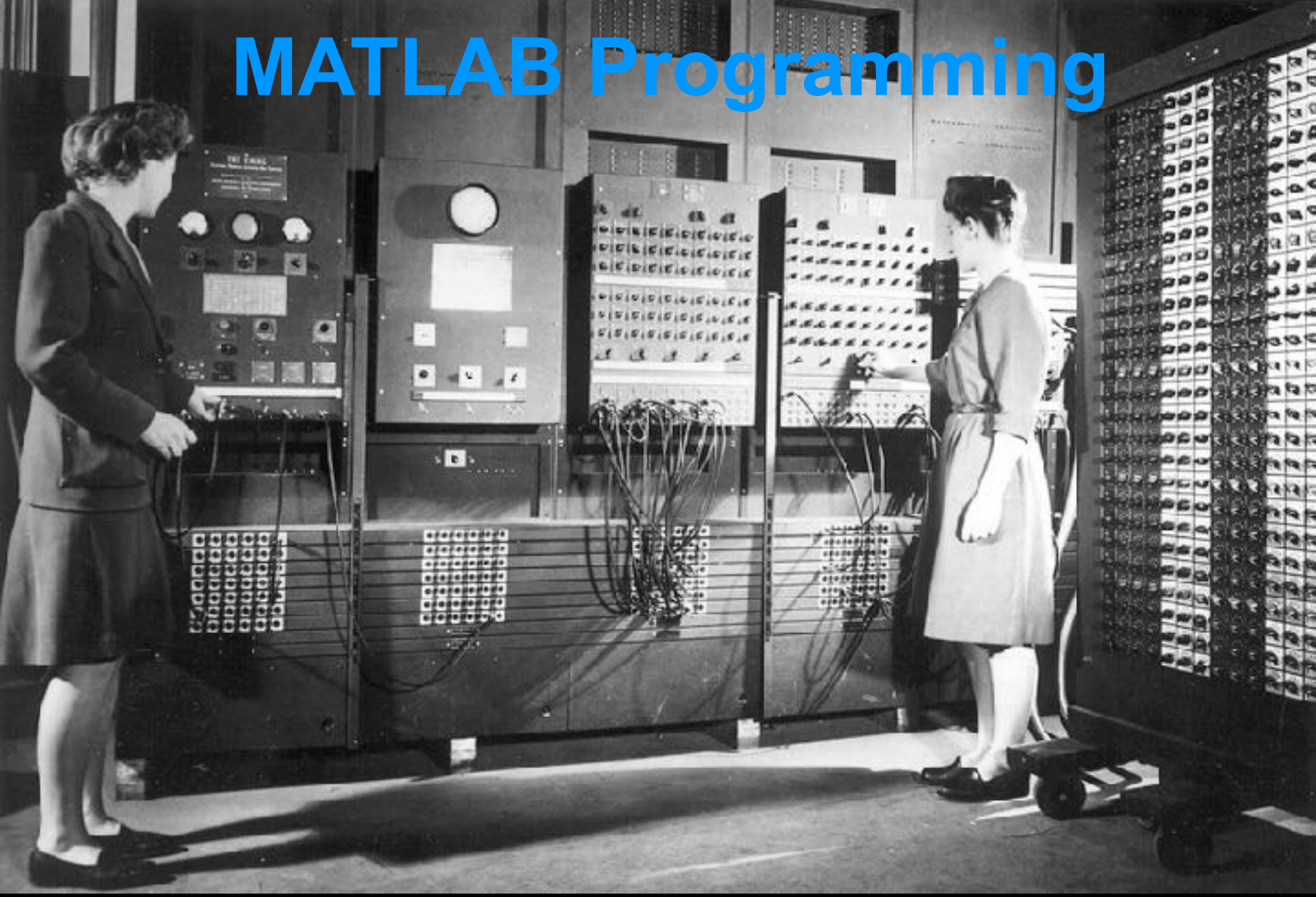
MATLAB Programming



Quiz on Friday

check out file ch02Slides.pdf
on the website

Excluded are:

- Complex numbers,
- Polynomials,
- Multiple plots

More on Script M-Files

- Initializing script M-files
  - ✓ The variables you use and define belong to the Workspace; i.e., they take on any values you assign on earlier in your MATLAB session and persist after the script finished executing.
  - ✓ Type "clear all" in a script M-file to delete all previously defined variables
  - ✓ Type "close all" to close all figure windows
- Echo
  - ✓ The commands in a script M-file do not automatically be displayed in the command window.
  - ✓ Add the command "echo on" and "echo off" to display desired command lines
- Startup M-file
  - ✓ When MATLAB starts, it searches the default path for a script M-file called "startup.m".
  - ✓ If you create such a file, the commands it contains will be run each time MATLAB starts.

More on Function M-Files

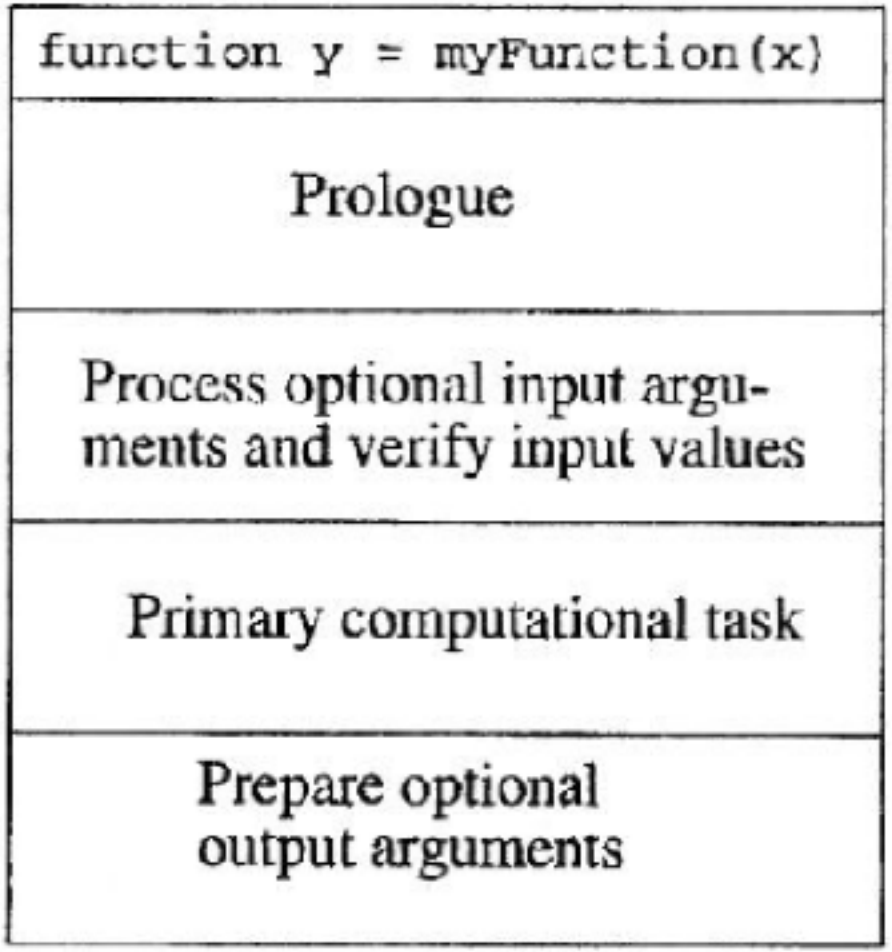
- The variables used in a function M-file are local, meaning that they are unaffected by, and have no effect on, the variables in your Workspace.
- Structure of Function M-Files
  - First line: function definition line: defines the function name, as well as the number and order of input and output arguments
  - The file name (without the .m extension) and the function name should match.
  - Starting from the second line, several comment lines (help text)
  - Command "help function-name" automatically retrieve comment lines following the first line of the function M-file.

Basic Structure of an m-File Function

Function definition

For defensive programming, e.g., check the ranges of input variables

Central part of the function



twosum.m

```
function twosum(x,y)
% twosum  Add two matrices
%         and print the result
x+y
```

threesum.m

```
function s = threesum(x,y,z)
% threesum  Add three variables
%           and return the result
s = x+y+z;
```

addmult.m

```
function [s,p] = addmult(x,y)
% addmult  Compute sum and product
%          of two matrices
s = x+y;
p = x*y;
```

Functions and m-files

Each m-file has only one accessible function!

but can use subfunctions to do its task

```
function y=f(x)
    y=sin(x) + squareplus1(x);

function z=squareplus1(x)
    z = x .* x + 1;
```

MATLAB Programming

- If you are already familiar with another programming language, you can pick up MATLAB programming quickly.

- Main programming structures: functions, flow control (conditions, loops)

- Branching with if

```
if logical expression
  execution;
elseif logical expression
  execution;
else logic expression
  execution;
end
```

Example of IF

signum.m

```
1   function y=signum(x)
2 -     if x>0
3 -       y=1;
4 -     elseif x==0
5 -       y=0
6 -     else
7 -       y= -1
8 -     end
```

Logical Expressions

- Relational operator (e.g., >=,==,~=)
- Type "help relop" for the available relational operators.
- If the inputs to a relational operator are vectors or matrices rather than scalars, then as for arithmetic operations such as + and .*, the operation is done term-by-term and the output is an array of 0 (FALSE) and 1 (TRUE).
- [2 3] < [3 2] gives [1 0]
- This differs from other programming languages, and may cause trouble.

An Example

```
x=[0:pi/4:pi]; % [ 0    0.7854    1.5708    2.3562    3.1416]

if x==0
   y=1;
else
   y=sin(x);
end
```

A Solution: Use Loop

```
x=[0:pi/4:pi];

y=ones(size(x));
for i=1:length(x)
    if x(i)~=0
       y=sin(x);
    end
end
```

Switch statement

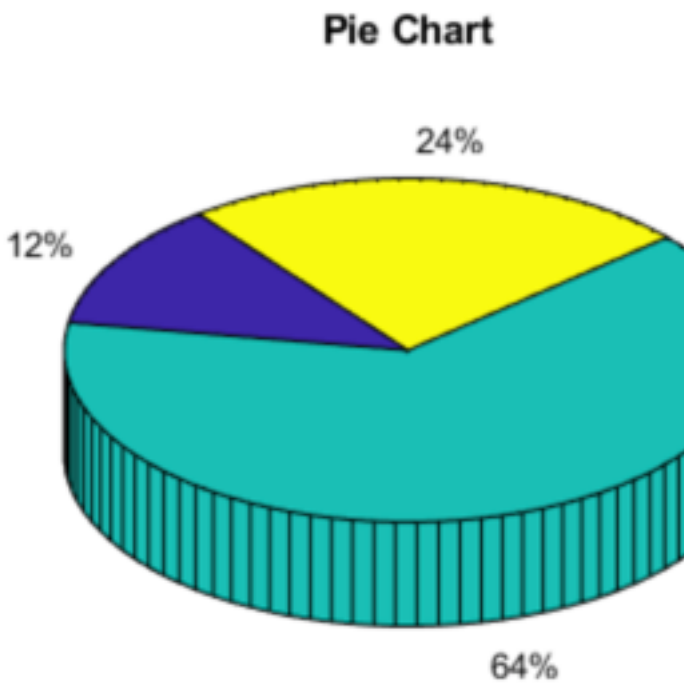
Syntax

```
switch switch_expression
   case case_expression
      statements
   case case_expression
      statements
    ...
   otherwise
      statements
end
```

Examples

```
x = [12 64 24];
plottype = 'pie3';

switch plottype
    case 'bar'
        bar(x)
        title('Bar Graph')
    case {'pie','pie3'}
        pie3(x)
        title('Pie Chart')
    otherwise
        warning('Unexpected plot type. No plot created.')
end
```



For Loop

- A loop specifies that a command or group of commands should be repeated several times.
- for variable = expression

  statement ..., statement

  end
- for R = 1:N

  for C = 1:N

  A(R,C) = 1/(R+C-1);

  end

  end
- Step S with increments of -0.1

  for S = 1.0: -0.1: 0.0

Break vs. Return

break

Use to skip out of current code block and continue with code

```
function z = test()
n = 10;
for i=1:10
  if(i>7)
    z=0;
    break;
  end
  disp(i);
end
disp(done!)
z=1;
```

break

Use to immediate return to the calling function

```
function z = test()
n = 10;
for i=1:10
  if(i>7)
    z=0;
    return;
  end
  disp(i);
end
disp(done!)
z=1;
```

While Statement

Syntax

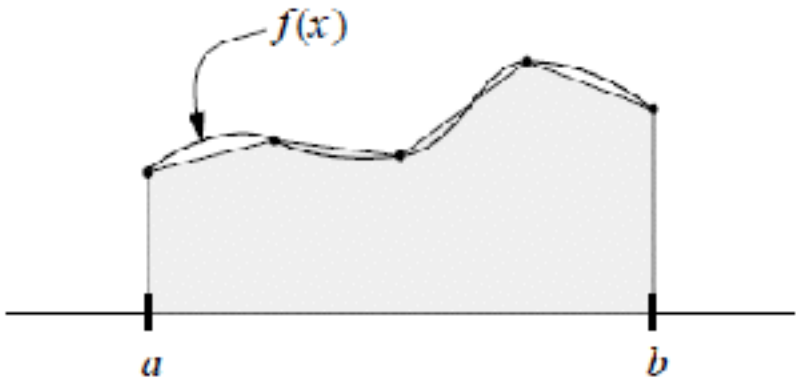
```
while expression
    statements
end
```

```
n = 10;
f = n;
while n > 1
    n = n-1;
    f = f*n;
end
disp(['n! = ' num2str(f)])
```

```
n! = 3628800
```

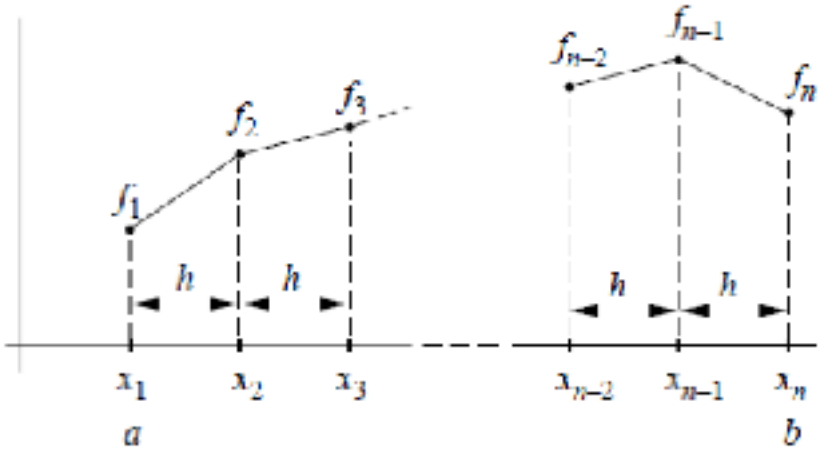
Class Exercise: Numerical Integration

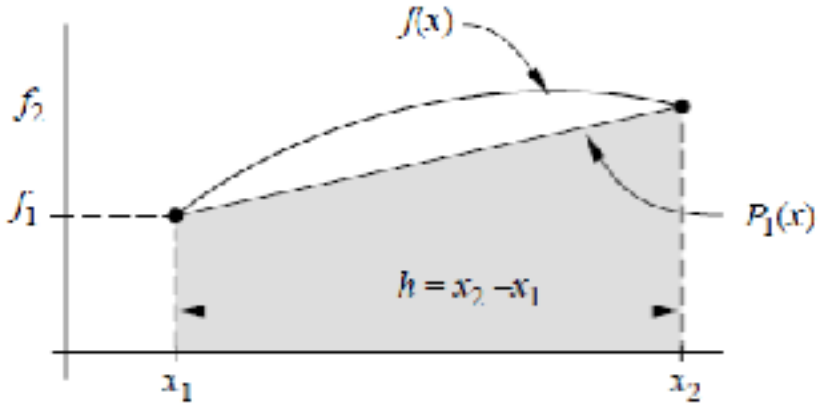
$$\int_a^b f(x)dx$$

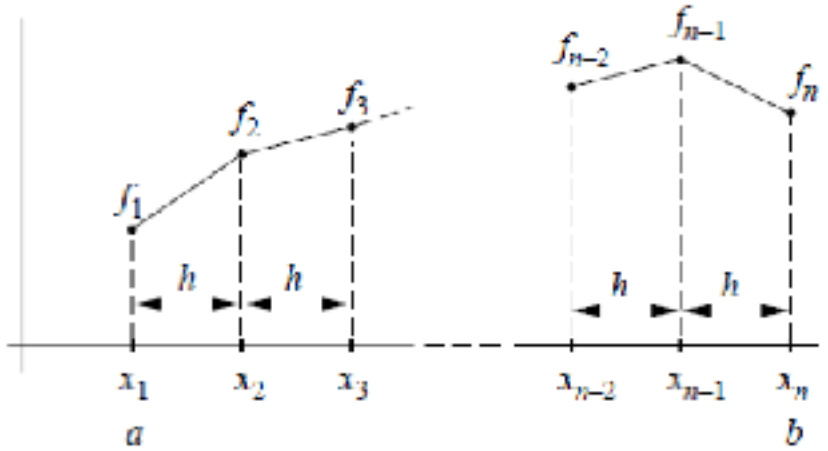


Trapezoidal rule

- Divide the integration interval into a number of panels
- Calculate the area below the function for each panel.
- Sum the areas together.







$$\int_{x_1}^{x_n} f(x)dx \approx h\left[\frac{1}{2}f_1 + f_2 + f_3 + \ldots + f_{n-2} + f_{n-1} + \frac{1}{2}f_n\right]$$

$$\int_0^{2\pi} xe^{-x}dx$$

Quiz on Friday

check out file ch02Slides.pdf

Excluded are:

- Complex numbers,
- Polynomials,
- Multiple plots