FY1005/TFY4165 Termisk fysikk. Institutt for fysikk, NTNU. Våren 2014.

Matlabtips til øving 10

Grunnleggende tips:

- To vanlige måter å lage en vektor som inneholder tall fra x til y: `a = 0:0.01:1` og `b = linspace(1,2,100)` Her er `a` en vektor med tall fra 0 til 1, med steglengde 0.01. `b` er en vektor med 100 tall fra 1 til 2.
- Funksjonen `length(a)` returnerer antall elementer i vektoren a.
- En `for`-løkke er praktisk for å plotte flere grafer i en og samme figur.
- Når du skal lage en ny vektor med tall basert på en funksjon av en gammel vektor, kan det være fristende å bruke en ny `for`-løkke. Men det er mye raskere å bruke innebygde elementvise operasjoner! Da kan vi lage en ny vektor `c` basert på `b` fra i sted slik:
  `c = b.*b` - her blir verdi nr. n i `c` lik kvadratet av verdi nr. n i `b`, slik at kommandoen `plot(b,c)` vil gi en parabel. Du kan bruke et punktum foran alle vanlige operasjoner.
- Etter det første plottet må du skrive `hold on` for at de neste kurvene skal komme i samme figur. (Du må skrive `hold off;` etter den siste kurven og deretter `figure;` dersom nye figurer skal lages senere i det nye programmet.) Det er også lurt å skalere aksene etter det første plottet, med kommandoen `axis([xmin xmax ymin ymax])`.

Et konkret eksempel er tatt med nedenfor. Du kan bruke dette eksemplet som utgangspunkt for å lage figur med isotermer.

```
%%FY1005/TFY4165, Oving 10, eksempel.
%%Filnavn: vdweks.m
zmin=1;
zmax=5;
Deltaz=1;
%%z = vektor med verdier mellom zmin og zmax, intervall Deltaz
z=zmin:Deltaz:zmax;
xmin=0.1;
xmax=pi;
Nx=500;
%%x = vektor med verdier mellom xmin og xmax, i alt Nx verdier
x=linspace(xmin,xmax,Nx);
%%length(z) = antall elementer i vektoren z
%%Bruker for-loekke fra i=1 til i=length(z) til aa regne ut en
%%funksjon y(x) for z-verdier z(1), z(2), ... , z(length(z))
for i = 1:length(z);
    y = sin(z(i).*x);
    fig = plot(x,y);
    %%y(x) for laveste z-verdi z(1): blaa kurve
    %%y(x) for hoeyeste z-verdi z(length(z)): roed kurve
    %%Mellomliggende kurver: gradvis mellom blaa og roed
    %%Tynne kurver, LineWidth = 1.0
    red=(i-1)/(length(z)-1);
    green = 0.0;
    blue=1-red;
```

```
    set(fig,'Color',[red green blue],'LineWidth',1.0);
    if i == 1;
        title('Noen harmoniske funksjoner','fontsize',18);
        xlabel('x','fontsize',18);
        ylabel('sin(zx)','fontsize',18);
        axis([0 xmax -1 1]);
        %%Kommandoen hold on; soerger for at paafoelgende
        %%kurver tegnes i samme figur
        hold on;
        %%Vi plotter ogsaa funksjonen sin(0.9zx) for laveste
        %%z-verdi, dvs for z(1)
        y2 = sin(0.9*z(i).*x);
        fig = plot(x,y2);
        %%Tykk blaa kurve for y2(x) ved z(1)
        set(fig,'LineWidth',1.5,'Color',[0 0 1]);
    end;
end;
hold off;
```

Videre skal Maxwells regel om like arealer brukes til å fastlegge koeksistenstrykket p_{co} for en gitt temperatur. En enkel algoritme for å løse dette problemet numerisk kan se omtrent slik ut:

- Basert på den aktuelle isotermen $p(V)$, gjett en verdi for væskevolumet V_v.
- Bestem tilhørende trykkverdi $p(V_v)$.
- Bestem tilsvarende gassvolum V_g, dvs slik at $p(V_g) = p(V_v)$.
- Bestem
$$\int_{V_v}^{V_g} p(V)\, dV$$
numerisk og sammenlign med arealet $p(V_v) \cdot (V_g - V_v)$.
- Gjett en ny verdi for V_v og gjenta prosedyren inntil p_{co} er bestemt med den ønskede nøyaktighet.

En innledende del til programmet er tatt med nedenfor, inkludert noen tips for den videre programmeringen:

```
%%FY1005/TFY4165, Oving 10, innledende del
%%
%%Velg utskrift av tallverdier med flere gjeldende siffer enn det som er standard
format long;
%%Temperatur i Kelvin
T=331;
%%V = molart volum (m^3)
Vmin=...;
Vmax=...;
NV=500;
%%V = vektor med verdier mellom Vmin og Vmax, i alt NV verdier
V=linspace(Vmin,Vmax,NV);
%%Verdier for a og b for R134a: ... , ...
R=8.314;
a=...;
b=...;
```

```
%%van der Waals tilstandsligning
p = R*T./(V-b) - a./(V.*V);
%%Plott av isotermen
fig = plot(V*1e3,p*1e-5);
%%Tekst paa figuren
title('Isoterm: van der Waals gass','fontsize',18);
xlabel('Molart volum (L)','fontsize',18);
ylabel('Trykk (bar)','fontsize',18);

%%HERFRA maa du selv skrive koden. Noen tips:
%%Elementer som maa eller boer inngaa, inklusive eksempler:
%%
%%A Innlesing av gjetning paa vaeskevolumet V1:
%%
%%Eksempel som ber om innlesing av tallverdi for stoerrelsen x1:
%%x1 = input('Les inn tallverdi for x1:  ')
%%
%%B Bestem trykkverdi P1 som tilsvarer volumet V1:
%%
%%Eksempel som bestemmer y1 = x1 - x1^3:
%%y1 = x1 - x1^3;
%%
%%C Bestem tilsvarende gassvolum V2. Tips: Velg en startverdi V_stor som
%%ut fra kurven p(V) helt sikkert er stoerre enn V2, reduser V_stor
%%med en passende skrittlengde dV, og bruk en while (...) ... end loekke
%%til aa lokalisere V2 slik at P2 = p(V2) er (omtrent) lik P1.
%%
%%Eksempel som lokaliserer x2 slik at y2 = x2 - x2^3 er (omtrent) lik
%%y1 = x1 - x1^3 med verdien av x1 et sted mellom -1.15 og -0.58 (valgt
%%slik at vi faktisk har en loesning y2 = y1):
%%x_stor = 1.5;
%%dx = 0.1; %%skrittlengde
%%y_find = x_stor - x_stor^3;
%%while (y_find < y1)
%%      x_stor = x_stor - dx;
%%      y_find = x_stor - x_stor^3;
%%end;
%%x2 = x_stor;
%%y2 = y_find;
%%
%%D Integrer p(V) numerisk fra V1 til V2, og sammenlign med P1*(V2-V1),
%%dvs arealet under den horisontale linjen.
%%
%%Eksempel som integrerer funksjonen
%%y = x - x^3 fra x1 til x2 med saakalt adaptiv Simpson-kvadratur, og
%%regner ut forskjellen mellom integralet I og arealet y1*(x2-x1) under
%%den horisontale linjen:
%%y = @(x) (x - x^3);
%%I = quad(y, x1, x2);
%%avvik = I - y1*(x2-x1);
```

```
%%
%%E Tegn inn linjen mellom (P1,V1) og (P2,V2). Skriv ut beregnede stoerrelser.
%%
%%Eksempel:
%%plot([x1,x2],[y1,y2]);
%%disp('x1,x2: ')
%%disp(x1)
%%disp(x2)
%%disp('y1,y2: ')
%%disp(y1)
%%disp(y2)
%%disp('Avvik mellom areal under kurve og rektangel: ')
%%disp(avvik)
%%
%%Endre x1, og eventuelt ogsaa x_stor og dx, og kjoer paa nytt,
%%for aa oppnaa et mindre avvik.
```

Et par kommentarer:

- Merk at skrittlengden dV setter en begrensning på hvor nøyaktig koeksistenstrykket p_{co} kan bestemmes.
- Merk at fortegnet på avviket mellom arealet under kurven $p(V)$ og rektanglet kan brukes til å avgjøre om gjetningen på V_v skal justeres opp eller ned.
- Den iterative prosessen for lokalisering av V_v, og dermed p_{co}, kan gjøres manuelt eller automatisk. I algoritmen og eksemplet ovenfor er det skissert en manuell løsning, der programmet simpelthen kjøres på nytt og ny verdi for $V1$ leses inn fra "kommandolinjen".