DC 004.94
DOI: 10.32626/2308-5916.2019-20.40-50

V. A. Ivanyuk, Ph. D., Associate Professor,
V. A. Fedorchuk, Doctor of Technical Sciences, Professor

Kamianets-Podilskyi Ivan Ohiienko National University,
Kamianets-Podilskyi

VECTOR-MATRIX METHOD OF NUMERICAL IMPLEMENTATION OF THE POLYNOMIAL INTEGRAL VOLTERRA OPERATORS

The article deals with the quadrature method for the numerical implementation of polynomial integral operators. With the computer implementation of Volterra-type integral models, the typical problem is the accumulation of calculations at each step of the computational process. For its acceleration it is suggested to apply the vector-matrix approach. The suggested approach is based on quadrature methods: rectangles, trapezoids, and Simpson's. For homogeneous polynomial integral Volterra operators of the first-, second- and third-degree, respectively, the objects in the form of vectors, matrices, and three-dimensional structures containing the coefficients of the corresponding quadrature formulas have been constructed. The suggested vector-matrix approach involves the reduction of computational operations to the elementary multiplication of elements of the corresponding structures and allows efficient use of parallel algorithms, which significantly accelerates the execution of computational tasks for the implementation of integral operators. In the research work the complexity of implementation is estimated depending on the number of possible parallel flows. The estimation of the suggested approximations of integral representations is researched by model examples, in which there are models in the form of second- and third-degree polynomial integrals of Volterra. The results of computational experiments showed that among the considered quadrature methods, the trapezoidal method is optimal in terms of «precision — complexity of implementation». The accuracy of the numerical implementation of integral models depends on the chosen method, the simulation step, the type of kernel, and does not depend on the dimensionality of the operator. The vector-matrix approach allows building of efficient algorithms for the numerical implementation of integral models and greatly simplifies their software implementation, as it allows easy scaling to a multidimensional case. Such representation allows to use advantages of matrix-oriented packages of applications (Matlab, Octave, Scilab), the peculiarity of which is the high speed of execution of matrix operations.

Key words: *polynomial Volterra integral operators, quadrature method, vector-matrix method.*

Introduction. The constant extension of the scope of integrated models stimulates the development of methods and means of their numerical implementation [1, 3–8]. Moreover, depending on the objects studied and the tasks, the basic ones for modeling linear dynamic systems based on integral models are Volterra operators, and for nonlinear dynamic systems, polynomial Voltaire operators [4, 7, 8].

One of the effective methods of numerical implementation of integral models is the method of quadrature formulas, which includes replacing of the integral by an approximating system of algebraic representations with respect to the discrete values of the desired function [2, 6, 9]. At the same time, the upper bound of the integration is fixed in the Volterra models, and formulas are used for approximate calculation of the integral. There are many quadrature formulas. They include the formulas of Newton-Cotes (including rectangles, trapezoids, Simpson), Gauss, Chebyshev, and others. This gives rise to many approaches and methods for applying the quadrature method. Cubature formulas [2] are applied to approximate polynomial integral models, but there is no general approach and definite recommendations for their application, depending on the form of the integral model.

The key problem of applying the algorithms of the quadrature method in the numerical implementation of Volterra integral models with an arbitrary kernel is the accumulation of the number of calculations at each step of the computational process [2]. This especially becomes apparent in the numerical implementation of polynomial integral operators. Therefore, developing new approaches to the numerical implementation of Volterra polynomial integral models that will accelerate the execution of necessary operations at each step of the numerical implementation is an urgent task.

Algorithms for numerical implementation of the integral polynomial Volterra operator. Consider the possibility of using quadrature formulas of rectangles, trapezoids and Simpson for the numerical realization of a nonhomogeneous Volterra integral polynomial operator of the third degree

$$y(t) = \int_0^t K_1(s)\,x(t-s)\,ds + \int_0^t\int_0^t K_2(s_1,s_2)\,x(t-s_1)\,x(t-s_2)\,ds_1 ds_2 + \int_0^t\int_0^t\int_0^t K_3(s_1,s_2,s_2)\,x(t-s_1)\,x(t-s_2)\,x(t-s_3)\,ds_1 ds_2 ds_3. \quad (1)$$

By entering a sustainable breaking down and replacing the integrals (1) by quadratic sums, we obtain [2, 9]:

$$y(t_i) = \sum_{j=1}^{i} A_{1,j} K_1(t_j) x(t_i - t_j) +$$

$$+ \sum_{j=1}^{i} \sum_{g=1}^{i} A_{2,jg} K_2(t_j, t_g) x(t_i - t_j) x(t_i - t_g) + \qquad (2)$$

$$+ \sum_{j=1}^{i} \sum_{g=1}^{i} \sum_{l=1}^{i} A_{3,jgl} K_3(t_j, t_g, t_l) x(t_i - t_j) x(t_i - t_g) x(t_i - t_l), i = \overline{1,n}.$$

In this formulation, the problem of numerical implementation arises because of the complexity of describing multidimensional approximation representations of integral operators and the considerable number of computational actions. It is suggested to apply a vector-matrix approach with the reduction of all operations to elementwise multiplication.

Consider separately homogeneous operators of the first, second and third degrees. The coefficients A_1, A_2, A_3 are represented in vector-matrix form. In the case of a one-dimensional operator of the coefficient A_1 is determined by a vector that has a different view, depending on the method used:

- rectangles:

$$A_1^{\text{п}} = h\begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \end{pmatrix};$$

- trapeze:

$$A_1^{\text{т}} = h\begin{pmatrix} \frac{1}{2} & 1 & 1 & \dots & 1 & 1 & \frac{1}{2} \end{pmatrix};$$

- Simpson:

$$A_1^{\text{с}} = h\begin{pmatrix} \frac{1}{3} & \frac{4}{3} & \frac{2}{3} & \dots & \frac{2}{3} & \frac{4}{3} & \frac{1}{3} \end{pmatrix}.$$

When approximating a two-dimensional operator, we obtain matrices that determine the coefficient A_2:

- for the method of rectangles:

$$A_2^{\text{п}} = h^2 \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix};$$

- for the method of trapeze:

$$A_2^{\text{т}} = h^2 \begin{pmatrix} 1/4 & 1/2 & 1/2 & \dots & 1/2 & 1/2 & 1/4 \\ 1/2 & 1 & 1 & \dots & 1 & 1 & 1/2 \\ 1/2 & 1 & 1 & \dots & 1 & 1 & 1/2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1/2 & 1 & 1 & \dots & 1 & 1 & 1/2 \\ 1/2 & 1 & 1 & \dots & 1 & 1 & 1/2 \\ 1/4 & 1/2 & 1/2 & \dots & 1/2 & 1/2 & 1/4 \end{pmatrix};$$

- for the Simpson method:

$$A_2^{\text{с}} = h^2 \begin{pmatrix} 1/9 & 4/9 & 2/9 & \dots & 2/9 & 4/9 & 1/9 \\ 4/9 & 16/9 & 8/9 & \dots & 8/9 & 16/9 & 4/9 \\ 2/9 & 8/9 & 4/9 & \dots & 4/9 & 8/9 & 2/9 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 2/9 & 8/9 & 4/9 & \dots & 4/9 & 8/9 & 2/9 \\ 4/9 & 16/9 & 8/9 & \dots & 8/9 & 16/9 & 4/9 \\ 1/9 & 4/9 & 2/9 & \dots & 2/9 & 4/9 & 1/9 \end{pmatrix}.$$

Replacing the triple integral with (1), we obtain a three-dimensional structure containing the coefficients of quadrature formulas. It can be represented as a cube (Fig. 1). This structure can be determine as following:

$$A_3 = \left\{ A_3^{\text{м},1};\ \ A_3^{\text{м},2};\ \ A_3^{\text{м},3};\ \ \dots\ \ A_3^{\text{м},n-2};\ \ A_3^{\text{м},n-1};\ \ A_3^{\text{м},n} \right\}, \qquad (3)$$

where $A_3^{\text{м},1}$; $A_3^{\text{м},2}$; $A_3^{\text{м},3}$; ... $A_3^{\text{м},n-2}$; $A_3^{\text{м},n-1}$; $A_3^{\text{м},n}$ — matrices of coefficient, which determined by the basic quadrature methods applied to each dimension $A_{3,j}$, $A_{3,g}$, $A_{3,l}$.

Using the rectangles method, we get the following matrices:

$$A_3^{\text{п},1} = A_3^{\text{п},2} = \dots = A_3^{\text{п},n-2} = A_3^{\text{п},n-1} = \\ = h^3 \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix};$$

$$A_3^{\text{п},n} = h^3 \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}.$$

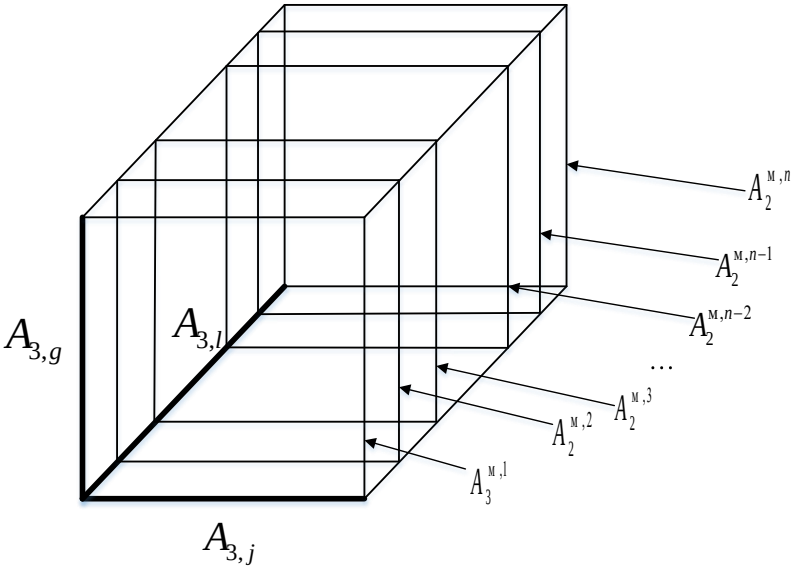

Fig. 1. *Graphic representation of the structure (3)*

Applying of the trapezoid method results in matrices:

$$A_3^{\text{т},1} = A_3^{\text{т},n} = = h^3 \begin{pmatrix} 1/8 & 1/4 & 1/4 & \dots & 1/4 & 1/4 & 1/8 \\ 1/4 & 1/2 & 1/2 & \dots & 1/2 & 1/2 & 1/4 \\ 1/4 & 1/2 & 1/2 & \dots & 1/2 & 1/2 & 1/4 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1/4 & 1/2 & 1/2 & \dots & 1/2 & 1/2 & 1/4 \\ 1/4 & 1/2 & 1/2 & \dots & 1/2 & 1/2 & 1/4 \\ 1/8 & 1/4 & 1/4 & \dots & 1/4 & 1/4 & 1/8 \end{pmatrix};$$

$$A_3^{\tau,2} = A_3^{\tau,3} = \ldots = A_3^{\tau,n-2} = A_3^{\tau,n-1} = h^3 \begin{pmatrix} 1/4 & 1/2 & 1/2 & \ldots & 1/2 & 1/2 & 1/4 \\ 1/2 & 1 & 1 & \ldots & 1 & 1 & 1/2 \\ 1/2 & 1 & 1 & \ldots & 1 & 1 & 1/2 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1/2 & 1 & 1 & \ldots & 1 & 1 & 1/2 \\ 1/2 & 1 & 1 & \ldots & 1 & 1 & 1/2 \\ 1/4 & 1/2 & 1/2 & \ldots & 1/2 & 1/2 & 1/4 \end{pmatrix}.$$

Applying the Simpson method, we have:

$$A_3^{c,1} = A_3^{c,2m+1} = h^3 \begin{pmatrix} 1/27 & 4/27 & 2/27 & \ldots & 2/27 & 4/27 & 1/27 \\ 4/27 & 16/27 & 8/27 & \ldots & 8/27 & 16/27 & 4/27 \\ 2/27 & 8/27 & 4/27 & \ldots & 4/27 & 8/27 & 2/27 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 2/27 & 8/27 & 4/27 & \ldots & 4/27 & 8/27 & 2/27 \\ 4/27 & 16/27 & 8/27 & \ldots & 8/27 & 16/27 & 4/27 \\ 1/27 & 4/27 & 2/27 & \ldots & 2/27 & 4/27 & 1/27 \end{pmatrix};$$

$$A_3^{c,2} = A_3^{c,4} = \ldots = A_3^{c,2m} =$$

$$= h^3 \begin{pmatrix} 4/27 & 16/27 & 8/27 & \ldots & 8/27 & 16/27 & 4/27 \\ 16/27 & 64/27 & 32/27 & \ldots & 32/27 & 64/27 & 16/27 \\ 8/27 & 32/27 & 16/27 & \ldots & 16/27 & 32/27 & 8/27 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 8/27 & 32/27 & 16/27 & \ldots & 16/27 & 32/27 & 8/27 \\ 16/27 & 64/27 & 32/27 & \ldots & 32/27 & 64/27 & 16/27 \\ 4/27 & 16/27 & 8/27 & \ldots & 8/27 & 16/27 & 4/27 \end{pmatrix};$$

$$A_3^{c,3} = A_3^{c,5} = \ldots = A_3^{c,2m-1} =$$

$$= h^3 \begin{pmatrix} 2/27 & 8/27 & 4/27 & \ldots & 4/27 & 8/27 & 2/27 \\ 8/27 & 32/27 & 16/27 & \ldots & 8/27 & 32/27 & 8/27 \\ 4/27 & 8/27 & 8/27 & \ldots & 8/27 & 8/27 & 4/27 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 4/27 & 8/27 & 8/27 & \ldots & 8/27 & 8/27 & 4/27 \\ 8/27 & 32/27 & 8/27 & \ldots & 8/27 & 32/27 & 8/27 \\ 2/27 & 8/27 & 4/27 & \ldots & 4/27 & 8/27 & 2/27 \end{pmatrix}.$$

Fig. 2 shows the structural representation of the specified operations for different homogeneous operators and their programmatic analogy in the Matlab environment.

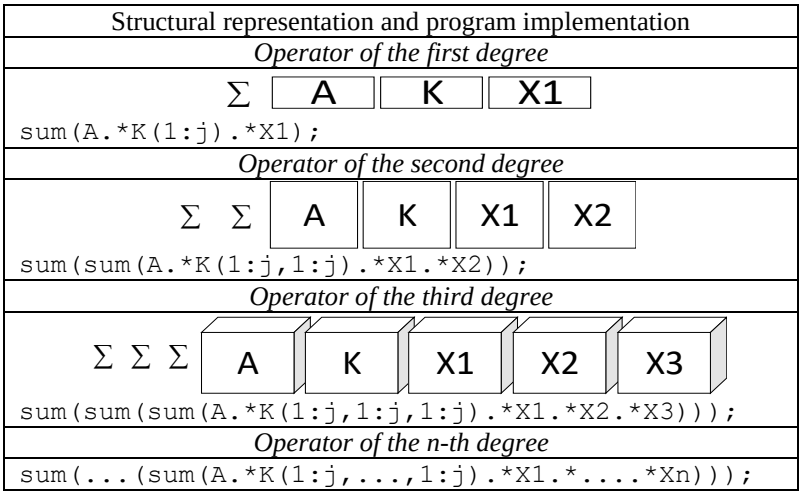

Fig. 2. *Structural representation and program implementation of the vector-matrix approach*

When implementing a first-degree operator: A — vector that defines the coefficients of the quadrature formula in accordance with the shown above; K — vector of kernel values according to the entered time variable partition; X1 — vector of values of input influence. The programmatic analogy of such representation is: `sum(A.*K(1:j).*X1)`. When the implementation of operator of the second degree: A and K are matrices, X1 is a matrix consisting of identical rows of vector x; X2 is a matrix consisting of identical columns of vector x. Elementwise operations are used in software implementation. The general expression for calculating an operator is `sum(sum(A.*K(1:j,1:j).*X1.*X2))`. Similarly, when implementing an operator of the third degree, everything is reduced to element-by-element operations, but already of three-dimensional structures.

This approach greatly simplifies the software implementation of polynomial operators because it allows easy scaling to a multidimensional case, as shown in fig. 2.

Model experiments. The estimation of the suggested approximations of integral representations was investigated on model examples.

Consider the model in the form:

$$y(t)=\int_0^t\int_0^t \frac{1}{2}e^{-\frac{1}{2}(s_1+s_2)}\sin\left(\frac{-\sqrt{2}}{2}s_1\right)\sin\left(\frac{-\sqrt{2}}{2}s_2\right)x(t-s_1)\,x(t-s_2)\,ds_1 ds_2. \quad (4)$$

Figure 3 shows graphs of the transient characteristic of an object, described by model (4), using the method of rectangles, trapezoids, Simpson, and exact transient characteristics. Figure 4 shows the calculation errors.

To analyze the numerical implementation of polynomial homogeneous Volterra operator of the third degree, consider the model

$$y(t)=\int_0^t\int_0^t\int_0^t \frac{\sqrt{2}}{4}e^{-\frac{1}{2}(s_1+s_2+s_3)}\sin\left(\frac{-\sqrt{2}}{2}s_1\right)\sin\left(\frac{-\sqrt{2}}{2}s_2\right)\sin\left(\frac{-\sqrt{2}}{2}s_3\right)\times \\ \times x(t-s_1)\,x(t-s_2)\,x(t-s_3)\,ds_1ds_2ds_3. \qquad (5)$$

The transient characteristic obtained by the considered methods and its exact value are presented in Fig. 5, calculation errors — in Fig. 6.

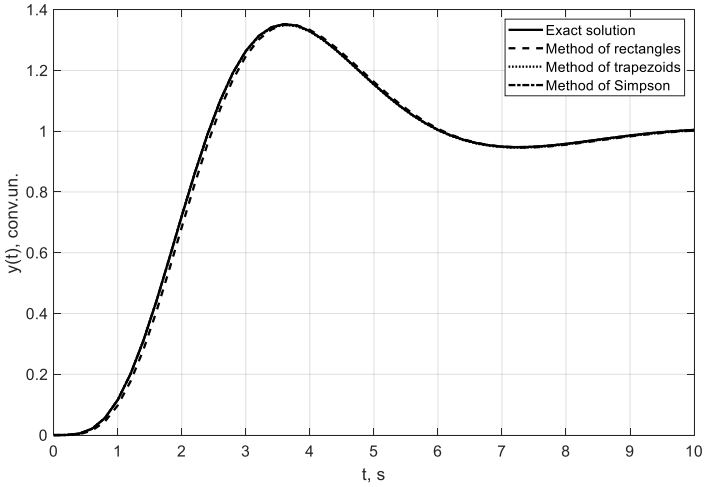

Fig. 3. *Graphs of the transient characteristic (4) obtained by different methods*

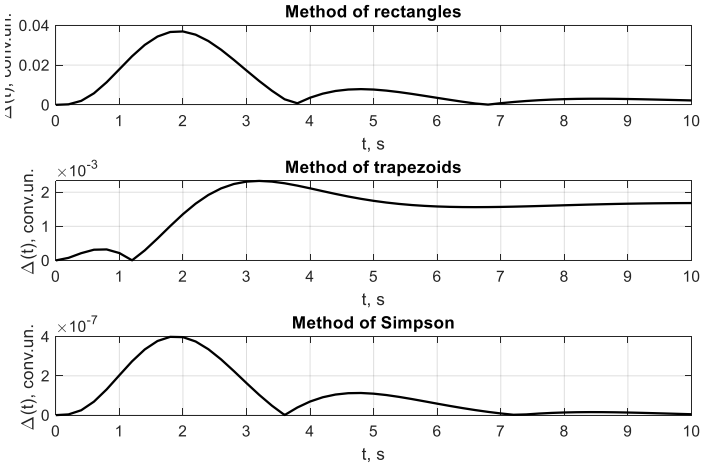

Fig. 4. *Absolute errors of the calculation of transient characteristic for the model (4)*

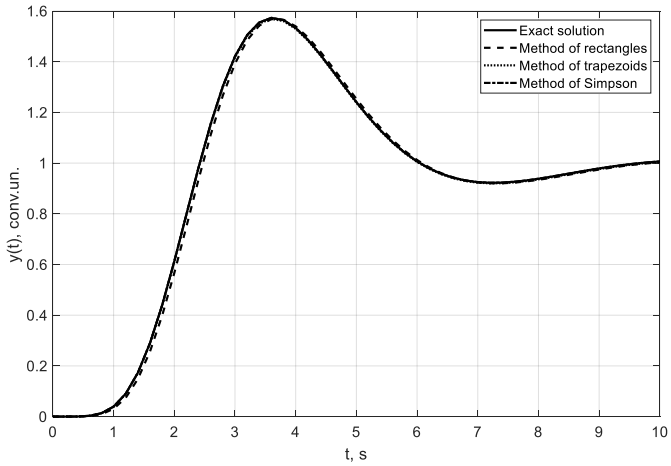

Fig. 5. *Graphs of the transient characteristics (5) obtained by different methods*

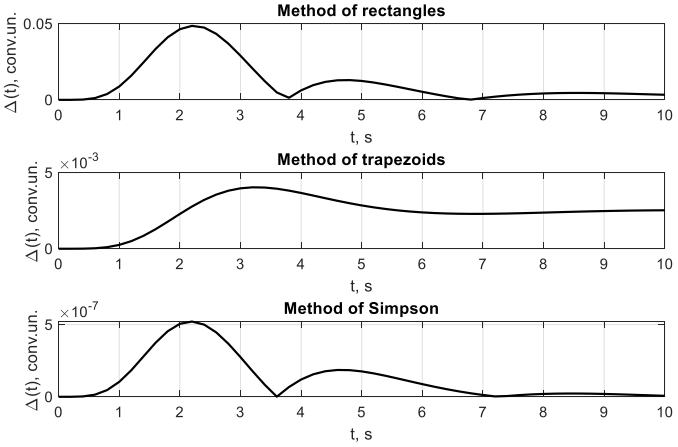

Fig. 6. *Absolute errors of the calculation of transient characteristic for a model (5)*

The obtained results of the computational experiments showed the effectiveness of the suggested methods, with the Simpsons method being the most accurate among the considered methods but applying of this method requires more computations that are associated with additional partitioning to find intermediate values at interpolation points. Optimal in terms of «accuracy — complexity of implementation» is the method of trapezoids, which allows obtaining solutions with a relative error of less than 1%, which is enough for engineering calculations.

It is important to note that the accuracy of implementation of integral models depends on the method chosen, the simulation step, and the type of

kernel and does not depend on the dimensionality of the operator. Therefore, the choice of the «best» method should be based, first, on the analysis of the kernel of the integral inhomogeneous polynomial Volterra operator.

Depending on the type of kernel, computational algorithms can be developed based on the use of different quadrature methods that are applied separately to each dimension. Moreover, not only the considered methods (of rectangles, trapezoids, Simpsons), but also methods based on the combination of Newton-Cotes quadrature formulas of higher order can be applied. Such approach will allow obtaining different cubature formulas and will expand the set of algorithms for approximation of integral models with finite sums and will allow choosing the best method depending on the set initial problem.

The suggested approach makes it possible to accomplish natural parallelism of computational algorithms, which greatly accelerates the numerical implementation of integral operators. Table 1 shows the complexity of the numerical implementation of polynomial integral operators depending on the number of possible parallel threads.

Table 1

The degree of complexity of the numerical implementation of polynomial integral operators

	Operator of the first degree	Operator of the second degree	Operator of the third degree
Usual approach	$O(n)$	$O^2(n)$	$O^3(n)$
Vector-matrix	$\frac{O(n)}{kp}$	$\frac{O^2(n)}{kp}$	$\frac{O^3(n)}{kp}$
n — number of partition points, kp — number of parallel threads			

Conclusions. Thus, the representation of quadrature and cubature formulas in vector-matrix form allows developing of effective algorithms and software means for numerical implementation of integral models. This representation allows taking advantages of matrix-oriented application packages (Matlab, Octave, Scilab), the peculiarity of which is a high speed of matrix operations.

References:

1. Apartsin A. S. On the mathematical modelling of nonlinear dynamical systems by Volterra series / A. S. Apartsin, S. V. Solodusha // Electronic modelling. — 1999. — № 2. — P. 3–12.
2. Verlan A. F. Integral equations: methods, algorithms, software / A. F. Verlan, V. S. Sizikov. — Kyiv, 1986. — 544 p.
3. Verlan A. F. Integral equation toolbox — software package for solving integral equations in the environment Matlab / A. F. Verlan, D. E. Contreras, B. C. Sizikov, S. T. Tykhonchuk, V. A. Fedorchuk. — Kyiv, 1997. — 44 p.
4. Verlan A. F., Fedorchuk V.A. Models of dynamics of electromechanical systems / A. F. Verlan, V.A. Fedorchuk. — Kyiv, 2013. — 221 p.

5. Ivaniuk V. A. Mathematical packages of applications : a tutorial / V. A. Ivaniuk. — Kamianets-Podilskyi, 2015. — 160 p.
6. Manzhirov A. V. Integral Equation Reference: Solution Methods / A. V. Manzhirov, A. D. Polianin. — Moscow, 2000. — 685 p.
7. Sidorov D. N. Methods of analysis of integral dynamic models. Theory and Applications / D. N. Sidorov. — Irkutsk, 2013. — 293 p.
8. Sytnyk O. O. Integral macro models of dynamic objects / O.O. Sytnyk, S. O. Protasov, V. A. Fedorchuk // Mat. and computer modelling. Tech. science. — 2013. — Issue 8. — P. 98-109.
9. Fedorchuk V. A. Integral equations in mathematical modelling problems : a tutorial / V. A. Fedorchuk, V. A. Ivaniuk, D. A. Verlan. — Kamianets-Podilskyi, 2014. — 144 p.

ВЕКТОРНО-МАТРИЧНИЙ МЕТОД ЧИСЛОВОЇ РЕАЛІЗАЦІЇ ПОЛІНОМІАЛЬНИХ ІНТЕГРАЛЬНИХ ОПЕРАТОРІВ ВОЛЬТЕРРИ

У статті розглядається метод квадратур для числової реалізації поліноміальних інтегральних операторів. При комп'ютерній реалізації інтегральних моделей типу Вольтерри характерною проблемою є накопичення кількості обчислень на кожному кроці обчислювального процесу. Для його пришвидшення пропонується застосовувати векторно-матричний підхід. В основі запропонованого підходу лежать методи квадратур: прямокутників, трапецій, Сімпсона. Для однорідних поліноміальних інтегральних операторів Вольтерри першого, другого та третього степеня побудовано, відповідно, у вигляді векторів, матриць та тривимірних структур об'єкти, які містять коефіцієнти відповідних квадратурних формул. Запропонований векторно-матричний підхід передбачає зведення обчислювальних операцій до поелементного множення елементів відповідних структур та дозволяє ефективно використовувати паралельні алгоритми, що значно пришвидшує виконання обчислювальних задач реалізації інтегральних операторів. В роботі оцінено складність реалізації в залежності від кількості можливих паралельних потоків. Оцінку запропонованих апроксимацій інтегральних представлень досліджено на модельних прикладах, в яких присутні моделі у вигляді поліноміальних інтегральних операторів Вольтерри другого та третього степеня. Результати обчислювальних експериментів показали, що серед розглянутих квадратурних методів оптимальним у відношенні «точність — складність реалізації» є метод трапецій. Точність числової реалізації інтегральних моделей залежить від вибраного методу, кроку моделювання, виду ядра, і не залежить від розмірності оператора. Векторно-матричний підхід дозволяє будувати ефективні алгоритми для числової реалізації інтегральних моделей та значно спрощує їх програмну реалізацію, оскільки дозволяє легке масштабування до багатовимірного випадку. Таке представлення дає змогу використовувати переваги матрично-орієнтованих пакетів прикладних програм (Matlab, Octave, Scilab), особливістю яких є висока швидкість виконання матричних операцій.

Ключові слова: *поліноміальні інтегральні оператори Вольтерри, метод квадратур, векторно-матричний метод.*

Отримано: 14.08.2019