ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ТЕХНІКА

https://doi.org/10.31649/1997-9266-2021-156-3-61-68

УДК 519.85

Б. І. Мокін[1]
О. Б. Мокін[1]
Д. О. Шалагай[1]

ПРО ОДИН ІЗ ПІДХОДІВ НАБЛИЖЕНОГО ОБЧИСЛЕННЯ ІНТЕГРАЛІВ СТІЛТЬЄСА І ЛЕБЕГА НА МОВІ PYTHON В ЗАДАЧАХ СИСТЕМНОГО АНАЛІЗУ З ДИСКРЕТНИМИ МОДЕЛЯМИ

[1]Вінницький національний технічний університет

Запропоновано програми наближеного обчислення інтегралів Стілтьєса та Лебега на мові Python, яких нині немає в програмних пакетах Sympy та Scipy, в яких зосереджені програмні функції обчислення лише однократних та багатократних інтегралів Рімана. Для реалізації цих програм здійснене коригування класичних математичних виразів, якими визначаються інтеграли Стілтьєса та Лебега і синтезовано алгоритми, придатні для розроблення програм наближеного обчислення цих інтегралів на мові Python. Особливістю алгоритму, який синтезовано для наближеного обчислення інтегралу Лебега, є врахування того, що міра Лебега дискретної функції, заданої на нульвимірній множині точок, розміщених на відрізку визначення її аргументу, є монотонною неперервною функцією координати функціональної осі, зростаючою від нуля в точці мінімального значення цієї функції до величини, що дорівнює довжині відрізка функціональної осі в межах від мінімального значення цієї функції до її максимального значення. В цьому алгоритмі значення дискретної функції, що інтегрується по Лебегу, відсортовуються так, щоб складати зростаючу послідовність, міра кожного значення якої задається відрізком функціональної осі в межах сусідніх значень цієї послідовності в бік її зростання. Розроблені програми інтегрування по Стілтьєсу та Лебегу на мові Python містять у своїй структурі стандартні, уже відомі програмні функції цієї мови. Показано, що запропоновані програми будуть корисними науковцям, які займаються задачами системного аналізу з дискретними моделями.

Ключові слова: інтеграл Стілтьєса, інтеграл Лебега, алгоритми адаптації інтегралів до мови програмування Python, програми обчислення інтегралів Стілтьєса та Лебега на мові Python.

Вихідні передумови та постановка задачі

Аналіз пакетів програм на мові Python, розміщених в програмному середовищі Anaconda [1], та матеріалів, розміщених в одному з найповніших навчальних посібників з програмування на мові Python [2], показав, що станом на сьогодні і в програмному пакеті Sympy і в програмному пакеті Scipy можна знайти достатньо програмних інструментів для обчислення як однократних, так і багатократних інтегралів Рімана, але в жодному нічого не сказано про те, як обчислювати інтеграли Стілтьєса та інтеграли Лебега, які донедавна ще не конкурували з широко розповсюдженими в математичному аналізі інтегралами Рімана, але завдяки поширенню ключових положень функціонального аналізу в сферу інформаційних технологій [3] починають знаходити місце для застосування в якості критеріальних функціоналів в задачах системного аналізу з дискретними моделями, заданими на нульвимірних множинах, для яких не можуть бути обчисленими інтеграли Рімана в принципі.

А тому у цій статті ми вирішили доповнити відомий програмний інструментарій реалізації процесів інтегрування неперервних функцій по Ріману на мові Python двома програмами інтегрування на цій же мові дискретних (решітчастих) функцій по Стілтьєсу та Лебегу, аби озброїти спеціалістів з IT-технологій, які користуються в повсякденній практиці програмування мовою Python, ефективним програмним інструментарієм реалізації процесів інтегрування по Стілтьєсу та Лебегу функцій, заданих на нульвимірних множинах в задачах перетворення законів розподілу випадкових величин та в моделях прогнозування часових рядів, в яких вони виступають як критеріальні функціонали.

Приступаючи до розв'язання поставленої задачі, одразу ж вкажемо на те, що потрібну нам інформацію стосовно суті інтегрування по Стілтьєсу та Лебегу ми будемо брати з роботи [3], не обтяжуючи себе посиланнями на те, звідки вона була взята авторами цієї роботи, а інформацію стосовно програмного інструментарію мови Python ми будемо брати з робіт [1], [2], теж не обтяжуючи себе посиланнями на те, звідки вона була взята авторами цих робіт.

Отже, як відомо [3], основною відмінністю інтеграла Стілтьєса від інтеграла Рімана є те, що в інтегралі Рімана інтегрування на відрізку $[a, b]$ неперервної та обмеженої функції $f(x)$ здійснюється з використанням приростів dx її аргументу на цьому ж відрізку числової осі, а в інтегралі Стілтьєса інтегрування на відрізку $[a, b]$ функції $f(x)$ здійснюється з використанням приростів $dg(x)$ іншої обмеженої функції $g(x)$, заданої на цьому ж відрізку числової осі, а сама функція $f(x)$, що інтегрується, називається інтегрованою за функцією $g(x)$ на відрізку $[a, b]$ числової осі. Символічно інтеграл Стілтьєса записується так:

$$S = \int_a^b f(x)\,dg(x), \tag{1}$$

а реалізує він, фактично, операцію визначення граничного числа S за виразом

$$S = \lim_{\Delta_i x \to 0} \sum_{i=1}^{n \to \infty} f(\xi_i)\big(g(x_i) - g(x_{i-1})\big), \tag{2}$$

в якому

$$\Delta_i x = x_i - x_{i-1},\ \ i = 1, 2, \ldots, n,\ \ \xi_i \in \Delta_i x. \tag{3}$$

Що ж до інтеграла Лебега, то його розгляд розпочнемо цитатою з роботи [3]: «Коли математики побачили, що існують функції, які не інтегруються за Ріманом, вони почали пошук такого узагальнення поняття визначеного інтеграла, за допомогою якого ці функції теж можна було б інтегрувати. І таке узагальнення вдалося здійснити Лебегу, який запропонував приріст координати, за якою здійснюється інтегрування функції $y = y(x)$, заданої на відрізку $[a, b]$, визначати не по осі аргументу x , а по функціональній осі y, адже у цьому випадку, навіть коли координата x задається на нульвимірній множині E скінченної чи зліченної кількості точок на осі x, координата y буде елементом множини дійсних чисел на відрізку $[m, M]$ осі y, мірою якого є його довжина, і лівою границею якого є дійсне число m, що є мінімальним значенням цієї функції на відрізку $[a, b]$, а правою границею є дійсне число M, що є максимальним значенням цієї функції на цьому ж відрізку $[a, b]$». І конструював свій інтеграл Лебег, висунувши умову, щоб вимірна і обмежена нижнім m та верхнім M значеннями функція $y = y(x)$, що задана на відрізку $[a, b]$ осі x, була визначеною на множині E з мірою

$$mE\big(m \le y < M\big). \tag{4}$$

А в результаті реалізації цієї конструкції Лебегом отримано вираз

$$L = \lim_{\Delta_i y \to 0} \sum_{i=1}^{n} y_{i-1} mE_i\big(y_{i-1} \le y < y_i\big), \tag{5}$$

який мав усі властивості інтеграла при інтегруванні по координаті y, а тому Лебег увів його в математику як нове трактування визначеного інтеграла, який інші математики назвали в його честь інтегралом Лебега, а суму (5) назвали інтегральною сумою Лебега на множині E з мірою, визначеною виразом (4). Але оскільки міра mE є монотонною функцією координати y, то, позначивши її як $g(y)$, вираз (5) можна переписати у вигляді

$$L = \lim_{\Delta_i y \to 0} \sum_{i=1}^{n} y_{i-1}\big(g(y_i) - g(y_{i-1})\big), \tag{6}$$

який уже має вигляд інтеграла Стілтьєса, а тому може бути записаним і так:

$$L = \int_m^M y dg(y). \tag{7}$$

Виразами (1)—(7) і будуть визначатись вихідні передумови для поставленої нами вище задачі.

Побудова основних розрахункових співвідношень та їх програмування

Почнемо розв'язання поставленої задачі з визначення алгоритму обчислення інтегралу Стілтьєса та синтезу програми на мові Python для реалізації цього алгоритму.

Аналізуючи вирази (2), (3), бачимо, що наближене значення граничного числа S можна визначити з виразу

$$S \approx \sum_{i=1}^{n} f(\xi_i)\big(g(x_i) - g(x_{i-1})\big), \tag{8}$$

в якому n — досить значне число, але задане в межах розумного вибору, продиктованого допустимим значенням похибки обчислення.

Цілком очевидно, що за достатньо великих значень n матимемо значення

$$\Delta_i x = x_i - x_{i-1}, \ \ i = 1, 2, \ldots, n \tag{9}$$

настільки малими, що замість співвідношення $\xi_i \in \Delta_i x$ можна використовувати співвідношення $\xi_i = x_{i-1}$ і, як наслідок, можна прийняти, що

$$f(\xi_i) = f(x_{i-1}). \tag{10}$$

Підставляючи вираз (10) у вираз (8), матимемо

$$S \approx \sum_{i=1}^{n} f(x_{i-1})\big(g(x_i) - g(x_{i-1})\big), \tag{11}$$

Ось, виходячи з цього виразу (11), ми і будемо синтезувати програму наближеного обчислення інтегралу Стілтьєса від дискретної функції $f(x_{i-1}),\ i = 1, 2, \ldots, n$, заданої на нульвимірній множині точок відрізку $[a, b]$ по заданій на цьому ж відрізку функції $g(x)$.

Команди програми, як і в роботі [2], ми писатимемо зліва послідовно одна за другою, а етапи алгоритму створення програми з використанням виразу (11) писатимемо у цьому ж рядку справа після символу #, переносячи словесні характеристики цих етапів в разі, якщо вони не поміщатимуться в один рядок, в рядок наступний, перед початком якого уже не стоятиме символ #, а будуть проставлені чотири крапки «….». Для доведення дії програми до конкретних обчислень ми в цій програмі використаємо цілком конкретні функції $f(x),\ g(x)$, а в разі необхідності користувачеві нашої програми обчислювати наближено інтеграли Стілтьєса для іншої пари функцій, він їх в поле програми вписуватиме замість тих, які використали при створенні програми ми.

Отже, програма на мові Python для обчислення інтегралу Стілтьєса від функції $f(x) = x^3 + 3x^2 - 2x - 4$ по функції $g(x) = 2\exp(x) - 3x$ дійсної змінної x, заданої на відрізку [0, 2] дискретно через проміжок $\Delta_i x = x_i - x_{i-1} = 0{,}1;\ \ i = 1, 2, \ldots, 20$.

```
In [1]: import numpy as np                  # Виклик ППП numpy під символом np
In [2]: x=np.linspace(0,2,21)               # Формування масиву значень x
In [3]: def f(x):                           # Формування функції f(x), що інтегрується по Стілтьєсу на
            return x**3+3*x**2-2*x-4        ….відрізку [0, 2] значень аргументу x
In [4]: fvec=np.vectorize(f)                # Векторизація функції f(x)
In [5]: f1=fvec(x); f1                      # Візуалізація векторизованої функції f(x) у вигляді списку f1
Out[5]:
array([-4. , -4.169, -4.272, -4.303, -4.256,
-4.125, -3.904, -3.587, -3.168, -2.641, -2.,
-1.239, -0.352, 0.667, 1.824, 3.125,
4.576, 6.183, 7.952, 9.889, 12. ])
In [6]: g=lambda x: 2*np.exp(x)-3*x         # Формування функції g(x), по якій інтегрується f(x) та її
In [7]: gvec=np.vectorize(g)                ….векторизація
In [8]: g1=gvec(x); g1                      # Візуалізація векторизованої функції g(x) у вигляді списку g1
```

```
Out[8]:
array([2., 1.91034184, 1.84280552,
1.79971762, 1.7836494 , 1.79744254,
1.8442376, 1.92750541, 2.05108186,
2.21920622, 2.43656366, 2.70833205,
3.04023385, 3.43859334, 3.91039993,
4.46337814, 5.10606485, 5.84789478,
6.69929493, 7.67178888, 8.7781122 ])
In [9]: g11=np.diff(g1); g11       # Визначення масиву g11 перших різниць векторизованої
                                   ….функції g(x) та його візуалізація
Out[9]:
array([-0.08965816, -0.06753632,
-0.0430879, -0.01606822, 0.01379315,
0.04679506, 0.08326781, 0.12357644,
0.16812437, 0.21735743, 0.27176839,
0.3319018, 0.39835949, 0.4718066,
0.55297821, 0.64268671, 0.74182993,
0.85140015, 0.97249396, 1.10632331])
In [10]: f11=f1[: -1]; f11         # Видалення з масиву f1 останнього елемента і приведення
                                   ….його до вигляду f11 для узгодження з масивом g11
Out[10]:
array([-4., -4.169, -4.272, -4.303, -4.256,
-4.125, -3.904, -3.587, -3.168, -2.641, -2.,
-1.239, -0.352, 0.667, 1.824, 3.125,
4.576, 6.183, 7.952, 9.889])
In [11]: s1=f11*g11; s1            # Поелементне перемноження масивів f11 та g11 та його
                                   ….візуалізація
Out[11]:
array([ 0.35863266, 0.28155892,
0.18407151, 0.06914155, -0.05870363,
-0.19302962,-0.32507755, -0.4432687,
-0.53261799, -0.57404098, -0.54353678,
-0.41122633, -0.14022254, 0.314695,
1.00863225, 2.00839596, 3.39461378,
5.2642071, 7.73327194, 10.94043125])
In [12]: S=s1.sum (); S            # Обчислення інтегралу Стілтьєса від дискретної функції
Out[12]: 28.33592779370339         ….f(x) по функції g(x) від дискретної функції f(x) по функції
                                   ….g(x) на відрізку [0, 2] значень аргументу x
```

Кінець програми наближеного обчислення інтегралу Стілтьєса

А далі продовжимо розв’язання поставленої задачі з визначення алгоритму обчислення інтегралу Лебега та синтезу програми на мові Python для реалізації цього алгоритму.

Аналізуючи вирази (5), (6), бачимо, що наближене значення лебегівської інтегральної суми L можна визначити з виразу, за структурою подібного до виразу (11), за допомогою якого ми обчислювали наближене значення інтегралу Стілтьєса. Але, зважаючи на те, що міра mE значень функції $y(x)$, заданої дискретно на відрізку $[a, b]$ осі аргументу x, на осі значень цієї функції y згідно з виразом (4) є функцією, монотонно зростаючою від нуля в точці m до значення, що дорівнює довжині M відрізку $[m, M]$, очевидним стає той факт, що мірою значення функції $y(x)$ у точці x_{i-1}, де $i = 1, 2, \ldots, n$, тобто значення y_{i-1} є відрізок осі у між значенням y_{i-1} та наступним значенням y_i, тобто

$$mE_i\left(y_{i-1} < y \le y_i\right) = y_i - y_{i-1} = \Delta_i y. \qquad (12)$$

Але, оскільки міра є функцією монотонно зростаючою, то це означає, що підставляти вираз (12) у вираз (5) для визначення інтегральної суми Лебега можна лише після того, як значення y_{i-1} будуть відсортованими за їх зростанням, тобто заміненими на значення y^*_{i-1}, для яких є справедливим співвідношення

$$y_0^* < y_1^* < y_2^* < \ldots < y_{(i-1)}^* < \ldots < y_n^*, \qquad (13)$$

$$y_0^* = m, \quad y_n^* = M, \quad n = \frac{b-a}{\Delta x} = \frac{b-a}{\text{const}}, \tag{14}$$

З урахуванням співвідношень (12)—(14) перепишемо вираз (6), який є відображенням виразу (7), у вигляді, придатному для наближеного обчислення інтегралу Лебега, тобто у вигляді

$$L \approx \sum_{i=1}^{n} y_{i-1}^* \Delta_i y^*. \tag{15}$$

Ось, виходячи з цього виразу (15), ми і будемо синтезувати програму наближеного обчислення інтегралу Лебега від дискретної функції $y\left(x_{i-1}\right) = y_{i-1},\ i = 1, 2, \ldots, n$, заданої на нульвимірній множині точок відрізку [a, b].

Для доведення дії і цієї програми до конкретних обчислень ми і в цій програмі використаємо цілком конкретну функцію $y = f(x)$, а за необхідності користувачеві нашої програми обчислювати наближено інтеграли Лебега для інших функцій, він їх в поле програми вписуватиме замість тієї, яку використали автори при створенні програми.

Отже, програма на мові Python для обчислення інтегралу Лебега від функції $f(x) = e^{-x} \sin 3x$ дійсної змінної x, заданої на відрізку [0, 3] дискретно в точках через проміжок $\Delta_i x = x_i - x_{i-1} = 0{,}15; \quad i = 1, 2, \ldots, 2.$

```
In [1]: import numpy as np                    # Виклик ППП numpy під символом np
In [2]: x=np.linspace(0,3,21)                 # Формування масиву значень x
In [3]: def f(x):                             # Формування функції f(x), що інтегрується по Лебегу на
          return np.exp(-x)*np.sin(3*x)       ….відрізку [0, 3] значень аргументу x
In [4]: fvec=np.vectorize(f)                  # Векторизація функції f(x)
In [5]: f1=fvec(x); f1                        # Візуалізація векторизованої функції f(x) у вигляді списку f1
Out[5]:
array([ 0. , 0.3743783 , 0.58030285,
0.62214868, 0.53445891, 0.36753575,
0.17375969, -0.00294201, -0.1332846,
-0.20441749, -0.21811645, -0.1866539,
-0.12773711, -0.05972154, 0.00205897,
0.0474343 , 0.07199992, 0.07646286,
0.06518194, 0.0443898 , 0.02051817])
In [6]: M=max(f1); M                          # Визначення максимуму M функції f(x) на відрізку [0, 3]
Out[6]: 0.6221486811452028                    ….значень аргументу x
In [7]: m=min(f1); m                          # Визначення мінімуму m функції f(x) на відрізку [0, 3]
Out[7]: -0.21811645170452654                  ….значень аргументу x
In [8]: mEf=M-m;mEf                           # Визначення міри M-m функції f(x) на відрізку [0, 3]
Out[8]: 0.8402651328497294                    ….значень аргументу x
In [9]: f11=np.sort(f1); f11                  # Трансформація масиву f1 в масив f11, відсортований за
Out[9]:                                       ….зростанням значень
array([-0.21811645, -0.20441749,
-0.1866539 , -0.1332846 , -0.12773711,
-0.05972154, -0.00294201, 0,
0.00205897,
0.02051817, 0.0443898 , 0.0474343,
0.06518194, 0.07199992, 0.07646286,
0.17375969, 0.36753575, 0.3743783,
0.53445891, 0.58030285, 0.62214868])
In [10]: g=np.diff(f11); g                    # Визначення масиву g перших різниць векторизованої
Out[10]:                                      ….функції f(x)
array([0.01369896, 0.0177636,
0.0533693, 0.00554749, 0.06801557,
0.05677952, 0.00294201, 0.00205897,
0.0184592, 0.02387163, 0.0030445,
0.01774765, 0.00681798, 0.00446294,
```

```
0.09729683, 0.19377606, 0.00684255,
0.16008061, 0.04584394, 0.04184583])
In [11]: f111=f11[: -1]; f111          # Видалення з масиву f11 останнього елемента і приведення
                                       ….його до вигляду f111 для узгодження з масивом g
Out[11]:
array([-0.21811645, -0.20441749,
-0.1866539, -0.1332846 , -0.12773711,
-0.05972154, -0.00294201, 0.,
0.00205897, 0.02051817, 0.0443898,
0.0474343 , 0.06518194, 0.07199992,
0.07646286, 0.17375969, 0.36753575,
0.3743783 , 0.53445891, 0.58030285])
In [12]: l1=f111*g; l1                 # Поелементне перемноження масивів f111 та g та його
                                       ….візуалізація
Out[12]:
array([-2.98796838e-03, -3.63118992e-03,
-9.96158766e-03, -7.39394368e-04,
-8.68811299e-03, -3.39096030e-03,
-8.65544179e-06, 0.00000000e+00,
3.80069447e-05, 4.89802141e-04,
1.35144637e-04, 8.41847084e-04,
4.44409318e-04, 3.21331035e-04,
7.43959408e-03, 3.36704688e-02,
2.51488210e-03, 5.99307062e-02,
2.45017000e-02, 2.42832565e-02])
In [13]: L=np.sum(l1); L               # Обчислення інтегралу Лебега від дискретної функції f(x)
Out[13]: 0.125203279831203             ….на відрізку [0, 3] значень аргументу x
```

Кінець програми наближеного обчислення інтегралу Лебега.

Висновки

Запропоновано програми наближеного обчислення інтегралів Стілтьєса та Лебега на мові Python, яких нині немає в програмних пакетах Sympy та Sciру, де зосереджені програмні функції обчислення лише однократних та багатократних інтегралів Рімана. Для реалізації цих програм здійснене коригування класичних математичних виразів, якими визначаються інтеграли Стілтьєса та Лебега і синтезовано алгоритми, придатні для розроблення програм наближеного обчислення цих інтегралів на мові Python. Показано, що запропоновані програми будуть корисними науковцям, які займаються задачами системного аналізу з дискретними моделями.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] *Python.* [Електронний ресурс]. Режим доступу: https://www.python.org/downloads/ .

[2] П. Г. Доля, *Введение в научный Python*. Харків: ХНУ ім. Каразіна, 2016, 265 с.

[3] Б. І. Мокін, В. Б. Мокін, О. Б. Мокін, *Функціональний аналіз, адаптований до прикладних задач в галузі інформаційних технологій*, навч. посіб. Вінниця: ВНТУ, 2020, 192 с.

Рекомендована кафедрою системного аналізу та інформаційних технологій ВНТУ

Стаття надійшла до редакції 16.06.2021

Мокін Борис Іванович — академік НАПН України, д-р техн. наук, професор, професор кафедри системного аналізу та інформаційних технологій, професор кафедри електромеханічних систем автоматизації в промисловості і на транспорті, e-mail borys.mokin@gmail.com ;

Мокін Олександр Борисович — д-р техн. наук, професор, професор кафедри системного аналізу та інформаційних технологій, e-mail abmokin@gmail.com ;

Шалагай Дмитро Олександрович — аспірант кафедри системного аналізу та інформаційних технологій, e-mail: d.shalagai@gmail.com .

Вінницький національний технічний університет, Вінниця

B. I. Mokin[1]
O. B. Mokin[1]
D. O. Shalagai[1]

On one of the Approaches to Approximate Calculation of Lebesgue–Stieltjes Integrals in Python in System Analysis Problems with Discrete Models

[1]Vinnytsia National Technical University

The article presents programs for the approximate calculation of Lebesgue-Stieltjes integrals in Python, which are not currently available in the SymPy and SciPy packages. Those packages include only functions for calculating single and multiple Riemann integrals. To implement these programs, there has been made the correction of classical mathematical expressions, which determine the Lebesgue–Stieltjes integrals, and synthesized algorithms suitable for the development of programs for the approximate calculation of these integrals in Python. The feature of the algorithm synthesized for the approximate calculation of the Lebesgue integral is that the Lebesgue measure of a discrete function given on a zero-dimensional set of points located on the segment of its argument is a monotonic continuous function of the coordinate of the functional axis. This axis value increases from zero at the point of the minimum value of this function to a value equals to the length of the segment of the functional axis in the range from the minimum value of this function to its maximum value. In this algorithm, the values of a discrete Lebesgue-integrated function are sorted to form an ascending sequence, the measure of each value of which is given by a segment of the functional axis within adjacent values of this sequence in the direction of its growth. The developed Python programs for Lebesgue–Stieltjes integration contain standard already known program functions of this programming language. The article shows that the proposed programs can be useful for scientists who work on problems of systems analysis with discrete models.

Keywords: Stieltjes integral, Lebesgue integral, algorithms for adapting integrals to the Python programming language, programs for calculating Lebesgue–Stieltjes integrals in Python.

Mokin Borys I. — Academician of NAPS of Ukraine, Dr. Sc. (Eng.), Professor, Professor of the Chair of System Analysis and Information Technologies, Professor of the Chair of Electromechanical Systems of Automation in Industry and Transport, e-mail: borys.mokin@gmail.com ;

Mokin Oleksandr B. — Dr. Sc. (Eng.), Professor, Professor of the Chair of System Analysis and Information Technologies, e-mail: abmokin@gmail.com ;

Shalagai Dmytro O. — Post-Graduate Student of the Chair of System Analysis and Information Technologies, e-mail: d.shalagai@gmail.com

Б. И. Мокин[1]
А. Б. Мокін[1]
Д. А. Шалагай[1]

Об одном из подходов приближенного вычисления интегралов Стилтьеса и Лебега на языке Python в задачах системного анализа с дискретными моделями

[1]Винницкий национальный технический университет

Предложены программы приближенного вычисления интегралов Стилтьеса и Лебега на языке Python, отсутствующих в программных пакетах Sympy и Scipy, в которых сосредоточены программные функции вычисления только однократных и многократных интегралов Римана. Для реализации этих программ осуществлены корректировки классических математических выражений, которыми определяются интегралы Стилтьеса и Лебега и синтезированы алгоритмы, пригодные для разработки программ приближенного вычисления этих интегралов на языке Python. Особенностью алгоритма, который синтезирован для приближенного вычисления интеграла Лебега, есть учет того, что мера Лебега дискретной функции, заданной на нольмерном множестве точек, расположенных на отрезке определения ее аргумента, является монотонной непрерывной функцией координаты функциональной оси, растущей от нуля в точке минимального значения этой функции до величины, равной длине отрезка функциональной оси в пределах от минимального значения этой функции до ее макси-

мального значения. В этом алгоритме значение дискретной функции, интегрирующиеся по Лебегу, сортируются так, чтобы составлять растущую последовательность, мера каждого значения которой задается отрезком функциональной оси в пределах соседних значений этой последовательности в сторону ее роста. Разработанные программы интегрирования по Стилтьеса и Лебегу на языке Python содержат в своей структуре стандартные, уже известные программные функции этого языка. Показано, что предложенные программы будут полезными для ученых, занимающихся задачами системного анализа с дискретными моделями.

Ключевые слова: интеграл Стилтьеса, интеграл Лебега, алгоритмы адаптации интегралов к языку программирования Python, программы вычисления интегралов Стилтьеса и Лебега на языке Python.

Мокин Борис Иванович — академик НАПН Украины, д-р техн. наук, профессор, профессор кафедры системного анализа и информационных технологий, профессор кафедры электромеханических систем автоматизации в промышленности и на транспорте, e-mail borys.mokin@gmail.com ;

Мокин Александр Борисович — д-р техн. наук, профессор, профессор кафедры системного анализа и информационных технологий, e-mail abmokin@gmail.com ;

Шалагай Дмитрий Александрович — аспирант кафедры системного анализа и информационных технологий, e-mail: d.shalagai@gmail.com