

УДК 519.6:004.94

DOI <https://doi.org/10.32782/tnv-tech.2023.2.14>

## ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ЗАСОБІВ ДОСЛІДЖЕННЯ ІНТЕГРАЛЬНИХ МОДЕЛЕЙ В СИСТЕМІ MATLAB

**Тихоход В. О.** – кандидат технічних наук,  
доцент кафедри цифрових технологій в енергетиці  
Навчально-наукового інституту атомної і теплової енергетики  
Національного технічного університету України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
ORCID ID: 0000-0002-1525-4770

Система комп'ютерної математики MATLAB надає багато можливостей для проведення різних досліджень. Однак в ній відсутні штатні засоби вирішення інтегральних рівнянь та їх систем. Числові реалізації систем інтегральних рівнянь Вольтера характеризуються значною кількістю обчислень та можуть вимагати великих масивів даних для збереження результатів обчислень, що вимагає оптимального використання пам'яті та максимально ефективно організації обчислень у модулях MATLAB. В роботі систематизовано підходи до створення модулів розширення системи MATLAB для обчислювальних задач різноманітного спрямування, описано шляхи побудови ефективних модулів за критеріями оптимізації використання пам'яті та швидкодії обчислень. Запропоновано процедуру виконання досліджень інтегральних моделей динамічних систем з використанням функцій MATLAB та описано сконструйовані модулі для розв'язання інтегральних рівнянь Вольтера та їх систем прямими методами наближеного розв'язання. Вони полягають у приведенні інтегральних рівнянь до більш простих рівнянь і систем, наприклад, систем лінійних алгебраїчних рівнянь, і подальшому вирішенні цих більш простих рівнянь. Описано способи представлення вихідних даних інтегральних моделей в системі MATLAB, що передбачають два способи опису вихідних структур: у табличній та аналітичній формі. Описано способи верифікації даних в середовищі MATLAB, що дозволяють забезпечити прогнозовану поведінку модулів та уникнути непередбачених ситуацій. Запропоновані в роботі рішення та рекомендації до побудови функцій в системі MATLAB знайшли застосування при побудові засобів дослідження інтегральних моделей у формі інтегральних рівнянь Вольтера. На прикладі реалізації методів квадратур та колокації до розв'язання інтегральних рівнянь показано особливості організації програмних модулів на мові програмування, що вбудована в систему MATLAB. Розробленим засобам характерно максимальне використання матрично-векторних обчислень, оптимальне використання пам'яті та безпечна перевірка та обробка коректності даних.

**Ключові слова:** функції MATLAB, прямі методи розв'язання, інтегральні рівняння Вольтера, засоби розв'язання інтегральних рівнянь, верифікація, векторно-матричні обчислення.

### **Tykhokhod V. O. Features of the implementation of integral models research tools in the MATLAB system**

The computer mathematics system MATLAB provides many opportunities for conducting various studies. However, it lacks inbuilt tools of solving integral equations and their systems. Numerical implementations of systems of Volterra integral equations require a significant number of calculations and may require large data sets to store the results of calculations, which requires optimal use of memory and the most efficient organization of calculations in MATLAB modules. The work systematizes approaches to creating modules for the MATLAB system expansion for computational tasks of various directions, describes the ways of building efficient modules according to the criteria of memory optimization and calculation speed. The procedure for conducting studies of integral models of dynamic systems using MATLAB functions is proposed and the designed modules for solving Volterra integral equations and their systems by direct methods of approximate solution are described. They consist in reducing integral equations to simpler equations and systems, for example, systems of linear algebraic equations, and further solving these simpler equations. Methods of presenting the initial data of integral models in the MATLAB system are described, which involve two

*ways of describing the initial structures: in tabular and analytical form. Methods of data verification in the MATLAB environment are described, which allows you to ensure the predicted behavior of modules and avoid unforeseen situations. The solutions and recommendations for the construction of functions in the MATLAB system proposed in the work were used in the construction of tools for the study of integral models in the form of Volterra integral equations. On the example of the implementation of quadrature and collocation methods for solving integral equations, the peculiarities of the organization of program modules in the programming language built into the MATLAB system are shown. The developed tools are characterized by maximum use of matrix-vector calculations, optimal use of memory, and safe verification and processing of data correctness.*

**Key words:** *MATLAB functions, direct methods, Volterra integral equation, tools for solving integral equations, verification, vector-matrix calculations.*

**Вступ.** Інтегральні рівняння Вольтера відносяться до класу непараметричних динамічних моделей [1], що найбільш ефективно описують процеси та об'єкти, задані динамічними характеристиками своїх елементів. Зокрема, опис динамічних систем на основі інтегральних рівнянь знайшов застосування в таких задачах динаміки як аналіз, дослідження та проектування систем керування, елементами яких є ланки з розподіленими параметрами. Інтегральний метод опису має такі позитивні якості як зручність та компактність математичного опису динамічних систем; високий рівень універсальності моделей; досить висока стійкість методів числової реалізації інтегральних залежностей.

На сучасному етапі існує ряд комп'ютерних програмних систем та пакетів програм для автоматизації математичних розрахунків, серед яких однією з найбільш універсальних та розвинутих є система комп'ютерної математики MATLAB [2]. Вона забезпечує автоматизацію математичних розрахунків, її основною характеристикою є вбудована підтримка матричних операцій.

Архітектура MATLAB дозволяє легко модифікувати та розширювати систему, що призвело до створення значної кількості пакетів прикладних програм (toolbox) для різних сфер науки і техніки. Це розширення досягається природним шляхом і реалізується в формі так званих *m*-файлів (з розширенням *.m*), що містять код програм на мові програмування MATLAB. Проте, попри розвиток та поширення на різні області, мало уваги приділяється реалізації апарату інтегральних рівнянь та їх систем. Наявність програмних засобів, які забезпечують вирішення завдань дослідження багатозв'язних динамічних систем на основі розв'язання систем інтегральних рівнянь, дозволяє значно розширити клас задач динаміки, що вирішуються комп'ютерними засобами. Проте особливості вбудованої мови програмування та інфраструктури системи MATLAB вимагають застосування спеціальних підходів для побудови ефективних програм.

**Аналіз останніх досліджень і публікацій.** Питанням комп'ютерної реалізації прямих методів наближеного розв'язання інтегральних рівнянь Вольтера присвячено різні роботи, зокрема [1; 3], їх систем – роботи [4; 5]. Але в існуючих роботах не розкриті питання конструювання ефективних модулів дослідження інтегральних моделей в системі MATLAB з врахування вбудованих можливостей системи моделювання та особливостей даного типу моделей.

**Мета.** У зв'язку з цим робота присвячена систематизації підходів до створення ефективних модулів розширення системи MATLAB для дослідження інтегральних моделей динамічних систем шляхом розв'язання інтегральних рівнянь та їх систем.

**Прямі методи наближеного розв'язання інтегральних рівнянь.** У загальному випадку система інтегральних рівнянь (CIP) записується у формі:

$$g_i(x)y_i(x) - \lambda \sum_{j=1}^n \int_{\Omega_j} K_{ij}(x,s)y_j(s)ds = f_i(x), x \in Q_i, i = \overline{1, n}. \quad (1)$$

де  $K(x,s)$  – ядра СР,  $f(x)$  – права частина рівняння з областю визначення  $Q$ ,  $\lambda$  – параметр рівняння,  $y(s)$  – шукана функція з областю визначення  $\Omega$ , що може бути змінною або постійною, одновимірною та багатовимірною. Функції  $K(x,s)$ ,  $f(x)$ ,  $g(x)$ , параметр  $\lambda$  та області  $Q$  та  $\Omega$  вважаються заданими, а функція  $y(s)$  – шуканою. У випадку змінної області визначення  $\Omega$  рівняння (1) є рівнянням Вольтера, при постійній області інтегрування мають справу з рівняннями Фредгольма. При  $g(x)=0$  рівняння (1) є рівнянням I роду, при  $g(x)\neq 0$  рівняння (1) є рівнянням II роду. Параметр  $i$  в рівнянні (1) вказує на кількість невідомих функцій, при  $i>0$  отримується система інтегральних рівнянь з кількістю рівнянь, що відповідає кількості невідомих функцій.

До основних традиційних методів наближеного розв'язання інтегральних рівнянь відносяться прямі методи. Вони полягають у приведенні інтегральних рівнянь до більш простих рівнянь і систем, наприклад систем лінійних алгебраїчних рівнянь, і подальшому вирішенні цих більш простих рівнянь. При розв'язанні інтегральних рівнянь Вольтера змінна верхня межа інтегрування, що характерна для цього типу рівнянь, в точках апроксимаційної сітки фіксується та виконується обчислення визначених інтегралів.

До прямих методів розв'язання інтегральних рівнянь та їх систем, зокрема, відносяться метод квадратур та метод колокацій. При числовій реалізації лінійних інтегральних рівнянь типу Вольтера II роду методом квадратур використовується [3] вираз:

$$\sum_{j=1}^m y_j^{(i)} (a_{rj} - K_{rji}^{(i)} A_i) = f_r^{(i)} + \sum_{j=1}^m \sum_{l=1}^{i-1} K_{rji}^{(l)} y_j^{(l)} A_l, \quad (2)$$

де

$$K_{rji}^{(l)} = K_{rj}(x_i, x_l), \quad y_j^{(l)} = y_j(x_l), \quad f_r^{(i)} = f_r(x_i),$$

$A_l$  – коефіцієнти квадратурної формули.

При розв'язанні систем лінійних інтегральних рівнянь типу Вольтера I роду методом колокацій з використанням кусково-гладких поліномів використовують [4] обчислювальний вираз

$$\sum_{j=1}^n \int_{x_{k,0}}^{x_{k,v}} K_{rij}(x_{k,v}, s) P_{j,k}(s) ds = f_{r,i}(x_{k,v}) - \psi_{r,i,k}(x_{k,v}), \quad (3)$$

де

$$\begin{aligned} \psi_{r,i,k}(x_{k,v}) = & \sum_{j=1}^n \int_a^{x_{1,0}} K_{rij}(x_{x,v}, s) P_{j,0}(s) ds + \\ & + \sum_{j=1}^n \int_{x_{1,0}}^{x_{2,0}} K_{rij}(x_{x,v}, s) P_{j,1}(s) ds + \dots + \\ & + \sum_{j=1}^n \int_{x_{k-1,0}}^{x_{k,0}} K_{rij}(x_{x,v}, s) P_{j,k-1}(s) ds, k = \overline{1, M-1}, v = \overline{1, l}. \end{aligned}$$

Після заміни в рівняннях (2) та (3) виразів обчисленими значеннями отримують системи лінійних алгебраїчних рівнянь (СЛАР), розв'язок яких є значення функцій  $y_r$  в точках  $x_i$ .

**Можливості MATLAB для розв'язання інтегральних рівнянь.** При розробці та проектуванні складних технічних систем з великою ефективністю застосовують комп'ютерні системи моделювання. Такі системи надають великий набір функцій, реалізують різні методи числового моделювання. Система MATLAB є однією з найбільш розвинутих, але в ній відсутні засоби розв'язання інтегральних рівнянь. Розширення MATLAB такими засобами потребує аналізу можливостей, що можуть бути використані для розв'язання інтегральних рівнянь.

До основних переваг MATLAB [5] відносяться: модульна структура, яка дозволяє розширювати можливості системи шляхом включення нових програмних модулів, зокрема написаних користувачем; внутрішню мову програмування, що орієнтована на операції у векторно-матричній формі; інтерпретатор, що дозволяє зручно виконувати команди та одразу отримувати їх результат; потужну систему візуалізації результатів досліджень.

Підтримка матрично-векторних обчислень полягає в тому, що дані в MATLAB можна представляти як матриці та вектори, а операції над цими структурами даних можна виконувати у векторизованому вигляді. Це забезпечує простоту та виразність синтаксичної структури за рахунок використання операторних команд для виконання операцій над матрицями та векторами, що роблять код більш читабельним, коротким та ефективним; швидкість виконання за рахунок використання алгоритмів та бібліотек, оптимізованих під векторизовані обчислення; підтримка лінійної алгебри, в основі якої лежать матрично-векторні обчислення.

Окрім простих операцій над векторами та матрицями, також MATLAB підтримує різні операції агрегування та перетворення матриць, серед яких: *length* – повертає довжину вектора, *size* – функція, що повертає розмірність масиву за заданим виміром, *sum* – підраховує суму елементів масиву чи вектору, *max*, *min* – функції, що шукають максимальний та мінімальний елементи масиву чи вектору, *\** операція поелементного добутку.

Програмні модулі, або програми, у MATLAB представляються у формі функцій. У MATLAB є ряд функцій для числового інтегрування, зокрема: *integral* – обчислення інтегралів одновимірних функцій за допомогою різних методів, таких як квадратура Гауса, квадратура трапецій та інші; *integral2* – обчислення подвійних інтегралів адаптивним алгоритмом; *integral3* – обчислює потрійні інтеграли адаптивним алгоритмом; *quad* – інтегрування методом низького порядку, використовуючи рекурсивне правило Симпсона; *quadl* – використовує адаптивне правило квадратури Гауса-Лобато дуже високого порядку; *quads* – інтегрування за допомогою квадратурних формул Ньютона-Котеса 8-го порядку; *trapz* – обчислення визначеного інтеграла методом трапецій; *dblquad* – обчислює значення подвійного інтеграла для підінтегральної функції, за замовчанням використовуючи квадратурну функцію *quad*; *cumtrapz* – обчислення кумулятивного інтегралу методом трапецій

Як було сказано вище, при застосуванні прямих методів розв'язання інтегральних рівнянь в кожній точці апроксимаційної сітки отримується СЛАР, що потрібно вирішити методами лінійної алгебри. У MATLAB вбудовано ряд функцій, що реалізують різні алгоритми розв'язання СЛАР. Деякі засоби розв'язання СЛАР, що представлені в системі MATLAB у матричному вигляді  $AX=B$ , наведено в таблиці 1. Вибір конкретного методу залежить від вимог до точності та швидкості алгоритмів та програм та від характеру даних в матрицях.

Таблиця 1

## Деякі функції розв'язання СЛАР, вбудовані в MATLAB

Числовий метод	Функція Matlab
Метод оберненої матриці	Розв'язання здійснюється шляхом знаходження оберненої матриці функцією $inv$ та її добутку на $B$ : $x=inv(A)*B$
Методи розв'язання на основі розкладання матриці	$[L, U]=lu(A)$ – виконує LU-розклад, $[Q, R]=qr(A)$ – виконує QR-розклад, $R=chol(A)$ – здійснює розклад Холецького, $[U, S, V]=svd(A)$ – метод сингулярного розкладу (SVD).
Метод спряжених градієнтів	$X=cgs(A, B)$

Для розв'язання СЛАР в MATLAB рекомендовано застосовувати оператор лівого ділення « $\backslash$ » (наступним чином:  $x=A/B$ ) або еквівалентну їй функцію  $mldivide$  та  $linsolve$ , які автоматично обирають найкращий метод розв'язання з врахуванням характеристик матриць. Алгоритми вибору конкретного методу розв'язання, який реалізований в методах  $mldivide$  та  $linsolve$  описано в [5] та [6] відповідно.

**Побудова ефективних модулів MATLAB.** Побудова ефективних користувачьких модулів високої споживачької якості передбачає дотримання наступних критеріїв: дотримання простоти та зручності формування вихідних даних; організація максимальної швидкодії обчислень; представлення результатів обчислень в зручній та зрозумілій формі; обробка виключних ситуацій та індикація процесу обчислень.

Розглянемо побудову модулів MATLAB для розв'язання СІР прямими методами наближеного розв'язання.

Для простоти та зручності формування вихідних даних для функцій розв'язування СІР передбачені різні інтерфейси взаємодії, які передбачають різну кількість аргументів функції. Наприклад, модуль розв'язання СІР методом колокації має такі можливі інтерфейси:

1.  $y=slvie2colloc(x,mm,K,f)$ .
2.  $y=slvie2colloc(x,mm,K,f,y0)$ .
3.  $y=slvie2colloc(x,mm,K,f,y0,B)$ .

Вхідними параметрами програми є:  $x, mm, K, f, y0, B$ , де  $mm$  – степінь полінома, інші параметри описані в таблиці 2.

Якщо кількість аргументів менша за кількість параметрів функції, відсутні аргументи приймають значення за замовчуванням. Основні параметри функцій, відповідні їм позначення та значення за замовчуванням описані в таблиці 2.

Таблиця 2

## Опис та характеристики основних параметрів функцій

Позначення	Формат	Опис	Значення за замовчуванням
1	2	3	4
$x$	вектор-рядок	сітка вузлів, на якій необхідно знайти розв'язки системи	–
$a$	число	ліва границя відрізка інтегрування	–
$b$	число	права границя відрізка інтегрування	–

Закінчення таблиці 2

1	2	3	4
$n$	число	довжина вектора $x$ , кількість точок розбиття	–
$m$	число	розмірність СІР	–
$h$	число	початковий крок інтегрування	0.1
$f$	вектор-рядок	права частина СІР у вигляді вектора анонімних функцій	–
$f$	масив $m \times n$	права частина системи інтегральних рівнянь у табличному вигляді	–
$K$	масив $m \times m$	масив ядер системи інтегральних рівнянь у вигляді анонімних функцій	–
$K$	масив $m \times m \times n \times n$	масив ядер системи інтегральних рівнянь, представлених у табличному вигляді (перші дві розмірності – номер ядра системи, третя та четверта розмірності – номер вузла)	–
$B$	масив $m \times m$	масив значень коефіцієнтів при $y_r, r = 1, m$ . Використовується у випадку СІР Вольтера II роду	діагональна матриця розмірності $m$ з одиничними елементами на головній діагоналі,
$E$	число	значення похибки обчислень	$10^{-3}$
$x0$	вектор-рядок	точки $x_r$ , у яких задані «розгінні» значення шуканих функцій	$f(a)$ для СІР Вольтера II роду
$y0$	двовимірний масив	початкові («розгінні») значення шуканих функцій $y_r(x)$ , що задані в довільній кількості точок $x_i$	вектор розмірності $m$ , елементи якого набувають значення правих частин $f(a)$

У програм розв'язання СІР Вольтера II роду можливі два види інтерфейсу, який залежить від виду вихідної СІР: для систем з постійною матрицею коефіцієнтів при у необхідно ввести масив коефіцієнтів і вектор початкових умов  $y0$ , для систем з одиничною матрицею коефіцієнтів ці два параметри передавати необов'язково.

Модулі, призначені для розв'язання на заданій сітці  $x$  розмірності  $n$  в результаті своєї роботи повертають масив розмірності  $m \times n$  значень функцій  $y_l, l = \overline{1, m}$ .

Передбачено два способи вихідних структур ядер та правих частин СІР: у табличній та аналітичній формі. При табличній формі дані передаються на обробку у вигляді сформованих масивів числових даних. При такому поданні даних потрібно більше пам'яті обчислювальної системи, проте витрати машинного часу будуть меншими порівняно з другим підходом. При аналітичній формі ядер і правих частин передбачається формування спеціальних функцій чи файлів, які містять дані у функціональному вигляді. Такий підхід дозволяє заощаджувати пам'ять обчислювальної системи, проте потребує більше часу для обчислень. Форма вихідних даних може залежати від способу синтезу моделей інтегральних рівнянь.

Розглянемо способи представлення вихідних даних інтегральних моделей в системі MATLAB.

1. В аналітичному вигляді ядра можуть представлятися у формі двовимірного масиву, кожна комірка якого є аналітичним описом відповідних ядер СІР.

2. В табличній формі ядра подаються у формі чотиривимірного числового масиву  $K(i, j, l, m)$ , де  $i, j$  – індекси ядер СІР, а  $l, m$  – індекси точок  $(x(l), x(m))$ ,  $m < l$ .

В модулях MATLAB реалізовано метод опису ядер та правих частин у формі так званих анонімних функцій MATLAB. Анонімна функція двох аргументів  $x$  та  $s$  записується в такому виді:

$$@ (x, s) \varphi(x, s),$$

де  $\varphi(x, s)$  – аналітична функція двох змінних.

Наприклад, праву частину СІР

$$2(1-x)\sin x - \cos x - x + 1$$

можна описати таким чином:

$$@ (x) (2*(1-x).*\sin(x)-\cos(x)-x+1).$$

Для того щоб сформувати вхідні дані в аналітичному представленні, потрібно дотримуватись наступного порядку.

1. Формується функція MATLAB з аргументами, що задають ядра СІР, що повинна мати наступний заголовок:

$$function k=Kern(i, j, x, s)$$

де  $k$  – вихідний масив розмірності  $m*m$ , кожен елемент якого є анонімною функцією від двох змінних  $x$  та  $s$ ; де  $x, s$  – незалежні змінні.

Наприклад, для системи другого порядку ця функція виглядає наступним чином:

$$\begin{aligned} function k=Kern(i, j, x, s) \\ k=\{... \\ @ (x, s) (K(1, 1)), @ (x, s) (K(1, 2)); ... \\ @ (x, s) (K(2, 1)), @ (x, s) (K(2, 2)) ... \\ \}; \end{aligned}$$

де  $K(i, j)$  – аналітична функція, що описує ядро СІР, яке міститься в підінтегральному виразі  $j$ -го інтегралу  $i$ -го рівняння СІР та залежить від змінних  $x$  та  $s$ .

2. Формується функція, що задає праві частини. Вона повинна мати наступний заголовок:

$$function f=F$$

де  $f$  – вихідний масив з розмірністю  $m*1$ , кожен елемент якого є анонімною функцією від змінної  $x$ ;  $x$  – незалежні змінні.

Так, для системи другого порядку відповідна структура виглядатиме так:

$$\begin{aligned} function f=F \\ f=\{... \\ @ (x) (fp(1)); ... \\ @ (x) (fp(2))... \\ \}; \end{aligned}$$

де  $fp(i)$  – аналітична функція, що описує праву частину, яка міститься в  $i$ -му рівнянні СІР та залежить від змінної  $x$ .

3. Завантажуємо сформовані на кроці 1 і кроці 2 функції в пам'ять, надавши значення цих функцій будь-яким змінним.

4. Викликається основна програма, якій передаються створені на кроці 3 змінні, а також інші необхідні вхідні дані.

Функції як результат повертають масив розв'язків  $u$ .

Числові реалізації СІР потребують виконання значної кількості обчислень та можуть вимагати великих масивів даних для збереження результатів обчислень, що вимагає оптимального використання пам'яті та максимально ефективної організації обчислень у модулях MATLAB. Враховуючи це та той факт, що мова програмування MATLAB є інтерпретованою, при розробці застосунків MATLAB рекомендується дотримуватись таких загальних рекомендацій :

1. Мінімізувати використання циклів та максимізувати використання матрично-векторних операцій (матричні операції порівняно з циклами інтерпретатор Matlab виконує в десятки та сотні разів швидше).

2. Видаляти з пам'яті непотрібні вже змінні з допомогою функцій *delete* та *clear*.

3. Виконувати попереднє резервування пам'яті в порядку від більших до менших за розміром. Ініціалізацію масивів даних здійснювати з допомогою вбудованих функцій *ones*, *eye* и *zeros*.

Дотримання цих рекомендацій скорочує час розрахунків і зменшує фрагментацію пам'яті в порівнянні з динамічним виділенням пам'яті під масиви.

Функції як результат повертають масив розв'язків  $u$ , розмірність якого за першим виміром відповідає кількості невідомих функцій СІР. Для кожної функції для візуального аналізу отриманих рішень потрібно побудувати графіки. В MATLAB реалізовано потужну систему візуалізації графіків. Ці команди, зокрема, включають: *figure(i)* – активізує вікно графіку з індексом  $i$ , *hold on/off* – вмикає/вимикає відображення сітки у вікні, *set* – встановлення значень властивостей графічного об'єкта, *xlabel*, *ylabel* – налаштування та відображення координатних осей графічного об'єкта, *plot* – будує двовимірний графік, *legend* – конфігурує легенду графіку, *gca* – посилання на поточні осі або діаграму, *clf* – очистка поточної фігури.

**Верифікація вихідних даних.** У відповідності з вимогами до пакетів прикладних програм комп'ютерного моделювання, всі розроблені функції повинні здійснювати перевірку коректності вихідних даних, їх відповідність задачі, а також забезпечувати коректне завершення роботи програми у разі виникнення помилки. Верифікація даних дозволяє забезпечити прогнозовану поведінку модулів та уникнути непередбачених ситуацій.

В середовищі MATLAB існує цілий ряд вбудованих функцій для перевірки типів даних та їх структур. Їх доцільно використовувати для перевірки аргументів функцій на коректність. Призначення деяких з цих функцій має наступний зміст: *nargin* – повертає кількість аргументів заданої функції; *isvector* – визнає, чи задана змінна є вектором; *isempty* – визнає, чи задана змінна є порожнім масивом; *isnumeric* – перевіряє, чи змінна містить числові дані.

Для викидання помилки та відображення повідомлення помилки в MATLAB призначена команда *error*.

**Приклад реалізації модуля розв'язання інтегрального рівняння в MATLAB.** Розглянемо можливу реалізацію модуля розв'язання системи лінійних інтегральних рівнянь Вольтера II роду з використанням методу квадратур за формулами (2) в системі MATLAB з врахуванням вищезазначених концепцій. Для простоти представлення форму вихідних даних обмежено аналітичним представленням у формі анонімних функцій.



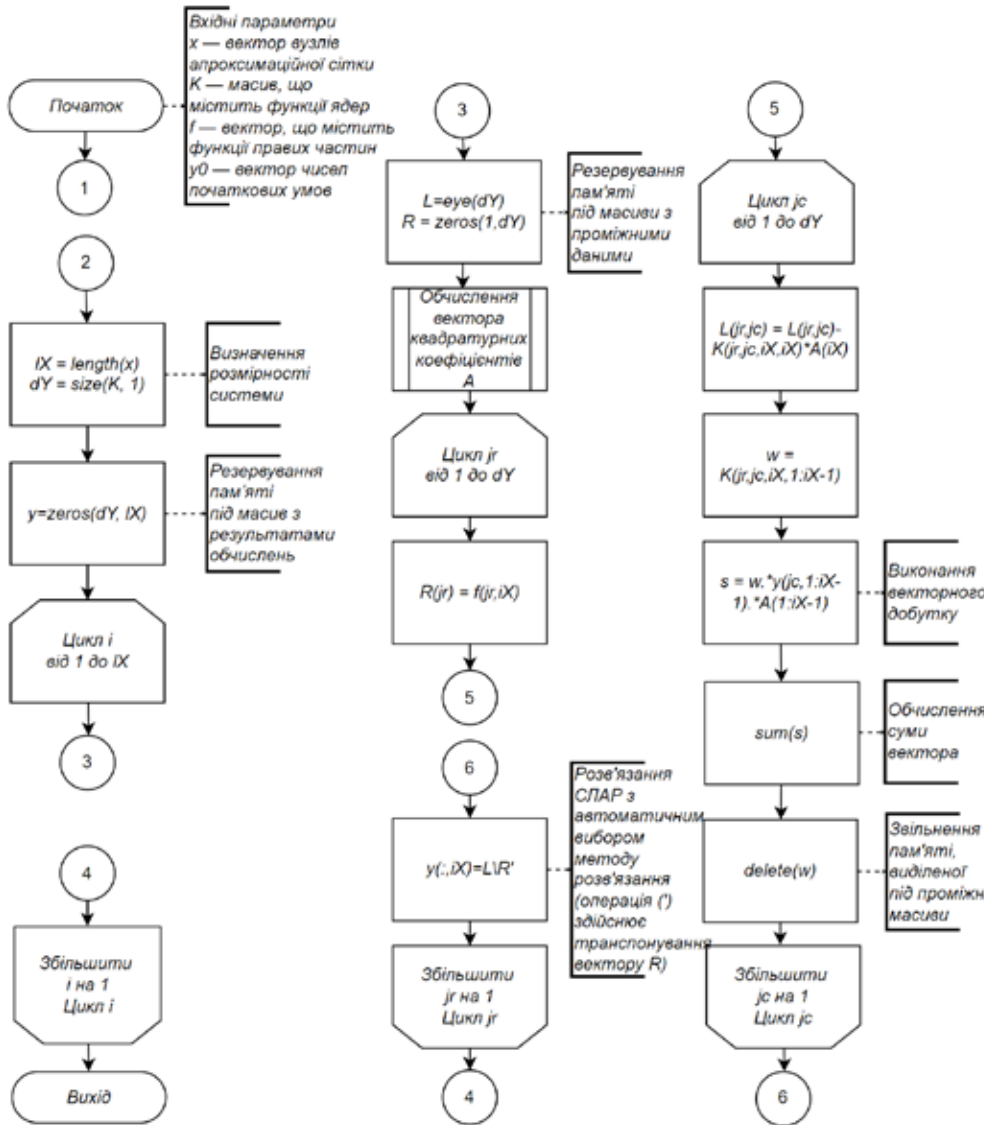


Рис. 1. Алгоритм роботи модуля розв'язання системи лінійних інтегральних рівнянь Вольтера II роду в системі MATLAB

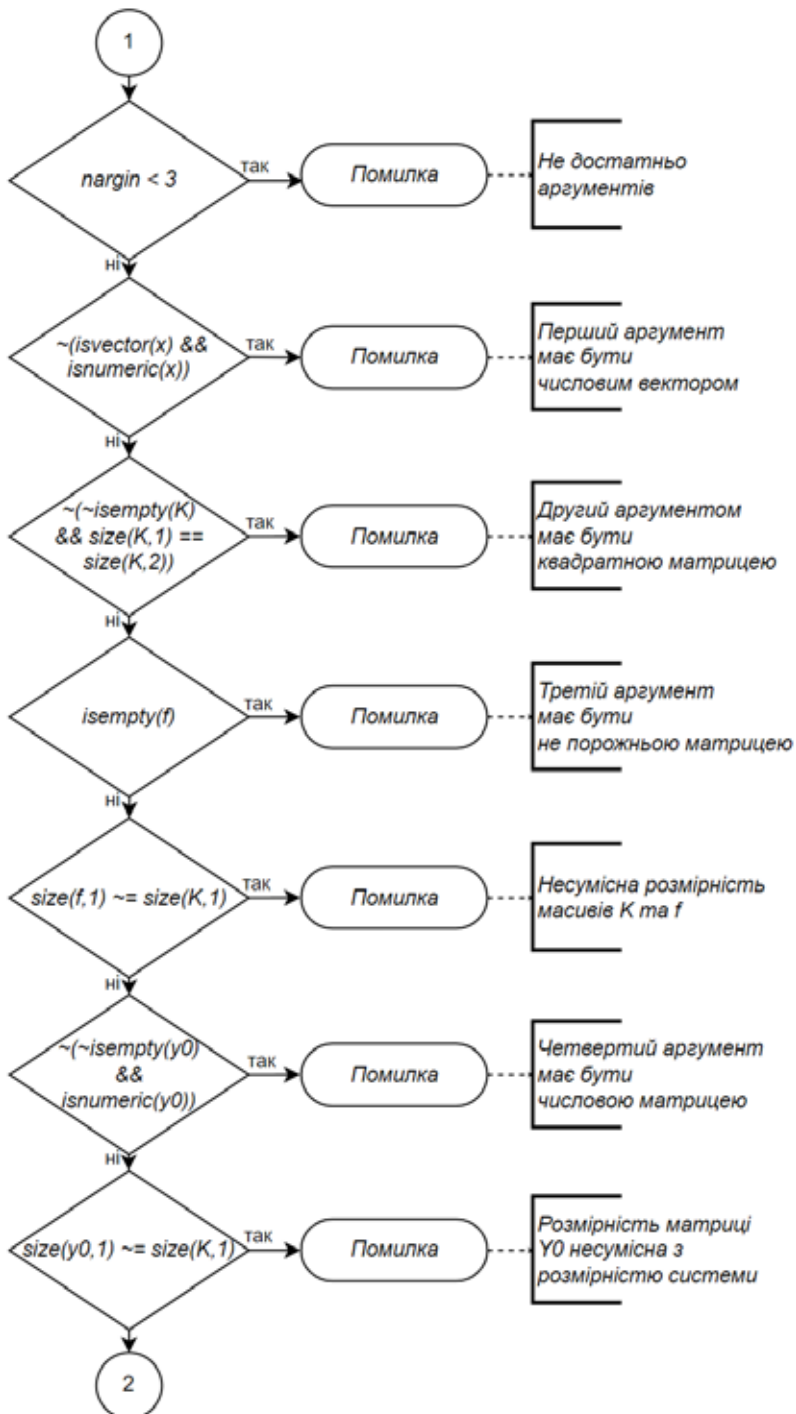


Рис. 2. Алгоритм роботи модуля розв'язання системи лінійних інтегральних рівнянь Вольтера II роду в системі MATLAB (процес перевірки коректності вихідних даних)

**Висновки.** Застосування запропонованих в роботі рішень та рекомендацій до побудови функцій в системі MATLAB дозволяє отримати обчислювальні модулі високої якості за критеріями оптимізації використання пам'яті та швидкодії обчислень, зручності використання та передбачуваності роботи. Ці рішення знайшли застосування при побудові засобів дослідження інтегральних моделей у формі інтегральних рівнянь Вольтера, для розв'язання яких вказані критерії якості є особливо критичними.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Верлань А.Ф., Сизиков В.С. Интегральные уравнения: методы, алгоритмы, программы. Київ : Наукова думка, 1986. 544 с.
2. MATLAB – MathWorks. URL: <https://www.mathworks.com/products/matlab.html> (дата звернення 29.05.2023). Назва з екрана.
3. Верлань А.Ф., Сизиков В.С. Методы решения интегральных уравнений с программами для ЭВМ. Справочное пособие. 1978. 292 с.
4. Горошко И.О., Тихоход В.А. Компьютерная реализация решения систем интегральных уравнений Вольтерры при исследовании многосвязных динамических объектов. *Электронное моделирование*. 2007. Том. 29. № 3. С. 101–107.
5. Тихоход В. О., Кублій Л. І., Онисько А. І. Дослідження багатозв'язних динамічних систем шляхом розв'язання систем інтегральних рівнянь Вольтера I роду методом колокацій. *Інформаційні технології та суспільство*. 2023. Випуск 1 (7).
6. Solve systems of linear equations  $Ax=B$  for  $x$  – MATLAB mldivide. URL: <https://www.mathworks.com/help/matlab/ref/mldivide.html> (дата звернення 29.05.2023). Назва з екрана.
7. Solve linear system of equations – MATLAB linsolve. URL: <https://www.mathworks.com/help/matlab/ref/linsolve.html> (дата звернення 29.05.2023). Назва з екрана.
8. MATLAB Performance – MATLAB & Simulink. URL: <https://www.mathworks.com/products/matlab/performance.html> (дата звернення 29.05.2023). Назва з екрана.

### REFERENCES:

1. Verlan A.F., Sizikov V.S. (1986) *Intehralnye uravneniya: metody, alhoritmy, prohramy [Integral Equations: Methods, Algorithms, Programs]*. Kyiv: Naukova dumka.
2. MATLAB – MathWorks. URL: <https://www.mathworks.com/products/matlab.html> (Viewed 29.05.2023). Title from the screen.
3. Verlan A.F., Sizikov V.S. (1978) *Metody resheniya integralnyh uravnenij s programmami dlya EVM. Spravochnoe posobie [Methods of solving integral equations with computer programs. Reference manual]*.
4. Goroshko I.O., Tikhokhod V.A. (2007) *Kompyuternaya realizaciya resheniya sistem integralnyh uravnenij Volterry pri issledovanii mnogovsyaznyh dinamicheskikh obektov. [Computer implementation of solving systems of Volterra integral equations in the study of multiply connected dynamic objects] Elektronnoe modelirovanie. (Vol. 29., № 3). (pp. 101–107).*
5. Tykhokhod V. O., Kublii L. I., Onysko A. I. (2023) *Doslidzhennia bahatozvi-aznykh dynamichnykh system shliakhom rozv'iazannia sistem intehralnykh rivnian Voltera I rodou metodom kolokatsii. [Study of multi-connected dynamical systems by solving systems of Volterra integral equations of the first kind by the method of collocations]. Information technologies and society. Vol. 1 (7).*
6. Solve systems of linear equations  $Ax=B$  for  $x$  – MATLAB mldivide. URL: <https://www.mathworks.com/help/matlab/ref/mldivide.html> (Viewed 29.05.2023). Title from the screen.

7. Solve linear system of equations – MATLAB linsolve. URL: <https://www.mathworks.com/help/matlab/ref/linsolve.html> (Viewed 29.05.2023). Title from the screen.

8. MATLAB Performance – MATLAB & Simulink. URL: <https://www.mathworks.com/products/matlab/performance.html> (Viewed 29.05.2023). Title from the screen.

---