

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА МОДЕЛИРОВАНИЯ ЭКОНОМИЧЕСКИХ СИСТЕМ

Мынзу Лариса Игоревна

Выпускная квалификационная работа бакалавра

Компьютерное моделирование работы цепи поставок

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,
кандидат физ.-мат. наук,
доцент

Трофимова И. В.

Санкт-Петербург

2016

Содержание

Введение. Обзор литературы	3
Глава 1. Постановка задачи.....	6
1.1 Описание модели	6
1.2 Математическая модель	7
1.3 Постановка задачи	8
1.4 Формирование вспомогательной задачи оптимального управления	9
Глава 2. Методы	12
2.1 Сведение задачи оптимального управления к интервальной задаче линейного программирования.....	12
2.2 Адаптивный метод.....	14
2.3 Алгоритм.....	24
Глава 3. Программная реализация.....	25
3.1 Примеры	26
Заключение. Выводы	29
Список литературы	30
Приложение	32

Введение. Обзор литературы

В современном мире управление цепями поставок является ключевым стратегическим фактором для повышения организационной эффективности работы компании и для более полной реализации организационных целей таких, как повышение конкурентоспособности, улучшение обслуживания клиентов и увеличение прибыльности. Управление цепочками поставок включает в себя все виды деятельности, необходимые для успеха предприятия, как, например, управление объектами поставок, оптимизация работы производителей, складов, транспортной сети, улучшение взаимодействия между элементами цепи и так далее [1].

Технологически производственно-логистические процессы постоянно модифицируются, что приводит к появлению как новых требований к качеству работы, так и новых видов управления и организации предприятиями.

Так, например, в 60-е гг. начался переход от ненасыщенного «рынка производителя» к насыщенному «рынку потребителя» [2]. После первоначального насыщения рынка существенную роль стали играть требования потребителя к качеству товаров, что проявилось в концепции всеобщего управления качеством. Возникающие при этом дополнительные затраты компенсировались за счет безупречного качества продукции. Непосредственным влиянием этой тенденции на производство и логистику стали требования к сокращению длительности производственного цикла, низкому уровню запасов, высокой надежности в соблюдении сроков поставки, а также к гибкости производства и логистики, при этом количество вариантов видов продукции росло чрезвычайно быстро.

В 80-90-е гг. закономерно произошел следующий виток развития промышленности и логистики: эффект скорости [2]. Потребителю стало важно получать продукцию в точно оговоренные сроки. Данному периоду

соответствует развитие концепций Lean Production («бережливое, гибкое» производство) и Just-in-Time (точно-во-время). Они продолжают развиваться и в настоящее время.

Новый виток развитию в области управления цепями поставок придали корпоративные информационные системы, а позже и Интернет-технологии, позволившие существенно повысить эффективность координации процессов в цепи поставок. Информационные технологии явились, с одной стороны, средой обеспечения и повышения эффективности цепи поставок, а с другой стороны - мощным инструментом развития новых концепций управления цепями поставок.

С развитием IT- технологий, управление цепями поставок стало развиваться стремительнее и сейчас оно приобретает все большую значимость для промышленных, логистических и торговых предприятий.

По мере развития «рынков клиента», глобализации и информационных технологий, значение цепей поставок как ключевого фактора повышения доходности и конкурентоспособности бизнеса непрерывно возрастает. Именно поэтому управление цепями поставок стремительно развивается, а инвестиции в цепи поставок будут увеличиваться и в будущем.

Сегодня на российском рынке представлено огромное количество как западных, так и отечественных программных продуктов, созданных для управления работой двух основных типов цепей поставок: централизованными и децентрализованными. В децентрализованной цепи поставок решения об оптимизации своей работы принимаются на каждом уровне автономно, независимо от других участников цепи. В централизованной же, напротив, существует единый инструмент, отслеживающий работу всех участников цепи одновременно и принимающий решение на основе общей ситуации [3]. Децентрализованное управление зачастую не отражают полной картины работы цепи поставок. Однако вопрос

о степени децентрализации цепи поставок может быть решен по-разному для различных компаний в определенных ситуациях. Компьютерное моделирование работы цепи поставок различной структуры и с перечисленными способами управления в ней может помочь производителю в решении этой проблемы.

В данной работе проводится компьютерное моделирование выбранной централизованной цепи поставок для нахождения оптимального выпуска с использованием методов теории управления [4,5], в частности адаптивного метода [6].

Глава 1. Постановка задачи

1.1 Описание модели

Рассмотрим цепь поставок, состоящую из трех элементов: производителя (завода), склада и клиента (Рис. 1). При этом под заводом будем понимать один или несколько заводов одного производителя, которые изготавливают только один тип продукции, выпускающийся с необходимой мощностью. Товар с завода поступает на склад, откуда в соответствии с уровнем спроса поставляется потребителю. Потребителем здесь может выступать как оптовый производитель, дилерские компании, так и конечные потребители. Будем рассматривать работу данной цепи во времени.

Следуя работе [7], рассмотрим математическую модель, описывающую работу такой обобщенной цепи поставок. На основе данной модели авторы определяют желаемую скорость производства и ограничения для скорости износа, обеспечивающую асимптотическую устойчивость данной модели с неопределенной скоростью износа.

В работах [8], [9] частные случаи данной модели использовались для описания работы цепей поставок различной структуры (в случаях с централизованным и децентрализованным управлениями) во времени, для управления их работой используются другие методы построения управлений.

В данной работе, используя такую модель, будем искать оптимальную скорость производства, обеспечивающую заданный уровень спроса, с учетом поступающей информации о запасах продукции на складе.

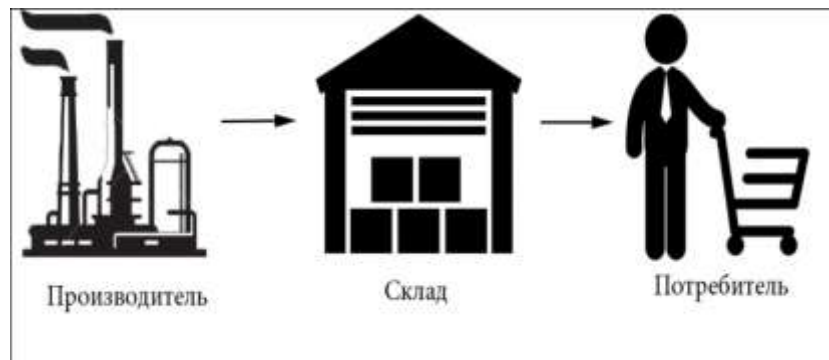


Рис. 1. Визуальное представление модели

1.2 Математическая модель

Опишем математическую модель динамики работы цепи поставок [7].

Пусть $I(t)$ – уровень запасов в момент времени t ,

$P_a(t)$ – фактическая скорость производства в момент времени t ,

$D(t)$ – уровень спроса в момент времени t ,

$u(t)$ – желаемый уровень производства в момент времени t ,

$\theta(t)$ – коэффициент износа,

α – обратное экспоненциальное время задержки

Все функции неотрицательны, непрерывны и дифференцируемы. Тогда можно составить следующую систему дифференциальных уравнений первого порядка:

$$\begin{cases} \dot{I}(t) = P_a - \theta(t)I(t) - D(t) \\ \dot{P}_a(t) = \alpha(u(t) - P_a(t)) \end{cases}$$

Приведя эту систему к виду $\dot{x}(t) = \hat{A}(t)x(t) + b(t)u(t) + f(t)$,

где $x = x(t)$ – есть n -вектор состояния динамической системы в момент времени;

$\hat{A}(t)$ – матрица $n \times n$;

$b(t)$ – n -мерная векторная функция;

$u(t)$ – значение скалярного управляющего воздействия;

$f(t)$ – кусочно-непрерывное ограниченное возмущение

получим

$$\frac{d}{dt} \begin{pmatrix} I \\ P_a \end{pmatrix} = \begin{pmatrix} -\theta & 1 \\ 0 & -\alpha \end{pmatrix} \begin{pmatrix} I \\ P_a \end{pmatrix} + \begin{pmatrix} 0 \\ \alpha \end{pmatrix} u(t) + \begin{pmatrix} -D(t) \\ 0 \end{pmatrix}, \quad (1)$$

где $\begin{pmatrix} I \\ P_a \end{pmatrix} = x(t)$, $\begin{pmatrix} -\theta(t) & 1 \\ 0 & -\alpha \end{pmatrix} = \hat{A}(t)$, $\begin{pmatrix} 0 \\ \alpha \end{pmatrix} = b(t)$, $\begin{pmatrix} -D(t) \\ 0 \end{pmatrix} = f(t)$

Параметры модели могут быть получены путем идентификации модели, опираясь на статистическую информацию о спросе, объемах выпуска, объемах запасов на складе за предыдущие периоды работы цепи. Входные параметры модели определяются с некоторой точностью, поэтому производителю для принятия решения о величине выпуска могут помочь результаты численных экспериментов для различных значений параметров, которые позволяет проводить компьютерное моделирование работы цепи поставок.

1.3 Постановка задачи

Рассмотрим цепь поставок (Рис. 1), состоящую из производителя, склада и потребителя. Пусть у производителя есть возможность получать информацию об уровне запасов на складе и сопоставлять ее с фактической скоростью производства в фиксированные моменты времени. В эти моменты производитель может менять стратегию производства (управлять величиной выпуска), чтобы обеспечить ожидаемый уровень спроса и максимизировать прибыль в конце периода планирования. Поскольку планируемый объем спроса может отличаться от текущего, то производителю приходится регулярно корректировать работу цепи. Будем считать, что управление

u принадлежит классу кусочно-постоянных функций, то есть производитель может менять объем выпуска в определенные моменты времени.

Таким образом, целью является нахождение оптимального выпуска (управления u) в цепи, описываемой уравнением (1), обеспечивающего ожидаемый уровень спроса, заданного ограничениями

$$Hx(t^*) = g, \quad g \in R^m, \text{rang } H = m < n \quad (2)$$

и максимизирующего прибыль, описываемую

$$c'x(t) \rightarrow \max. \quad (3)$$

А так же проведение численных экспериментов для различных значений параметров.

1.4 Формирование вспомогательной задачи оптимального управления

Рассмотрим (1) с постоянными $\hat{A}(t) = \hat{A}, b(t) = b$ и заранее определенными $\bar{x}, \bar{u}, \bar{f}$. Так как в уравнении присутствует возмущение $f(t)$, составим уравнение в отклонениях.

Для этого введем замену переменных:

$$y(t) = x(t) - \bar{x},$$

$$v(t) = u(t) - \bar{u},$$

$$q(t) = f(t) - \bar{f}.$$

И отнимем от уравнения

$$\dot{x}(t) = \hat{A}x(t) + bu(t) + f(t)$$

уравнение

$$\dot{\bar{x}} = \hat{A}\bar{x} + b\bar{u}(t) + \bar{f}(t).$$

Полученное уравнение

$$\dot{y}(t) = \hat{A}y(t) + bv(t) + q(t) \quad (4)$$

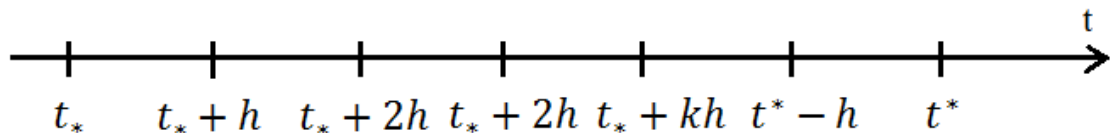
есть уравнение в отклонениях.

Итак, рассмотрим (2) с начальными условиями:

$$y(t_*) = y_0 \quad (5)$$

на заданном промежутке $t \in [t_*, t^*] = T, t_* < t^* < +\infty$.

Зададим класс допустимых управлений: отрезок T разобьем на N частей $N + 1$ точкой следующим образом:



$h = \frac{t^* - t_*}{N}$ – период квантования, N – натуральное число. При этом будем считать, что в полученные моменты времени поступает информация о векторе состояний системы.

Ограниченную кусочно-постоянную функцию $v(t), t \in T$ назовем дискретно - управляющим воздействием, если

$$\begin{cases} v(t) = v(t_* + kh) = v_{kh}, t \in [t_* + kh, t_* + (k + 1)h[\\ k = 0, N + 1 \end{cases} \quad (6)$$

И, кроме того, $|v(t)| \leq l$.

Обозначим множество точек разбиения $T_v \triangleq \{t_*, t_* + h, \dots, t_* + kh, t^* - h\}$.

Таким образом, можно сформулировать вспомогательную задачу оптимального управления:

$$\dot{y}(t) = \hat{A}y(t) + bv,$$

$$c'y(t^*) \rightarrow \max,$$

$$Hy(t^*) = g, \tag{7}$$

$$|v(t)| \leq l,$$

$$y(t_i) = y_i, i = \overline{1, N}$$

Решение данной задачи можно разбить на два последовательных шага:

- сведение задачи оптимального управления к интервальной задаче линейного программирования
- решение задачи линейного программирования адаптивным методом

Глава 2. Методы

2.1 Сведение задачи оптимального управления к интервальной задаче линейного программирования

Для начала сведем (7) к канонической задаче [4,5]: запишем решение уравнения в (7), используя формулу Коши решения линейного дифференциального уравнения:

$$y(t, t_*, y_0) = Y(t)[Y^{-1}(t_*)y_0 + \int_{t_*}^t Y^{-1}(\tau)bv(\tau)d\tau], \quad (8)$$

где $Y(t)$ – фундаментальная матрица соответствующей однородной линейной системы. Подставляя $t = t^*$ в (8), получим:

$$y(t^*) = y(t^*, t_*, y_0) = Y(t^*)[Y^{-1}(t_*)y_0 + \int_{t_*}^{t^*} Y^{-1}(\tau)bv(\tau)d\tau] \quad (9)$$

Подставим формулу (9) в целевую функцию (7):

$$c'y(t^*) = c'Y(t^*)[Y^{-1}(t_*)y_0 + \int_{t_*}^{t^*} Y^{-1}(\tau)bv(\tau)d\tau] \quad (10)$$

Так как первое слагаемое в (10) не зависит от управления, то целевая функция (7) будет эквивалентна:

$$\int_{t_*}^{t^*} c'Y(t^*)Y^{-1}(\tau)bv(\tau)d\tau \rightarrow \max$$

Поскольку $v(t)$ ищется в классе ограниченных кусочно-постоянных функций, то:

$$\sum_{t \in T_v} \int_t^{t+h} c'Y(t^*)Y^{-1}(\tau)bv(\tau)d\tau \rightarrow \max \quad (11)$$

Рассмотрим ограничение в задаче (7) с учетом (9), оно представится в виде

$$HY(t^*)[Y^{-1}(t_*)y_0 + \int_{t_*}^{t^*} Y^{-1}(\tau)bv(\tau)d\tau] = g$$

Вектор $y_0(t) = Y(t)Y^{-1}(t_*)y_0$ является решением уравнения в (7) с начальными условиями $y(t_*) = y_0$ при $v(t) = 0, t \in T$. Тогда последнее условие можно записать в виде

$$\int_{t_*}^{t^*} HY(t^*)Y^{-1}(\tau)bv(\tau)d\tau = g_0,$$

где $g_0 = g - Hy_0(t^*)$ – m -мерный вектор

Аналогично:

$$\sum_{t \in T_v} \int_t^{t+h} HY(t^*)Y^{-1}(\tau)bv(\tau)d\tau = g_0 \quad (12)$$

Очевидно, что можно вычислить заранее интегралы по элементарным промежуткам. Введем для них следующие обозначения:

$$\tilde{c}_h(t) = \int_t^{t+h} c'Y(t^*)Y^{-1}(\tau)bd\tau \quad (13)$$

$$d_h(t) = \int_t^{t+h} HY(t^*)Y^{-1}(\tau)bd\tau, \quad (14)$$

где $t \in \{t_*, t_* + h, \dots, t_* + kh, t^* - h\}$. Обозначим:

$V = (v_0, v_h, \dots, v_{kh}, \dots, v_{(N-1)h})'$ – N -мерный вектор, состоящий из управлений (6);

$\tilde{C} = (\tilde{c}_h(t_*), \tilde{c}_h(t_* + h), \dots, \tilde{c}_h(t_* + kh), \tilde{c}_h(t^* - h))'$ – N -мерный вектор, составленный из (13);

$D = (d_h(t_*), d_h(t_* + h), \dots, d_h(t_* + kh), d_h(t^* - h))'$ – матрица ($m \times N$), составленная из (14).

Таким образом, получили следующую задачу:

$$\begin{cases} \tilde{C}'V \rightarrow \max \\ DV = g_0 \\ |V_i| < 1, i = \overline{1, N} \end{cases} \quad (15)$$

Такая задача является канонической задачей с двухсторонними ограничениями. Данную задачу легко свести к интервальной задаче линейного программирования (ИЗЛП).

$$\begin{cases} c'x \rightarrow \max \\ b_* \leq Ax \leq b^* \\ d_* \leq x \leq d^* \end{cases} \quad (16)$$

Переобозначим целевую функцию $\tilde{C}'V \rightarrow \max$ как $c'x \rightarrow \max$, в качестве вектора x в данном случае используется вектор управлений V . В качестве полученной матрицы D используем матрицу A , а вектор g_0 b_* и b^* становятся равен основным ограничениям b_* и b^* . Прямые ограничения остаются те же.

2.2 Адаптивный метод

Список терминов, используемых в этом разделе

План—любой вектор x , удовлетворяющий ограничениям задачи линейного программирования.

План x^0 —оптимальный, если на нем достигается максимальное значение целевой функции $c'x^0 = \max c'x$.

План x^ε —субоптимальный, если на нем выполняется неравенство $c'x^0 - c'x^\varepsilon \leq \varepsilon$.

Пара $\{I_{\text{оп}}, J_{\text{оп}}\}$ —опора $K_{\text{оп}}$, состоящая из подмножеств $I_{\text{оп}} \in I$, $J_{\text{оп}} \in J$, где I и J —множества индексов строк и столбцов матрицы A из задачи (16).

Опора $K_{\text{оп}}$ называется пустой ($K_{\text{оп}} = \emptyset$), если она состоит из пустых множеств $I_{\text{оп}}$ и $J_{\text{оп}}$.

Для решения задачи (16) будем использовать адаптивный метод. Данный метод является конечным, точным и релаксационным интервальным

методом. Именно эти факторы и были ключевыми при выборе данного метода:

- Конечность. Можно найти такое число $k^0 < \infty$, что $x^0 = x^{k^0}$.
- Точность. Все последовательные приближения x^k точно соответствуют ограничениям задачи.
- Релаксационность. При остановке метода на любой итерации имеется некоторая компенсация за проделанную работу.

Преимуществом данного метода также является то, что он направлен на решение «ленточных» задач линейного программирования [10]. В данном случае (16) имеет такой вид при большом числе отрезков разделения N исходного интервала $[t_*, t^*]$.

Конечной целью этого метода является нахождение - оптимального x^0 либо субоптимального x^ε планов задачи линейного программирования. Для данной работы роль оптимального плана играет оптимальное управление. Так как адаптивный итеративный, то он состоит из конечного числа шагов-итераций, на каждой из которых находится новое приближение.

В основе метода лежит принцип уменьшения оценки субоптимальности: каждую итерацию необходимо строить так, чтобы каждый раз уменьшалась оценка субоптимальности β :

$$\beta(\bar{x}, \overline{K_{\text{оп}}}) < \beta(x, K_{\text{оп}}) \quad (17)$$

Адаптивный метод состоит из двух фаз:

- на первой фазе адаптивного метода строится первое приближение-начальный план и опора $\{x^1, x_{\text{оп}}^1\}$;
- на второй фазе строятся последующие приближения $\{x^k, x_{\text{оп}}^k\}$, $k=1,2,3\dots$

Каждая из фаз состоит из двух этапов: на первой фазе происходит построение плана и опоры, вторая фаза состоит из замены плана и опоры для уменьшения оценки субоптимальности.

Кратко изложим основные этапы адаптивного метода [6]:

А. Первая фаза адаптивного метода. Построение начального плана

Пусть \tilde{x}_j – значение j -й компоненты плана, вычисленной по рекомендациям специалиста. Если такие рекомендации отсутствуют, то $\tilde{x}_j = (d_j^* + d_{*j})/2$. \tilde{z}_i – значение i -й компоненты вектора затрат, полученной в соответствии с рекомендациями специалиста. Если таковых нет, положим:

$$\tilde{z}_i = \begin{cases} b_{*i}, & \text{если } A(i, J)\tilde{x} < b_{*i} \\ b_i^*, & \text{если } A(i, J)\tilde{x} > b_i^* \\ A(i, J)\tilde{x}, & \text{если } b_{*i} \leq A(i, J)\tilde{x} \leq b_i^*, i \in I \end{cases}$$

$\tilde{\omega} = \tilde{z} - A\tilde{x}$ – вектор невязок. Если $\tilde{\omega} = 0$, то \tilde{z} – вектор затрат на векторе \tilde{x} , следовательно \tilde{x} – план. Осталось только проверить его на субоптимальность. Если $\tilde{\omega} \neq 0$, то построим множества $I_+ = \{i \in I: \tilde{\omega}_i > 0\}$, $I_- = \{i \in I: \tilde{\omega}_i < 0\}$, $I_0 = \{i \in I: \tilde{\omega}_i = 0\}$

Сформулируем задачу первой фазы:

$$\sum_{i \in I_+ \cup I_-} x_{n+i} \rightarrow \min$$

$$\begin{cases} b_{*i} \leq A(i, J)x + x_{n+i} \leq b_i^*, i \in I_+ \\ b_{*i} \leq A(i, J)x - x_{n+i} \leq b_i^*, i \in I_- \\ b_{*i} \leq A(i, J)x \leq b_i^*, i \in I_0 \end{cases} \quad (18)$$

$$d_* \leq x \leq d^*, 0 \leq x_{n+i} \leq |\tilde{\omega}_i|, i \in I_+ \cup I_-$$

Для задачи первой фазы начальным планом является совокупность

$$(\tilde{x}, x_{n+i} = |\tilde{\omega}_i|, i \in I_+ \cup I_-) \quad (19)$$

Итак, чтобы решить задачу (16), необходимо добавить к начальному плану (19) опору и решить ее адаптивным методом. В итоге через конечное число итераций получим оптимальный опорный план $\{(x^*, x_{n+i}^*), K_{\text{оп}}^*\}$. При этом x_{n+i}^* может принимать следующие значения:

1. $x_{n+i}^* \neq 0$; при этом задача (12) некорректна и не имеет планов;
2. $x_{n+i}^* = 0$, $K_{\text{оп}}^*$ не содержит индексов $(n+i)$, $i \in I_+ \cup I_-$;
3. $x_{n+i}^* = 0$, $K_{\text{оп}}^*$ содержит некоторые индексы $(n+i)$, $i \in I_+ \cup I_-$ (индексы искусственных переменных).

Во втором случае $\{x^*, K_{\text{оп}}^*\}$ – опорный план и необходимо переходить ко второй фазе.

Если же реализовал себя третий случай, то сформулируем буферную задачу, которая поможет опоре $K_{\text{оп}}^*$ избавиться от индексов $(n+i)$, $i \in I_+ \cup I_-$

$$c'x \rightarrow \max,$$

$$\begin{cases} b_{*i} \leq A(i, J)x + x_{n+i} \leq b_i^*, i \in \tilde{I}_+ \\ b_{*i} \leq A(i, J)x - x_{n+i} \leq b_i^*, i \in \tilde{I}_- \\ b_{*i} \leq A(i, J) \leq b_i^*, i \in I \setminus (\tilde{I}_+ \cup \tilde{I}_-) \end{cases}$$

$$d_* \leq x \leq d^*, 0 \leq x_{n+i} \leq 0, i \in \tilde{I}_+ \cup \tilde{I}_-$$

$\tilde{I}_+ \cup \tilde{I}_-$ —множество искусственных индексов, составляющие искусственные индексы $j = n+i$ компоненты $J_{\text{оп}}^*$ опоры $K_{\text{оп}}^*$

Решаем эту задачу с помощью второй фазы. На каждой итерации из опоры будет удаляться один искусственный индекс. В результате получится план $(x^*, x_{n+i}^* = 0)$ и опора без искусственных индексов. Этот план и опору используем в качестве начальных на второй фазе.

В. Первая фаза адаптивного метода. Построение начальной опоры

Как и при построении начального плана, так и при построении начальной опоры можно использовать опыт эксперта. Выделяем $\tilde{I} \subset I$ на элементах которого основные ограничения задачи (16) по мнению эксперта активны на оптимальном плане. $A(i, J)\tilde{x} = b_i^* \vee b_{*i}$. Из множества J выделяем подмножество не критических индексов: $d_{*j} \leq x \leq d_j^*$, выравниваем количество элементов выделенных множеств.

Также в качестве начальной опоры можно взять пустую опору.

С. Вторая фаза адаптивного метода. Процедура замены плана.

Новый план \bar{x} строится из старого плана x по правилу

$$\bar{x} = x + \theta l,$$

где n -вектор l — направление плана

число θ — шаг вдоль направления l , $\theta > 0$.

l — допустимое направление в точке x относительно множества X , если существует такое число $\theta_0 > 0$, что $x + \theta l \in X, \forall \theta \in [0, \theta_0]$

Для уменьшения (17) направление l должно быть выбрано так, чтобы выполнялось неравенство $c'l \geq 0$. Множество допустимых направлений, удовлетворяющих этому неравенству, называют подходящим. Обозначим такое множество E_{Π} .

В качестве l для быстрого уменьшения оценки субоптимальности разумно взять вектор l^0 , вдоль которого скорость изменения целевой функции максимальна.

$$c'l^0 = \max c'l, l \in E_{\Pi} \cap N, \text{ где } N\text{-компакт} \quad (20)$$

Множество N называется нормировочным множеством. Для его выбора используем задачу (16) и положим, что $N^0 = \{l \in R^n: x + l \in X\}$ -

оптимальное нормировочное множество. Для облегчения решения полученной задачи уберем неопорные основные ограничения $b_{*н} \leq A(I_н, J)(x + l) \leq b_н^*$ и опорные прямые ограничения $d_{*оп} \leq x_{оп} + l_{оп} \leq d_{оп}^*$ и заменим $\bar{x} = x + l$. Тогда задача (16) примет вид:

$$\begin{cases} c'\bar{x} - c'x \rightarrow \max \\ b_{*оп} \leq A(I_{оп}, J)(x + l) \leq b_{оп}^* \\ d_{*н} \leq x_н + l_н \leq d_н^* \end{cases} \quad (21)$$

Для вычисления адаптивного направления l_a , полученного как решение задачи (21) воспользуемся формулой $l_a = \bar{x} - x$, где \bar{x} – псевдоплан.

Найдем максимально допустимый шаг $\theta_0, \theta_0 \leq 1$:

Пусть θ_x – макс θ , при котором выполняются прямые ограничения;

θ_z – макс θ , при котором выполняются основные ограничения;

θ_j – макс θ , при котором нарушается j-е прямое ограничение;

θ_i – макс θ , при котором нарушается i-е основное ограничение;

$$\theta_j = \begin{cases} \frac{d_j^* - x_j}{l_{a_j}}, \text{ если } l_{a_j} > 0 \\ \frac{d_{*j} - x_j}{l_{a_j}}, l_{a_j} < 0 \\ +\infty, l_{a_j} = 0 \end{cases} \quad (22)$$

$$\theta_x = \min \theta_j, j \in J_{оп}$$

$$\theta_i = \begin{cases} \frac{b_i^* - z_i}{p_{a_i}}, \text{ если } p_{a_i} > 0 \\ \frac{b_{*i} - z_i}{p_{a_i}}, p_{a_i} < 0 \\ +\infty, p_{a_i} = 0 \end{cases} \quad (23)$$

$$\theta_z = \min \theta_i, i \in I_н$$

$\theta_0 = \min [1, \theta_x, \theta_z]$. Таким образом, новый план получается в виде $\bar{x} = x + \theta_0 l_a$.

При этом оценка субоптимальности нового плана равна:

$$\beta(\bar{x}, K_{\text{оп}}) = c' \bar{x} - c' x = c' \bar{x} - c'(x + \theta_0 l_a) = (1 - \theta_0) \beta(x, K_{\text{оп}})$$

В случае $\theta_0 = 1$ получим $\beta(\bar{x}, K_{\text{оп}}) = 0$, что означает, что новый план \bar{x} – оптимальный.

Если $(1 - \theta_0) \beta(x, K_{\text{оп}}) \leq \varepsilon$, то \bar{x} – ε -оптимальный план.

Если $(1 - \theta_0) \beta(x, K_{\text{оп}}) > \varepsilon$ необходимо перейти к процедуре замены опоры.

D. Вторая фаза адаптивного метода. Процедура замены опоры

Замена опоры проводится согласно следующей схеме: $K_{\text{оп}} \rightarrow \lambda_{\text{сопр}} \rightarrow \overline{\lambda_{\text{сопр}}} \rightarrow \overline{K_{\text{оп}}}$, где $\lambda_{\text{сопр}}$ –сопровождающий двойственный план. Следуя (17), необходимо строить двойственный сопровождающий план так, чтобы двойственная целевая функция уменьшалась: $\psi(\overline{\lambda_{\text{сопр}}}) \leq \psi(\lambda_{\text{сопр}})$

Для этого построим кусочно-линейную непрерывную ломаную линию $\lambda(\sigma)$, $\sigma > 0$, которая начинается в $\lambda(0) = \lambda_{\text{сопр}}$ и ее угловые точки представляют собой сопровождающие двойственные планы $\lambda^k = \lambda(\sigma^k)$, $k = 1, 2, \dots, k_*$. Построение ломаной $\lambda(\sigma)$ начинается с направления $\Delta \lambda$ при движении вдоль которого двойственная целевая функция убывает.

Итак, после замены плана шаг θ^0 равен либо $\theta_z = \theta_{i_0}$, либо $\theta_x = \theta_{j_0}$

Рассмотрим эти случаи отдельно:

1) Пусть $\theta^0 = \theta_z = \theta_{i_0}$. Для построения $\Delta \lambda$ необходимо построить направление Δu вектора потенциалов r . Для первого звена ломанной $\lambda(\sigma) = r + \sigma \Delta u$ для начала зададим неопорную компоненту Δu_n :

$$\Delta y_H = \begin{cases} \text{sgn}(\xi_{i_0} - \bar{z}_{i_0}), & \text{если } i = i_0 \\ 0, & \text{если } i \in I_H \setminus i_0 \end{cases},$$

ξ – вектор псевдозатрат [6].

$$\Delta y_{\text{оп}} = -\Delta y_{i_0} A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1}$$

Если опора $K_{\text{оп}}$ -пустая, то полагаем, что $\Delta y_{\text{оп}} = 0$. Зная $\Delta y = (\Delta y_{\text{оп}}, \Delta y_H)$ построим

$$y_i(\sigma) = \begin{cases} 0, & \text{если } i \in I_H \setminus i_0 \\ \sigma \Delta y_{i_0}, & \text{если } i = i_0, \\ r_i + \sigma \Delta y_i, & \text{если } i \in I_{\text{оп}} \end{cases} \quad (24)$$

$$\text{где } \Delta y_{i_0} = \begin{cases} -1, & \text{если } \bar{z}_{i_0} = b_{*i_0} \\ 1, & \text{если } \bar{z}_{i_0} = b_{i_0}^* \end{cases}$$

Если опора $K_{\text{оп}}$ -пустая, то полагаем, что $y_i(\sigma) = 0, i \in I \setminus i_0$

$$\delta_j(\sigma) = \begin{cases} \Delta_j + \sigma \Delta \delta_j, & j \in J_H \\ 0, & j \in J_{\text{оп}} \end{cases},$$

Δ – вектор оценок [6].

$$\Delta \delta_j = \begin{cases} -\Delta y' A(I, j), & j \in J_H \\ 0, & j \in J_{\text{оп}} \end{cases}$$

Если $K_{\text{оп}} = \emptyset$, то $\Delta \delta_j = \Delta y_{i_0} A(i_0, j), j \in J$

2) Рассмотрим случай $\theta^0 = \theta_x = \theta_{j_0}$, который возможен только в случае непустой опоры. Для построения направления $\Delta \lambda$ определим направление $\Delta \delta_{\text{оп}}$ изменения опорной компоненты вектора оценок:

$$\Delta \delta_{\text{оп}} = e_{j_0} \text{sgn}(\alpha_{j_0} - \bar{x}_{j_0}), \text{ где } e_{j_0} \text{ – единичный орт} \quad (25)$$

$$\Delta \delta_H = -\Delta y_{\text{оп}} A(I_{\text{оп}}, J_H) \quad (26)$$

При этом

$$\Delta y = (\Delta y_{\text{оп}}, \Delta y_{\text{н}}); \Delta y_{\text{н}} = 0, \Delta y_{\text{оп}} = -e_{j_0} A_{\text{оп}}^{-1} \Delta \delta_{j_0}, \quad (27)$$

$$\text{где } \Delta \delta_{j_0} = \begin{cases} -1, & \text{если } \bar{x}_{j_0} = d_{*j_0} \\ 1, & \text{если } \bar{x}_{j_0} = d_{j_0}^* \end{cases}.$$

Далее рассмотрим вычисление короткого двойственного шага. При движении вдоль $\Delta \lambda$ скорость изменения целевой функции когда-то изменится. Закон поведения функции $\lambda(\sigma)$ зависит от поведения функций $y(\sigma)$ и $\delta(\sigma)$. Найдем максимальный отрезок $[0, \sigma^1]$, на котором $y_i(\sigma)$ и $\delta_i(\sigma)$ сохраняют свои первоначальные знаки. Число σ^1 – короткий двойственный шаг.

1. Рассмотрим первый случай $\theta^0 = \theta_z = \theta_{i_0}$

Из (24) следует, что функции $y_i(\sigma), i \in I_{\text{н}}$ сохраняют знаки при любых значениях $\sigma > 0$. Рассмотрим $y_i(\sigma), i \in I_{\text{оп}}$ при $K_{\text{оп}} \neq \emptyset$. Пусть σ_i – значение σ , при котором $y_i(\sigma)$ меняет знак. Тогда

$$\sigma_i = \begin{cases} -\frac{r_i}{\Delta y_i}, & \text{если } r_i \Delta y_i < 0 \\ \infty, & \text{если } r_i \Delta y_i \geq 0 \end{cases} \quad (28)$$

Обозначим i_* индекс той компоненты $y_i(\sigma)$, которая первой меняет знак. Тогда:

$$\sigma_{i_*} = \min_{i \in I_{\text{оп}}} \sigma_i \quad (29)$$

При пустой опоре $K_{\text{оп}}$ предполагается, что $\sigma_i = \infty, i \in I_{\text{оп}}$. В таком случае $\sigma_{i_*} = \infty$.

Аналогично для функций $\delta_i(\sigma)$: по построению $\delta_j(\sigma), j \in J_{\text{оп}}$, то есть менять знак могут только $\delta_j(\sigma), j \in J_{\text{н}}$. Тогда

$$\sigma_j = \begin{cases} -\frac{\Delta_j}{\Delta\delta_i}, & \text{если } \Delta_j\Delta\delta_j < 0 \\ \infty, & \text{если } \Delta_j\Delta\delta_j \geq 0 \end{cases} \quad (30)$$

$$\sigma_{j_*} = \min \sigma_i, i \in I_H \quad (31)$$

Следовательно, если опора непустая, то $\sigma^1 = \min [\sigma_{j_*}, \sigma_{i_*}]$. При пустой опоре $\sigma^1 = \sigma_{j_*}$

2. Рассмотрим второй случай $\theta^0 = \theta_x = \theta_{j_0}$, он возможен только если $K_{оп} \neq \emptyset$.

Из (27) следует, что при $i \in I_H$ функции $y_i(\sigma)$ сохраняют знаки при любых значениях $\sigma > 0$. Тогда вычисление σ_i, σ_{i_*} при $i \in I_{оп}$ происходит по формулам (28), (29).

Функции $\delta_j(\sigma), j \in J_{оп}$ не меняют знаки. Для подсчета $\sigma_j, \sigma_{j_*}, j \in J_H$ справедливы формулы (30), (31)

Из этого следует, что при замене опоры справедливы четыре случая:

1. $\theta^0 = \theta_{i_0}, \sigma^1 = \sigma_{i_*}$
2. $\theta^0 = \theta_{i_0}, \sigma^1 = \sigma_{j_*}$
3. $\theta^0 = \theta_{j_0}, \sigma^1 = \sigma_{i_*}$
4. $\theta^0 = \theta_{j_0}, \sigma^1 = \sigma_{j_*}$

При этом, если опора непустая, то возможна реализация только второго случая. Построим новую опору $\overline{K_{оп}}$ для каждого из этих случаев:

- 1) $\overline{K_{оп}} = (\overline{I_{оп}}, \overline{J_{оп}})$, где $\overline{I_{оп}} = (I_{оп} \setminus i_*) \cup i_0, \overline{J_{оп}} = J_{оп}$
- 2) $\overline{K_{оп}} = (\overline{I_{оп}}, \overline{J_{оп}})$, где $\overline{I_{оп}} = I_{оп} \cup i_0, \overline{J_{оп}} = J_{оп} \cup i_*$. В этом случае размер $\overline{K_{оп}}$ на единицу больше размера $K_{оп}$.
- 3) $\overline{K_{оп}} = (\overline{I_{оп}}, \overline{J_{оп}})$, где $\overline{I_{оп}} = I_{оп} \setminus i_*, \overline{J_{оп}} = J_{оп} \setminus i_0$. В этом случае размер $\overline{K_{оп}}$ на единицу меньше размера $K_{оп}$.

$$4) \overline{K}_{\text{оп}} = (\overline{I}_{\text{оп}}, \overline{J}_{\text{оп}}), \text{ где } \overline{I}_{\text{оп}} = I_{\text{оп}}, \overline{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_*) \cup j_0.$$

Оценка субоптимальности на новом опорном плане равна:

При $\beta(\bar{x}, \overline{K}_{\text{оп}}) \leq \varepsilon$ адаптивный метод заканчивает свою работу. Если же $\beta(\bar{x}, \overline{K}_{\text{оп}}) > \varepsilon$, то происходит переход к новой итерации.

2.3 Алгоритм работы

1. Сведение вспомогательной задачи оптимального управления (7) с начальным условием $y(t_i) = y_i$ к канонической задаче линейного программирования.
2. Преобразование канонической задачи в интервальную задачу линейного программирования (ЛП).
3. Решение интервальной задачи ЛП адаптивным методом. В результате решения определяется оптимальное управление v_i на промежутке $[t_i, t_{i+1}]$.
4. Замыкание нелинейного дифференциального уравнения (4) полученным управлением v_i .
5. Получение нового реального условия в момент времени t_{i+1} : $y(t_{i+1}) = y_{i+1}$.
6. Возвращение к шагу 1, с той лишь разницей, что вместо $y(t_i) = y_i$ берется новое $y(t_{i+1}) = y_{i+1}$.

Данный цикл повторяется до того момента, пока не определятся все оставшиеся v_k на промежутках $[t_k, t_{k+1}]$, $k = \overline{l, N}$.

Глава 3. Программная реализация

Для реализации был выбран пакет прикладных программ MATLAB, поскольку данный пакет содержит в себе встроенный инструментарий для работы с матрицами, дифференциальными уравнениями и обработкой данных, включая построение графиков [11]. Использование MATLAB для данной задачи позволяет сократить временные ресурсы за счет встроенных функций [12].

Программная реализация содержит в себе три файла- основной скрипт Main.m и две функции Optimal_control.m и Adaptivny_method.m. В функции Optimal_control.m описано сведение задачи оптимального управления к задаче ЛП. Входными параметрами для этой функции являются матрицы \hat{A} и H , вектора c , b и g , начальное условие y_0 , период квантования h и начальный и конечный моменты времени t . На выход функция возвращает вектора \tilde{C} и g_0 , а также матрицу D .

Adaptivny_method.m-функция, позволяющая решить задачу линейного программирования адаптивным методом. В качестве входных параметров функция принимает следующие значения: матрицу D вектора v -начальный план g_0 , \tilde{C} , $I_{оп}$, $J_{оп}$ -индексы опоры. На выход функция возвращает v -новый план.

В основном скрипте Main.m запускается алгоритм, включающий цикл нахождения оптимального управления на каждом отрезке $[t_i, t_{i+1}]$, $i = \overline{0, N}$. Нелинейное дифференциальное уравнение (4) решается с применением встроенной функцией ode45. Для решения данная функция использует формулы Рунге-Кутты 4-го и 5-го порядка.

3.1 Примеры

Пример № 1. Входные параметры для примера № 1 приведены в Таблице 1.

Входные параметры для функции Optimal_control	\hat{A}		H	c	b	g	y_0	t_*	t^*	h
	-0.05 0	1 -5	1 -3	-2 2	0 5	-0.6	1 0	0	π	$\frac{\pi}{4}$
Входные параметры для функции Adaptivny_meth od	D	\tilde{c}		g_0	d_*	d^*	$I_{оп}$	$J_{оп}$		v
	0.691	-1.383		-1.45	0	2	\emptyset	\emptyset		1
	0.718	-1.437								0
	0.686	-1.449								1
	-2.361	0.8								0
Параметры уравнения (4)	q			\bar{x}			\bar{u}			
	$-2\cos(t_i)$			0.2			0.5			
	0			0						

Таблица 1

Результат решения задачи (7) приведен на Рис. 2:

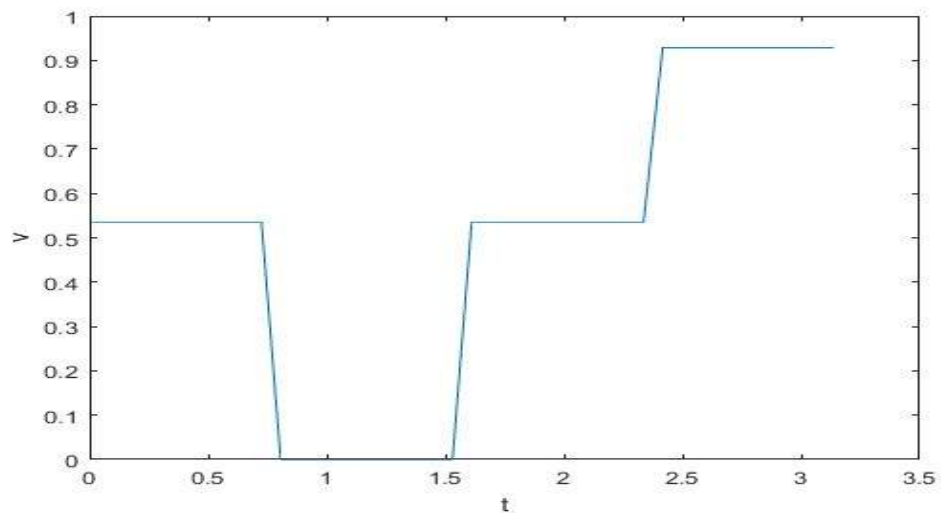


Рис. 2 Оптимальное управление v в примере №1

Пример №2. Входные параметры для примера №2 приведены в Таблице 2.

Входные параметры для функции Optimal_control	\hat{A}		H	c	b	g	y_0	t_*	t^*	h
	-0.05 0	1 0.2	-2 3	2 -0.5	0 0.2	2.6	2 4	0	π	$\frac{\pi}{6}$
Входные параметры для функции Adaptivny_meth od	D	\tilde{C}		g_0	d_*	d^*	$I_{оп}$	$J_{оп}$		v
	-1.113	0.743		16.15	0	2	\emptyset	\emptyset		0
	-1.025	0.68								1
	-0.946	0.624								0
	-0.876	0.574								2
	-0.8143	0.529								0
	-0.758	0.489								0
Параметры уравнения (4)	q			\bar{x}			\bar{u}			
	$\sin(t_i)$			2			0.5			
	0			1						

Таблица 2

Результаты решения задачи (7) с входными параметрами из Таблицы 2 приведены на Рис 3.

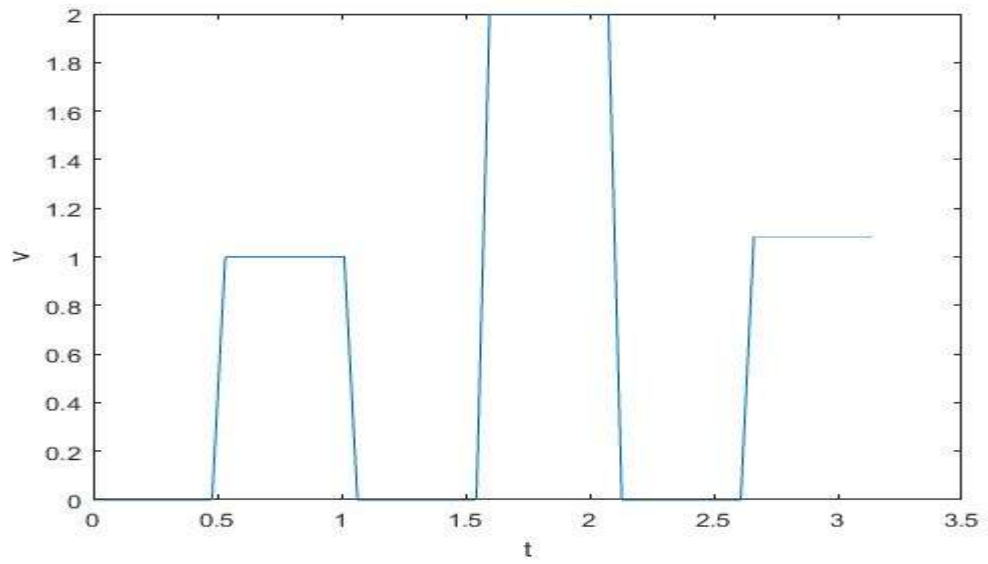


Рис 3. Оптимальное управление v в примере №2

Заключение. Выводы

Здесь было проведено моделирование работы выбранной цепи поставок и найдено оптимальное управление (желаемая скорость производства) на заранее определенных промежутках времени. Данное моделирование проведено для помощи производителю в решении об объемах продукции, которую необходимо выпускать, чтобы обеспечить ожидаемый спрос, а также максимизировать прибыль производства.

Таким образом, в данной работе были получены следующие результаты:

1. выбрана математическая модель цепи поставок;
2. изучены методы теории управления для нахождения оптимального управления;
3. проведено компьютерное моделирование данной модели с помощью изученных методов;
4. сделаны численные эксперименты.

В дальнейшем планируется расширить исходную модель до двух продуктов производства и сравнить результаты моделирования с результатами, полученными использованием других методов нахождения оптимального управления.

Список литературы

1. SCM.// Wikipedia URL:
https://en.wikipedia.org/wiki/Supply_chain_management
2. Иванов Д. А. Управление цепями поставок. СПб: Изд-во Политехн. ун-та, 2009. 660 с.
3. Прасолов А.В. Структурная оптимизация цепочки снабжения // 4я международная конференция "Государство и бизнес", 2012. С.176–186
4. Балашевич Н. В., Габасов Р., Кириллова Ф. М. Вычисление оптимальных программы и управления в линейной задаче с фазовым ограничением// Журнал вычислительной математики и математической физики, 2005. Т. 45. С. 2112-2130
5. Габасов Р., Кириллова Ф.М., Ружицкая Е.А. Демпфирование и стабилизация маятника при больших начальных возмущениях // Известия Академии наук. Сер. Теория и системы управления. 2001. №1. С. 29-38
6. Альсевич В. В., Габасов Р., Глушенков В. С. Оптимизация линейных экономических моделей: Статистические задачи: Учеб. Пособие. М.: БГУ, 2000. 210 с
7. Bukhari F. Adaptive Control of a Production-Inventory Model with Uncertain Deterioration Rate//Applied Mathematics, 2011, Vol. 2, P. 1170-1174
8. Benhadid Y., Tadj L., Bounkhel M. Optimal Control Of Production Inventory Systems With Deteriorating Items And Dynamic Costs// Applied Mathematics E-Notes, 2008, P. 194-202
9. Perea E., Grossmann I., Ydstie E., Tahmassebi T. Dynamic modeling and classical control theory for supply chain management// Computers and Chemical Engineering, 2000, P. 1143-1149

10. Габасов Р., Кириллова Ф. М. Методы линейного программирования часть 3. Специальные задачи. М.: Книжный дом «ЛИБРОКОМ», 2010. 368 с.

11. MATLAB// mathworks URL:

<http://www.mathworks.com/help/matlab/index.html>

12. Chapman J. S. MATLAB programming for Engineers, USA, 2008

Приложение

1. Функция Optimal_control

```
function [cvolna, D, g_sh]=Optimal_control(X1,A,c,b,g,tstart,tend,N,H,h)
Y=@(t)expm(A*t);% фундаментальная матрица Y
Y1=@(t)expm(A*(-t));% обратная фундаментальная матрица Y

func=@(t)(c*Y(tend)*Y1(t)*b);
func1=@(t)(H*Y(tend)*Y1(t)*b);
cvolna = zeros(0,N);

for i=1:N % Расчет вектора cvolna
    cvolna(i)=integral(func, tstart+(i-1)*h, tstart+i*h, 'ArrayValued',true);
% по формуле 13
end

for i=1:N % Расчет матрицы D по формуле 14
    sup = integral(func1, tstart+(i-1)*h, tstart+i*h, 'ArrayValued',true);
    D(:,i)=sup;
end

g_sh=g-H*Y(tend)*Y(0)*X1 %вычисление g0
end
```