TMV225/TMV176 Inledande matematik M+TD

DATORÖVNING 3 — BISEKTIONSALGORITMEN

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: bisektionsalgoritmen, konvergent följd.
Programmering: `while`-loop, `if`-sats, funktion som argument till funktion.

Litteratur. Adams 1.4, BM kap 4

Instruktioner. Skapa en ny filkatalog ("directory") `studio-3` 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.

Introduktion. Bisektionsalgoritmen är en metod för att beräkna lösningar till ekvationer på formen $f(x) = 0$. Idén är att utgå från ett intervall $[a, b]$ sådant att $f(a)$ och $f(b)$ har olika tecken.

OBS! Att $f(a)$ och $f(b)$ har olika tecken innebär att produkten $f(a)f(b) < 0$. Detta villkor kommer vi att utnyttja i våra MATLAB-program.

Vi studerar det konkreta exemplet $f(x) = x^2 - 2$, och utgår från intervallet $[a, b] = [1, 2]$.

Vi söker alltså $\bar{x}$ sådant att $f(\bar{x}) = \bar{x}^2 - 2 = 0$, och eftersom vi startar med intervallet $[1, 2]$ är lösningen $\bar{x} = \sqrt{2} = 1.414213562...$ Hade vi i stället velat beräkna den negativa roten kunde vi startat med intervallet $[-2, -1]$. Med bisektionsalgoritmen kan vi beräkna en lösning i taget.

Notera att $f(a) = f(1) = 1^2 - 2 = -1 < 0$ och $f(b) = f(2) = 2^2 - 2 = 2 > 0$ har olika tecken. Vi drar slutsatsen att $\bar{x} \in [1, 2]$.

Nästa steg är att beräkna mittpunkten $\frac{a+b}{2} = \frac{1+2}{2} = 1.5$ på intervallet $[a, b]$.

Nu undersöker vi vad f har för tecken i mittpunkten: $f(1.5) = 1.5^2 - 2 = 0.25 > 0$.

Men detta innebär att $f(a) = f(1) < 0$ och $f(1.5) > 0$ har olika tecken. Vi drar slutsatsen att lösningen $\bar{x}$ ligger i vänster halva av intervallet $[a, b]$: $\bar{x} \in [1, 1.5]$.

Vi definierar därför om b (höger ändpunkt) till $b = 1.5$. (Vi tar alltså mittpunkten som det nya värdet på b.)

Vi har nu funnit ett intervall $[a, b] = [1, 1.5]$ (hälften så långt som vårt ursprungliga intervall) som innehåller $\bar{x}$.

Vi upprepar nu allting:

Beräkna mittpunkten $\frac{a+b}{2} = \frac{1+1.5}{2} = 1.25$ på intervallet $[a, b]$.

Nu undersöker vi vad f har för tecken i mittpunkten: $f(1.25) = 1.25^2 - 2 = -0.4375 < 0$.

Date: 2012–09–20, Stig Larsson, Matematiska vetenskaper, Chalmers tekniska högskola (mindre uppdateringar av Niklas Ericsson).

Men detta innebär att $f(1.25) < 0$ och $f(b) = f(1.5) > 0$ har olika tecken. Vi drar slutsatsen att lösningen $\bar{x}$ ligger i höger halva av intervallet $[a, b]$: $\bar{x} \in [1.25, 1.5]$.

Vi definierar därför om a (vänster ändpunkt) till $a = 1.25$. (Vi tar alltså mittpunkten som det nya värdet på a.)

Vi har nu funnit ett intervall $[a, b] = [1.25, 1.5]$ (en fjärdedel så långt som vårt ursprungliga intervall) som innehåller $\bar{x}$.

På detta sätt kan vi fortsätta så länge vi vill. Vi får om vi fortsätter ytterligare några steg nedanstående följd av intervall:

a	b
1.000000000000000	2.000000000000000
1.000000000000000	1.500000000000000
1.250000000000000	1.500000000000000
1.375000000000000	1.500000000000000
1.375000000000000	1.437500000000000
1.406250000000000	1.437500000000000
1.406250000000000	1.421875000000000
1.414062500000000	1.421875000000000
1.414062500000000	1.417968750000000
...	...

Efter i stycken *bisektioner* (bisektion = delning av intervallet mitt itu) har vi stängt in $\bar{x}$ i ett intervall med längd $(b-a)/2^i$. Väljer vi slutligen mittpunkten på detta intervall som approximation till $\bar{x}$ får vi ett fel som är mindre än $(b-a)/2^{i+1}$.

I stället för att uppdatera a och b hela tiden, kan vi införa beteckningarna x_i och X_i för vänster resp. höger ändpunkt hos intervallet som vi funnit efter i stycken bisektioner:

$$
\begin{aligned}
[x_0, X_0] &= [1, 2] \\
[x_1, X_1] &= [1, 1.5] \\
[x_2, X_2] &= [1.25, 1.5] \\
\ldots &\quad \ldots
\end{aligned}
$$

Talen i följderna $x = \{x_0, x_1, x_2, ..., x_i, ...\}$ och $X = \{X_0, X_1, X_2, ..., X_i, ...\}$ närmar sig bägge $\bar{x}$ mer och mer när i växer. Vi säger att talföljderna konvergerar mot $\bar{x}$. Notera att $X_i - x_i = (b-a)/2^i$.

Uppgifter.

1. Skriv ett program som implementerar bisektionsalgoritmen enligt programskalet `bisekt.m` (kopiera filen genom att klicka på länken eller ladda ned den från kurshemsidan) och flödesschemat `bisekt.pdf`. Mycket av programmet finns redan i programskalet. Din uppgift är att komplettera koden på de platser där det står ??.

2. Använd programmet för att beräkna alla rötter till följande ekvationer. Tips: skriv ekvationerna på formen $f(x) = 0$, skriv en funktionsfil för varje exempel, kalla dem `funk1.m`, `funk2.m` och så vidare, plotta funktionerna för att få en grov uppfattning om var rötterna ligger.

(1) $x^2 = 2$
(2) $x^4 = 2$
(3) $x^2 - 4x + 8 = 0$
(4) $\sin(x) = 0$
(5) $\cos(x) = x$
(6) $11x^3 - 2x^2 - 77x + 14 = 0$

(7) Hitta på en ekvation själv.

Tips: när funktionen finns som en funktionsfil är det bekvämt att använda `fplot`:

```
>> fplot(@funk1, [-5, 7])
```

3. Lägg till kod som skriver ut ett felmeddelande på skärmen och sedan avbryter och returnerar ett tomt värde `x = []` om $f(a)$ och $f(b)$ inte har olika tecken från början.

Tips: Använd `disp` och `return`, samt gör tilldelningen `x = []`, inuti en `if`-sats i början av ditt program. Jämför med hur vi lite längre ner i programmet (rad 46–48 i programskalet `bisekt.m`) avbryter med `return` om vi funnit en lösning.

4. Skriv ett program som implementerar bisektionsalgoritmen enligt programskalet `bisektdemo.m` och flödesschemat `bisektdemo.pdf`. Obs att skillnaden jämfört med `bisekt.m` är att vi nu sparar följderna x_i och X_i i varsin kolonnvektor. (Anledningen till att vi skriver två nästan likadana program är att du ska förstå programmeringen och matematiken bättre.) Även i detta programskal är det din uppgift att komplettera koden där det står ??. Lite mer denna gång. Du kommer successivt att få skriva mer och mer på egen hand.

5. Kör programmet på något av föregående exempel. Observera hur de beräknade följderna konvergerar. Hur många steg behöver algoritmen för att felet ska bli mindre än 10^{-10}?

Jämför med det teoretiska resulatet $|x_i - X_i| \leq (b-a)2^{-i} \leq 10^{-10}$. Tips: lös ut antalet steg i ur denna olikhet.

6. MATLAB har programmet `fsolve`. Läs om detta och prova på något av våra exempel.

`/stig`