ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №10 (Вариант 24)

**Тема:** метод Ньютона-Котеса и трапеций

**Задание:** Решить определенный интеграл методом Ньютона-Котеса и методом трапеций

**Теория:**

**Метод Ньютона-Котеса**

Выше были рассмотрены три схожих метода интегрирования функций – метод прямоугольников, метод трапеций, метод Симпсона. Их объединяет общая идея: интегрируемая функция интерполируется на отрезке интегрирования по равноотстоящим узлам многочленом Лагранжа, для которого аналитически вычисляется значение интеграла. Семейство методов, основанных на таком подходе, называется **методами Ньютона-Котеса**.

В выражении

http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image076.png

коэффициенты http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image042.png правильнее называть **весовыми коэффициентами**. Величину

http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image077.png,

определяющую погрешность численного интегрирования, называют **остатком**.

Для семейства методов Ньютона-Котеса можно записать общее выражение:

http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image078.png(2.16)

где *n* – порядок метода Ньютона-Котеса, *N* – количество частичных отрезков,

http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image079.png, http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image080.png, http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image081.png.

Из выражения (2.16) легко можно получить формулу прямоугольников для http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image082.png, формулу трапеций для http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image083.png, и формулу Симпсона для http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image084.png. Коэффициенты http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image085.png могут быть заданы в табличной форме (таблица.2.1).

**Решение:**

f(x) = (3.3\*x-2.8)/(4.5\*x^3+7.4)

Вычисление интеграла на интервале [2; 3,2] методом Ньютона-Котеса с коэффициентами 10 порядка

**Разобьем интервал [2; 3,2] на 1 часть**

Находим шаг:

http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image079.png, где n – порядок метода Ньютона-Котеса.

Делаем разбиение на 1:

H = (3,2 - 2)/(1) = 1,2

Делаем разбиение на 10:

h = (3,2 - 2)/(10\*1) = 0,12

H[0] = 0,0268

H[1] = 0,177

H[2] = -0,081

H[3] = 0,454

H[4] = -0,435

H[5] = 0,713

H[6] = -0,435

H[7] = 0,454

H[8] = -0,081

H[9] = 0,177

H[10] = 0,026

**Рассмотрим промежуток 0 интервал которого = [2; 3,2]**

**Разделим его на 10 частей**

Для этого воспользуемся следующей формулой:

http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image081.png

xi = 2 + (0 + 0 \* 10) \* 0,12 = 2

I = 0 + H[0] \* f(2) \* 0,12 = 0,00028

xi = 2 + (1 + 0 \* 10) \* 0,12 = 2,12

I = 0,00028 + H[1] \* f(2,12) \* 0,12 = 0,0020

xi = 2 + (2 + 0 \* 10) \* 0,12 = 2,24

I = 0,0020 + H[2] \* f(2,24) \* 0,12 = 0,0012

xi = 2 + (3 + 0 \* 10) \* 0,12 = 2,36

I = 0,0012 + H[3] \* f(2,36) \* 0,12 = 0,0053

xi = 2 + (4 + 0 \* 10) \* 0,12 = 2,48

I = 0,0053 + H[4] \* f(2,48) \* 0,12 = 0,0016

xi = 2 + (5 + 0 \* 10) \* 0,12 = 2,6

I = 0,0016 + H[5] \* f(2,6) \* 0,12 = 0,0074

xi = 2 + (6 + 0 \* 10) \* 0,12 = 2,72

I = 0,0074 + H[6] \* f(2,72) \* 0,12 = 0,0041

xi = 2 + (7 + 0 \* 10) \* 0,12 = 2,84

I = 0,0041 + H[7] \* f(2,84) \* 0,12 = 0,0073

xi = 2 + (8 + 0 \* 10) \* 0,12 = 2,96

I = 0,0073 + H[8] \* f(2,96) \* 0,12 = 0,0068

xi = 2 + (9 + 0 \* 10) \* 0,12 = 3,08

I = 0,0068 + H[9] \* f(3,08) \* 0,12 = 0,0079

xi = 2 + (10 + 0 \* 10) \* 0,12 = 3,2

I = 0,0079 + H[10] \* f(3,2) \* 0,12 = 0,0081

I = 0,0081 \* 10 = 0,081

**Интеграл от 2 до 3.2 функции (3.3\*x - 2.8)/(4.5\*x^3 + 7.4) равен 0.0810**

**Протокол решения в Scilab:**

function l=f(x), l=(3.3\*x-2.8)/(4.5\*x^3+7.4) endfunction

i=0, maxf=0, a=2, b=3.2, n=1, tabl=[]

fm = (3.3\*%z-%2.8)/(4.5\*%z^3+7.4)

disp(varn(fm,"x"),'Подинтегральная функция:')

disp('Интеграл определён на промежутке от '+string(a)+' до '+string(b))

h=(b-a)/n

disp('Для получения точности 0.001 отрезок необходимо поделить на '+string(n)+' частей')

disp('Следовательно шаг разбиения: '+string(h))

tabl(1,:)=[0 a f(a)]

su=(f(a)+f(b))/2

for k=1:n-1

su=su+f(a+k\*h)

tabl(k+1,:)=[k a+k\*h f(a+k\*h)]

end

tabl(n,:)=[n b f(b)]

disp('Все вычисления сведём в таблицу:')

disp(tabl,' i xi f(xi)')

su=h\*su

disp('Вычисленное значение интеграла: '+string(su))

**Вывод в консоли:**

--> function l=f(x), l=(3.3\*x-2.8)/(4.5\*x^3+7.4) endfunction

--> i=0, maxf=0, a=2, b=3.2, n=1, tabl=[]

i =

0.

maxf =

0.

a =

2.

b =

3.2

n =

1.

tabl =

[]

--> fm = (3.3\*%z-%2.8)/(4.5\*%z^3+7.4)

fm = (3.3\*%z-%2.8)/(4.5\*%z^3+7.4)

--> disp(varn(fm,"x"),'Подинтегральная функция:')

Неопределённая переменная: fm

--> disp('Интеграл определён на промежутке от '+string(a)+' до '+string(b))

Интеграл определён на промежутке от 2 до 3.2

--> h=(b-a)/n

h =

1.2

--> disp('Для получения точности 0.001 отрезок необходимо поделить на '+string(n)+' частей')

Для получения точности 0.001 отрезок необходимо поделить на 1 частей

--> disp('Следовательно шаг разбиения: '+string(h))

Следовательно шаг разбиения: 1.2

--> tabl(1,:)=[0 a f(a)]

tabl =

0. 2. 0.0875576

--> su=(f(a)+f(b))/2

su =

0.0688343

--> for k=1:n-1

> su=su+f(a+k\*h)

> tabl(k+1,:)=[k a+k\*h f(a+k\*h)]

> end

--> tabl(n,:)=[n b f(b)]

tabl =

1. 3.2 0.0501111

--> disp('Все вычисления сведём в таблицу:')

Все вычисления сведём в таблицу:

--> disp(tabl,' i xi f(xi)')

i xi f(xi)

1. 3.2 0.0501111

--> su=h\*su

su =

0.081012

--> disp('Вычисленное значение интеграла: '+string(su))

Вычисленное значение интеграла: 0.081012

**Теория:**

**Метод трапеций**

Суть метода трапеций.

Поставим перед собой следующую задачу: пусть нам требуется приближенно вычислить определенный интеграл формула, где подынтегральная функция y=f(x) непрерывна на отрезке [a;b].

Разобьем отрезок [a;b] на n равных интервалов длины h точками формула. В этом случае шаг разбиения находим как формула и узлы определяем из равенства формула.

Рассмотрим подынтегральную функцию на элементарных отрезках формула.

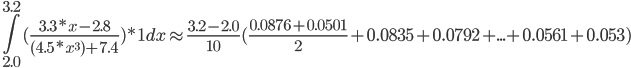
В качестве приближенного значения интеграла формула возьмем выражение формула, то есть, примем формула.

Сути метода трапеций, которая состоит в представлении определенного формула интеграла в виде суммы интегралов вида формула на каждом элементарном отрезке и в последующей приближенной замене формула.

**Решение:**

f(x) = (3.3\*x-2.8)/(4.5\*x^3+7.4)

***Формула трапеций****:*

https://chart.googleapis.com/chart?cht=tx&chl=\int\limits_%7ba%7d%5e%7bb%7d%7bf(x)%20dx%7d%20\approx%20%20\frac%7bb-a%7d%7bn%7d%20%5b\frac%7by_%7b0%7d%2By_%7bn%7d%7d%7b2%7d%20%2B%20y_%7b1%7d%20%2B%20y_%7b2%7d%20%2B%20...%20%2B%20y_%7bn-1%7d%5d  
https://chart.googleapis.com/chart?cht=tx&chl=h=\frac%7bb-a%7d%7bn%7d%20=%20\frac%7b3.2-2.0%7d%7b10%7d%20=%200.12 

Остаточный член квадратурной формулы:

https://chart.googleapis.com/chart?cht=tx&chl=R_%7bn%7d%20=%20-%20\frac%7bb-a%7d%7b12%7d\cdot%20h\cdot%20f%5e%7b\prime%20\prime%7d(c)

C:\Users\Владимир\Downloads\chart (1).png

Найдем максимальное значение второй производной функции на интервале [2.0;3.2].

y = x/((4.5\*x^3+7.4)^2)\*((364.5\*x^3\*(3.3\*x-2.8))/(4.5\*x^3+7.4)-178.2\*x+75.6)

[2.0;3.2]

Находим первую производную функции:

y' =

Приравниваем ее к нулю:

≠ 0

Глобальных экстремумов нет

Находим стационарные точки:

Вычисляем значения функции на концах отрезка

f(2) =

f(3.2) =

**Ответ:**

Имеются только локальные экстремумы (на заданном интервале)

fmin = , fmax =   
  
https://chart.googleapis.com/chart?cht=tx&chl=R_%7bn%7d%20=%20-%20\frac%7bb-a%7d%7b12%7d\cdot%20h%5e%7b2%7d\cdot%20f%5e%7b\prime%20\prime%7d(c)%20=%20\frac%7b3.2-2.0%7d%7b12%7d\cdot%200.12%5e%7b2%7d\cdot%20%20=%200  
Таким образом, I = 0.0811 ± 0

**Протокол решения в Scilab:**

function l=f(x), l=(3.3\*x-2.8)/(4.5\*x^3+7.4) endfunction

i=0, maxf=0, a=2.0, b=3.2, n=40, tabl=[]

fm =(3.3\*%z-%2.8)/(4.5\*%z^3+7.4)

disp(varn(fm,"x"),'Подинтегральная функция:')

disp('Интеграл определён на промежутке от '+string(a)+' до '+string(b))

h=(b-a)/n

disp('Для получения точности 0.001 отрезок необходимо поделить на '+string(n)+' частей')

disp('Следовательно шаг разбиения: '+string(h))

tabl(1,:)=[0 a f(a)]

su=(f(a)+f(b))/2

for k=1:n-1

su=su+f(a+k\*h)

tabl(k+1,:)=[k a+k\*h f(a+k\*h)]

end

tabl(n,:)=[n b f(b)]

disp('Все вычисления сведём в таблицу:')

disp(tabl,' i xi f(xi)')

su=h\*su

disp('Вычисленное значение интеграла: '+string(su))

**Вывод в консоли:**

--> disp(varn(fm,"x"),'Подинтегральная функция:')

Подинтегральная функция:

3.3x - 2.8

------------

4.5x^3 + 7.4

--> disp('Интеграл определён на промежутке от '+string(a)+' до '+string(b))

Интеграл определён на промежутке от 2 до 3.2

--> h=(b-a)/n

h =

0.03

--> disp('Для получения точности 0.001 отрезок необходимо поделить на '+string(n)+' частей')

Для получения точности 0.001 отрезок необходимо поделить на 40 частей

--> disp('Следовательно шаг разбиения: '+string(h))

Следовательно шаг разбиения: 0.03

Все вычисления сведём в таблицу:

--> disp(tabl,' i xi f(xi)')

i xi f(xi)

0. 2. 0.0875576

1. 2.03 0.086559

2. 2.06 0.0855404

3. 2.09 0.0845056

4. 2.12 0.0834583

5. 2.15 0.0824017

6. 2.18 0.0813387

7. 2.21 0.0802717

8. 2.24 0.0792033

9. 2.27 0.0781353

10. 2.3 0.0770697

11. 2.33 0.0760082

12. 2.36 0.0749521

13. 2.39 0.0739028

14. 2.42 0.0728614

15. 2.45 0.071829

16. 2.48 0.0708063

17. 2.51 0.0697941

18. 2.54 0.0687932

19. 2.57 0.067804

20. 2.6 0.066827

21. 2.63 0.0658627

22. 2.66 0.0649113

23. 2.69 0.0639731

24. 2.72 0.0630484

25. 2.75 0.0621374

26. 2.78 0.06124

27. 2.81 0.0603565

28. 2.84 0.0594868

29. 2.87 0.0586309

30. 2.9 0.0577889

31. 2.93 0.0569607

32. 2.96 0.0561462

33. 2.99 0.0553454

34. 3.02 0.0545581

35. 3.05 0.0537842

36. 3.08 0.0530236

37. 3.11 0.0522762

38. 3.14 0.0515417

40. 3.2 0.0501111

--> su=h\*su

su =

0.0810836

--> disp('Вычисленное значение интеграла: '+string(su))

Вычисленное значение интеграла: 0.0810836

**Вывод:**

Можно заметить, что при нахождении ответов решения системы есть небольшие разбежности, потому что считая вручную используем ε = 0,001 (допускаемое приближение).

**Список используемой литературы:**

1. Мак-Кракен Д., Дорн У. Численные методы и программирование на ФОРТРАНе. – М.: Мир, 1977. – 584 с.
2. Чисельні методи : навчальний посібник / В. М. Задачин, І. Г. Конюшенко. – Х.: Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с. (Укр. мов.) ст 94