

**ЗАСТОСУВАННЯ МЕТОДУ КВАДРАТУР
ДЛЯ ЧИСЕЛЬНИХ РОЗВ'ЯЗКІВ ЛІНІЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ**

Ситник О. О., к.т.н., професор, завідувач кафедри електротехнічних систем,
Кисельова Г. О., старший викладач кафедри електротехнічних систем,
Кисельов В. Б., асистент кафедри електротехнічних систем
Черкаський державний технологічний університет
18006, м. Черкаси, бул. Шевченка, 460, annakis.777@yandex.ru

Анотація. Розглядається чисельний метод наближеного обчислення лінійних диференціальних рівнянь із застосуванням квадратурних формул Ньютона-Котеса різної точності.

Ключові слова: диференціальне рівняння, інтегральне рівняння, тест, Матлаб.

**APPLICATION OF QUADRATURE METHOD
FOR NUMERICAL CALCULATIONS OF LINEAR DIFFERENTIAL EQUATIONS**

Sytnyk O. O., Ph.D.(Engineering), professor, head of the electrical systems department,
Kyselyova G. O., senior lecturer of the electrical systems department,
Kyselyov V. B., assistant of the electrical systems department
Cherkasy State Technological University
18006, Cherkasy city, Shevchenko Boulevard, 460, annakis.777@yandex.ru

Abstract. The article deals with a numerical method for the approximate calculation of linear differential equations with the use of quadrature formulas of Newton-Cotes varying accuracy.

Keywords: differential equations, integral equations, test, Matlab.

Вступ. Диференціальні рівняння широко використовуються в багатьох областях науки та техніки для математичного моделювання різних фізичних процесів. Значну частину диференціальних рівнянь, що реально застосовуються для математичного моделювання динамічних систем, складають лінійні диференціальні рівняння. Для розв'язку лінійних диференціальних рівнянь використовують аналітичні, чисельно-аналітичні та чисельні методи.

Постановка проблеми. Методи чисельного диференціювання застосовуються, якщо вихідну функцію $f(x)$ важко, або неможливо продиференціювати аналітично. Існує багато різних методів чисельного диференціювання функцій. Для розв'язання лінійних диференціальних рівнянь чисельними методами найбільше поширення отримав метод Рунге-Кутти. Цей метод, або його модифікації використовує переважна більшість сучасних математичних програм. Так в програмі MATLAB існує два солвери [1] – ode45 (використовується явний метод Рунге-Кутти 4-го та 5-го порядків) та ode23 (використовується явний метод Рунге-Кутти 2-го та 3-го порядків), в програмі MATHCAD [2] – rkfixet та rkadapt (використовується метод Рунге-Кутти 4-го порядку з фіксованим та змінним кроком відповідно). Цей метод не є швидкодіючим, але практично завжди дає непогані результати. **Основним недоліком** даного методу є невисока точність обчислень. Для зменшення похибки та підвищення точності можлива зміна кроку інтегрування, або підвищення порядку методу. **Основною проблемою** чисельних розв'язків диференціальних рівнянь з фіксованим кроком інтегрування є значне збільшення кількості обчислень при підвищенні порядку методу, що зменшує швидкодію програм (тому наявні алгоритми обмежуються 4-5 порядком).

Шляхи вирішення проблеми. Одним з можливих способів розв'язку лінійних диференціальних рівнянь є перехід від диференціальних до інтегральних рівнянь Вольтери 2-го роду, які можуть бути розв'язані чисельно методом квадратур. Це дає можливість використати переваги інтегральних методів при розв'язанні лінійних диференціальних рівнянь [3, 4, 5]. Перева-

гою чисельного розв'язку рівнянь Вольтери 2-го роду із застосуванням квадратурних формул Ньютона-Котеса різної точності є незалежність кількості обчислень від порядку застосованих формул [6], що забезпечує зменшення похибки при незмінному часі обчислень.

Основними способами переходу від диференціальних до інтегральних рівнянь Вольтери 2-го роду є методи старшої похідної [3, 4] та послідовного інтегрування [4].

Метод старшої похідної [3, 4] полягає в представленні лінійного диференціального рівняння виду

$$D[y] \equiv y^{(p)}(x) + \sum_{k=1}^p a_k(x) \cdot y^{(k-1)}(x) = f(x), \quad y^{(k)}(0) = C_k, \quad k = \overline{0, p-1}, \quad (1)$$

де C_k – початкові умови, у вигляді інтегрального рівняння Вольтери другого роду:

$$u(x) - \int_0^x K(x, s) \cdot u(s) ds = j(x), \quad (2)$$

де $K(x, s)$ – ядро інтегрального рівняння:

$$K(x, s) = -\sum_{k=1}^p a_k(x) \cdot \frac{(x-s)^{k-1}}{(k-1)!}; \quad (3)$$

$j(x)$ – права частина:

$$j(x) = f(x) - C_{p-1} \cdot a_1(x) - (C_{p-1} \cdot x + C_{p-2}) \cdot a_2 - \dots - \left(C_{p-1} \cdot \frac{x^{p-1}}{(p-1)!} + C_1 \cdot x + C_0 \right) \cdot a_p(x). \quad (4)$$

При такому підході визначається функція $u(x) = y^{(p)}(x)$ (старша похідна), а для отримання значення шуканої функції $y(x)$ необхідно проводити додаткове інтегрування $u(x)$. Формула інтегрування $u(x)$ буде такою:

$$y(x) = C_{p-1} \cdot \frac{x^{p-1}}{(p-1)!} + \dots + C_1 \cdot x + C_0 + \frac{1}{(p-1)!} \cdot \int_0^x (x-s)^{p-1} u(s) ds. \quad (5)$$

Метод послідовного інтегрування [4] застосовується для розв'язку диференціальних рівнянь з постійними коефіцієнтами A_k . Здійснюючи послідовне інтегрування лінійного диференціального рівняння, отримуємо еквівалентне інтегральне рівняння виду

$$y(x) - \int_0^x K(x, s) \cdot y(s) ds = y(x), \quad (6)$$

де $K(x, s)$ – ядро інтегрального рівняння:

$$K(x, s) = -\sum_{k=1}^p A_k \cdot \frac{(x-s)^{k-1}}{(k-1)!}, \quad (7)$$

$y(x)$ – права частина:

$$y(x) = \frac{1}{(n-1)!} \int_0^x (x-s)^{p-1} \cdot f(s) ds + \sum_{k=0}^{p-1} C_k \cdot \frac{x^k}{k!} + C_0 \cdot \sum_{k=0}^{p-1} A_k \cdot \frac{x^k}{k!} + C_1 \cdot \sum_{k=0}^{p-2} A_k \cdot \frac{x^{(k+1)}}{(k+1)!} + \dots + C_{p-2} \cdot \frac{x^{(p-1)}}{(p-1)!}. \quad (8)$$

Постановка завдання. Дані методи використовуються для чисельно-аналітичних розв'язків диференціальних рівнянь (спочатку аналітично отримують еквівалентне інтегральне рівняння, а потім проводиться його чисельний розв'язок методом квадратур) тому **метою даної роботи** є отримання рекурентних формул, на основі яких можливе створення програм для чисельного розв'язку диференціальних рівнянь методом квадратур.

Основна частина. Аналіз формул ядра (3, 7) та правих частин (4, 8) еквівалентних інтегральних рівнянь показує, що вони можуть бути представлені у матричному вигляді і процес

переходу від диференціального до інтегрального рівняння можемо проводити чисельно. В результаті такого підходу отримуємо методи чисельного розв'язку диференціальних рівнянь з використанням квадратурних формул.

Якщо розв'язок диференціального рівняння проводиться на проміжку від 0 до b з кроком h , то кількість розрахункових точок x_i

$$n = \frac{b}{h} + 1.$$

Ядро (3) інтегрального рівняння для методу старшої похідної представимо рекурентною формулою

$$K_{ij} = - \sum_{j=1}^i \sum_{k=1}^p a_k(x_i) \cdot \frac{(x_i - x_j)^{k-1}}{(k-1)!}, \quad i = \overline{1, n}. \quad (9)$$

Праву частину (4) інтегрального рівняння за методом старшої похідної також представимо рекурентною формулою

$$j_i = f(x_i) - \sum_{k=1}^p \left[a_k(x_i) \cdot \sum_{g=0}^{k-1} C_{p-1-g} \cdot \frac{x_i^{k-1-g}}{(k-1-g)!} \right], \quad i = \overline{1, n}. \quad (10)$$

Отримане рівняння Вольтери (2) для старшої похідної $u(x_i) = y^{(p)}(x_i)$ розв'язуємо з використанням рекурентної формули [3, 4, 5]

$$u_i = \frac{f_i + \sum_{j=1}^{i-1} A_j \cdot K_{ij} \cdot u_j}{1 - A_i \cdot K_{ii}}, \quad i = \overline{1, n}, \quad (11)$$

за умови

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

де A_j – коефіцієнти застосованої квадратурної формули інтегрування.

Для чисельного розв'язку використовуємо формули Ньютона-Котеса та алгоритм представлений в [6].

Значення шуканої функції $y(x)$ (5) представимо рекурентною формулою

$$y_i = c_i + \frac{1}{(p-1)!} \cdot \sum_{j=1}^{i-1} A_j \cdot k_{ij}, \quad i = \overline{1, n}, \quad (12)$$

де k_{ij} – підінтегральний вираз:

$$k_{ij} = \sum_{j=1}^i (x_i - x_j)^{p-1} \cdot u_j, \quad i = \overline{1, n}, \quad (13)$$

c_i – вільний член:

$$c_i = \sum_{k=1}^p C_{k-1} \cdot \frac{x_i^{k-1}}{(k-1)!}, \quad i = \overline{1, n}. \quad (14)$$

Для остаточного визначення шуканої функції $y(x)$ використовуємо формули Ньютона-Котеса та алгоритм представлений в [6] і адаптований під відповідну рекурентну формулу.

Формування матриці ядра інтегрального рівняння за методом послідовного інтегрування (7) аналогічно методу старшої похідної (3), тому обчислення проводимо за формулою (9).

Праву частину інтегрального рівняння (8) за методом послідовного інтегрування представляємо рекурентною формулою

$$y_i = z_i + \frac{1}{(p-1)!} \cdot \sum_{j=1}^{i-1} A_j \cdot r_{ij}, \quad i = \overline{1, n}, \quad (15)$$

де r_{ij} – підінтегральний вираз:

$$r_{ij} = \sum_{j=1}^i (x_i - x_j)^{p-1} \cdot f_j, \quad i = \overline{1, n}, \quad (16)$$

c_i – вільний член:

$$z_i = \sum_{g=0}^{p-1} C_g \cdot \frac{x_i^g}{g!} + \sum_{g=0}^{p-2} \left(C_g \cdot \sum_{k=1}^{p-1-g} a_k \cdot \frac{x_i^{k+g}}{(k+g)!} \right) \quad i = \overline{1, n}. \quad (17)$$

Отримані результати. На основі отриманих рекурентних формул (9)-(17), розроблено програми `voltdifn` та `voltdify`, в яких для чисельних розв’язків рівнянь Вольтери другого роду використано алгоритм `voltnew` [6] з застосуванням формул Ньютона-Котеса різної точності.

Відмінність програми `voltdifn` від `voltdify` полягає в тому, що в ній розв’язок диференціальних рівнянь проводиться за методом старшої похідної, а в програмі `voltdify` – за методом послідовного інтегрування, що впливає на можливості застосування програм та величину похибки обчислень.

В програмах використовується метод квадратур на основі алгоритму `voltnew` [6], в якому передбачена можливість вибору квадратурної формули від двох- до дев’ятиточкової формули Ньютона-Котеса, що підвищує універсальність даного алгоритму та дозволяє відслідковувати зміну величини похибки від точності апроксимуючої формули. Враховуючи, що в даному алгоритмі значення y_2 обчислюється за формулою трапецій, яка має достатньо велику похибку, його доопрацьовано для отримання більш точного значення функції шляхом зменшення початкового кроку розрахунку. Таким чином в програмах `voltdifn` та `voltdify` використано квадратурний алгоритм з уточненням y_2 , y_3 та y_4 , значення яких обчислюються з меншим кроком та за формулами вищої точності.

На тестових задачах [2, 3] проводимо порівняння штатних солверів `ode45` та `ode23` MATLAB та отриманих програм `voltdifn` і `voltdify`. Визначаємо похибку та кількість точок, в яких обчислено функцію для отримання необхідної точності.

Задача 1 (приклад 1.14 [3]):

$$\begin{aligned} y''(x) + y'(x) + y(x) &= 3 \cdot e^{-2x}, \\ y'(0) &= -2, \quad y(0) = 1, \end{aligned}$$

точний розв’язок:

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

Задача 2 (приклад 6.16 [2]):

$$\begin{aligned} y'' + 2 \cdot y' + 10 \cdot y &= \sin(x); \\ y(0) &= 1; \quad y'(0) = 0, \end{aligned}$$

точний розв’язок:

$$\begin{aligned} y(x) &= -\frac{2}{85} \cdot \left(\cos(x) - \frac{9}{2} \cdot \sin(x) \right) + \\ &+ \frac{87}{85} \cdot e^{-x} \cdot \left(\cos(3 \cdot x) + \frac{26}{87} \cdot \sin(3 \cdot x) \right). \end{aligned}$$

Абсолютна похибка обчислень Δ_{abs} визначається як модуль різниці між точним значенням диференціального рівняння $y_m(x)$ та наближеним $y_i(x)$, отриманим відповідним чисельним методом

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

Результати розв’язку задач 1 та 2 програмами `voltdifn` (за методом старшої похідної) та `voltdify` (за методом послідовного інтегрування), а також штатних солверів MATLAB, для максимальних значень $\max \Delta_{abs}$ похибки, при різній кількості точок розрахунку n , зведені в табл. 1. Діаграми найбільш характерних випадків максимальних значень $\max \Delta_{abs}$ похибки представлені на рис. 1. Поведінку похибки Δ_{abs} найбільш характерних випадків зображено на рис. 2, 3.

Розв’язок задач штатними солверами програми MATLAB здійснювався з підбором величини абсолютної (`AbsTol`) та відносної (`RelTol`) похибки так, щоб кількість точок розрахунку мінімально відрізнялась і порівняння отриманих даних було найбільш коректним.

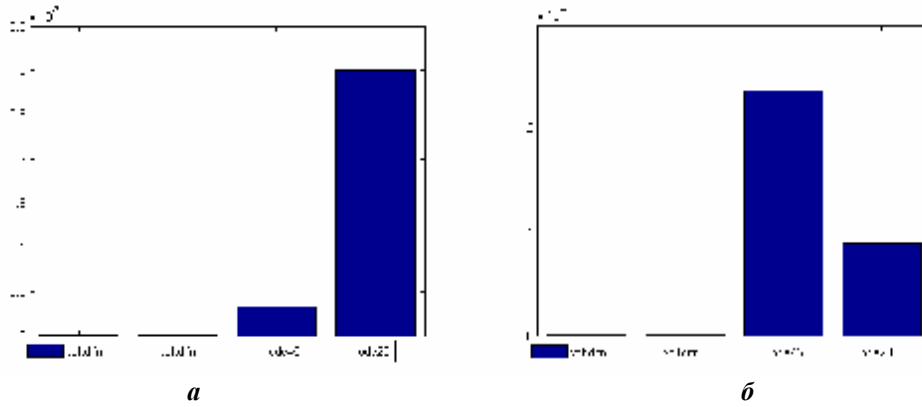


Рис. 1. Максимальна величина похибки обчислень програм voldifn, voldtify та ode113: *а* – задача 1 на проміжку [0; 1]; *б* – задача 2 на проміжку [0; 5].

Таблиця 1

Результати розв'язку задач 1 та 2

Програма	Кількість точок обчислень, <i>n</i>	Максимальна похибка обчислень, $\max \Delta_{abs}$	Кількість точок обчислень, <i>n</i>	Максимальна похибка обчислень, $\max \Delta_{abs}$
voldifn	41	1.3517e-10	101	2.7099e-07
voldtify	41	7.8723e-11	101	3.3356e-07
ode45	41	3.3439e-08	101	2.3669e-04
ode23	63	3.0041e-07	116	8.8910e-05

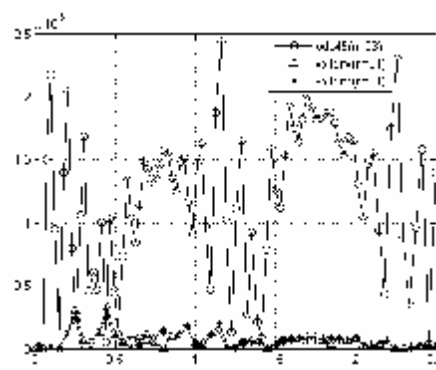
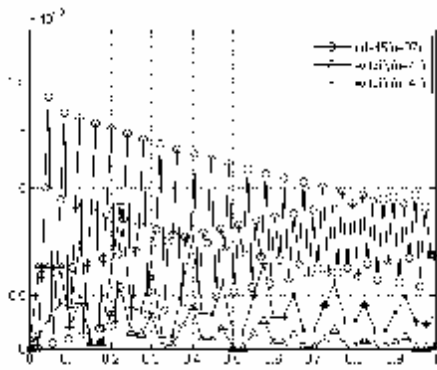


Рис. 2. Графіки зміни величини похибки обчислень програм voldifn, voldtify та ode45: *а* – задача 1 на проміжку [0; 1]; *б* – задача 2 на проміжку [0; 2,5].

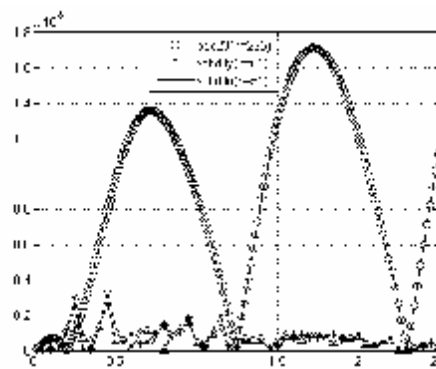
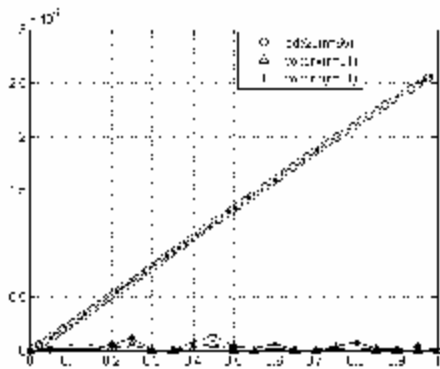


Рис. 3. Графіки зміни величини похибки обчислень програм voldifn, voldtify та ode23: *а* – задача 1 на проміжку [0; 1]; *б* – задача 2 на проміжку [0; 5].

Висновки. Мінімальну похибку обчислень тестової задачі 1 ($\max \Delta_{abs} = 7,8723 \cdot 10^{-11}$) отримано програмою `voldify`, другий результат ($\max \Delta_{abs} = 1,3517 \cdot 10^{-10}$) – програмою `voldifn`. Мінімальну похибку обчислень тестової задачі 2 ($\max \Delta_{abs} = 2,7099 \cdot 10^{-07}$) отримано програмою `voldifn`, другий результат ($\max \Delta_{abs} = 3,3356 \cdot 10^{-07}$) – отримано програмою `voldify`. Похибки штатних солверів MATLAB, при обчисленні тестових задач, на декілька порядків більші за похибки програм із застосуванням методу квадратур. На основі цього можемо зробити висновок про високу точність обчислень, проведених за допомогою програм `voldifn` та `voldify`, в яких використовувалась восьмиточкова формула Ньютона-Котеса, що підтверджує ефективність застосування формул високого порядку для розв'язку як диференціальних, так і інтегральних рівнянь методом квадратур [6]. Таким чином, запропонований метод може ефективно використовуватись для чисельного розв'язку лінійних диференціальних рівнянь. Застосування квадратурних формул Ньютона-Котеса вищої точності дає можливість будувати алгоритми для чисельного розв'язку лінійних диференціальних рівнянь, які є більш ефективними за методи Рунге-Кутти, реалізовані програмою MATLAB.

Список літератури

1. Клетков Ю. Л., Клетков А. Ю., Шульц М. М., MATLAB 7 : программирование, численные методы. СПб. : БХВ-Петербург, 2005. – 752 с.
2. Макаров Е. Инженерные расчеты в Mathcad 15: Учебный курс. СПб. : Питер, 2011. – 400 с.
3. Верлань А. Ф., Сизиков В. С. Интегральные уравнения: методы, алгоритмы, программы. – К. : Наукова думка, 1986. – 544 с.
4. Верлань А. Ф., Москалюк С. С. Математическое моделирование непрерывных динамических систем. – К. : Наукова думка, 1988. – 288 с.
5. Арушанян И. О. Численное решение интегральных уравнений методом квадратур. Практикум на ЭВМ. – М. : МГУ, 2002. – 72 с.
6. Ситник О. О., Кисельова Г. О., Кисельов В. Б. „Універсальний алгоритм розрахунку інтегрального рівняння Вольтерри II роду із застосуванням формул Ньютона-Котеса”, Вісник ЧДТУ, 2010, №3. – С. 36–42.

References

1. Kletkov Yu. L., Kletkov A. Yu., Shul'ts M. M. MATLAB 7: programmirovaniye, chislennyye metody. SPb. : BKhV-Peterburg, 2005. – 752 p.
2. Makarov E. Inzhenernyye raschety v Mathcad 15: Uchebnyy kurs. SPb. : Piter, 2011. – 400 p.
3. Verlan' A. F., Sizikov V. S. Integral'nyye uravneniya: metody, algoritmy, programmy. – K. : Naukova dumka, 1986. – 544 p.
4. Verlan' A. F., Moskalyuk S. S. Matematicheskoye modelirovaniye nepreryvnykh dinamicheskikh sistem. – K. : Naukova dumka, 1988. – 288 p.
5. Arushanyan I. O. Chislennoye resheniye integral'nykh uravneniy metodom kvadratur. Praktikum na EVM. – M.: MGU, 2002. – 72 p.
6. Sytnyk O. O., Kysel'ova G. O., Kysel'ov V. B. „Universal'nyy algorytm rozrahunku integral'nogo rivnjannja Vol'terry II rodu iz zastosuvannjam formul N'jutona-Kotesa”, Visnyk ChDTU, 2010, №3. – P. 36–42.

Стаття надійшла до редакції 18.04.2013.

Відомості про авторів:

Ситник О. О., кандидат технічних наук, професор, завідувач кафедри електротехнічних систем, Черкаський державний технологічний університет

Кисельова Г. О., старший викладач кафедри електротехнічних систем, Черкаський державний технологічний університет

Кисельов В. Б., асистент кафедри електротехнічних систем, Черкаський державний технологічний університет