Практикум 7. Непрерывность функции

Цель работы **–** изучение понятия непрерывности функции в точке, классификация точек разрыва, использование функции brentq для нахождения нулей функции и корней уравнения, символическое решение уравнений и систем

Продолжительность работы - 2 часа.

Оборудование, приборы, инструментарий – работа выполняется в компьютерном классе с использованием пакета Anaconda.

Порядок выполнения

1. Упражнения выполняются параллельно с изучением теоретического материала.
2. После выполнения каждого упражнения результаты заносятся в отчёт.
3. При выполнении упражнений в случае появления сообщения об ошибке рекомендуется сначала самостоятельно выяснить, чем оно вызвано, и исправить команду; если многократные попытки устранить ошибку не привели к успеху, то проконсультироваться с преподавателем.
4. Дома доделать упражнения из раздела «Краткие теоретические сведения и практические упражнения», которые Вы не успели выполнить во время аудиторного занятия.
5. После выполнения упражнений выполнить дополнительные упражнения для самостоятельной работы и ответить на контрольные вопросы и (см. ниже).
6. Подготовить отчёт**,** в который включить упражнения из раздела «Краткие теоретические сведения и практические упражнения» и упражнения для самостоятельной работы. Отчёт представить в виде документа Microsoft Word, имя файла (пример): mp_10_Ivanov_P_01_s_1 (факультет_группа_Фамилия студента_Инициал_номер лабораторной, семестр). Отчет должен содержать по каждому выполненному упражнению: № упражнения, текст упражнения; команды, скопированные из командного окна, с комментариями к ним и результаты их выполнения, включая построенные графики; тексты python функций; выводы.

Краткие теоретические сведения

и практические упражнения

1. Непрерывность функции в точке и точки разрыва функции.

Определение. Функция $y = f(x)$ называется непрерывной в точке x_0, если $\lim\limits_{x \to x_0} f(x) = f(x_0)$.

Если функция не является непрерывной в точке x_0 области определения функции или функция не определена в этой точке, но определена в некоторой её окрестности, то точка x_0 называется ***точкой разрыва*** функции.

Точки разрыва классифицируются следующим образом:

1) если оба предела $f(x_0+0)$ и $f(x_0-0)$ конечны и $f(x_0+0) = f(x_0-0) \neq f(x_0)$, то такая точка x_0 является точкой устранимого разрыва, причем $f(x)$ может быть и определена, и не определена в точке x_0;

2) если оба предела $f(x_0+0)$ и $f(x_0-0)$ конечны и $f(x_0+0) \neq f(x_0-0)$, то функция $y = f(x)$ имеет в точке x_0 разрыв первого рода;

3) если хотя бы один из односторонних пределов $f(x_0+0)$, $f(x_0-0)$ не существует или бесконечен, то функция $y = f(x)$ имеет в точке x_0 разрыв второго рода.

2. Нули непрерывной функции. Численное решение уравнений

Из курса математического анализа нам известно следующее свойство непрерывной функции:

если $f(a) \cdot f(b) < 0$, то существует точка $x_0 \in (a,b)$ такая, что $f(x_0) = 0$. Это утверждение означает, что график функции $y = f(x)$ непрерывной на отрезке $[a,b]$, хотя бы в одной точке пересекает отрезок ось Ox, если точки $(a, f(a))$ и $(b, f(b))$ лежат по разные стороны от оси Ox.

Найти приближенно точку x_0 можно с помощью функции **brentq()**.

Базовый формат вызова этой функции включает два аргумента и имеет следующий вид:

brentq (fun, a, b)

Параметры a, b представляет собой границы интервала, при этом значения fun должны менять знак на концах интервала, это гарантирует нахождение, по крайней мере, одного корня на этом интервале.

Пример 1.

Нули функции $f(x) = \cos x - \sin x$ на отрезке $[0, \pi / 2]$ можно найти с помощью команды

```
from scipy import optimize
from math import sin , cos, pi

def f(x):
    return (cos(x)-sin(x))

root = optimize.brentq(f, 0, pi/2)
print('x =',root)
print('f(x) =',f(root))
```

```
x = 0.7853981633974484
f(x) = -1.1102230246251565e-16
```

Судя по значению функции точность нахождения нуля функции достаточно высока.

Поскольку brentq не проверяет функцию fun на непрерывность, то применение brentq в некоторых случаях может привести к парадоксальным (на первый взгляд) результатам. Например, попытка найти нули функции $y = \operatorname{tg} x$ на интервале [1, 2] приводит к следующему.

Пример 2.

```
from scipy import optimize
from math import tan

def f(x):
    return (tan(x))

root = optimize.brentq(f, 1, 2)
print('x =',root)
print('f(x) =',f(root))
```

```
x = 1.570796326793586
f(x) = 763027653393.7319
```

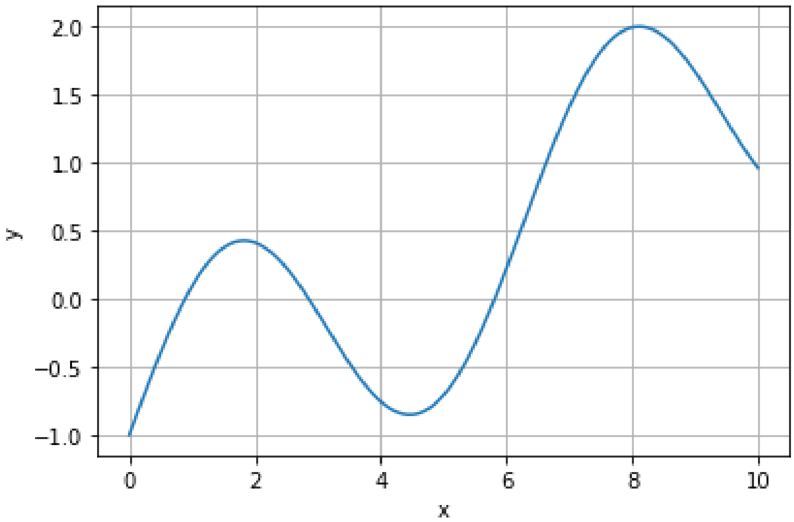
Полученное значение аргумента соответствует приближенному значению $\pi / 2$ и на самом деле является не нулем функции $y = \operatorname{tg} x$, а ее точкой разрыва, при переходе через которую функция меняет знак. Выведенное значение функции в найденной точке показывает нам, что найден не корень.

Условие обнаружение интервала, на концах которого функция принимает значения разных знаков, является принципиальным для алгоритма, использованного в функции brentq. Например, для такой тривиальной функции, как $y = x^2$, функция brentq найти нуля не может.

Логично задать вопрос: как мы можем получить начальное приближение или отрезок, на концах которого функция принимает значения с разными знаками? Часто это проще всего сделать, построив график функции.

Пример 3.

Хотим решить уравнение $x = 4(1 - \sin x)$. Преобразуем его к виду $\sin x - 1 + x/4 = 0$ и воспользуемся тем, что корни уравнения $\sin x - 1 + x/4 = 0$ можно интерпретировать как нули функции $f(x) = \sin x - 1 + x/4$.



Из рисунка видно, что корни уравнения лежат на отрезках $[0,1]$, $[2,3]$ и $[5,6]$. (Можно было использовать другой подход: построить графики функций x и $4(1 - \sin x)$ в одном окне и увидеть, в каких точках они пересекаются.)

```
def f(x):
    return (sin(x)- 1 + x/4)

x1 = optimize.brentq(f, 0, 1)
print('x1 =',x1)
print('f(x1) =',f(x1),'\n')

x2 = optimize.brentq(f, 2, 3)
print('x2 =',x2)
print('f(x2) =',f(x2),'\n')
```

```
x3 = optimize.brentq(f, 5, 6)
print('x3 =',x3)
print('f(x3) =',f(x3),'\n')
```

```
x1 = 0.89048708074438
f(x1) = 2.7755575615628914e-17

x2 = 2.849968934455141
f(x2) = 6.661338147750939e-16

x3 = 5.8128260902615505
f(x3) = -2.220446049250313e-16
```

Упражнение 1. Для следующих функций найти точки разрыва, исследовать их характер, сделать геометрическую иллюстрацию:

а) $y_1 = \dfrac{x^2-4}{|x^2-4|}$, б) $y_2 = e^{\frac{1}{x-2}}$.

Упражнение 2.

Найдите все корни уравнений:

а) $e^{\frac{x}{2}} = 3+\sqrt{x}$; б) $x^3-8x^2+17x+\sqrt{x}=10$.

3. Символическое решение уравнений.

Для решения уравнений можно использовать функцию **solve.** Применение функции **solve** более ограниченно: она позволяет решать уравнения, заданные символическими выражениями или строками. Символическое выражение или строка, которая не содержит уравнения, будут использованы как левая часть уравнения и приравнены к нулю.

Пример 4.

Найдем корни уравнения для $f(x) = 0$, где $f(x) = x^4 + x^3 - x - 1$.

```
from sympy.solvers import solve
from sympy import Symbol
x = Symbol('x')
ans=solve(x**4 + x**3 - x - 1, x)
print(ans)
```

```
[-1, 1, -1/2 - sqrt(3)*I/2, -1/2 + sqrt(3)*I/2]
```

Найдем корни уравнения для $f(y) = 0$, где $f(y) = y^2 - y$.

```
from sympy.solvers import solve
from sympy import Symbol
y = Symbol('y')
ans=solve(y**2 +y, y)
print(ans)
```

```
[-1, 0]
```

Пример 5.

Найдем корни уравнения для $f(x) = 0$, где $f(x) = \cos(x)$.

```
from sympy.solvers import solve
from sympy import Symbol, cos
x = Symbol('x')
ans=solve(cos(x), x)
ans
```

```
[pi/2, 3*pi/2]
```

Из этого примера видно, что решения ищутся символически. Но, конечно, точное решение не всегда возможно найти, и в таких случаях solve вернёт приближённое численное решение.

Для решение нелинейных уравнений и систем можно использовать функцию **nsolve()**. Главное отличие от предыдущей функции, для решения необходимо начальное приближение к корню.

Пример 6.

Найдем корни уравнения для $f(x) = 0$, где $f(x) = \sin(x) - x - 1$.

```
from sympy.solvers import nsolve
from sympy import Symbol, sin

x = Symbol('x')
eq = sin(x) - x - 1
a = nsolve(eq, 0)

print ('x = ', a)
print ('f(x) = ', eq.subs(x,a))
```

```
x = -1.93456321075202
f(x) = 0
```

В данном примере для удобства мы передали в переменную **eq** символьную запись необходимого уравнения. Метод **subs** позволяет заменить одну переменную на другую (в этом примере x поменяли на решение уравнения).

Упражнение 3.

Решите следующие уравнения:

а) $e^{3x} = (3x)^e$; б) $\sin x = \frac{x}{3}$.

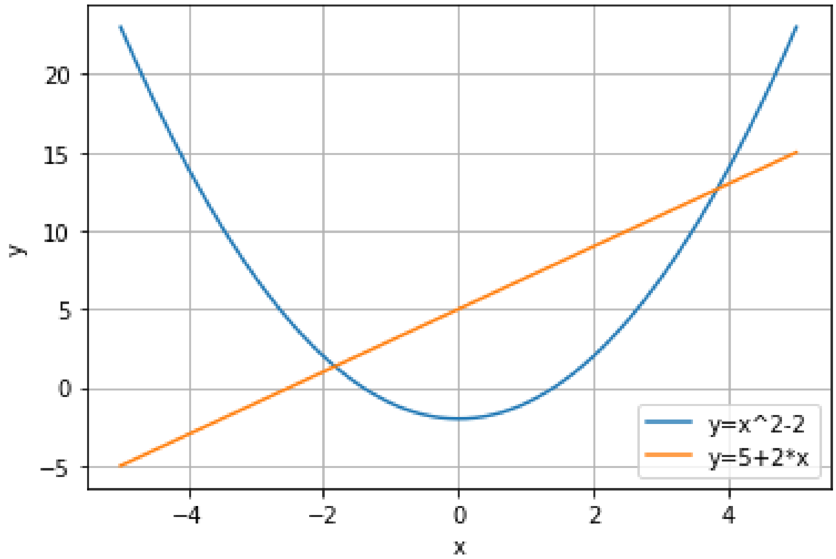
Попробуйте определить, являются ли полученные решения точными или приближёнными.

4. Решение системы уравнений

Функция nsolve также может решать системы нескольких уравнений:

Пример 8.

Необходимо решить систему из двух уравнений: $\begin{cases} x^2 - y - 2 = 0 \\ 2x - y + 5 = 0 \end{cases}$



Из графика видно, что система имеет два решения, значит для функции fsolve будем использовать два начальных приближения.

```
from sympy.solvers import nsolve
from sympy import Symbol, sin

x = Symbol('x')
```

```
y = Symbol('y')
f1 = x**2 - y - 2
f2 = y-2*x-5

a = nsolve((f1, f2), (x, y), (0, 0))
print('x = ', a)

a = nsolve((f1, f2), (x, y), (3, 12))
print('x = ', a)
```

```
x = Matrix([[-1.82842712474619], [1.34314575050762]])
x = Matrix([[3.82842712474619], [12.6568542494924]])
```

Упражнение 4.

Решите систему уравнений:

$$\begin{cases} w+x+4y+3z=5 \\ 2w+3x+y-2z=1 \\ w+2x-5y+4z=3 \\ w-3z=9 \end{cases}$$

Задания для самостоятельной работы

1. Выполнить упражнения из раздела «Краткие теоретические сведения и практические упражнения», которые не успели сделать в аудитории.

2. Самостоятельно выполнить упражнения:

Упражнение С1. Для следующих функций найти точки разрыва, исследовать их характер, сделать геометрическую иллюстрацию:

а) $y_3=\dfrac{\arcsin x}{x}$, б) $y_4=\dfrac{2}{x^2-1}$.

Упражнение С2.

Найдите все корни уравнения $x^2-5x\sin 3x+3=0$.

Упражнение С3.

Решите следующие уравнения:

а) $x+\dfrac{1}{x}=2$; б) $\ln(1+x)=-\ln(1-x^2)$.

Попробуйте определить, являются ли полученные решения точными или приближёнными.

3. Ответить на контрольные вопросы:

1) Когда функция является непрерывной в точке?

2) Дайте классификацию точек разрыва.
3) Назовите аргументы функции brentq.
4) К каким результатам может привести использование функции brentq?
5) Какие уравнения можно решать с помощью функции solve, а какие с помощью nsolve?

Список рекомендуемой литературы

1. http://orioks.miet.ru/oroks-miet/scripts/login.pl?DBnum=9 - ОМА. Предел и непрерывность. Предел функции

2. Сборник задач по математике для втузов под ред. А.В.Ефимова и А.С.Поспелова, часть 2, М.2002, - 5.4.

3. https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.brentq.html - Описание функции brentq().

4. https://docs.sympy.org/latest/modules/solvers/solvers.html - Описание функций solve() и nsolve() .

5. https://www.geeksforgeeks.org/python-sympy-subs-method-2/ - Описание функций subs() .