

Санкт-Петербургский государственный университет
Факультет прикладной математики – процессов управления

Бойко Алина Владимировна

Магистерская диссертация

**Применение адаптивного метода
для класса нелинейных задач
оптимального управления**

Направление 01.04.02 «Прикладная математика и информатика»

Основная образовательная программа магистратуры ВМ.5517
«Методы прикладной математики и информатики
в задачах управления»

Научный руководитель
доктор физ.-мат. наук,
профессор Смирнов Н. В.

Рецензент
кандидат физ.-мат. наук,
Шахов Я. А.

Санкт-Петербург

2019

Содержание

Введение. Обзор литературы	4
Постановка задачи	5
Нелинейная задача оптимального управления	5
Глава 1. Методы решения нелинейных задач оптимального управления	6
§1.1. Адаптивный метод	6
Линейная задача оптимального управления	7
Сведение к интервальной задаче линейного программирования	8
Особенности адаптивного метода	10
§1.2. Метод динамического программирования	10
Принцип оптимальности	11
Динамическое программирование	11
Глава 2. Модель макроэкономического роста	12
§2.1. Математическая постановка задачи	12
§2.2. Применение адаптивного метода	15
Сведение к ИЗЛП	16
Первый способ преобразования ограничений	18
Второй способ преобразования ограничений	19
§2.3. Численная реализация	19
§2.4. Сравнение адаптивного метода и метода динамического программирования	22
Применение метода динамического программирования	22

Результаты и выводы	23
§2.5. Сравнение симплекс-метода и адаптивного метода	26
Заключение	27
Список литературы	29
Приложение	32

Введение. Обзор литературы

Математические методы широко используются для моделирования различных систем и процессов. Некоторые параметры моделей поддаются управлению для достижения лучших результатов. В связи с этим теория оптимального управления получила широкое распространение.

Развитие теории оптимального управления связано с такими учеными, как Р. Е. Калман [1], Л. С. Понтрягин [2], В. И. Зубов [3] и многими другими. Теория управления находит применение в различных задачах моделирования экономической динамики. Примеры можно найти в известных монографиях [4,5] и недавних работах [6–10].

Существует множество подходов к поиску оптимального управления, таких как принцип оптимальности Беллмана [4] и принцип максимума Понтрягина [2]. С возникновением вычислительных машин появилась необходимость управления в режиме реального времени, Р. Габасов и его научная школа разработали адаптивный метод [11,12].

В данной работе мы рассмотрим два подхода к решению задач оптимального управления. Первый подход заключается в применении принципа оптимальности и составления дифференциального уравнения Беллмана [5]. Второй подход состоит в сведении задачи оптимального управления к задаче линейного программирования и нахождение решения адаптивным методом Р. Габасова. Во второй части мы рассмотрим на примере решение нелинейной задачи макроэкономического роста с нелинейными ограничениями.

Постановка задачи

Главной задачей является нахождение оптимального программного управления, доставляющее максимум (минимум) заданному функционалу и удовлетворяющего заданным ограничениям. Задачи оптимального управления могут быть как линейными так и не линейными в зависимости от вида исходной системы дифференциальных уравнений и максимизируемого (минимизируемого) функционала. Рассмотрим подробнее математическую постановку задачи.

Нелинейная задача оптимального управления

Рассмотрим общую задачу управления

$$\begin{aligned} \max_{u(t)} J &= \int_{t_0}^{t_1} G(x, u, t) dt + I(x_1, t_1), \\ \dot{x} &= f(x, u, t), \\ x(t_0) &= x_0, \\ x(t_1) &= x_1, \\ u(t) &\in U, \quad t \in [t_0, t_1]. \end{aligned} \tag{1}$$

Здесь управлением является $u(t)$, а фазовой переменной $x(t)$. В общем случае функции $f(x, u, t)$, $G(x, u, t)$ и $I(x, t)$ нелинейны, а система (1) называется *нелинейной задачей оптимального управления*. Множество U предполагается выпуклым, компактным, а также инвариантным относительно времени.

Допустимым управлением называется кусочно-непрерывная вектор-функция времени, значения которой лежат внутри U в любой момент времени $t \in [t_0, t_1]$. Требуется найти такое допустимое управление, удовлетворяющее системе дифференциальных уравнений (1) и доставляющее максимум функционалу J .

Для решения нелинейных задач оптимального управления можно либо свести систему к линейной и применить известные способы нахождения оптимального управления для линейных задач [11], либо погрузить нелинейную задачу в более широкий класс задач и, основываясь на принципе оптимальности, задать рекуррентное соотношение для нахождения искомого управления [5].

Глава 1. Методы решения нелинейных задач оптимального управления

В данной главе описывается 2 подхода к решению нелинейных задач оптимального управления. Первый подход основан на линеаризации системы и дальнейшего применения к ней адаптивного метода, второй основан на принципе оптимальности Беллмана и динамическом программировании.

§1.1. Адаптивный метод

Рассмотрим первый подход к решению нелинейных задач управления. Для начала нелинейная система сводится к линейной посредством линеаризации. Затем линейная задача преобразуется в интервальную задачу линейного программирования (ИЗЛП), к которой применяется адаптивный метод для нахождения оптимального плана.

Процесс линеаризации для нелинейных систем не является универсальным, поскольку функции $f(x, u, t)$, $G(x, u, t)$ и $I(x, t)$ из нелинейной системы (1) произвольны. Но для большинства реальных задач функционал и система дифференциальных уравнений имеют нелинейности, для которых легко подобрать линеаризацию. В главе 2 мы подробно рассмотрим на примере реальной

задачи процесс линеаризации и проведем сравнительный анализ решений адаптивным методом и методом динамического программирования.

Рассмотрим подробнее алгоритм применения адаптивного метода после линеаризации нелинейной задачи оптимального управления.

Линейная задача оптимального управления

Пусть имеется линейная система дифференциальных уравнений с заданными начальными условиями

$$\begin{aligned} \dot{x} &= A(t)x + B(t)u, \\ x(t_0) &= x_0, \end{aligned} \quad (2)$$

$$t \in [t_0, t_1], \quad t_0 < t_1 < +\infty,$$

где $x \in \mathbb{R}^n$, $u \in \mathbb{R}^1$. $A(t)$ — кусочно-непрерывная $(n \times n)$ -матричная функция, а $B(t)$ — n -мерная векторная функции, $t \in [t_0, t_1]$.

Для решения задачи (2) необходимо максимизировать линейную целевую функцию

$$C^T x(t_1) \rightarrow \max_u, \quad (3)$$

Управление $u(t)$ будет задано в классе кусочно-постоянных функций с периодом квантования $h = \frac{t_1 - t_0}{N}$, где N — некоторое натуральное число

$$u(t) = u(t_0 + (i - 1)h) = u_i, \quad t \in [t_0 + (i - 1)h, t_0 + ih), \quad i = \overline{1, N}. \quad (4)$$

Поскольку на практике ресурсы обычно ограничены, закономерно ввести дополнительные ограничения на управление:

$$W_1 \leq u(t) \leq W_2. \quad (5)$$

В задачах оптимального управления могут быть ограничения не только на управляющий параметр, но и на положение системы в конечный момент време-

ни, что часто встречается в экономических задачах. Введем следующие линейные ограничения вида

$$h_1 \leq Kx(t_1) \leq h_2, \quad (6)$$

где $h_1, h_2 \in \mathbb{R}^k$, $k = \text{rang}K < n$.

Таким образом, получили задачу оптимального управления (2)–(6), в которой требуется найти такое допустимое управление $u^0(t)$, что $C^T x(t_1, u^0) \geq C^T x(t_1, u)$ для $\forall u$ из (4).

Сведение к интервальной задаче линейного программирования (ИЗЛП)

Рассмотрим как задача (2)–(6) может быть сведена к ИЗЛП. Выпишем формулу Коши для линейной однородной системы дифференциальных уравнений из (2)

$$x(t_1, t_0, x_0) = Y(t_1) \left(Y^{-1}(t_0)x_0 + \int_{t_0}^{t_1} Y^{-1}(\tau)B(\tau)u(\tau)d\tau \right), \quad (7)$$

где $Y(t)$ — фундаментальная матрица системы (2), нормированная в точке t_0 ($Y(t_0) = E$). Подставим (7) в максимизируемый функционал (3):

$$C^T Y(t_1)Y^{-1}(t_0)x_0 + \int_{t_0}^{t_1} C^T Y(t_1)Y^{-1}(\tau)B(\tau)u(\tau)d\tau \longrightarrow \max_u.$$

Первое слагаемое можно отбросить поскольку оно не зависит от управления $u(t)$ и является константой, не меняющей характер стремления функционала к максимуму. Поскольку управление является кусочно-постоянной вектор-функцией, максимизируемое относительно управления выражение преобразуется в следующий вид

$$\sum_{i=1}^N \int_{t_0+(i-1)h}^{t_0+ih} C^T Y(t_1)Y^{-1}(\tau)B(\tau)d\tau u_k \longrightarrow \max_u.$$

Умножим граничные условия (7) слева на матрицу K , получим:

$$Kx(t_1, t_0, x_0) = KY(t_1) \left(Y^{-1}(t_0)x_0 + \int_{t_0}^{t_1} Y^{-1}(\tau)B(\tau)u(\tau)d\tau \right).$$

Учитывая вид управления (4) граничные условия (6) преобразуются в следующую систему неравенств относительно u_i :

$$\begin{aligned} h_1 - KY(t_1)Y^{-1}(t_0)x_0 &\leq \sum_{i=1}^N \int_{t_0+(i-1)h}^{t_0+ih} KY(t_1)Y^{-1}(\tau)B(\tau)d\tau u_i \leq \\ &\leq h_2 - KY(t_1)Y^{-1}(t_0)x_0. \end{aligned}$$

Введём следующие обозначения:

$$\begin{aligned} s_i &= \int_{t_0+(i-1)h}^{t_0+ih} C^T Y(t_1)Y^{-1}(\tau)B(\tau)d\tau, \quad i = \overline{1, N}, \quad S = (s_1, \dots, s_N)^T, \\ \mu_i &= \int_{t_0+(i-1)h}^{t_0+ih} KY(t_1)Y^{-1}(\tau)B(\tau)d\tau, \quad i = \overline{1, N}, \quad M = (\mu_1, \dots, \mu_N), \\ U &= (u_1, \dots, u_N)^T, \end{aligned}$$

$$V_1 = h_1 - KY(t_1)Y^{-1}(t_0)x_0, \quad V_1 \in \mathbb{R}^m.$$

$$V_2 = h_2 - KY(t_1)Y^{-1}(t_0)x_0, \quad V_2 \in \mathbb{R}^m.$$

Здесь U и S — N -мерные векторы, M — $(m \times N)$ -матрица.

Таким образом нам удалось свести исходную линейную задачу оптимального управления (2)—(6) к ИЗЛП следующего вида

$$S^T U \rightarrow \max_u,$$

$$V_1 \leq MU \leq V_2, \tag{8}$$

$$W_1 \leq u_i \leq W_2, \quad i = \overline{1, N}.$$

Для решение интервальной задачи линейного программирования применяется адаптивный метод Габасова. Рассмотрим подробнее особенности применения адаптивного метода.

Особенности адаптивного метода

Для нахождения оптимального плана в задачах линейного программирования обычно используют симплекс-метод, но в данной работе мы рассмотрим адаптивный метод, имеющий ряд преимуществ перед симплекс-методом. Прежде всего адаптивный метод не требует введения новых переменных и увеличения размерности задачи. Метод работает с любыми точками множества допустимых планов, а не только с вершинами симплекса, именно это позволяет быстрее строить оптимальный план.

При реализации алгоритма адаптивного метода на каждой итерации выбирается направление и длина шага вдоль выбранного направления по определенным правилам. В адаптивном методе помимо плана существует понятие опоры. Опорой называется подмножества строк и столбцов матрицы основных ограничений (для задачи (8) подмножества матрицы M). Опора не связана с планом и меняется независимо от него, что также является преимуществом в ходе реализации алгоритма.

Для задач большой размерности, использование адаптивного метода более эффективно по быстродействию, чем симплекс-метода. В главе 2 на примере будет проведен сравнительный анализ эффективности адаптивного и симплекс-метода. Более подробно об алгоритмах адаптивного метода в книгах Р. Габасова [11, 13].

§1.2. Метод динамического программирования

Рассмотрим еще один подход к решению нелинейных задач оптимального управления с помощью метода динамического программирования. Данный подход основан на принципе оптимальности Беллмана, из которого выводится основное

рекуррентное соотношение с заданными граничными условиями.

Принцип оптимальности

Максимальное значение функционала J с начальным состоянием x и начальным временем t из (1) называется *функцией оптимального поведения* и обозначается $J^*(x, t)$, а $J^* = J^*(x_0, t_0)$.

Основное рекуррентное соотношение для нелинейной задачи оптимального управления (1) выглядит следующим образом (см [5])

$$J^*(x, t) = \max_{u(t)} [G(x, u, t)\Delta t + J^*(x + \Delta x, t + \Delta t)],$$

где Δt — приращение времени, а Δx — приращение фазовой переменной.

Если предположить, что функция оптимального поведения является однозначной и непрерывно дифференцируемой функцией, то разложение ее в ряд Тейлора даст следующее соотношение

$$-\frac{dJ^*}{dt} = \max_{u(t)} \left[G(x, u, t) + \frac{dJ^*}{dx} f(x, u, t) \right].$$

Это соотношение называется *уравнением Беллмана*.

Динамическое программирование

Как и в §1.1. время рассматривается как дискретная величина с шагом h

$$t_0, t_0 + h, t_0 + 2h, \dots, t_1, \quad h > 0.$$

$x(t)$ — состояние системы в момент времени t , $u(t)$ — управление системы в момент t , тогда состояние системы в момент $t + h$ задается

$$x(t + h) = f(x(t), u(t), t), \quad t = \overline{t_0, t_1 - h}.$$

Таким образом, получим рекуррентное соотношение

$$J^*(x, t) = \max_{u(t)} [G(x(t), u(t), t) + J^*(f(x(t), u(t), t), t + h)], \quad (9)$$

с граничным условием

$$J^*(x_1, t_1) = I(x_1, t_1).$$

Решая рекуррентное соотношение (9), найдем последовательность управлений

$$\{u(t_0), u(t_0 + h), \dots, u(t_1 - h)\},$$

каждое из которых лежит в заданном множестве U

$$u(t) \in U, \quad t = \overline{t_0, t_1 - h}$$

и максимизирует целевую функцию

$$J = \sum_{i=t_0}^{t_1-h} G(x(i), u(i), i) + I(x_1, t_1).$$

Глава 2. Модель макроэкономического роста

В данной главе рассматривается применение адаптивного метода, на примере неоклассической модели макроэкономического роста. Приведен сравнительный анализ решения адаптивным методом и методом динамического программирования. Для полученной ИЗЛП проведено сравнение времени работы алгоритмов симплекс-метода и адаптивного метода.

§2.1. Математическая постановка задачи

Рассмотрим агрегированную замкнутую экономику, все затраты и весь выпуск находятся внутри системы в любой момент времени t [11, 14]. Согласно уравнению баланса, суммарный выпуск P состоит из потребления C и инвестиций I

$$P = C + I.$$

Величины P , C и I могут быть выражены в единицах национальной валюты.

Инвестиции обеспечивают рост капитала и компенсируют его амортизацию. Обозначим $K(t)$ значение капитала в момент времени t , тогда $\dot{K}(t)$ — скорость изменения капитала. Предположим, что амортизация капитала происходит пропорционально капиталу и равна $\mu K(t)$, где μ — норма амортизации. Получим дифференциальную систему уравнений модели экономического роста

$$I = \dot{K} + \mu K. \quad (10)$$

Следующим важным элементом модели экономического роста являются производственные функции. Производственная функция связывает величину произведенной продукции с факторами производства (капитал, трудовые ресурсы и т.д.).

$$P = F(K, L),$$

здесь $L = L(t)$ — трудовые ресурсы в момент времени t .

В качестве производственной функции была выбрана динамизированная функция Кобба-Дугласа [15, 16]

$$P(t) = a_0 L(t)^{\alpha_1} K(t)^{\alpha_2} e^{bt}, \quad (11)$$

где a_0, α_1, α_2 положительные константы, $\alpha_1, \alpha_2 \in (0, 1)$, и b — коэффициент технологического прогресса.

Предположим, что трудовые ресурсы растут с постоянной скоростью

$$\dot{L} = rL, \quad (12)$$

где r — темп роста трудовых ресурсов.

Начальное значение величины трудовых ресурсов в точке t_0 задается, как $L(t_0) = L_0$. Начальное значение капитала в точке t_0 , как $K(t_0) = K_0$

Введем также значение капитала в конечный момент времени $K(t_1)$

$$K(z) = K_1.$$

Поскольку потребление $C(t)$ в любой момент времени не может превышать выпуск $P(t)$, получаем следующие ограничения на потребление

$$0 \leq C(t) \leq P(t), \quad t \in [t_0, t_1].$$

Требуется максимизировать сумму потребления за весь период времени $[t_0, t_1]$ и капитала в конечный момент времени $K(t_1)$.

Перепишем (10), учитывая уравнение баланса

$$\dot{K} = I - \mu K = P - C - \mu K, \quad (13)$$

$$t \in [t_0, t_1], \quad K(t_0) = K_0.$$

Поскольку ресурсы растут с постоянной скоростью, выпишем решение дифференциальной системы (12)

$$L(t) = L_0 e^{rt}.$$

Тогда производственная функция Кобба-Дугласа примет вид

$$P(t) = \xi(t)K(t)^{\alpha_2}, \quad (14)$$

где $\xi(t) = a_0 L_0^{\alpha_1} e^{(\alpha_1 r + b)t}$.

Подставим преобразованную нелинейную производственную функцию (14) в дифференциальную систему (13)

$$\dot{K}(t) = \xi(t)K(t)^{\alpha_2} - C(t) - \mu K(t). \quad (15)$$

Подставим производственную функцию (14) в ограничения на управления, получим следующие нелинейные неравенства

$$0 \leq C(t) \leq \xi(t)K(t)^{\alpha_2}.$$

Целевой функционал представляет собой сумму потребления за весь период времени $[t_0, t_1]$ и капитала в конечный момент времени t_1

$$J = e^{-\delta t_1} K_1 + \int_{t_0}^{t_1} e^{-\delta t} C(t) dt,$$

где δ — коэффициент дисконтирования.

Окончательно задача макроэкономического роста примет вид

$$J = e^{-\delta t_1} K_1 + \int_{t_0}^{t_1} e^{-\delta t} C(t) dt \rightarrow \max_{C(t)}, \quad (16)$$

$$\dot{K}(t) = \xi(t)K(t)^{\alpha_2} - \mu K(t) - C(t), \quad (17)$$

$$K(t_0) = K_0, \quad K(t_1) = K_1,$$

$$0 \leq C(t) \leq \xi(t)K(t)^{\alpha_2}, \quad t \in [t_0, t_1]. \quad (18)$$

Полученная система может рассматриваться как нелинейная задача оптимального управления, управлением здесь будет являться потребление $C(t)$, а фазовой переменной — капитал $K(t)$. Требуется найти такую пару $(K(t), C(t))$, удовлетворяющую системе (17) и доставляющую максимум функционалу (16).

Задача может быть решена одним из рассмотренных в Главе 1 способом. Рассмотрим подробнее подход, основанный на адаптивном методе.

§2.2. Применение адаптивного метода

Для применения адаптивного метода нам необходимо линеаризовать систему (16)—(18) и привести ее к линейной задаче управления. (см. §1.1.).

Рассмотрим дифференциальную систему (17). В ней присутствует нелинейность $K(t)^{\alpha_2}$, $t \in [t_0, t_1]$, которую требуется линеаризовать.

Поскольку производственная функция и ее элемент $K(t)^{\alpha_2}$ достаточно гладкие, т.к. $\alpha_2 \in (0, 1)$, в качестве линеаризации выберем отрезок прямой

$\omega K(t) + \zeta$. Коэффициенты ω , ζ могут быть найдены из системы

$$\begin{cases} \omega K_0 + \zeta = K_0^{\alpha_2}, \\ \omega K_1 + \zeta = K_1^{\alpha_2}. \end{cases}$$

Получим

$$\omega = \frac{K_0^{\alpha_2} - K_1^{\alpha_2}}{K_0 - K_1},$$

$$\zeta = K_0^{\alpha_2} - \omega K_0.$$

После замены нелинейного элемента в системе (17), получим

$$J = e^{-\delta z} K(z) + \int_0^z e^{-\delta t} C(t) dt \rightarrow \max, \quad (19)$$

$$\dot{K}(t) = A(t)K(t) - C(t) + F(t),$$

$$K(0) = K_0, \quad K(z) = K_1,$$

$$0 \leq C(t) \leq \xi(t)K(t)^{\alpha_2}, \quad t \in [0, z],$$

где

$$A(t) = \omega \xi(t) - \mu, \quad F(t) = \zeta \xi(t).$$

Заметим, что нелинейность присутствует также в ограничениях на управление.

Линеаризацию ограничений подробнее рассмотрим ниже.

Сведение к ИЗЛП

Далее преобразуем линейную модель экономического роста (19) в ИЗЛП. Для этого выпишем формулу Коши линейной однородной системы

$$K(t, t_0, K_0) = Y(t) \left[Y^{-1}(t_0) K_0 + \int_{t_0}^t Y^{-1}(\tau) (F(\tau) - C(\tau)) d\tau \right], \quad (20)$$

где

$$Y(t) = e^{\int_{t_0}^t A(\tau) d\tau}.$$

Подставим (20) в линейную систему (19)

$$J = e^{-\delta t_1} Y(t_1) \left[Y^{-1}(t_0) K_0 + \int_{t_0}^z Y^{-1}(\tau) (F(\tau) - C(\tau)) d\tau \right] + \int_{t_0}^{t_1} e^{-\delta t} C(t) dt \rightarrow \max, \quad (21)$$

$$Y(t_1) \left[Y^{-1}(t_0) K_0 + \int_{t_0}^{t_1} Y^{-1}(\tau) (F(\tau) - C(\tau)) d\tau \right] = K_1. \quad (22)$$

Отбросим все слагаемые без управления $C(t)$ в функционале (21)

$$\int_{t_0}^{t_1} (e^{-\delta \tau} - e^{-\delta t_1} Y(t_1) Y^{-1}(\tau)) C(\tau) d\tau \rightarrow \max. \quad (23)$$

Преобразуем (22)

$$Y(t_1) \int_{t_0}^{t_1} Y^{-1}(\tau) C(\tau) d\tau = Y(t_1) \left[Y^{-1}(t_0) K_0 + \int_{t_0}^{t_1} Y^{-1}(\tau) F(\tau) d\tau \right] - K_1. \quad (24)$$

Учитывая вид управления (4) разделим интегралы (23), (24) на части

$$\int_{t_0}^{t_0+h} (e^{-\delta \tau} - e^{-\delta t_1} Y(t_1) Y^{-1}(\tau)) d\tau C_1 + \int_{t_0+h}^{t_0+2h} (e^{-\delta \tau} - e^{-\delta t_1} Y(t_1) Y^{-1}(\tau)) d\tau C_2 + \dots + \int_{t_1-h}^{t_1} (e^{-\delta \tau} - e^{-\delta t_1} Y(t_1) Y^{-1}(\tau)) d\tau C_N \rightarrow \max \quad (25)$$

$$Y(t_1) \left[\int_{t_0}^{t_0+h} Y^{-1}(\tau) d\tau C_1 + \int_{t_0+h}^{t_0+2h} Y^{-1}(\tau) d\tau C_2 + \dots + \int_{t_1-h}^{t_1} Y^{-1}(\tau) d\tau C_N \right] = Y(t_1) \left[Y^{-1}(t_0) K_0 + \int_{t_0}^{t_1} Y^{-1}(\tau) F(\tau) d\tau \right] - K_1. \quad (26)$$

Перепишем (25), (26) в векторной форме

$$S^T C \rightarrow \max, \quad (27)$$

$$MC = V, \quad (28)$$

где $C = C_{N \times 1} = (C_1, \dots, C_N)^T$ - искомый вектор управления,

$$S = S_{N \times 1} = (S_h(t_0 + h), \dots, S_h(t_0 + Nh))^T,$$

$$\begin{aligned}
S_h(t) &= \int_{t-h}^t (e^{-\delta\tau} - e^{-\delta t_1} Y(t_1) Y^{-1}(\tau)) d\tau, \\
M &= M_{1 \times N} = (M_h(t_0 + h), \dots, M_h(t_0 + Nh)), \\
M_h(t) &= Y(t_1) \int_{t-h}^t Y^{-1}(\tau) d\tau, \\
V &= -K_1 + Y(t_1) \left(Y^{-1}(t_0) K_0 + \int_{t_0}^{t_1} F(\tau) Y^{-1}(\tau) d\tau \right).
\end{aligned}$$

Для ограничений на управление в (19) мы можем также использовать аппроксимацию отрезком прямой, либо формулу Коши для нелинейности. Рассмотрим два способа приведения ограничений к стандартному виду прямых ограничений ИЗЛП.

Первый способ преобразования ограничений

Подставим формулу Коши (20) в (18)

$$\begin{aligned}
0 \leq C(t) \leq \xi(t) Y(t)^{\alpha_2} \left[Y^{-1}(t_0) K_0 + \int_{t_0}^t F(\tau) Y^{-1}(\tau) d\tau - \int_{t_0}^t Y^{-1}(\tau) C(\tau) d\tau \right]^{\alpha_2} \leq \\
\xi(t) Y(t)^{\alpha_2} \left[Y^{-1}(t_0) K_0 + \int_{t_0}^t F(\tau) Y^{-1}(\tau) d\tau \right]^{\alpha_2}, \quad (29)
\end{aligned}$$

последнее неравенство выполняется поскольку $C_i \geq 0$, $i = \overline{1, N}$.

Запишем ограничения в векторной форме

$$0 \leq C \leq W, \quad (30)$$

где

$$\begin{aligned}
W &= W_{N \times 1} = (W_h(t_0), \dots, W_h(t_0 + (N-1)h))^T, \\
W_h(t) &= \xi(t) \left(Y(t) \left(Y^{-1}(t_0) K_0 + \int_{t_0}^t F(\tau) Y^{-1}(\tau) d\tau \right) \right)^{\alpha_2}.
\end{aligned}$$

Тогда система (16), (17) и (30) представляет собой ИЗЛП.

Второй способ преобразования ограничений

Во втором случае мы можем сначала заменить нелинейность $K(t)^{\alpha_2}$ отрезком прямой, а затем подставить формулу Коши (20).

$$0 \leq C(t) \leq \xi(t)(\omega K(t) + \zeta). \quad (31)$$

$$0 \leq C(t) \leq \omega \xi(t) Y(t) \left[Y^{-1}(t_0) K_0 + \int_{t_0}^t F(\tau) Y^{-1}(\tau) d\tau - \int_{t_0}^t Y^{-1}(\tau) C(\tau) d\tau \right] + \zeta \xi(t).$$

Последнее слагаемое в квадратных скобках перепишем в виде

$$\int_{t_0}^t Y^{-1}(\tau) C(\tau) d\tau = \int_{t_0}^{t_0+h} Y^{-1}(\tau) d\tau C_1 + \\ + \int_{t_0+h}^{t_0+2h} Y^{-1}(\tau) d\tau C_2 + \dots + \int_{t-h}^t Y^{-1}(\tau) d\tau C_m, \quad (32)$$

где $m = \lceil \frac{t}{h} \rceil$. Получим ограничения аналогичные (30).

Используя один из двух способов линеаризации, можно получить стандартную форму прямых ограничений ИЗЛП. Мы будем использовать первый способ, поскольку он проще и не использует линеаризацию.

§2.3. Численная реализация

Основываясь на алгоритме применения адаптивного метода был разработан пакет программ в среде MATLAB для нахождения оптимального управления. В качестве тестовых параметров были выбраны следующие значения:

No.	Параметр	Значение	No.	Параметр	Значение
1	N	60	7	K_1	$1,1K_0$
2	a_0	1	8	r	0,1
3	α_1	0,3	9	t_0	0
4	α_2	0,2	10	t_1	5
5	L_0	2	11	μ	0,05
6	K_0	5	12	δ	0,09

Полученные значения аппроксимации $\omega \approx 0,05$, $\zeta \approx 1,11$.

В качестве меры погрешности аппроксимации выбрана норма разности на отрезке времени $[t_0, t_1]$ $e \approx 0,002$.

На рисунке 1 изображена аппроксимация нелинейности $K(t)^{\alpha_2}$. Пунктирной линией изображен отрезок прямой $\omega K + \zeta$, а сплошной линией нелинейность $K(t)^{\alpha_2}$ на отрезке $[t_0, t_1]$. Как видно из графика благодаря гладкости нелинейного элемента нам удалось довольно просто аппроксимировать ее отрезком прямой. Для повышения точности аппроксимации можно использовать ломаную линию.

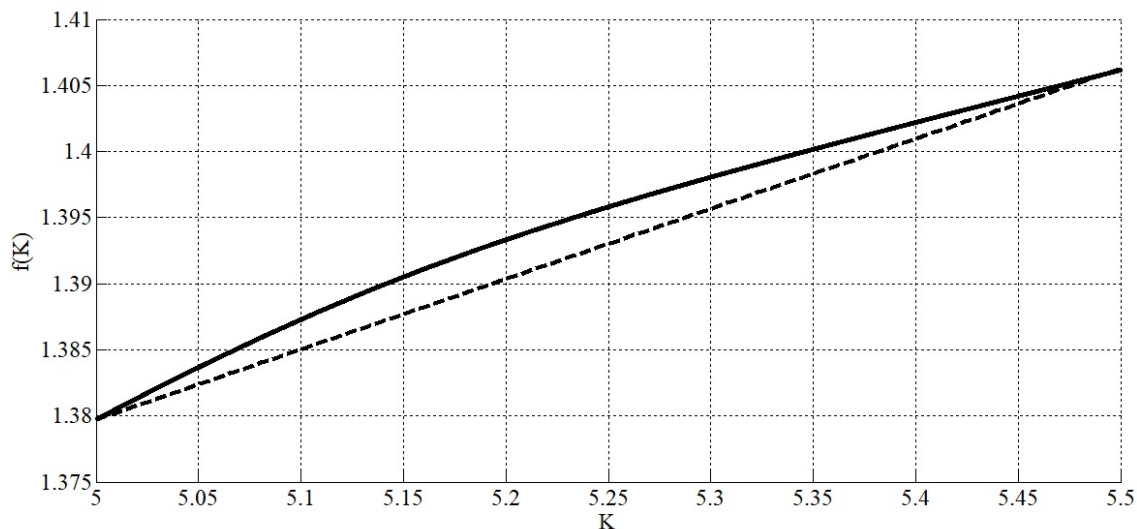


Рис. 1. Аппроксимация нелинейности

Полученное управление, изображенное на графике 2, является кусочно-постоянной функцией. Исходя из графика, искомое потребление $C(t)$ возрастает с течением времени.

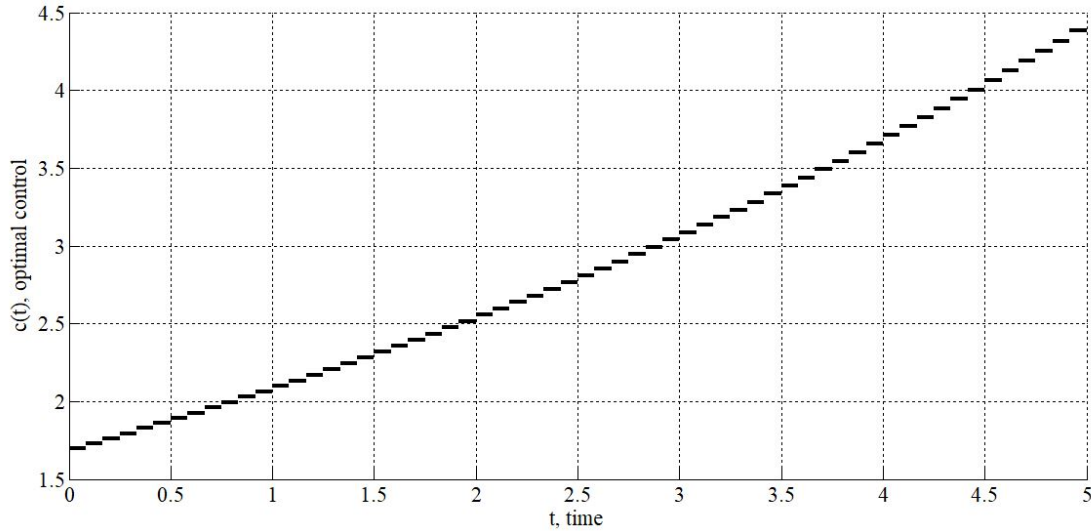


Рис. 2. Оптимальное управление

Нам удалось найти оптимальное управление для нелинейной системы (16) — (18), являющееся искомым потреблением $C(t)$. Особенностью этой задачи является нелинейная производственная функция. Нелинейность удалось достаточно просто аппроксимировать в соответствующей области фазового пространства.

В то же время ограничения в ИЗЛП оказались динамическими, что необходимо было учитывать при разработке программного кода. Программный пакет, созданный в среде MATLAB, позволяет нам строить оптимальное управление для многих нелинейных задач. Мы также можем рассмотреть эту модель с другими производственными функциями, что приводит к целому классу нелинейных задач, решаемых с помощью адаптивного метода. Меняться будет только способ линеаризации системы.

§2.4. Сравнение адаптивного метода и метода динамического программирования

Рассмотрим модель экономического роста с линейно-однородной функцией Кобба-Дугласа (при $\alpha_1 + \alpha_2 = 1$) и коэффициенте технологического прогресса $b = 0$.

В этом случае система выглядит следующим образом

$$\begin{aligned}
 J &= e^{-\delta t_1} K(t_1) + \int_{t_0}^{t_1} e^{-\delta t} C(t) dt \rightarrow \max, \\
 \dot{K}(t) &= \xi(t) K(t)^{1-\alpha_1} - \mu K(t) - C(t), \\
 K(t_0) &= K_0, \quad K(t_1) = K_1, \\
 0 &\leq C(t) \leq \xi(t) K(t)^{1-\alpha_1}, \\
 \xi(t) &= a_0 L_0^{\alpha_1} e^{\alpha_1 r t}, \quad t \in [t_0, t_1].
 \end{aligned} \tag{33}$$

Применение метода динамического программирования

Найдем оптимальное управление с помощью метода динамического программирования.

В условиях задачи основное рекуррентное соотношение примет вид

$$J^*(K, t) = \max_{C(t)} [e^{-\delta t} C(t) + J^*([\xi(t) K(t)^{1-\alpha_1} - \mu K(t) - C(t)], t + h)] \tag{34}$$

С граничным условием

$$J^*(K_1, t_1) = e^{-\delta t_1} K_1.$$

Рекуррентное соотношение (34) позволяет найти значение функционала J^* при оптимальном управлении $C(t)$.

Пусть количество шагов метода $N = \frac{(t_1 - t_0)}{h}$ равно двум. Тогда, учитывая

(34) и вид управления, получим

$$\begin{aligned} J^*(K(t_1 - h), t_1 - h) &= \max_{C(t_1 - h)} [e^{-\delta(t_1 - h)} C(t_1 - h) + \\ &+ e^{-\delta t_1} (\xi(t_1 - h) K(t_1 - h)^{1 - \alpha_1} - \mu K(t_1 - h) - C(t_1 - h))] = \\ &= e^{-\delta t_1} [e^{\delta h} \xi(t_1 - h) K(t_1 - h)^{1 - \alpha_1} - \mu K(t_1 - h)]. \end{aligned}$$

Управление на первом шаге равно

$$C(t_1 - h) = \xi(t_1 - h) K(t_1 - h)^{1 - \alpha_1}.$$

На втором шаге

$$J^*(K(t_0), t_0) = \max_{C(t_0)} [\psi C(t_0) + \theta (\eta - C(t_0))^{1 - \alpha_1} - \epsilon],$$

где $\psi = e^{-\delta t_0} + \mu e^{-\delta t_1}$, $\theta = e^{(t_1 - h)(\alpha_1 r - \delta)} a_0 L_0^{\alpha_1}$,

$\eta = \xi(t_0) K_0^{1 - \alpha_1} - \mu K_0$, $\epsilon = \mu e^{-\delta t_1} (\xi(t_0) K_0^{1 - \alpha_1} - \mu K_0)$.

Управление на втором шаге равно

$$C(t_0) = \eta - \left(\frac{\psi}{\theta(1 - \alpha_1)} \right)^{-\frac{1}{\alpha_1}}.$$

Результаты и выводы

Для численной реализации выберем параметры из §2.3., при $N = 2$, а $\alpha_1 = 0,8$. Подставим параметры в рекуррентные уравнения (34) и систему (33), найдем управление двумя способами, методом динамического программирования и адаптивным методом.

Полученные значения оптимального управления приведены на рисунке 3. Пунктирная линия - управление, построенное при помощи адаптивного метода, сплошная линия - управление, построенное методом динамического программирования.

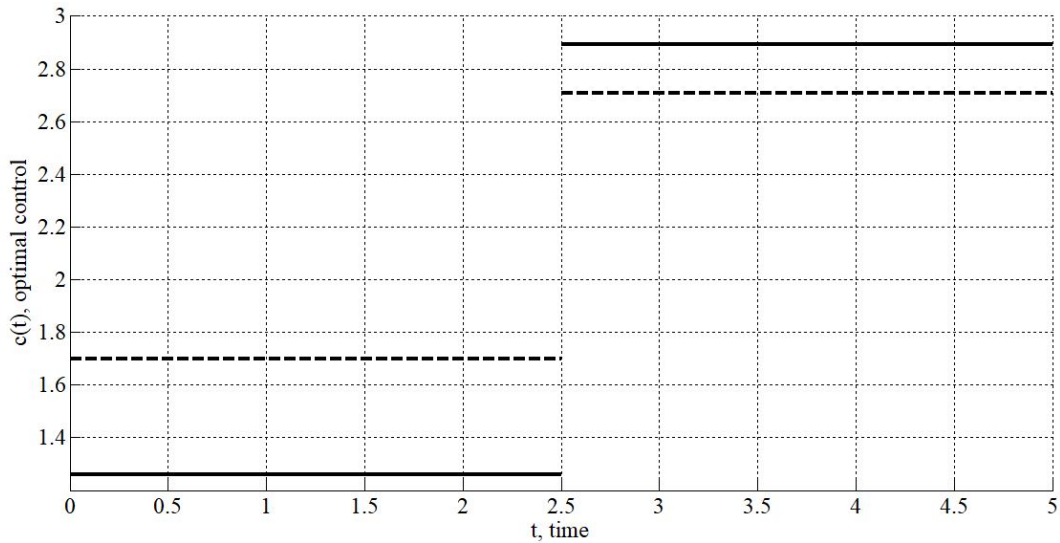


Рис. 3. Оптимальное управление метода динамического программирования и адаптивного метода

Полученные целевые функции, при соответствующих управлениях изображены на рисунке 4. Пунктирная линия — целевой функционал J^* для адаптивного метода, сплошная линия — целевой функционал для метода динамического программирования.

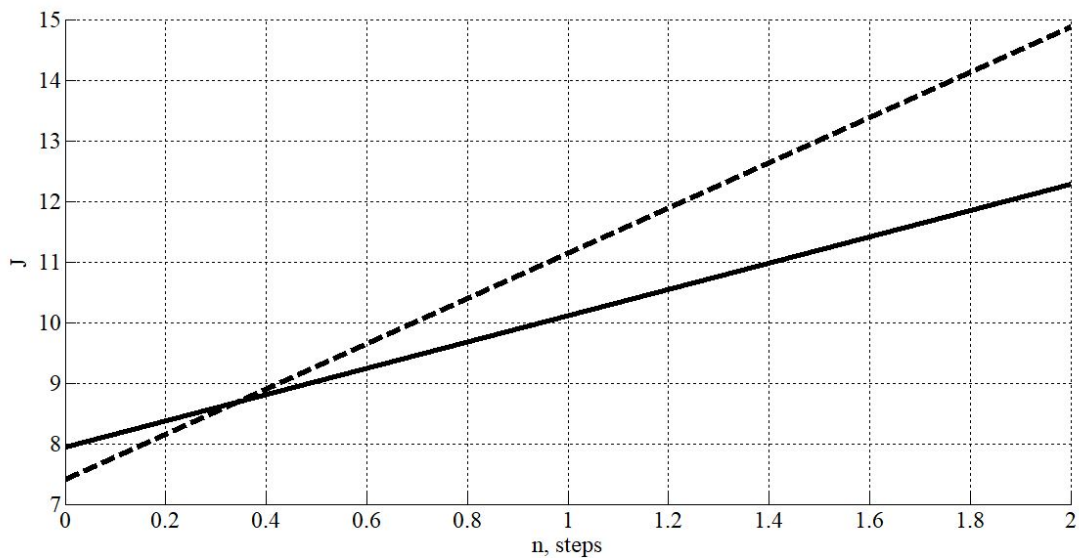


Рис. 4. График целевого функционала J

На примере решения модели макроэкономического роста мы провели сравнительный анализ двух методов нахождения оптимального управления для нелинейных задач. На построенных графиках видно, что управления достаточно близки, но не совпадают, а целевая функция в случае адаптивного метода по значению в конечной точке превосходит построенную методом динамического программирования.

При применении адаптивного метода задача сводится к более простой задаче ИЗЛП, увеличение числа рассматриваемых интервалов времени N не ведет к усложнению системы, в методе динамического программирования каждый новый шаг увеличивает сложность вычислений. В то же время метод динамического программирования не требует линеаризации задачи и применяется к нелинейной системе, что уменьшает ошибки при линеаризации. Оба метода находят оптимальное управление, однако в случае большого числа интервалов времени использование динамического программирования порождает сложные вычисления. Правильная линеаризация системы позволяет использовать более простой в реализации алгоритм решения.

Во многих задачах может не быть точно задано значение системы в конечный момент времени. В рамках рассмотренной задачи капитал может быть больше, чем предполагаемое значение K_1 , и мы имеем неравенство в ограничении на фазовую переменную. В таком случае использование метода динамического программирования затрудняется, даже при методе прямой прогонки нам точно не известно в какую точку система должна попасть. При переходе к задаче линейного программирования таких проблем нет.

§2.5. Сравнение симплекс-метода и адаптивного метода

После сведения системы (33) к ИЗЛП вида (16), (17) и (30) для сравнения работы адаптивного метода и симплекс-метода посчитаем оптимальный план с помощью встроенных функций MATLAB и измерим время работы алгоритмов. В таблице 2 приведено время работы алгоритма адаптивного метода и симплекс-метода, а также время сведения системы (33) к ИЗЛП.

Н, число шагов	Время работы Адаптивного метода,сек	Время работы Симплекс-метода,сек	Время сведения к ИЗЛП, сек
2	0.01	0.02	0.49
5	0.01	0.13	1.13
10	0.01	0.13	2.36
20	0.02	0.3	4.41
30	0.03	0.13	6.35
40	0.04	0.12	8.15
60	0.12	0.18	12.08

Как видно из таблицы 2 адаптивный метод работает быстрее симплекс-метода, что упрощает нахождение управления в режиме реального времени. Таким образом, можно сказать, что при правильной линеаризации системы использование адаптивного метода для нелинейных задач оптимального управле-

ния более эффективно.

Заключение

В ходе работы над данным проектом:

1. Изучено два подхода к решению нелинейных задач оптимального управления: адаптивный метод, использующий линеаризацию и метод динамического программирования, основанный на принципе оптимальности Беллмана.
2. Применение адаптивного метода рассмотрено на примере задачи макроэкономического роста. Построено оптимальное управление для рассмотренной задачи, результаты приведены на графиках.
3. Проведен сравнительный анализ метода динамического программирования и адаптивного метода, а также симплекс- метода для построения оптимального плана в ИЗЛП.
4. Разработан пакет программ в среде MATLAB для реализации всех этапов рассмотренного в §1.1. подхода, фрагменты которого приведены в приложении.
5. Результаты работы докладывались на следующих конференциях:
 - XLIX международная научная конференция аспирантов и студентов «Процессы управления и устойчивость» Control Processes and Stability (CPS'18);
 - Международная научная конференция "Динамические системы: устойчивость, управление, оптимизация"(DSSCO'18) к 100-летию со дня рождения Е.А. Барбашина, 24–29 сентября 2018 года, г. Минск (Беларусь);
 - 3rd International Conference on Applications in Information Technology, November 1-3, 2018, Aizu-wakamatsu.

- Международная конференция "Game theory and management 2019"GTM'19.
Санкт-Петербург 03 - 05 июля 2019. Статья принята к публикации.
- 13th International Symposium on Intelligent Distributed Computing. IDC'19.
Санкт-Петербург 7-9 октября 2019. Статья принята к публикации.

6. Результаты частично опубликованы в работах [21–23].

Список литературы

1. Калман Р. Е. Об общей теории систем управления. Труды I Междунар. конгресса ИФАК. М.: Изд-во АН СССР, 1961. Т. 2. С. 521–547.
2. Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. Математическая теория оптимальных процессов. М.: Наука, 1969. 384 с.
3. Zubov V. I. Математические методы исследования систем автоматического регулирования. Л.: Машиностроение, 1974. 336 с.
4. R. Bellman and R. Kalaba. Dynamic Programming and Modern Control Theory. New York. 1965. Academic Press.
5. M.D. Intriligator. Mathematical Optimization and Economic Theory. Englewood Cliffs, New Jersey. 1971. Prentice-Hall, Inc.
6. W.R. Esposito. Dynamic Programming: Continuous-time Optimal Control. Encyclopedia of Optimization. Springer Boston, MA. 2008. URL:https://doi.org/10.1007/978-0-387-74759-0_146.
7. N.V. Smirnov and T.E. Smirnova and K.M. Volik and V.P. Peresada. Modelling of investment programs based on the impulse program controls. 2015 International Conference on “Stability and Control Processes” in Memory of V. I. Zubov SCP 2015. 494–497. doi:10.1109/SCP.2015.7342182.
8. W. Xu and Z. Wang and L. Hong and L. He and X. Chen. The uncertainty recovery analysis for interdependent infrastructure systems using the dynamic inoperability input-output model. International Journal of Systems Science. Volume 46. Number 7. 2015. p. 1299–1306. doi: 10.1080/00207721.2013.822121.

9. M. Dombi. Modeling the material stock of manufactured capital with production function. *Resources, Conservation and Recycling*. Volume 138. 2018. Pages 207–214. doi: 10.1016/j.resconrec.2018.07.015.
10. B. Gong. Agricultural reforms and production in China: Changes in provincial production function and productivity in 1978–2015. *Journal of Development Economics*. Volume 132. 2018. Pages 18–31. doi: 10.1016/j.jdeveco.2017.12.005.
11. Альсевич В. В., Габасов Р., Глушенков В. С. Оптимизация линейных экономических моделей. Минск: Изд-во БГУ, 2000. 211 с.
12. Балашевич Н. В., Габасов Р., Кириллова Ф. М. Численные методы программной и позиционной оптимизации линейных систем управления// Журн. вычисл. математики и мат. физики. 2000. Вып. 40, № 6. С. 838–859.
13. Габасов Р. Методы оптимизации: пособие. Минск: Четыре четверти, 2011. 472 с.
14. J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton 1953. Princeton University Press
15. R. Shephard. *Theory of Cost and Production Functions*. Princeton, NJ 1970. Princeton University Press.
16. Л.Л. Терехов. Производственные функции. Москва 1974. Издательский до "Статистика".
17. Беллман Р., Гликсберг И., Гросс О. Некоторые вопросы математической теории процессов управления. М.: ИЛ, 1962. 336 с.
18. Петросян Л. А., Захаров В. В. Математические модели в экологии. СПб.:Изд-во С.-Петербург. ун-та, 1997. 253 с.

19. Попков А. С., Баранов О. В. Об оптимальном управлении вращательным движением вала электродвигателя // Процессы управления и устойчивость. 2014. Т. 1. № 1. С. 31–36.
20. Ключенков А. Л. Реализация адаптивного метода в одной задаче оптимального управления // Процессы управления и устойчивость. 2015. Т. 2. № 1. С. 53–58.
21. Бойко А. В., Зубаков А. В. Применение адаптивного метода в неоклассической модели экономического роста // Процессы управления и устойчивость. 2016. Т. 3. № 1. С. 607–611.
22. Бойко А. В. Применение адаптивного метода к задаче оптимального распределения капитальных вложений в отрасли // Процессы управления и устойчивость. 2017. Т. 4. № 1. С. 581–585.
23. Boiko. A.V., Smirnov. N.V. Approach to optimal control in the economic growth model with a non-linear production function. ACM International Conference Pro-ceeding Series. 1 November 2018, Pages 85-89. ICAIT'2018 Proceedings of the 3rd International Conference on Applications in Information Technology.[doi:10.1145/3274856.3274874](https://doi.org/10.1145/3274856.3274874).

Приложение

В приложении приведен код функций метода (MatLab).

Алгоритм адаптивного метода

Сведение к ИЗЛП

```
1 function [ C,D,g0,R,t0,t1,N,K ] = izlp
2 syms t
3 L0=2;
4 k_0=5;
5 n=0.1; %темп роста L
6 z=5;
7 k_1=double(k_0*1.1);
8 alpha_1=0.3;
9 alpha_2=0.2;
10 p=0.1;
11 a_0=1; %коэффициент Кобба-Дугласса
12 [a,b]=linearization(k_0,k_1,alpha_2);
13 mu=0.05;
14 betta=a_0*(L0^alpha_1)*exp((alpha_1*n+p)*t);
15 delta=0.09;
16 A=betta*a-mu; %после линеаризации
17 N=60;
18 t0=0;
19 t1=z;
20 f=betta*b;
21 c=exp(-delta*z);
22 x=exp(-delta*t);
23 h=(t1-t0)/N;
24 Y=exp(int(A,0,t));
25 Z=Y^(-1);
26 psi=alpha_1*n+p;
27 ksi=(a_0*a*L0^alpha_1)/psi;
28 dzeta=a_0*L0*b;
29
30 Z1=@(w)(exp(ksi+mu*w-ksi*(exp(psi*w))));
31 Zf=@(w)(exp(ksi+mu*w-ksi*(exp(psi*w))+psi*w))*dzeta;
32
33 g0=double(-k_1+subs(Y,t,t1)*(subs(Z,t,0)*k_0+quad(Zf,0,t1)));
34
35 for i=1:N
36     D(:,i)=double(subs(Y,t,t1))*quad(Z1,(i-1)*h,i*h);
37 end
```



```

38
39 for i=1:N
40     C(i,:) = int(x, (i-1)*h, i*h) - c*(subs(Y,t,t1))*quad(Z1,(i-1)*h, i*h);
41 end
42
43 for i=1:N
44     R(i,:) = subs(betta, t, (i-1)*h)*(double(subs(Y,t,(i-1)*h))^(alpha_2))*
45     *(double(subs(Z,t,0))*k_0+quad(Zf,0,(i-1)*h))^(alpha_2);
46 end
47
48
49 for i=1:N
50     ZR=@(w)(exp(ksi+mu*w-ksi*(exp(psi*w)))*R(i));
51     k(i) = double(subs(Y,t,(i-1)*h)*(subs(Z,t,0))*k_0+
52     +quad(Zf,0,(i-1)*h)-quad(ZR,0,(i-1)*h));
53 end
54
55 C=double(C)
56 D=double(D)
57 g0=double(g0)
58 R=double(R)
59 \end {lstlisting}
60 \subsection*{Первый шаг алгоритма}
61 \vskip -1.25 cm
62 \begin{lstlisting}
63 function [U, Delta] = step1(c, A, IOP, JOP, INOP, JNOP)
64
65 [M, N] = size(A);
66
67 U(1:M, 1) = 0;
68 Delta(1:N, 1) = 0;
69
70 [tmp, NOP] = size(IOP);
71
72 Atran = A';
73
74 Arev = A(IOP, JOP)^-1;
75
76 U(INOP) = 0;
77 Delta(JNOP) = c(JNOP);
78
79 if NOP > 0
80     crev = (c(JOP));
81     U;
82     Delta;
83     U(IOP) = Arev' * crev;
84     Delta(JOP) = 0;
85

```

```

86     Delta(JNOP) = Delta(JNOP) - (A(IOP, JNOP))' * U(IOP);
87 end
88
89 end

```

Второй шаг алгоритма

```

1  function [opt, k0, flag] = step2(x, A, IOP, JNOP, DN, DV, BN, BV, U, Delta)
2
3  [tmp, NOP] = size(IOP);
4  [tmp, NNOP] = size(JNOP);
5
6
7
8  for p = 1 : 1 : NNOP
9
10     % index stolbca v opore
11     j = JNOP(p);
12     if (x(j) == DN(j) && Delta(j) >= 0) || (x(j) == DV(j) && Delta(j) <= 0) ||
13     || (x(j) > DN(j) && x(j) < DV(j) && Delta(j) ~= 0)
14         opt = false;
15         k0 = p;
16         flag = 2;
17         return
18     end
19 end
20
21 for p = 1 : 1 : NOP
22
23     %index sroki v opore
24     i = IOP(p);
25     %A(i,J)*x <=> A(i, :) * x
26     tmp = A(i,:) * x;
27     if (tmp == BN(i) && U(i) >= 0) || (tmp == BV(i) && U(i) <= 0) ||
28     || (tmp > BN(i) && tmp < BV(i) && U(i) ~= 0) %?
29         opt = false;
30         k0 = p;
31         flag = 1;
32         return
33     end
34 end
35
36
37 opt = true;
38 flag = 0;
39 k0 = 0;
40
41 end

```

Третий шаг алгоритма

```
1 function [L] = step3(flag, k0, A, IOP, JOP, JNOP, U, Delta)
2 %flag = 1 if k = i0 from IOP
3 %flag = 2 if k = j0 from JNOP
4
5 [M, N] = size(A);
6 L = zeros(N, 1);
7 AOP = A(IOP, JOP);
8 Arev = AOP ^ -1;
9 if AOP == 0
10     Arev=0;
11 end
12 if flag == 2
13     j0 = JNOP(k0);
14     L(j0) = sign(Delta(j0));
15     L(JOP) = - Arev * A(IOP, j0) * L(j0);
16
17 elseif flag == 1
18     i0 = IOP(k0);
19     L(i0) = sign(U(i0));
20     L(JOP) = - Arev * L(IOP);
21 end
```

Четвертый шаг алгоритма

```
1 function [Tetak, zamOpFlag, k] = step4(x, A, IOP, JOP, INOP, JNOP, L, DN, DV, BN, BV, flag, k0)
2 [tmp, NOP] = size(JOP);
3 [tmp, NNOP] = size(INOP);
4
5 TetaI = [];
6 TetaJ = [];
7 for p = 1:1:NOP
8     k = JOP(p);
9     TmpTeta = Inf;
10    if L(k) > 0
11        TmpTeta = (DV(k) - x(k)) / L(k);
12    end
13    if L(k) < 0
14        TmpTeta = (DN(k) - x(k)) / L(k);
15    end
16    TetaJ = [TetaJ [TmpTeta; k]];
17 end
18
19 if flag == 2
20     k = JNOP(k0);
21     TmpTeta = Inf;
22     if L(k) > 0
```

```

23         TmpTeta = (DV(k) - x(k)) / L(k);
24     end
25     if L(k0) < 0
26         TmpTeta = (DN(k) - x(k)) / L(k);
27     end
28     TetaJ = [TetaJ [TmpTeta; k]];
29 end

```

Шестой шаг алгоритма

```

1  function [NEWIOP, NEWJOP, NEWINOP, NEWJNOP] = step6(IOP, JOP, INOP, JNOP, M, N, flag, zamOpFlag, k0, k)
2  I = 1:1:M;
3  J = 1:1:N;
4
5
6  if flag == 2 && zamOpFlag == 2
7      disp(1)
8      NEWIOP = IOP;
9      I(NEWIOP) = [];
10     NEWINOP = I;
11
12     JOP(k) = [];
13     NEWJOP = sort([JOP JNOP(k0)]);
14     J(NEWJOP) = [];
15     NEWJNOP = J;
16 end
17
18 if flag == 2 && zamOpFlag == 1
19     disp(2)
20     NEWIOP = sort([IOP INOP(k)]);
21
22     NEWJOP = sort([JOP JNOP(k0)]);
23     I(NEWIOP) = [];
24     J(NEWJOP) = [];
25     NEWINOP = I;
26     NEWJNOP = J;
27 end
28
29 if flag == 1 && zamOpFlag == 2
30     disp(3)
31     IOP(k0) = [];
32     NEWIOP = IOP;
33     I(NEWIOP) = [];
34     NEWINOP = I;
35     JOP(k) = [];
36     NEWJOP = JOP;
37     J(NEWJOP) = [];
38     NEWJNOP = J ;

```

```

39 end
40
41 if flag == 1 && zamOpFlag == 1
42     disp(4)
43     IOP(k0) = [];
44     NEWIOP = sort([IOP INOP(k)]);
45     I(NEWIOP) = [];
46     NEWINOP = I;
47
48     NEWJOP = JOP;
49     J(NEWJOP) = [];
50     NEWJNOP = J;
51
52 end
53
54 end

```

Main

```

1 while 1
2     opt = false;
3     zamOpFlag = false;
4     Tetak = 0;
5     L = zeros(n, 1);
6     k0 = 1;
7     k = 1;
8
9     [U, Delta] = step1(c, A, IOP, JOP, INOP, JNOP);
10    [opt, k0, flag] = step2(x, A, IOP, JNOP, DN, DV, BN, BV, U, Delta);
11    if opt
12        break
13    end
14    [L] = step3(flag, k0, A, IOP, JOP, JNOP, U, Delta);
15    [Tetak, zamOpFlag, k] = step4(x, A, IOP, JOP, INOP, JNOP, L, DN, DV, BN, BV, flag, k0);
16    x = x + Tetak * L;
17
18    if zamOpFlag ~= 0
19        [IOP, JOP, INOP, JNOP] = step6(IOP, JOP, INOP, JNOP, M, N, flag, zamOpFlag, k0, k);
20    end
21    %проверки
22    if (A*x<=BV)
23        ('ok')
24    else
25        break
26    end
27 end

```