Variational

by Ali Salehi

December 14, 2020

1 Introduction

Here we wish to investigate the effectiveness of variational method in solving electrodynamics problems. To do this we first solve the Poisson equation for a one dimension Dirichlet problem. And then put four different functions into the action and try to minimize it. Finally we plot all the results together to judge various suggested functions.

To do this project, I used SymPy. SymPy is a Python library for symbolic mathematics. you can find additional information about this library in it's official website.

2 Calculations

First we need to import the library.
p.s. The first line is for higher quality plots.

[1]:
```
%config InlineBackend.figure_format = 'svg'
import sympy as sy
import sympy.plotting as plt
```

Now we Define mathematical variables and functions

[2]:
```
rho, C1, C2, I1, I2, I3, I4 = sy.symbols('rho C1 C2 I1 I2 I3 I4')
g, psi = sy.symbols('g psi', cls=sy.Function)
```

2.1 Poisson Equation

We know that the potential only depends ρ so we can write Poisson equation as follows:

[3]:
```
diffeq = sy.Eq(1/rho * (rho * psi(rho).diff(rho)).diff(rho), -g(rho))
sy.simplify(diffeq)
```

[3]:

$$g(\rho) = -\frac{d^2}{d\rho^2}\psi(\rho) - \frac{\frac{d}{d\rho}\psi(\rho)}{\rho}$$

2.1.1 General Solution

Now that we have defined the differential equation to solve, we can use SymPy's *dsolve* function.

[4]:
```
sy.dsolve(diffeq, psi(rho))
```

[4]:

$$\psi(\rho) = C_1 + C_2 \log(\rho) - \log(\rho) \int \rho g(\rho)\, d\rho + \int \rho g(\rho) \log(\rho)\, d\rho$$

This is the most general form of potential, to have a complete solution, we need to have $g(\rho)$ and also find C_1 and C_2 by dictating boundary conditions.

Solution with definite g(ρ)

[5]:
```
g = -5 * (1-rho) + 10000 * rho**5 * (1-rho)**5
diffeq = sy.Eq(1/rho * (rho * psi(rho).diff(rho)).diff(rho), -g)
equation = sy.dsolve(diffeq, psi(rho))
equation
```

[5]:

$$\psi(\rho) = C_1 + C_2 \log(\rho) + \frac{625\rho^{12}}{9} - \frac{50000\rho^{11}}{121} + 1000\rho^{10} - \frac{100000\rho^9}{81} + \frac{3125\rho^8}{4} - \frac{10000\rho^7}{49} - \frac{5\rho^3}{9} + \frac{5\rho^2}{4}$$

At $\rho = 0$, the derivative of $\psi(\rho)$ should vanish, so we have:

[6]:
```
C2_solved = sy.nonlinsolve([rho, equation.rhs.diff(rho)], (rho, C2))
C2_solved.args[0][1]
```

[6]:
0

[7]:
```
equation = equation.subs({C2:C2_solved.args[0][1]})
equation
```

[7]:

$$\psi(\rho) = C_1 + \frac{625\rho^{12}}{9} - \frac{50000\rho^{11}}{121} + 1000\rho^{10} - \frac{100000\rho^9}{81} + \frac{3125\rho^8}{4} - \frac{10000\rho^7}{49} - \frac{5\rho^3}{9} + \frac{5\rho^2}{4}$$

Also at $\rho = 1$, the potential itself should vanish, so:

[8]:
```
C1_solved = sy.nonlinsolve([rho - 1, equation.rhs], (rho, C1))
C1_solved.args[0][1]
```

[8]:

$$\frac{464675}{960498}$$

Now we can arrive at a complete solution.

2.1.2 Complete solution

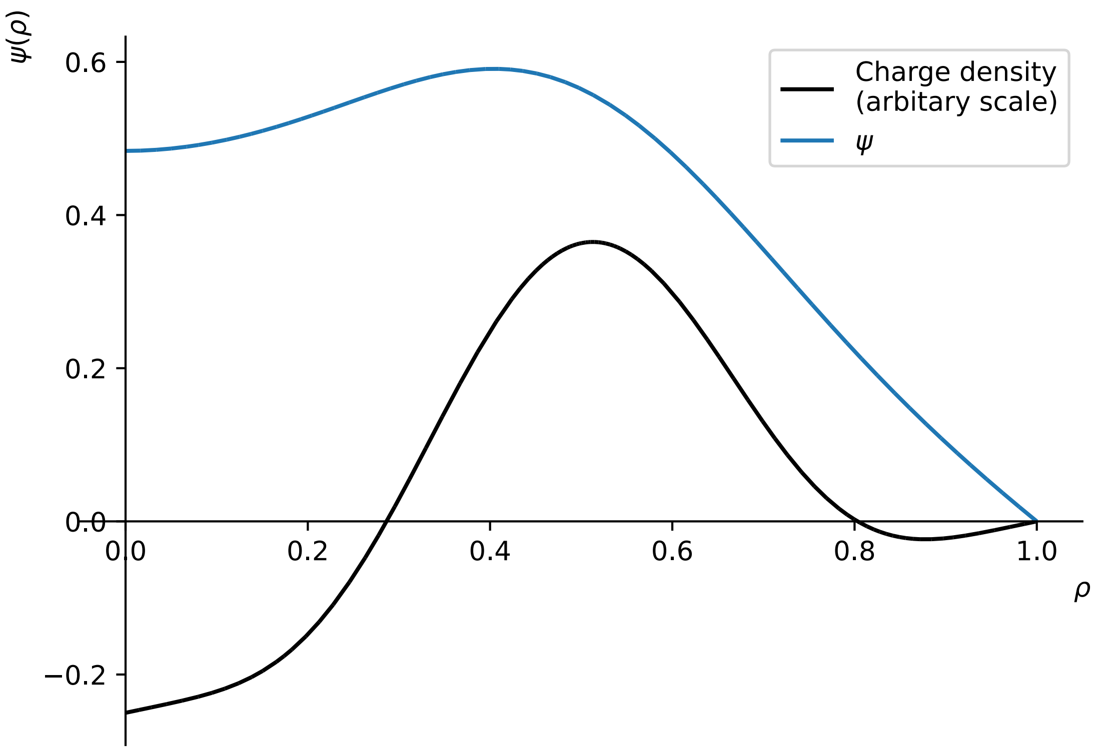
[9]:
```
equation = equation.subs({C1:C1_solved.args[0][1]})
equation
```

[9]:

$$\psi(\rho) = \frac{625\rho^{12}}{9} - \frac{50000\rho^{11}}{121} + 1000\rho^{10} - \frac{100000\rho^9}{81} + \frac{3125\rho^8}{4} - \frac{10000\rho^7}{49} - \frac{5\rho^3}{9} + \frac{5\rho^2}{4} + \frac{464675}{960498}$$

[10]:
```
p1 = plt.plot(g*0.05, (rho, 0, 1), line_color='black', xlabel=r'$\rho$',␣
↪ylabel=r'$\psi(\rho)$', label='Charge density\n(arbitary scale)',␣
↪legend=True, show=False)
p2 = plt.plot(equation.rhs, (rho, 0, 1), label=r'$\psi$', show=False)
```

```
p1.append(p2[0])
p1.style = 'wireframe'
p1.show()
```



2.2 First trial function: $\psi_1(\rho)$

From here until the end, instead of solving the Poisson equation, we need to put trial functions into the action given by $I[\psi] = \frac{1}{2}\int_V \nabla\psi \cdot \nabla\psi d^3x - \int_V g\psi d^3x$ and try to minimize it with respect to the free parameters.

[11]:
```
alpha1, beta1, gamma1 = sy.symbols('alpha1 beta1 gamma1')
g, psi1 = sy.symbols('g psi1', cls=sy.Function)
```

[12]:
```
equation1 = sy.Eq(psi1(rho), alpha1 * (1-rho) + beta1 * (1-rho)**2 + gamma1 *␣
 ↪(1-rho)**3)
equation1
```

[12]:

$$\psi_1(\rho) = \alpha_1 (1-\rho) + \beta_1 (1-\rho)^2 + \gamma_1 (1-\rho)^3$$

Since we are working in cylindrical coordinates and our functions are only dependent on ρ, the integral we must calculate, significantly simplifies to: $I[\psi] = 2\pi \times (\frac{1}{2}\int_0^1 \psi'(\rho)^2 \rho d\rho - \int_0^1 g\psi\rho d\rho)$, where $\psi'(\rho)$ means derivative with respect to ρ.

[13]:

```
sy.Eq(I1, 2*sy.pi * (sy.integrate(equation1.rhs.diff(rho)**2 * rho, (rho, 0,␣
↪1))/2 - sy.integrate(equation1.rhs * g(rho) * rho, (rho, 0, 1))))
```

[13]:

$$\frac{I_1}{2\pi} = \frac{\alpha_1^2}{4} + \frac{\alpha_1\beta_1}{3} + \frac{\alpha_1\gamma_1}{4} + \frac{\beta_1^2}{6} + \frac{3\beta_1\gamma_1}{10} + \frac{3\gamma_1^2}{20} + \int_0^1 (-\alpha_1\rho g(\rho))\,d\rho + \int_0^1 \alpha_1\rho^2 g(\rho)\,d\rho + \int_0^1 (-\beta_1\rho g(\rho))\,d\rho + \int_0^1 2\beta_1\rho^2 g(\rho)\,d\rho + \int_0^1 (-\beta_1\rho^3 g(\rho))\,d\rho + \int_0^1 (-\gamma_1\rho g(\rho))\,d\rho + \int_0^1 3\gamma_1\rho^2 g(\rho)\,d\rho + \int_0^1 (-3\gamma_1\rho^3 g(\rho))\,d\rho + \int_0^1 \gamma_1\rho^4 g(\rho)\,d\rho$$

Action with definite g(ρ)

[14]:

```
g = -5 * (1-rho) + 10000 * rho**5 * (1-rho)**5
I_psi1 = 2*sy.pi * (sy.integrate(equation1.rhs.diff(rho)**2 * rho, (rho, 0, 1))/
↪2 - sy.integrate(equation1.rhs * g * rho, (rho, 0, 1)))
sy.Eq(I1, I_psi1)
```

[14]:

$$I_1 = 2\pi\left(\frac{\alpha_1^2}{4} + \frac{\alpha_1\beta_1}{3} + \frac{\alpha_1\gamma_1}{4} - \frac{1665\alpha_1}{4004} + \frac{\beta_1^2}{6} + \frac{3\beta_1\gamma_1}{10} - \frac{1997\beta_1}{12012} + \frac{3\gamma_1^2}{20} - \frac{997\gamma_1}{18018}\right)$$

Depending on the set of equations we want to solve, we need to use either of these SymPy functions: *solveset*, *linsolve* or *nonlinsolve*.

[15]:

```
params1 = sy.linsolve([I_psi1.diff(alpha1), I_psi1.diff(beta1), I_psi1.
↪diff(gamma1)], [alpha1, beta1, gamma1])
params1
```

[15]:

$$\left\{\left(\frac{2855}{2574}, \frac{2855}{5148}, -\frac{34985}{27027}\right)\right\}$$

$\psi_1(\rho)$ with minimized parameters.

[16]:

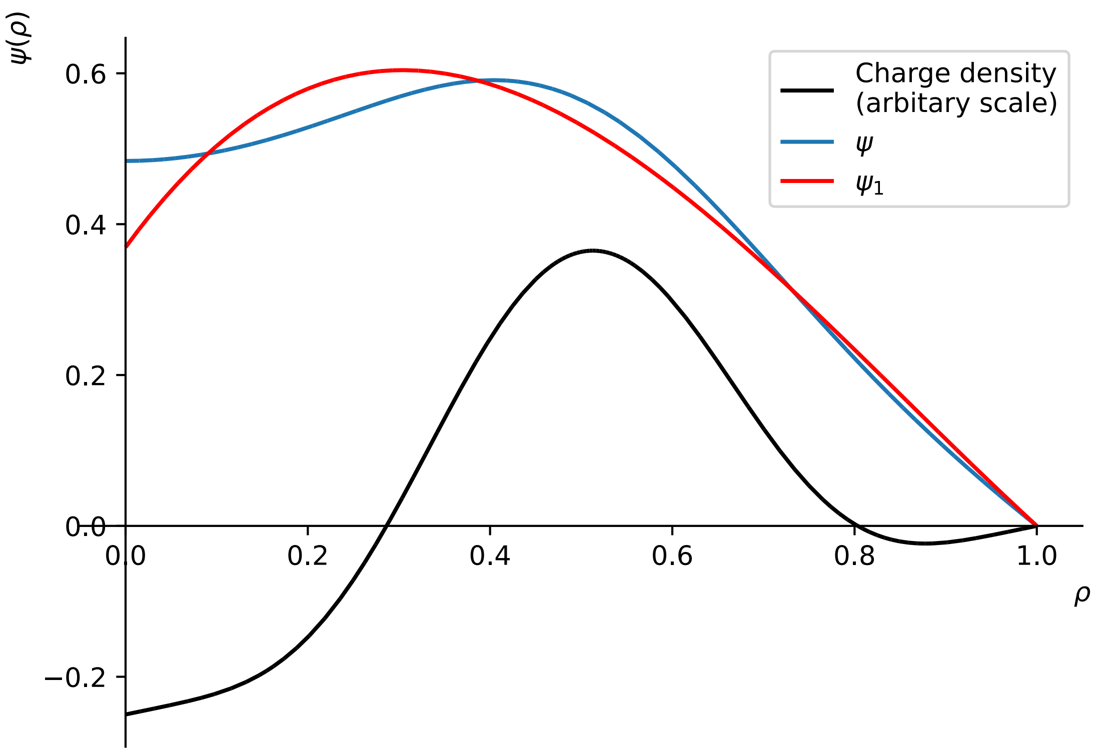
```
alpha1_sovled = params1.args[0][0]
beta1_sovled = params1.args[0][1]
gamma1_sovled = params1.args[0][2]
equation1 = equation1.subs({alpha1:alpha1_sovled, beta1:beta1_sovled, gamma1:
↪gamma1_sovled})
equation1
```

[16]:

$$\psi_1(\rho) = -\frac{2855\rho}{2574} - \frac{34985(1-\rho)^3}{27027} + \frac{2855(1-\rho)^2}{5148} + \frac{2855}{2574}$$

[17]:

```
p3 = plt.plot(equation1.rhs, (rho, 0, 1), line_color='red',␣
↪label=r'$\psi_{1}$', show=False)
p1.append(p3[0])
p1.show()
```



2.3 Second trial function: $\psi_2(\rho)$

Here the procedure is the same as above, so I won't talk much more.

[18]:
```
alpha2, beta2, gamma2 = sy.symbols('alpha2 beta2 gamma2')
g, psi2 = sy.symbols('g psi2', cls=sy.Function)
```

[19]:
```
equation2 = sy.Eq(psi2(rho), alpha2 * rho**2 + beta2 * rho**3 + gamma2 * rho**4␣
↪- alpha2 - beta2 - gamma2)
equation2
```

[19]:

$$\psi_2(\rho) = \alpha_2\rho^2 - \alpha_2 + \beta_2\rho^3 - \beta_2 + \gamma_2\rho^4 - \gamma_2$$

[20]:
```
sy.Eq(I2, 2*sy.pi * (sy.integrate(equation2.rhs.diff(rho)**2 * rho, (rho, 0,␣
↪1))/2 - sy.integrate(equation2.rhs * g(rho) * rho, (rho, 0, 1))))
```

[20]:

$$\frac{I_2}{2\pi} = \frac{\alpha_2^2}{2} + \frac{6\alpha_2\beta_2}{5} + \frac{4\alpha_2\gamma_2}{3} + \frac{3\beta_2^2}{4} + \frac{12\beta_2\gamma_2}{7} + \gamma_2^2 - \int\limits_0^1 \rho\,(\rho-1)\left(\alpha_2\rho + \alpha_2 + \beta_2\rho^2 + \beta_2\rho + \beta_2 + \gamma_2\rho^3 + \gamma_2\rho^2 + \gamma_2\rho + \gamma_2\right) g(\rho)\,d\rho$$

[21]:
```
g = -5 * (1-rho) + 10000 * rho**5 * (1-rho)**5
I_psi2 = 2*sy.pi * (sy.integrate(equation2.rhs.diff(rho)**2 * rho, (rho, 0, 1))/
↪2 - sy.integrate(equation2.rhs * g * rho, (rho, 0, 1)))
sy.Eq(I2, I_psi2)
```

[21]:

$$I_2 = 2\pi\left(\frac{\alpha_2^2}{2} + \frac{6\alpha_2\beta_2}{5} + \frac{4\alpha_2\gamma_2}{3} + \frac{7993\alpha_2}{12012} + \frac{3\beta_2^2}{4} + \frac{12\beta_2\gamma_2}{7} + \frac{7244\beta_2}{9009} + \gamma_2^2 + \frac{7940\gamma_2}{9009}\right)$$

[22]:
```
params2 = sy.linsolve([I_psi2.diff(alpha2), I_psi2.diff(beta2), I_psi2.
↪diff(gamma2)], [alpha2, beta2, gamma2])
params2
```

[22]:

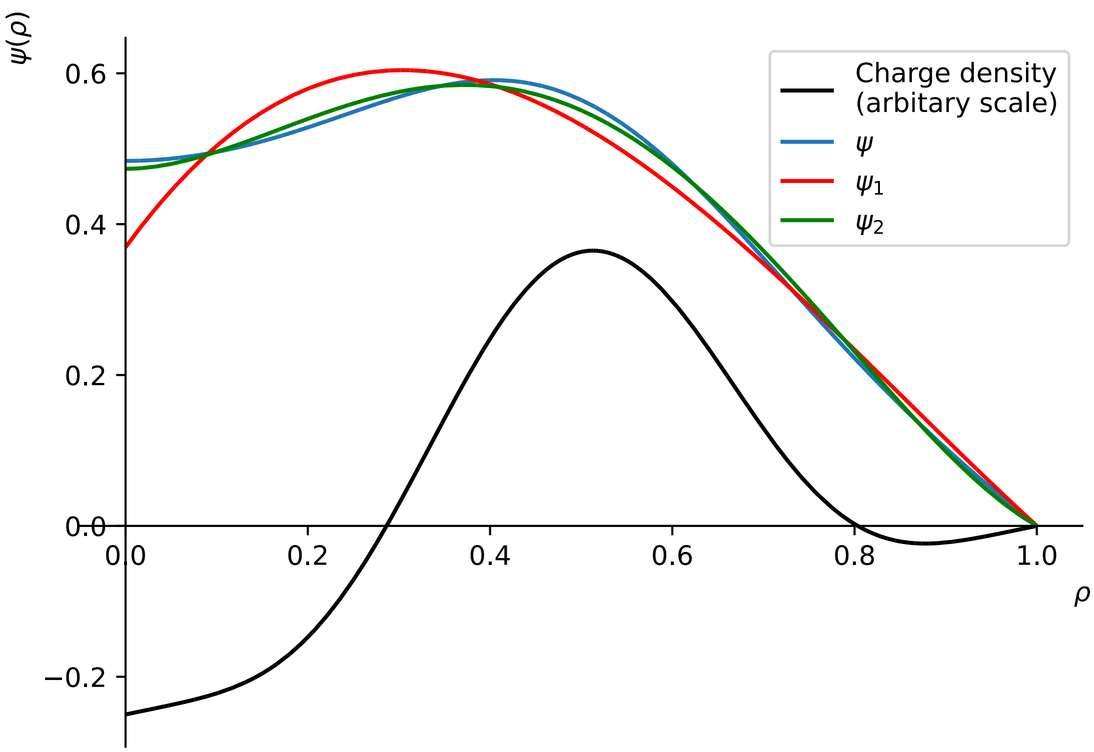
$$\left\{\left(\frac{35015}{12012}, -\frac{27145}{3861}, \frac{3125}{858}\right)\right\}$$

[23]:
```
alpha2_sovled = params2.args[0][0]
beta2_sovled = params2.args[0][1]
gamma2_sovled = params2.args[0][2]
equation2 = equation2.subs({alpha2:alpha2_sovled, beta2:beta2_sovled, gamma2:
↪gamma2_sovled})
equation2
```

[23]:

$$\psi_2(\rho) = \frac{3125\rho^4}{858} - \frac{27145\rho^3}{3861} + \frac{35015\rho^2}{12012} + \frac{51175}{108108}$$

[24]:
```
p4 = plt.plot(equation2.rhs, (rho, 0, 1), line_color='green',␣
↪label=r'$\psi_{2}$', show=False)
p1.append(p4[0])
p1.show()
```



2.4 Third trial function: $\psi_3(\rho)$

We may be interested in seeing what will happen if we add one higher term to the trial function.

[25]:
```
alpha3, beta3, gamma3, omega3 = sy.symbols('alpha3 beta3 gamma3 omega3')
g, psi3 = sy.symbols('g psi3', cls=sy.Function)
```

[26]:
```
equation3 = sy.Eq(psi3(rho), alpha3 * rho**2 + beta3 * rho**3 + gamma3 * rho**4␣
↪+ omega3 * rho**5 - alpha3 - beta3 - gamma3 - omega3)
equation3
```

[26]:
$$\psi_3(\rho) = \alpha_3\rho^2 - \alpha_3 + \beta_3\rho^3 - \beta_3 + \gamma_3\rho^4 - \gamma_3 + \omega_3\rho^5 - \omega_3$$

[27]:
```
sy.Eq(I3, 2*sy.pi * (sy.integrate(equation3.rhs.diff(rho)**2 * rho, (rho, 0,␣
↪1))/2 - sy.integrate(equation3.rhs * g(rho) * rho, (rho, 0, 1))))
```

[27]:

$$\frac{I_3}{2\pi} = \frac{\alpha_3^2}{2} + \frac{6\alpha_3\beta_3}{5} + \frac{4\alpha_3\gamma_3}{3} + \frac{10\alpha_3\omega_3}{7} + \frac{3\beta_3^2}{4} + \frac{12\beta_3\gamma_3}{7} + \frac{15\beta_3\omega_3}{8} + \gamma_3^2 + \frac{20\gamma_3\omega_3}{9} + \frac{5\omega_3^2}{4} - \int\limits_0^1 \rho\,(\rho-1)\left(\alpha_3\rho + \alpha_3 + \beta_3\rho^2 + \beta_3\rho + \beta_3 + \gamma_3\rho^3 + \gamma_3\rho^2 + \gamma_3\rho + \gamma_3 + \omega_3\rho^4 + \omega_3\rho^3 + \omega_3\rho^2 + \omega_3\rho + \omega_3\right) g(\rho)\,d\rho$$

[28]:
```
g = -5 * (1-rho) + 10000 * rho**5 * (1-rho)**5
I_psi3 = 2*sy.pi * (sy.integrate(equation3.rhs.diff(rho)**2 * rho, (rho, 0, 1))/
↪2 - sy.integrate(equation3.rhs * g * rho, (rho, 0, 1)))
sy.Eq(I3, I_psi3)
```

[28]:

$$\frac{I_3}{2\pi} = \frac{\alpha_3^2}{2} + \frac{6\alpha_3\beta_3}{5} + \frac{4\alpha_3\gamma_3}{3} + \frac{10\alpha_3\omega_3}{7} + \frac{7993\alpha_3}{12012} + \frac{3\beta_3^2}{4} + \frac{12\beta_3\gamma_3}{7} + \frac{15\beta_3\omega_3}{8} + \frac{7244\beta_3}{9009} + \gamma_3^2 + \frac{20\gamma_3\omega_3}{9} + \frac{7940\gamma_3}{9009} + \frac{5\omega_3^2}{4} + \frac{1133375\omega_3}{1225224}$$

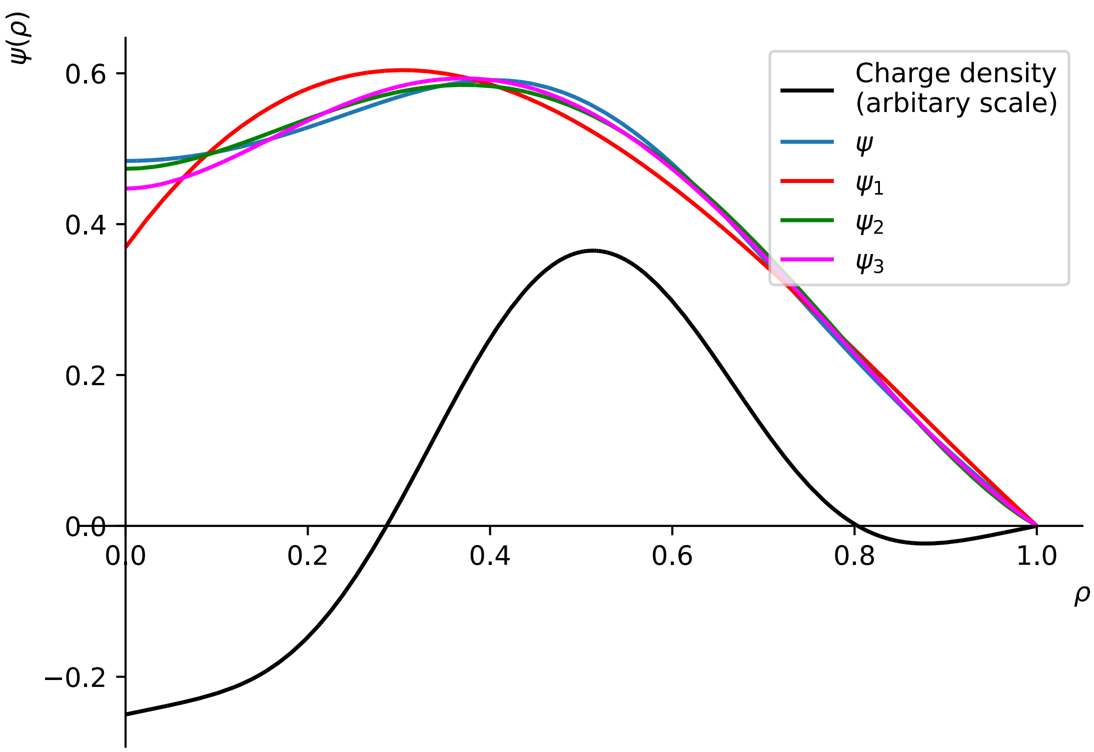
[29]:
```
params3 = sy.nonlinsolve([I_psi3.diff(alpha3), I_psi3.diff(beta3), I_psi3.
↪diff(gamma3), I_psi3.diff(omega3)], [alpha3, beta3, gamma3, omega3])
params3
```

[29]: $$\left\{\left(\frac{28415}{6732}, -\frac{761465}{65637}, \frac{44375}{4862}, -\frac{16000}{7293}\right)\right\}$$

[30]:
```
alpha3_sovled = params3.args[0][0]
beta3_sovled = params3.args[0][1]
gamma3_sovled = params3.args[0][2]
omega3_sovled = params3.args[0][3]
equation3 = equation3.subs({alpha3:alpha3_sovled, beta3:beta3_sovled, gamma3:
↪gamma3_sovled, omega3:omega3_sovled})
equation3
```

[30]: $$\psi_3(\rho) = -\frac{16000\rho^5}{7293} + \frac{44375\rho^4}{4862} - \frac{761465\rho^3}{65637} + \frac{28415\rho^2}{6732} + \frac{10675}{23868}$$

[31]:
```
p5 = plt.plot(equation3.rhs, (rho, 0, 1), line_color='magenta',␣
↪label=r'$\psi_{3}$', show=False)
p1.append(p5[0])
p1.show()
```



Amazingly adding another higher term not only didn't improve the result, It made it even worse!

2.5 Fourth trial function: $\psi_4(\rho)$

We try one last option, to see how our trial function behaves if we only have odd powers of ρ.

[32]:
```
alpha4, beta4, gamma4, omega4 = sy.symbols('alpha4 beta4 gamma4 omega4')
g, psi4 = sy.symbols('g psi4', cls=sy.Function)
```

[33]:
```
equation4 = sy.Eq(psi4(rho), alpha4*rho**3 + beta4*rho**5 + gamma4*rho**7 +␣
↪omega4*rho**11 - alpha4 - beta4 - gamma4 - omega4)
equation4
```

[33]:

$$\psi_4(\rho) = \alpha_4\rho^3 - \alpha_4 + \beta_4\rho^5 - \beta_4 + \gamma_4\rho^7 - \gamma_4 + \omega_4\rho^{11} - \omega_4$$

[34]:
```
sy.Eq(I4, 2*sy.pi * (sy.integrate(equation4.rhs.diff(rho)**2 * rho, (rho, 0,␣
↪1))/2 - sy.integrate(equation4.rhs * g(rho) * rho, (rho, 0, 1))))
```

[34]:

$$\frac{I_4}{2\pi} = \frac{3\alpha_4^2}{4} + \frac{15\alpha_4\beta_4}{8} + \frac{21\alpha_4\gamma_4}{10} + \frac{33\alpha_4\omega_4}{14} + \frac{5\beta_4^2}{4} + \frac{35\beta_4\gamma_4}{12} + \frac{55\beta_4\omega_4}{16} + \frac{7\gamma_4^2}{4} + \frac{77\gamma_4\omega_4}{18} + \frac{11\omega_4^2}{4} - \int_0^1 \rho(\rho-1)\left(\alpha_4\rho^2 + \alpha_4\rho + \alpha_4 + \beta_4\rho^4 + \beta_4\rho^3 + \beta_4\rho^2 + \beta_4\rho + \beta_4 + \gamma_4\rho^6 + \gamma_4\rho^5 + \gamma_4\rho^4 + \gamma_4\rho^3 + \gamma_4\rho^2 + \gamma_4\rho + \gamma_4 + \omega_4\rho^{10} + \omega_4\rho^9 + \omega_4\rho^8 + \omega_4\rho^7 + \omega_4\rho^6 + \omega_4\rho^5 + \omega_4\rho^4 + \omega_4\rho^3 + \omega_4\rho^2 + \omega_4\rho + \omega_4 g(\rho)\, d\rho$$

[35]:
```
g = -5 * (1-rho) + 10000 * rho**5 * (1-rho)**5
I_psi4 = 2*sy.pi * (sy.integrate(equation4.rhs.diff(rho)**2 * rho, (rho, 0, 1))/
↪2 - sy.integrate(equation4.rhs * g * rho, (rho, 0, 1)))
sy.Eq(I4, I_psi4)
```

[35]:

$$\frac{I_4}{2\pi} = \frac{3\alpha_4^2}{4} + \frac{15\alpha_4\beta_4}{8} + \frac{21\alpha_4\gamma_4}{10} + \frac{33\alpha_4\omega_4}{14} + \frac{7244\alpha_4}{9009} + \frac{5\beta_4^2}{4} + \frac{35\beta_4\gamma_4}{12} + \frac{55\beta_4\omega_4}{16} + \frac{1133375\beta_4}{1225224} + \frac{7\gamma_4^2}{4} + \frac{77\gamma_4\omega_4}{18} + \frac{215903\gamma_4}{223839} + \frac{11\omega_4^2}{4} + \frac{3863630\omega_4}{3936933}$$

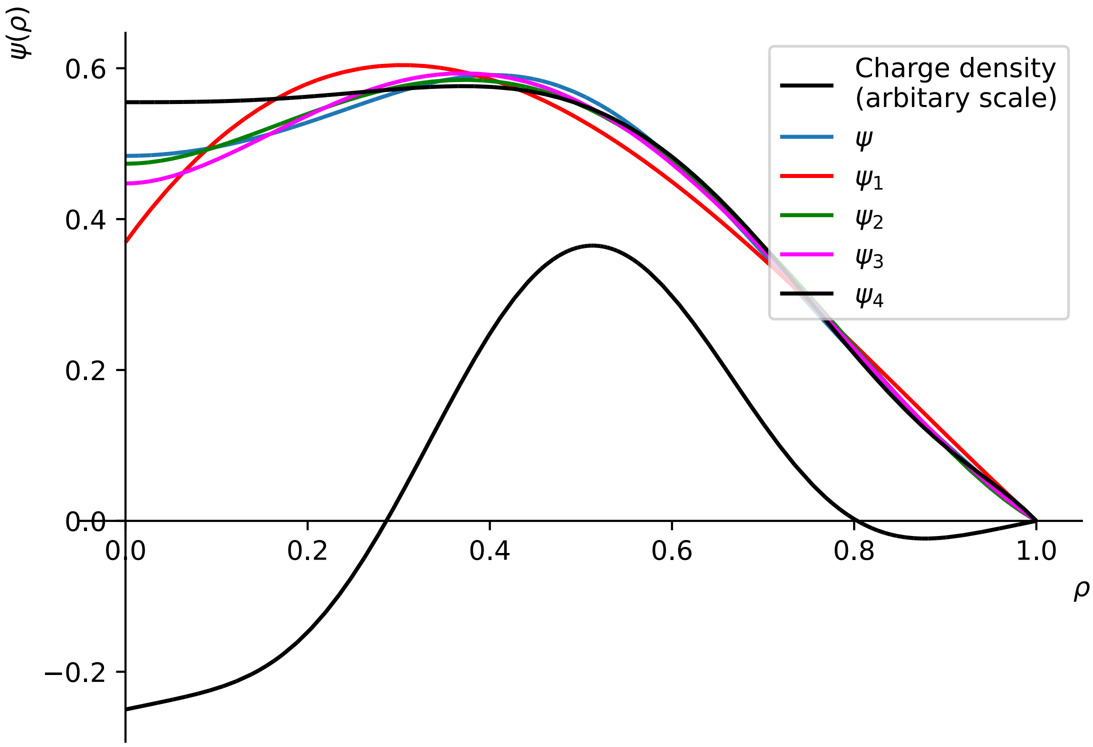
[36]:
```
params4 = sy.nonlinsolve([I_psi4.diff(alpha4), I_psi4.diff(beta4), I_psi4.
↪diff(gamma4), I_psi4.diff(omega4)], [alpha4, beta4, gamma4, omega4])
params4
```

[36]:
$$\left\{\left(\frac{3878195}{3374514}, -\frac{407422720}{66927861}, \frac{585412115}{104110006}, -\frac{43462385}{35057451}\right)\right\}$$

[37]:
```
alpha4_sovled = params4.args[0][0]
beta4_sovled = params4.args[0][1]
gamma4_sovled = params4.args[0][2]
omega4_sovled = params4.args[0][3]
equation4 = equation4.subs({alpha4:alpha4_sovled, beta4:beta4_sovled, gamma4:
↪gamma4_sovled, omega4:omega4_sovled})
equation4
```

[37]:
$$\psi_4(\rho) = -\frac{43462385\rho^{11}}{35057451} + \frac{585412115\rho^7}{104110006} - \frac{407422720\rho^5}{66927861} + \frac{3878195\rho^3}{3374514} + \frac{8579910635}{15460335891}$$

[38]:
```
p6 = plt.plot(equation4.rhs, (rho, 0, 1), line_color='black',␣
↪label=r'$\psi_{4}$', show=False)
p1.append(p6[0])
p1.show()
```



3 Conclusion

By looking at how trial functions behaved, we can conclude that adding higher terms to our trial functions is only effective when those terms are themselves present in the original wave function. otherwise they can be destructive and we will get better results by only looking at first few lower terms. Also it is a lot more computationally economic.