UMEÅ UNIVERSITET
Institutionen för Datavetenskap

Februari 2008

# Laboration 4, TVB, VT08
# ODE-system av första ordningen
# "Bungee Jumping"

---

## Redovisning

Syftet med att skriva rapport är bland annat att

- Träna på att kommunicera skriftligt
- Visa förmåga att kritiskt granska resultat
- Bättre förstå vad man kan och vad man behöver träna mer på

Därför är det ungefär lika bra att ge fel svar med tydliga resonemang som att ge rätt svar utan att beskriva resonemanget. Att hitta och åtgärda fel blir lättare med en välskriven rapport.

Rapporten förväntas innehålla följande:

- Försättsblad med **namn**, **datum**, **sökväg till källkoden**, samt **användarnamn på alla gruppmedlemmar**
- Väl **förklarade lösningar** på alla ingående deluppgifter
- **Kommenterade resultat** med egna reflektioner och slutsatser
- **Välformaterad källkod** utskriven i fixed-width typsnitt (t.ex. courier)
- Kommentarer om labben: rolig/tråkig, svår/lätt, oklarheter?

Vid komplettering (betyget O) lämnas orginalrapporten in tillsammans med modifierade bitar av rapporten/koden. Ett nytt försättsblad måste skrivas ut.

---

Uppgiften löses enskilt eller i par
och inlämnas senast
**Onsdagen den 19 mars 2008 klockan 12:00**

---

Rapporten lämnas i facket, märkt med kursens namn och kurskod, utanför institutionen.

# Inledning

Den här uppgiften behandlar bungee jumping som var en fluga för några år sedan. Det går att ställa upp fysikaliska modeller för bungee jumping som resulterar i ODE-system av första ordningen. Oavsett hur svåra dessa är att lösa analytiskt är de ungefär lika lätt att skatta numeriskt.

Som bakgrund har vi följande beskrivning som är en fri översättning från engelska och hämtat från boken "Engineering Problem Solving with Matlab" av Etter, D.M.

> En bungee-hoppare ska hoppa från en bro mha ett 150 meter långt bungee-rep. Han vill uppskatta sin maximala acceleration, hastighet samt hur långt han faller så att han kan vara säker på att inte utsättas för för höga G-krafter och att längden på repet är kort nog. Ekvationen han använder för analysen är Newtons andra lag:
>
> $$F = ma \tag{1}$$
>
> där $F$ summan av gravitationskraften, luftmotståndet, och bungee-kraften som verkar på honom. Vidare är $m$ hans massa (70kg) och $a$ är hans acceleration. Han börjar med att definiera sträckan han faller som $x$ (vilken är en funktion av tiden; $x(t)$). Hans hastighet och acceleration representeras av $x'$ respektive $x''$. Han omordnar Newtons ekvation (ekv. 1) för att lösa ut accelerationen:
>
> $$x'' = \frac{F}{m}. \tag{2}$$
>
> Sedan bestämmer han krafterna som utgör $F$. Gravitationskraften är hans vikt:
>
> $$W = mg = (70[\mathrm{kg}]) \times (9.8[\mathrm{m/s^2}]) = 686[\mathrm{N}]. \tag{3}$$
>
> Han vet att luftmotståndet, $D$, kommer vara proportionell mot kvadraten på hans hastighet, $D = c \times (x')^2$, men han vet inte proportionalitetskonstanten $c$. Men han vet dock utifrån sin erfarenhet som en fallskärmshoppare att hans maximala hastighet (eng. *terminal velocity*) i fritt fall är ungefär 55m/s. I den hastigheten är luftmotståndet lika med hans vikt, så han bestämmer $c$ mha:
>
> $$c = \frac{D}{(x')^2} = \frac{686[\mathrm{N}]}{(55[\mathrm{m/s}])^2} \approx 0.227[\mathrm{kg/m}]. \tag{4}$$
>
> Efter han fallit i 150 meter så är bungee-repet sträckt och kommer att börja påverka hopparen med en kraft $B$ om 10[N] för varje meter som repet sträcks utöver 150 meter. Därför kommer det finnas två regioner för att beräkna accelerationen. Den första används när sträckan $x$ är mindre än eller lika med 150 meter:
>
> $$\begin{aligned} x'' = \frac{F}{m} &= \frac{W - D}{m} \\ &\approx \frac{686 - 0.227(x')^2}{70} \\ &\approx 9.8 - 0.00324(x')^2[\mathrm{m/s^2}]. \end{aligned} \tag{5}$$

Den andra ekvationen för accelerationen kommer användas när $x$ överstiger 150 meter:

$$\begin{aligned} x'' = \frac{F}{m} &= \frac{W - D - B}{m} \\ &\approx \frac{686 - 0.227(x')^2 - 10(x - 150)}{70} \\ &\approx 31.24 - 0.00324(x')^2 - 0.143x[\mathrm{m/s}^2]. \end{aligned} \tag{6}$$

Nedanstående problem har anknytning till bungee-problemet som beskrivits ovan. Till er hjälp har ni filen `rungekutta.m` som ni hittar under MBIB på kursens hemsida. Den filen definierar en funktion som använder den klassiska Runge-Kutta–metoden.

**a)** Utgående från ekvation (5) eller (6) kan vi skriva ett ODE-system av första ordningen på *standardform* (se "Grundkurs i Numeriska Metoder s. 201f). Eftersom de två olika ekvationerna har en kontinuerlig övergång vid $x = 150$ kan vi utan problem slå samman bägge ekvationerna och använda den i den numeriska lösaren (dvs `rungekutta.m`). Överför de två ekvationerna till standardform. Skriv en matlab-funktion `y = f(t, u)` som tar som indata tiden $t$ och en vektor $\mathbf{u}$ och ger som utdata en vektor $\mathbf{y} = f(t, \mathbf{u})$ (dvs implementera en funktion som beräknar en av de två standardformernas funktion beroende på vilken som är tillämpbar). Redovisa tydligt vilka hjälpfunktioner ni inför och och bifoga som vanligt matlab-koden. Tips: tänk på att tecknet för luftmotståndet måste ändras på vägen upp eftersom kraften är motriktad hastigheten.

**b)** Använd `rungekutta.m` för att lösa ODE-systemet för en tillräckligt lång tidsrymd så att ett par tre svängningar hinner inträffa. Bestäm med hjälp av de experimentella resultaten maxvärdena för (beloppet av) acceleration, hastighet, och avstånd under denna tidsrymd. Hur högt upp måste bron vara för att ett säkerhetsavstånd på 10 meter skall gälla?

**c)** Vi lägger till en extra kraft för att simulera en viskös friktionskraft hos linan. Vi betecknar denna med $R$ och den sätter i så fort linan sträckts ut (dvs när $x > 150$). Det gäller att

$$R = 1.5x'. \tag{7}$$

Man är intresserad av att ta reda på maximal massa för en person som hoppar bungee från bron på den höjd som beräknades fram i uppgift b), *men med ett säkerhetsavstånd på* 5m. Ekvationerna (5) och (6) måste nu skrivas om så att parametern $m$ (massan) ingår, såväl som den nya kraften $R$ (som bara är tillämpbar i ekvation (6)). Skapa på motsvarande sätt som i a)-uppgiften en matlab-funktion som beräknar standardformernas funktion. Tänk på att massan är en fri variabel och kan därför med fördel deklareras som `global` (se matlabs hjälpfiler). På detta sätt kan massan ändras utifrån och vi har nu den infrastruktur som behövs för att lösa problemet. Formulera problemet med att hitta maximal massa som en ickelinjär ekvation $g(m) = 0$ och lös med hjälp av sekantmetoden så att $m$ får två korrekta decimaler. Här får ni anta att trunkeringsfelet i Runge-Kutta är försumbart jämfört med felet i ekvationslösningen.