Single Variable Optimization

Video: 4A-4C

Reading:

Chong, An Introduction to Optimization, 2011, Chapter 7 – Sections 7.1 and 7.3

https://onlinelibrary-wiley-com.proxy1.library.jhu.edu/doi/book/10.1002/9781118033340

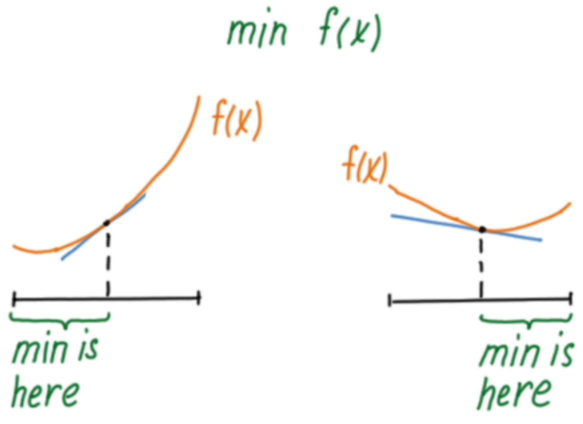
Rao, Engineering Optimization: Theory and Practice, 2009, Chapter 5 – Sections 5.6, 5.8 and 5.12.1

https://onlinelibrary-wiley-com.proxy1.library.jhu.edu/doi/book/10.1002/9780470549124

In this module we will assume that f(x) fits all the requirements of the corresponding methods, i.e., it is unimodal and differentiable or twice differentiable when needed to be.

Bisection method

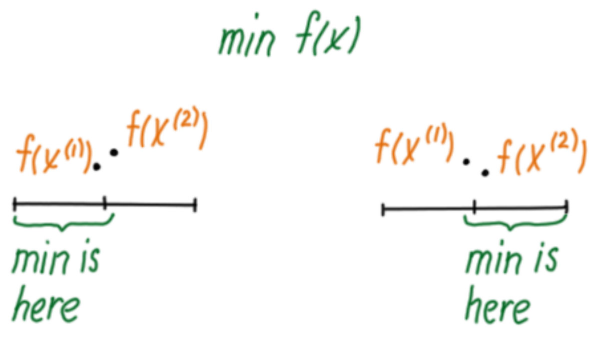
In the bisection method, we find the derivative of the objective function at a point inside the interval and this derivative indicates whether the minimum is located to the left or to the right from this point. In order to minimize the number of iterations in the worst-case scenario, the bisection point must be located half way between the ends of the interval.



Question 1: How many bisection iterations are needed to find a minimum of a differentiable unimodal function on the [0, 1] interval in the worst-case scenario if the tolerance is $\varepsilon = 10^{-5}$? For the purpose of this and the next four questions, let's define tolerance as the length of the interval containing the optimal point at which the algorithm stops.

Golden Section Search

If, instead of the derivative, we use the values of the objective function, we need two points instead of one, as shown in the picture.

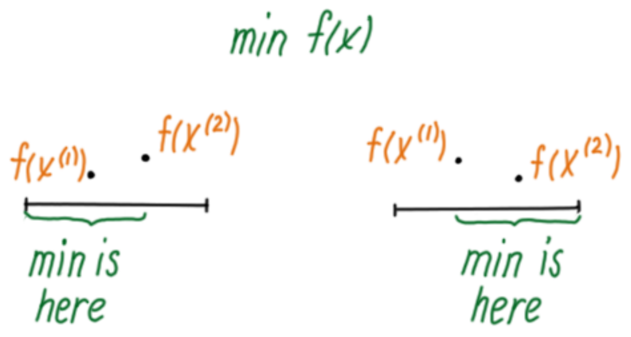


If these two points are in the middle of the interval and are also located very close to each other (the distance between them approaches zero) then we’re, essentially, computing the derivative, so this is similar to the bisection method. In this case, if the length of the search interval is 1 and the tolerance is ε, then, in the worst-case scenario, the value of the objective function has to be evaluated ? times.

Question 2: Replace the question mark with the correct expression

- $\left\lceil \log_{\frac{1}{2}} \varepsilon \right\rceil$
- $\lceil \log_2 \varepsilon \rceil$
- $2\left\lceil \log_{\frac{1}{2}} \varepsilon \right\rceil$
- $2\lceil \log_2 \varepsilon \rceil$
- none of the above

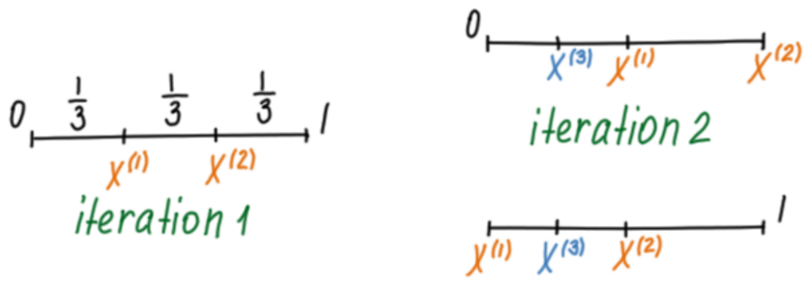
Can we improve the convergence rate by moving the two points away from the interval midpoint? Let’s place the test points at the 1/3 and 2/3 of the interval, as shown on the picture.



Question 3: With the above approach, how many times the objective function must be evaluated in the worst-case scenario if the length of the search interval is 1 and the tolerance is $\varepsilon = 10^{-5}$?

Moving the test points apart actually makes convergence slower, but what if we reuse a test point from the previous step? That is, on each iteration, one of the test points becomes the end point of the new search interval and the other test point falls inside of this interval. The value of the objective function at this point is already known and can be reused at the next iteration.

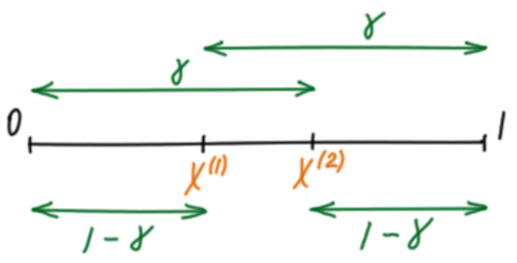
Consider a scenario when one point is reused and the new one is always placed in the middle of the biggest subinterval, as in the picture below:



Question 4: With the above approach, how many times the objective function must be evaluated in the worst-case scenario if the length of the search interval is 1 and the tolerance is $\varepsilon = 10^{-5}$? If the two subintervals are of equal length, the interval to be split is chosen arbitrary.

We can further improve the worst-case scenario if we place the test points in such a way that the relative contraction of the search interval is the same at each iteration, that is, if the test points always split the search interval in the same proportion. Let γ be the relative contraction of the search interval at each iteration. The length of the search region will become γ after the first iteration, γ^2 after the second iteration and so on.

The distance between 0 and the first split point is $1-\gamma$.



If $(x^{(2)}, 1]$ is rejected at the first iteration then $x^{(1)}$ is reused as the right split point at the second iteration. If the $x^{(1)}$ to $x^{(2)}$ region is rejected at the second iteration then the new search region is $[0, x^{(1)}]$ and its length is $1-\gamma$. However, we have established before that the length of the search region should be γ^2 after the second iteration, so we get

$$1-\gamma=\gamma^2$$

$$\gamma=\frac{-1+\sqrt{5}}{2}$$

This is the golden section search algorithm.

Question 5: With the golden section search algorithm, how many times the objective function must be evaluated in the worst-case scenario if the length of the search interval is 1 and the tolerance is ε ?

Newton's Method

Newton's method is based on the Taylor series expansion,

$$f(x)=f(x^{(0)})+f'(x^{(0)})(x-x^{(0)})+\frac{1}{2}f''(x^{(0)})(x-x^{(0)})^2+\ldots$$

Differentiating both sides, we get

$$f'(x)=f'(x^{(0)})+f''(x^{(0)})(x-x^{(0)})+\ldots$$

or, when $x-x^{(0)}$ is small,

$$f'(x)\approx f'(x^{(0)})+f''(x^{(0)})(x-x^{(0)})$$

When $f(x)$ is minimum/maximum and f(x) is differentiable, $f'(x)=0$, so we're looking for x such that

$$f'(x)=0$$

which gives

$$x\approx x^{(0)}-\frac{f'(x^{(0)})}{f''(x^{(0)})}$$

When the Taylor expansion doesn't have any terms beyond the $\frac{1}{2}f''(x^{(0)})(x-x^{(0)})^2$, i.e., when the third and any further derivatives are zero, the approximate equality becomes

$$x = x^{(0)} - \frac{f'(x^{(0)})}{f''(x^{(0)})}$$

and this equation gives the point x where $f'(x) = 0$. If the third or higher order derivatives are not zero, applying the above equation gets us closer to such a point (with some exceptions), but more than one iteration is needed.

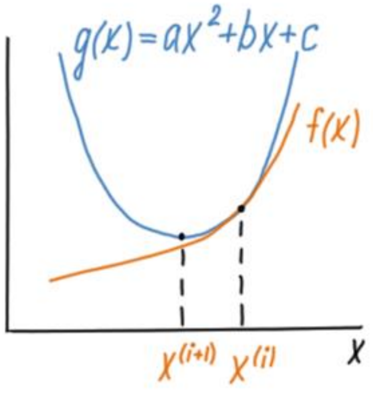
Question 6: Perform three iterations of Newton's method to solve max f(x) = x+ln(2x)+sin(3x)+cos(4x) on [4, 5]. The starting point is x_0=4.5. After the first iteration, x_1=

Question 7: (not for forum discussion, continued from the previous question) After the second iteration, x_2=

Question 8: (not for forum discussion, continued from the previous question) After the third iteration, x_3=

Question 9: (not for forum discussion, continued from the previous question) Use MATLAB's fminunc() to maximize f(x). What is the absolute difference between the value of x on the third iteration of Newton's method (x_3) and the precise solution to max f(x) = x+ln(2x)+sin(3x)+cos(4x)?

Viewing Newton's method from another perspective, at iteration i+1, the objective function in the neighborhood of point $x^{(i)}$ is approximated by a quadratic function and the minimum/maximum of this quadratic function is the $x^{(i+1)}$.



Question 10: (not for forum discussion, continued from the previous question) The second iteration involved approximating f(x) with a quadratic function g(x) = ax^2+bx+c where a =

Question 11: (not for forum discussion, continued from the previous question) b =

Question 12: (not for forum discussion, continued from the previous question) c =

Question 13: (not for forum discussion, continued from the previous question) g(x) = ax^2+bx+c is maximum on [4,5] at x =

Question 14: (can be discussed, but without giving out the answers to the previous four questions) Plot f(x) and g(x) for on the same plot in MATLAB. Please attach a screenshot of your plot here.

Multivariable Optimization

Gradient and Hessian

Video: 4D

Readings: Rao, Engineering Optimization: Theory and Practice, 2009, Chapter 6 – Section 6.8

https://onlinelibrary-wiley-com.proxy1.library.jhu.edu/doi/book/10.1002/9780470549124

The gradient $\nabla f(x)$ and the Hessian $\nabla^2 f(x)$ are multivariable generalizations of the first and the second derivatives,

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

they are vector and matrix valued functions, correspondingly.

Note that x is no longer a scalar variable, it is a column vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}$$

and $f(x)$ is a multivariate function.

Derivatives of multivariate functions are analogous to derivatives of single variable functions and the equations look only slightly different. For example, let's find the gradient of the quadratic multivariate function,

$$f(x) = x^T Q x + b^T x$$

Written in terms of $x_1, x_2, ..., x_n$,

$$f(x) = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \dots & \dots & \dots & \dots \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

The term of the above sum that includes x_i is $q_{ii}x_i^2 + \sum_{j=1, j\neq i}^{n} \left(q_{ij} + q_{ji}\right) x_i x_j + b_i x_i$, so

$\dfrac{\partial f}{\partial x_i} = 2q_{ii}x_i + \sum_{j=1, j\neq i}^{n} \left(q_{ij} + q_{ji}\right) x_j + b_i$ and

$$\nabla f(x) = \begin{bmatrix} 2q_{11}x_1 + \sum_{j=1, j\neq 1}^{n} \left(q_{1j} + q_{j1}\right) x_j + b_1 \\ 2q_{22}x_2 + \sum_{j=1, j\neq 2}^{n} \left(q_{2j} + q_{j2}\right) x_j + b_2 \\ \dots \\ 2q_{nn}x_n + \sum_{j=1, j\neq n}^{n} \left(q_{nj} + q_{jn}\right) x_j + b_n \end{bmatrix} = \begin{bmatrix} q_{11}x_1 + \sum_{j=1, j\neq 1}^{n} q_{1j}x_j \\ q_{22}x_2 + \sum_{j=1, j\neq 2}^{n} q_{2j}x_j \\ \dots \\ q_{nn}x_n + \sum_{j=1, j\neq n}^{n} q_{nj}x_j \end{bmatrix} + \begin{bmatrix} q_{11}x_1 + \sum_{j=1, j\neq 1}^{n} q_{j1}x_j \\ q_{22}x_2 + \sum_{j=1, j\neq 2}^{n} q_{j2}x_j \\ \dots \\ q_{nn}x_n + \sum_{j=1, j\neq n}^{n} q_{jn}x_j \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=1}^{n} q_{1j}x_j \\ \sum_{j=1}^{n} q_{2j}x_j \\ \dots \\ \sum_{j=1}^{n} q_{nj}x_j \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n} q_{j1}x_j \\ \sum_{j=1}^{n} q_{j2}x_j \\ \dots \\ \sum_{j=1}^{n} q_{jn}x_j \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \dots & \dots & \dots & \dots \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} + \begin{bmatrix} q_{11} & q_{21} & \dots & q_{n1} \\ q_{12} & q_{22} & \dots & q_{n2} \\ \dots & \dots & \dots & \dots \\ q_{1n} & q_{2n} & \dots & q_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

$$= Qx + Q^T x + b = \left(Q + Q^T\right) x + b$$

We have derived the gradient of the quadratic function,

$\nabla f(x) = \left(Q + Q^T\right) x + b$ or, if Q is symmetric,

$\nabla f(x) = 2Qx + b$

Question 15: Use the above logic to derive the Hessian H of the quadratic function. H =

- Q
- Qx

- $x^T Q$
- $Q + Q^T$
- $\left(Q + Q^T\right)x$
- $x^T\left(Q + Q^T\right)$
- none of the above

Similar reasoning can be used to derive Taylor series expansion for a vector function,

$$f(x) = f(x^{(0)}) + f'(x^{(0)})(x - x^{(0)}) + \frac{1}{2} f''(x^{(0)})(x - x^{(0)})^2 + \ldots$$

$$f(x) = f(x^{(0)}) + \left(\nabla f(x^{(0)})\right)^T \left(x - x^0\right) + \frac{1}{2}\left(x - x^0\right)^T \nabla^2 f(x^{(0)})\left(x - x^0\right) + \ldots$$

and

$$x = x^{(0)} - \frac{f'(x^{(0)})}{f''(x^{(0)})}$$

becomes

$$x = x^{(0)} - \left(\nabla^2 f(x^{(0)})\right)^{-1} \nabla f(x^{(0)})$$

which hints us that Newton's method is applicable to the multivariate case as well.

Steepest Descent

Video: 4E – 4F

Reading: Chong, An Introduction to Optimization, 2011, Chapter 8

https://onlinelibrary-wiley-com.proxy1.library.jhu.edu/doi/book/10.1002/9781118033340

Rao, Engineering Optimization: Theory and Practice, 2009, Chapter 6 – Section 6.9

https://onlinelibrary-wiley-com.proxy1.library.jhu.edu/doi/book/10.1002/9780470549124

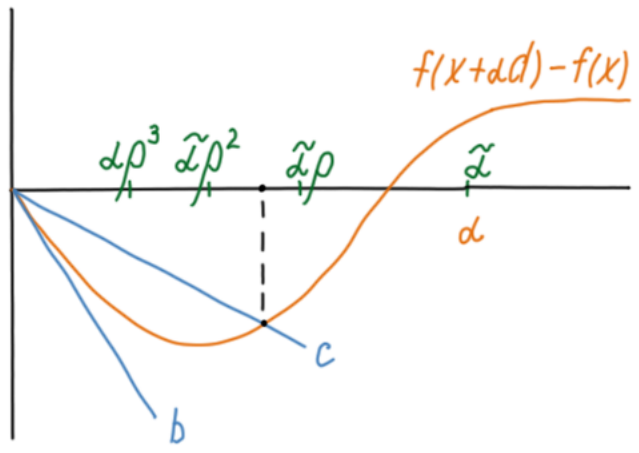
The most intuitive multivariate minimization method is the steepest descend. The direction $d^{(k)}$ that is opposite to the gradient of the objective function is the direction of the steepest descent, i.e., the direction in which the function decreases most rapidly. On each iteration, we find the direction of the steepest descent at the current point and make a step of length α in this direction.

Ideally, α should minimize the value of the objective function in the direction of the step,

$$\min_{\alpha} f(x^{(k)} + \alpha d^{(k)})$$
s.t. $\alpha \geq 0$

However, exact line search is slow, and using an approximate minimizer often makes the algorithm more efficient despite the increased number of iterations. One of the inexact line search approaches is the *Armijo rule*.

Let's plot $f(x^{(k)} + \alpha d^{(k)}) - f(x^{(k)})$ as a function of α.



Since $f(x^{(k)})$ is fixed, the exact line search can be written as

$$\min_{\alpha} f(x^{(k)} + \alpha d^{(k)}) - f(x^{(k)})$$
s.t. $\alpha \geq 0$

Armijo's line search looks for any α to the left from the intersection of $f(x^{(k)} + \alpha d^{(k)}) - f(x^{(k)})$ with line c, but as close to the intersection as possible.

Question 16: Line b has the slope (if applicable, c is a constant)
- $\nabla f(x^{(k)})^T$
- $c\nabla f(x^{(k)})^T$
- $\nabla f(x^{(k)})^T d^{(k)}$
- $c\nabla f(x^{(k)})^T d^{(k)}$
- none of the above

Question 17: Line marked by c has the slope (if applicable, c is a constant)
- $\nabla f(x^{(k)})^T$
- $c\nabla f(x^{(k)})^T$

- $\nabla f(x^{(k)})^T d^{(k)}$
- $c\nabla f(x^{(k)})^T d^{(k)}$
- none of the above

Therefore, the search starts with some initial value of α denoted by $\tilde{\alpha}$ and α is scaled by $\rho \in (0,1)$ at each step until $f(x^{(k)} + \alpha d^{(k)}) - f(x^{(k)})$ becomes smaller than $\alpha c \nabla f(x^{(k)})^T d^{(k)}$.

While simple and intuitive, the steepest descent method is slow because it zigzags. To illustrate, consider this optimization problem:

$$\min f(x,y) = \left(x^2 + y^2 - 9\right)^2 + \left(x + 3\right)^2$$

with $(x^{(0)}, y^{(0)}) = (0,3)$ as the starting point.

Using an exact line search, at the first iteration, $d^{(0)} = \begin{bmatrix} -1? \\ 2? \end{bmatrix}$ and $(x^{(1)}, y^{(1)}) = \begin{bmatrix} -3? \\ 4? \end{bmatrix}$

Question 18: Replace the 1?, 2?, 3? and 4? with the correct numbers.

At the second iteration, $d^{(1)} = \begin{bmatrix} 1? \\ -2? \end{bmatrix}$ and $(x^{(2)}, y^{(2)}) = \begin{bmatrix} 3? \\ 4? \end{bmatrix}$ or $(x^{(2)}, y^{(2)}) = \begin{bmatrix} 3? \\ -4? \end{bmatrix}$

Question 19: Replace the 1? and 2? with the correct numbers.

Question 20: Replace the 3? with the correct number.

Question 21: Replace the 4? with the correct number.

At the third iteration, $d^{(2)} = \begin{bmatrix} -1? \\ 2? \end{bmatrix}$, at the fourth iteration, $d^{(3)} = \begin{bmatrix} 3? \\ \dots \end{bmatrix}$ and at the fifth iteration, $d^{(4)} = \begin{bmatrix} \dots \\ 4? \end{bmatrix}$

Question 22: (not for forum discussion) Replace the 1?, 2?, 3? And 4? with the correct numbers.

Let's plot the first two iterations.

Question 23: Plot $(x^{(0)}, y^{(0)})$, $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$ and the objective function contours that pass through these points on the same plot in MATLAB. Please upload a screenshot of your plot here.

Question 24: Is $(x^{(2)}, y^{(2)})$ closer to the optimal solution than to the starting point $(x^{(0)}, y^{(0)})$?
Note: please use the $(x^{(2)}, y^{(2)})$ with the positive $y^{(2)}$ to answer this question.
- yes
- no

We will compare this search trajectory with one produced by Newton's method in the next section.

Newton's Method

Reading:
Chong, An Introduction to Optimization, 2011, Chapter 9 – Sections 9.1 and 9.2
https://onlinelibrary-wiley-com.proxy1.library.jhu.edu/doi/book/10.1002/9781118033340
Rao, Engineering Optimization: Theory and Practice, 2009, Chapter 6 – Section 6.11
https://onlinelibrary-wiley-com.proxy1.library.jhu.edu/doi/book/10.1002/9780470549124

Let's solve the same optimization problem using Newton's method. The search direction is now not $-\nabla f(x,y)$, but $-\left[\nabla^2 f(x,y)\right]^{-1}\nabla f(x,y)$. We will, again, use exact line search to provide for a fair comparison between the methods.

At the first iteration, $d^{(0)} = \begin{bmatrix} 1? \\ 2? \end{bmatrix}$ and $(x^{(1)}, y^{(1)}) = \begin{bmatrix} 3? \\ 4? \end{bmatrix}$

Question 25: Replace the 1?, 2?, 3? and 4? with the correct numbers.

A hint: MATLAB can be used to invert a matrix.

At the second iteration, $d^{(1)} = \begin{bmatrix} 1? \\ 2? \end{bmatrix}$ and $(x^{(2)}, y^{(2)}) = \begin{bmatrix} 3? \\ 4? \end{bmatrix}$

Question 26: (not for forum discussion) Replace the 1? with the correct number.

Question 27: (not for forum discussion) Replace the 2? with the correct number.

Question 28: (not for forum discussion) Replace the 3? with the correct number.

A hint: fminunc() can be used for exact line search.

Question 29: (not for forum discussion) Replace the 4? with the correct number.

Question 30: (can be discussed, but without giving out the answers to the previous four questions) Plot $(x^{(0)}, y^{(0)})$ $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$ and the objective function contours that pass through these points on the same plot in MATLAB. Please upload a screenshot of your plot here.

Question 31: (can be discussed, but without giving out $(x^{(2)}, y^{(2)})$) Is $(x^{(2)}, y^{(2)})$ closer to the optimal solution than to the starting point $(x^{(0)}, y^{(0)})$?

- yes
- no