

TMV035, Analys och linjär algebra, del A, 2007/08

Programmering och Matlab

Laboration 3

Laboration 3 redovisas på studio 7.1 På studio 7.2 kommer vi att ha sista redovisnings tillfälle i grupper för alla laborationer. De som inte hinner vid detta tillfälle måste redovisa senare till Alexei.

Ni får inte poäng för kursen utan att redovisa laborationer.

Allmän uppgift är: att göra om programmet från Laboration 2 för att lösa ekvationer av typ $f(x)=0$ med hjälp av Newtons metod. Det finns beskrivning av Newtons metod i Adams på sid. 246, en länk med en detaljerad text om Newtons metod och lämpliga exempel av Jan på kursens hemsida. Vi hade också en föreläsning med genomgång av Newtons metod och programmerings detaljer i laborationen. Ni kan utnyttja för Laboration 3 mest av texter ni skrivit för Laboration 2 och bara ändra dem lite i den del (cirka 4 rader!) som skiljer Newtons metod och metoden med fixpunktiterationer.

- i) En ny detalj i programmering blir att $f(x)$ måste innehålla en global parameter a .
- ii) En annan möjlighet är att definiera era exempel-funktioner som funktioner av två variabler t.ex. $f(x, a)$ inklusive parametern a . För att anropa dem som funktion av bara en variabel x i huvudprogrammet, som är lämpligt för Newtons program, får konstruktionen med anonymous funktion användas. Läs beskrivningen av den metoden i detaljerad beskrivning under.

Välja själv vilken variant av funktion med parameter du vill använda.

Detaljerad uppgift:

1. Skriv en funktion i Matlab med namn till exempel NewtonMin som löser en godtycklig ekvation $f(x)=0$ med hjälp av Newtons metod. Det enkelt att göra om texten av funktionen för fixpunktiterationen för detta.
Skillnaden blir att nya funktionen NewtonMin behöver ett funktions argument f_{prim} till för derivatan av funktionen f : `NewtonMin(f, fprim, x0, tol, Nmax)`.
Man måste också ändra den viktiga raden där står beräkningen av nästa approximation från gamla approximationen av lösningen.
2. Skriv ett huvudprogram som använder er funktion för Newtons metod och löser ekvationen $f(x)=0$ för en godtycklig funktion f .
3. Funktionen $f(x)$ måste innehålla en parameter. Till exempel parametern a i funktioner $f(x) = 0.5 + \sin(ax)$ eller $f(x) = 0.5 - \exp(-ax^2)$. Den parametern måste beskrivas som global parameter i huvudprogrammet och få ett värde där. Ni kan igen utnyttja ert huvudprogram från laboration 2 och bara göra om den lite.

Fortsättningen på nästa sida !

4. Hitta ett lämpligt eget exempel av funktionen $f(x)$ i ekvationen (**inte samma som i boken och inte samma som gruppen bredvid !**) och skriv två funktions filer: en för funktionen $f(x)$ och en för dess derivata $f'(x)$. Funktionen $f(x)$ måste ha en parameter a som kan varieras och kan definieras som global variabel i huvudprogrammet och på samma sätt i funktionsfiler för $f(x)$ och för $f'(x)$. Ni måste förstås kunna skriva ner en formel för $f'(x)$. I exempel ovan $f'(x) = a \cos(ax)$ och $f'(x) = -2ax \exp(-ax^2)$.
5. Alternativt, definiera din exempel funktion som funktion av två variabler t.ex. $f(x, a)$ inklusive parametern a (samma för derivatan). Att anropa den som funktion av bara en variabel x i huvudprogrammet, är lämpligt för Newtons program. Det kan göras med hjälp av anonymous funktions konstruktion.
 Du kan skapa i ditt huvudprogram en handle (referens till funktion) med bara en variabel x av din tvåvariabels funktion $f(x, a)$ med följande kommandot: `MyAnonymFunk=@(x)`
 $f(x, a)$. Konstanten a måste vara definierad i huvud programmet före detta kommando, x här är bara en formel symbol.
`MyAnonymFunk` kan senare användas (utan `@` !) när du anropar den funktion för Newtons metod och också direkt `y=MyAnonymFunk(x)`, när du ritar graf till $f(x, a)$. Det görs för att undvika skriva flera onödiga argument. Samma typ av handle måste skapas för derivatan av f .
Välj själv vilken variant av funktion med parameter du vill använda: parametern som global variabel, eller anonymous funktioner.
6. Skriv funktionen för Newtons metod (t.ex. `MyNewton`) och huvudprogram så att beräknings processen illustreras på samma sätt som i Adams på sidan 246.
 Huvudprogrammet skall rita en graf till er funktion $y=f(x)$. Grafen måste ha en skärnings punkt med x -axeln för att kunna lösa ekvationen $f(x)=0$. Initiala approximationen x_0 kommer att väljas med hjälp av kommandot `ginput` exakt på samma sätt som i `Laborration2`
 Funktionen `MyNewton(f, fprim, x0, tol, Nmax)` måste rita en bruten linje mellan konsekventa iterationer som på Fig. 4.51 på sid. 246 i Adams. Den är lite annorlunda än spiraler i fixpunktiterationer och är också lätta att programmera med `plot` kommandot.
7. Analysera hur många iterationer behövs för toleransen 10^{-6} i fall de konvergerar.