MAT 1206 – Introduction to MATLAB

CHAPTER 04: Applications in Linear algebra and basic calculus

Lesson 2:

Content

- Calculating Limits
- Derivatives of functions
- Finding maxima and minima of curves
- Solving differential equations

Calculating Limits

MATLAB provides the limit command for calculating limits.

Syntax: L = **limit**(f, x, a), where 'f' is the function, 'x' is the variable, and 'a' is the value the variable approaches.

```
syms x
f = (x^3 + 5)/(x^4 + 7);
limit(f, x, 0)
```

```
syms x
f = sin(x) / x;
limit(f, x, 0)
```

```
syms x
f = (x^2 - 4) / (x - 2);
limit(f, x, 2)
```

```
syms x
f = (exp(x) - 1 - x) / x^2;
L = limit(f, x, 0);
disp(L);
```

Cont.

```
syms x
f = (exp(x) - 1 - x) / x^2;
L = limit(f, x, 0);
disp(L);
```

```
syms z
f = (2*z^2 - 17*z + 8)/(8 - z);
L = limit(f, z, 8);
disp(L);
```

```
syms x m n
f = (1-cos(m*x))/(1-cos(n*x));
L = limit(f, x, 0);
disp(L);
```

Cont.

Infinite Limits:

MATLAB can also calculate limits that tend to infinity or negative infinity. To compute these limits, we can use the 'inf' or '-inf' values as the limit.

```
syms x
f = (x^2 + 3*x + 2) / (2*x^2 - 5*x + 1);
L = limit(f, x, inf);
disp(L);
```

```
syms x
f = (3*x^3 + 2*x^2 + x) / (4*x^3 - 5*x^2 + 3);
L = limit(f, x, -inf);
disp(L);
```

```
syms x
f = (log(x + 1) - log(x)) / x;
L = limit(f, x, inf);
disp(L);
```

```
syms t
f = nthroot(t,3)+12*t-2*t^2;
L = limit(f, t, -inf);
disp(L);
```

Derivatives of functions

MATLAB provides the diff command for computing symbolic derivatives. In its simplest form, you pass the function you want to differentiate to **diff** command as an argument.

```
syms t
f = 3*t^2 + 2*t^(-2);
diff(f)
```

```
syms x
f = sin(x^2);
diff(f)
```

```
syms x
f = (x^2 - 2*x + 1)*(3*x^3 - 5*x^2 + 2);
diff(f)
```

```
syms x
f = sin(2*x) / sqrt(1 + tan(x/2)^2);
diff(f)
```

Cont.

Computing Higher Order Derivatives:

diff(f,n) computes the nth derivative of f

```
syms x
f = sin(x);
diff(f,2)
```

```
syms x
f = x^4;
diff(f,3)
```

```
syms x
f = sin(x) + x^2 + exp(x) + log(x) + cos(x)^2;
diff(f,2)
```

Cont.

Computing partial derivatives:

```
syms x y
f = x^2 * sin(y) + y^3 * cos(x); % Define the function with two variables
df_dx = diff(f, x) % Compute the partial derivative with respect to x
df_dy = diff(f, y) % Compute the partial derivative with respect to y


syms x y z
f = x^2 * y * sin(z) + exp(x*y*z) - log(x*y*z); % Define the function with three variables
df_dx = diff(f, x) % Compute the partial derivative with respect to x
df_dy = diff(f, y) % Compute the partial derivative with respect to y
df_dz = diff(f, z) % Compute the partial derivative with respect to z
```

Cont.

Computing higher order partial derivatives:

```
syms x y
f = x^3 * y^2 + sin(x*y); % Define the function with two variables
d2f_dx2 = diff(f, x, 2) % Compute the second partial derivative with respect to x
d2f_dy2 = diff(f, y, 2) % Compute the second partial derivative with respect to y
d2f_dxdy = diff(diff(f, x), y) % Compute the mixed partial derivative
```

```
syms x y z
f = x^2 * y * sin(z) + exp(x*y*z) - log(x*y*z); % Define the function with three variables
d3f_dx3 = diff(f, x, 3); % Compute the third partial derivative with respect to x
d3f_dydz2 = diff(diff(diff(f, y), z), z); % Compute the mixed partial derivative with respect to y and two z
```

Finding maxima and minima of curves

If we are searching for the local **maxima** and **minima** for a graph, we are basically looking for the highest or lowest points on the graph of the function for a particular range of values of the symbolic variable.

For a function y = f(x) the points on the graph where the graph has zero slope are called **stationary points**. In other words stationary points are where **f'(x) = 0**.

To find the stationary points of a function, the derivative of the function equate to zero and solve the equation.

Cont.

Example 1: Find the points of maxima and minima of a function: $y = 2x^3 + 3x^2 - 12x + 17$

```
syms x
y = 2*x^3 + 3*x^2 - 12*x + 17; % defining the function
df = diff(y, x) % Compute the first derivative of f(x)
```

df = 6*x^2 + 6*x – 12

```
critical_points = solve(df == 0) % Solve for the critical points where df/dx = 0
```

critical_points =

-2

1

Cont.

```
% Evaluate the second derivative at the critical points
second_derivative = diff(df, x)

        second_derivative =
         12*x + 6

% Evaluate the second derivative at the critical points
function_values = subs(second_derivative, x, critical_points)

        function_values =
        -18
         18
```

We can substitute a value in a symbolic function by using the **subs** command.

fnew = **subs(f,old,new)** returns a copy of f, replacing all occurrences of old with new, and then evaluates s.

Cont.

Minimum and maximum values on the function are -2 and 1.

The maximum and minimum function values:

```
Max=subs(y,x,-2)
Min=subs(y,x,1)
```

```
Max =
 37

 Min =
 10
```

Cont.

Exercise:

Find the points of maxima and minima of a function: $y = 2x^3 - 3x^2 + 6$

Questions/queries?