

# Laboration 1, differentialekvationslösning för en modell med förarlösa bilar

Labben redovisas under den schemalagda övningen onsdagen 29:e september. Kom väl förberedd och med välordnade resultat till redovisningen. Numeriska resultat ska finnas noterade. Båda i laborationsgruppen ska kunna redogöra för teori och algoritmer!

Häftet "MATLAB 8 i korthet" ger en stegvis introduktion till MATLAB som kan vara användbar. Flitigt utnyttjande av MATLABs `help`-kommando rekommenderas också.

Vi ska se på en modell med ett antal förarlösa bilar som använder samma styrsystem. Antag att  $M$  bilar kör efter varandra på en väg. Positionen till bil  $i$  vid tiden  $t$  benämns  $x_i(t)$ . Bilarna är ordnade så att

$$x_i(t) < x_{i+1}(t), \quad \text{för } i = 1, \dots, M-1. \quad (1)$$

Varje bil anpassar sin hastighet med hänsyn till avståndet till framförvarande bil. Positionerna till bil  $1, \dots, M-1$  ges därför av differentialekvationer

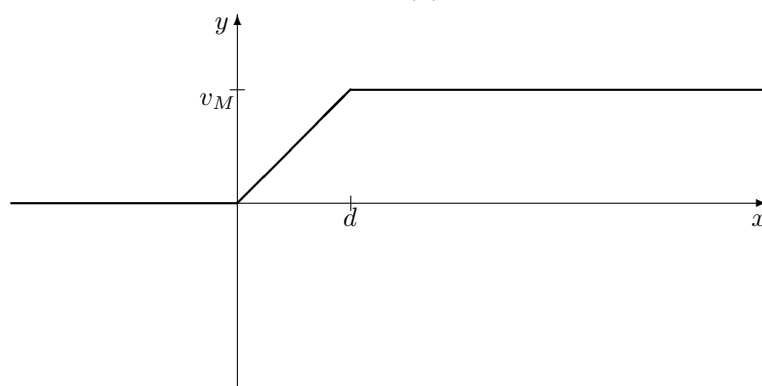
$$x'_i(t) = f(x_{i+1}(t) - x_i(t)), \quad i = 1, \dots, M-1. \quad (2)$$

Bil  $M$ , som ligger främst i ledet av bilar och därför inte har någon annan bil framför sig, löser differentialekvationen

$$x'_M(t) = g(x_M(t)). \quad (3)$$

Funktionen  $g$  kommer att senare att specificeras när de båda fallen inbromsning respektive acceleration behandlas.

- a) I figuren nedan visas grafen  $y = f(x)$  till funktionen  $f$ .



Konstanten  $v_M$  betecknar bilarnas maxhastighet (hastighetsbegränsningen), och konstanten  $d$  är det avstånd till framförvarande bil vid vilket bilarnas styrsystem börjar lätta på gasen. Vi kan här låta  $v_M = 25$  m/s och  $d = 75$  m. (Bilarna är alltså programmerade att lyda "tresekundersregeln".) Skapa funktionen  $f$  i Matlab, dvs. en Matlab-funktion som givet indata  $x$  returnerar  $f(x)$ . Skriv ut grafen till din Matlab-funktion.

- b) Implementera Eulers metod för att lösa systemet av differentialekvationer (2), (3):

$$\begin{aligned} x_i^{n+1} &= x_i^n + h f(x_{i+1}^n - x_i^n), \quad i = 1, \dots, M-1, \\ x_M^{n+1} &= x_M^n + h g(x_M^n). \end{aligned} \quad (4)$$

Här betecknar  $x_i^n$  Euler-approximationen efter  $n$  steg, dvs. en approximation av  $x_i(t_n)$ , där  $t_n = nh$ . Lös Euler-systemet (4) för följande två fall. För varje fall, plotta lösningskurvorna  $x_i(t)$  på tidsintervallet  $0 \leq t \leq 40$  för alla bilarna i samma figur (en figur för varje fall). Välj lämpligt tidssteg  $h$  och antal bilar  $M$ , t.ex.  $M = 10$ .

- (i) *Inbromsning*. Låt bilarnas positioner vid tiden  $t = 0$  ges av  $x_i(0) = d \cdot i$ ,  $i = 1, \dots, M$ . Bil  $M$ , som kör längst fram, kör långsamt med hastighet  $g = 5$  m/s.
  - (ii) *Acceleration*. Låt bilarnas positioner vid tiden  $t = 0$  ges av  $x_i(0) = 10i$ ,  $i = 1, \dots, M$ . Bil  $M$  har fri väg framför sig, så  $g = v_M$ .
- c) Skriv en kod som genererar en film av bilarnas framfart. Om t.ex. bilarnas positioner sparas i en matris  $\mathbf{x}$  med dimension  $N_{\text{steg}} \times M$ , där varje rad i matrisen är positionen av bilarna vid en viss tidpunkt, och tidssteget i Eulers metod är  $h$ , så kan följande kod användas:

```
for n=1:Nsteg
    plot(x(n,:),zeros(1,Ncars),'r*');
    axis([0 1000 -1 1])
    drawnow
    pause(h)
end
```

Koden ovan ger en film som visar bilarnas framfart i realtid, förutsatt att tidssteget  $h$  inte är för litet. Ett tidssteg  $h=0.1$  bör t.ex. fungera väl.

- d) Denna deluppgift löses med papper och penna. Om tidssteget  $h$  är för stort kan det hända att bilarna "kör förbi varandra" i Euler-lösningen, dvs.  $x_i^n > x_{i+1}^n$ , för något tidssteg  $n$ . Bestäm en övre gräns för tidssteget  $h$  sådan att det är säkert att detta inte sker.
- e) Denna deluppgift löses med papper och penna. Skriv upp iterationsformeln för metoden bakåt Euler (implicit Euler) tillämpad på systemet av differentialekvationer (2), (3). Om vi betraktar fallen inbromsning eller acceleration, dvs. om  $g$  är konstant, gäller det att givet en bakåt Euler-lösning vid tidssteg  $n$ ,  $x_i^n$ ,  $i = 1, \dots, M$ , det för varje tidssteg  $h$  existerar en bakåt Euler-lösning i nästa tidssteg  $x_i^{n+1}$ ? Om det finns en lösning i tidssteg  $n+1$ , är denna entydig?
- f) Kan det ske några förbikörningar i bakåt Euler-modellen, eller kommer det att gälla att  $x_i^n < x_{i+1}^n$  för alla tidssteg  $h$ ?
- g) Om tidssteget  $h$  är tillräckligt litet kan fixpunktsmetoden användas för att bestämma bakåt Euler-lösningen  $\mathbf{x}^{n+1}$  givet  $\mathbf{x}^n$ , där

$$\mathbf{x}^n = \begin{pmatrix} x_1^n \\ x_2^n \\ \vdots \\ x_M^n \end{pmatrix}.$$

Skriv ett MATLAB-program som med hjälp av fixpunktsmetoden beräknar en bakåt Euler-lösning till systemet av differentialekvationer (2), (3).

- h) Om tidssteget  $h$  är för stort fungerar inte fixpunktsmetoden (åtminstone inte i sin mest uppenbara form) för att beräkna bakåt Euler-lösningen. För vårt system av differentialekvationer är det dock möjligt att hitta ett uttryck med papper och penna som bestämmer bakåt Euler-lösningen  $\mathbf{x}^{n+1}$  givet  $\mathbf{x}^n$ . Finn detta uttryck med papper och penna, för fallet inbromsning. Skriv ett MATLAB-program som använder det exakta uttrycket för bakåt Euler-lösningen. Kontrollera att programmet kan beräkna lösningen även med stort tidssteg  $h$ . Med denna lösning som referens, visa grafiskt

hur fixpunktsiterations-felet beror av antalet fixpunktsiterationer. Gör detta genom att välja en steglängd sådan att fixpunktsiterationerna konvergerar. Plotta sedan absolutbeloppet av skillnaden mellan bakåt Euler-approximationen av  $x_1(40)$  beräknad med hjälp av fixpunktsiterationer och den exakta bakåt Euler-lösningen, mot antalet fixpunktsiterationssteg. Använd logaritmisk skala i  $y$ -led, dvs. använd MATLAB-kommandot `semilogy`.

- i) Diskutera huruvida systemet av differentialekvationer (2), (3) är en rimlig modell för trafikflöden.