Lecture 4 MATLAB programming
(2)

Dr. Tony Cahill
Objectives

• For loops
• While loops
• Loop speed-up
  – Dot operator
  – Array mask
The For Loop

• The for loop is a loop that executes a block of statements for a specified number of times.

\[ \text{for index} = \text{expr} \]
\[ \text{statement 1} \]
\[ \text{statement 2} \]
\[ \ldots \]
\[ \text{statement n} \]
\[ \text{end} \]

index is the loop variable, expr is the control expression.
The For Loop

• Control expression
  – The control expression is evaluated by MATLAB at the beginning of the loop.
  – Each time a loop is executed, the control variable will be assigned the value in the next column of the control expression.
  – Several typical forms of the control expression
    • first:last (e.g., for ii=1:10)
    • first:increment:last (e.g., for ii=0.5:0.1:1.5)
    • Row vector (e.g., for ii=[5 9 7])
    • Matrix (e.g., for ii=[1 2 3; 4 5 6])
Example: factorial

• Write a program to calculate the factorial:
  
  $N!=1, \; N=0$

  $N!=N\times(N-1)\times(N-2)\times\ldots\times3\times2\times1, \; N>0$

  ```matlab
  n=input('Please enter N:');
n_factorial=1;
for ii=1:n
    n_factorial=n_factorial*ii;
end
disp ([num2str(n),'!=',num2str(n_factorial)]);
```
Good Programming Practice

• Indent the bodies of loops

```latex
for ii=1:n
    n_factorial=n_factorial*ii;
end
```

```latex
for ii=1:n
    n_factorial=n_factorial*ii;
end
```

• Don’t modify the loop variable in the loop

```latex
for ii=1:n
    ....
    ii=5; % WRONG! SHOULD NOT MODIFY ii
    ...
end
```
Exercise

• Modify the factorial code to calculate
  \( s(n) = 1 + 2 + 3 + \ldots + n \)
The While Loop

• The while loop is a block of statements that are repeated indefinitely as long as some condition is satisfied.

```plaintext
while expression
    statement 1
    statement 2
    ...
end
```

The loop will be repeated if the value of the logical expression is true.
Example: statistical analysis

Problem:

Enter from the keyboard an arbitrary number of measurements (>0), and calculate the mean and the standard deviation.

\[
-x = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

\[
s = \sqrt{\frac{N \sum_{i=1}^{N} x_i^2 - \left( \sum_{i=1}^{N} x_i \right)^2}{N(N-1)}}
\]
Algorithm

while done~=true
    Read a number from the input, save in x
    if x>0
        Increase the number of data points by 1
        Calculate sum_x=sum_x+x
        Calculate sum_x2=sum_x2+x**2
    else
        Set done=true
    end
end
One possible solution, but with a bug. Can you find the bug?

```matlab
n=0;sum_x=0;sum_x2=0; % initialize
done=false;
while done~=true % loop to input data
    x=input('Enter a number (0 to finish):');
    n=n+1;
    if x>0 % valid data points
        sum_x=sum_x+x;
        sum_x2=sum_x2+x^2;
    else
        done=true; % need to stop
    end
end
if n<2 % not enough data points % check number of data points
    disp('At least 2 numbers are needed.');
else
    x_bar=sum_x/n; % mean
    std_dev=sqrt((n*sum_x2-sum_x^2)/(n*(n-1)));% standard deviation
    disp(['number of data points:',num2str(n)]); % display results
    disp(['mean=',num2str(x_bar)]);
    disp(['std_dev=',num2str(std_dev)]);
end
```
Break and continue

• Break: stop the current **for** or **while** loop

```plaintext
while done~=true
    x=input('Enter a number (0 to finish):');
    if x>0
        n=n+1;
        sum_x=sum_x+x;
        sum_x2=sum_x2+x^2;
    else
        done=true;
    end
end
```

```plaintext
while true
    x=input('Enter a number (0 to finish):');
    if x>0
        n=n+1;
        sum_x=sum_x+x;
        sum_x2=sum_x2+x^2;
    else
        break;
    end
end
```
Break and continue

- **Continue**: skipping the remaining statements in the loop and move to the top of the loop. In a for loop, the loop control variable will take the next value.

```matlab
for ii=1:5
    if ii==3
        continue;
    end
    fprintf('ii=%d\n',ii);
end
>> test_continue
ii = 1
ii = 2
ii = 4
ii = 5
```
**dot operator**

- Can be used to simplify and speed-up for loops

```matlab
x=0:0.1:10;
n=length(x);
y=zeros(1,n);
for i=1:n
    y(i)=x(i)*2;
end
plot(x,y);
```

```matlab
x=0:0.1:10;
y=x.^2;
plot(x,y);
```
Array mask

• Can be used to simplify and speed-up for loops with if constructs

```matlab
a=[1 2 3; 4 5 6; 7 8 9];
for ii=1:size(a,1)
    for jj=1:size(a,2)
        if a(ii,jj)>5
            a(ii,jj)=sqrt(a(ii,jj));
        end
    end
end
```

```matlab
a=[1 2 3;4 5 6; 7 8 9];
b=a>5;
a(b)=sqrt(a(b));
```
Array mask

\[ a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \]

for ii=1:size(a,1)
    for jj=1:size(a,2)
        if a(ii,jj)>5
            a(ii,jj)=sqrt(a(ii,jj));
        else
            a(ii,jj)=a(ii,jj)^2;
        end
    end
end

\[ a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \]
\[ b = a > 5 \]
\[ a(b) = \sqrt{a(b)} \]
\[ a(\sim b) = a(\sim b)^2 \]
Exercise

Plot the following function:

\[ y(t) = \begin{cases} 
-3t^2 + 5, & t \geq 0 \\
3t^2 + 5, & t < 0 
\end{cases} \quad \text{with } t \text{ in } [-9,9] \]
Study

• Study the two examples in section 4.4
  – Fitting a linear curve through data points (linear regression)
  – Trajectories of a projectile