1) $f(x_1, x_2) = 2x_1^2 - 4x_1x_2 + 1.5x_2^2 + x_2$

Show stationary point (zero gradient) of the func^n is a saddle (indefinite Hessian)

$\rightarrow f(x_1, x_2) = 2x_1^2 - 4x_1x_2 + 1.5x_2^2 + x_2$

Finding gradient.

$g_1 = \nabla f_{x_1} = 4x_1 - 4x_2$ — ①

$g_2 = \nabla f_{x_2} = -4x_1 + 3x_2 + 1$ — ②

$$g(x_1, x_2) = \begin{bmatrix} 4x_1 - 4x_2 \\ -4x_1 + 3x_2 + 1 \end{bmatrix}$$

Finding Hessians;

$$H(x_1, x_2) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_2 \partial x_1} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

$$\therefore H = \begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix}$$

for stationary point (zero gradient) we equate ① & ② to zero.

$$\therefore 4x_1 - 4x_2 = 0$$

$$-4x_1 + 3x_2 + 1 = 0$$

$\therefore$ We get.

$$4x_1 = 4x_2, \quad \therefore x_1 = x_2$$

$$-4x_2 + 3x_2 + 1 = 0$$

$$\therefore -x_2 + 1 = 0$$

$$\therefore x_2 = 1$$

Also, $x_1 = 1$

$$(x_1, x_2) \equiv (1, 1)$$

This is the stationary point.

finding the eigen values.

$$Hx = \lambda x$$

$$\therefore (H - \lambda I)x = 0$$

$$\therefore \left(\begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) x = 0$$

$$\therefore \begin{bmatrix} 4-\lambda & -4 \\ -4 & 3-\lambda \end{bmatrix} x = 0$$

$$\therefore (4-\lambda)(3-\lambda) - 16 = 0$$

$$\therefore 12 - 7\lambda + \lambda^2 - 16 = 0$$

$$\therefore \lambda^2 - 7\lambda - 4 = 0$$

Solving with quadratic formula.

$\lambda = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$, we get.

$\lambda_1 = 7.53112$
$\lambda_2 = -0.53112$

$\because \lambda_1 > 0$ & $\lambda_2 < 0$

H is indefinite.

Thus we have a saddle point @ (1, 1)

- Find directions of downslopes away from saddle

Using Taylor's expansion show that.

$$f(x_1, x_2) = f(1,1) + (a\partial x_1 - b\partial x_2)(c\partial x_1 - d\partial x_2)$$

such that

$$f(x_1, x_2) - f(1,1) = (a\partial x_1 - b\partial x_2)(c\partial x_1 - d\partial x_2) < 0$$

$\partial x_1 = x_1 - 1$
$\partial x_2 = x_2 - 1$

→ Using 2nd order Taylor's expansion, we have.

$$f(x) \approx f(x_0) + \nabla_x f\Big|_{x_0}^{T} (x - x_0)$$

$$+ \frac{1}{2} (x - x_0)^T H\Big|_{x_0} (x - x_0)$$

@ (1, 1)

$$\therefore f(x_1, x_2) \approx f(1, 1) + g_0^T \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}$$

$$+ \frac{1}{2} \begin{bmatrix} x_1 - 1, & x_2 - 1 \end{bmatrix} H_0 \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}$$

~~for g_0, we get~~

for $f(1, 1)$ we get

$$f(1, 1) = 2(1)^2 - 4(1)(1) + 1.5(1)^2 + 1$$

$$= 0.5$$

for g_0, we get

$$g_0 = \begin{bmatrix} 4(1) - 4(1) \\ -4(1) + 3(1) + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

H_0 stay the same as H.

$x_1 - 1 = \partial x_1$, $x_2 - 1 = \partial x_2$ - from above

$$\therefore f(x_1, x_2) = 0.5 + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} \partial x_1 \\ \partial x_2 \end{bmatrix}$$

$$+ \frac{1}{2} \begin{bmatrix} \partial x_1 , & \partial x_2 \end{bmatrix} \begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix} \begin{bmatrix} \partial x_1 \\ \partial x_2 \end{bmatrix}$$

$$= 0.5 + \frac{1}{2} \begin{bmatrix} \partial x_1 , & \partial x_2 \end{bmatrix} \begin{bmatrix} 4 \partial x_1 - 4 \partial x_2 \\ -4 \partial x_1 + 3 \partial x_2 \end{bmatrix}$$

$$= 0.5 + \frac{1}{2} \left(4 \partial x_1^2 - 4 \partial x_2 \partial x_1 - 4 \partial x_1 \partial x_2 + 3 \partial x_2^2 \right)$$

$$= 0.5 + \frac{1}{2} \left[4 \partial x_1^2 - 8 \partial x_1 \partial x_2 + 3 \partial x_2^2 \right]$$

$$= 0.5 + \left[2 \partial x_1^2 - 4 \partial x_1 \partial x_2 + \frac{3}{2} \partial x_2^2 \right]$$

Solving the quadratic eqn for roots.

Roots are $\partial x_1 = \frac{3}{2}\partial x_2$, $\partial x_1 = \partial x_2/2$

★ $\therefore f(x_1, x_2) = 0.5 + \left[\left(\partial x_1 - \frac{3}{2}\partial x_2\right)\left(\partial x_1 - \partial x_2/2\right)\right]$

Comparing with given equation we have

$$a = 1, \quad b = \frac{3}{2}, \quad c = 1, \quad d = \frac{1}{2}$$

– Direction of downslopes.

The given condition is

$$f(x_1, x_2) - f(1,1) = (a\,\partial x_1 - b\,\partial x_2)(c\,\partial x_1 - d\,\partial x_2) < 0$$

$$\therefore f(x_1, x_2) - 0.5 = \left(\partial x_1 - \frac{3}{2}\partial x_2\right)\left(\partial x_1 - \partial x_2 \cdot \frac{1}{2}\right) < 0$$

2) find point in plane $x_1 + 2x_2 + 3x_3 = 1$ in $\mathbb{R}^3$ nearest to the point $(-1, 0, 1)^T$.

Is this a convex problem?

→ $x_1 + 2x_2 + 3x_3 = 1$

To unconstrain it we can write.

$$x_1 = 1 - 2x_2 - 3x_3$$

$\therefore$

for the main equation we have.

Using lagrange multiplier.

We get.

$$d = \sqrt{(x_1+1)^2 + x_2^2 + (x_3-1)^2} \quad \text{— distance}$$

Thus, we have.

$$\min \; (x_1+1)^2 + x_2^2 + (x_3-1)^2$$

→ Now using this we can write the unconstrained equation.

$$\min (1 - 2x_2 - 3x_3 + 1)^2 + x_2^2 + (x_3 - 1)^2$$

this is now unconstrained.

Now we find the gradient & Hessian.

$$g_1 = \nabla_{x_2} f = \frac{\partial}{\partial x_2} (2 - 2x_2 - 3x_3)^2 + x_2^2 + (x_3 - 1)^2$$

$$= 2(-2)(2 - 2x_2 - 3x_3) + 2x_2$$

$$g_2 = \nabla_{x_3} f = \frac{\partial}{\partial x_3} (2 - 2x_2 - 3x_3)^2 + x_2^2 + (x_3 - 1)^2$$

$$= (2)(-3)(2 - 2x_2 - 3x_3) + 2(x_3 - 1)$$

$$g = \begin{bmatrix} (-4)(2 - 2x_2 - 3x_3) + 2x_2 \\ (-6)(2 - 2x_2 - 3x_3) + 2(x_3 - 1) \end{bmatrix}$$

$$= \begin{bmatrix} -8 + 8x_2 + 12x_3 + 2x_2 \\ -12 + 12x_2 + 18x_3 + 2x_3 - 2 \end{bmatrix}$$

$$g = \begin{bmatrix} 10x_2 + 12x_3 - 8 \\ 12x_2 + 20x_3 - 14 \end{bmatrix} \quad - (1)$$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_3 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_3} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix}$$

$$H = \begin{bmatrix} 10 & 12 \\ 12 & 20 \end{bmatrix}$$

finding eigen values.

$$Hx = \lambda x$$

$$(H - \lambda I)x = 0.$$

$$\therefore \left(\begin{bmatrix} 10 & 12 \\ 12 & 20 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) x = 0$$

$$\therefore \begin{pmatrix} (10-\lambda) & 12 \\ 12 & (20-\lambda) \end{pmatrix} x = 0$$

$\therefore (10-\lambda)(20-\lambda) - 144 = 0$

$\therefore \lambda^2 - 30\lambda + 56 = 0$

Solving this quadratic eq^n for roots we get;

$\lambda = 2 \quad , \quad \lambda = 28.$

$\therefore$ both $\lambda > 0$, H is positive definite.

This is a strictly convex problem.

- We also need to find out the values of x_1, x_2, x_3

Using gradient as zero & equating to formulated equations.

$\therefore \quad 10x_2 + 12x_3 - 8 = 0$

$12x_2 + 20x_3 - 14 = 0$

$\therefore \quad x_2 = (8 - 12x_3)/10$

$\therefore \quad \frac{12}{10}(8 - 12x_3) + 20x_3 - 14 = 0$

$\therefore x_3 = 0.7857$

$x_3 = 0.7857$

$\therefore \; x_2 = -0.14284$

$\therefore \quad x_1 = 1 - 2x_2 - 3x_3$

$\therefore \; x_1 = -1.0714$

These are the values at stationary points.

3) Prove hyperplane is a convex set

$\mathbb{R}^n$: $a^T x = c$ for $x \in \mathbb{R}^n$

a is normal direction of hyperplane
c is a constant

→ As a set we can write

$$S = \{x \in \mathbb{R}^n \mid a^T x = c\}$$

We choose two points x_1 & x_2 from the hyperplane

$$x_1, x_2 \in S$$

Using the equation

$\forall \lambda \in [0, 1]$ $\lambda x_1 + (1-\lambda) x_2$ we check if it belongs to hyperplane.

For this it needs to satisfy the eqn

$$a^T x = c$$

$\therefore$ Let $x = \lambda x_1 + (1-\lambda) x_2$

$\therefore \; a^T(\lambda x_1 + (1-\lambda) x_2) = c$ — (to prove)

$LHS = \lambda a^T x_1 + (1-\lambda) a^T x_2$

$\because$ x_1 & x_2 both belong to the hyperplane

We can write

$$a^T x_1 = c \; ; \quad a^T x_2 = c$$

Substituting these in LHS, we get.

$$LHS = \lambda c + (1-\lambda) c$$

$$= \lambda c + c - \lambda c$$

$$= c$$

$$= RHS.$$

Thus, it is proved that a hyperplane is a convex set.

4)

$$\min_{p} \max_{k} \{h(a_k^T p, I_t)\}$$

s.t. $0 \leq p_i \leq p_{max}$.

$p := [p_1, \ldots, p_n]^T$ - power output of n lamps.

a_k for $k = 1, \ldots, m$, fixed parameters for m mirrors

I_t - target intensity level

$$h(I, I_t) = \begin{cases} I_t / I & \text{if } I \leq I_t \\ I / I_t & \text{if } I_t \leq I \end{cases}$$

a) Show that problem is convex.

→ $$\min_{p} \max_{k} \{h(a_k^T p, I_t)\}$$

The max of the equation is maximum of a bunch of h i.e.

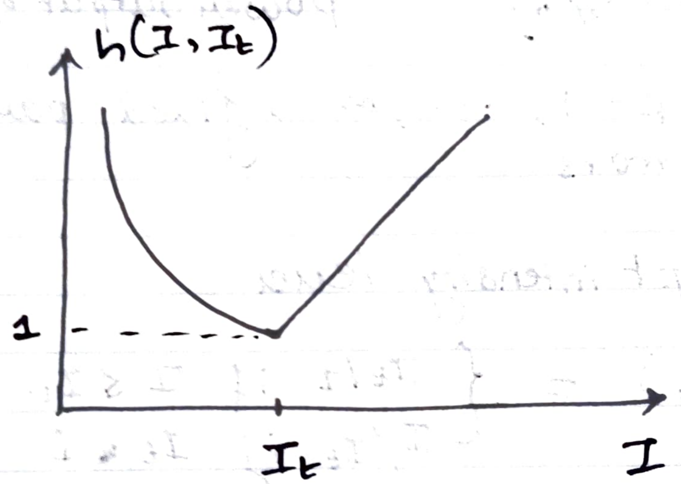
$$\max\{h_1, h_2, h_3, \ldots, h_N\}$$

where each h consists of $(a_k^T p, I_t)$

Now for h

$$h(I, I_t) = \begin{cases} I_t / I & I \le I_t \\ I / I_t & I > I_t \end{cases}$$

looking at the graph of this function



there we clearly see that its a convex function with p.s.d Hessian.

but h also consists of $(a_k^T p)$

So we also have to prove that part to be convex for h to be convex wrt. p.

For that we find the hessian of the function wrt. p.

$$\therefore \frac{\partial h}{\partial p} = \frac{\partial}{\partial p} h(a^T p, I_t)$$

– (for one of the h)

$$\therefore \frac{\partial h}{\partial p} = \frac{dh}{dI} \frac{\partial a^T p}{\partial p} \quad \text{— (chain rule)}$$

$$g = h' \cdot a^T$$

$$\therefore H = \frac{\partial^2 h}{\partial p^2} = \frac{\partial}{\partial p}(h' \cdot a^T)$$

$$= \frac{\partial h'}{\partial I} \cdot \frac{\partial a^T p}{\partial p} a^T$$

$$= h'' \cdot a \cdot a^T$$

$$= h'' a a^T$$

a_k goes from $(k = 1, \dots, M]$

So aa^T should give us a $M \times M$ matrix.

from the graph we know h is convex

$\therefore h'' \geq 0$ (as the hessian of curve will be positive and hessian of line would be zero as it is linear) (p.s.d.)

Now for aa^T

for $a \in \mathbb{R}^{M \times N}$; $a = a^T$

$\forall\, w \in \mathbb{R}^m, w \neq 0$
we can write.

$$w^T a w \geq 0$$

$\therefore$ for the equation, we can write

$$w^T a_k a_k^T w$$

let $a_k^T w = y_i$ then $w^T a_k = y_i$

as it is just a number.

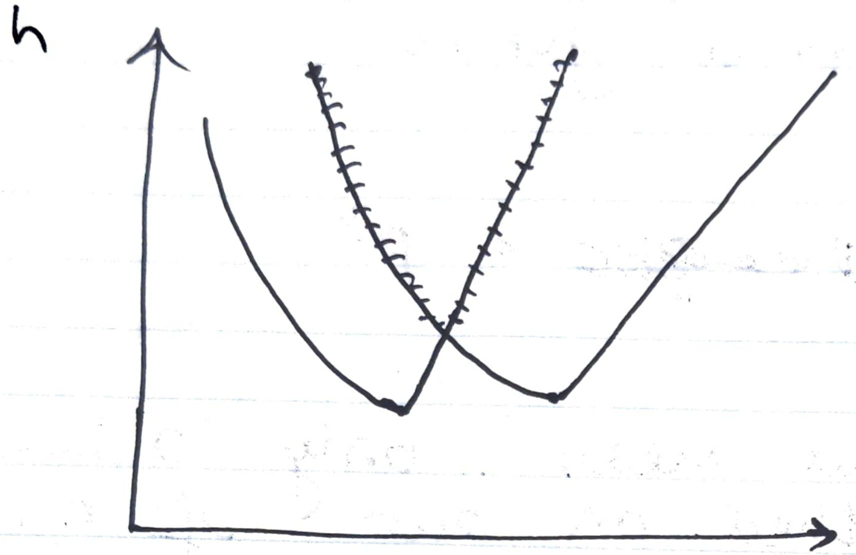
then ~~$\quad$~~ $w^T a_k a_k^T w = y_i^2 \geq 0$

this would mean $a_k a_k^T$ is a positive semi-definite matrix.

thus h is ~~[illegible]~~ convex function wrt. p.

Now a bunch of h in the main equation, each of them are a convex function.

In graphical terms; we can say.



Max of a set of convex functions is also a convex function.

Thus;

$$\min_{p} \max \{ h(a_t^T p, I_t) \}$$

is convex.

b) If we require the overall power output of any of the 10 lamps to be less than p^*, will the problem have a unique solution.

→ The equation we get is

$$\sum_{i=1}^{10} (p_i) < p^*$$

This would mean, only 10 lamps are switched on out of n lamps i.e.

$$\begin{array}{c} \text{10th lamp} \\ \downarrow \end{array}$$

$$\begin{bmatrix} 1, 1, \dots 1, 0, 0, 0 \dots \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix} < p^*$$

This now turns into a linear equation. It is a feasible set so it is convex.

Thus we get an unique solution.

c) If we require no more than 10 lamps to be switched on ($p>0$), will the problem have a unique solution?

→ for no more than 10 lamps we do not have a feasible set which is convex anymore

We might not reach the target intensity as prescribed

Hence we won't get a unique solution.

We might get multiple local solutions at some points.

5) $C(x)$ - cost of producing x amount of product A

Y - price set for product

profit is given as;

$$C^*(y) = \max_x \{xy - c(x)\}$$

Show that $C^*(y)$ is a convex function wrt y.

→ The given equation is $xy - c(x)$

It is a linear equation. The given function is the max of many linear functions.

Now if we take the Hessian of the linear function to check convexity.

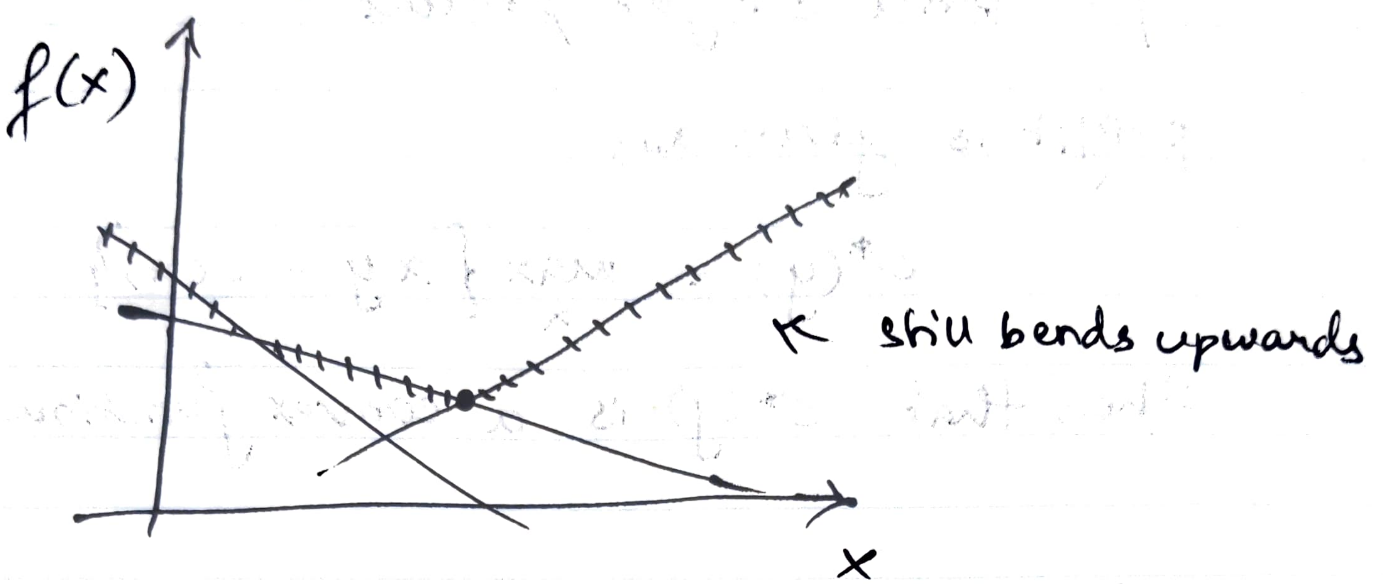
If we differentiate the function twice to find Hessian we get a zero.

Now, if $H = 0$ it is positive semi definite

Thus the linear function is convex.

$$\boxed{\begin{aligned} g &= \frac{\partial f}{\partial y} = x && \text{- gradient.} \\ H &= \frac{\partial^2 f}{\partial y^2} = 0 && \text{- Hessian.} \end{aligned}}$$

Thus if we have a max of such many linear functions, we can show that as



We can see it is still convex.

Thus the max of linear function is also convex.

from the question we can conclude that the profit of that product is a convex function with respect to the price.

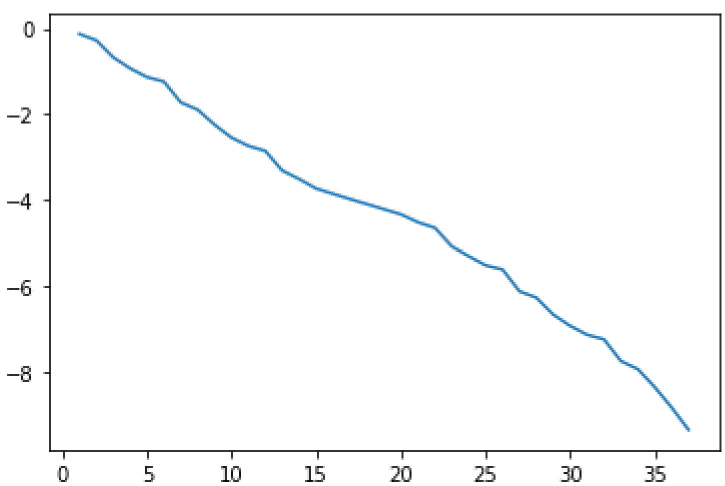
i.e. there is only one solution for max. profit.

In [2]:

```
import numpy as np
import matplotlib.pyplot as mpl
def f(x):
    return ((2 - (2*x[0]) - (3*x[1]))**2) + (x[0]**2) + ((x[1]-1)**2)
def gradient(x):
    return np.array([10*x[0] + 12*x[1] - 8, 12*x[0] + 20*x[1] - 14])
#def hessian(x):
    #return np.matrix([[10,12],[12,20]])

t = 0.5
x_ini = np.array([1,1])
gr_values = [] # output value list

# Indirect line search
def backtrack(x):
    alpha = 1
    while f(x - alpha*gradient(x)) > f(x) - (t * np.matmul(gradient(x).T,gradient(x)) *
        alpha = 0.5 * alpha
    return alpha


while np.linalg.norm(gradient(x_ini)) > 0.0001:  # epsilon  = 0.0001
    alpha = backtrack(x_ini) #step size
    x_ini = x_ini + alpha * (-gradient(x_ini))
    gr_values.append(f(x_ini))

#print(x_ini)
#print(gr_values)
#print(len(gr_values))
y_ax = []
for i in range(len(gr_values)-1):
    j = np.log10(gr_values[i] - gr_values[-1])
    y_ax.append(j)

x_ax = range(1,len(gr_values))
mpl.plot(x_ax,y_ax)
```

Out[2]:

```
[<matplotlib.lines.Line2D at 0x1e99b6ace20>]
```

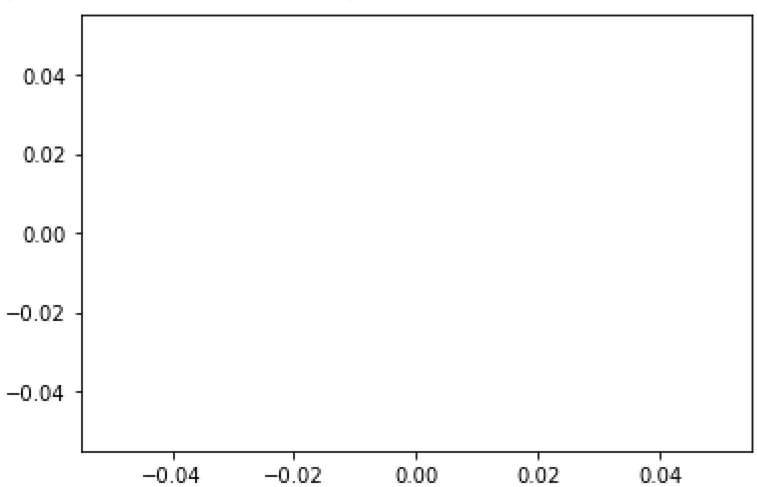


In [9]:

```
import numpy as np
import matplotlib.pyplot as mpl
```

```
def f(x):
    return ((2 - (2*x[0]) - (3*x[1]))**2) + (x[0]**2) + ((x[1]-1)**2)
def gradient(x):
    return np.array([10*x[0] + 12*x[1] - 8, 12*x[0] + 20*x[1] - 14])
def hessian(x):
    return np.array([[10,12],[12,20]])

t = 0.5
x_ini = np.array([1,1])
newt_values = [] # output value list

# Indirect line search
def backtrack(x):
    a1 = np.matmul(gradient(x).T,np.linalg.inv(hessian(x)))
    a2 = np.matmul(a1,gradient(x))
    alpha = 1
    while f(x - alpha*(-np.linalg.inv(hessian(x)))*gradient(x)).all() > (f(x) - (t* a2*
        alpha = 0.5 * alpha
    return alpha
#print(f(x_ini - alpha*gradient(x_ini)))
#print((f(x_ini) - (t*np.matmul(gradient(x_ini).T,(np.matmul(np.linalg.inv(hessian(x_in

while np.linalg.norm(gradient(x_ini)) > 0.0001:  # epsilon  = 0.0001
    alpha = backtrack(x_ini) #step size
    x_ini = x_ini + alpha *np.matmul(-np.linalg.inv(hessian(x_ini)),gradient(x_ini))
    newt_values.append(f(x_ini))

print(x_ini)
#print(newt_values)
#print(len(newt_values))
y_ax = []
for i in range(len(newt_values)-1):
    j = np.log10(newt_values[i] - newt_values[-1])
    y_ax.append(j)

x_ax = range(1,len(newt_values))
mpl.plot(x_ax,y_ax)
mpl.show()
```

```
[-0.14285714  0.78571429]
```




```
In [ ]:
```