# Lab 03: Functions and Control Flows

Melissa Butler

September 06, 2021

## 1 Output file

```
lab_03_script
rowSums =
  <a href="matlab:helpPopup function_handle" style="font-weight:bold">function_handle</a> with value:
    @(mat)sum(mat,2)
magicMat5 =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
magicMat7 =
    30    39    48     1    10    19    28
    38    47     7     9    18    27    29
    46     6     8    17    26    35    37
     5    14    16    25    34    36    45
    13    15    24    33    42    44     4
    21    23    32    41    43     3    12
    22    31    40    49     2    11    20
magicMat5RowSums =
    65
    65
    65
    65
    65
magicMat7RowSums =
   175
   175
   175
   175
   175
   175
   175
f =
  <a href="matlab:helpPopup function_handle" style="font-weight:bold">function_handle</a> with value:
    @(x)x.*log(x)
g =
  <a href="matlab:helpPopup function_handle" style="font-weight:bold">function_handle</a>
      with value:
    @(y)y.*exp(y)
h =
  <a href="matlab:helpPopup function_handle" style="font-weight:bold">function_handle</a>
      with value:
    @(z)g(f(z))
z =
    1.0000
    1.4000
    1.8000
    2.2000
    2.6000
    3.0000
    3.4000
    3.8000
    4.2000
    4.6000
    5.0000
hz =
   1.0e+04 *
         0
    0.0001
    0.0003
    0.0010
    0.0030
    0.0089
    0.0267
    0.0810
    0.2499
    0.7853
    2.5147
matProd =
  <a href="matlab:helpPopup function_handle" style="font-weight:bold">function_handle</a>
      with value:
    @(A,B)A*B*B'*A'
A =
     1     2     3
     4     5     6
B =
     7    10    13    16
     8    11    14    17
     9    12    15    18
matProdAB =
       25336       62416
       62416      153766
```

```
p =
  <a href="matlab:helpPopup function_handle" style="font-weight:bold">function_handle</a>
      with value:
    @(x)x.^3.*(x<-1)+x.*(-1<=x&x<=1)+x.^2.*(x>1)
 n   factorialRecursive   factorialIterative
 1                    1                    1
 2                    2                    2
 3                    6                    6
 4                   24                   24
 5                  120                  120
 6                  720                  720
 7                 5040                 5040
 8                40320                40320
 9               362880               362880
10              3628800              3628800
11             39916800             39916800
12            479001600            479001600
13           6227020800           6227020800
14          87178291200          87178291200
15        1307674368000        1307674368000
16       20922789888000       20922789888000
17      355687428096000      355687428096000
18     6402373705728000     6402373705728000
19   121645100408832000   121645100408832000
20  2432902008176640000  2432902008176640000
ans =
    'Wednesday'
ans =
    'Saturday'
ans =
    'Monday'
ans =
    'Monday'
ans =
    'Thursday'
ans =
    'Wednesday'
maxBottles1 =
    15
maxBottles2 =
        1069
diary off
```

## 2 Script file

```
% Math 3341, Fall 2021
% Lab 03: Functions and Control Flows
% Author: Melissa Butler
% Date: 09/06/2021

clc             % clear command window
close all       % close figure windows
clear           % clear variables workspace
format compact  % show results in compact format

%% 1 Anonymous Function

% 1(a)
rowSums = @(mat) sum(mat, 2)
magicMat5 = magic(5)
magicMat7 = magic(7)
magicMat5RowSums = rowSums(magicMat5)
magicMat7RowSums = rowSums(magicMat7)

% 1(b)
f = @(x) x .* log(x)
g = @(y) y .* exp(y)
h = @(z) g(f(z))
z = linspace(1, 5, 11)'
hz = h(z)

% 1(c)
matProd = @(A, B) A * B * B' * A'
A = reshape(colon(1, 6), 3, 2)'
B = reshape(colon(7, 18), 3, 4)
matProdAB = matProd(A, B)

% 1(d)
p = @(x) x.^3 .* (x < -1) + x .* (-1 <= x & x <= 1) + x.^2 .* (x > 1)
fplot(p, [-2, 2])
print(gcf, '-dpng', 'lab_03_1d.png')

%% 2 Function Files

% 2(c)
fprintf('%2s %20s %20s\n', 'n', 'factorialRecursive', 'factorialIterative');
for n = 1:20
    f1 = factorialRecursive(n);
    f2 = factorialIterative(n);
    fprintf('%2d %20d %20d\n', n, f1, f2);
end

%% 3 Application: Real-Life Problems

% 3(a)
dayOfWeek(1970, 1, 7)
dayOfWeek(1970, 3, 7)
dayOfWeek(1971, 3, 8)
dayOfWeek(1988, 8, 8)
dayOfWeek(1999, 9, 9)
dayOfWeek(2021, 2, 10)

% 3(b)
maxBottles1 = maxBeverageBottles(10, 2, 4, 2)
maxBottles2 = maxBeverageBottles(1000, 2, 5, 3)
```

# 3 Figure file

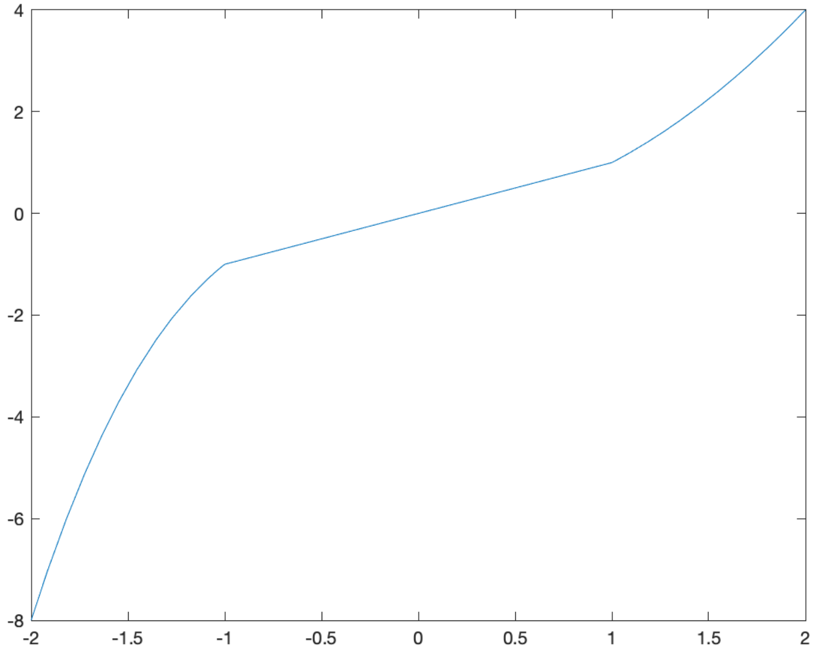

Figure 1: Piecewise Function $p(x)$

# 4 Function files

factorialRecursive.m:

```
function f = factorialRecursive(n)

if n == 0
    f = 1;
else
    f = n * factorialRecursive(n - 1);
end

end
```

factorialIterative.m:

```
function f = factorialIterative(n)

f = 1;
for i = 1:n
    f = f * i;
end

end
```

dayOfWeek.m:

```
function dow = dayOfWeek(year, month, day)

numberOfDays = 0;
for y = 1970:(year - 1)
    if isLeapYear(y)
        numberOfDays = numberOfDays + 366;
    else
        numberOfDays = numberOfDays + 365;
    end
end

for m = 1:(month - 1)
    switch m
        case {1, 3, 5, 7, 8, 10, 12}
            numberOfDays = numberOfDays + 31;
        case {4, 6, 9, 11}
            numberOfDays = numberOfDays + 30;
        case 2
            if isLeapYear(year)
                numberOfDays = numberOfDays + 29;
            else
                numberOfDays = numberOfDays + 28;
            end
    end
end

numberOfDays = mod(numberOfDays + day - 1, 7);

```

```
switch numberOfDays
    case 0
        dow = 'Thursday';
    case 1
        dow = 'Friday';
    case 2
        dow = 'Saturday';
    case 3
        dow = 'Sunday';
    case 4
        dow = 'Monday';
    case 5
        dow = 'Tuesday';
    case 6
        dow = 'Wednesday';
end

end
```

maxBeverageBottles.m:

```
function totalBottles = maxBeverageBottles(money, pricePerNewBottle, capsPerNewBottle,
    emptyBottlesPerNewBottle)

totalBottles = floor(money / pricePerNewBottle);
caps = totalBottles;
emptyBottles = totalBottles;

while caps >= capsPerNewBottle || emptyBottles >= emptyBottlesPerNewBottle
    newBottles = floor(caps / capsPerNewBottle) + floor(emptyBottles / emptyBottlesPerNewBottle);
    caps = mod(caps, capsPerNewBottle) + newBottles;
    emptyBottles = mod(emptyBottles, emptyBottlesPerNewBottle) + newBottles;
    totalBottles = totalBottles + newBottles;
end

end
```