

INTRODUKTIONSÖVNINGAR I MATLAB

Detta är ett övningshäfte för MATLAB och förutsätter att man parallellt studerar läroboken "Användarhandledning för MATLAB 6" eller på annat sätt inhämtar grunderna i MATLAB. Tanken är alltså att du ska ha läroboken tillgänglig för att kunna lösa uppgifterna. På så sätt lär Du dig också att hitta i boken vilket är mycket bra för kommande behov.

Uppgifterna ger Dig en introduktion till MATLAB. Du kan göra dem i din egen takt och efter eget urval men tveka aldrig att rådgöra med din handledare. Appendix A i boken innehåller även en elementär introduktion till MATLAB.

För uppgifterna 1 - 19 är kapitel 1 och 2 i MATLAB-boken relevanta. Ögna gärna i genom dessa kapitel först, men fördjupa Dig inte i sådant som verkar svårt eller obekant. Det kommer att klarna så småningom.

Att komma igång, hjälpfunktioner mm

1. Se till att du har tillgång till MATLAB-6 från ditt login. Om inte kan du fixa detta med kommandot "tomten". Starta MATLAB med kommandot **matlab** i UNIX. Som du märker startas en webbläsare som laddar in en www-sida med tre delar: kommandofönster för MATLAB, historiefönster och hjälpfönster (launch pad). Det är i kommandofönstret som vi kommunicerar med MATLAB. Hjälpfönstret används för att söka information om olika funktioner och program i MATLAB. I följande uppgift ska vi se exempel på det. Historiefönstret är användbart när man vill upprepa något man nyligen gjort i MATLAB. Det är bara att klicka på aktuella rader och med "return" få dem exekverade igen i MATLAB. Alternativt kan man flytta in text från historiefönstret (med musen) till kommandofönstret för att få kommandona utförda igen.
2. Använd "MATLAB-help" i "launch pad" för att lista alla elementära matematiska funktioner som finns i MATLAB. Hur gör du om du vill ha mer detaljerad beskrivning av en viss funktion?
3. Använd "MATLAB-help" i "launch pad" för att ta fram en beskrivning av funktionen **fzero**. Försök förstå vad funktionen gör.
4. Använd "MATLAB-demos" i "launch pad" för att köra ett demonstrationsprogram som ritar en tredimensionell nivåyta (**mesh**).

Ett tips: Du kan få en demonstration av hur de olika fönstren kan användas genom "MATLAB-demos-Desktop Environment" och sedan knappen "Run" på det alternativ Du vill få demonstrerat.

5. Ett enkelt alternativ till att söka information enligt uppgift 3 ovan är att använda MATLAB-kommandot **help**, i kommando-fönstret alltså. Testa det på samma funktion som i uppgift 3.

6. Ett alternativt sätt att ta reda på vilka funktioner som finns i MATLAB för ett visst ändamål är att använda **lookfor**. Kommandot **lookfor text** letar upp alla rutiner som har texten **text** i första raden av sin help-text. Om man i stället skriver **lookfor -all text** så letar MATLAB igenom hela help-texten, inte bara första raden. Ge kommandot **lookfor logarithm** för att se vilka logaritmfunktioner som finns och hur de anropas. Man kan lista innehållet i en MATLAB-funktion (inbyggd eller en som man skapat själv) med kommandot **type**, se övning 16 nedan.

Anm. Du avbryter en pågående exekvering i MATLAB med **Control-c**. Du avslutar MATLAB genom att skriva **exit** i kommando-fönstret.

MATLAB som räknedosa

7. Beräkna följande uttryck i MATLAB. Elementära funktioner hittar du i boken på sid 26-27 eller med **help elfun** och speciella fördefinierade variabler kan du läsa om på sid 23 i boken.

$$\sin(\pi), \ln(3^2), \ln(\sqrt{e}), \log_{10}(100), e^2, |5 + 3i|, e^{i\pi}, 1/0, 0/0, (1/0) * 0, (0/0) * (1/0)$$

OBS. MATLAB skiljer på små och stora bokstäver. Kolla vad som händer om Du skriver t.ex. **SIN(2)**.

8. Tilldela variabeln **x** värdet π , variabeln **y** värdet 355 och variabeln **z** värdet 226. Beräkna sedan följande uttryck och skriv ut resultatet med 5 resp 15 siffror, se **help format**:

$$a = x/2, b = y/z, c = a - b, d = a - \sin(y)$$

Ange utgående från dessa beräkningar en bra rationell approximation av π . Hur stort är felet?

Anm. Om man inte vill att MATLAB ska svara på alla beräkningar med att skriva ut variabeln eller det beräknade värdet så avslutar man beräkningssatsen med semikolon (;).

9. Eftersom datorn räknar med ändlig precision blir de flesta resultat närmevärden. Vi vill undersöka hur stora felen kan bli i följande (matematiskt korrekta) trigonometriska formler:

$$\sin(2x) = 2 \sin(x) \cos(x) \text{ och } \sin^2(x/2) = (1 - \cos(x))/2.$$

Välj några olika värden på x och låt MATLAB beräkna och skriva ut skillnaden mellan vänster- och högerled i formlerna dividerat med vänsterledets värde, dvs vi tittar på **relativa** felet. Är det några värden som verkar ge stora fel i någon av formlerna? Jämför med variabeln *eps*, se sid 23 i boken. Hur blir det speciellt med små x -värden i den andra formeln?

10. Vi antar att $R \gg r$ (innebär mycket större än) och betraktar formeln $R(1 - \cos(r/R))$, som då kan approximeras med $r^2/(2R)$, där approximationen blir bättre och bättre ju mindre r är. (Detta följer av Taylorutveckling som vi skall gå igenom senare i kursen.) Låt

nu $R = 4 * 10^7$ och undersök genom att utföra beräkningarna i MATLAB hur stort felet i approximationen är om $r = 10^p$, för $p = 0, 1, 2, 3, 4, 5$. Verkar resultatet rimligt?

11. Rita kurvorna $y = x$ och $y = 2 \sin x$. (Använd gärna MATLAB.) Definiera en talföljd x_n genom $x_0 = 1, 5$ och $x_{n+1} = 2 \sin x_n$. Undersök $\lim_{n \rightarrow \infty} x_n$. Har gränsvärdet något med ekvationen $x = 2 \sin x$ att göra? Kan du bevisa det?
12. (a) Visa att ekvationen $x^3 - x = 1$ har en rot mellan 1 och 2.
(b) Skriv ekvationen $x = x^3 - 1$ och försök lösa ekvationen rekursivt genom att sätta $a_0 = 1$ och $a_{n+1} = a_n^3 - 1, n = 0, 1, 2, \dots$. Vad händer?
(c) Skriv ekvationen $x = (x + 1)^{1/3}$ och försök lösa ekvationen genom att låta $a_0 = 1$ och $a_{n+1} = (a_n + 1)^{1/3}, n = 0, 1, 2, \dots$. Vad händer?
(d) Försök bevisa din observation i (c) (nämligen att $a_n \rightarrow a$ där a löser ekvationen) genom att bevisa att $a_n \leq 2$ och att a_n är en växande följd.

Programering i MATLAB

För att få lättöverskådligare och effektivare program i MATLAB finns några programmeringsredskap. Vi kommer att öva lite på dessa. De är beskrivna i avsnitt 12.1 - 12.4 i läroboken. Det handlar om villkorssatser, som vi brukar kalla **if-satser** och repetitionssatser av två slag **for-satser** och **while-satser**. Vid if-satserna används ofta relationsoperatorer, se avsnitt 3.9, och logiska operatorer, se avsnitt 3.10.

13. Skriv en MATLAB-sekvens som skriver ut texten "mellan -1 och 2" om variabeln x har värden mellan -1 och 2, annars skrivs texten "ej i intervallet" ut.
14. Skriv en MATLAB-sekvens som ger text-variabeln kvot värdet "low" om variabeln test har värde mindre än 100, värdet "medium" om test har värde mellan 100 och 150 och värdet "high" om test har värde större än 150. Testa med olika värden på variabeln test och skriv ut motsvarande innehåll i variabeln kvot.
15. Skriv en for-sats för beräkning av summan av de 20 första termerna i serien 1, 1/2, 1/4, 1/8, 1/16, ...
16. Skriv en while-sats för beräkning av en delsumma av serien i uppgift 15, där summeringen avbryts när termerna är mindre än en miljondel.

Vi ska nu se hur man kan skapa egna kommandon och funktioner i MATLAB. Dessa ska ha namn som slutar med **.m** och skapas som filer i en editor. MATLAB har en egen editor men vi rekommenderar att Du använder EMACS, som Du har tillgång till på Din arbetsstation och kan få hjälp med. Innan Du gör denna uppgift bör Du läsa avsnitt 2.9 i boken ganska noggrant, utom exempel 2.20, som vi inte är riktigt mogna för ännu.

17. Skriv nu en kommandofil med namnet **addera.m** som helt enkelt består av matlabberäkningen $a = b + c$. Spara filen och testa att exekvera den från MATLAB. Variablerna **b** och **c** måste ha fått värden innan förstås.

Lägg nu till kommentaren "Jag adderar b plus c i variabeln a" *först* i filen **addera.m**. (Kommentarer skrivs med procent-tecken i första position.) Vad händer om Du nu skriver **help addera** i MATLAB. Just det, Du har skapat en egen help-information.

Gör om **addera.m** till en funktionsfil i stället, med parametrarna **b** och **c**, och anropa den från MATLAB som en funktion.

Ge nu kommandot **type addera** och konstatera att din funktion betraktas som MATLABs egna, detta under förutsättning att den ligger i aktuellt MATLAB-bibliotek dvs aktuellt directory där Du startade MATLAB från. Man kan kolla vilka egna m-filer man har i biblioteket genom kommandot **what** i MATLAB. (Man kan även komplettera de bibliotek som man vill att MATLAB ska söka i genom kommandot **matlabpath**)

18. Anta att du upprepade gånger vill använda if-satsen i uppgift 13. Det kan då vara relevant att lägga satsen i en m-kommandofil. Gör det och kalla filen t.ex. **intelligent.m**. Exekvera den i MATLAB med några olika värden på variabeln **test**.
19. Harmoniska serien är $1, 1/2, 1/3, 1/4, \dots$. Skriv en funktion **harmonisk.m** med parameter **n**, som beräknar partialsumman av **n** termer ($1+1/2+1/3+1/4+\dots+1/n$) genom att på lämpligt sätt använda en for-loop.

Harmoniska serien är inte konvergent dvs summan blir godtyckligt stor bara man tar med tillräckligt många termer. Vi vill veta hur många termer som behövs för att summan ska överstiga värdet **tak**. Skriv en ny funktion, som har **tak** som parameter och som returnerar antalet termer. Använd lämpligen en while-loop. Hur många termer behövs då $\text{tak}=12$?

Grafik i MATLAB

20. Rita med **plot**-kommandot ut funktionen $\cos(x)$ på intervallet $0 \leq x \leq 6\pi$. Använd lämplig upplösning i x -variabeln.

I uppgiften som följer blir det aktuellt att använda de s.k. elementvisa aritmetiska operationerna, som används för vektorer och matriser, se avsnitt 3.6 i boken. Försök förstå hur dessa fungerar. Det blir lättare att till fullo inse nödvändigheten och kraftfullheten av dessa operationer när vi lite senare studerat just vektorer och matriser i den linjära algebran.

21. På samma sätt som i uppgift 20 kan man generera en vektor med funktionsvärden motsvarande funktionerna $\sin(x)$ och e^x och rita ut en graf över funktionerna. Hur gör man lämpligen om man vill rita ut funktionerna $\sin(x) + \cos(x)$, $\sin(x) * \cos(x)$ och $\sin(x)/e^x$ på samma intervall som i uppgift 20?

22. Rita kurvor för funktionerna

$$f(x) = x(1 - \sin \pi x) \text{ och } g(x) = 5e^{-x/2}/(3 - 2 \cos 2\pi x)$$

i intervallet $0 \leq x \leq 4$. Prova dig fram med lämpligt steg. Använd sedan kommandot **subplot** för att i nya bildrutor rita summafunktionen $f(x) + g(x)$, produkten $f(x) * g(x)$ och kvoten $f(x)/g(x)$.

23. Skriv en funktionsfil **step.m** som definierar funktionen

$$f(x) = \begin{cases} 0, & 0 \leq x \leq 1 \\ 1, & 1 < x \leq 2 \\ 0, & 2 < x \leq 3 \end{cases}$$

Skriv en kommandofil som ritar ut funktionen genom att anropa funktionsfilen. Testa även alternativet att använda MATLAB-funktionen **fplot**.

Extra: Försök skriva funktionsfilen **step.m** så att du kan rita genom satserna

$x = 0 : 0.01 : 3$; `plot(x, step(x))`

dvs på samma sätt som för exempelvis $x = 0 : 0.01 : 3$; `plot(x, sin(x))`

24. Du vet att funktionsfilen **decrease.m** definierar en funktion som avtar strikt från ett positivt värde då $x = -1$ till ett negativt värde då $x = 1$. Skriv en kommandofil som ritar ut funktionen från $x = -1$ så länge den är positiv. Testa ditt program på funktionen $f(x) = -2(\sin x + 0.3)$, som Du alltså lägger i funktionsfilen **decrease.m**.

Extra 1: Försök skriva kommandofilen utan att använda while-loop, utan i stället med hjälp av **min** och **find** funktionerna i MATLAB.

Extra 2: Gör om kommandofilen till en funktionsfil **rita.m**, som har den funktion som ska ritas ut som parameter. Hur man gör när man vill ha en funktion som parameter till en annan funktion framgår av avsnitt 12.4 i boken. Nu ska Du alltså kunna rita ut den funktion som är definierad i **decrease.m** (så länge den är positiv) med anropet **rita('decrease')** i MATLAB.

25. Rita funktionen $\tan(x)$ på intervallet $(-\pi/2, 9\pi/2)$. Här får man tänka sig för lite på grund av de stora funktionsvärdena. En ledning är att Du kan använda kommandot **axis** och en for-loop på lämpligt sätt. Resultatet bör bli något liknande Figur 1.

EXTRA: Lös uppgiften utan for-sats genom att på lämpligt sätt utnyttja att NaN-värden inte ritas ut.

26. Man bör försöka att undvika for-loopar och while-loopar om man vill få så effektiv MATLAB-kod som möjligt. Ofta kan man utnyttja att MATLAB är konstruerat för att hantera vektorer och matriser på ett bra sätt. Vi ger ett exempel. Följande två alternativa MATLAB-sekvenser ritar funktionen $\sin(t)$ för $0 \leq t \leq 20$:

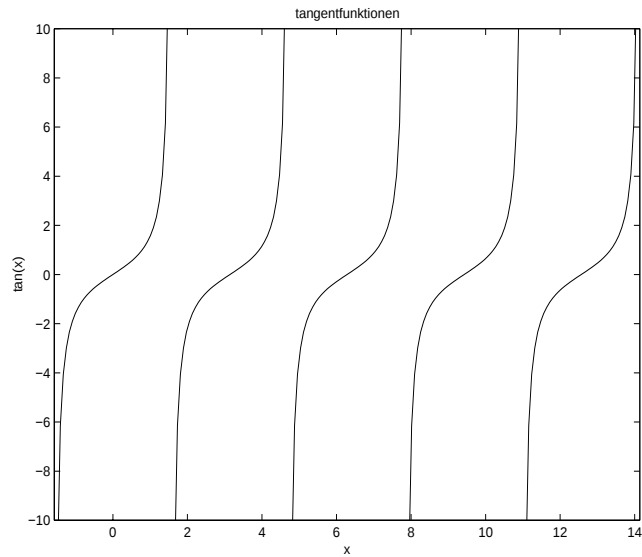


Figure 1: Tangentfunktionen i MATLAB

Alternativ 1:

```
t = 0 : 0.001 : 20;
```

```
plot(t, sin(t))
```

Alternativ 2:

```
n = 0;
```

```
for t = 0 : 0.001 : 20;
```

```
    n = n + 1;
```

```
    x(n) = t :
```

```
    y(n) = sin(t);
```

```
    plot(x, y)
```

```
end
```

Undersök hur lång tid, rutinerna **tic** och **toc** i MATLAB, som de båda alternativen kräver.

27. Cirklar ritas smidigt i MATLAB med parameterframställning: $x = x_c + r \cos \phi$, $y = y_c + r \sin \phi$, där (x_c, y_c) är centrumkoordinaterna, r är radien och där vinkeln ϕ går från 0 till 2π . Rita en cirkel, som har centrum i (0.72, 1.42) och radien 2.5. Använd steget (upplösningen) $2\pi/60$ map vinkeln. Markera ut mittpunkten.

Obs. Använd **axis equal** och **axis square** (efter plot-satsen) för att få skalorna på axlarna lika, annars ser det ut som en oval.

28. Slumpa ut tio cirkelar med radie 5 på området $0 \leq x \leq 60$, $0 \leq y \leq 45$, jämför uppgift 15. Klicka på tre av cirkelarna, som skall rödfärgas. Använd **ginput** och **fill** i MATLAB.

MATLAB-övningar på vektorer och matriser

29. Skapa följande vektorer med hjälp av "kolon-operatorn":

$$x = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10], y = [0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1]$$

Skapa vektorn $x_3 = [1 \ 8 \ 27 \ 64 \dots 1000]$ med hjälp av vektorn x .

Bilda delvektorerna x_1 och x_2 som första resp andra hälften av vektorn x .

Återskapa x ur delvektorerna x_1 och x_2

30. Låt $x = [1 \ 2 \ 3]^T$ och $y = [3 \ 2 \ 1]^T$. Transponat ges av apostrof ' i MATLAB. Vi transponerar för att få kolonnvektorer, vilket är det normala att arbeta med i linjär algebra. Undersök och försök förstå följande beräkningar:

$$x', y', x * y, x ./ y, x' * y, x * y', (x * y') * x \text{ Extra: } x / y, x \backslash y$$

31. **Ingen utmaning:** Beräkna $n!$ med kolonoperatorn och **prod**-funktionen i MATLAB, jämför övning 7.

32. **Lite större utmaning:** Beräkna summan av n termer i den harmoniska serien med endast ett anrop av **sum**-funktionen i MATLAB, jämför med övning 8.

33. Bilda diagonalmatrisen

$$D = \text{diag}(2, 3, 4) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}.$$

Skriv **help diag** så får Du veta hur man gör. Bilda sedan matrisprodukterna DA och AD , där

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

Vilka transformationer på A åstadkommer multiplikationerna med diagonalmatrisen.

34. Bilda en så kallad permutationsmatris P genom att byta första och tredje raden i enhetsmatrisen av ordning 3. Detta görs med satserna (försök förstå vad som händer):

$$P = \text{eye}(3), \quad P([1, 3], :) = P([3, 1], :)$$

Bilda nu matrisprodukterna PA och AP , där A är matrisen i uppgift 18. Vilka transformationer på A åstadkommer multiplikationerna med permutationsmatrisen.

35. Givet matrisen $A = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix}$

bilda delmatriserna A_{ij} , av ordning 2×2 , i partitioneringen $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$. Hur kan man sedan återskapa A från delmatriserna?

36. Kontrollera att associativa och distributiva lagarna gäller för matriserna

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 1 & 1 \\ 4 & 1 & 0 \\ -2 & 2 & 1 \end{bmatrix}$$

dvs att $A(BC) = (AB)C$ respektive $A(B+C) = AB+AC$ och $(A+B)C = AC+BC$. Visa att matrismultiplikation inte är kommutativ genom att visa att $AC \neq CA$. Vad gäller för AB och BA ?

37. Beräkna $x^T y$ och xy^T där x och y är vektorerna i uppgift 22. Bilda nu produkten $Axy^T B$, dels som $A(xy^T)B$ och dels som $(Ax)(y^T B)$, där A och B är matriserna i uppgift 26. Undersök med vilket som kräver minst beräkningsarbete. Försök förstå skillnaden.

Linjära ekvationssystem

38. Lös det linjära ekvationssystemet $Ax = b$, där A är matrisen i uppgift 25 och $b = [23 \ 32 \ 33 \ 31]^T$. Är lösningen korrekt? Är det enda lösningen?

39. Låt $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 1 \end{bmatrix}$ och $b = [1 \ 3]^T$. Lös ekvationssystemet $Ax = b$. Vad ger backslash i MATLAB. Är det enda lösningen? Använd **rref**-kommandot för att ta fram reducerad trappstegsform och därur bestämma lösningsmängden.

40. Lös matrisekvationen $AX = B$ där $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 5 \\ 7 & 8 & 0 \end{bmatrix}$ och $B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ Vad blir X om B är enhetsmatrisen?

41. Lös matrisekvationen (dvs bestäm X i uttrycket) $AXB + 2AX = C$, där $A = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$,

$$B = \begin{bmatrix} 1 & 4 & 2 \\ 0 & -1 & 3 \\ 1 & 1 & 1 \end{bmatrix} \text{ och } C = \begin{bmatrix} 1 & 1 & 2 \\ -1 & 3 & 1 \end{bmatrix}.$$

Ekvationslösning

42. Ljudnivån L (i decibel) på avståndet r meter från en ljudkälla ges av formeln

$$L = L_0 - 20^{10} \log(r) - \beta r$$

där L_0 är ljudnivån 1 meter från ljudkällan och β är en parameter som anger hur ljudet försvagas då det går genom luften. (β beror på luftens viskositet, temperatur och fuktighet.) Vi vill för $L_0 = 80$ dB bestämma det avstånd där ljudnivån är 20 dB då $\beta = 1.15 \cdot 10^{-3}$. Formulera en ekvation som ska lösas, rita en lämplig graf och läs av en approximation av avståndet med **ginput**. Bestäm sedan en noggrannare approximation med **fzero**.

43. En lång stång upphetas momentant vid tiden $t = 0$ i mittpunkten. Med denna punkt i $x = 0$ och x -axeln utmed stängen bestäms temperaturen $T(x, t)$ för $t > 0$ av uttrycket

$$T(x, t) = C e^{-x^2/kt} / \sqrt{t}$$

där parametrarna $k = 5.17$ och $C = 28.3$ beror av värmeledningsförmågan respektive den tillförda värmen. Beräkna under hur lång tid som temperaturen vid $x = 1$ överstiger 20° . Använd **fplot**, **ginput** och **fzero**.

44. (a) Bestäm alla reella nollställena till

$$f(x) = x^3 - x - 0.3847$$

Utnyttja **fplot**, eventuellt **zoom** och **ginput** för att lokalisera nollställena grovt och **fzero** för att beräkna dem noggrannare.

- (b) Studera känsligheten i funktionen med avseende på x -koefficienten, i ostörd variant är alltså denna koefficient $= -1$. Stör nu koefficienten med störningar $+0.1$ resp -0.1 och se vad som händer med nollställena.
- (c) Använd även rutinen **roots** för att bestämma nollställena. Vad är det för skillnad på **fzero** och **roots** vad gäller vilka nollställena (rötter) som kan bestämmas?
- (d) Skriv en egen funktionsfil för Newtons metod med funktionen och derivatan som parametrar. Använd funktionsfilen för att beräkna nollställena. Använd **ginput** för att få startapproximationer. Jämför antal beräkningstiden för **fzero**, **roots** och din egen Newtonfunktion.

Integraler

45. Beräkna integralen $\int_0^1 \ln(1+x^2) dx$ exakt och med rutinerna **quad** och **quadl** i MATLAB. Jämför noggrannhet och hastighet.

Ordinära differentialekvationer

46. Beräkna $y(4)$ där y är lösningen till begynnelsevärdesproblemet $y' = e^{-y} + t$, $y(0) = 0$ genom att använda **ode45** i MATLAB. Räkna fram en approximation med Eulers framåt-

metod och steg $h = 0.1$. Rita ut lösningen från **ode45** och approximationen från Eulers metod i samma diagram. Hur stort blir felet i $t = 4$?

47. För att kyla en kropp med temperatur T ($^{\circ}C$) används en fläkt. Enligt Newtons avsvlningslag gäller att temperaturminskningen är proportionell mot temperaturdifferensen mellan kroppen och den omgivande luftströmmen. Om luftens temperatur är 30 $^{\circ}C$ får vi därför $T' = -k(T - 30)$, där k är en avsvlningskonstant. Låt $k = 0.04$ och anta att kroppens temperatur är 100 $^{\circ}C$ vid kylningens början. Bestäm kroppens temperatur efter 20 tidsenheter. Använd Eulers framåtmetod och några olika steglängder. Verifiera att felet är proportionellt mot h , som teorin säger. Jämför med exakta lösningen som är $T(t) = 30 + 70 e^{-0.004 t}$.

48. En viss fastspänd bladfjäder, som vi böjer ut och släpper, beskrivs av differentialekvationen $x'' = -x^3 - 0.05 x'$, $x(0) = 1$, $x'(0) = 0$. Om vi låter $y = (y_1, y_2) = (x, x')$ får vi standardformen för ett system med två komponenter $y' = f(t, y)$, $y(0) = c$ där f och c är vektorerna

$$f = \begin{bmatrix} y_2 \\ -y_1^3 - 0.05 y_2 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Använd **ode45** för att beräkna lösningen och rita ut den på lämpligt sätt.