

TANA17 Matematiska beräkningar med Matlab

Laboration 3. Funktioner

Namn: _____

Personnummer: _____

Epost: _____

Namn: _____

Personnummer: _____

Epost: _____

Godkänd den: _____

Sign: _____

Retur: _____

1 Introduktion

I denna laboration skall vi träna på att använda Matlab för att lösa simuleringsuppgifter som kräver lite mer omfattande programmering.

I de flesta fall skall både Matlab program och grafer redovisas. Enklast är därför att skriva de Matlab kommandon som löser uppgiften i editorn. Då blir det enkelt att skriva ut det som skall redovisas efter att uppgiften är löst.

För att skriva ut Matlab program kan kommandot `a2ps` användas. Du använder då ett terminal-fönster och skriver `a2ps *.m` så skrivs alla Matlab program i aktuell katalog ut till skrivaren.

2 En Talföljd

Med utgångspunkt ifrån ett godtyckligt positivt heltal bildas en talföljd enligt följande princip:

1. Om ett tal i följderna är udda bildas nästa tal genom att multiplicera det med 3 och addera 1 till produkten.
2. Om talet är jämnt bildas nästa tal i följderna genom att dividera med 2.

Om man börjar med siffran 6 får alltså talföljden:

6 3 10 5 16 8 4 2 1.

Man förmodar att talföljder som bildas på detta sätt alltid slutar med 1 men har inte lyckats bevisa det. Börjar man med ett tal mellan 1 och 100 så kommer man i vilket fall till 1 så småningom.

Uppgift 2.1 Skriv en funktion `BildaTalfoljd` som bildar en talföljd enligt metoden ovan. Start-siffran skall vara inparameter och utparameter skall vara längden på den talföljd man får. Både start och slutsiffran skall räknas så längden på den talföljd man får då man startar med siffran 6 är 9. $\square$

Tips Du kan kontrollera om ett tal är jämnt eller udda genom att dela det med två, avrunda resultatet till närmaste heltal, och sedan multiplicera med två igen. Om talet är jämnt så får du tillbaka samma tal igen. Läs hjälptexten till funktionen `round`. Alternativ är att använda funktionen `mod`.

Uppgift 2.2 Skriv ett program som använder funktionen `BildaTalfoljd` för att undersöka hur långa talföljder man får då man startar med talen 1–100. Du skall skriva ut för vilken startsiffra som den längsta talföljden erhålls. Hur lång blev den längsta talföljden? $\square$

Svar: _____

Uppgift 2.3 Utöka funktionen `BildaTalfoljd` så att den även kan skriva ut den talföljd som bildas. Anropar vi funktionen med ett extra argument så skall utskrift göras enligt mönstret

```
>> n = BildaTalfoljd( 6 , 'print' );  
Talföljd: 6 3 10 5 16 8 4 2 1.  
Längd: 9
```

Anropas funktionen med ett inargument, dvs som

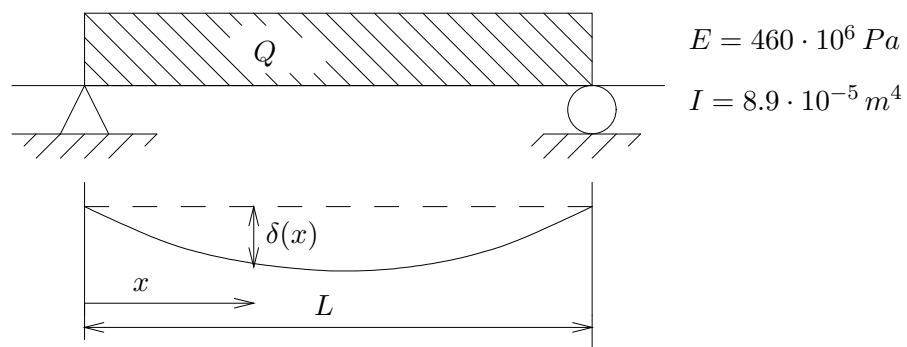
```
>> n = BildaTalfoljd( 6 );
```

så skall inga utskrifter göras.

Tips Funktionen `nargin` kan användas för att ta reda på om ett eller två inargument gavs värden då funktionen anropades. $\square$

Redovisa Skriv ut dina program. Det räcker om du skriver ut den slutgiltiga versionen av `BildaTalfoljd`.

3 Ett snötäckt tak



Figur 1: En balk som belastas av en jämt fördelad last Q .

En balk som belastas kommer att böjas. Hur stor nedböjningen blir beror dels på balkens dimensioner, material, och hur lasten ser ut. I denna uppgift skall vi försöka dimensionera taket på en uteplats så att det klarar av att bära upp ett tjockt snötäcke utan att nedböjningen märks allt för mycket.

Snötäcket approximeras med en jämt fördelad last vilket ger en formel

$$\delta(x) = \frac{QL^3}{EI} \left(\frac{x}{24L} - \frac{x^3}{12L^3} + \frac{x^4}{24L^4} \right),$$

där $\delta(x)$ är nedböjningen, Q är den totala lasten, E är materialets elasticitetsmodul, I är balkens tröghetsmoment, och L är balkens längd. Situationen illustreras i Figur 1.

I fallet med snötäcket antar vi att vi har en konstant last på varje längdenhet. Vi har alltså en last q [N/m] och den totala lasten beräknas som $Q = qL$.

Uppgift 3.1 Låt balkens egenskaper vara givna enligt Figur 1. Skriv en funktion `BalkForm` med inparametrar x , q , och L som beräknar nedböjningen δ som funktion av x . Du skall kunna anropa din funktion på följande sätt:

```
>> delta = BalkForm( x , q , L );
```

där `delta` då blir en vektor av samma storlek som `x`.

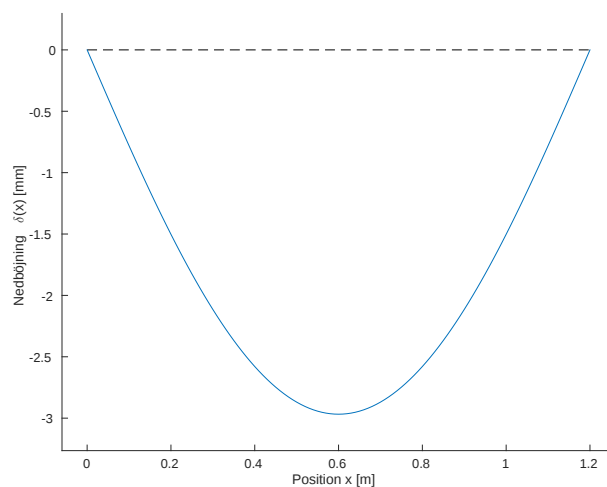
Tillåtna värden på x är endast intervallet $0 \leq x \leq L$. Ifall något x -värde ligger utanför intervallet skall en felutskrift ges med hjälp av `error` funktionen. Titta på `help error`. $\square$

Uppgift 3.2 För att illustrera nedböjningsfallet så skall vi plotta både funktionen $\delta(x)$ och utgångsläget utan last i en graf. Skriv en funktion `RitaBalk` som gör detta. Din graf skall ha samma utseende som i Figur 2.

Tips För att få en lagom stor utsträckning på koordinat axelarna beräkna $c = 0.1 * \delta_{max}$ och låt axlarna ha gränser $-0.05L \leq x \leq 1.05L$ och $-c - \delta_{max} \leq y \leq c$. Läs hjälptexten till `axis` funktionen för att se hur du skall göra.

Tips För att få grekiska tecken på y -axeln skriv

```
ylabel('Nedböjning \delta(x) [mm]');
```



Figur 2: Den graf som ritas upp av `RitaBalk` för fallet då $L = 1.2m$ och $q = 4500 N/m$. Den maximala nedböjningen blir här $3.0 mm$.

Uppgift 3.3 Vi vill att ett tak skall klara ett snölager som motsvarar en last $q = 1370\text{ N/m}$. Balken som skall bära upp snötäcket är 4 m lång och nedböjningen blir som störst 111.5 mm . Vi är beredda att acceptera 10 mm och skall därför placera ut extra stolpar som stödjer upp balken. Den maximala nedböjningen δ_{max} blir en funktion av avståndet L mellan två av stolparna. Vill vi att den maximala nedböjningen skall vara högst 10 mm . Vi bestämma det högsta tillåtna avståndet mellan två stolpar genom att lösa ekvationen

$$\delta_{max}(L) - 10.0 \cdot 10^{-3} = 0.$$

Skriv en funktion som beräknar nedböjningen som funktion av avståndet mellan två stolpar L . Använd sedan `fzero` för att hitta det största tillåtna avståndet L .

Hur många extra stolpar krävs för att stödja balken? $\square$.

Svar: _____

Redovisa Skriv ut dina program.