

УДК 372.06

Д. А. Верлань*, аспирант,
С. Ю. Протасов**, канд. техн. наук

* Киевский национальный университет
имени Тараса Шевченко, г. Киев,

**Черкасский государственный технологический
университет, г. Черкассы

НЕКОТОРЫЕ ОСОБЕННОСТИ ЧИСЛЕННОЙ РЕАЛИЗАЦИИ НЕЛИНЕЙНЫХ ИНТЕГРАЛЬНЫХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

Рассмотрены различные виды нелинейных интегральных динамических моделей, и также подходы к построению численных алгоритмов их компьютерной реализации.

Ключевые слова: *нелинейная модель, интегральное уравнение, вырожденное ядро, алгоритм.*

Введение. Нелинейные динамические модели. Большой класс задач динамики описывается нелинейными интегральными уравнениями вида

$$y(x) - \int_0^x K[x, s, y(s)] ds = f(x), \quad (1)$$

где y — искомая, а K и f — заданные функции. Выражение (1) является нелинейным уравнением Вольтера второго рода, к которому принципиально сводятся произвольные задачи Коши для нелинейных обыкновенных дифференциальных уравнений. Уравнение (1), полученное непосредственно как описание некоторой нелинейной динамической задачи, не всегда может быть сведено без аппроксимации к дифференциальным уравнениям.

Еще более общей формой описания нелинейных динамически задач является интегральное уравнение вида

$$F[y(x)] - \int_0^x K[x, s, y(s)] ds = f(x), \quad (2)$$

или в частном случае

$$F[y(x)] - \int_0^x K(x, s)G[y(s)] ds = f(x), \quad (3)$$

где F и G — также заданные функции.

Для компьютерного решения нелинейных интегральных уравнений типа (1)–(3) в принципе могут быть применены любые подходящие численные методы, обеспечивающие получение результата. В случае задач

управления к алгоритмам решения нелинейных интегральных уравнений, реализуемым в специализированных ЭВМ, предъявляются несколько иные требования из-за необходимости учитывать режим реального времени. При этом важное значение приобретают малое количество циклических процедур, вычислительных операций, быстрая сходимость.

В связи с этим, актуальной задачей является выбор или разработка необходимого, часто специального, численного алгоритма. Ниже рассматривается частный случай данной проблемы применительно к сравнению некоторых алгоритмов для методов квадратур и последовательных приближений.

Численные алгоритмы. Согласно методу квадратур для уравнений вида (1) записывается выражение [1]:

$$y(x_i) - \int_0^{x_i} K[x_i, s, y(s)] ds = f(x_i), \quad i = \overline{1, n}, \quad (4)$$

от которого путем замены интеграла конечной суммой переходят к системе нелинейных рекуррентных соотношений

$$y(x_i) - \sum_{j=1}^i A_j K[x_i, x_j, y(x_j)] = f(x_i). \quad (5)$$

С помощью выражений (5) можно последовательно определить искомые приближенные значения $y(x_i)$, решая n нелинейных уравнений:

$$\begin{aligned} y_1 - A_1 K_{11}(y_1) &= f_1 = f(0), \\ y_2 - A_2 K_{22}(y_2) &= f_2 + A_1 K_{21}(y_1), \\ &\dots\dots\dots \\ y_n - A_n K_{nn}(y_n) &= f_n + \sum_{j=1}^{i-1} A_j K_{ij}(y_j), \end{aligned} \quad (6)$$

где A_j — коэффициенты квадратурных формул, $i = 2, 3, \dots, n$;

$$y_i = y(x_i); \quad x_i = (i-1)h; \quad K_{ij}(y_j) = K[x_i, x_j, y(x_j)].$$

Таким образом, метод квадратур приводит к получению аппроксимирующей системы конечных уравнений относительно дискретных значений искомой функции, т.е. включает в себя необходимый этап преобразования исходной задачи при компьютерной реализации. Использование аппроксимирующих выражений может быть различным. Одна из особенностей уравнений второго рода состоит в том, что они позволяют без особых затруднений получать различные итерационные схемы вычисления искомой функции. Для уравнения вида (3) можно получить, например, выражение

$$y_k(x) = F^{-1} \left[f(x) + \int_0^x K(x, s) G[y_{k-1}(s)] ds \right]. \quad (7)$$

Если в уравнении (3)

$$F[y(x)] = F_1[y(x)] + F_2[y(x)],$$

то итерационное соотношение может иметь вид

$$y_k(x) = F_1^{-1}[f(x) - F_2[y_{k-1}(x)] + \int_0^x K(x, s)G[y_{k-1}(s)]ds], \quad (8)$$

или

$$y_k(x) = F_2^{-1}[f(x) - F_1[y_{k-1}(x)] + \int_0^x K(x, s)G[y_{k-1}(s)]ds]. \quad (9)$$

Для уравнения

$$y(x) + F[y(x)] - \int_0^x K(x, s)G[y(s)]ds = f(x) \quad (10)$$

итерационное выражение получается наиболее просто:

$$y_k(x) = f(x) - F[y_{k-1}(x)] + \int_0^x K(x, s)G[y_{k-1}(s)]ds. \quad (11)$$

При численной реализации метода итераций интеграл заменяется конечной суммой. В этом случае (7)–(9) и (11) соответственно принимают вид

$$y_k(x_i) = F^{-1}[f(x_i) + \sum_{j=1}^i A_j K(x_i, x_j)G[y_{k-1}(x_j)]]; \quad (7a)$$

$$y_k(x_i) = F_1^{-1}[f(x_i) - F_2[y_{k-1}(x_i)] + \sum_{j=1}^i A_j K(x_i, x_j)G[y_{k-1}(x_j)]]; \quad (8a)$$

$$y_k(x_i) = F_2^{-1}[f(x_i) - F_1[y_{k-1}(x_i)] + \sum_{j=1}^i A_j K(x_i, x_j)G[y_{k-1}(x_j)]]; \quad (9a)$$

$$y_k(x_i) = f(x_i) - F[y_{k-1}(x_i)] + \sum_{j=1}^i A_j K(x_i, x_j)G[y_{k-1}(x_j)]. \quad (11a)$$

В случаях, когда в уравнениях (1) и (3) ядро является разделяющимся [2], т.е. имеет вид

$$K[x, s, y(s)] = \sum_{r=1}^m \alpha_r(x)\beta_r(s, y(s)), \quad (12)$$

$$K(x, s) = \sum_{r=1}^m \alpha_r(x)\beta_r(s), \quad (13)$$

где $r = \overline{1, m}$, имеется возможность повысить эффективность процесса решения, т.е. сократить время вычисления, машинную память, уменьшить ошибки [1].

Реализация алгоритмов. Несколько подробнее свойства метода квадратур можно рассмотреть на примере решения интегрально уравнения

$$y(x) - \int_0^x \frac{x^{-1}s + \frac{\sin s}{x}}{1 + y(s)} ds = \cos x - \operatorname{tg}\left(\frac{x}{2}\right), \quad (14)$$

относящиеся к уравнениям типа (1). После замены интеграла конечной суммой в соответствии с (5) имеем

$$y(x_i) - \frac{h}{2} A_i \frac{1 + \frac{\sin x_i}{x_i}}{1 + y(x_i)} = \cos x_i - \operatorname{tg}\left(\frac{x_i}{2}\right) + h \sum_{j=1}^{i-1} A_j \frac{x_i^{-1} x_j + \frac{\sin x_j}{x_i}}{1 + y(x_j)}, \quad (15)$$

где $A_j = \begin{cases} 0.5 & \text{при } j = 1, \\ 1 & \text{при } j > 1. \end{cases}$

Из (15) можно получить

$$y^2(x_i) + (1 - M)y(x_i) - C = 0, \quad (16)$$

где

$$M = \cos x_i - \operatorname{tg}\left(\frac{x_i}{2}\right) + h \sum_{j=1}^{i-1} A_j \frac{x_i^{-1} x_j + \frac{\sin x_j}{x_i}}{1 + y(x_j)},$$

$$C = \frac{h}{2} A_i \left(1 + \frac{\sin x_i}{x_i}\right) + M.$$

Окончательная расчетная формула численной реализации уравнения (14) имеет вид

$$y(0) = f(0),$$

$$y(x_i) = \frac{-(1 - M) + \sqrt{(1 - M)^2 + 4C}}{2}. \quad (17)$$

Используя свойство разделяемости ядра (12), можно построить модифицированный алгоритм вычислений. При этом (16) принимает вид

$$y^2(x_i) + (1 - M^*)y(x_i) - C^* = 0, \quad (18)$$

$$M^* = \cos x_i - \operatorname{tg}\left(\frac{x_i}{2}\right) + x_i^{-1} h \sum_{j=1}^{i-1} A_j \frac{x_j + \sin x_j}{1 + y(x_j)},$$

$$C^* = \frac{h}{2} A_i \left(1 + \frac{\sin x_i}{x_i}\right) + M^*.$$

Блок-схема модифицированного алгоритма вычислений приведена на рис. 1. Результаты решения уравнения (14) по алгоритму (17) и модифицированному алгоритму при $h = 0.02$ для 150 точек представлены в табл. 1, где также приведено точное решение. Время вычислений для уравнения (14) по традиционному алгоритму превышает время вычисления по модифицированному алгоритму, так как количество операций в традиционном алгоритме (17) зависит от номера шага.

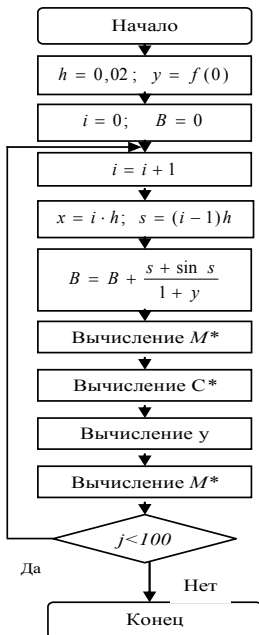


Рис. 1. Блок-схема модифицированного алгоритма

Таблица 1

Результат решения уравнения (14)

x	y	$y_{\text{мод}}$	$y_{\text{точ}}$
0	1	1	1
0.03	0.999553	0.999551	0.99955
0.09	0.995957	0.995956	0.995953
0.21	0.978039	0.978039	0.978031
0.6	0.825359	0.825358	0.825336
1.02	0.523409	0.523408	0.523366
1.41	0.160174	0.160173	0.160104
1.8	-0.22708	-0.22709	-0.2272
2.19	-0.58021	-0.58022	-0.58039
2.61	-0.86177	-0.86179	-0.862
3	-0.98977	-0.98978	-0.98999

Таким образом, если есть возможность, то квадратурные формулы целесообразно применять после выполнения операции разделения ядра.

Сравнение алгоритмов, вытекающих из методов итераций и квадратурных формул, позволяет получить данные для выбора необходимой процедуры, что можно рассмотреть на примере уравнения

$$\sqrt{2y(x)} - 1 + e^{-\sqrt{2y(x)}} = \int_0^x e^{-(x-s)} y(s) ds + y(x), \quad (19)$$

относящегося к уравнениям вида (3) и имеющего два точных решения $y(x) = 0$ и $y(x) = \frac{x^2}{2}$.

Воспользуемся выражениями (8а), (9а) и (11а), которые приводят к следующим расчетным выражениям:

$$y_k(x_i) = \frac{1}{2} \left[-y_{k-1}(x_i) + e^{-\sqrt{2y_{k-1}(x_i)}} - 1 + \sum_{j=1}^i A_j e^{x_j - x_i} y_{k-1}(x_j) \right]^2 = \psi_1(y, x_i); \quad (20)$$

$$y_k(x_i) = \frac{1}{2} \ln^2 \left[y_{k-1}(x_i) - \sqrt{2y_{k-1}(x_i)} + 1 - \sum_{j=1}^i A_j e^{x_j - x_i} y_{k-1}(x_j) \right] = \psi_2(y, x_i); \quad (21)$$

$$y_k(x_i) = \sqrt{2y_{k-1}(x_i)} + e^{-\sqrt{2y_{k-1}(x_i)}} - 1 + \sum_{j=1}^i A_j e^{x_j - x_i} y_{k-1}(x_j) = \psi_3(y, x_i). \quad (22)$$

Сходимость метода итераций зависит от вида представления уравнения (19) относительно неизвестной функции. Необходимое условие сходимости имеет вид [1]:

$$\max \left| \frac{\partial \psi(y, x_i)}{\partial y} \right| = m < 1. \quad (23)$$

При этом даже сходящиеся процедуры могут не привести к получению нужного результата, если исходное уравнение имеет несколько решений.

Проверка показывает, что в случае представлений уравнений (19) в виде (20), (22) процесс итераций не сходится, а итерации в виде (21) дают сходимость к нулевому решению.

Прямое применение к уравнению (19) метода квадратурных формул приводит к системе нелинейных уравнений относительно искомых значений $y(x_i)$, т.е. к системе, содержащей большое количество уравнений, что создает затруднения при реализации алгоритма в специализированных ЭВМ, несмотря на то, что матрица решаемой системы конечных уравнений является треугольной. Более простые расчетные выражения свойственны итерационным алгоритмам.

В связи с этим большей эффективности машинной реализации можно добиться применением комбинированных алгоритмов. Примером может служить следующий алгоритм применительно к уравнению (19). Интеграл заменяется конечной суммой по формуле трапеций с шагом h , что приводит к выражению

$$\begin{aligned} y(x_i) &= \sqrt{2y(x_i)} + e^{-\sqrt{2y(x_i)}} - 1 + h \sum_{j=1}^i A_j e^{x_j - x_i} y(x_j) = \\ &= \sqrt{2y(x_i)} + e^{-\sqrt{2y(x_i)}} - 1 + h \sum_{j=1}^{i-1} A_j e^{x_j - x_i} y(x_j) + \frac{h}{2} A_i e^{x_i - x_i} y(x_i). \end{aligned}$$

Используя теперь локальные итерации, получаем окончательную расчетную формулу

$$y_k(x_i) = \sqrt{2y_{k-1}(x_i)} + e^{-\sqrt{2y_{k-1}(x_i)}} - 1 + B + \frac{h}{2} y_{k-1}(x_i) = \psi_4(y, x_i), \quad (24)$$

где

$$B = h \sum_{j=1}^{i-1} A_j e^{x_j - x_i} y(x_j).$$

Выражение (23) позволяет записать условие сходимости

$$\left| \frac{\partial \psi_4(y, x_i)}{\partial y} \right| = \left| \frac{1 - e^{-\sqrt{2y}}}{\sqrt{2y}} + \frac{h}{2} \right| < 1,$$

и определить отсюда шаг $h \leq 0.7$, обеспечивающий его выполнение.

Итерационный процесс ведется до тех пор, пока не выполнится условие

$$\left| \frac{y_k(x_i) - y_{k-1}(x_i)}{y_k(x_i)} \right| = \delta \leq \varepsilon, \quad (25)$$

где ε — заданная относительная ошибка.

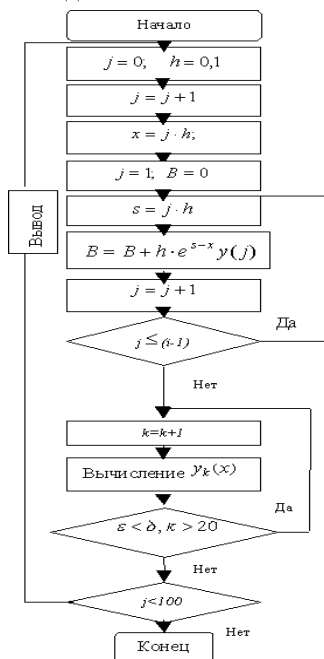


Рис. 2. Блок-схема алгоритма по расчетной формуле 24

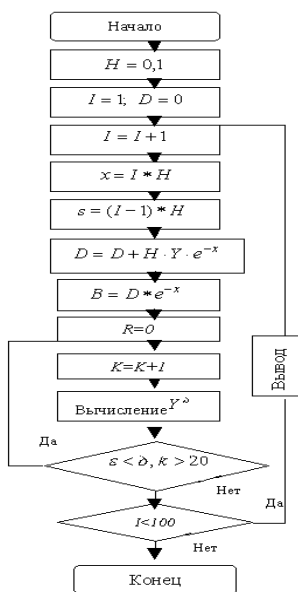


Рис. 3. Блок-схема алгоритма по расчетной формуле (26)

Блок-схема алгоритма вычислений представлена на рис. 2, а результаты вычислений в табл. 1 (для 100 точек при $h = 0.1$; $\varepsilon = 0,001$; $k = 20$), где также приведено точное решение. Свойство разделяемости ядра (13) позволяет построить модифицированный алгоритм, который приводит к сокращению времени вычислений и уменьшению ошибок округлений. В этом случае выражение (24) принимает вид

$$y_k(x_i) = \sqrt{2y_{k-1}(x_i)} + e^{-\sqrt{2y_{k-1}(x_i)}} - 1 + B^* + \frac{h}{2} y_{k-1}(x_i), \quad (26)$$

где $B^* = e^{-x_i} h \sum_{j=1}^{i-1} A_j e^{x_j} y(x_j)$.

Блок-схема модифицированного алгоритма приведена на рис. 3, а соответствующие результаты вычислений в табл. 2.

Таблица 2

Результаты решения уравнения (19)

x	y	y_{mod}	$y_{точ}$
0.0000	0.0000	0.0000000	0.0000000
1.0000	0.5000	0.47998315	0.47998315
2.0000	2.0000	1.9721012	1.9721003
3.0000	4.5000	4.4726210	4.4726229
4.0000	8.0000	7.9821358	7.9821377
5.0000	12.5000	12.501597	12.501593
6.0000	18.0000	18.32098	18.032047
7.0000	24.5000	24.574768	24.574722
8.0000	32.0000	32.130829	32.130783
9.0000	40.5000	40.701447	40.701416
10.0000	50.0000	50.287933	50.287903

Выводы. Наличие различных форм нелинейных интегральных динамических моделей приводят к ряду особенностей алгоритмов их численной реализации. Применение прямых квадратурных методов численного решения нелинейных интегральных уравнений включает в себя итерационные процедуры решения нелинейных алгебраических систем. Применение итерационных методов требует обеспечения условий сходимости и учёта неоднозначности получаемых результатов.

Список использованной литературы:

1. Верлань А. Ф. Методы решения интегральных уравнений с программы для ЭВМ / А. Ф. Верлань, В. С. Сизиков. — К. : Наук. думка, 1978. — 292 с.
2. Верлань Д. А. Алгоритм реалізації методу вироджених ядер при розв'язуванні інтегральних рівнянь Вольтерри / Д. А. Верлань // Математичне та комп'ютерне моделювання. Серія: Технічні науки : зб. наук. пр. / Ін-т кібернетики ім. В. М. Глушкова НАН України, Кам'янець-Подільський національний університет імені Івана Огієнка. — Кам'янець-Подільський : Кам'янець-Подільський національний університет імені Івана Огієнка, 2010. — Вип. 4. — С. 64–69.

Various types of nonlinear integral dynamic models, and the approaches to the construction of Num-represented algorithms for their computer implementation.

Key words: *non-linear model, the integral equation, the degenerate kernel algorithm.*

Отримано: 28.02.2014

УДК 627.324.2/3:532.72

А. П. Власюк*, д-р техн. наук, професор,

Т. П. Цветкова**, старший викладач

*Міжнародний економіко-гуманітарний університет імені академіка С. Дем'янчука, м. Рівне,

**Національний університет водного господарства та природокористування, м. Рівне

ЧИСЕЛЬНЕ МОДЕЛЮВАННЯ МАСОПЕРЕНЕСЕННЯ СОЛЬОВИХ РОЗЧИНІВ У НЕНАСИЧЕНИХ ШАРУВАТИХ ҐРУНТАХ

Здійснено математичне моделювання процесу масоперенесення сольових розчинів при вологоперенесенні в шарах області неповного насичення з врахуванням осмотичних явищ в нелінійному випадку. Чисельні розв'язки відповідних крайових задач знайдено методом скінчених різниць. В результаті програмної реалізації задачі проведено ряд чисельних експериментів та зроблено їх аналіз.

Ключові слова: *математична модель, масоперенесення, вологоперенесення, неповне насичення, шаруватий ґрунт, концентрація, напір, осмос.*

Вступ. Стрімке індустріальне виробництво, розвиток меліорації, будівництво підземних споруд та комунікацій зробили важливою необхідністю дослідження процесів, що протікають в приповерхневих шарах земної кори. Особливо важливе значення вивчення даних процесів стосується сільського господарства. Оскільки встановлення напорів вологи в ґрунтах є центральною проблемою для встановлення кількості водних ресурсів в засушливих регіонах, проектування дренажних та зрошувальних систем.

Питаннями масоперенесення та вологоперенесення займалось багато вчених. Однак процес масоперенесення солей розглядався без врахування вологоперенесення [1–3].

У зв'язку з цим, в даній роботі проведемо дослідження процесів масоперенесення з врахуванням вологоперенесення та осмотичних явищ в шаруватому ґрунті.