اسم المادة : مبادئ التحليل العددي

تجمع طلبة كلية التكنولوجيا والعلوم التطبيقية - جامعة القدس المفتوحة

acadeclub.com

وُجد هذا الموقع لتسهيل تعلمنا نحن طلبة كلية التكنولوجيا والعلوم التطبيقية وغيرها من خلال توفيروتجميع كتب وملخصات وأسئلة سنوات سابقة للمواد الخاصة بالكلية, بالإضافة لمجموعات خاصة بتواصل الطلاب لكافة المواد:

للوصول للموقع مباشرة اضغط هنا

وفقكم الله في دراستكم وأعانكم عليها ولا تنسوا فلسطين من الدعاء ♡

دليل الجانب العملي لمقرر مبادئ التحليل العددي

1281

باستخدام برمجية مات لاب (MatLab R2017a)

إعداد المادة العلمية والتنسيق

د. حازم إسماعيل الشيخ أحمد

أ. مي محمد زكريا

أ. عاطف محمد عساف

أ. منار سعيد فياض

جامعة القدس المفتوحة

2019

المقدمة

يعد الجانب العملي من مقرر التحيل العددي من الأهمية بمكان بحيث تطلب إفراد دليل خاص بالجانب العملي وقد تم استخدام برمجية الماث لاب كونها من البرمجيات الحديثة والتي أثبتت قدرتها على حل العديد من الإشكاليات الرياضية في هذا المجال ويحتوي الدليل العملي على ثماني وحدات تتفق مع ما يقدم في الجانب النظري لهذا المقرر على النحو التالي: -

تتحدث الوحدة الأول عن تعريف ببرمجية الماث لاب ومكوناتها وأنواع المتغيرات التي سيتم التعامل معها وبعض الوظائف الخاصة بالمصفوفات كما تتحدث الوحدة الثانية عن طرق حل المعادلات غير الخطية مثل طريقة التنصيف والنقطة الثابتة ونيوتن، والوحدة الثالثة تناقش موضوعات توظيف المصفوفات في حل أنظمة المعادلات الخطية كما تعرض الوحدة الرابعة أساليب حل أنظمة المعادلات تتابعيا من خلال خوارزميات محددة منها جاكوبي و جاوس سايدل و SOR ، والوحدة الخامسة تتطرق إلى خوارزمية نيوتن في حل أنظمة المعادلات غير الخطية ، والوحدة السادسة تناقش التقريب وإيجاد المنحنيات الموائمة ، أما الوحدة السابعة فتناقش طرق الاستكمال العددي من خلال خوارزميات لاجرانج والفروق المقسومة والفروق التقدمية والرجعية لنيوتن ، أخيرا تركز الوحدة الأخيرة على خوارزميات حل المعادلات التكاملية باستخدام قواعد شبه المنحرف وسيمبسون.

المحتويات

أهمية التحليل العددي

التحليل العددي هو دراسة الطرق الرياضية العددية لإيجاد الحل التقريبي لبعض المسائل الرياضية التي تظهر عند تطبيق الرياضيات باختلاف فروعها في العلوم البحتة والتطبيقية وتحليل تقاربها ودقتها واستقرارها.

تعريف ببرمجية مات لاب

ويمكن تعريف برنامج مات لاب MATLAB على انه لغة ذات مستوى عالي للحسابات وتصميم الرسومات والبرمجة مدعمة ببرامج تسهل استخدامه وتشمل المعالجات الرياضية وتطوير الخوارزميات والنمذجة والمحاكاة وتحليل البيانات. ويؤمن برنامج MATLAB أدوات واجهة التخاطب الرسومية Graphical User Interface (GUI) التي تجعله برنامجا متطورا وطيعا.

نبذة تاريخية عن برنامج مات لاب

تاريخيا يعتبر عالم الرياضيات كليف مولر (Cleve Moler) المؤلف الأول لبرنامج مات لاب ففي العام ١٩٧٠م قام العالم مولر وعدد من زملائه بتطوير مكتبات لغة فورتران البرمجية LINPACK و EISPACK وهما المكتبات الخاصة ببرمجة المصفوفات، وقد صمم البرنامج على أساس التعامل مع البيانات والمتغيرات في صورة مصفوفات، وقد أتت تسمية البرنامج مات لاب MATLAB اختصارا للكلمات Matrix Laboratory بمعنى مختبر المصفوفات متفقة مع تصميم البرنامج. في العام ١٩٨٣ انضم العالم جاك ليتل (jack little) إلى فريق العالم مولر لتعاد كتابة مات لاب بلغة السي.

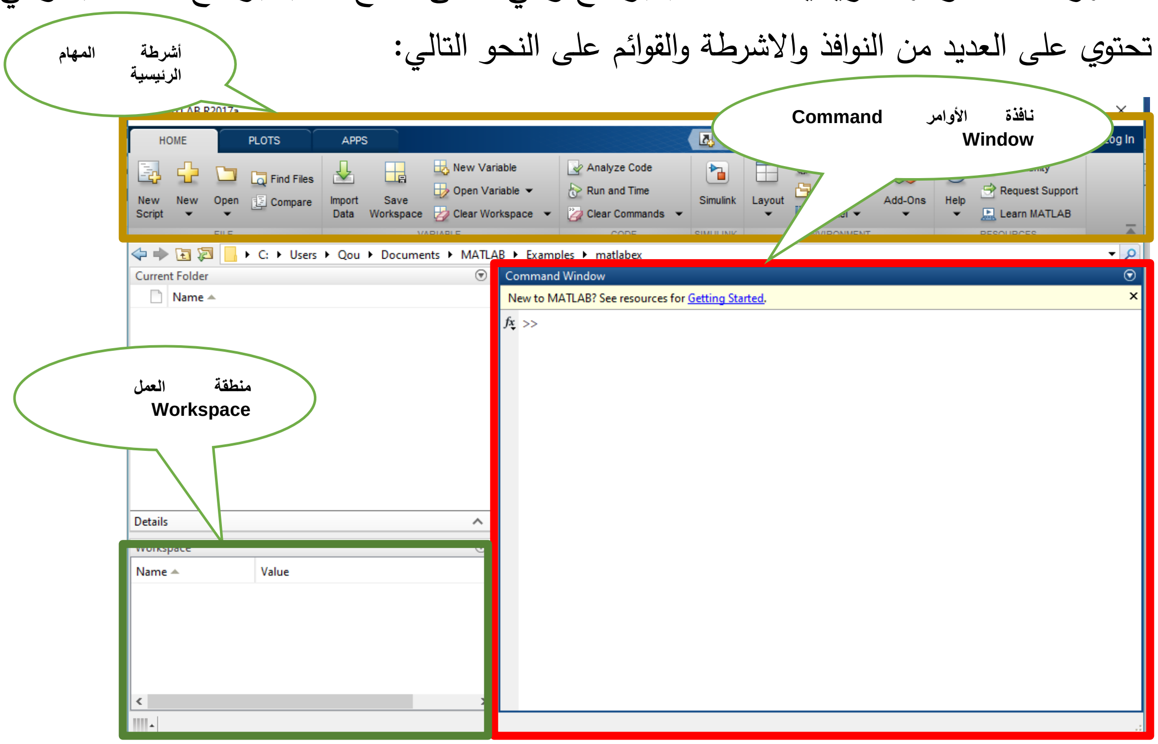
أهمية برنامج مات لاب في التحليل العددي

يعد برنامج مات لاب من البرامج التطبيقية الهامة والمتعددة الاستخدام، يعمل البرنامج في بيئة نظام التشغيل Windows ويستخدم على نطاق واسع في حل المسائل الهندسية والرياضية بكفاءة ودقة، ويمتلك برنامج مات لاب العديد من المميزات التي جعلته ينتشر بين الأكاديميين وفي المؤسسات العلمية والتقنية، ولعل اهم تلك المميزات قدرته على رسم المنحنيات والاشكال الفراغية ثلاثية الابعاد بالإضافة الى قدرته على التعامل مع الكثير من المسائل الرياضية المتعلقة بالنهايات والتفاضل الجزئي والتكامل المتعدد والعديد من التطبيقات الرياضية المتنوعة.

بعد تلك المقدمة العامة حول برنامج مات لاب فقد تبين أهمية البرنامج في العديد من التطبيقات الرياضية والتي سنوظفها في حل المسائل المتعلقة بمقرر مبادئ التحليل العددي، وسنبدأ بعرض عام عن مكونات سطح المكتب الخاص بذلك البرنامج ومن ثم كيفية استخدامه.

مكونات واجهة برنامج مات لاب

عند النقر على اختصار برنامج مات لاب على سطح الكتب او تشغيله من خلال قائمة ابدأ، ستظهر امامك الواجهة الرئيسية الخاصة بالبرنامج والتي تسمى سطح مكتب برنامج مات لاب والتي تحتوي على العديد من النوافذ والاشرطة والقوائم على النحو التالي:

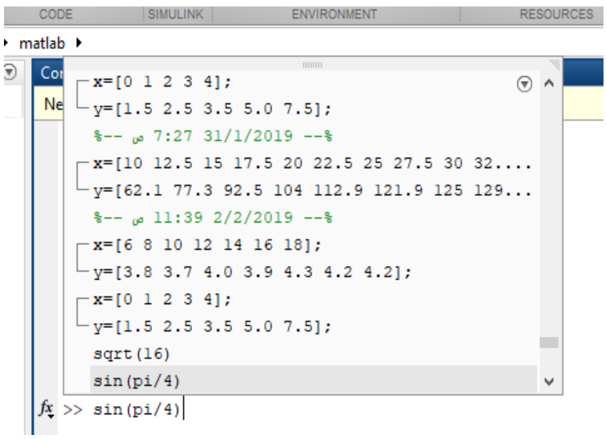


يبين الشكل السابق الواجهة الرئيسية للبرنامج بأجزائها المختلفة على النحو التالي:

أولا: نوافذ البرنامج

- نافذة الأوامر Command Window
- منطقة العمل Workspace

تستخدم نافذة الأوامر لإدخال البيانات والتعليمات البرمجية وكذلك يتم عرض نتائج المعالجات عبر تلك النافذة، في حين يتم تسجيل المدخلات والمخرجات في منطقة العمل وهناك بعض النوافذ التي تظهر عند العمل على مهمات محددة.

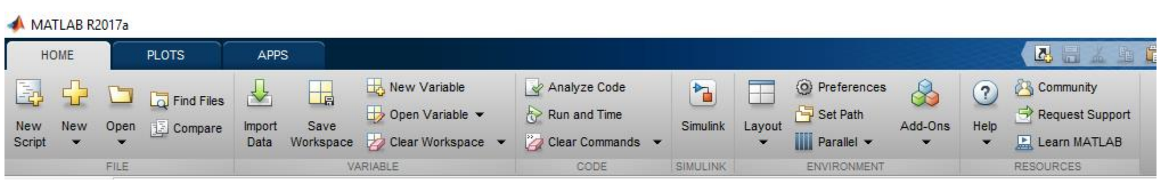


يتم تسجيل جميع الأوامر التي يتم إدخالها من خلال نافذة الأوامر Command Window، ويتم تسجيل تلك الأوامر مع الوقت والتاريخ، وعند الضغط على مفتاح السهم لأعلى على لوحة المفاتيح يظهر مربع حوار يحتوي على الأوامر المطبقة مسبقا كما في الشكل.

ثانيا: أشرطة المهام الرئيسية

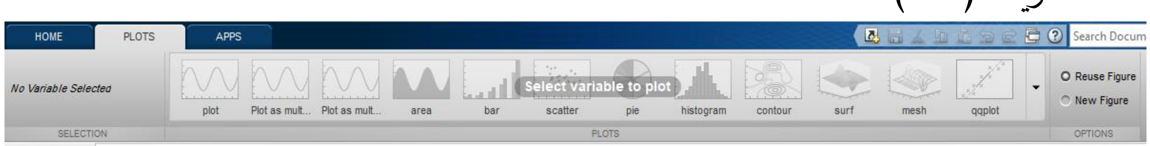
يحتوي هذا الجزء على أشرطة أوامر المهام الرئيسية التي تشتمل على مجمل المهام والتعليمات المتعلقة بمعالجة وتحليل البيانات المدخلة وهي على النحو التالي:

- شريط (Home)



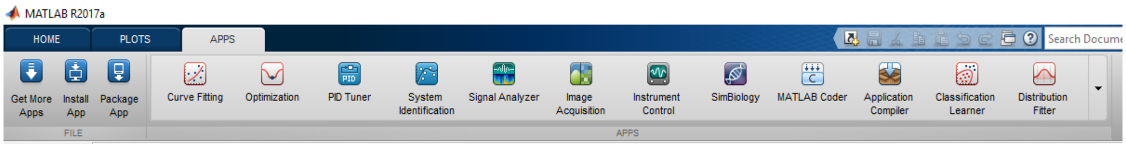
من الوظائف التي يحتويها هذا الشريط جزء مخصص للتعامل مع الملفات من انشاء ملفات جديدة مثل ملفات الأكواد والتي تمكننا من كتابة مجموعة من الاوامر لتنفيذها ككتلة واحدة وأيضا جزء خاص بالتعامل مع المتغيرات وجزء آخر متخصص بالتعامل مع الأكواد وبعض الأوامر الأخرى.

- شريط (Plot)



وهذا الجزء مخصص لتحويل البيانات وتمثيلها بيانيا

- شريط (Apps)



وهذا الجزء مخصص لتطبيقات مخصصة لعمليات مشهورة مثل التقريب أو التعامل مع الصور وتحليل البيانات وغيرها من الوظائف المشهورة.

وهناك بعض الأشرطة المخصصة والتي تظهر عند البدء بالعمل على مهمات محددة.

١. الوحدة الأولى

١. البيانات في برنامج مات لاب
٢. أنواع المتغيرات في برنامج مات لاب
٣. الأوامر والتعليمات البرمجية
٤. التطبيقات العملية

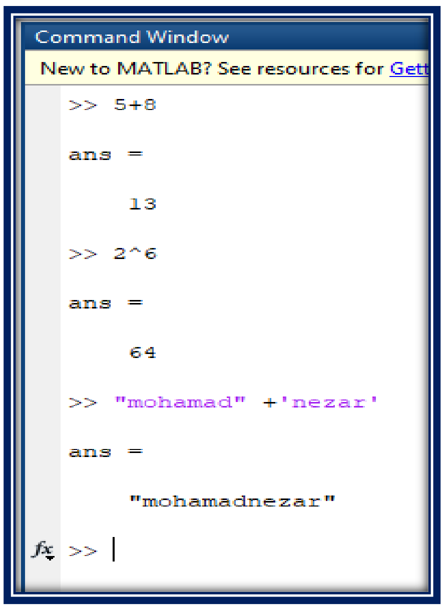
مات لاب هي لغة ذات مستوى عالي للحسابات والبرمجة، وتعتمد على معالجة البيانات باعتبارها مصفوفات بغض النظر عن نوعها، وسنستعرض من خلال هذا الفصل أنواع البيانات المستخدمة في برنامج مات لاب

١.١ البيانات في برنامج مات لاب

يمكن التعامل مع البيانات في برنامج مات لاب بطريقتين: –

- إما بشكل مباشر

حيث يستطيع برنامج مات لاب التعرف على البيانات العددية والرمزية والتعامل معها واجراء العمليات عليها فمثلا يمكنك جمع رقمين وسيقوم مات لاب بإنشاء متغير باسم ans ليخزن ناتج العملية فيه كما في الشكل المقابل.



يمكن لبرنامج مات لاب التعامل مع أنواع مختلفة من البيانات مثل بيانات رقمية ونصوص وصور إشارات صوتية وغيرها.

- أو من خلال تخزينها في متغير

تعتمد برمجية مات لاب على المكونات الرمزية والعددية في صياغة كافة المتغيرات والتعليمات البرمجية الخاصة بالبرنامج، حيث يمكن استخدام الحروف الابجدية باللغة الإنجليزية كبيرة وصغيرة والاعداد العربية والرموز الخاصة.

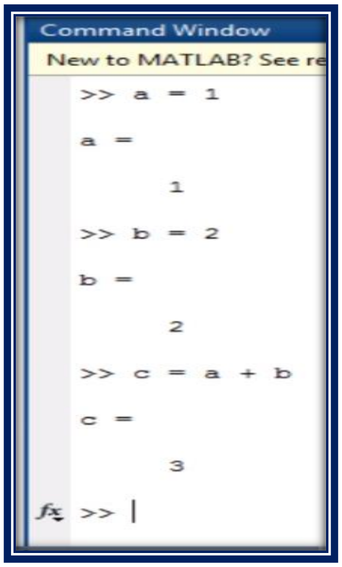
٢.١ بعض الشروط الواجب مراعاتها في تسمية المتغيرات:

- يجب تجنب الكلمات المحجوزة في تسمية كافة أنواع المتغيرات
- تسمية المتغيرات حساسة للحروف الكبيرة والصغيرة
- تعريف المتغير بما لا يتعدى ٦٣ رمز على ان تبدأ بأحد الحروف الابجدية الإنجليزية.
- لا يسمح باستخدام الرموز الخاصة او الفراغ مع استثناء رمز underscore عند تعرف المتغيرات

٣.١. أنواع المتغيرات في برنامج مات لاب

تصنف المتغيرات في برنامج مات لاب إلى عدة أنواع نذكر منها: −

١.٣.١. المتغيرات العددية Numerical Variables

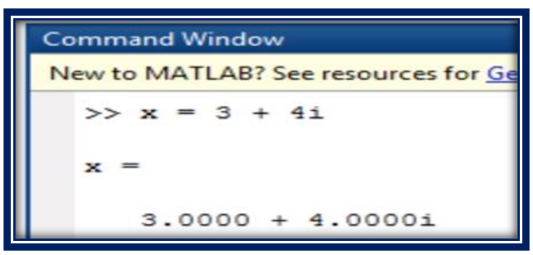


تعرف تلك المتغيرات من خلال الحروف والاعداد الرموز ويمكن ان تكون قيمة ذلك المتغير صحيحة او حقيقية او مركبة.

مثال: الصيغ التالية تستخدم لتعريف متغير واسناد قيمة له والشكل المقابل يبين ادخال قيم لمتغيرات مع تنفيذ عملية الجمع من خلال نافذة الأوامر

a=1; b=2; c=a+b;

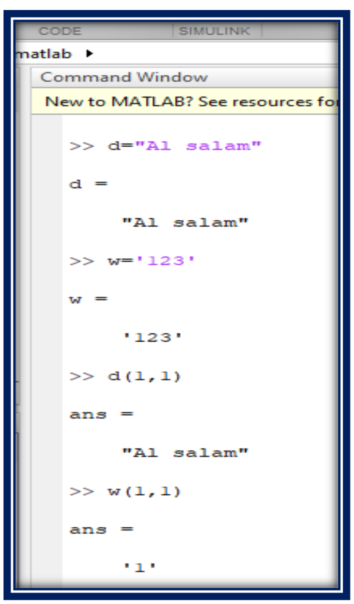
ملاحظة: يمكن تعريف باقي العمليات الحسابية مباشرة بين المتغيرات والقيم العددية كما في الشكل السابق.



هناك صنف من الأعداد يسمى بالأعداد التخيلية ويمكن تعريفة في مات لاب كما في الشكل المقابل.

٢.٣.١. المتغيرات الرمزية Symbolic Variables

تعرف تلك المتغيرات بنفس طريقة تعريف المتغيرات العددية، مع اختلاف القيم المعرفة بقيم رمزية



او عددية محصورة بين علامتي اقتباس وهناك خياران لعلامتي الاقتباس الأول زوجية وفي هذه الحالة يتم تعريف الثابت على أنه مصفوفة من عنصر واحد، والثاني فردية وهنا يتم تعريف الثابت على أنه مصفوفة من الحروف كما في الشكل المقابل.

D="Al Salam "

W='123'

١.٣.٣. المصفوفات الرمزية Symbolic matrices

يمكن تعريف مصفوفة رمزية بكتابة ثوابت داخل علامتي اقتباس مفصولة بفاصلة للبعد الواحد والفاصلة المنقوطة لإضافة أبعاد متعددة للمصفوفة كما يلي.

str = ["Mercury"،"Gemini"،"Apollo";

"Skylab"،"Skylab B"،"ISS"]

للوصول إلى أي عنصر في المصفوفة نكتب اسم المصفوفة متبوعا بأقواس مربعة"[]"، داخلها عنوان العنصر str [2،2]،

```
Command Window
New to MATLAB? See resources for Getting Started.
>> str = ["Mercury","Gemini","Apollo";
        "Skylab","Skylab B","ISS"]

str =

  2×3 string array

    "Mercury"     "Gemini"      "Apollo"
    "Skylab"      "Skylab B"    "ISS"

>> str(2,2)

ans =

    "Skylab B"
```

١.٣.٤. المصفوفات العددية Numerical matrices

تعرف المصفوفات العددية في برنامج مات لاب على النحو الوارد في الشكل المقابل حيث يفصل بين عناصر الصف بالفراغ بينما يتم الفصل بين الصفوف المختلفة بفاصلة منقوطة.

```
>> a = [1 2 3; 4 5 6; 7 8 10]

a =

     1     2     3
     4     5     6
     7     8    10
```

هناك العديد من العمليات الحسابية التي يمكن تطبيقها على المصفوفات نذكر منها: –

- منقول المصفوفة b=a'

حيث تقوم هذه العملية بتدوير عناصر المصفوفة كما في الشكل المقابل.

- حاصل ضرب مصفوفتين c=a*b

هنا تتم عملية ضرب مصفوفي لعناصر المصفوفتين وتخزينها في ثالثة كما في الشكل المقابل

- حاصل الضرب المتماثل لمصفوفتين c=a.*b

هنا نقوم بضرب كل عنصر من المصفوفة a في العنصر المناظر له من المصفوفة b

```
Command Window
New to MATLAB? See resources fo
>> b = a'

b =

     1     4     7
     2     5     8
     3     6    10

>> c = a*b

c =

    14    32    53
    32    77   128
    53   128   213

>> c = a.*b

c =

     1     8    21
     8    25    48
    21    48   100
```

- يمكن حساب معكوس المصفوفة المربعة

c=inv(a)

- يمكن تحديد أي عنصر في المصفوفة المعرفة من خلال تدوين الصف والعمود الذي يقع فيه العنصر، فلتحديد العنصر الواقع في الصف الثاني العمود الثالث في المصفوفة a السابقة نستخدم الامر

a(2،3)

```
Command Window
New to MATLAB? See resources for Getting Started.
>> a = [ 1 2 3; 2 3 4; 1 2 5]
inv(a)

a =

     1     2     3
     2     3     4
     1     2     5


ans =

   -3.5000    2.0000    0.5000
    3.0000   -1.0000   -1.0000
   -0.5000         0    0.5000

>> a(2,3)

ans =

     4

fx >>
```

- يمكن حساب مقياس المصفوفة $||A||$ (الأول والثاني واللانهائي) من خلال الدوال التالية:

norm(a،1)

norm(a،2)

norm(a،inf)

- لحساب القيم المميزة للمصفوفة المربعة نستخدم الدالة

eig(a)

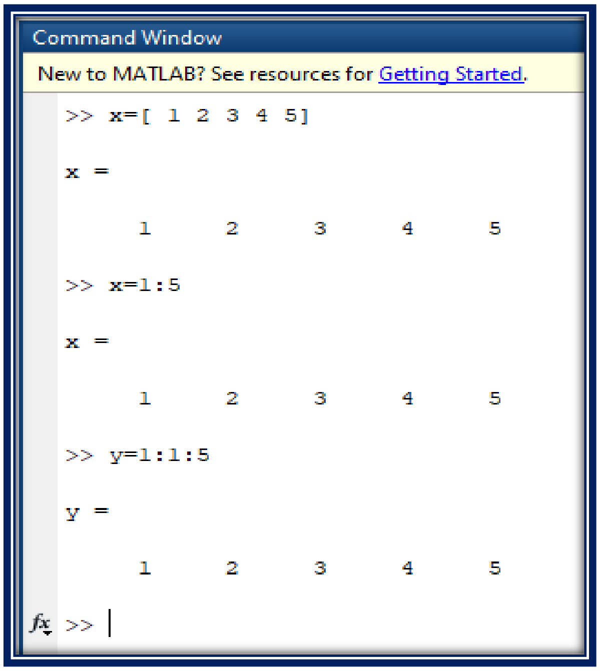
```
Command Window
New to MATLAB? See resources for Getting Started.
>> norm(a,1)

ans =

    12

>> norm(a,2)

ans =

    8.4518

>> norm(a,inf)

ans =

     9

>> eig(a)

ans =

    7.9639
   -0.2027
    1.2388

fx >>
```

١.٣.٥. المتجهات

تعتبر المتجهات حالة خاصة من المصفوفات وتعرف بعدة طرق والشكل المقابل يبين كيفية تعريف قائمة من القيم العددية على صورة متجه عددي:

يمكن حصر جميع القيم العددية بين حدين بزيادة ثابتة من خلال الصيغة التالية التي تطلب من البرنامج طباعة جميع الاعداد المحصورة بين ١ ، ٥ بزيادة 1 في كل مرة.

هناك العديد من الاستخدامات للمتجهات والتي سنتعرف عليها تباعا خلال هذا الدليل.



١.٣.٦. كثيرات الحدود العددية

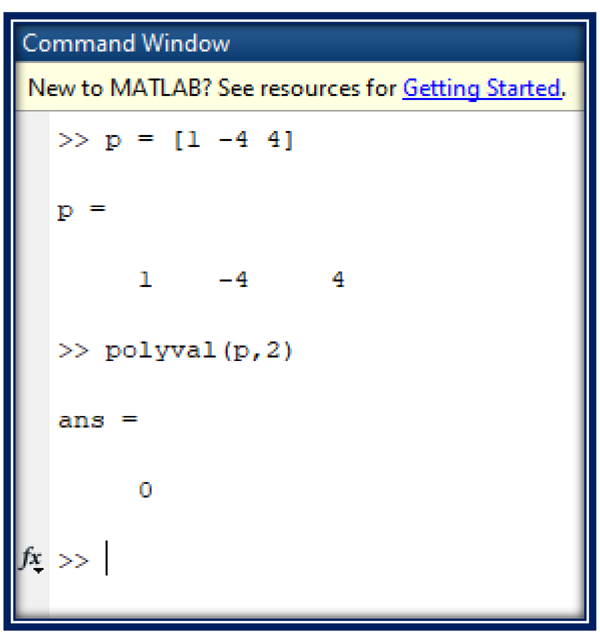
يمكن لبرنامج مات لاب التعامل مع المصفوفات على أنها كثيرات حدود فمثلا: –

لو أردنا تعريف كثيرة الحدود $p(x) = x^2 - 4x + 4$ في برنامج مات لاب فيمكن تعريفها كمصفوفة أحادية البعد كالتالي p = [1 −4 4] حيث يراعي مات لاب ترتيب المعاملات على الشكل التالي p = [p2 p1 p0] ولحساب الدالة عن قيمة محددة مثلا عند x=2 فنستخدم الدالة polyval(p,2) كما في الصورة التالية

مثال آخر لتعريف كثيرة حدود: –

$$p(x) = 4x^5 - 3x^2 + 2x + 33$$

p = [4 0 0 −3 2 33];



ملاحظة: يمكن حساب جذور كثيرات الحدود باستخدام الامر roots

```
Command Window
>> p=[1 3 -4]
roots(p)

p =

     1     3    -4


ans =

    -4
     1
```

مثال : احسب جذور كثيرة الحدود $P(x) = x^2 + 3x - 4$ من خلال برنامج ماتلاب

الحل: يتم تعريف كثيرة الحدود وحساب جذورها كما في الخطوات المبينة في الشكل المقابل

٤.١. الأوامر والتعليمات البرمجية

تتعدد الأوامر والتعليمات البرمجية المستخدمة في برنامج مات لاب، ويشار في هذا السياق الى ضرورة استخدام الحروف الابجدية الإنجليزية الصغيرة اثناء كتابة التعليمات والاوامر البرمجية بما فيها الدوال، ويمكن تصنيف تلك الأوامر من خلال مجموعة الدوال التي تعبر عن تنفيذ بعض المهام الحسابية والمنطقية على النحو التالي:

١.٤.١. الدوال المكتبية

مجموعة الدوال الرياضية المتاحة مباشرة من خلال برنامج مات لاب والتي تستدعى وتنفذ في أي وقت، الجدول التالي يبين بعض اهم تلك الدوال والغرض من استخدامها:

بيان الدالة	صيغة الدالة في الماتلاب	ملاحظة
الجذر التربيعي sqrt	>>sqrt(16)	
القيمة المطلقة abs	>>abs(−6)	
دالة الاس الطبيعي e	>>exp(1)	
دالة اللوغاريتم العادي للأساس ١٠	>>log10(20)	
دالة اللوغاريتم الطبيعي	>>log(2)	
دالة الجيب المثلثية	>>sin(pi/4)	تعرف الدوال المثلثية ومقلوباتها للزوايا بالقياس الدائري، $pi = \pi$
دالة جيب التمام	>>cos(pi/4)	تعطي ناتج $\cos(\frac{\pi}{4})$

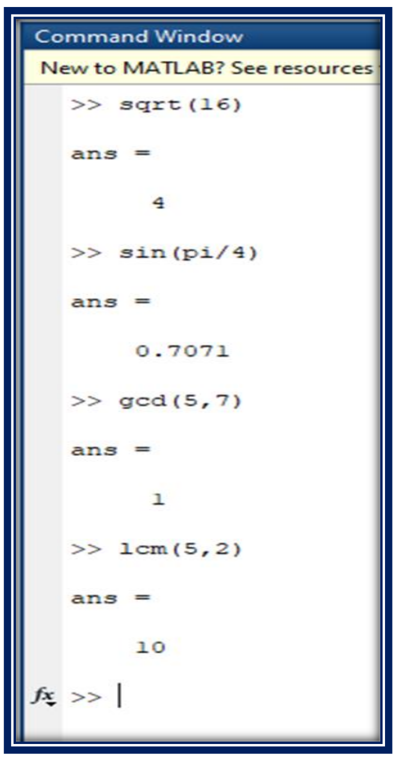
دالة معكوس الجيب	>>asin(0)	تعطي ناتج $\sin^{-1}(0)$
دالة معكوس الظل	>>atan(1)	تعطي ناتج $\tan^{-1}(1)$
دالة مقلوب جيب التمام	>>sec(pi)	
دالة الجيب الزائدي	>>sinsh(10)	
القاسم المشترك الأعظم لعددين	>>gcd(x،y)	تنتج قيمة القاسم المشترك الأعظم للقيم x،y
المضعف المشترك الأصغر لعددين	>>lcm(x،y)	تنتج قيمة المضعف المشترك الاصغر للقيم x،y

امثلة عملية على بعض هذه الدوال:

اوجد كل مما يلي:

١. احسب الجذر التربيعي للعدد ١٦ $\sqrt{16}$
٢. أوجد قيمة التعبير $\sin\left(\frac{\pi}{4}\right)$
٣. القاسم المشترك الأعظم للعددين ٧،٥
٤. المضعف المشترك الأصغر للعددين ٥،٢

الحلول كما في الشكل المقابل مع مراعاة الترتيب.



١.٤.٢. دوال التحويل للأنظمة العددية

يحتوي مات لاب على العديد من الأوامر والتعليمات البرمجية التي يمكن من خلالها تحويل القيم العددية من نظام عددي الى اخر فعلى سبيل المثال:

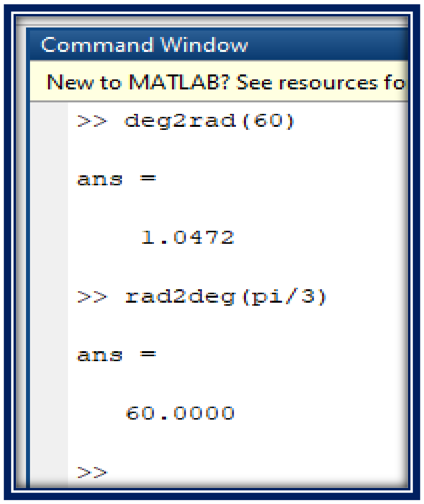
- يمكن تحويل الزوايا من القياس الدائري الى الستيني والعكس من خلال الدوال rad2deg و deg2rad.

مثال:

١. حول الزاوية التي قياسها 60^0 الى القياس الدائري المكافئ

٢. حول الزاوية التي قياسها $\frac{\pi}{3}$ الى القياس الزاوي المكافئ

الحل كما في الشكل المقابل



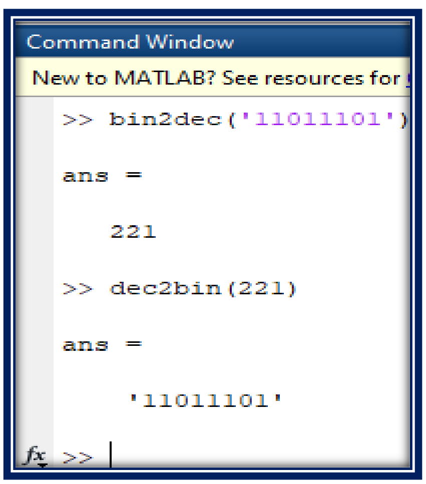
- يمكن تحويل الاعداد العشرية الى ثنائية والعكس من خلال الدوال bin2dec و dec2bin

مثال:

١. حول الرقم الثنائي 11011101 الى القيمة العشرية المكافئة

٢. حول الرقم العشري 221 الى القيمة الثنائية المكافئة

الحل كما في الشكل المقابل

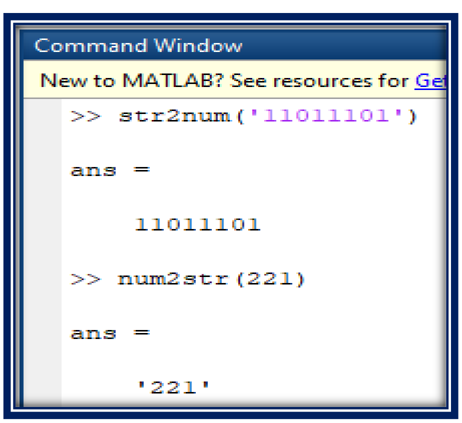


- يمكن تحويل النصوص إلى أعداد والعكس من خلال الدوال num2str و str2num

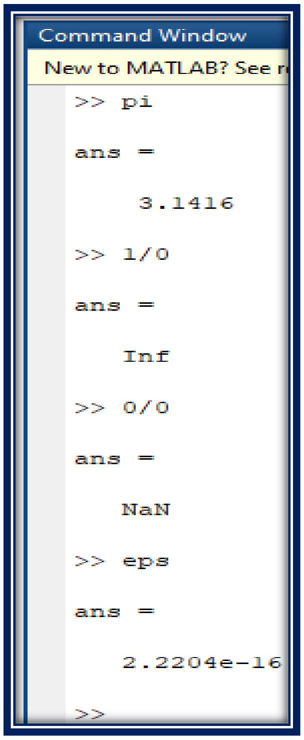
١. حول الرقم الثنائي 11011101 الى نص

٢. حول النص "221" الى رقم

الحل كما في الشكل المقابل



١.٤.٣. دوال القيم الخاصة والثوابت المحجوزة



مجموعة الثوابت والقيم الخاصة الرياضية التي تتعرف تلقائيا في البرنامج دون الحاجة لتعريفها او إدخالها، ونذكر بعض تلك القيم الخاصة المستخدمة:

١. النسبة التقريبية $\pi = \frac{22}{7}$ وتعرف في البرنامج على الصيغة pi.
٢. اللانهاية ∞ وتعرف في البرنامج من خلال $1/0$. (INF)
٣. القيمة غير المعينة $0/0$. (NaN)
٤. قيمة ϵ وهي متناهية بالصغر وتقدر عند القيمة 2^{-52} وتكتب من خلال eps

١.٤.٤. الدوال التحكمية

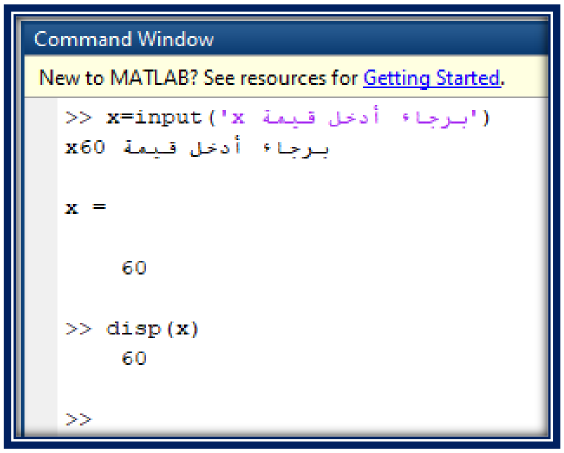
تستخدم الدوال التحكمية في برنامج مات لاب لمسح او تنظيف الأوامر والتعليمات البرمجية ضمن شاشات البرنامج على النحو التالي:

- $cls \gg$ مسح محتويات نافذة الأوامر Command Window مع بقائها في نافذة منطقة العمل Workspace
- $clear \gg$ مسح محتويات نافذة منطقة العمل Workspace
- $clear\ a\ b \gg$ مسح المتغيرات b،a من نافذة منطقة العمل Workspace
- $who \gg$ عرض محتويات نافذة منطقة العمل Workspace

١.٤.٥. أوامر الادخال والإخراج

عند صياغة بعض البرامج في برنامج مات لاب نستخدم أوامر الادخال لتمرير متغير معين للبرنامج ونستخدم بعض أوامر العرض والإخراج لعرض نتائج المعالجة ولعل اهم تلك الأوامر المستخدمة في ذلك:

- امر الادخال input:



المثال في الشكل المقابل يطلب من المستخدم إدخال قيمة للمتغير x عن طريق عرض جملة لتوضيح ما المراد إدخاله.

- امر الإخراج disp:

لعرض قيمة المتغير x

١.٤.٦. جمل التحكم و التكرار

تستخدم الجمل الشرطية في التعليمات البرمجية لبرنامج مات لاب والتي يتم من خلالها الحصول على ناتج معين من اختبار شرطي، وتستخدم الجملة الشرطية if وكذلك جملة التوزيع switch case لتنفيذ مثل تلك المهام. وسيتم التعرف على الجملة الشرطية if statement وطريقة التعامل معها في برنامج مات لاب.

- **جمل التحكم**

تستخدم هذه الجملة للتحكم في سير البرنامج بناء على شرط ما، وتكتب على احدى الصيغ الثلاث التالية:

١- جملة if

في حال وجود شرط وحيد تكتب الجملة الشرطية على الصورة التالية:

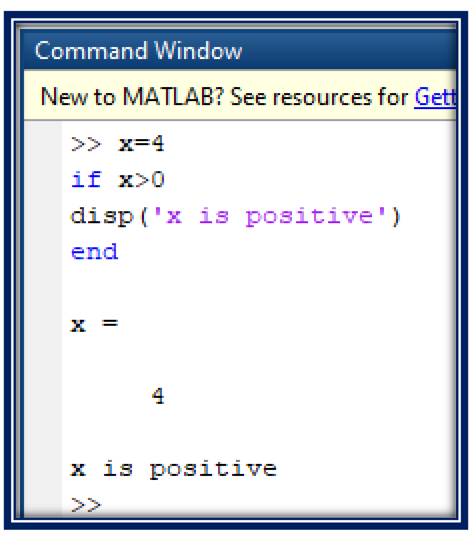
if condition

Statement

end

مثال:

أكتب برنامج لتحديد هل المتغير x موجب



٢- جملة الشرط if/else

تستخدم هذه الصيغة لتحقيق احدى نتيجتين، فاذا تحقق شرط الجملة الشرطية if تحققت النتيجة الواردة خلف الجملة if مباشرة والا تحققت النتيجة الواردة خلف else

مثال:

أكتب برنامج لتحديد هل المتغير x موجب أم لا

```
Command Window
New to MATLAB? See resources for Ge
>> x=-5
if x>0
disp('x is positive')
else
disp('x is negative')
end

x =

    -5

x is negative
>>
```

٣- جملة الشرط if/elseif

تتيح هذه الجملة الشرطية إمكانية تحقق اكثر من شرط ، المثال التالي يبين عمل هذا النوع من الجمل الشرطية:

مثال: بين ما إذا كانت القيمة المدخلة موجبة او سالبة او تساوي صفر

```
Command Window
New to MATLAB? See resources for Ge
>> x=-5
if x>0
disp('x is positive')
elseif  x<0
disp('x is negative')
else
disp('x=0')
end

x =

    -5

x is negative
>>
```

- **جمل التكرار**

١. جملة التكرار المحدود for loop statement

تستخدم تلك الجملة لتكرار جملة او مجموعة من الجمل والتعليمات البرمجية بشكل محدود، وتصاغ على الصور التالية:

```
Command Window
New to MATLAB? See resources for Getting
>> for v = 1.0:-0.2:0.0
   disp(v)
end
```

تطبيقات عملية:

١- احسب عدد الاعداد المختلفة الذي يستوعبه نظام $F(2،3،-1،1)$

الحل:

يعتمد نظام F على المعالم التالية: $F(b،p،M_1،M_2)$ حيث b يرمز قيمة القاعدة ، p يرمز لطول الكلمة ، M_1 اصغر اس يقبله الحاسوب ، M_2، اكبر اس يقبله الحاسوب.

وتحسب عدد الاعداد المختلفة بالاعتماد على الصيغة التالية:

$$1+2(b-1)b^{p-1}(M_2-M_1+1)$$

```
Command Window
New to MATLAB? See resources for Getting Started.
>> b=2

b =

     2

>> p=3

p =

     3

>> m1=-1

m1 =

    -1

>> m2=1

m2 =

     1

>> f=1+2*(b-1)*b^(p-1)*(m2-m1+1)

f =

    25
```

٢- لديك المصفوفة $a = \begin{bmatrix} -1 & 4 & 8 \\ 4 & 2 & 8 \\ 9 & 1 & 0 \end{bmatrix}$

احسب باستخدام برنامج ماتلاب كل مما يلي:

١- محدد المصفوفة a

٢- معكوس المصفوفة a

٣- المقياس الأول والثاني واللانهائي للمصفوفة a

٤- القيم المميزة للمصفوفة a

٥- باستخدام الامر diag اوجد عناصر القطر الرئيسي للمصفوفة

```
Command Window
New to MATLAB? See resources for Getting Started.
>> a=[-1 4 8 ; 4 2 8 ; 9 1 0]

a =

    -1     4     8
     4     2     8
     9     1     0

>> det(a)

ans =

   184

>> inv(a)

ans =

   -0.0435    0.0435    0.0870
    0.3913   -0.3913    0.2174
   -0.0761    0.2011   -0.0978

>> norm(a,1)

ans =

    16

>> norm(a,2)

ans =

   12.5896

>> norm(a,inf)

ans =

    14

>> eig(a)

ans =

   11.2082
   -8.2082
   -2.0000

>> diag(a)

ans =

    -1
     2
     0
```

٢. الوحدة الثانية

طرق حل المعادلات غير الخطية ذات المتغير الواحد

١. طريقة التنصيف Bisection Method

٢. طريقة النقطة الثابتة Method Fixed Point

٣. طريقة نيوتن رافسون Newton-Raphson Method

٤. طريقة القاطع Secant Method

١.٢. طريقة التنصيف

باستخدام طريقة التنصيف أوجد جذرا تقريبيا للمعادلة $f(x)=e^x+3x=0$ على الفترة $\alpha\in[-0.5,0]$

الكود التالي في الماتلاب يجد جذرا تقريبيا باستخدام 15 تكرار كالتالي:

```
f= inline('exp(t)+3*t');الإقتران المراد ايجاد الجذر التقريبي له
a=-0.5;b=0; بداية ونهاية الفترة
n=10; عدد التكرارات
format long
c = f(a); d = f(b);حساب قيمة الاقتران عند اطراف الفترة
if c*d > 0.0
 إذا كان حاصل الضرب للقيمتين موجب أي أكبر من صفر فلا يوجد جذر داخل تلك الفترة ويتم طباعة الجملة التالية والتي توضح ذلك
    error('An Error Occured The Function has same sign at both
endpoints.')
end
disp('           x                   y')
for i = 1:n
    x = (a + b)/2;
    y = f(x);
    disp([    x      y])
    if y == 0.0      % solved the equation exactly
        e = 0;
        break      % jumps out of the for loop
    end
    if c*y < 0
        b=x;
    else
        a=x;
    end
end
e = (b-a)/2;
[x e]
```

بعد تنفيذ الاوامر أعلاه يعطي البرنامج النتائج التالية:

```
         x                     y
  -0.250000000000000    0.028800783071405

  -0.375000000000000   -0.437710721209028

  -0.312500000000000   -0.205884371053358

  -0.281250000000000   -0.088910398010993

  -0.265625000000000   -0.030148403929180

  -0.257812500000000   -0.000697392705427

  -0.253906250000000    0.014045776561826

  -0.255859375000000    0.006672715161449

  -0.256835937500000    0.002987292396774

  -0.257324218750000    0.001144857682882


ans =

  -0.257324218750000    0.000244140625000
```

قيمة الجذر التقريبية بعد التكرار العاشر

قيمة الاقتران عند تلك القيمة

٢.٢. طريقة النقطة الثابتة

أوجد جذرا تقريبيا للاقتران $g(x) = \frac{1}{2}(x + e^{-x}), \quad x > 0$ باستخدام النقطة الثابتة.

```
% Finding the nontrivial root of
% f(x) =0.5*(exp(-x)+x);
% using the Simple Fixed-Point Iteration
clear all
x = 0.0      %initial guess النقطة الابتدائية
Es = 0.1     %tolerance القيمة التي يتوقف عندها التكرار
Ea = 1000;   %randomly large relative approximate error
xold = x;
n = 0;       %iteration counter
while Ea > Es
    x = 0.5*(exp(-x)+x);  (x=g(x)) الاقتران على صورة
    Ea = abs((x-xold)/x)*100; حساب الخطأ المطلق
    xold = x;  تعديل القيم
    n = n + 1;  العداد
    [n Ea x'] طباعة النتائج
```

```
end
x  %the root
n   %number of iterations
Ea  %the error
```

٣.٢. طريقة نيوتن رافسون

مثال (١١) صفحة ٨٩

يتم كتابة أوامر البرنامج في ملف M-file وتخزينه باسم mynewton كالتالي:

```
function x = mynewton(f،f1،x0،n)
% Solves f(x) = 0 by doing n steps of Newton’s method starting at x0.
% Inputs: f -- the function f = @(x)exp(x)+3*x
% f1 -- it’s derivative f1 = @(x) exp(x)+3
% x0 -- starting guess، a number zero
% n -- the number of steps to do 5
% Output: x -- the approximate solution
x = x0; % set x equal to the initial guess x0
for i = 1:n % Do n times
x = x - f(x)/f1(x) % Newton’s formula، prints x too
% in the command window write mynewton(f،f1،x0،n)
end
end
```

في صفحة command window يتم كتابة الاوامر التالية:

```
Command Window
>> format long
f = @(x)exp(x)+3*x;
f1 = @(x) exp(x)+3;
x0=0;
n=4;
mynewton(f,f1,x0,n)

x =

  -0.250000000000000


x =

  -0.257621672780536


x =

  -0.257627653046074


x =

  -0.257627653049737


ans =

  -0.257627653049737
```

٤.٢. طريقة القاطع

مثال ١٤ صفحة ٩٦

يتم كتابة أوامر البرنامج في ملف M-file وتخزينه باسم mysecant كالتالي:

```
function x = mysecant(f،x0،x1،n)
% Solves f(x) = 0 by doing n steps of the secant method
% starting with x0 and x1.
% Inputs: f -- the function
% x0 -- starting guess، a number
% x1 -- second starting guess
% n -- the number of steps to do
% Output: x -- the approximate solution
y0 = f(x0);
y1 = f(x1);
for i = 1:n % Do n times
x = x1 - (x1-x0)*y1/(y1-y0) % secant formula.
y=f(x) % y value at the new approximate solution.
% Move numbers to get ready for the next step
x0=x1;
y0=y1;
x1=x;
y1=y;
end
end
```

في صفحة command window يتم كتابة الاوامر التالية:

```
f=@(x) exp(x)+3*x
x0=-0.5;
x1=0;
n=4;
mysecant(f،x0،x1،n)
```

تظهر النتائج التالية :

Command Window

```
x =

  -0.264065537984426


y =

  -0.024273404085180


x =

  -0.257807668276102


y =

   -6.791638578717008e-004


x =

  -0.257627534527687


y =

     4.471698279440162e-007
```

٣. الوحدة الثالثة

الطرق المباشرة في حل أنظمة من المعادلات الخطية غير المتجانسة

١-طريقة جاوس Gaussian Method

٢-طريقة جاوس – جوردان Gauss Jordan Method

٣-طريقة النظير الضربي Inverse Method

٤-تحليل المصفوفة LU Decomposition Method

١.٣. طريقة جاوس والتعويض العكسي :

يتم تكوين المصفوفة الممتدة $[A:b]$ واجراء عمليات الصف البسيط حيث يتم تحويل مصفوفة المعاملات إلى مصفوفة مثلثية علوية

الأوامر التالية في برنامج مات لاب تمثل طريقة جاوس والتعويض العكسي (مثال صفحة ١٤٩ من الكتاب)

```
% Code from "Gauss elimination using MATLAB"
% a is the matrix of coefficients with column b
%we want to solve the the linear system page 149 from the book:
%EQ1: X1-X2+2X3-X4=-8
%EQ2:2X1-2X2+3X3-3X4=-20
%EQ3:X1+X2+X3       =-2
%EQ4:X1-X2+4X3+3X4=4
%we input the augmented matrix [A:b] as follows:

a = [1 -1 2 -1 -8
    2 -2 3 -3 -20
    1 1 1 0 -2
    1 -1 4 3 4];
%Gauss elimination method [m،n)=size(a);
[m،n]=size(a);
for j=1:m-1
    for z=2:m
        if a(j،j)==0
            t=a(j،:);a(j،:)=a(z،:);
            a(z،:)=t;
        end
    end
    for i=j+1:m
        a(i،:)=a(i،:)-a(j،:)*(a(i،j)/a(j،j));
    end
end
x=zeros(1،m);
for s=m:-1:1
    c=0;
    for k=2:m
        c=c+a(s،k)*x(k);
    end
    x(s)=(a(s،n)-c)/a(s،s);
end
disp('Gauss elimination method:');
a
x'
```

بعد التنفيذ تظهر النتائج التالية :

```
Gauss elimination method:

a =

     1    -1     2    -1    -8
     0     2    -1     1     6
     0     0    -1    -1    -4
     0     0     0     2     4


ans =

    -7
     3
     2
     2

>>
```

أي أن X1=-7،X2=3 ،X3=2 ، X4=2

٢.٣. طريقة جاوس_ جوردان :

يتم تكوين المصفوفة الممتدة واجراء عمليات الصف البسيط حيث يتم تحويل مصفوفة المعاملات إلى مصفوفة قطرية (تدريب ٤ صفحة ١٥٥)

```
% Code from "Gauss-Jordan elimination using MATLAB"
% a is the matrix of coefficients with column b
%we want to solve the the linear system
%EQ1: 2X1-X2+X3=2
%EQ2:-X1+2X2-X3=1
%EQ3:X1+X2+X3=5
%we input the augmented matrix [A:b] as follows:

a = [2 -1 1 2
    -1 2 -1 1
    1 1 1 5];

%Gauss-Jordan method
[m،n]=size(a);
for j=1:m-1
    for z=2:m
        if a(j،j)==0
            t=a(1،:);a(1،:)=a(z،:);
            a(z،:)=t;
        end
    end
    for i=j+1:m
        a(i،:)=a(i،:)-a(j،:)*(a(i،j)/a(j،j));
```

```
    end
end

for j=m:-1:2
    for i=j-1:-1:1
        a(i،:)=a(i،:)-a(j،:)*(a(i،j)/a(j،j));
    end
end

for s=1:m
    a(s،:)=a(s،:)/a(s،s);
    x(s)=a(s،n);
end
disp('Gauss-Jordan method:');
a
x'
```

بعد التنفيذ يعطي النتائج:

```
Gauss-Jordan method:

a =

     1     0     0     1
     0     1     0     2
     0     0     1     2


ans =

     1
     2
     2
```

أي أن X1=1،X2=2 ، X3=2

٣.٣. طريقة النظير الضربي لمصفوفة المعاملات

لإيجاد الحل للنظام السابق في المثال أعلاه

$$EQ1: 2x_1 - x_2 + x_3 = 2$$
$$EQ2: -x_1 + 2x_2 - x_3 = 1$$
$$EQ3: x_1 + x_2 + x_3 = 5$$

باستخدام الماتلاب يتم كالتالي في شاشة الاوامر: حيث الامر inv(A) في الماتلاب يعطي النظير الضربي للمصفوفة A وحل النظام هو **$X=A^{-1}b$** وقيم المتغيرات هي كما ظهرت في المثال السابق باستخدام طريقة جاوس جوردان.

```
Command Window
>> A=[2 -1 1;-1 2 -1;1 1 1];
>> b=[2;1;5];
>> sol=inv(A)*b

sol =

    1.0000
    2.0000
    2.0000

>>
```

٤.٣. طريقة تحليل المصفوفة LU

لإيجاد الحل للنظام السابق باستخدام تحليل مصفوفة المعاملات

$$EQ1: 2x_1 - x_2 + x_3 = 2$$
$$EQ2: -x_1 + 2x_2 - x_3 = 1$$
$$EQ3: x_1 + x_2 + x_3 = 5$$

يتم ادخال مصفوفة المعاملات A في الماتلاب كالتالي A=[2 -1 1;-1 2 -1;1 1 1]

ثم ادخال عمود الحل b كالتالي b=[2;1;5] والأمر [L U P]=lu(A) يقوم بتحليل مصفوفة المعاملات A الى ثلاثة مصفوفات مصفوفة مثلثية سفلية L مصفوفة مثلثية علوية U ومصفوفة قطرية P

```
Command Window
>> b=[2;1;5];
A=[2 -1 1;-1 2 -1;1 1 1];
[L U P]=lu(A);
D=P*b;
Y=L\D;
>> X=U\Y

X =

     1
     2
     2

>>
```

٤. الوحدة الرابعة

الطرق غير المباشرة في حل أنظمة من المعادلات الخطية غير المتجانسة

١-طريقة الجاكوبي Jacobi Iterative Method

٢-طريقة جاوس سايدل Gauss Seidel Iterative Method

٣-طريقة SOR Successive Over Relaxation Method

١.٤. طريقة الجاكوبي

نريد حل النظام الخطي التالي بطريقة الجاكوبي :

$$E1: 4x_1 + x_2 - 4x_3 = -5$$
$$E2: x_1 + 2x_2 + x_3 = 1$$
$$E3: \quad -x_2 + 4x_3 = 9$$

باستخدام المتجه الابتدائي : $\vec{X} = [1 \quad 1 \quad 1]$

يتم ذلك في الماتلاب

```
% jacobi Method

clc
clear
format short
A=[4 1 -4;1 2 1;0 -1 4];
b=[-5;1;9];
l=tril(-A،-1);
u=triu(-A،1);
d=diag(diag(A));
tj=inv(d)*(l+u);
cj=inv(d)*b;
xj=[1;1;1];
N=5;
for i=1:N
    xj=tj*xj+cj;
    [i xj']
end
```

وتظهر النتائج التالية عند تنفيذ الاوامر اعلاه:

```
Command Window

ans =

    1.0000   -0.5000   -0.5000    2.5000


ans =

    2.0000    1.3750   -0.5000    2.1250


ans =

    3.0000    1.0000   -1.2500    2.1250


ans =

    4.0000    1.1875   -1.0625    1.9375


ans =

    5.0000    0.9531   -1.0625    1.9844
```

حيث العمود الأول هو رقم التكرار و العمود الثاني والثالث والرابع قيم المتغيرات X1،X2، X3 على الترتيب في كل تكرار.

٢.٤. طريقة جاوس سايدل التتابعية

لحل المثال أعلاه يتم تنفيذ الأوامر التالية في الماتلاب

```
% Gauss Seidel Method
clc
clear
format short
A=[4 1 -4;1 2 1;0 -1 4];
b=[-5;1;9];
l=tril(-A،-1);
u=triu(-A،1);
d=diag(diag(A));
tg=inv(d-l)*u;
cg=inv(d-l)*b;
xg=[1;1;1];
N=5;
for i=1:N
    xg=tg*xg+cg;
    [i xg']
end
```

تظهر النتائج التالية:

```
Command Window

ans =

    1.0000   -0.5000    0.2500    2.3125


ans =

    2.0000    1.0000   -1.1563    1.9609


ans =

    3.0000    1.0000   -0.9805    2.0049


ans =

    4.0000    1.0000   -1.0024    1.9994


ans =

    5.0000    1.0000   -0.9997    2.0001
```

حيث العمود الأول هو رقم التكرار و العمود الثاني والثالث والرابع قيم المتغيرات X1،X2، X3 على الترتيب في كل تكرار.

٣.٤. طريقة SOR

لحل المثال أعلاه يتم تنفيذ الأوامر التالية في الماتلاب باستخدام ثابت التسارع $\omega = 1.2$

```
% SOR Method using Accelerating constant =1.2

clc
clear
format short
A=[4 1 -4;1 2 1;0 -1 4];
b=[-5;1;9];
w=1.2;
l=tril(-A،-1);
u=triu(-A،1);
d=diag(diag(A));
ts=inv(d-w*l)*((1-w)*d+w*u);
cs=inv(d-w*l)*b;
```

```
xs=[1;1;1];
N=5;
for i=1:N
    xs=ts*xs+cs;
    [i xs']
End
```

تظهر النتائج التالية:

```
Command Window

ans =

    1.0000   -0.5500    0.0300    2.0590


ans =

    2.0000    1.3218   -1.5345    1.3779


ans =

    3.0000    0.5994   -0.3795    1.8606


ans =

    4.0000    0.9767   -1.1265    1.5399


ans =

    5.0000    0.7405   -0.6430    1.7491
```

حيث العمود الأول هو رقم التكرار و العمود الثاني والثالث والرابع قيم المتغيرات X1، X2، X3 على الترتيب في كل تكرار.

٥. الوحدة الخامسة

حل الأنظمة غير الخطية باستخدام طريقة نيوتن

لحل النظام الغير خطي لتالي باستخدام طريقة نيوتن على برمجية الماث لاب :

$$f_1(x_1,x_2)=x_1+2x_2-3=0$$
$$f_2(x_1,x_2)=2x_1^2+x_2^2-5=0$$

باستخدام القيم الابتدائية : [1.5,0.5]

نقوم بكتابة الكود التالي

```
% mymultnewton
format long;
n=4; % set some number of iterations، may need adjusting
f = @(x)[x(1)+2*x(2)-3 ; 2*x(1)^2+x(2)^2-5]; % the vector function
% the matrix of partial derivatives
Df = @(x)[1،2 ;4*x(1)،2*x(2)];
x = [1.5;1.0;] % starting guess
for i = 1:n
Dx = -Df(x)\f(x); % solve for increment
x = x + Dx; % add on to get new guess
[i x']
end
```

فتظهر النتائج لأربعة تكرارات :

```
x =

   1.500000000000000
   1.000000000000000


ans =

   1.000000000000000   1.500000000000000   0.750000000000000


ans =

   2.000000000000000   1.488095238095238   0.755952380952381


ans =

   3.000000000000000   1.488033873343152   0.755983063328424


ans =

   4.000000000000000   1.488033871712585   0.755983064143708
```

حيث العمود الأول يمثل رقم التكرار والعمود الثاني قيمة X_1 والعمود الثالث قيمة X_2 في كل تكرار.

٦. الوحدة السادسة

التقريب

١-المستقيم الموائم

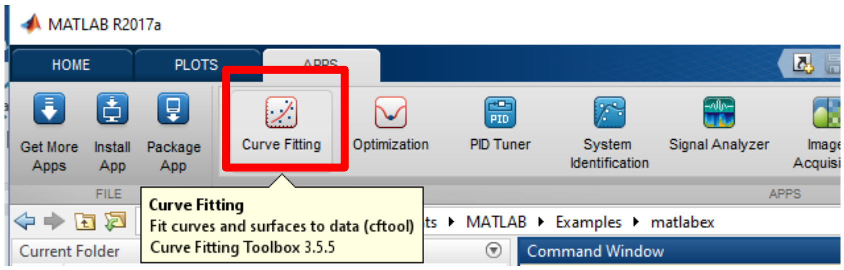
٢-تقريب كثيرة حدود

٣-التقريب الأسي

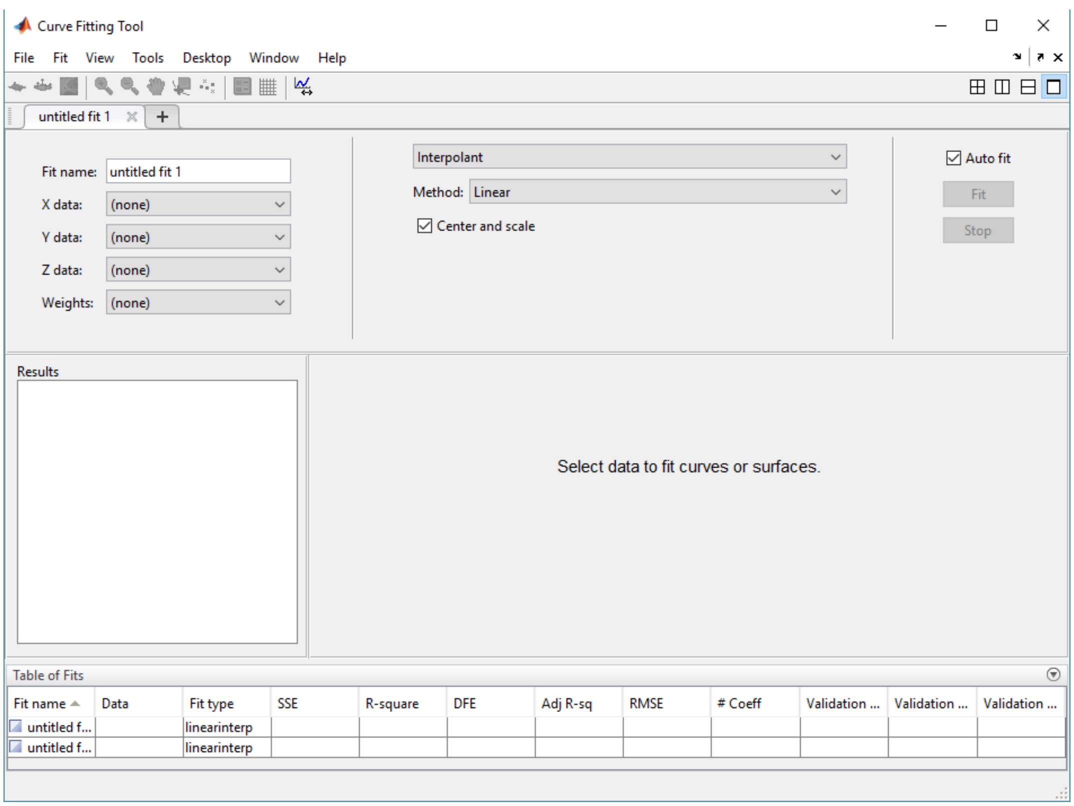
٤-تقريب لاقتران اسي من الدرجة الثانية

١.٦. التقريب

التقريب هي من العمليات الرئيسية في علم الرياضيات حيث يمكن تقريب الأرقام والمتجهات وغيرها حيث يمكن تقريبها الى دوال متعددة الحدود أو دوال أسية وغيرها لتسهيل تمثيل هذه البيانات بمعادلات كثيرة الحدود. وقد اهتم برنامج الماتلاب بهذه العملية وأفرز لها تطبيقا خاصا داخل البرنامج كما هو موضح في الصورة التالية: -

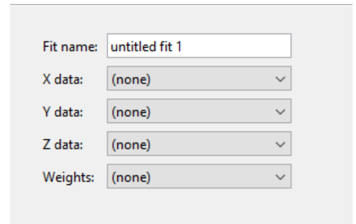


عند تشغيل هذا البرنامج تظهر الشاشة التالية: -

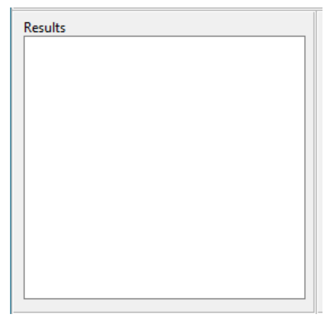


تحتوي هذه الشاشة على عدة عناصر منها: -

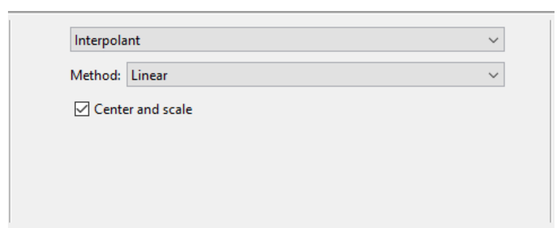
أولا- منطقة تحديد البيانات المراد تقريبها والتي يتم اختيار متجهات البيانات المراد تقريبها منها.



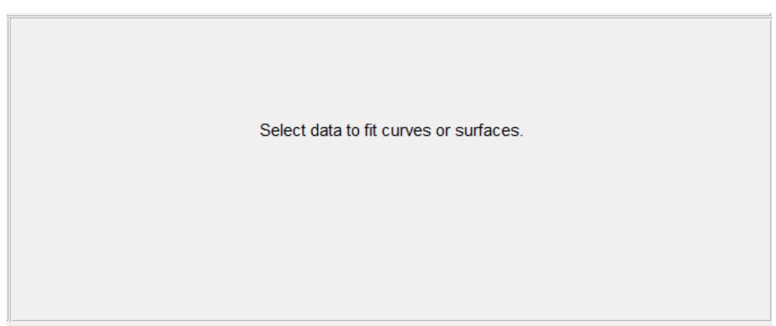
ثانيا- المنطقة التي ستحتوي ناتج التقريب



ثالثا- المنطقة التي يتم اختيار أسلوب التقريب المراد تطبيقه على البيانات وتختلف محتويات هذه المنطقة حسب اختلاف نوع التقريب



رابعا- منطقة الرسم والتي سيظهر فيها رسم بياني للبيانات ومعادلة التقريب



خامسا- المنطقة التي ستظهر فيها التقارير المحتملة للبيانات.

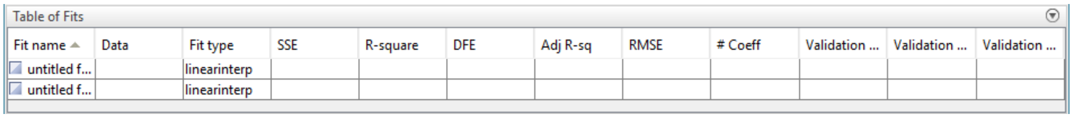


Table of Fits

Fit name	Data	Fit type	SSE	R-square	DFE	Adj R-sq	RMSE	# Coeff	Validation ...	Validation ...	Validation ...
untitled f...		linearinterp									
untitled f...		linearinterp									

٢.٦ المستقيم الموائم

مثال ٣ صفحة ٢٨٠

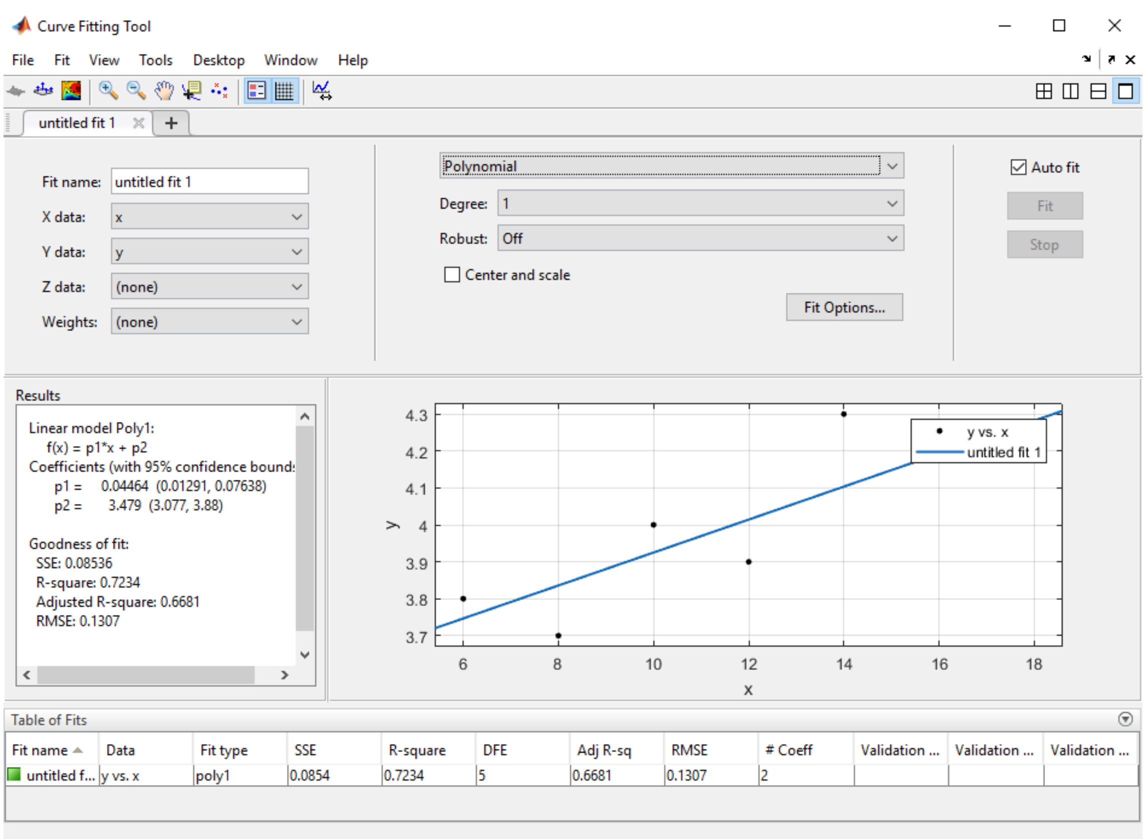
إيجاد المستقيم الموائم باستخدام برنامج مات لاب

أولا- في سطر الأوامر قم بإدخال البيانات كالتالي

x=[6 8 10 12 14 16 18];

y=[3.8 3.7 4.0 3.9 4.3 4.2 4.2];

ثانيا- تشغيل برنامج التقريب في برنامج مات لاب ونقوم بإدخال البيانات اللازمة وبعدها تظهر النتائج أنظر الصورة التالية:-

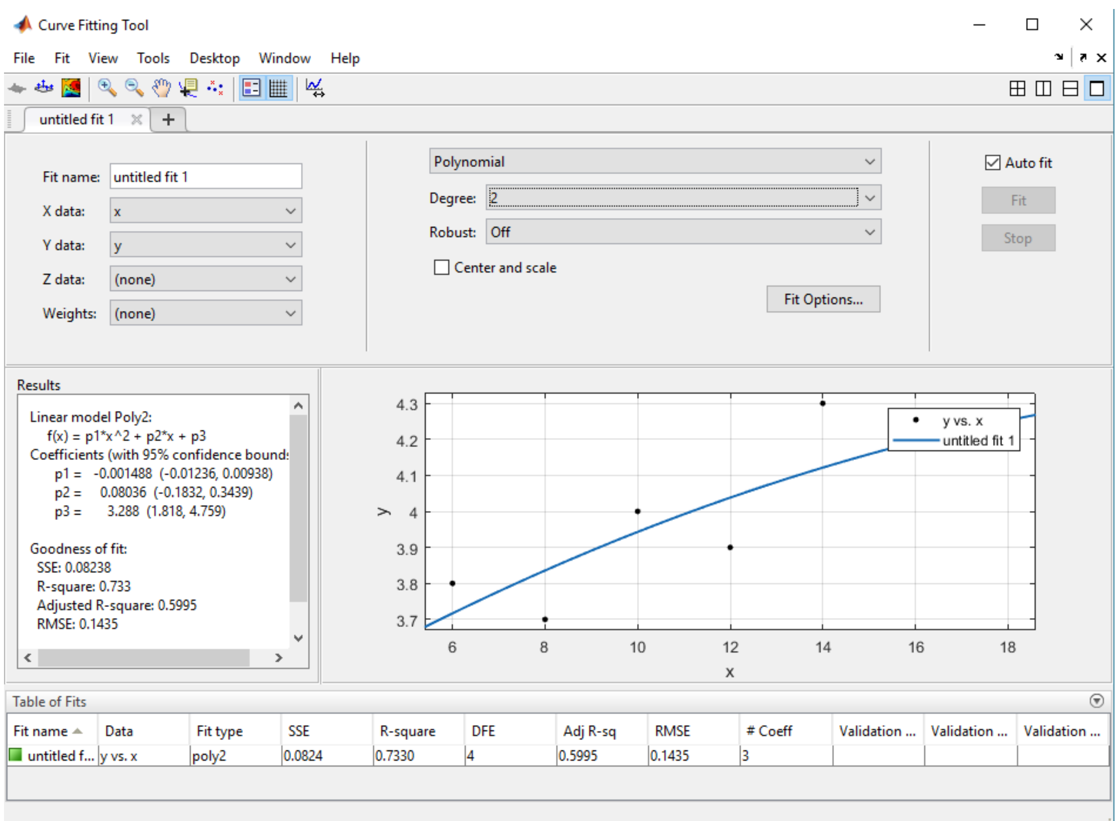


نلاحظ في الشكل بأن التقريب هو تقريب كثيرة حدود من الدرجة الأولى والكود البرمجي له يعطى على الصورة :-

```
function [fitresult، gof] = createFit(x، y)
 [xData، yData] = prepareCurveData( x، y );
ft = fittype( 'poly1' );
[fitresult، gof] = fit( xData، yData، ft );
figure( 'Name'، 'untitled fit 1' );
h = plot( fitresult، xData، yData );
legend( h، 'y vs. x'، 'untitled fit 1'، 'Location'، 'NorthEast' );
xlabel x
ylabel y
grid on
```

٣.٦ تقريب كثيرة حدود

لو أردنا حساب تقريب كثيرة حدود من الدرجة الثانية أو الثالثة نغير في درجة كثيرة الحدود بحيث تصبح الحل للمثال ٤ صفحة ٢٨٢ كالتالي: -



والكود البرمجي يعطى على الصورة

```
function [fitresult، gof] = createFit(x، y)
 [xData، yData] = prepareCurveData( x، y );
```

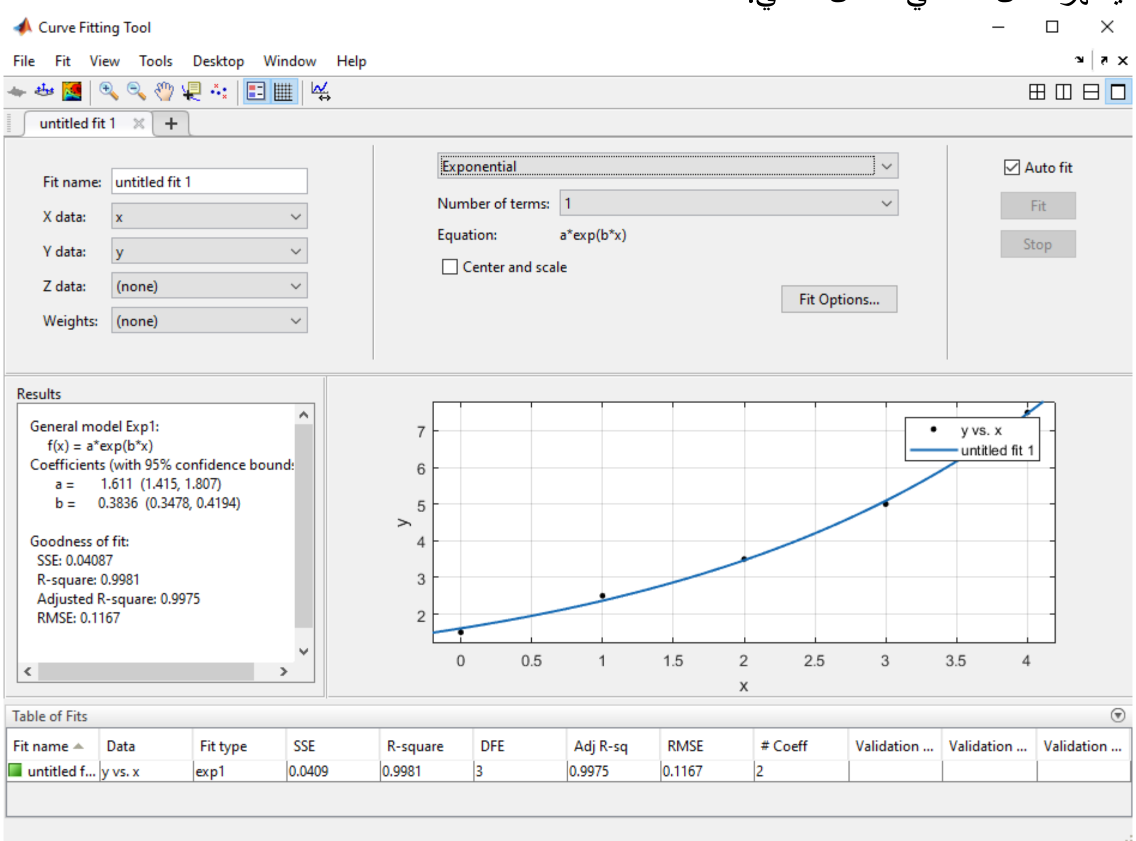
```
ft = fittype( 'poly2' );
 [fitresult، gof] = fit( xData، yData، ft );
figure( 'Name'، 'untitled fit 1' );
h = plot( fitresult، xData، yData );
legend( h، 'y vs. x'، 'untitled fit 1'، 'Location'، 'NorthEast' );
xlabel x
ylabel y
grid on
```

٤.٦. التقريب الأسي

في المثال الخامس صفحة ٢٨٦ يتم استخدام التقريب الأسي حيث يتم عمليا حل هذا المثال كالتالي :- نقوم بإدخال البيانات

x=[0 1 2 3 4];
y=[1.5 2.5 3.5 5.0 7.5];

ثم من برنامج التقريب نختار البيانات التي سيتم تطبيق التقريب عليها ثم نختار نوع التقريب الأسي فيظهر الحل كما في الشكل التالي:-



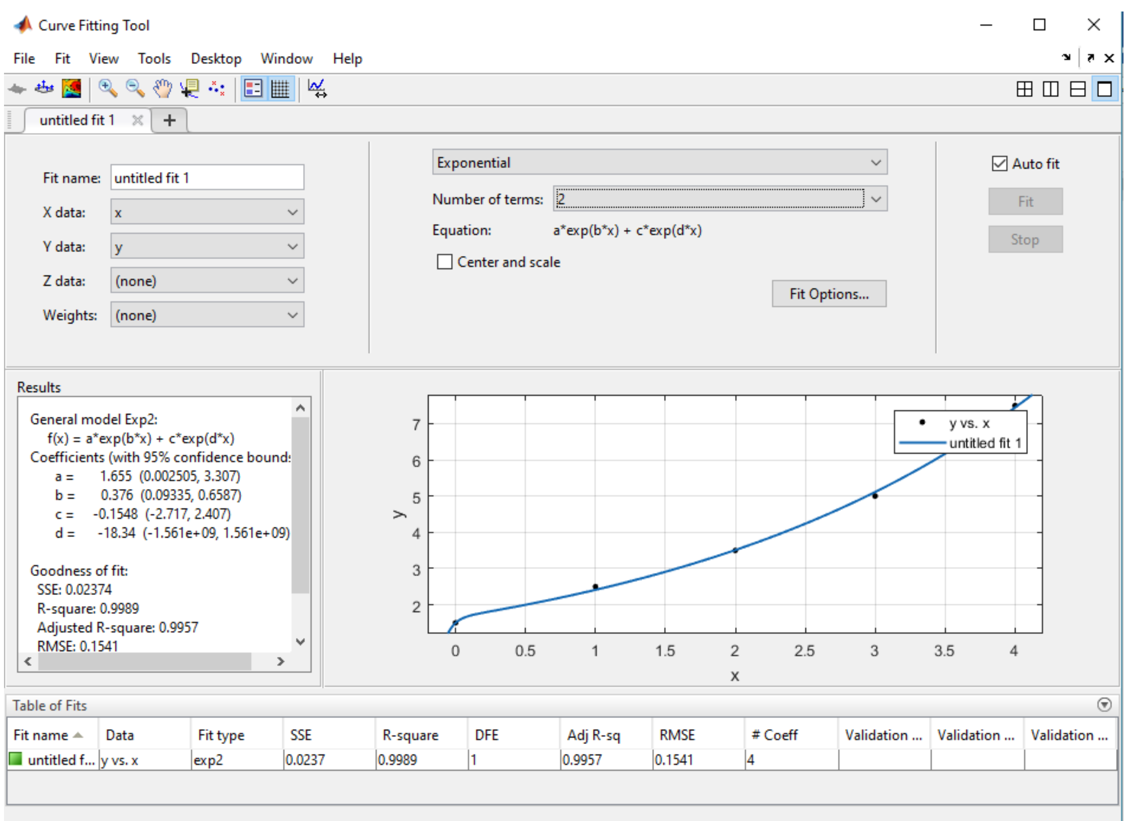
ويكون الكود البرمجي للتقريب الأسى السابق على الصورة التالية :-

```
function [fitresult، gof] = createFit(x، y)
 [xData، yData] = prepareCurveData( x، y );

% Set up fittype and options.
```

```
ft = fittype( 'exp1' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.StartPoint = [1.63987437617133 0.37688590118819];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'untitled fit 1' );
h = plot( fitresult, xData, yData );
legend( h, 'y vs. x', 'untitled fit 1', 'Location', 'NorthEast' );
% Label axes
xlabel x
ylabel y
grid on
```

٥.٦. تقريب لاقتران اسي من الدرجة الثانية

و لإيجاد تقريب لاقتران اسي من الدرجة الثانية نزيد عدد الإقترانات كما في الشكل التالي



٧. الوحدة السابعة

الاستكمال

١. حدودية لاجرانج
٢. حدودية الفروق المقسومة
٣. حدودية نيوتن للفروق المتقدمة
٤. حدودية نيوتن للفروق الرجعية

١.٧. حدودية لاجرانج

جد حدودية لاجرانج للاستكمال الداخلي للجدول التالي (تدريب ٢ صفحة ٣٥٠) :

X	3	4	5	6
Y	3	5	7	9

البرنامج التالي في الماتلاب لإيجاد حدودية لاجرانج لأي عدد N من الأزواج المرتبة، حيث يتم إدخال قيم المتغير X في متجه وقيم المتغير Y في متجه أخر لهما نفس البعد ومن ثم يتم حساب القيمة التقديرية ل Y عند قيمة معينة ل X كالتالي :

```
X=[3 4 5 6];
Y=[3 5 7 9];
N=length(X)-1;
Xp=4.3;%approximate Y when X=4.3
sm=0;
for i=1:N+1
pr=1;
for j=1:N+1
if j~=i
pr=pr*(Xp-X(j))/(X(i)-X(j));
end
end
sm=sm+Y(i)*pr;
end
Yp=sm
```

عند التنفيذ يعطي النتيجة 5.6

٢.٧. ايجاد الحدودية باستخدام الفروق المقسومة

البرنامج التالي في الماتلاب لإيجاد الحدودية المطلوبة باستخدام الفروق المقسومة لأي عدد N من الأزواج المرتبة، حيث يتم إدخال قيم المتغير X في متجه وقيم المتغير Y في متجه أخر لهما نفس البعد ومن ثم يتم حساب القيمة التقديرية ل Y عند قيمة معينة ل X كما في المثال التالي : (مثال ١١ صفحة ٣٦٥)

X	-1	0	2	3
Y	4	1	0	2

```
function F = divided_diff(x,y,x0)

%getting the number of points from the x-vector
```

```
n = size(x،1);
if n == 1
   n = size(x،2);
end


%the 1st column in the divided differences table
for i = 1:n
   F(i،1) = y(i);
end

%the rest of the entries in the table
for i = 2:n
   for j = 2:i
      F(i،j)=(F(i،j-1)-F(i-1j،-1))/(x(i)-x(i-j+1));
   end
end


%evaluating the polynomial at the specified point
fx0 = F(n،n);
for i = n-1:-1:1
   fx0 = fx0*(x0-x(i)) + F(i،i);
end

%command window outputs
disp('Point x0 where approximation of f(x0) is needed')
x0
disp('Evaluation of the polynomial at the specified point yields')
fx0
disp('Divided-differences table')
```

في شاشة الاوامر يتم ادخال

```
x=[-1 0 2 3];
y=[4 1 0 2];
x0=1;
divided_diff(x،y،x0)
```

فتظهر النتائج :

```
Command Window
>> x=[-1 0 2 3];
y=[4 1 0 2];
x0=1;
divided_diff(x,y,x0)
Point x0 where approximation of f(x0) is needed

x0 =

     1

Evaluation of the polynomial at the specified point yields

fx0 =

   -0.3333

Divided-differences table

ans =

    4.0000         0         0         0
    1.0000   -3.0000         0         0
         0   -0.5000    0.8333         0
    2.0000    2.0000    0.8333         0

fx >>
```

٣.٧. ايجاد الحدودية باستخدام الفروق المتقدمة لنيوتن:

البرنامج التالي في الماتلاب لإيجاد الحدودية المطلوبة باستخدام الفروق المتقدمة لنيوتن لأي عدد N من الأزواج المرتبة، حيث يتم إدخال قيم المتغير X في متجه وقيم المتغير Y في متجه أخر لهما نفس البعد ومن ثم يتم حساب القيمة التقديرية ل Yعند قيمة معينة ل X كما في المثال التالي : (تدريب ١٥صفحة ٣٧٧)

X	1.00	1.01	1.02	1.03	1.04	1.05
Y	1.0000	.99433	.98840	.98355	.97844	.96530

احسب $P_4(1.018)$ حيث أن النقطة تقع في بداية الفترة

```
clc;
clear all;

x = input('Enter x: ');
```

```
y = input('Enter y: ');

req = input('Enter required x: ');

rows = length(y);
cols = rows+1;
h = x(2) - x(1);

table = zeros(rows، cols);

% add data to the table
for i = 1:rows
    table(i، 1) = x(i);
    table(i، 2) = y(i);
end

% find the deltas
n = 1;
for j = 3:cols
    for i = 1:rows-n
        table(i، j) = table(i+1، j-1) - table(i، j-1);
    end
    n = n +1;
end

% find the y0
p = 0;
for i = 1:rows
   temp = (req-x(i))/h;
   if(( temp > 0 && temp < 1))
        p = temp;
        pi = i;
   end
end

%final answer
answer = table(pi، 2);
r = 1;
for i = 2:rows
    r = r * (p-i+2);
    answer = answer + (r * (table(pi، i+1)/factorial(i-1)));
end

disp(table);

fprintf('Final answer is: %f\n'، answer);
```

تظهر النتيجة كالتالي:

```
Command Window
Enter x: [1.00 1.01 1.02 1.03 1.04 1.05];
Enter y: [1.0000 .99433 .98840 .98355 .97844 .96530];
Enter required x: 1.018
    1.0000    1.0000   -0.0057   -0.0003    0.0013   -0.0027   -0.0037
    1.0100    0.9943   -0.0059    0.0011   -0.0013   -0.0064         0
    1.0200    0.9884   -0.0048   -0.0003   -0.0078         0         0
    1.0300    0.9836   -0.0051   -0.0080         0         0         0
    1.0400    0.9784   -0.0131         0         0         0         0
    1.0500    0.9653         0         0         0         0         0

Final answer is: 0.989570
>>
```

٤.٧ . ايجاد الحدودية باستخدام الفروق الرجعية لنيوتن:

البرنامج التالي في الماتلاب لإيجاد الحدودية المطلوبة باستخدام الفروق الرجعية لنيوتن لأي عدد N من الأزواج المرتبة، حيث يتم إدخال قيم المتغير X في متجه وقيم المتغير Y في متجه أخر لهما نفس البعد ومن ثم يتم حساب القيمة التقديرية ل Yعند قيمة معينة ل X كما في المثال التالي : (تدريب ١٥صفحة ٣٧٧)

X	1.00	1.01	1.02	1.03	1.04	1.05
Y	1.0000	.99433	.98840	.98355	.97844	.96530

احسب $P_4(1.043)$ حيث نجد ان النقطة تقع في نهاية الفترة

```
clc;
clear all;

x = input('Enter x: ');
y = input('Enter y: ');

req = input('Enter required x: ');

rows = length(y);
cols = rows+1;
h = x(2) - x(1);

table = zeros(rows، cols);

% add data to the table
for i = 1:rows
    table(i، 1) = x(i);
```

```
    table(i, 2) = y(i);
end

% find the deltas
n = 1;
for j = 3:cols
    for i = 1:rows-n
        table(i, j) = table(i+1, j-1) - table(i, j-1);
    end
    n = n +1;
end

% find the y0
p = 0;
for i = 1:rows
   temp = (req-x(i))/h;
   if(( temp > 0 && temp < 1))
        p = (req-x(i+1))/h;
        pi = i+1;

   end
end

%final answer
answer = table(pi, 2);

r = 1;
n = 1;
for i = 2:rows
    r = r * (p+i-2);
    if(pi-n < 1)
        break;
    end
    answer = answer + (r * (table(pi-n, i+1)/factorial(i-1)));
    n = n + 1;
end

disp(table);

fprintf('Final answer is: %f\n', answer);
```

في شاشة الاوامر تظهر النتيجة التالية :

```
Command Window
Enter x: [1.00 1.01 1.02 1.03 1.04 1.05];
Enter y: [1.0000 .99433 .98840 .98355 .97844 .96530];
Enter required x: 1.043;
    1.0000    1.0000   -0.0057   -0.0003    0.0013   -0.0027   -0.0037
    1.0100    0.9943   -0.0059    0.0011   -0.0013   -0.0064         0
    1.0200    0.9884   -0.0048   -0.0003   -0.0078         0         0
    1.0300    0.9836   -0.0051   -0.0080         0         0         0
    1.0400    0.9784   -0.0131         0         0         0         0
    1.0500    0.9653         0         0         0         0         0

Final answer is: 0.975928
fx >>
```

٨. الوحدة الثامنة

التكاملات العددية

١-قاعدة منتصف الفترة

٢-طريقة شبه المنحرف

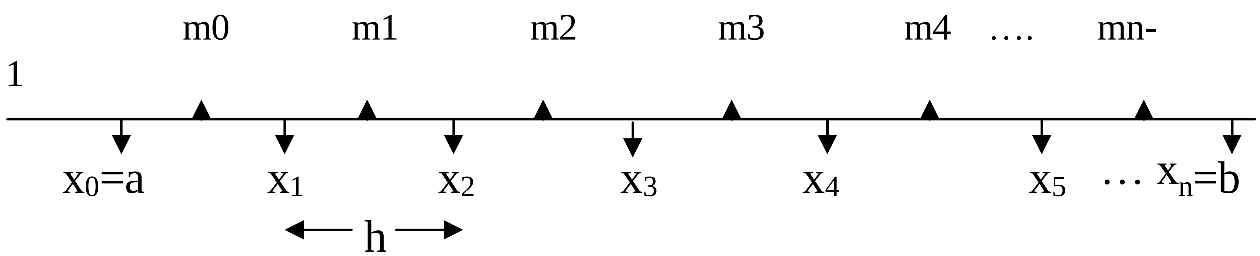
٣-طريقة سمبسون

ليس من السهل حساب بعض التكاملات لذلك نلجأ إلى طرق عددية لحساب هذه التكاملات مثل طريقة منتصف الفترة وطريقة شبه المنحرف وطريقة سمبسون. وبما أن الحل العددي الناتج عن استخدام أي طريقة هو قيمة تقريبية للحل الدقيق لذلك سيحتوي هذا الحل على أخطاء من المهم قياسها.

١.٨. قاعدة منتصف الفترة: Midpoint Rule

لنفرض ان لدينا الاقتران $f(x)=\sin x$ والمطلوب حساب المساحة بين هذا الاقتران ومحور السينات للفترة المحصورة بين 0 و π لذلك سيتم تقسيم الفترة الي عدة أجزاء بمعني أخر تقسيم المساحة تحت المنحني الي عدة شرائح عمودية متساوية العرض عددها n كما هو موضح في الشكل (١) .

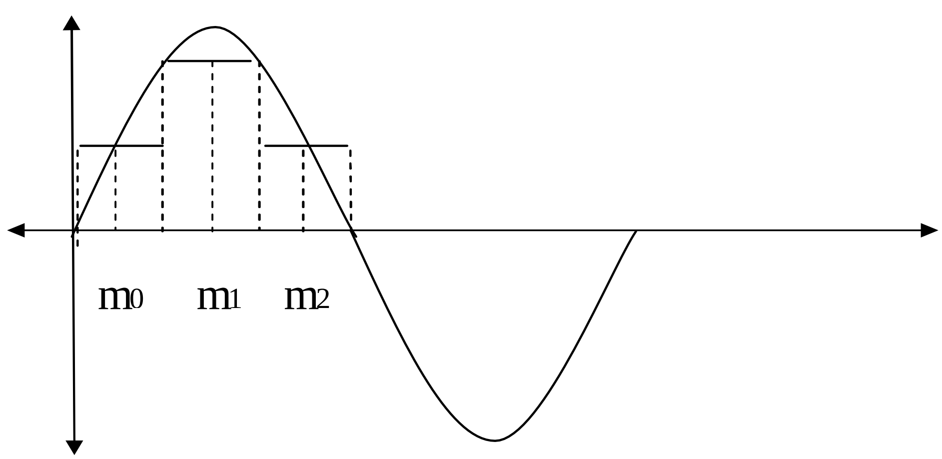
$a=0$ ، $b=\pi$ ، $h=\frac{b-a}{n}$ ، $m_0=a+\frac{1}{2}h$ $\quad m_1=a+\frac{3}{2}h$ $\quad \dots \quad m_n=a+\frac{n+1}{2}h$



الشكل رقم (١)

فاذا كان عدد الشرائح n فان قيم m تكون كما هي موضحة في الشكل رقم (٢)

$$h=\frac{b-a}{n} \qquad m_0=a+\frac{1}{2}h \qquad m_1=a+\frac{3}{2}h \qquad \dots \qquad m_n=a+\frac{n+1}{2}h$$



الشكل رفم (٢)

المساحة تحت منحني اقتران $f(x)=sinx$ مقسمة الي عدة شرائح عمودية

ولإيجاد المساحة الكلية المحصورة بين الاقتران ومحور السينات علينا ايجاد حاصل جمع المستطيلات المرسومة تحت المنحني. علما بأنه كلما زاد عدد الشرائح (المستطيلات) كانت المساحة اقرب الي الحقيقة أو أكثر دقة.

$$A = hf(m_0) + hf(m_1) + hf(m_2) + hf(m_3) + \dots\dots\dots + hf(m_{n-1})$$

or :

$$A = h\sum_{i=0}^{n-1} f(m_i)$$

مثال(١)

استخدم قاعدة منتصف الفترة والتجزئة S_6 لإيجاد قيمة التكامل $\int_0^{\pi} sinx\, dx$.

الحل:

$$h = \frac{b-a}{6} = \frac{\pi-0}{6} = \frac{\pi}{6} \quad ، \quad x_j = a + jh$$

$$x_0 = 0 ، \quad x_1 = \frac{\pi}{6} ، x_2 = \frac{2\pi}{6} ، x_3 = \frac{3\pi}{6} ، \quad x_4 = \frac{4\pi}{6} ، x_5 = \frac{5\pi}{6} ، x_6 = \frac{6\pi}{6}$$

$$m_0 = \frac{\pi}{12} ، \quad m_1 = \frac{3\pi}{12} ، m_2 = \frac{5\pi}{12} ، m_3 = \frac{7\pi}{12} ، \quad m_4 = \frac{9\pi}{12} ، m_5 = \frac{11\pi}{12}$$

N	m_n	$f(m_n)$	$hf(m_n)$
0	π\12	0.2588	0.1355
1	3π\12	0.7071	0.3702
2	5π\12	0.9659	0.5057
3	7π\12	0.9659	0.5057

4	9π\12	0.7071	0.3702
5	11π\12	0.2588	0.1355
			Total = 2.0228

R= **2.0228** the approximated value.

نستخدم برنامج (Matlab) في حساب قيمة التكامل كما يلي:

$a = 0\,;\quad b = \mathrm{pi}\ \ ;\ n = 6\,;\ \ h = (b - a)/n;$
$m = \mathrm{pi}/12 : pi/6 : 11 * pi/12\ \ ;$
$y = \sin(m)\,;$
$R = h * \mathrm{sum}(y)$

The result is R = **2.0230**

مثال (٢)

استخدم قاعدة منتصف الفترة والتجزئة S_6 لإيجاد قيمة التكامل $\int_0^3 e^x\, dx$.

الحل:

$$h = \frac{b-a}{6} = \frac{3-0}{6} = 0.5$$

$x_0 = 0$ ، $x_1 = 0.5$ ، $x_2 = 1$، $x_3 = 1.5$، $x_4 = 2$، $x_5 = 2.5$ ، $x_6 = 3$

$m_0 = 0.25$ ، $m_1 = 0.75$ ، $m_2 = 1.25$ ، $m_3 = 1.75$ ، $m_4 = 2.25$ ، $m_5 = 2.75$

N	***m_n***	***$f(m_n)$***	***$hf(m_n)$***
0	0.25	1.2840	0.6420

1	0.75	2.1170	1.0585
2	1.25	3.4903	1.7452
3	1.75	5.7546	2.8773
4	2.25	9.4877	4.7439
5	2.75	15.6426	7.8213
			Total = 18.8882

R= 18.8882 the approximated value.

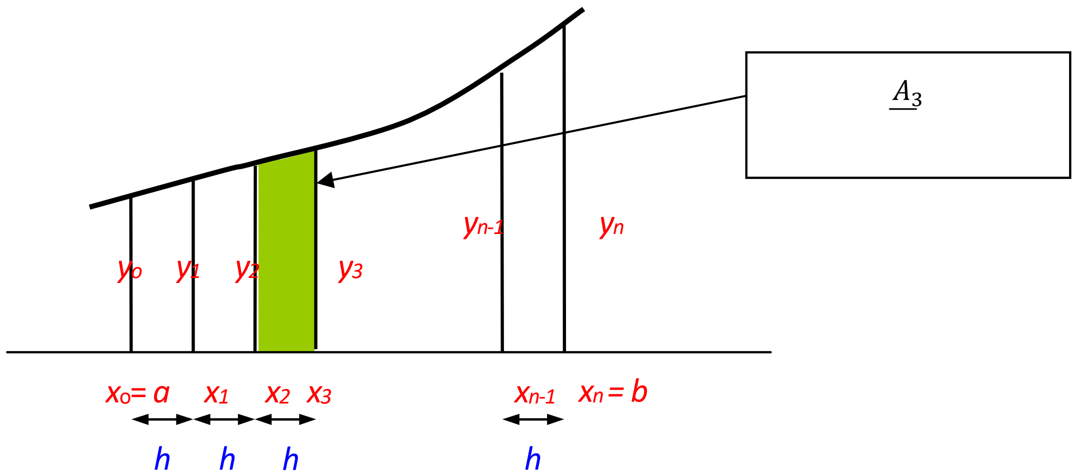
نستخدم برنامج (Matlab) في حساب قيمة التكامل كما يلي:

$a = 0$; $b = 3; n = 6$; $h = (b - a)/n$;
m = 0.25 : 0.5 : 2.75 ;
$y = \exp(m)$;
R = h ∗ sum(y)

The result is: R = 18.8882

٢.٨. طريقة شبه المنحرف Trapezoidal Rule

وفي هذه الطريقة يتم تقسيم المساحة تحت المنحني إلي شرائح عمودية متساوية العرض وليكن عددها يساوى n وعرض الشريحة الواحدة يساوى h حيث $h = \frac{b-a}{n}$. عندها تكون كل شريحة أشبه بشبه منحرف. وتكون مساحة شبه المنحرف الأول هي $A_1 = \frac{h}{2}(y_0 + y_1)$ ومساحة شبه المنحرف الثاني $A_2 = \frac{h}{2}(y_1 + y_2)$ وهكذا كما هو موضح في شكل (3).



شكل (3)

والآن يمكننا حساب المساحة التقريبية تحت المنحني وذلك من مجموع الشرائح العمودية تحت المنحني كما يلي:

$$\int_{a}^{b} \boldsymbol{f(x)dx} = \frac{h}{2}(y_0 + y_1) + \frac{h}{2}(y_1 + y_2) + \ \ldots + \ \frac{h}{2}(y_{n-1} + y_n)$$

و عند إجراء التبسيط لهذه المعادلة سيكون

$$\int_{a}^{b} \boldsymbol{f(x)dx} = \frac{b-a}{2n}\ (\ y_0\ +\ 2y_1\ + \cdots +\ 2y_{n-1} + y_n)$$

مثال (١)

استخدم قاعدة شبه المنحرف والتجزئة S_6 لإيجاد قيمة التكامل $\boldsymbol{\int_0^3 e^x\, dx}$.

الحل:

$y = e^x$ ، $a = 0$ ، $b = 3$ ، $n = 6$

$$h = \frac{b-a}{n} = \frac{3-0}{6} = 0.5$$

أي أن عرض الشريحة يساوى 0.5 . وبناء على قانون هذه الطريقة يكون:

$y_0 = f(x_0)$ ، $y_1 = f(x_1)$ ، $y_2 = f(x_2)$ ، ….. ، $y_n = f(x_n)$

وعند استخدام الجدول وتسجيل البيانات المطلوبة فيه يسهل علينا حساب التكامل المطلوب. حيث تمثل القيمة m معامل المتغير y في المعادلة الأخيرة المستخدمة في عملية الحساب.

N	x_n	$y_n(=e^x)$	***C***	cy_n
0	0	1	1	1
1	0.5	1.64872	2	3.29744
2	1	2.71828	2	5.43656
3	1.5	4.48169	2	8.96338
4	2	7.38906	2	14.77811
5	2.5	12.18249	2	24.36499
6	3	20.08554	1	20.08554
			Total	**76.92602**

وعليه ستكون قيمة التكامل النهائية تساوى:

$$\int_a^b f(x)dx = \frac{h}{2}\left(y_0 + 2y_1 + \cdots + 2y_{n-1} + y_n\right)$$

$$= \frac{0.5}{2}(76.926) = 19.2315$$

The approximated value is R = 19.2315 .

نستخدم برنامج (Matlab) في حساب قيمة التكامل كما يلي:

```
a = 0;
b = 3;
n = 6;
h = (b-a) / n;
x = 0 : 0.5 : 3
y = exp(x) ;
R=(h/2)*(y(1)+2*y(2)+2*y(3)+2*y(4)+2*y(5)+2*y(6)+y(7))
```

The result is : R = 19.4815

مثال (٢)

استخدم قاعدة شبه المنحرف والتجزئة S_4 لإيجاد قيمة التكامل $\int_0^{\frac{\pi}{2}} x\, cosx\, dx$.

الحل:

$$a = 0 \quad ، \quad b = \frac{\pi}{2} \quad ، \quad n = 4$$

$$h = \frac{b-a}{n} = \frac{\frac{\pi}{2}-0}{4} = \frac{\pi}{8} = 0.3927 \; ، \quad x_j = a + jh$$

N	x_n	$y_n = x\cos x$	*c*	cy_n
0	0	0	1	0
1	π/8	0.3628	2	0.7256
2	π/4	0.5554	2	1.1108
3	3π/8	0.4508	2	0.9016
4	π/2	0	1	0
			Total	**2.7380**

$$\int_0^{\frac{\pi}{2}} x\, cosx\, dx = \frac{h}{2}\,(y_0 + 2y_1 + \cdots + 2y_{n-1} + y_n)$$

$$= \frac{\frac{\pi}{8}}{2}\,(2.7380) = 0.5369$$

The approximated value is R = 0.5369 .

نستخدم برنامج (Matlab) في حساب قيمة التكامل كما يلي:

```
a = 0  ;
b = pi / 2  ;
n = 4  ;
h = (b – a) /n   ;
x = 0 : pi/8 : pi/2   ;
y = x.*cos x ;
R = (h/2) * ( y(1) + 2*y(2) + 2*y(3) 2*y(4) + y(5))
```

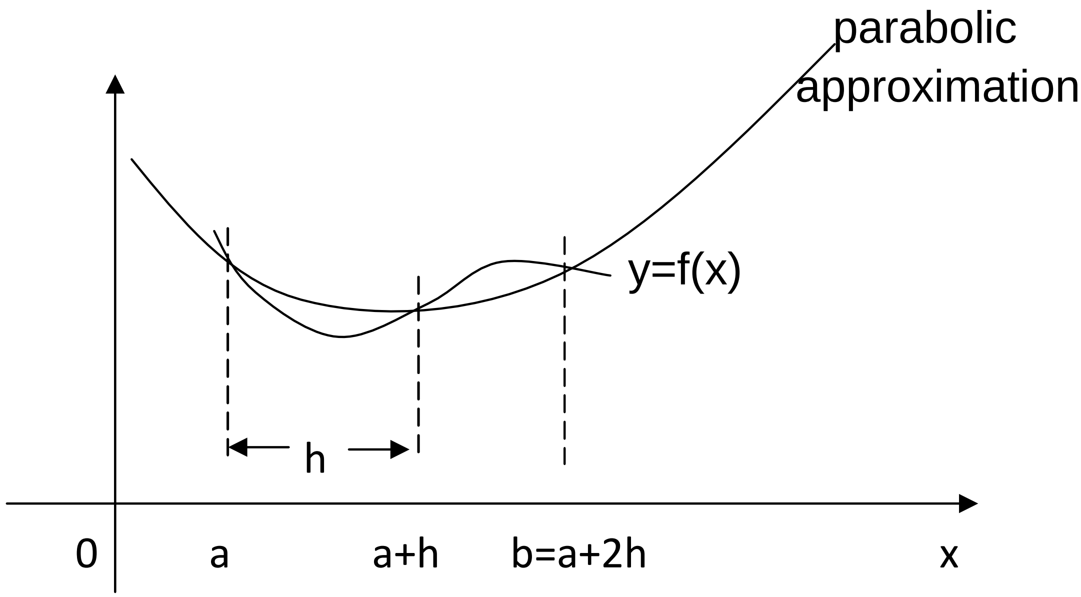
The result is R = 0.5376

٣.٨. طريقة سمبسون Simpson Rule

وفي هذه الطريقة يتم تقسيم المساحة تحت المنحني إلي شرائح عمودية متساوية العرض وليكن عددها يساوى 2m وعرض الشريحة الواحدة يساوى h حيث $h = \frac{b-a}{2m}$. وكل شريحة تتكون من شريحتين عموديتين متساويتين وكل شريحة من هاتين الشريحتين تكون أشبه بشبه المنحرف حيث من الممكن حساب مساحتهما التقريبية. وقلنا تقريبية لان الشكل ليس شبه منحرف تماما بل أشبه بذلك، وذلك لأن حده الأعلى خط منحني وهو جزء من منحني الدالة المراد إيجاد المساحة تحتها. وسيمر منحني الدالة بثلاث نقاط ضمن الشريحة الاصلية وهذه النقاط هي:

$$\big(a \text{ ، } f(a)\big) \text{ ، } \big(a+h \text{ ، } f(a+h)\big) \text{ ، } \big(a+2h \text{ ، } f(a+2h)\big)$$

حيث تمثل القيمة h عرض احدى الشريحتين كما هو موضح في الشكل التالي:



شكل (٤)
الشرائح العمودية تحت المنحني

والآن يمكننا حساب المساحة التقريبية تحت المنحني وذلك من مجموع الشرائح العمودية تحت المنحني **كما يلي:**

$$\int_a^b \boldsymbol{f(x)dx} \cong \frac{h}{3}\left[f(x_0) + 4f(x_1) + \boldsymbol{2f(x_2)} + 4f(x_3) + \cdots + \boldsymbol{2f(x_{2m-2})} + \boldsymbol{4f(x_{2m-1})} + \boldsymbol{f(x_{2m})} \right]$$

$$\boldsymbol{h} = \frac{\boldsymbol{b-a}}{\boldsymbol{2m}} \quad ، \quad x_j = a + jh \ ، \ j = 0،1،2، \dots ،2m$$

مثال (١)

استخدم قاعدة سمبسون والتجزئة S_6 لإيجاد قيمة التكامل $\int_0^3 e^x\, dx$.

الحل:

a=0 ، b=3 ، h=0.5

$X_0=0$ $x_1=0.5$ $x_2=1$ $x_3=1.5$ $x_4=2$ $x_5=2.5$
$x_6=3$

شكل (٥)

$$\int_0^3 e^x \, dx = \frac{1}{6}[\, e^0 + 4e^{0.5} + 2e^1 + 4\, e^{1.5} + 2e^2 + 4e^{2.5} + e^3 \,]$$

$$= 19.092$$

The approximated value is R= 19.092

ان النتيجة المحسوبة بطريقة سمبسون وهي (19.092) هي أقرب للقيمة الدقيقة وهي (19.0855). لذلك تعتبر هذه الطريقة أكثر دقة من طريقة شبه المنحرف ومن طريقة التنصيف أيضا. ويمكن حساب قيمة الخطأ وفق العلاقة:

$$e = |19.0855 - 19.092| = 0.0065$$

مع العلم أنه كلما زاد عدد الشرائح قل الخطأ.

نستخدم برنامج (Matlab) في حساب قيمة التكامل كما يلي:

```
a = 0 ;
b = 3  ;
n = 6  ;
h = (b-a) / n;
x = 0 : 0.5 : 3
y = exp(x) ;
R=(h/3)*(y(1)+4*y(2)+2*y(3)+4*y(4)+2*y(5)+4*y(6)+y(7))
```

The result is : R = 19.0920

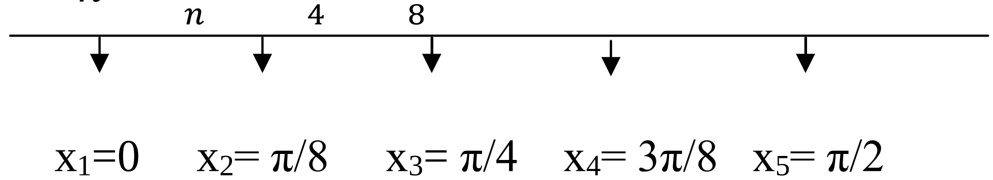
مثال (٢)

استخدم قاعدة سمبسون والتجزئة S_4 لإيجاد قيمة التكامل $\int_0^{\frac{\pi}{2}} x\, cosx\, dx$. اعطي اجابتك مقربة لأربع منازل عشرية.

الحل:

$$a = 0 \quad ، \quad b = \frac{\pi}{2} \quad ، \quad n = 4$$

$$h = \frac{b-a}{n} = \frac{\frac{\pi}{2}-0}{4} = \frac{\pi}{8}$$



شكل (٦)

$$\int_0^{\frac{\pi}{2}} \boldsymbol{x\, cosx\, dx} \cong \frac{h}{3}\, [\, f(x_1) + 4f(x_2) + \boldsymbol{2f(x_3)} + 4f(x_4) + \cdots + \boldsymbol{2f(x_{n-2})} + \boldsymbol{4f(x_{n-1})} + \boldsymbol{f(x_n)}\ ]$$

$$= \frac{\boldsymbol{\pi}}{\boldsymbol{24}}\, (\boldsymbol{0 + 1.4512 + 1.1107 + 1.8034 + 0}) = \boldsymbol{0.5714}$$

The approximated value is R = $\boldsymbol{0.5714}$.

وهنا نلاحظ أن القيمة المحسوبة بهذه الطريقة وهي (0.5714) قريبة جدا للقيمة الدقيقة التي تم حسابها من اجراء التكامل وتساوى (0.5708) . وهذه يؤكد ان هذه الطريقة أفضل من الطرق السابقة.

نستخدم برنامج (Matlab) في حساب قيمة التكامل كما يلي:

```
a = 0  ;
b = pi / 2  ;
n = 4  ;
h = (b – a) /n   ;
x = 0 : pi/8 : pi/2   ;
y = x.*cos x ;
R = (h/3) * ( y(1) + 4*y(2) + 2*y(3) + 4*y(4) + y(5) )
```

The result is R = 0.5714

٤.٨. الاستنتاجات:

لأجل مقارنة النتائج المحسوبة بهذه الطرق نكتبها على شكل جدول حتى يتسنى لنا ملاحظة الفرق بينها من حيث تقارب القيم التقريبية مع القيم الحقيقية.

الاقتران	القيم الحقيقية	القيم التقريبية		
		شبه المنحرف	سمبسون	منتصف الفترة
$\int_0^3 e^x\,dx$	19.0855	19.4815	19.092	18.8882
$\int_0^{\frac{\pi}{2}} x\cos x\,dx$	0.6708	0.5376	0.5714	0.5874

نلاحظ من الجدول أعلاه أن طريقة سمبسون هي الأفضل. كما أن زيادة عدد الشرائح المأخوذة يؤدى الي زيادة الدقة في حساب المساحات تحت المنحنى.