

УДК 519.6

Л. В. Мосенцова, аспірантИнститут проблем моделирования в энергетике
им. Г.Е. Пухова НАН Украины, г. Киев

ЧИСЛЕННАЯ РЕАЛИЗАЦИЯ В СИСТЕМЕ MATLAB КВАДРАТУРНЫХ АЛГОРИТМОВ РЕШЕНИЯ ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ ВОЛЬТЕРРА II РОДА В ФОРМЕ ГАММЕРШТЕЙНА

Работа посвящена реализации в вычислительной среде Matlab квадратурных алгоритмов решения нелинейных интегральных уравнений Вольтера II рода в форме Гаммерштейна.

Ключевые слова: *Matlab, интегральное уравнение Вольтера II рода в форме Гаммерштейна, метод квадратур, формула трапеции.*

Введение. В наши дни компьютерная математика интенсивно развивается как передовое научное направление на стыке математики и информатики и предлагает целый набор интегрированных программных систем и пакетов программ для автоматизации математических расчетов. Одной из самых эффективных систем компьютерного программного обеспечения является пакет универсальных интегрированных программ MATLAB [1]. Модифицируемость системы привела к созданию дополнительных пакетов программ, так называемых тулбоксов, направленных на решение задач специфических задач науки и техники. Однако, несмотря на развитие в разных направлениях вычислений, при этом мало внимания уделяется реализации аппарата интегральных уравнений, в частности нелинейных. В связи с этим в данной работе предлагается реализация в системе Matlab квадратурных алгоритмов решения нелинейных интегральных уравнений Вольтера II рода в форме Гаммерштейна путем создания дополнительных модулей.

Метод квадратур для нелинейных интегральных уравнений Вольтера II рода. Рассмотрим нелинейное уравнение Вольтера II рода в форме Урысона [2].

$$y(x) - \int_a^x K(x, t, y(t)) dt = f(x) \quad (1)$$

на отрезке $a \leq x \leq b$. Будем считать, что $K(x, t, y(t))$ и $f(x)$ — непрерывные функции.

На основании уравнения (1) найдем $y(a) = f(a)$. Возьмем постоянный шаг интегрирования h и рассмотрим дискретное множество точек $x_i = a + h(i-1)$, где $i = 1, 2, \dots, n$. В точках $x = x_i$ уравнение (1) принимает вид

$$y(x_i) - \int_a^{x_i} K(x_i, t, y(t)) dt = f(x_i). \quad (2)$$

Выбирая x_i в качестве узлов квадратурной формулы трапеции [3], для которой ошибкой аппроксимации будем пренебрегать, и заменяя ею интеграл в выражении (2), получим следующую систему нелинейных алгебраических (или трансцендентных) уравнений:

$$\begin{aligned} y_1 &= f_1, \\ y_i &= \sum_{j=1}^i A_{ij} K_{ij}(y_j) = f_i, \quad i = 2, 3, \dots, n \end{aligned} \quad (3)$$

где A_{ij} — коэффициенты квадратурной формулы на отрезке $[a, x_i]$, y_i — приближенные значения решения $y(x)$ в узлах x_i , f_i , $K_{ij}(y_j) = K(x_i, t_j, y_j)$.

Соотношения (3) можно записать в виде последовательности рекуррентных нелинейных уравнений

$$\begin{aligned} y_1 &= f_1, \\ y_i - A_{ii} K_{ii}(y_i) &= f_i + \sum_{j=1}^{i-1} A_{ij} K_{ij}(y_j), \quad i = 2, 3, \dots, n \end{aligned} \quad (4)$$

для определения приближенного значения искомого решения в узловых точках.

Применительно к уравнению Вольтера II рода в форме Гаммерштейна [2]

$$y(x) - \int_a^x Q(x, t) \Phi(t, y(t)) dt = f(x) \quad (5)$$

основные соотношения метода квадратур имеют вид ($x_i = a$)

$$\begin{aligned} y_1 &= f_1, \\ y_i - \sum_{j=1}^i A_{ij} Q_{ij} \Phi_j(y_j) &= f_i, \quad i = 2, \dots, n, \end{aligned} \quad (6)$$

где $Q_{ij} = Q(x_i, t_j)$ и $\Phi_j(y_j) = \Phi(t_j, y_j)$. Они приводят к последовательности нелинейных рекуррентных уравнений

$$\begin{aligned} y_1 &= f_1, \\ y_i - A_{ii} Q_{ii} \Phi(y_i) &= f_i + \sum_{j=1}^{i-1} A_{ij} Q_{ij} \Phi_j(y_j), \quad i = 2, \dots, n, \end{aligned} \quad (7)$$

решения которых дают приближенные значения искомой функции.

Описание программы. Алгоритм решения задачи реализован в программе Volt_nonLinear_II_G, составленной в среде моделирования

Matlab и реализован на ЭВМ типа IBM PC. Исходные данные для решения задачи:

N — число интервалов, на которые разбивается отрезок интегрирования; A — нижняя граница интегрирования; B — верхняя граница интегрирования; F — массив размерности $N+1$, содержащий значения правых частей f_i ; $Kern$ — массив *размерности* $N+1$, содержащий значения ядер.

В программе предусмотрены следующие интерфейсы:

1. $[y]= Volt_nonLinear_II_G(a, b, N, K, f)$,
2. $[y]= Volt_nonLinear_II_G(a, b, N, K, f, formula)$.

Параметр *formula* — параметр, указывающий, какая квадратурная формула используется при решении. Перед проведением вычислений программа проверяет корректность введенных данных и в случае неправильности данных возвращает код ошибки, на чем работа программы прекращается. В результате работы программы выдается матрица Y размерностью $N+1$, в которой найдены значения y_i в точках x_1, x_2, \dots, x_{n+1} .

Решение модельных примеров

Пример 1. Задано интегральное уравнение

$$u(x) = 1 - x + \int_0^x [xe^{s(x-2s)} + e^{-2s^2}] u^2(s) ds, \quad (8)$$

имеющее точное решение

$$u(x) = e^{x^2}. \quad (9)$$

Решение уравнения будем искать на отрезке $[0, 0,3]$, который равномерно разбивается на N частей точками $x_i = 0 + h(i - 1)$. На рис. 1 приведены результаты вычислений при $N = 12$.

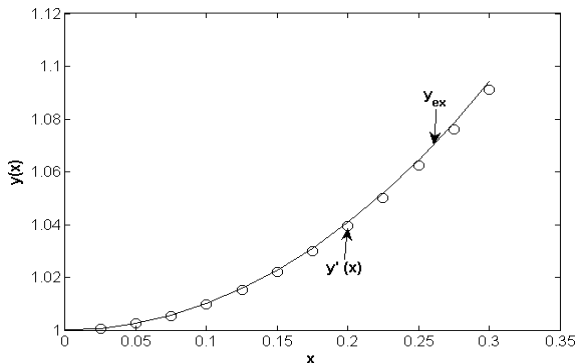


Рис. 1. Графики решения уравнения (8) методом квадратур и точного решения (9) на отрезке $[0, 0,3]$ при $N = 12$.

Графики абсолютной и относительной ошибки приведены на рис. 2 и рис.3. Относительная ошибка численного решения не превышает 0.003, или 0,3%.

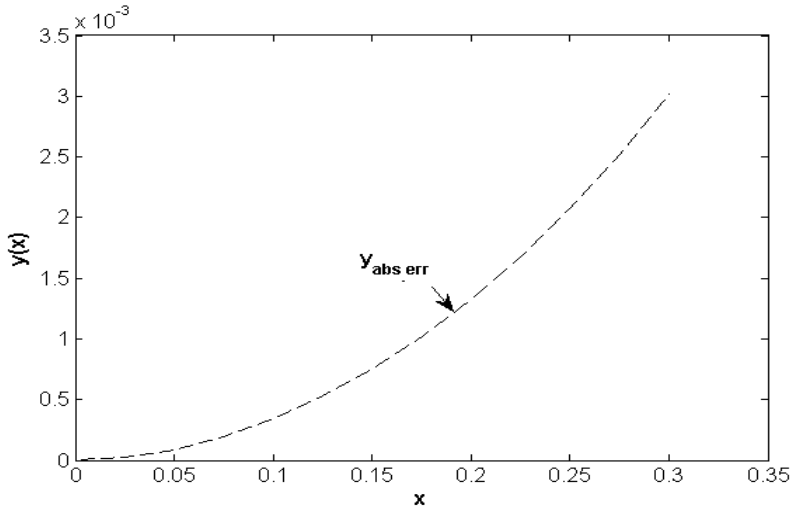


Рис. 2. График абсолютной ошибки решения уравнения (8).

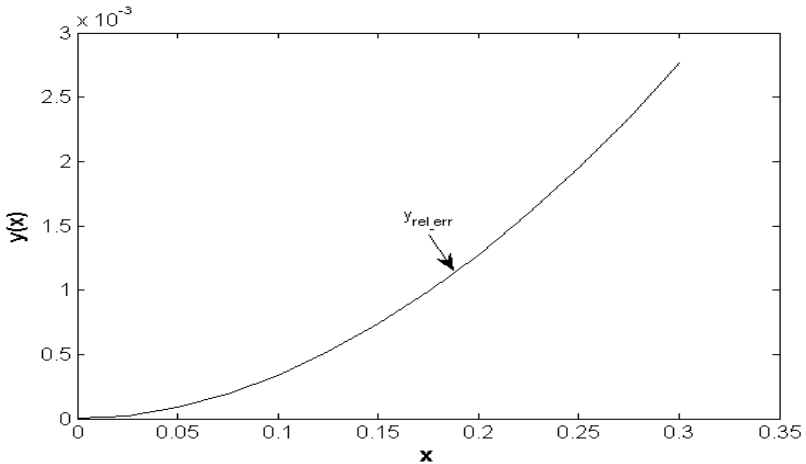


Рис. 3. График относительной ошибки решения уравнения (8).

Пример 2. Задано интегральное уравнение

$$y_0(x) = \cos(x) - \sin(2x) - \sin(x) + 3 \int_0^x \cos(x-s) y_0^2(s) ds. \quad (10)$$

Решение уравнения (10) ищется на отрезке $[0,025, 0,200]$ методом квадратур с использованием формулы трапеций [2]. При этом отрезок интегрирования разбивается равномерно точками на N частей. На рис. 4 представлены графики решения уравнения (10) при $N = 8$ и точного решения.

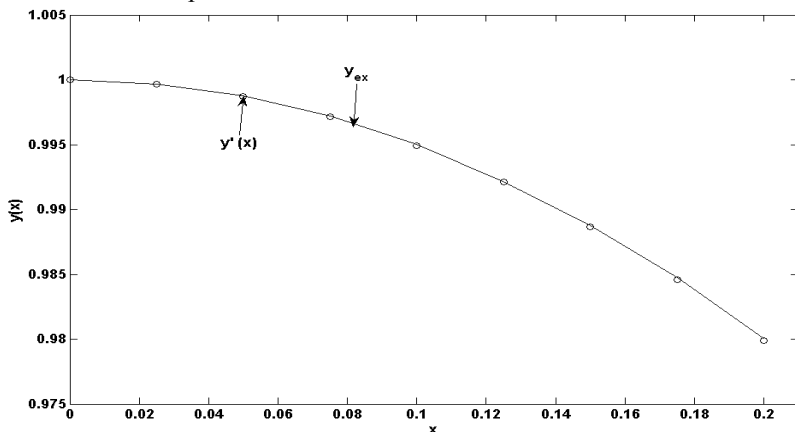


Рис. 4. График решения уравнения (10) $y'(x)$ и точное решение y_{ex} .

Графики абсолютной и относительной ошибки приведены на рис. 5 и рис.6. Относительная ошибка метода не превышает 0.00018, или 0,018%.

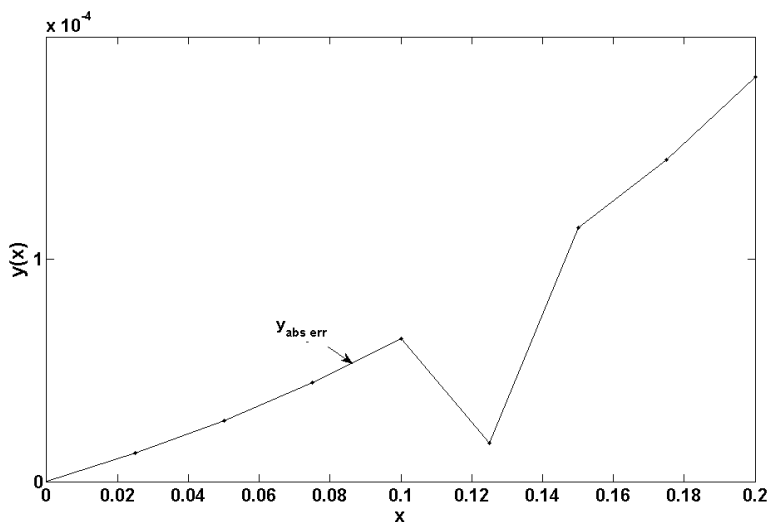


Рис. 5. График абсолютной ошибки решения уравнения (10).

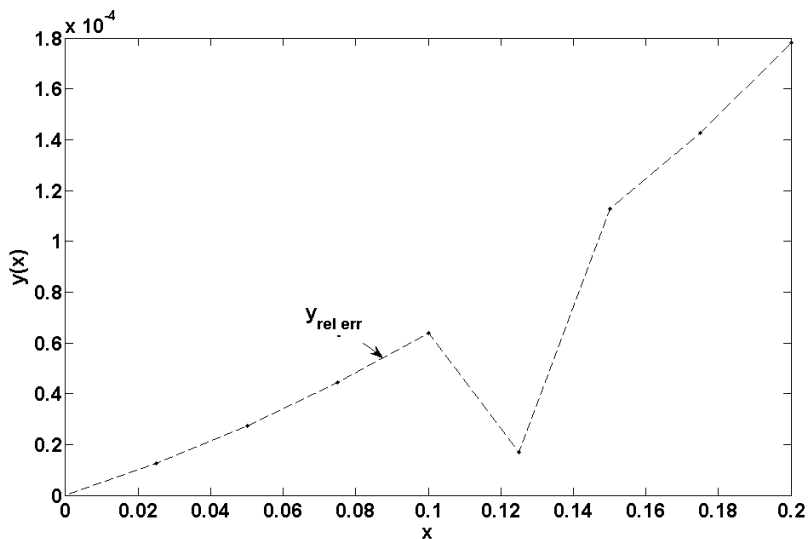


Рис. 6. Графік відносної помилки рішення рівняння (10).

Заключення. В роботі пропонується реалізація в середовищі Matlab квадратурних алгоритмів рішення інтегральних рівнянь Вольєрра II роду в формі Гаммерштейна шляхом створення додаткових спеціалізованих модулів, сукупність яких є основою створення відповідного тулбокса, органічно доповнює апробовані блоки системи.

Список использованной литературы:

1. Чен К. Matlab в математических исследованиях / К. Чен, П. Джиблин, А. Ирвинг Дёч; пер. с англ. В. Е. Кондрашова и С. Б. Королева. — М. : Мир, 2001. — 346 с.
2. Манжиров А. В. Методы решения интегральных уравнений: справочник / А. В. Манжиров, А. Д. Полянин. — М. : Факториал, 1999. — 272 с.
3. Калиткин Н. Н. Численные методы / Н. Н. Калиткин. — М. : Наука, 1978. — 512 с.

The article is devoted to realization in computing environment Matlab of quadrature algorithms of solving for non-linear integral Volterra equations of the second kind in the Hammerstein form.

Key words: Matlab, integral Volterra equation of the second kind in the Hammerstein form, quadrature method, trapezoid rule.

Отримано 21.09.2010