جَامعة المَنارة
MANARA UNIVERSITY

الجلسة العملية الخامسة

عنوان الجلسة: ايجاد جذور التوابع

الغاية من الجلسة :

دراسة خوارزميات ايجاد جذور التوابع بالطرق العددية و تحويلها الى أكواد برمجية بلغة البايثون

Bisection Method Algorithm

1-خوارزمية تنصيف المجالات

- **طريقة تنصيف المجالات**: في الرياضيات، طريقة التنصيف هي إحدى طرق ايجاد الجذر والتي يتم بها تنصيف المجال الذي يقع فيه الجذر بصورة تكرارية واختيار مجال فرعي منه يقع فيه الجذرأيضا من أجل تحسين المعالجة.

مع أنها بسيطة جدا ومرنة إلا أن طريقة التنصيف بطيئة نسبيا.

الخوارزمية:

إذا كانت الدالة f(x)= 0 مستمرة ومعرفة في الفترة [a,b] حيث f(a)* f(b) < 0 أي أنهما مختلفتان في الإشارة فإن معنى ذلك ان أحد جذور الدالة f(x) على الأقل يقع في نطاق الفترة [a,b] . في هذه الحالة نتبع الخوارزمية التالية للوصول إلى حل هذه الدالة:

1- اختبر اذا كان المجال مجال الحل

1-1- نوجد متوسط القيمتين a,b المعرف عندهما الدالة وليكن x1 حيث أن x1=(a+b)/2.

1-2- إذا كانت قيمة f(x1)<tol فإن x1 جذر للدالة f(x) وحلا لها.

1-3- والا إذا كانت f(x1) * f(b)< 0 فإننا نضع a = x1 لنقترب من الحل.

1-4- و الا إذا كانت f(x1) * f(a)< 0 فإننا نضع b = x1 لنقترب من الحل.

2 - نقوم بتكرار الخطوات 1و 2 و 3 و4 حتي نصل لقيمة تكون فيها f(xi)= 0 أو تكون فيها f(xi) < tol حيث أن tol تمثل درجة الدقة المطلوبة في الحل.

2- والا ليس لدينا حل في هذا المجال

التمرين الأول:

بما أن الخوارزمية هي خوارزمية تكرارية يمكن حل التمرين باستخدام الحلقات او باستخدام التوابع العودية

لا حظ أن نص التمرين يطلب الحل بشكل تابع

استخدم بايثون:

1- لكتابة تابع لايجاد الجذر التقريبي لتابع معطى مستخدما خوارزمية تنصيف المجال

1-استخدم التابع السابق لايجاد الجذور التقريبية للمعادلة $f(x)=x^3-9x+1$ على المجال [2,4]

بدقة $\varepsilon=0.5$. و اطبع على الشاشة قيمة التابع f عند هذا الجذر

الأدوات اللازمة لحل التمرين:

1-كتابة تابع يتم استدعاؤه في كل مرة من أجل تابع جديد. (تعرفنا إليه سابقا)

2-أدوات الشرط في python

3-استخدام Lambda لتعريف التابع المطلوب

2- أدوات الشرط في python:

الأدوات اللازمة لحل التمرين:

In Python, there are three forms of the `if...else` statement.

1. `if` statement
2. `if...else` statement
3. `if...elif...else` statement

2-أدوات الشرط في python

الأدوات اللازمة لحل التمرين:

Python Nested if statements

We can also use an `if` statement inside of an `if` statement. This is known as a nested if statement.

The syntax of nested if statement is:

```
# outer if statement
if condition1:
    # statement(s)
    # inner if statement
    if condition2:
    # statement(s)
```

Notes:

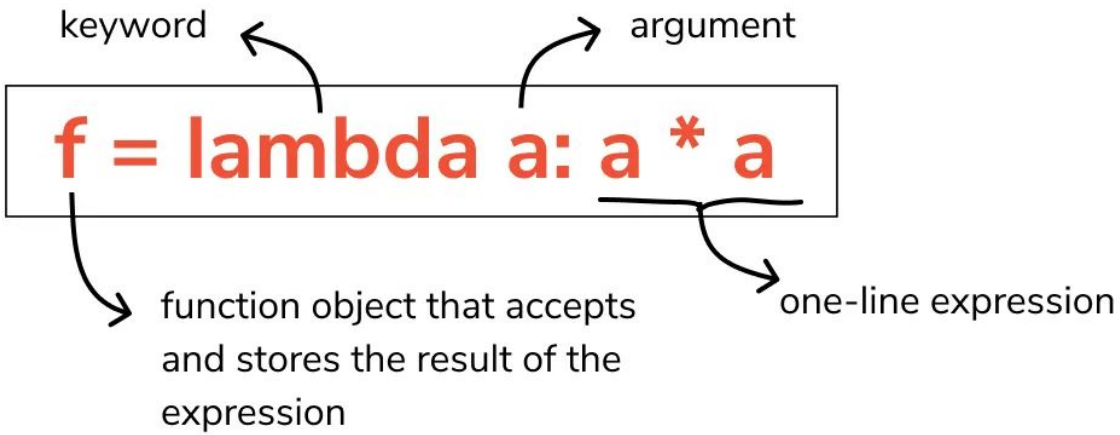
- We can add `else` and `elif` statements to the inner `if` statement as required.
- We can also insert inner `if` statement inside the outer `else` or `elif` statements(if they exist)
- We can nest multiple layers of `if` statements.

الأدوات اللازمة لحل التمرين:

في لغة بايثون ، تعد Lambda function تابعا مجهولا(anonymous) ، مما يعني أنها تابع بدون اسم.

- يمكن أن تحتوي على أي عدد من الوسطاء ولكن تعبير واحد فقط يتم تقييمه وإرجاعه.
- ليس من الضروري أن يكون لها قيمة معادة

استخدام Lambda في python:



```python
import numpy as np
def my_bisection(f, a, b, tol):
        # check if a and b bound a root
    if f(a)* f(b)<0:
       # get midpoint
        mid = (a + b)/2
        if np.abs(f(mid)) < tol:
         # stopping condition, report m as root
              return mid
        elif f(a)*f(mid)<0:
         # case where mid is an improvement on a.
         # Make recursive call with a = m
            return my_bisection(f, a, mid, tol)
        elif f(b)*f(mid)<0:
         # case where mid is an improvement on b.
         # Make recursive call with b = m
            return my_bisection(f, mid, b, tol)
    else:
       print("No Solution")
```

الكود البرمجي المتعلق بكتابة التابع:

لاحظ أن هذا التابع هو تابع عودي يطلب نفسه من داخل جسم التابع

Using `my_bisection` function

```
f = lambda x: x**3 - 9*x + 1

r1 = my_bisection(f, 2, 4, 0.5)
print("r1 =", r1)
print("f(r1) =", f(r1),"\n")

r2 = my_bisection(f, 2, 4, 0.05)
print("r2 =", r2)
print("f(r2) =", f(r2),"\n")



r3 = my_bisection(f, 2, 4, 0.0005)
print("r3 =", r3)
print("f(r3) =", f(r3))
```

جامعة المنارة
MANARA UNIVERSITY

الكود البرمجي المتعلق باستخدام التابع السابق ,من أجل معرفة جذر التابع :

f(x)=X3-9*x+1**

على المجال [2,4]

وبدقة ε=0.05 وبدقة ε=0.05

بدقةε=0.0005

OUTPUT

```
r1 = 2.9375
f(r1) = -0.090087890625

r2 = 2.9453125
f(r2) = 0.04237794876098633

r3 = 2.94281005859375
f(r3) = -0.0001697912650797661 6
```

لاحظ أن قيمة السماحية او الدقة تؤثر على قيمة الجذرr و قيمة التابع f .

خرج الكود البرمجي المتعلق باستخدام التابع السابق ,من أجل معرفة جذر التابع :

f(x)=X3-9*x+1**

على المجال [2,4]

بدقة ε=0.5

بدقة ε=0.05

بدقةε=0.0005

الكود البرمجي المتعلق باستخدام التابع السابق ,من أجل معرفة جذر التابع :

$$f(x)=X**3-9*x+1$$

على المجال [3,4]

بدقة $\varepsilon=0.5$

```
f = lambda x: x**3 - 9*x + 1


r1 = my_bisection(f, 3, 4, 0.5)
print("r1 =", r1)
print("f(r1) =", f(r1),"\n")
```

OUTPUT

```
No Solution
r1 = None

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-11-7ae2f9b8cd4b> in <cell line: 5>()
      3 r1 = my_bisection(f, 3, 4, 0.5)
      4 print("r1 =", r1)
----> 5 print("f(r1) =", f(r1),"\n")

<ipython-input-11-7ae2f9b8cd4b> in <lambda>(x)
----> 1 f = lambda x: x**3 - 9*x + 1
      2
      3 r1 = my_bisection(f, 3, 4, 0.5)
      4 print("r1 =", r1)
      5 print("f(r1) =", f(r1),"\n")

TypeError: unsupported operand type(s) for ** or pow():
'NoneType' and 'int'
```

جامعة المنارة
MANARA UNIVERSITY

خرج الكود البرمجي المتعلق باستخدام التابع السابق ,من أجل معرفة جذر التابع :

f(x)=X**3-9*x+1

على المجال [3,4]

بدقة $\varepsilon=0.5$

Newton_Raphson Algorithm

2-خوارزمية نيوتن رافسون

خوارزمية تكرارية تعتمد على الانطلاق من نقطة تعتبر حلا تقريبيا للتابع للاقتراب من الحل .

الخوارزمية:

إذا كانت الدالة f(x)= 0 معرفة ومستمرة و قابلة للاشتقاق في الفترة [a,b] و لدينا x0 قيمة ابتدائية قريبة من الحل. في هذه الحالة نتبع الخوارزمية التالية للوصول إلى حل هذه الدالة:

1- اختبر اذا كان f(x0)<tol :

1-1 - نعيد قيمة x0 كجذر للتابع

2 -و الا نحسب القيمة :

$$x0=x0-(f(x0)/f'(x0))$$

نكرر الخطوات 1 و 2 حتى نصل الى الحل

التمرين الأول:

بما أن الخوارزمية هي خوارزمية تكرارية يمكن حل التمرين باستخدام الحلقات او باستخدام التوابع العودية

لا حظ أن نص التمرين يطلب الحل بشكل تابع

استخدم بايثون:

1- لكتابة تابع لايجاد الجذر التقريبي لتابع معطى مستخدما خوارزمية نيوتن رافسون

1-استخدم التابع السابق لايجاد الجذر التقريبي للمعادلة $f(x)=x^3-9x+1$ من أجل الحل الابتدائي 1.5

بدقة 1e-6. و اطبع على الشاشة قيمة التابع f عند هذا الجذر .

الكود البرمجي المتعلق بكتابة التابع:

```
import numpy as np
from mpmath import *    #importing differntiation

def my_newton(f, df, x0, tol):
    # output is an estimation of the root of f
    # using the Newton Raphson method
    # recursive implementation
    if abs(f(x0)) < tol:
        return x0
    else:
        return my_newton(f, df, x0 - (f(x0)/df(x0)), tol)
```

الكود البرمجي المتعلق باستخدام التابع السابق ,من أجل معرفة جذر التابع :

f(x)=X**3-9*x+1

على المجال [3,4]

بدقة ε=0.5

```
f = lambda x: x**3-x - 1
df=lambda x:diff(f,x, 1)

newton_raphson = my_newton(f,df,1.5, 1e-6)
print("newton_raphson =", newton_raphson)
print(f(newton_raphson))
```

OUTPUT

```
newton raphson = 1.32471817399905
9.24377759670136e-7
```