Практикум 5. Предел последовательности

Цель работы – изучение символьных переменных, выражений и операций, способов решений неравенств, нахождения предела последовательности, построение иллюстраций определения предела.

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 либо в формате интерактивного питон-скрипта (ipynb), либо в виде документа Microsoft Word. Файл следует назвать по следующей схеме pin_10_Ivanov_P_01_s1 (группа, фамилия, инициалы, 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 фукнций; выводы.

Краткие теоретические сведения

и практические упражнения

1. Символьные переменные, константы и выражения.

Поскольку переменные в системе MATLAB по умолчанию задаются как векторные, матричные, числовые и т.д, то есть не имеющие отношения к символьной математике, то для реализации символьных вычислений нужно прежде всего позаботиться о создании специальных *символьных переменных*.

Для создания *символьных переменных* используется функция **sp.Symbol**.

Пример 1.

```
import sympy as sp
x = sp.Symbol('x')
%whos
```

Для создания группы символьных объектов служит функция **symbols**. Нам будет удобно использовать эту функцию вместе со специальной формой создания списков **[f(i) for i in M]** --- составить список из выражений **f(i)**, где переменная **i** принимает все значения из множества **M**.

Пример 2.

```
import sympy as sp
n = 3 # число переменных
# a = список из символьных перменных a1 …, an :
a = [sp.symbols('a%d' % i) for i in range(n)]
# оператор for в данном выражении выступает в качестве квантора «для любых», а не в качестве обозначения цикла
sp.pprint(a)
# печать в "красивом" формате символьных выражений
```

Для всех латинских и греческих букв в пакете **sympy.abs** предусмотрены предопределенные символы. Мы можем использовать их без объявления:

```
import sympy as sp
from sympy.abc import x,y,z,alpha,beta
sp.pprint(alpha*x**2+2*x*y-3*beta*z-z**2)
```

Команда **sp.pi** даёт доступ к символьной версии числа π, не обладающее погрешностью представления числа π в формате с плавающей запятой. Заметим, что эту особую константу **sp.pi** не стоит путать с аналогичным символом из пакета `sympy.abc` (в этом пакете pi – это просто греческая буква).

```
import sympy as sp
from sympy.abc import pi
sp.pprint(sp.sin(pi)) # просто греческая буква
print(sp.sin(sp.pi)) # символьное число пи
```

Упражнение 1. Используя символьное число π, вычислить значения $\cos\frac{\pi k}{2}$ при $k = 1, 2, \ldots, 10$, а затем проделать тоже самое с численным значением π системной константы (из модуля numpy), снова вычислить $\cos\frac{\pi k}{2}$. Проверить равенство полученных результатов с помощью логической операции. Проделать операцию с помощью конструкции **[f(i) for i in M].** Объяснить результат.

2. Символьные операции с выражениями.

Функция **simplify(S)** упрощает символьное выражение S.

Пример 3.

```
import sympy as sp
from sympy.abc import x,y
v = sp.sin(x)**2+sp.cos(x)**2
sp.pprint(v)
print('\n after simplify = '+str(sp.simplify(v)))
# \n означает перевод на новую строку
```

Если упрощение невозможно, то возвращается исходное выражение.

Функция **factor(S)** осуществляет разложение выражения S на множители.

Упражнение 3. Разложить $\dfrac{x^7 + 3x^2 - 4}{x - 1}$ на множители и упростить.

Упражнение 4. Разложить на множители**:**

а) $x^4 + 4$; б) $x^7 + 1$; в) $2x^3 + x^2y - 5xy^2 + 2y^3$.

3. Решение неравенств. Решение неравенств осуществляется с помощью команды **solve_univariate_inequality.** При этом, если неравенство имеет точное решение в радикалах, то выдаётся это решение, иначе – приближённое численно.

```
import sympy as sp
from sympy.abc import x
a = sp.solve_univariate_inequality(x**2>=4,x)
# Без указаний дополнительных параметров мы получим символьное описание интервала
sp.pprint(a)
```

Решение системы неравенств осуществляется с помощью команды **Intersection**, которая объединит результат двух **solve_univariate_inequality.**

```
import sympy as sp
from sympy.abc import x
a = sp.solve_univariate_inequality(x**2>=4,x,relational=False)
# указав relational=False мы запишем в a решение в форме множества
b = sp.solve_univariate_inequality(x**2<=9,x,relational=False)
# тоже самое делаем для решения второго неравенства
c = sp.Intersection(a,b)
# берем объдинение двух множеств и получам наш ответ
sp.pprint(c)
```

Пример 4. Решение неравенств с модулями.

```
import sympy as sp
from sympy.abc import x
sp.solve_univariate_inequality(sp.Abs(x**2-3)>3,x)
sp.N(a,4) # округление до 4 десятичных знаков
```

Замечание: Если внутри неравенства вместо натурального числа 3 указать 3.0, то вместо точного решения сразу будет искаться приближенное.

Упражнение 5. Решить неравенство $x^3 + 3x > 3$. Округлить ответ до 5 знаков.

4. Определение предела последовательности.

Определение. Число a называется пределом последовательности $\{x_n\}$ при n, стремящемся к бесконечности ($a = \lim\limits_{n\to\infty} x_n$), если для любого $\varepsilon > 0$ найдётся номер $n_0(\varepsilon)$ такой, что при всех $n > n_0(\varepsilon)$ выполняется неравенство $|x_n - a| < \varepsilon$.

Пример 5. Найти $n_0(0.001)$ для предела последовательности $x_n = \dfrac{n^3+3n+10}{2n^3-2n+5}$.

Решение. Так как $a = \lim\limits_{n\to\infty} x_n = \dfrac{1}{2}$, то $n_0(0.001)$ находим из неравенства $\left|\dfrac{n^3+3n+10}{2n^3-2n+5} - \dfrac{1}{2}\right| < 0.001$:

```
import sympy as sp
from sympy.abc import n
a = sp.solve_univariate_inequality(sp.Abs(((n**3+3*n+10)/(2*n**3-2*n+5))-0.5)<0.001,n,relational=False)
b = sp.solve_univariate_inequality(n>0,n,relational=False)
# отбросим отрицательные значения
sp.Intersection(a,b)
```

$n_0(0.001) = 45.$

Замечание. Обратите внимание на то, что для того, чтобы не выводить лишние отрицательные промежутки, к основному неравенству было добавлено ограничение $n > 0$.

Упражнение 6. Написать скрипт с параметрами x(n), n0, a, epsilon, осуществляющий следующие действия:

1. Задаёт массив n номеров от n0–2 до n0+10.
2. Строит график последовательности x(n) на указанном промежутке.
3. Строит прямые $y = a - epsilon,\ y = a + epsilon.$

Для последовательностей $\{x_n\}$:

а) $x_n = \dfrac{n^3 - 1}{2n^3 + 2n}$; б) $\lim\limits_{n\to\infty} \dfrac{n^3 + 2n}{2n^4 + 1}$

выполнить:

1. Найти $a = \lim\limits_{n\to\infty} x_n$.
2. Вычислить $n_0(0.01),\ n_0(0.001)$. Результат оформить в виде таблицы

	а)	б)
$n_0(0.01)$		
$n_0(0.001)$		

3. С помощью созданного М-файла построить графическую иллюстрацию.

Указание. При задании формулы x(n) не забывайте, что n – это массив.

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. Решить неравенство $x^3 - 3x + 1 > 0.$, получить точный и приближённый ответ. Для поиска приближенного решения попробуйте задавать числа в неравенстве с плавающей точкой (3.0 или 1.0). Объяснить результат.

Упражнение С2. Для последовательностей $\{x_n\}$:

а) $x_n = \dfrac{2n^5 + 2n^4 + 3}{n^5 - 5n - 10000}$; б) $x_n = \dfrac{\sqrt{2n^5 + 3}}{\sqrt{n^5 - 5}}$;

в) $x_n = \sqrt[3]{n^3 + n^2} - \sqrt[3]{n^3 - n^2}$.

выполнить:

1) Найти $a = \lim\limits_{n \to \infty} x_n$.

2) Вычислить $n_0(0.01)$, $n_0(0.001)$. Результат оформить в виде таблицы

	а)	б)
$n_0(0.01)$		
$n_0(0.001)$		

3) С помощью созданного скрипта построить графическую иллюстрацию.

3. Ответить на контрольные вопросы:

1) Каким образом задаются символьные переменные и массивы?
2) Каким образом можно упростить символьное выражение?
3) Каким образом можно разложить на множители алгебраическое выражение или натуральное число?
4) В каком случае при решении неравенства выдаётся символьное выражение?
5) Как получить численное решение неравенства?
6) Дайте определение предела последовательности.

Список рекомендуемой литературы

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

2. Сборник задач по математике для втузов под ред. А.В.Ефимова и А.С.Поспелова, часть 2, М.2002, - 5.3.