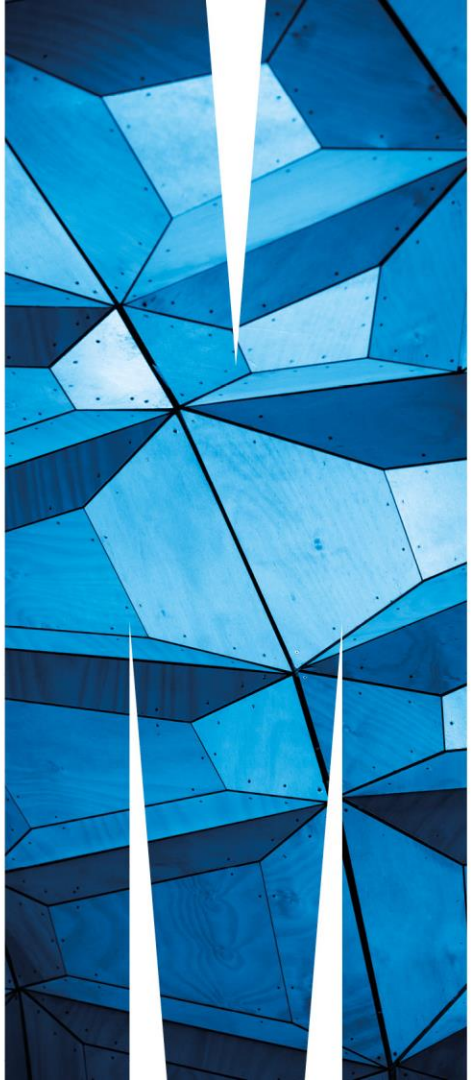
MONASH University

MONASH
ENGINEERING
ENG1060



FUNCTIONS: MULTIPLE OUTPUTS

Presented by Tony Vo

Slides by Tony Vo

GROUP OF EIGHT AUSTRALIA

RECALL: HOW TO CREATE A FUNCTION

- Start a new m-file (script file)
  - Declare it as a function with a function header

- The function declaration

  function outputs = function_name(inputs)

SineTaylor.m

```
1   function taylor_result = SineTaylor(x,n)
2   taylor_result = (-1)^n / factorial(2*n + 1) * x.^(2*n + 1);
```

RECALL: HOW TO CALL A FUNCTION

- Do not run a function
  - Clicking run on a function does not work because no inputs have been provided
  - Instead, call a function using a separate/complimentary m-file

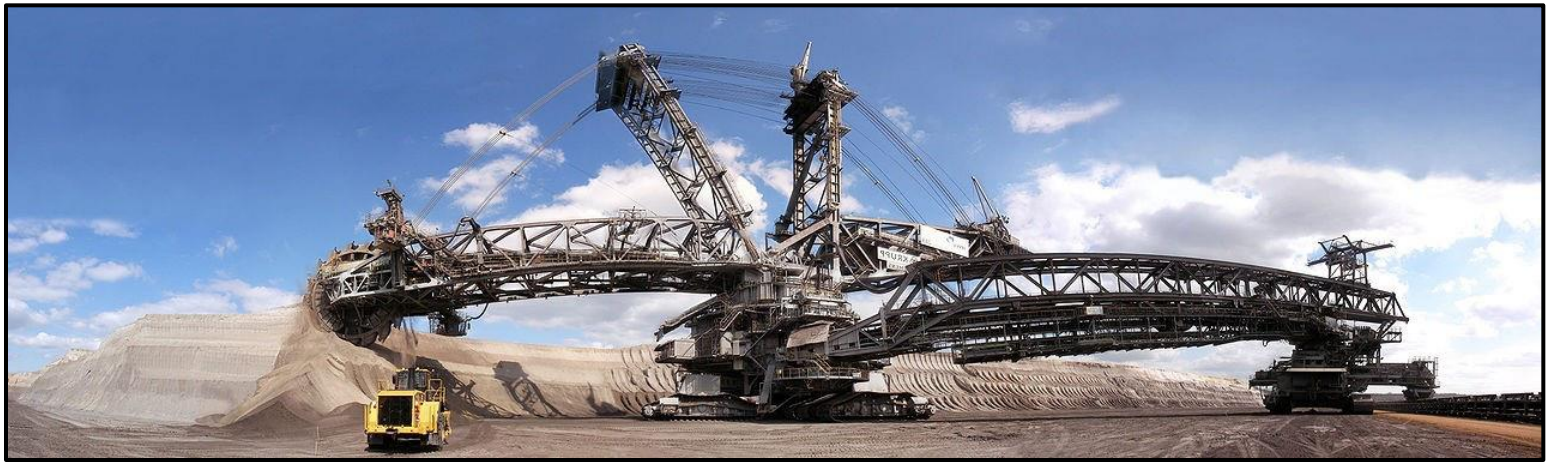
SineTaylor.m

```
function taylor_result = SineTaylor(x,n)
% taylor_result = SineTaylor(x,n)
% Written by Tony Vo
% Created on 12/07/2015
%
% Input Arguments
% ----
% x - input values for Taylor term (can be a vector)
% n - n value for Taylor term (must be a scalar)
% Outputs
% ----
% taylor_result - taylor series term values (can be vector)
taylor_result = (-1)^n / factorial(2*n + 1) * x.^(2*n + 1);

```

```
x = -pi:0.1:pi;

% n= 0
sin_n0 = SineTaylor(x, 0)
% n= 1
sin_n1 = SineTaylor(x, 1)
% n= 2
sin_n2 = SineTaylor(x, 2)
% n= 3
sin_n3 = SineTaylor(x, 3)
```

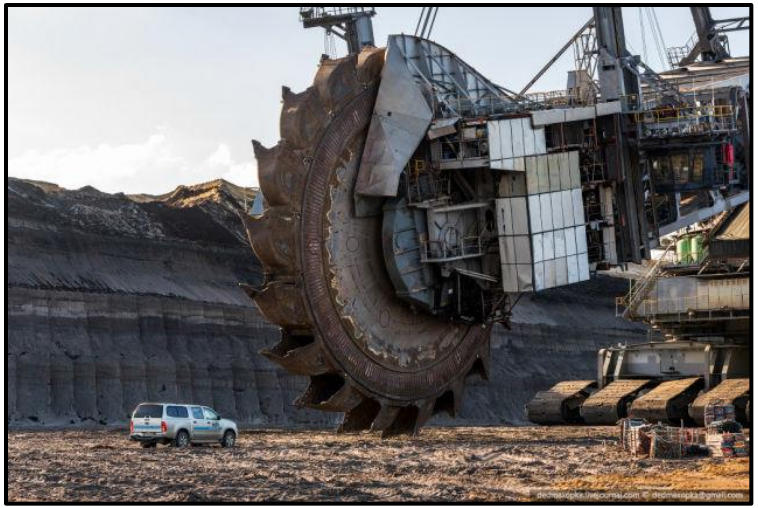
BAGGER 293

- Rotating buckets dig material from the earth and moves it along the conveyer belt
  - Located in Germany



BAGGER 293

- 96 metres tall and 225 metres long at 14,200 tonnes
- Operated by 5 people at one time
- Can move 240,000 m^3 of earth per day



BAGGER 293

- You work for a mining company that uses the Bagger 293
  - The company wants to know the risks posed to workers if debris falls from the conveyer belt
  - The height of the conveyer belt changes day-to-day

- Programming task
  - Code a function that calculates the time taken for the debris to hit the ground and the velocity when it hits the ground
  - Function must work with vector inputs

THE MATHS

- Step 1: Describe the problem using mathematics

$$t = \sqrt{\frac{2h}{g}} \qquad v = gt$$

  - t is the falling time
  - h is the height of the conveyer belt
  - g is the acceleration due to gravity
  - v is the velocity of the debris as it hits the ground

THE INPUTS AND OUTPUTS

- Step 2: Determine the input arguments and outputs of the MATLAB function

$$t = \sqrt{\frac{2h}{g}} \qquad v = gt$$

  - v is an ???
  - t is an ???
  - h is an ???
  - g is an ???

THE FUNCTION

- Step 3: Write the MATLAB function
  - Start with the function header
  - Multiple outputs need to be in a vector (square brackets)

DebrisDrop.m

```
1   function [drop_time, drop_velocity] = DebrisDrop(conveyer_height)
```

THE FUNCTION

- Step 3: Write the MATLAB function
  - Aim of function is to calculate the outputs

DebrisDrop.m

```
1  function [drop_time, drop_velocity] = DebrisDrop(conveyer_height)
2  % [drop_time, drop_velocity] = DebrisDrop(conveyer_height)
3  % Written by Wai Ho Li, ID: 12345678
4  %
5  % Calculates the time it takes for debris that dropping from
6  % the RB293 conveyer belt to hit the ground. Also returns the
7  % velocity of the debris at the time of impact with the ground.
8  % ** All units are metric (metres, seconds etc)
9  %
10 % Input Arguments
11 % ---
12 % conveyer_height - Height of the conveyer belt (m)
13 %
14 % Return Value
15 % ---
16 % drop_time - Drop time of debris from belt to ground (s)
17 % drop_velocity - Velocity of debris when it hits the ground (m/s)
18 g = 9.81; % Acceleration due to gravity on earth (m/s^2)
19 drop_time = sqrt(2 * conveyer_height / g); % t = sqrt(2d/g)
20 drop_velocity = conveyer_height ./ drop_time; % v = d / t
```

THE INPUTS AND OUTPUTS

MONASH University

DebrisDrop.m

```
1   function [drop_time, drop_velocity] = DebrisDrop(conveyer_height)
```

- Questions
  1. Calculate the fall time and velocity of debris with a conveyer height at 50m
  2. Repeat question 1 for conveyer heights of 10, 15, 20 … 90m
  3. What happens if this command is executed? X = DebrisDrop(B)

USING THE FUNCTION

1. Calculate the fall time and velocity of debris with a conveyer height at 50m

DebrisDrop.m

```
1   function [drop_time, drop_velocity] = DebrisDrop(conveyer_height)
```

DebrisDrop.m | Debris_example.m

```
1   % Written by: Tony Vo
2   % Created on 13/07/2015
3   % Demonstrating that the variable name does not need to
4   % match the variable name used in the function
5   % [drop_time, drop_velocity] = DebrisDrop(conveyer_height)
6   clear all; close all; clc;
7
8   % two outputs
9   % blah variable to demonstrate that variable names
10  % do not need to match those in function
11  blah = 50;
12  [drop_time, drop_velocity] = DebrisDrop(blah);
```

Workspace

Name	Value
blah	50
drop_time	3.1928
drop_velocity	15.6605

USING THE FUNCTION

MONASH University

2. Repeat question 1 for conveyer heights of 10, 15, 20 … 90m

DebrisDrop.m

```
1   function [drop_time, drop_velocity] = DebrisDrop(conveyer_height)
18  g = 9.81; % Acceleration due to gravity on earth (m/s^2)
19  drop_time = sqrt(2 * conveyer_height / g); % t = sqrt(2d/g)
20  drop_velocity = conveyer_height ./ drop_time; % v = d / t
```

```
% output variables can also be different to those defined
% in the function
blah2 = 10:5:90;
[A, B] = DebrisDrop(blah2);
```

Workspace

Name	Value
A	*1x17 double*
B	*1x17 double*
blah2	*1x17 double*

USING THE FUNCTION

3. What happens if this command is executed? X = DebrisDrop(B)

DebrisDrop.m

```
1   function [drop_time, drop_velocity] = DebrisDrop(conveyer_height)
```

```
% problem with specifying on one output
B = 80;
X = DebrisDrop(B);
```

Workspace

Name	Value
B	80
X	4.0386

- If only one output is specified, then only the first one is provided
  - Good practice is to copy entire function header without the word "function" when calling functions

SUMMARY

- Create a function with multiple outputs
- Call a function with multiple outputs

- Is it possible to obtain only the second output, without assigning a variable to the first output?