Caleb Logemann
MATH 565 Continuous Optimization
Homework 1

1. Problem 1

   Compute the gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$ of the Rosenbrock function

$$f(x) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$$

The gradient of f is defined as $\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)^T$. The first derivatives of f are stated below.

$$\begin{aligned}\frac{\partial f}{\partial x_1} &= -400x_1\left(x_2 - x_1^2\right) - 2(1 - x_1)\\ \frac{\partial f}{\partial x_2} &= 200\left(x_2 - x_1^2\right)\end{aligned}$$

So the gradient of f is

$$\nabla f(x) = \begin{bmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{bmatrix}$$

The Hessian of f is the matrix

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

The second derivatives are shown below.

$$\begin{aligned}\frac{\partial^2 f}{\partial {x_1}^2} &= \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_1}\right)\\ &= -400\left(x_2 - x_1^2\right) + 800x_1^2 + 2\\ &= -400\left(x_2 - 3x_1^2\right) + 2\\ \frac{\partial^2 f}{\partial x_1 \partial x_2} &= \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_2}\right)\\ &= -400x_1\\ \frac{\partial^2 f}{\partial {x_2}^2} &= \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_2}\right)\\ &= 200\end{aligned}$$

Note that $\frac{\partial^2 f}{\partial x_2 \partial x_1} = \frac{\partial^2 f}{\partial x_1 \partial x_2}$ as f has continuous second derivatives. Thus the Hessian of f is

$$\nabla^2 f(x) = \begin{bmatrix} -400(x_2 - 3x_1^2) + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}$$

2. Problem 8

   Suppose that f is a convex function. Show that the set of global minimizers of f is a convex set.

Proof. Let f be a convex function. This implies that

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

for $\lambda \in [0,1]$ and $x, y \in \mathbb{R}^n$. Let $S = \{x^* \in \mathbb{R}^n : f(x^*) \leq f(x) \forall x \in \mathbb{R}^n\}$ be the set of all global minimizers. Let $x_1^*, x_2^* \in S$ and let $\lambda \in [0,1]$. Note that $f(x_1^*) = f(x_2^*)$ because in order for both of these to be global minimizers they must have the same functional value. Now consider the point $\lambda x_1^* + (1-\lambda)x_2^*$. Since f is convex.

$$\begin{aligned} f(\lambda x_1^* + (1-\lambda)x_2^*) &\leq \lambda f(x_1^*) + (1-\lambda)f(x_2^*) \\ &= \lambda f(x_1^*) + (1-\lambda)f(x_1^*) \\ &= f(x_1^*) \end{aligned}$$

This shows tha Also since x_1^* is a global minimizer.

$$f(\lambda x_1^* + (1-\lambda)x_2^*) \leq f(x_1^*) \leq f(x)$$

for any $x \in \mathbb{R}^n$. Thus $\lambda x_1^* + (1-\lambda)x_2^* \in S$ and is a global minimizer. This shows that S is convex. □

3. Problem 9

Consider the function $f(x_1, x_2) = (x_1 + x_2^2)^2$. At the point $x^T = (1,0)$ we consider the search direction $p^T = (-1,1)$. Show that p is a descent direction and find all minimizers of the problem (2.10).

In order to show that this is a descent direction, it is first necessary to find the gradient of f at $x^T = (1,0)$.

$$\nabla f = \begin{bmatrix} 2(x_1 + x_2^2) \\ 4x_2(x_1 + x_2^2) \end{bmatrix}$$

At the point $x^T = (1,0)$, the gradient is

$$\nabla f(x) = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

This is a descent direction if $p^T \nabla f(x) < 0$. Now note that $p^T \nabla f(x) = (-1) \times 2 + 1 \times 0 = -2 < 0$. Therefore p is a descent direction.

Secondly we want to find the minimizers of the problem.

$$\min_{\alpha > 0}\{f(x + \alpha p)\} = \min_{\alpha > 0}\{\phi(\alpha)\}$$

We can find the vector $x + \alpha p$ to be

$$x + \alpha p = \begin{bmatrix} 1 - \alpha \\ \alpha \end{bmatrix}$$

Now the function $\phi(\alpha)$ can be given as

$$\phi(\alpha) = (1 - \alpha + \alpha^2)^2$$

In order to find the minimizers of this function we must first find the derivative of ϕ.

$$\phi'(\alpha) = 2(\alpha^2 - \alpha + 1)(2\alpha - 1)$$

Now it is obvious that the roots of $\phi'(\alpha)$ are the roots of $2\alpha - 1$ and $\alpha^2 - \alpha + 1$. The first is zero when $\alpha = \frac{1}{2}$ and the second has no real roots. Therefore $\phi'(\alpha) = 0$ when $\alpha = \frac{1}{2}$. So this is the only candidate for a minimizer of $\phi(\alpha)$. To check that this is a minimizer note that $\phi'(0) = -2$ and $\phi'(1) = 2$. This shows that the slope goes from negative to positive and thus $\alpha = \frac{1}{2}$ is the only minimizer.

4. Problem 13
   Show that the sequence $x_k = 1/k$ is not Q-linearly convergent, though it does converge to zero.

   Consider the limit of $\frac{|x_{k+1}|}{|x_k|}$ as $k \to \infty$.

$$\begin{aligned}\lim_{k\to\infty}\left(\frac{|x_{k+1}|}{|x_k|}\right) &= \lim_{k\to\infty}\left(\frac{1/(k+1)}{1/k}\right)\\ &= \lim_{k\to\infty}\left(\frac{k}{k+1}\right)\\ &= 1 \not< 1\end{aligned}$$

   This shows that x_k is not Q-linearly convergent to 0. If x_k was Q-linearly convergent to 0 then this limit would be less than 1.

5. Problem 14
   Show that the sequence $x_k = 1 + (0.5)^{2^k}$ is Q-quadractically convergent to 1.

   First note that $|x_k - 1| = (0.5)^{2^k}$ and that

$$|x_{k+1} - 1| = (0.5)^{2^{(k+1)}} = \left((0.5)^{2^k}\right)^2 = |x_k - 1|^2$$

   Therefore for any $k \in \mathbb{N}$.

$$\frac{|x_{k+1} - 1|}{|x_{k+1} - 1|^2} = 1 \leq M$$

   This shows that the sequence $x_k = 1 + (0.5)^{2^k}$ converges Q-quadractically to 1.

6. Consider the following fixed point iteration scheme:

$$x_{k+1} = x_k - \frac{[g(x_k)]^2}{g(x_k + g(x_k)) - g(x_k)}$$

   Prove that if this method is converges to a root x^* of $g(x)$ such that $g'(x^*) \neq 0$ and $g''(x^*) \neq 0$, then the rate of convergence is quadractic: $p = 2$.

   Proof. First note the following two Taylor series for $g(x_k)$ and $g(x_k + g(x_k)$.

$$\begin{aligned}g(x_k) &= g(x^*) + (x_k - x^*)g'(x^*) + \frac{1}{2}(x_k - x^*)^2 g''(\xi_1)\\ g(x_k + g(x_k)) &= g(x_k) + g(x_k)g'(\xi_2)\end{aligned}$$

   where $\xi_1 \in (x_k, x^*)$ and $\xi_2 \in (x_k, x_k + g(x_k))$. Now consider the iteration scheme

$$\begin{aligned}x_{k+1} &= x_k - \frac{[g(x_k)]^2}{g(x_k + g(x_k)) - g(x_k)}\\ x_{k+1} - x^* &= x_k - x^* - \frac{[g(x_k)]^2}{g(x_k + g(x_k)) - g(x_k)}\end{aligned}$$

Now use the Taylor series for $g(x_k + g(x_k))$.

$$\begin{aligned}
x_{k+1} - x^* &= x_k - x^* - \frac{[g(x_k)]^2}{g(x_k) + g(x_k)g'(\xi_2) - g(x_k)} \\
x_{k+1} - x^* &= x_k - x^* - \frac{[g(x_k)]^2}{g(x_k)g'(\xi_2)} \\
x_{k+1} - x^* &= x_k - x^* - \frac{g(x_k)}{g'(\xi_2)}
\end{aligned}$$

Now use the Taylor expansion of $g(x_k)$.

$$\begin{aligned}
x_{k+1} - x^* &= x_k - x^* - \frac{g(x^*) + (x_k - x^*)g'(x^*) + \frac{1}{2}(x_k - x^*)^2 g''(\xi_1)}{g'(\xi_2)} \\
x_{k+1} - x^* &= x_k - x^* - \frac{(x_k - x^*)g'(x^*) + \frac{1}{2}(x_k - x^*)^2 g''(\xi_1)}{g'(\xi_2)} \\
x_{k+1} - x^* &= x_k - x^* - \frac{(x_k - x^*)g'(x^*)}{g'(\xi_2)} - \frac{\frac{1}{2}(x_k - x^*)^2 g''(\xi_1)}{g'(\xi_2)} \\
x_{k+1} - x^* &= (x_k - x^*)\left(1 - \frac{g'(x^*)}{g'(\xi_2)}\right) - \frac{\frac{1}{2}(x_k - x^*)^2 g''(\xi_1)}{g'(\xi_2)} \\
\frac{x_{k+1} - x^*}{(x_k - x^*)^2} &= \frac{1}{x_k - x^*}\left(1 - \frac{g'(x^*)}{g'(\xi_2)}\right) - \frac{g''(\xi_1)}{2g'(\xi_2)}
\end{aligned}$$

Note that as $k \to \infty$, $g(x_k) \to 0$ since this method is convergent. This also implies that $\xi_2 \to x^*$ and $\xi_1 \to x^*$, therefore

$$\lim_{k\to\infty}\left(\frac{|x_{k+1} - x^*|}{|x_k - x^*|^2}\right) = \lim_{k\to\infty}\left(\left|\frac{1}{x_k - x^*}\left(1 - \frac{g'(x_k)}{g'(\xi_2)}\right) - \frac{g''(\xi_1)}{2g'(\xi_2)}\right|\right)$$

Note that since this iteration scheme converges $\frac{g'(x^*)}{g'(\xi_2)} \to 1$ faster than $x_k - x^* \to 0$, so

$$\begin{aligned}
\lim_{k\to\infty}\left(\frac{|x_{k+1} - x^*|}{|x_k - x^*|^2}\right) &= \lim_{k\to\infty}\left(\frac{g''(\xi_1)}{2g'(\xi_2)}\right) \\
\lim_{k\to\infty}\left(\frac{|x_{k+1} - x^*|}{|x_k - x^*|^2}\right) &= \frac{g''(x^*)}{2g'(x^*)}
\end{aligned}$$

Thus this iteration scheme is Q-quadractically convergent. □

7. Implement the method from problem 6 in Matlab (or Python if you prefer). Use it to solve problems 8 and 9.

The following function implements the method from problem 6.

```
import numpy as np
def problem7(g, x0, TOL, MaxIter):
    x = np.zeros(MaxIter+2)
    x[0] = x0
    nIter = 0
    mstop = 1
    gx = g(x[nIter])
    delta = -np.square(gx)/(g(x[nIter] + gx) + gx)

    while nIter <= MaxIter and mstop:
        nIter+=1
        x[nIter] = x[nIter - 1] + delta
        gx = g(x[nIter])
        if abs(gx) < TOL and abs(delta) < TOL:
            mstop = 0
        else:
            delta = -np.square(gx)/(g(x[nIter] + gx) + gx)
    x = x[:nIter+1]
    return x
```

8. The van der Waal equation

$$\left(P + \frac{a}{V^2}\right)(V - b) = nRT$$

generalizes the ideal gas law $PV = nRT$. In each equation, P represents the pressure (atm), V represents the volume (liters), n is the number of moles of gas, and T represents the temperature (K). R is the universal gas constant and has the value

$$R = 0.08205\frac{\text{liters} \cdot \text{atm}}{\text{mole} \cdot \text{K}}$$

Determine the volume of 1 mole of isobutane at a temperature of $T = 313K$ and a pressure of $P = 2$ atm, given that, for isobutane, $a = 12.87\text{atm} \cdot \text{liters}^2$ and $b = 0.1142$ liters. Compare this to the value predicted from the ideal gas law. You may use any one of your methods (make clear in your writeup which one you are using, what initial guesses or intervals you are using, etc...).

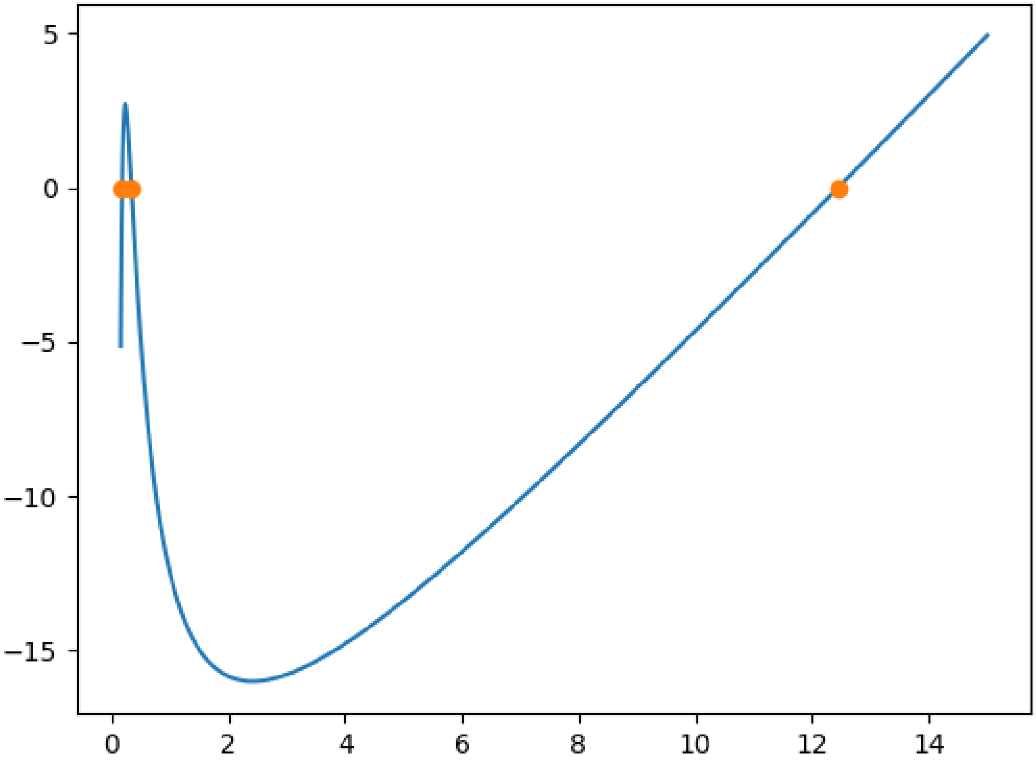
The following script uses the method implemented in problem 7 to find the volume that satisfies the van der Waal equation. The script uses three different initial guesses, 12, 0.175, and 0.3375.

```
import numpy as np
import matplotlib.pyplot as plt
execfile('01_7.py')

P = 2
a = 12.87
b = 0.1142
n = 1
R = 0.08205
T = 313
g = lambda V: (P + a/np.square(V))*(V - b) - n*R*T

TOL = 1e-10
MaxIter = 1000

V0 = 12
sol1 = problem7(g, V0, TOL, MaxIter)
V0 = .175
sol2 = problem7(g, V0, TOL, MaxIter)
V0 = .3375
sol3 = problem7(g, V0, TOL, MaxIter)
```

```
print 'The␣possible␣volumes␣of␣the␣isobutane␣are␣{0:.10f},␣{1:.10f},␣and
    ↪ ␣{2:.10f}'.format(sol1[-1], sol2[-1], sol3[-1])
x = np.linspace(.15,15,10000)
plt.plot(x, g(x), '-', [sol1[-1], sol2[-1], sol3[-1]], [g(sol1[-1]), g(
    ↪ sol2[-1]), g(sol3[-1])], 'o')
```

The output of this script is shown below.

```
The possible volumes of the isobutane are 12.4425967050, 0.1750704236, and 0.3373578713
liters
```



Note that the ideal gas law states that $V = \frac{nRT}{P} = \frac{0.08205 \times 313}{2} = 12.840825$ liters. This is very close to the first possible volume that satisfies the van der Waal equation, 12.442597 liters.

9. According to Archimedes' law, when a solid of density σ is placed in a liquid of density ρ, it will sink to a depth h that displaces an amount of liquid whose weight equals the weight of the solid. For a sphere of radius r, Archimedes law is

$$\frac{1}{3}\pi\Big(3rh^2 - h^3\Big)\rho = \frac{4}{3}\pi r^3 \sigma$$

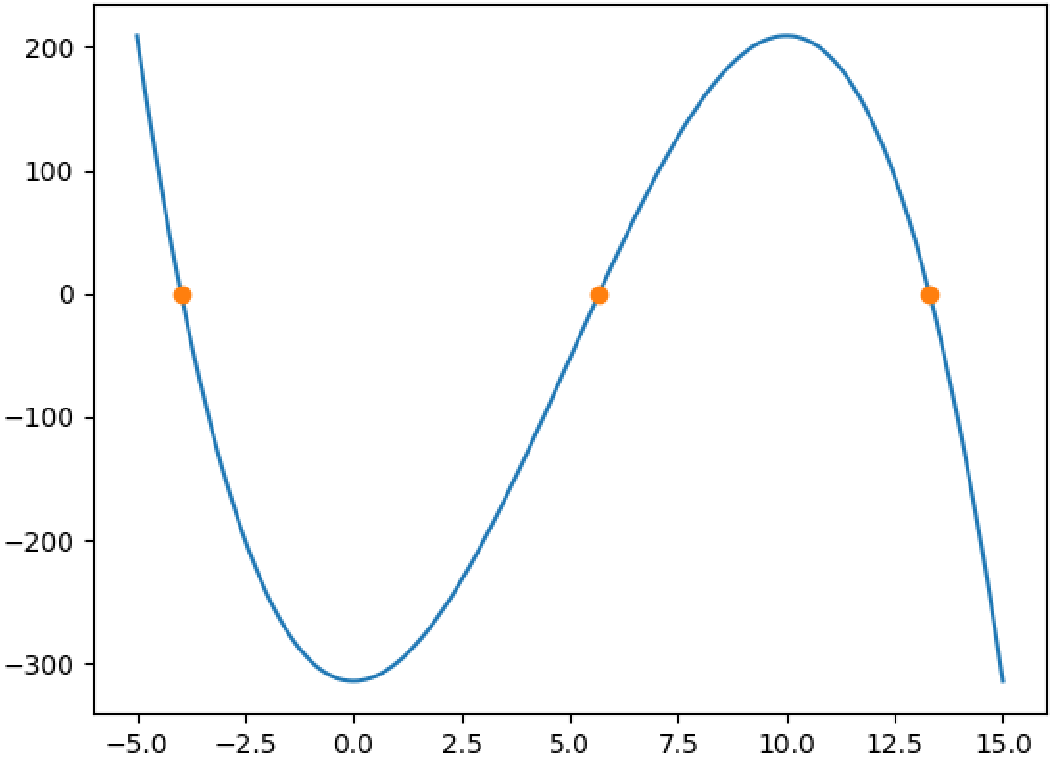
Given $r = 5$, $\rho = 1$, and $\sigma = 0.6$, determine h. You may use any one of your methods (make clear in your writeup which one you are using, what initial guesses or intervals you are using, etc...).

The following script uses the method in implemented in problem 7 to find the height of the sphere according to Archimedes' law. The script uses three different initial guesses and gives three different roots. The three initial guesses were -4, 13, and 5.67.

```
import numpy as np
import matplotlib.pyplot as plt
execfile('01_7.py')

r = 5.0
rho = 1.0
sigma = .6
g = lambda h: 1.0/3.0 * math.pi * (3.0*r*np.square(h) - np.power(h, 3.0)
    ↪ )*rho - 4.0/3.0*math.pi*np.power(r, 3.0)*sigma
h0 = 5.67

TOL = 1e-10
MaxIter = 1000

h0 = 5.67
sol1 = problem7(g, h0, TOL, MaxIter)
h0 = -4
sol2 = problem7(g, h0, TOL, MaxIter)
h0 = 13
sol3 = problem7(g, h0, TOL, MaxIter)

print 'The␣possible␣depths␣of␣the␣sphere␣are␣{0:.10f},␣{1:.10f},␣and␣
    ↪ {2:.10f}'.format(sol1[-1], sol2[-1], sol3[-1])

x = np.linspace(-5,15,10000)
plt.plot(x, g(x), '-', [sol1[-1], sol2[-1], sol3[-1]], [g(sol1[-1]), g(
    ↪ sol2[-1]), g(sol3[-1])], 'o')
```

The output of this script is shown below.

```
The possible depths of the sphere are 5.6706892285, -3.9760987462, and 13.3054095177
```


200
100
0
−100
−200
−300
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0