УДК 681.142

В. П. Бубнов, А. С. Еремин, С. А. Сергеев

ОСОБЕННОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ ЧИСЛЕННО-АНАЛИТИЧЕСКОГО МЕТОДА РАСЧЕТА МОДЕЛЕЙ НЕСТАЦИОНАРНЫХ СИСТЕМ ОБСЛУЖИВАНИЯ

Бубнов В.П., Еремин А.С., Сергеев С.А. **Особенности программной реализации численно-аналитического метода расчета моделей нестационарных систем обслуживания.**

Аннотация. В статье описывается численно-аналитический метод расчета моделей нестационарных систем обслуживания. Находится решение системы уравнений Чепмена — Колмогорова в аналитическом виде. Приводится алгоритм построения решения и особенности его программной реализации на языке Java. Также приводятся результаты сравнения времени работы и точности выходных данных метода со временем работы и точностью выходных данных численного метода типа Рунге — Кутты, который используется в Matlab для решения аналогичных задач.

Ключевые слова: нестационарные системы обслуживания, уравнения Чепмена — Колмогорова, численно-аналитический метод.

Bubnov V.P., Eremin A.S., Sergeev S.A. **Program Implementation of the Numerical-Analytical Method for Computation of Non-Stationary Service System Models.**

Abstract. A numerical-analytical method for non-stationary queueing systems models computation is presented. The solution of Chapman—Kolmogorov equations is found in the analytical form. The algorithm and its practical implementation with Java language are discussed. Computation time and results precision for the presented method and the Runge—Kutta type method used in Matlab are compared.

Keywords: non-stationary queueing systems, Chapman—Kolmogorov equations, numerical-analytical method

1. Введение. Отличительной чертой современных аппаратно-программных комплексов (АПК) является то, что при создании они, прежде всего, должны быть ориентированы на функционирование не только в нормальных, но и в критических (кризисных) условиях. Это обусловлено с одной стороны возрастанием угроз, вызванных техногенными, природными и человеческими факторами, а с другой — желанием использовать уже существующие АПК для решения новых более сложных задач. В рассматриваемых ситуациях мониторинг и прогнозирование должны сопровождаться целенаправленными процедурами реконфигурации структур (в общем случае, управления структурами) как самих АПК, так и систем управления (СУ) ими для обеспечения максимально допустимого уровня их работоспособности и пропускной способности. Для определения возможности реализации всех операций, связанных с технологическим циклом управления, на заданном временном интервале применяют математическое моделирование. Математической базой является теория массового обслужива-

ния (ТМО), позволяющая решать разнообразные задачи анализа и синтеза АПК путем определения технико-экономических показателей эффективности функционирования комплексов в целом при известных технических параметрах их элементов и рабочей нагрузке. Большинство авторов используют модели ТМО в предположении, что очередь заявок бесконечна, существует стационарный режим, а коэффициент загрузки не превышает единицы [1, 2]. Однако наибольший практический и теоретический интерес представляют модели нестационарных систем обслуживания, учитывающие поведение АПК в контуре управления технологическими процессами и объектами, функционирующими в условиях перегрузок на заданном (директивном) временном интервале. Этим объясняется появление в последнее время публикаций, связанных с исследованием поведения моделей ТМО в переходных режимах [3–7].

На основе результатов работ [8–11] разработан комплекс программ [12] расчета надежности и планирования испытаний программных средств, в котором системы обыкновенных дифференциальных уравнений (ОДУ) для различных моделей нестационарных систем обслуживания решаются с помощью численных методов. Популярным программным инструментом, например, является Matlab, предлагающий несколько процедур для решения систем ОДУ. Основными недостатками традиционных численных методов являются накапливаемая на каждом шаге интегрирования погрешность и время работы алгоритма. Причем, вычислительная погрешность может привести и к отсутствию физического смысла получаемого решения.

В частности, мы использовали для решения систем, которые будут описаны в следующем разделе, процедуру ode45 из пакета Matlab (при настройках по умолчанию), основанную на паре вложенных методов Рунге — Кутты порядков 4 и 5, разработанных Э. Фельбергом [13]. Глобальная погрешность этой процедуры имеет порядок $O(h^5)$, где h — максимальная величина шага, в данном случае, по времени. Положив число заявок на обслуживание равным 100 (что дает 5151 дифференциальное уравнение в системе Чепмена — Колмогорова), выбрав интенсивность поступления всех заявок 1, а интенсивность обработки 2, мы получили, что многие состояния имели отрицательную вероятность, в частности, вероятность состояния номер 5087 стала $-9{,}271964480522870 \cdot 10^{-9}$. Повышение же точности расчетов значительно увеличивает время работы программы.

В то же время, желание учета большего числа факторов реальных процессов, подлежащих моделированию, приводит к увеличению числа уравнений Чепмена — Колмогорова относительно вероятностей

состояний. Время численного решения такого рода систем уравнений, как показывают эксперименты, экспоненциально зависит от общего числа состояний систем обслуживания. Это накладывает серьезные ограничения на разрабатываемые модели нестационарных систем обслуживания.

Высказанные соображения свидетельствуют об актуальности разработки альтернативных методов расчета вероятностей состояний нестационарных систем обслуживания с конечным источником (НСО), рассматриваемых в настоящей работе.

Нами был разработан численно-аналитический подход к решению возникающих в НСО дифференциальных уравнений, пресдтавленный в данной статье.

2. Суть численно-аналитического метода решения систем дифференциальных уравнений. Суть численно-аналитического метода представлена на модели НСО. На вход последовательно поступает N запросов на обработку. Распределения временных интервалов между моментами поступления запросов описываются экспоненциальными законами с интенсивностями $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$, где λ_i соответствует i-й поступающей заявке. Считаем, что система не имеет потерь. Закон распределения времен обслуживания тоже экспоненциальный с интенсивностями $\{\mu_1, \mu_2, \dots, \mu_N\}$, где μ_j соответствует j-й обслуживаемой заявке.

Состояния системы в каждый момент времени характеризуются числом находящихся в системе запросов $i = \overline{0, N}$ и числом уже получивших обслуживание запросов $j = \overline{0, N-i}$. Вероятности пребывания системы в этих состояниях обозначается через $P_{i,j}(t)$. Их общее число $K = (N+1)(N+2)/2$. На рис. 1 представлена диаграмма переходов между состояниями системы.

Для определения распределения вероятностей нахождения системы обслуживания в состояниях (i, j) необходимо решить относительно $P_{i,j}(t)$ систему ОДУ, каждое из которых выглядит как

$$\begin{aligned}\dot{P}_{i,j}(t) = u(i)\left(\lambda_{i+j}P_{i-1,j}(t) - \mu_{j+1}P_{i,j}(t)\right) + \\ + u(j)\mu_j P_{i+1,j-1}(t) - u(N-i-j)\lambda_{i+j+1}P_{i,j}(t).\end{aligned} \quad (1)$$

Здесь $u(t)$ — функция Хевисайда, заданная как

$$u(t) = \begin{cases}1, & t > 0,\\ 0, & t \le 0.\end{cases} \quad (2)$$

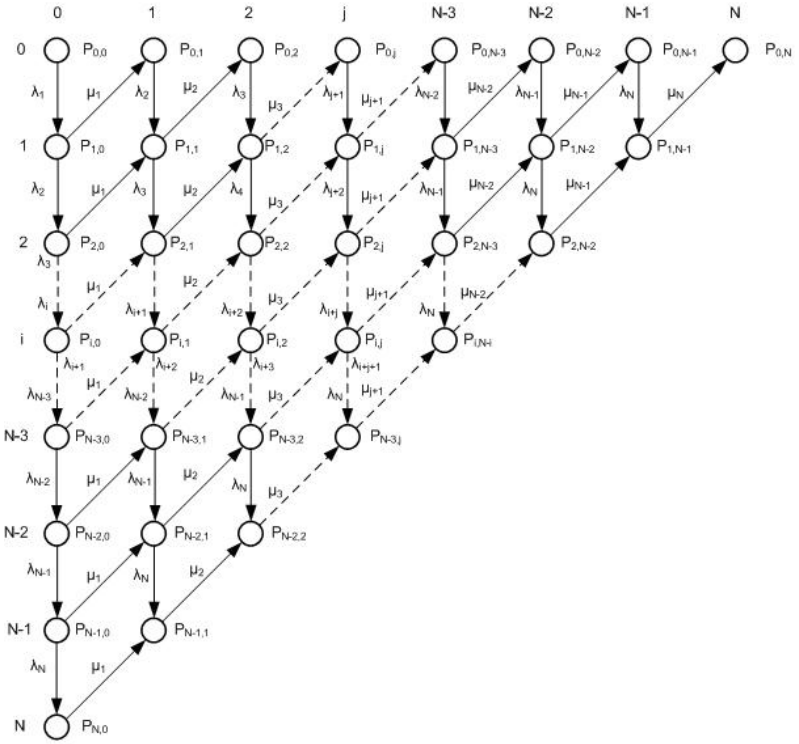

Рис. 1. Диаграмма переходов между состояниями НСО

В качестве начальных условий выбирают обычно нахождение в состоянии $(0, 0)$, то есть

$$P_{i,j}(0) = 1 - u(i + j). \tag{3}$$

Получаемое решение позволяет затем найти вероятности нахождения в системе i запросов, ожидаемое число запросов, вероятность обслуживания всех или не менее некоторого числа поступивших запросов и другие интересующие управленца характеристики.

Алгоритм нумерации состояний НСО. Рассмотренная выше система ОДУ (1) является линейной однородной системой уравнений. Она может быть представлена в матричной форме:

$$\dot{x}(t) = \mathbf{A}x(t), \tag{4}$$

где $x(t)$ — вектор неизвестных функций размерности K, а $\mathbf{A}$ — квадратная матрица.

Для указанной системы существует явное аналитическое решение, если только известны собственные числа матрицы **A**. Очевидным является тот факт, что в случае треугольного вида матрицы **A** (для определенности будем считать ее нижнетреугольной) ее собственные числа выписаны в явном виде на диагонали. Таким образом, аналитическое решение системы вида (4) можно легко найти, если только матрица **A** — треугольная.

Если пронумеровать состояния по возрастанию числа обработанных запросов, а внутри этих групп по возрастанию числа поступивших запросов:

(i,j)	$(0,0)$	$(1,0)$	...	$(N,0)$	$(0,1)$	...	$(N-1,1)$	...	$(0,N)$
k	1	2	...	$N+1$	$N+2$	...	$2N+1$	...	$(N+1)(N+2)/2$

то ни одно из состояний полученного списка не будет иметь зависимости от последующих. На графе это будет выглядеть как нумерация сверху вниз с перемещением по столбцам слева направо. Соответственно, полученная матрица **A** системы (4) будет нижнетреугольной.

Перепишем систему (4), учитывая треугольность матрицы **A**:

$$\begin{cases} \dfrac{dx_1}{dt} = a_{11}x_1, \\ \dfrac{dx_2}{dt} = a_{21}x_1 + a_{22}x_2, \\ \cdots \\ \dfrac{dx_i}{dt} = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ii}x_i, \\ \cdots \\ \dfrac{dx_K}{dt} = a_{K1}x_1 + a_{K2}x_2 + \cdots + a_{KK}x_K. \end{cases}, \tag{5}$$

где a_{ij} — интенсивность перехода из состояния с номером j в состояние с номером i (в векторе $x(t)$).

Решение системы ОДУ. Решение системы (5) разбивается на последовательное решение скалярных уравнений, первое из которых будет однородным, а все последующие будут включать в себя решения предыдущих уравнений в качестве неоднородности. Это позволяет сформулировать следующий алгоритм нахождения решения.

Решением первого уравнения (5) очевидно является

$$x_1(t) = x_1(0)e^{a_{11}t}, \tag{6}$$

и ему соответствует фундаментальное решение $\tilde{x}_1(t) = e^{a_{11}t}$.

Предположим, что решения первых $i-1$ уравнений найдены в форме:

$$x_j(t) = \sum_{w=1}^{j} k_{jw}\tilde{x}_w(t), \quad j = \overline{1, \iota - 1}, \tag{7}$$

причем $\tilde{x}_j(t) = t^{v_j-1}e^{a_{jj}t}$, где v_j — кратность собственного числа a_{jj} в системе первых j уравнений. Тогда, после подстановки (7) в i-е уравнение системы (5), последнее можно записать в виде:

$$\frac{dx_i}{dt}(t) = a_{ii}x_i(t) + \sum_{j=1}^{i-1} b_{ij}\tilde{x}_j(t), \tag{8}$$

где $b_{ij} = \sum_{w=j}^{i-1} a_{iw}k_{wj}$. Решение (8) будет иметь вид

$$x_i(t) = \sum_{j=1}^{i} k_{ij}\tilde{x}_j(t), \quad x_i(t) = t^{v_j-1}e^{a_{ii}t}. \tag{9}$$

Пусть a_{ii} — собственное число, кратность которого в системе первых i уравнений равна v_i. Собственные числа $a_{i_1i_1} = a_{i_2i_2} = \cdots = a_{i_{v_i}i_{v_i}}$, $i_{v_i} = i$, пронумерованы в порядке нахождения на диагонали матрицы A. Тогда

$$k_{ii_{v_i}} = \frac{b_{ii_{v_i-1}}}{v_i}, \qquad v_i > 1. \tag{10}$$

Коэффициенты k_{ij} при тех $\tilde{x}_j(t)$, для которых $a_{ii} \neq a_{jj}$:

$$k_{ij} = \begin{cases} \frac{b_{ij} - k_{iw}(v_j - 1)}{a_{jj} - a_{ii}}, & v_j > 1, \\ \frac{b_{ij}}{a_{jj} - a_{ii}}, & v_j = 1. \end{cases} \tag{11}$$

где w такая, что $a_{ww} = a_{jj}$ и $v_w = v_j - 1$.

После нахождения всех коэффициентов (10) и (11), через начальные данные найдем оставшиеся:

$$k_{ii_1} = x_i(0) - \sum_{\substack{j=1 \\ v_j=0}}^{i-1} k_{ij}. \tag{12}$$

Следует отметить, что нахождение решения системы ОДУ в данном случае, в отличие от численного метода, дающего конечный набор точек, представляет собой построение процедуры, позволяющей определить вероятности состояний НСО в произвольный момент времени. Это, во-первых, дает возможность вывести решение ОДУ с произвольной степенью детальности, а во-вторых, скорость нахождения решения в любой сколь угодно удаленный момент времени одинакова и не требует расчета многих предыдущих временных состояний, как это потребовалось бы численному методу, имеющему к тому же и методическую погрешность, оценка которой дополнительно увеличивает сложность решения.

3. Особенности программной реализации метода. Описанный в разделе 2 метод был реализован на объектно-ориентированном языке программирования Java. Первая версия реализации данного метода значительно выигрывала в скорости у Matlab. В таблице 1 приведено сравнение времени работы методов. Для всех заявок были установлены интенсивности поступления $\lambda = 1$ и обработки $\mu = 2$. Рассчитывались вероятности состояний при $t = 100$.

Таблица 1. Сравнение времени работы методов

Количество заявок (N)	Время расчета на Java, мс	Время расчета в Matlab, мс
3	1	24
5	3	29
10	21	49
15	22	60

Тестирование метода при больших N показало, что используемая реализация непригодна к практическому применению, в связи с появляющейся погрешностью. Это объясняется особенностями хранения чисел с плавающей точкой чисел в памяти ЭВМ. Стандартный тип хранения таких данных в Java — тип double, размерность которого равна 64 бита. На всех этапах алгоритма происходит округление переменных, вследствие чего происходит накопление погрешности и при больших N итоговая погрешность становится неприемлемо высокой.

Способ борьбы с погрешностью вычислений. Для решения данной проблемы было принято решение использовать числа с практически неограниченной разрядностью, реализованные в классе BigDecimal [14]. Эти числа представляют собой сочетание двух значений. Первое — неограниченное значение строкового типа (String, s), для хранения мантиссы числа, второе — 32-битное целое число (Integer, I), в котором хранится десятичный показатель. Таким образом, значение числа типа BigDecimal представляется в виде: $s \cdot 10^I$.

Для контроля погрешности в программу была добавлена возможность задания точности чисел, т. е. определения числа разрядов в мантиссе. Необходимо учитывать тот факт, что повышение точность хранения чисел приводит к увеличению времени на совершение арифметических операций над ними, и, соответственно, к увеличению времени, требуемого для нахождения решения системы.

В классе BigDecimal реализованы все необходимые в аналитическом методе арифметические операции кроме вычисления экспоненты. Данная функциональность была реализована в ходе разработки; в ее основе лежит ряд Тейлора

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}. \quad (13)$$

После реализации алгоритма с числами типа BigDecimal, его точность, но, к сожалению, и время работы значительно возросли. Так, например, при $N = 100$, $\lambda = 1$, $\mu = 2$ с использованием точности чисел в 1000 знаков расчет вероятностей состояний занимал около полутора часов до проведения оптимизация программной реализации.

Оптимизация реализации численно-аналитического метода. В ходе анализа времени работы алгоритма было выяснено, что основное время занимает расчет коэффициентов k по формулам (10)–(12). Для ускорения этой части программы было сделано несколько изменений в коде.

1) Изначально при вычислении коэффициентов по (11) выполнялся цикл от $j = 0$ до $j = i$. При такой реализации, при очередном j, таком, что $\nu_j > 1$, приходилось пересчитывать все предыдущие k_{iw}, для которых $a_{ww} = a_{jj}$. Это приводило к многократному пересчету одних и тех же значений и значительно увеличивало время работы программы. После оптимизации цикл стал выполняться от $j = i$ к $j = 0$. Таким образом, каждое значение k вычисляется только один раз.

2) Было установлено, что большинство элементов матрицы **A** равны 0, поэтому в (8) было добавлено условие:

$$b_{ij} = \sum_{w=j}^{i-1} a_{iw} k_{wj}, a_{iw} \neq 0, \quad (14)$$

что позволило значительно сократить время работы алгоритма, за счет отбрасывания операций, не влияющих на результат.

3) Изначально в программе выделалась оперативная память для хранения всех значений b, но было замечено, что при вычислении значений k_{ij}, используются только значения b_{is}, с тем же первым индексом i. Было принято решение выделять память только для хранения набора коэффициентов b_{ij} при фиксированном i, а при переходе к вычислению значений при новом i освобождать использованную память. Такая оптимизация позволила сократить требуемый размер памяти для хранения коэффициентов b в $(N+1)(N+2)/2$ раз и, как следствие, увеличить скорость работы программы.

4) Большая часть алгоритма может выполняться в многопоточном режиме, поэтому были определены точки, в которых потоки должны синхронизироваться, и реализована многопоточная структура алгоритма. Этими точками при вычислении k для любого i являются:

а) Окончание расчета коэффициентов b_{ij} (14),

б) Окончание расчета коэффициентов k_{ij}, для которых $a_{ii} = a_{jj}$ и $\nu_j > 1$ (10) и для которых $a_{ii} \neq a_{jj}$ (11).

в) Окончание расчета оставшихся коэффициентов k_{ij} (11), (12), выполняющийся в один поток вычислений.

Кроме того была проведена оптимизация расчета вероятностей состояний при вычисленных коэффициентах k:

1) Из формулы 7 видно, что значения $\tilde{x}_j(t)$ для двух собственных чисел таких, что $a_{i_j i_j} = a_{i_{j+1}, i_{j+1}}$, будут удовлетворять уравнению $\tilde{x}_{i_{j+1}}(t) = t\tilde{x}_{i_j}(t)$. Таким образом, при использовании этого уравнения в (7), удается значительно сократить время работы алгоритма.

Было замечено, что при вычислении вероятностей состояний на промежутке времени от t_1 до t_2 с шагом h в (7) целесообразно сохранять значения экспоненты в отдельный вектор E длиной K. Действительно, если значения экспоненты для момента времени t_1 сохранены, тогда для вычисления вероятностей состояний в момент времени $t_1 + h$ можно воспользоваться выражением $\tilde{x}_j(t+h) = t^{\nu_j - 1} e^{a_{jj}h} E_j$. Это значительно уменьшило время работы алгоритма.

Алгоритм вычисления вероятностей состояний по (7) также был разделен на потоки.

В результате внесения приведенных выше изменений, удалось добиться существенного ускорения работы программы при повышенной точности расчетов. В разделе 5 представлено сравнение скорости с Matlab.

4. Оценка точности метода. Одним из способов проверки точности работы метода является суммирование вероятностей всех состояний в какой-либо момент времени, так как суммарная вероятность всегда

остается равной 1. В рассматриваемой модели НСО для всех заявок были установлены интенсивности поступления $\lambda = 1$ и обработки $\mu = 2$, число заявок взято $N = 100$, точность чисел — 1000 знаков. Рассчитывались вероятности состояний при $t = 50$. Сумма всех вероятностей состояний, рассчитанных численно-аналитическим методом составила $1 + 616856 \cdot 10^{-678}$. При этом, ни одна вероятность не была отрицательной. В Matlab сумма всех состояний была равна 1, но были состояния, вероятность которых меньше нуля, в частности состояние номер 5087 имело вероятность $-9{,}271964480522870 \cdot 10^{-9}$.

Для дополнительной проверки точности на Java был реализован «классический» метод Рунге — Кутты четвертого порядка с постоянным шагом. В таблице 2 приведены значения вероятности поглощающего состояния с номером 5150 при $t = 50,\ 100,\ 150$. Интенсивность поступления заявок $\lambda = 1$, интенсивность обработки $\mu = 2$, $N = 100$. Шаг по времени в методе Рунге — Кутты обозначен h.

Таблица 2. Значения состояния 5150, вычисленные разными методами

	$t = 50$	$t = 100$	$t = 150$
Matlab	$4{,}24408126 \cdot 10^{-8}$	0,47343**46**	0,9999919**7951**
Аналитический метод	$2{,}24023344 \cdot 10^{-11}$	0,47343**80**	0,9999919**9164**
Метод Рунге — Кутты ($h = 0.4$)	$3{,}64428442 \cdot 10^{-11}$	0,47343**53**	0,9999919**9068**
Метод Рунге — Кутты ($h = 0.2$)	$2{,}23763751 \cdot 10^{-11}$	0,47343**78**	0,9999919**9158**

Из таблицы видно, что чем h меньше, тем ближе значение вероятности к результату, полученному аналитическим методом и наоборот, чем шаг больше, тем значение ближе к значению, полученному с использованием Matlab.

5. Сравнительная оценка скорости работы программной реализации метода. Был проведен эксперимент по сравнению скорости работы метода с Matlab. Параметры модели брались теми же, что и в предыдущем пункте. В таблице 3 приведено время работы метода и Matlab при расчете вероятностей состояний в различные моменты времени T, с учетом времени расчета коэффициентов k.

Таблица 3. Сравнение времени расчета вероятностей

	Нахождение решения в $T = 50$, с	Нахождение решения в $T = 100$, с	Нахождение решения в $T = 150$, с	Нахождение решения в $T = 200$, с
Matlab	8,370605	14,19324	20,578149	25,971104
Аналитический метод	55,961	59,98	58,637	61,616

Особенность нашего подхода такова что, рассчитав значения коэффициентов k, можно получить значения вероятностей состояний в

любой момент времени. Поэтому был проведен еще один эксперимент с теми же исходными данными, в котором аналитический метод использовал для расчета вероятностей заранее сохраненные значения коэффициентов k. Результаты эксперимента приведены в таблице 4.

Таблица 4. Сравнение времени расчета вероятностей при использовании уже найденных k

	Нахождение решения в $T = 50$, с	Нахождение решения в $T = 100$, с	Нахождение решения в $T = 150$, с	Нахождение решения в $T = 200$, с
Matlab	8,370605	14,19324	20,578149	25,971104
Аналитический метод	4,782	04,298	04,673	04,457

В третьем эксперименте проверялось влияние размерности чисел на скорость работы алгоритма. Интенсивность поступления заявок $\lambda = 1$, интенсивность обработки заявок $\mu = 2$, количество заявок $N = 100$, момент времени $T = 100$. При 500 знаках расчет занимает 58,086 с, а при 1000 — 59,980 с.

Из таблиц 3 и 4 видно, что наш метод тратит почти одинаковое время, независимо от момента T, для которого необходимо произвести расчет; кроме того, рассчитав и сохранив матрицу коэффициентов k (трудоемкий, но однократный процесс), можно получать вероятности состояний для любого момента времени намного быстрее по сравнению с MatLab.

Таким образом, рекомендуется использовать предложенный метод для решения класса задач, описанных во введении. Как видно из результатов, при определенных условиях, он значительно превосходит численные методы (на примере методов Рунге — Кутты) по скорости.

Литература

1. *Osogami T., Raymond R.* Analysis of transient queues with semidefinite optimization // Queueing Systems, 2013. vol. 73. pp. 195–234.
2. *Wolff R. W., Yao Y.-C.* Little's law when the average waiting time is infinite // Queueing Systems, 2014. vol. 76. pp. 267–281.
3. *Sudhesh R., Vijayashree K. V.* Stationary and transient analysis of M/M/1 G-queues // Int. J. of Mathematics in Operational Research, 2013. vol. 5. no. 2. pp. 282–299.
4. *Sudhesh R., Francis Raj L.* Stationary and transient solution of Markovian queues — an alternate approach // Int. J. of Mathematics in Operational Research, 2013. vol. 5. no. 3. pp. 407–421.
5. *Czachórski T., Nycz M., Nycz T., Pekergin F.* Analytical and numerical means to model transient states in computer networks // Computer Networks. Proceedings of the 20th International Conference, CN 2013. Springer Communications in Computer and Information Science, 2013. vol. 370. pp. 426–435.
6. *Wei Y., Yu M., Tang Y., Gu J.* Queue size distribution and capacity optimum design for N-policy Geo(λ1,λ2,λ3)/G/1 queue with setup time and variable input rate // Mathematical and Computer Modelling, 2013. vol. 57. no. 5–6. pp. 1559–1571.

7. *Бубнов В. П., Тырва А. В., Еремин А. С.* Комплекс моделей нестационарных систем обслуживания с распределениями фазового типа // Труды СПИИРАН, 2014. № 6(37), С. 61–71.
8. *Бубнов В. П., Сафонов В. И., Смагин В. Л.* О загрузке вычислительной системы с изменяющейся интенсивностью поступления заданий // Автоматика и вычислительная техника, 1987. № 6, С. 19–22.
9. *Бубнов В. П., Тырва А. В., Хомоненко А. Д.* Обоснование стратегии отладки программ на основе нестационарной модели надежности // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета, 2010. № 2(97), С. 85–92.
10. *Bubnov V. P., Tyrva A. V., Khomonenko A. D.* Model of reliability of the software with Coxian distribution of length of intervals between the moments of detection of errors // Proceedings of 34th Annual IEEE Computer Software and Applications Conference (COMPSAC 2010), 2010. pp. 238–243.
11. *Bubnov V. P., Khomonenko A. D., Tyrva A. V.* Software Reliability Model with Coxian Distribution of Length of Intervals Between Errors Detection and Fixing Moments // Proceedings of 35th Annual IEEE Computer Software and Applications Conference (COMPSAC 2011), Munich, 18–22 July 2011. pp. 310–314.
12. *Тырва А. В., Хомоненко А. Д., Бубнов В. П.* Комплекс программ расчета надежности и планирования испытаний программных средств // Федеральная служба по интеллектуальной собственности, патентам и товарным знакам. Свидетельство о государственной регистрации программ для ЭВМ № 2010615617. Москва, 2010.
13. *Fehlberg E.* Low-order classical Runge—Kutta formulas with step size control and their application to some heat transfer problems // NASA Technical Report 315 (1969), extract published in Computing, 1970. vol. 6, no. 1–2. pp. 61–71.
14. Class BigDecimal. Java™ Platform, Standard Edition 7. Application Programming Interface (API) Specification // URL: http://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html (дата обращения 10.02.2015).

References

1. Osogami T., Raymond R. Analysis of transient queues with semidefinite optimization. Queueing Systems, 2013. vol. 73. pp. 195–234.
2. Wolff R.W., Yao Y.-C. Little's law when the average waiting time is infinite. Queueing Systems, 2014. vol. 76. pp. 267–281.
3. Sudhesh R., Vijayashree K. V. Stationary and transient analysis of M/M/1 G-queues. Int. J. of Mathematics in Operational Research, 2013. vol. 5. no 2. pp. 282–299.
4. Sudhesh R., Francis Raj L. Stationary and transient solution of Markovian queues — an alternate approach. Int. J. of Mathematics in Operational Research, 2013. vol. 5. no. 3. pp. 407–421.
5. Czachórski T., Nycz M., Nycz T., Pekergin F. Analytical and numerical means to model transient states in computer networks. Computer Networks. Proceedings of the 20th International Conference, CN 2013, Lwówek Śląski, Poland, June 17–21, 2013 / Springer Communications in Computer and Information Science, 2013. vol. 370. pp. 426–435.
6. Wei Y., Yu M., Tang Y., Gu J. Queue size distribution and capacity optimum design for N-policy Geo(λ1,λ2,λ3)/G/1 queue with setup time and variable input rate. Mathematical and Computer Modelling, 2013. vol. 57. no. 5–6. pp. 1559–1571.
7. Bubnov V.P., Tyrva A.V, Eremin A.S. [A set of non-stationary queuing system models with phase-type distributions]. *Trudy SPIIRAN – SPIIRAS Proceedings*, 2014. vol. 6(37), pp. 61–71. (In Russ.)

8. Bubnov V.P., Safonov V.I., Smagin V.A. [The load of a computational system with varying customer arrival rate]. *Avtomatika i Vychislitel'naya Tekhnika – Automation and Computer Engineering.* 1987. vol. 6. pp. 19–22. (In Russ.).
9. Bubnov V.P., Tyrva A.V., Khomonenko A.D. [Software debugging strategy based on a non-stationary reliability model]. *Nauchno-tehnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politehnicheskogo universiteta – St. Petersburg State Polytechnical University Journal*, 2010. vol. 2(97), pp. 85–92.
10. Bubnov V.P., Tyrva A.V., Khomonenko A.D. Model of reliability of the software with Coxian distribution of length of intervals between the moments of detection of errors. Proceedings of 34th Annual IEEE Computer Software and Applications Conference (COMPSAC 2010), 2010. pp. 238–243.
11. Bubnov V.P., Khomonenko A.D., Tyrva A.V. Software Reliability Model with Coxian Distribution of Length of Intervals Between Errors Detection and Fixing Moments. Proceedings of 35th Annual IEEE Computer Software and Applications Conference (COMPSAC 2011), Munich, 18–22 July 2011. pp. 310–314.
12. Tyrva A.V., Khomonenko A.D., Bubnov V.P. [The program complex for software reliability computation and tests planning]. Russian Federal Service for Intellectual Property (Rospatent). Certificate of the state registration of a computer program No. 2010615617. Moscow, 2010. (In Russ.).
13. Fehlberg E. Low-order classical Runge—Kutta formulas with step size control and their application to some heat transfer problems. NASA Technical Report 315 (1969), extract published in Computing vol. 6, no. 1–2, 1970, pp. 61–71.
14. Class BigDecimal. Java™ Platform, Standard Edition 7. Application Programming Interface (API) Specification. Available at: http://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html (Accessed: 10.02.2015).

Бубнов Владимир Петрович — д-р техн. наук, профессор кафедры информационных и вычислительных систем факультета автоматизации и интеллектуальных технологий, Петербургский государственный университет путей сообщения императора Александра I (ПГУПС). Область научных интересов: вероятностные модели аппаратно-программных комплексов, марковские процессы, дифференциальные уравнения. Число научных публикаций — 150. bubnov1950@yandex.ru, http://www.pgups.ru; Московский пр., д. 9, г. Санкт-Петербург, 190031, РФ; р.т.: +79052807904, Факс: +7(812)457-8606.

Bubnov Vladimir Petrovish — Ph.D., Dr. Sci., professor of informatics and computer systems department, Petersburg state transport university. Research interests: probabilistic models of hardware and software complexes, Markovian processes, differential equations. The number of publications — 150. bubnov1950@yandex.ru, http://www.pgups.ru; Moskovsky pr., 9, Saint-Petersburg, 190031, Russian Federation; office phone: +79052807904, Fax: +7(812)457-8606.

Еремин Алексей Сергеевич — к-т техн. наук, доцент кафедры информационных систем факультета прикладной математики — процессов управления, Санкт-Петербургский государственный университет (СПбГУ). Область научных интересов: численные методы решения дифференциальных уравнений, уравнения с запаздывающих аргументом, вероятностные модели. Число научных публикаций — 13. ereminh@gmail.com, http://www.spbu.ru; Университетский пр. 35, Петергоф, г. Санкт-Петербург, 198504, РФ; р.т.: +7(812)428-7159, Факс: +7(812)428-7159.

Eremin Alexey Sergeevich — Ph.D., associate professor of information systems department of the Faculty of Applied Mathematics and Control Processes, Saint-Petersburg State University. Research interests: numerical solution of differential equations, delay differential equations,

probabilistic models. The number of publications — 13. ereminh@gmail.com, http://www.spbu.ru; Universitetskii prospekt 35, Petergof, Saint-Petersburg, 198504, Russian Federation; office phone: +7(812)428-7159, Fax: +7(812)428-7159.

Сергеев Сергей Александрович — аспирант кафедры информационных и вычислительных систем факультета автоматизации и интеллектуальных технологий, Петербургский государственный университет путей сообщения императора Александра I (ПГУПС). Область научных интересов: настационарные системы массового обслуживания, программные комплексы. Число научных публикаций — 12. serega_svetl@mail.ru; Московский пр., д. 9, г. Санкт-Петербург, 190031, РФ; р.т.: 8(911)959-53-25.

Sergeev Sergei Aleksandrovich — Ph.D. student of the informatics and computer systems department, Petersburg State Transport University. Research interests: non-stationary queueing systems, software complexes. The number of publications — 12. serega_svetl@mail.ru; Moskovsky pr., 9, Saint-Petersburg, 190031, Russian Federation; office phone: 8(911)959-53-25.

Поддержка исследований. Работа выполнена при финансовой поддержке РФФИ (гранты № 13-07-00279, 13-08-00702, 13-08-01250, 13-06-00877, 13-07-12120-офи-м), Программы фундаментальных исследований ОНИТ РАН (проект No2.11)

Acknowledgements. This research is partially supported by the RFBR (grants 13-07-00279, 13-08-00702, 13-08-01250, 13-06-00877, 13-07-12120-офи-м), and by the fundamental scientific research support ONITRAS Project No. 2.11.

РЕФЕРАТ

Бубнов В. П., Еремин А. С., Сергеев С. А. **Особенности программной реализации численно-аналитического метода расчета моделей нестационарных систем обслуживания.**

В статье описывается численно-аналитический метод расчета моделей нестационарных систем обслуживания (НСО). В отличие от традиционных моделей, они позволяют моделировать процессы обслуживания на заданном временном интервале при общих предположениях о законах распределения временных интервалов между поступлениями и обслуживаниями заявок. Важным в этом случае становится решение системы уравнений Чепмена — Колмогорова.

Рассматривается способ нумерации уравнений, позволяющий найти решение в аналитическом виде, как стационарной однородной системы линейных обыкновенных дифференциальных уравнений (ОДУ) с нижнетреугольной матрицей.

Приводится алгоритм построения решения и особенности его программной реализации на языке Java. Обсуждаются вопросы повышения скорости расчетов и точности путем использования чисел типа BigDecimal.

Проводится сравнение эффективности предлагаемого подхода с результатами расчетов в комплексе MatLab, в котором реализовано решение систем ОДУ численными методами типа Рунге — Кутты.

SUMMARY

Bubnov V.P., Eremin A.S., Sergeev S.A. **Program Implementation of the Numerical-Analytical Method for Computation of Non-Stationary Service System Models.**

Numerical-analytical method for solving non-stationary queueing systems (NQS) is presented. Such models describe the processes of customers servicing in the specified time interval under general assumptions on the time distribution between customer arrival and service. The Chapman—Kolmogorov equations solution becomes important in this case.

The equations are numerated so, that the solution can be found analytically, as a solution of a stationary homogeneous system of linear ordinary differential equations (ODEs), which has a lower triangular matrix.

The algorithm of the solution computation and its implementation peculiarities with Java language are described. Computation time and precision improvements with using of the BigDecimal type are discussed.

The efficiency of the approach is compared to results of computation in MatLab, where ODEs are solved with Runge—Kutta type methods.