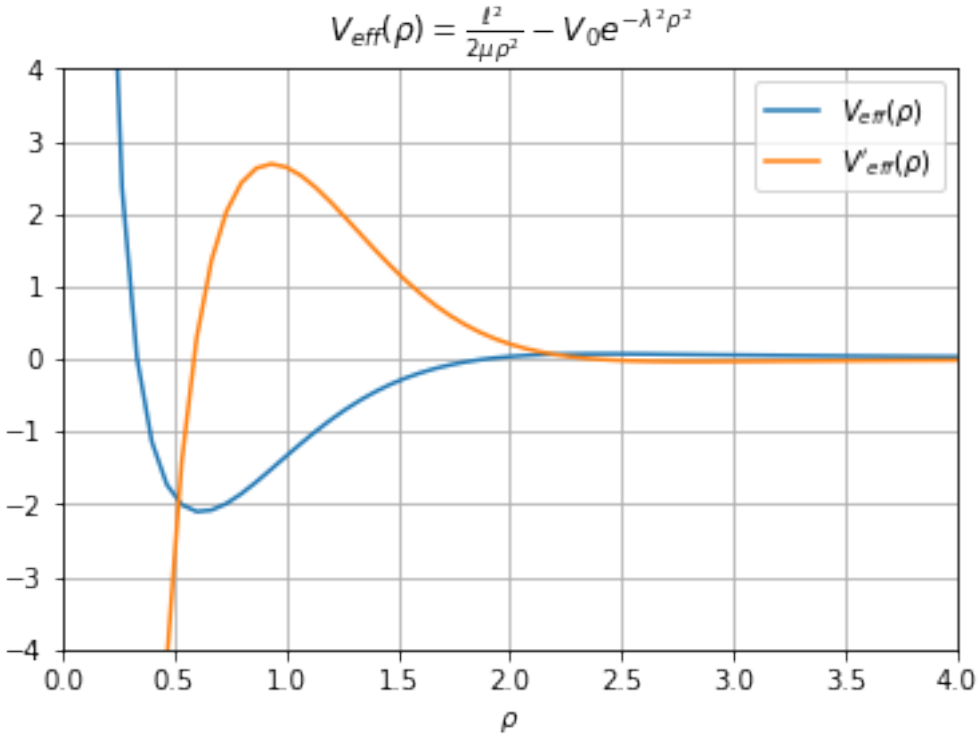
Untitled12

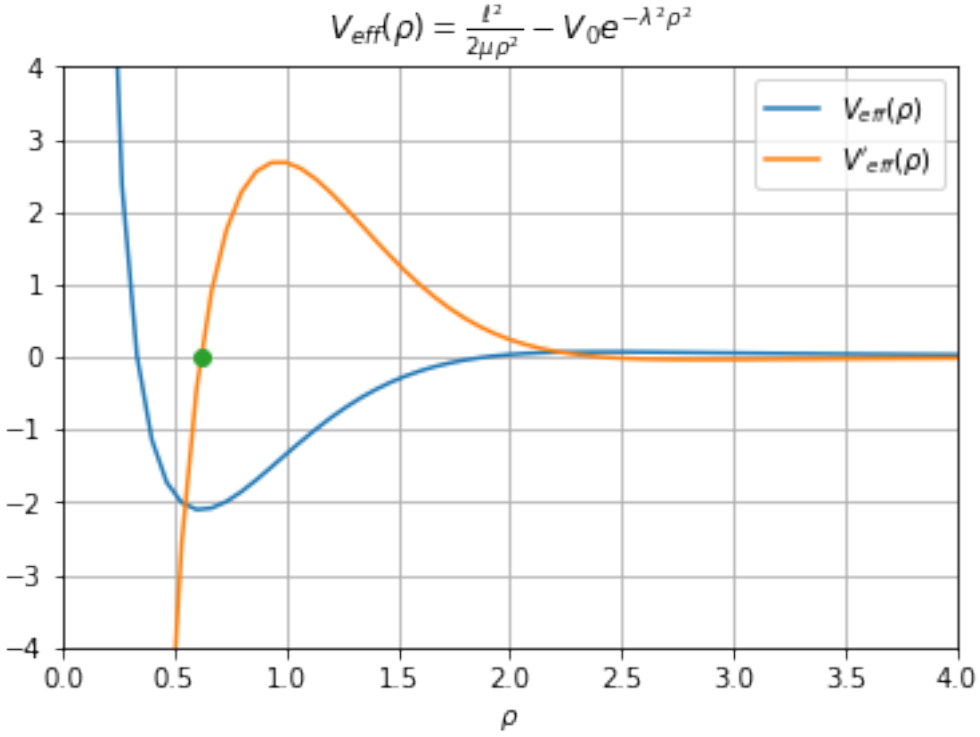
June 20, 2022

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve
np.seterr(divide='ignore') # Division durch Null Warnung ignorieren

r = np.linspace(0,100,1500)
mu = 1
ell = 1
V_0 = 5
lam = 1

V_eff = (ell**2)/(2*mu*r**2) - V_0*np.exp(-(lam*r)**2)
dV_eff = np.diff(V_eff) / np.diff(r) # np.diff() : Differenzial berechnen dV[i]␣
 ↪= V[i+1] - V[i], i=Element des Arrays

## Graphisch Minimum finden:

plt.title(r"$V_{eff}(\rho)=\frac{\ell^2}{2\mu\rho^2}-V_0e^{-\lambda^2\rho^2}$")
plt.plot(r, V_eff, label=r"$V_{eff}(\rho)$")
plt.plot(r[:-1], dV_eff, label=r"$V'_{eff}(\rho)$")
plt.xlabel(r"$\rho$") # \rho = LaTeX Code fuers Symbol rho
plt.xlim([0,4]) # Passende x und y Grenzwerte waehlen
plt.ylim([-4,4])
plt.legend() # Bezeinungen zeigen
plt.grid() # Gitter zeigen
plt.show()

## Nun numerisch ..

def dVeff(r): # Funktion definieren
    dV = -(ell**2)/(mu*r**3) + 2*(lam**2)*r*V_0*np.exp(-(lam*r)**2)
    return dV

rho_guess = 0.5

root = fsolve(dVeff, rho_guess) # Root Solver rufen: fsolve(FunktionName,␣
 ↪[Vermutung])
```

```
print("Root found: r=",root) # Loesung zeigen

# Nochmal plotten ..
plt.title(r"$V_{eff}(\rho)=\frac{\ell^2}{2\mu\rho^2}-V_0e^{-\lambda^2\rho^2}$")
plt.plot(r, V_eff, label=r"$V_{eff}(\rho)$")
plt.plot(r, dVeff(r), label=r"$V'_{eff}(\rho)$")
plt.plot(root, dVeff(root),"o") # Nullpunkt im Plot markieren
plt.xlabel(r"$\rho$")
plt.xlim([0,4])
plt.ylim([-4,4])
plt.legend()
plt.grid()
```




```
Root found: r= [0.61884228]
```


$V_{eff}(\rho) = \frac{\ell^2}{2\mu\rho^2} - V_0 e^{-\lambda^2\rho^2}$
$V_{eff}(\rho)$
$V'_{eff}(\rho)$
4
3
2
1
0
−1
−2
−3
−4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
ρ

[ ]: