لايجاد مصفوفات P,H التي يتم الحصول عليها عند تطبيق طريقة ال

Hessenberg Decomposition Method

نطبق الكود التالي

```
a = im2double(imread('cameraman.tif'));
imshow(a)
H= hess(a);
[P,H]= hess(a);
x1=P;
 x2=H;
subplot(2,2,1); subimage(a); title('Original Image');
subplot(2,2,2); subimage(x1); title('P matrix');
subplot(2,2,3); subimage(x2); title('H matrix');
```

لايجاد مصفوفات U,S,V التي يتم الحصول عليها عند تطبيق طريقة ال

SVD Method

نطبق الكود التالي

```
a = im2double(imread('cameraman.tif'));
imshow(a)
 [U,S,V] = svd(a);
x1=U;
 x2=S;
 x3=V;
subplot(2,2,1); subimage(a); title('Original Image');
subplot(2,2,2); subimage(x1); title('U matrix');
subplot(2,2,3); subimage(x2); title('S matrix');
subplot(2,2,4); subimage(x3); title('V matrix');
```

الدالة imresize

يمكن تغيير حجم الصورة الحالية الى حجم آخر وباكثر من طريقة كتابة كما موضح في الصيغ التالية

B=imresize(A,m,method);

حيث A تمثل الصورة الرقمية السابقة وm حجم التكبير (رقم اكبر من 1) او التصغير (رقم اصغر من 1) والـ method هي اسلوب التكبير بصيغة تكرار البكسلات المجاورة من خلال اهمالها او كتابة 'nearest' او اسلوب توليد خطي ثنائي 'bilinear' .

```
a = im2double(imread('cameraman.tif'));
B=imresize(a,5,'bilinear');
imshow(a)
 x1=B;
subplot(2,2,1); subimage(a); title('Original Image');
 subplot(2,2,2); subimage(x1); title('B');
```

الطريقة الثانية هي بتحديد الحجم الجديد للصورة من خلال عدد الصفوف والاعمدة للصورة الجديدة وقد ينتج عنه بعض التشويه اذا كان التكبير او التصغير غير متناسب من ناحية الابعاد، وتكتب الصيغة بالطريقة التالية

B = imresize(A,[mrows mcols],method)

```
a = im2double(imread('cameraman.tif'));
 B = imresize(a,[5 5],'nearest')
imshow(a)
 x1=B;
subplot(2,2,1); subimage(a); title('Original Image');
 subplot(2,2,2); subimage(x1); title('B');
```

والمثال التالي يوضح تحويل الصورة الى نصف حجمها الاصلي.

```
a = im2double(imread('cameraman.tif'));
b=imresize(a,0.5);
imshow(a);
figure,imshow(b);
```

دوال ملفات M

M – File Function

الدوال function :- الدوال هي مجموعة من الاوامر او الجمل البرمجية المكتوبة في ملفات M – File والتي تستخدم لتؤدي وظيفة معينة ولها اسم مميز يعبر عن وظيفتها لتؤدي امر او مجموعة من الاوامر الجاهزة .

انواع الدوال : function types

يحتوي برنامج MATLAB على مئات من الدوال الداخلية الجاهزة مبنية في بنية البرنامج MATLAB Built in Function بدلا من كتابة او برمجة هذه الدوال في كل مرة مثل , sum prod , mean , inv , det , size , length , round , fix , rem , angle , abs وغيرها من الدوال التي تم برمجتها مسبقا في برنامج MATLAB .

دوال يتم انشاءها من قبل المستخدم User – Defined Function تكتب بصيغة برنامج MATLAB ويسمى هذا النوع من البرمجة بدوال ملفات (M – File Function) M .

- تضاف هذه الدوال بعد انشاءها الى مكتبة الدوال الداخلية الجاهزة

(MATLAB Built in Function) ، 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 variables)
```

OR

```
function  function_name (list of input variables)
```

شروط تسمية وانشاء دوال ملفات (M – File Function) M :-

يسمى اسم الملف بنفس اسم الدالة التي تم كتابتها في البرنامج .

السطر الاول من البرنامج يجب ان يبدأ بالامر function .

يتبع اسم الدالة شروط تسمية المتغيرات في برنامج MATLAB .

لايجوز استخدام اسم الدالة من اسماء مشابهة لاسماء المتغيرات المستخدمة في البرنامج .

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

- برنامج الدوال يبدأ بالامرfunction بينما برنامج الـ script يبدأ بالبرنامج مباشرة .
- يتطلب كتابة برنامج الدوال function تعريف المخرجات والمدخلات مع اسم الدالة بينما في البرمجة النصية الـ script لايحتاج الى ذلك .
- تستطيع في برنامج الدوال function عند تنفيذه تغيير قيم المدخلات بينما في البرمجة النصية الـ scriptلايمكن ذلك .
- المتغيرات التي تحسب داخل الـ function لا تحفظ في الـ workspace بينما في الـ script تحفظ في الـ workspace .

معرفة عدد المدخلات والمخرجات للدوال(nargout ,nargin) :-

يستخدم الامرnargin لمعرفة عدد المعاملات (المتغيرات) المستخدمة داخل الدالة بالشكل التالي

a = nargin (' function_name')

-: ex

a = nargin ('sum')

a=

3

ويستخدم الامر nargout لمعرفة عدد المعاملات (المتغيرات) المستخدمة كمخرجات لهذه الدالة وبالشكل التالي :-

a = nargout (' function_name')

-: ex

a = nargout (' sum')

a=

1

مثال 2 :- اكتب برنامج بلغة MATLAB لايجاد القيمة العظمى من بين ثلاث قيم ندخلها للبرنامج

Sol.

```
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
```

لتنفيذ البرنامج اكتب اسم البرنامج مع ادخال القيم مع اسم البرنامج

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

مثال 3 :- اكتب برنامج بلغة MATLABلادخال رقم واختبار هل هو زوجي ام فردي .

Sol .

```
function   n= test_number( a )
if   mod  ( a  , 2 ) == 0
n = ' even  no.' ;
```

```
else
n = ' odd  no.' ;
end
```

لتنفيذ البرنامج ، اكتب اسم البرنامج مع اعطاء قيمة بين القوسين :-

```
>>test_number (  4  )
even no.
```