

## УНІВЕРСАЛЬНИЙ АЛГОРИТМ РОЗРАХУНКУ ІНТЕГРАЛЬНОГО РІВНЯННЯ ВОЛЬТЕРРИ ІІ РОДУ ІЗ ЗАСТОСУВАННЯМ ФОРМУЛ НЬЮТОНА-КОТЕСА

Ситник О.О., к.т.н.,

Кисельова Г.О.,

Кисельов В.Б.

Черкаський державний технологічний університет

*Рассмотрен универсальный алгоритм численного решения интегрального уравнения Вольтерры II рода с использованием формул Ньютона-Котеса различной точности.*

**Ключевые слова:** интегральное уравнение, тест, алгоритм, Matlab.

**Вступ.** Застосування інтегральних рівнянь є одним із найефективніших методів обчислення задач багатьох областей природничих наук, але розрахунок аналітичними методами можливий лише в деяких окремих випадках. Тому, для розрахунків інтегральних рівнянь використовують чисельні методи. Найбільш застосовуваними у розрахунках рівнянь Вольтерри II роду є квадратурні алгоритми, в основу яких покладені формули трапецій та прямокутників [1, 2, 3, 4]. Основними перевагами подібних алгоритмів є простота та універсальність, тобто можливість розраховувати інтегральні рівняння, як при рівномірному, так і при нерівномірному розташуванні вузлів.

**Постановка проблеми.** Формули трапецій та прямокутників мають невисоку точність (порядка  $\hat{I}(h^2)$ ), що, в свою чергу, впливає на величину похибки квадратурних алгоритмів, які використовують подібні формули. Специфікою обчислень інтегральних рівнянь Вольтерри за допомогою квадратурних формул є необхідність використання для кожного наступного кроку результатів, які отримані при попередніх розрахунках, тому має місце накопичення похибки, яка може збільшуватися і при зменшенні кроку обчислень. Величина похибки також збільшується при збільшенні проміжку інтегрування, враховуючи це, виникає необхідність використання, при розрахунках інтегральних рівнянь чисельними методами, формул високої алгебраїчної точності [1].

**Мета.** Створити універсальний алгоритм розрахунку інтегрального рівняння Вольтерри II роду з використанням формул Ньютона-Котеса високої алгебраїчної точності.

**Шляхи вирішення проблеми.** Формули чисельного розрахунку інтегралів можна, умо-

*The universal algorithm of numeral decision of Volterra's integral equations of the second kind with the use of Newton-Cotes' formula of different exactness is considered in the article.*

**Key words:** integral equations, test, algorithm, Matlab.

вно, поділити на дві основні групи: формули з рівномірним розподілом вузлів інтегрування – формули Ньютона-Котеса; формули з нерівномірним розподілом вузлів інтегрування – формули Гауса. При однаковій кількості вузлів інтегрування формули Гауса мають більш високу точність завдяки більш раціональному розміщенню вузлів та компенсації похибки на різних проміжках інтегрування [4, 7], але переважна більшість моделей, в яких використовуються інтегральні рівняння (особливо, коли змінною величиною є час) вимагають розрахунку з постійним кроком при рівномірному розподілі вузлів інтегрування (розрахунок на рівномірних сітках). Для розрахунків з постійним кроком використовуються формули Ньютона-Котеса.

В основі квадратурних методів рішення інтегральних рівнянь лежить заміна інтегральних операторів кінцевими сумами із застосуванням квадратурних формул [1, 2, 3]. При чисельному розрахунку рівнянь Вольтерри II роду змінні межі інтегрування фіксуються, що дає можливість використовувати формули наближеного обчислення визначених інтегралів, які в загальному випадку мають вигляд

$$\int_a^b j(x) dx = \sum_{i=1}^n A_i \cdot j(x_i) + R[j],$$

де  $x_i$  – фіксовані абсциси проміжку  $[a, b]$ , або

вузли інтегрування;

$A_i$  – чисельні коефіцієнти;

$R[j]$  – залишок (похибка квадратурної формули);

При розрахунку лінійного інтегрального рівняння Вольтерри II роду методом квадратур використовується вираз [1]:

$$y(x_i) - \int_a^{x_i} K(x_i, s) y(s) ds = f(x_i), \quad i = \overline{1, n},$$

де  $x_i$  – вузли інтегрування;

$K(x_i, s)$  – значення ядра інтегрального рівняння у вузлах інтегрування;

$f(x_i)$  – значення правої частини інтегрального рівняння у вузлах інтегрування;

$y(x_i)$  – значення шуканої функції у вузлах інтегрування.

Приймаючи значення  $x_i$  в якості вузлів квадратурної формули та замінюючи за її допомогою інтеграл кінцевою сумою, отримуємо систему:

$$y(x_i) - \sum_{j=1}^i A_j \cdot K(x_i, x_j) \cdot y(x_j) = f(x_i) + R_i[y],$$

де  $x_i$  – вузли інтегрування;

$K(x_i, x_j)$  – значення ядра інтегрального рівняння у вузлах інтегрування;

$A_j$  – чисельні коефіцієнти;

$f(x_i)$  – значення правої частини інтегрального рівняння у вузлах інтегрування;

$R_i[y]$  – залишок (похибка квадратурної формули);

$y(x_i)$  – значення шуканої функції у вузлах інтегрування.

Нехтуючи величиною похибки  $R_i[y]$  та приймаючи позначення

$$y(x_i) = y_i; \quad f(x_i) = f_i; \quad K(x_i, x_j) = K_{ij},$$

отримуємо систему лінійних алгебраїчних рівнянь, яка може бути приведена до вигляду

$$-\sum_{j=1}^{i-1} A_j \cdot K_{ij} \cdot y_j + (1 - A_i \cdot K_{ii}) \cdot y_i = f_i, \quad i = \overline{1, n},$$

звідки отримуємо рекурентну формулу [1, 2] для визначення  $y_i$

$$y_i = \frac{f_i + \sum_{j=1}^{i-1} A_j \cdot K_{ij} \cdot y_j}{1 - A_i \cdot K_{ii}}, \quad i = \overline{1, n},$$

при умові

$$1 - A_i \cdot K_{ii} \neq 0.$$

Особливістю чисельних розрахунків інтегральних рівнянь за даною методикою є необхідність визначення  $y_i$  при будь-якому значенні  $i = \overline{2, n}$ , що не створює проблем при застосуванні формул трапецій та прямокутників, чим і обумовлене таке поширення даних методів, але при розрахунках з постійним кроком із застосуванням формул Ньютона-Котеса вищої

точності виникають труднощі, пов'язані з можливістю розраховувати значення інтегралів лише при  $i$ , значення яких залежать від кількості точок апроксимації застосовуваної формули. Так, для формули Сімпсона (трюхточкова апроксимація) розрахунок можливий лише при

$$i = 2 \cdot z + 1, \quad z = 1, 2, 3, \dots,$$

для формули Буля (чотирьохточкова апроксимація) – при

$$i = 3 \cdot z + 1, \quad z = 1, 2, 3, \dots,$$

або, в загальному випадку,

$$i = (m - 1) \cdot z + 1, \quad z = 1, 2, 3, \dots,$$

де  $m$  – кількість точок апроксимації застосованої формули. Тому використання формул Ньютона-Котеса вищої точності при розрахунках інтегральних рівнянь Вольтерри можливе лише при комбінації з іншими методами, або сумісному застосуванні формул різної точності, при цьому значення  $y_2$  розраховується лише за формулами трапецій або прямокутників, значення  $y_3$  – за формулою Сімпсона, і т.д.

Блок-схему алгоритму чисельного рішення рівняння Вольтерри II роду за формулами Ньютона-Котеса вищої точності зображено на рис. 1.

На першому етапі роботи даного алгоритму створюються файл-функції правої частини (fprav) та ядра (kern) інтегрального рівняння.

На другому етапі вводяться межі (a, b) та крок (h) інтегрування. На третьому етапі обчислюється кількість вузлів інтегрування

$$n = \frac{b - a}{h} + 1,$$

значення абсцис

$$x_i = a + (i - 1) \cdot h, \quad i = \overline{1, n}$$

та величини

$$\left. \begin{aligned} y_1 &= \text{fprav}(x_1), \\ y_2 &= \frac{\text{fprav}(x_2) + 0,5 \cdot h \cdot \text{kern}(x_2, x_1) \cdot y_1}{1 - 0,5 \cdot h \cdot \text{kern}(x_2, x_2)}, \end{aligned} \right\}$$

попередньо перевіривши умову

$$1 - 0,5 \cdot h \cdot \text{kern}(x_2, x_2) \neq 0$$

(якщо умова не виконується, подальший розрахунок припиняється), після чого величині  $i$  присвоюється значення  $i = 2$ .

На четвертому етапі перевіряється умова  $i \geq n$ ,

при виконанні якої розрахунок завершується та відбувається виведення даних  $y_i$  та  $x_i$ .

На п'ятому етапі величині  $i$  присвоюється значення  $i = i + 1$  та вводяться проміжні величини:  $g = 0, \quad k = i, \quad t = 0$ .

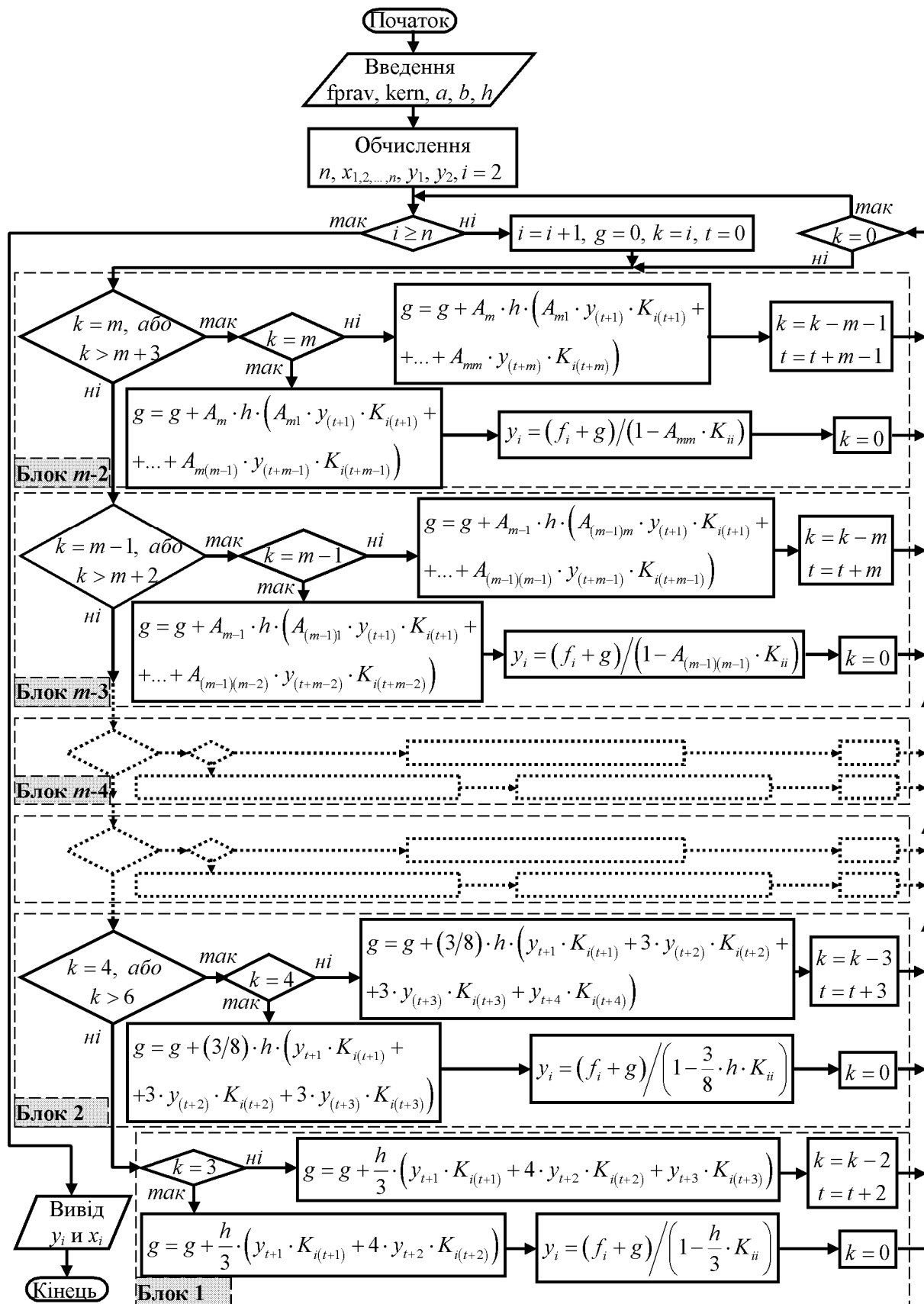


Рис. 1. Блок-схема квадратурного алгоритму чисельного рішення інтегрального рівняння Вольтерри II роду за формулами Ньютона-Котеса

На шостому етапі перевіряються умови

$$k = m, \text{ або } k > m + 3,$$

де  $m$  – кількість точок апроксимації застосованої формули. При невиконанні умов здійснюється перехід до сьомого етапу. При виконанні умови  $k = m$  розраховуються величини

$$g = g + A_m \cdot h \cdot \left( A_{m1} \cdot y_{(t+1)} \cdot \text{kern}(x_i, x_{(t+1)}) + \dots + A_{m(m-1)} \cdot y_{(t+m-1)} \cdot \text{kern}(x_i, x_{(t+m-1)}) \right);$$

$$y_i = \frac{\text{fprav}(x_i) + g}{1 - A_{mm} \cdot \text{kern}(x_i, x_i)},$$

величині  $k$  присвоюється значення  $k = 0$  та здійснюється перехід до восьмого етапу. При виконанні умови

$$k > m + 3,$$

розраховуються величини

$$g = g + A_m \cdot h \cdot \left( A_{m1} \cdot y_{(t+1)} \cdot \text{kern}(x_i, x_{(t+1)}) + \dots + A_{mm} \cdot y_{(t+m)} \cdot \text{kern}(x_i, x_{(t+m)}) \right),$$

величинам  $k$  та  $t$  присвоюються значення

$$k = k - m - 1;$$

$$t = t + m - 1;$$

та здійснюється перехід до восьмого етапу.

На сьомому етапі перевіряються умови

$$k = m - 1, \text{ або } k > m + 2.$$

При невиконанні умов здійснюється перехід до наступного етапу. При виконанні умови

$$k = m - 1,$$

розраховуються величини

$$g = g + A_{(m-1)} \cdot h \cdot \left( A_{(m-1)1} \cdot y_{(t+1)} \cdot \text{kern}(x_i, x_{(t+1)}) + \dots + A_{(m-1)(m-2)} \cdot y_{(t+m-2)} \cdot \text{kern}(x_i, x_{(t+m-2)}) \right);$$

$$y_i = \frac{\text{fprav}(x_i) + g}{1 - A_{(m-1)(m-1)} \cdot \text{kern}(x_i, x_i)},$$

величині  $k$  присвоюється значення  $k = 0$  та здійснюється перехід до восьмого етапу. При виконанні умови

$$k > m + 2,$$

розраховуються величини

$$g = g + A_{(m-1)} \cdot h \cdot \left( A_{(m-1)1} \cdot y_{(t+1)} \cdot \text{kern}(x_i, x_{(t+1)}) + \dots + A_{(m-1)(m-1)} \cdot y_{(t+m-1)} \cdot \text{kern}(x_i, x_{(t+m-1)}) \right),$$

величинам  $k$  та  $t$  присвоюються значення

$$k = k - m - 2;$$

$$t = t + m - 2;$$

та здійснюється перехід до восьмого етапу.

Кількість таких перевірок відповідає кількості однотипних блоків алгоритму або величині  $m - 2$  та закінчується перевіркою умови  $k = 3$ , при виконанні якої здійснюється розрахунок за трьохточковою формулою Сімпсона

$$g = g + \frac{h}{3} \cdot \left( y_{t+1} \cdot K_{i(t+1)} + 4 \cdot y_{t+2} \cdot K_{i(t+2)} \right);$$

$$y_i = (f_i + g) / \left( 1 - \frac{h}{3} \cdot K_{ii} \right),$$

величині  $k$  присвоюється значення  $k = 0$  та здійснюється перехід до восьмого етапу. При невиконанні умови розрахунок проводиться за формулою

$$g = g + \frac{h}{3} \cdot \left( y_{t+1} \cdot K_{i(t+1)} + 4 \cdot y_{t+2} \cdot K_{i(t+2)} + y_{t+3} \cdot K_{i(t+3)} \right),$$

величинам  $k$  та  $t$  присвоюються значення

$$k = k - 2;$$

$$t = t - 2;$$

та здійснюється перехід до восьмого етапу.

На восьмому етапі перевіряється умова  $k = 0$ , при виконанні якої здійснюється перехід до четвертого етапу. При невиконанні даної умови відбувається перехід до шостого етапу.

**Отримані результати.** Використовуючи отримані формули та розрахунковий алгоритм, в програмі MATLAB розроблено програми *voltr* (з використанням формули трапецій), аналогічну програмі *volts1* [1], та *voltnew* (з використанням формул Ньютона-Котеса різної точності), в основу якої покладено розроблений алгоритм від двох- до дев'ятиточкової формули Ньютона-Котеса.

Відмінність програми *voltr* від *volts1* полягає в тому, що в програмі *volts1* похибки обчислень  $\Delta u_i$  лише розраховуються за допомогою підпрограми *udu*, а в програмі *voltr* розраховані величини похибок  $\Delta u_i$  додаються до відповідних значень  $y_i$  (компенсація похибки), що дає змогу (якщо ядро інтегрального рівняння не має особливостей та вирази для розрахунку похибки є вірними) отримати більш точні значення інтегрального рівняння.

В програмі *voltnew* передбачена можливість вибору методу розрахунку від формули трапецій до дев'ятиточкової формули Ньютона-Котеса, що підвищує універсальність даного алгоритму та надає змогу відслідковувати зміну величини похибки від точності апроксимуючої формули.

Звертання до програми `voltr` аналогічне використуваному в [1] звертання до програми `volts1`. Основною відмінністю є необхідність задавати величину кроку, а не кількість вузлів інтегрування, що є більш зручним при застосуванні даного алгоритму в реальних розрахунках, але призводить до необхідності попереднього узгодження величини кроку та меж інтегрування.

$$[y_i, x_i] = \text{voltr}(\text{prav}, \text{kern}, a, b, h)$$

де  $y_i$  – рішення інтегрального рівняння;  
 $x_i$  – вузли інтегрування;  
`prav` – права частина інтегрального рівняння;  
`kern` – ядро інтегрального рівняння;  
`a, b` – межі інтегрування;  
`h` – крок інтегрування.

Звернення до програми `voltnew` аналогічне зверненню до програми `voltr`. Основною відмінністю є необхідність задавати метод інтегрування (кількість точок апроксимуючої формули).

$$[y_i, x_i] = \text{voltnew}(\text{prav}, \text{kern}, a, b, h, \text{met})$$

де  $y_i$  – рішення інтегрального рівняння;  
 $x_i$  – вузли інтегрування;  
`prav` – права частина інтегрального рівняння;  
`kern` – ядро інтегрального рівняння;  
`a, b` – межі інтегрування;  
`h` – крок інтегрування;  
`met` – метод інтегрування (кількість точок апроксимуючої формули від „2” до „9”).

**Тест програм.** Для порівняння ефективності використання формул Ньютона-Котеса різної точності при розрахунках інтегральних рівнянь Вольтерри II роду розраховуємо за їх допомогою кілька прикладів [1], які мають точні рішення, та визначаємо похибку розрахунків. Наводимо ці приклади.

Приклад 1 (приклад 1.2 [1]):

$$y(x) - \int_0^x e^{x^2-s^2} y(s) ds = e^{x^2};$$

точне рішення:

$$y(x) = e^{x^2+x}.$$

Приклад 2 (приклад 1.11 [1]):

$$y(x) - \int_0^x (1 - (x-s) \cdot e^{2x}) y(s) ds = (1 - x \cdot e^{2x}) \cdot \cos 1 - e^{2x} \cdot \sin 1;$$

точне рішення:

$$y(x) = e^x \cdot [\cos(e^x) - e^x \cdot \sin(e^x)].$$

Абсолютна похибка розрахунків  $\Delta_{abs}$  визначається як модуль різниці між точним значеннями інтегрального рівняння  $y_m(x)$  та наближеними  $y_i(x)$ , розрахованими відповідним чисельним методом

$$\Delta_{abs} = |y_m(x) - y_i(x)|.$$

Отримані результати розрахунків прикладів 1 та 2 за програмами `voltr` (з використанням формули трапецій з компенсацією похибки) та `voltnew` (з використанням формул різної точності від формули трапецій до дев'ятиточкової формули Ньютона-Котеса) для максимальних значень  $\max \Delta_{abs}$  похибки, при різних проміжках інтегрування та величинах кроку  $h$ , зведені в таблицю 1. Діаграми найбільш характерних випадків для програм `voltr` та `voltnew` представлені на рис. 2–5. Поведінку похибки  $\Delta_{abs}$  та самої функції  $y(x)$  на кінцях відповідних проміжків, де величина похибки є максимальною, для програм `voltr` та `voltnew` з використанням восьмиточкової формули Ньютона-Котеса, яка дає найбільш стабільні результати, зображено на рис. 6–9.

Таблиця 1

Максимальна величина похибки розрахунків рівнянь Вольтерри II роду для програм `voltr` та `voltnew` з використанням апроксимуючих формул Ньютона-Котеса для прикладів 1 та 2

№ п/п	№ прикладу	Межі інтегрування, $[a; b]$	Крок, $h$	К-сть точок апроксимації	Максимальна величина похибки розрахунку								
					Формула трапецій, <code>voltr</code>	Кількість точок апроксимації формули Ньютона-Котеса, <code>voltnew</code>							
						2	3	4	5	6	7	8	9
1	Приклад 1	[0; 2.5]	0.1	26	0.5002	13.18	1.042	0.075	0.074	0.068	0.075	0.069	0.071
2		[0; 2.5]	0.01	251	6.4e-4	0.132	1.3e-3	7.8e-6	7.5e-6	6.8e-6	8.0e-6	7.5e-6	8.5e-6
3		[0; 3.7]	0.01	371	5.3877	1100	9.158	0.05	0.043	0.039	0.045	0.043	0.048
4		[0; 3.7]	0.002	1851	0.0438	44.0	0.081	8.0e-5	6.8e-5	6.2e-5	7.3e-5	6.9e-5	7.9e-5
5		[0; 4.9]	0.002	2451	5.85+3	6e+6	1e+4	8.907	6.835	6.254	7.385	6.934	7.912
6	Приклад 2	[0; 2.5]	0.1	26	19.5449	37.96	14.90	8.175	17.7	10.41	16.66	10.27	12.76
7		[0; 2.5]	0.01	251	0.0385	0.29	0.051	1.1e-3	6.4e-4	5.4e-4	4.0e-4	5.5e-6	4.8e-6
8		[0; 3.7]	0.01	371	50.5127	137.0	61.4	4.647	6.497	6.168	4.934	0.207	0.363
9		[0; 3.7]	0.002	1851	0.3407	5.267	0.455	7.6e-3	3.2e-3	2.6e-3	1.8e-3	6.6e-6	9.8e-6
10		[0; 5.4]	0.002	2701	19053	28540	13609	1152	2143	1440	1401	935.3	939.5

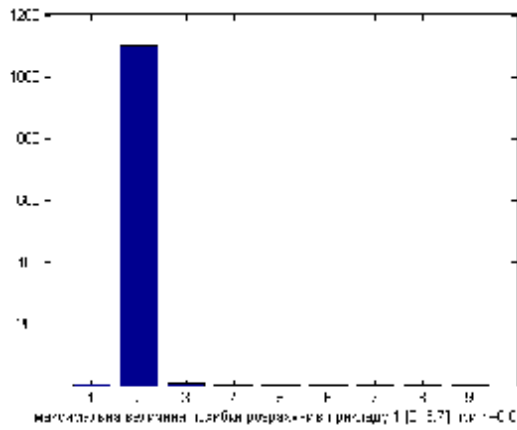


Рис. 2. Максимальна величина похибки розрахунків програм voltrr та voltnew з різною кількістю точок апроксимуючої формули від „2” до „9” для прикладу 1 [0, 3.7] при  $h = 0.01$

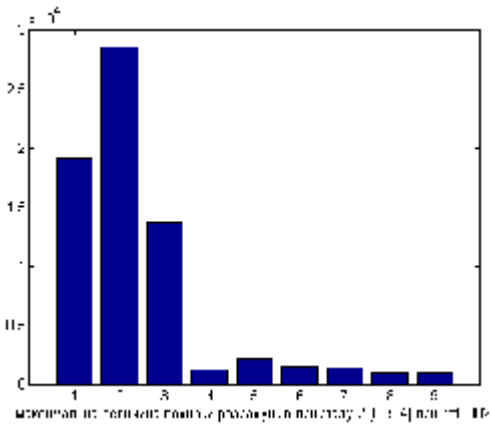


Рис. 5. Максимальна величина похибки розрахунків програм voltrr та voltnew з різною кількістю точок апроксимуючої формули від „2” до „9” для прикладу 1 [0, 5.4] при  $h = 0.002$

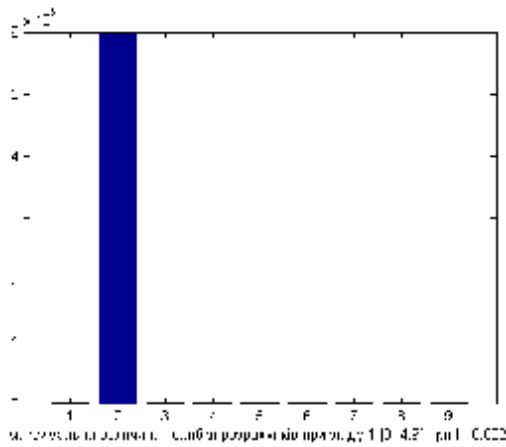


Рис. 3. Максимальна величина похибки розрахунків програм voltrr та voltnew з різною кількістю точок апроксимуючої формули від „2” до „9” для прикладу 1 [0, 4.9] при  $h = 0.002$

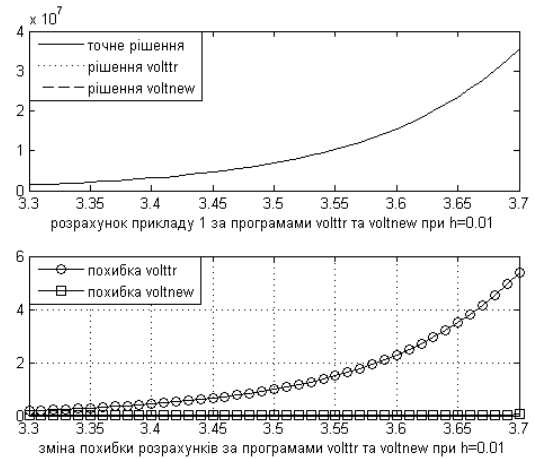


Рис. 6. Графік зміни величини похибки розрахунків програм voltrr та voltnew з восьмиточковою апроксимуючою формулою для прикладу 1 [0, 3.7] при  $h = 0.01$

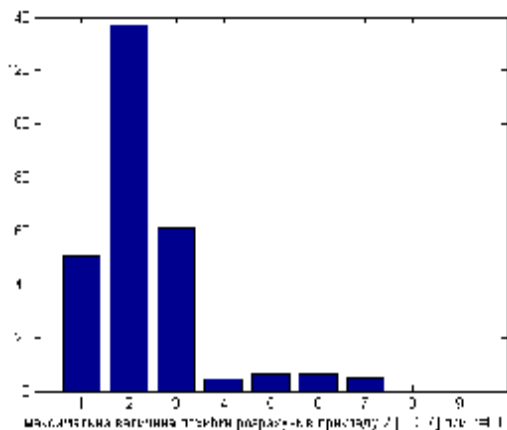


Рис. 4. Максимальна величина похибки розрахунків програм voltrr та voltnew з різною кількістю точок апроксимуючої формули від „2” до „9” для прикладу 2 [0, 3.7] при  $h = 0.01$

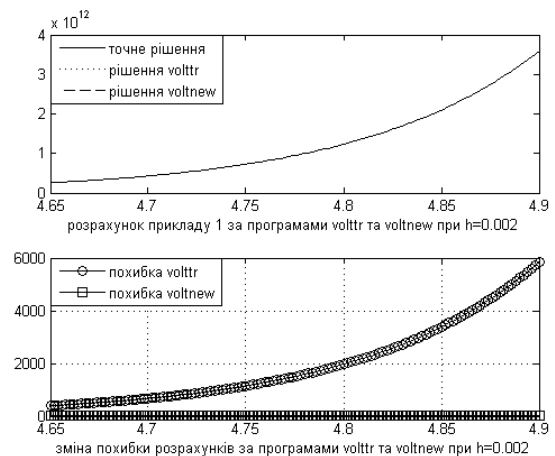
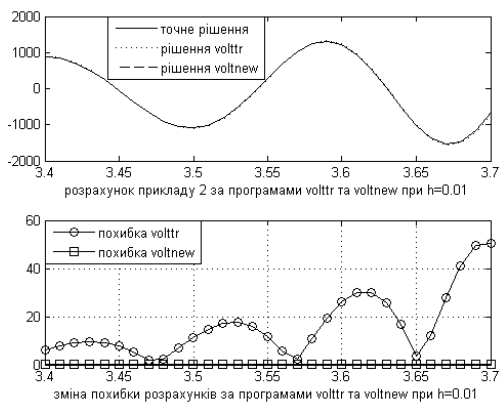


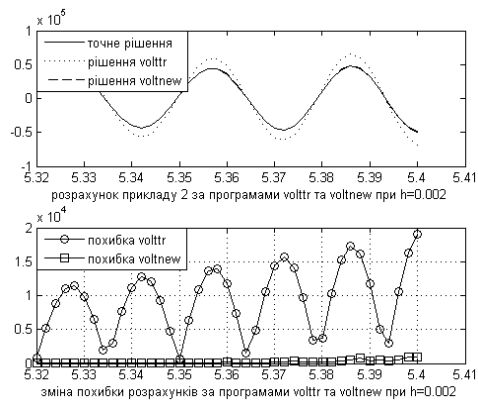
Рис. 7. Графік зміни величини похибки розрахунків програм voltrr та voltnew з восьмиточковою апроксимуючою формулою для прикладу 1 [0, 4.9] при  $h = 0.002$



**Рис. 8. Графік зміни величини похибки розрахунків програм volttr та voltnew з восьми-точковою апроксимуючою формулою для прикладу 2 [0, 3.7] при  $h = 0.01$**

**Висновки.** Результати порівняння найбільш характерних дослідів рис. 2-9 показують, що величина похибки розрахунків програми volttr для прикладу 1 на проміжку [0, 3,7] при  $h = 0,01$  складає 5,3877, на проміжку [0, 4,9] при  $h = 0,002$  – 5853, для прикладу 2 на проміжку [0, 3,7] при  $h = 0,01$  – 50,5127, на проміжку [0, 5,4] при  $h = 0,002$  – 19053, а для програми voltnew з використанням восьмиточкової формули Ньютона-Котеса для прикладу 1 на проміжку [0, 3,7] при  $h = 0,01$  складає 0,043, на проміжку [0, 4,9] при  $h = 0,002$  – 6,934, для прикладу 2 на проміжку [0, 3,7] при  $h = 0,01$  – 0,207 на проміжку [0, 5,4] при  $h = 0,002$  – 935,3. Таким чином можна зробити висновок, що програма voltnew з використанням восьмиточкової формули Ньютона-Котеса дає на декілька порядків точніші результати програми volttr, яка, в свою чергу, є більш точною за програму voltnew з використанням формули трапецій без компенсації похибки. Порядок похибки програми volttr фактично відповідає порядку похибки програми voltnew з використанням формули Сімпсона, але, враховуючи, що при розрахунках похибки програми volttr, для її компенсації, необхідно проводити додаткові обчислення, тому розрахунок за програмою voltnew потребує менше часу. Найбільш важливим в даному порівнянні є не величина похибки, а можливість розраховувати інтегральні рівняння на більшому проміжку при достатній точності.

Порівнюючи результати тестів, можна зробити висновок про високу стабільність та якість розрахунків, проведених за допомогою програми voltnew, в якій застосована восьмиточкова формула Ньютона-Котеса. Використання дев'ятиточкової формули практично не покращує отримані результати, що пов'язане з особливостями даного алгоритму та використаних прикладів. Аналіз даних, отриманих при розрахунках програмою volttr, в якій застосована формула трапецій з компенсацією похибки, та програмою voltnew з використанням формули трапецій без компенсації похибки, показує ефективність підпрограми *udy*, за допомогою якої розраховується похибка. Таким чином, програма



**Рис. 9. Графік зміни величини похибки розрахунків програм volttr та voltnew з восьми-точковою апроксимуючою формулою для прикладу 2 [0, 5.4] при  $h = 0.002$**

ма volttr може бути застосована при розрахунках лінійних інтегральних рівнянь Вольтерри II роду при нерівномірному розподілі вузлів інтегрування. Застосування формул Ньютона-Котеса вищої точності дає можливість будувати алгоритми для розрахунків лінійних та нелінійних інтегральних рівнянь Вольтерри II роду чисельними методами, однак необхідно враховувати властивості ядра, правої частини та рішення даних рівнянь, а саме наявність вихідних відповідних порядків. Для складання програм розрахунку нелінійних рівнянь Вольтерри II роду за даним алгоритмом, найбільш ефективним є метод послідовних наближень.

**ЛІТЕРАТУРА**

1. Верлань А.Ф., Сизиков В.С. Интегральные уравнения: методы, алгоритмы, программы. – К.: Наукова думка, 1986. – 544 с.
2. Арушанян И.О. Численное решение интегральных уравнений методом квадратур. Практикум на ЭВМ. – М.: МГУ, 2002. – 72 с.
3. А.В. Манжиров, А.Д. Полянин. Методы решения интегральных уравнений: Справочник. – М.: Факториал, 1999. – 272 с.
4. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Наука, 2003.
5. Шуп Т. Решение инженерных задач на ЭВМ. – М.: Наука, 1982. – 238 с.
6. Ю.Л. Клетков, А.Ю. Клетков, М.М. Шульц, MATLAB 7: программирование, численные методы. – С.-Пб.: БХВ-Петербург, 2005. – 752 с.
7. Алексеев Е.Р., Чеснокова О.В., Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. – М.: НТ Пресс, 2006. – 496 с.

**Ситник О.О.**, професор, к.т.н., зав. кафедри електротехнічних систем Черкаського державного технологічного університету.

**Кисельова Г.О.**, старший викладач кафедри електротехнічних систем Черкаського державного технологічного університету.

**Кисельов В.Б.**, асистент кафедри електротехнічних систем Черкаського державного технологічного університету.