



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине: «Методы математического моделирования сложных процессов
и систем»

Студент Косенков Александр Александрович

Группа РК6-12М

Тип задания Лабораторная работа №3

Тема лабораторной работы Разработка архитектуры
графоориентированного решателя

Студент

подпись, дата

Косенков А.А.

фамилия, и.о.

Преподаватель

подпись, дата

Соколов А.П.

фамилия, и.о.

Оценка _____

Москва, 2022 г.

Оглавление

Задание на лабораторную работу	3
Цель выполнения лабораторной работы.....	6
Выполненные задачи	6
1. Получение аналитического решения прямой задачи	7
2. Математическая постановка алгоритма решения прямой задачи.....	12
Метод Рунге-Кутты 4-го порядка.....	12
3. Построение графовой модели для решения прямой задачи	13
Описание графоориентированного подхода	13
Описание входных данных для решателя прямой задачи	15
Разработка графовой модели	16
4. Математическая постановка алгоритма решения обратной задачи	18
Метод Монте-Карло.....	18
5. Построение графовой модели для решения обратной задачи.....	19
Описание входных данных для решателя прямой задачи	19
Разработка графовой модели	20
6. Анализ разработанных графовых моделей	24
Заключение	25
Список использованных источников	25

Задание на лабораторную работу

Вариант 4

Задача 4 (разработка шаблонов файлов входных данных и графовых моделей программной реализации сложных вычислительных методов (СВМ)).

Разработать графовую модель реализации СВМ, позволяющего решать определённую задачу указанными методами согласно конкретному варианту далее.

Требования (базовая часть):

1. Представить разработанную графовую модель в файле в формате aDOT, включающую определение наименований и описаний (файл должен содержать комментарии): состояний данных $\{S_i\}_1^k$, функций-предикатов $\{p_j\}_1^n$, функций-обработчиков $\{f_k\}_1^m$ и функций-селекторов $\{h_l\}_1^p$.
2. Разработать текстовый файл в формате aINI, определяющий формат входных данных разрабатываемого графоориентированного решателя.
3. Включить в состав отчета соответствующие листинги разработанных определений графовой(ых) модели(ей) в формате aDOT и листинг формата входных данных на языке aINI. Все листинги должны быть снабжены описаниями и ссылками на них в тексте отчета.
4. Представить иллюстрацию разработанной графовой модели.
5. Разрабатываемая графовая модель должна обеспечивать масштабируемую программную реализацию указанного вариантом метода(ов). Обосновать, что построенная графовая модель действительно будет реализовывать масштабируемую программную реализацию требуемого вычислительного метода.

Требования (продвинутая часть):

1. Какие ограничения в части использования вашей графовой модели вы бы могли выделить?

Вариант 4.1 (построение графовой модели метода решения обратной задачи идентификации параметров СОДУ).

Пусть сформулирована прямая задача в форме ОДУ (в общем случае СОДУ) и достаточного количества начальных условий, необходимых для единственности решения:

$$L[y] = f(y, t), t \in R_+; \quad (1)$$

$$y : R_+ \rightarrow R^n;$$

$$f : R \times R_+ \rightarrow R^n;$$

$$y^{(p)}(t_0) = y_0^{(p)}, p = 1, 2, \dots, n,$$

где $L[y] = L[y, \mathbf{g}^1]$ – дифференциальный оператор, $f(y, t) = f(y, t, \mathbf{g}^2)$ – некоторая заданная функция, $\mathbf{g}^1, \mathbf{g}^2$ – векторы известных внутренних параметров модели, описываемой этим ОДУ (для общего случая СОДУ, при $n > 1$), p – порядок производной. Если не оговорено специально, то $n = 1$.

До выполнения задания определить значения параметров при $N = 4$ – номер по списку группы:

$$V = (N \bmod 4) + 1 = 1;$$

$$k = (N \bmod 2) + 1 = 1;$$

$$d = (N \bmod 4) + 1 = 1;$$

$$h = (N \bmod 2) + 1 = 1.$$

Для заданного варианта прямая задача сформулирована следующим образом:

$$\left\{ \begin{array}{l} L[y, \mathbf{g}^1] = f(y, t); \\ \mathbf{g}^1 = (g_0^1, g_1^1, g_2^1), \text{ где } g_i^1 = \text{const для } i = 0, \dots, 2; \\ L[y, \mathbf{g}^1] = g_2^1 \frac{d^2 y}{dt^2} + g_1^1 \frac{dy}{dt} + g_0^1; \\ f(y, t) = t; \\ y'(t_0) = y_0'; \\ y''(t_0) = y_0''; \end{array} \right. \quad (2)$$

где $y = y(t)$ – неизвестная функция, y_0', y_0'' – известные начальные условия.

Значения постоянных внутренних параметров g_i^1, g_j^2 , а также начальные условия $y_0^{(p)}$ являются известными входными данными в постановке прямой задачи.

Формулировка обратной задачи в рамках обозначений соответствующей формулировки прямой задачи (2) для заданного варианта:

$$\begin{cases} g_j^1 = [\underline{g}_j^1; \overline{g}_j^1] \text{ для } j = 0, \dots, 2; \\ \mathbf{g}^{1*} = \operatorname{argmin}_{\mathbf{g}^1} \|\hat{y}(t, \mathbf{g}^1) - y(t, \mathbf{g}^1)\|; \\ y'(t_0) = y'_0; \\ y''(t_0) = y''_0; \end{cases} \quad (3)$$

где $g_j^1 = [\underline{g}_j^1; \overline{g}_j^1]$; $y(t, \mathbf{g}^1)$ – решение прямой задачи (1) при конкретных \mathbf{g}^1 и не изменяющихся и известных \mathbf{g}^2 , получаемое методом Рунге-Кутты 4-го порядка; $\hat{y}(t, \mathbf{g}^1)$ – известное решение прямой задачи (например, получено экспериментально или каким-то иным методом) при известных $\hat{\mathbf{g}}^1$ и не изменяющихся и также известных \mathbf{g}^2 .

Разрабатываемая графовая модель должна обеспечивать решение прямой задачи путём реализации численного метода интегрирования СОДУ Рунге-Кутты, а также решение обратной задачи путём реализации численного метода глобальной многомерной оптимизации Монте-Карло.

Дополнительно к общим требованиям задачи 4 требуется (**базовая часть**):

1. Построить графовую модель программной реализации вычислительного метода решения обратной задачи (2).
2. Определить файл входных данных обратной задачи в формате aINI.

Дополнительно к общим требованиям задачи 4 требуется (**продвинутая часть**):

1. Аналитически решить прямую задачу (1) и представить общий вид решения $y(t)$ (опционально).
2. Представить ответы на вопросы о том, как изменится ваша графовая модель (как для решения прямой, так и для решения обратной задачи) в

случае, если будет требоваться реализовать синтаксический анализатор, который должен будет обеспечивать:

- а) ввод произвольной правой части $f(y, t)$ с клавиатуры;
- б) ввод произвольного дифференциального оператора $L[y]$ с клавиатуры

Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – изучить принципы графоориентированного подхода к программной реализации сложных вычислительных методов на примере решения прямой и обратной задач, сформулированных в форме ОДУ.

Выполненные задачи

1. Получение аналитического решения прямой задачи.
2. Математическая постановка алгоритма решения прямой задачи.
3. Построение графовой модели прямой задачи.
4. Математическая постановка алгоритма решения обратной задачи.
5. Построение графовой модели обратной задачи.
6. Анализ разработанных графовых моделей.

1. Получение аналитического решения прямой задачи

Из (2) следует задача Коши, решение которого является искомым для решения прямой задачи:

$$\begin{cases} g_2^1 \frac{dy^2}{dt^2} + g_1^1 \frac{dy}{dt} = t - g_0^1; \\ y'(t_0) = y_0'; \\ y''(t_0) = y_0''. \end{cases} \quad (4)$$

Данное ДУ является неоднородным линейным ДУ 2-го порядка с постоянными коэффициентами. Решение ДУ данного типа представляет собой сумму общего решения однородного линейного ДУ и частного решения неоднородного.

Так, решим однородное линейное ДУ

$$g_2^1 \frac{dy^2}{dt^2} + g_1^1 \frac{dy}{dt} = 0, \quad (5.1)$$

для чего составим характеристическое уравнение:

$$\begin{aligned} g_2^1 \lambda^2 + g_1^1 \lambda &= 0 \\ \lambda(g_2^1 \lambda + g_1^1) &= 0. \end{aligned} \quad (5.2)$$

Корнями квадратного уравнения (5.2) являются:

$$\begin{cases} \lambda_1 = 0; \\ \lambda_2 = -\frac{g_1^1}{g_2^1}. \end{cases} \quad (5.3)$$

Общее решение ОЛДУ 2-го порядка (5.1) имеет вид:

$$y_{0.0}(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t} = C_1 + \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t}}. \quad (5.4)$$

Для проверки корректности составления общего решения (5.4) возможна его подстановка в однородное ЛДУ (5.1):

$$y'_{0.0}(t) = -\frac{g_1^1}{g_2^1} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t}};$$

$$y''_{0.0}(t) = \frac{g_1^{1^2}}{g_2^{1^2}} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t}};$$

$$g_2^1 \frac{dy^2}{dt^2} + g_1^1 \frac{dy}{dt} = g_2^1 \frac{g_1^{1^2}}{g_2^{1^2}} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t}} - g_1^1 \frac{g_1^1}{g_2^1} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t}} = \frac{g_1^{1^2}}{g_2^1} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t}} - \frac{g_1^{1^2}}{g_2^1} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t}} = 0.$$

Частное решение неоднородного ДУ может быть получено с помощью метода вариации постоянной, который подразумевает следующую замену:

$$y_{0.0}(t) = C_1 + \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t}} = C_1 Y_1(t) + C_2 Y_2(t) = C_1(t) Y_1(t) + C_2(t) Y_2(t) = y_{0.н};$$

и последующее решение системы:

$$\begin{cases} C_1'(t) Y_1(t) + C_2'(t) Y_2(t) = 0; \\ a_0(C_1'(t) Y_1'(t) + C_2'(t) Y_2'(t)) = f(y, t), \end{cases} \quad (6)$$

где a_0 – коэффициент при старшей производной ЛДУ. Тогда:

$$\begin{cases} C_1'(t) + C_2'(t) \frac{1}{e^{\frac{g_1^1}{g_2^1} t}} = 0; \\ C_2'(t) \left(-\frac{g_1^1}{g_2^1} \frac{1}{e^{\frac{g_1^1}{g_2^1} t}} \right) = \frac{(t - g_0^1)}{g_2^1}, \end{cases} \quad (7)$$

откуда:

$$C_1(t) = \frac{1}{g_1^1} \left(\int t dt - g_0^1 \int dt \right) = \frac{1}{g_1^1} \left(\frac{t^2}{2} - g_0^1 t \right) + C_1, \quad (8.1)$$

$$C_2(t) = \frac{1}{g_1^1} \left(g_0^1 \int e^{\frac{g_1^1 t}{g_2^1}} dt - \int t e^{\frac{g_1^1 t}{g_2^1}} dt \right) = \frac{1}{g_1^1} \left(g_0^1 \frac{g_2^1}{g_1^1} e^{\frac{g_1^1 t}{g_2^1}} - \int t e^{\frac{g_1^1 t}{g_2^1}} dt \right) + C_2. \quad (8.2)$$

Интеграл $\int t e^{\frac{g_1^1 t}{g_2^1}} dt$ раскрывается как интеграл по частям по формуле $\int u dv = uv - \int v du$:

$$\begin{aligned} \int t e^{\frac{g_1^1 t}{g_2^1}} dt &= \left| \begin{array}{ll} u = t & v = \int e^{\frac{g_1^1 t}{g_2^1}} dt = \frac{g_2^1}{g_1^1} e^{\frac{g_1^1 t}{g_2^1}} \\ dv = e^{\frac{g_1^1 t}{g_2^1}} dt & du = 1 dt \end{array} \right| = \frac{g_2^1}{g_1^1} t e^{\frac{g_1^1 t}{g_2^1}} - \frac{g_2^1}{g_1^1} \int e^{\frac{g_1^1 t}{g_2^1}} dt \\ &= \frac{g_2^1}{g_1^1} t e^{\frac{g_1^1 t}{g_2^1}} - \frac{g_2^{1^2}}{g_1^{1^2}} e^{\frac{g_1^1 t}{g_2^1}} + C. \end{aligned}$$

Тогда выражение (8.2) преобразуется в:

$$C_2(t) = \frac{1}{g_1^1} \left(g_0^1 \frac{g_2^1}{g_1^1} e^{\frac{g_1^1 t}{g_2^1}} - \frac{g_2^1}{g_1^1} t e^{\frac{g_1^1 t}{g_2^1}} + \frac{g_2^{1^2}}{g_1^{1^2}} e^{\frac{g_1^1 t}{g_2^1}} \right) + C_2 = \frac{g_2^1}{g_1^{1^2}} e^{\frac{g_1^1 t}{g_2^1}} \left(g_0^1 - t + \frac{g_2^1}{g_1^1} \right) + C_2. \quad (8.3)$$

$$y_{0.н} = C_1(t) + C_2(t) \frac{1}{e^{\frac{g_1^1 t}{g_2^1}}} = \frac{1}{g_1^1} \left(\frac{t^2}{2} - g_0^1 t \right) + C_1 + \frac{g_2^1}{g_1^{1^2}} \left(g_0^1 - t + \frac{g_2^1}{g_1^1} \right) + C_2 \frac{1}{e^{\frac{g_1^1 t}{g_2^1}}}. \quad (9)$$

Для проверки корректности полученного решения (9) достаточно подставить его в НЛДУ (4):

$$y'_{0.н}(t) = \frac{1}{g_1^1} t - \frac{g_0^1}{g_1^1} - \frac{g_2^1}{g_1^{1^2}} - \frac{g_1^1}{g_2^1} \frac{C_2}{e^{\frac{g_1^1 t}{g_2^1}}};$$

$$y''_{0.н}(t) = \frac{1}{g_1^1} + \frac{g_1^{1^2}}{g_2^{1^2}} \frac{C_2}{e^{\frac{g_1^1 t}{g_2^1}}};$$

$$g_2^1 \frac{dy^2}{dt^2} + g_1^1 \frac{dy}{dt} = \frac{g_2^1}{g_1^1} + \frac{g_1^{1^2}}{g_2^1} \frac{C_2}{e^{\frac{g_1^1 t}{g_2^1}}} + t - g_0^1 - \frac{g_2^1}{g_1^1} - \frac{g_1^{1^2}}{g_2^1} \frac{C_2}{e^{\frac{g_1^1 t}{g_2^1}}} = t - g_0^1;$$

$$t - g_0^1 = t - g_0^1.$$

После получения общего решения неоднородного ЛДУ (9) необходимо учесть начальные условия и решить задачу Коши.

$$\begin{aligned} & \begin{cases} y'(t_0) = y'_0; \\ y''(t_0) = y''_0, \end{cases} \\ & \begin{cases} \frac{1}{g_1^1} t_0 - \frac{g_0^1}{g_1^1} - \frac{g_2^1}{g_1^{1^2}} - \frac{g_1^1}{g_2^1} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t_0}} = y'_0; \\ \frac{1}{g_1^1} + \frac{g_1^{1^2}}{g_2^{1^2}} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t_0}} = y''_0, \end{cases} \\ & \begin{cases} C_2 = -\frac{g_2^1}{g_1^1} e^{\frac{g_1^1}{g_2^1} t_0} \left(y'_0 - \frac{1}{g_1^1} t_0 + \frac{g_0^1}{g_1^1} + \frac{g_2^1}{g_1^{1^2}} \right); \\ C_2 = \frac{g_2^{1^2}}{g_1^{1^2}} e^{\frac{g_1^1}{g_2^1} t_0} \left(y''_0 - \frac{1}{g_1^1} \right), \end{cases} \end{aligned} \quad (10)$$

Решение задачи Коши подразумевает нахождение частного решения из общего путем уточнения неизвестных констант C_i , однако в случае представленных начальных условий система (10) является переопределенной и может быть разрешена только при условии равенства правых частей, откуда:

$$\begin{aligned} & -\frac{g_2^1}{g_1^1} e^{\frac{g_1^1}{g_2^1} t_0} \left(y'_0 - \frac{1}{g_1^1} t_0 + \frac{g_0^1}{g_1^1} + \frac{g_2^1}{g_1^{1^2}} \right) = \frac{g_2^{1^2}}{g_1^{1^2}} e^{\frac{g_1^1}{g_2^1} t_0} \left(y''_0 - \frac{1}{g_1^1} \right), \\ & y'_0 - \frac{1}{g_1^1} t_0 + \frac{g_0^1}{g_1^1} + \frac{g_2^1}{g_1^{1^2}} = \frac{g_2^1}{g_1^{1^2}} - \frac{g_2^1}{g_1^1} y''_0, \\ & y''_0 = -\frac{g_1^1}{g_2^1} y'_0 + \frac{1}{g_2^1} t_0 - \frac{g_0^1}{g_2^1}. \end{aligned} \quad (11)$$

При этом задача (4) все еще не содержит единственного решения, поскольку константа C_1 остается неопределенной в виду ее редуцирования в процессе взятия производной для удовлетворения начальных условий и отсутствия начального условия для $y(t_0)$. А условие $y''(t_0) = y''_0$ является избыточным при учете (11).

Для достижения единственности решения, а также для корректного использования дальнейших численных методов, было принято решение изменить начальные условия на следующие:

$$\begin{cases} y(t_0) = y_0; \\ y'(t_0) = y'_0, \end{cases} \quad (12)$$

Откуда решение задачи Коши при начальных условиях (12) принимает вид:

$$\begin{cases} \left\{ \begin{aligned} &\frac{1}{g_1^1} \left(\frac{t_0^2}{2} - g_0^1 t_0 \right) + C_1 + \frac{g_2^1}{g_1^{1^2}} \left(g_0^1 - t_0 + \frac{g_2^1}{g_1^1} \right) + C_2 \frac{1}{e^{\frac{g_1^1}{g_2^1} t_0}} = y_0; \\ &\frac{1}{g_1^1} t_0 - \frac{g_0^1}{g_1^1} - \frac{g_2^1}{g_1^{1^2}} - \frac{g_1^1}{g_2^1} \frac{C_2}{e^{\frac{g_1^1}{g_2^1} t_0}} = y'_0, \end{aligned} \right. \\ \\ \left\{ \begin{aligned} &C_1 = y_0 + \frac{g_2^1}{g_1^1} \left(y'_0 - \frac{1}{g_1^1} t_0 + \frac{g_0^1}{g_1^1} + \frac{g_2^1}{g_1^{1^2}} \right) - \frac{g_2^1}{g_1^{1^2}} \left(g_0^1 - t_0 + \frac{g_2^1}{g_1^1} \right) - \frac{1}{g_1^1} \left(\frac{t_0^2}{2} - g_0^1 t_0 \right); \\ &C_2 = -\frac{g_2^1}{g_1^1} e^{\frac{g_1^1}{g_2^1} t_0} \left(y'_0 - \frac{1}{g_1^1} t_0 + \frac{g_0^1}{g_1^1} + \frac{g_2^1}{g_1^{1^2}} \right), \end{aligned} \right. \\ \\ \left\{ \begin{aligned} &C_1 = y_0 + \frac{g_2^1}{g_1^1} y'_0 - \frac{1}{g_1^1} \left(\frac{t_0^2}{2} - g_0^1 t_0 \right); \\ &C_2 = -\frac{g_2^1}{g_1^1} e^{\frac{g_1^1}{g_2^1} t_0} \left(y'_0 - \frac{1}{g_1^1} t_0 + \frac{g_0^1}{g_1^1} + \frac{g_2^1}{g_1^{1^2}} \right), \end{aligned} \right. \end{cases} \quad (13)$$

$$\begin{aligned} y(t) = & \frac{1}{g_1^1} \left(\frac{t^2}{2} - g_0^1 t \right) + \frac{g_2^1}{g_1^{1^2}} \left(g_0^1 - t + \frac{g_2^1}{g_1^1} \right) \\ & - \frac{g_2^1}{g_1^1} e^{\frac{g_1^1}{g_2^1} t_0} \left(y'_0 - \frac{1}{g_1^1} t_0 + \frac{g_0^1}{g_1^1} + \frac{g_2^1}{g_1^{1^2}} \right) \frac{1}{e^{\frac{g_1^1}{g_2^1} t}} + y_0 + \frac{g_2^1}{g_1^1} y'_0 \\ & - \frac{1}{g_1^1} \left(\frac{t_0^2}{2} - g_0^1 t_0 \right). \end{aligned} \quad (14)$$

Выражение (14) является аналитическим решением прямой задачи (2) при начальных условиях (12).

2. Математическая постановка алгоритма решения прямой задачи

Метод Рунге-Кутты 4-го порядка

Метод Рунге-Кутты 4-го порядка является численным методом решения ОДУ и основан на аппроксимации ряда Тэйлора [2] и имеет следующую формулировку:

$$\begin{aligned}\omega_i &\approx y(t_i), & \omega_0 &= y_0; \\ \omega_{i+1} &= \omega_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), & i &= 1, 2, \dots, m-1,\end{aligned}\quad (15)$$

где:

$$\begin{aligned}k_1 &= hf(t_i, \omega_i), \\ k_2 &= hf\left(t_i + \frac{h}{2}, \omega_i + \frac{1}{2}k_1\right), \\ k_3 &= hf\left(t_i + \frac{h}{2}, \omega_i + \frac{1}{2}k_2\right), \\ k_4 &= hf(t_i + h, \omega_i + k_3),\end{aligned}\quad (16)$$

где h - шаг сетки интегрирования.

Однако, для использования численного метода необходимо предварительно свести исходное ДУ к нормальной системе ДУ 1-го порядка:

$$\left\{\begin{array}{l}y_1(t) = y(t); \\ y_1'(t) = f_1(t, y_1, y_2) = y_2; \\ y_2' = f_2(t, y_1, y_2) = \frac{t - g_0^1 - g_1^1 y_2}{g_2^1}; \\ y_1(t_0) = y_0; \\ y_2(t_0) = y_0',\end{array}\right.\quad (17)$$

В таком случае схема решения прямой задачи (2) методом Рунге-Кутты 4-го порядка будет выглядеть следующим образом при:

$$\begin{aligned}\omega_i^1 &\approx y_1(t_i), & \omega_i^2 &\approx y_2(t_i); \\ \omega_0^1 &= y_0, & \omega_0^2 &= y_0';\end{aligned}$$

$$\begin{aligned}\omega_{i+1}^1 &= \omega_i^1 + \frac{1}{6}(k_1^1 + 2k_2^1 + 2k_3^1 + k_4^1), & i = 1, 2, \dots, m-1; \\ \omega_{i+1}^2 &= \omega_i^2 + \frac{1}{6}(k_1^2 + 2k_2^2 + 2k_3^2 + k_4^2), & i = 1, 2, \dots, m-1,\end{aligned}\tag{18.1}$$

где:

$$\begin{aligned}k_1^j &= hf_j(t_i, \omega_i^1, \omega_i^2), \\ k_2^j &= hf_j\left(t_i + \frac{h}{2}, \omega_i^1 + \frac{1}{2}k_1^1, \omega_i^2 + \frac{1}{2}k_1^2\right), \\ k_3^j &= hf_j\left(t_i + \frac{h}{2}, \omega_i^1 + \frac{1}{2}k_2^1, \omega_i^2 + \frac{1}{2}k_2^2\right), \\ k_4^j &= hf_j(t_i + h, \omega_i^1 + k_3^1, \omega_i^2 + k_3^2), \\ &j = 1, 2.\end{aligned}\tag{18.2}$$

Таким образом получение решения прямой задачи подразумевает собой итеративное вычисление точек ω_i^1 , результатом которого является дискретный набор пар точек $\langle t_j, \omega_j^1 \rangle, j = 0, 1, \dots, m$, где m – количество точек сетки по оси t .

3. Построение графовой модели для решения прямой задачи

Описание графоориентированного подхода

Графоориентированный подход для реализации сложных вычислительных методов (СВМ) был разработан с целью создания набора инструментов для организации программного кода, позволяющего уменьшить трудозатраты на его дальнейшую верификацию, валидацию и поддержку [3].

Сложным вычислительным методом (СВМ) называется множество вычислительных методов, взаимосвязанных друг с другом по входным и выходным данным.

Алгебраическим фундаментом графоориентированного подхода является категория состояний СВМ с копроизведением. Графовая модель, в свою очередь, представляет собой граф переходов между состояниями S_i с помощью морфизмов (функций-обработчиков) f_{ij} (граф потока управления [3]), таким образом описывая внутреннюю и внешнюю взаимосвязь данных СВМ.

Для гибкости архитектуры СВМ для перехода между состояниями S_i и S_j удобно использовать функции перехода $\langle p_{ij}, f_{ij} \rangle$, которые выполняют сценарий f в случае удовлетворения входным данным в соответствии с предикатом p или создают исключительную ситуацию в ином случае ($p = 0$), где p_{ij} - функция-предикат, возвращающий 0 или 1 в зависимости от входных данных, f_{ij} - функция-обработчик, осуществляющий переход от состояния S_i к состоянию S_j . Для подобного графа узлами являются состояния СВМ S_i , в то время как ребрами являются функции перехода $\langle p, f \rangle$.

Многие вычислительные процедуры предполагают три основных этапа вычислений: препроцессор, процессор и постпроцессор. Препроцессная часть подразумевает предварительную подготовку модели (загрузка геометрии, наложение краевых условий, преобразование модели к нужному виду для численного метода и т.д.). Процессор подразумевает собой реализацию конкретного алгоритма. Постпроцессная часть – обработка результата (например, вывод только конкретных степеней свободы результата с целью уменьшения размерностей файлов).

Помимо существенного объема работ по написанию исходных кодов, требования обеспечения масштабируемости и пр., важной проблемой является и существенный объем исходных данных. Для решения этой проблемы вводится формат входных данных СВМ, представляющий собой массив, состоящий из пар ключ-значение, где ключом выступает название переменной, а значением выступает ее тип (целочисленный, вещественный, строковый и т.д.). По данному формату составляются описания непосредственных начальных данных, где переменные имеют конкретные значения, соответствующие их типу, описанному в формате.

В качестве языка описания графовых моделей был использован язык описания данных aDOT [4]. В свою очередь, описание начальных состояний систем (формат входных данных СВМ) осуществлено с помощью формата описания данных aINI [5]. При этом, состояние S_1 в разработанных далее

моделях является отображением файла входных данных формата aINI, благодаря чему предварительная загрузка данных и инициализация модели не требуется.

Сама разработка графовых моделей подразумевает:

- предварительную подготовку инициализационных данных в формате aINI;
- выделение состояний и функций перехода между состояниями описываемого CBM;
- формализацию графа состояний на языке aDOT.

Описание входных данных для решателя прямой задачи

Для прямой задачи входными данными являются:

- непосредственно параметры системы – вектор \mathbf{g}^1 и начальные условия y'_0 и y''_0 ;
- параметры численного метода: время начала и конца, шаг интегрирования;
- имя файла с результатами.

Входные данные для прямой задачи, записанные на языке aINI, приведены в Листинге 1. Параметры разделены по секциям в соответствии со списком выше.

Кроме того, Листинг 2 содержит формат входных данных CBM с информацией о типах приведенных параметров.

Листинг 1. Файл forward-solver.aini – входные данные решателя прямой задачи в формате aINI

```
[PARAMS]

g1_0=3.2
g1_1=2.1
g1_2=1.3

d1y_0=1
d2y_0=1

[TIME]

t_start=0.0
t_end=10.0
t_step=0.1

[DATA]

output=forward-task.csv
```

```

[PARAMS]

g1_0=@double@
g1_1=@double@
g1_2=@double@

d1y_0=@double@
d2y_0=@double@

[TIME]

t_start=@double@
t_end=@double@
t_step=@double@

[DATA]

output=@filename@

```

Разработка графовой модели

Поскольку решение прямой задачи подразумевает собой однократное применение численного метода Рунге-Кутты, данный случай является тривиальным и представляет собой линейную структуру графа со следующими состояниями:

- S_1^F – начальное состояние системы с проинициализированными параметрами в соответствии с входными данными;
- S_2^F – получено решение прямой задачи с помощью метода Рунге-Кутты 4-го порядка;
- S_3^F – осуществлена стадия постпроцессинга (обработаны результаты решения прямой задачи)

Морфизмы, выраженные в функциях перехода $\langle p_{ij}, f_{ij} \rangle$ представлены на рисунке графовой модели (рис. 1) в виде их фактических имен в формальной записи графа в формате aDOT (Листинг 3) для удобочитаемости. Таким образом, была составлена таблица описания функций-переходов:

Таблица 1. Описание морфизмов графовой модели прямой задачи

Функция перехода	Фактическое имя	Описание

$\langle p_{12}, f_{12} \rangle$	$\langle \text{PRED_F_PROC}, \text{HANDLE_F_PROC} \rangle$	Решение прямой задачи (4) с помощью метода Рунге-Кутты 4-го порядка
$\langle p_{23}, f_{23} \rangle$	$\langle \text{PRED_F_POSTPROC}, \text{HANDLE_F_POSTPROC} \rangle$	Сохранение результатов в файл формата .csv

В соответствии со списком состояний и морфизмов (Таблица 1) была построена графовая модель решателя прямой задачи (4):

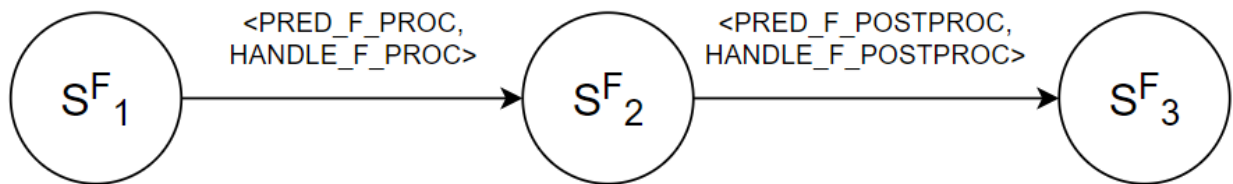


Рис. 1. Графовая модель решателя прямой задачи.

Данная графовая модель была описана с помощью формата aDOT в соответствии с рис.1:

Листинг 3. Файл forward_solver.adot – описание графовой модели решателя прямой задачи в формате aDOT

```

digraph forward_solver_graph {
    // Функции-предикаты
    PRED_F_PROC [module="forward_solver", entry_func="pred_f_proc"]
    PRED_F_POSTPROC [module="forward_solver", entry_func="pred_f_postproc"]

    // Функции-обработчики
    HANDLE_F_PROC [module="forward_solver", entry_func="handle_f_proc"]
    HANDLE_F_POSTPROC [module="forward_solver",
entry_func="handle_f_postproc"]

    // Функции перехода
    TRANS_F_PROC [predicate=PRED_F_PROC, function=HANDLE_F_PROC]
    TRANS_F_POSTPROC [predicate=PRED_F_POSTPROC, function=HANDLE_F_POSTPROC]

    // Описание графа
    BEGIN --> SF_1
    SF_1 --> SF_2 [morphism=TRANS_F_PROC]
    SF_2 --> SF_3 [morphism=TRANS_F_POSTPROC]
    SF_3 --> END
}

```

Таким образом Листинг 3 является формальным описанием графоориентированного решателя прямой задачи.

4. Математическая постановка алгоритма решения обратной задачи

Метод Монте-Карло

Метод Монте-Карло относится к классу прямых стохастических методов поиска и предназначен для решения многомерной задачи глобальной условной оптимизации, в которой область допустимых значений определяется как ограничениями типа равенств, так и ограничениями типа неравенств [6].

Для многомерной задачи глобальной условной оптимизации:

$$\min_{X \in D \subset R^n} \Phi(X) = \Phi(X^*) = \Phi^*,$$

где множество допустимых значений

$$D = \{X \mid h_i(X) = 0, g_j(X) \geq 0, i \in [1, m], j \in [1, l]\}$$

определяется как ограничениями типа неравенств, так и ограничениями типа равенств.

Схема метода представляет собой следующий алгоритм:

- 1 Задание общего количества испытаний N и установка счетчика числа итераций $r = 1$;
- 2 Генерация n компонент вектора $X^1 \in D$ с помощью какого-либо программного генератора случайных чисел;
- 3 Вычисление $\Phi(X^1)$, $X^* = X^1$, $\Phi^* = \Phi(X^1)$, $r = r + 1$;
- 4 Генерация случайной точки $X^r \in D$ аналогично п.2. Вычисление соответствующего критерия оптимальности $\Phi(X^r) = \Phi^*$;
- 5 Выполнение присваивания:

$$X^* = \begin{cases} X^r, & \text{если } \Phi^r < \Phi^*; \\ X^*, & \text{иначе;} \end{cases}$$

- 6 Если $r < N$, то $r = r + 1$ и переход на п.4, иначе X^* , Φ^* принимаются в качестве приближенного решения задачи и вычисления заканчиваются.

В простейшем случае точки $X^r \in D$ генерируются равномерно распределенными в области D . С целью сокращения вычислительных затрат и при наличии априорной информации о положении точки глобального минимума, целесообразно использовать законы распределения, в которых вероятность генерации точки в окрестности предполагаемого глобального минимума выше, чем вне этой окрестности.

5. Построение графовой модели для решения обратной задачи

Описание входных данных для решателя прямой задачи

Для обратной задачи входными данными являются:

- диапазоны значений вектора \mathbf{g}^1 ;
- источник с эталонными данными;
- количество итераций метода;
- параметры системы из прямой задачи (начальные условия, время интегрирования)
- имя файла с результатами.

Входные данные для обратной задачи, записанные на языке aINI, приведены в Листинге 4. Параметры разделены по секциям в соответствии со списком выше.

Кроме того, Листинг 5 содержит формат входных данных CBM с информацией о типах приведенных параметров.

Листинг 4. Файл backward-solver.ini – входные данные решателя обратной задачи в формате aINI.

```
[INCLUDE]
AnotherFile=forward-solver.ini

[PARAMS]

g1_0=[3.0;1.0:10.0;0.1]
g1_1=[2.0;1.0:5.0;0.1]
```

```

g1_2=[1.0;0.5:5.0;0.1]

[BACKWARD_METHOD]

iterations=1000

[DATA]

reference_values=forward-task.csv
output=backward-task.csv

```

Листинг 5. Формат входных данных решателя прямой задачи.

```

[INCLUDE]
AnotherFile=@filename@

[PARAMS]

g1_0=@range@
g1_1=@range@
g1_2=@range@

[BACKWARD_METHOD]

iterations=@int@

[DATA]

reference_values=@filename@
output=@filename@

```

Поскольку формат aINI позволяет подключать внешние файлы входных данных — данные о времени интегрирования и начальных условиях подключаются из входных данных прямой задачи для гибкости.

Разработка графовой модели

Решатель обратной задачи включает в себя расчет случайных значений вектора \mathbf{g}^1 , решение прямой задачи, расчет целевой функции оптимизации. Таким образом, данный СВМ можно описать следующими состояниями:

- S_1^B — начальное состояние с проинициализированными параметрами системы и метода в соответствии с входными данными, загружены данные об эталонном решении $\hat{y}(t, \hat{\mathbf{g}}^1)$;
- S_2^B — получен новый вектор случайных значений \mathbf{g}^1 ;
- S_3^B — получено решение прямой задачи для конкретного \mathbf{g}^1 ;
- S_4^B — произведен расчет целевой функции оптимизации;

- S_5^B – выбраны оптимальные значения \mathbf{g}^1 и получено очередное приближенное решение обратной задачи;
- S_6^B – осуществлена стадия постпроцессинга (обработаны результаты решения обратной задачи)

Морфизмы, выраженные в функциях перехода $\langle p_{ij}, f_{ij} \rangle$ представлены на рисунке графовой модели (рис. 2) в виде их фактических имен в формальной записи графа в формате aDOT (Листинг 6) для удобочитаемости аналогично прямой задаче. Таким образом, была составлена таблица описания функций-переходов:

Таблица 2. Описание морфизмов графовой модели обратной задачи

Функция перехода	Фактическое имя	Описание
$\langle p_{12}, f_{12} \rangle$	<code><PRED_B_GENERATE_G1, HANDLE_B_GENERATE_G1></code>	Генерация нового вектора \mathbf{g}^1 на основе случайных значений
$\langle p_{23}, f_{23} \rangle$ $\langle p_{52}, f_{52} \rangle$	<code><PRED_B_PROC_FORWARD, HANDLE_B_PROC_FORWARD ></code>	Решение прямой задачи (4) на основе полученного ранее вектора \mathbf{g}^1
$\langle p_{34}, f_{34} \rangle$	<code><PRED_B_TARGET_FUNC, HANDLE_B_TARGET_FUNC></code>	Расчет значения целевой функции оптимизации
$\langle p_{45}, f_{45} \rangle$	<code><PRED_B_G1_ASSIGN, HANDLE_B_G1_ASSIGN></code>	Определение оптимального \mathbf{g}^1 на текущей итерации метода
$\langle p_{56}, f_{56} \rangle$	<code><PRED_B_POSTPROC, HANDLE_B_POSTPROC></code>	Сохранение результатов в формате .csv

Решение прямой задачи внедрено в виде подграфа ранее реализованной графовой модели прямой задачи. Заикливание осуществлено с помощью функции-селектора [3] CHECK_ITER, осуществляющей проверку счетчика итераций метода.

В соответствии со списком состояний и морфизмов (Таблица 1) была построена графовая модель решателя обратной задачи (3) (рис. 2). Данная графовая модель была описана с помощью формата aDOT в соответствии с рис.2 (Листинг 6).

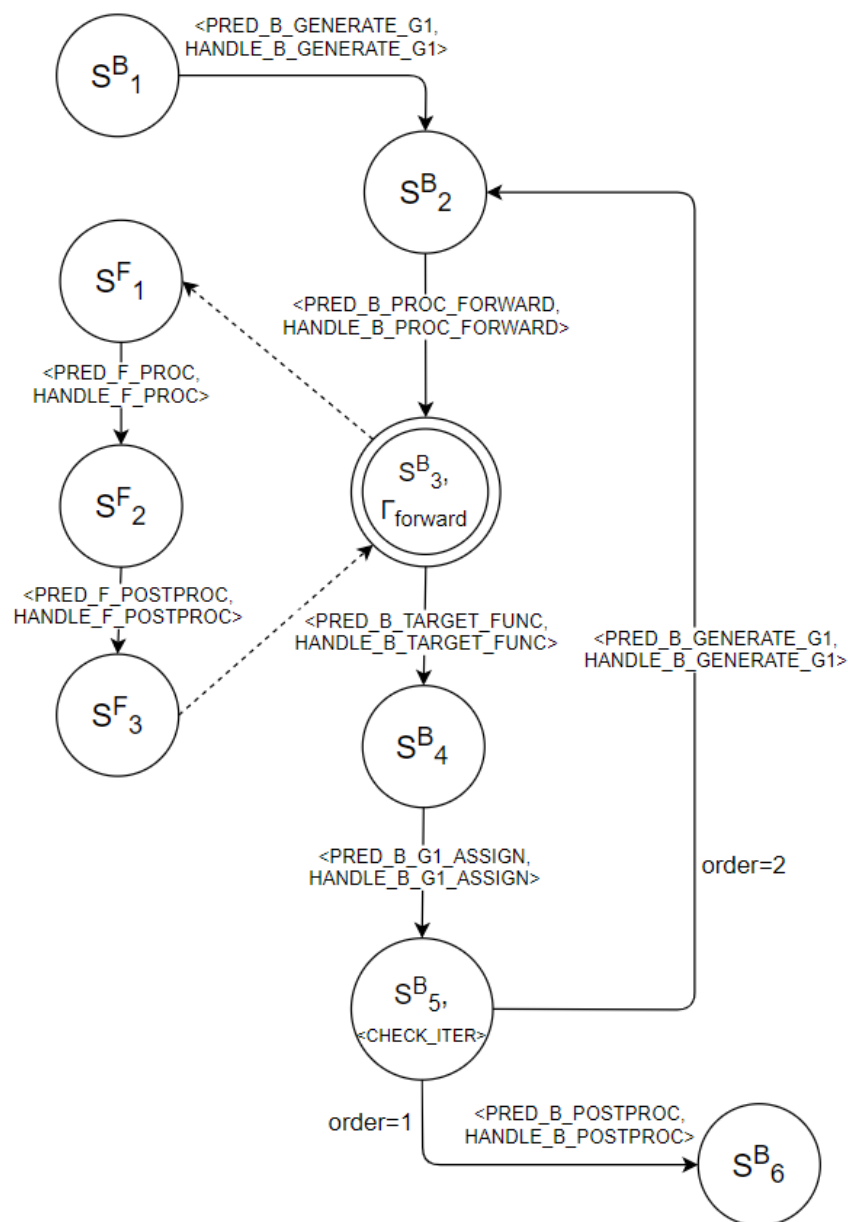


Рис. 2. Графовая модель решателя обратной задачи.

Листинг 6. Файл backward_solver.adot – описание графовой модели решателя прямой задачи в формате aDOT

```

digraph backward_solver_graph {
    // Функции-предикаты
    PRED_B_GENERATE_G1 [module="backward_solver",
entry_func="pred_b_generate_g1"]
    PRED_B_PROC_FORWARD [module="backward_solver",
entry_func="pred_b_proc_forward"]
    PRED_B_TARGET_FUNC [module="backward_solver",
entry_func="pred_b_target_func"]
    PRED_B_G1_ASSIGN [module="backward_solver",
entry_func="pred_b_g1_assign"]
    PRED_B_POSTPROC [module="backward_solver", entry_func="pred_b_postproc"]

    // Функции-обработчики
    HANDLE_B_GENERATE_G1 [module="backward_solver",
entry_func="handle_b_generate_g1"]
    HANDLE_B_PROC_FORWARD [module="backward_solver",
entry_func="handle_b_proc_forward"]
    HANDLE_B_TARGET_FUNC [module="backward_solver",
entry_func="handle_b_target_func"]
    HANDLE_B_G1_ASSIGN [module="backward_solver",
entry_func="handle_b_g1_assign"]
    HANDLE_B_POSTPROC [module="backward_solver",
entry_func="handle_b_postproc"]

    // Функции перехода
    TRANS_B_GENERATE_G1 [predicate=PRED_B_GENERATE_G1,
function=HANDLE_B_GENERATE_G1]
    TRANS_B_PROC_FORWARD [predicate=PRED_B_PROC_FORWARD,
function=HANDLE_B_PROC_FORWARD]
    TRANS_B_TARGET_FUNC [predicate=PRED_B_TARGET_FUNC,
function=HANDLE_B_TARGET_FUNC]
    TRANS_B_G1_ASSIGN [predicate=PRED_B_G1_ASSIGN,
function=HANDLE_B_G1_ASSIGN]
    TRANS_B_POSTPROC [predicate=PRED_B_POSTPROC, function=HANDLE_B_POSTPROC]

    // Функция-селектор
    CHECK_ITER [module="backward_solver", entry_func="check_iter"]

    // Определение структуры графа
    SB_3 [subgraph=forward-solver.adot]
    SB_5 [selector=CHECK_ITER]

    // Описание графа
    BEGIN__ -> SF_1
    SB_1 -> SB_2 [morphism=TRANS_B_GENERATE_G1]
    SB_2 -> SB_3 [morphism=TRANS_B_PROC_FORWARD]
    SB_3 -> SB_4 [morphism=TRANS_B_TARGET_FUNC]
    SB_4 -> SB_5 [morphism=TRANS_B_G1_ASSIGN]
    SB_5 -> SB_2, SB_6 [morphism=(TRANS_B_GENERATE_G1, TRANS_B_POSTPROC),
order=(2, 1)]
    SB_6 -> __END__
}

```

Таким образом Листинг 6 является формальным описанием графоориентированного решателя обратной задачи.

6. Анализ разработанных графовых моделей

Разработанные графовые модели отвечают требованиям масштабируемости.

Модель решателя прямой задачи позволяет заменять/добавлять функцию перехода для постобработки, тем самым достигая необходимого конечного вида результатов (вывод в БД, графический вывод, вывод в консоль и т.д.). Сам численный метод прямой задачи в общем виде не масштабируем – особенности внутренней реализации, параллелизм вычисления k_i^j остается в зоне ответственности программиста, реализующего данный метод. Шаги интегрирования являются последовательными и не могут быть распараллелены.

Модель решателя обратной задачи также позволяет заменять/дополнять этап постпроцессинга, но помимо этого структура метода позволяет выделить его масштабируемую топологию, среди которых гибкими местами остаются:

- стратегия расчета нового вектора \mathbf{g}^1 ;
- решатель прямой задачи;
- целевая функция для оптимизации;
- критерий выбора оптимального вектора \mathbf{g}^1 .

Все перечисленные места подразумевают их замену или дополнение, что делает структуру более гибкой. Кроме того, для итераций метода Монте-Карло можно также ввести параллелизм, позволяющие решать прямую задачу с разными векторами \mathbf{g}^1 .

В качестве ограничений в части использования модели можно выделить жестко запрограммированное ОДУ без возможности изменения левой или правой части.

От данного ограничения можно избавиться путем внедрения синтаксического анализатора, позволяющего вводить произвольную правую часть $f(y, t)$ и дифференциальный оператор $L[y]$ с клавиатуры. Для обеспечения

данной возможности графовые модели могут быть дополнены еще одним состоянием стадии препроцессинга, которое заключается в сформированной математической модели ОДУ после парсинга пользовательского ввода, который являлся бы функцией-обработчиком для перехода в данное состояние.

Заключение

В ходе лабораторной работы были изучены принципы графоориентированного подхода для решения сложных вычислительных методов (СВМ). Разработаны и проанализированы графовые модели решения прямой и обратной задачи в форме ОДУ (2, 4) с помощью СВМ: Рунге-Кутты 4-го порядка и Монте-Карло соответственно.

Для формального описания входных данных и графовых моделей были изучены и применены форматы описания данных aDOT и aINI.

Список использованных источников

1. **Соколов, А.П.** Методы математического моделирования сложных процессов и систем. Лабораторные работы. // Кафедра РК6 (Системы автоматизированного проектирования). МГТУ им. Н.Э. Баумана, Москва, 2018 г., 25 с.
2. **Першин, А.Ю.** Лекции по вычислительной математике (черновик). [Электронный ресурс] // Кафедра РК6 (Системы автоматизированного проектирования). МГТУ им. Н.Э. Баумана, Москва, 2020 г., 142 с.
3. **Соколов, А.П.** Графоориентированный программный каркас для реализации сложных вычислительных методов. / А.П. Соколов, А.Ю. Першин // Программирование. – Т. 47, №5 – 2019, с. 43-55.
4. **Соколов, А.П.** Описание формата данных aDOT (advanced DOT) / А.П. Соколов, А.Ю. Першин // кафедра «Системы автоматизированного проектирования» МГТУ им. Н.Э.Баумана, Москва, 2020г., 6с.

5. **Соколов, А.П.** Описание формата данных aINI (advanced INI) / А.П.Соколов // кафедра «Системы автоматизированного проектирования» МГТУ им. Н.Э.Баумана, Москва, 2020г., 4с.
6. Метод Монте-Карло [Электронный ресурс] // <http://bigor.bmstu.ru/?cnt/?doc=MO/ch1003.mod/?cou=MO/base.cou> - (Дата обращения: 07.01.2022)