

УДК 004.422.8

ПОСТРОЕНИЕ КОМПАРТМЕНТАЛЬНЫХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ СИСТЕМ С ПРИМЕНЕНИЕМ ПРОГРАММНОГО КОМПЛЕКСА СИМВОЛЬНЫХ ВЫЧИСЛЕНИЙ НА ЯЗЫКЕ JULIA

© 2024 г. А. В. Демидова^{а, *}, О. В. Дружинина^{б, **}, О. Н. Масина^{с, ***}, А. А. Петров^{с, ****}^аРоссийский университет дружбы народов, 117198 Москва, ул. Миклухо-Маклая, д. 6, Россия^бФедеральный исследовательский центр “Информатика и управление” Российской академии наук
119333 Москва, ул. Вавилова, д. 44, к. 2, Россия^сЕлецкий государственный университет им. И. А. Бунина
399770 Елец, Липецкая обл., ул. Коммунаров, д. 28, Россия

*E-mail: demidova-av@rudn.ru,

**E-mail: ovdruzh@mail.ru,

***E-mail: olga121@inbox.ru,

****E-mail: xeal91@yandex.ru

Поступила в редакцию 01.08.2023

После доработки: 10.09.2023

Принята к публикации 01.10.2023

В работе рассмотрены вопросы построения компартментальных моделей динамических систем с применением программного комплекса символьных вычислений на языке Julia. Программный комплекс направлен на решение задачи унификации формализованного построения моделей с учетом сущностного описания возможных взаимодействий компартментов и влияния различных факторов на эволюцию систем. Развивается подход к разработке инструментально-методического обеспечения моделирования динамических систем, поведение которых может быть охарактеризовано одношаговыми процессами. Разработанное программное обеспечение позволяет получить символьное представление дифференциальных уравнений модели как в стохастическом, так и в детерминированном случае. Предложенный программный комплекс реализован с помощью языка Julia и использует библиотеку компьютерной алгебры Julia Symbolics. Представлено сравнение инструментария Julia Symbolics с другими системами компьютерной алгебры. Рассмотрено применение разработанного программного комплекса к компартментальной модели распространения эпидемии. Результаты могут найти применение при решении задач конструирования и исследования динамических моделей естествознания, представляемых одношаговыми процессами.

Ключевые слова: компартментальные модели, динамические системы, компьютерная алгебра, язык программирования Julia, программный комплекс символьных вычислений

DOI: 10.31857/S0132347424020051 EDN: RPCESS

1. ВВЕДЕНИЕ

Направление, связанное с формализованным представлением моделей динамических систем в символьном виде, относится к числу актуальных научных направлений [1–5]. В частности, теоретический и прикладной интерес представляет построение моделей динамических систем с привлечением методов и средств компьютерной алгебры.

В настоящее время возникает необходимость описания и изучения новых факторов, влияющих на динамику распространения эпидемий [6–10]. Математическому моделированию процессов, связанных с пандемией Covid-19, посвящены, в частности, работы [11–13]. Моделирование эпидемиологических процессов позволяет учесть факторы,

влияющие на распространение заболеваний, и оценить действенность различных мер, направленных на снижение риска заболевания. Например, действие таких мер, как дистанцирование, ношение масок, различные протоколы обслуживания в медицинских учреждениях и многое другое, могут быть проанализированы и оптимизированы с помощью эпидемиологических моделей.

Описание различных эффектов в моделях распространения эпидемий связано с повышением размерности, что приводит к достаточно громоздким формализованным представлениям взаимосвязей переменных и параметров. Поэтому целесообразно использовать инструменты компьютерной алгебры для реализации возможностей построения моделей из принципов взаимодействия элементов системы.

Развиваемый в настоящей работе подход к разработке инструментально-методического обеспечения моделирования динамических систем базируется на реализации возможностей построения и исследования систем, поведение которых может быть описано одношаговыми процессами. Как известно, класс одношаговых процессов охватывает многие процессы физики, химии, экологии, демографии, социологии. Большинство моделей для описания процессов распространения эпидемий относится к классу компартментальных моделей, которые являются частным случаем моделей одношаговых процессов.

Для описания компартментальных моделей может быть использован метод построения самосогласованных стохастических моделей [14–17]. В основе этого метода лежит комбинаторная методология [18, 19], в соответствии с которой предполагается, что эволюция многомерной системы рождения–гибели может быть рассмотрена как результат индивидуальных взаимодействий между элементами этой системы. Благодаря указанному методу возможно получить стохастическую модель с согласованной стохастической и детерминистической частями, что позволяет оценить влияние введения стохастики на поведение системы. Метод построения самосогласованных стохастических моделей удобен для дальнейшей разработки программного обеспечения, поскольку позволяет реализовать автоматизированное построение компартментальных моделей по описанию взаимодействий между элементами и по характеру переходов из состояния в состояние.

Одной из важных задач, возникающих при моделировании компартментальных систем, является создание эффективного и удобного для использования программного обеспечения, позволяющего достаточно просто формализовать возможные переходы из состояния в состояние. Разработка таких инструментов компьютерной алгебры, которые являются высокопроизводительными, гибкими, масштабируемыми, предполагает использование широкого спектра технологий программирования. Одним из современных языков программирования, ориентированных на математическое моделирование с высоким уровнем абстракции, является Julia [20]. Данный язык позволяет реализовать алгоритмы символьных вычислений для решения различных задач [21–23].

Настоящая работа посвящена проблеме построения как детерминированных, так и стохастических компартментальных моделей с применением программного комплекса символьных вычислений на

языке Julia. Структура статьи следующая. В разделе 2 приводится описание подходов к построению эпидемиологической модели SIR и ее модификаций. В разделе 3 рассмотрены особенности использования языка Julia для решения задач моделирования динамических систем. Представлено сравнение инструментария Symbolics.jl с другими системами компьютерной алгебры. В разделах 4 и 5 изложены основные результаты работы. В разделе 4 дано описание программного комплекса символьных вычислений с учетом различных функциональных модулей комплекса и приведены основные фрагменты кода. В разделе 5 рассмотрено применение разработанного программного комплекса на примере модели распространения эпидемии Covid-19. В разделе 6 приведено обсуждение результатов работы.

2. ОСОБЕННОСТИ И МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ МОДЕЛИРОВАНИЯ ЭПИДЕМИОЛОГИЧЕСКИХ СИСТЕМ

С целью построения различных классов эпидемиологических моделей обычно используют два подхода: подход, приводящий к построению статистических моделей, и подход, приводящий к построению компартментальных моделей.

Первый подход связан с применением методов математической статистики и машинного обучения для прогнозирования процесса эпидемии в краткосрочной перспективе при отсутствии резких изменений ситуации [24, 25]. Прогнозирование базируется на анализе статистических данных, таких как отчеты о заболевших, умерших и вылечившихся от конкретного заболевания. Второй подход основан на разделении популяции на различные группы (компарменты) по отношению к инфекции и на определении правил перехода из одной группы в другую. При этом процесс распространения эпидемии представляется в виде графа переходов. Реализация таких компартментальных моделей заключается в описании уравнений переходов и последующем анализе динамики численности каждой группы в течение определенного периода времени. В качестве математического аппарата для этого класса моделей чаще всего используются системы дифференциальных или разностных уравнений, однако в последнее время набирает популярность алгоритмическое моделирование, в частности, построение имитационных и агентно-ориентированных моделей [11, 26].

Как известно, при моделировании распространения инфекционных заболеваний существенное значение могут иметь демографические процессы.

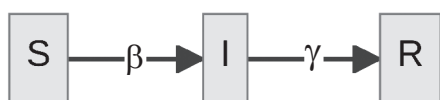


Рис. 1. Диаграмма модели SIR.

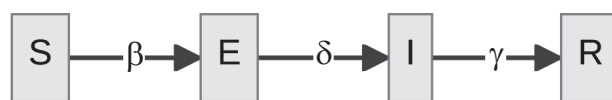


Рис. 2. Диаграмма модели SEIR.

Поэтому при построении модели в зависимости от специфики инфекционной болезни может потребоваться учет таких процессов, как рождаемость, смертность и миграция. Так, например, при моделировании заболеваний с длительным инкубационным или инфекционным периодом (туберкулез, ВИЧ/СПИД) необходимо учитывать рождаемость и смертность. Это связано с тем, что продолжительность болезни сравнима с продолжительностью жизни, и демографические процессы могут оказать большое влияние на распространение инфекции. В свою очередь, при моделировании эпидемий заболеваний с коротким инфекционным периодом (корь, коклюш, грипп) демографическими процессами можно пренебречь [27].

Компартментальные модели удобны для изучения влияния различных факторов, таких как вакцинация, изоляция и т.п., на течение динамики эпидемии в связи с тем, что для учета дополнительного фактора достаточно добавить в граф переходов новый узел и определить его связи с остальными узлами графа.

Классической моделью эпидемии является модель SIR, предложенная В. Кермаком и А. МакКендриком [28]. В этой модели различают три состояния индивидуумов: S – восприимчивый, I – инфицированный, R – выздоровевший (или невосприимчивый). Процессы, характерные для модели SIR, могут быть представлены диаграммой на рис. 1, где β – интенсивность заражения, γ – интенсивность выздоровления.

Система дифференциальных уравнений, соответствующая данной диаграмме, имеет вид:

$$\frac{dS}{dt} = -\beta SI, \quad \frac{dI}{dt} = \beta SI - \gamma I, \quad \frac{dR}{dt} = \gamma I. \quad (1)$$

Самой известной модификацией модели SIR является модель SEIR, которая учитывает количество бессимптомных переносчиков инфекции E . Диаграмма для данной модели может быть получена добавлением узла E в диаграмму модели SIR. На рис. 2 представлена диаграмма модели SEIR, где β – интенсивность заражения, γ – интенсивность выздоровления, δ – величина, обратная среднему инкубационному периоду заболевания.

В последнее время появилось большое количество работ, в которых рассматриваются модификации модели SIR для моделирования эпидемии COVID-19. Указанные модификации получены

усложнением диаграммы SIR с помощью добавления в нее различных узлов. Так, например, в [12] описана такая модель SAPHIRE, которая помимо компартментов S , I и R учитывает дополнительные группы индивидуумов: подвергшиеся воздействию инфекции (E), бессимптомные больные (P), зарегистрированные больные (J), незарегистрированные больные (A) и больные, изолированные в больнице (H).

Другая модель SIDARTHE распространения COVID-19 [13] включает в себя дополнительные группы индивидуумов по отношению к модели SAPHIRE, а именно, больные с опасными для жизни симптомами и бессимптомные больные (разделенные на выявленных и невыявленных).

Таким образом, компартментальные модели могут иметь сколь угодно сложную структуру за счет расширения первоначальной модели добавлением новых узлов. В результате усложнения структуры приобретает актуальность решение задачи формализованного построения таких систем дифференциальных уравнений, которые соответствуют графам рассматриваемых моделей.

В данной работе развивается подход к построению компартментальных моделей в виде систем дифференциальных уравнений для описания процессов распространения эпидемий с возможностью учета разнообразных факторов. В рамках данного подхода реализуется возможность построения как детерминированных, так и стохастических компартментальных моделей.

Развиваемый нами подход к построению компартментальных моделей распространения эпидемий согласуется с методологией построения моделей одношаговых процессов [14, 15]. В рамках этой методологии сначала необходимо сформулировать в терминах одношаговых процессов все взаимодействия между ее элементами системы. Далее необходимо составить такую схему взаимодействия элементов, на основе которой записываются операторы состояния системы и оператор изменения состояния системы. В результате произведения оператора изменения состояния системы и вектора интенсивностей переходов можно получить коэффициенты сноса и диффузии для уравнения Фоккера–Планка. Таким образом, можно получить кроме стохастического описания системы также детерминистическое, которому соответствует вектор сносов.

В следующем разделе мы анализируем такие возможности и особенности языка Julia и системы компьютерной алгебры Symbolics.jl, которые используются для программной реализации описанного подхода к построению компартментальных моделей.

3. ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ЯЗЫКА JULIA ДЛЯ РЕШЕНИЯ ЗАДАЧ МОДЕЛИРОВАНИЯ

Язык программирования Julia представляет собой высокоуровневый интерпретируемый язык с динамической типизацией, который был создан специально для научных вычислений. Одним из важнейших принципов разработки данного языка является обеспечение максимально высокой производительности программного обеспечения, что достигается благодаря использованию Low Level Virtual Machine (LLVM) в качестве кодогенератора. В ряде работ выполнен сравнительный анализ производительности в научных задачах (см., например, [29]). Результаты проведенного анализа демонстрируют, что скорость выполнения операций с применением языка Julia сравнима с наиболее высокопроизводительными математическими пакетами и языками программирования.

Следует отметить, что уровень выразительности языка по отношению к задачам математического моделирования является достаточно высоким. По сравнению с классическими объектно-ориентированными языками (такими, как Python, C++, Java) Julia основана на концепции множественной диспетчеризации. Концепция характеризуется тем, что главными объектами являются так называемые мультиметоды, а система типов образует иерархическую структуру. По сравнению с Python, где математические операции над тензорами требуют наличия дополнительных библиотек типа Numpy, матрицы и векторы в Julia являются элементами базовой иерархии типов. Благодаря этому факту существует возможность достаточно гибко определять логику программы.

Развитые средства метапрограммирования в Julia позволили создать удобные и высокопроизводительные средства численных методов, математического моделирования и компьютерной алгебры. Пакет DifferentialEquations.jl на сегодняшний день является одним из наиболее функциональных пакетов для численного решения дифференциальных уравнений. В частности, поддерживаются функции решения следующих типов уравнений:

- дискретные уравнения (функциональные карты, дискретные стохастические (Gillespie/Markov) симуляции);
- обыкновенные дифференциальные уравнения (ODEs);
- сплит и разделенные ODE (симплектические интеграторы, методы IMEX);
- стохастические обыкновенные дифференциальные уравнения (SODE или SDE);
- стохастические дифференциально-алгебраические уравнения (SDAE);
- случайные дифференциальные уравнения (RODE или RDE);
- дифференциальные алгебраические уравнения (DAEs);
- дифференциальные уравнения с запаздыванием (DDE);
- нейтральные, отстающие и алгебраические дифференциальные уравнения с запаздыванием (NDDE, RDDE и DDAE);
- стохастические дифференциальные уравнения с запаздыванием (SDDE);
- экспериментальная поддержка стохастических нейтральных, отстающих и алгебраических дифференциальных уравнений с запаздыванием (SNDDE, SRDDEs и SDDAEs);
- смешанные дискретные и непрерывные уравнения (гибридные уравнения) Jump Diffusions;
- детерминированные и стохастические дифференциальные уравнения в частных производных.

Для DifferentialEquations.jl поддерживается широкая интеграция с возможностями других пакетов, в частности ускорение на специализированных процессорах (GPU), автоматическое детектирование разреженности, символьное вычисления якобианов, интерполяция решений. Возможности пакета более подробно рассмотрены, например, в [30]. Средства символьной компьютерной алгебры в языке Julia реализуются посредством пакета Symbolics.jl. Высокопроизводительным символьным вычислениям в Julia посвящена работа [21]. Существует ряд работ, посвященных применению языка Julia при моделировании процессов в различных областях знания. В частности, вопросам биохимического моделирования в Julia средствами Catalyst.jl посвящена работа [31].

Следует отметить, что разработан ряд библиотек, направленных на решение прикладных вопросов математического моделирования. В частности, в [32] рассмотрены вопросы моделирования распространения патогенов и вопросы разработки специализированной библиотеки Pathogen.jl.

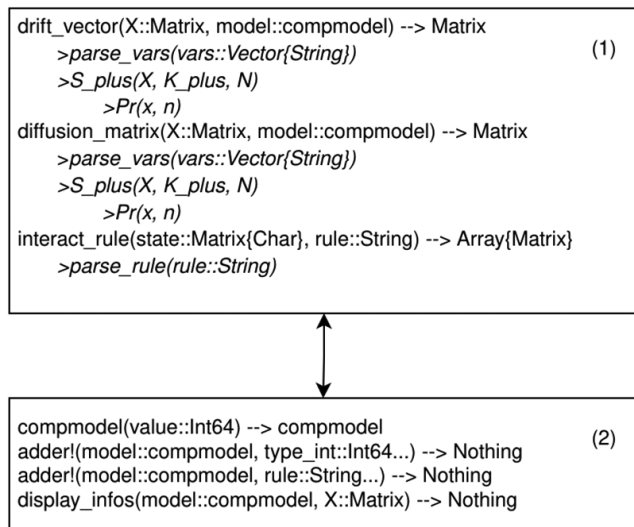


Рис. 3. Структура программного комплекса.

Вопросы применения численно-аналитических методов для моделирования систем, задаваемых дифференциальными уравнениями, с применением языка Julia изложены в работе [23]. Язык Julia может быть использован для моделирования сложных технических систем [33, 34].

Сравнение Symbolics.jl с другими известными системами компьютерной алгебры представлено в табл. 1.

Следует отметить, что основными преимуществами Symbolics.jl являются высокая производительность, открытость и простота интеграции благодаря возможности генерации кода для языка C. Выразительность языка Julia по отношению к научным задачам и развитые возможности метапрограммирования предоставляют удобные инструменты для решения задач формализации компартментальных моделей в символьной форме.

4. ОПИСАНИЕ ПРОГРАММНОГО КОМПЛЕКСА СИМВОЛЬНЫХ ВЫЧИСЛЕНИЙ

В данном разделе мы приводим описание программного комплекса символьных вычислений, который позволяет получать детерминистическое и

стохастическое описание модели по схеме взаимодействий ее элементов. Разработка указанного программного комплекса осуществлена на языке Julia с применением библиотеки Symbolics.jl, а также результатов [5].

Программный комплекс состоит из конструирующего и функционального блоков. Конструирующий блок определяет структуру модели. Указанная структура включает в себя размерность фазового пространства системы, матрицы переходов, счетчик коэффициентов интенсивности переходов, текстовое представление коэффициентов интенсивности переходов, операторы состояний M , N и оператор переходов r . Фрагмент программного кода, описывающего структуру модели и функцию-конструктор, представлен в листинге 1.

```

using Symbolics
""" value - количество компартментов
    """
mutable struct compmodel
    value::Int64
    M::Union{Vector,Matrix}
    N::Union{Vector,Matrix}
    eq_nub::Int64
    coef::Array{String}
    interaction::Matrix{Int64}
end
compmodel(value::Int64) =
    ↪ compmodel(value, zeros(value),
    ↪ zeros(value), 0, [],
    ↪ Array{Any}(undef, 0, 0))

```

Листинг 1. Программный код структуры модели.

Кроме того, конструирующий блок включает в себя связанные со структурой модели функции, которые задают и отображают схему взаимодействий. Указанным функциям соответствуют метод display_infos и мультиметод adder!.

Метод display_infos отвечает за вывод правил переходов и может быть использован для проверки

Таблица 1. Сравнение систем компьютерной алгебры

CAS / Особенности	Цена и лицензия	Быстродействие	Сложность интеграции	Функциональность
Symbolics.jl	Бесплатно/ открытая	Высокое	Невысокая	Средняя
SymPy	Бесплатно/ открытая	Низкое	Невысокая	Высокая
Maxima	Бесплатно/ открытая	Высокое	Высокая	Высокая
Symbolic Math Toolbox (MATLAB)	Варьируется/ закрытая	Высокое	Невысокая	Высокая
Mathematica	Варьируется/ закрытая	Высокое	Невысокая	Высокая

корректности работы программного комплекса. Мультиметод `adder!` позволяет добавлять в систему правила переходов двумя способами. Первый способ основан на задании схемы взаимодействий в виде предварительно определенного вектора и номеров компартментов. Второй способ базируется на задании схемы взаимодействий в символьном виде. Заголовки функций метода `adder!` представлены в листингах 2, 3.

```
function adder!(model::compmodel,
  ↪ type_int::Int64,
  id_pop_1::Int64, id_pop_2::Int64,
  coef = nothing,
  ↪ interact::Union{Nothing,Matrix} =
  ↪ nothing)
```

Листинг 2. Заголовок метода `adder!` с использованием таблицы взаимодействий.

```
function adder!(model::compmodel,
  ↪ rule::String,
  state::Matrix, coef::String)
```

Листинг 3. Заголовок метода `adder!` с использованием правил.

В листинге 2 представлен заголовок метода, получающего следующие аргументы: модель, тип взаимодействия (задается таблицей взаимодействий), номера компартментов `id_pop_1` и `id_pop_2`, коэффициент интенсивности перехода в символьной форме. Отметим, что по умолчанию коэффициент назначается из диапазона $k_1, k_2, \dots, k_n$, где n — количество правил переходов.

В листинге 3 представлен заголовок метода, получающего следующие аргументы: модель, тип взаимодействия в символьной форме, фазовые переменные в символьной форме, коэффициент интенсивности переходов. Следует отметить, что обе реализации мультиметода являются in-place, т.е. изменяют состояние модели, при этом возвращается значение “nothing”.

Функциональный блок программного комплекса содержит в себе функции, которые предназначены для построения схемы взаимодействия на основе описания, а также для вывода вектора сноса и матрицы диффузии.

Основным методом блока является метод `interact_rule`, который преобразует текстовое описание правила в форму

$$“a + b \sim c + d” \text{ или } “a \sim c”$$

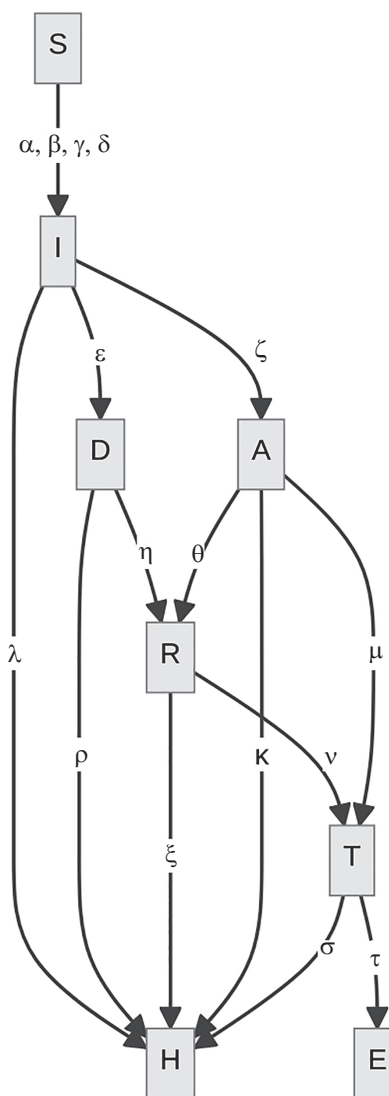


Рис. 4. Диаграмма модели SIDARTHE.

в векторы M, N для соответствующей модели. Указанный метод использует вспомогательный метод `parse_rule`, который разбирает текстовое правило на антецеденты и консеквенты.

Методы `drift_vector` и `diffusion_matrix` отвечают за расчет вектора сноса и матрицы диффузии соответственно. Указанные методы принимают в качестве аргументов модель и вектор состояний в символьной форме.

Кроме того, в функциональном блоке присутствуют служебные методы `S_plus`, `Pr` и `parse_vars`. Методы `S_plus`, `Pr` реализуют математические операции для методов `drift_vector` и `diffusion_matrix`. Метод `parse_vars` отвечает за преобразования коэффициентов в текстовой форме в символьные переменные.

Общая структура разработанного программного комплекса представлена на рис. 3.

На рис. 3 через (1) обозначен функциональный блок программного комплекса, через (2) обозначен конструирующий блок программного комплекса. Приведены заголовки разработанных методов с возвращаемыми типами. Вызовы служебных методов отмечены курсивным текстом. Диаграмма на рис. 3 приведена для пояснения функциональных возможностей разработанного программного комплекса.

Таким образом, программный комплекс позволяет формализовать компартментальные модели в символьной форме по описанию взаимодействий между элементами. Далее рассмотрим пример построения компартментальной модели с применением разработанного программного обеспечения.

5. ПРИМЕР ПОСТРОЕНИЯ КОМПАРТМЕНТАЛЬНОЙ МОДЕЛИ

Для демонстрации возможностей реализованного программного обеспечения предлагается рассмотреть модель SIDARTHE распространения эпидемии Covid-19 [13]. Выбор данной модели обусловлен тем, что в ней учитывается достаточно много дополнительных факторов по сравнению с моделью SIR. Для модели SIDARTHE известна система дифференциальных уравнений, что позволяет провести верификацию результатов.

В модели SIDARTHE рассматриваются следующие компартменты:

- S – восприимчивый (незараженный);
- I – невыявленный бессимптомный или малосимптомный инфицированный;
- D – диагностированный бессимптомный инфицированный;
- A – невыявленный с симптоматическим инфицированием;
- R – диагностированный с симптоматическим инфицированием;
- T – диагностированный и госпитализированный с опасными для жизни симптомами;
- H – выздоровевший инфицированный;
- E – умерший инфицированный.

Данная компартментальная модель учитывает следующие процессы: заражение; диагностирование заболевания; появление симптомов у инфицированных; госпитализация; выздоровление; летальный исход.

Заражение восприимчивого индивидуума (S) в рассматриваемой модели может произойти в результате его взаимодействия с любым типом инфицированных индивидуумов, кроме госпитализиро-

ванных индивидуумов. В частности, инфицирование происходит с коэффициентом α после взаимодействия с индивидуумом из компартмента I , с коэффициентом β – с индивидуумом из компартмента D , с коэффициентом γ – с индивидуумом из компартмента A и с коэффициентом δ – с индивидуумом из компартмента R . Выявление инфицированного индивидуума происходит с коэффициентом ε из компартмента I и с коэффициентом θ из компартмента A .

В данной модели предусмотрен учет проявления симптомов болезни с коэффициентом ζ у невыявленных инфицированных I и с коэффициентом η у выявленных инфицированных (D). В данной модели предусмотрено проявление у индивидуумов из компартмента A с коэффициентом μ и из компартмента R с коэффициентом ν симптомов, угрожающих жизни, с последующей госпитализацией индивидуумов.

Кроме того, в рассматриваемой модели учитывается выздоровление инфицированных из всех компартментов: из компартмента I с коэффициентом λ , из A – с коэффициентом κ , из D – с коэффициентом ρ , из R – с коэффициентом ξ , из T – с коэффициентом σ . Наконец, в данной модели учитывается случай возможного летального исхода тяжело больных индивидуумов (T) с коэффициентом τ .

Модели SIDARTHE соответствует диаграмма, представленная на рис. 4.

Система дифференциальных уравнений рассматриваемой модели предложена в [13] и имеет вид:

$$\begin{aligned} \frac{dS}{dt} &= -S(\alpha I + \beta D + \gamma A + \delta R), \\ \frac{dI}{dt} &= S(\alpha I + \beta D + \gamma A + \delta R) - (\varepsilon + \zeta + \lambda)I, \\ \frac{dD}{dt} &= \varepsilon I - (\eta + \rho)D, \\ \frac{dA}{dt} &= \zeta I - (\theta + \mu + \kappa)A, \\ \frac{dR}{dt} &= \eta D + \theta A - (\nu + \xi)R, \\ \frac{dT}{dt} &= \mu A + \nu R - (\sigma + \tau)T, \\ \frac{dH}{dt} &= \lambda I + \rho D + \kappa A + \xi R + \sigma T, \\ \frac{dE}{dt} &= \tau T. \end{aligned} \quad (2)$$

Рассмотрим пример использования программного комплекса для формализации модели SIDARTHE. Формализация производится с исполь-

зованием графа, представленного на рис. 4, и с учетом описания процессов, происходящих в рассматриваемой системе. Для формализации необходимо задать количество компартментов ($n = 8$), символьные обозначения компартментов ['s' 'i' 'd' 'a' 'r' 't' 'h' 'e'] с помощью переменной `state_sidarthe`, список правил переходов между компартментами с помощью переменной `rules_sidarthe` и коэффициенты модели с помощью переменной `coefs_sidarthe`. Само построение модели производится с помощью функции

```
map((rule,coef) -> adder!(sidarthe,
  ↪ rule, state_sidarthe, coef),
  ↪ rules_sidarthe, coefs_sidarthe).
```

Программный код представлен в листинге 4.

```
sidarthe = compmodel(8)
```

```
@variables s i d a r t h e
matrix_sidarthe = [s i d a r t h e]
state_sidarthe = ['s' 'i' 'd' 'a' 'r' 't' 'h' 'e']
  ↪ ['t' 'h' 'e']
rules_sidarthe = ["s + i ~ i + i",
  "s + d ~ i + d", "s + a ~ i + a",
  "s + r ~ i + r",
  "i ~ a", "i ~ d", "i ~ h",
  "a ~ r", "a ~ t", "a ~ h",
  "d ~ r", "d ~ h",
  "r ~ t", "r ~ h",
  "t ~ e", "t ~ h"]
coefs_sidarthe = ['a', 'β', 'γ',
  ↪ 'δ', 'ζ', 'ε', 'λ', 'θ', 'μ',
  ↪ 'κ', 'η', 'ρ', 'ν', 'ξ', 'τ',
  ↪ 'σ']
```

```
map((rule,coef) -> adder!(sidarthe,
  ↪ rule, state_sidarthe, coef),
rules_sidarthe, coefs_sidarthe)
```

Листинг 4. Реализация модели.

Для получения вектора сносов можно использовать метод `drift_vector`. Вектор сносов для системы (2) имеет вид, представленный на рис. 5.

Таким образом, полученный вектор сносов с точностью до обозначений соответствует правой части детерминированной системы дифференциальных уравнений (2).

Кроме того, с помощью метода `diffusion_matrix` можно получить матрицу диффузии уравнения Фоккера–Планка, что, в свою очередь, позволяет запи-

```
1 drift_vector(matrix_sidarthe, sidarthe)
```

$$\begin{bmatrix} -a\gamma - d\beta - i\alpha - r\delta \\ a\gamma - i\epsilon + d\beta - i\zeta - i\lambda + i\alpha + r\delta \\ i\epsilon - d\eta - d\rho \\ i\zeta - a\theta - a\kappa - a\mu \\ a\theta + d\eta - r\nu - r\xi \\ a\mu + r\nu - t\sigma - t\tau \\ a\kappa + d\rho + i\lambda + r\xi + t\sigma \\ t\tau \end{bmatrix}$$

Рис. 5. Вектор сносов для модели SIDARTHE.

сать стохастическое дифференциальное уравнение в форме Ланжевена. Результат работы программного комплекса применительно к получению матрицы диффузии здесь не приводится ввиду достаточно громоздкого представления.

6. ОБСУЖДЕНИЕ

Таким образом, рассмотренный в разделе 5 пример демонстрирует возможности разработанного программного комплекса для построения компартментальных моделей с большим количеством переменных и с достаточно сложной структурой. Результаты конструирования модели SIDARTHE в точности соответствуют формальному описанию в виде системы дифференциальных уравнений (2).

Подход к построению правил взаимодействий в системе продемонстрирован в листинге 6. Мы предлагаем два типа правил, описывающих эволюцию системы, а именно, бинарные и небинарные правила. Конкретная формулировка правил для системы SIDARTHE содержится в переменной `rules_sidarthe`. Представление правил осуществлено в текстовом формате, что позволяет манипулировать содержанием базы правил с использованием штатных средств языка Julia. Разработанный программный комплекс продемонстрировал возможности удобного инструментария для конструирования компартментальных моделей на основе сущностного описания взаимодействий элементов.

Перспективой развития работы является реализация возможности численного расчета траекторий компартментальных систем для детерминистического и стохастического случая. Реализация такой возможности позволит выполнять комплекс исследований, связанных с качественным поведением систем. Кроме того, в дальнейшем целесообразно разработать модуль программного комплекса для усовершенствования визуализации результатов.

7. ЗАКЛЮЧЕНИЕ

В современных условиях быстрого изменения течения эпидемий и пандемий актуальной задачей является разработка программного обеспечения для оперативного и удобного получения таких новых математических моделей распространения заболеваний, которые учитывают дополнительные специфические факторы.

В настоящей работе представлены результаты разработки программного комплекса символьных вычислений для построения детерминированных и стохастических моделей динамических систем. Программный комплекс направлен на решение задачи формализации компартментальных моделей исходя из сущностного описания взаимодействия компартментов. Использование функциональности разработанного программного обеспечения позволяет избежать некорректностей технического характера при построении моделей высоких размерностей. Применение языка Julia дало возможность привлечь удобные и высокопроизводительные средства численных методов, математического моделирования и компьютерной алгебры.

Результаты могут найти применение при решении задач конструирования и исследования динамических моделей одношаговых процессов, в частности, характерных для биологических, экологических и социо-экономических систем.

СПИСОК ЛИТЕРАТУРЫ

- Кулябов Д.С. Аналитический обзор систем символьных вычислений // Вестник РУДН. Сер. Математика. Информатика. Физика. 2007. № 1–2. С. 38–45.
- Колесов Ю.Б., Сениченков Ю.Б. Компонентное моделирование сложных динамических систем. — СПб: Санкт-Петербургский политехнический университет Петра Великого, 2020.
- Банщиков А.В., Бурлакова Л.А., Иртегов В.Д., Титов Т.Н. Символьные вычисления в моделировании и качественном анализе динамических систем // Вычислительные технологии. 2014. № 6. С. 3–18.
- Banshchikov A., Vetrov A. Application of software tools for symbolic description and modeling of mechanical systems // CEUR Workshop Proceedings. 2. ser. "ICCSDE 2020 — Proceedings of the 2nd International Workshop on Information, Computation, and Control Systems for Distributed Environments". 2020. P. 33–42.
- Демидова А.В., Дружинина О.В., Масина О.Н., Петров А.А. Разработка алгоритмического и программного обеспечения моделирования управляемых динамических систем с применением символьных вычислений и стохастических методов // Программирование. 2023. № 2. С. 54–68.
- Кабанихин С.И., Криворотько О.И. Математическое моделирование эпидемии Уханьского коронавируса COVID-2019 и обратные задачи // Журнал вычислительной математики и математической физики. 2020. Т. 60. № 11. С. 1950–1961.
- Hamelin F., Iggidr A., Rapaport A., Sallet G. Observability, Identifiability and Epidemiology A survey. 2023. Access mode: <https://arxiv.org/abs/2011.12202>.
- Chebotaeva V., Vasquez P.A. Erlang-Distributed SEIR Epidemic Models with Cross-Diffusion // Mathematics. 2023. Vol. 11, no. 9. P. 2167. Access mode: <https://www.mdpi.com/2227-7390/11/9/2167>.
- Киселевская-Бабинина В.Я., Романюха А.А., Санникова Т.Е. Математическая модель течения Covid-19 и прогноз тяжести инфекции // Математическое моделирование. 2023. Т. 35. № 5. С. 31–46.
- Ghosh S., Volpert V., Banerjee M. An Epidemic Model with Time Delay Determined by the Disease Duration // Mathematics. 2022. Vol. 10, no. 15. P. 2561. Access mode: <https://www.mdpi.com/2227-7390/10/15/2561>.
- Ariffin M., Gopal K., Krishnarajah I., Cheilias I., Adam M., Arasan J., Rahman N., Dom N., Sham N. Mathematical epidemiologic and simulation modelling of first wave COVID-19 in Malaysia // Scientific Reports. 2021. Vol. 11. P. 20739.
- Roman H.E., Croccolo F. Spreading of Infections on Network Models: Percolation Clusters and Random Trees // Mathematics. 2021. Vol. 9, no. 23. P. 3054. Access mode: <https://www.mdpi.com/2227-7390/9/23/3054>.
- Giordano G., Blanchini F., Bruno R., Colaneri P., Filippa A., Di Matteo A., Colaneri M. Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy // Nature Medicine. 2020. Vol. 26. P. 855–860.
- Демидова А.В. Уравнения динамики популяций в форме стохастических дифференциальных уравнений // Вестник РУДН. Серия: Математика. Информатика. Физика. 2013. № 1. С. 67–76. Режим доступа: <https://journals.rudn.ru/miph/article/view/8319>.
- Gevorkyan M.N., Velieva T.R., Korolkova A.V., Kulyabov D.S., Sevastyanov L.A. Stochastic Runge–Kutta Software Package for Stochastic Differential Equations // Dependability Engineering and Complex Systems / ed. by Zamojski W., Mazurkiewicz J., Sugier J., Walkowiak T., and Kacprzyk J. — Cham : Springer International Publishing. — 2016. Vol. 470. P. 169–179.
- Gevorkyan M.N., Demidova A.V., Korolkova A.V., Kulyabov D.S. Issues in the Software Implementation of Stochastic Numerical Runge–Kutta // Distributed Computer and Communication Networks / ed. by Vishnevskiy V. M. and Kozyrev D. V. Cham : Springer International Publishing. 2018. Vol. 919. P. 532–546.

17. Геворкян М.Н., Демидова А.В., Велиева Т.Р., Королькова А.В., Кулябов Д.С., Севастьянов Л.А. Реализация метода стохастизации одношаговых процессов в системе компьютерной алгебры // Программирование. 2018. № 2. С. 18–27.
18. Gardiner C.W. Handbook of Stochastic Methods: For Physics, Chemistry and the Natural Sciences. Heidelberg: Springer, 1985.
19. Van Kampen N. Stochastic Processes in Physics and Chemistry. Amsterdam: Elsevier, 1992.
20. Bezanson J., Karpinski S., Shah V., Edelman A. Julia: A Fast Dynamic Language for Technical Computing. 2012. Access mode: <https://arxiv.org/abs/1209.5145>.
21. Gowda S., Ma Y., Cheli A., Gwozdz M., Shah V.B., Edelman A., Rackauckas C. High-Performance Symbolic-Numerics via Multiple Dispatch // ACM Commun. Comput. Algebra. 2022. jan. Vol. 55, no. 3. P. 92–96. Access mode: <https://doi.org/10.1145/3511528.3511535>.
22. Кулябов Д.С., Королькова А.В. Компьютерная алгебра на Julia // Программирование. 2021. № 2. С. 44–50.
23. Fedorov A.V., Masolova A.O., Korolkova A.V., Kulyabov D.S. Application of a numerical-analytical approach in the process of modeling differential equations in the Julia language // Journal of Physics: Conference Series. 2020. dec. Vol. 1694, no. 1. P. 012026. Access mode: <https://dx.doi.org/10.1088/1742-6596/1694/1/012026>.
24. Abotaleb M.S., Makarovskikh T. Analysis of Neural Network and Statistical Models Used for Forecasting of a Disease Infection Cases // 2021 International Conference on Information Technology and Nanotechnology (ITNT). 2021. P. 1–7.
25. Tuluri F., Remata R., Walters W.L., Tchounwou P.B. Application of Machine Learning to Study the Association between Environmental Factors and COVID-19 Cases in Mississippi, USA // Mathematics. 2022. Vol. 10, no. 6. P. 850. Access mode: <https://www.mdpi.com/2227-7390/10/6/850>.
26. Roman H.E., Crococolo F. Spreading of Infections on Network Models: Percolation Clusters and Random Trees // Mathematics. 2021. Vol. 9, no. 23. P. 3054. Access mode: <https://www.mdpi.com/2227-7390/9/23/3054>.
27. Романюха А.А. Математические модели в иммунологии и эпидемиологии инфекционных заболеваний. Москва : Бином. Лаборатория знаний, 2011.
28. Kermack W.O., McKendrick A.G. Contributions to the mathematical theory of epidemics // Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character. 1927. V. 115. P. 700–721.
29. Strauss R.R., Bishnu S., Petersen M.R. Comparing the Performance of Julia on CPUs versus GPUs and Julia-MPI versus Fortran-MPI: a case study with MPASOcean (Version 7.1) // EGUsphere. 2023. Vol. 2023. P. 1–22. Access mode: <https://egusphere.copernicus.org/preprints/2023/egusphere-2023-57/>.
30. Rackauckas C., Nie Q. DifferentialEquations.jl — A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia // Journal of Open Research Software. 2017.
31. Loman T.E., Ma Y., Ilin V., Gowda S., Korsbo N., Yewale N., Rackauckas C., Isaacson S.A. Catalyst: Fast Biochemical Modeling with Julia. 2022. Access mode: <https://www.biorxiv.org/content/early/2022/08/02/2022.07.30.502135>.
32. Angevaere J., Feng Z., and Deardon R. Pathogen.jl: Infectious Disease Transmission Network Modeling with Julia // Journal of Statistical Software. 2022. Vol. 104, no. 4. P. 1–30. Access mode: <https://www.jstatsoft.org/index.php/jss/article/view/v104i04>.
33. Апреутесей А.М.Ю., Королькова А.В., Кулябов Д.С. Возможности гибридного моделирования систем с управлением на языках Modelica и Julia // Распределенные компьютерные и телекоммуникационные сети: управление, вычисление, связь (DCCN–2020). Материалы XXIII Международной научной конференции. 2020. С. 433–440.
34. Апреутесей А.М.Ю., Королькова А.В., Кулябов Д.С. Hybrid modelling of the red algorithm in the Julia language // Journal of Physics: Conference Series. 7. Information Technology, Telecommunications and Control Systems, ITTCS 2020. 2020. P. 012025.

CONSTRUCTING COMPARTMENTAL MODELS OF DYNAMIC SYSTEMS USING A SOFTWARE PACKAGE FOR SYMBOLIC COMPUTATION IN JULIA

© 2024 A. V. Demidova^a, O. V. Druzhinina^b, O. N. Masina^c, A. A. Petrov^c

^aRUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russia

^bFederal Research Center "Computer Science and Control" of Russian Academy of Sciences 44-2, Vavilov St., Moscow, 119333 Russia

^cBunin Yelets State University, 28, Kommunarov St., Yelets, Lipetsk region, 399770 Russia

This paper considers the problem of constructing compartmental models of dynamic systems by using a software package for symbolic calculation written in Julia. The software package is aimed at unifying the formalized construction of compartmental models, taking into account the meaningful description of possible interactions among compartments and the influence of various factors on the evolution of systems. An approach to the development of the instrumental and methodological basis for modeling the dynamic systems the behavior of which can be described by one-step processes is developed. The proposed software package enables the symbolic representation of the differential equations of the model in both stochastic and deterministic cases. It is implemented in Julia and uses the Julia Symbolics computer algebra library. A comparison between the Julia Symbolics tools and some other computer algebra systems is carried out. The application of the developed software package to a compartmental model is considered. The results can be used to solve problems of constructing and studying dynamic models in natural sciences that are represented by onestep processes.

Keywords: compartmental models, dynamic systems, computer algebra, Julia programming language, symbolic computing software package

REFERENCES

1. Kulyabov D.S. Analytical overview of symbolic computation systems, Vestn. Ross. Univ. Druzhby Nar., Ser. Mat. Inf. Fiz. 2007. №. 1–2. P. 38–45.
2. Kolesov Yu.B., Senichenkov Yu.B. Komponentnoe modelirovanie slozhnykh dinamicheskikh sistem (Component Modeling of Complex Dynamic Systems), St. Petersburg: S.-Peterb. Politekh. Univ. Petra Velikogo, 2020.
3. Banshchikov A.V., Burlakova L.A., Irtegov V.D., Titorenko T.N. Symbolic computation in modeling and qualitative analysis of dynamic systems // Vychisl. Tekhnol. 2014. № 6. P. 3–18.
4. Banshchikov A., Vetro A. Application of software tools for symbolic description and modeling of mechanical systems, Proc. 2nd Int. Workshop Information, Computation, and Control Systems for Distributed Environments (ICCSDE). 2020. P. 33–42.
5. Demidova A.V., Druzhinina O.V., Masina O.N., Petrov A.A. Development of algorithms and software for modeling controlled dynamic systems using symbolic computations and stochastic methods // Program. Comput. Software. 2023. V. 49. P. 108–121.
6. Kabanikhin S.I., Krivorotko O.I. Mathematical modeling of the Wuhan COVID-2019 epidemic and inverse problems // Comput. Math. Math. Phys. 2020. V. 60. P. 1889–1899.
7. Hamelin F., Iggidr A., Rapaport A., Sallet G. Observability, identifiability, and epidemiology: A survey, 2023. <https://arxiv.org/abs/2011.12202>.
8. Chebotaeva V., Vasquez P.A. Erlang-distributed SEIR epidemic models with cross-diffusion, Mathematics. 2023. V. 11. № 9. P. 2167. <https://www.mdpi.com/2227-7390/11/9/2167>.
9. Kisselevskaya-Babinina V.Ya., Romanyukha A.A., Sannikova T.E. Mathematical model of COVID-19 progression: Prediction of severity and outcome, Math. Models Comput. Simul. 2023. V. 15. P. 987–998.
10. Ghosh S., Volpert V., Banerjee M. An epidemic model with time delay determined by the disease duration // Mathematics. 2022. V. 10. № 15. P. 2561. <https://www.mdpi.com/2227-7390/10/15/2561>.
11. Ariffin M., Gopal K., Krishnarajah I., Cheilias I., Adam M., Arasan J., Rahman N., Dom N., Sham N. Mathematical epidemiologic and simulation modeling of first wave COVID-19 in Malaysia // Sci. Rep. 2021. V. 11. P. 20739.
12. Roman H.E., Croccolo F. Spreading of infections on network models: Percolation clusters and random trees, Mathematics. 2021. V. 9. № 23. P. 3054. <https://www.mdpi.com/2227-7390/9/23/3054>.
13. Giordano G., Blanchini F., Bruno R., Colaneri P., Filippa A., Di Matteo A., Colaneri M. Modeling the COVID-19 epidemic and implementation of population-wide interventions in Italy // Nat. Med. 2020. V. 26. P. 855–860.
14. Demidova A.V. Equations of population dynamics in the form of stochastic differential equations // Vestn. Ross. Univ. Druzhby Nar., Ser. Mat. Inf. Fiz. 2013. № 1. P. 67–76. <https://journals.rudn.ru/miph/article/view/8319>.
15. Gevorkyan M.N., Velieva T.R., Korolkova A.V., Kulyabov D.S., Sevastyanov L.A. Stochastic Runge–Kutta software package for stochastic differential equations // Dependability Engineering and Complex Systems, Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., and Kacprzyk, J., Eds., Springer. 2016. V. 470. P. 169–179.

16. *Gevorkyan M.N., Demidova A.V., Korolkova A.V., Kulyabov D.S.* Issues in the software implementation of stochastic numerical Runge–Kutta // *Distributed Computer and Communication Networks*, Vishnevskiy, V.M. and Kozyrev, D., Eds., Springer. 2018. V. 919. P. 532–546.
17. *Gevorkyan M.N., Demidova A.V., Velieva T.R., Korolkova A.V., Kulyabov D.S., Sevastyanov L.A.* Implementing a method for stochastization of one-step processes in a computer algebra system, *Program. Comput. Software*. 2018. V. 44. P. 86–93.
18. *Gardiner C.W.* Handbook of Stochastic Methods: For Physics, Chemistry and the Natural Sciences, Heidelberg: Springer, 1985.
19. *Van Kampen N.* Stochastic Processes in Physics and Chemistry, Amsterdam: Elsevier, 1992.
20. *Bezanson J., Karpinski S., Shah, V., Edelman A.* Julia: A fast dynamic language for technical computing, 2012. <https://arxiv.org/abs/1209.5145>.
21. *Gowda S., Ma Y., Cheli A., Gwozdz M., Shah V.B., Edelman A., Rackauckas C.* High-performance symbolic-numeric via multiple dispatch // *ACM Commun. Comput. Algebra*. 2022. V. 55. № 3. P. 92–96. <https://doi.org/10.1145/3511528.3511535>
22. *Kulyabov D.S., Korolkova A.V.* Computer algebra in JULIA // *Program. Comput. Software*. 2021. Vol. 47. P. 133–138.
23. *Fedorov A.V., Masolova A.O., Korolkova A.V., Kulyabov D.S.* Application of a numerical-analytical approach in the process of modeling differential equations in the Julia language // *J. Phys.: Conf. Ser.* 2020. V. 1694. № 1. P. 012026. <https://doi.org/10.1088/1742-6596/1694/1/012026>.
24. *Abotaleb M.S., Makarovskikh T.* Analysis of neural network and statistical models used for forecasting of a disease infection cases // *Technol. Nanotechnol.* 2021. P. 1–7.
25. *Tuluri F., Remata R., Walters W. L., Tchounwou P.B.* Application of machine learning to study the association between environmental factors and COVID-19 cases in Mississippi, USA // *Mathematics*. 2022. V. 10. № 6. P. 850. <https://www.mdpi.com/2227-7390/10/6/850>.
26. *Roman H.E., Croccolo F.* Spreading of infections on network models: Percolation clusters and random trees // *Mathematics*. 2021. V. 9. № 23. P. 3054. <https://www.mdpi.com/2227-7390/9/23/3054>.
27. *Romanyukha A.A.* Matematicheskie modeli v immunologii i epidemiologii infektsionnykh zabolevaniy (Mathematical Models in Immunology and Epidemiology of Infectious Diseases), Moscow: Binom. Laboratoriya znaniy, 2011.
28. *Kermack W.O., McKendrick A.G.* Contributions to the mathematical theory of epidemics // *Proc. R. Soc. London, Ser. A*. 1927. V. 115. P. 700–721.
29. *Strauss R.R., Bishnu S., Petersen M.R.* Comparing the performance of Julia on CPUs versus GPUs and Julia-MPI versus Fortran-MPI: A case study with MPAS-Ocean (Version 7.1), *EGUsphere*. 2023. V. 2023. P. 1–22.
30. *Rackauckas C., Nie Q.* Differentialequations.jl: A performant and feature-rich ecosystem for solving differential equations in Julia, *J. Open Res. Software*, 2017.
31. *Loman T.E., Ma Y., Ilin V., Gowda S., Korsbo N., Yewale N., Rackauckas C., Isaacson S.A.* Catalyst: Fast biochemical modeling with Julia, 2022. <https://www.biorxiv.org/content/early/2022/08/02/2022.07.30.502135>.
32. *Angevaere J., Feng, Z., Deardon R.* Pathogen.jl: Infectious disease transmission network modeling with Julia // *J. Stat. Software*. 2022. V. 104. № 4. P. 1–30. <https://www.jstatsoft.org/index.php/jss/article/view/v104i04>.
33. *Apreatesei A.M.Yu., Korolkova A.V., Kulyabov D.S.* Capabilities of hybrid modeling of systems with control in Modelica and Julia languages // *Mater. XXIII Mezhdunar. Nauchn. Konf. “Raspredelemnnye komp’yuternye i telekommunikatsionnye seti: upravlenie, vychislenie, svyaz”* (Proc. 23rd Int. Sci. Conf. Distributed Computer and Communication Networks: Control, Computation, Communications). 2020. P. 433–440.
34. *Apreatesei A.M.Y., Korolkova A.V., Kulyabov D.S.* Hybrid modeling of the red algorithm in the Julia language // *J. Phys.: Conf. Ser.* 7. 2020. P. 012025.