

Санкт-Петербургский государственный университет
Кафедра механики управляемого движения

Горбунова Анна Андреевна

Магистерская диссертация

Адаптивный метод оптимизации функций в динамике

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа «Математическое моделирование в задачах
естествознания»

Научный руководитель,
доктор физ.-мат. наук,
профессор
Бабаджаниянц Л.К.

Санкт-Петербург

2017

Содержание

Введение.....	4
Постановка задачи.....	4
Обзор литературы.....	6
Глава 1. Идентификация параметров кинетических моделей.....	7
1.1. Постановка задачи.....	7
1.2. Градиентные уравнения.....	8
1.3. Уравнения в вариациях.....	10
1.4. Функционалы нелинейного метода наименьших квадратов.....	11
Глава 2. Метод GEM минимизации функций многих переменных.....	13
2.1. Метод градиентных уравнений.....	13
2.2. Метод четвертой точки.....	14
2.2.1. Построение четвертой точки.....	14
2.2.2. Ускорение.....	15
2.2.3. Замедление.....	15
2.2.4. Схема метода четвертой точки.....	16
2.3. Схема метода GEM.....	17
Глава 3. Модель Лотки-Вольтерры.....	18
3.1. Система «хищник-жертва».....	18
3.2. Дифференциальные уравнения.....	20
Глава 4. Численные эксперименты.....	22
Заключение.....	28
Список литературы.....	29
Приложения.....	31

АННОТАЦИЯ

Магистерская диссертация студентки 2 курса магистратуры кафедры механики управляемого движения факультета ПМ-ПУ СПбГУ Горбуновой Анны Андреевны «Адаптивный метод оптимизации функций в динамике». Рассматривается новый метод минимизации функций многих переменных, сочетающий в себе метод градиентных уравнений и метод четвертой точки. Данный метод применяется к задаче идентификации параметров кинетических моделей. Приводятся результаты численного эксперимента идентификации параметров модели Лотки-Вольтерры. Все расчеты выполняются с использованием пакета MATLAB R2017a.

Ключевые слова: минимизация функций многих переменных, задача идентификации параметров.

ABSTRACT

Graduate qualification work of the Master "An adaptive method of optimizing functions in Dynamics" written by Anna Gorbunova, the senior year student of the controlled motion dynamics department of Applied Mathematics and Control Processes faculty of Saint-Petersburg State University, Saint-Petersburg, Russia. We consider a new method of minimizing functions of many variables which combines the gradient equations method and the method of the fourth point. This method is applied to the problem of identification of parameters of kinetic models. The results of numerical experiment of parameter identification Lotka-Volterra model. All computations are performed using MATLAB version R2017a.

Key words: minimization of functions of many variables, the problem of parameter identification.

Введение.

Постановка задачи.

В прикладной математике очень часто нахождение решения определенного рода задач сводится к задаче минимизации функций многих переменных. Разработано множество численных методов ее решения, для программной реализации которых важны две главные характеристики: скорость и надежность при заданной точности.

Различные методы ориентированы на различные области применения. В сложных задачах, требующих высокой точности, наиболее часто используется метод градиентных уравнений, который приводит к положительным результатам в тех задачах, где многие другие методы отказывают. С другой стороны, метод градиентных уравнений является трудоемким, и его целесообразно использовать в комбинации с менее трудоемкими методами.

В рассмотренном в настоящей работе методе GEM метод градиентных уравнений используется в комбинации с гораздо более экономным методом четвертой точки. Отметим, что метод четвертой точки является обобщением метода оврагов и использует простейший метод градиента, экстраполяцию и интерполяцию в пространстве параметров.

Метод ориентирован на сложные задачи, в которых само вычисление функции требует больших вычислительных затрат. К таким относится, в частности, задача идентификации параметров моделей, описываемых системами дифференциальных уравнений.

Таким образом, в настоящей работе были поставлены следующие цели:

1. Рассмотреть новый метод минимизации функций многих переменных, сочетающий в себе метод градиентных уравнений и метод четвертой точки.

2. Сформулировать задачу идентификации параметров модели Лотки-Вольтерры. Составить для данной задачи уравнения в вариациях и выбрать функционал метода наименьших квадратов.

3. Провести численные эксперименты по решению сформулированной задачи в соответствии со схемой предложенного в настоящей работе метода GEM.

4. Повторить численные эксперименты в соответствии с методами, предложенными в работе [7].

5. Сравнить результаты проведенных экспериментов по двум показателям: точности и затраченному процессорному времени.

Обзор литературы.

Известно множество численных методов для решения задачи минимизации функций многих переменных, большая часть которых относится к градиентным методам.

Метод покоординатного спуска является несложным и легко программируемым. Но он сходится медленно, а при наличии оврагов может вообще не сходиться. Его очень часто используют в качестве первой попытки нахождения минимума. К быстрым методам минимизации функции относятся различные варианты метода Ньютона. Реализации этих методов можно найти в работах [2,5,13,14].

В сложных задачах «овражного типа» эффективен метод оврагов [4,6] и метод градиентных уравнений (метод тяжелого шарика) [2]. Основная идея метода оврагов заключается в линейной экстраполяции вдоль «оврага» с последующим исправлением полученной точки одним из градиентных методов. Основная сложность этого метода заключается в выборе «овражных» шагов при итерациях.

Метод тяжелого шарика основан на численном интегрировании (например, методами Рунге-Кутты) градиентных уравнений по тому или иному параметру. К сожалению, этот метод является очень медленным.

Решение задачи минимизации функций многих переменных находит свое практическое применение при решении задачи идентификации параметров кинетических моделей [1,8–12]. В ней рассматривается функционал метода наименьших квадратов, зависящий от параметров рассматриваемой модели, который в дальнейшем необходимо минимизировать.

Глава 1. Идентификация параметров кинетических моделей.

1.1. Постановка задачи.

Рассматривается кинетическая модель, описываемая задачей Коши:

$$\frac{dx_j}{dt} = f_j(x_1, \dots, x_n; k_1, \dots, k_m) \quad (1)$$

$$x_j(t_0) = x_{j0}, \quad j \in [1, n] \quad (2)$$

где k_1, \dots, k_m – параметры, которые необходимо определить или уточнить на основе (1), (2) и известного ряда измерений:

$$x_{jk} = x_j(t_k), k \in [1, M], t_1 < t_2 < \dots < t_M, j \in [1, q], q \leq n.$$

Для этих параметров также задано начальное приближение $k^0 = (k_1^0, \dots, k_m^0)$.

Задачу определения этих параметров обычно сводят к минимизации того или иного функционала нелинейного метода наименьших квадратов, зависящего от экспериментальных данных и решения задачи Коши (1), (2):

$$\Phi(k) = \frac{1}{2} \sum_{j=1}^q \sum_{p=1}^M \alpha_{jp} \cdot (x_j(t_p, k) - x_{jp})^2.$$

Здесь используются обозначения: $\alpha_{jp} \geq 0$ – фиксированные весовые коэффициенты, $x_j(t_p, k)$, $j \in [1, q]$ – значения первых q компонент решения задачи (1), (2) в точке t_p при заданных $k = (k_1, \dots, k_m)$. Предполагается, что значение k^* , доставляющее минимум функционалу $\Phi(k)$, является адекватным приближением к реальному значению параметра $k = (k_1, \dots, k_m)$ для рассматриваемой модели процесса. Величину k^* называют МНК-решением задачи идентификации параметров кинетической модели.

1.2. Градиентные уравнения.

Рассмотрим вещественнозначную функцию $\Phi(k)$ аргумента $k = (k_1, \dots, k_m)$, $(k_1, \dots, k_m) \in R^m$. Пусть $l = (l_1, \dots, l_m) \in R^m$ и $l_1^2 + \dots + l_m^2 = 1$.

Тогда производная функции $\Phi(k)$ по направлению l :

$$\frac{\partial \Phi(k)}{\partial l} = \lim_{h \rightarrow 0} \frac{\Phi(k + hl) - \Phi(k)}{h} = \frac{d}{dh} \Phi(k + hl)|_{h=0} \quad (3)$$

характеризует скорость изменения $\Phi(k)$ при изменении аргумента k в направлении вектора l .

Из формулы (3) получаем:

$$\frac{\partial \Phi(k)}{\partial l} = \frac{d}{dh} \Phi(k_1 + hl_1, \dots, k_m + hl_m)|_{h=0} = \frac{\partial \Phi(k)}{\partial k_1} l_1 + \dots + \frac{\partial \Phi(k)}{\partial k_m} l_m = (\nabla \Phi(k), l)$$

где $\nabla \Phi(k) = \left(\frac{\partial \Phi(k)}{\partial k_1}, \dots, \frac{\partial \Phi(k)}{\partial k_m} \right)$ – градиент $\Phi(k)$, из чего следует:

$$\begin{aligned} \frac{\partial \Phi(k)}{\partial l} &= |\nabla \Phi(k)| \cos(\nabla \Phi(k), l) \\ -|\nabla \Phi(k)| &\leq \frac{\partial \Phi(k)}{\partial l} \leq |\nabla \Phi(k)| \\ \frac{\partial \Phi(k)}{\partial l} &= \begin{cases} |\nabla \Phi(k)|, & \frac{\nabla \Phi(k)}{|\nabla \Phi(k)|} = \frac{l}{|l|} \\ -|\nabla \Phi(k)|, & \frac{\nabla \Phi(k)}{|\nabla \Phi(k)|} = -\frac{l}{|l|} \end{cases} \end{aligned}$$

Получаем, что вектор $\nabla \Phi(k)$ ($-\nabla \Phi(k)$) является направлением наискорейшего роста (убывания) функции $\Phi(k)$ в точке k .

Градиентной кривой функции $\Phi(k)$ называется кривая $k(\tau) = (k_1(\tau), \dots, k_m(\tau))$, $(k_1(\tau), \dots, k_m(\tau)) \in R^m$, $\tau \in [0, +\infty]$, касательное направление к которой в каждой точке τ совпадает с направлением наискорейшего ее убывания.

Другими словами, $k(\tau) = (k_1(\tau), \dots, k_m(\tau))$ удовлетворяет следующим дифференциальным уравнениям:

$$\frac{dk_i}{d\tau} = -\frac{\partial \Phi(k_1, \dots, k_m)}{\partial k_i}, \quad i \in [1, m] \quad (4)$$

Начальные условия для уравнений (4):

$$k_i(0) = k_i^0, \quad i \in [1, m] \quad (5)$$

Решение задачи Коши (4), (5) определяет градиентную кривую, проходящую через точку k^0 . Рассмотрим это решение как вектор-функцию $k(k^0, \tau) = (k_1(k^0, \tau), \dots, k_m(k^0, \tau))$, зависящую от аргументов $k^0 \in R^m$ и $\tau \in [0, +\infty]$.

Далее ищем точку k^* локального минимума неотрицательной функции $\Phi(k)$, если она существует и достаточно близка к начальному приближению k^0 . Движение вдоль траектории решения $k(k^0, \tau)$ (то есть движение вдоль градиентной кривой, проходящей через k_0) можно считать наилучшим путем к точке минимума k^* .

Если решение задачи (4), (5) существует при $\tau \in [0, +\infty]$, то для любого τ получаем:

$$\frac{d\Phi(k(\tau))}{d\tau} = - \left(\nabla\Phi(k(\tau)), \nabla\Phi(k(\tau)) \right) < 0 \quad \text{при} \quad \nabla\Phi(k(\tau)) \neq 0,$$

$$\frac{d\Phi(k(\tau))}{d\tau} = 0 \quad \text{при} \quad \nabla\Phi(k(\tau)) = 0.$$

В этом случае мы можем ожидать, что $\lim_{\tau \rightarrow +\infty} k(k^0, \tau) = k^*$.

1.3. Уравнения в вариациях.

Обратим внимание на функционалы, зависящие от параметров k_1, \dots, k_m не напрямую, а через задачу Коши (1), (2):

$$\begin{aligned}\frac{dx_j}{dt} &= f_j(x_1, \dots, x_n; k_1, \dots, k_m) \\ x_j(t_0) &= x_{j0}, \quad j \in [1, n]\end{aligned}$$

Тогда градиентные уравнения будут зависеть от производных по искомым параметрам решения задачи (1), (2). Для этого необходимые производные нужно вычислить. Дифференцируя уравнения (1), (2) по k_r , получаем, что функции

$$y_{jr}(t) = \frac{\partial x_j(t, k_1, \dots, k_m)}{\partial k_r}, \quad j \in [1, n], r \in [1, m] \quad (6)$$

удовлетворяют дифференциальным уравнениям:

$$\frac{dy_{jr}}{dt} = \frac{\partial f_j}{\partial k_r} + \sum_{p=1}^n \frac{\partial f_j}{\partial x_p} y_{pr} \quad (7)$$

с начальными условиями:

$$y_{jr}(t_0) = 0, \quad j \in [1, n], r \in [1, m] \quad (8)$$

Уравнения (7) относительно производных (6) называются уравнениями в вариациях для уравнений (1).

1.4. Функционалы нелинейного метода наименьших квадратов.

Естественно использовать различные функционалы для разных экспериментов, поскольку каждый из них имеет свои возможности. Рассмотрим три вида функционалов, зависящих от набора величин x_{jp} .

В эксперименте первого типа рассматриваются одни и те же начальные данные x_0 , измеряются значения $x_{qp} = x_q(t_p, t_0, x_0)$ одной из переменных $x_q(t_p, t_0, x_0)$ в различные моменты времени $t = t_p, p \in [1, M]$.

В эксперименте второго типа рассматриваются начальные данные $x_0 = x_0^r, r \in [1, N]$, измеряются значения $x_{jM}^r = x_j(t_M, t_0, x_0^r)$ q величин $x_j(t, t_0, x_0), j \in [1, q]$ в один и тот же момент времени $t = t_M$.

В эксперименте третьего типа рассматриваются начальные данные $x_0 = x_0^r, r \in [1, N]$, измеряются значения $x_{jp}^r = x_j(t_{jp}, t_0, x_0^r)$ q величин $x_j(t, t_0, x_0), j \in [1, q]$ в моменты времени $t = t_{jp}, j \in [1, q], p \in [1, M_j]$.

Для каждого эксперимента выпишем соответствующие функционалы:

$$\Phi_I(k) = \frac{1}{2} \sum_{p=1}^M \alpha_p \cdot (x_q(t_p, x_0, k) - x_{qp})^2$$

$$\Phi_{II}(k) = \frac{1}{2} \sum_{r=1}^N \sum_{j=1}^q \alpha_{rj} \cdot (x_j(t_M, x_0^r, k) - x_{jM}^r)^2$$

$$\Phi_{III}(k) = \frac{1}{2} \sum_{r=1}^N \sum_{j=1}^q \sum_{p=1}^{M_j} \alpha_{rjp} \cdot (x_j(t_{jp}, x_0^r, k) - x_{jp}^r)^2$$

Здесь $\alpha_p, \alpha_{rj}, \alpha_{rjp}$ – фиксированные весовые коэффициенты.

Для рассмотренных функционалов выпишем градиентные уравнения и соответствующие им начальные условия

$$\frac{dk_i}{d\tau} = - \frac{\partial \Phi_I(k)}{\partial k_i} = - \sum_{p=1}^M \alpha_p \cdot (x_q(t_p, x_0, k) - x_{qp}) \cdot y_{qi}(t_p, x_0, k) \quad (9)$$

$$\frac{dk_i}{d\tau} = -\frac{\partial\Phi_{II}(k)}{\partial k_i} = -\sum_{r=1}^N \sum_{j=1}^q \alpha_{rj} \cdot (x_j(t_M, x_0^r, k) - x_{jM}^r) \cdot y_{ji}(t_M, x_0^r, k) \quad (10)$$

$$\frac{dk_i}{d\tau} = -\frac{\partial\Phi_{III}(k)}{\partial k_i} = -\sum_{r=1}^N \sum_{j=1}^q \sum_{p=1}^{M_j} \alpha_{rjp} \cdot (x_j(t_{jp}, x_0^r, k) - x_{jp}^r) \cdot y_{ji}(t_{jp}, x_0^r, k) \quad (11)$$

$$k_i(0) = k_i^0, \quad i \in [1, m]$$

В формулах (9), (10), (11) величины x_{qp}, x_{jM}^r и x_{jp}^r соответственно известны из эксперимента как его результаты, величины $x_q(t_p, x_0, k), x_j(t_M, x_0^r, k)$ и $x_j(t_{jp}, x_0^r, k)$ находятся из задачи Коши (1), (2) путем ее численного интегрирования, а величины $y_{qi}(t_p, x_0, k), y_{ji}(t_M, x_0^r, k)$ и $y_{ji}(t_{jp}, x_0^r, k)$ получаются численным интегрированием уравнений в вариациях (7) с начальными условиями (8).

Глава 2. Метод GEM минимизации функций многих переменных.

Как уже говорилось ранее, метод градиентных уравнений достаточно трудоемкий и гораздо эффективнее было бы использовать его в комбинации с другими, менее трудоемкими, методами. В методе GEM (Gradient Equations Method) метод градиентных уравнений используется в сочетании с гораздо более экономным методом четвертой точки. Его схему можно увидеть на рис.2.

2.1. Метод градиентных уравнений.

Метод градиентных уравнений нахождения локального минимума функционала $\Phi(k)$ заключается в численном интегрировании задачи Коши (4), (5) вдоль оси τ до достижения точки $k(\tau)$, достаточно близкой к k^* .

При использовании данного метода в комбинации с высокоточными методами численного интегрирования последовательно получают точки k^1, k^2, \dots , лежащие достаточно близко от градиентной кривой, начинающейся в начальной точке k^0 .

В методе градиентных уравнений для задачи идентификации параметров кинетических моделей численно интегрируются градиентные уравнения (9), (10) или (11) при заданных начальных условиях.

2.2. Метод четвертой точки.

Метод четвертой точки является обобщением метода оврагов и использует простейший метод градиента, экстраполяцию и интерполяцию в пространстве параметров. Схему метода можно увидеть на рис.1.

2.2.1. Построение четвертой точки.

Предполагаются известными три точки k^1, k^2, k^3 , лежащие достаточно близко от градиентной кривой функционала $\Phi(k)$, начинающейся в точке k^1 и такие, что $\Phi(k^1) > \Phi(k^2) > \Phi(k^3)$. В соответствии с методом, мы можем найти четвертую точку k^4 такую, что $\Phi(k^3) > \Phi(k^4)$. Построение искомой точки производится в два этапа.

1. Построение начального приближения \tilde{k}^4 .

Через известные точки k^1, k^2, k^3 , проводится полином следующего вида:

$$k(S) = aS^2 + bS + c, \quad S \in [0,1].$$

Рассматриваемый полином должен удовлетворять условиям:

$$k(S_i) = k^{i+1}, \quad i = 0, 1, 2, \quad S_0 = 0, S_3 = 1,$$
$$\frac{S_3 - S_2}{S_2 - S_1} = \frac{S_2 - S_1}{S_1 - S_0} = \frac{\|k^3 - k^2\|}{\|k^2 - k^1\|}$$

Далее полагаем, что $\tilde{k}^4 = k(S_3) = k(1)$ и получаем:

$$\tilde{k}^4 = a + b + c.$$

Здесь используются следующие обозначения:

$$a = \frac{1}{S_1 S_2 (S_1 - S_2)} [S_2 \cdot (k^2 - k^1) - S_1 \cdot (k^3 - k^1)],$$

$$b = \frac{1}{S_1 S_2 (S_1 - S_2)} [S_1^2 \cdot (k^3 - k^1) - S_2^2 \cdot (k^2 - k^1)],$$

$$c = k^1.$$

2. Построение k^4 .

Точка k^4 находится при помощи нескольких последовательных приближений простейшим методом градиента, исходя из начального приближения \tilde{k}^4 .

Приближения производятся по формулам:

$$k^{i+1} = k^i - h \cdot \nabla \Phi(k^i).$$

2.2.2. Ускорение.

После того, как по точкам k^1, k^2, k^3 получили точку k^4 , по точкам k^2, k^3, k^4 получаем точку k^5 . Далее из этих пяти точек выделяем три точки k^1, k^3, k^5 затем, чтобы в дальнейшем строить по ним следующую точку. Переход к выделенным точкам назовем ускорением.

2.2.3. Замедление.

Если по трем точкам k^1, k^2, k^3 не получается найти четвертую, удовлетворяющую условиям в п.2.2.1, то включается алгоритм замедления. Он заключается в том, что по известным точкам k^1, k^2, k^3 ищется такая точка k^* , что выполняется условие: $\Phi(k^2) > \Phi(k^*) > \Phi(k^3)$. Построение искомой точки производится в 2 этапа.

1. Построение начального приближения \tilde{k}^* .

Мы также используем полином $k(S) = aS^2 + bS + c, S \in [0,1]$, удовлетворяющий следующим условиям:

$$k(S_i) = k^{i+1}, \quad i = 0, 1, 2, \quad S_0 = 0, S_3 = 1, \\ \frac{S_3 - S_2}{S_2 - S_1} = \frac{S_2 - S_1}{S_1 - S_0} = \frac{\|k^3 - k^2\|}{\|k^2 - k^1\|}$$

Далее полагаем, что

$$\tilde{k}^* = aS^{*2} + bS^* + c, \quad \text{где } S^* = \frac{S_1 + S_2}{2}.$$

2. Построение k^* .

Точка k^* находится при помощи нескольких последовательных приближений простейшим методом градиента, исходя из начального приближения \tilde{k}^* .

2.2.4. Схема метода четвертой точки.



Рис.1 Метод четвертой точки

2.3. Схема метода GEM.

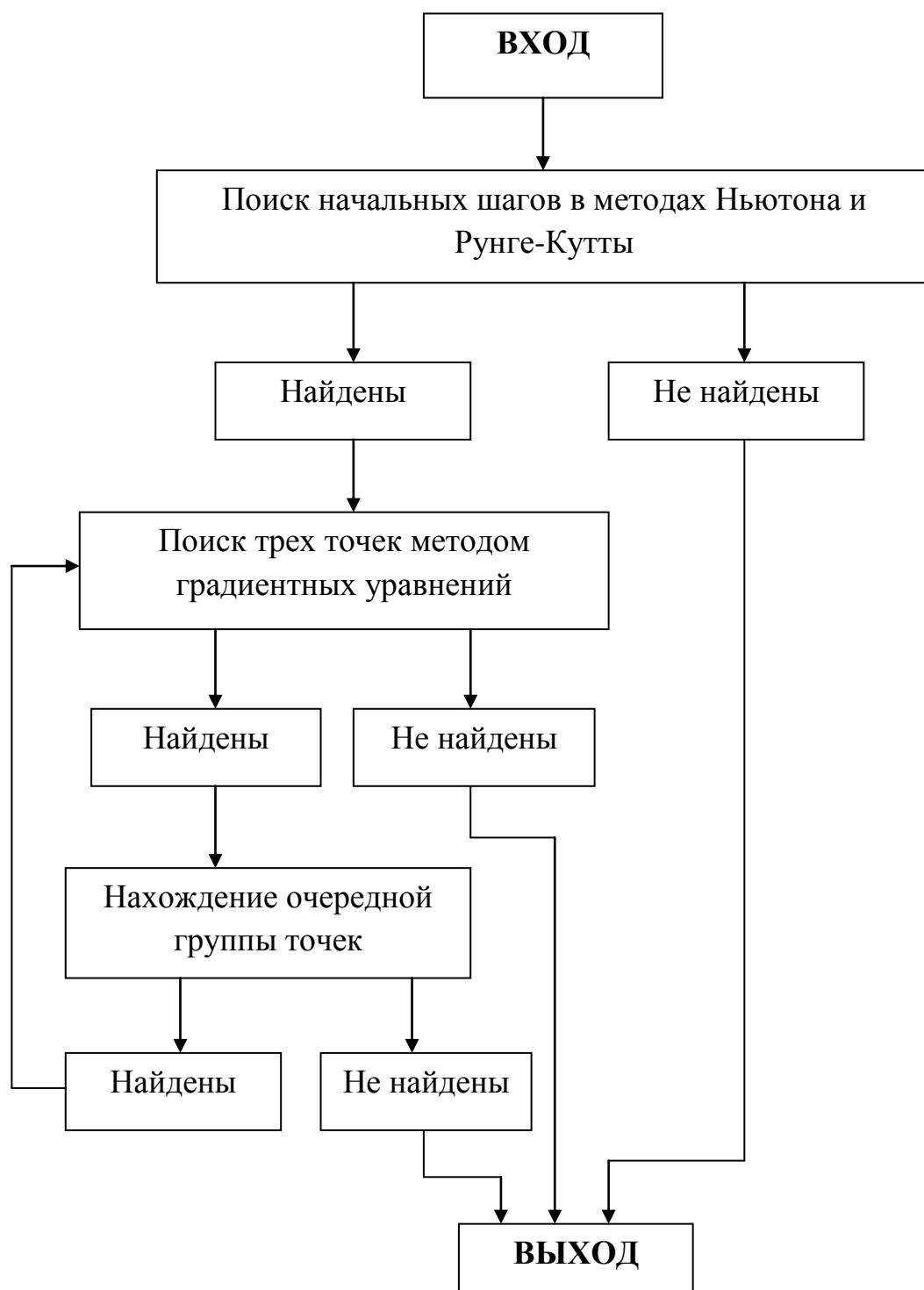


Рис.2 Общая схема метода GEM

Глава 3. Модель Лотки-Вольтерры.

3.1. Система «хищник-жертва».

Первая модель типа “хищник – жертва” предлагалась в трудах Лотки [10]. Эта экологическая модель основана на следующих уравнениях (ОДУ Лотки):

$$\begin{aligned}\frac{d[A]}{dt} &= -k_1[A] \cdot [X], \\ \frac{d[X]}{dt} &= k_1[A] \cdot [X] - k_2[X] \cdot [Y], \\ \frac{d[Y]}{dt} &= k_2[X] \cdot [Y] - k_3[Y], \\ \frac{d[B]}{dt} &= k_3[Y].\end{aligned}\tag{1}$$

Предполагается, что в некоторой местности живут жертвы X и хищники Y (зайцы и волки). Строится математическая модель изменения следующих величин: A (количество растительной пищи для зайцев), X (численность жертв), Y (численность хищников) и B (число умерших волков). При этом предполагается:

1. Увеличение численности зайцев зависит от наличия зайцев X и пищи для них A . Это можно записать с помощью формулы:



2. Увеличение численности волков зависит от наличия волков Y и пищи для них (зайцев) X . Это условие записывается формулой:



3. Численность волков уменьшается от старости или болезней, что выражается следующей формулой:



4. Скорости изменения численности зайцев и волков, а также скорость увеличения числа умерших волков по формулам (2), (3), (4) соответственно равны: $k_1[A] \cdot [X]$, $k_2[X] \cdot [Y]$, $k_3[Y]$. Здесь используются обозначения:

$[A]$ – количество растительной пищи;

$[X]$ – число жертв (зайцев);

$[Y]$ – число хищников (волков);

k_1, k_2, k_3 – положительные коэффициенты.

5. Скорость изменения каждой из величин $[A]$, $[X]$, $[Y]$, $[B]$ равна сумме скоростей изменения количеств растительной пищи, зайцев и волков в процессах, описываемых формулами (2), (3), (4).

Система уравнений Лотки (1) следует из перечисленных выше условий.

Еще один ученый, Вольтерра, предлагал более общие модели поведения n хищников и m жертв в разных экологических ситуациях [3]. Поэтому уравнения Лотки (1) часто называются уравнениями Лотки-Вольтерры.

Чаще всего используется более упрощенная (по сравнению с моделью Лотки) модель. Предполагается, что величина $[A]$ постоянна, т.е. зайцы имеют неограниченное количество растительной пищи. Из четырех уравнений остается только два:

$$\frac{d[X]}{dt} = k_0[X] - k_2[X] \cdot [Y],$$
$$\frac{d[Y]}{dt} = k_2[X] \cdot [Y] - k_3[Y].$$

Здесь $k_0 = k_1[A]$.

3.2. Дифференциальные уравнения.

Задача Коши для уравнений Лотки выглядит следующим образом:

$$\begin{aligned}\frac{dx_1}{dt} &= -k_1 x_1 x_2, \\ \frac{dx_2}{dt} &= k_1 x_1 x_2 - k_2 x_2 x_3, \\ \frac{dx_3}{dt} &= k_2 x_2 x_3 - k_3 x_3, \\ \frac{dx_4}{dt} &= k_3 x_3.\end{aligned}\tag{5}$$

Начальные условия для уравнений (5):

$$x_j(t_0) = x_{j0}, \quad j \in [1,4].\tag{6}$$

Теперь запишем уравнения в вариациях для построенной системы:

$$\begin{aligned}\frac{dy_{11}}{dt} &= -k_1 x_2 y_{11} - x_1 (x_2 + k_1 y_{21}), \\ \frac{dy_{12}}{dt} &= -k_1 (x_2 y_{12} + x_1 y_{22}), \\ \frac{dy_{13}}{dt} &= -k_1 (x_2 y_{13} + x_1 y_{23}), \\ \frac{dy_{21}}{dt} &= x_1 x_2 + k_1 x_2 y_{11} - k_2 x_2 y_{31} + (k_1 x_1 - k_2 x_3) y_{21}, \\ \frac{dy_{22}}{dt} &= (k_1 x_1 - k_2 x_3) y_{22} - x_2 (x_3 - k_1 y_{12} + k_2 y_{32}), \\ \frac{dy_{23}}{dt} &= (k_1 x_1 - k_2 x_3) y_{23} + x_2 (k_1 y_{13} - k_2 y_{33}), \\ \frac{dy_{31}}{dt} &= k_2 x_3 y_{21} + (k_2 x_2 - k_3) y_{31}, \\ \frac{dy_{32}}{dt} &= x_2 x_3 + k_2 x_3 y_{22} + (k_2 x_2 - k_3) y_{32}, \\ \frac{dy_{33}}{dt} &= x_3 (k_2 y_{23} - 1) + (k_2 x_2 - k_3) y_{33}, \\ \frac{dy_{41}}{dt} &= k_3 y_{31}, \\ \frac{dy_{42}}{dt} &= k_3 y_{32}, \\ \frac{dy_{43}}{dt} &= x_3 + k_3 y_{33}.\end{aligned}\tag{7}$$

Уравнения в вариациях дополним следующими начальными условиями:

$$y_{ij}(t_0) = 0, \quad i \in [1,4], \quad j \in [1,3]. \quad (8)$$

Задача Коши (5), (6), (7), (8) является полиномиальной.

Используем следующий функционал для решения задачи идентификации модели Лотки-Вольтерры:

$$\Phi_I(k) = \frac{1}{2} \sum_{p=1}^M \alpha_p \cdot (x_q(t_p, x_0, k) - x_{qp})^2.$$

Градиентные уравнения для данного функционала будут следующими:

$$\frac{dk_i}{d\tau} = - \frac{\partial \Phi_I(k)}{\partial k_i} = - \sum_{p=1}^M \alpha_p \cdot (x_q(t_p, x_0, k) - x_{qp}) \cdot y_{qi}(t_p, x_0, k).$$

Глава 4. Численные эксперименты.

В данной главе описывается постановка и результаты численного эксперимента, проведенного в соответствии с рассмотренной схемой метода GEM. Был проведен тот же эксперимент, но в соответствии с методами, предложенными в работе [7]. Далее результаты этих двух экспериментов сравниваются по затраченному процессорному времени.

Обратимся к системе дифференциальных уравнений для модели Лотки-Вольтерры. Действуем по схеме:

1. Зафиксируем начальные данные:

$$x_1(0) = 3; x_2(0) = 0,9; x_3(0) = 2,1; x_4(0) = 0,$$

а также параметры:

$$k_1 = 1; k_2 = 1,5; k_3 = 0,1.$$

2. При выбранных начальных данных и параметрах численным интегрированием решаем задачу Коши (5), (6) предыдущей главы. Далее находим значения величины $x_4(t, k_1, k_2, k_3)$ в моменты времени $t_i = i$, $i = 1, \dots, M = 30$.

Далее «имитируем» измерения величин x_{4i} по следующей формуле:

$$x_{4i} = (1 + \xi_i)x_4(25 * i/M, k_1, k_2, k_3),$$

В этой формуле ξ_i – независимые случайные величины, равномерно распределенные между значениями -0,05 и +0,05.

3. В рассматриваемом функционале нелинейного метода наименьших квадратов весовые коэффициенты выбираем $\alpha_i = 1, i = [1, M]$. Тогда функционал принимает следующий вид:

$$\Phi(k_1, k_2, k_3) = \frac{1}{2} \sum_{i=1}^M (x_4(25 * i/M, k_1, k_2, k_3) - x_{4i})^2.$$

Выписываем для него градиентные уравнения:

$$\frac{dk_1}{d\tau} = - \sum_{i=1}^M (x_4(25 * i/M, k_1(\tau), k_2(\tau), k_3(\tau)) - x_{4i}) \cdot y_{41}(25 * i/M),$$

$$\frac{dk_2}{d\tau} = - \sum_{i=1}^M (x_4(25 * i/M, k_1(\tau), k_2(\tau), k_3(\tau)) - x_{4i}) \cdot y_{42}(25 * i/M),$$

$$\frac{dk_3}{d\tau} = - \sum_{i=1}^M (x_4(25 * i/M, k_1(\tau), k_2(\tau), k_3(\tau)) - x_{4i}) \cdot y_{43}(25 * i/M).$$

4. Зафиксируем начальное приближение для идентифицируемых параметров:

$$k^0 = (k_1^0, k_2^0, k_3^0) = (0,5, 0,7, 0,4).$$

Находим приближенное значение точки локального минимума $k^* = (k_1^*, k_2^*, k_3^*)$ используемого функционала предложенным выше методом GEM.

Эффективность метода проверяем, исходя из двух величин: относительной погрешности и затраченного процессорного времени.

Покажем, насколько решение задачи Коши (5), (6) при фиксированных параметрах из п.1 отличается от решения при начальном приближении из п.4 для этих параметров (рис.3).

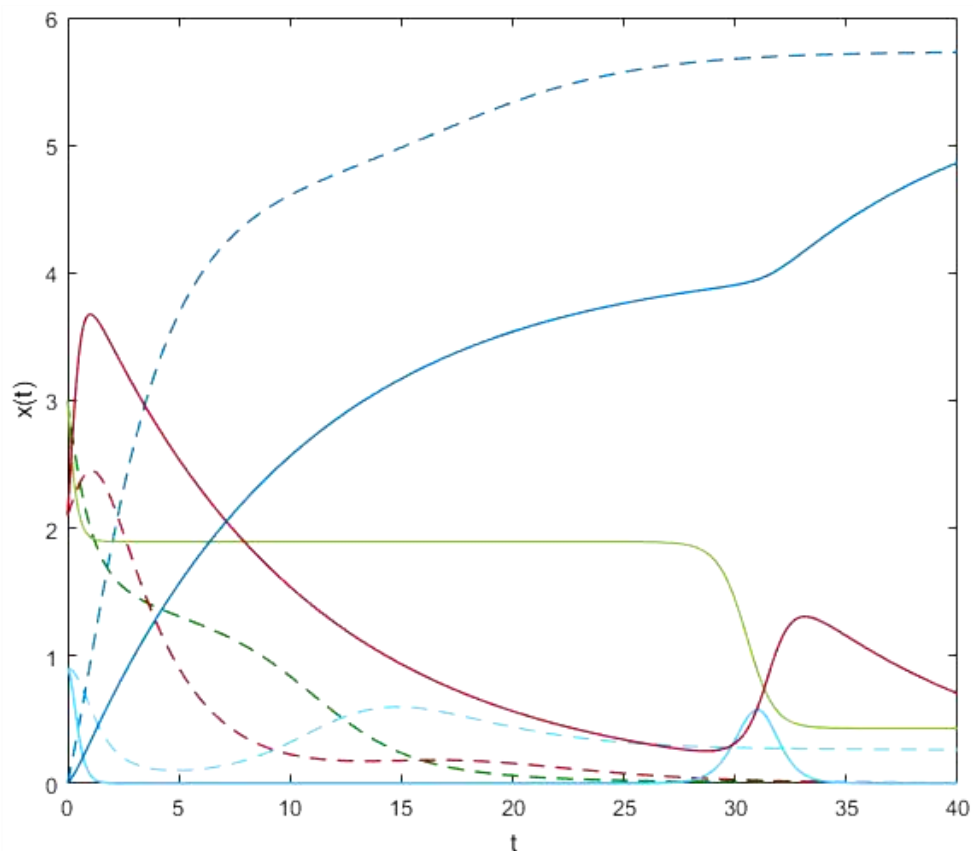


Рис.3 Истинная модель при параметрах (1, 1,5, 0,1) (сплошные линии) и модель при начальном приближении параметров (0,5, 0,7, 0,4) (пунктирные линии)

Как видно из рисунка 3, модели при истинных и начальных параметрах очень сильно отличаются.

Результаты проведенного численного эксперимента приведены на рисунках 4 – 5.

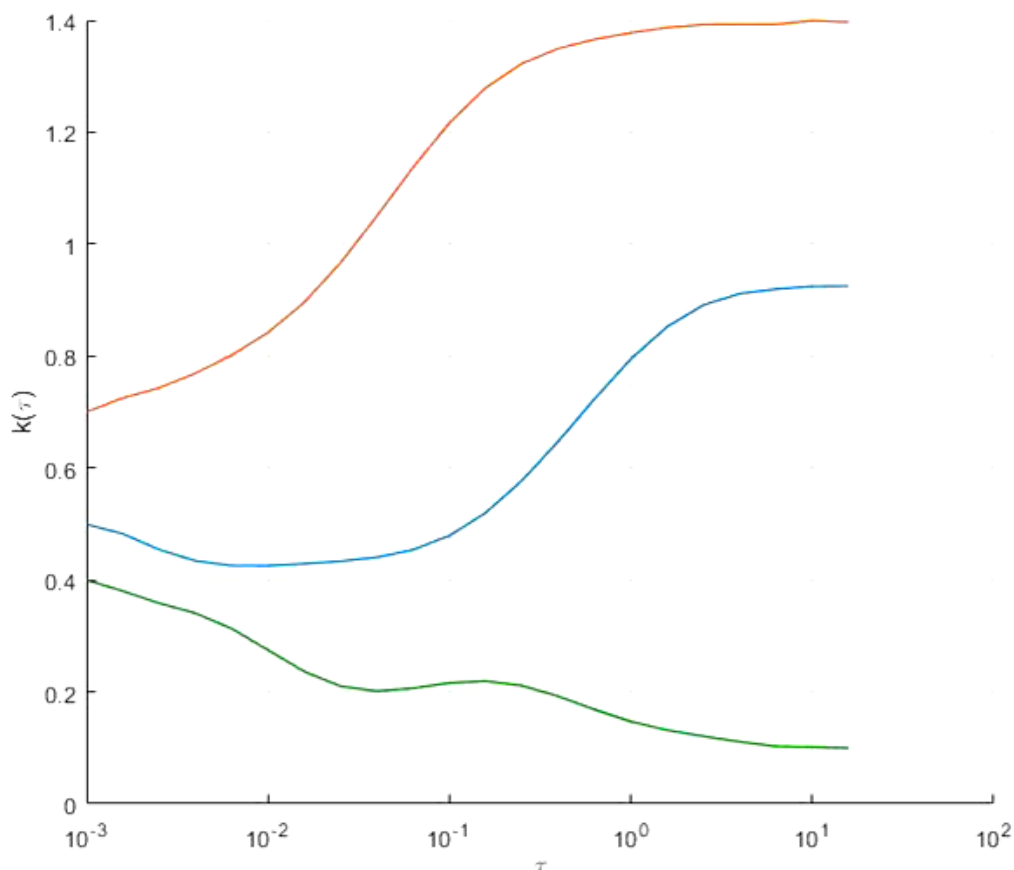


Рис.4 Траектория параметров k_1, k_2, k_3 (идентификация методом GEM)

В проведенном эксперименте параметры идентифицируются с большой точностью и приводят к точке минимума, достаточно близкой к точке (1,1,5,0,1). Вычисляя относительную погрешность по формуле

$$\frac{1}{\sqrt{k_1^2 + k_2^2 + k_3^2}} \cdot \max_{i \in [1:3]} |k_i - k_i^*|, \text{ получаем число } 0.036, \text{ или } 3.6\%. \text{ Что}$$

касается затраченного на решение задачи процессорного времени, то компьютер выдал ответ спустя 9 минут после запуска программы.

На рисунке 5 сравниваются истинное решение задачи Коши (5), (6) при параметрах (1,1,5,0,1) и решение той же задачи Коши при идентифицированных параметрах.

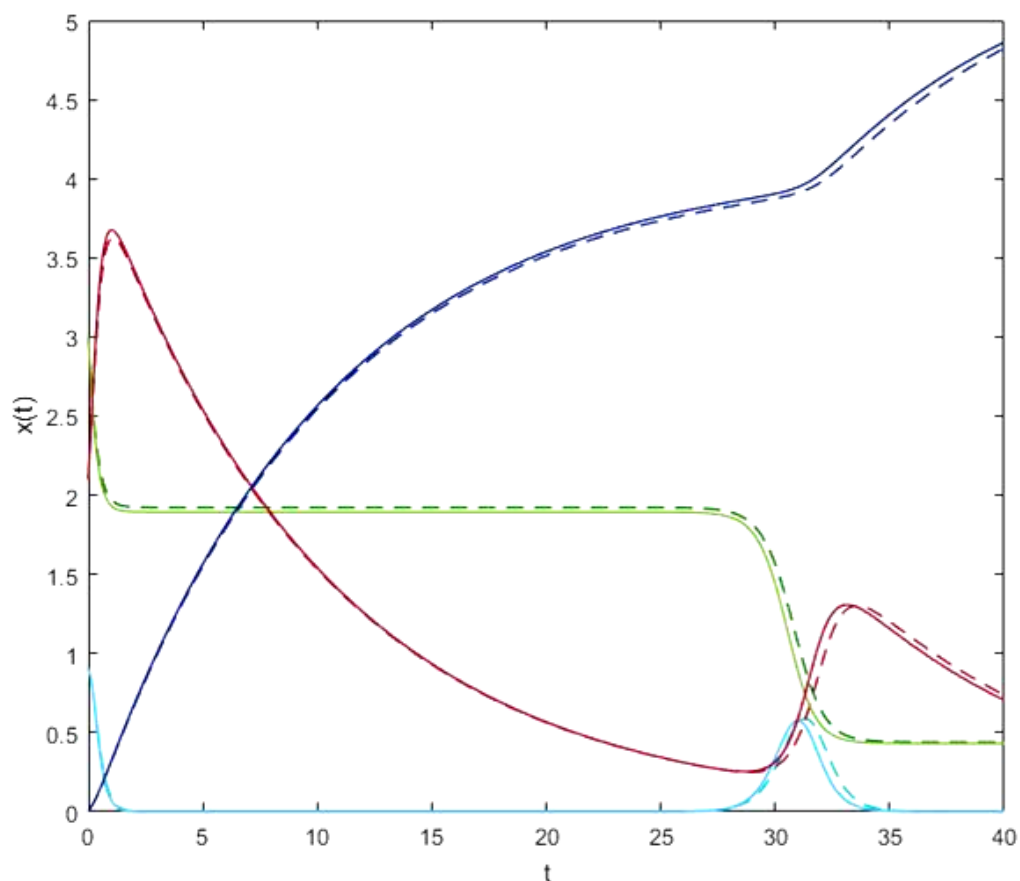


Рис.5 Поведение переменных x_1, x_2, x_3, x_4 при истинных (сплошная линия) и идентифицированных (пунктирная линия) параметрах (идентификация методом GEM)

Из рисунка 5 можно увидеть, что поведение каждой из переменных при истинных и идентифицированных параметрах отличаются незначительно.

Для сравнения результатов был проведен еще один эксперимент в соответствии с методами, предложенными в работе [7]. Результаты этого эксперимента приведены на рисунках 6 – 7.

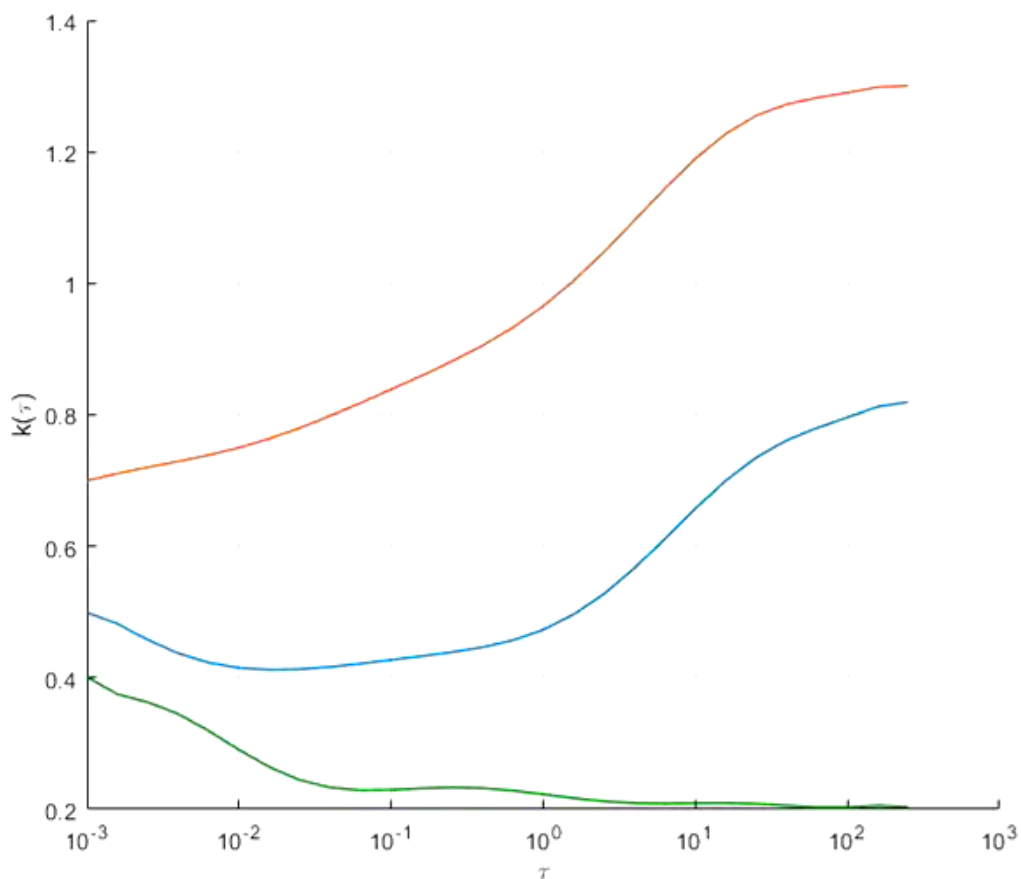


Рис.6 Траектория параметров k_1 , k_2 , k_3 (идентификация методом градиентных уравнений)

Так же, как и в предыдущем эксперименте, была вычислена относительная погрешность, она составила 0.084, или 8.4%. Было подсчитано затраченное на решение задачи процессорное время, которое составило 140 минут. На рисунке 7 сравниваются истинное решение задачи Коши (5), (6) при параметрах (1,1,5,0,1) и решение той же задачи Коши при идентифицированных параметрах.

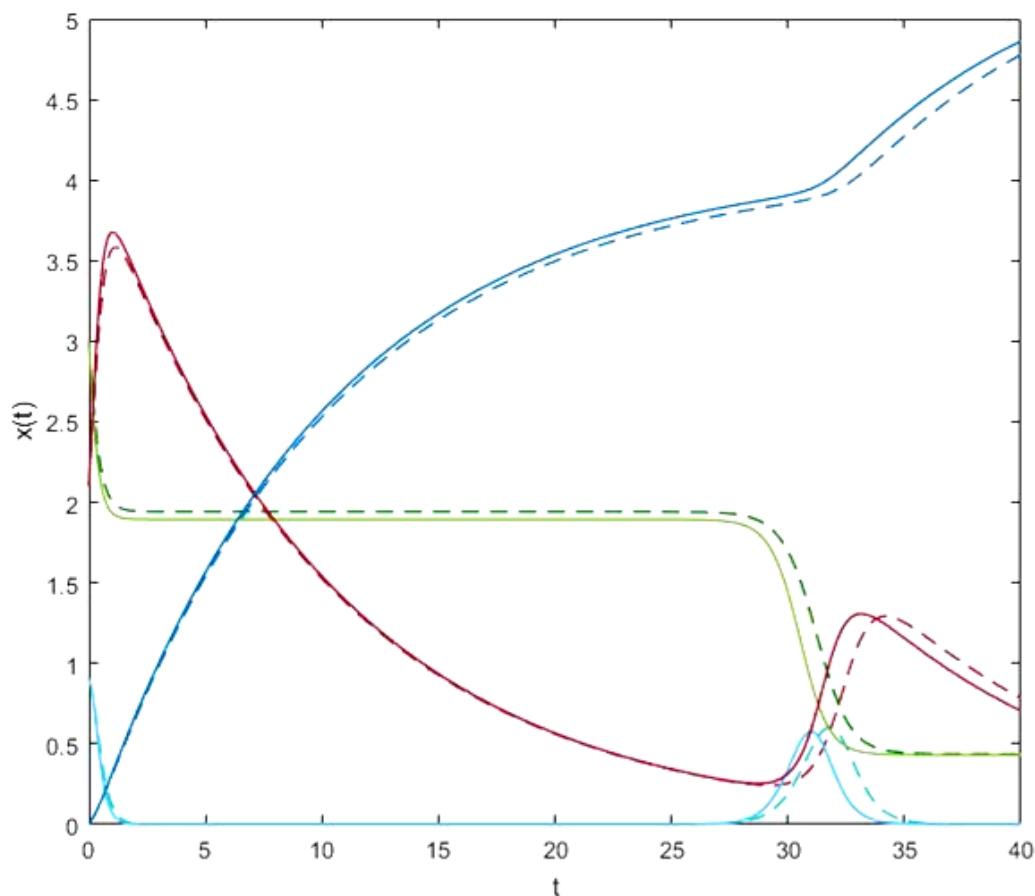


Рис.7 Поведение переменных x_1, x_2, x_3, x_4 при истинных (сплошная линия) и идентифицированных (пунктирная линия) параметрах (идентификация методом градиентных уравнений)

Сравнивая два показателя: скорость и точность, можно сделать вывод о том, что метод GEM превосходит метод градиентных уравнений по обеим этим величинам. И если разница в точности этих двух методов небольшая, т.е. они являются достаточно надежными при решении подобного вида задач, то в скорости метод градиентных уравнений значительно уступает методу GEM.

Заключение.

В настоящей работе были получены следующие результаты:

6. Рассмотрен новый метод минимизации функций многих переменных, сочетающий в себе метод градиентных уравнений и метод четвертой точки.

7. Поставлена задача идентификации параметров модели Лотки-Вольтерры, для которой были составлены уравнения в вариациях и выбран функционал метода наименьших квадратов.

8. Поставленная задача решена в полном соответствии со схемой предложенного метода GEM с использованием пакета MatLab R2017a.

9. В соответствии с методами, предложенными в работе [7], был проведен численный эксперимент, для реализации которого использовался пакет MatLab R2017a.

10. Проведено сравнение результатов обоих экспериментов, выявлена эффективность предложенного в работе метода оптимизации.

Список литературы.

1. Бабаджанянц Л.К. Идентификация параметров осциллирующих процессов в живой природе, моделируемых дифференциальными уравнениями //Проблемы современного естествознания – Биология и Медицина, 2002.
2. Бахвалов Н.С, Жидков Н.П, Кобельков Г.М. Численные методы. //М., Наука, 1987.
3. Вольтерра В. Математическая теория борьбы за существование. М.: Наука, 1976.
4. Гельфанд И.М., Вул Е.Б., Гинзбург С.Л., Федоров Ю.Г. Метод оврагов в задачах рентгеноструктурного анализа. М.: Наука, 1966.
5. Лапчик М.П., Рагулина М.И., Хеннер Е.К. Численные методы: учеб пособие для студ. вузов. М.: Академия, 2004.
6. Цетлин М. Исследования по теории автоматов и моделированию биологических систем. М., 1969.
7. Babadzanjanz L., Boyle J., Sarkissian D., Zhu J. Parameter Identification for Oscillating Chemical Reactions Modeled by Systems of ODE. //Journal of Computational Methods for Sciences and Engineering, 2002.
8. Bower J., Bolouri H. Computational modeling of genetic biochemical networks. //The MIT Press, Cambridge, MA, 2001.
9. Haberman R. Mathematical Models. Mechanical Vibrations, Population Dynamics, and Traffic Flow. Classics in Applied Mathematics, 21. //SIAM, Philadelphia, 1977.
10. Lotka A. Elements of Physical Biology. //Williams and Wilkins, Baltimore, 1925.
11. Murray D. Mathematical Biology. //Springer-Verlag, Berlin, 1993.
12. Rust B., ACMD, Ashton R., Chemical Science and Technology Laboratory. Parameter Identifications. 2001.

13. Schittkowski K. in Yang X. et al. eds., Progress in Optimization.
//Kluwer Academic Publishers, Dordrecht, Boston, 2000.
14. Электронная библиотека методов МГУ.

Приложение 1. Решение задачи идентификации методом GEM

```
function ur=diff0(~, x)
global m k;
ur = [- k(m-1,1)* x(1) * x(2);
k(m-1,1) * x(1) * x(2) - k(m-1,2) * x(2) * x(3);
k(m-1,2) * x(2) * x(3) - k(m-1,3) * x(3);
k(m-1,3) * x(3)];
end
function g = varur(~, y)
global m k x;
g = [- k(m-1,1) * x(2) * y(1) - x(1) * (x(2) + k(m-1,1) * y(4));
- k(m-1,1) * (x(2) * y(2) + x(1) * y(5));
- k(m-1,1) * (x(2) * y(3) + x(1) * y(6));
x(1) * x(2) + k(m-1,1) * x(2) * y(1) - k(m-1,2) * x(2) * y(7) + (k(m-1,1) * x(1) -
k(m-1,2) * x(3)) * y(4);
(k(m-1,1) * x(1) - k(m-1,2) * x(3)) * y(5) - x(2) * (x(3) - k(m-1,1) * y(2) + k(m-
1,2) * y(8));
(k(m-1,1) * x(1) - k(m-1,2) * x(3)) * y(6) + x(2) * (k(m-1,1) * y(3) - k(m-1,2) *
y(9));
k(m-1,2) * x(3) * y(4) + (k(m-1,2) * x(2) - k(m-1,3)) * y(7);
x(2) * x(3) + k(m-1,2) * x(3) * y(5) + (k(m-1,2) * x(2) - k(m-1,3)) * y(8);
x(3) * (k(m-1,2) * y(6) - 1) + (k(m-1,2) * x(2) - k(m-1,3)) * y(9);
k(m-1,3) * y(7);
k(m-1,3) * y(8);
x(3) + k(m-1,3) * y(9)];
end
global m y x x4 h k
while (abs(h(1))>=10^-4)&&(abs(h(2))>=10^-4)&&(abs(h(3))>=10^-4)
```

```

k(m,1:3)=k(m,1:3)+h*10^-10;
m=m+1;
[t, x] = ode15s(@diff0, t0, x0); %решаем систему дифф.уравнений с
заданными параметрами
[t, y] = ode15s(@varur, t0, y0); %решаем систему уравнений в вариациях с
заданными параметрами
m=m-1;
%Находим решения градиентных уравнений
for p = 1:1:3,
gradur = 0;
for l = 1:1:30,
gradur = gradur - abs(x(l, 4) - x4(l)) * y(l, 9 + p);
end
h(p) = gradur;
end
end
%Моделирование наблюдений для случая M = 30
clc, clear all
global m k x4 x;
m=2;
x0 = [3 0.9 2.1 0];
k(1,1:3) = [1 1.5 0.1];
t0 = 5/6:5/6:25;
[t, x] = ode15s (@diff0, t0, x0);
%моделируем наблюдения
q = rand (30, 1);
a = - 0.05;
b = 0.05;
c = a + (b - a) * q;
x4(1:30) = (ones(30,1) + c(1:30))'* x(1:30, 4);

```



```

%Реализация алгоритма
x0 = [3 0.9 2.1 0]; %начальные данные
k(m,1:3)=[0.5 0.7 0.4];
t0 = 1:1:30; %промежутки времени
y0=zeros(1,12); %начальные условия для уравнений в вариациях
[t, x] = ode15s(@diff0, t0, x0); %решение системы дифференциальных
уравнений с заданным начальным приближением параметров
%нахождение функционала, который нужно минимизировать
func = 0;
%задаем цикл для получения значения функционала
for j = 1:1:30
func = func + 1/2 * (x(j, 4) - x4(j))^2 ;
end
f = func; %переменной f присваиваем значение функционала
d(2) =f;
%Находим вторую и третью точки методом градиентных уравнений
for m = 3 : 1 : 4
[t, x] = ode15s(@diff0, t0, x0); %решаем систему дифф.уравнений с
заданными параметрами
[t, y] = ode15s(@varur, t0, y0); %решаем систему уравнений в вариациях с
заданными параметрами
%Находим решения градиентных уравнений
for p = 1:1:3
gradur = 0;
for l = 1:1:30
gradur = gradur - (x(l, 4) - x4(l)) * y(l, 9 + p);
end
h(p) = gradur;
end

```

```

k(m,1:3) = k(m-1,1:3) + h *10^-7; %находим новое приближение k для
параметров
func = 0;
for j = 1:1:30
func = func + 1/2 * (x(j, 4) - x4(j))^2 ;
end
f = func;
d(m) =f;
end
tic
%Находим четвертую и пятую точку
for m = 5:1:6
k(m,1:3) = fourpoint(k(m - 3,:), k(m - 2,:), k(m - 1,:));
[t, x] = ode15s(@diff0, t0, x0);
[t, y] = ode15s(@varur, t0, y0);
for p = 1:1:3
gradur = 0;
for l = 1:1:30
gradur = gradur - (x(l, 4) - x4(l)) * y(l, 9 + p);
end
h(p) = gradur;
end
searchh
func = 0;
for j = 1:1:30
func = func + 1/2 * (x(j, 4) - x4(j))^2 ;
end
f = func; %переменной f присваиваем значение функционала
d(m) =f;
end

```

```

m = 7;
k(m,1:3) = fourpoint(k(m-5,:), k(m-3,:), k(m-1,:));
[t, x] = ode15s(@diff0, t0, x0);
[t, y] = ode15s(@varur, t0, y0);
for p = 1:1:3
gradur = 0;
for l = 1:1:30
gradur = gradur - (x(l, 4) - x4(l)) * y(l, 9 + p);
end
h(p) = gradur;
end
searchh
func = 0;
for j = 1:1:30
func = func + 1/2 * (x(j, 4) - x4(j))^2 ;
end
f = func;
d(m) =f;
while abs(d(m) - d(m-1))>= 10^-6
if d(m) < d(m-1)
    k(m,1:3) = fourpoint (k(m - 5,:), k(m - 3,:), k(m - 1,:));
    searchh
    m = m + 1;
[t, x] = ode15s(@diff0, t0, x0);
func = 0;
for j = 1:1:3
func = func + 1/2 * (x(j, 4) - x4(j))^2 ;
end
f = func;
d(m) =f;

```

```

else
    k(m,1:3) = minimum(k(m - 3,:), k(m - 2,:), k(m-1,:));
    searchh
    m = m + 1;
[t, x] = ode15s(@diff0, t0, x0);
func = 0;
for j = 1:1:30
func = func + 1/2 * (x(j, 4) - x4(j))^2 ;
end
f = func;
d(m) =f;
end
end
toc
r=10^-3:10^-3:(m-1)/10^3;
semilogx(r, k1), %график поведения параметра k1
hold on;
semilogx(r, k2), %график поведения параметра k2
semilogx(r, k3), %график поведения параметра k3
hold off;
[t, x] = ode45(@diff0,t0,x0); %решаем систему с итоговыми параметрами
plot(t, x,'--')
hold on;
k = [1 1.5 0.1]; %верные параметры
[t, x] = ode45(@diff0, t0, x0); %решение системы с верными параметрами
plot(t, x) %графики переменных
hold off

```

Приложение 2. Решение задачи идентификации методом градиентных уравнений.

```
function ur = diff0(t, x);
global k;
ur = [- k(1)*x(1)*x(2);
k(1)*x(1)*x(2)-k(2)*x(2)*x(3);
k(2)*x(2)*x(3)-k(3)*x(3);
k(3)*x(3)];
end
function g = varur(t,y);
global k x;
g=[-k(1)*x(2)*y(1)-x(1)*(x(2)+k(1)*y(4));
-k(1)*(x(2)*y(2)+x(1)*y(5));
-k(1)*(x(2)*y(3)+x(1)*y(6));
x(1)*x(2)+k(1)*x(2)*y(1)-k(2)*x(2)*y(7)-(k(1)*x(1)-k(2)*x(3))*y(4);
(k(1)*x(1)-k(2)*x(3))*y(5)-x(2)*(x(3)-k(1)*y(2)+k(2)*y(8));
(k(1)*x(1)-k(2)*x(3))*y(6)+x(2)*(k(1)*y(3)-k(2)*y(9));
k(2)*x(3)*y(4)+(k(2)*x(2)-k(3))*y(7);
x(2)*x(3)+k(2)*x(3)*y(5)+(k(2)*x(2)-k(3))*y(8);
x(3)*(k(2)*y(6)-1)+(k(2)*x(2)-k(3))*y(9);
k(3)*y(7);
k(3)*y(8);
x(3)+k(3)*y(9)];
end
%Моделирование наблюдений для случая M = 40
clc, clear all
global k x4;
x0 = [3 0.9 2.1 0];
k = [1 1.5 0.1];
```

```

t0 = 1/2:1/2:25;
[t, x] = ode45 (@diff0, t0, x0);
%моделируем наблюдения
q = rand (30, 1);
a = -0.05;
b = 0.05;
c = a + (b - a) * q;
for i=1:1:30
x4(i) = (1 + c(i)) * x(i, 4);
end
%Реализация алгоритма
global k x4 x;
x0 = [3 0.9 2.1 0]; %начальные данные
k = [0.5 0.7 0.4]; %начальные приближения параметров
t0 = 0:2/3:40; %промежутки времени
y0=[0 0 0 0 0 0 0 0 0 0 0]; %начальные условия для уравнений в вариациях
[t, x] = ode45(@diff0, t0, x0); %решение системы дифференциальных
уравнений с заданным начальным приближением параметров
%нахождение функционала, который нужно минимизировать
func = 0;
%задаем цикл для получения значения функционала
for j = 1:1:30
func = func + 1/2 * (x(j, 4) - x4(j))^2 ;
end
f = func, %переменной f присваиваем значение функционала, выводим на
экран
d(1) = f; %первому элементу массива d присваиваем полученное значение
k1(1) = k(1);
k2(1) = k(2);
k3(1) = k(3);

```

```

[t, x] = ode45(@diff0, t0, x0);
%задаем цикл для получения значения функционала
[t, x] = ode45(@diff0, t0, x0); %решаем систему дифф.уравнений с заданными
параметрами
[t, y] = ode45(@varur, t0, y0); %решаем систему уравнений в вариациях с
заданными параметрами
%Находим решения градиентных уравнений
for p = 1:1:3
gradur = 0;
for l = 1:1:30
gradur = gradur - (x(l, 4) - x4(l)) * y(l, 9 + p);
end
h(p) = gradur;
end
u = 10^-7;
func = 0;
for j = 1:1:30
func = func + 1/2 * (x(j, 4) - x4(j))^2 ;
end
f = func;
d(2) = f;
k1(2) = k(1)+ h(1)* 10^-7;
k2(2) = k(2)+ h(2)* 10^-7;
k3(2) = k(3)+ h(3)* 10^-7;
%Задаем цикл для нахождения минимума функционала, считаем время
работы
tic
for m = 3:1:4*10^5
[t, x] = ode45(@diff0, t0, x0);
[t, y] = ode45(@varur, t0, y0);

```

```

for p = 1:1:3
gradur = 0;
for l = 1:1:30
gradur = gradur - (x(l, 4) - x4(l)) * y(l, 9 + p);
end
h(p) = gradur;
end
k1(m) = k1(m-1)+ h(1)* u + 0.5*(k1(m-1)-k1(m-2));
k2(m) = k2(m-1)+ h(2)* u + 0.5*(k2(m-1)-k2(m-2));
k3(m) = k3(m-1)+ h(3)* u + 0.5*(k3(m-1)-k3(m-2));
k =[k1(m) k2(m) k3(m)]
k1(m)=k(1);
k2(m)=k(2);
k3(m)=k(3);
[t, x] = ode45(@diff0, t0, x0);
func = 0;
for l = 1:1:30
func = func + 1/2 * (x(l, 4) - x4(l))^2 ;
end
f=func;
d(m) = f;
if d(m)<d(m-1)
    if (abs(k1(m)-k1(m-1))>=10^-8) || (abs(k2(m)-k2(m-1))>=10^-8) || (abs(k3(m)-
k3(m-1))>=10^-8)
        m = m+1;
    else
        if (abs(k1(m)-k1(m-1))<10^-8) && (abs(k2(m)-k2(m-1))<10^-8) &&
(abs(k3(m)-k3(m-1))<10^-8)
            break;
        end
    end
end

```



```

    end
else
    if d(m)>=d(m-1)
        u = u/2;
        m = m+1;
    end
end
end
end
toc
r=10^-3:10^-3:m/10^3;
semilogx(r, k1), %график поведения параметра k1
hold on;
semilogx(r, k2), %график поведения параметра k2
semilogx(r, k3), %график поведения параметра k3
hold off;
[t, x] = ode45(@diff0,t0,x0); %решаем систему с итоговыми параметрами
plot(t, x,'--')
hold on;
k = [1 1.5 0.1]; %верные параметры
[t, x] = ode45(@diff0, t0, x0); %решение системы с верными параметрами
plot(t, x) %графики переменных
hold off

```