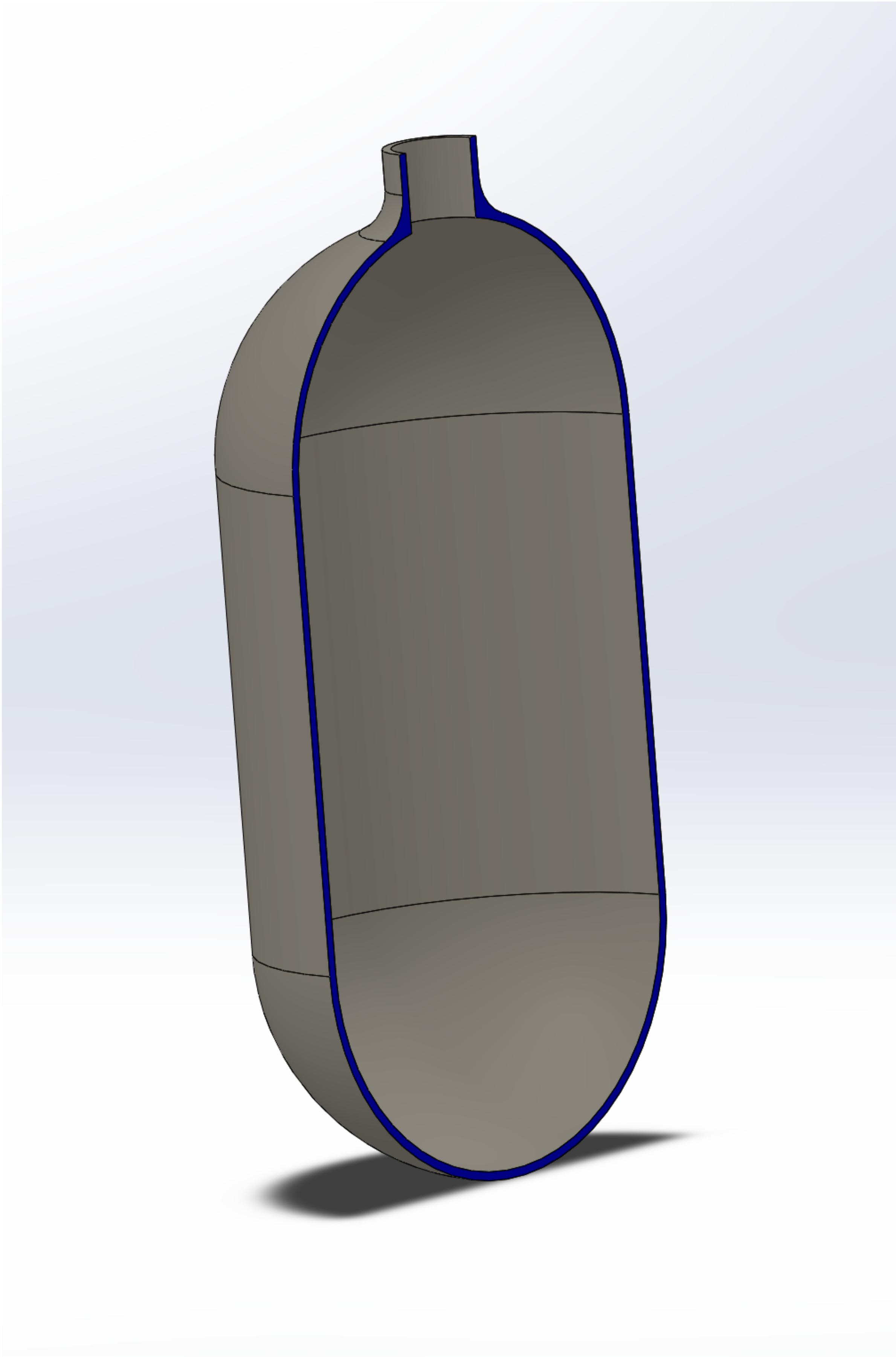
First, we import sympy, numpy, matplotlib,and ipython.display libraries; and, initiate latex printing.

In [1]:
```
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import Image
init_printing()
```

We want to optimize the acceleration at launch of the rocket design shown below.

In [2]:
```
Image(filename='SLDWORKS_lJ3xjb2x4o.png')
```

Out[2]:



Rocket design with cross section.

We want the maximum possible acceleration, **a** to be achieved by our rocket. In order to do this we are optimizing the rocket Equation shown below. This equation came from https://www.grc.nasa.gov/WWW/K-12/rocket/rktslaunch.html (https://www.grc.nasa.gov/WWW/K-12/rocket/rktslaunch.html) and describes the acceleration of a pressurized rocket based on variables: P_i-internal pressure,P_{atm}-external pressure, **a**-acceleration, **g**-gravitational acceleration, **A**-cross sectional area, **t**-time, **W**-weight.

We initiate these symbols and establish a symbolic relationship for the acceleration, **a**.

In [3]:
```
Pi,po,a,g,A,t,W,ro=symbols('P_i,P_atm,a,g,A,t,W,r_o')
a = g*(((Pi-po)*A)/W-1)
```

The expression for **a** is given as:

In [4]:
```
print('a=')
a
```

a=

Out[4]:

$$g\left(\frac{A\left(-P_{atm}+P_i\right)}{W}-1\right)$$

This equation currently does not satisfy the candidate design problem project requirement for as is. There are only 3 design variables when the project requires there to be 4. This can be fixed by substituting in an expression for **A**.

In [5]:
```
r=symbols('r')
A1 = pi*r**2

a = a.subs(A,A1)
a
```

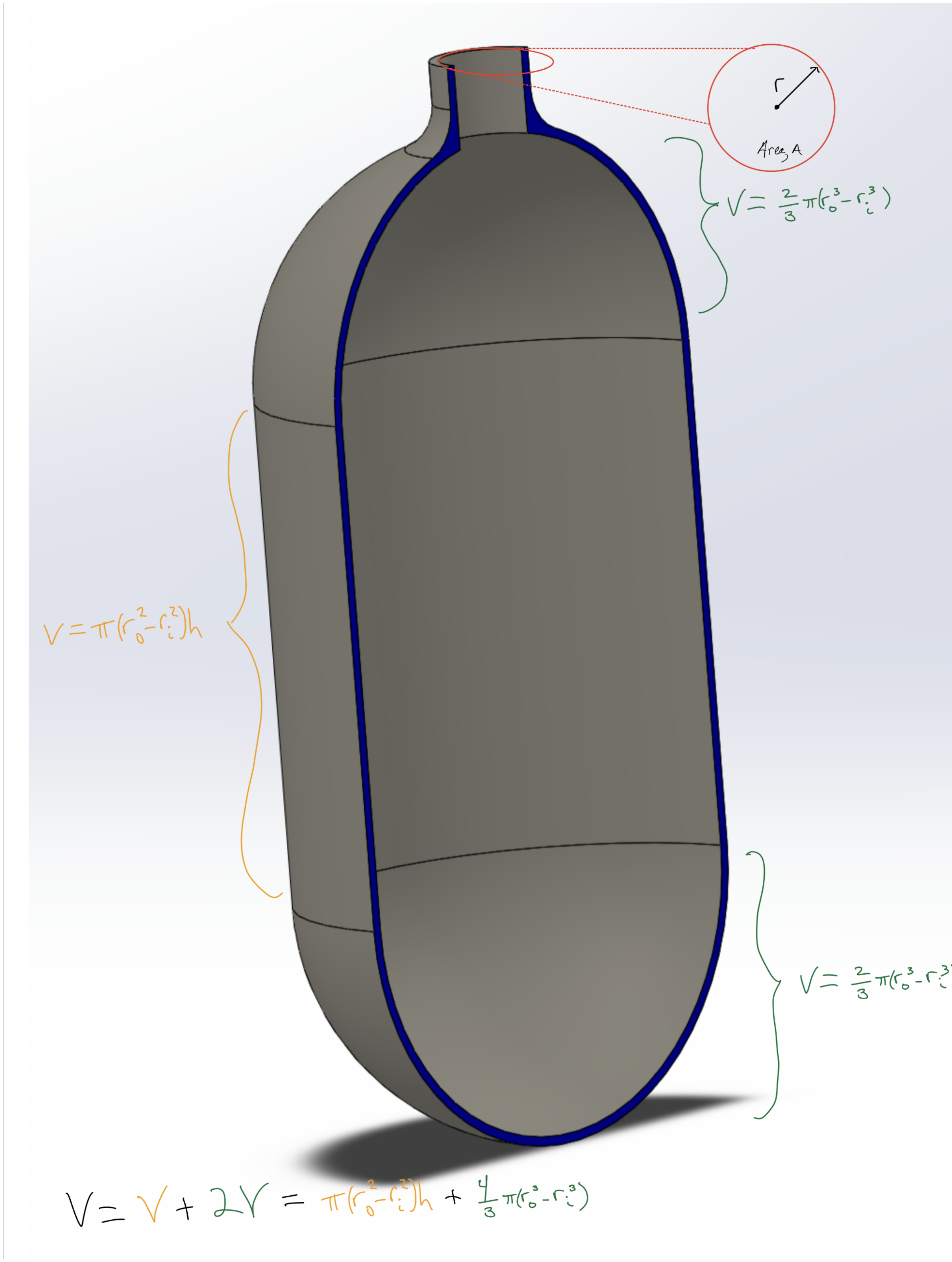
Out[5]:

$$g\left(-1+\frac{\pi r^2\left(-P_{atm}+P_i\right)}{W}\right)$$

This objective equation also does not meet the minimum criteria for the design candidate problem of at least 4 design variables. The weight however is a function of the designs volume. In order to meet the criteria for minimum design variables, we add an expression for volume and substitute it into the candidate design problem with the followiung additional design variables: r_i - inner radius, r_o - outer radius, and **h** - cylinder height.

In [6]:
```
Image(filename='20201106_055134491_iOS.png')
```

Out[6]:



In [7]:
```
ri,h,V,rho,w1=symbols('r_i, h,V,rho,w1')
V = pi*(ro**2-ri**2)*h+(4/3)*pi*(ro**3-ri**3)
```

Now we need a function for W to sub in. We establish W = ρ***V**. Where ρ is the density and **V** is the Volume. We sub in our expression for volume.

In [8]:
```
w1 = rho*V*g
w1
```

Out[8]: $g\rho\left(\pi h\left(-r_i^2+r_o^2\right)+1.33333333333333\pi\left(-r_i^3+r_o^3\right)\right)$

Now that we have a new expression for W we sub it into our candidate design problem.

In [9]:
```
a = a.subs(W,w1)
```

The new candidate objective problem becomes,

In [10]:
```
print('fx=',end='')
a
```

fx=

Out[10]:

$$g\left(-1+\frac{\pi r^2\left(-P_{atm}+P_i\right)}{g\rho\left(\pi h\left(-r_i^2+r_o^2\right)+1.33333333333333\pi\left(-r_i^3+r_o^3\right)\right)}\right)$$

Now, we want to change then design variables to the form:

x_1 - radius of exit throat
x_2 - inner radius
x_3 - outer radius
x_4 - height of midsection
x_5 - internal pressure

We establish these variables symbolically and make these subsitutions.

In [11]:
```
x1,x2,x3,x4,x5=symbols('x_1,x_2,x_3,x_4,x_5')
a = a.subs(r,x1)
a = a.subs(ri,x2)
a = a.subs(ro,x3)
a = a.subs(h,x4)
a = a.subs(Pi,x5)
```

Our candidate objective function becomes.

In [12]:
```
print('fx='+str(a))
a
```

```
fx=g*(-1 + pi*x_1**2*(-P_atm + x_5)/(g*rho*(pi*x_4*(-x_2**2 + x_3**2) + 1.333
33333333333*pi*(-x_2**3 + x_3**3))))
```

Out[12]:

$$g\left(-1 + \frac{\pi x_1^2\left(-P_{atm} + x_5\right)}{g\rho\left(\pi x_4\left(-x_2^2 + x_3^2\right) + 1.33333333333333\pi\left(-x_2^3 + x_3^3\right)\right)}\right)$$

This objective function meets the candidate criteria. Now we begin to define objective function constraints. The value of the internal pressure cannot exceed the critical pressure P_{cr} this imposes g_1 : P_{cr}-P_i $<= 0$ as a constraint. The value of the thickness of the vessel should not be below **2mm** this allows for the constraint: g_2 : r_o-r_i-$.002m$ $<= 0$. The outer radius cannot be greater than 200mm due to the 3D printer bed size. This implies constraint g3: r_o-$.200m$ $<= 0$

We establish these constraints programattically.

In [13]:
```
g1,g2,g3,Pcr=symbols('g_1,g_2,g_3,Pcr')

g1 = Pcr-Pi <= 0
g2 = ri-ro-.002 <= 0
g3 = ro-.200 <= 0
```

This gives us constraints:

In [14]:
```
g1
```

Out[14]: $-P_i + Pcr \leq 0$

In [15]:
```
g2
```

Out[15]: $r_i - r_o - 0.002 \leq 0$

In [16]:
```
g3
```

Out[16]: $r_o - 0.2 \leq 0$

We want these in terms of variables $x_1, x_2, x_3, x_4,$ and x_5. Subbing in the values $x_5 = P_i$ - internal pressure, $x_2 = r_i$ - inner radius, and $x_3 = r_o$ - outer radius into g_1, g_2, g_3.

In [17]:
```
g1 = g1.subs(Pi,x5)
g2 = g2.subs(ri,x2)
g2 = g2.subs(ro,x3)
g3 = g3.subs(ro,x3)
```

The new constraint equations are.

In [18]:
```
g1
```

Out[18]: $Pcr - x_5 \leq 0$

In [19]:
```
g2
```

Out[19]: $x_2 - x_3 - 0.002 \leq 0$

In [20]:
```
g3
```

Out[20]: $x_3 - 0.2 \leq 0$

Now, we want to conduct 1D parameter studies. Using the function param_study1D we study parameters 1, 2, 3, 4, and 5. To do this we import some libararies for math, interactive widgets, and plotting. We also establish 3 functions:

fun*eval(xi): This function evaluates the objective function with vector xi of $x*{1},x*{2},x*{3},x*{4},x*{5}$ values.
study_function(xdata): This function creates vector of y axis data iteratively.
param_study1D: This function takes input parameter and upper and lower graph bounds and outputs paramter study.

In [21]:
```
from ipywidgets import interactive,widgets
```

In [22]:
```
def fun_eval(xi):
    #from numpy import Matrix
    import math
    rho = 1.24 #kg/m^3
    g=   9.81 #m/s^2
    Patm=101325#N/m^2
    #print('xi:\n'+str(xi))
    yi=g*(-1+(((math.pi*xi[0]**2)*(-Patm+xi[4])))/(g*rho*(math.pi*xi[3]*(-1.01
*xi[1]**2+xi[2]**2))+(4/3)*math.pi*(-1.01*xi[1]**3+xi[2]**3)))
    return yi
```

In [23]:
```
def study_function(xdata):
    from numpy import append
    sn,nd= xdata.shape
    ydata=[]
    for i in range(0,sn):
        xi=xdata[i,:]
        yi=fun_eval(xi)
        ydata=append(ydata,yi)
    return ydata
```

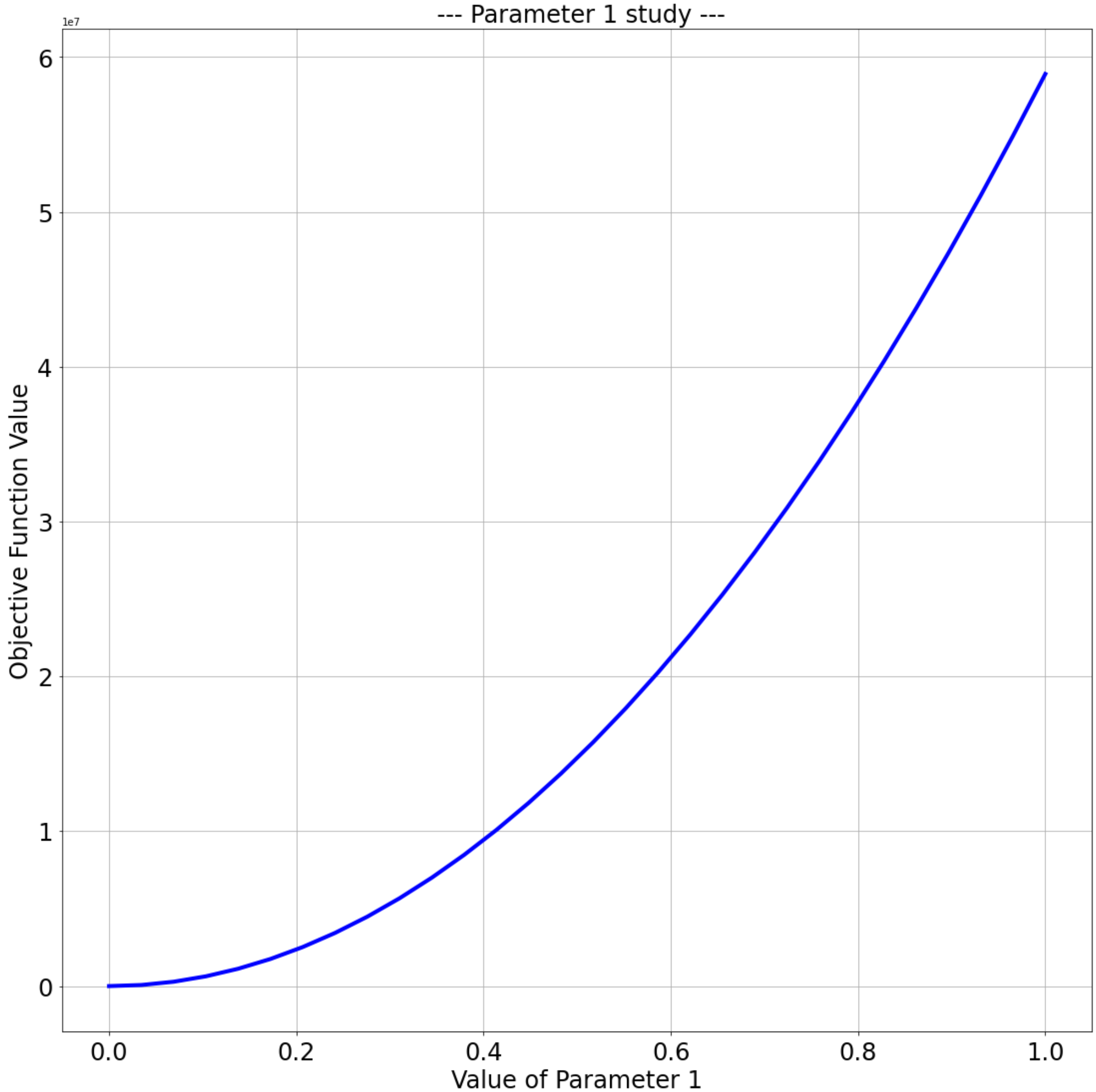
In [24]:
```
def param_study1D(parameter,LB,UB):
    from numpy import ones,arange,transpose,array,linspace
    from matplotlib.pyplot import subplots,plot,show,xlabel,ylabel,legend,titl
e,tick_params,grid,ylim,xlim,xticks,yticks
    %matplotlib inline
    fig, ax = subplots()
    sn=30
    xdata=array(ones([sn,5])*.5)
    #print('xdata:\n'+str(xdata))
    xs=transpose([array(linspace(LB,UB,sn))])
    xdata[:,parameter-1]= xs[:,0]
    #print('xdata:\n'+str(xdata))
    ydata=study_function(xdata)
    #print('ydata:\n'+str(ydata))
    plot(xs,ydata,'b-',linewidth=4)
    #title('parameter %i' %parameter_to_study)
    xlabel('Value of Parameter '+str(parameter),fontsize=24)
    ylabel('Objective Function Value',fontsize=24)
    title(' --- Parameter '+str(parameter)+' study ---',fontsize=24)
    fig.set_size_inches(18.5,18.5)
    tick_params(labelsize=24,pad=6)
    grid()
```

We create an interactive plot 1D study with the function param_study1D we have established

In [25]:
```
interactive(param_study1D,parameter=[1,2,3,4,5],LB=[0,.1,.15,.2,.25,.3,.35,.40
,.45,.50,.55,.60,.65,.70,.75,.80,.85,.90,.95,1],UB=[1,.1,.2,.3,.4,.5,.6,.7,.8,
.9])
```

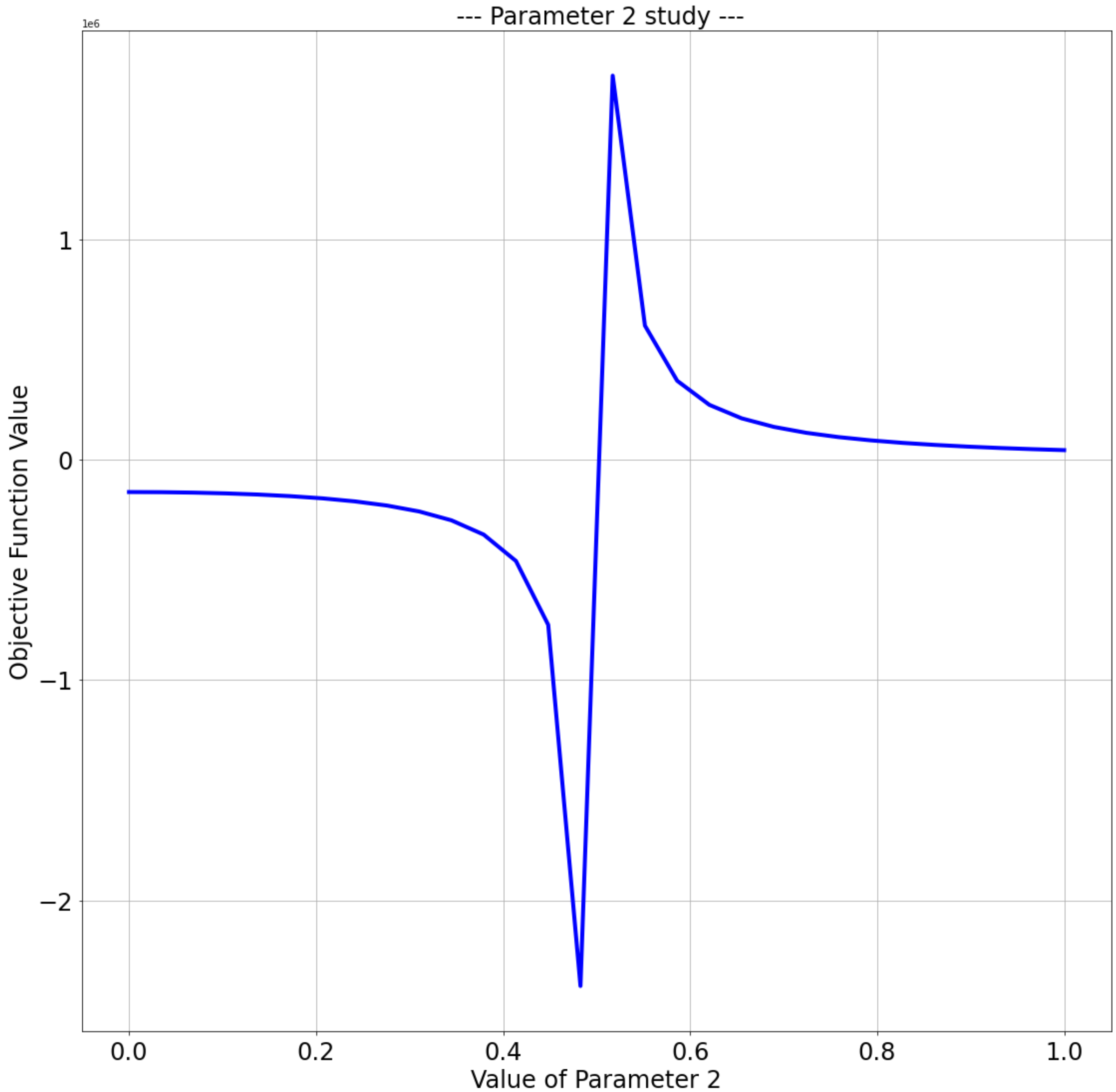
Now, we want to look at these studies individually.

In [26]:
```
param_study1D(1,0,1)
```



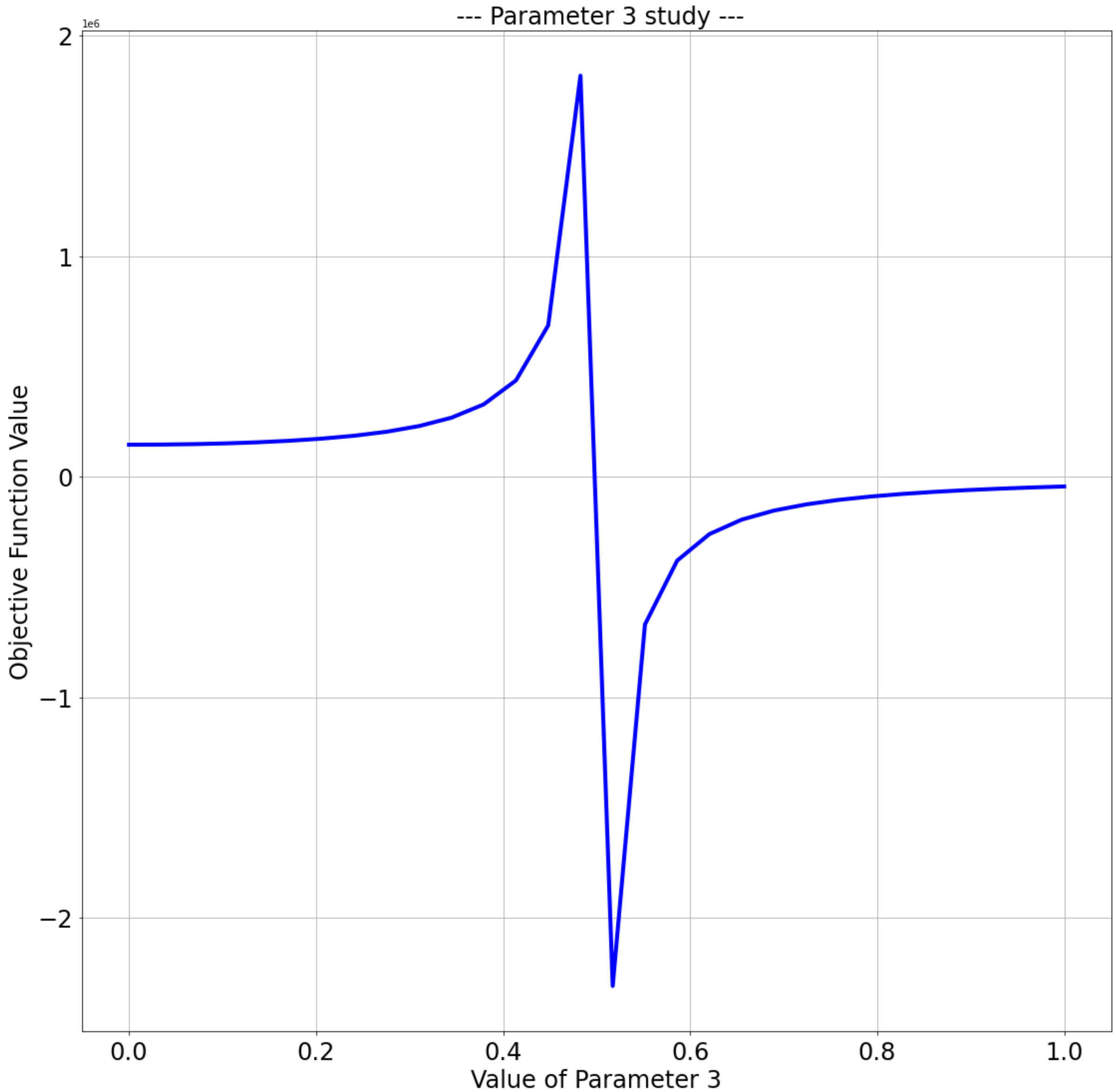
This shows that design parameter x_1 the radius of the exit throat has an exponential relationship with the objective function value.

In [27]:
```
param_study1D(2,0,1)
```



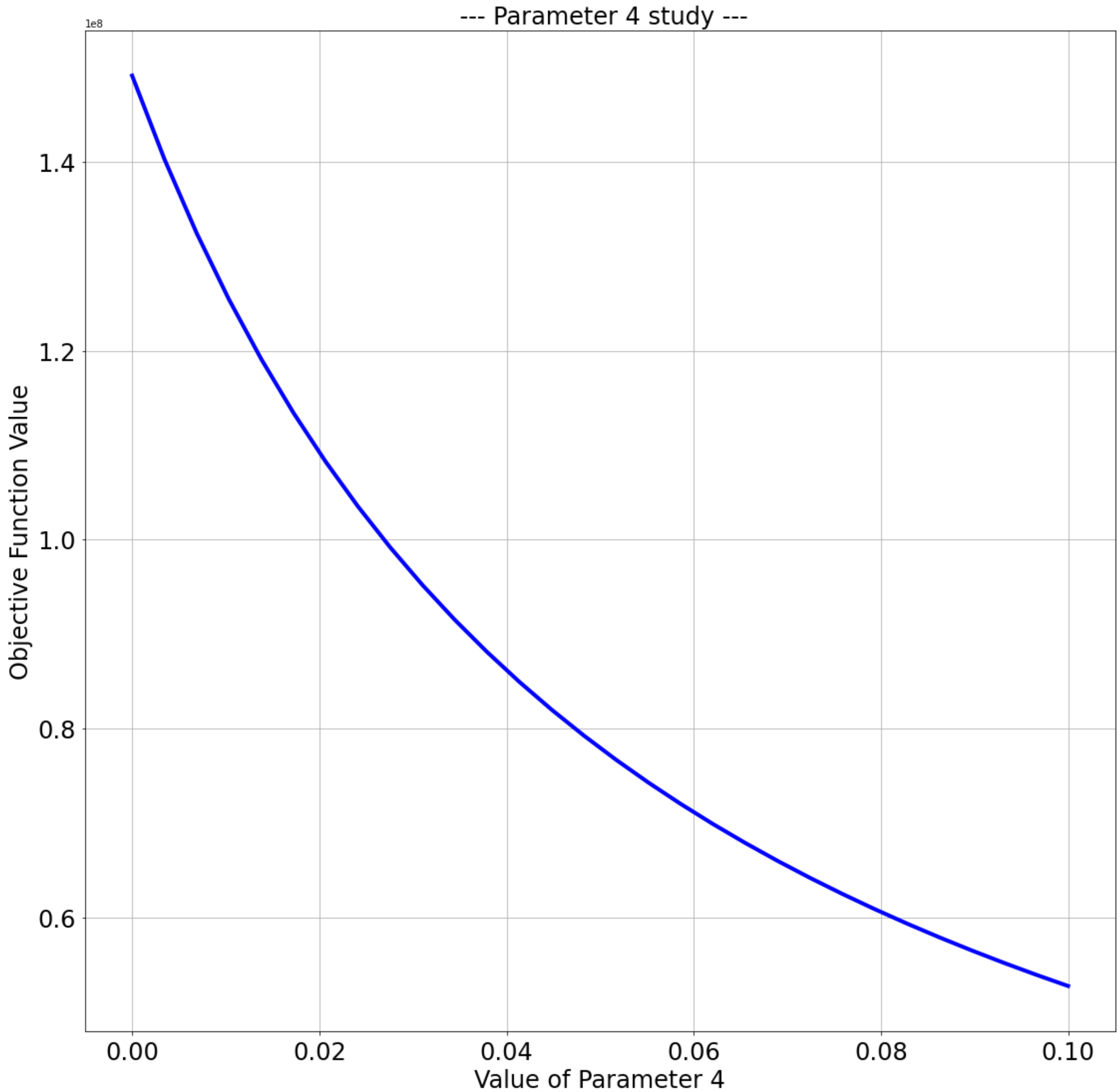
This shows that design parameter x_2 the inner radius of the pressure vessel has a hyperbolic relationship with the objective function value.

In [28]:
```
param_study1D(3,0,1)
```



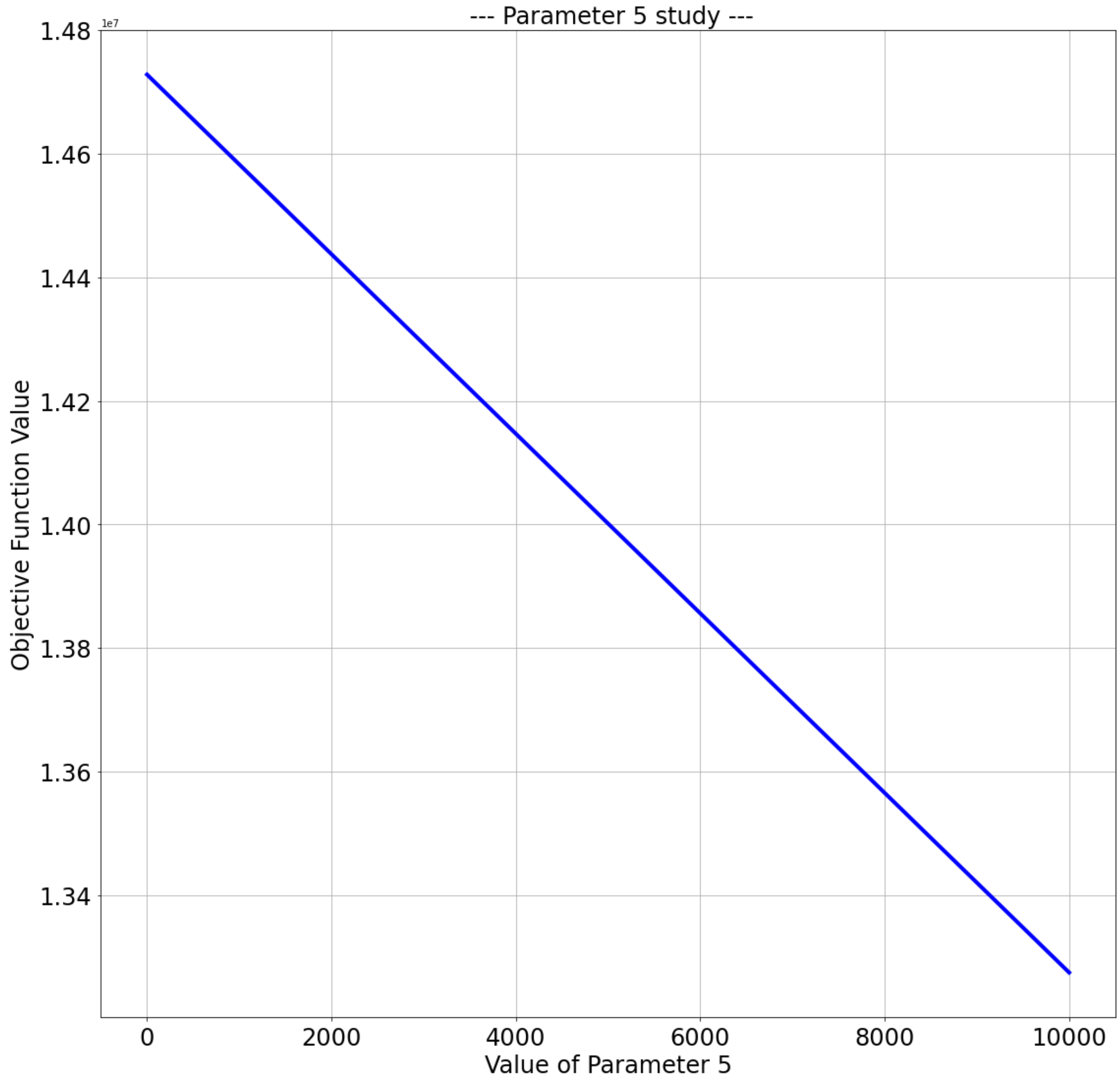
This shows that design parameter x_3 the outer radius of the pressure vessel has a hyperbolic relationship with the objective function value.

In [29]:
```
param_study1D(4,0,.1)
```



This shows that design parameter x_4 the height of the midsection of the pressure vessel has a hyperbolic relationship with the objective function value.

In [30]:
```
param_study1D(5,0,10000)
```



This shows that design parameter x_5 the internal pressure of the pressure vessel has a linear relationship with the objective function value.

The 1D parameter studies above show that there is a strong need for the functions to be bounded around variables: $x_2, x_3,$ and x_5 due to their position on the bottom of a fraction in then objective function. We could still use to know more about the behavior of the design variables so we continue and establish a basis for studying two design parameters vs the objective function.

We establish function: param_study2D(parameter1, parameter2, LB, UB, gridn, angle)

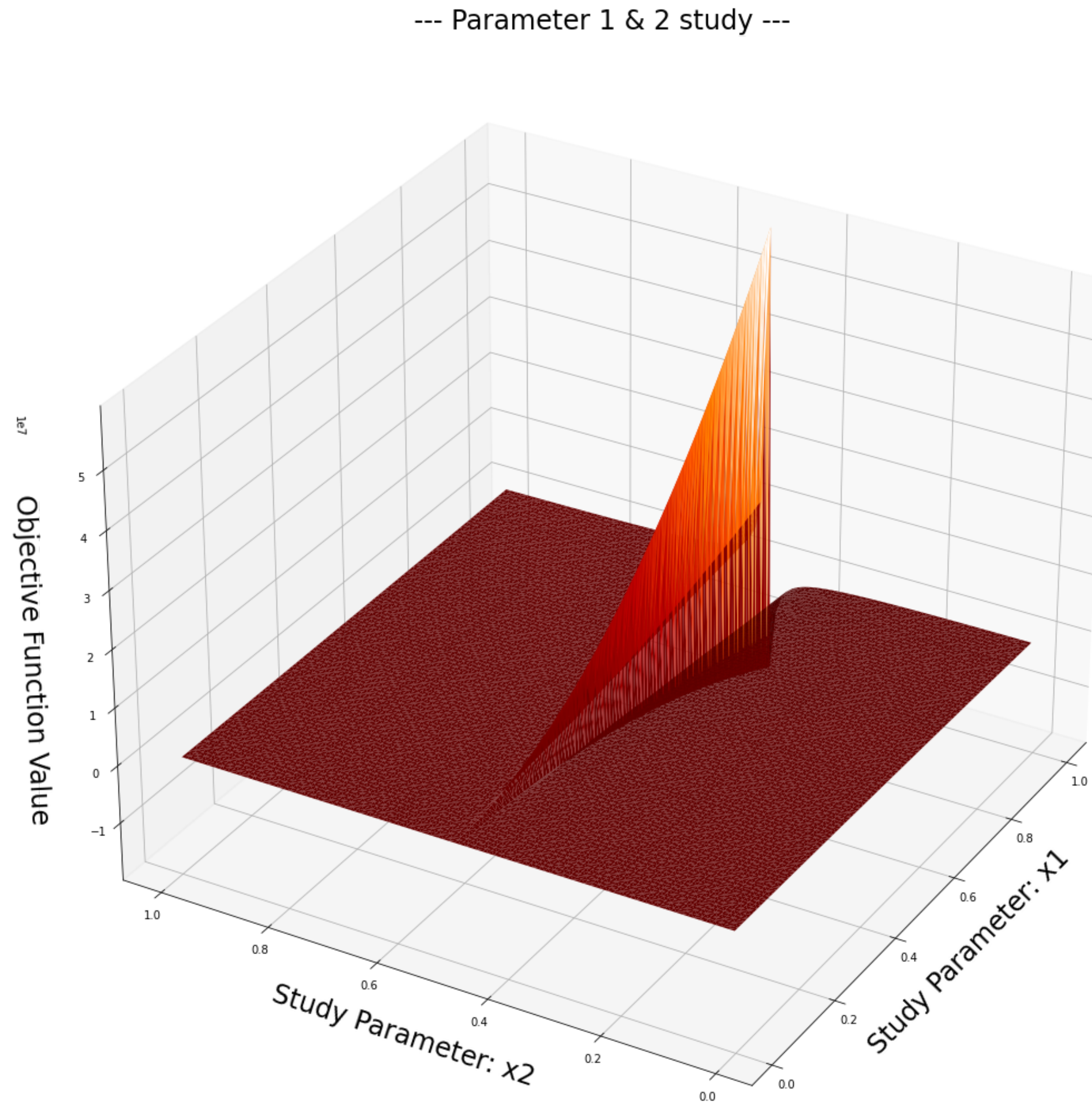
this function takes inputs of: the first study paramter, the second study parameter, the function upperbounds, the function lowerbounds, the amount of grid points, and the angle of the output graph.

In [31]:
```
def param_study2D(parameter1,parameter2,LB,UB,gridn,angle):

    from numpy import ones,arange,transpose,array,linspace,meshgrid,shape,resh
ape,zeros
    from matplotlib.pyplot import subplots,plot,show,xlabel,ylabel,legend,titl
e,tick_params,grid,ylim,xlim,xticks,yticks,close,pause
    from mpl_toolkits import mplot3d
    %matplotlib inline
    xs=transpose([array(linspace(LB,UB,gridn))])
    [xs1,xs2]=meshgrid(xs,xs)
    shape=shape(xs1)
    xs1=xs1.flatten('F')
    xs2=xs2.flatten('F')
    sn=len(xs1)
    parameter1=parameter1-1
    parameter2=parameter2-1


    xdata=ones([sn,5])*.5#integer to right of sn, needs to equal number of des
ign variables in objective function
    xdata[:,parameter1]=xs1
    xdata[:,parameter2]=xs2
    #print('xs1:\n'+str(xs1))
    ydata=study_function(xdata)
    ydata=ydata.flatten('F')
    close("All")
    fig = plt.figure()
    ax = plt.axes(projection='3d')
    ax.plot_trisurf(xs1,xs2,ydata,cmap='gist_heat')
    #ax.scatter(xs1,xs2,ydata,c=ydata,cmap='RdGy',linewidth=1)
    fig.set_size_inches(18.5,18.5)
    xlabel('Study Parameter: x'+str(parameter1+1),fontsize=24,labelpad=15)
    ylabel('Study Parameter: x'+str(parameter2+1),fontsize=24,labelpad=15)
    title(' --- Parameter '+str(parameter1+1)+' & '+str(parameter2+1)+' study
 ---',fontsize=24)
    ax.set_zlabel('Objective Function Value',fontsize=24,labelpad=20)
    ax.view_init(30, angle)
```

In [32]:
```
interactive(param_study2D,parameter1=[1,2,3,4,5],parameter2=[2,1,3,4,5],LB=[0.
01,.08,.09,.1,.11,.12,.13,.14,.15,.25,.4,.5,.75,1,10,100,1000,10000,100000,100
00000],UB=[.1,.11,.12,.13,.14,.15,.25,.3,.5,.6,.75,1,1.1,1.5,100,1000,10000,10
0000,1000000],gridn=[10,15,20,25,30,40,50,60,70,80,90,100],angle=widgets.IntSl
ider(min=0, max=360, step=15, value=255))
```

Now that we have established a 2D study interactive widget and function. We look at the indivdual 2D studies in more details.

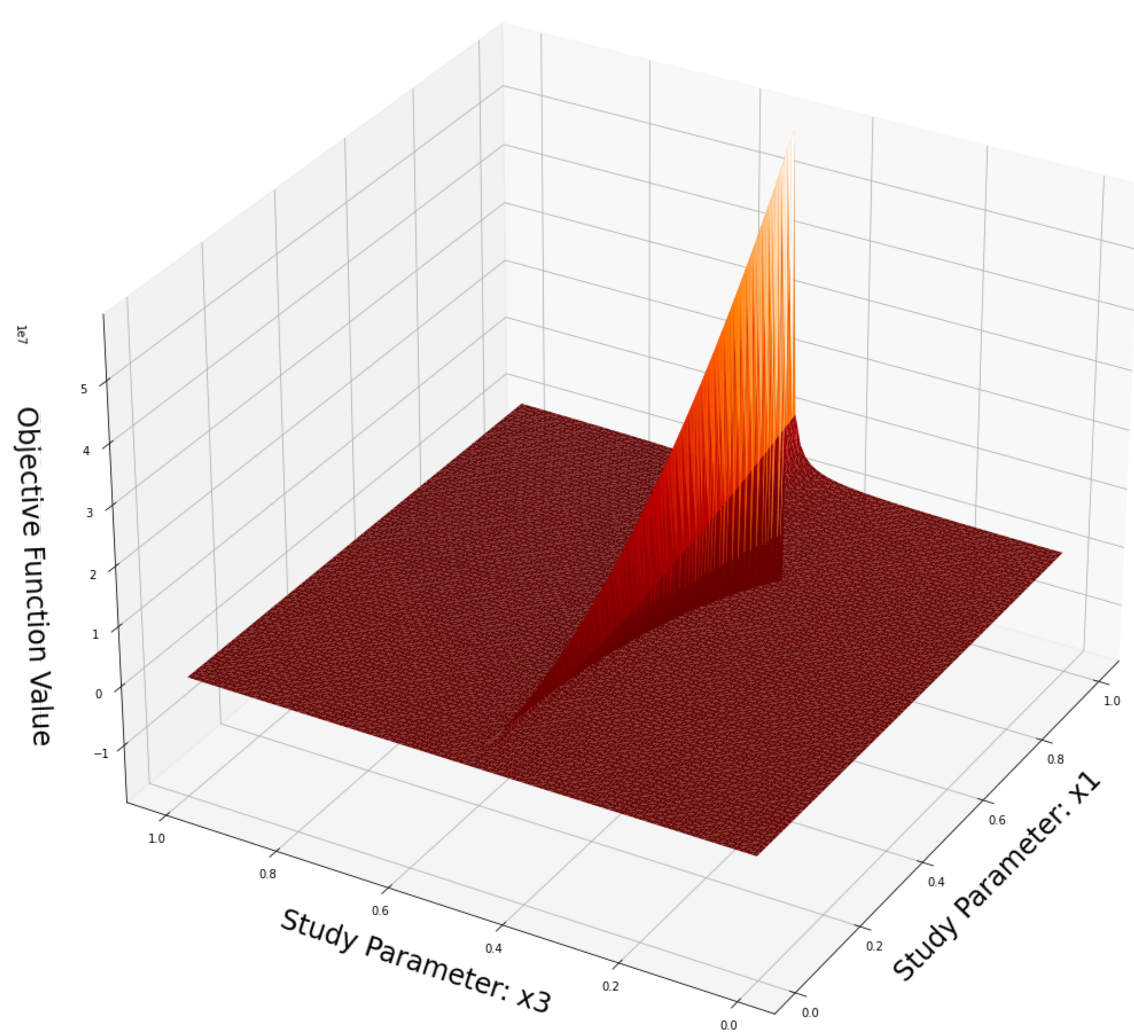
In [40]:
```
param_study2D(1,2,0.01,1,100,210)
```



This 2d design study shows that there is a hyperbolic relationship between the radius of the exit throat and the inner radius of the pressure vessel ax design variable x_2 changes.

In [41]:
```
param_study2D(1,3,0.01,1,100,210)
```

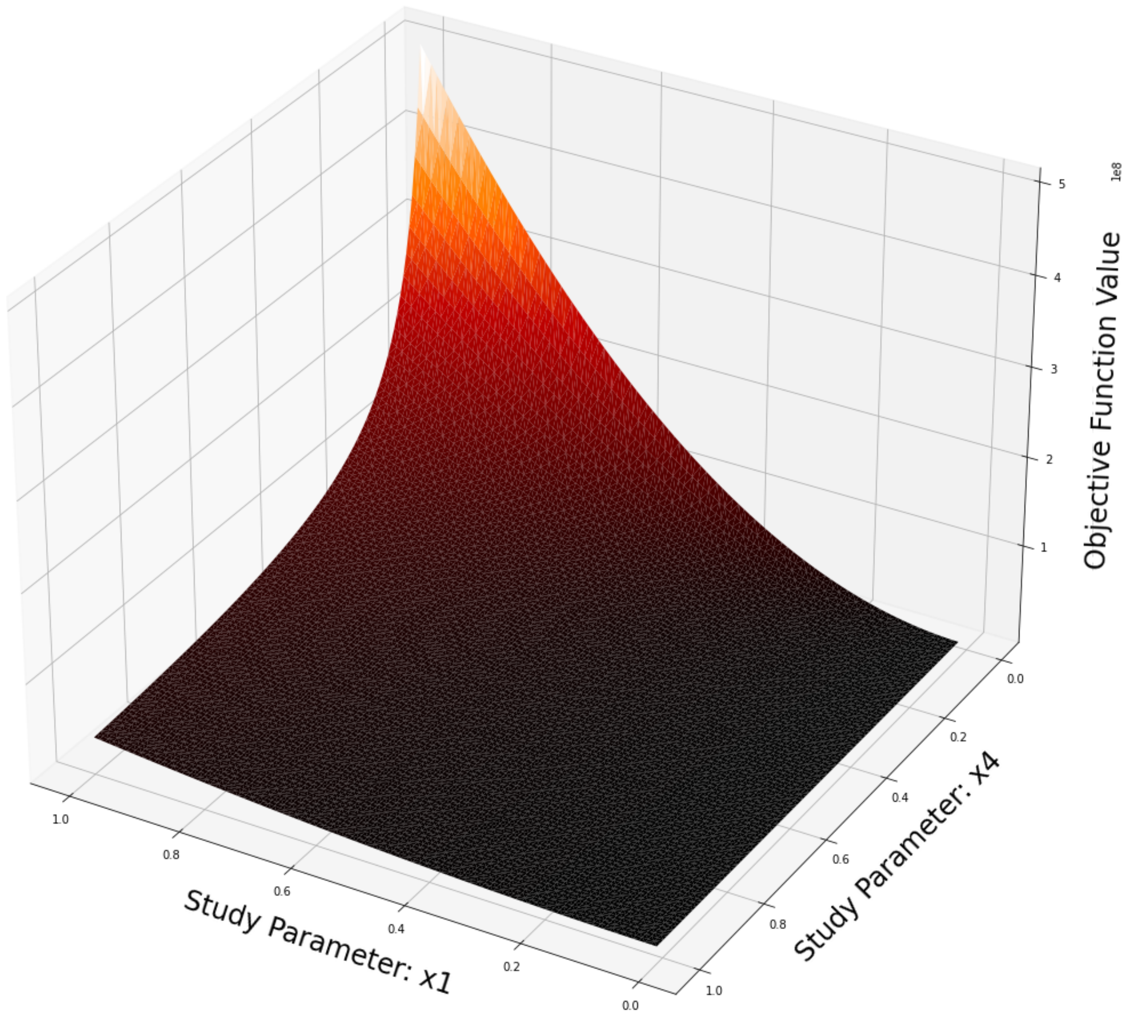
--- Parameter 1 & 3 study ---



This 2d design study shows that there is a hyperbolic relationship between the radius of the exit throat and the objective function as design variable x_3 changes.

In [45]:
```
param_study2D(1,4,0.01,1,100,120)
```

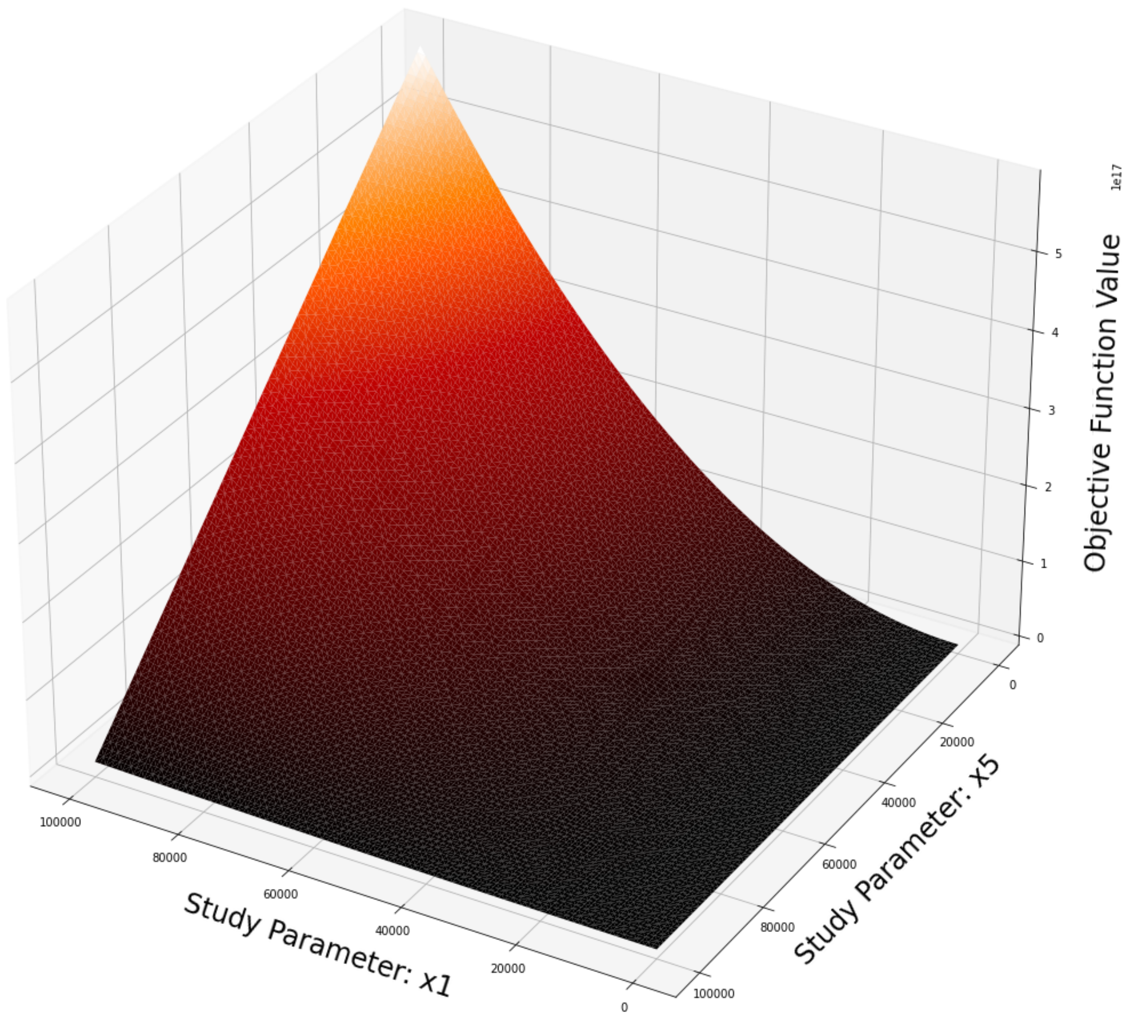
--- Parameter 1 & 4 study ---



This 2d design study shows that there is an exponential relationship between the objective function and design variables x_4 and x_4. x_4 can be seen to approach infinity faster and thus has a stronger exponential relationship.

In [47]:
```
param_study2D(1,5,0.01,100000,100,120)
```

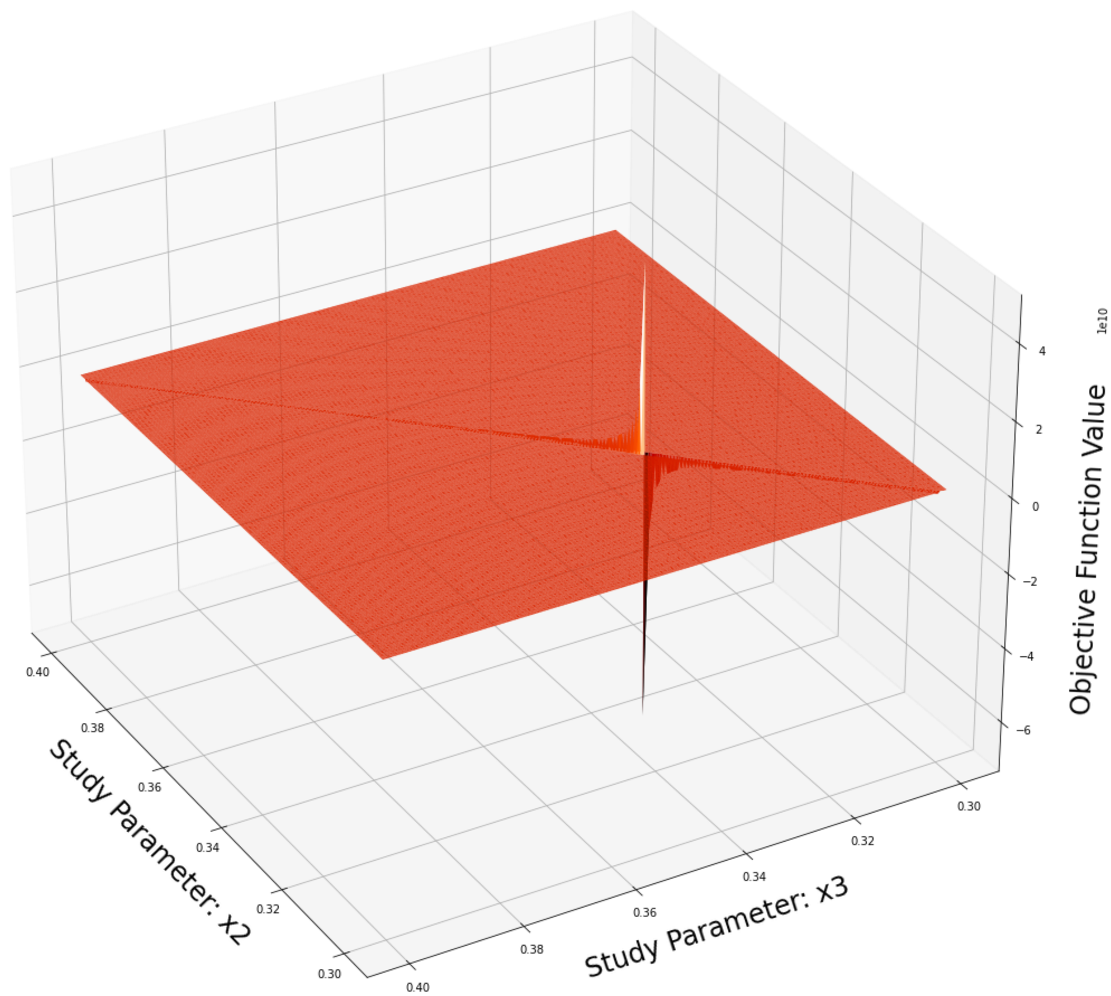
--- Parameter 1 & 5 study ---



This 2d design study shows that there is an exponential relationship between the objective function and design variable x_1 and a linear relatinship between the objective function and design variable x_5.

In [53]:
```
param_study2D(2,3,0.3,.4,250,150)
```

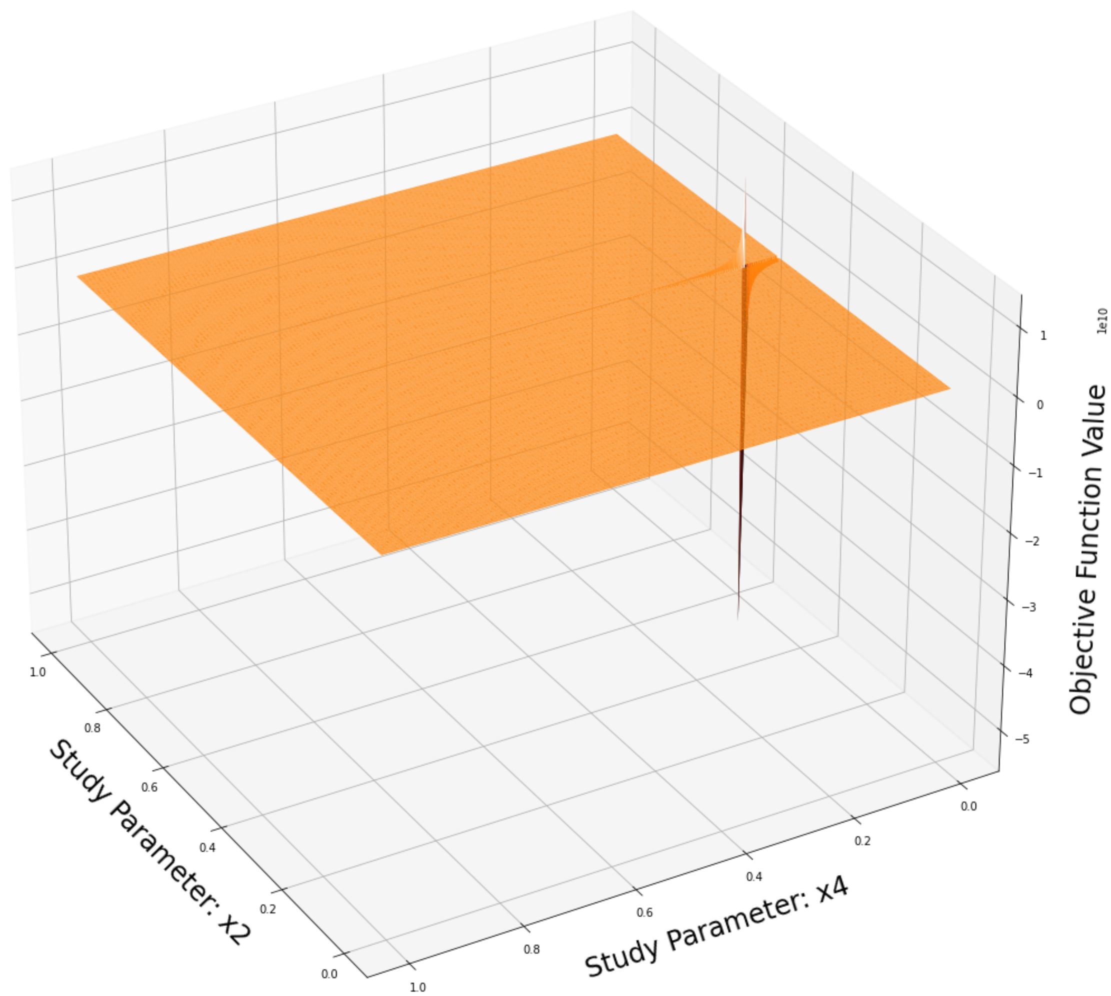
--- Parameter 2 & 3 study ---



This 2D design study shows that there is a point where design variables x_2 and x_3 approach positive and negative infinity. This is the location where the design approaches 0 pressure vessel thickness.

In [56]:
```
param_study2D(2,4,0,1,250,150)
```

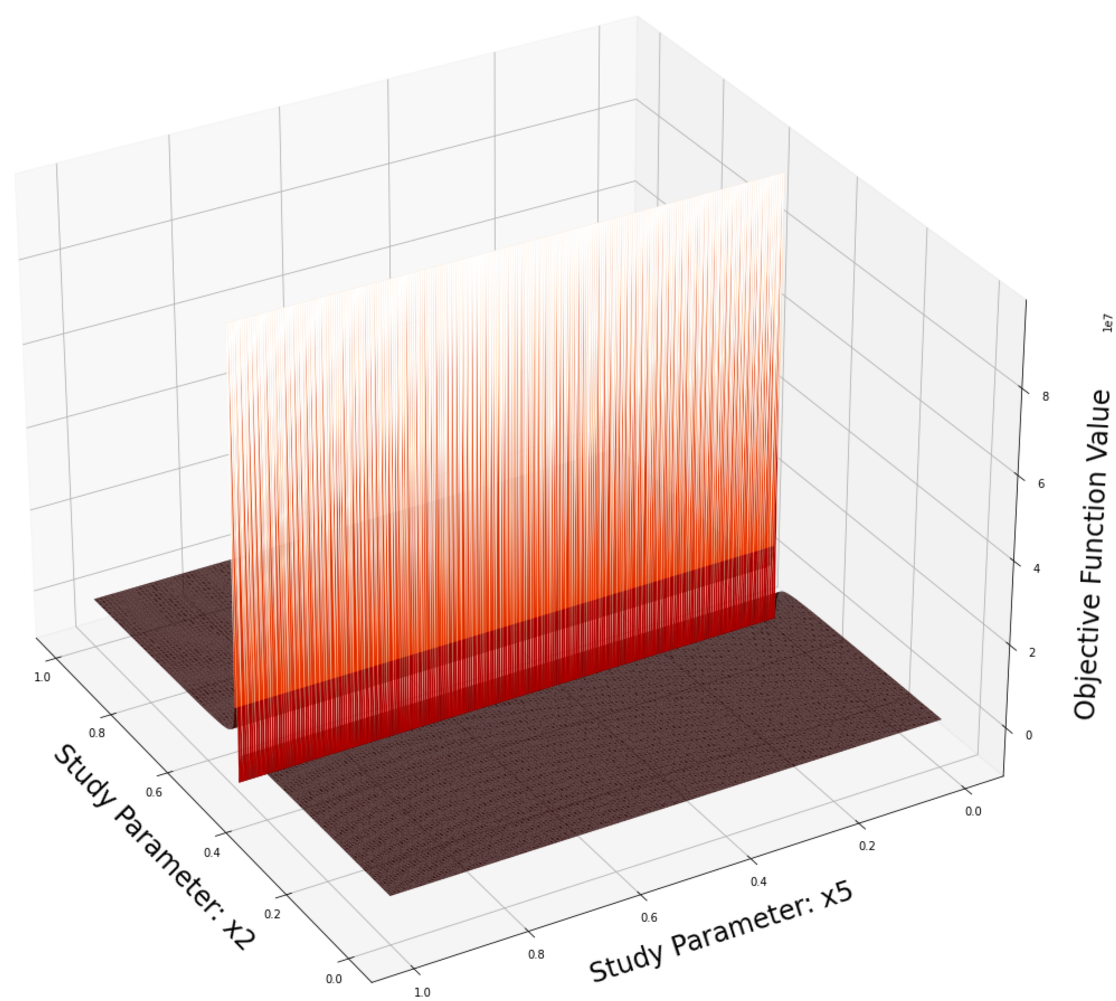
--- Parameter 2 & 4 study ---



This 2D design study shows that there is a point where design variables x_2 and x_4 approach positive and negative infinity.

In [57]:
```
param_study2D(2,5,0,1,250,150)
```

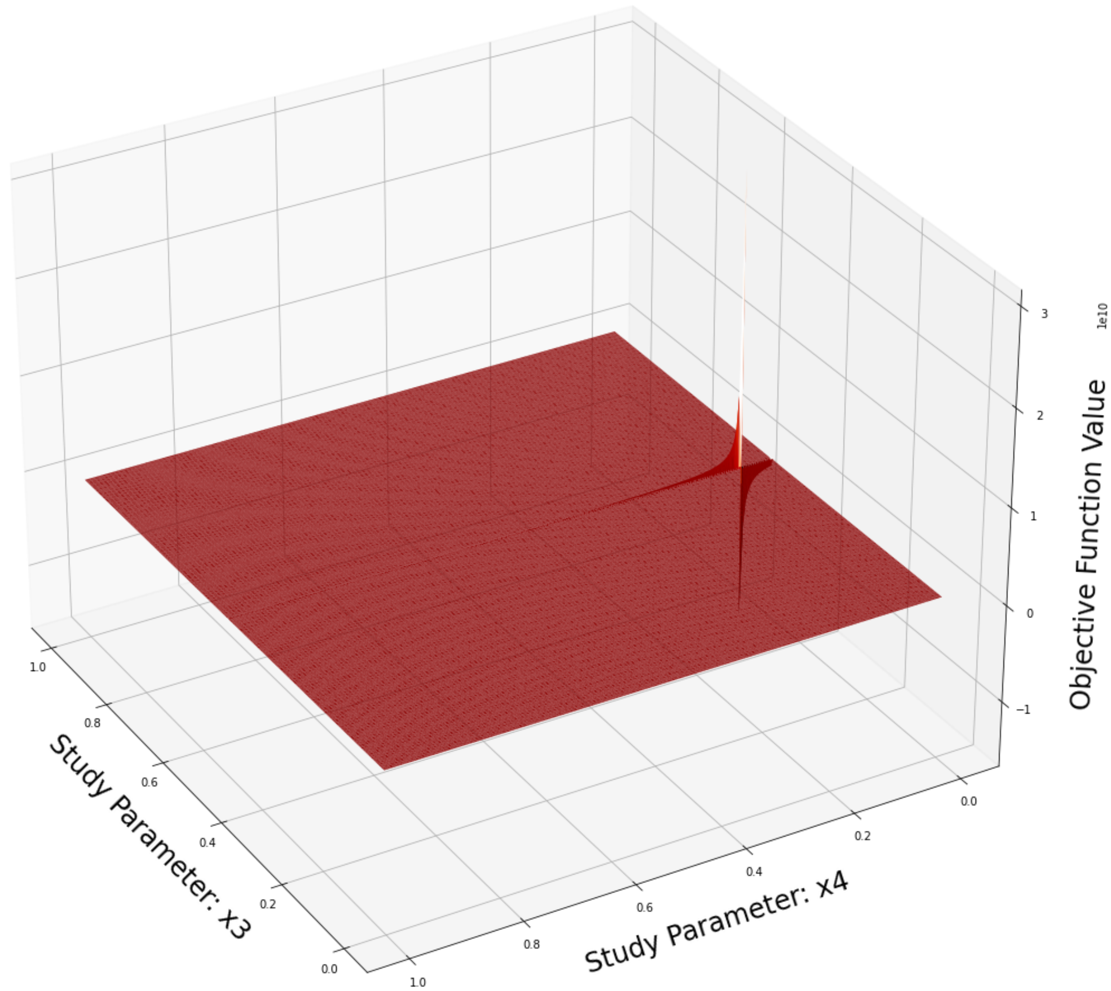
--- Parameter 2 & 5 study ---



This 2D design study shows that design prameter x_2 causes the objective function value to approach positive and negative infinity around the value of .5. This is due to the fact that the thickness of the pressure vessel approaches 0 at this value in the study.

In [58]:
```
param_study2D(3,4,0,1,250,150)
```

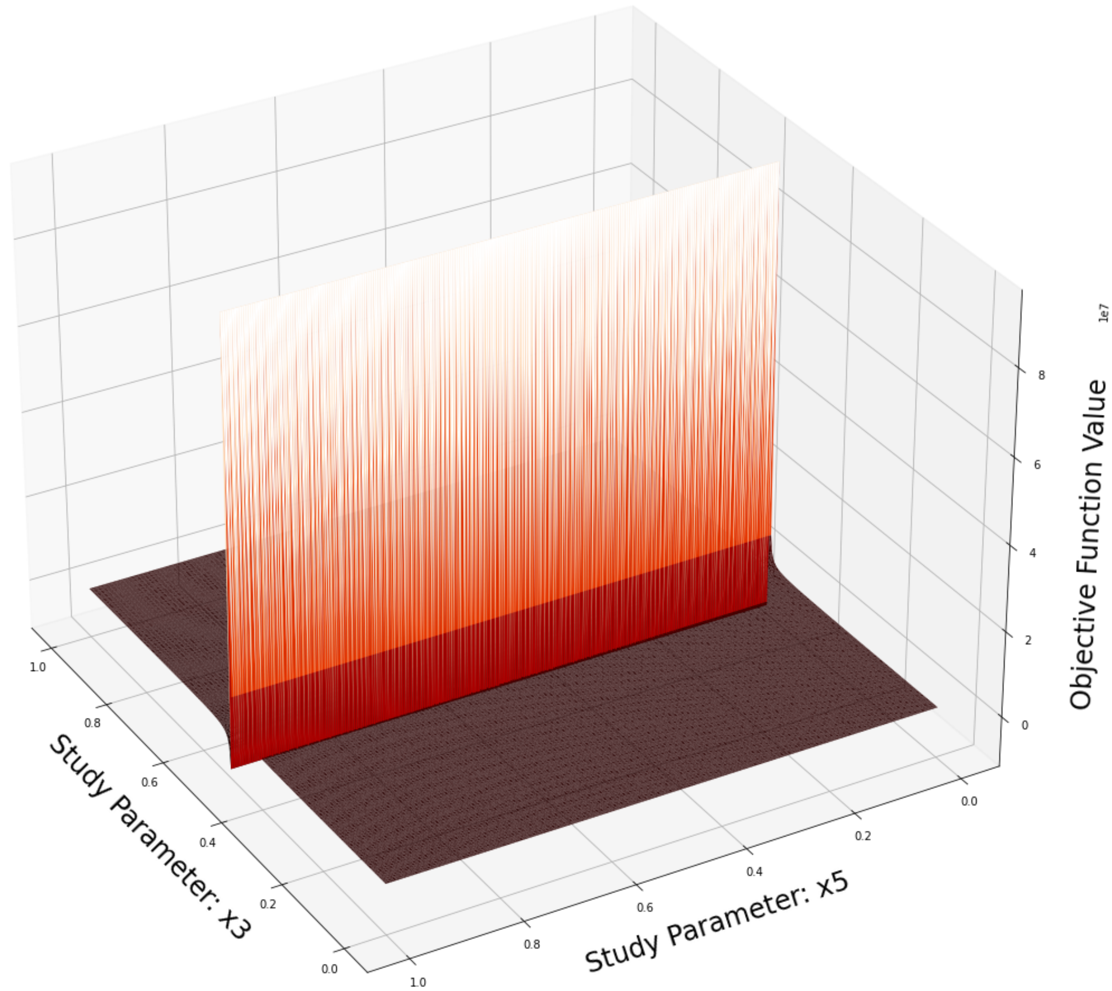
--- Parameter 3 & 4 study ---



This 2D design study shows that there is a point where the value of the objective function approaches infinity and negative infinity when we vary design variables x_3 and x_4.

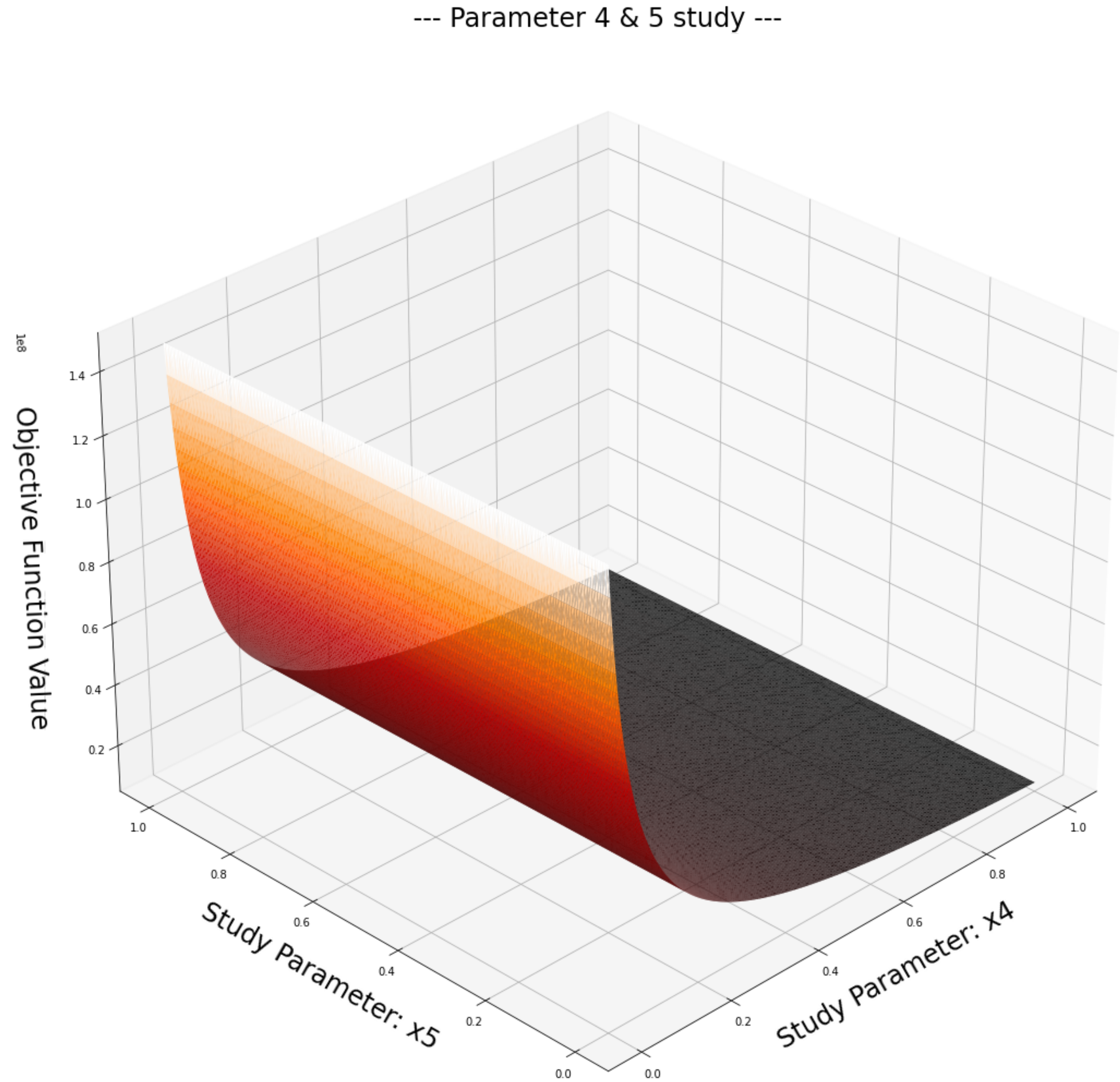
In [59]:
```
param_study2D(3,5,0,1,250,150)
```

--- Parameter 3 & 5 study ---



This 2D design study shows that there is a point where the value of the objective function approaches infinity and negative infinity when we vary design variable x_3.

In [63]:
```
param_study2D(4,5,0,1,250,225)
```



This 2D design study shows that there is an exponential relationship for x_4 and the objective function value.

This concludes the 2D design studies.

Now, we want to define all design variables except two design variables. So that our objective function is 2D and we can visualize the data.

In [34]:
```
a = a.subs(x1,x3)#104 cm radius
a = a.subs(x2,.094)#inner radius of rocket
#a = a.subs(x3,.104)#outer radius of the rocket
#a = a.subs(x4,.109)#109 mm height
a = a.subs(x5,101325+861845)#125 psi converted to pascals internal pressure, t
his value was chosen within an assumed range
a = a.subs(po,101325)#101325 pascal atmopsheric pressure
a = a.subs(rho,1.24*.2)#1.24 Kg/m^3 with 20% infill density
a = a.subs(g,-9.81)#m/s^2 gravitational constant
```

Now the objective function becomes,

In [35]:
```
a
```

Out[35]:

$$\frac{3475181.4516129\pi x_3^2}{\pi x_4\left(x_3^2 - 0.008836\right) + 1.33333333333333\pi\left(x_3^3 - 0.000830584\right)} + 9.81$$

We divide by 9.81 to convert acceleration function to gs

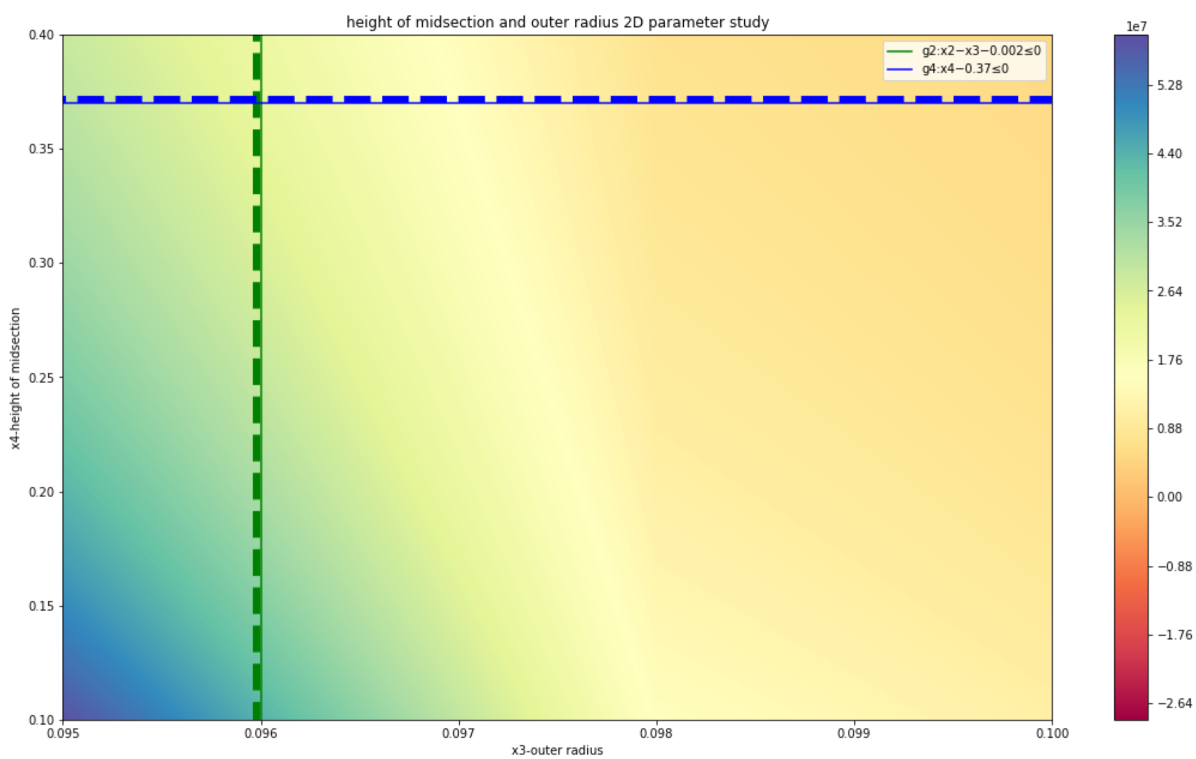
In [36]:
```
a=a/9.81
a
```

Out[36]:

$$\frac{354248.873762783\pi x_3^2}{\pi x_4\left(x_3^2 - 0.008836\right) + 1.33333333333333\pi\left(x_3^3 - 0.000830584\right)} + 1.0$$

In [37]:
```
delta = 0.003
x3max=.1
x3min=.095
x4max=.4
x4min=.1
x3 = np.arange(x3min-delta,x3max+delta, delta)
x4 = np.arange(x4min-delta,x4max+delta, delta)
X3,X4 = np.meshgrid(x3,x4)
a = a.subs(x3,X3)
a = a.subs(x4,X4)
fx = 1+((354248.8737*pi*X3**2)/(pi*X4*(X3**2-.008836)+1.3333*pi*(X3**3-.000830
584)))
```

In [38]:
```
fig,ax=plt.subplots()
CS=ax.contourf(X3,X4,fx,1000,cmap='Spectral')
plt.xlim(x3min,x3max)
plt.ylim(x4min,x4max)
plt.xlabel('x3-outer radius')
plt.ylabel('x4-height of midsection')
plt.colorbar(CS)
fig.set_size_inches(18.5, 10.5)
x3 = np.arange(-1,2,delta)
x4 = np.arange(-1,2,delta)
x2 = .094
g4 = x3*0+x2+.002
g5 = x4*0+.37



g1p=plt.plot(g4*.9998,x3,color='green',linewidth=6,linestyle='--')
g2p=plt.plot(g4,x3,color='green',label='g2:x2-x3-0.002≤0')
g3p=plt.plot(x4,g5*1.005,color='blue',linewidth=6,linestyle='--')
g4p=plt.plot(x4,g5,color='blue',label='g4:x4-0.37≤0')
plt.title('height of midsection and outer radius 2D parameter study')

legend = plt.legend(loc=0)
```



In [39]:
```
V
```

Out[39]: $\pi h\left(-r_i^2+r_o^2\right)+1.33333333333333\pi\left(-r_i^3+r_o^3\right)$

In [ ]:

In [ ]:

In [ ]:

In [ ]: