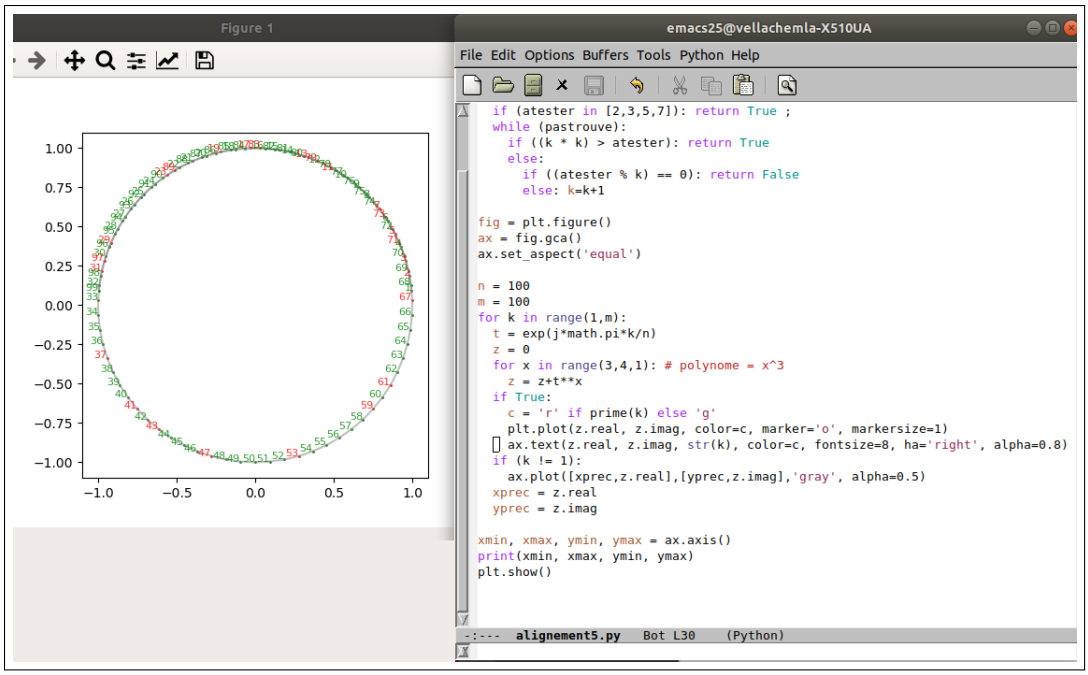
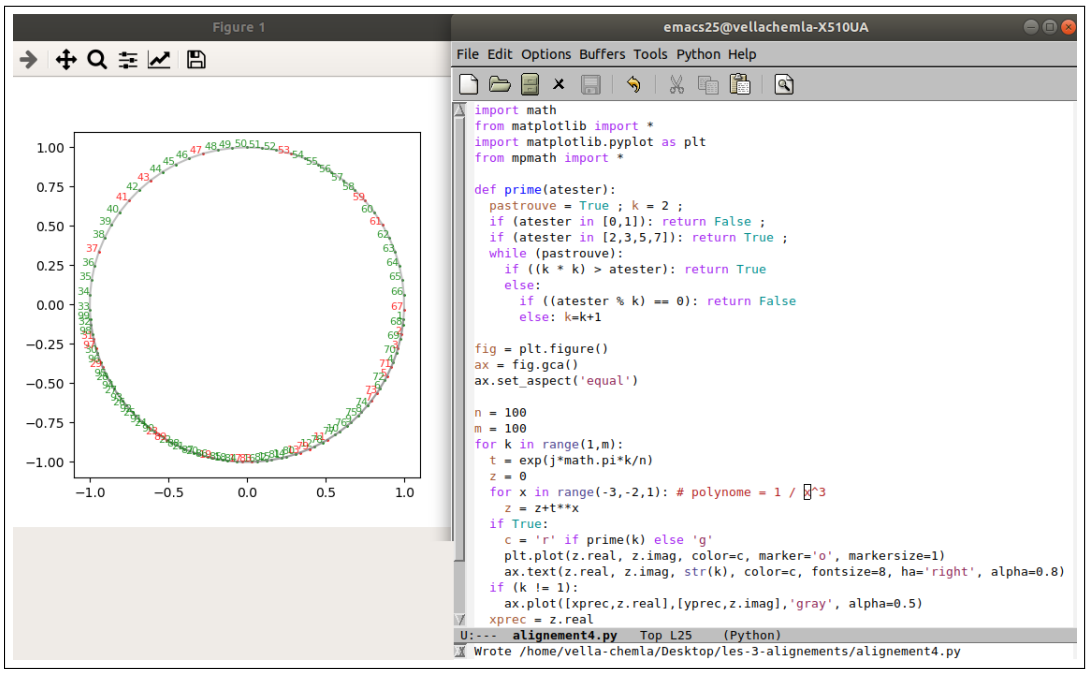
Inversions (Denise Vella-Chemla, 8.10.2020)





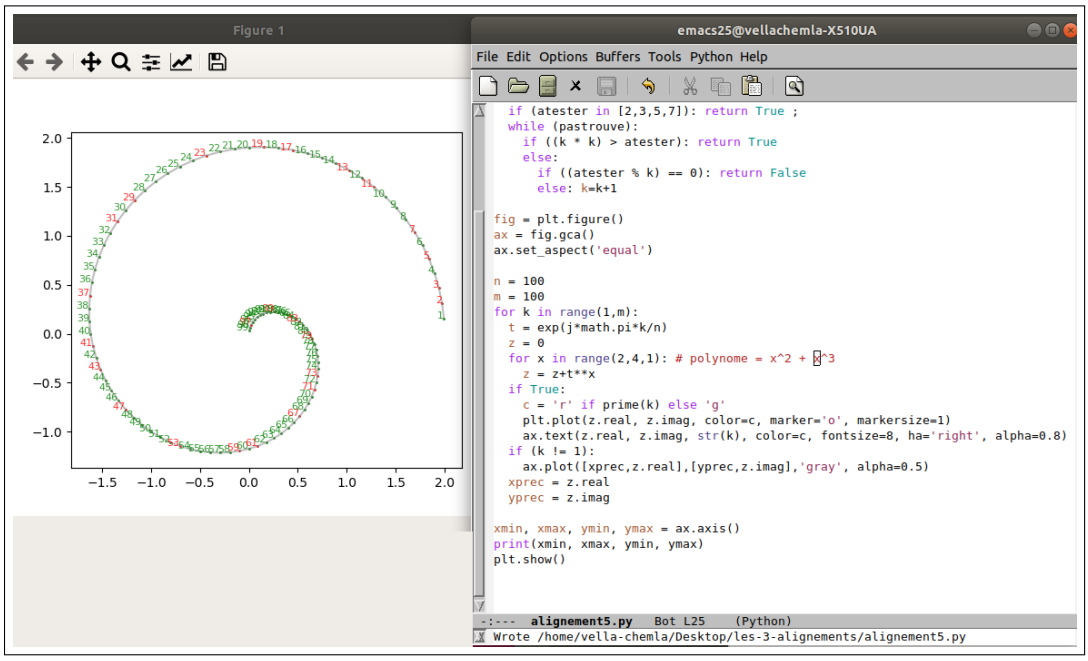


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(2,4,1): # polynome = x^2 + x^3
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
yprec = z.imag
xmin, xmax, ymin, ymax = ax.axis()
print(xmin, xmax, ymin, ymax)
plt.show()
-:--- alignement5.py Bot L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement5.py

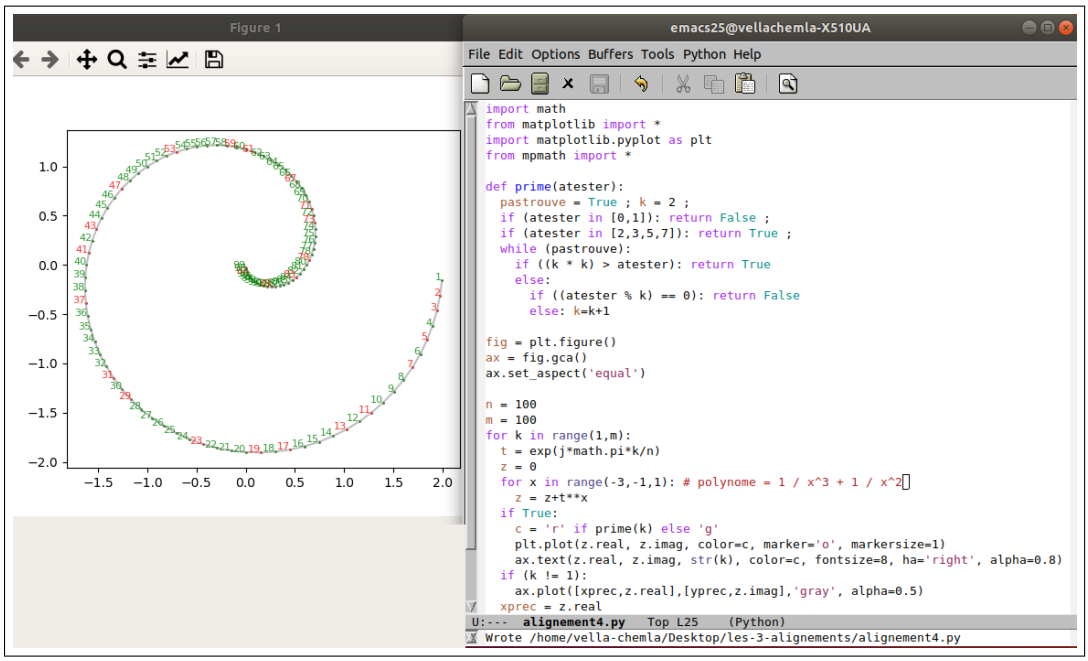


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
import math
from matplotlib import *
import matplotlib.pyplot as plt
from mpmath import *
def prime(atester):
pastrouve = True ; k = 2 ;
if (atester in [0,1]): return False ;
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(-3,-1,1): # polynome = 1 / x^3 + 1 / x^2
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
U:--- alignement4.py Top L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement4.py

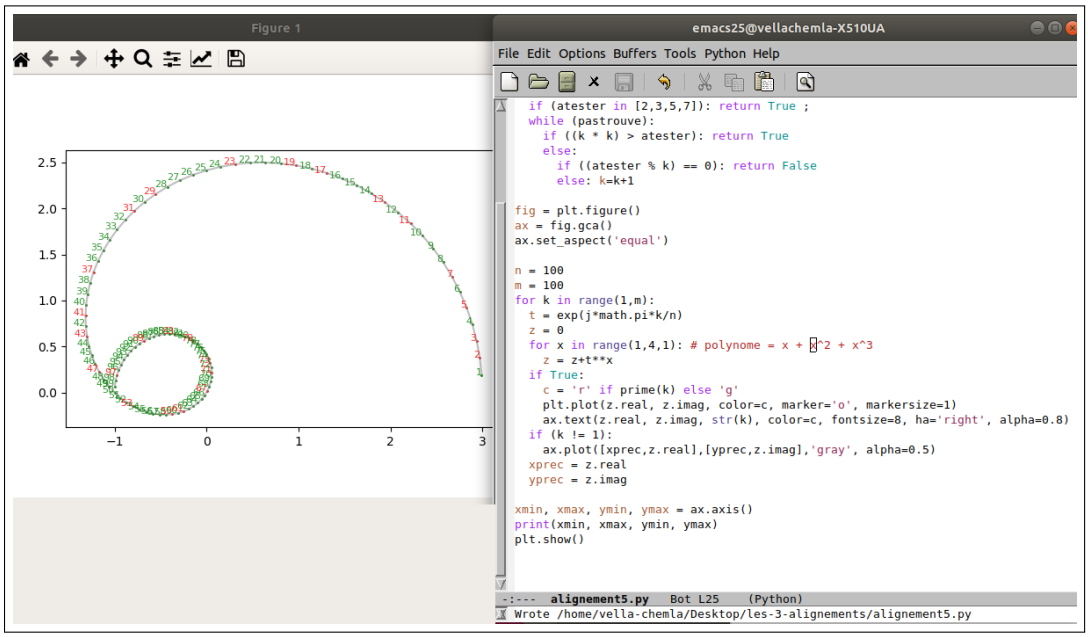


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(1,4,1): # polynome = x + x^2 + x^3
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
yprec = z.imag
xmin, xmax, ymin, ymax = ax.axis()
print(xmin, xmax, ymin, ymax)
plt.show()
-:--- alignement5.py Bot L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement5.py

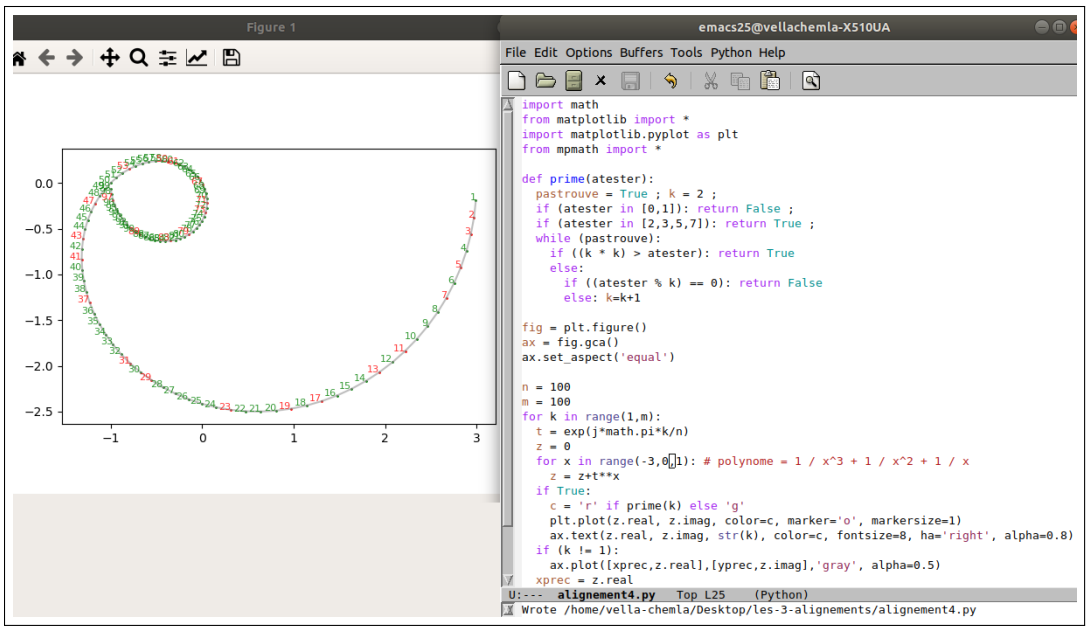


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
import math
from matplotlib import *
import matplotlib.pyplot as plt
from mpmath import *
def prime(atester):
pastrouve = True ; k = 2 ;
if (atester in [0,1]): return False ;
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(-3,0,1): # polynome = 1 / x^3 + 1 / x^2 + 1 / x
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
U:--- alignement4.py Top L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement4.py

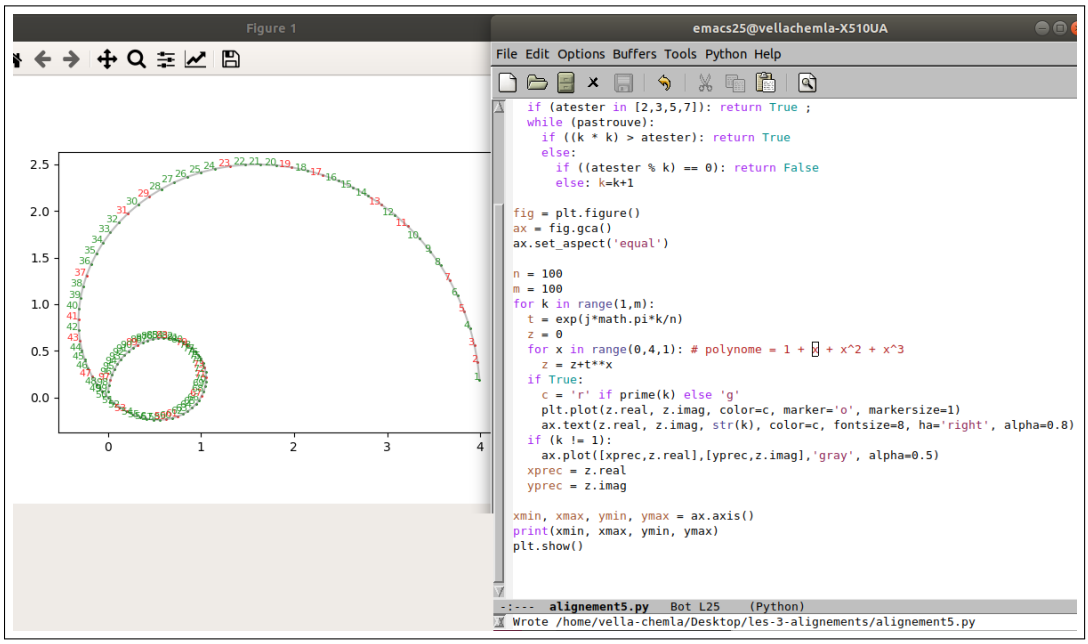


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(0,4,1): # polynome = 1 + x + x^2 + x^3
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
yprec = z.imag
xmin, xmax, ymin, ymax = ax.axis()
print(xmin, xmax, ymin, ymax)
plt.show()
-:--- alignement5.py Bot L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement5.py

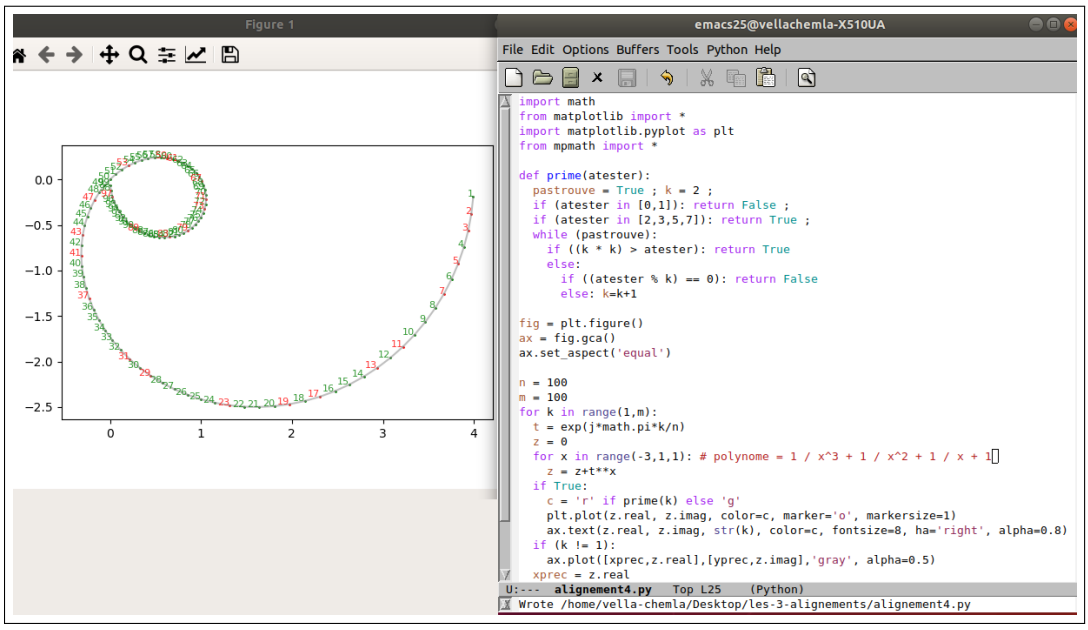


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
import math
from matplotlib import *
import matplotlib.pyplot as plt
from mpmath import *
def prime(atester):
pastrouve = True ; k = 2 ;
if (atester in [0,1]): return False ;
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(-3,1,1): # polynome = 1 / x^3 + 1 / x^2 + 1 / x + 1
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
U:--- alignement4.py Top L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement4.py

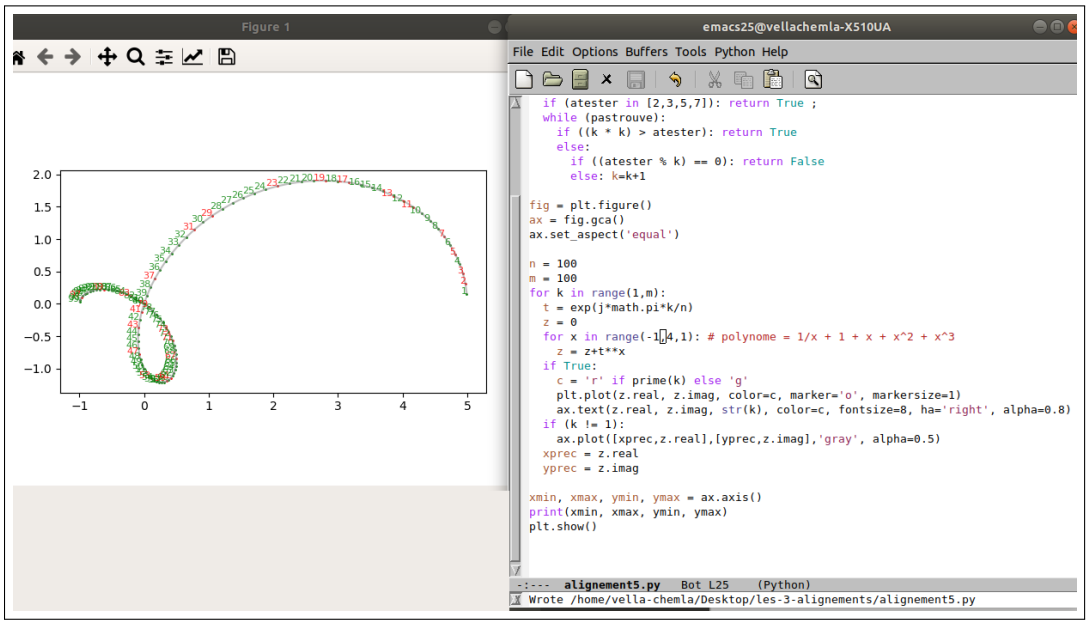


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(-1,4,1): # polynome = 1/x + 1 + x + x^2 + x^3
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
yprec = z.imag
xmin, xmax, ymin, ymax = ax.axis()
print(xmin, xmax, ymin, ymax)
plt.show()
-:--- alignement5.py Bot L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement5.py

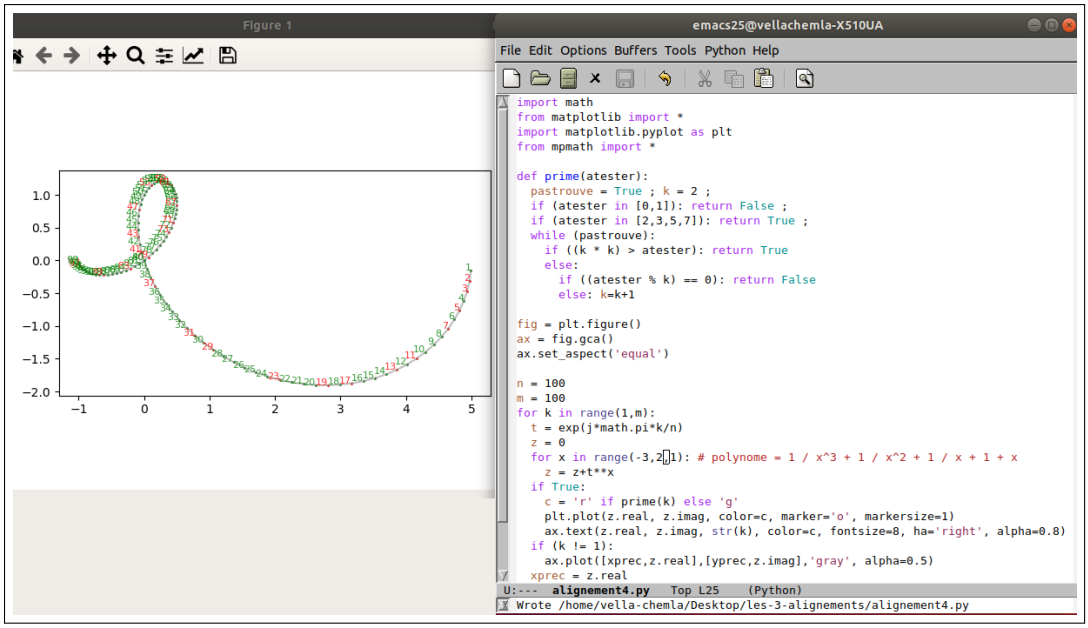


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
import math
from matplotlib import *
import matplotlib.pyplot as plt
from mpmath import *
def prime(atester):
pastrouve = True ; k = 2 ;
if (atester in [0,1]): return False ;
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(-3,2,1): # polynome = 1 / x^3 + 1 / x^2 + 1 / x + 1 + x
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
U:--- alignement4.py Top L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement4.py

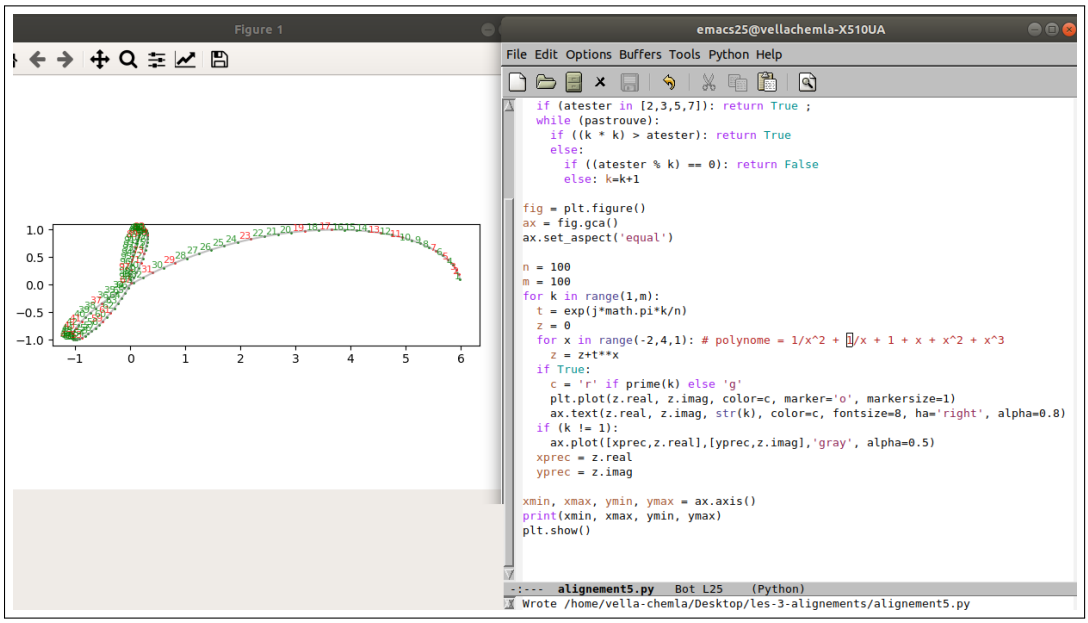


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(-2,4,1): # polynome = 1/x^2 + 1/x + 1 + x + x^2 + x^3
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
yprec = z.imag
xmin, xmax, ymin, ymax = ax.axis()
print(xmin, xmax, ymin, ymax)
plt.show()
-:--- alignement5.py Bot L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement5.py

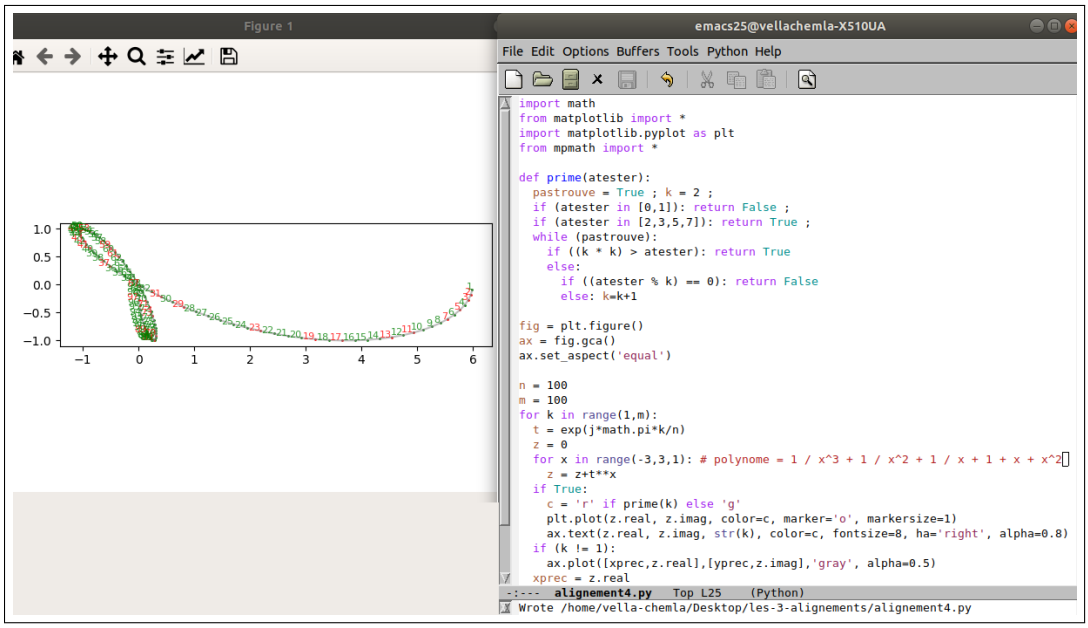


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
import math
from matplotlib import *
import matplotlib.pyplot as plt
from mpmath import *
def prime(atester):
pastrouve = True ; k = 2 ;
if (atester in [0,1]): return False ;
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(-3,3,1): # polynome = 1 / x^3 + 1 / x^2 + 1 / x + 1 + x + x^2
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
xprec = z.real
-:--- alignement4.py Top L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement4.py

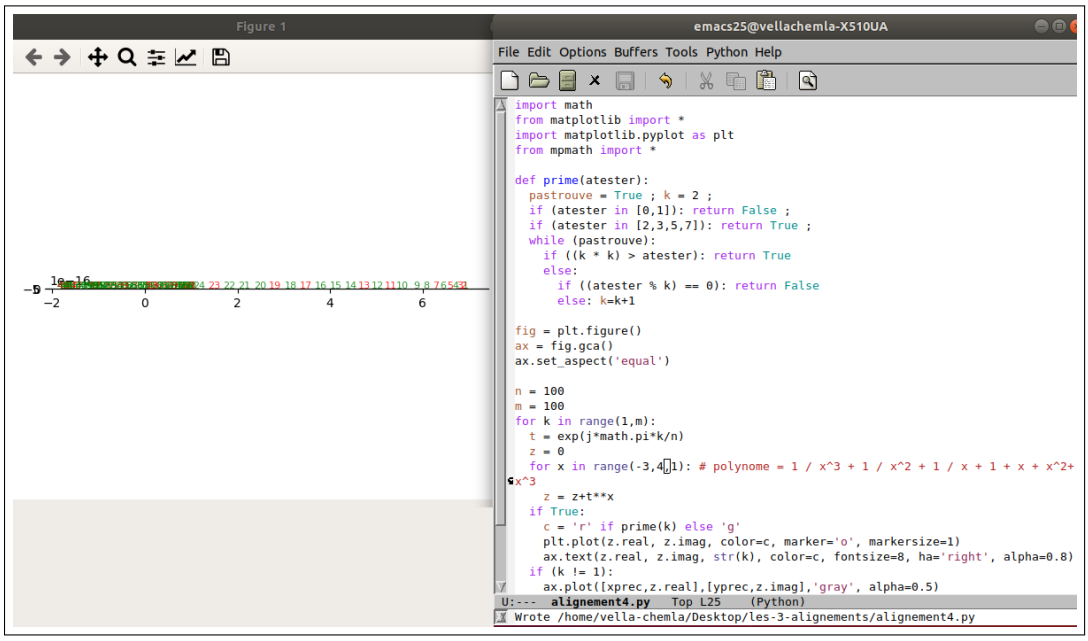


Figure 1
emacs25@vellachemla-X510UA
File Edit Options Buffers Tools Python Help
import math
from matplotlib import *
import matplotlib.pyplot as plt
from mpmath import *
def prime(atester):
pastrouve = True ; k = 2 ;
if (atester in [0,1]): return False ;
if (atester in [2,3,5,7]): return True ;
while (pastrouve):
if ((k * k) > atester): return True
else:
if ((atester % k) == 0): return False
else: k=k+1
fig = plt.figure()
ax = fig.gca()
ax.set_aspect('equal')
n = 100
m = 100
for k in range(1,m):
t = exp(j*math.pi*k/n)
z = 0
for x in range(-3,4,1): # polynome = 1 / x^3 + 1 / x^2 + 1 / x + 1 + x + x^2+ x^3
z = z+t**x
if True:
c = 'r' if prime(k) else 'g'
plt.plot(z.real, z.imag, color=c, marker='o', markersize=1)
ax.text(z.real, z.imag, str(k), color=c, fontsize=8, ha='right', alpha=0.8)
if (k != 1):
ax.plot([xprec,z.real],[yprec,z.imag],'gray', alpha=0.5)
U:--- alignement4.py Top L25 (Python)
Wrote /home/vella-chemla/Desktop/les-3-alignements/alignement4.py