10.3 Steepest Descent Techniques

- The advantage of the Newton and quasi-Newton methods for solving systems of nonlinear equations is their speed of convergence once a sufficiently accurate approximation is known.

- A weakness of these methods is that an accurate initial approximation to the solution is needed to ensure convergence.

- The Steepest Descent method considered in this section converges only linearly to the solution, but it will usually converge even for poor initial approximations.

- As a consequence, this method is used to find sufficiently accurate starting approximations for the Newton-based techniques in the same way the Bisection method is used for a single equation.
Consider the system of nonlinear equations of the form

\[
\begin{align*}
  f_1(x_1, x_2, \cdots, x_n) &= 0, \\
  f_2(x_1, x_2, \cdots, x_n) &= 0, \\
  &\vdots \\
  f_n(x_1, x_2, \cdots, x_n) &= 0.
\end{align*}
\]

Define the function

\[
g(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} \left[ f_i(x_1, x_2, \cdots, x_n) \right]^2.
\]

The function \( g(x_1, x_2, \cdots, x_n) \) is nonnegative.

If \( x = [x_1, x_2, \cdots, x_n]^t \) is a solution to the system \( \mathbf{F}(x) = 0 \), then the function \( g \) reaches its local minimum, and the minimum value is 0.
The method of Steepest Descent for finding a local minimum for $g : \mathbb{R}^n \rightarrow \mathbb{R}$ is as follows:

1. Evaluate $g$ at an initial approximation $x^{(0)} = [x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}]^t$.

2. Determine a direction from $x^{(0)}$ that results in a decrease in the value of $g$.

3. Move an appropriate amount in this direction and call the new value $x^{(1)}$.

4. Repeat step 1 through step 3 with $x^{(0)}$ replaced by $x^{(1)}$. 
Illustration of the Method of Steepest Descent
Review of Gradient of a Function

For $g : \mathbb{R}^n \rightarrow \mathbb{R}$, the **gradient** of $g$ at $x = [x_1, x_2, \cdots, x_n]^t$ is denoted $\nabla g(x)$ and defined by

$$\nabla g(x) = \left[ \frac{\partial g}{\partial x_1}(x), \frac{\partial g}{\partial x_2}(x), \cdots, \frac{\partial g}{\partial x_n}(x) \right]^t.$$ 

For example, $g(x_1, x_2) = x_1^2 - 2x_1x_2 + \sin(2x_1 + x_2)$,

$$\nabla g(x) = [2x_1 - 2x_2 + 2 \cos(2x_1 + x_2), -2x_1 + \cos(2x_1 + x_2)]^t.$$
The **directional derivative** of $g$ at at $x = [x_1, x_2, \cdots, x_n]^t$ in the direction of a unit vector $v$ measures the rate of change of $g$ in the direction $v$. It is defined by

$$D_v g(x) = \lim_{h \to 0} \frac{g(x + hv) - g(x)}{h} = v \cdot \nabla g(x)$$

The direction that produces the **maximum value for the directional derivative** occurs when $v$ is chosen to be parallel to $\nabla g(x)$. This direction yields the **greatest increase** of the value of $g$.

The direction of **greatest decrease** in the value of $g$ at $x$ is the direction given by $-\nabla g(x)$. 
Illustration of greatest decrease direction for a function $z = g(x_1, x_2)$. 
The object is to reduce \( g(x) \) to its minimum value of zero.

We choose \( x^{(1)} \) by moving away from \( x^{(0)} \) in the direction \(-\nabla g(x^{(0)})\) that gives the greatest decrease at \( x^{(0)} \). That is

\[
x^{(1)} = x^{(0)} - \alpha \nabla g(x^{(0)}), \quad \text{for some constant } \alpha > 0.
\]

The remaining task is to choose an appropriate value of \( \alpha \) so that \( g(x^{(1)}) \) is significantly less than \( g(x^{(0)}) \).
To determine the value $\alpha$, we consider the single-variable function

$$h(\alpha) = g(x^{(0)} - \alpha \nabla g(x^{(0)})).$$

The value of $\alpha$ that minimizes the function $h$ is the value needed.

Taking derivative of $h(\alpha)$ is computationally too costly. Instead, we consider an approximation of $h(\alpha)$.

To do this, we choose three numbers $\alpha_1 < \alpha_2 < \alpha_3$ that, we hope, are close to where the minimum value of $h(\alpha)$ occurs.

We then construct a quadratic polynomial $P(x)$ that agrees with the value of $h$ at $\alpha_1$, $\alpha_2$, and $\alpha_3$.

Let $\alpha$ be the minimizer of $P(x)$ in $[\alpha_1, \alpha_3]$. Then, $\alpha$ is used for approximating the minimum value of $g$ is

$$x^{(1)} = x^{(0)} - \alpha \nabla g(x^{(0)}).$$
First, we choose $\alpha_1 = 0$, that is
\[
g_1 = g(x^{(0)} - \alpha_1 \nabla x^{(0)}) = g(x^{(0)})
\]

Next, we choose a number $\alpha_3$ with
\[
g_3 = g(x^{(0)} - \alpha_3 \nabla x^{(0)}) < g(x^{(0)}) = g_1.
\]

Since $\alpha_1$ does not minimize $h$, such a number $\alpha_3$ does exist.

1. First choose $\alpha_3 = 1$, and calculate $g_3 = g(x^{(0)} - \alpha_3 \nabla x^{(0)})$.
2. If $g_3 < g_1$, then this $\alpha_3$ works.
3. If $g_3 > g_1$, then let $\alpha_3 = \alpha_3/2$ and go back to Step 1.

Finally, we choose $\alpha_2 = \frac{\alpha_3}{2}$. Compute $g_2 = g(x^{(0)} - \alpha_2 \nabla x^{(0)})$
Now we obtain three data points \((\alpha_1, g_1), (\alpha_2, g_2), (\alpha_3, g_3)\). Use the quadratic interpolation (coefficients \(h_1\) and \(h_2\) to be determined)

\[
P(x) = g_1 + h_1(x - \alpha_1) + h_2(x - \alpha_1)(x - \alpha_2)
\]

\[
\begin{align*}
  h_1 &= \frac{g_2 - g_1}{\alpha_2 - \alpha_1}, \\
  h_2 &= \frac{g_3 - g_1 - h_1(\alpha_3 - \alpha_1)}{(\alpha_3 - \alpha_1)(\alpha_3 - \alpha_2)}
\end{align*}
\]

Note that \(\alpha_1 = 0\), and \(\alpha_2 = \alpha_3/2\), then

\[
P(x) = g_1 + h_1x + h_2x(x - \alpha_2)
\]

We can find that

\[
\begin{align*}
  h_1 &= \frac{g_2 - g_1}{\alpha_2} = \frac{2(g_2 - g_1)}{\alpha_3} \\
  h_2 &= \frac{g_3 - g_1 - h_1\alpha_3}{\alpha_3(\alpha_3 - \alpha_2)} = \frac{2(g_1 - 2g_2 + g_3)}{\alpha_3^2}
\end{align*}
\]
Note that

\[ P(x) = g_1 + h_1 x + h_2 x (x - \alpha_2) \]

The critical point \( \alpha_0 \) of \( P(x) \) can be obtained by

\[ P'(x) = h_1 + 2h_2 x - h_2 \alpha_2 = 0, \quad \Rightarrow \quad \alpha_0 = \frac{h_2 \alpha_2 - h_1}{2h_2} = \frac{1}{2}(\alpha_2 - \frac{h_1}{h_2}). \]

The minimum value of \( P \) on \([\alpha_1, \alpha_3]\) occurs either at the only critical point \( \alpha_0 \) of \( P \) or at the right endpoint \( \alpha_3 \).

Choose \( \alpha \) from \( \{\alpha_0, \alpha_3\} \) so that

\[ g(x - \alpha \nabla g(x^{(0)})) = \min\{g_0, g_3\}. \]

Update the solution by

\[ x^{(1)} = x^{(0)} - \alpha \nabla g(x^{(0)}) \]
Example 5.

Use the Steepest Descent method with $x^{(0)} = [0, 0, 0]^t$ to find a reasonable starting approximating to the solution of

\[
\begin{align*}
3x_1 - \cos(x_2x_3) - \frac{1}{2} &= 0, \\
x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 &= 0, \\
e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} &= 0.
\end{align*}
\]
Solution (1/5).

\[ g(x_1, x_2, x_3) = [f_1(x_1, x_2, x_3)]^2 + [f_2(x_1, x_2, x_3)]^2 + [f_3(x_1, x_2, x_3)]^2 \]

\[ \nabla g(x) = \begin{bmatrix}
2 f_1(x) \frac{\partial f_1}{\partial x_1}(x) + 2 f_2(x) \frac{\partial f_2}{\partial x_1}(x) + 2 f_3(x) \frac{\partial f_3}{\partial x_1}(x) \\
2 f_1(x) \frac{\partial f_1}{\partial x_2}(x) + 2 f_2(x) \frac{\partial f_2}{\partial x_2}(x) + 2 f_3(x) \frac{\partial f_3}{\partial x_2}(x) \\
2 f_1(x) \frac{\partial f_1}{\partial x_3}(x) + 2 f_2(x) \frac{\partial f_2}{\partial x_3}(x) + 2 f_3(x) \frac{\partial f_3}{\partial x_3}(x)
\end{bmatrix} = 2 \begin{bmatrix}
\frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_3}{\partial x_1}(x) \\
\frac{\partial f_1}{\partial x_2}(x) & \frac{\partial f_2}{\partial x_2}(x) & \frac{\partial f_3}{\partial x_2}(x) \\
\frac{\partial f_1}{\partial x_3}(x) & \frac{\partial f_2}{\partial x_3}(x) & \frac{\partial f_3}{\partial x_3}(x)
\end{bmatrix} \begin{bmatrix}
f_1(x) \\
f_2(x) \\
f_3(x)
\end{bmatrix} = 2 [J(x)]^t \mathbf{F}(x) \]
Solution (2/5).

- For $x^{(0)} = [0, 0, 0]^t$, we have

  $$g(x^{(0)}) = 111.975$$

  $$\nabla g(x^{(0)}) = 2 \left[ J(x^{(0)}) \right]^t F(x^{(0)}) = \begin{bmatrix}
  -9 \\
  -8.1 \\
  419.38
\end{bmatrix}.$$ 

- Normalize the gradient vector

  $$z = \frac{1}{\| \nabla g(x^{(0)}) \|_2} \nabla g(x^{(0)}) = \frac{1}{419.55} \nabla g(x^{(0)}) = \begin{bmatrix}
  -0.021451 \\
  -0.019306 \\
  0.99958
\end{bmatrix}.$$
Solution (3/5).

- Set $\alpha_1 = 0$, we have $g_1 = g(x^{(0)} - \alpha_1 z) = 111.975$.
- Set $\alpha_3 = 1$, we have $g_3 = g(x^{(0)} - \alpha_3 z) = 93.565$.
- Since $g_3 < g_1$, we let $\alpha_2 = \frac{\alpha_3}{2} = 0.5$, $g_2 = g(x^{(0)} - \alpha_2 z) = 2.5356$.
- The data points $(\alpha_1, g_1), (\alpha_2, g_2), (\alpha_3, g_3)$ are 
  
  $(0, 111.975), (0.5, 2.5356), (1, 93.565)$

- We construct $P(x) = g_1 + h_1(x - \alpha_1) + h_2(x - \alpha_1)(x - \alpha_2)$ that interpolates the data points.

  - $P(\alpha_1) = g_1$ is automatically satisfied.
  - $P(\alpha_2) = g_2$ yields $h_1 = \frac{2(g_2 - g_1)}{\alpha_3} = -218.878$
  - $P(\alpha_3) = g_3$ yields $h_2 = \frac{2(g_1 - 2g_2 + g_3)}{\alpha_3^2} = 400.94$
Solution (4/5).

- The interpolating polynomial is
  \[ P(x) = 111.975 - 218.878x + 400.94x(x - 0.5) \]

- The critical point is
  \[ \alpha_0 = \frac{1}{2} (\alpha_2 - \frac{h_1}{h_2}) = 0.5230 \]

- Evaluating \( g(x) \) at \( \alpha_0 \) yields \( g_0 = g(x^{(0)} - \hat{\alpha}z) = 2.3276 \).
- Since \( g_0 < g_3 \), we choose \( \alpha = \alpha_0 = 0.52296 \).

\[ x^{(1)} = x^{(0)} - \alpha z = \begin{bmatrix} 0.0112 \\ 0.0101 \\ -0.5227 \end{bmatrix} \]
Solution (5/5).

- We use Matlab programming to finish the rest of the computation.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$x_1^{(k)}$</th>
<th>$x_2^{(k)}$</th>
<th>$x_3^{(k)}$</th>
<th>$g(x_1^{(k)}, x_2^{(k)}, x_3^{(k)})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.137860</td>
<td>$-0.205453$</td>
<td>$-0.522059$</td>
<td>1.27406</td>
</tr>
<tr>
<td>3</td>
<td>0.266959</td>
<td>0.00551102</td>
<td>$-0.558494$</td>
<td>1.06813</td>
</tr>
<tr>
<td>4</td>
<td>0.272734</td>
<td>$-0.00811751$</td>
<td>$-0.522006$</td>
<td>0.468309</td>
</tr>
<tr>
<td>5</td>
<td>0.308689</td>
<td>$-0.0204026$</td>
<td>$-0.533112$</td>
<td>0.381087</td>
</tr>
<tr>
<td>6</td>
<td>0.314308</td>
<td>$-0.0147046$</td>
<td>$-0.520923$</td>
<td>0.318837</td>
</tr>
<tr>
<td>7</td>
<td>0.324267</td>
<td>$-0.00852549$</td>
<td>$-0.528431$</td>
<td>0.287024</td>
</tr>
</tbody>
</table>

- A true solution to the nonlinear system is $x = (0.5, 0, -0.5235988)^t$, so the second iterative solution $x^{(2)}$ would likely be adequate as an initial approximation for Newton's method or Broyden's method.

- One of these quicker converging techniques would be appropriate at this stage, since 70 iterations of the Steepest Descent method are required to find $\|x - x^{(k)}\|_{\infty} \leq 10^{-2}$.
% ex10_3_1: Steepest Descent Method
clc
clear
%% Initialization
MaxIt = 100; % max number of iterations
tol = 10^(-2); % tolerance
x0 = [0; 0; 0]; % initial approximation

%% Steepest Descent Method
for i = 1:MaxIt
    disp(['Iteration = ', int2str(i)]);
    F = Fun(x0);
    J = jacobian(x0);
    grad = 2*J'*F;
    z = (1/norm(grad))*grad; % normalize the vector
    a1 = 0; g1 = FunG(x0-a1*z);
    a3 = 1; g3 = FunG(x0-a3*z);
    while g3 > g1
        a3 = a3/2;
        g3 = FunG(x0-a3*z);
    end
    a2 = a3/2; g2 = FunG(x0-a2*z);
    h1 = 2*(g2-g1)/a3;
    h2 = 2*(g1-2*g2+g3)/(a3*a3);
    a0 = (1/2)*(a2-h1/h2); % a0 is the critical point
    g0 = FunG(x0-a0*z);
    if g0 < g3 % select the smaller one from g0 and g3
        a = a0;
    elseif g0 >= g3
        a = a3;
    end
    g = FunG(x0-a*z)
    x = x0 - a*z
    if abs(g) < tol % Stopping Criteria
        break;
    end
    x0 = x;
end

%% Function F
function F = Fun(x)
y1 = 3*x(1)-cos(x(2).*x(3))-1/2;
y2 = x(1).^2-81*(x(2)+0.1).^2+sin(x(3)) + 1.06;
y3 = exp(-x(1).*x(2))+20*x(3)+(10*pi-3)/3;
F = [y1;y2;y3];
end

%% Jacobian Function
function J = jacobian(x)
n = length(x);
J = zeros(n);
J(1,1) = 3;
J(1,2) = x(3).*sin(x(2).*x(3));
J(1,3) = x(2).*sin(x(2).*x(3));
J(2,1) = 2*x(1);
J(2,2) = -162*(x(2)+0.1);
J(2,3) = cos(x(3));
J(3,1) = -x(2).*exp(-x(1).*x(2));
J(3,2) = -x(1).*exp(-x(1).*x(2));
J(3,3) = 20;
end

%% Function g
function u = FunG(x)
y1 = 3*x(1)-cos(x(2).*x(3))-1/2;
y2 = x(1).^2-81*(x(2)+0.1).^2+sin(x(3)) + 1.06;
y3 = exp(-x(1).*x(2))+20*x(3)+(10*pi-3)/3;
u = y1.^2 + y2.^2 + y3.^2;
end