كلية العلوم

القسم : الفيزياء

السنة : الثالثة

المادة : فيزياء حاسوبية

المحاضرة : الرابعة/عملي/

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

٦

كلية العلوم ، كلية الصيدلة ، الهندسة التقنية ، تكنولوجيا المعلومات والاتصالات

يمكنكم طلب المحاضرات برسالة نصية (SMS) أو عبر (What's app-Telegram) على الرقم 0931497960

جامعة طرطوس
كلية العلوم
قسم الفيزياء
السنة الثالثة

# الفيزياء الحاسوبية

## المحاضرة الرابعة

## القسم العملي

### دوال ملفات M
### M- File Function

الدوال function: الدوال هي مجموعة من الأوامر أو الجمل البرمجية المكتوبة في ملفات M-Files والتي تستخدم لتؤدي وظيفة معينة ولها اسم مميز يعبر عن وظيفتها لتؤدي أمر أو مجموعة من الأوامر الجاهزة.

أنواع الدوال function type:

1- يحتوي برنامج MATLAB على مئات من الدوال الداخلية الجاهزة مبنية في بيئة البرنامج بدلاً من كتابة أو برمجة هذه الدوال في كل مرة مثل:
sum, prod, mean, inv, det, size, length, angle, abs, ..... وغيرها من الدوال التي تم برمجتها مسبقاً في برنامج MATLAB
2- دوال يتم إنشاءها من قبل المستخدم تكتب بصيغة برنامج MATLAB ويسمى هذا النوع من البرمجة بدوال ملفات M
تضاف هذه الدوال بعد إنشائها إلى مكتبة الدوال الداخلية الجاهزة
ويتم تسميتها من قبل المستخدم بتصميمه لها حسب وظيفتها ويتم كتابة البرامج بصيغة function بالشكل التالي:

```
Function [ list of output variables] =
function_name(list of input variable
```

**OR**
**Function function_name(list of input variable)**

شروط تسمية وإنشاء دوال ملفات M (M-File Function) :

1- يسمى اسم الملف بنفس اسم الدالة التي تم كتابتها في البرنامج
2- السطر الأول من البرنامج يجب أن يبدأ بالأمر function
3- يتبع اسم الدالة شروط تسمية المتغيرات في برنامج MATLAB
4- لا يجوز استخدام اسم الدالة من أسماء مشابهة لأسماء المتغيرات المستخدمة في البرنامج

الفرق بين برمجة الدوال (function) والبرمجة النصية (script) :

1- برنامج الدوال يبدأ بالأمر function بينما برنامج ال script يبدأ بالبرنامج مباشرة
2- يتطلب كتابة برنامج الدوال function تعريف المخرجات والمدخلات مع اسم الدالة بينما في البرمجة النصية script لا يحتاج إلى ذلك
3- تستطيع في برنامج الدوال function عند تنفيذه تغيير قيم المدخلات بينما في البرمجة النصية ال script لا يمكن ذلك
4- المتغيرات التي تحسب داخل ال function لا تحفظ في work space بينما في ال script تحفظ في work space وتسمى متغيرات محلية.

مثال 1 : اكتب برنامج يحل معادلة من الدرجة الثانية باستخدام Function M-File
وإذا كانت قيمة الجذر أقل من الصفر اطبع العبارة (the root is complex)

quadratic_equation.m

```
function quadratic_equation(a,b,c)
delta = b^2 - 4*a*c
if delta >0
x1 = (-b + sqrt(delta)) / (2*a)
x2 =(-b - sqrt(delta)) / (2*a)
elseif delta <0
disp ('the root is complex')
else
x1_2 = (-b / (2*a))
end
```

نتائج تنفيذ البرنامج

```
>> quadratic_equation(4,6,2)

delta =

     4


x1 =

   -0.5000


x2 =

    -1
```

مثال 2 : اكتب برنامج بلغة MATLAB لإيجاد القيمة العظمى من بين ثلاث قيم ندخلها للبرنامج باستخدام Function M-file :

```
function max1(a,b,c)
if a> b
    max= a;
     if c > max
         max = c;
     end
else
    max = b;
     if c > max
         max = c;
     end
end
max
```

```
>> max1 (10,2,18)

max =

    18
```

نتائج تنفيذ البرنامج

مثال 3: اكتب برنامج بلغة MATLAB لإدخال رقم واختبار هل هو زوجي أم فردي :

test_number.m

```
1   function n = test_number(a)
2 -  if mod(a,2)==0
3 -      n = 'even number';
4 -  else
5 -      n = 'odd number';
6 -  end
7
8
```

```
>> test_number(4)

ans =

even number
```

مثال 4: اكتب برنامج بلغة MATLAB لإيجاد معامل عدد :

fact.m

```
1   function c =fact(n )
2
3 -  v = 1;
4 -  for i=1:n
5 -      v = v * i;
6 -  end
7 -  v
8
9 -  end
10
```

```
>> fact(5)

v =

   120
```

مثال 5: اكتب برنامج بلغة MATLAB لإظهار جدول الضرب لأي رقم :

production.m

```
function production (num)
if length(num) ~= 1 | ~isnumeric(num)
    disp ('error: please enter one number')
else
    for i= 1 : 10
        disp([num2str(num),'X',num2str(i),'=',num2str(num*i)])
    end
end
```

ملاحظة:
1- يفحص البرنامج إذا كان الرقم المدخل هو ليس عنصر واحد أو قيم ليست عددية فيطبع العبارة التصحيحية لتبين عدم إدخال صحيح
2- يعمل الأمر disp على طبع مصفوفة مكونة من 5 عناصر يجب تحويل كل عنصر من عناصرها إلى نصوص (سلسة حرفية string) عن طريق num2str حيث لا يمكن استخدام مصفوفة مكونة من أرقام وحروف، يجب أن تكون المصفوفة إما حروف لوحدها أو أرقام لوحدها

نتائج تنفيذ البرنامج السابق

```
>> production(4)
4X1=4
4X2=8
4X3=12
4X4=16
4X5=20
4X6=24
4X7=28
4X8=32
4X9=36
4X10=40
```

```
>> production([1 2 ])
error: please enter one number
>> production('m')
error: please enter one number
```

فيما يلي جدول يوضح معاملات المقارنة والمعاملات المنطقية المستخدمة في كتابة البرامج :

| المعامل | الوصف |
|---|---|
| < | أصغر من |
| <= | أصغر أو يساوي |
| > | أكبر من |
| >= | أكبر أو يساوي |
| == | إشارة المساواة |
| ~= | إشارة عدم المساواة |
| & | AND (وَ) |
| \| | OR (أو) |
| ~ | NOT(نفي) |

**الدوال الرياضية في MATLAB:**

يتضمن البرنامج مكتبة رياضية تحتوي على الدوال الرياضية إضافة للعمليات الحسابية واللوغاريتمات والدوال المثلثية والمثلثية العكسية ودوال التقريب إضافة إلى دوا ل خاصة تتعامل مع التحويلات، يلخص الجدول التالي بعض الدوال المستخدمة في الماتلاب:

| اسم الدالة | وظيفة الدالة |
|---|---|
| abs | لإيجاد القيمة المطلقة لرقم أو مصفوفة |
| sqrt | لإيجاد الجذر التربيعي لرقم أو مصفوفة |
| mean | لإيجاد متوسط مصفوفة |
| power | الرفع لقوة |
| pow2 | العدد 2 مرفوع للقوة x |
| log | لإيجاد ln رقم أو مصفوفة |
| log10 | لإيجاد log رقم أو مصفوفة |
| exp | قيمة الدالة الأسية للأساس e |
| max | لإيجاد أكبر قيمة في متجه |
| min | لإيجاد أصغر قيمة في متجه |

| اسم الدالة المثلثية | وظيفة الدالة |
|---|---|
| cos | لإيجاد cos رقم |
| sin | لإيجاد sin رقم |
| tan | لإيجاد tanرقم |
| acos | لإيجاد: $\cos^{-1}$ |
| asin | لإيجاد: $\sin^{-1}$ |
| atan | لإيجاد: $\tan^{-1}$ |
| csc | لإيجاد csc التجيب القطعي |
| sec | لإيجاد sec الجيب القطعي |
| cot | لإيجاد cot |
| cosh | لإيجاد cosh رقم |
| sinh | لإيجاد sinh رقم |
| tanh | لإيجاد tanh رقم |
| **دوال الأعداد المركبة** | **وظيفة الدالة** |
| abs | لإيجاد طويلة العدد المركب |
| angle | لإيجاد زاوية العدد المركب |
| conj | لإيجاد مرافق العدد المركب |
| imag | لإيجاد قيمة الجزء التخيلي |
| real | لإيجاد قيمة الجزء الحقيقي |

| دوال التقريب والتحويلات | وظيفة الدالة |
|---|---|
| mod(y,x) | إيجاد باقي القسمة الصحيحة ل y على x |
| rem | لإيجاد باقي القسمة |
| round | لتقريب الرقم العشري باتجاه أقرب رقم صحيح |
| rad2deg | لتحويل رقم من radians إلى degree |
| deg2rad | لتحويل رقم من degree إلى radians |

```
>> y = 9;
>> x = 5;
>> mod(y,x)

ans =

     4
```

```
>> z = 2+3*i

z =

   2.0000 + 3.0000i

>> theta= angle(z)

theta =

    0.9828
```

```
>> abs(z)

ans =

    3.6056

>> rad2deg(theta)

ans =

   56.3099
```

في المثال السابق لدينا z= 2+3*i عدد مركب وعند استخدام الدالة angle أرجعت قيمة الزاوية والتي تعتبر مقدرة بالراديان ولإظهار النتيجة بالدرجات استخدمنا rad2deg

## كثيرات الحدود :

يتم التعامل مع كثيرات الحدود باستخدام برنامج MATLAB من خلال توابع خاصة، يتم تحويل كثير الحدود إلى متجه سطري (نسق من الأعداد)، تمثل الأعداد ضمن هذا المتجه معاملات كثير الحدود، يتم ترتيب الأعداد ضمن النسق تنازلياً بدءاً من القوة n (أعلى رتبة للمتحول في كثير الحدود أو درجة كثير الحدود) وحتى القوة 0

مثال: للتعبير عن كثير الحدود من الدرجة 4 له الصيغة :

$$B = 7x^4 + 6x^3 + 3x^2 + 5x + 5$$

## العمليات الحسابية على كثيرات الحدود :

### إيجاد جذور كثير الحدود :

جذور كثير الحدود هي الأعداد التي تجعل قيمة كثير الحدود معدومة، إن التابع الخاص لإيجاد جذور كثير الحدود في الماتلاب هو **roots :**

لإيجاد جذور كثير الحدود السابق :

```
>> roots(B)

ans =

   0.3868 + 0.8314i
   0.3868 - 0.8314i
  -0.8154 + 0.4295i
  -0.8154 - 0.4295i
```

لإيجاد جذور كثير الحدود :

$$C = x^2 - 2x + 1 = 0$$

```
>> C = [ 1 -2 1]

C =

     1    -2     1

>> roots(C)

ans =

     1
     1
```

كما يمكن الحصول على الجذور مباشرة دون الحاجة لإسناد معاملات كثير الحدود لمتغير:

```
>> roots([1 -2 1])

ans =

     1
     1
```

### جمع وطرح كثيرات الحدود :

تتم عمليات الجمع والطرح على كثيرات الحدود من خلال إجراء العمليات الحسابية على الأنساق الممثلة لمعاملات كثيرات الحدود. فإما أن يكون كثيري الحدود المراد جمعهما أو طرحهما من نفس الدرجة وعندها يتم جمع أو طرح الأعداد المتقابلة في النسقين مباشرة أو يكون كثيري الحدود من درجتين مختلفتين فيجب إضافة أصفار إلى النسق الأصغر درجة لجعل النسقين من نفس الأبعاد.

مثال:

لجمع وطرح كثيري الحدود :

$$A = 2x^4 + x^3 + 3x^2 + x + 1$$
$$B = 4x^2 - x - 1$$

```
>> A= [2 1 3 1 1]

A =

     2     1     3     1     1

>> B = [0 0 4 -1 -1]

B =

     0     0     4    -1    -1

>> C = A+B

C =

     2     1     7     0     0

>> D= A-B

D =

     2     1    -1     2     2
```

لجمع كثيري الحدود :

$$A = 2x^2 + x + 3$$
$$B = 4x^2 - x - 1$$

```
>> A= [2 1 3]

A =

     2     1     3

>> B=[4 -1 -1]

B =

     4    -1    -1

>> C = A+B

C =

     6     0     2
```

## جداء وقسمة كثيرات الحدود:

يستخدم التابع **conv** مختصر كلمة (convolution) للحصول على جداء كثيرات الحدود بشرط أن تعرف كثيرات الحدود بأنساق موافقة ومعبرة عن معاملاتها

أما لقسمة كثيرات الحدود نستخدم **deconv** ويحفظ الناتج في متحولين [q,r] حيث q هي ناتج القسمة و r هي باقي القسمة

مثال:

بفرض لدينا كثيرا الحدود من الشكل : $U(x) = x + 1$ $V(x) = x^2 + x + 1$

```
>> v = [1 1 1]

v =

     1     1     1

>> u = [1 1]

u =

     1     1

>> [q,r] = deconv(v,u)

q =

     1     0


r =

     0     0     1
```

```
>> U= [1 1]

U =

     1     1

>> V= [1 1 1]

V =

     1     1     1

>> W=conv(U,V)

W =

     1     2     2     1
```

**حساب قيمة كثير الحدود من أجل قيمة معينة لمتحول :**

يستخدم التابع **polyval(p,x)** حيث يعبر الرمز p
عن النسق الممثل لكثير الحدود
والرمز x يعبر عن قيمة المتحول المراد
حساب كثير الحدود عنده

مثال: لحساب قيمة كثير الحدود
$g(x) = x^2 + x + 1$ من أجل x=3

```
>> P = [1 1 1 ]

P =

     1     1     1

>> x=3

x =

     3

>> g = polyval(P,x)

g =

    13
```

**استخدام الأمر syms :**

يقوم الأمر **syms** بتحويل المتحول إلى رمز يمكن التعامل معه من قبل البرنامج والتعرف عليه، ويستخدم الأمر **subs** لحساب قيمة كثير الحدود عند قيمة المتحول المطلوبة

```
>> syms x y
>> g = x^2+x+1

g =

x^2 + x + 1

>> subs(g,3)

ans =

13
```

```
>> syms x y
>> g = 3*x^2 - 2*y^2 +10

g =

3*x^2 - 2*y^2 + 10

>> subs(g,x,3)

ans =

37 - 2*y^2
```

قيمة كثير الحدود من أجل x=3

**اشتقاق كثير الحدود:**

يستخدم التابع **polyder** لاشتقاق كثير الحدود

مثال:

لاشتقاق كثير الحدود $g(x) = x^4 + x^2 + 2x + 1$

```
>> g = [1 0 1 2 1];
>> f = polyder(g)

f =

     4     0     2     2
```

كما يمكن استخدام الأمر syms وعند الاشتقاق يستخدم الأمر diff

```
>> syms x
>> g = x^4+x^2+2*x+1

g =

x^4 + x^2 + 2*x + 1

>> f = diff(g)

f =

4*x^3 + 2*x + 2
```

مثال:
اشتقاق التابع بالنسبة للمتحول x $g = 2\sin(x * y) + e^{x.y}$

```
>> syms x y
>> g = 2*sin(x*y)+exp(x*y)

g =

exp(x*y) + 2*sin(x*y)

>> f = diff(g)

f =

2*y*cos(x*y) + y*exp(x*y)
```

**تكامل كثير الحدود:**
يستخدم التابع **polyint** لإجراء التكامل. عند أخذ ثابت التكامل بعين الاعتبار يكتب الأمر على الشكل التالي polyint(p,k) حيث p يعبر عن كثير الحدود و k يمثل ثابت التكامل
مثال:
تكامل التابع $g = 6x^2 + 10x - 6$ على اعتبار أن قيمة ثابت التكامل k= -5

```
>> p= [6 10 -6];
>> k = -5;
>> g = polyint(p,k)

g =

     2     5    -6    -5
```

كما يمكن استخدام الأمر syms والأمر int لإيجاد التكامل:

```
>> syms x
>> g = 6*x^2 +10*x -6

g =

6*x^2 + 10*x - 6

>> f = int(g)

f =

x*(2*x^2 + 5*x - 6)
```

أما لإيجاد التكامل المحدد:

```
>> h = int(g,1,2)

h =

23
```

**تدريب 1** :

اكتب برنامج في MATLAB يقوم بحساب مجوع الأعداد الزوجية المحصورة بين 0 و 20 وطباعة الاعداد الزوجية ثم أخيراً طباعة المجموع على الشاشة

**تدريب 2:**

ليكن z = 5+4i والمطلوب:
أوجد مرافق z
أوجد طويلة z
أوجد زاوية z مقدرة بالدرجات degrees
أوجد الجذر التربيعي للجزء التخيلي

**تدريب 3 :**

$$r = \frac{1}{2}t^2.\sin(h) - h.\cos(t)$$

أوجد تكامل هذا التابع إذا علمت أن حدود التكامل ثابت التكامل : 5
أوجد مشتق هذا التابع بالنسبة t

**تدريب 4** :

اكتب برنامج في MATLAB يقوم بحساب مجموع ثلاثة أعداد مدخلة وإظهاره ثم اختبار ناتج الجمع إن كان عدد زوجي ام فردي وعرض نتيجة الاختبار على الشاشة

**تدريب 5**:

اكتب برنامج MATLAB يقوم بحساب المتوسط الحسابي لمجموعة أعداد مدخلة

اشرح عمل التابع التالي واكتب الخرج الذي سيقوم بإظهاره عند استدعاؤه :

to_meter.m

```
function to_meter(n, unit)
switch unit
    case('km')
        n = n*1000;
    case('hm')
        n = n*100;
    case ('dam')
        n = n*10;
    case ('dm')
        n = n/10;
    case('cm')
        n = n/100;
    case('mm')
        n = n/1000;
    case('m')
        n = n;
    otherwise
        disp('unvalid unit');
end
fprintf('the meter value is %f\n',n);
end
```

انتهت المحاضرة