

Supplementary material

The aim of this course "Scientific Computing I" is to teach:

How to prepare a scientific document.

To achieve this, you need to learn the basics of

linux operating system

emacs text editor

L^AT_EX document preparation system

python programming language

You will also learn the basics of using the following databases

ADS (SAO/NASA Astrophysics Data System)

SIMBAD (SIMBAD Astronomical Database)

There are other operating systems, text editors, document preparation systems and programming languages, as well as databases. We choose the ones mentioned above, because they are widely used in natural sciences, like in Physics or Astronomy. Furthermore, they are all freely available open source products. Although no complicated, impressive and fancy stuff is presented, but the examples given here will get you started. The rest is up to yourself. The three simple principles are:

1. Principle: *Memorizing is not learning, but doing is.*
2. Principle: *Solving any problem once is enough, twice is too much.*
3. Principle: *Nearly all problems have already been solved, one only has to find out where?*

This first version of supplementary material contains the **python** programs mentioned during the lectures in autumn 2018. These programs are given in the same order as in the lectures. The **'additional programs'** are not mentioned in the lectures. They are completely voluntary material that will be discussed **only if** lecturing time allows it. Every program can be copied from the [www-link](#) marked with "👤".

Hello.py 🐍

```
#!/usr/bin/env python3
# Kommenttirivi: Tama on ohjelmani Hello.py
print("Hello_world!")
```

ForLooppiAlku.py 🐍

```
# Kommenttirivi: Tama on ohjelmani ForLooppiAlku.py
# Ohjauksen rivi paattyy merkiin ":",
# jota seuraa sisennys
for i in range(5):
    print('Hei_maaailma!')
for i in range(5):
    print(i)
```

PythonFunktioita.py 🐍

```
# Kommenttirivi: Tama on python ohjelmani PythonFunktioita.py
import os ; os.system('clear') # Tyhjennetaan naytto
import numpy as np # importoidaan koko numpy-moduli
# aletaan kayttaa "numpy" sijasta lyhennetta "np"
a=np.pi ; print(a) # Radiaanit trigonometrisille funktioille
b=np.sin(a) ; print(b) #-"-
c=np.cos(a) ; print(c) #-"-
d=np.tan(a) ; print(d) #-"-
e=1 ; print(e) #-"-
f=np.arcsin(e) ; print(f) #-"-
g=np.arccos(e) ; print(g) #-"-
h=np.arctan(e) ; print(h) #-"-
i=9 ; print(i) #
J=np.sqrt(i) ; print(J) # neliojuuri. Huom: iso "J" ei pieni "j",
# koska "j" symbolia kaytetaan kompleksiluvuille
k=1j*1j ; print(k) #-"-
l=complex(0,1)*complex(0,1) ; print(l)
m=1+1j ; print(m) #-"-
n=2+3j ; print(n) #-"-
o=m+n ; print(o) #-"-
p=abs(o) ; print(p) #
q=np.exp(1) ; print(q) # e
r=np.log(q) ; print(r) # ln
s=np.log10(q) ; print(s) # log
t=-3 ; print(t) #
u=abs(t) ; print(u) # itseisarvo
v=np.power(t,4.); print(v) # potenssi
```

[.Myalias](#)

```
alias c='clear'
alias cp='cp-i'
alias rm='rm-i'
alias python='python3'
```

[Tunnus.py](#)

```
# Tama on python ohjelmani Tunnus.py
import os # Tyhjennetaan naytto
os.system('clear') #-"-
a=[1,2]; b=a
print('Alku: a[0]=',a[0],', a[1]=',a[1])
print('Alku: b[0]=',b[0],', b[1]=',b[1])
a[1]='Ahaa' # Muuta a[1] arvo ja tyyppi
print('Loppu: a[0]=',a[0],', a[1]=',a[1])
print('Loppu: b[0]=',b[0],', b[1]=',b[1])
print('.....')
c=[1,2]; d=tuple(c)
print('Alku: c[0]=',c[0],', c[1]=',c[1])
print('Alku: d[0]=',d[0],', d[1]=',d[1])
c[1]='Ahaa' # Muuta c[1] arvo ja tyyppi
print('Loppu: c[0]=',c[0],', c[1]=',c[1])
print('Loppu: d[0]=',d[0],', d[1]=',d[1])
print('.....')
e=[1,2]; f=e[:]
print('Alku: e[0]=',e[0],', e[1]=',e[1])
print('Alku: f[0]=',f[0],', f[1]=',f[1])
e[1]='Ahaa' # Muuta e[1] arvo ja tyyppi
print('Loppu: e[0]=',e[0],', e[1]=',e[1])
print('Loppu: f[0]=',f[0],', f[1]=',f[1])
```

[Ifmalli1.py](#)

```
# Komenttirivi: Tama on python ohjelmani Ifmalli1.py
import os ; os.system('clear') # Tyhjennetaan naytto
a=2 ; b=1 ; c=2 ; d=1 # 1. kombinaatio
#a=2 ; b=1 ; c=1 ; d=2 # 2. kombinaatio
#a=1 ; b=2 ; c=2 ; d=1 # 3. kombinaatio
#a=1 ; b=2 ; c=1 ; d=2 # 4. kombinaatio
print('a=',a,'b=',b,'c=',c,'d=',d) # Tulostetaan a,b,c,d
print('Kaydaan aina rivilla AAA') #
# Ohjausrakenteen aloitusrivit paattyvat merkkiin ":"
# Muut ohjausrakenteen rivit sisennetty
if(a > b): # 1. ohjausrakenne alkaa
    print('Koska a>b, kaydaan rivilla BBB')
    print('Tarkistus: a=',a,'b=',b)
if(c > d): # 2. ohjausrakenne alkaa
    print('Koska a>b ja c>d, kaydaan rivilla CCC')
    print('Tarkistus: a=',a,'b=',b,'c=',c,'d=',d)
print('Koska a>b, kaydaan rivilla DDD')
print('Tarkistus: a=',a,'b=',b)
```

[Ifmalli2.py](#)

```
# Komenttirivi: Tama on python ohjelmani Ifmalli2.py
import os ; os.system('clear')           # Tyhjennetaan naytto
import numpy as np                       # numpy moduli
import random as ra                      # random moduli
x=ra.uniform(0,1)                        # Tasajakauma 0-1
print('Arvottu_luku_',x)                 # Tarkistus
if (x<=1./3.):                           # Alussa?
    print("on_alueen_(0,1)_alussa")      # -"-
elif (x>1./3. and x<=2./3.):             # Keskella?
    print("on_alueen_(0,1)_keskella")    # -"-
else:                                     # Lopussa?
    print("on_alueen_(0,1)_lopussa")    # -"-
```

[Inputmalli1.py](#)

```
#=====
# - Tama ohjelmani Inputmalli1.py on malli "input" komennon
# kaytosta. Se lukee annetun syotteen stringina
nimi = input('Kerro_nimesi_')
print('Hauska_tutustua', nimi)
ika = input('Kerro_ikasi_vuosissa_')
print('Olet_siis', nimi, 'ja_olet', ika, 'vuotta_vanha.')
```

ika1 = eval(ika) # Tapa 1: Muuntaa stringin ika numeroarvoksi
*print('Olet_siis', ika1*12, 'noin_kuukautta_vanha.')*
ika2 = float(ika) # Tapa 2: Muuntaa stringin ika numeroarvoksi
*print('Seka_suurin_piiirtein_', ika2*365, 'paivaa_vanha.')*

```
#=====
```

Funktiomalli.py

```
# Tama on ohjelmani Funktiomalli.py,  
# jossa on kaksi omaa aliohjelmaa  
def Kaksisanaa(sana1, sana2):  
    tx=sana1 + ' ja ' + sana2  
    return tx  
def Kolmaspotenssi(x):  
    y=x**3.0  
    return y  
# Paa-ohjelma alkaa  
# -----  
a='kissat' ; b='koirat'  
c=Kaksisanaa(a,b) ; print(c)  
a='puurot' ; b='velli'  
c=Kaksisanaa(a,b) ; print(c)  
a=20.  
b=Kolmaspotenssi(a) ; print(b)
```

ForLooppi2.py

```
# Kommenttirivi: Tama on python ohjelmani ForLooppi2.py  
import os ; os.system('clear') # Tyhjennetaan naytto  
for i in range(3): # 1. Looppi: indeksilla  
    print('i=', i) #  
a=['A', 'B', 'C'] #  
for b in a: # 2. Looppi alkioilla  
    print('b=', b) #  
    print(b.find("A")) #  
k=0 # Kahdessa sisakkaisessa loopissa muuttuva luku  
for j in range(2): # 3. Looppi  
    for b in a: # 4. Looppi  
        print('b=', b, ', k=', k) # 4. Looppi: print  
        k=k+1 # 4. Looppi: k=k+1  
    print('j=', j, ', k=', k) # 3. Looppi: print  
    k=k+1 # 3. Looppi: k=k+1
```

WhileMalli1.py

```
# Tama on python ohjelmani WhileMalli1.py
import os           # Import operation system
os.system('clear') # Tyhjennetaan naytto
i=0                # i:n alkuarvo
while (i <=9):     # Ohjausrakenne
    print(i)       # Lohkon sisennys
    i=i+2          # -"-
```

WhileMalli2.py

```
# Tama on python ohjelmani WhileMalli2.py
import os ; os.system('clear') # Tyhjennetaan naytto
print('Kokeile arvoja k=-11, -9 ja 20') #
k=input("Anna luku k=") # Luetaan k tekstina
k=float(k) # k muutetaan luvuksi
print('Alussa k=',k) # k arvo alussa
while (k >=10): # Ohjausrakenne
    print('1. ehdolla jatketään ja k=',k) # 1. sisennys
    k=k-4 # Vahenna 4
print('1. while jälkeen k=',k) #
while (k > -10.5 and k < 10): # Ohjausrakenne
    print('2. ehdolla jatketään ja k=',k) # 2. sisennys
    k=k-3 # Vahenna 2
print('2. while jälkeen k=',k) #
```

AliOhjelmamalli1.py

```
# Tama on ohjelmani
# AliOhjelmamalli1.py
# Aliohjelma alkaa =====
def puolita(a):
    puolet=a/2.0
    return a, puolet
# Aliohjelma loppuu ===
b=4.0
c,d=puolita(b)
print('Input: b=', b)
print('Output: c=', c)
print('Output: d=', d)
```

AliOhjelmamalli2.py

```
# Tama on ohjelmani
# AliOhjelmamalli2.py
# Aliohjelma alkaa =====
def puolita(a):
    puolet=a/2.0
    return a, puolet
# Aliohjelma loppuu ===
b=4.0
e=puolita(b) # Kokeilu 1
print('Input: b=', b)
print('Output: e=', e)
print('Output: e[0]=', e[0])
print('Output: e[1]=', e[1])
print('type(e)=', type(e))
```

AliOhjelmamalli3.py

```
# Tama on ohjelmani
# AliOhjelmamalli3.py
# Aliohjelma alkaa =====
def puolita(a):
    puolet=a/2.0
    a=a-9 # Kokeilu 2
    jotain=a/9. # Kokeilu 3
    return a, puolet
# Aliohjelma loppuu ===
b=4.0
c,d=puolita(b)
print('Input: b=', b)
print('Output: c=', c)
print('Output: d=', d)
print('jotain=', jotain)
```

[Summia.py](#)

```
# Tama on ohjelmani Summia.py
# -----
import numpy as np
n=input('Anna_n=_')
n=float(n) ;
k=1.+np.arange(n)
A=np.sum(k) # p3: 1. CE 333
B=n*(n+1.)/2.
print('n=',n,'A=',A,'B=',B)
C=np.sum(k**2) # p3: 2. CE 333
D=n*(n+1.)*(2*n+1.)/6.
print('n=',n,'C=',C,'D=',D)
x=input('Anna_x=_')
x=float(x)
E=np.sum((np.sin(k*x))**2.) # p37: 1. AD (361.3)
ny1=n/2.
ny2=np.cos((n+1.)*x)*np.sin(n*x)
ny3=2.*np.sin(x)
F=ny1-ny2/ny3
print('n=',n,'x=',x,'E=',E,'F=',F)
```

[RayleighAliOhjelma1.py](#)

```
# Komenttirivi: Tama on python ohjelmani RayleighAliOhjelma1.py
import numpy as np # numpy importoitu
def rayleightest(t, f): # Funktio alkaa
    pi=np.pi; theta=2.0*pi*f*t ; n=len(t)
    z=(1.0/n)*(sum(np.cos(theta))**2+sum(np.sin(theta))**2)
    return n, z
# -----
import os ; os.system('clear') # Tyhjennetaan näyttö
t=10.*np.random.sample(10) # Satunnaisajat 0 < t_i < 10 (n=10)
f=1. # Frekvenssi on f=1.
n,z=rayleightest(t, f) # Aliohjelman soveltaminen
print('n=',n) # n
print('z=_',z) # z
```


[Stringmalli1.py](#)

```
# Komenttirivi: Tama on python ohjelmani Stringmalli1.py
# - Joitakin esimerkkeja stringien manipuloinnista
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
a="karhunkierros" ; b=a.find("karhu")    # Etsi teksti
print('b_=',b)                          # Tulos=====
d='AbCd' ; e=d.upper()                  # Isoiksi kirjaimiksi
print('d_=',d,'e=',e)                 # Tulos=====
f='EfGh' ; g=f.lower()                  # Pieniksi kirjaimiksi
print('f_=',f,'g=',g)                 # Tulos=====
j='xku_xnkkx' ; k=j.replace('x','a')     # Korvaa 'x' merkillä 'a'
print('j_=',j,'k=',k)                 # Tulos=====
l='ABNyt_OKABAB' ; m=l.strip('AB')      # Poista merkit 'AB'
print('l_=',l,'m=',m)                 # Tulos=====
o='x;h;8;9' ; p=o.split(';')           # Jaa erottimella '='
print('o_=',o,'p=',p)                 # Tulos=====
print('p[0]=' ,p[0])                    # p:n 1. elementti
```

[Formatointimalli1.py](#)

```
# Komenttirivi: Tama on python ohjelmani Formatointimalli1.py
# - Joitakin string muunnoksen formatointi esimerkkeja
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
# Reaaliluvun muunnos stringiksi=====
a=1.458                                  # Luku
b=str(a)                                  # Muunnos stringiksi
print('b=',b,'_type(b)=' ,type(b))      # Tarkistus
# Reaaliluvun formatointi: leveys=8, desimaalit=5=====
c=7.89                                    # Luku
d='%8.5f' %c                              # Formatointi
print('d=',d)                            # Tarkistus
# Kokonaisluvun formatointi: leveys=6=====
e=71                                       # Luku
f='%6i' %e                                  # Formatointi
print('f=',f)                             # Tarkistus
# Reaaliluvun ja kokonaisluvun yhdistetty formatointi=====
g=2.34 ; h=-912                            # Luvut
i='%4.1f%5i' %(g,h)                       # Formatointi
print('i=',i)                             # Tarkistus
# Reaaliluku, string ja kokonaisluku yhdistetty=====
j=9.84 ; k='tekstia' ; l=812              # Yhdistettavat
m='%6.0f%10s%7i' %(j,k,l)                 # Formatointi
print('m=',m)                             # Tarkistus
# Reaaliluku, string, kokonaisluku ja '\&' yhdistetty===
n=9.23 ; o=-87 ; p='\\\\' '.....#_Yhdistettavat
q='%8.3e & %3i %2s' %_%(n,o,p).....#_Formatointi
print('q=',q).....#_Tarkistus
```

[Formatointimalli2.py](#)

```
# Kommenttirivi: Tama on python ohjelmani Formatointimalli2.py
# - Joitakin print komennon formatointi esimerkkeja
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
# Esimerkkeja formatoidusta print komennosta=====
a=0.123456789 ; b='abcde' ; c=123456     # float , string , integer
print ("%6.2f"% (a))                     # 6 merkkia , 2 desimaali
print ("%12.5f"% (a))                   # 12 merkkia , 5 desimaalia
print ("%8s"% (b))                       # 12 merkkia
print ("%30s"% (b))                      # 30 merkkia
print ("%6i"% (c))                       # 6 merkkia
print ("%15i"% (c))                      # 15 merkkia
print ("%12.3f%10s%8i"% (a,b,c))        # Yhdistelma
print ("%12.3f_&_%10s_&_%8i"% (a,b,c))  # Lisatty kaksi " & "merkkia
```

[Mallidata1.dat](#)

```
1 A
2 B
3 C
```

[LueTiedosto1A.py](#)

```
# Kommenttirivi: Tama on ohjelmani LueTiedosto1A.py
import os ; os.system('clear')           # Tyhjennetaan naytto
luku=[] ; sana=[]                       # Luo kaksi tyhjaa muuttujaa
file = open('Mallidata1.dat', 'r')       # Avaa tiedosto Mallidata1.dat
rivi = file.readline()                   # LUETAAN RIVI KERRALLAAN
while (len(rivi) > 0):                   # Ei tyhja rivi
    osat=rivi.split()                    # Tyhjien mukaan osiksi
    lukunyt=osat[0]                       # 1. elementti
    sananyt=osat[1]                       # 2. elementti
    luku.append(float(lukunyt))            # Lisaa arvo muuttujaan luku
    sana.append(sananyt)                 # Lisaa arvo muuttujaan sana
    print(luku,sana)                     # Nayta nykyiset arvot
    rivi = file.readline()                # Lue seuraava rivi
file.close()                              # Sulje Mallidata1.dat
```

[LueTiedosto1B.py](#)

```
# Kommenttirivi: Tama on ohjelmani LueTiedosto1B.py
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
f = open('Mallidata1.dat', 'r')          # Avaa Mallidata1.dat
data = f.readlines()                     # LUETAAN RIVILISTANA
f.close()                                 # Sulje Mallidata1.dat
luku=[] ; sana=[]                        # Luo kaksi tyhjaa muuttujaa
for rivi in data:                          # Looppi muuttujan data riveille
    osat=rivi.split()                      # Tyhjien mukaan osiksi
    lukunyt=osat[0]                         # 1. elementti
    sananyt=osat[1]                         # 2. elementti
    luku.append(float(lukunyt))             # Lisaa arvo muuttujaan luku
    sana.append(sananyt)                   # Lisaa arvo muuttujaan sana
    print(luku,sana)                       # Nayta nykyiset arvot
```

[LueTiedosto1C.py](#)

```
# Kommenttirivi: tama on python ohjelmani LueTiedosto1C.py
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
import numpy as np                       # numpy importoitu
file='Mallidata1.dat'                   # Tiedoston nimi
luku=np.loadtxt(file ,dtype='float' ,usecols=(0,)) # Lue 1. kolumni
sana=np.genfromtxt(file ,dtype='str' ,usecols=(1,)) # Lue 2. kolumni
print(luku ,sana)                       # Nayta muuttujien arvot
print(np.cos(luku)) # Muuttuja luku=vektori. Siita voi laskea esim. cosinin
```

[Mallidata2.dat](#)

```
Mallidata2.dat
HJD      Year M D H M
2454395.96 2007 10 22 10 59
2454399.48 2007 10 25 23 35
2454403.01 2007 10 29 12 11
```

[LueTiedosto2A.py](#)

```
# Komenttirivi: Tama on ohjelmani LueTiedosto2A.py
import os ; os.system('clear')           # Tyhjennetaan naytto
date=[]                                  # Tyhja muuttuja
file = open('Mallidata2.dat' , 'r')      # Avaa tiedosto Mallidata2.dat
rivi = file.readline()                  # LUETAAN RIVI KERRALLAAN
while (len(rivi) > 0):                   # Ei tyhja rivi
    osat=rivi.split()                   # Tyhjen mukaan osiksi
    osa=osat[0]                          # 1 alkio
    q=osa.find("245")                   # Etsi "245" tekstia
    if (q>-1):                           # Jos loytyy ... ,
# String muutetaan desimaaliluvuksi kaskylla "float"
        date.append(float(osa))          # Lisaa arvo muuttujaan date
    print(date)                          # Nayta nykyinen date
    rivi = file.readline()               # Lue seuraava rivi
file.close()                             # Sulje Mallidata2.dat
print("date_=" ,date)                    # Tarkistus
print("type(date)=" ,type(date))         # Tarkistus
```

[LueTiedosto2B.py](#)

```
# Komenttirivi: Tama on python ohjelmani LueTiedosto2B.py
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
f = open('Mallidata2.dat' , 'r')         # Avaa Mallidata2.dat
data = f.readlines()                    # LUETAAN RIVILISTANA
f.close()                                # Sulje Mallidata2.dat
date = []                                 # Luo tyhja lista
for rivi in data:                        # Looppi datan riveille
    osat=rivi.split()                   # Tyhjen mukaan osiksi
    osa=osat[0]                          # 1 alkio
    q=osa.find("245")                   # Etsi "245" tekstia
    if (q>-1):                           # Jos loytyy ... ,
# String muutetaan desimaaliluvuksi kaskylla "float"
        date.append(float(osa))          # Lisaa arvo muuttujaan date
    print(date)                          # Nayta nykyinen date
print("date_=" ,date)                    # Tarkistus
print("type(date)=" ,type(date))         # Tarkistus
```

[LueTiedosto2C.py](#)

```
# Kommenttirivi: tama on ohjelmani LueTiedosto2C.py
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
import numpy as np                       # numpy importoitu
file='Mallidata2.dat'                   # Tiedoston nimi
# Komennolla "skiprows=2" jatetaan lukematta 2 ensimmaista rivia
date=np.loadtxt(file,skiprows=2,usecols=(0,)) # Lue 1. kolumni
print("date_=",date)                     # Tarkistus
print("type(date)=",type(date))         # Tarkistus
print(np.cos(date))                     # Tarkistus
```

[KirjoitaTiedosto1.py](#)

```
# Komenttirivi: Tama on ohjelmani KirjoitaTiedosto1.py
# - Kirjoitetaan tiedostoon rivi kerrallaan
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
import numpy                             # Importoidaan Numpy
x=np.arange(3.) ; print("x=",x)          # Luodaan 1 vektori
y=np.sin(x) ; print("y=",y)             # Luodaan 2 vektori
# -----
# - Kirjoitetaan vain lukuja
f=open('KirjoitaData1a.dat','w')         # Avaa tiedoston
for i in range(len(x)):                 # for-looppi
    f.write("%6.1f%8.4f\n" %(x[i],y[i])) # Kirjoittaa rivin
f.close()                               # Sulkee tiedoston
# -----
# - Kirjoitetaan lukuja ja stringeja esim: Latex kayttoon
f=open('KirjoitaData1b.dat','w')         # Avaa tiedoston
for i in range(len(x)):                 # for-looppi
    f.write("%6.1f%3s%8.4f%4s\n"%(x[i], '&',y[i], '\\\\'))
f.close()                               # Sulkee tiedoston
```

[KirjoitaTiedosto2.py](#)

```
# Komenttirivi: Tama on ohjelmani KirjoitaTiedosto2.py
# - Eras nopea tapa kirjoittaa kerralla kaikki data
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
import numpy as np                       # Importoidaan Numpy
x=np.arange(3.) ; print("x=",x)          # Luodaan 1 vektori
y=np.sin(x) ; print("y=",y)             # Luodaan 2 vektori
# -----
np.savetxt('KirjoitaData2a.dat',(x,y),fmt='%10.3f')
np.savetxt('KirjoitaData2b.dat',np.transpose((x,y)),fmt='%10.3f')
```

[KirjoitusVinkki.py](#)

```
# KirjoitusVinkki.py
# - ENSIN kokeilu "print(...)"
# - SITTEN kun onnistuu, "f.write(...)" ja "\n"
# -----
import os ; os.system('clear')           # Tyhjennetaan naytto
import numpy                             # Importoidaan Numpy
x=np.arange(3.) ; y=np.sin(x)
f=open('KirjoitusVinkki.tex','w')
for i in range(len(x)):
    print("%6.1f%3s%8.4f%4s"%(x[i], '&',y[i], '\\\\'))
    # f.write("%6.1f%3s%8.4f%4s\n"%(x[i], '&',y[i], '\\\\'))
f.close()
```

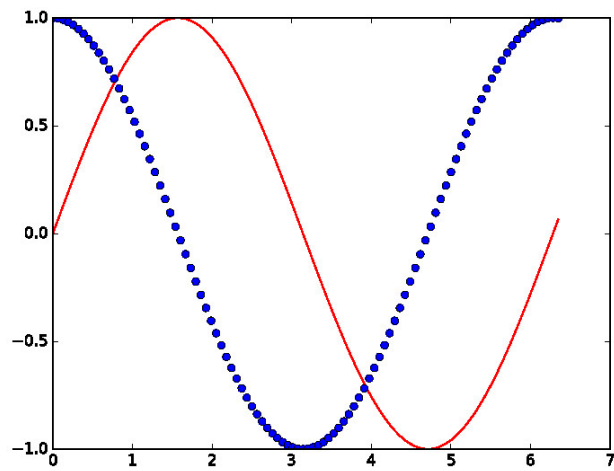


Figure 1: `Plottimalli1.py` produces this figure.

[Plottimalli1.py](#)

```

# -----
# Kommenttirivi: Tama on python ohjelmani Plottimalli1.py
# - Plottaa ensimmäisen kerran kuvan
# -----
import os ; os.system('clear')      # Tyhjennetaan näyttö
import numpy as np                  # numpy importoitu
import pylab as pl                  # pylab importoitu
x = np.pi*np.arange(100)/49.0     # x-vektori
y = np.sin(x)                       # y-vektori
z = np.cos(x)                       # z-vektori
pl.plot(x,y,'r')                    # x,y plotti
pl.plot(x,z,'ob')                   # x,z plotti
pl.savefig('Plotti1.eps')           # Tallenna eps-tiedostoon

```

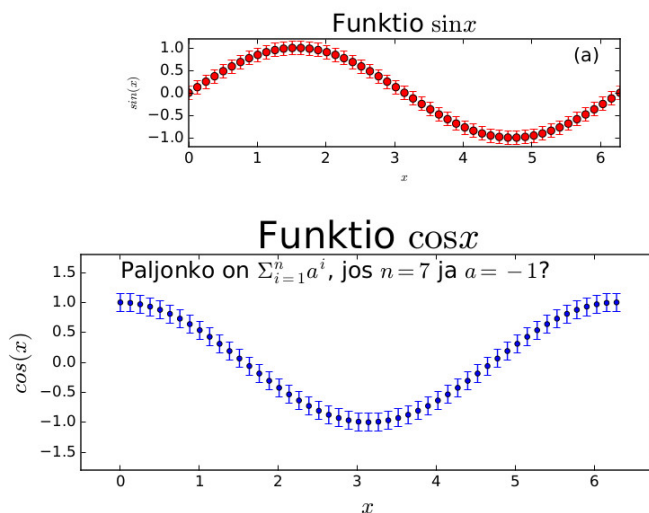


Figure 2: `Plottimalli2.py` produces this figure.

[Plottimalli2.py](#)

```

# Kommenttirivi: Tama on ohjelmani Plottimalli2.py
# -----
import os ; os.system('clear')           # Tyhjennetaan näyttö
import numpy as np                       # Importoi numpy
import pylab as pl                       # -"-      pylab
x = np.pi*np.arange(51)/25.0           # x-vektori
y = np.sin(x)                           # y-vektori
z = np.cos(x)                           # z-vektori
e = np.zeros(len(y))+0.15               # virhe
# -----
pl.axes([0.3,0.7,0.6,0.2])               # 1. plotin tila: pienempi
pl.xlim([0.0, 2.0 * np.pi])             # x rajat (vahan tilaa)
pl.ylim([-1.2, 1.2])                     # y rajat (vahan tilaa)
pl.errorbar(x,y,e,fmt='or')              # plottaa x,y,e
pl.title('Funktio_$$\sin{x}$$', fontsize=20,) # Paateksti
pl.xlabel('$x$', fontsize=10)            # x-teksti
pl.ylabel('$\sin(x)$', fontsize=10)      # y-teksti
pl.text(5.6,0.7, '(a)', fontsize=15)    # Kirjoitetaan jotain
# -----
pl.axes([0.15,0.1,0.80,0.4])             # 2. plotin tila: suurempi
pl.xlim([-0.5, 2.0 * np.pi +0.5])      # x rajat (paljon tilaa)
pl.ylim([-1.8, 1.8])                     # y rajat (paljon tilaa)
pl.errorbar(x,z,e,fmt='ob',ms=4)         # plottaa x,z,e
pl.title('Funktio_$$\cos{x}$$', fontsize=30) # Paateksti
pl.xlabel('$x$', fontsize=18)            # x-teksti
pl.ylabel('$\cos(x)$', fontsize=18)      # y-teksti
txt='Paljonko on_$$\Sigma_{i=1}^n a^i$$, jos_$$n=7$$_ja_$$a=-1$$?'
pl.text(0.0,1.4,txt, fontsize=18)        # Kirjoitetaan jotain
# -----
pl.savefig('Plotti2.pdf')                 # Kuva tiedosto

```

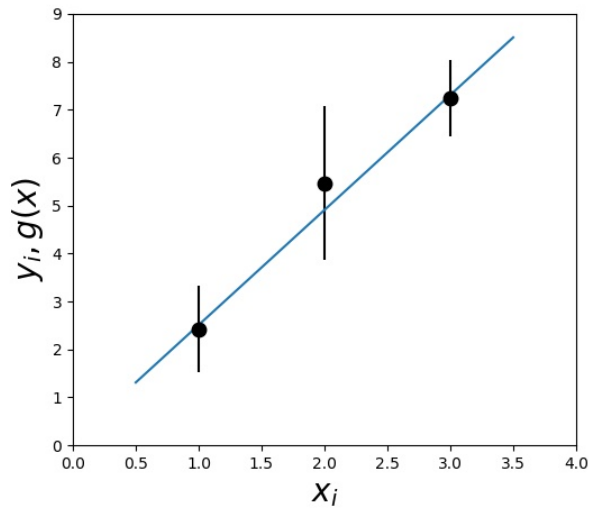


Figure 3: `PNSmalli1.py` produces this figure.

[PNSdata1.dat](#)

```
1 2.42 0.9
2 5.47 1.6
3 7.24 0.8
```

[PNSmalli1.py](#)

```
# -----
# Tama on ohjelmani PNSmalli1.py
# - Pienimman neliosumman sovitus
# -----
import os ; os.system('clear') # Tyhjennetaan
import numpy as np ; import pylab as pl # Importoidaan
from scipy.optimize import curve_fit # -"-
# -----
def funct(x,a,b):
    g=a+b*x
    return g
# -----
file='PNSdata1.dat' #Data
x=np.loadtxt(file,skiprows=0,usecols=(0,)) #Data: x
y=np.loadtxt(file,skiprows=0,usecols=(1,)) #Data: y
e=np.loadtxt(file,skiprows=0,usecols=(2,)) #Data: e
beta0=np.ones(2) ; print('beta0=',beta0) # Alkuarvo
beta1,covmatrix=curve_fit(funct,x,y,beta0,sigma=e)
print('beta1=',beta1) # Loppuarvo
pl.axes([0.15,0.15,0.70,0.80]) # paikka
pl.xlim([0,4]) ; pl.ylim([0,9]) # xy rajat
pl.errorbar(x,y,e,fmt='ok',markersize=9) # Data plot
xx=0.5+np.arange(21)/20.*3 # Mallin xx
g=beta1[0]+beta1[1]*xx ; pl.plot(xx,g) # Malli g(xx)
pl.xlabel('$x_i$',fontsize=20) # x-label
pl.ylabel('$y_i, g(x)$',fontsize=20) # y-label
pl.savefig('PNSmalli1.jpg') # Kuva&Nayta
```

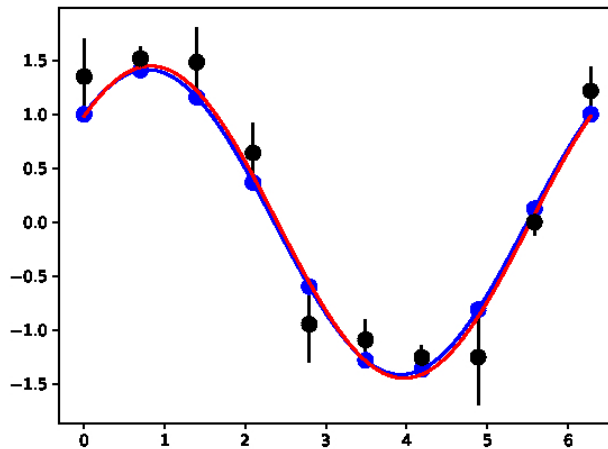


Figure 4: `PNSmalli2.py` produces this figure.

[PNSmalli2.py](#)

```

# -----
# Tämä on ohjelmani PNSmalli2.py
# - Pienimman neliosumman sovitus simuloituun dataan  $y=y(x)+e$ 
# -----
import os ; os.system('clear')           # Tyhjennetaan näyttö
import numpy as np ; import pylab as pl # Importoidaan
from scipy.optimize import curve_fit    # -----
# -----
def funct(x,a,b):
    g=a*np.cos(x)+b*np.sin(x)
    return g
# -----
n = input('Anna datan maara n= ') ; n=int(n)
s = input('Anna datan virhe s= ') ; s=float(s)
# -----
x=2*np.pi*np.arange(1.*n)/(n-1)        # Simuloidun datan x
beta0=np.ones(2)                         # Alkuarvo beta0
g0=beta0[0]*np.cos(x)+beta0[1]*np.sin(x) # Alkuperäinen malli
e=s*np.random.normal(0.,s,n)            # Gaussian error
y=g0+e                                   # Keinotekoisen datan y
beta1,covmatrix=curve_fit(funct,x,y,beta0,sigma=e) # PNS
print('Alku_beta0=',beta0) ; print('Loppu_beta1=',beta1)
xx=2*np.pi*(np.arange(101.)/100)       # Sovituksen xx
gg=beta0[0]*np.cos(xx)+beta0[1]*np.sin(xx) # Alkuperäinen malli
yy=beta1[0]*np.cos(xx)+beta1[1]*np.sin(xx) # Sovituksen malli
# -----
pl.plot(x,g0,'ob',markersize=9) # Alkuperäinen malli: siniset pisteet
pl.plot(xx,gg,'b',markersize=9) # Alkuperäinen malli: sininen viiva
pl.errorbar(x,y,e,fmt='ok',markersize=9) # Simuloitu data: mustat pisteet
pl.plot(xx,yy,'r') # Simuloituun sovitettu malli: punainen viiva
# -----
pl.savefig('PNSmalli2.eps')              # Kuva tiedosto

```



```

gg=          a0*np.cos(xx)+          b0*np.sin(xx)          # Simuloitu malli g0
# -----
pl.axes([z1[i],z2[i],lev,kor])          # MUUTTUU!!!...
x1=0.0          ; x2=2.0*np.pi          # x-rajat
pl.xlim([x1,x2])          # --
y1=g0.min()-3.*s0 ; y2=g0.max()+3.*s0          # y-rajat
pl.ylim([y1,y2])          # --
pl.plot(x0,g0,'ob') ; pl.plot(xx,gg,'b')          # plotit
pl.errorbar(x0,y0,e0,fmt='or')          # Simuloitu data
pl.plot(xx,yy,'r')          # Sim. data sovitus
pl.xticks(fontsize=8)          # x numerokoko
pl.yticks(fontsize=8)          # y numerokoko
# -----
txt1='%3s%3.0f%6s%5.2f' %('n=',n0,'_ja_s0=',s0)
pl.text(x1+0.0*(x2-x1),y1+1.36*(y2-y1),txt1,fontsize=10)
txt2='%14s%5.2f%6s%5.2f' %('Simuloitu:_A=',a0,'_ja_B=',b0)
pl.text(x1+0.0*(x2-x1),y1+1.20*(y2-y1),txt2,fontsize=10)
txt3='%14s%5.2f%6s%5.2f' %('Sovitettu:_A=',beta1[0],'_ja_B=',beta1[1])
pl.text(x1+0.0*(x2-x1),y1+1.04*(y2-y1),txt3,fontsize=10)
# -----
pl.savefig('PNSmalli3.eps')          # Kuva tiedosto

```

Additional programs

[Chooseinput.py](#)

```
# Chooseinput.py
# Combines Ifmalli1.py and Inputmalli1.py ideas
import os ; os.system('clear')
print('Combinatios are:')
print('1. combination: a=2; b=1; c=2; d=1')
print('2. combination: a=2; b=1; c=1; d=2')
print('3. combination: a=1; b=2; c=2; d=1')
print('4. combination: a=1; b=2; c=1; d=2')
combi=input('Which combination do you choose: 1, 2, 3 or 4 = ... ')
combi=eval(combi)
if (combi == 1):
    a=2 ; b=1 ; c=2 ; d=1
if (combi == 2):
    a=2 ; b=1 ; c=1 ; d=2
if (combi == 3):
    a=1 ; b=2 ; c=2 ; d=1
if (combi == 4):
    a=1 ; b=2 ; c=1 ; d=2
print('a=',a, 'b=',b, 'c=',c, 'd=',d)
print('We always visit THIS LINE A!')
if(a > b):
    print('Since a>b, we visit THIS LINE B!')
    print('Checking: a=',a, ', b=',b)
    if(c > d):
        print('Since a>b and c>d, we visit THIS LINE C!')
        print('Tarkistus: a=',a, 'b=',b, 'c=',c, 'd=',d)
    print('Since a>b, we visit THIS LINE D!')
    print('Checking: a=',a, ', b=',b)
```

ForAndWhile.py

```
# ForAndWhile.py
# -----
# for and while examples
import numpy as np
#
n1=3
print('-----Example_1A-----')
for i in range(n1):
    print(i)
print('-----Example_1B-----')
I=np.arange(n1)
for i in I:
    print(i)
print('-----Example_1C-----')
i=0
while (i<n1):
    print(i)
    i=i+1
n1=2 ; n2=5
print('-----Example_2A-----')
for i in range(n1, n2):
    print(i)
print('-----Example_2B-----')
I=n1+np.arange(n2-n1)
for i in I:
    print(i)
print('-----Example_2C-----')
i=n1
while (i<n2):
    print(i)
    i=i+1
n1=2 ; n2=11 ; n3=4
print('-----Example_3A-----')
for i in range(n1, n2, n3):
    print(i)
print('-----Example_3B-----')
I=n1+np.arange(3)*n3
for i in I:
    print(i)
print('-----Example_3C-----')
i=n1
while (i<n2):
    print(i)
    i=i+n3
```

[Trymalli.py](#)

```
# Komenttirivi: Tama on python ohjelmani Trymalli.py
import os ; os.system('clear') # Tyhjennetaan naytto
a = [1,2] # a[0]=1,a[1]=2 mutta alkiota a[2] ei ole!
#
# "print a[2]" komennon virheviesti olisi normaalisti
# "IndexError: list index out of range"
# Esimerkki 1 -----
try:                                # try alkaa
    print(a[2])                     # Komento yrittys
except:                              # Jos ei onnistu,
    print("Ei_onnistu.")           # printataan tama
# Esimerkki 2 -----
try:                                # try alkaa
    print(a[2])                     # Komento yrittys
except IndexError:                  # Jos virhetyyppi IndexError,
    print("Virhe_tunnistettu")     # niin printataan tama.
except:                              # Jos virhetyyppi ei IndexError,
    print("Virhe_tunnistamatta")   # niin printataan tama.
# Esimerkki 3 -----
try:                                # Ohjauskomento alkaa
    print(a[2])                     # Sisennys
except TypeError:                   # Jos virhetyyppi TypeError,
    print("Virhe_tunnistettu")     # niin printataan tama.
except:                              # Jos virhetyyppi ei TypeError,
    print("Virhe_tunnistamatta")   # niin printataan tama.
```

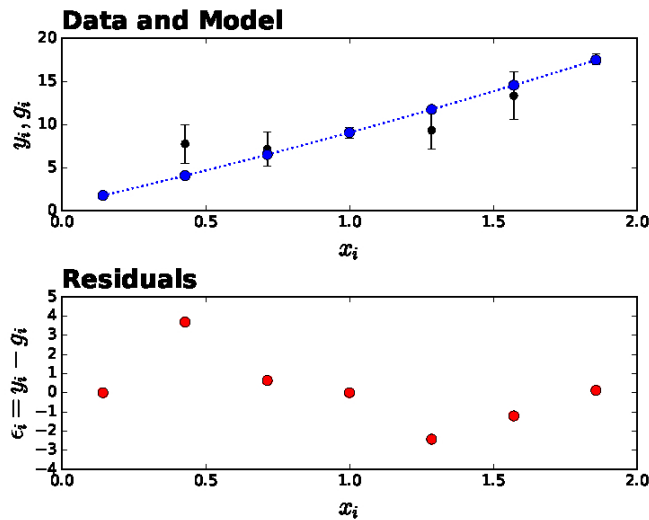


Figure 6: `PNSintro.py` produces this figure.

[PNSintro.py](#)

```

# -----
# Tämä on ohjelmani PNSintro.py
# – Pienimman neliosumman sovitus
# -----
import os
os.system('clear')
import numpy as np
import pylab as pl
from scipy.optimize import curve_fit
# -----
def funct(x, a0, a1, a2):
    g=a0+a1*x+a2*x**2.0
    return g
# -----
def myfit(x, y, e, beta0):
    betal, covmatrix=curve_fit(funct, x, y, beta0, sigma=e)
    return betal
# -----
def mymodel(x, beta):
    g=beta[0]+beta[1]*x+beta[2]*x**2.0
    return g
# -----
def myplot(x, y, e, g, epsilon, xx, gg):
    pl.axes([0.15, 0.60, 0.80, 0.32])
    x1=np.floor(np.min(x))
    x2=np.floor(np.max(x))+1.
    pl.xlim([x1, x2])
    y1=np.floor(np.min(y-e))-1.
    y2=np.floor(np.max(y+e))+2.
    pl.ylim([y1, y2])
    pl.errorbar(x, y, e, fmt='ok', markersize=6)
    pl.plot(x, g, 'ob', markersize=8.0)
    pl.plot(xx, gg, 'b')
    pl.xlabel('$x_i$', fontsize=20)
    pl.ylabel('$y_i, g_i$', fontsize=20)
    pl.text(x1+0.0*(x2-x1), y1+1.05*(y2-y1), \
            'Data_and_Model', fontsize=20, fontweight='bold')
    pl.axes([0.15, 0.12, 0.80, 0.32])

```

```

x1=np.floor(np.min(x))
x2=np.floor(np.max(x))+1.
pl.xlim([x1,x2])
y1=np.floor(np.min(epsilon))-1.
y2=np.floor(np.max(epsilon))+2.
pl.ylim([y1,y2])
pl.plot(x,epsilon,'or',markersize=8.0)
pl.xlabel('$x_i$',fontsize=20)
pl.ylabel('\epsilon_i=y_i-g_i$',fontsize=20)
pl.text(x1+0.0*(x2-x1),y1+1.05*(y2-y1),\
        'Residuals',fontsize=20,fontweight='bold')
pl.savefig('PNSintro.eps')
return

# -----
n=7
Dt=2.0
#x=Dt*np.random.uniform(0,1,n)
x=(1./n)+Dt*(np.arange(n)/n)
y=4.0+2.*x+3.*x**2.0
e=2.0
ny=np.random.normal(0.,e,np.size(x))
y=y+ny
e=np.random.normal(0.,e,np.size(x))
beta0=np.ones(3)
beta1=myfit(x,y,e,beta0)
print('beta1=',beta1)
g=mymodel(x,beta1)
epsilon=y-g
xx=np.min(x)+(np.arange(101)/100.)*(np.max(x)-np.min(x))
gg=mymodel(xx,beta1)
myplot(x,y,e,g,epsilon,xx,gg)

```

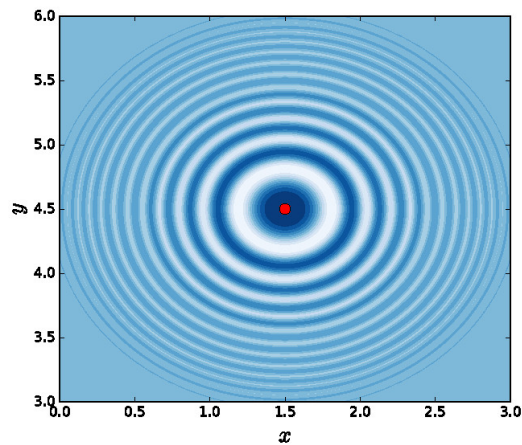


Figure 7: `tila2dim.py` produces this figure.

[tila2dim.py](#) 

```

# tila2dim.py
#=====
# - Two-dimensional plot
# - Keep input, computation and plotting apart
# -----
import numpy as np                                # For math
from numpy import unravel_index                   # For 2-dimensional ..
import matplotlib.pyplot as plt                  # For plots & "contourf"
# -----
def xybox(x,y):
    m=x.size ; xx=np.zeros(m*m).reshape(m,m) ; yy=np.zeros(m*m).reshape(m,m)
    for i in range(m):
        xx[i,:]=x
    for i in range(m):
        yy[:,i]=y
    return xx,yy
# -----
def zzcompute(a,b,c,d,x0,y0,f):
    x=a+(np.arange(1001)/1000.)*(b-a) ; y=c+(np.arange(1001)/1000.)*(d-c)
    xx,yy=xybox(x,y)
    r=(xx-x0)**2.+(yy-y0)**2.
    zz=(1./(r+1.))**2.*np.cos(2.*np.pi*f*r)
    kk=unravel_index(zz.argmax(),zz.shape)
    return xx,yy,zz,kk
# -----
def myplot(xx,yy,zz,kk):
    ny1=np.min(zz) ; ny2=np.max(zz) ; levels=ny1+np.arange(11)*(ny2-ny1)/10.
    plt.axes([0.15,0.10,0.70,0.80])
    plt.xlabel('$x$',fontsize=20)
    plt.ylabel('$y$',fontsize=20)
    c1=plt.contourf(xx,yy,zz,levels,origin='lower',cmap='Blues')
    c1.cmap.set_under('white')
    plt.plot(xx[kk],yy[kk], 'or', markersize=10)
    plt.show
    plt.savefig('tila2dim.eps')
# -----
a=0. ; b=3. ; c=3. ; d=6. ; x0=1.5 ; y0=4.5 ; f=5. # Input
xx,yy,zz,kk=zzcompute(a,b,c,d,x0,y0,f)           # Computation
myplot(xx,yy,zz,kk)                               # Plotting
# -----

```