene enačbe izrazimo neko neznanko (x_1) in dobljeno enačbo vstavljamo v ostale enačbe. Postopek ponavljamo dokler nam ne ostane ena enačba z eno neznanko.

Kaj lahko:

- vsako en., lahko množimo s konstanto
- vrstni red en. lahko spreminjamo
- vsaki en. smemo prišteti/odšteti, s konstanto pomnoženo drugo en.

Te operacije lahko spremenijo lastnosti, ne spremenijo pa rešitve.

S pravili lahko:

- s 1. Pravilom lahko normiramo (vsi koef. manjši ali enaki 1). Ugodno pa vpliva natančnost rešitve.
- na 3. pravilu je grajen Gaussov postopek,

če pa pri tem slučajno delimo z nič:

- nam 2. pravilo omogoča zamenjavo en.

34.SISTEM 2 LINEARNIH ENAČB

Imamo dve enačbi in dve neznanki:

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

Možnosti:

1. Sist. en. ima eno rešitev (določen sist.)
2. Sist. en. ima dve rešitvi (nezdružljiv ali nekonstinenten)
3. Sist. en. ima neskončno rešitev.
4. Sist. en. ima trivialno rešitev.

35.LU RAZCEP

Produkt dveh mat. A=BC lahko zapišemo na veliko načinov. Če je B, sp. Trik. Mat. (L) in C zg. trik. mat. (U) dobimo poseben primer razcepa: A=LU.

Mat. koef. razcepimo zato, da hitreje rešimo sist. en. Razcep na zg. in sp. trik. mat. pa je tudi bolj ekonomičen, če rešujemo sist. z veliko desnih strani.

Matriko razcepimo samo enkrat in rešimo dva trik. sist. en.:

$$Ax=b \Rightarrow LUx=b$$

Vzamemo za novo neznanko $Ux=b'$ in zapišemo: $Lb'=b$. Ta sist. pa je sp. trikoten. Ko izračunamo komponente pomožnega vektorja b' (b_1',b_2',b_3'). Nato iz sist. en. $Ux=b'$ izračunamo neznanke x (vektor rešitve).

Z GAUSSOM:

Imamo mat. A in vektor desnih strani. U (A=LU) izračunamo po Gaussu (a_{22}^*, a_{23}^*, a_{33}^{**}-enako kot pri Gaussu leva stran), L pa sestavimo iz faktorjev (f_{21}, f_{23} ,f_{32}^* - postavimo levo spodaj v mat.), ter po diagonali postavimo enke. Zaradi teh enk dobimo še tri potrebne enačbe. S tem je sist. enolično določen. Če L in U zmnožimo, da res dobimo mat. A.

Algoritem:

1. Izvedemo Gaussov postopek in dobimo mat. U in L.
2. Izračunamo pomožni vektor b' po formulah:

$$b_1'=b_1 \qquad b_i'=b_i-\sum_{k=1}^{i-1} L_{ik}\, b_k' \qquad (i=2,3,\ldots n)$$

kjer so L_{ik} elementi mat. L.

3. Izračunamo vektor neznank x po formulah:

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

Kjer so u_{ik} elementi mat. U.

LU razcep je bolj zapleten če med eliminacijo pivotiramo in normiramo.