Christopher Lum
lum@uw.edu

Lecture 10b
Taylor Series

Lecture is on YouTube

The YouTube video entitled ‘The Taylor Series’ that covers this lecture is located at https://youtu.be/k-bV9LdQXVtg.

Outline

-Introduction
-Derivation
-Taylor and Maclaurin Series
-Usage of Series to Evaluate Integrals
-Taylor Series in Several Variables

Introduction

A Taylor series is a representation of a function as an infinite sum of terms that are calculated from the values of the function’s derivatives at a single point.

Derivation

Suppose that f is any function that can be represented by a power series

$$f(x) = c_0 + c_1(x-a) + c_2(x-a)^2 + c_3(x-a)^3 + c_4(x-a)^4 + \ldots \qquad |x-a| < R \qquad \textbf{(Eq.1)}$$

where c_n = coefficients
a = expansion point
R =radius of convergence

We seek the coefficients c_n that with make Eq.1 true.

Notice that if $x = a$, Eq.1 becomes

$$f(a) = c_0$$

So we see the coefficient c_0 is simply the function evaluated at the point a.

If we differentiate Eq.1 w.r.t. x once, we obtain

$$f'(x) = c_1 + 2c_2(x-a) + 3c_3(x-a)^2 + 4c_4(x-a)^3 + \ldots \qquad |x-a| < R \qquad \textbf{(Eq.2)}$$

Once again, notice that if $x = a$, Eq.2 becomes

$$f'(a) = c_1$$

If we differentiate Eq.1 w.r.t. x twice, we obtain

$$f''(x) = 2c_2 + 2\cdot 3c_3(x-a) + 3\cdot 4c_4(x-a)^2 + \ldots \qquad |x-a| < R \qquad \textbf{(Eq.3)}$$

Once again, notice that if $x = a$, Eq.3 becomes

$$f''(a) = 2c_2$$

If we differentiate Eq.1 w.r.t. x three times, we obtain

$$f'''(x) = 2\cdot 3c_3 + 2\cdot 3\cdot 4c_4(x-a) + \ldots \qquad |x-a| < R \qquad \textbf{(Eq.4)}$$

Once again, notice that if $x = a$, Eq.4 becomes

$$f'''(a) = 2\cdot 3c_3 = 3!\,c_3$$

So we notice a pattern that

$$f^{(n)}(a) = n!\,c_n$$

Or solving for the coefficient c_n we obtain

$$c_n = \frac{f^{(n)}(a)}{n!} \qquad \textbf{(Eq.5)}$$

This result can be summarized in a theorem

Theorem: Taylor Series Coefficients

If f has a power series representation (expansion) at a, that is, if

$$f(x) = \sum_{n=0}^{\infty} c_n(x-a)^n \qquad |x-a| < R$$

then the coefficients are given by

$$c_n = \frac{f^{(n)}(a)}{n!}$$

So applying this to our situation, we see that we obtain

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n \qquad \textbf{(Eq.6)}$$

$$= f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots$$

Eq.6 is called the **Taylor series of the function *f* at *a*** (or **about *a*** or **centered at *a***)

It is common to abbreviate the infinite series and consider partial sums. In other words

$$f(x) \approx S_N(x) \triangleq \sum_{n=0}^{N} \frac{f^{(n)}(a)}{n!} (x-a)^n \qquad \textbf{(Eq.6.B)}$$

This is often referred to as the a **Taylor series approximation of the function *f* at a**.

An interesting property to note is that the Taylor series approximation of f is exact at $x = a$ as long as $N \geq 0$. In other words, this regardless of how many terms in the expansion are included, the approximation equals the function at $x = a$.

The special case of a Taylor series with $a = 0$ is referred to as the **Maclaurin series**.

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n \qquad \textbf{(Eq.7)}$$

$$= f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \ldots$$

Note: This shows that <u>if</u> f can be represented as a power series about a, then f is equal to the sum of its Taylor series. There exists functions that are not equal to the sum of their Taylor series as we will see.

Example 1: Simple Maclaurin and Taylor Series

Find

1. The Maclaurin series of the function $f(x) = e^x$
2. The Taylor series of the function $f(x) = e^x$ at $a = 2$

<u>Solution:</u>

We note that if $f(x) = e^x$, then

$$f^{(n)}(x) = e^x \text{ for all } n$$

```
In[1]:= f[x_] := Exp[x];
```

For the Maclaurin series, by Eq.7, we have

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \ldots$$

$$= e^0 + \frac{e^0}{1!}x + \frac{e^0}{2!}x^2 + \frac{e^0}{3!}x^3 + \ldots$$

$$= 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

$$f(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

In[2]:=
```
fMaclaurin[x_, P_] := Sum[x^p/p!, {p, 0, P}]
```

For the Taylor series expansion about $a = 2$, by Eq.6 we have

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots$$

$$= e^a + \frac{e^a}{1!}(x-a) + \frac{e^a}{2!}(x-a)^2 + \frac{e^a}{3!}(x-a)^3 + \ldots$$

$$= e^2 + \frac{e^2}{1!}(x-2) + \frac{e^2}{2!}(x-2)^2 + \frac{e^2}{3!}(x-2)^3 + \ldots$$

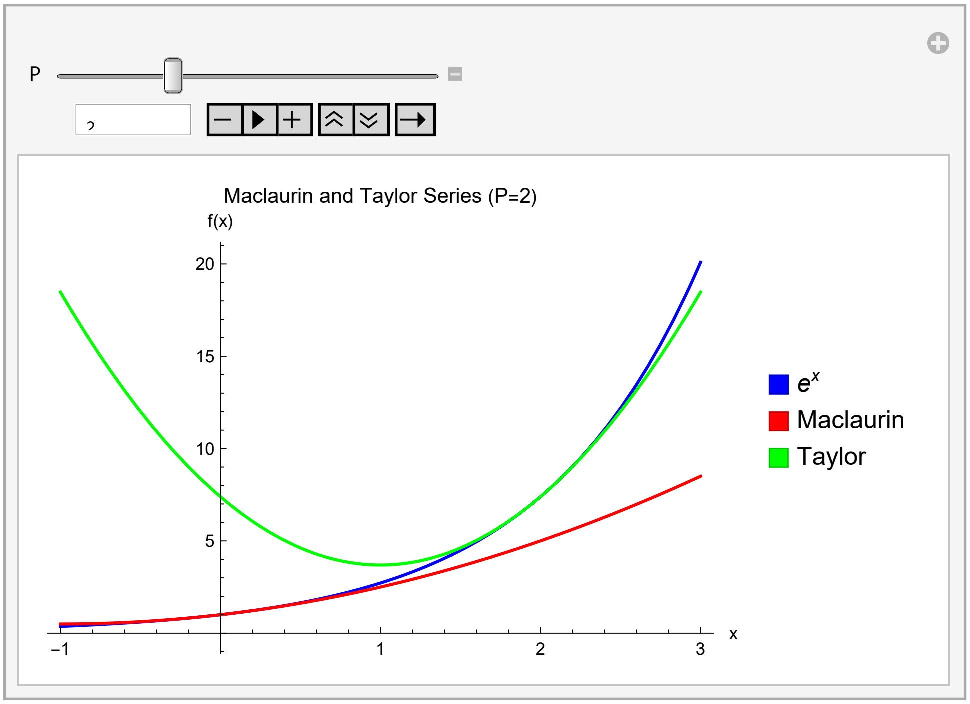
$$f(x) = \sum_{n=0}^{\infty} \frac{e^2}{n!}(x-2)^n$$

In[3]:=
```
fTaylor[x_, P_] := Sum[Exp[2]/p! (x - 2)^p, {p, 0, P}]
```

We can plot the Maclaurin and Taylor series (using a finite number of terms)

In[4]:=
```
xMin = -1;
xMax = 3;

Manipulate[
 Legended[
  Show[
   (*f(x)=e^x*)
   Plot[f[x], {x, xMin, xMax}, PlotStyle → {Blue}],

   (*Maclaurin*)
   Plot[fMaclaurin[x, P], {x, xMin, xMax}, PlotStyle → {Red}],

   (*Taylor*)
   Plot[fTaylor[x, P], {x, xMin, xMax}, PlotStyle → {Green}],

   (*Plot Options*)
   PlotLabel → StringJoin["Maclaurin and Taylor Series (P=", ToString[P], ")"],
   AxesLabel → {"x", "f(x)"}
  ],

  (*Add legend information*)
  SwatchLegend[{Blue, Red, Green}, {"e^x", "Maclaurin", "Taylor"}]
 ],

 (*Manipulated parameter*)
 {P, 0, 7, 1}
]
```

Out[6]=



We can make a table to see what these partial sums look like for various values of *P*.

In[∘]:=
```
Grid[
 Table[{P, fMaclaurin[x, P], fTaylor[x, P]}, {P, 0, 5, 1}],
 Frame → All,
 Alignment → Left]
```

Out[∘]=

0	1	e^2
1	$1+x$	$e^2+e^2(-2+x)$
2	$1+x+\frac{x^2}{2}$	$e^2+e^2(-2+x)+\frac{1}{2}e^2(-2+x)^2$
3	$1+x+\frac{x^2}{2}+\frac{x^3}{6}$	$e^2+e^2(-2+x)+\frac{1}{2}e^2(-2+x)^2+\frac{1}{6}e^2(-2+x)^3$
4	$1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}$	$e^2+e^2(-2+x)+\frac{1}{2}e^2(-2+x)^2+\frac{1}{6}e^2(-2+x)^3+\frac{1}{24}e^2(-2+x)^4$
5	$1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}$	$e^2+e^2(-2+x)+\frac{1}{2}e^2(-2+x)^2+\frac{1}{6}e^2(-2+x)^3+\frac{1}{24}e^2(-2+x)^4+\frac{1}{120}e^2(-2+x)^5$

In[∘]:=
```
Clear[xMin, xMax, f, fTaylor, fMaclaurin]
```

End Example 1:

Often, it will be useful to investigate a first order approximation of a function at the point *a*. This is obtained by only retaining the first order terms in the Taylor series and dropping all higher order terms.

$$f(x) \approx f(a) + \frac{f'(a)}{1!}(x-a)$$

Sometimes this is colloquially referred to as "linearizing" a function as it reduces to the equation of a line (or hyperplane in higher dimensions). A more mathematically accurate description would be an

affine approximation as the y-offset is not zero in general. We refer to the affine representation as **the first order approximation** of the function.

We can create the linear transformation from the first order approximation by defining

$$\Delta f(x) = f(x) - f(a)$$
$$\Delta x = x - a$$

We can interpret Δx as the distance from the expansion point a and $\Delta f(x)$ as the distance from the expansion point $f(a)$. In other words, these are perturbations from the expansion point.

We can now rewrite the first order approximation as

$$f(x) - f(x_o) \approx \frac{f'(a)}{1!}(x - x_o)$$

$$\Delta f(x) \approx \frac{f'(a)}{1!}\Delta x$$

This can now be considered linear if we are only interested in examining perturbations from the point $(a, f(a))$.

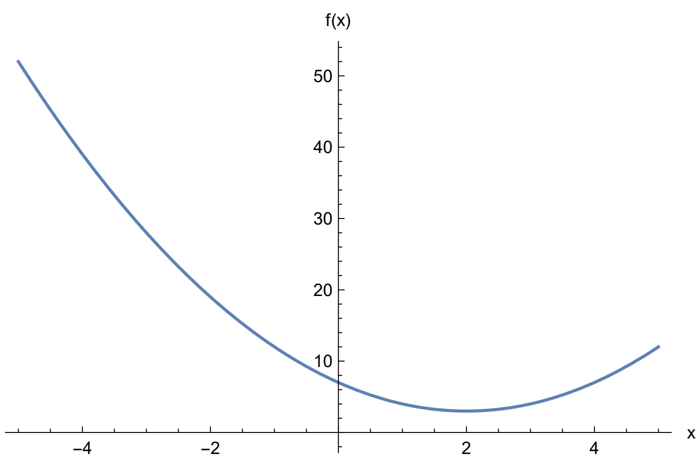
Example 2: First Order Approximation

Consider the function

$$f(x) = (x-2)^2 + 3$$

In[ ]:=
```
f[x_] = (x - 2)^2 + 3;
Plot[f[x], {x, -5, 5},
 AxesLabel → {"x", "f(x)"}]
```

Out[ ]=



Let us linearize about the point $x_o = -1$. So the first order approximation of the function is

$$f(x) \approx f(x_o) + \frac{\partial f(x_o)}{\partial x}(x - x_o) \qquad \textbf{\textit{(Eq.1)}}$$

```
In[•]:= xo = -1;
        dfdx[x_] = D[f[x], x];
        ffirstorder[x_] = f[xo] + dfdx[xo] (x - xo)

Out[•]= 12 - 6 × (1 + x)
```

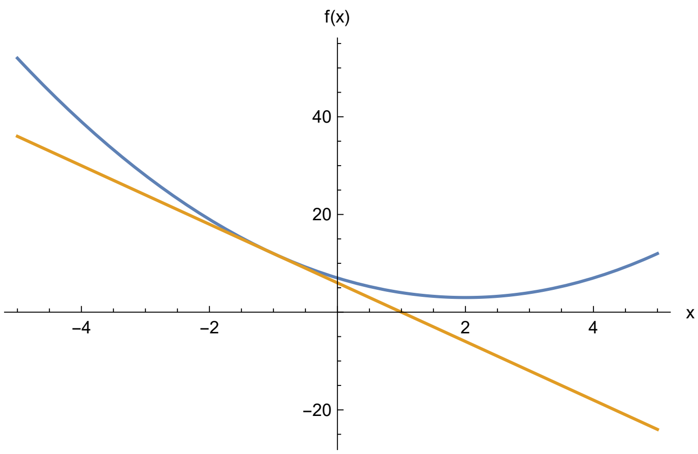
So we have

$$f_{\text{first}}(x) = 12 - 6\,(x + 1)$$

Plotting this and the original function yields

```
In[•]:= Plot[{f[x], ffirstorder[x]}, {x, -5, 5},
         AxesLabel → {"x", "f(x)"}]
```

Out[•]=



As can be seen, this is a decent approximation of the function and it matches the function exactly at the point we are linearizing about.

As discussed previously, this is a first order or affine approximation of the function. It is not linear in the mathematical sense as the function does not pass through the origin and will therefore fail the linearity test. If we would like generate a linear approximation, we can simply shift the origin and instead plot deviations from the Taylor series expansion point. In other words, we move the origin to the point $(x_o, f(x_o))$ and instead look at differences from this point.

$$\Delta\text{f}(x) = f(x) - f(x_o)$$
$$\Delta\, x = x - x_o$$

We can use these relationships to rewrite Eq.1 as

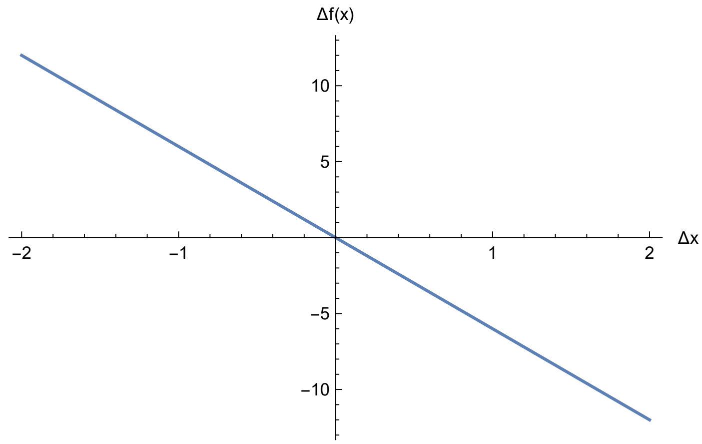
$$\Delta\, f(x) \approx \frac{\partial f(x_o)}{\partial x}\,\Delta\, x$$

$$= (2\,x_o - 4)\,\Delta\, x$$

$$= -6\,\Delta\, x$$

We have to be careful about this notation as this really shows the perturbations from the trim point. If we were to plot this, we need to be very careful of how we label the axes

In[ ]:=
```
Plot[dfdx[xo] Δx , {Δx, -2, 2},
 AxesLabel → {"Δx", "Δf(x)"}]
```

Out[ ]=


This is a measure of how "sensitive" the function is at the trim point. For example, if we had chosen linearization trim point as $x_o = 2$

In[ ]:=
```
D[f[x], x] /. {x → 2}
```

Out[ ]= 0

We see that the function is not sensitive at all to changes in x at this point

Usage of Series To Evaluate Integrals (OPTIONAL)

One reason that Taylor series are important is that they enable us to integrate functions that we couldn't previously handle. In fact, Newton often integrated function by first expressing them as a power series and then integrating the series term by term.

Example: Integrating a Power Series

Consider $f(x) = e^{-x^2}$. Compute the integrals

$$\int e^{-x^2} dx$$

and

$$\int_0^1 e^{-x^2} dx$$

Solution:

We note that this cannot be integrated by traditional techniques because its anti-derivative is not an elementary function. When we try this with Mathematica, we obtain

In[ ]:=
```
Integrate[Exp[-x^2], x]
```

Out[ ]= $\frac{1}{2}\sqrt{\pi}\,\mathrm{Erf}[x]$

Recall that the error function is defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt$$

effectively showing us that Mathematica does not have a closed form expression for the integral.

However, we can use the Maclaurin series for $f(x) = e^{-x^2}$ to evaluate the integral.

Although we could use the direct method to calculate the Maclaurin series, we recall from our previous example that the Maclaurin series for e^x is given by

$$e^x = \sum_{n=0}^{\infty}\frac{x^n}{n!}$$

Therefore, replacing x with $-x^2$ yields

$$e^{-x^2} = \sum_{n=0}^{\infty}\frac{\left(-x^2\right)^n}{n!}$$

$$e^{-x^2} = \sum_{n=0}^{\infty}(-1)^n\frac{x^{2n}}{n!} \qquad \textbf{\textit{(Eq.2.1)}}$$

So for the indefinite integral, we can write

$$\int e^{-x^2}\,dx = \int\sum_{n=0}^{\infty}(-1)^n\frac{x^{2n}}{n!}\,dx$$

$$= \int\left(1 - \frac{x^2}{1!} + \frac{x^4}{2!} - \frac{x^6}{3!}\ \ldots + (-1)^n\frac{x^{2n}}{n!} + \ldots\right)dx$$

$$\int e^{-x^2}\,dx = C + x - \frac{x^3}{3\cdot 1!} + \frac{x^5}{5\cdot 2!} - \frac{x^7}{7\cdot 3!} + \ldots + (-1)^n\frac{x^{2n+1}}{(2n+1)\,n!} + \ldots$$

And now for the definite integral

$$\int_0^1 e^{-x^2}\,dx = \left[C + x - \frac{x^3}{3\cdot 1!} + \frac{x^5}{5\cdot 2!} - \frac{x^7}{7\cdot 3!} + \ldots + (-1)^n\frac{x^{2n+1}}{(2n+1)\,n!} + \ldots\right]\Bigg|_{x=0}^{x=1}$$

$$= 1 - \frac{1^3}{3\cdot 1!} + \frac{1^5}{5\cdot 2!} - \frac{1^7}{7\cdot 3!} + \frac{1^9}{9\cdot 4!} - \ldots$$

If we truncate this after this many terms, we obtain an approximate value of the definite integral

$$\int_0^1 e^{-x^2}\,dx \approx 1 - \frac{1^3}{3\cdot 1!} + \frac{1^5}{5\cdot 2!} - \frac{1^7}{7\cdot 3!} + \frac{1^9}{9\cdot 4!}$$

$$= 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42} + \frac{1}{216}$$

$$\int_0^1 e^{-x^2} \, dx \approx 0.7475$$

In[ ]:=
```
1 - 1/(3*1!) + 1/(5*2!) - 1/(7*3!) + 1/(9*4!) // N
```
Out[ ]= 0.747487

We can compare with a numerical solution using the Erf function

In[ ]:=
```
Integrate[Exp[-x^2], {x, 0, 1}]
% // N
```
Out[ ]= $\frac{1}{2}\sqrt{\pi}$ Erf[1]

Out[ ]= 0.746824

Taylor Series in Several Variables

Previously, we considered the function $f : \mathbb{R}^1 \to \mathbb{R}^1$. However, this can be generalize to the situation where f is a function of d variables. That is $f : \mathbb{R}^n \to \mathbb{R}^1$.

$$f = f(x_1, x_2, \ldots, x_n)$$

In this case, the Taylor series of f expanded about a point $a_1, a_2, \ldots, a_d$ is given by

$$f(x_1, x_2, \ldots, x_n) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \cdots \sum_{n_n=0}^{\infty} \frac{(x_1-a_1)^{n_1} (x_2-a_2)^{n_2} \ldots (x_n-a_n)^{n_n}}{n_1! \, n_2! \ldots n_n!} \left(\frac{\partial^{n_1+n_2+\ldots+n_n} f(a_1,a_2,\ldots,a_n)}{\partial x_1^{n_1} \, \partial x_2^{n_2} \ldots \partial x_n^{n_n}} \right) \qquad \textbf{\textit{(Eq.3.1)}}$$

We can write out several terms in this expression

constant term (akin to $f(a)$) $= f(a_1, a_2, \ldots, a_n)$

1st order term (akin to $\frac{f'(a)}{1!}(x-a)$) $= \sum_{j=1}^{n} \frac{\partial f(a_1,a_2,\ldots,a_n)}{\partial x_j} \left(x_j - a_j\right)$

2nd order term (akin to $\frac{f''(a)}{2!}(x-a)^2$) $= \frac{1}{2!} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 f(a_1,a_2,\ldots,a_n)}{\partial x_j \, \partial x_k} \left(x_j - a_j\right)(x_k - a_k)$

3rd order term (akin to $\frac{f'''(a)}{3!}(x-a)^3$) $= \frac{1}{3!} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{\partial^3 f(a_1,a_2,\ldots,a_n)}{\partial x_j \, \partial x_k \, \partial x_l} \left(x_j - a_j\right)(x_k - a_k)(x_l - a_l)$

$\vdots$

Compact Vector Notation

Using more compact vector notation, we can define vectors x and a as

$x = (\; x_1 \quad x_2 \quad \dots \quad x_n \;)^T$
$a = (\; a_1 \quad a_2 \quad \dots \quad a_n \;)^T$

Then the first order approximation of the function can be written as

$$f(x) \approx f(a) + \nabla f(a)^T \cdot (x - a) \qquad \textbf{(Eq.3.2)}$$

where $\quad \nabla f(a)$ = gradient of f evaluated at a

Then the second order approximation of the function can be written as

$$f(x) \approx f(a) + \nabla f(a)^T \cdot (x - a) + \frac{1}{2!}(x - a)^T \cdot \nabla^2 f(a) \cdot (x - a) \qquad \textbf{(Eq.3.3)}$$

where $\quad \nabla^2 f(a)$ = Hessian matrix evaluated at a

The Hessian matrix corresponding to f is an n – by – n matrix of all second partial derivatives of f. Its elements are given by

$$H(f)_{\text{ij}} = \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} f$$

Or using the ∇^2 notation (nabla-squared or del-squared)

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} \qquad \textbf{(Eq.3.3)}$$

Notation may be confusing at points as some authors use $\nabla^2 f$ to denote the Laplacian of f (recall that $\nabla^2 f = \Delta f = \frac{\partial f^2}{\partial x^2} + \frac{\partial f^2}{\partial y^2} + \frac{\partial f^2}{\partial z^2}$). It is therefore important to consider the context in which the term $\nabla^2 f$ is used.

It is important to note that this matrix is symmetric because the order of the partial derivatives do not matter

$$H(f)_{\text{ij}} = \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} f \qquad \text{same as} \quad H(f)_{\text{ji}} = \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_i} f$$

Example: 2D Function

Consider the function

$$f(x_1, x_2) = e^{x_1} \ln(1 + x_2)$$

Compute

1. The first order approximation of the function at the point $a = \left(\frac{3}{2}, 1\right)$
2. The second order approximation of the function at the point $a = \left(\frac{3}{2}, 1\right)$

Solution 1: First Order Approximation

We can first define the function

In[ ]:=
```
f[x1_, x2_] = Exp[x1] Log[1 + x2];
```

We can compute the gradient

In[ ]:=
```
gradF[x1_, x2_] = ( D[f[x1, x2], x1]
                    D[f[x1, x2], x2] );
gradF[x1, x2] // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} e^{x1} \text{Log}[1 + x2] \\ \frac{e^{x1}}{1+x2} \end{pmatrix}$$

We can define some other intermediate variables

In[ ]:=
```
(*Define the point we are interested in*)
a1 = 3 / 2;
a2 = 1;
a = ( a1
      a2 );

(*define the vector x*)
x = ( x1
      x2 );
```

So the first order approximation of the function is given by

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

In[ ]:=
```
temp = f[a1, a2] + Transpose[gradF[a1, a2]].(x - a);
fFirstOrder[x1_, x2_] = temp[[1, 1]]
```

Out[ ]= $\frac{1}{2} e^{3/2} (-1 + x2) + e^{3/2} \text{Log}[2] + e^{3/2} \left(-\frac{3}{2} + x1\right) \text{Log}[2]$

So we see that the first order approximation is given by

$$f_{\text{first}}(x_1, x_2) = \frac{1}{2} e^{3/2}(x_2 - 1) + e^{3/2} \ln(2) + e^{3/2}\left(x_1 - \frac{3}{2}\right) \ln(2)$$

We can show that this approximation is exact at the point a

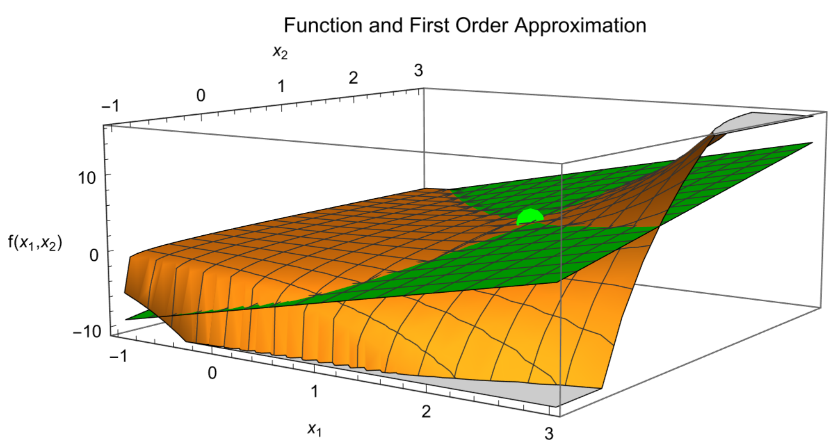
In[ ]:=
```
fFirstOrder[a1, a2] == f[a1, a2]
```

Out[ ]= True

We can plot this first order approximation along with the original function

In[]:=
```
(*Plot options*)
x1Min = -1;
x1Max = 3;

x2Min = -1;
x2Max = 3;

(*Plot this on the surface*)
Show[
 (*Plot the function f(x1,x2)*)
 Plot3D[f[x1, x2], {x1, x1Min, x1Max}, {x2, x2Min, x2Max}],

 (*Plot the point a*)
 Graphics3D[
  {
   AbsolutePointSize[15], Green, Point[{a1, a2, f[a1, a2]}]
  }
 ],

 (*Plot the first order approximation*)
 Plot3D[fFirstOrder[x1, x2], {x1, x1Min, x1Max},
  {x2, x2Min, x2Max}, PlotStyle → Directive[Green]],

 (*Plot options*)
 AxesLabel → {"x₁", "x₂", "f(x₁,x₂)"},
 PlotLabel → "Function and First Order Approximation"
]
```

Out[]=



So we see that the first order approximation is the tangent plane to the surface at the point *a*.

<u>Solution 2: Second Order Approximation</u>

We can compute the Hessian matrix

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

```
In[]:= H[x1_, x2_] = ( D[f[x1, x2], x1, x1]  D[f[x1, x2], x1, x2]
                       D[f[x1, x2], x2, x1]  D[f[x1, x2], x2, x2] );
        H[x1, x2] // MatrixForm
```

Out[]//MatrixForm=

$$\begin{pmatrix} e^{x1}\,\mathrm{Log}[1+x2] & \frac{e^{x1}}{1+x2} \\ \frac{e^{x1}}{1+x2} & -\frac{e^{x1}}{(1+x2)^2} \end{pmatrix}$$

Note that this is symmetric as we discussed previously.

So the second order approximation of the function is given by

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

```
In[]:= temp = f[a1, a2] + Transpose[gradF[a1, a2]].(x - a) + 1/2! * Transpose[x - a].H[a1, a2].(x - a);
        fSecondOrder[x1_, x2_] = Collect[Expand[temp[[1, 1]]], {x1, x1^2, x2, x2^2}]
```

Out[]= $\frac{e^{3/2}}{8} - \frac{1}{8} e^{3/2}\, x2^2 + \frac{5}{8} e^{3/2}\,\mathrm{Log}[2] + \frac{1}{2} e^{3/2}\, x1^2\,\mathrm{Log}[2] + x1\left(-\frac{e^{3/2}}{2} + \frac{1}{2} e^{3/2}\, x2 - \frac{1}{2} e^{3/2}\,\mathrm{Log}[2]\right)$

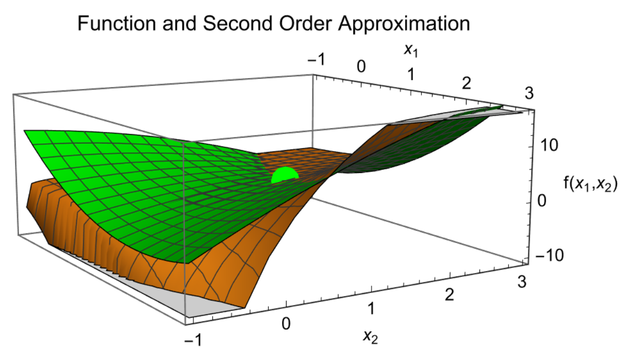
Once again, we can show that this approximation is exact at the point a

```
In[]:= fSecondOrder[a1, a2] == f[a1, a2]
Out[]= True
```

We can plot this first order approximation along with the original function

In[ ]:=
```
(*Plot this on the surface*)
Show[
 (*Plot the function f(x1,x2)*)
 Plot3D[f[x1, x2], {x1, x1Min, x1Max}, {x2, x2Min, x2Max}],

 (*Plot the point a*)
 Graphics3D[
  {
   AbsolutePointSize[15], Green, Point[{a1, a2, f[a1, a2]}]
  }
 ],

 (*Plot the first order approximation*)
 Plot3D[fSecondOrder[x1, x2], {x1, x1Min, x1Max},
  {x2, x2Min, x2Max}, PlotStyle → Directive[Green]],

 (*Plot options*)
 AxesLabel → {"x1", "x2", "f(x1,x2)"},
 PlotLabel → "Function and Second Order Approximation"
]
```

Out[ ]=



So we see that the second order approximation is the quadratic surface which fits the original surface at the point *a*

In[ ]:=
```
Clear[x2Max, x2Min, x1Max, x1Min, fSecondOrder, H, fFirstOrder, temp, x, a2, a1, a, gradF]
```