

Ordning och oordning i fysikaliska system, SEE065

# Energiomvandlingsprocesser

Numeriskt projekt VT 2021

Frågor: [therese.karmstrand@chalmers.se](mailto:therese.karmstrand@chalmers.se)

---

## Instruktioner

Uppgifterna skall lösas delvis numeriskt med hjälp av Python och delvis med analytiska räkningar för hand.

Lösningarna och resultaten skall redovisas med en skriftlig rapport skriven i L<sup>A</sup>T<sub>E</sub>X. I rapporten skall antaganden för lösningarna motiveras och använda ekvationer redovisas. Vidare skall samtliga frågor i uppgiftsformuleringen besvaras och alla plottar inkluderas. För att bli godkänd krävs även att en fungerande python-kod med förklarande kommentarer lämnas in.

Samarbete för att lösa uppgifterna uppmuntras men rapporten skall lämnas in individuellt.

Det erbjuds 2 handledningstillfällen där man kan ställa frågor om uppgifterna och programmeringen.

## Handledning

Räkneövning 8, 12e februari

Räkneövning 10, 19e februari

## Deadlines för inlämning

Ordinarie: 2021-02-26

Komplettering: 2021-03-12

---

## Introduktion

I detta projekt skall ni studera **energiförbrukningen** för ett typiskt svenskt småhus under **ett år** för att sedan göra en **energi- och exergi-analys** av **två** olika **uppvärmningsmetoder**; direkt **eluppvärmning** och **bergvärmepump**.

Projektet behandlar flera av kursens centrala koncept; värmeledning, värmeflöden, Carnotprocessen och exergi. Utöver att träna er i att applicera teori på praktiska problem med numeriska lösningar, syftar projektet till att ge en djupare förståelse för den icke-ideala Carnotprocessen samt begreppet exergi som ett mått på energikvalitet.

Relevanta kapitel i Beckman är kap 8, 11, 13, 14.

## Uppgifter

### 1) Värmeläckage

För att beräkna uppvärmningsbehovet behöver ni först uppskatta **värmeläckaget** från huset. Hastigheten med vilken huset utbyter värme med sin omgivning beror på **temperaturskillnaden** och är **1 MJ/K per timme**. Antag att inomhustemperaturen hålls vid konstant temperatur **22°C**.

Ladda ner filen '*smhi-opendata\_adapted.xlsx*' från kurshemsidan i Canvas. Filen innehåller meteorologiska data för dygnsmedeltemperaturen i Göteborg varje dag år 2019. Det underlättar om ni sparar filen i samma mapp som ert python-script.

- Skapa en plot som visar utomhustemperaturens variation över året. Beräkna även årsmedelvärdet.
- Använd väderdatat för att beräkna husets värmeläckage. Skapa en plot som visar hur värmeläckaget varierar över året.

### 2) Värmeledning

Värmeläckaget för huset beror på ett spontant värmeflödet från ett varmare området till ett kallare område. Hur stort värmeflödet är beror på temperaturdifferensen mellan de två regionerna och på värmeledningsförmågan hos materialen som skiljer regionerna åt, samt värmeövergångskoefficienten vid skiljeväggens in- respektiver ut-sida.

I svenska småhus byggs väggarna generellt med olika isoleringslager där väggens totala värmeledningsegenskaper anges med hjälp av värmeledningskoefficienten  $U$  W/m<sup>2</sup>K. För svenska trähus ligger ytterväggarnas  $U$ -värdet ofta på runt 0.15 – 0.20 W/m<sup>2</sup>K, men det finns stora variationer beroende på när de är byggda.

- Använd värden från tabell 8.1 s 127 i Beckman för att konstruera en vägg som har ett  $U$ -värde på ca 0.20. Din vägg skall ha träfasad och bestå av olika lager inklusive minst ett lager isolering med mineralull. Glöm inte konvektionens bidrag till väggens totala värmemotstånd.  
Redogör tydligt i rapporten vilka material som använts och vilken tjocklek de har. Illustrationer uppskattas.

- Gör en uppskattning av värmeläckaget för ett hus med dina väggar. Huset ska bestå av ett plan med arean  $122\text{ m}^2$  och ha takhöjden  $2.75\text{ m}$ . Låt huset vara rektangulärt till formen med måtten  $12.2\text{ m} \times 10\text{ m}$ .
- Använd informationen i tabell 1 till att förfinas din uppskattning av värmeläckaget genom att även inkludera effekter av värmeledning genom husgrunden, taket samt fönster och dörrar. Husets fönster och dörrar uppskattas uppta  $25\text{ m}^2$  av väggarean.
- Hur stor andel av det totala värmeläckaget,  $1\text{ MJ/K}$  per timme, motsvarar din uppskattning? Var sker resterande värmeförluster?

Konstruktionsdel	U-värde $\text{W/m}^2\text{K}$
Platta på mark/källargolv	0.24
Tak	0.18
Fönster och dörrar	1.9

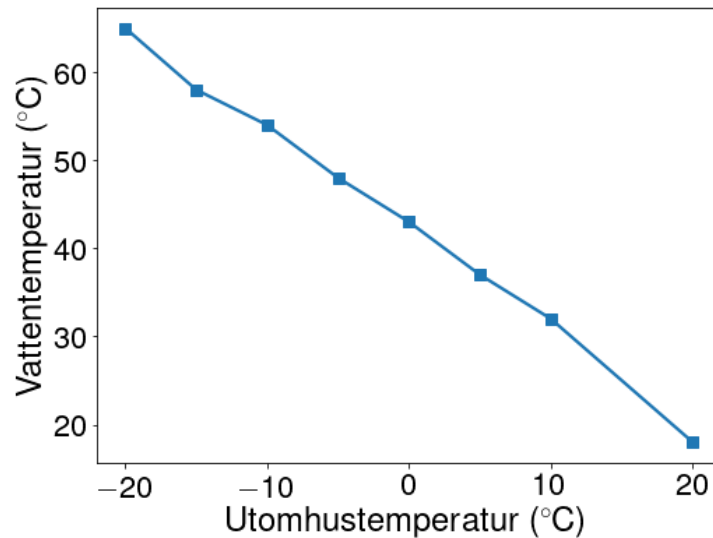
Tabell 1: Vanliga U-värden för olika delar av klimatskalet för svenska småhus. Värdena är tagna från en undersökning utförd av Boverket 2010.

### 3) Radiatorsystem

Låt oss anta att husets uppvärmning sker med hjälp av ett vattenburet radiatorsystem. Framledningstemperaturen till radiatorerna regleras av en termostat beroende på utomhustemperaturen. En typisk inställning för väderförhållandena i mellersta Sverige ges av värmekurvan i figur 1.

Ladda ner filen '*radiator\_Vkurva.xlsx*' från kurskansidan i Canvas. Filen innehåller datat som användes för att skapa värmekurvan i figur 1.

- Läs in data för värmekurvan och gör en linjär-kurvanpassning ( $y = kx + m$ ) för att ta fram en funktion för vattentemperaturen i radiatorerna som funktion av utomhustemperaturen.



Figur 1: Typisk radiatorkurva för väderförhållandena i södra Sverige.

#### 4) Värmefaktor

En vanlig uppvärmningmetod i Sverige är att använda sig av en värmepump.

I denna uppgift skall ni studera en bergvärmepump som använder berggrunden under huset som kallreservoar. Temperaturen i berggrunden är  $\sim 8^\circ\text{C}$  konstant. När utomhus-temperaturen överstiger  $20^\circ\text{C}$  stängs värmepumpen av.

- Utgå från uttrycket  $\varepsilon_v = \frac{Q_{ut}}{W}$  för värmefaktorn och använd kända relationer för en ideal och reversibel Carnotprocess för att visa relationen nedan.

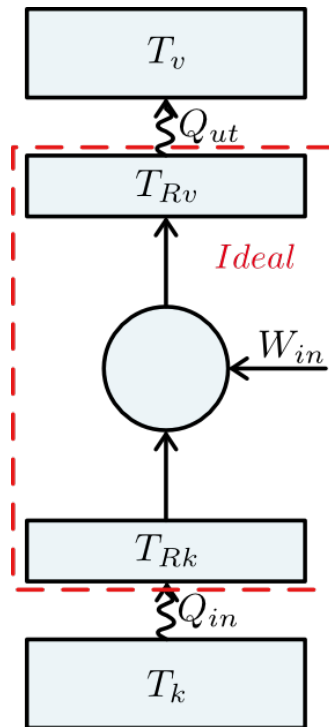
$$\varepsilon_v = \frac{T_v}{T_v - T_k} \quad \begin{cases} v = \text{varm} \\ k = \text{kall} \end{cases}$$

- Gör en graf som visar hur den ideala värmefaktorn varierar över året, samt beräkna årsmedelvärdet. Hur väl stämmer ert medelvärde överens med normala värden på värmefaktorn (COP) för värmepumpar?

I en mer realistisk modell för värmepumpen håller kylmediet en något högre temperatur  $T_{Rv}$  (R = refrigerant) i kondensatorn än temperaturen  $T_v$  på radiatorvattnet och en något lägre temperatur  $T_{Rk}$  i förångaren än temperaturen  $T_k$  i berggrunden.

En schematisk skiss över modellen, som är en omvänd *Curzon-Ahlborn*-värmemaskin, ges i figur 2. De kurvade pilarna indikerar värmeöverföring inklusive irreversibla värmeförluster. I övrigt innebär modellen en ideal Carnotprocess som arbetar mellan de nya temperaturerna  $T_{Rv}$  och  $T_{Rk}$ .

- Varför är det en mer realistisk modell?
- Vad blir uttrycket för värmefaktorn i detta fall? Är den större eller mindre än värmefaktorn i det ideala fallet? Klargör vilka antaganden du gör.
- Gör en graf som visar hur den icke-ideala värmefaktorn varierar över året om  $T_{Rv} = T_v + 20^\circ\text{C}$  och  $T_{Rk} = 0^\circ\text{C}$ . Hur är årsmedelvärdet i detta fall jämfört med normala COP värden?



Figur 2: Schematisk illustration av en omvänd *Curzon-Ahlborn*-värmemaskin som är en modell för en icke-ideal Carnotprocess.

## 5) Total elförbrukning

I många svenska villor används istället direktverkande elvärme där en elpanna värmer vattnet till husets vattenburna system.

- Beräkna husets årsförbrukning av el på grund av värmeförlusterna (uppgift 1) i de tre fallen; ideal värmepump, icke-ideal värmepump, direktverkande el. Svara i antal kWh och jämför med typiska värden för elförbrukningen hos svenska småhus. Diskutera eventuella skillnader.

## 6) Energi- och exergi-analys

I denna uppgift skall ni undersöka varför enbart en energianalys inte ger en fullständig bild av hur bra och effektivt ett termodynamiskt system presterar.

- Beräkna energi- och exergi- verkningsgraderna för de två fallen ideal bergvärmepump och direktverkande elvärme när utomhus temperaturen är  $10^{\circ}\text{C}$ . Som referenstemperatur använd samma temperatur som i berggrunden, det vill säga  $T_0 = 8^{\circ}\text{C}$ .
- Vad blir exergiverkningsgraden för den icke-ideala värmepumpen? Hur är den jämfört med exergiverkningsgraden för direktverkande elvärme?
- Fick ni några nya insikter av analysen?
- Genom att välja  $T_0 = T_k$  förenklar vi räkningarna i analysen. Reflektera kring andra för- och nackdelar med valet av referenstemperaturen. Hade det till exempel varit bättre med en annan referenstemperatur?

Tips: kapitel 14 i Beckman är användbart.

I fallet med den icke-ideala värmepumpen, tänk efter noga vilka temperaturer som används vid beräkningarna för exergin i värmereservoarerna samt för arbetet som används av värmepumpen.

$$\text{Energigiverkningsgrad: } \eta_{en} = \frac{E_{nyttig}}{E_{in}} \quad (1)$$

$$\text{Exergigiverkningsgrad: } \eta_{ex} = \frac{Ex_{ut}}{Ex_{in}} \quad (2)$$

$$\text{Exergi från värme: } Ex = Q\left(1 - \frac{T_0}{T}\right), \quad T > T_0 \quad (3)$$

**LYCKA TILL!**

# Python

Nedan följer tips som ni kan använda er av när ni skriver er Python-kod. Noter att vissa skillnader i hur man skriver finns beroende på om man använder någon mjukvara som t.ex. Spyder eller Jupyter Notebooks. Dessa skillnader behandlas ej i detta dokument.

Det är också bra att lära sig att läsa dokumentationen för en funktion för att förstå hur den fungerar. De flesta populära Python-bibliotek som numpy och matplotlib har webbsidor där man kan hitta dokumentation, tutorials och exempel. Vi rekommenderar er att titta på dessa sidor för att ta del av den fullständiga dokumentationen till funktionerna som presenteras nedan och/eller för att prova andra metoder.

I ert projekt ska ni klara er helt på att använda er av paketen **numpy** och **matplotlib** med webbsidorna:

- [numpy.org](http://numpy.org)
- [matplotlib.org](http://matplotlib.org)

## Preamble

På Canvas finns en fil *'preamble.py'*. Filen visar ett exempel på hur ni kan börja ert python-script. Förutom att visa hur ni importerar bibliotek och skriver funktioner, så finns också ett exempel på hur ni kan bestämma saker som textstorlek och figurstorlek mm när ni använder biblioteket matplotlib.pyplot för att skapa figurer.

## Arbeta med externt data

### Läsa in data från fil

För att läsa data från en text fil (.txt) kan man använda numpy-funktionen loadtxt. Det vanliga är att man döper alla funktioner 'np.funktionsnamn' i sitt lokala workspace när man importerar numpy. Detta görs genom att skriva;

```
1 import numpy as np
```

Om textfilen innehåller två kolumner med data, där säg kolumn ett är data för variabeln x och kolumn två är data för variabeln y. Då kan datat läsa in direkt genom att skriva;

```
1 data = np.loadtxt('filnamn.txt')
2 x_data = data[:,0] # all rows, first column
3 y_data = data[:,1] # all rows, second column
```

Variabeln data kommer då att vara en 2-dimensionell numpy array där första kolumnen är x-datat och andra kolumnen är y-datat.

Men, det är vanligt att datafiler kommer med några rader text som beskriver vad det är för data. Då kan man använda argumentet skiprows.

```
1 data = np.loadtxt('filnamn.txt', skiprows = 1) # don't read first row
2 x_data = data[:,0] # all rows, first column
3 y_data = data[:,1] # all rows, second column
```

Det finns också ett argument `usecols` som tillåter en att definiera vilka kolumner man vill spara. Tänk på att numreringen börjar på 0 så att `usecols = 1` innebär att man bara läser in den andra kolumnen i textfilen.

```
1 x_data = np.loadtxt('filnamn.txt', usecols = 0)
2 y_data = np.loadtxt('filnamn.txt', usecols = 1)
```

## Kurvanspassning

För att göra kurvanspassningarna med numpy finns funktionen `polyfit` som gör en polynomanspassning till något input data. `Polyfit` tar x- och y-data samt graden på polynomet man vill anpassa som input och ger koefficienterna till anpassningen som output. Så, om ni vill anpassa ett förstgrads polynom,

$$y = kx + m \quad (4)$$

då är x och y datat du ger som input och `polyfit` returnerar k och m som en numpy array.

```
1 fit_coeffs = np.polyfit(x_data, y_data, 1) #1 deg polynomial fit, y=kx+m
2 k = fit_coeffs[0]
3 m = fit_coeffs[1]
```

När du vet koefficienterna är det lätt att skriva en egen funktion som räknar ut  $y(x)$ .

## Plotta med `matplotlib.pyplot`

Precis som med numpy är det vanligt att man ger `matplotlib.pyplot` ett namn när man importerar paketet. Det går såklart att kalla paketet vad man vill, men det vanligaste namnet till `matplotlib.pyplot` är `plt`, då kallar man på funktionerna i paketet genom att skriva '`plt.funktionsnamn`'.

```
1 import matplotlib.pyplot as plt
```

## Skapa en figur

Det finns olika funktioner för att skapa plottar. En vanlig metod är att använda sig av `plt.subplots` som gör det enkelt att skapa figurer med flera plottar sida vid sida eller i ett rutnät.

Här kommer vi bara att visa hur ni skapar en enkel figur eftersom det skall räcka för ert projekt. Den som är intresserad av hur man skapar flera plottar i samma figur med `subplots` kan studera dessa [exempel](#).

De viktiga argumenten att kunna för `subplots` är `nrows`, `ncols` som är antal rader respektive kolumner och `figsize` som bestämmer storleken på figuren i inches; `plt.subplots(nrows,ncols,figsize=(width,height))`. Ingen av dessa argument är obligatoriska och anges inget kommer default-värden att användas.

Skapa en figur:

```
1 fig, ax = plt.subplots() # create figure with default parameters
```



eller

```
1 fig, ax = plt.subplots(1, 1) # create a figure with nrows = 1, ncols = 1 (
  same as defaults)
```

och ändra figurstorleken manuellt (anges i inches)

```
1 fig, ax = plt.subplots(1,1, figsize(width, height))
```

Objekten fig och ax är figuren (som python-objekt) respektive axlarna som ni plottar på i figuren. Om ni skapat en figur med flera plottar kommer ax att vara en array, då behöver ni specificera vilken axel ni vill plotta på. Återigen hänvisar vi till detta [exempel](#) om ni underar mer om detta, men det skall ej behövas för att kunna göra en bra rapport.

Nedan ges ett exempel på hur ni skapar en figur, plottar samt konfigurerar saker som figurtitel och labels på axlarna.

```
1 fig, ax = plt.subplots(1,1)
2 ax.plot(x_data, y_data)
3 ax.set_xlabel('parameter x-axel')
4 ax.set_ylabel(r'parameter y-axel med latex-kod $\lambda$')
5 # for latex(math-mode) add r before string indicator '' and put $$ around
  latex math expression
6 ax.set_title('Titel')
7 plt.show()
```

Förutom vilket data ni vill plotta kan ni även specificera saker som linjebredd (linewidth), linjetyp (solid, dashed, dotted etc.), linjefärg mm. För mer om detta hänvisar vi till dokumentationen till subplots.

Om ni plottar fler än en linje i samma plot är det bra att även skapa en legend som talar om vad som är vad. Detta gör du genom att först tilldela varje linje en label med hjälp av argumentet label när du anropar plot-funktionen och sedan skapa legend:en med funktionen `ax.legend()`. Hur det ser ut i kod illustreras nedan.

Kommandot `plt.show()` skrivs sist för att visa figuren. Notera att `plt.show()` visar alla figurer om ni har skapat flera figurer i ert script (tänk också på att ge era figurer olika variabelnamn, tex. `fig1`, `fig2`, ... om ni gör flera). Vill ni visa enbart t.ex. figur 1 med variabelnamn `fig1`, skriver ni `fig1.show()`.

### Exempel 1

```
1 fig, ax = plt.subplots(1,1)
2 # plot data
3 ax.plot(x_data, y_data, 'r--', linewidth = 7 , label = 'data 1' ) # very
  thick red and dashed line and legend label is 'data 1'
4 ax.set_xlabel('parameter x-axel')
5 ax.set_ylabel(r'parameter y-axel med latex-kod $\lambda$')
6 # for latex(math-mode) add r before string indicator '' and put $$ around
  latex math expression
7 ax.set_title('Titel')
8 # add a legend to the plot located in the upper right corner, labels will
  automatically be taken from the labels specified when plotting
9 ax.legend(loc = 'upper right')
10 plt.show()
```

### Exempel 2

```
1 fig, ax = plt.subplots(1,1)
2 # latex expression can also be used in data labels
3 ax.plot(x_data, y_data, ':', label = 'data 1', color = 'b', linewidth = 1 )
  # thin blue and dotted line, legend label is data 1
```

```

4 ax.legend(loc = 'center')
5 ax.set_xlabel('parameter x-axel')
6 ax.set_ylabel(r'parameter y-axel med latex-kod $\lambda$')
7 ax.set_title('Titel')
8 plt.show()

```

## Anpassa ticks och tick-labels

Ofta vill man manuellt bestämma antalet ticks och deras namn/värden för att göra informationen i plotten mer tydlig. För ert projekt kan det t.ex vara mer illustrativt att ha årets månader på x-axeln istället för dagnummer.

Börja med att skapa en numpy vektor med värdena på ticksen ni vill ha och en lista med era ticks-labels (namn/värden). Använd sedan `set_ticks` och `set_ticklabels` för att uppdatera de nya parametrarna.

```

1 months = np.linspace(0,365, 12)
2 months_labels = ['J','F','M','A','M','J','J','A','S','O','N','D']
3 ax.set_xticks(months, minor=False)
4 ax.set_xticklabels(months_labels)

```

## Spara en figur som png

När ni är klara med er figur vill ni såklart spara den så att ni sedan kan importera den till ert latex-dokument. Det går att spara i många olika filformat. I exemplet används formatet png som är lätt att importera i Latex. Filen sparas automatiskt i samma mapp som ni kompilerar i.

Exempel 1

```

1 fig_name = 'figur_namn.png'
2 # save figure defined as fig when you created it
3 fig.savefig(fig_name, bbox_inches='tight')
4 # bbox_inches='tight' removes excess white space around figure

```

## Andra användbara plot-kommandon

Två andra användbara kommandon är;

```

1 ax.set_ylim([ymin, ymax])

```

som tillåter er att bestämma största och minsta värde på y-axlen (byt ut ylim mot xlim för att skala x-axel).

```

1 _ = ax.ticklabel_format(axis="y", style="sci", scilimits=(0,0))

```

som ändrar formatet på axeln till 'scientific' vilket innebär att ticksen skrivs som en tiopotens.