Chapter 11

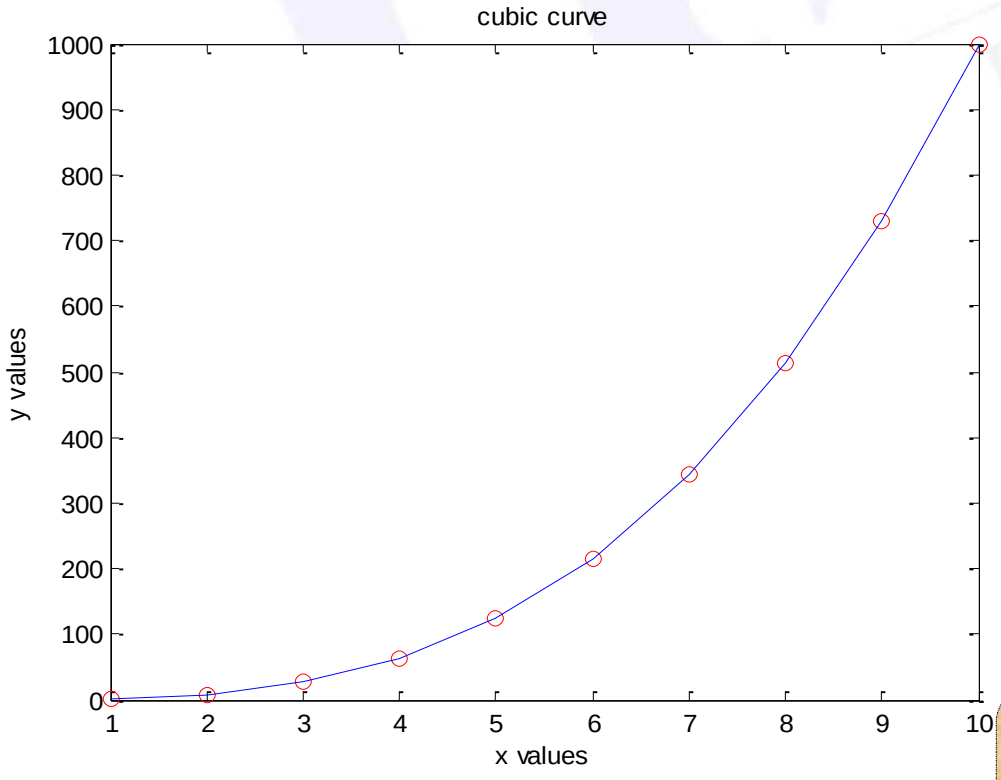
Plotting

من الأفضل أن تمشى ببطء إلى الأمام
من أن تمشى مسرعاً إلى الخلف.

- يستخدم الأمر plot والدوال التابعة له لتمثيل عدد من النقاط بيانياً حسب الأمثلة التالية.

- كالعادة فإن النتائج التي تلي كل أمر تعبر عن معناه ولا حاجة للإسهاب في شرح كل منها.

```
>> x = 1:10
x =
   1   2   3   4   5   6   7   8   9   10

>> y = x.^3
y =
       1   8   27   64   125   216   343   512   729   1000

>> plot  (x, y, 'ro', x, y, 'b-')          % 'r'  = red,  'o' = circle

>> xlabel  ('x values')

>> ylabel  ('y values')

>> title   ('cubic curve')
```

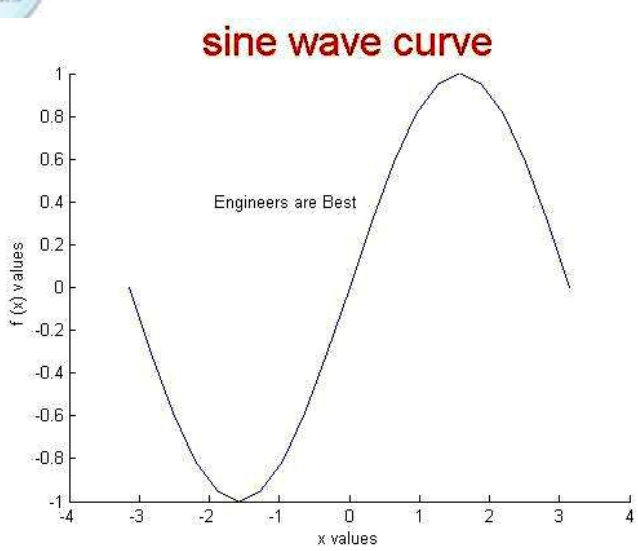


الشـــخص الوحيـــد الـــذي سيستمر معك طوال حياتك هو أنت . . . فلا تُحمّلـه ما لا يطيق من القلق والتردد.

ـ تابع استخدام الأمر plot والدوال التابعة له بالمثال التالي.

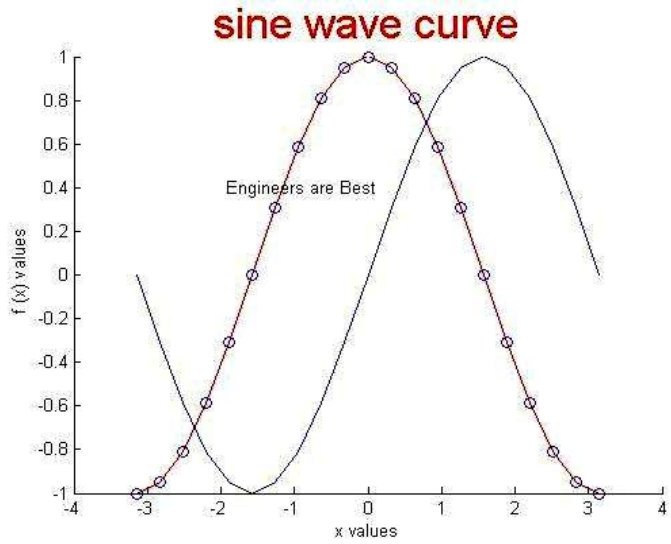
```
>> x  = -pi:0.1*pi:pi
x =

Columns 1 through 7
  -3.1416   -2.8274   -2.5133   -2.1991   -1.8850   -1.5708   -1.2566

Columns 8 through 14
  -0.9425   -0.6283   -0.3142         0    0.3142    0.6283    0.9425

Columns 15 through 21
   1.2566    1.5708    1.8850    2.1991    2.5133    2.8274    3.1416

>> y = sin (x)
y =

Columns 1 through 7
  -0.0000   -0.3090   -0.5878   -0.8090   -0.9511   -1.0000   -0.9511

Columns 8 through 14
  -0.8090   -0.5878   -0.3090         0    0.3090    0.5878    0.8090

Columns 15 through 21
   0.9511    1.0000    0.9511    0.8090    0.5878    0.3090    0.0000

>> plot (x, y)

>>  title ('sine wave curve', 'fontsize', 24, 'color', 'red')

>> xlabel ('x values')

>> ylabel ('f (x) values')

>> box off

>> text (-2, 0.4, 'Engineers are Best')
```

ألـيس مـن الممكـن أن يكـون وضـعك الحالي أفضل وبنفس الإمكانات المتاحة.



ـ في حال الرغبة بإضافة رسم جديد على نفس الشكل الحالي (Figure 1) فلابد من استخدام الأمر (hold on) أولاً وإلا سيقوم بمسح الرسم الحالي وإظهار الرسم الجديد فقط.

```
>> hold on

>> z = cos (x);

>> plot (x, z, 'bo', x, z, 'r-')          % 'b' = blue, '-' = solid line

>> close        % لإغلاق نافذة الرسم الحالية
```



روح عن نفسك بين الحين
والآخر بنكت وألعاب ذكاء.

1. Solve the following introductory problems on strings.
   a. Write a function dayName that consumes a parameter, day, containing the numerical value of a day in the moth of September 2008. Your function should return the name of that day as a string. For example:
   dayName (8) should retrun 'Monday'

Hint: you should probably be concatenating the day names and the delimiters

   b. You are now given a variable named day, a vector that contains the numeric values of days in the month of September 2008. Write a script that will convert each numeric value in the vector days into a string named days of week with the day names separated by a comma and a space.

For example:
if days = [8, 9, 10], days of week should be 'monday, Tuesday, Wednesday'
Notice that there is no separator before the first day name or after the last one.

- يمكن من خلال الأمر calendar (2008, 9) معرفة أن شهر سبتمبر يبدأ يوم الاثنين.
- الدالة التالية تستلم رقم اليوم في شهر سبتمبر 2008 وتعيد اسم اليوم المقابل:

```
function [ dName ] = dayName (day)
d = mod (day, 7);
switch (d)
   case 1
      dName = 'Monday';
   case 2
      dName = 'Tuesday';
   case 3
      dName = 'wednesday';
   case 4
      dName = 'Thursday';
   case 5
      dName = 'Friday';
   case 6
      dName = 'Saturday';
   case 0
      dName = 'Sunday';
   otherwise
      dName = 'Invalid day';
end;
end
```

- الدالة التالية تستدعي الدالة dayName بعدد حدود عناصر المصفوفة days لتكوين الجملة المطلوبة.

```
function [ s ] = daysOfWeek (days)

s = dayName (days (1));

for i = 2:length (days)
   day = days (i);
   dName = dayName (day);
   s = [s  ', '  dName];          % concatenate
end;

end
```

- يمكن اختيار الدالة من خلال شاشة (command window):

```
>> [p] = daysOfWeek ([8 9 10])

p =

Monday, Tuesday, wednesday
```

الشرطي للمرأة التي تقود السيارة: ألم تقرأي (قف)؟ المرأة: قرأت (قف) وليس (قفي)!

2. Consider the problem the MATLAB system has in parsing the string: 'v = [1 2 3 4; 5, 6, 7; 8; 9 10] ' Your task is to use strtok to parse this line and construct the array it represents. You will write a function arraypareese that consumes a string and returns two variables: a string that is the variable name and an array.

a. Tokenize the string first using '=' as the delimiter to isolate the variable name and the expression to be evaluated. Return the variable name to the user and save the rest of the line as the variable str1 for further processing. You may assume that there are no espaces outside the characters '[]'.
b. Tokenize str1 with '[' and '] ' to remove the concatenation operators and save the first token as str2.

Hint: To understand string to kenizing, in the CommandWindow, enter>> help strtok.

c. Tokenize str2 using 'i' as the delimiter. This will produce 0 or more strings that represent the rows of the array. Save each in the variable rowstring. You may assume for now that the first row is the longest one.
d. Using nested while loops, tokenize each reowString with' ; and ' ' as delimiters and use sscanf (…) to extract the numerical value of each array entry. Save it as rowEntry.
e. Concatenate the rowEntry elements horizontally to produce each row of the array. If the row is too short, pad it with zeros.
f. Concatenate each row vertically to produce the resulting array and return that array to the caller.
g. Test the function with cases like.
empty = [ ]
row = [1 2 3 4]
diag = [0 0 0 1; 0 0 0 1; 0 1; 1]

```
function [vName, x] = arrayParse (s)
  x = [];

  [vName, str1] = strtok (s, '=');
  [temp, str2]  = strtok (str1, '[');
  [str2, temp]  = strtok (str2, '[');
  [str2, temp]  = strtok (str2, ']');

  n = 1;
  while length (str2) ~= 0
    [rowString, str2] = strtok (str2, ';');
    rowEntry = '';

    while length (rowString) ~= 0
      [str3, rowString] = strtok (rowString, ',');
      rowEntry = [rowEntry sscanf(str3, '%d')];
    end;

    m = length (rowEntry);
    if (n == 1)
      len = m;
    end;

    if (m < len)
      rowEntry (len) = 0;
    end;

    for i = 1:len
      x (n, i) = rowEntry (i);
    end;

    n = n + 1;
  end;          % while (str2)
end
```

```
>> [vName, x] = arrayParse ('empty=[]')

vName =
  empty

x =
   []

>> [vName, x] = arrayParse ('diag=[0 0 0 1; 0 0 1; 0 1; 1]')

vName =

  diag

x =
    0    0    0    1
    0    0    1    0
    0    1    0    0
    1    0    0    0

>> s = 'v=[1 2 3 4; 5, 6, 7; 8; 9 10]'

s =
  v=[1 2 3 4; 5, 6, 7; 8; 9 10]

>> [vName, x] = arrayParse (s)

vName =
  v

x =
    1    2    3    4
    5    6    7    0
    8    0    0    0
    9   10    0    0
```

الأستاذ: ما أكثر شيء يعجبك في المدرسة؟ التلميذ: الجرس!

3. Write a function called DNA complement that consumes a set of letters as a character array that form a DNA sequence such as 'gattaca'. The function will produce the complement of the sequence so that a 's become t's, g's become c's, and vice versa. The string 'gattaca' would therefore become 'ctaatgt'. You may assume that all the letters in the sequence will be lowercase and that they will all be either a, t, g, or c.

Note: you may be tempted to use iteration for this problem, but you don't need it.

```
function [y] = DNAcomplement (x)

y = x;

y (findstr (y, 'a')) = '?';
y (findstr (y, 't'))  = 'a';
y (findstr (y, '?')) = 't';


y (findstr (y, 'g')) = '?';
y (findstr (y, 'c')) = 'g';
y (findstr (y, '?')) = 'c';

end
```

ـ حيث يمكن اختبار الدالة من خلال شاشة (command window):

```
>> [x] = DNAcomplement ('gattaca')

x =

    ctaatgt

>> [z] = DNAcomplement (x)

z =

    gattaca
```

الدكتور لطلبته: أنتم أمل المستقبل، أنتم مصابيح الغد؟ طالب يشير إلى زميله النائم: لقد احترق المصباح بجانبي!

4. The function rot (s, n) is a simple Caesar cipher encryption algorithm that replaces each English letter in places forward or backward along the alphabet in the strings. For example, the result of rot ('Baz!', 3) is 'Edc!'. An encrypted string can be deciphered by simply performing the inverse rotation on it, i.e., rot ('Edc!', 3), which rotates each English letter in the strings three places to the left.

Numbers, symbols, and non-letters are not transformed. Implement the following function:

Function rotated Text = rot (text, n)

To assist you as you solve this problem, you could write several functions as local functions in the rot.m file:

is Uppercase Letter (letter), get Uppercase Letter (n), get Lowercase Letter (n), and get position (letter).

You may also wish to use the built-in MATLAB functions is letter (…), find (…), and mod (…).

```
function [result] = rot (s, n)
result = '';
  for i = 1:length (s)
     if isletter (s (i))
        if isUppercaseLetter (s (i)) == true
          result (i) = rotateUpper (s (i), n);
        else
           result (i) = rotateLower (s (i), n);
        end;
     else
        result (i) = s (i);
    end;
  end;
end

function [check] = isUppercaseLetter (ch)
if (ch >= 'A' && ch <= 'Z')
   check = true;
else
   check = false;
end;
end

function [ch1] = rotateUpper (ch, n)
  value = ch + n;
  if value >= 91
     value = mod (value, 91) + 65;
  end;
  if value <= 64
     value = value + 26;
  end;
    ch1 = char (value);
end

function [ch1] = rotateLower (ch, n)
  value = ch + n;
  if value >= 123
     value = mod (value, 123) + 97;
  end;
  if value <= 96
     value = value + 26;
  end;
  ch1 = char (value);
end
```

ـ حيث يمكن اختبار الدالة من خلال شاشة (command window):

```
>> [r] = rot ('Baz!', 3)

r =

    Edc!

>> [g] = rot (r, -3)

g =

    Baz!
```

الأستاذ: كم عدد سكان الدولة؟ التلميذ: سبعة ملايين وواحد! الأستاذ: بالكتاب سبعة ملايين، فلماذا هذا الواحد؟ التلميذ: لأني أمي ولدت أختا لي صباح هذا اليوم!

5. You have a big problem. In one of your CS courses, your professor decides that the only way you will pass the class is if you write a MATLAB function to get him out of a mess. All the grades in his class have been accidentally stored into one long string of characters containing only the letters A, B, C, D, F, and Y.

a. Your job is to write a function called Crazy Grade that will take in the string and flip the grades according to the following specifications:

A becomes F - B becomes D - C remains unchanged
D becomes B - F becomes A - Y becomes W

Your function should take in a string and return an inverted string.
You may assume that the string will only consist of valid letter grades. For example,
Crazy Grade ('BADDAD') should return 'DFBBFB'
Crazy Grade ('BAYBAY') should return 'DFWDFW'

b. To make matters worse, he wants you to organize this modified grade set. Write a function called GradeDist to bunch together all the similar grades (put all the A'S next to each other, B'S next to each other , etc.) Then calculate and return the professor's grade distribution. Your function should take in a string and return a string with all similar grades grouped together, along with an array containing percentage values from A'S all the way to F'S. **For example**, if there are 15% A'S, 16% B'S, 33% C'S, 16% D'S, 16 F'S and 4% W'S, GradeDist should return [15 16 33 16 16 4].

a)

```
function [ y ] = crazyGrade ( x )

  y = x;

  y (findstr (y, 'A'))  = '?';
  y (findstr (y, 'F'))  = 'A';
  y (findstr (y, '?'))   = 'F';

  y (findstr (y, 'B'))  = '?';
  y (findstr (y, 'D'))  = 'B';
  y (findstr (y, '?'))   = 'D';

  y (findstr (y, 'Y'))  = 'W';

end
```

```
>> [ y ] = crazyGrade ('BADDED')

y =

    DFBBEB

>> [ y ] = crazyGrade ('BAYBAY')

y =

    DFWDFW
```

b)

```
function [ z, p ] = GradeDist ( y )

 for i = 1:6
    count (i) = 0;
 end;

 for i = 1:length (y)
    switch y (i)
       case 'A'
          count (1) = count (1) + 1;
       case 'B'
          count (2) = count (2) + 1;
       case 'C'
          count (3) = count (3) + 1;
       case 'D'
          count (4) = count (4) + 1;
       case 'F'
          count (5) = count (5) + 1;
       case 'W'
          count (6) = count (6) + 1;
    end;
 end;

 z = repmat  ('A', 1, count (1));
 z = strcat (z, repmat ('B', 1, count (2)));
 z = strcat (z, repmat ('C', 1, count (3)));
 z = strcat (z, repmat ('D', 1, count (4)));
 z = strcat (z, repmat ('F', 1, count (5)));
 z = strcat (z, repmat ('W', 1, count (6)));

 for i = 1:6
   p (i) = round (count (i) * 100 / length (y));
 end;
```

```
>> [z, p] = GradeDist ('ACBWFFACCBDDDFF')

z =

   AABBCCCDDDFFFFW

 p =

    13   13   20   20   27    7
```

الإصرار أحد أهم أصول النجاح.