

```

def complexmat(N, z0, z1):
    # skapar en N x N-matris med komplexa tal a + bi
    # där re(z0) <= a <= re(z1) och im(z1) <= b <= im(z0)
    # (jämnt fördelade i matrisen)

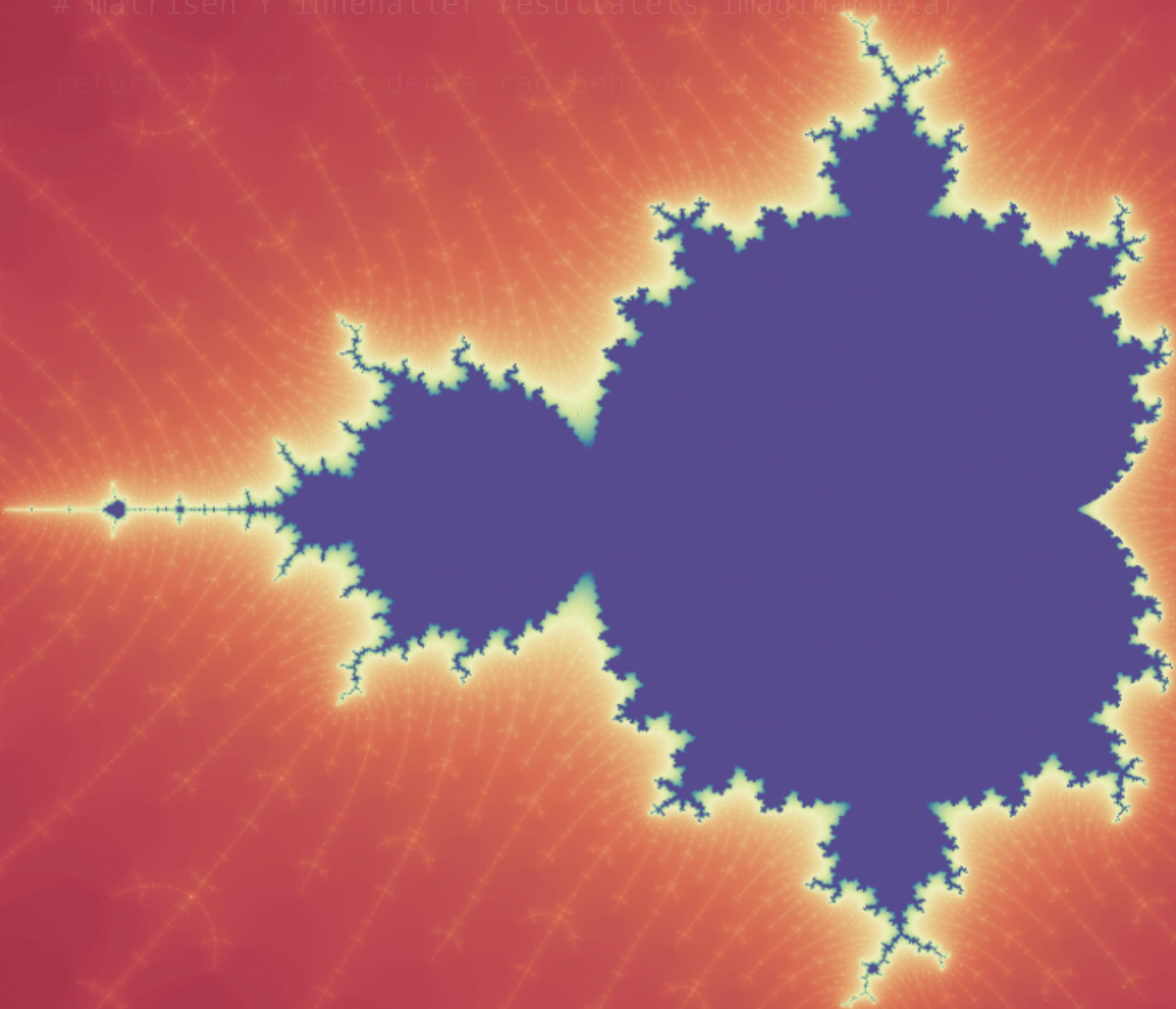
    xs = np.linspace(z0.real, z1.real, num = N) # realdelar
    ys = np.linspace(z0.imag, z1.imag, num = N) # imaginärdelar

    # skapa två matriser med real- respektive imaginärdelar
    [X, Y] = np.meshgrid(xs, ys)

    # matrisen X innehåller resultatets realdelar
    # matrisen Y innehåller resultatets imaginärdelar

    return X + 1j * Y # 1j är den rena behövs för komplexa tal

```



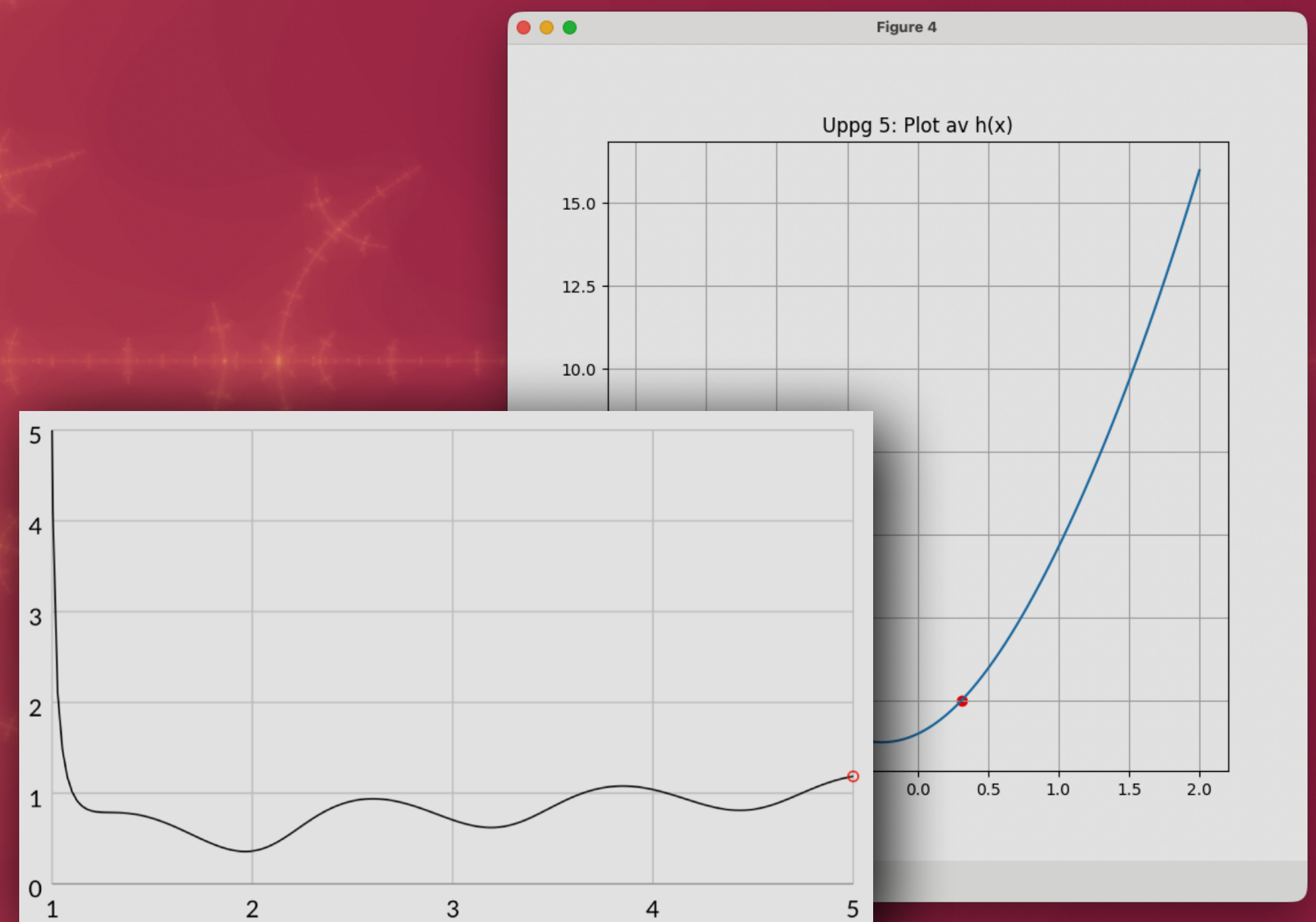
$$3\alpha + 5\beta + 6\gamma = -39$$

$$5\alpha + 15\beta - 10\gamma = -345$$

$$10\alpha + 20\beta + 15\gamma = -240$$

## Beräkningsprogrammering (numpy)

EDAA85 PROGRAMMERINGSTEKNIK, GRUNDKURS 2026 | PATRIK PERSSON



# Varför denna delkurs?

- Under din utbildning kommer du ofta att göra beräkningar...
- ... men kalkylatorappen i datorn är lite klumpig för större beräkningar
- Syfte med beräkningsprogrammeringen:
  - Du får ett verktyg som du kan använda i din utbildning
  - Du lär dig använda numpy som praktiskt verktyg för tekniska beräkningar
  - Du lär dig kombinera beräkningar med Python-programmering

demo!

# Beräkningsprogrammering med numpy

Python har ett antal praktiska paket (numpy, scipy, matplotlib) för att enkelt utföra beräkningar, t.ex.:

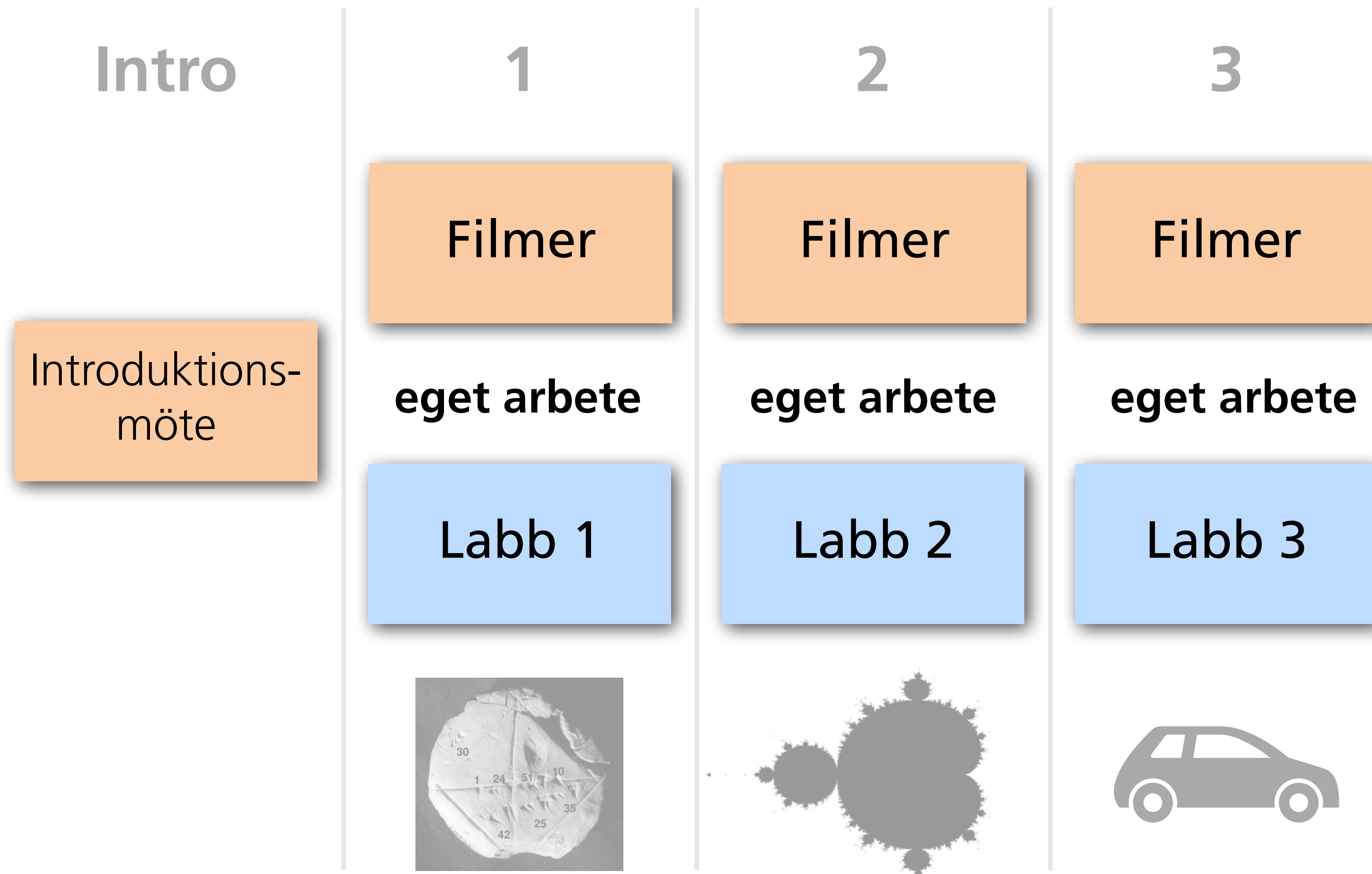
- plotta grafer
- lösa ekvationssystem och differentialekvationer
- beräkna integraler
- anpassa kurvor
- ...

Matris- och vektoroperationer: kan räkna med stora mängder data samtidigt

Kan användas interaktivt – vi för en dialog med Python-tolken

Går utmärkt att kombinera med "vanlig" Python-programmering

# Upplägg: introduktion + tre moduler



# Upplägg våren 2026

	v5	v6	v7	v8	...	v10	...	v19
<b>F1, I1, M1</b>			Intro + Modul 1			Modul 2		Modul 3
<b>E1</b>	Intro + Modul 1	Modul 2		Modul 3				

# Laborationer: obligatoriska (3st)

## Webbaserade laborationer (länkar finns i Moodle)

Vissa uppgifter genereras; ibland finns ledtrådar och/eller facit

Ni genomför laborationen **i par om två studenter**, gärna i förväg

## För att bli godkänd:

visa handledaren att du kan använda Python/numpy för att lösa uppgifterna

**D9** Vi söker en matris



$$M = \begin{pmatrix} 4 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

# Tips: skriv ut resultat

Gör utskrifter med resultat från dina beräkningar.

Då blir det lätt att följa logiken i ditt program och redovisningen går smidigare.

```
### Uppgift 2 ###  
pi = 3.141592653589793  
roten ur 3 = 1.7320508075688772  
roten ur -3 = 1.7320508075688772j  
roten ur i = (0.7071067811865476+0.7071067811865475j)  
s^(pi * i) = (-1+1.2246467991473532e-16j)  
#####
```

# Ditt arbete i beräkningsprogrammeringen (numpy)

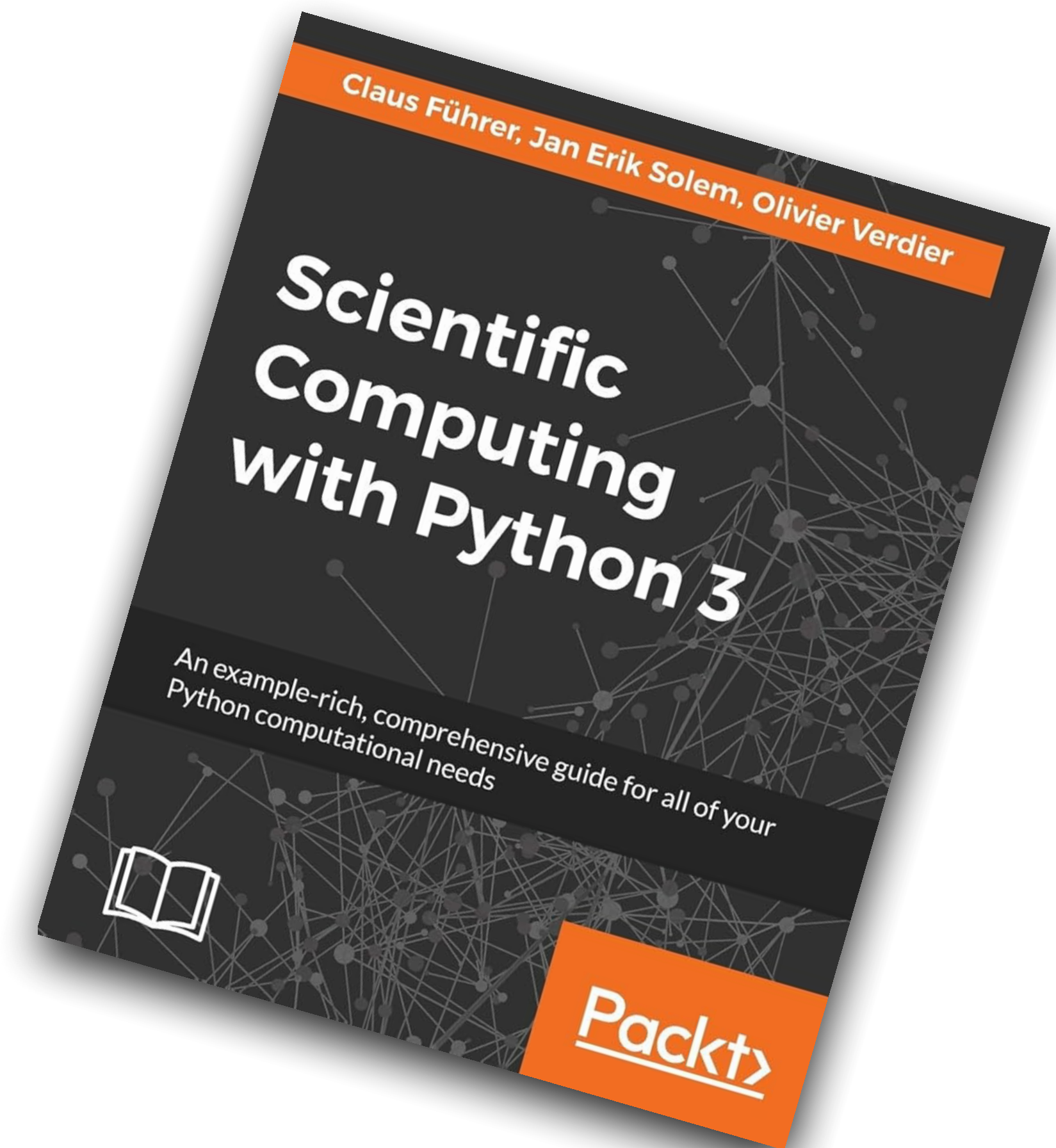
1,5hp = 40 timmar **självständigt arbete**

**Förbered och genomför** veckans numpy-laboration  
i grupper om två studenter!

**Redovisas på särskild labbtid**

**Material och hjälp:**

inspelade filmer,  
läsanvisningar (e-bok),  
särskilda resurstider för beräkningsprogrammering



# Redovisning av de tre numpy-laborationerna

- Redovisas på särskilda tillfällen för beräkningsprogrammering (kan **inte** redovisas på "vanliga" laborationer)
- Om du inte blir klar på laborationstillfället, redovisa på en **numpy**-resurstid senare (eller komplettera i LP4)
- Redovisning:
  - Visa era lösningar, svara på handledarens frågor, prata om vad ni gjort
  - Kanske får ni visa hur ni löser en (omgenererad) uppgift
  - Okej om ni kört fast på någon enstaka fråga:  
passa gärna på att själv ställa frågor till handledaren!
  - Om du glömt bort detaljer från matematiken så får du gärna fråga
- Ni jobbar i par om två studenter, men **båda svarar för hela lösningen**

# Att göra härnäst:

- Om du använder egen dator, installera då numpy m.m. **snarast** (installationsinstruktioner i Moodle)
- Det går utmärkt att använda E-husets Linuxdatorer också!  
(I så fall kan du behöva kontakta Patrik eller Nazila om din labbtid inte är i E-huset.)
- Laboration 1 redovisas på torsdag/fredag;  
använd gärna numpy-resurstiden på tisdag/onsdag för att få hjälp



**LUNDS**  
**UNIVERSITET**

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 2 * np.pi, 100)
print('x =', x)

y = np.sin(x)
print('y =', y)

plt.plot(x, y)
plt.show()
```