Christopher Lum
lum@uw.edu

Lecture10c
Unconstrained Optimization

Lecture is on YouTube

The YouTube video entitled ‘Unconstrained Optimization’ that covers this lecture is located at https://youtu.be/6NB4QiKId2w.

Outline

-Unconstrained Optimization
- -1D Example
- -Higher Order Extrema

-Conditions for Unconstrained Optimality
- -Definiteness of Hermitian Matrices
- -Analytically Finding Minima

-Problems with a Pure Theory Approach to Solving Unconstrained Optimization Problems

Unconstrained Optimization

Within the context of unconstrained optimization, the problem is often stated as

$$(\wp) \quad J^* = \min_{x \in \mathbb{R}^n} f(x)$$

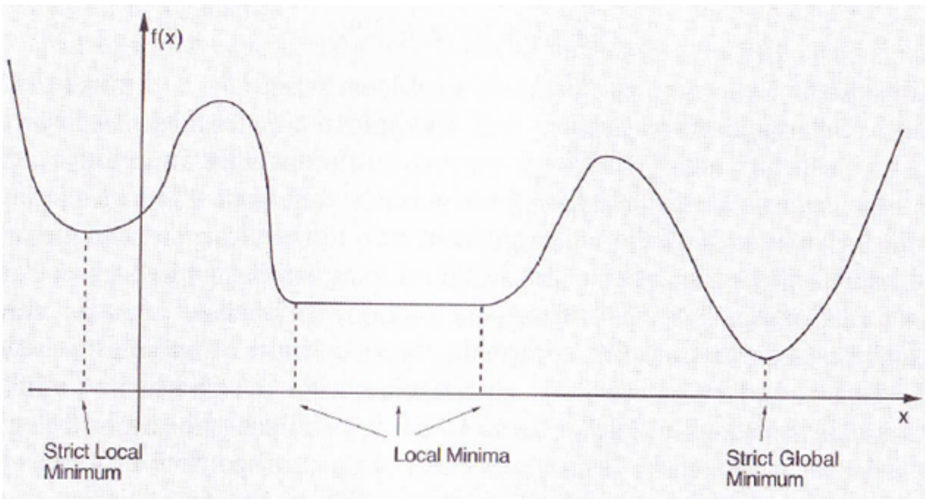
The problem stated in $(\wp)$ asks to find the minimum of the cost function by choosing any possible value of $x \in X$. If $X = \mathbb{R}^n$ then the problem is unconstrained. Often, it is just as important (if not more so) to find the value of x which yields this minimum. In this case, the problem is typically stated as

$$(\wp) \quad x^* = \arg\min_{x \in \mathbb{R}^n} f(x)$$

At this point, it becomes useful to only consider the case of minimizing a cost function and therefore only discuss finding minimums. As discussed previously, we do so without a loss of generality due to the ability to convert a maximization to a minimization by simply multiplying the cost/objective function by -1.

1D Example

We can start by considering a simple 1D scenario. Conceptually, we can visualize different situations of minima as shown below.



From elementary calculus, we know that for a 1D function, the function has a local minimum or maximum at locations where the derivative is equal to 0.

$$\frac{d f(x^*)}{d x} = 0$$

Once we find the locations x^* that satisfy $\frac{d f(x^*)}{d x} = 0$, we can apply the so-called 'second derivative test' to determine if this is a local min or max. If the function f is twice-differentiable at the point x^* then

$\frac{d^2 f(x^*)}{d x^2} < 0$ then f has a local maximum at x^*

$\frac{d^2 f(x^*)}{d x^2} > 0$ then f has a local minimum at x^*

$\frac{d^2 f(x^*)}{d x^2} = 0$ then the test in inconclusive

Example: 1D Function

```
f[x_] = -x^2 - x^3/3 + x^4/4; (*leads to nominal/normal 2nd derivative test*)
(*f[x_]=(x-1)^3(x+2);  (*leads to inconclusive 2nd derivative test*)*)

Print["df/dx"]
dfdx[x_] = D[f[x], x]
Print[" "]

Print["x*"]
temp = Solve[dfdx[x] == 0, x];
xstar1 = x /. temp[[1]]
xstar2 = x /. temp[[2]]
xstar3 = x /. temp[[3]]
Print[" "]

Print["f(x*)"]
f[xstar1]
f[xstar2]
f[xstar3]

Print["df^2/dx^2"]
df2dx2[x_] = D[dfdx[x], x]
Print[" "]
Print["2nd Derivative Test"]
df2dx2[xstar1]
df2dx2[xstar2]
df2dx2[xstar3]
Print[" "]
```

df/dx

$-2x - x^2 + x^3$

x^*

-1

0

2

$f(x^*)$

$-\frac{5}{12}$

0

$-\frac{8}{3}$

```
df^2/dx^2
```

$-2 - 2x + 3x^2$

```
2nd Derivative Test
```

3

-2

6

We can visualize this situation

```
(*Plot f*)
xMin = -1.5;
xMax = 2.5;

fighfA = Plot[f[x], {x, xMin, xMax}];
```

$$\text{ptsf} = \begin{pmatrix} \text{xstar1} & \text{f[xstar1]} \\ \text{xstar2} & \text{f[xstar2]} \\ \text{xstar3} & \text{f[xstar3]} \end{pmatrix};$$

```
fighfB = ListPlot[ptsf, PlotStyle → {Green, PointSize[0.03]}];

Show[fighfA, fighfB,
 AxesLabel → {"x", "f(x)"}]

(*Plot dfdx*)
fighdfdxA = Plot[dfdx[x], {x, xMin, xMax}];
```

$$\text{ptsdfdx} = \begin{pmatrix} \text{xstar1} & 0 \\ \text{xstar2} & 0 \\ \text{xstar3} & 0 \end{pmatrix};$$

```
fighdfdxB = ListPlot[ptsdfdx, PlotStyle → {Green, PointSize[0.03]}];
```

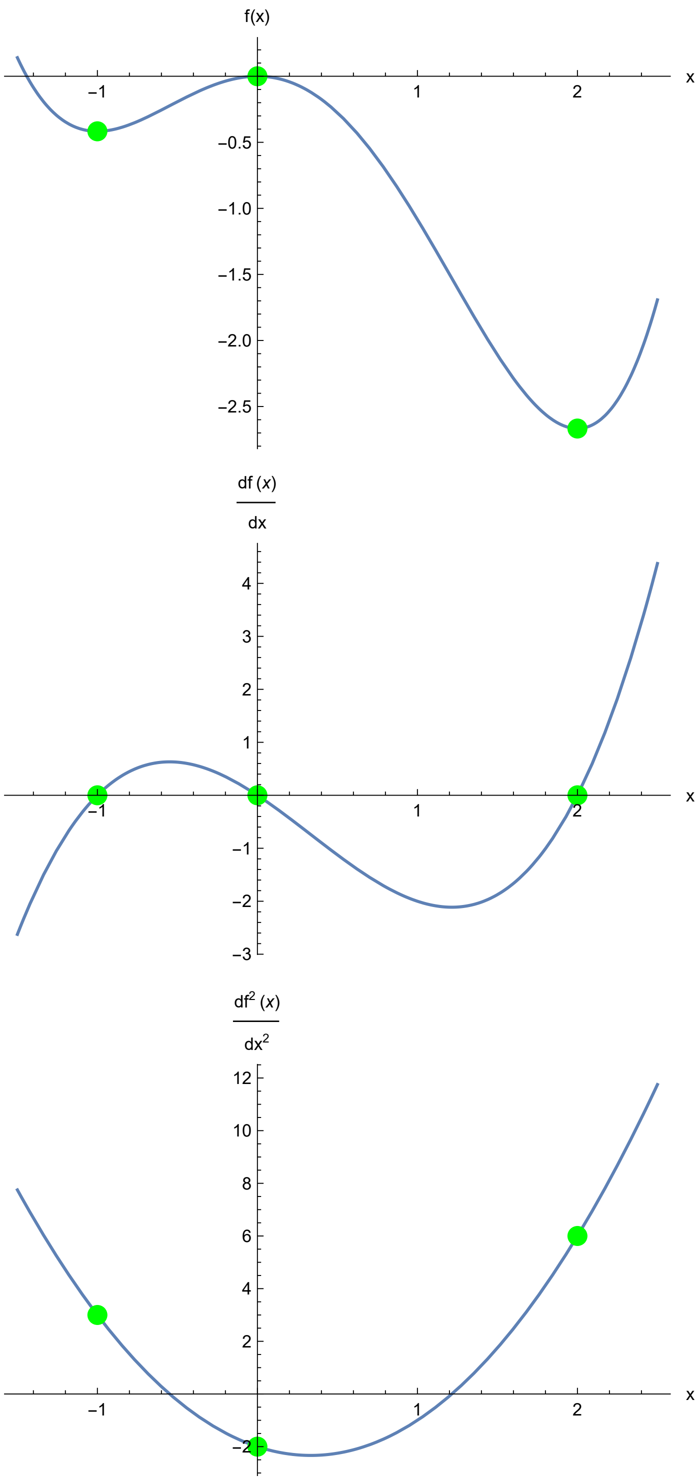
$$\text{Show}\left[\text{fighdfdxA, fighdfdxB, AxesLabel} \to \left\{\text{"x"}, \text{"}\frac{\text{df (x)}}{\text{dx}}\text{"}\right\}\right]$$

```
(*Plot df2dx2*)
fighdf2dx2A = Plot[df2dx2[x], {x, xMin, xMax}];
```

$$\text{ptsdf2dx2} = \begin{pmatrix} \text{xstar1} & \text{df2dx2[xstar1]} \\ \text{xstar2} & \text{df2dx2[xstar2]} \\ \text{xstar3} & \text{df2dx2[xstar3]} \end{pmatrix};$$

```
fighdf2dx2B = ListPlot[ptsdf2dx2, PlotStyle → {Green, PointSize[0.03]}];
```

$$\text{Show}\left[\text{fighdf2dx2A, fighdf2dx2B, AxesLabel} \to \left\{\text{"x"}, \text{"}\frac{\text{df}^2\text{ (x)}}{\text{dx}^2}\text{"}\right\}\right]$$



Higher Dimension Extrema

Consider a region *B* where *f* is defined over this region. By definition, *f* has a **minimum** at the point x^* in *B* if

$$f(x) \geq f(x^*) \qquad \text{for all } x \in B \qquad \Rightarrow \qquad x^* \text{ is a minimum in } B$$

Similarly, f has a **maximum** at x^* if

$$f(x) \leq f(x^*) \qquad \text{for all } x \in B \qquad \Rightarrow \qquad x^* \text{ is a maximum in } B$$

Minima and maxima together are referred to as **extrema**.

In an unconstrained scenario, we consider finding a point $x^* \in \mathcal{R}^n$ which has special properties.

A point x^* is an **unconstrained local minimum** of f if it is no worse than its neighbors; that is, if there exists an $\epsilon > 0$ such that

$$f(x^*) \leq f(x) \qquad \forall\, x \text{ with } \| x - x^* \| < \epsilon$$

We can extend this idea of an unconstrained local minimum to the concept of an **unconstrained global minimum**. A point x^* is said to be an unconstrained global minimum if

$$f(x^*) \leq f(x) \qquad \forall\, x \in \mathcal{R}^n$$

The unconstrained local or global minimum x^* is said to be strict if the corresponding inequalities above are strict for $x \neq x^*$.

Conditions for Unconstrained Optimality

If the cost function is differentiable, we can use gradients and Taylor series expansions to compare the cost of a point with the cost of its close neighbors.

See the YouTube video entitled ‘The Taylor Series’ before proceeding (located at https://youtu.be/k-bV9LdQXVtg)

Go to Taylor Series lecture here

For example, we consider a first order Taylor series expansion of the cost function expanded about the point x^*. This can be written as

$$f(x) \approx f(x^*) + \nabla f(x^*)^T \cdot (x - x^*) \qquad \textbf{\textit{(Eq.1)}}$$

We now consider small variations, Δx, from the given point x^*. In other words, we consider points $x = x^* + \Delta x$. Substituting this into Eq.1 yields

$$f(x^* + \Delta x) \approx f(x^*) + \nabla f(x^*)^T \cdot (x^* + \Delta x - x^*)$$

$$\approx f(x^*) + \nabla f(x^*)^T \cdot \Delta x$$

$$f(x^* + \Delta x) - f(x^*) = \nabla f(x^*)^T \cdot \Delta x \qquad \textbf{\textit{(Eq.2)}}$$

In Eq.2, we see that the LHS represents the cost function change from the point x^* to the point $x^* + \Delta x$. We see that in order for the point x^* to be a local minimum, we require that no matter which direction we move in, Δx, the LHS is non-negative. Conversely, in order for x^* to be a local maximum, we require that no matter Δx, the LHS is non-positive. Since Δx can be any direction, we see that a local min or max can occur only if the gradient at the point x^* is zero.

$$\nabla f(x^*) = \overline{0} \qquad \textbf{\textit{(Eq.3)}}$$

A point x^* at which Eq.3 holds is called a **stationary point** of f. These are also sometimes referred to as **critical points** but in optimization, they are more commonly referred to using the stationary point nomenclature.

Recall that from 1D calculus, we learned that a max/min of a function occurs when the derivative is 0. In a very rough sense, this is the multi-dimensional version of this concept.

Eq.3 is necessary for an extremum of f at x^* in the interior of R, but is not sufficient. In other words

$\{x^*$ is an extremum$\}$ implies $\left\{\nabla f(x^*) = \overline{0}\right\}$

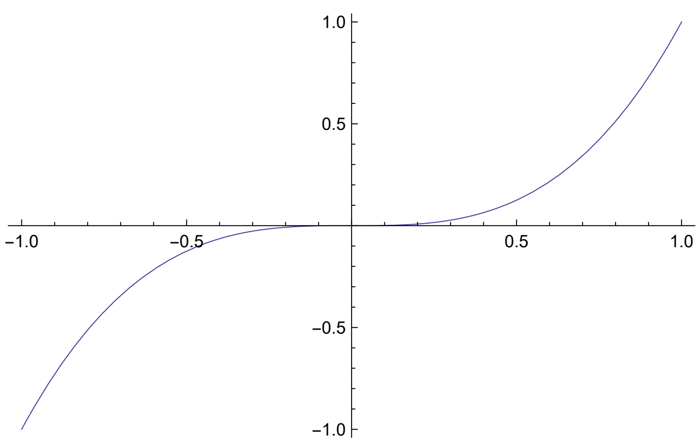
$\left\{\nabla f(x^*) = \overline{0}\right\}$ does not imply $\{x^*$ is an extremum$\}$

Example: Show that $\left\{\nabla f(x^*) = \overline{0}\right\}$ Does Not Imply $\{x^*$ is an extremum$\}$

We can show that $\left\{\nabla f(x^*) = \overline{0}\right\}$ does not imply $\{x^*$ is an extremum$\}$

Consider the function $f = x^3$ defined over the region $R = x \in [-1, 1]$.

```
f[x_] = x^3;
Plot[f[x], {x, -1, 1}]
```

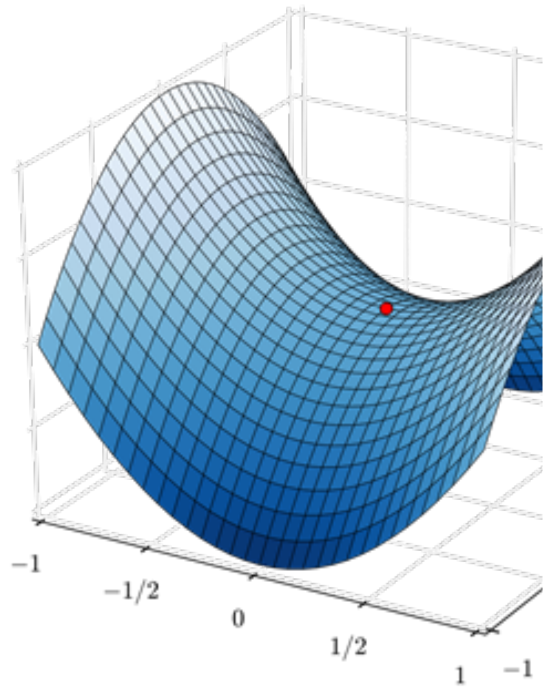


We can easily find the stationary points by finding where the gradient is zero. This obviously occurs at $x^* = 0$

```
Solve[D[f[x], x] == 0, x]
{{x → 0}, {x → 0}}
```

However, it is easy to see that this is not an extremum of the function.

In higher dimensions, examples which illustrate this fact typically involve **saddle points**.



The consequence of this is that after solving Eq.3, we can only consider the stationary points to be candidate extremum and must do additional testing to determine if it is indeed an extremum.

We can repeat this analysis for higher dimension functions except now we use a second order Taylor series expansion of the cost function expanded about the point x^*, which can be written as

$$f(x) \approx f(x^*) + \nabla f(x^*)^T \cdot (x - x^*) + \frac{1}{2!} (x - x^*)^T \cdot \nabla^2 f(x^*) \cdot (x - x^*) \qquad \textbf{\textit{(Eq.4)}}$$

where $\quad \nabla^2 f(x^*)$ = Hessian of f at x^*

We again consider small variations, Δx, from the given point x^*. In other words, we consider points $x = x^* + \Delta x$. Substituting this into Eq.4 yields

$$f(x^* + \Delta x) \approx f(x^*) + \nabla f(x^*)^T \cdot (x^* + \Delta x - x^*) + \frac{1}{2!} (x^* + \Delta x - x^*)^T \cdot \nabla^2 f(x^*) \cdot (x^* + \Delta x - x^*)$$

$$\approx f(x^*) + \nabla f(x^*)^T \cdot \Delta x + \frac{1}{2!} \Delta x^T \cdot \nabla^2 f(x^*) \cdot \Delta x$$

$$f(x^* + \Delta x) - f(x^*) \approx \nabla f(x^*)^T \cdot \Delta x + \frac{1}{2!} \Delta x^T \cdot \nabla^2 f(x^*) \cdot \Delta x$$

At a stationary point, x^*, Eq.3 states that $\nabla f(x^*) = \overline{0}$, so this becomes

$$f(x^* + \Delta x) - f(x^*) \approx \frac{1}{2!} \Delta x^T \cdot \nabla^2 f(x^*) \cdot \Delta x \qquad \textbf{\textit{(Eq.5)}}$$

In the situation of unconstrained optimization, a point x^* is an **unconstrained local minimum** of f if it is no worse than its neighbors. From Eq.5, we see that we therefore require that the entire right side is non-negative

$$\frac{1}{2!}\Delta x^T \cdot \nabla^2 f(x^*) \cdot \Delta x \geq 0 \qquad \forall\, \Delta x$$
$$\Delta x^T \cdot \nabla^2 f(x^*) \cdot \Delta x \geq 0 \qquad \forall\, \Delta x \qquad \textbf{\textit{(Eq.6)}}$$

The inequality in Eq.6 is equivalent to requiring $\nabla^2 f(x^*)$ (the Hessian matrix of the cost function at the point x^*) to be positive semi-definite

$$\{\Delta x^T \cdot \nabla^2 f(x^*) \cdot \Delta x \geq 0\, \forall\, \Delta x\} \iff \{\nabla^2 f(x^*) \text{ is positive semi-definite}\} \qquad \textbf{\textit{(Eq.7)}}$$

Once again, we learned that for a 1D case, in order for a function to have a minimum, its first derivative must be zero and the second derivative must be positive. The concept of a positive semi-definite Hessian matrix is the multi-dimensional extension of this concept.

We can now use the below theorem to analyze whether $f(x)$ has any local or global minima or maxima.

Theorem 1: Local Optimality Conditions Without Constraints
For a function $f : \mathcal{R}^n \to \mathcal{R}$ of class C^2, consider the problem of minimizing f over all $x \in \mathcal{R}^n$

(a) (necessary condition). If x^* is a locally optimal solution, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semi-definite.
(b) (sufficient condition). If x^* is such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, then x^* is a strict locally optimal solution.

Therefore, a somewhat naive method for finding minimum of the cost function is

Step 1: Find stationary points x^* that satisfy $\nabla f(x^*) = \overline{0}$
Step 2: At each of these stationary points, determine the definiteness of $\nabla^2 f(x^*)$
- If $\nabla^2 f(x^*) \geq 0$ (positive semi-definite) then x^* is a local minimum
- If $\nabla^2 f(x^*) > 0$ (positive definite) then x^* is a (strict) local minimum
- otherwise x^* may not be a local minimum

This is sometimes referred to as the **Second Partial Derivative Test** (https://en.wikipedia.org/wiki/Second_partial_derivative _test)and is stated as follows

If $\nabla^2 f(x^*) > 0$ (positive definite) then x^* is a (strict) local minimum
If $\nabla^2 f(x^*) < 0$ (negative definite) then x^* is a (strict) local maximum
If $\nabla^2 f(x^*)$ has both positive and negative eigenvalues and x^* is a saddle point
otherwise the test is inconclusive

This naturally raises the question, how can one check if a matrix is positive semi-definite, positive definite, or otherwise?

Definiteness of Hermitian Matrices

Theorem 2: Eigenvalues and Definiteness of a Hermitian Matrix
See https://en.wikipedia.org/wiki/Definite_matrix#Eigenvalues

Let M be an $n \times n$ Hermitian matrix.

M is positive definite $\Longleftrightarrow$ all of its eigenvalues are positive
M is positive semi-definite $\Longleftrightarrow$ all of its eigenvalues are non-negative
M is negative definite $\Longleftrightarrow$ all of its eigenvalues are negative
M is negative semi-definite $\Longleftrightarrow$ all of its eigenvalues are non-positive
M is indefinite $\Longleftrightarrow$ it has at least one positive eigenvalue and at least one negative eigenvalue

Or alternatively written as

$$
\begin{aligned}
M > 0 &\Longleftrightarrow \lambda_i > 0 \quad \forall i \\
M \geq 0 &\Longleftrightarrow \lambda_i \geq 0 \quad \forall i \\
M < 0 &\Longleftrightarrow \lambda_i < 0 \quad \forall i \\
M \leq 0 &\Longleftrightarrow \lambda_i \leq 0 \quad \forall i
\end{aligned}
$$

M is indefinite $\Longleftrightarrow$ it has at least one positive eigenvalue and at least one negative eigenvalue

Recall that Hermitian matrices are the complex extensions of symmetric matrices.

Also note that since the Hessian matrix is symmetric, all the eigenvalues are real and therefore, we do not need to consider imaginary eigenvalues in the theorem statement.

Example: 2D Symmetric Matrices and Definiteness

Let us consider 5 symmetric matrices

$$\mathbf{H1} = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix}; \quad \text{(* ++ *)}$$

$$\mathbf{H2} = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}; \quad \text{(* +0 *)}$$

$$\mathbf{H3} = \begin{pmatrix} -8 & -5 \\ -5 & -5 \end{pmatrix}; \quad \text{(* -- *)}$$

$$\mathbf{H4} = \begin{pmatrix} -5 & -5 \\ -5 & -5 \end{pmatrix}; \quad \text{(* -0 *)}$$

$$\mathbf{H5} = \begin{pmatrix} 5 & 3 \\ 3 & -4 \end{pmatrix}; \quad \text{(* +- *)}$$

We can now look at the functions

$$f_i(x_1, x_2) = x^T H_i x$$

where $x = \begin{pmatrix} x_1 & x_2 \end{pmatrix}^T$
H_i =symmetric matrix

We can plot each of these function to get an idea of its behavior of various values of x_1 and x_2

```
x1Min = -2;
x1Max = 2;
x2Min = -2;
x2Max = 2;

(*Define a module to analyze the scneario*)
AnalyzeScenario[H_, x1Min_, x1Max_, x2Min_, x2Max_] := Module[{x, f},
  (*Display the H matrix*)
  Print["H"];
  Print[MatrixForm[H]];
  Print[" "];

  (*Eigenvalues of the H matrix*)
  Print["Eigenvalues of H"];
  Print[N[Eigenvalues[H]]];
  Print[" "];

  (*Define the function f=x^T*H*x  *)
  x = (x1
       x2);
  f[x1_, x2_] = (Transpose[x].H.x)[[1, 1]] // Expand;

  (*Print out the function*)
  Print["f=x^T*H*x"];
  Print[f[x1, x2]];
  Print[" "];

  (*Plot the function*)
  Plot3D[f[x1, x2], {x1, x1Min, x1Max}, {x2, x2Min, x2Max},
   AxesLabel -> {"x1", "x2", "f(x1,x2)"}]
 ]

(*Look at the different cases*)
AnalyzeScenario[H1, x1Min, x1Max, x2Min, x2Max]
AnalyzeScenario[H2, x1Min, x1Max, x2Min, x2Max]
AnalyzeScenario[H3, x1Min, x1Max, x2Min, x2Max]
AnalyzeScenario[H4, x1Min, x1Max, x2Min, x2Max]
AnalyzeScenario[H5, x1Min, x1Max, x2Min, x2Max]
```

H

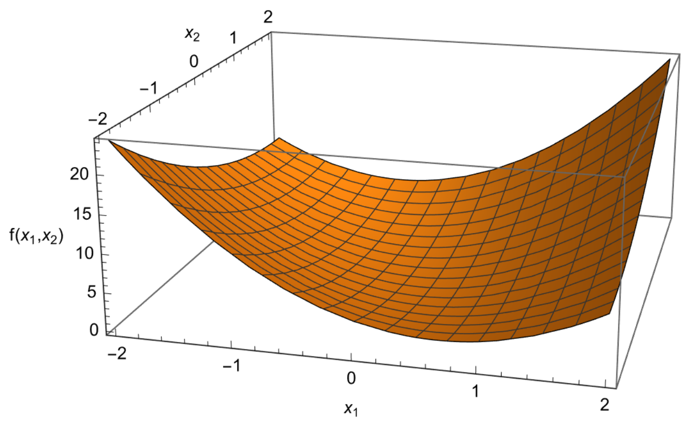
$$\begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix}$$

Eigenvalues of H

{3.41421, 0.585786}

f=x^T*H*x

$3\,x1^2 + 2\,x1\,x2 + x2^2$



H

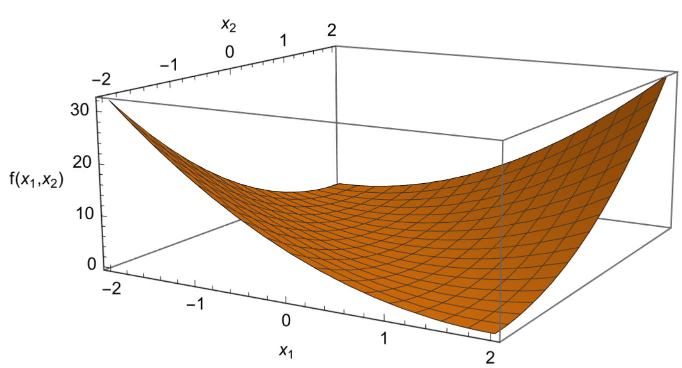
$$\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$

Eigenvalues of H

{4., 0.}

f=x^T*H*x

$2\,x1^2 + 4\,x1\,x2 + 2\,x2^2$



H

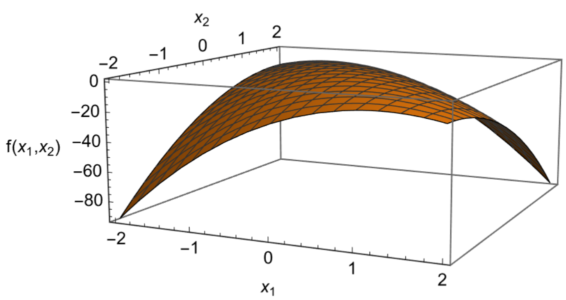
$$\begin{pmatrix} -8 & -5 \\ -5 & -5 \end{pmatrix}$$

Eigenvalues of H

{-11.7202, -1.27985}

f=x^T*H*x

$-8\,x1^2 - 10\,x1\,x2 - 5\,x2^2$



H

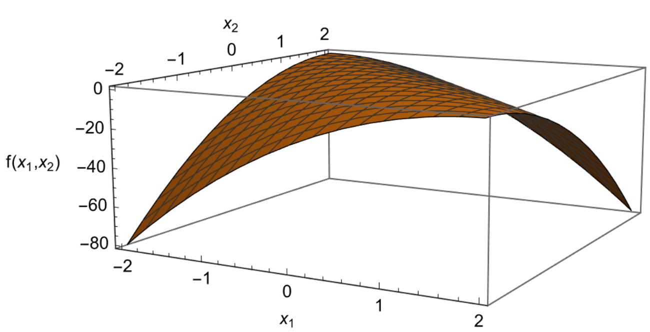
$$\begin{pmatrix} -5 & -5 \\ -5 & -5 \end{pmatrix}$$

Eigenvalues of H

{-10., 0.}

f=x^T*H*x

$-5\,x1^2 - 10\,x1\,x2 - 5\,x2^2$



H

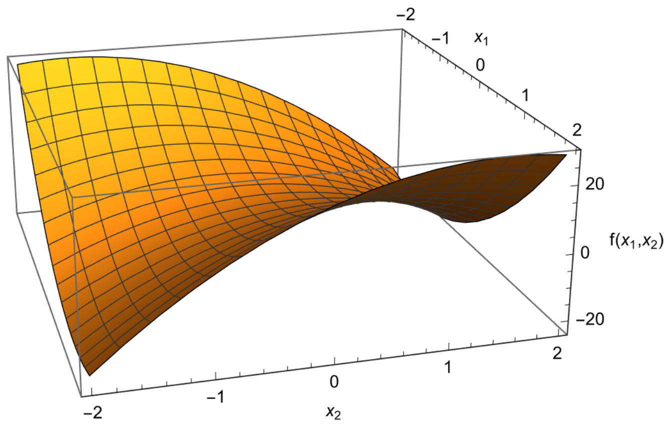
$$\begin{pmatrix} 5 & 3 \\ 3 & -4 \end{pmatrix}$$

Eigenvalues of H

{5.90833, -4.90833}

$f=x^T * H * x$

$5\ x1^2 + 6\ x1\ x2 - 4\ x2^2$



From this example, we notice that the behavior is based on the eigenvalues, λ_1, λ_2 = eigenvalues(H).

$$\begin{array}{lll}
\lambda_1 > 0, \lambda_2 > 0 & \Rightarrow & f_i(x) = x^T H_i x > 0 \quad \forall x \neq 0 \\
\lambda_1 > 0, \lambda_2 = 0 & \Rightarrow & f_i(x) = x^T H_i x \geq 0 \quad \forall x \neq 0 \\
\lambda_1 < 0, \lambda_2 < 0 & \Rightarrow & f_i(x) = x^T H_i x < 0 \quad \forall x \neq 0 \\
\lambda_1 < 0, \lambda_2 = 0 & \Rightarrow & f_i(x) = x^T H_i x \leq 0 \quad \forall x \neq 0 \\
\lambda_1 > 0, \lambda_2 < 0 & \Rightarrow & f_i(x) = x^T H_i x \text{ is positive or negative}
\end{array}$$

Analytically Finding Minima

So combining Theorem 1 and Theorem 2, we have the simple test for local min/max of stationary points

If $\nabla^2 f(x^*)$ has all eigenvalues ≥ 0 then x^* is a local minimum
If $\nabla^2 f(x^*)$ has all eigenvalues > 0 then x^* is a (strict) local minimum
otherwise x^* may not be a local minimum

In theory, the previous procedure seems like a reasonable procedure for finding stationary points and local minima of a function.

Example: Analytically Find Stationary Point

Consider the cost function shown below

```
Print["f"]
fA[x1_, x2_] = (x1 + x2) (x1 x2 + x1 x2^2) // Expand
```

```
Print[" "]

Print["∇f"]
gradFA[x1_, x2_] = Grad[fA[x1, x2], {x1, x2}];
gradFA[x1, x2] // MatrixForm
Print[" "]

Print["Solve ∇f=0"]
temp = Solve[{gradFA[x1, x2][[1]] == 0, gradFA[x1, x2][[2]] == 0}, {x1, x2}];
Print["Number of solutions = ", Length[temp]]

Print["x* solution 1"]
x1star1 = x1 /. temp[[1]]
x2star1 = x2 /. temp[[1]]

Print["x* solution 2"]
x1star2 = x1 /. temp[[2]]
x2star2 = x2 /. temp[[2]]

Print["x* solution 3"]
x1star3 = x1 /. temp[[3]]
x2star3 = x2 /. temp[[3]]

Print["x* solution 4"]
x1star4 = x1 /. temp[[4]]
x2star4 = x2 /. temp[[4]]

Print["x* solution 5"]
x1star5 = x1 /. temp[[5]]
x2star5 = x2 /. temp[[5]]
Print[" "]

Print["f(x*) solution 1"]
fA[x1star1, x2star1]

Print["f(x*) solution 2"]
fA[x1star2, x2star2]

Print["f(x*) solution 3"]
fA[x1star3, x2star3]

Print["f(x*) solution 4"]
fA[x1star4, x2star4]

Print["f(x*) solution 5"]
fA[x1star5, x2star5]
```

```
Print[" "]
```

f

$x1^2\, x2 + x1\, x2^2 + x1^2\, x2^2 + x1\, x2^3$

▽f

$$\begin{pmatrix} 2\, x1\, x2 + x2^2 + 2\, x1\, x2^2 + x2^3 \\ x1^2 + 2\, x1\, x2 + 2\, x1^2\, x2 + 3\, x1\, x2^2 \end{pmatrix}$$

Solve ▽f=0

Number of solutions = 5

x^* solution 1

0

0

x^* solution 2

0

−1

x^* solution 3

1

−1

x^* solution 4

$\frac{3}{8}$

$-\frac{3}{4}$

x^* solution 5

0

0

$f(x^*)$ solution 1

0

$f(x^*)$ solution 2

0

$f(x^*)$ solution 3

```
0
f(x*) solution 4
```

$$\frac{27}{1024}$$

```
f(x*) solution 5
0
```

Note that solution 5 is the same as solution 1. So we have 4 stationary points

$$x^* \text{ solution } 1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$x^* \text{ solution } 2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

$$x^* \text{ solution } 3 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$x^* \text{ solution } 4 = \begin{pmatrix} 3/8 \\ -3/4 \end{pmatrix}$$

```
Print["∇²f(x*)"]
H[x1_, x2_] = ( D[fA[x1, x2], x1, x1]  D[fA[x1, x2], x1, x2]
                D[fA[x1, x2], x2, x1]  D[fA[x1, x2], x2, x2] );
H[x1, x2] // MatrixForm
Print[" "]

Print["2nd Partial Derivative Test"]
Eigenvalues[H[x1star1, x2star1]] // N
Eigenvalues[H[x1star2, x2star2]] // N
Eigenvalues[H[x1star3, x2star3]] // N
Eigenvalues[H[x1star4, x2star4]] // N
Print[" "]
```

$\nabla^2 f(x^*)$

$$\begin{pmatrix} 2\,x2 + 2\,x2^2 & 2\,x1 + 2\,x2 + 4\,x1\,x2 + 3\,x2^2 \\ 2\,x1 + 2\,x2 + 4\,x1\,x2 + 3\,x2^2 & 2\,x1 + 2\,x1^2 + 6\,x1\,x2 \end{pmatrix}$$

```
2nd Partial Derivative Test
{0., 0.}
{-1., 1.}
{-2.41421, 0.414214}
{-0.75, -0.28125}
```

So we classify the 4 stationary points as

x^* solution $1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $\lambda_1 = 0,\ \lambda_2 = 0$ inconclusive (saddle point)

x^* solution $2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ $\lambda_1 = -1,\ \lambda_2 = 1$ inconclusive (saddle point)

x^* solution $3 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ $\lambda_1 = -2.4,\ \lambda_2 = 0.4$ inconclusive (saddle point)

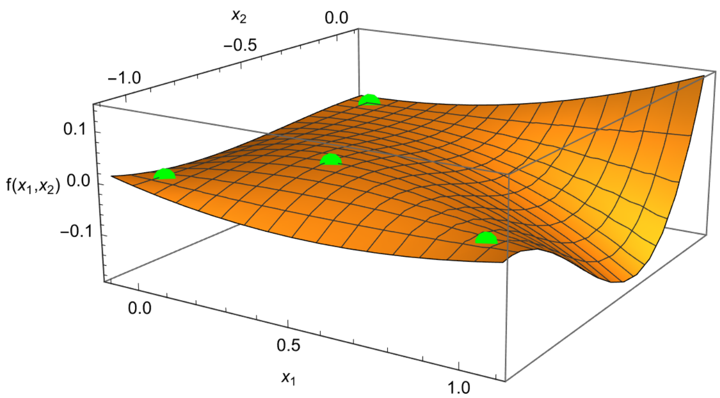
x^* solution $4 = \begin{pmatrix} 3/8 \\ -3/4 \end{pmatrix}$ $\lambda_1 = -0.75,\ \lambda_2 = -0.28$ strict local maximum

We can visualize the scenario

```
(*Plot fA*)
x1Min = -0.1;
x1Max = 1.1;

x2Min = -1.1;
x2Max = 0.1;

fighfA = Plot3D[fA[x1, x2], {x1, x1Min, x1Max}, {x2, x2Min, x2Max}];
```

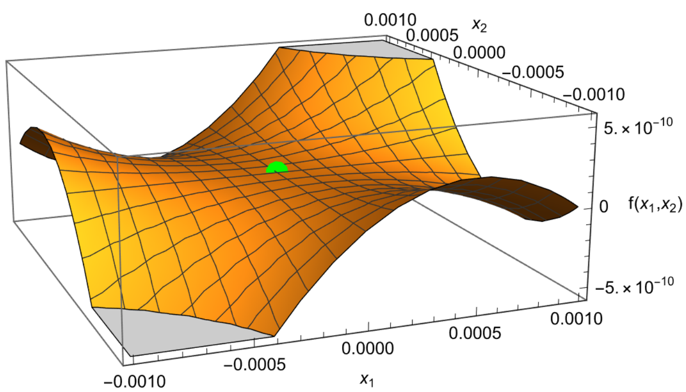
$$\text{ptsf} = \begin{pmatrix} \text{x1star1} & \text{x2star1} & \text{fA[x1star1, x2star1]} \\ \text{x1star2} & \text{x2star2} & \text{fA[x1star2, x2star2]} \\ \text{x1star3} & \text{x2star3} & \text{fA[x1star3, x2star3]} \\ \text{x1star4} & \text{x2star4} & \text{fA[x1star4, x2star4]} \end{pmatrix};$$

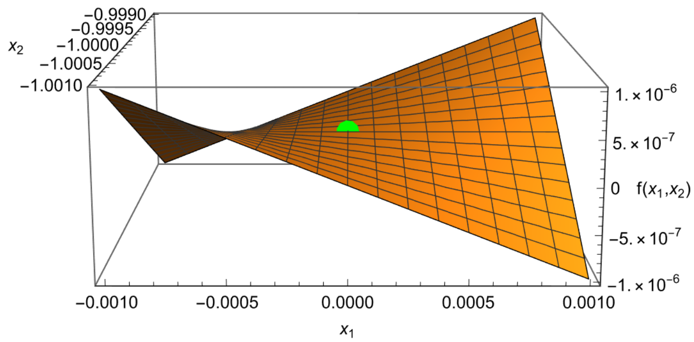
```
pts = Point[ptsf];
fighfB = Graphics3D[
   {AbsolutePointSize[12], Green, pts}
  ];

Show[fighfA, fighfB,
 AxesLabel → {"x1", "x2", "f(x1,x2)"}]
```

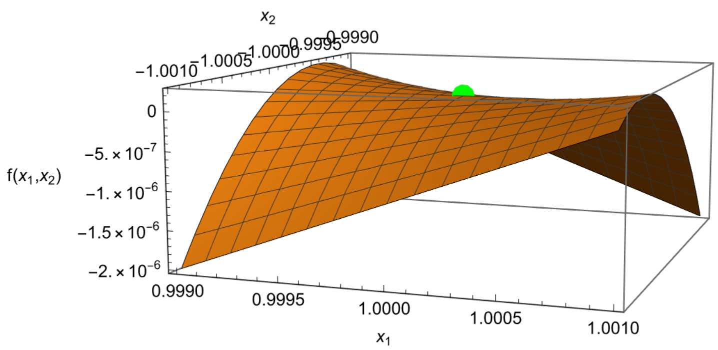


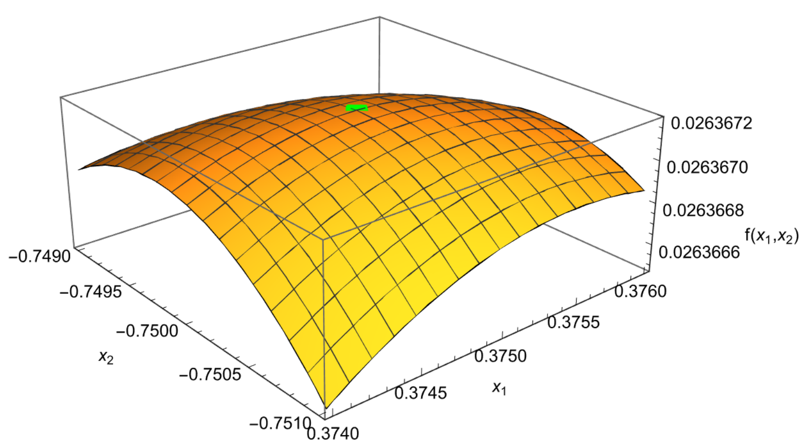
We can zoom into the plot at each of these stationary points to better visualize the behavior of the function

```
deltaX1 = 0.001;
deltaX2 = 0.001;

(*xstar1*)
fighfA1 = Plot3D[fA[x1, x2], {x1, x1star1 + deltaX1, x1star1 - deltaX1},
   {x2, x2star1 + deltaX2, x2star1 - deltaX2}];
Show[fighfA1, fighfB,
 AxesLabel → {"x1", "x2", "f(x1,x2)"}]

(*xstar2*)
fighfA2 = Plot3D[fA[x1, x2], {x1, x1star2 + deltaX1, x1star2 - deltaX1},
   {x2, x2star2 + deltaX2, x2star2 - deltaX2}];
Show[fighfA2, fighfB,
 AxesLabel → {"x1", "x2", "f(x1,x2)"}]

(*xstar3*)
fighfA3 = Plot3D[fA[x1, x2], {x1, x1star3 + deltaX1, x1star3 - deltaX1},
   {x2, x2star3 + deltaX2, x2star3 - deltaX2}];
Show[fighfA3, fighfB,
 AxesLabel → {"x1", "x2", "f(x1,x2)"}]

(*xstar4*)
fighfA4 = Plot3D[fA[x1, x2], {x1, x1star4 + deltaX1, x1star4 - deltaX1},
   {x2, x2star4 + deltaX2, x2star4 - deltaX2}];
Show[fighfA4, fighfB,
 AxesLabel → {"x1", "x2", "f(x1,x2)"}]
```









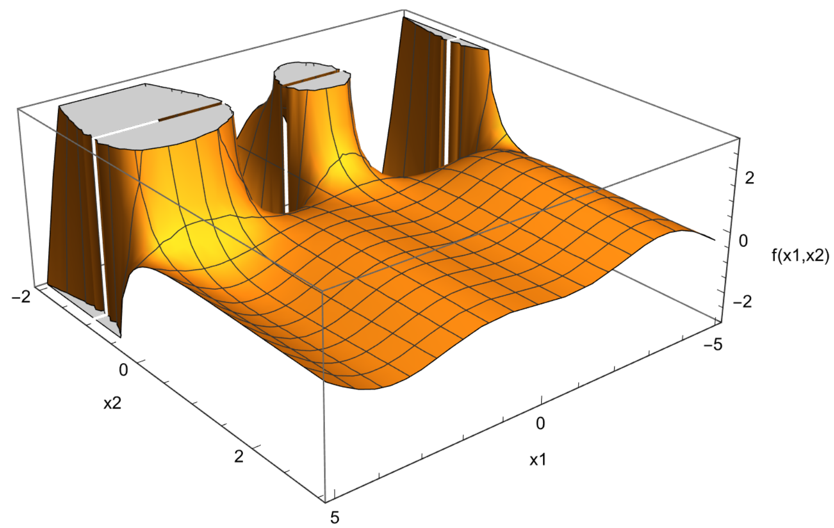
Problems with a Pure Theory Approach to Solving Unconstrained Optimization Problems

However, if the cost function is somewhat complicated, the solving for locations where the gradient is 0 is difficult in practice as is shown in the following example.

Example: Attempt to Analytically Find Stationary Point

Consider the cost function below

```
fB[x1_, x2_] = Cos[x1] x1 x2 / (x2 + 1)^2;
Plot3D[fB[x1, x2], {x1, -5, 5}, {x2, -2, 3},
 AxesLabel → {"x1", "x2", "f(x1,x2)"}]
```



We can compute the gradient

```
gradfB[x1_, x2_] = Grad[fB[x1, x2], {x1, x2}];
gradfB[x1, x2] // MatrixForm
```

$$\begin{pmatrix} \frac{x2 \text{Cos}[x1]}{(1+x2)^2} - \frac{x1\, x2 \text{Sin}[x1]}{(1+x2)^2} \\ -\frac{2\, x1\, x2 \text{Cos}[x1]}{(1+x2)^3} + \frac{x1 \text{Cos}[x1]}{(1+x2)^2} \end{pmatrix}$$

As can be seen, it is not entirely clear if there are any points where the gradient is zero. In fact, if we try to explicitly solve

```
Solve[{gradfB[x1, x2][[1]] == 0, gradfB[x1, x2][[2]] == 0}, {x1, x2}]
```

Solve: This system cannot be solved with the methods available to Solve.

$$\text{Solve}\left[\left\{\frac{x2 \text{Cos}[x1]}{(1+x2)^2} - \frac{x1\, x2 \text{Sin}[x1]}{(1+x2)^2} == 0,\ -\frac{2\, x1\, x2 \text{Cos}[x1]}{(1+x2)^3} + \frac{x1 \text{Cos}[x1]}{(1+x2)^2} == 0\right\}, \{x1, x2\}\right]$$

This difficultly in analytically computing the stationary points motivates the development of a numerical algorithms to find the stationary points.