

УДК 004.41+004.94

DOI: 10.32626/2308-5916.2023-24.26-34

В. А. Іванюк, д-р техн. наук,**М. О. М'ястковська**, канд. пед. наук,**В. В. Понеділок**, канд. техн. наукКам'янець-Подільський національний університет
імені Івана Огієнка, м. Кам'янець-Подільський

АВТОМАТИЗОВАНІ ЗАСОБИ ТЕСТУВАННЯ ПРОГРАМНИХ МОДУЛІВ РОЗВ'ЯЗУВАННЯ ІНТЕГРАЛЬНИХ РІВНЯНЬ ВОЛЬТЕРРИ ДРУГОГО РОДУ

У статті розглянута методологія автоматизованого тестування програмних модулів розв'язування інтегральних рівнянь Вольтерри другого роду. Для реалізації автоматизованого тестування вибрано програмне середовище Matlab, яке має широкий спектр можливостей тестування програмного забезпечення, зокрема: функції для генерації наборів даних для тестування; функції для порівняння результатів тестування; функції для формування звітів тестування тощо. У якості бази для розробки автоматизованих засобів тестування вибрано Unit Testing Framework, що є складовою MATLAB Test Framework, має багато готових методів для перевірки коректності значень і формування статистики помилкових випадків. Розроблено набір тестових задач для інтегральних рівнянь Вольтерри другого роду, який поділяється на різні типи рівнянь, включаючи лінійні інтегральні рівняння Вольтерри другого роду, які містять ядра із степеневими, експоненціальними, гіперболічними, логарифмічними, тригонометричними, оберненими тригонометричними функціями та їх комбінації. Розроблені засоби автоматизованого тестування використано для перевірки якості програмних модулів, побудованих на основі методів лівих прямокутників, правих прямокутників, методу трапецій та методу Сімпсона. Розроблений набір тестових задач дозволяє покривати широкий спектр можливих умов роботи програмних модулів. Результати тестування дозволили виявити шляхи вдосконалення наявних програмних модулів для досягнення виставлених умов їх функціонування. Проведенні дослідження будуть сприяти розробці більш надійних і ефективних програмних модулів розв'язування інтегральних рівнянь Вольтерри другого роду. Отримані результати є основою для подальших досліджень, які будуть спрямовані в таких напрямках: розробка методів тестування для більш складних типів інтегральних рівнянь Вольтерри другого роду, включаючи рівняння з нелінійними та нестационарними ядрами.

Ключові слова: *автоматизоване тестування, інтегральні рівняння, програмні засоби, очікуваний розв'язок, набір тестових завдань, математичне моделювання.*

Вступ. Інтегральні рівняння Вольтерри другого роду є важливим математичним інструментом, який широко використовується в різних галузях науки і техніки, зокрема в математичній фізиці, механіці, електротехніці та ін. Вони використовуються для моделювання широкого спектру динамічних процесів, таких як поширення хвиль, коливання, управління системами тощо [2, 8, 9, 10].

Ефективне розв'язання інтегральних рівнянь Вольтерри другого роду є складною задачею. Воно вимагає використання чисельних методів, які можуть бути досить складними та чутливими до параметрів [7]. Тому розробка надійних і точних програмних модулів для розв'язування таких рівнянь є важливою задачею.

Наявні сучасні математичні пакети прикладних програм не містять серійних засобів для чисельної реалізації інтегральних рівнянь. Винятками є вузькопрофільні бібліотеки, які направлені на розв'язування певних типів інтегральних рівнянь і не забезпечують універсальності [5, 11].

Розробка програмних модулів для розв'язування інтегральних рівнянь Вольтерри другого роду є складною задачею, яка вимагає значних зусиль і часу. Тому важливо використовувати ефективні методи тестування, які дозволяють виявити помилки на ранніх етапах розробки.

Автоматизовані засоби тестування дозволяють значно підвищити ефективність процесу тестування програмного забезпечення. Вони дозволяють генерувати набори тестових випадків автоматично, що дозволяє покривати більш широкий спектр можливих умов роботи програмного модуля. Крім того, автоматизовані засоби тестування дозволяють легко аналізувати результати тестування та виявляти помилки [1, 3].

Метою статті є розробка методів автоматизованого тестування програмних модулів розв'язування інтегральних рівнянь Вольтерри другого роду.

Програмне середовище. В межах дослідження в якості середовища моделювання вибрано Matlab, який є потужним інструментом для наукових досліджень в галузі моделювання, а також має широкий спектр можливостей для тестування програмних засобів. До них відносяться: функції для генерації наборів даних для тестування; функції для порівняння результатів тестування; функції для формування звітів тестування тощо.

Matlab пропонує широкий спектр бібліотек для автоматизації тестування, які дозволяють розробникам створювати ефективні та надійні

тестові набори, зокрема це: MATLAB Test Framework – базова бібліотека для автоматизації тестування, яка надає широкий спектр функцій для створення та запуску тестових наборів; MATLAB Test Explorer – графічний інтерфейс користувача, який дозволяє розробникам створювати та запускати тестові набори без необхідності писати код; MATLAB Test Executor – бібліотека, яка дозволяє виконувати тестові набори на віддаленій машині; MATLAB TestNG – бібліотека, яка надає підтримку для тестування на основі сценаріїв; MATLAB JUnit – бібліотека, яка надає підтримку для тестування на основі класів [4].

Розглянемо детальніше, як переваги Matlab можна використати для тестування програмних модулів розв'язування інтегральних рівнянь Вольтерри другого роду. Matlab може використовуватися для генерації наборів даних, що містять різні типи інтегральних рівнянь Вольтерри другого роду, а також різні значення параметрів цих рівнянь. Це дозволить перевірити, здатність програмного модуля розв'язувати інтегральні рівняння різних типів і з різними параметрами. Порівняння результатів тестування здійснюється шляхом порівняння результатів розв'язання інтегральних рівнянь програмним модулем з очікуваними значеннями. Це дозволяє виявити помилки в алгоритмі розв'язання рівнянь, що використовується програмним модулем. Matlab має засоби для ефективного формування звітів про результати тестування, що містять інформацію про типи інтегральних рівнянь, які були розв'язані програмним модулем. Ці звіти будуть основою для аналізу результатів тестування та виявлення тенденцій.

Unit Testing Framework. Основою для розробки автоматизованих засобів тестування вибрано Unit Testing Framework, що є складовою MATLAB Test Framework. Unit Testing Framework має багато готових методів для перевірки коректності значень і для виведення повідомлень про помилки. Unit Testing Framework створений в середовищі xUnit, тобто в ньому можна створювати різні набори тестів, що мають можливість попередніх дій перед запуском цих тестів та після їх запуску для того, щоб порядок виконання тестів не мав значення.

Інструментарій дозволяє писати юніт-тести, виконувати їх в автоматичному режимі, збирати результати, порівнювати їх з очікуваними результатами і візуалізувати загальний пробний пуск. Тести можна запускати окремо або згруповані в набір тестів. Для того щоб написати відповідні модульні тести, у вікні інструментів знаходяться декілька функцій, які викликають повідомлення про помилку, коли результат тестуючої функції не відповідає вимогам [4, 5].

Отже, в середовищі Unit Testing Framework розроблено набір тестів для виявлення помилкових ситуацій на основі вхідних даних та очікуваних результатів програмних модулів розв'язування інтегральних рівнянь Вольтерри другого роду.

Тестові засоби розроблені у вигляді класів. Кожен клас визначає інтегральне рівняння Вольтерри другого роду, яке має типовий вигляд [12].

$$y(x) - \int_a^x K(x,s)y(s)ds = f(x),$$

де $K(x,s)$ – ядро інтегрального рівняння, $f(x)$ – функція лівої частини. Також в кожному класі визначаються $f_T(x)$ – функція точного розв’язку, a – значення нижньої межі інтегрування.

Для дослідження якості розроблених модулів розроблено тести на основі модельних задач із порівнянням з очікуваними розв’язками, враховуючи різні типи похибок: абсолютна, відносна, інтегральна, також обмеження накладаються на час виконання поставленого завдання. Перед використанням комплексу тестів створюється структура із такими значеннями: крок розв’язування, кінцевий час моделювання, необхідні абсолютна, відносна та інтегральні похибки, час моделювання, а також посилання на програмний модуль, який підлягає тестуванню.

Набір тестових задач. Для того, щоб використовувати можливість Matlab для тестування програмних модулів розв’язування інтегральних рівнянь Вольтерри другого роду, ключовою проблемою є розробка наборів тестових випадків, що покриває широкий спектр можливих умов роботи програмного модуля. Набір тестових випадків розробляється на основі знань про теорію інтегральних рівнянь Вольтерри другого роду [7], а також на основі власного досвіду розробки програмних модулів для розв’язування таких рівнянь [12].

Модельні задачі поділяються на різні типи рівнянь. Розглядаються лінійні рівняння, які містять ядра із степеневими, експоненціальними, гіперболічними, логарифмічними, тригонометричними, оберненими тригонометричними функціями та їх комбінації. Також розглядаються інтегральні рівняння Вольтерри другого роду з виродженими та різницевиими ядрами.

Набір інтегральних рівнянь та їх точні розв’язки, які використовуються для тестування програмних засобів розв’язування інтегральних рівнянь Вольтерри другого роду, подано в таблиці 1.

Таблиця 1

Тестові випадки для інтегральних рівнянь Вольтерри другого роду

№	Інтегральне рівняння	Точний розв’язок
1	$y(x) = e^x + \int_0^x e^{x-t} y(t) dt$	$y(x) = e^{2x}$

Продовження таблиці 1

2	$y(x) = \sin x + 2 \int_0^x e^{x-t} y(t) dt$	$y(x) = \frac{1}{5} e^{3x} - \frac{1}{5} \cos x + \frac{2}{5} \sin x$
3	$y(x) = x3^x - \int_0^x 3^{x-t} y(t) dt$	$y(x) = 3^x (1 - e^{-x})$
4	$y(x) = 1 - 2x - \int_0^x e^{x^2-t^2} y(t) dt$	$y(x) = e^{x^2-x} - 2x$
5	$y(x) = e^{x^2+2x} + 2 \int_0^x e^{x^2-t^2} y(t) dt$	$y(x) = e^{x^2+2x} (1 + 2x)$
6	$y(x) = 1 + x^2 + \int_0^x \frac{1+x^2}{1+t^2} y(t) dt$	$y(x) = e^x (1 + x^2)$
7	$y(x) = xe^{\frac{x^2}{2}} + \int_0^x e^{-(x-t)} y(t) dt$	$y(x) = e^{\frac{x^2}{2}} (x+1) - 1$
8	$y(x) = e^{-x} + \int_0^x e^{-(x-t)} \sin(x-t) y(t) dt$	$y(x) = e^{-x} \left(\frac{x^2}{2} + 1 \right)$
9	$y(x) = 1 - \int_0^x ty(t) dt$	$y(x) = e^{-\frac{x^2}{2}}$
10	$y(x) = x + \int_0^x xty(t) dt$	$y(x) = xe^{\frac{x^3}{3}}$
11	$y(x) = \operatorname{ch} x + \int_0^x \frac{\operatorname{ch} x}{\operatorname{ch} t} y(t) dt$	$y(x) = e^x \operatorname{ch} x$
12	$y(x) = \frac{1}{1+x^2} + \int_0^x \frac{1+t^2}{1+x^2} y(t) dt$	$y(x) = \frac{e^x}{1+x^2}$
13	$y(x) = e^x + 2 \int_0^x \cos(x-t) y(t) dt$	$y(x) = e^x (1+x)^2$
14	$y(x) = x - \int_0^x e^{x-t} y(t) dt$	$y(x) = x - \frac{x^3}{2}$

Продовження таблиці 1

15	$y(x) = e^{2x} + \int_0^x e^{t-x} y(t) dt$	$y(x) = \frac{1}{2}(3e^{2x} - 1)$
16	$y(x) = x - \int_0^x (x-t) y(t) dt$	$y(x) = \sin x$
17	$y(x) = \cos x - \int_0^x (x-t) \cos(x-t) y(t) dt$	$y(x) = \frac{1}{3}(2 \cos \sqrt{3}x + 1)$
18	$y(x) = 1 + x + \int_0^x e^{-2(x-t)} y(t) dt$	$y(x) = \frac{x}{2} - \frac{1}{4} + \frac{1}{4} e^{-2x}$
19	$y(x) = x + \int_0^x \sin(x-t) y(t) dt$	$y(x) = x + \frac{x^3}{6}$
20	$y(x) = \sin x + \int_0^x (x-t) y(t) dt$	$y(x) = \frac{1}{2} \sin x + \frac{1}{2} \operatorname{sh} x$
21	$y(x) = x - \int_0^x \operatorname{sh}(x-t) y(t) dt$	$y(x) = x - \frac{x^3}{6}$
22	$y(x) = 1 - 2x - 4x^2 + \int_0^x (3 - 6(x-t) - 4(x-t)^2) y(t) dt$	$y(x) = x$
23	$y(x) = \operatorname{sh} x - \int_0^x \operatorname{ch}(x-t) y(t) dt$	$y(x) = \frac{2}{\sqrt{5}} \operatorname{sh} \frac{\sqrt{5}}{2} x$
24	$y(x) = 1 + 2 \int_0^x \cos(x-t) y(t) dt$	$y(x) = 1 + 2xe^x$
25	$y(x) = 4x + 2 \int_0^x \frac{1}{2t+1} y(t) dt$	$y(x) = (4x+2) \ln(2x+1) + 4x$
26	$y(x) = 1 + \int_0^x \frac{\sin x}{\cos t} y(t) dt$	$y(x) = \sin^2 x + 1$
27	$y(x) = \cos x e^x + \int_1^x \frac{x \cos x}{t \cos t} y(t) dt$	$y(x) = (x \ln x + 1) \cos x e^x$

Продовження таблиці 1

28	$y(x) = x^3 \cos x + \int_{\pi}^x \frac{x^2}{t^3} y(t) dt$	$y(x) = x^3 (\sin x + \cos x)$
29	$y(x) = 1 + \int_e^x \frac{2}{t \ln x} y(t) dt$	$y(x) = 2 \ln x - 1$
30	$y(x) = e^{x-\sin x} + \int_0^x \cos x e^{x-t} y(t) dt$	$y(x) = (1 - x \cos x) e^{x-\sin x}$

Тестування засобів розв'язування інтегральних рівнянь Вольтерри II роду. Розроблені засоби автоматизованого тестування використано для перевірки якості програмних модулів. Розглянуто випадок застосування програмних модулів побудованих на основі методів лівих прямокутників, правих прямокутників, методу трапецій та методу Сімпсона [6]. Вимоги, що виставлені до програмних засобів у даному випадку, одночасно включали значну вимогу щодо точності результатів розв'язування та значне обмеження часу виконання. Результати тестування подано в таблиці 2.

Таблиця 2

Результати тестувань модулів розв'язування інтегральних рівнянь Вольтерри другого роду

Програмний засіб	Метод	Passed	Failed
Volt_II_Rect_L	Лівих прямокутників	14	16
Volt_II_Rect_R	Правих прямокутників	14	16
Volt_II_Rect_T	Трапецій	20	10
Volt_II_Rect_S	Сімпсона	18	12

Отриманні результати показали працездатність всіх тестованих програмних модулів. Однак, значна кількість невдалих тестів (таблиця 2) вказує, що всі програмні засоби не можуть бути використанні в умовах, які визначалися перед тестуванням. Детальний огляд результатів тестування показує, що тестові випадки, які не пройшли вимоги – це розв'язки, із швидкозростаючими функціями, наприклад, тестовий випадок 1: функція розв'язку має вигляд $y(x) = e^{2x}$, де із зростанням аргументу x швидко зростає значення функції $y(x)$. Вплив на цей результат здійснює, як функція вільної частини рівняння $f(x) = e^{2x}$, так і ядро інтегрального оператора e^{x-t} .

Виходом із цієї ситуації можливе застосування ряду алгоритмів, зокрема, застосування змінного кроку моделювання. Але у цьому

випадку слід враховувати можливе значне зростання часу виконання, для вирішення якого слід застосувати підхід з апроксимаційним перетворенням ядра інтегрального оператора та зведення його до виродженого вигляду. Такий підхід теж пов'язаний із певними недоліками – будь-які апроксимаційні перетворення призводять до втрати точності обчислень.

Загалом, підготовка програмних модулів для розв'язування прикладних задач повинна здійснюватись із врахуванням всіх критичних умов, а наявність множини тестових задач, дозволяє виявити проблеми на етапі розробки.

Висновки. Запропонована методологія автоматизованого тестування дозволяє ефективно перевіряти якість програмних модулів розв'язування інтегральних рівнянь Вольтерри другого роду. Розроблено набір тестових задач для інтегральних рівнянь Вольтерри другого роду, який поділяється на різні типи рівнянь, включаючи лінійні інтегральні рівняння Вольтерри другого роду з виродженими та різницевиими ядрами. Розроблені засоби автоматизованого тестування використано для перевірки якості програмних модулів, побудованих на основі методів лівих прямокутників, правих прямокутників, методу трапецій та методу Сімпсона, та дозволили виявити проблемні ситуації та знайти шляхи для їх розв'язання. Проведені дослідження будуть сприяти розробці більш надійних і точних програмних модулів розв'язування інтегральних рівнянь Вольтерри другого роду. Отримані результати є основою для подальших досліджень: розробка методів тестування для більш складних типів інтегральних рівнянь Вольтерри другого роду, включаючи рівняння з нелінійними ядрами, з нестационарними ядрами.

Список використаних джерел:

1. Jackvony K. The Complete Software Tester: Concepts, Skills, and Strategies for High-Quality Testing. Independently published. 512 p.
2. Rainer Kress. Linear Integral Equations: Springer, 3rd ed. 2014. 428 p.
3. Stormy Attaway, Boston University. MATLAB: A Practical Introduction to Programming and Problem Solving. Butterworth-Heinemann, 2022. 592 p.
4. Testing Frameworks / Help Center Matlab. URL: <https://www.mathworks.com/help/matlab/matlab-unit-test-framework.html>
5. Yair Altman. Accelerating MATLAB Performance: 1001 Tips to Speed Up MATLAB Programs. CRC Press, Inc., 2014. 792 p.
6. Верлань А. Ф., Контрерас Д. Е., Сізіков В. С. та ін. Integral equation toolbox – пакет програм для розв'язування інтегральних рівнянь в середовищі Matlab. Київ, 1997. 44 с.
7. Верлань А. Ф., Сизиков В. С. Интегральные уравнения: методы, алгоритмы, программы. Київ, 1986. 544 с.

8. Верлань А. Ф., Федорчук В. А. Моделі динаміки електромеханічних систем. Київ, 2013. 221 с.
9. Верлань А. Ф., Федорчук В. А., Іванюк В. А. Комп'ютерне моделювання в задачах динаміки електромеханічних систем. Кам'янець-Подільський, 2010. 204 с.
10. Верлань А. А., Федорчук В. А. Моделювання комп'ютерно-інтегрованих силових енергетичних установок: монографія. Кам'янець-Подільський, 2023. 296 с.
11. Іванюк В. А. Математичні пакети прикладних програм : навчальний посібник. Кам'янець-Подільський, 2015. 160 с.
12. Федорчук В. А., Іванюк В. А., Верлань Д. А. Інтегральні рівняння в задачах математичного моделювання: навчальний посібник. Кам'янець-Подільський, 2014. 144 с.

AUTOMATED MEANS OF TESTING SOFTWARE MODULES FOR SOLVING VOLTERRA INTEGRAL EQUATIONS OF THE SECOND KIND

The article is part of the methodology of automated testing of software modules for solving Volterra integral levels of the second kind. For the implementation of automated testing, the Matlab software environment was selected, which has a wide range of software testing capabilities, in particular: functions for generating data sets for testing; functions for comparing test results; functions for generating test reports, etc. For the development of automated testing tools, the Unit Testing Framework was selected, which is a component of the MATLAB Test Framework, has many ready-made methods for checking the correctness of values and the formation of statistical errors. A set of test problems has been developed for Volterra integral equations of the second kind, which are divided into different types, including linear Volterra integral equations of the second kind, which consist of a kernel with power, exponential, hyperbolic, logarithmic, trigonometric, inverse trigonometric functions and their combination. Developed testing tools used for automated quality control of software modules built on the basis of left rectangle, right rectangle, trapezoidal, and Simpson methods. The developed set of test tasks covers a wide range of possible operating conditions of software modules. The results of the testing allowed to improve the existing software modules to achieve the set conditions for their operation. The conducted research should contribute to the development of more reliable and efficient software modules for solving Volterra integral levels of the second kind. The obtained results are the basis for further research, which will be used in the following directions: development of testing methods for more complex types of Volterra integral levels of the second kind, including the level with nonlinear and non-stationary kernels.

Key words: *automated testing, integral equation, software tools, expected solution, set of test tasks, mathematical modeling.*

Отримано: 15.11.2023