

DATORÖVNING 4 — FIXPUNKTSITERATION

Allmänt. Dokumentera ditt arbete i ett pdf-dokument. Spara detta så att du kan läsa på inför duggor och tentamen. Detta ger träning i att skriva matematik. Det är viktigt i sig. Men du ska se att skrivandet också hjälper dig att lära matematik. Detta är ännu viktigare. När man arbetar med att förbättra resonemanget och texten ökar man samtidigt förståelsen.

Datorövningarna examineras endast vid duggor (2–3) samt vid tentamen. Detta sker genom ett flertal frågor av typen: “Hur gjorde du det och det i MATLAB i Datorövning 3?” “Skissa den graf som du gjorde då.” “Vad är syftet med denna beräkning?”

Datorövningarna är alltså inga vanliga “laborationer” som man gör och redovisar i datorsal. Det är material som du jobbar med, dokumenterar och lär in, på Chalmers och hemma. Samarbete uppmuntras, men detta är inget grupparbete. Varje student måste göra sina egna datorprogram och sina egna dokument.

Spara alla filer och textdokument i en separat filkatalog för varje övning. Om ni sitter två vid samma dator under den schemalagda datorövningstiden, tänk på att ni båda sparar filerna i er egen filkatalog så att ni kan läsa på inför duggor och tentamen, där material från datorövningarna kommer att utgöra en väsentlig del.

Mål.

Matematik: Fixpunktsekvation $x = g(x)$. Fixpunktsiteration.

Programmering: Funktion. Iteration.

Litteratur. [BM kap 5](#), Adams 4.2 (4.6, 6e uppl)

Instruktioner. Skapa en ny filkatalog (“directory”) **studio-4** för denna övning.

Skriv funktionsfiler (m-filer) för de nämnda funktionerna. Dokumentera allt ditt arbete i ett ordbehandlingsdokument. Exportera slutligen dokumentet till pdf-format. (Både MS Word och OpenOffice Writer stödjer detta.) Slutprodukten skall alltså vara pdf.

Uppgifter.

1. Fixpunktsalgoritmen är

- (a) Givet: en kontraktion $g : I \rightarrow I$, en startpunkt $x_0 \in I$ och en tolerans TOL.
- (b) Start: $i = 1$, $x_i = x_0$, $x_{i+1} = g(x_i)$.
- (c) Om $|x_{i+1} - x_i| \leq \text{TOL}$, stoppa. Annars, $i = i + 1$, $x_{i+1} = g(x_i)$, upprepa (c).

Skriv ett program som implementerar algoritmen enligt programskalet [fixpunkt1.m](#). Kopiera filen genom att klicka på denna länk eller gå till “Matlab/” på kurshemsidan (se under Datorövningar) och ladda ned den därifrån. Använd **while**-loop.

2. Använd programmet på följande fixpunktsekvationer.

- (1) $x = x/2 + 1/x$
- (2) $x = 2/x$
- (3) $\cos(x) = x$

Du måste själv hitta en startpunkt.

3. Observera hur de beräknade följderna konvergerar eller divergerar. Hur många steg behövs för att felet ska bli mindre än 10^{-10} ?

4. Det är onödigt att spara hela följderna x_i . Istället för i och x_i räcker det att använda två variabler x och x_1 :

- (a) Givet: en kontraktion $g : I \rightarrow I$, en startpunkt $x_0 \in I$ och en tolerans TOL.
- (b) Start: $x_1 = x_0$, $x = g(x_1)$.
- (c) Om $|x_1 - x| \leq \text{TOL}$, stoppa. Annars, $x_1 = x$, $x = g(x_1)$, upprepa (c).

Skriv ett program som implementerar denna algoritm enligt programskalet [fixpunkt.m](#).

5. System av ekvationer. En viktig fördel med fixpunktsiterationen jämfört med bisektionsalgoritmen är att den fungerar även för system av ekvationer. Det gör inte bisektion. Nackdelen är att det ofta är svårt att skriva ekvationssystemet på fixpunktsform så att iterationen konvergerar.

Prova nu ditt program `fixpunkt.m` på följande ekvationssystem:

$$5x_1 + 2x_2 = -2$$

$$x_1 + 6x_2 = 1$$

Skriv om på fixpunktsform genom att lösa ut x_1, x_2 ur diagonaltermerna:

$$x_1 = (-2 - 2x_2)/5$$

$$x_2 = (1 - x_1)/6$$

Detta blir bra (liten Lipschitz-konstant, $L < 1$) eftersom diagonaltermerna är stora. Skriv sedan en funktionsfil `gfunk1.m`

```
function y=gfunk1(x)
    y(1,1)=(-2-2*x(2))/5;
    y(2,1)=(1-x(1))/6;
```

Kör sedan `fixpunkt` med en kolonnvektor som startpunkt:

```
>> x=fixpunkt(@gfunk1,[0;0],1e-7)
```

Lösningen är $(-0.5, 0.25)$.

/stig