Nonlinear optimization

CE 367R
Moving to fancier optimization

- Network optimization
- Linear programming

All of these problems had very convenient structures which led to efficient solution methods: a network structure or linearity.
Not all optimization problems are as convenient as these.

- What if the objective function is not linear?
- What if the constraints are not linear functions?
- What if a decision variable must be an integer?
- What if our problem has little mathematical structure in general (e.g., simulation-based optimization)

The rest of the semester is aimed at these more challenging problems. Part of the art in optimization is balancing the realism of the formulation with the resources available to collect data and solve the problem.
NONLINEAR OPTIMIZATION
The *general* nonlinear optimization problem (where the objective and constraints can be any functions whatever) is extremely difficult and probably impossible.

However, if the objective and constraints are “nice” functions, there are efficient algorithms for finding the global minimum.
As you learned in calculus, differentiable functions have stationary points where their derivative is equal to zero (gradient vector, for multivariable functions).

If there are no constraints, this method is still valid: enumerate the stationary points, find the one which minimizes or maximizes the objective (but be careful if the function is unbounded).

If there are constraints, we must account for the possibility that the minimum or maximum point lies along the boundary of the feasible region.
Convex sets

Nonlinear optimization is much easier if we can assume that the constraints and objective function are sufficiently “nice” (even if nonlinear).

In particular, we will assume that our feasible region satisfies the following properties:

Closed: If a sequence of solutions $x_i$ is feasible and $x_i \to x$, then $x$ is feasible as well.

Bounded: The constraints do not allow any decision variable to be arbitrarily large in magnitude (positive or negative).

Convex: A weighted average of any two feasible solutions is feasible. ($x$ feasible, $y$ feasible, and $\lambda \in [0, 1]$ imply that $\lambda x + (1 - \lambda)y$ is feasible)

For many real-world problems these are not restrictive.
All of the examples from earlier in the semester have closed, bounded, and convex feasible regions.

An optimization problem whose only constraints are $x > 0$ and $x < 5$ does not have a closed feasible region.

An optimization problem whose only constraint is $x \geq 0$ does not have a bounded feasible region.

An optimization whose only constraints are $x^2 \geq 1$ and $x^2 \leq 4$ does not have a convex feasible region.
Convex functions

Consider a function $f(x)$ of one variable, whose domain $X$ is a convex set. Geometrically, this function is convex if it “lies below its secants”.

Mathematically, $f$ is convex if, for every $x_1 \in X$ and $x_2 \in X$, and for every $\lambda \in [0, 1]$, we have

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

Furthermore, if $f$ is differentiable, a function is convex iff it “lies above its tangents”:

$$f(x_2) \geq f(x_1) + f'(x_1)(x_2 - x_1)$$

Furthermore, if $f$ is twice differentiable, a function is convex iff

$$f''(x) \geq 0$$

for all $x \in X$

A function is strictly convex if these inequalities can be made strict.
Examples

Which of the following functions are convex? Strictly convex? (Assume their domain is convex.)

1. $f(x) = x^2$
2. $f(x) = 3x$
3. $f(x) = \sin x$
4. $f(x) = |x|$
Nonlinear programs of the following form are extremely common:

- There are $n$ decision variables $x_1, \ldots, x_n$
- The objective is to minimize a function of the form $\sum_{i=1}^{n} f_i(x_i)$
- Each $f_i$ is a convex function.
- The only constraints are $x_1 + x_2 + \cdots + x_n = B$, and $x_1, \ldots, x_n \geq 0$.

Interpretation: You have a finite resource $B$ which you can divide among $n$ possible options. Each option gives you a different return depending on how much of the resource it gets, but returns are diminishing. How should you divide the resource to maximize the total benefit?
This is called a **resource allocation problem**. Examples:

- Assigning a fleet of buses to routes, to minimize total delay.
- Allocating maintenance resources to different regions or facility types.
- Allocating a budget to different departments in an organization.
- Managing an investment portfolio.
- How you manage the hours in your day.

If the functions $f_i(x_i)$ exhibit diminishing marginal returns (i.e., throwing more and more resources at an option helps less and less), and if all options are used in the optimal solution, the problem is easy to solve.
Under the conditions in the previous slide, at the optimal solution, the marginal return on all the alternatives is equal.

\[ \frac{df_i}{dx_i} = \lambda \]

for some \( \lambda \)

If this condition were not true, you could make a small adjustment to the allocation (from smaller to larger marginal return) and improve the objective.
Transit frequency-setting problem

\[
\min_n D(n) = \sum_{r \in R} \frac{d_r T_r}{2n_r}
\]

s.t.
\[
\sum_{r \in R} n_r = B \\
n_r \geq 0 \quad \forall r \in R
\]

Assume there are two routes; twelve buses \((B = 12)\), and the routes have the following characteristics:

<table>
<thead>
<tr>
<th>Route</th>
<th>Demand ((d_r))</th>
<th>Traversal time ((T_r))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>4000</td>
<td>10</td>
</tr>
</tbody>
</table>
The objective is to minimize \( \frac{5000}{n_1} + \frac{20000}{n_2} \) subject to \( n_1 + n_2 = 12 \) and nonnegativity.

At optimality, the derivatives of the reward functions \( \frac{5000}{n_1} \) and \( \frac{10000}{n_2} \) must be equal:

\[
\frac{5000}{n_1^2} = \frac{20000}{n_2^2}
\]

and \( n_1 + n_2 = 12 \).

Therefore \( n_1 = 4 \) and \( n_2 = 8 \) is the optimal allocation. Marginal costs are equal: assigning one more bus to each route would reduce delay by the same amount.
A systematic way to approach this type of problem is by a “line search” on the marginal return $\lambda$.

For any value of $\lambda$, we can solve for the allocations $x_i$ by solving $\frac{df_i}{dx_i} = \lambda$; call this $x_i(\lambda)$.

The total resource “imbalance” is $U(\lambda) = \sum_i x_i(\lambda) - B$; this should exactly equal zero at an optimal solution (this is the resource constraint).

If it does not, we need to adjust $\lambda$ higher or lower so that this is the case. Increasing $\lambda$
Bisection Algorithm

1. Choose \( \lambda_{lo}, \lambda_{hi} \) such that \( U(\lambda_{lo}) \) and \( U(\lambda_{hi}) \) have opposite signs.
2. Pick \( \lambda = (\lambda_{lo} + \lambda_{hi})/2 \) as the midpoint.
3. Evaluate each \( x_i(\lambda) \) and \( U(\lambda) \).
4. If the signs of \( U(\lambda) \) and \( U(\lambda_{lo}) \) match, set \( \lambda_{lo} = \lambda \).
5. If the signs of \( U(\lambda) \) and \( U(\lambda_{hi}) \) match, set \( \lambda_{hi} = \lambda \).
6. If \( \lambda_{hi} \) and \( \lambda_{lo} \) are sufficiently close (or if \( U(\lambda) \) is sufficiently close to zero), terminate. Otherwise return to step 2.
What if there is a lower bound on the allowable value of each allocation (so $x_i \geq \ell_i$)?

We can replace the formula for $x_i(\lambda)$ with this procedure:

After solving $\frac{df_i}{dx_i} = \lambda$, use the resulting value if it is more than $\ell_i$. Otherwise $x_i(\lambda) = \ell_i$. 
Transit frequency-setting problem (expanded)

\[
\min_{\mathbf{n}} \quad D(\mathbf{n}) = \sum_{i \in R} \frac{d_i T_i}{2x_i} \\
\text{s.t.} \quad \sum_{i \in R} x_i = B \\
\quad x_i \geq 0 \quad \forall i \in R
\]

Assume there are five routes; thirty buses \((B = 30)\), and the routes have the following characteristics:

<table>
<thead>
<tr>
<th>Route</th>
<th>Demand (d_r)</th>
<th>Traversal time (T_r)</th>
<th>Minimum allocation (\ell_r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4000</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>600</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2500</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>40</td>
<td>2</td>
</tr>
</tbody>
</table>

Nonlinear optimization | Resource allocation problem
For each route $i$, we have $f_i(x_i) = \frac{d_i T_i}{2x_i}$ and $\frac{df_i}{dx_i} = -\frac{d_i T_i}{2x_i^2}$.

Therefore $x_i(\lambda) = \max\{\ell_i, \sqrt{-\frac{d_i T_i}{2\lambda}}\}$.

If $\lambda = -500$, $x(\lambda) = [3.16, 6.32, 3, 8.66, 2]$ and 23.1 buses are used, so $U(\lambda) = -6.85$.

If $\lambda = -100$, $x(\lambda) = [7.07, 14.1, 3.87, 19.36, 4.47]$ and 48.9 buses are used, so $U(\lambda) = 18.92$.

These have opposite signs, so we can use them as $\lambda_{lo}$ and $\lambda_{hi}$ in the bisection algorithm.
<table>
<thead>
<tr>
<th>$\lambda_{lo}$</th>
<th>$\lambda_{hi}$</th>
<th>$\lambda$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>Total</th>
<th>$U$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-500$</td>
<td>$-100$</td>
<td>$-300$</td>
<td>4.08</td>
<td>8.16</td>
<td>3</td>
<td>11.18</td>
<td>2.58</td>
<td>29.01</td>
<td>$-0.99$</td>
</tr>
<tr>
<td>$-300$</td>
<td>$-100$</td>
<td>$-200$</td>
<td>5</td>
<td>10</td>
<td>3</td>
<td>13.7</td>
<td>3.16</td>
<td>34.86</td>
<td>4.86</td>
</tr>
<tr>
<td>$-300$</td>
<td>$-200$</td>
<td>$-250$</td>
<td>4.47</td>
<td>8.94</td>
<td>3</td>
<td>12.25</td>
<td>2.83</td>
<td>31.49</td>
<td>1.49</td>
</tr>
<tr>
<td>$-300$</td>
<td>$-250$</td>
<td>$-275$</td>
<td>4.26</td>
<td>8.53</td>
<td>3</td>
<td>11.68</td>
<td>2.70</td>
<td>30.17</td>
<td>0.17</td>
</tr>
</tbody>
</table>
OTHER NONLINEAR OPTIMIZATION PROBLEMS
Not all nonlinear optimization problems are resource allocation problems. There is still hope for solving them if the objective is a convex function, and the constraints form a convex set.

We’ll tackle some techniques, in increasing order of difficulty.
If there are no constraints and a single variable, \( \min f(x) \) is easy to solve if \( f \) is differentiable.

Set \( f'(x) = 0 \) and solve for \( x \).

Since \( f \) is convex we don’t have to check whether this point is a minimum or a maximum, local or global, etc.

If there are multiple decision variables (but no constraints), set \( \nabla f(x) = 0 \) and solve for \( x \).
What happens if there is one variable, but a nonnegativity constraint: 
\[ \min f(x) \text{ such that } x \geq 0 \]

There are two possibilities:
- \( x \geq 0 \) and \( f'(x) = 0 \) (same as before, just checking that \( x \) is feasible)
- \( x = 0 \) and \( f'(x) \geq 0 \) (the optimal point is at \( x = 0 \))

We want to summarize these conditions with equations. We need both \( x \) and \( f'(x) \) to be nonnegative, but \textit{at least one of them must be zero}. 
The following equations do this:

\[
\begin{align*}
    x & \geq 0 \\
    f'(x) & \geq 0 \\
    xf'(x) & = 0
\end{align*}
\]

These are called the \textit{optimality conditions} for the problem. If we can find a value of \( x \) that satisfies all three conditions, it is the optimal solution.
Example

Minimize $f(x) = x^2 + 3x + 5$ such that $x \geq 0$
What if there is more than one decision variable, all of which must be nonnegative?

\[
\begin{align*}
\min_x & \quad f(x) \\ 
\text{s.t.} & \quad x \geq 0
\end{align*}
\]
Using the same logic as before, we can derive the following optimality conditions which must hold for every decision variable $x_i$:

\[
\begin{align*}
    x_i & \geq 0 \\
    \frac{\partial f(x)}{\partial x_i} & \geq 0 \\
    x_i \frac{\partial f(x)}{\partial x_i} & = 0 
\end{align*}
\]

This can be compactly written as

\[
0 \leq x \perp \nabla f(x) \geq 0
\]
Example

Minimize \( f(x_1, x_2) = x_1^2 + x_2^2 + 3x_1 - 3x_2 + 5 \) such that \( x_1, x_2 \geq 0 \)
LINEAR EQUALITY CONSTRAINTS
Now, assume we have an optimization problem with multiple variables, a linear equality constraint, but no nonnegativity constraint.

An example is

$$\min_{x_1, x_2} \quad x_1^2 + x_2^2$$

s.t. \quad x_1 + x_2 = 5

The technique of Lagrange multipliers can solve this problem.
The Lagrangian function $\mathcal{L}$ includes the original objective function, and a term for each constraint.

$$\mathcal{L}(x_1, x_2, \kappa) = x_1^2 + x_2^2 + \kappa(5 - x_1 - x_2)$$

where the new variable $\kappa$ is a Lagrange multiplier.

You can think of the new term in the Lagrangian function as a “penalty” for violating the constraint.

The optimal solution of the original problem is a stationary point of the Lagrangian, that is, a place where $\nabla \mathcal{L}$ is zero.
Example

\[
\begin{align*}
\min_{x_1, x_2} & \quad x_1^2 + x_2^2 \\
\text{s.t.} & \quad x_1 + x_2 = 5
\end{align*}
\]
What if we have both linear equality constraints and nonnegativity constraints on each variable?

\[
\min_{x_1, \ldots, x_n} \quad f(x_1, \ldots, x_n)
\]
\[
\text{s.t.} \quad \sum_{i=1}^{n} a_{1i} x_i = b_1 \\
\sum_{i=1}^{n} a_{2i} x_i = b_2 \\
\vdots \\
\sum_{i=1}^{n} a_{mi} x_i = b_m \\
x_1, \ldots, x_n \geq 0
\]

The Lagrangian function contains a multiplier and term for each constraint

\[
\mathcal{L}(x_1, \ldots, x_n, \kappa_1, \ldots, \kappa_m) = f(x_1, \ldots, x_n) + \kappa_1 \left( b_1 - \sum_{j=1}^{n} a_{1j} x_j \right) + \\
\kappa_2 \left( b_2 - \sum_{j=1}^{n} a_{2j} x_j \right) + \cdots + \kappa_m \left( b_m - \sum_{j=1}^{n} a_{mj} x_j \right)
\]
We combine the Lagrangian approach with the optimality conditions from before, to obtain

\[
\frac{\partial L}{\partial x_i} \geq 0 \quad \forall i \in \{1, \ldots, n\}
\]

\[
\frac{\partial L}{\partial \kappa_j} = 0 \quad \forall j \in \{1, \ldots, m\}
\]

\[
x_i \geq 0 \quad \forall i \in \{1, \ldots, n\}
\]

\[
x_i \frac{\partial L}{\partial x_i} = 0 \quad \forall i \in \{1, \ldots, n\}
\]

as the optimality conditions.
What do these optimality conditions look like for the resource allocation problem?
DESCENT METHODS
Assume that we are given a nonlinear minimization problem.

A descent method is an iterative algorithm consisting of the following steps:

1. Choose an initial feasible solution $\mathbf{x} \leftarrow \mathbf{x}_0$.
2. Identify a feasible “target” solution $\mathbf{x}^*$ in a “downhill direction.”
3. Choose a step size $\lambda \in [0, 1]$ and set $\mathbf{x} \leftarrow \lambda \mathbf{x}^* + (1 - \lambda)\mathbf{x}$
4. Test for termination, and return to step 2 if we need to improve further.
Hiking analogy
The two key questions are:

1. How do I choose the target solution?
2. How do I choose \( \lambda \)

You might also ask how to choose the initial solution \( \mathbf{x}^0 \). For a convex optimization problem, it won’t matter too much, and if we answer the previous two questions correctly we will converge to the global optimum from any starting point.
A few different ways these steps can be performed:

For choosing the target $x^*$, I will show you the **conditional gradient** and **gradient projection** methods.

For choosing the step size $\lambda$, I will show you the **method of successive averages**, **limited minimization rule**, and **Armijo rule**.

For a sufficiently nice (differentiable, convex, etc.) optimization problem, any combination of these will converge to the global optimum from any starting point.
As an example, minimize \((x_1 - 1)^2 + (x_2 - 2)^4\) over the set \(0 \leq x_1, x_2 \leq 2\).
TARGET SELECTION
Conditional gradient

In the conditional gradient rule, we assume that the slope of the objective is the *same* throughout the feasible region. (This is not true, but gives us a direction to move in.)

This is equivalent to replacing the objective function with its linear approximation at the current point $x$.

We then solve the optimization problem for the approximate objective function $f(x^*) = f(x) + \nabla f(x)(x^* - x)$ with the same constraints, and use the optimal $x^*$ value for the target.

If all of the constraints are linear, the conditional gradient method is nothing more than solving a linear program.
Example

Starting with the initial solution $\mathbf{x} = [0 \ 0]$, the gradient at this point is $[-2 \ -32]$.

The linearized objective $f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot (\mathbf{x}^\ast - \mathbf{x})$ is

$$17 + [-2 \ -32] \cdot \begin{bmatrix} x_1^\ast \\ x_2^\ast \end{bmatrix}$$

The solution to $\min -2x_1 - 32x_2$ subject to $0 \leq x_1, x_2 \leq 2$ is $(2, 2)$. This is the target $\mathbf{x}^\ast = [2 \ 2]$

Notice that we can always simplify $f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot (\mathbf{x}^\ast - \mathbf{x})$ by removing constants. We can just as well minimize $\nabla f(\mathbf{x}) \cdot \mathbf{x}^\ast$
If we had chosen $\mathbf{x} = [2 \ 0]$ as the initial solution, the gradient is $[2 \ -32]$.

The linearized objective is then $\min 2x_1 - 32x_2$, and the target is $\mathbf{x}^* = [0 \ 2]$.

If we had chosen $\mathbf{x} = [1 \ 1]$ as the initial solution, the gradient is $[0 \ -4]$.

The linearized objective is then $\min -4x_2$, and any vector with $x_2 = 0$ can be chosen as the target.
Gradient projection

In the gradient projection method, the target is found by calculating the point \( x - s \nabla f(x) \) (where \( s \) is another step size), and then calculating the projection of that point onto the feasible region.

The projection of a point \( x \) onto a set \( X \) is the point in \( X \) closest to \( x \), and is denoted by \( \Pi_X(x) \).
Examples of projection

Project the point (5, 10) onto the set defined by $0 \leq x_1 \leq 7$, $0 \leq x_2 \leq 7$.

Project the point (4, 2) onto the set defined by $x_1 + x_2 = 5$. 
Assume that $s = 1/2$ (this is another parameter that can be tuned)

At the point $x = [0 \ 0]$, the gradient is $[-2 \ -32]$.

The target is selected to be the projection of $[0 \ 0] - \frac{1}{2} [-2 \ -32]$ onto the feasible set.

The projection of $[1 \ 16]$ onto the feasible set is $[1 \ 2]$, so this is the target.
As a rule of thumb, the gradient projection target tends to be “better”, but calculating the projection of a point onto complicated feasible regions is not easy.
STEP SIZE SELECTION
There are a number of ways to choose the step size $\lambda$ after the target has been chosen. We’ll go through three:

- The **method of successive averages** is simplest and fastest, but not very intelligent.
- The **line minimization rule** tends to work well in practice, but can be slower.
- The **Armijo rule** uses trial and error to quickly find a “reasonably good” $\lambda$ value.
Method of successive averages

The method of successive averages uses a fixed sequence of $\lambda$ values, rather than trying to customize $\lambda$ at each step of the algorithm.

There are two risks with using fixed values of $\lambda$: if $\lambda$ is too small, convergence will be very slow. If $\lambda$ is too large, the algorithm may not converge at all.

The method of successive averages tries to avoid both of these difficulties by starting with larger values of $\lambda$ and moving to smaller ones.

A typical sequence of $\lambda$ values is $1/2$, $1/3$, $1/4$, etc.
Example

(Conditional gradient method plus MSA).

When we started the conditional gradient method with \([0 \ 0]\) the target point was \([2 \ 2]\).

Using the method of successive averages the new point is

\[
\frac{1}{2} [2 \ 2] + (1 - \frac{1}{2}) [0 \ 0] = [1 \ 1]
\]

(objective reduced from 17 to 1.16).
Nonlinear optimization
Step size selection
Example

(Conditional gradient method plus MSA).

When we started the conditional gradient method with \([0 \ 0]\) the target point was \([2 \ 2]\).

Using the method of successive averages the new point is 
\[
\frac{1}{2} [2 \ 2] + (1 - \frac{1}{2}) [0 \ 0] = [1 \ 1]
\]
(objective reduced from 17 to 1.16).

At this point the gradient is \([0 \ -4]\); the new target is any point where 
\(x_2 = 2\). If \((1, 2)\) is the new target, then \(x\) is updated to 
\[
\frac{1}{3} [1 \ 2] + \frac{2}{3} [1 \ 1] = [1 \ 4/3].
\]
(Objective reduced from 1.16 to 0.358)

Here you see one downside of MSA — the global optimum would have been reached if we had chosen \(\lambda = 1\). MSA does not have the ability to detect such cases, it always follows the pre-set sequence of step sizes.

Nonlinear optimization

Step size selection
Nonlinear optimization

Step size selection
The line minimization rule chooses the value of $\lambda \in [0, 1]$ which minimizes the objective function along the line connecting $x$ to $x^*$. This value can be found using the bisection method or Newton’s method.

Specifically, we want to choose $\lambda$ to minimize $f(\lambda x^* + (1 - \lambda)x)$ subject to $0 \leq \lambda \leq 1$. 
Example

(Conditional gradient plus line minimization). When we started the conditional gradient method with $[0 \ 0]$ the target point was $[2 \ 2]$. Using the line minimization rule, $\lambda = 0.705$ and the new point is $[1.41 \ 1.41]$. (Objective reduced from 17 to 0.289)
Example

(Conditional gradient plus line minimization). When we started the conditional gradient method with \([0 \ 0]\) the target point was \([2 \ 2]\).

Using the line minimization rule, \(\lambda = 0.705\) and the new point is \([1.41 \ 1.41]\). (Objective reduced from 17 to 0.289)

At this point the gradient is \([0.82 \ -2.36]\). The new target minimizes \(0.82x_1 - 2.36x_2\), so \(x^* = [0 \ 2]\).

Using the line minimization rule, \(\lambda = 0.328\) and the new point is \([0.948 \ 1.60]\). (Objective reduced to 0.027)
The Armijo rule tries to overcome disadvantages of both MSA (not very smart) and line minimization (it takes too long).

The Armijo rule does not try to find the value of $\lambda$ which minimizes $f$, but is content to find a value of $\lambda$ which reduces $f$ “enough.”

This rule is a “trial-and-error” technique, where we try different values of $\lambda$ until we find an acceptable one.
An “acceptable” $\lambda$ is defined as one for which
\[
\frac{f(x) - f(x(\lambda))}{\lambda} \geq \alpha |f'(x(0))|
\]
where $x(\lambda)$ is the new point as a function of $\lambda$, and $f'$ is the derivative of $f$ at $x$, in the direction of $x^*$. 
A good rule of thumb is to set $\alpha = 0.1$, and to try the sequence \(\{1, 1/2, 1/4, 1/8, \ldots\}\) of $\lambda$ values until one of them is acceptable.
Example

(Conditional gradient plus Armijo rule with $\alpha = 0.1$). At the initial point $[0 \ 0]$, the target point was $[2 \ 2]$, so

$$x(\lambda) = \lambda [2 \ 2] + (1 - \lambda) [0 \ 0] = [2\lambda \ 2\lambda]$$

So, $f(x(\lambda)) = (2\lambda - 1)^2 + (2\lambda - 2)^4$ and $f'(x(0)) = 4(2(0) - 1) + 8(2(0) - 2) = -20$

For a point to be acceptable in the Armijo rule, the decrease in the objective function (from 17) divided by $\lambda$ must be at least $2 = 0.1 \times 20$.

Start by trying $\lambda = 1$. In this case, the objective function decreases to 1; $16/1 > 2$ so this point is acceptable and we move to $[2 \ 2]$. 

Nonlinear optimization  Step size selection
Nonlinear optimization

Step size selection
The new point is \([2 \ 2]\), the gradient is \([2 \ 0]\), so the new target point minimizes \(2x_1\); say \(x^* = [0 \ 2]\).

So, \(x(\lambda) = [2 - 2\lambda \ 2]\), \(f(x(\lambda)) = (1 - 2\lambda)^2\), and \(f'(x(0)) = -4(1 - 2(0)) = -4\).

For a point to be acceptable in the Armijo rule, the decrease in the objective function (from 1) divided by \(\lambda\) must be at least 0.4.

If \(\lambda = 1\), the new point is \([0 \ 2]\) and the objective function is 1, so there is no decrease... unacceptable.

If \(\lambda = 1/2\), the new point is \([1 \ 2]\) and the objective function is 0; since \((1 - 0)/2 > 0.4\) this point is acceptable.
Nonlinear optimization
Step size selection