ВВЕДЕНИЕ

В учебном пособии рассмотрено несколько относительно простых экономико-математических моделей. Практикум, основанный на этих моделях, в течение ряда лет выполняется студентами 2–3 курсов экономических специальностей ВГУЭС.

Изначально практикум строился на работах, предложенных в известной книге «Математическая экономика на персональном компьютере» под редакцией М. Кубонива [1]. Со временем выявился ряд ограничений.

Во-первых, вышедшая в свет в 1984 г. и переизданная в России в 1991 г., книга стала библиографической редкостью и практически недоступна студентам.

Во-вторых, тексты программ в книге изложены на языке BASIC. Для персональных компьютеров 1984 г. это было неплохим решением. За прошедшие годы аппаратные возможности и программное обеспечение персональных компьютеров неизмеримо выросли. И хотя язык BASIC по-прежнему можно эффективно применять для решения многих задач математической экономики, у современных студентов не вызывает энтузиазма уныло-синий интерфейс QBASIC'а.

В-третьих, для студенческой аудитории оказались недостаточными краткие описания теоретических разделов работ, которые характерны для стиля книги.

Оставаясь в рамках идей и подходов, предложенных в книге, в данном пособии значительно расширены теоретические разделы работ. Пользователю предложен выбор программирования на языках QBASIC, C++, в среде C++Builder, а также возможность работы с математическим пакетом MathCAD – 2000.

Для студентов 2–3 курсов, по-видимому, оптимальной будет работа с математическим пакетом MathCAD. Страницы кода на языках QBASIC или C++ превращаются в десяток строк матричных уравнений или индексных выражений. Важным достоинством пакета является доброжелательный к пользователю графический интерфейс и понятный и привычный синтаксис математических выражений. Время адаптации пользователя к MathCAD'у ничтожно по сравнению с освоением программирования на языке высокого уровня.

Автор надеется, что учебное пособие окажется полезным для студентов, изучающих математические методы экономики, как под руководством преподавателя, так и самостоятельно, а также поможет в работе преподавателям, ведущим аналогичные занятия.

ОЦЕНКА ПАРАМЕТРОВ ПРОИЗВОДСТВЕННОЙ ФУНКЦИИ

1. Оценка параметров функции Кобба-Дугласа

1.1. Производственная функция Кобба-Дугласа

Функция Кобба-Дугласа относится к наиболее известным производственным функциям неоклассического типа [2, 3].

Если обозначить объем выпуска за некоторый временной промежуток символом Y, величину использованных в этом интервале производственных фондов (капитала) символом K и объем привлеченного труда – L, то статистические данные по объемам выпусков можно аппроксимировать (приближенно описать) степенной зависимостью от K и L с тремя параметрами:

$$Y(K,L) = cK^{\alpha}L^{\beta}, \tag{1}$$

где c, α и β – параметры, $c > 0$, $\alpha > 0$, $\beta > 0$.

Экономист Дуглас в сотрудничестве с математиком Коббом в 20-х годах XX столетия аппроксимировали функцией (1) данные по американской обрабатывающей промышленности за период с 1899 по 1922 г. В результате они получили, что $\alpha + \beta \approx 1$, т.е.

$$Y(K,L) = cK^{\alpha}L^{1-\alpha}, \tag{2}$$

где $c > 0$, $0 < \alpha < 1$.

Производственная функция (2) называется функцией Кобба-Дугласа. Она является однородной функцией. Это означает, что увеличение аргументов в m раз приводит к увеличению значения функции тоже в m раз. Действительно, если

$$Y(K,L) = cK^{\alpha}L^{1-\alpha},$$

то

$$\begin{aligned} Y_1(K,L) &= c(mK)^{\alpha}(mL)^{1-\alpha} \\ &= cK^{\alpha}L^{1-\alpha}m^{\alpha}m^{1-\alpha} \\ &= mY(K,L). \end{aligned}$$

Если в качестве множителя m выбрать $m = 1/L$, то получим выпуск, нормированный на величину затраченного труда:

$$\frac{Y(K,L)}{L} = c\left(\frac{K}{L}\right)^{\alpha}\left(\frac{L}{L}\right)^{1-\alpha} = c\left(\frac{K}{L}\right)^{\alpha}.$$

Введем следующие обозначения: $k = K/L$ – капиталовооруженность труда, $y(k) = Y(K,L)/L$ – производительность труда. В этом случае функция Кобба-Дугласа принимает вид

$$y(k) = ck^{\alpha}. \tag{3}$$

С 20-х годов XX столетия функция Кобба-Дугласа претерпела ряд модификаций. Одной из них является введение в функцию техногенного фактора – сомножителя $e^{\lambda t}$, учитывающего технический прогресс, где λ – постоянная темпа прогресса, а t – время:

$$y(k) = ce^{\lambda t}k^{\alpha}. \tag{4}$$

В настоящей работе предлагается аппроксимировать данные, которые приведены в работе Дугласа [4] по объемам производства Y_i, $i = 1,2,...,n$; и соответствующие им данные по K_i и L_i, $i = 1,2,...,n$; зависимостями (1), (2), (4). Развитие вычислительной техники и программного обеспечения сделало эту задачу доступной студенту, знакомому с численными методами аппроксимации и языками программирования или математическими программными средами типа Mathcad, Maple, MatLab и другие.

Рассмотрим общий подход к решению данной задачи.

1.2. Линейный регрессионный анализ

Пусть имеется линейная относительно величин $\gamma_0, \gamma_1, \gamma_2$ и γ_3 функциональная зависимость, например:

$$y(x) = \gamma_0 x^0 + \gamma_1 x^1 + \gamma_2 x^2 + \gamma_3 x^3, \tag{5}$$

где аргумент x может принимать значения $x_0, x_1, x_2,..., x_n$. Если коэффициенты $\gamma_0, \gamma_1, \gamma_2, \gamma_3$, которые будем называть параметрами, известны, то рассчитать значения функции y_i для заданного набора x_i, $i = 1,2,...,n$ не представляет особой сложности. В данном случае мы имеем дело с, так называемой, прямой задачей.

На практике чаще встречается *обратная задача*, когда известен дискретный набор значений наблюдаемых величин y_i^u, $i = 1,2,...,n$; определенных с некоторыми погрешностями (неточное знание величин y_i будем отмечать верхним индексом *u*), а также соответствующих им значений $x_i, i = 1,2,...,n$. Предположим, что значения x_i известны точно, а y_i^u и x_i связаны некоторой зависимостью, например (5). Необходимо по наблюдаемому дискретному набору (x_i, y_i^u) с неточными значениями y_i^u установить значения параметров $\gamma_0, \gamma_1, \gamma_2, \gamma_3$.

Поскольку y_i^u отличаются от истинных значений, то можно определить лишь приближенные значения (оценки) параметров $\tilde{\gamma}_0, \tilde{\gamma}_1, \tilde{\gamma}_2, \tilde{\gamma}_3$, и, следовательно, построить лишь приближенную кривую

$$\tilde{y}(x) = \tilde{\gamma}_0 x^0 + \tilde{\gamma}_1 x^1 + \tilde{\gamma}_2 x^2 + \tilde{\gamma}_3 x^3, \tag{6}$$

которая, конечно, отличается от истинной

$$y(x) = \gamma_0 x^0 + \gamma_1 x^1 + \gamma_2 x^2 + \gamma_3 x^3.$$

Предварительно необходимо выработать метод, критерий, который позволил бы провести кривую (6) через область расположения точек y_i^u «наилучшим образом», или, другими словами, определить «наилучшие параметры».

В качестве критерия используется, как правило, *минимизация суммы квадратов уклонений*: «наилучшая кривая» $\tilde{y}(x)$ должна проходить через область расположения наблюдаемых значений y_i^u так, чтобы обеспечить

$$\min M = \min \sum_{i=0}^{n} \left[y_i^u - \tilde{y}(x_i) \right]^2. \tag{7}$$

Это упрощенная форма критерия, т.к. в нем не учтены погрешности наблюдаемых значений. При учете погрешностей в (7) должны войти, так называемые, веса для точек y_i^u. В нашем случае веса предполагаются одинаковыми, что, по сути, означает равенство ошибок наблюдений для всех точек.

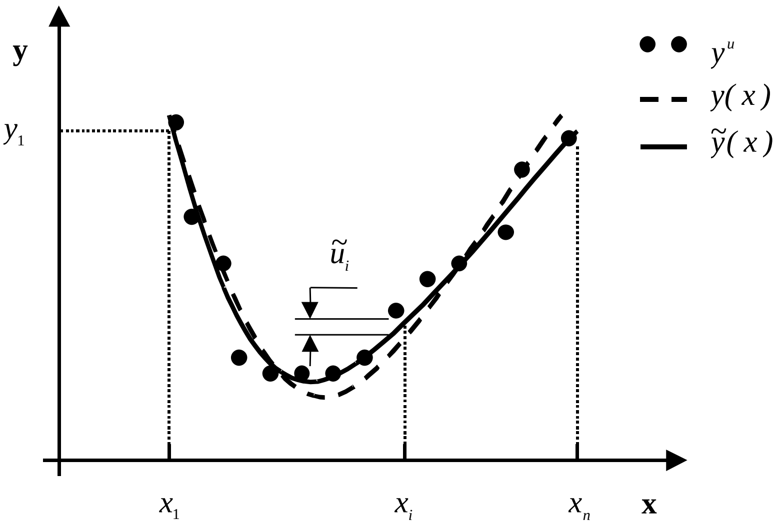

Рис. 1. • значения наблюдаемых величин y_i^u; – – – – предполагаемая истинная кривая (5); ——— «наилучшая кривая» (6)

Введем уклонения $\tilde{u}_i$ кривой $\tilde{y}(x_i)$ от наблюдаемых значений y_i^u (рис. 1):

$$\tilde{u}_i = y_i^u - \tilde{y}(x_i).$$

Тогда критерий (7) принимает вид

$$M = \sum_{i=0}^{n} \tilde{u}_i^{\,2} \rightarrow min. \qquad (8)$$

Метод, обеспечивающий построение аппроксимирующей кривой по этому критерию, носит название *метода наименьших квадратов*.

Усредненная кривая $\tilde{y}(x)$, проведенная «наилучшим образом», носит название *кривой регрессии*. *Регрессия* – зависимость среднего значения от какой-либо величины (совокупности величин). Модель, описывающая поведение усредненной кривой, приведенной к линейной относительно неизвестных параметров форме, носит название *линейной регрессионной модели.*

1.3. Матричный формализм и оценка параметров

Для дальнейшего изложения рациональнее перейти к матричной форме записи уравнений. Для этого введем следующие обозначения.

Совокупность наблюдаемых значений $x_i, i = 1,2,...,n$; запишем в виде вектора

$$\vec{x} = \begin{pmatrix} x_0 \\ x_1 \\ ... \\ x_n \end{pmatrix}, \tag{9}$$

а набор y_i^u, $i = 1,2,...,n$; – в виде вектора

$$\vec{y}^u = \begin{pmatrix} y_0^u \\ y_1^u \\ ... \\ y_n^u \end{pmatrix}. \tag{10}$$

В системе уравнений, определяющих значения $\tilde{y}(x)$ в точках x_i,

$$\begin{aligned} \tilde{\gamma}_0 + \tilde{\gamma}_1 x_0^1 + \tilde{\gamma}_2 x_0^2 + \tilde{\gamma}_3 x_0^3 &= \tilde{y}(x_0), \\ \tilde{\gamma}_0 + \tilde{\gamma}_1 x_1^1 + \tilde{\gamma}_2 x_1^2 + \tilde{\gamma}_3 x_1^3 &= \tilde{y}(x_1), \\ .... \\ \tilde{\gamma}_0 + \tilde{\gamma}_1 x_n^1 + \tilde{\gamma}_2 x_n^2 + \tilde{\gamma}_3 x_n^3 &= \tilde{y}(x_n), \end{aligned} \tag{11}$$

перейдем к матричной форме записи. Для этого введем матрицу

$$\mathbf{X} = \begin{pmatrix} 1 & x_0^1 & x_0^2 & x_0^3 \\ 1 & x_1^1 & x_1^2 & x_1^2 \\ ... & ... & ... & ... \\ 1 & x_n^1 & x_n^2 & x_n^3 \end{pmatrix}, \tag{12}$$

вектор $\vec{\tilde{\gamma}} = \begin{pmatrix} \tilde{\gamma}_0 \\ \tilde{\gamma}_1 \\ \tilde{\gamma}_2 \\ \tilde{\gamma}_3 \end{pmatrix}$ и вектор $\vec{\tilde{y}}(\vec{x}) = \begin{pmatrix} \tilde{y}(x_0) \\ \tilde{y}(x_1) \\ ... \\ \tilde{y}(x_n) \end{pmatrix}$. Тогда система уравнений (11) запишется в виде одного матричного уравнения

$$\begin{pmatrix} 1 & x_0^1 & x_0^2 & x_0^3 \\ 1 & x_1^1 & x_1^2 & x_1^2 \\ ... & ... & ... & ... \\ 1 & x_n^1 & x_n^2 & x_n^3 \end{pmatrix} \cdot \begin{pmatrix} \tilde{\gamma}_0 \\ \tilde{\gamma}_1 \\ \tilde{\gamma}_2 \\ \tilde{\gamma}_3 \end{pmatrix} = \begin{pmatrix} \tilde{y}(x_0) \\ \tilde{y}(x_1) \\ ... \\ \tilde{y}(x_n) \end{pmatrix}$$

или в более компактной записи:

$$\mathbf{X}\vec{\tilde{\gamma}} = \vec{\tilde{y}}(\vec{x}). \qquad (13)$$

Критерий (7) можно переписать в виде

$$min\ M = min\left(\vec{y}^{u} - \vec{\tilde{y}}(\vec{x})\right)^{T}\left(\vec{y}^{u} - \vec{\tilde{y}}(\vec{x})\right). \qquad (14)$$

Если ввести вектор уклонений

$$\vec{\tilde{u}} = \vec{y}^{u} - \vec{\tilde{y}}(\vec{x}) = \begin{pmatrix} y_0^u - \tilde{y}(x_0) \\ y_1^u - \tilde{y}(x_1) \\ \dots \\ y_n^u - \tilde{y}(x_n) \end{pmatrix} = \begin{pmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \dots \\ \tilde{u}_n \end{pmatrix},$$

то критерий (14) запишется еще более компактно:

$$min\ M = min\ \vec{\tilde{u}}^{T} \cdot \vec{\tilde{u}}\ . \qquad (15)$$

Оптимальные значения компонент вектора $\vec{\tilde{\gamma}}$ (обозначим их как $\vec{\tilde{\gamma}}^{*}$) будут найдены, если продифференцировать M по вектору $\vec{\tilde{\gamma}}$ и приравнять производные нулю. М зависит от вектора $\vec{\tilde{\gamma}}$:

$$\begin{aligned} M &= \vec{\tilde{u}}^{T} \cdot \vec{\tilde{u}} \\ &= \left(\vec{y}^{u} - \vec{\tilde{y}}(\vec{x})\right)^{T}\left(\vec{y}^{u} - \vec{\tilde{y}}(\vec{x})\right) \\ &= \left(\vec{y}^{u} - X\,\vec{\tilde{\gamma}}\right)^{T}\left(\vec{y}^{u} - X\,\vec{\tilde{\gamma}}\right), \end{aligned}$$

поэтому $\vec{\tilde{\gamma}}^{*}$ определяем из

$$\left(\frac{\partial M}{\partial \vec{\tilde{\gamma}}}\right)^{T} = 0, \qquad (16)$$

где транспонирование введено, чтобы обеспечить переход к вектору столбцу после дифференцирования.

Дифференцируя М по $\vec{\tilde{\gamma}}$, имеем

$$\begin{aligned} \frac{\partial M}{\partial \vec{\tilde{\gamma}}} &= 2(\vec{y}^{u} - X\vec{\tilde{\gamma}})^{T}\frac{\partial}{\partial \vec{\tilde{\gamma}}}(\vec{y}^{u} - X\vec{\tilde{\gamma}}) \\ &= -2(\vec{y}^{u} - X\vec{\tilde{\gamma}})^{T} X. \end{aligned}$$

Произведем согласно (16) операцию транспонирования и получим

$$\begin{aligned} \left(\frac{\partial M}{\partial \vec{\tilde{\gamma}}}\right)^{T} &= \left[-2(\vec{y}^{u} - X\vec{\tilde{\gamma}})^{T} X\right]^{T} \\ &= -2X^{T}(\vec{y}^{u} - X\vec{\tilde{\gamma}}) = 0. \end{aligned} \qquad (17)$$

Из (17) следует, что

$$X^T(\vec{y}^u - X\vec{\tilde{\gamma}}) = 0$$

и

$$X^T X\vec{\tilde{\gamma}} = X^T\vec{y}^u .$$

Чтобы найти вектор $\vec{\tilde{\gamma}}^*$, умножим слева последнее выражение на обратную матрицу $(X^T X)^{-1}$:

$$(X^T X)^{-1} X^T X\vec{\tilde{\gamma}} = (X^T X)^{-1} X^T\vec{y}^u .$$

Так как

$$(X^T X)^{-1}(X^T X) = I ,$$

где I – единичная матрица, а

$$I\vec{\tilde{\gamma}} = \vec{\tilde{\gamma}} ,$$

то

$$\boxed{\vec{\tilde{\gamma}}^* = (X^T X)^{-1} X^T\vec{y}^u}. \qquad (18)$$

Подставляя вектор $\vec{\tilde{\gamma}}^*$ в выражение (13), можно определить регрессионную кривую

$$\vec{\tilde{y}}(\vec{x}) = \mathbf{X}\vec{\tilde{\gamma}}^* .$$

Итак, чтобы найти оптимальные значения компонент вектора $\vec{\tilde{\gamma}}^*$, которые соответствуют регрессионной кривой, необходимо:

а) *сформировать вектор* $\vec{y}^u$ *– выражение (10);*

б) *сформировать матрицу* X *– выражение (12);*

в) *найти обратную матрицу* $(X^T X)^{-1}$ *;*

г) *произвести умножение матриц согласно выражению (18).*

1.4. Оценка параметров производственных функций

1.4.1. Производственная функция с тремя параметрами

Согласно изложенному выше, оценка параметров производственной функции (1) осуществляется следующим образом. В наличии имеются значения объемов выпусков Y_i, соответствующие объемы капитала K_i и затраченного труда L_i, $i = 0,1,...,n$. Уравнение (1) логарифмированием приведем к линейному относительно оцениваемых параметров виду

$$ln Y_i = ln c + \alpha ln K_i + \beta ln L_i , \quad i = 0,1,...,n. \qquad (19)$$

Сформируем вектор

$$\vec{y}^u = \begin{pmatrix} ln Y_0 \\ ln Y_1 \\ \ldots \\ ln Y_n \end{pmatrix},$$

матрицу

$$X = \begin{pmatrix} 1 & ln K_0 & ln L_0 \\ 1 & ln K_1 & ln L_1 \\ \ldots & \ldots & \ldots \\ 1 & ln K_n & ln L_n \end{pmatrix}$$

и вектор параметров

$$\vec{\tilde{\gamma}} = \begin{pmatrix} \tilde{\gamma}_0 \\ \tilde{\gamma}_1 \\ \tilde{\gamma}_2 \end{pmatrix} = \begin{pmatrix} ln c \\ \alpha \\ \beta \end{pmatrix}.$$

Тогда вектор наилучших параметров $\vec{\tilde{\gamma}}^*$ определяется согласно (18), а регрессионная кривая для производственной функции (1) имеет вид

$$\tilde{Y}(K,L) = e^{\tilde{\gamma}_0^*} K^{\tilde{\gamma}_1^*} L^{\tilde{\gamma}_2^*}. \qquad (20)$$

1.4.2. Производственная функция Кобба–Дугласа

Производственную функцию Кобба–Дугласа (3) логарифмированием приводим к линейному виду

$$\begin{aligned} ln\, y(k) &= ln c + \alpha ln k \\ &= ln c + \alpha ln\left(\frac{K}{L}\right). \end{aligned} \qquad (21)$$

Векторы $\vec{\tilde{\gamma}}$, $\vec{y}^u$ и матрица X для этого случая имеют вид

$$\vec{\tilde{\gamma}} = \begin{pmatrix} \tilde{\gamma}_0 \\ \tilde{\gamma}_1 \end{pmatrix} = \begin{pmatrix} ln c \\ \alpha \end{pmatrix}, \quad \vec{y}^u = \begin{pmatrix} ln \frac{Y_0}{L_0} \\ ln \frac{Y_1}{L_1} \\ \ldots \\ ln \frac{Y_n}{L_n} \end{pmatrix}, \quad X = \begin{pmatrix} 1 & ln \frac{K_0}{L_0} \\ 1 & ln \frac{K_1}{L_1} \\ \ldots & \ldots \\ 1 & ln \frac{K_n}{L_n} \end{pmatrix}.$$

Вектор наилучших оценок параметров $\vec{\tilde{\gamma}}^*$ вновь определяется из (18), а регрессионная кривая для функции (3) теперь имеет вид

$$\tilde{y}(K,L) = e^{\tilde{\gamma}_0^*}\left(\frac{K}{L}\right)^{\tilde{\gamma}_1^*}. \tag{22}$$

1.4.3. Модифицированная функция Кобба–Дугласа

Для производственной функции (4) линейное относительно параметров уравнение имеет вид

$$\ln y(k) = \ln c + \alpha \ln k + \lambda t.$$

Вектор $\vec{y}^u$ такой же, как и в предыдущем разделе, матрица X состоит из трех столбцов:

$$X = \begin{pmatrix} 1 & \ln\dfrac{K_0}{L_0} & t_0 \\ 1 & \ln\dfrac{K_1}{L_1} & t_1 \\ \dots & \dots & \dots \\ 1 & \ln\dfrac{K_n}{L_n} & t_n \end{pmatrix},$$

а вектор параметров $\vec{\tilde{\gamma}}$ содержит три компоненты

$$\vec{\tilde{\gamma}} = \begin{pmatrix} \tilde{\gamma}_0 \\ \tilde{\gamma}_1 \\ \tilde{\gamma}_2 \end{pmatrix} = \begin{pmatrix} \ln c \\ \alpha \\ \lambda \end{pmatrix}.$$

Наилучшая оценка для вектора $\vec{\tilde{\gamma}}$ определяется из (18), и, наконец, регрессионная кривая –

$$\tilde{y}(K,L) = e^{\tilde{\gamma}_0^*}\left(\frac{K}{L}\right)^{\tilde{\gamma}_1^*} e^{\tilde{\gamma}_2^* t}. \tag{23}$$

2. Контрольные вопросы

1. Что такое производственная функция и функция Кобба-Дугласа в частности?
2. Почему множитель в производственной функции (4), отражающий технический процесс, имеет экспоненциальный вид – $e^{\lambda t}$?
3. Почему, решая обратную задачу на основе наблюдаемых данных, можно найти только приближенные значения параметров?

4. Сформулируйте критерий, использованный для аппроксимации наблюдаемых данных в этой работе. В чем отличие аппроксимирующей кривой от интерполирующей кривой?
5. Какое уравнение является основным для расчета оптимальных значений параметров?
6. Для чего можно использовать аппроксимирующие кривые наблюдаемых данных?

3. Программы

Основная сложность, с точки зрения численных расчетов, заключается в нахождении обратной матрицы [5]. Программируя эту задачу на языке высокого уровня, можно воспользоваться матричными библиотеками, которые обеспечивают, как правило, возможность нахождения обратных матриц и, конечно, умножения матриц.

В данной работе предлагаются тексты программ на языках Qbasic [6], C++ [7] и в среде Borland C++Builder [8], аппроксимирующих данные Дугласа [4] зависимостью (1). За основу взята программа на языке Qbasic из работы [1].

Если воспользоваться средой Mathcad [9], то вычисление выражения (18) для большой области практических задач не представляет каких-либо сложностей.

3.1. Qbasic

Список переменных:
P – число элементов в выборке;
M = m – число оцениваемых параметров;
X(I,J) – фактические данные для независимой переменной;
Y(I) = Y_i – фактические данные для зависимой переменной;
A(I,J) – коэффициенты системы нормальных уравнений (элементы матриц X^TX, X^TY, а также матрицы $(X^TX)^{-1}$, обратной матрице X^TX;
B(I) = β_i – расчетные значения параметров;
SD(I) – стандартная ошибка параметра;
YM – среднее значение зависимой переменной;
A – расчетное значение зависимой переменной;
W(I) – регрессионный остаток;
S(1) – сумма квадратов регрессионных остатков (в программе обозначается символом AA);
S(2) – скорректированный коэффициент детерминации.

```
100 'Регрессионный анализ'
110 CLS: PRINT "Функция Кобба-Дугласа": PRINT
120 READ P, M
130 PRINT "Число точек (лет) ="; P
140 PRINT "Число параметров ="; M
150 M1 = M + 1
160 DIM X(P, M), Y(P), A(M, M1), S(4), SD(M)
170 DIM U(M), V(M), B(M), W(P)
180 'Считывание данных
190 FOR I = 1 TO P: READ Y(I)
200   FOR J = 1 TO M – 1: READ X(I, J)
210 NEXT J, I
220 'Преобразование переменных
230 FOR I = 1 TO P
240   Y(I) = LOG(Y(I)): X(I, 3) = LOG(X(I, 2))
250   X(I, 2) = LOG(X(I, 1)): X(I, 1) = 1
260 NEXT I
600 GOSUB 1000
610 GOSUB 2000
620 GOSUB 3000
630 PRINT: PRINT: INPUT "Нажмите ENTER"; X$
640 GOTO 4000


1000 '******** Подпрограммы **************
1010 FOR I = 1 TO M: FOR J = 1 TO M: S = 0
1020   FOR K = 1 TO P: S = S + X(K, I) * X(K, J): NEXT K
1030   A(I, J) = S: A(J, I) = S
1040 NEXT J: NEXT I
1050 FOR I = 1 TO M: S = 0
1060   FOR K = 1 TO P: S = S + X(K, I) * Y(K): NEXT K
1070   A(I, M1) = S
1080 NEXT I
1090 RETURN
2000 '------ Обращение матрицы ----------
2010 FOR I = 1 TO M: U(I) = 0: V(I) = 0: NEXT I
2020 FOR K = 1 TO M
2030   XM = 0
2040   FOR J = 1 TO M
2050     IF U(J) <> 0 THEN 2100
2060     FOR I = 1 TO M
2070       IF V(I) <> 0 OR ABS(A(I, J)) < XM THEN 2090
```

```
2080      XM = ABS(A(I, J)): Q = J: R = I
2090    NEXT I
2100  NEXT J
2110  U(Q) = R: V(R) = Q
2120  FOR J = 1 TO M: IF J = Q THEN 2170
2130    A(R, J) = A(R, J) / A(R, Q)
2140    FOR I = 1 TO M: IF I = R THEN 2160
2150      A(I, J) = A(I, J) – A(R, J) * A(I, Q)
2160    NEXT I
2170  NEXT J
2180  A(R, Q) = 1 / A(R, Q)
2190  FOR I = 1 TO M: IF I = R THEN 2210
2200    A(I, Q) = -A(R, Q) * A(I, Q)
2210  NEXT I
2220 NEXT K
2230 FOR K = 1 TO M
2240  IF K = U(K) THEN 2280
2250  FOR J = 1 TO M: SWAP A(K, J), A(U(K), J): NEXT J
2260  FOR I = 1 TO M: SWAP A(I, K), A(I, V(K)): NEXT I
2270  U(V(K)) = U(K): V(U(K)) = V(K): U(K) = K: V(K) = K
2280 NEXT K
2290 RETURN
3000 '---------- Результат -------------
3010 FOR I = 1 TO M: S = 0
3020  FOR J = 1 TO M: S = S + A(I, J) * A(J, M1): NEXT J
3030  B(I) = S
3040 NEXT I
3050 PRINT: PRINT "   Log a(0)";
3060 FOR I = 2 TO M: PRINT "     a("; I – 1; ")";: NEXT I
3070 PRINT
3080 FOR I = 1 TO M: PRINT B(I),: NEXT I
3090 S = 0
3100 FOR I = 1 TO P: S = S + Y(I): NEXT I
3110 YM = S / P: AA = 0: S = 0
3120 FOR I = 1 TO P
3130  A = 0: S = S + (Y(I) – YM) ^ 2
3140  FOR J = 1 TO M: A = A + X(I, J) * B(J): NEXT J
3150  W(I) = Y(I) – A
3160  AA = AA + (Y(I) – A) ^ 2
3170 NEXT I
3180 S(1) = AA: S(3) = AA / (P – M)
3190 S(2) = 1 – S(3) * (P – 1) / S
3200 S = 0
```

```
3210 FOR I = 2 TO P: S = S + (W(I) – W(I – 1)) ^ 2: NEXT I
3220 S(4) = S / AA
3230 PRINT
3240 FOR I = 1 TO M
3250   SD(I) = SQR(S(3) * A(I, I))
3260   PRINT "("; SD(I); ")",
3270 NEXT I
3280 PRINT: PRINT "a( 0)="; EXP(B(1))
3290 PRINT "R ="; S(1); " DW ="; S(4); " *R2 ="; S(2)
3300 RETURN
4000 '----------- График --------------------
4010 SCREEN 2: CLS: KEY OFF
4020 L = 24
4030 DIM T$(L), A$(2), L(2)
4040 L(1) = &HFFFF: L(2) = &HAAAA
4050 A$(1) = "Реальные значения": A$(2) = "Оценочные значения"
4060 FOR J = 1 TO L / 4: T$(J) = STR$(1896 + J * 4): NEXT J
4070 FOR J = 1 TO L
4080   X(J, 1) = EXP(Y(J)): X(J, 2) = X(J, 1) * EXP(-W(J))
4090 NEXT J
4100 MAX = 250
4110 LINE (48, 8)-(48, 351 / 2): LINE -(552, 351 / 2)
4120 FOR J = 1 TO L / 4
4130   LINE (48 * J + 24, 343 / 2)-(48 * J + 24, 351 / 2)
4140     FOR K = 1 TO 3
4150     LINE (48 * J + 24 + 12 * K, 347 / 2)-(48 * J + 24 + 12*K, 351/ 2)
4160     NEXT K
4170   LOCATE 24, 6 * J + 1: PRINT T$(J);
4180 NEXT J
4190 FOR I = 31 TO 351 STEP 64
4200  LINE (48, I / 2)-(52, I / 2)
4210    LOCATE 2 + 4 * (I – 31) / 64, 1
4220    PRINT USING "###"; INT(MAX – MAX * (I – 31) / 320);
4230 NEXT I
4240 FOR I = 1 TO 2
4250    LINE (546, 25 * I + 112)-(590, 25 * I + 112), 1,, L(I)
4260    LOCATE 3 * I + 15, 50: PRINT A$(I)
4270    PSET (72, 351 / 2 – INT(X(1, I) / MAX * 160)), 1
4280    FOR J = 1 TO L
4290      X = INT(X(J, I) / MAX * 160)
4300      LINE -(12 * J + 60, 351 / 2 – X), 1,, L(I)
4310    NEXT J
4320 NEXT I
```

```
4330 LOCATE 1, 25: PRINT 'Реальные и оценочные значения"
4340 GOTO 4340
5000 '*** Данные ***
5010 DATA 24,3
5020 DATA      100.0,      100.0,      100.0
5030 DATA      101.0,      107.0,      104.8
5040 DATA      112.0,      114.0,      110.0
5050 DATA      122.0,      122.0,      117.2
5060 DATA      124.0,      131.0,      121.9
5070 DATA      122.0,      138.0,      115.6
5080 DATA      143.0,      149.0,      125.0
5090 DATA      152.0,      163.0,      134.2
5100 DATA      151.0,      176.0,      139.9
5110 DATA      126.0,      185.0,      123.2
5120 DATA      155.0,      198.0,      142.7
5130 DATA      159.0,      208.0,      147.0
5140 DATA      153.0,      216.0,      148.1
5150 DATA      177.0,      226.0,      155.0
5160 DATA      184.0,      236.0,      156.2
5170 DATA      169.0,      244.0,      152.2
5180 DATA      189.0,      266.0,      155.8
5190 DATA      225.0,      298.0,      183.0
5200 DATA      227.0,      335.0,      197.5
5210 DATA      223.0,      366.0,      201.1
5220 DATA      218.0,      387.0,      195.9
5230 DATA      231.0,      407.0,      194.4
5240 DATA      179.0,      417.0,      146.4
5250 DATA      240.0,      431.0,      160.5
```

3.2. C++

```
//-------------------------------------------------------------
#include <iostream.h>
#include <math.h>
#include <conio.h>
#include <graphics.h>
#include <stdio.h>
//-------------------------------------------------------------
#define M  3
#define M 1 4
#define P  24
//-------------------------------------------------------------
float XTXANDXTY(float X[P][M], float A[M][M1], float Y[P]);
```

```
float XTXINV(float A[M][M1]);
void plotGraphic(float,float,float[],float[]);
//----------------------------------------------------------

main()
{
//Описания и данные
  float X[P][M],Ylog[P],Yapp[P];
  float A[M][M1];
  float B[M];
  float K[P]={100,107,114,122,131,138,149,163,176,185,198,208,
                  216,226,236,244,266,298,335,366,387,407,417,431};
  float L[P]={100,104.8,110,117.2,121.9,115.6,125,134.2,139.9,
                  123.2,142.7,147,148.1,155,156.2,152.2,155.8,183,
                  197.5,201.1,195.9,194.4,146.4,160.5};
  float Y[P]={100,101,112,122,124,122,143,152,151,126,155,159,
                  153,177,184,169,189,225,227,223,218,231,179,240};

//Вычисление параметров
  for(int i=0; i<P; i++)
  { Ylog[i]=log(Y[i]);
    X[i][0]=1.;  X[i][1]=log(K[i]);  X[i][2]=log(L[i]); }
    XTXANDXTY(X,A,Ylog);
    XTXINV(A);
  for(int i=0; i<M; i++)
  { float s=0;
    for(int j=0; j<M; j++)
      s+=A[i][j]*A[j][M1-1];
    B[i]=s;
  }

//Вывод значений параметров
  cout<<"\n\n\n";
  for(int i=0; i<M; i++)
    cout << "    B["<<i<<"] = " <<B[i]<<endl;
  getch();

//Аппроксимирующая кривая
  for(int i=0; i<P; i++)
    Yapp[i]=exp(B[0])*pow(K[i],B[1])*pow(L[i],B[2]);

//Открытие графического режима
  int dr = DETECT, mod;
```

```
  initgraph(&dr, &mod, "D:\\bc5\\bgi");

//Графики
  settextstyle(SMALL_FONT,HORIZ_DIR,0);
  outtextxy(150,50,"Original data");
  outtextxy(150,80,"Approximation");
  setcolor(YELLOW);
  line(250,57,300,57);
  setcolor(LIGHTRED);
  line(250,87,300,87);
  setcolor(WHITE);
  plotGraphic(0,23,Y,Yapp);
  getch();

//Закрытие графического режима
 closegraph();
return 0;
}
//---------  Функции  ----------------------------------------
//Матрицы XTX и XTY
float XTXANDXTY(float X[P][M], float A[M][M1], float Y[P])
{   for(int i=0; i<M; i++)
       for(int j=0; j<M; j++)
       { float s=0;
         for(int k=0; k<P; k++)
          s+=X[k][i]*X[k][j];
         A[i][j]=s;  A[j][i]=s;
       }
    for(int i=0; i<M; i++)
     { float s=0;
       for(int k=0; k<P; k++)
         s+=X[k][i]*Y[k];
       A[i][M1-1]=s;
     }
    return 0;
  }
//-----------------------------------------------------------
// Обращение матрицы
float XTXINV(float A[M][M1])
  { int U[M],V[M];  int q,r;
    for(int i=0; i<M; i++)
    { U[i]=0; V[i]=0; }
    for(int k=0; k<M; k++)
```

```
  { float xm=0;
    for(int j=0; j<M; j++)
    { if (U[j]!=0)  continue;
      for(int i=0; i<M; i++)
      { if (V[i]!=0 || fabs(A[i][j])<xm)  continue;
        xm=fabs(A[i][j]);
        q=j;  r=i;
      }
    }
  U[q]=r;  V[r]=q;
  for (int j=0; j<M; j++)
  { if (j==q)  continue;
    A[r][j]/=A[r][q];
    for(int i=0; i<M; i++)
    { if (i==r)  continue;
      A[i][j]-=A[r][j]*A[i][q];
    }
  }
  A[r][q]=1/A[r][q];
  for(int i=0; i<M; i++)
  { if (i==r)  continue;
    A[i][q]*=-A[r][q];
  }
 }
 for(int k=0; k<M; k++)
 { float t; int kk;
  if (k==U[k])  continue;
  for (int j=0; j<M; j++)
  { kk=U[k];  t=A[k][j];  A[k][j]=A[kk][j];  A[kk][j]=t; }
  for (int i=0; i<M; i++)
  { kk=V[k];  t=A[i][k];  A[i][k]=A[i][kk];  A[i][kk]=t; }
  kk=V[k];  U[kk]=U[k];
  kk=U[k];  V[kk]=V[k];
  U[k]=k;  V[k]=k;
 }
 return 0;
}
//----------------------------------------------------------
//Построение графика функции [10]
void plotGraphic(float a,float b,float f[],float ff[])
{ float xStep=pow(10, floor(log(b-a)/log(10.0)));
 float xMin=xStep*floor(a/xStep);
 float xMax=xStep*ceil(b/xStep);
```

```
float *fVal=new float[24];
float *ffVal=new float[24];
for(int i=0; i<24; i++)
  { fVal[i]=f[i];  ffVal[i]=ff[i]; }
float  yMin=fVal[0],  yMax=fVal[0];
for(int i=1; i<24; i++)
   if(fVal[i]<yMin)        yMin=fVal[i];
   else
   if(fVal[i]>yMax)        yMax=fVal[i];

float  yStep=pow(10, floor(log(yMax-yMin)/log(10.0)));
   yMin=yStep*floor(yMin/yStep);
   yMax=yStep*ceil(yMax/yStep);

int  x0=60;
int  x1=getmaxx()-20;
int  y0=10;
int  y1=getmaxy()-40;

line(x0,y0,x1,y0);
line(x1,y0,x1,y1);
line(x1,y1,x0,y1);
line(x0,y1,x0,y0);

float  kx=(x1-x0)/(xMax-xMin);
float  ky=(y1-x0)/(yMax-yMin);
float  x=a;
float  h=(b-a)/24.0;

setcolor(YELLOW);
circle(x0+(x-xMin)*kx,y0+(yMax-fVal[0])*ky,4);
for(int i=1; i<24; i++,x+=h)
  circle(x0+(x-xMin)*kx,y0+(yMax-fVal[i])*ky,4);
x=a;
moveto(x0+(x-xMin)*kx,y0+(yMax-fVal[0])*ky);
for(int i=1; i<24; i++,x+=h)
  lineto(x0+(x-xMin)*kx,y0+(yMax-fVal[i])*ky);
x=a;
setcolor(LIGHTRED);
moveto(x0+(x-xMin)*kx,y0+(yMax-ffVal[0])*ky);
for(int i=1; i<24; i++,x+=h)
  lineto(x0+(x-xMin)*kx,y0+(yMax-ffVal[i])*ky);
setcolor(WHITE);
```

```
  char   str[128];
  settextstyle(SMALL_FONT,HORIZ_DIR,0);
  for(x=xMin;x<=xMax;x+=xStep)
  {  int  ix=x0+(x-xMin)*kx;
     line(ix,y1,ix,y1+10);
     if(x+xStep<=xMax)
       for(int i=1; i<10;i++)
          line(ix+i*xStep*kx*0.1,y1,ix+i*xStep*kx*0.1,y1+5);
     sprintf(str,"%g",x);
     outtextxy(ix-textwidth(str)/2,y1+15,str);
  }
  for(float y=yMin;y<=yMax;y+=yStep)
  {  int  iy=y0+(yMax-y)*ky;
     line(x0-10,iy,x0,iy);
     if(y+yStep<=yMax)
       for(int i=1; i<10;i++)
          line(x0-5,iy-i*yStep*ky*0.1,x0,iy-i*yStep*ky*0.1);
     sprintf(str,"%g",y);
     outtextxy(x0-10-textwidth(str),iy,str);
  }
  delete fVal;
}
```

3.3. Borland C++Builder

На форму вызваны четыре компонеты ListBox для вывода данных, кнопка Button для запуска программы и Chartfx для построения графика.

```
#ifndef cobdugH
#define cobdugH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <chartfx3.hpp>
#include <OleCtrls.hpp>
//---------------------------------------------------------------------------
class TForm1: public TForm
{   published:         // IDE-managed Components
        TListBox *ListBox1;
```

```
        TChartfx *Chartfx1;
        TButton *Button1;
        TListBox *ListBox2;
        TListBox *ListBox3;
        TListBox *ListBox4;
        void __fastcall Button1Click(TObject *Sender);
private:            // User declarations
public:            // User declarations
    __fastcall TForm1(TComponent* Owner);
   float __fastcall XTXANDXTY(float X[24][3], float A[3][4], float Y[24]);
   float __fastcall XTXINV(float A[3][4]);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm1 *Form1;
//---------------------------------------------------------------------------
#endif


//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <math.h>
#include "cobdug.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{ char out[6];
  float y;
  int M=3, M1=4, P=24;
  float App[24], A[3][4];
  float B[3];
  float X[24][3]={ 1,100,100, 1,107,104.8, 1,114,110, 1,122,117.2,
                1,131,121.9, 1,138,115.6, 1,149,125, 1,163,134.2,
                1,176,139.9, 1,185,123.2, 1,198,142.7, 1,208,147,
```

```
           1,216,148.1, 1,226,155, 1,236,156.2, 1,244,152.2,
           1,266,155.8, 1,298,183, 1,335,197.5, 1,366,201.1,
           1,387,195.9, 1,407,194.4, 1,417,146.4, 1,431,160.5};
 float Y[24]={100,101,112,122,124,122,143,152,151,126,155,159,
            153,177,184,169,189,225,227,223,218,231,179,240};
 for(int i=0; i<P; i++)
 { Y[i]=log(Y[i]); X[i][1]=log(X[i][1]);  X[i][2]=log(X[i][2]); }
 XTXANDXTY(X,A,Y);
 XTXINV(A);
 for(int i=0; i<M; i++)
 { float s=0;
   for(int j=0; j<M; j++)   s+=A[i][j]*A[j][M1-1];
   B[i]=s;
 }
 for(int i=0; i<P; i++)
  App[i]=exp(B[0])*pow(exp(X[i][1]),B[1])*pow(exp(X[i][2]),B[2]);
   //Открытие данных для графика
    Chartfx1->OpenDataEx(COD_VALUES,1,24);
    Chartfx1->OpenDataEx(COD_XVALUES,1,24);  //Только  для  графика
scatter
   //Надписи над колонками таблиц
    ListBox1->Items->Append("     K");
    ListBox2->Items->Append("     L");
    ListBox3->Items->Append("     Y");
    ListBox4->Items->Append("Approxim");
   //Заполнение серии для графика
   for(int i=0; i<24; i++)
   {y=App[i];
    Chartfx1->ThisSerie=0;
    Chartfx1->Value[i]=y;
    Chartfx1->ThisSerie=1;
    Chartfx1->Value[i]=exp(Y[i]);
    Chartfx1->ThisSerie=0;
    Chartfx1->XValue[i]=(float)i;
    //Форматирование выводимых величин
    sprintf(out, "%f", exp(X[i][1]));
    ListBox1->Items->Append(out);
    sprintf(out, "%f", exp(X[i][2]));
    ListBox2->Items->Append(out);
    sprintf(out, "%f", exp(Y[i]));
    ListBox3->Items->Append(out);
    sprintf(out, "%f", y);
    ListBox4->Items->Append(out);
```

```
   }
Form1->Print();
 //Закрытие данных графика
 Chartfx1->CloseData(COD_VALUES);
 Chartfx1->CloseData(COD_XVALUES);
}
//---------------------------------------------------------------------------
float __fastcall TForm1::XTXANDXTY(float X[24][3], float A[3][4], float
Y[24])
//Матрицы XTX и XTY
{ int M=3, M1=4, P=24;
   for(int i=0; i<M; i++)
     for(int j=0; j<M; j++)
        { float s=0;
          for(int k=0; k<P; k++)
            s+=X[k][i]*X[k][j];
          A[i][j]=s;  A[j][i]=s;
        }
    for(int i=0; i<M; i++)
      { float s=0;
        for(int k=0; k<P; k++)
          s+=X[k][i]*Y[k];
        A[i][M1-1]=s;
      }
    return 0;
}
float __fastcall TForm1::XTXINV(float A[3][4])
{ int M=3, M1=4, P=24;
   int U[3],V[3];  int q,r;
    for(int i=0; i<M; i++)
    { U[i]=0; V[i]=0; }
    for(int k=0; k<M; k++)
    { float xm=0;
      for(int j=0; j<M; j++)
      { if (U[j]!=0)  continue;
        for(int i=0; i<M; i++)
        { if (V[i]!=0 || fabs(A[i][j])<xm)  continue;
          xm=fabs(A[i][j]);
          q=j;  r=i;
        }
      }
    U[q]=r;  V[r]=q;
    for (int j=0; j<M; j++)
    { if (j==q)  continue;
```

```
     A[r][j]/=A[r][q];
     for(int i=0; i<M; i++)
     { if (i==r)  continue;
       A[i][j]-=A[r][j]*A[i][q];
     }
   }
   A[r][q]=1/A[r][q];
   for(int i=0; i<M; i++)
   { if (i==r)  continue;
     A[i][q]*=-A[r][q];
   }
  }
  for(int k=0; k<M; k++)
  { float t; int kk;
   if (k==U[k])  continue;
   for (int j=0; j<M; j++)
   { kk=U[k];  t=A[k][j];  A[k][j]=A[kk][j];  A[kk][j]=t; }
   for (int i=0; i<M; i++)
   { kk=V[k];  t=A[i][k];  A[i][k]=A[i][kk];  A[i][kk]=t; }
   kk=V[k];  U[kk]=U[k];
   kk=U[k];  V[kk]=V[k];
   U[k]=k;  V[k]=k;
  }
  return 0;
}
```

3.4. Mathcad

Вычисления в оболочке Mathcad просты, так как операции с индексными переменными легко выполняются. Расчет по формуле (18) не представляет трудностей, достаточно лишь сформировать необходимые векторы и матрицы. Приводится лишь начало страницы расчетов с исходными данными и формированием матрицы X.

Аппроксимация функцией Кобба-Дугласа:

431   160.5   240 $Ylg_i := \log(Y_i)$

Результаты расчетов в среде MathCAD представлены на рис. 1 и 2.

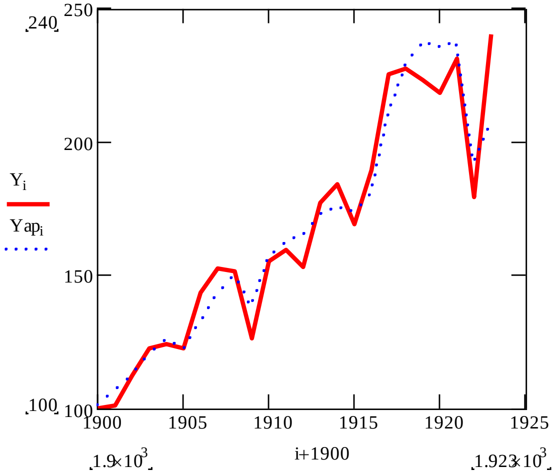

Рис. 1. Объемы выпусков промышленности в зависимости от года: сплошная кривая – исходные данные, пунктирная – аппроксимирующая кривая

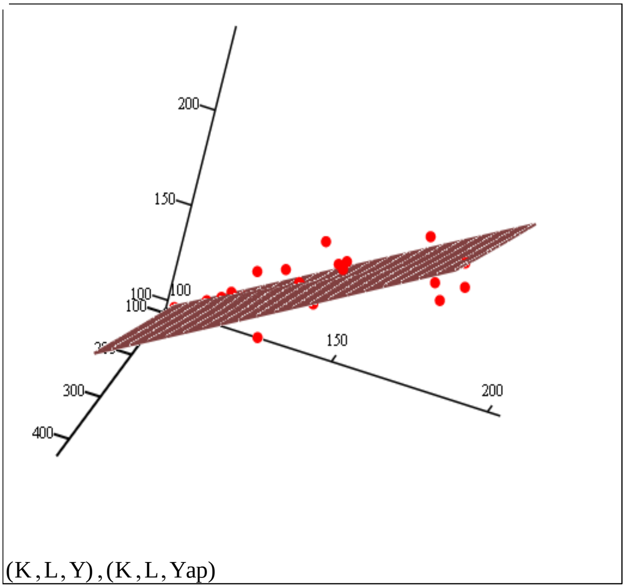

Рис. 2. Трехмерное изображение объемов выпусков в зависимости от капитала и труда: по осям отложены: значения объемов капитала – K, объемов труда – L и объемов выпусков – Y. Представлены исходные данные (точки) и аппроксимирующая поверхность

4. Задания

1. Вычислите значения параметров c, α, β в производственной функции (1) на основе метода наименьших квадратов, примененного к преобразованному уравнению (19). Полный текст программы приведен в разделе 3.

2. Взяв за основу эту программу, преобразуйте ее для оценки параметров c, α производственной функции Кобба-Дугласа (3) при условии, что $\alpha+\beta=1$;

3. Далее преобразуйте программу в программу оценки параметров c, α, λ в производственной функции (4) для случая, учитывающего технический прогресс.

4. Убедитесь в том, что получены наилучшие значения параметров. Для этого измените на несколько процентов значение одного из параметров (например c) в сторону увеличения и уменьшения. Как будет изменяться график аппроксимирующей кривой?

5. Если Вы работаете в среде Mathcad, начертите исходные данные по объемам выпусков, а также все полученные аппроксимирующие зависимости как функции времени (номера итерации) на одном графике. Какие выводы можно сделать из этих графиков?

6. Если Вы работаете в среде Mathcad – 2000, постройте трехмерные графики исходных данных – Y = Y(K,L) и аппроксимирующую поверхность (любую из трех рассмотренных случаев) на одном графике.

МОДЕЛЬ ОБЩЕГО РАВНОВЕСИЯ

1. Базовые положения модели

На основе идеализированной замкнутой экономики модель имитирует динамику стабилизации рыночных цен, объемов выпусков и потребления через обмен товарами и факторами производителей и потребителей [1].

В модели представлены два хозяйственных субъекта: «производитель 1» и «производитель 2» (рис. 1), каждый из которых потребляет один ресурс объемом L_1^P и L_2^P и выпускает один вид продукции конечного спроса Y_1^S и Y_2^S, соответственно. Будем считать, что объемы производства определяются производственными функциями [2, 3, 11]

$$Y_i^S = c_i (L_i^D)^{a_i}, \quad i = 1{,}2; \tag{1}$$

где c_i и a_i – коэффициенты производственных функций, $c_i > 0$, а $0 < a_i < 1$.

Обобщенный потребитель обуславливает спрос на выпускаемую продукцию – Y_1^D, Y_2^D и поставляет на рынок факторов постоянный объем ресурса $L^S \geq L_1^D + L_2^D$, где равенство относится к равновесному состоянию рынка. Будем считать, что состояние потребителя определяется функцией полезности [3, 11]

$$U(Y_1^D, Y_2^D) = b_1 \ln(Y_1^D) + b_2 \ln(Y_2^D), \tag{2}$$

где коэффициенты $b_1, b_2 > 0$.

Предполагается, что $Y_i^S \geq Y_i^D$, где равенство соответствует условию равновесия.

Цены на товары $p_1(t), p_2(t)$ и ресурс $\omega(t)$, так же как и объемы выпусков $Y_1^S(t), Y_2^S(t)$ и потребления $Y_1^D(t), Y_2^D(t)$, могут меняться с течением времени *t*. Функционирование рынков во времени и установление экономического равновесия, когда цены стабильны, а спрос на товары и факторы равен предложению ($Y_1^D = Y_1^S$, $Y_2^D = Y_2^S$ и $L_1^D + L_2^D = L^S$), в модели можно обеспечить с помощью итерационной (повторяющейся) процедуры, схема которой приведена на рис. 2. Для этого, введя единичный лаг во времени, будем фиксировать состояния экономики в дискретные моменты времени 0,1, 2, …, (t-1), t, (t+1), …, которые в модели соответствуют номеру итерации.

В начальный момент времени $t = 0$ задаются цены $p_1(0)$, $p_2(0)$ на товары и цена $\omega(0)$ на фактор, исходя из каких-то априорных оценок. Конечно, начальные цены не являются равновесными, но желательно, чтобы они отражали примерную величину цен. От их близости к равновесным ценам зависит сходимость итерационной процедуры. При значительном отличии начальных оценок для цен от их равновесных значений получить равновесное состояние рынка в данной модели проблематично.

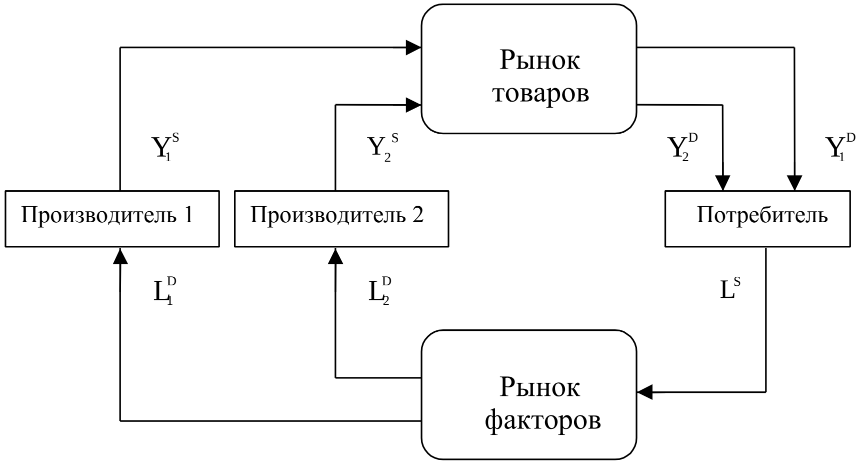

Рис. 1. Функциональная схема модели общего равновесия

Стратегия производителей заключается в получении максимальной прибыли. Для этого, исходя из предложенных цен на товары и факторы и своих производственных мощностей, задаваемых производственными функциями (1), они определяют оптимальные объемы закупаемых факторов $L_1^{D*}(t)$, $L_2^{D*}(t)$ и, следовательно, оптимальные объемы выпускаемой продукции $Y_1^{S*}(t), Y_2^{S*}(t)$, которые обеспечивают им максимальную прибыль в рассматриваемом временном интервале (на данной итерации).

Потребитель стремится обеспечить себе максимум благополучия, продавая фактор на рынке факторов и покупая товары на товарном рынке. Для этого потребитель, при заданном уровне цен и наличии ограниченного дохода от продажи факторов, определяет такие объемы $Y_1^{D*}(t)$, $Y_2^{D*}(t)$ покупаемых товаров, которые обеспечивают максимум функции полезности (2).

Так как цены на товары и фактор изначально неравновесные, то объемы спроса и предложения не совпадают и, следовательно, возникает отличный от нуля избыточный спрос:

$$E_i(t) = Y_i^{D*}(t) - Y_i^{S*}(t), \qquad i = 1,2. \qquad (3)$$

Он может быть как положительным, так и отрицательным.

Чтобы изменить ситуацию в экономике, *необходимо изменить первоначальные цены, увеличив или уменьшив их в соответствии с величиной и знаком избыточного спроса*, чтобы объемы производства товаров стремились к объемам спроса на них: $Y_1^S \to Y_1^D$, $Y_2^S \to Y_2^D$. Если избыточный спрос положительный (спрос превышает предложение), то цены необходимо увеличивать. В случае отрицательного избыточного спроса цены должны быть уменьшены.

С измененными ценами можно повторить рассуждения (расчеты) сначала. Этот итеративный (повторяющийся) процесс можно продолжать до тех пор, пока цены не перестанут изменяться.

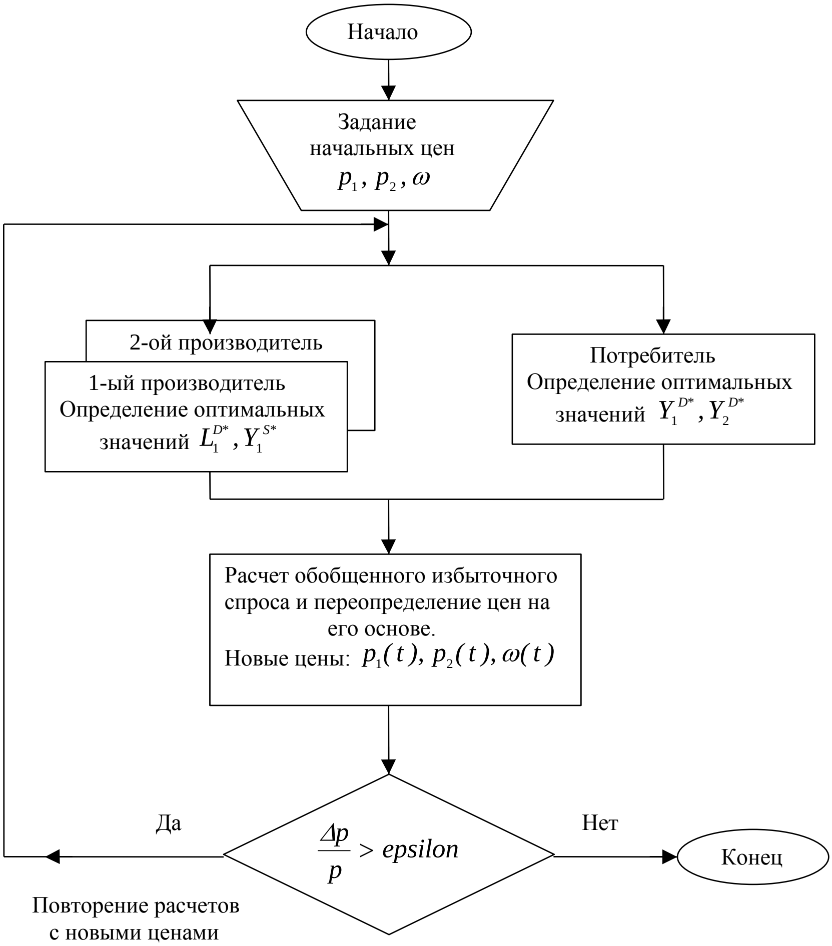

Рис. 2. Схема итерационного процесса

2. Математическая формулировка модели

Основные задачи, которые необходимо решить на данном этапе в рамках предложенной блок-схемы, следующие:

1. Разработать способ нахождения значений $L_i^{D*}(t)$ и $Y_i^{S*}(t)$, i = 1,2, определяющих наилучшие состояния производителей при заданных ценах.
2. Разработать способ определения значений $Y_i^{D*}(t)$, i = 1,2, соответствующих наилучшему состоянию обобщенного потребителя при тех же ценах.
3. Определить способ изменения цен в сторону равновесных.

2.1. Поведение производителя

Как отмечалось выше, введем единичный лаг во времени и будем фиксировать состояния нашей экономической системы в дискретные моменты времени 0, 1, 2, …, t-1, t, t+1, …, которые в модели будут соответствовать номеру итерации.

Пусть в момент времени t заданы цены на товары $p_1(t)$, $p_2(t)$ и цена на фактор $\omega(t)$. Полагаем, что стратегия производителей заключается в достижении максимальной прибыли. Планируемая прибыль i-го производителя $\pi_i(t)$, i=1,2 равна разности между планируемым доходом $p_i(t)Y_i^S(t)$ и расходом $\omega \cdot L_i^D$:

$$\pi_i(t) = p_i(t)Y_i^S(t) - \omega(t)L_i^D(t), \tag{4}$$

или в более кратком виде:

$$\pi_i = p_i Y_i^S - \omega L_i^D .$$

Для упрощения записи в последнем выражении мы опустили (и будем опускать далее, пока не понадобится обратное) явное подчеркивание зависимости переменных от времени, имея в виду, что такая зависимость существует.

Функция, отражающая ограниченные возможности производителя, – это производственная функция

$$c_i(L_i^D)^{a_i} = Y_i^S .$$

Необходимо найти наилучшее решение для производителя (максимальную прибыль) при наличии ограничения, записанного в виде

$$Y_i^S - c_i(L_i^D)^{a_i} = 0\,. \qquad (5)$$

Итак, необходимо найти максимум π_i, варьируя Y_i^S и L_i^D, при наличии ограничений (5):

$$\begin{cases} max\ \pi_i \\ Y_i^S, L_i^D; \\ Y_i^S - c_i(L_i^D)^{a_i} = 0; \\ Y_i^S \geq 0,\ L_i^D \geq 0. \end{cases}$$

Данную задачу можно решить с помощью метода множителей Лагранжа [3, 11]. Введем функцию Лагранжа:

$$Lag_i^{np}(Y_i^S, L_i^D) = p_i Y_i^S - \omega L_i^D + y_i^{np}(Y_i^S - c_i(L_i^D)^{a_i})\,,$$

где $p_i \cdot Y_i^S - \omega \cdot L_i^D$ – целевая функция, максимум которой мы ищем, а y_i^{np} – множитель Лагранжа, дополнительная неизвестная величина.

Дифференцируя функцию Лагранжа по переменным Y_i^S, L_i^D и y_i^{np} и приравнивая производные нулю, получаем систему уравнений:

$$\begin{cases} \dfrac{\partial Lag_i^{np}}{\partial Y_i^S} = p_i + y_i^{np} = 0, \\ \dfrac{\partial Lag_i^{np}}{\partial L_i^D} = -\omega - y_i^{np} c_i a_i (L_i^D)^{a_i - 1} = 0, \\ \dfrac{\partial Lag_i^{np}}{\partial y_i^{np}} = Y_i^S - c_i (L_i^D)^{a_i} = 0. \end{cases} \qquad (6)$$

Из первого уравнения найдем y_i^{np} и подставим во второе, тогда

$$\omega = p_i c_i a_i (L_i^D)^{a_i - 1},$$

а наилучшая оценка для L_i^D равна

$$\boxed{L_i^{D*} = \left(\frac{p_i c_i a_i}{\omega}\right)^{\frac{1}{1-a_i}}}. \qquad (A)$$

Подставляя L_i^{D*} в третье уравнение системы, имеем

$$\boxed{Y_i^{S*} = c_i(L_i^{D*})^{a_i}}. \qquad (A^*)$$

Полученные значения L_i^{D*} и Y_i^{S*} обеспечивают максимум прибыли i-го производителя при заданных ценах p_1, p_2, ω на товары и фактор для уровня производства, определенного функцией (1) с коэффициентами a_i, c_i.

2.2. Поведение потребителя

2.2.1. Функция Лагранжа для потребителя

Стратегия потребителя, как отмечалось выше, заключается в обеспечении максимума функции полезности:

$$U(Y_1^D, Y_2^D) = b_1 \ln(Y_1^D) + b_2 \ln(Y_2^D), \tag{7}$$

где b_1 и b_2 – постоянные коэффициенты, а потребитель может менять объемы закупаемых товаров – Y_1^D и Y_2^D. При этом он ограничен существующим уровнем цен на товары, а также величиной возможного дохода ωL^S:

$$p_1 Y_1^D + p_2 Y_2^D \le \omega L^S. \tag{8}$$

Математическая формулировка этой задачи следующая: необходимо найти максимум целевой функции (7) при наличии ограничения (8), причем $Y_1^D, Y_2^D \ge 0$, т.е.

$$\begin{cases} \max\limits_{Y_1^D, Y_2^D;} U(Y_1^D, Y_2^D) \\ \omega L^S - p_1 Y_1^D - p_2 Y_2^D \ge 0; \\ Y_1^D \ge 0, Y_2^D \ge 0. \end{cases} \tag{9}$$

Функция Лагранжа в этом случае имеет вид:

$$Lag^{потр}(Y_1^D, Y_2^D) = U(Y_1^D, Y_2^D) + y^{потр}(\omega L^S - p_1 Y_1^D - p_2 Y_2^D). \tag{10}$$

В качестве устойчивого численного способа определения оптимальных значений Y_1^{D*} и Y_2^{D*} в (9) может быть использован градиентный метод [12]. Рассмотрим сущность градиентного метода, чтобы затем применить его к нашей задаче.

2.2.2. Градиентный метод

Предположим, что график функции Лагранжа (10) представляет собой некоторую выпуклую, без локальных возвышений, поверхность с максимальной точкой $Lag_{max}^{потр}(Y_1^{D*}, Y_2^{D*})$. Необходимо из текущей точки А со значением функции в ней $Lag_A^{потр}(Y_{1A}^D, Y_{2A}^D) < Lag_{max}^{потр}(Y_1^{D*}, Y_2^{D*})$, наилучшим способом перейти в точку $Lag_{max}^{потр}$. Для этого необходимо двигаться в направлении наибольшего изменения функции Лагранжа, то есть изменять координаты Y_1^D и Y_2^D в направлении градиента функции:

$$grad\vec{d}\ Lag^{nomp} = \frac{\partial Lag^{nomp}}{\partial Y_1^D}\vec{i} + \frac{\partial Lag^{nomp}}{\partial Y_2^D}\vec{j}\,, \tag{12}$$

где $\vec{i}$ и $\vec{j}$ – единичные векторы по осям Y_1^D и Y_2^D.

Если рассматривать изменяющиеся координаты Y_1^D и Y_2^D как функции времени, то необходимо изменять их так, чтобы

$$\lim_{t\to\infty} Y_i^D(t) = Y_i^{D*} = const, \qquad i = 1{,}2. \tag{13}$$

Для нахождения Y_i^{D*} численно решаются дифференциальные уравнения

$$\frac{dY_i^D(t)}{dt} = f(Y_i^D)$$

при некоторых заданных начальных условиях. Решения должны удовлетворять условию (13). В качестве функций $f(Y_i^D)$ выбирают производные функции Лагранжа, т.е. решаются уравнения

$$\frac{dY_i^D(t)}{dt} = \beta_i\,\frac{\partial Lag^{nomp}(Y_i^D)}{\partial Y_i^D}, \qquad i = 1{,}2; \tag{14}$$

где β_i – подбираемые при решении коэффициенты.

Если лаг во времени единичный ($dt = 1$), то, переходя к разностному уравнению, имеем

$$Y_i^D(t+1) - Y_i^D(t) \approx \beta_i\left(\frac{\partial Lag^{nomp}}{\partial Y_i^D}\right).$$

Значения коэффициентов β_i подбираются при численном решении, поэтому знак приближенного равенства можно заменить символом точного равенства. Тогда имеем

$$Y_i^D(t+1) = Y_i^D(t) + \beta_i\left(\frac{\partial Lag^{nomp}}{\partial Y_i^D}\right). \tag{15}$$

Новое значение i-й координаты отличается от предыдущего на величину, пропорциональную производной от функции Лагранжа по этой координате. Такая стратегия позволяет удовлетворить условию (13) и, в конечном счете, получить значения Y_i^{D*}, i = 1,2.

2.2.3. Реализация градиентного метода в модели

Вернемся к рассмотрению поведения потребителя. Подставим в выражение (15) функцию Лагранжа (10):

$$Y_i^D(t+1) = Y_i^D(t) + \beta_i \left\{ \frac{\partial U(Y_1^D, Y_2^D)}{\partial Y_i^D} - y^{потр} p_i \right\}. \tag{16}$$

Следует сделать несколько замечаний относительно этого выражения.

Во-первых, поскольку коэффициент β_i подбирается в численных вычислениях, реализующих последовательные приближения, можно положить $y^{потр} = 1$.

Во-вторых, производная для функции полезности (7) равна

$$\frac{\partial U(Y_i^D)}{\partial Y_i^D} = \frac{b_i}{Y_i^D}.$$

И, в-третьих, так как в фигурных скобках стоит разность между двумя величинами, не исключено, что результат в фигурных скобках окажется отрицательным и по абсолютной величине больше $Y_i^D(t)$. Необходимо исключить возможность получения отрицательных значений, так как общий результат (16) не должен быть отрицательным. При численных расчетах вводится условие: если $Y_i^D < 0$, то оно приравнивается нулю.

С учетом этих трех замечаний, выражение для $Y_i^D(t)$ будет иметь вид

$$\boxed{Y_i^D(t+1) = max\left[Y_i^D(t) + \beta_i \left(\frac{b_i}{Y_i^D(t)} - p_i \right),\ 0 \right]}. \tag{В}$$

Если первое выражение в квадратных скобках больше нуля, то для $Y_i^D(t+1)$ принимается это значение, если же оно меньше нуля, то $Y_i^D(t+1)$ считается равным нулю.

Выражение (В) обеспечивает реализацию градиентного метода и определяет изменяющиеся значения величины спроса в итерационной процедуре.

2.3. Регулирование цен

Для регулирования цен будем исходить из того факта, что изменение цены во времени связано с избыточным спросом. В частности, если цены возрастают, т.е. $\partial p / \partial t > 0$, то и избыточный спрос $E > 0$ (спрос превышает предложение, выражение (3)). Если цены падают, т.е. $\partial p/\partial t < 0$, то избыточный спрос отрицателен: $E < 0$ (предложение выше спроса). И, наконец, при стабильных ценах, когда $\partial p/\partial t = 0$, соответственно $E = 0$ (предложение равно спросу). Из этих соотношений можно сделать вывод, что производную от цены по времени можно связать с величиной избыточного спроса некоторой функциональной зависимостью

$$\frac{\partial p}{\partial t} = f(E). \tag{17}$$

В численных расчетах последовательных приближений для (17) можно принять простую линейную зависимость

$$\frac{\partial p_i}{\partial t} = \alpha_i E_i \quad i = 1,2;$$

с подбираемыми коэффициентами α_i.

Переходя к дискретной форме записи этой связи и вводя единичный лаг во времени, имеем:

$$\frac{p_i(t+1) - p_i(t)}{(t+1) - t} \approx \alpha_i E_i.$$

Значения коэффициентов α_i подбираются, поэтому опять, как было сделано выше, знак приближенного равенства заменяем символом точного равенства. В этом случае получаем

$$p_i(t+1) = p_i(t) + \alpha_i E_i. \tag{18}$$

Соотношение (18) по форме аналогично выражению (15). Принимая во внимание, что цена на товар (или фактор) не может быть отрицательной, окончательно, аналогично выражению (B), имеем для цен на товары

$$\boxed{p_i(t+1) = max\left[p_i(t) + \alpha_i\left(Y_i^D - Y_i^S\right), 0\right]}, \quad i = 1,2; \quad \text{(C)}$$

а для цены на фактор

$$\boxed{\omega(t+1) = max\left[\omega(t) + \gamma\left(L_1^D + L_2^D - L^S\right), 0\right]}, \quad \text{(D)}$$

где γ – коэффициент, аналогичный α_i.

По формулам (C) и (D) в модели будут рассчитываться цены на товары и фактор при отличных от нуля избыточных спросах.

3. Контрольные вопросы

1. Перечислите основные положения модели общего равновесия и напишите соответствующие им уравнения.
2. Начертите схему функционирования модели.
3. Начертите схему итерационного процесса для математического моделирования установления рыночного равновесия.
4. Сформулируйте оптимизационные задачи для потребителей и обобщенного потребителя. Приведите соответствующие функции Лагранжа.
5. Что такое градиентный метод? Где и как он используется в данной модели?
6. Как в модели осуществляется регулирование цен?

4. Программы

4.1. Алгоритм вычислений

В цикле с переменной цикла t, имитирующей изменение времени, по формулам (A) и (A*) рассчитываются оптимальные значения L_i^{D*} и Y_i^{S*}, $i = 1,2$; по формулам (B) производится вычисление Y_i^{D*}, $i = 1,2$; и по формулам (C) и (D) определяются новые цены. Окончание итерационной процедуры задается верхним значением для t и может быть изменено пользователем.

Признаком установления экономического равновесия является равенство нулю обобщенного избыточного спроса

$$\boxed{E^D = \omega(L_1^D + L_2^D - L^S) + p_1(Y_1^D - Y_1^S) + p_2(Y_2^D - Y_2^S)}. \qquad \text{(E)}$$

На каждом итерационном шаге определяется функция полезности (7).

В предлагаемых программах выпущены строки кода, соответствующие уравнениям (A) – (E). Пользователю предлагается дописать их самостоятельно в приемлемом для него варианте работы на QBASIC[6], C++[7], Borland C++ Builder[8] или MathCAD[9].

4.2. Список переменных

YS(I)= Y_i^S – объем предложения i-го продукта;

LS= L^S – объем предложения ресурса;

YD(I) = Y_i^d – размер спроса на i-продукт;

LD(I) = L_i^d – размер спроса i-го предприятия на ресурс;

P(I) = P_i – цена i-го продукта;
W = ω – цена ресурса;
A(I) = a_i, B(I) = b_i, C(I) = c_i – коэффициенты производственной функции и функции полезности;
AL(I) = α_i, BE(I) = β_i, GA = γ – коэффициенты в выражениях (B), (C), (D).
ED – обобщенный избыточный спрос;
U – функция полезности.

4.3. QBASIC

```
100 'МОДЕЛЬ ОБЩЕГО РАВНОВЕСИЯ [1]
110 SCREEN 2: CLS: KEY OFF
120 DIM YD(2), YS(2), LD(2), P(2), A(2), B(2), C(2), AL(2), BE(2)
130 READ TT, AL(1), AL(2), BE(1), BE(2), GA, LS
140 FOR I = 1 TO 2
150 READ A(I), B(I), C(I)
160 NEXT I
170 READ W
180 FOR I = 1 TO 2
190 READ P(I), YD(I)
200 NEXT I
210 PRINT "    альфа(1)="; AL(1); "  альфа(2)="; AL(2);
220 PRINT "  бета(1)="; BE(1); "  бета(2)="; BE(2)
230 PRINT " гамма="; GA; " YD(1)="; YD(1); "  YD(2)="; YD(2): PRINT
240 PRINT "   T    U       ED     W     P(1)   P(2)";
250 PRINT "  LD(1)  LD(2)   YS(1) YS(2)": PRINT
260 FOR T = 0 TO TT
270 '----------------------------------------------------
280 '----Расчет LD(I) и YS(I) /формулы (A), (A*),  предприятия/----
290
300
310
320
330 '----------------------------------------------------
340 '------Расчет YD(I) /формула (B), потребитель/-------
350
360
370
380
390 '----------------------------------------------------
400 '-----------Расчет ED и U /формулы (E) и (7)/-----------
```

```
410
420
430 '-------------Вывод результатов итерации--------------
440    GOSUB 600
450 '----------------------------------------------------
460 '--------Переопределение цен /формулы (C), (D)/---------
470
480
490
500
510
520
530 NEXT T
540 GOTO 540
600 '----------------------------------------------------
610 '------------------ПОДПРОГРАММА-----------------------
620 PRINT USING "####"; T;: PRINT USING "#####.##"; U;
630 PRINT USING "####.##"; ED; W; P(1); P(2);
640 PRINT USING "####.##"; LD(1); LD(2); YS(1); YS(2)
650 RETURN
800 '*** ДАННЫЕ ***
810 DATA 20,1,0.5,0.2,0.5,0.01,100
820 DATA 0.5,300,1
830 DATA 0.5,100,3
840 DATA 2.5
850 DATA 30,10
860 DATA 10,10
```

4.4. C++

```
#include <iostream.h>
#include <math.h>
#include <stdlib.h>
main()
{ float a[]={0.5,0.5}, b[]={300.,100.}, c[]={1.,3.};
  float alpha[]={1.,0.5}, beta[]={0.2,0.5}, gamma=0.01;
  float p[]={30.,10.}, omega=2.5;
  float Yd[]={10.,10.}, Ls=100.;
  float Ld[2],Ys[2],Ed,U;
  int tt=40;
  cout << "\n  t    U        Ed   omega   p[1]   p[2]  Ld[1]  Ld[2]  "
                                    <<"Ys[1]  Ys[2]\n\n";
  for(int t=0; t<=tt; t++)
```

```
 {
   //--- Расчет Ld[i] и Ys[i] – формулы (A),(A*), предприятия ---
   //------- Расчет Yd[i]  –  формула (B), потребитель ----------
   for(int i=0; i<2; i++)
    { …..
       …..
    }
   //---------- Расчет Ed и U – формулы (E) и (7) ---------------
       …..
   //----------- Вывод результатов итерации ---------------------
   cout.precision(2);
   cout.setf(0x1000);
   cout.width(3);
   cout<<t;
   cout.width(8);
   cout<<U;
   cout.width(8);
   cout<<Ed;
   cout.width(7);
   cout<<omega;
   cout.width(7);
   cout<<p[0];
   cout.width(7);
   cout<<p[1];
   cout.width(7);
   cout<<Ld[0];
   cout.width(7);
   cout<<Ld[1];
   cout.width(7);
   cout<<Ys[0];
   cout.width(7);
   cout<<Ys[1]<<endl;
//------- Переопределение цен – формулы (C) и (D) ----------------
   for(int i=0; i<2; i++)
       …..
 }
 cout << "\n  t    U        Ed   omega   p[1]   p[2]  Ld[1]  Ld[2]  "
                                  <<"Ys[1]  Ys[2]";
 return 0;
}
```

4.5. Borland C++Builder

На форму вызваны два компонента VCL – кнопка Button1 для запуска программы и ListBox1 для вывода данных.

```
#include <vcl.h>
#pragma hdrstop
#include "equil_model_BCB.h"
#include <math.h>
#include <stdio.h>
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
   : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1:: Button1Click(TObject *Sender)
{ float a[]={0.5,0.5}, b[]={300.,100.}, c[]={1.,3.};
  float alpha[]={1.,0.5}, beta[]={0.2,0.5}, gamma=0.01;
  float p[]={30.,10.}, omega=2.5;
  float Yd[]={10.,10.}, Ls=100.;
  float Ld[2],Ys[2],Ed,U;
  int tt=40;
  AnsiString NewString = "  t           U           Ed       omega       p[1]"
    "         p[2]        d[1]       Ld[2]       Ys[1]     Ys[2]";
  ListBox1->Items->Append(NewString);
  for(int t=0; t<=tt; t++)
  {     //--- Расчет Ld[i] и Ys[i] – формулы (A),(A*), предприятия ---
        //------- Расчет Yd[i]  –  формула (B), потребитель ----------
    for(int i=0; i<2; i++)
     { …….
       …….
     }
        //---------- Расчет Ed и U – формулы (E) и (7) ---------------
     ……
        //----------- Вывод результатов итерации ---------------------
    char out[100];
    sprintf(out,
"%3i%12.2f%12.2f%12.2f%12.2f%12.2f%12.2f%12.2f%12.2f%12.2f",
                     t,U,Ed,omega,p[0],p[1],Ld[0],Ld[1],Ys[0],Ys[1]);
    ListBox1->Items->Append(out);
```

```
      //------- Переопределение цен – формулы (C) и (D) ----------------
   for(int i=0; i<2; i++)
     ……
}
  ListBox1->Items->Append(NewString);
}
```

4.6. MathCAD

Модель общего равновесия

```
M :=  α ← (1, 0.5)ᵀ
      β ← (0.2, 0.5)ᵀ
      γ ← 0.01
      Ls ← 100
      a ← (0.5, 0.5)ᵀ
      b ← (300, 100)ᵀ
      c ← (1, 3)ᵀ
      w ← 2.5
      p ← (27, 10)ᵀ
      Yd ← (10, 10)ᵀ
      tt ← 40
      for  t ∈ 0 .. tt
          for  i ∈ 0 .. 1
              ▪
              ▪
          for  i ∈ 0 .. 1
              ▪
              ▪
          ▪
          ▪
          M⟨t⟩ ← (t, U, Ed, w, p_0, p_1, Ld_0, Ld_1, Yd_0, Yd_1, Ys_0, Ys_1)ᵀ
          for  i ∈ 0 .. 1
              ▪
              ▪
          ▪
          ▪
      M
```

Поведение производителя

Поведение потребителя

Избыточный спрос и функция полезности

Сохранение данных в текущей итерации

Переопределение цен

Результаты расчетов в среде MathCAD приведены на рисунках.

$M^T =$

	t	U	Ed	w	p0	p1	Ld0	Ld1	Yd0	Yd1	Ys0	Ys1
	0	1	2	3	4	5	6	7	8	9	10	11
0	0	345.245	-26.7	2.5	27	10	29.16	36	10.6	10	5.4	18
1	1	372.08	14.937	2.152	32.2	6	55.992	17.497	9.82	12	7.483	12.549
2	2	402.02	1.966	1.886	34.538	5.726	83.794	20.726	9.023	13.304	9.154	13.658
3	3	396.047	-0.482	1.932	34.406	5.549	79.312	18.565	8.791	14.288	8.906	12.926
4	4	402.561	1.095	1.91	34.292	6.23	80.547	23.924	8.758	14.673	8.975	14.674
5	5	397.712	4.238	1.955	34.075	6.229	75.936	22.838	8.794	14.966	8.714	14.337
6	6	401.248	4.481	1.943	34.155	6.544	77.257	25.521	8.786	15.035	8.79	15.156
7	7	398.365	5.017	1.971	34.151	6.483	75.078	24.351	8.785	15.119	8.665	14.804
8	8	400.368	4.124	1.965	34.271	6.641	76.047	25.697	8.761	15.106	8.72	15.208
9	9	398.652	3.78	1.982	34.311	6.59	74.89	24.861	8.747	15.121	8.654	14.958
10	10	399.757	2.857	1.98	34.404	6.671	75.487	25.543	8.726	15.092	8.688	15.162
11	11	398.769	2.414	1.99	34.442	6.636	74.87	25.015	8.714	15.087	8.653	15.005
12	12	399.368	1.747	1.989	34.503	6.677	75.222	25.356	8.699	15.062	8.673	15.107
13	13	398.818	1.41	1.995	34.528	6.655	74.898	25.044	8.691	15.054	8.654	15.013
14	14	399.139	0.991	1.994	34.565	6.676	75.099	25.214	8.682	15.038	8.666	15.064
15	15	398.84	0.777	1.997	34.58	6.663	74.932	25.036	8.677	15.031	8.656	15.011

Рис. 1. Результаты расчетов в первых шестнадцати итерациях

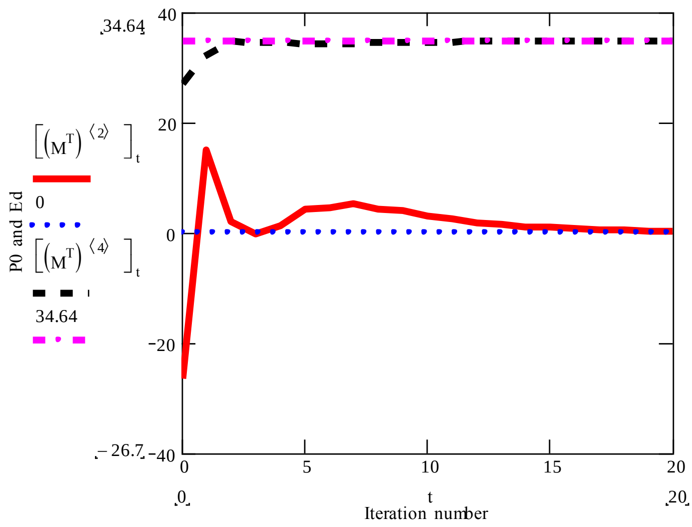

Рис. 2. Динамика стабилизации цены и избыточного спроса в зависимости от времени (номера итерации): сплошная кривая – избыточный спрос, пунктирная – цена. Для наглядности представлены уровень равновесной цены (34.64) и нулевой уровень

$$c0 := \begin{pmatrix} 0.75 \\ 0.8 \\ 0.85 \\ 0.9 \\ 0.95 \\ 1 \\ 1.05 \\ 1.1 \\ 1.15 \\ 1.2 \end{pmatrix} \quad p0 := \begin{pmatrix} 46.165 \\ 43.293 \\ 40.752 \\ 38.489 \\ 36.464 \\ 34.641 \\ 32.991 \\ 31.492 \\ 30.123 \\ 28.868 \end{pmatrix}$$

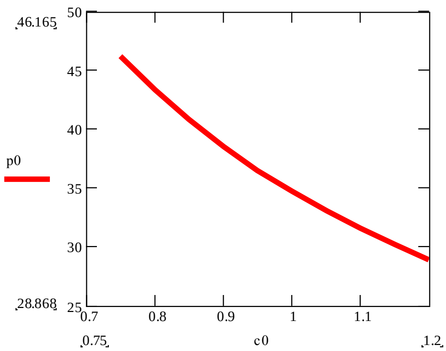

Рис. 3. Изменение равновесной цены p_0 на первый продукт при увеличении объема производства (изменение коэффициента c_0)

5. Задания

1. Введите в компьютер текст программы и дополните его недостающими операторами. Запустите программу на счет.

2. Проанализируйте полученные данные: как изменяются величины избыточного спроса, цен на продукты, функции полезности, объёмы выпусков в зависимости от номера итерации – времени функционирования рынка. Постройте соответствующие графики.

3. Варьируйте поочередно значениями $a_1, a_2; b_1, b_2; c_1, c_2$. Оцените примерные границы изменения этих параметров, при которых итерационная процедура сходится. При варьировании следующего параметра значение предыдущего должно быть возвращено к исходному среднему значению, т.к. итерационная процедура сходится в ограниченном диапазоне значений этих параметров.

4. Изменяйте объём выпуска продукта 1, меняя коэффициент c_1 через 5% от 0.75 до 1.25. Запишите равновесные цены на первый продукт и объёмы выпусков этого продукта для каждого значения c_1. Постройте графики зависимости равновесной цены от коэффициента c_1 и объема выпуска.

5. Верните значение $c_1 = 1$ и проделайте те же исследования, изменяя коэффициент b_1 в функции полезности. Постройте графики, подобные предыдущим.

6. Верните коэффициент b_1 в среднее положение, задайте начальное значение для цены на первый продукт $p_1 = 1$ (т.е. минимальное) и проанализируйте данные первых итераций. Почему цена p_1 начинает возрастать?

ДВУХСЕКТОРНАЯ МОДЕЛЬ

1. Основные положения модели

Двухсекторная модель является одной из разновидностей моделей общего равновесия [1]. Она отличается от рассмотренной раннее модели общего равновесия тем, что в ней введено два вида факторов – труд L и капитал K, а предприятия рассматриваются как отрасли, производящие обобщенные виды продукции Y_1^s *и* Y_2^s. Схема функционирования экономики в обсуждаемой модели представлена на рис.1.

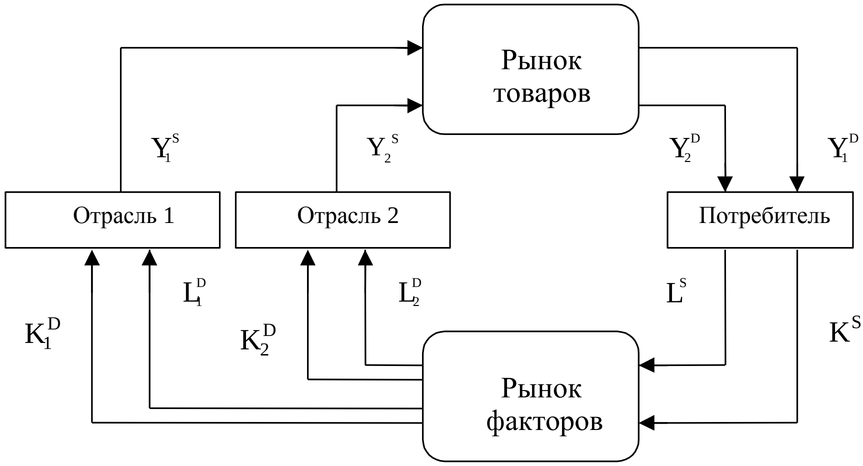

Рис. 1. Схема функционирования экономики в двухсекторной модели: L^S и K^S – объемы предложения факторов; $K_1^D, K_2^D, L_1^D, L_2^D$ – объемы спроса факторов; Y_1^S и Y_2^S – объемы предложения продукции первого и второго видов; Y_1^D и Y_2^D – объемы спроса продукции

Отрасли, приобретая на рынке факторов труд и капитал, производят два вида продукции и реализуют их на рынке товаров. Потребитель обуславливает спрос на эту продукцию, имея доход от продажи факторов.

Предположим, что производственные возможности отраслей определяются производственными функциями [2, 3, 11]

$$Y_i^S = c_i(K_i^D)^{a_i}(L_i^D)^{1-a_i}; \quad c_i > 0, \quad 0 < a_i < 1; \quad i = 1,2. \tag{1}$$

Для дальнейших рассуждений удобно произвести нормирование на L_i^D и перейти к выражению Y_i^S / L_i^D – удельного выпуска на единицу затраченного труда (производительности труда) в зависимости от капиталовооруженности труда K_i / L_i:

$$\frac{Y_i^D}{L_i^D} = \frac{c_i\,(K_i^D)^{a_i}\,(L_i^D)^{1-a_i}}{L_i^D}$$

$$= c_i \left(\frac{K_i^D}{L_i^D}\right)^{a_i} \left(\frac{L_i^D}{L_i^D}\right)^{1-a_i},$$

или

$$f_i(k_i) = c_i(k_i)^{a_i} \tag{2}$$

где $f_i = Y_i^S / L_i^D$, $k_i = K_i^D / L_i^D$ – производительность труда и капиталовооруженность труда, соответственно.

Предположим, что состояние обобщенного потребителя, так же как и в рассмотренной выше модели общего равновесия, описывается функцией полезности [3, 11]

$$U(Y_1^D, Y_2^D) = b_1 \ln(Y_1^D) + b_2 \ln(Y_2^D), \tag{3}$$

где Y_1^D и Y_2^D – объемы спроса продукции отраслей, а b_1 и b_2 – коэффициенты, определяющие уровни вкладов двух слагаемых в функцию полезности.

Обозначим цены на товары отраслей p_1 и p_2, цену единицы труда – ω, а цену аренды капитала – ρ. Будем считать, что цена аренды капитала постоянна, в то время как p_1, p_2 и ω в процессе установления рыночного равновесия меняются.

2. Математическая реализация динамики рынка

2.1. Схема алгоритма расчета

Математическая модель, имитирующая установление рыночного равновесия в двухсекторной модели, аналогична описанной ранее в модели общего равновесия. Схема модели приведена на рис. 2.

В момент времени $t = t_0$ задаются значения цен на товары и факторы. Определяется стратегия отраслей и уравнения, описывающие эту стратегию. Из уравнений находятся оптимальные для заданных цен объемы спроса на факторы и объемы выпуска продукции.

Аналогично задается стратегия обобщенного потребителя, на основе которой устанавливаются уравнения, позволяющие определить объемы спроса на продукцию отраслей.

Для неравновесных цен объемы предложений и объемы спроса не совпадают, возникает отличный от нуля избыточный спрос:

$$E_i = Y_i^D - Y_i^S, \qquad i = 1,2. \tag{4}$$

С учетом величин избыточных спросов осуществляется корректировка цен на товары.

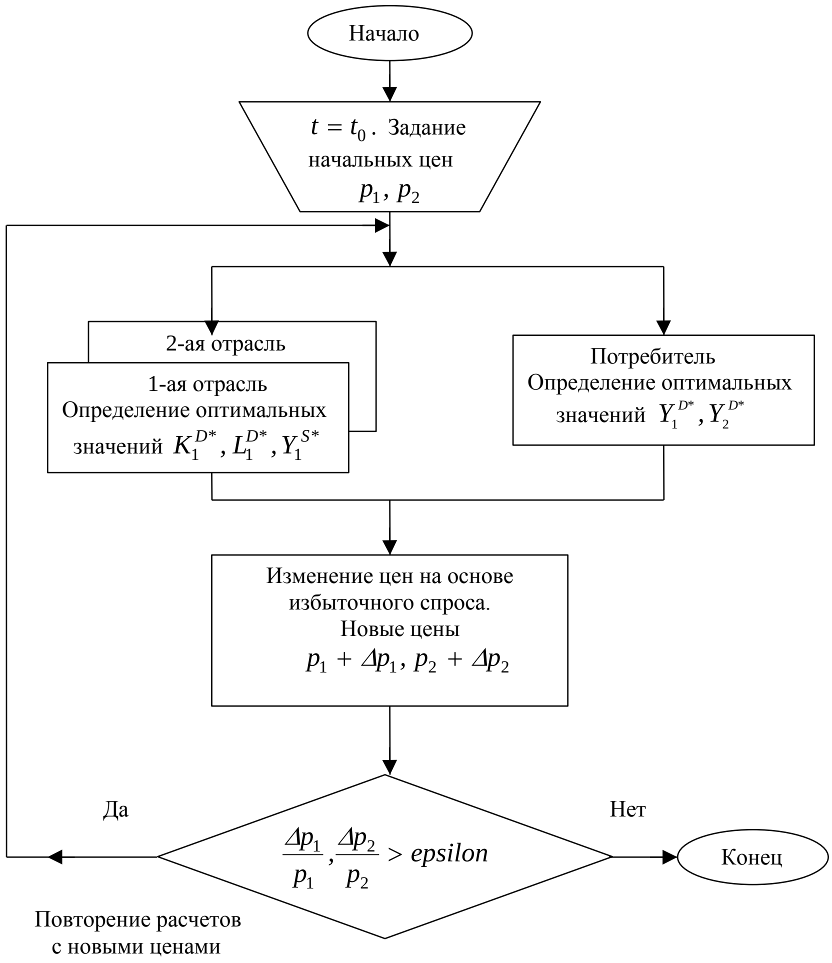

Рис. 2. Схема алгоритма расчетов

Производится дискретный сдвиг по времени Δt и следует расчет новых состояний производителей и потребителя в момент времени $t = t_0 + \Delta t$, новое вычисление избыточного спроса, корректировка цен и т.д.

Программная реализация данного процесса – итерационный (повторяющийся) расчет состояний производителей и потребителя с приближением к рыночному равновесию, организованный с помощью цикла. Роль дискретно изменяющегося времени в нем играет переменная цикла.

2.2. Отрасли

В дальнейшем изложении будем придерживаться следующих соглашений.

Для обозначения отраслей будем использовать индекс i, $i = 1$, если обсуждаются уравнения 1-й отрасли. Для второй отрасли $i = 2$.

При записи уравнений, чтобы не усложнять выражений, не будем указывать явно зависимость от времени, хотя необходимо иметь в виду, что большинство величин в уравнениях зависят от времени: цены, объемы предложения и спроса, прибыли и т.д.

Предположим, что *стратегия производителей (отраслей) заключается в получении максимальной прибыли*. Прибыль i-й отрасли Π_i равна разности между доходом от продажи продукции $p_i Y_i^S$ и расходами на факторы ρK_i^D, ωL_i^D:

$$\Pi_i = p_i Y_i^S - \rho K_i^D - \omega L_i^D . \qquad (5)$$

Произведем нормирование обеих частей равенства (5) на величину труда и будем далее оперировать с прибылью π_i, приходящейся на единицу затраченного труда:

$$\begin{aligned} \pi_i &= \frac{\Pi_i}{L_i^D} \\ &= p_i \frac{Y_i^S}{L_i^D} - \rho \frac{K_i^D}{L_i^D} - \omega \frac{L_i^D}{L_i^D} \\ &= p_i f_i - \rho k_i - \omega , \end{aligned} \qquad (6)$$

где f_i соответствует выражению (2).

Если отрасль осуществляет производство в ограниченных масштабах, то чистая прибыль в равновесном состоянии должна быть равна нулю, потому ограничение на деятельность отрасли имеет вид

$$p_i f_i = \rho k_i + \omega . \qquad (7)$$

Оптимизационная задача для отрасли формулируется следующим образом: необходимо найти максимум прибыли π_i, варьируя k_i, при ограничении (7)

$$\begin{cases} max \;\; \pi_i , \\ k_i ; \\ p_i \, f_i \; = \; \rho k_i \; + \; \omega \; . \end{cases} \tag{8}$$

Для решения этой задачи введем функцию Лагранжа $Lag_i^{отр}(k_i)$, состоящую из целевой функции (прибыли) и ограничивающего соотношения (7), умноженного на множитель Лагранжа y_i [3, 11]:

$$Lag_i^{отр}(k_i) = \pi_i + y_i(p_i f_i(k_i) - \rho k_i - \omega)$$

$$= p_i f_i(k_i) - \rho k_i - \omega + y_i(p_i f_i(k_i) - \rho k_i - \omega). \tag{9}$$

Продифференцируем $Lag_i^{отр}(k_i)$ по k_i и y_i и приравняем производные нулю, чтобы определить координаты экстремальной точки:

$$\begin{cases} \left.\dfrac{\partial Lag_i^{отр}(k_i)}{\partial k_i}\right|_{k_i = k_i^*} = p_i f_i' - \rho + y_i p_i f_i' - y_i \rho = 0 \\ \left.\dfrac{\partial Lag_i^{отр}(k_i)}{\partial y_i}\right|_{k_i = k_i^*} = p_i f_i - \rho k_i^* - \omega = 0 , \end{cases} \tag{10}$$

где $f_i' = \left.\dfrac{\partial f_i}{\partial k_i}\right|_{k_i = k_i^*}$.

Наша задача – из уравнений (10) при заданных ценах p_1 и p_2 найти выражение для оптимальной капиталовооруженности, как функцию параметров производственных функций и цен:

$$k_i^* = k_i^*(p_1, p_2, a_1, a_2, c_1, c_2) . \tag{11}$$

Если будет известно оптимальное значение k_i^*, то, используя (2) и (1), можно определить оптимальные величины Y_i^{S*} и L_i^{D*}. Для того чтобы найти k_i^*, проведем ряд преобразований уравнений (10).

Во-первых, из первого уравнения системы (10) следует:

$$p_i f_i' + y_i p_i f_i' = \rho + y_i \rho ,$$

$$p_i f_i'(1 + y_i) = \rho(1 + y_i) ,$$

$$f_i' = \frac{\rho}{p_i} . \tag{12}$$

Во-вторых, из второго уравнения системы имеем:

$$\omega = p_i f_i - \rho k_i^*,$$

$$\frac{\omega}{\rho} = \frac{p_i}{\rho} f_i - k_i^*,$$

и, используя (12), получаем

$$\frac{\omega}{\rho} = \frac{f_i}{f_i'} - k_i^*. \qquad (13)$$

Так как $f_i(k_i^*) = c_i(k_i^*)^{a_i}$, то $f_i' = c_i a_i(k_i^*)^{a_i - 1}$. Подставим эти выражения в (13)

$$\frac{\omega}{\rho} = \frac{c_i(k_i^*)^{a_i}}{c_i a_i(k_i^*)^{a_i-1}} - k_i^*$$

$$= \frac{k_i^*}{a_i} - k_i^*$$

$$= \frac{1 - a_i}{a_i} k_i^*.$$

Перенесем множитель, стоящий перед k_i^*, влево и получим промежуточное выражение для оптимального значения k_i^*:

$$k_i^* = \frac{a_i}{1 - a_i} \frac{\omega}{\rho}. \qquad (14)$$

Осталось выразить отношение ω / ρ через параметры производственных функций, чтобы получить (11) в явном виде. Для этого представим (12) следующим образом:

$$p_i = \frac{\rho}{f_i'}$$

$$= \frac{\rho}{c_i a_i(k_i^*)^{a_i-1}}.$$

Тогда отношение цен принимает вид

$$\frac{p_1}{p_2} = \frac{c_2 a_2(k_2^*)^{a_2-1}}{c_1 a_1(k_1^*)^{a_1-1}}. \qquad (15)$$

Подставим в (15) выражение для k_i^* из (14):

$$\frac{p_1}{p_2} = \frac{c_2 a_2 \left(\frac{a_2}{1-a_2}\right)^{a_2-1} \left(\frac{\omega}{\rho}\right)^{a_2-1}}{c_1 a_1 \left(\frac{a_1}{1-a_1}\right)^{a_1-1} \left(\frac{\omega}{\rho}\right)^{a_1-1}} . \quad (16)$$

Из (16) найдем значение ω / ρ :

$$\left(\frac{\omega}{\rho}\right)^{a_2-a_1} = \frac{p_1 c_1 a_1 a_1^{a_1-1} (1-a_2)^{a_2-1}}{p_2 c_2 a_2 a_2^{a_2-1} (1-a_1)^{a_1-1}}$$

$$= \frac{p_1 c_1 a_1^{a_1} (1-a_1)^{1-a_1}}{p_2 c_2 a_2^{a_2} (1-a_2)^{1-a_2}} ,$$

и, возводя левое и правое выражения в степень $1/(a_2 - a_1)$, окончательно имеем

$$\frac{\omega}{\rho} = \left[\frac{p_1 c_1 a_1^{a_1} (1-a_1)^{1-a_1}}{p_2 c_2 a_2^{a_2} (1-a_2)^{1-a_2}}\right]^{\frac{1}{a_2-a_1}} . \quad (17)$$

Оптимальное значение капиталовооруженности получим, подставив (17) в (14):

$$k_i^* = \frac{a_i}{1-a_i} \left[\frac{p_1 c_1 a_1^{a_1} (1-a_1)^{1-a_1}}{p_2 c_2 a_2^{a_2} (1-a_2)^{1-a_2}}\right]^{\frac{1}{a_2-a_1}} \quad (18)$$

Нахождение оптимальных значений объемов потребляемых факторов L_i^{D*}, K_i^{D*} и объемов выпускаемой продукции Y_i^{S*} каждой отраслью теперь не сложно. Действительно, полная занятость населения и полная загруженность производственных фондов (капитала) определяется системой уравнений:

$$\begin{cases} (L_1^{D*} + L_2^{D*} = L^S \\ K_1^{D*} + K_2^{D*} = K^S , \end{cases} \quad (19)$$

где предложение труда L^S и предложение капитала K^S – постоянные в данной модели величины.

Так как $K_i^{D*} / L_i^{D*} = k_i^*$, а $K^S / L^S = k$, то систему (19) можно записать в виде

$$\begin{cases} L_1^{D*} + L_2^{D*} = L^S \\ k_1^* L_1^{D*} + k_2^* L_2^{D*} = kL^S , \end{cases} \quad (20)$$

Решение системы (20) относительно неизвестных L_1^{D*} и L_2^{D*} дает оптимальные размеры спроса на труд (на данном этапе, т.е. в текущей итерации):

$$\boxed{L_1^{D*} = \frac{k_2^* - k}{k_2^* - k_1^*} L^S}, \tag{A}$$

$$\boxed{L_2^{D*} = \frac{k - k_1^*}{k_2^* - k_1^*} L^S}, \tag{B}$$

откуда оптимальные размеры спроса i-й отрасли на капитал

$$\boxed{K_i^{D*} = k_i^* L_i^{D*}}. \tag{C}$$

Оптимальные значения объемов выпусков при текущих ценах (в текущей итерации) получаются из уравнения (1)

$$\boxed{Y_i^{S*} = c_i (K_i^{D*})^{a_i} (L_i^{D*})^{1-a_i}}. \tag{D}$$

2.3. Потребитель

Предположим, что *стратегия потребителя*, устанавливающего спрос Y_1^D, Y_2^D на продукты производства, *заключается в стремлении к максимальному благополучию*, находясь, конечно, в рамках своих бюджетных возможностей.

С математической точки зрения это означает, что необходимо максимизировать функцию полезности (3), учитывая бюджетное ограничение: расход $p_1 Y_1^D + p_2 Y_2^D$ не должен превышать дохода $\rho K^S + \omega L^S$ или в условиях установившегося равновесия

$$p_1 Y_1^D + p_2 Y_2^D = \rho K^S + \omega L^S .$$

Таким образом, можно сформулировать оптимизационную задачу для потребителя:

$$\begin{cases} max\ U(Y_1^D, Y_2^D) \\ Y_1^D, Y_2^D ; \\ p_1 Y_1^D + p_2 Y_2^D = \rho K^S + \omega L^S \ . \end{cases} \tag{21}$$

Как и ранее, найдем решение этой задачи с использованием функции Лагранжа:

$$Lag^{потр}(Y_1^D, Y_2^D) = U(Y_1^D, Y_2^D) + y(\rho K^S + \omega L^S - p_1 Y_1^D - p_2 Y_2^D), \tag{22}$$

где y – множитель Лагранжа. Продифференцируем функцию Лагранжа по варьируемым переменным Y_1^D, Y_2^D и множителю Лагранжа, чтобы отыскать оптимальные для данной итерации значения $(Y_1^D)^*$ и $(Y_2^D)^*$. С учетом выражения (3) для функции полезности получаем:

$$\begin{cases} \dfrac{\partial Lag^{потр}}{\partial Y_1^D} = \dfrac{b}{Y_1^D} - yp_1 = 0, \\ \dfrac{\partial Lag^{потр}}{\partial Y_2^D} = \dfrac{b}{Y_2^D} - yp_2 = 0, \\ \dfrac{\partial Lag^{потр}}{\partial y} = \rho K^S + \omega L^S - p_1 Y_1^D - p_2 Y_2^D = 0 . \end{cases} \tag{23}$$

Преобразуем систему (23) к удобному для решения виду:

$$\begin{cases} p_1 Y_1^D + 0 \cdot Y_2^D - b_1 \dfrac{1}{y} = 0, \\ 0 \cdot Y_1^D + p_2 Y_2^D - b_2 \dfrac{1}{y} = 0, \\ p_1 Y_1^D + p_2 Y_2^D + 0 \cdot \dfrac{1}{y} = \rho K^S + \omega L^S . \end{cases}$$

Найдем решение для Y_1^D и Y_2^D по правилу Крамера:

$$Y_1^{D*} = \frac{\begin{vmatrix} 0 & 0 & -b_1 \\ 0 & p_2 & -b_2 \\ \rho K^S + \omega L^S & p_2 & 0 \end{vmatrix}}{\begin{vmatrix} p_1 & 0 & -b_1 \\ 0 & p_2 & -b_2 \\ p_1 & p_2 & 0 \end{vmatrix}}$$

$$= \frac{b_1 p_2 (\rho K^S + \omega L^S)}{b_1 p_1 p_2 + b_2 p_1 p_2}$$

$$= \frac{b_1}{b_1 + b_2} \frac{\rho K^S + \omega L^S}{p_1} . \tag{24}$$

$$Y_2^{D*} = \frac{\begin{vmatrix} p_1 & 0 & -b_1 \\ 0 & 0 & -b_2 \\ p_1 & \rho K^S + \omega L^S & 0 \end{vmatrix}}{b_1 p_1 p_2 + b_2 p_1 p_2}$$

$$= \frac{b_2 p_1 (\rho K^S + \omega L^S)}{b_1 p_1 p_2 + b_2 p_1 p_2}$$

$$= \frac{b_2}{b_1 + b_2} \frac{\rho K^S + \omega L^S}{p_2}. \qquad (25)$$

Аналогично можно найти и множитель Лагранжа, но в рамках нашей задачи нет необходимости находить значение и обсуждать смысл множителя Лагранжа. Он играет здесь вспомогательную роль.

В выражениях (24) и (25) цены входят в виде отношений, поэтому для численных расчетов удобно произвести масштабирование цен. В частности, можно за единицу измерения цен ввести цену аренды капитала, т.е. положить $\rho = 1$. Тогда выражения (24) и (25) преобразуются к виду

$$\boxed{Y_1^{D*} = \frac{b_1}{b_1 + b_2} \frac{K^S + \omega L^S}{p_1}}, \qquad (E)$$

$$\boxed{Y_2^{D*} = \frac{b_2}{b_1 + b_2} \frac{K^S + \omega L^S}{p_2}}. \qquad (F)$$

2.4. Преобразование цен

Как и в рассмотренной ранее модели общего равновесия полагаем, что

$$\frac{\partial p_i(t)}{\partial t} = \alpha_i E_i(t),$$

где α_i – коэффициенты пропорциональности (подбираются при численных расчетах), а E_i – избыточный спрос (4). Переходя к дискретным единичным интервалам, получаем

$$\frac{p_i(t+1) - p_i(t)}{(t+1) - t} = \alpha_i E_i(t),$$

или

$$p_i(t+1) = p_i(t) + \alpha_i E_i(t)$$

$$= p_i(t) + \alpha_i (Y_i^{D*}(t) - Y_i^{S*}(t)).$$

Как и ранее, предотвратим возможность появления отрицательных цен:

$$\boxed{p_i(t+1) = max\left\{ p_i(t) + \alpha_i(Y_i^{D*}(t) - Y_i^{S*}(t)),\ 0 \right\}} \qquad (G)$$

Для расчета $\omega(t+1)$ используется выражение (17). Если цену услуг капитала использовать для масштабирования цен, то нормированное на ρ =1 выражение для $\omega(t+1)$ имеет вид

$$\boxed{\omega(t+1) = \left[\frac{p_1(t+1)c_1 a_1^{a_1}(1-a_1)^{1-a_1}}{p_2(t+1)c_2 a_2^{a_2}(1-a_2)^{1-a_1}} \right]^{\frac{1}{a_2 - a_1}}}. \qquad (H)$$

В итерационном процессе она пересчитывается при определении оптимальных оценок для отраслей.

3. Контрольные вопросы

1. Сформулируйте основные положения двухсекторной модели.
2. Начертите схему алгоритма итерационного приближения к равновесному состоянию.
3. Что является аналогом времени в итерационном процессе?
4. Что такое избыточный спрос? Для чего он используется в данной модели?
5. Объясните, почему при ограниченном масштабе производства оптимальные прибыли отраслей равны нулю.
6. Какова структура функций Лагранжа в оптимизационных задачах данной модели?

4. Программы

4.1. Алгоритм вычислений

В цикле по переменной цикла t, имитирующей изменение времени, по формулам (A)-(D) рассчитываются оптимальные значения L_i^{D*}, K_i^{D*} и Y_i^{S*}, i = 1,2; по формулам (E), (F) производятся вычисления Y_i^{D*}, i = 1,2; и по формулам (G), (H) определяются новые цены. Окончание итерационной процедуры задается верхним значением для t и может быть изменено пользователем.

Признаком установления экономического равновесия является равенство нулю избыточных спросов, определяемых по формуле (4). Производится распечатка для каждой итерации значений цен на товары и факторы, величин избыточных спросов и оптимальных значений капиталовооруженностей отраслей. Строится график динамики избыточного спроса на первый товар в зависимости от цены.

В предлагаемых программах выпущены строки кода, соответствующие уравнениям (A) – (F). Пользователю предлагается дописать их самостоятельно в приемлемом для него варианте работы на QBASIC[6], C++[7], Borland C++ Builder[8] или MathCAD[9].

4.2. Список переменных

A(I) = a_i, C(I) = c_i – коэффициенты производственных функций;
B(I) = b_i – коэффициенты функции полезности;
P(I) = P_i – цена i-го продукта;
W = ω – цена ресурса;
LS = L^S – объем предложения труда;
KS = K^S – объем предложения капитала;
LD(I) = L_i^d – размер спроса i-й отрасли на труд;
KD(I) = L_i^d – спрос на капитал со стороны i-й отрасли;
YS(I) = Y_i^S – объем предложения i-го продукта;
YD(I) = Y_i^d – размер спроса на i-й продукт;
ED(I) = E_i – избыточный спрос на i-й продукт;
KL(I) = k_i – отношение капитал/труд в i-й отрасли;
AL(I) = α_i – коэффициенты в выражении (G);
U – функция полезности;
TT – верхняя граница числа итераций.

4.3. QBASIC

```
100 '2-ух СЕКТОРНАЯ МОДЕЛЬ [1]
110 DIM YD(2), YS(2), ED(2), KD(2), LD(2)
120 DIM KL(2), P(2), A(2), B(2), C(2), AL(2)
130 READ TT, LS, KS
140 KL = KS / LS
150 FOR I = 1 TO 2
160  READ A(I), B(I), C(I), AL(I), P(I)
170 NEXT I
180 IF GR = 0 THEN GOSUB 1000 ELSE GOSUB 1100
190 FOR T = 0 TO TT
200 '-----ОТРАСЛЬ---,формулы (A)-(D),(H)
```

```
210
220
230
240
250
260
270
280
290
300
310
320 '----ПОТРЕБИТЕЛЬ----,формулы (E),(F)
330
340
350
360
370  IF GR = 0 THEN GOSUB 1500 ELSE GOSUB 1600
380 '----ПРЕОБРАЗОВАНИЕ ЦЕН----,формула (G)
390
400
410
420
430
440 IF GR = 1 THEN 440
450 LOCATE 25, 60: INPUT "НАЖМИТЕ ENTER"; Q$
460 GR = 1: RESTORE: GOTO 130
1000 '---- УСТАНОВКА ЭКРАНА ----
1010 '(  GR=0)
1020 SCREEN 2: CLS: KEY OFF
1030 PRINT "       АЛЬФА(1)="; AL(1); "АЛЬФА(2)="; AL(2): PRINT
1040 PRINT "  T   U      W/R  P(1)/R  P(2)/R  ";
1050 PRINT "KL(1)  KL(2) ED(1)  ED(2)"
1060 RETURN
1100 '(  GR=1)
1110 SCREEN 2: CLS
1120 X0 = 32: Y0 = 200
1130 LINE (X0, 8)-(X0, 192)
1140 LINE (0, Y0 / 2)-(600, Y0 / 2)
1150 LOCATE 1, 6: PRINT "+ED(1)"
1160 LOCATE 24, 6: PRINT "-ED(1)";
1170 LOCATE Y0 / 16, 75: PRINT "P(1)";
1180 LOCATE 1, 30: PRINT "ИЗМЕНЕНИЕ ЦЕН И ИЗБЫТОЧНОГО
СПРОСА"
```

```
1190 XX = 1.1: X1 =.8: YY = 25
1200 SX = (XX – X1) / (600 – X0): SY = YY / Y0
1210 LOCATE 1, 1: PRINT YY
1220 LOCATE 23, 1: PRINT -YY;
1230 LOCATE Y0 / 16 + 1, 6: PRINT X1;
1240 LOCATE Y0 / 16 + 1, 70: PRINT XX;
1250 RETURN
1500 '--- ВЫВОД РЕЗУЛЬТАТА ---
1510 '(  GR=0)
1520 U = B(1) * LOG(YS(1)) + B(2) * LOG(YS(2))
1530 PRINT USING "###"; T;
1540 PRINT USING "####.##"; U; W; P(1); P(2);
1550 PRINT USING "####.##"; KL(1); KL(2); ED(1); ED(2)
1560 RETURN
1600 '(  GR=1)
1610 IF T = 0 THEN 1640
1620 LINE ((X0 – X1) / SX + X0, 200 / 2 – Y0 / SY / 2)-((P(1) – X1) / SX +
X0, 200 / 2 -
ED(1) / SY / 2)
1630 GOTO 1650
1640 PSET ((P(1) – X1) / SX + X0, 200 / 2 – ED(1) / SY / 2)
1650 X0 = P(1): Y0 = ED(1)
1660 RETURN
2000 '***ДАННЫЕ***
2010 DATA 50,100,100
2020 DATA 0.4,8,2,0.0007,1.0
2030 DATA 0.6,12,3,0.0003,0.7
```

4.4. C++

```
//Двухсекторная модель
/*Программная оболочка Borland C++ 5.x
При открытии нового проекта в окне New Target в Target Type необходимо
выбрать Application, в Platform выбрать DOS(Standard), в Target Model –
Large  и включить кнопку  BGI.
Сведения о функциях BGI и файлы примеров расположены:
Help => Keyword search => Book Shelf => Borland C++ DOS Reference =>
Borland Grafic Interface => About.... */

#include <iostream.h>
#include <graphics.h>
#include <math.h>
#include <conio.h>
```

```
#include <stdio.h>

void func_k_opt(float *, float &);
void func_LdKdYs(float *,float *,float *,float *,float &);
void func_Yd(float *,float &);
void func_p_E(float *,float *,float *);
void plotGraphic(float,float,float *);

const int n=2;                          //Число отраслей
const int t_end=100;                    //Верхняя граница числа итераций

float a[]={0.4,0.6},c[]={2.,3.};        //Коэффициенты в производственных
//функциях
float b[]={8.,12.};                     //Коэффициенты в функции полезно-
сти
float alpha[]={7.e-4,3.e-4};            //Коэффициенты для корректиров-
ки цен
float p[]={1.,0.7};                     //Цены на продукты производства
float Ks=100.,Ls=100.,                  //Объемы предложения факторов

main()
{
//----------------------------------------------------------------------------------------
  float w;                              //Цена единицы труда
  float Ys[n],Yd[n];                    //Объемы предложения и спроса
  float Kd[n],Ld[n];                    //Объемы спроса факторов
  float k_opt[2];                       //Оптимальные капиталовооруженности
  float E[n],U;                         //Избыточные спросы и функция полез-
ности
  float fy_graph[t_end];                //Массив для вывода графика
//----------------------------------------------------------------------------------------
  float k=Ks/Ls;
  cout.precision(3);
  cout.setf(ios::right);
  cout<<"   t   U      w     p[0]   p[1]";
  cout<<"  k_opt1  k_opt2    E[0]     E[1]\n";
 //-- Итерационное приближение к равновесию ---------------------------------
  for(int t=0;t<=t_end;t++)
  { func_k_opt(k_opt,w);
    func_LdKdYs(Ld,Kd,Ys,k_opt,k);
    func_Yd(Yd,w);
    func_p_E(Yd,Ys,E);
    U=b[0]*log(Ys[0])+b[1]*log(Ys[1]);
//-- Вывод результатов ------------------
```

```
   cout.width(4);
   cout<<t;
   cout.width(6);
   cout<<U;
   cout.width(7);
   cout<<w;
   cout.width(7);
   cout<<p[0];
   cout.width(7);
   cout<<p[1];
   cout.width(7);
   cout<<k_opt[0];
   cout.width(8);
   cout<<k_opt[1];
   cout.width(10);
   cout<<E[0];
   cout.width(10);
   cout<<E[1]<<"\n";
//-- Заполнение массива для графика -----
   fy_graph[t]=E[0];
}
  cout<<"   t   U      w     p[0]   p[1]";
  cout<<"  k_opt1  k_opt2    E[0]     E[1]\n";
  getch();

//-- Открытие графического режима --------
  int dr = DETECT, mod;
  initgraph(&dr, &mod, "D:\\bc5\\bgi");

//-- Графики -------------------------------------
  settextstyle(SMALL_FONT,HORIZ_DIR,6);
  outtextxy(150,50,"  E = E(t)");
  setcolor(WHITE);
  plotGraphic(9,(float)t_end,fy_graph);
  getch();

//-- Закрытие графического режима --------
 closegraph();
return 0;
}

//---------------------------------------------------------------------------
//Функция для вычисления  w  и  k_opt: уравнения (H) и (18)
```

```
void func_k_opt(float * k_opt,float & w)
{
   …….
   …….
  return;
}
//--------------------------------------------------------------------------------
//Функция для вычисления Ld[i],  Kd[i]  и  Ys[i]: уравнения (A) – (D)
void func_LdKdYs(float *Ld,float *Kd,float *Ys,float *k_opt,float &k)
{
   …….
   …….
  return;
}
//------------------------------------------------------------------------
//Функция для вычисления  Yd[i]:  формулы (E), (F)
void func_Yd(float *Yd,float &w)
{
..…..
  return;
}
//------------------------------------------------------------------------
//Функция для вычисления E[i] и p[i]: формулы (4) и (G)
void func_p_E(float *Yd,float *Ys,float *E)
{
  …….
  …….
  return;
}
//---------------------------------------------------------------------
//Построение графика функции[10]
void plotGraphic(float a,float b,float *fy)
{ float xStep=pow(10, floor(log(b-a)/log(10.0)));
  float xMin=xStep*floor(a/xStep);
  float xMax=xStep*ceil(b/xStep);
  float *fVal=new float[t_end];
  for(int i=0; i<t_end; i++)
    fVal[i]=fy[i];
  float  yMin=fVal[0],  yMax=fVal[0];
  for(int i=1; i<t_end; i++)
    if(fVal[i]<yMin)       yMin=fVal[i];
    else
    if(fVal[i]>yMax)       yMax=fVal[i];
```

```
float  yStep=pow(10, floor(log(yMax-yMin)/log(10.0)));
   yMin=yStep*floor(yMin/yStep);
   yMax=yStep*ceil(yMax/yStep);

int  x0=60;
int  x1=getmaxx()-20;
int  y0=10;
int  y1=getmaxy()-40;

line(x0,y0,x1,y0);
line(x1,y0,x1,y1);
line(x1,y1,x0,y1);
line(x0,y1,x0,y0);

float  kx=(x1-x0)/(xMax-xMin);
float  ky=(y1-x0)/(yMax-yMin);
float  x=a;
float  h=(b-a)/t_end;

setcolor(YELLOW);
x=a;
moveto(x0+(x-xMin)*kx,y0+(yMax-fVal[0])*ky);
for(int i=1; i<t_end; i++,x+=h)
  lineto(x0+(x-xMin)*kx,y0+(yMax-fVal[i])*ky);
setcolor(LIGHTRED);
line(x0,y0+yMax*ky,x1,y0+yMax*ky);
setcolor(WHITE);

char   str[128];
settextstyle(SMALL_FONT,HORIZ_DIR,0);
for(x=xMin;x<=xMax;x+=xStep)
{  int  ix=x0+(x-xMin)*kx;
   line(ix,y1,ix,y1+10);
   if(x+xStep<=xMax)
     for(int i=1; i<10;i++)
        line(ix+i*xStep*kx*0.1,y1,ix+i*xStep*kx*0.1,y1+5);
   sprintf(str,"%g",x);
   outtextxy(ix-textwidth(str)/2,y1+15,str);
}
for(float y=yMin;y<=yMax;y+=yStep)
{  int  iy=y0+(yMax-y)*ky;
   line(x0-10,iy,x0,iy);
```

```
    if(y+yStep<=yMax)
      for(int i=1; i<10;i++)
        line(x0-5,iy-i*yStep*ky*0.1,x0,iy-i*yStep*ky*0.1);
    sprintf(str,"%g",y);
    outtextxy(x0-10-textwidth(str),iy,str);
  }
  delete fVal;
}
//------------------------------------------------------------
```

4.5. Borland C++Builder

На форму вызваны три компонента VCL – кнопка Button для запуска программы, ListBox для вывода данных и Chartfx для построения графика.

```
//-----------------------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include <stdio.h>
#include "double_sector_BCB.h"
//-----------------------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----------------------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----------------------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{ const int n=2;                          //Число отраслей
  int t_end=100;                          //Верхняя граница числа итераций
  float a[]={0.4,0.6},c[]={2.,3.};        //Коэффициенты в производственных
                                          //функциях
  float b[]={8.,12.};                //Коэффициенты в функции полезности
  float alpha[]={7.e-4,3.e-4};         //Коэффициенты для корректировки цен
  float p[]={1.,0.7};                //Цены на продукты производства
  float w;                                //Цена единицы труда
  float Ys[n],Yd[n];                      //Объемы предложения и спроса
  float Kd[n],Ld[n];                      //Объемы спроса факторов
```

```
  float k_opt[n];                    //Оптимальные капиталовооруженности
  float E[n],U;                //Избыточные спросы и функция полезности
  float Ks=100.,Ls=100.;              //Объемы предложения факторов
//---------------------------------------------------------------------------------------
  float k=Ks/Ls;
  char out[100];                      //Вспомогательный массив
 //-- Открытие данных для графика -------------------------
  Chartfx1->OpenDataEx(COD_VALUES,1,101);
  //Вывод строки сообщений
  AnsiString NewString = "  t          U          w        p[0]"
    "       p[1]       k[0]      k[1]      E[0]       E[1]";
  ListBox1->Items->Append(NewString);
//-- Основная итерационная процедура -----------------------------------------
  for(int t=0; t<=t_end; t++)
  {
//-- Расчет  w  и  k_opt[i]  –  формулы (H) и (18) ----------
       …….
       …….
//-- Расчет  Ld[i],Kd[i] и Ys[i] – формулы (A) – (D) -------
       ……
       ……
 //-- Расчет  Yd[i] и E[i] – формулы  (E),(F),(4) ------------
       ……
……
//-- Вычисление функции полезности – формула (3) -----
       ……
//-- Вывод результатов итерации ----------------------------
   sprintf(out,
"%3i%12.2f%12.2f%12.2f%12.2f%12.2f%12.2f%12.2f%12.2f",
                  t,U,w,p[0],p[1],k_opt[0],k_opt[1],E[0],E[1]);
   ListBox1->Items->Append(out);
//-- Расчет  p[i] – формула (G) -------------------------------
       ……
……
 //-- Вывод графика -------------------------------------------
   Chartfx1->ThisSerie=0;
   Chartfx1->Value[t]=E[0];
  }
//-- Закончилась итерационная процедура -------------------------------------
   ListBox1->Items->Append(NewString);
 //-- Закрытие данных графика -------------------------------
   Chartfx1->CloseData(COD_VALUES);
}
```

```
for i ∈ 0 .. 1                      Вычисление k_opt
                                    Поведение
                                    производителя
for i ∈ 0 .. 1
                                    Поведение потребителя
                                    и избыточный спрос
                                    Функция полезности
M⟨t⟩ ← (t, w, p_0, p_1, k_opt_0, k_opt_1, E_0, E_1, ...)
                                    Сохранение данных в текущей
                                    итерации
for i ∈ 0 .. 1
                                    Переопределение цен
M
```

//---

4.6. MathCAD

Вычисления в среде MathCAD [9] удобны для изучения и исследования модели. Дружественный пользователю интерфейс, неплохие графические средства, возможность программирования в обозначениях, понятных большинству пользователей, позволяют успешно использовать MathCAD в этой работе.

Ниже приведен текст программы с частично пропущенным кодом, который пользователю предлагается дописать самостоятельно. С помощью таблицы M осуществлен вывод результатов расчетов. Приведены графики динамики избыточного спроса в зависимости от номера итерации, а также изменение цены на продукт в зависимости от объема производства.

Результаты расчетов в среде MathCAD приведены на рисунках.

$M^T =$

	t	w	p0	p1	k_opt0	k_opt1	E0	E1	U
	0	1	2	3	4	5	6	7	8
0	0	0.784	1	0.7	0.522	1.175	29.932	-88.92	95.645
1	1	1.055	1.021	0.673	0.704	1.583	-34.691	50.003	96.674
2	2	0.838	0.997	0.688	0.559	1.257	15.41	-57.182	97.115
3	3	1.003	1.007	0.671	0.669	1.505	-23.339	27.33	97.338
4	4	0.87	0.991	0.679	0.58	1.305	7.745	-38.614	97.53
5	5	0.975	0.997	0.668	0.65	1.462	-16.471	14.934	97.579
6	6	0.889	0.985	0.672	0.593	1.334	3.605	-27.301	97.669
7	7	0.958	0.988	0.664	0.638	1.437	-12.125	7.976	97.676
8	8	0.901	0.979	0.666	0.601	1.351	1.336	-20.082	97.721
9	9	0.947	0.98	0.66	0.631	1.421	-9.241	3.992	97.717
10	10	0.908	0.974	0.662	0.605	1.362	0.092	-15.262	97.742
11	11	0.941	0.974	0.657	0.627	1.411	-7.24	1.687	97.737
12	12	0.913	0.969	0.658	0.609	1.369	-0.574	-11.906	97.75
13	13	0.936	0.968	0.654	0.624	1.404	-5.795	0.356	97.746
14	14	0.916	0.964	0.654	0.611	1.374	-0.91	-9.481	97.755
15	15	0.933	0.963	0.651	0.622	1.399	-4.714	-0.397	97.751

Рис. 1. Результаты расчетов в первых шестнадцати итерациях

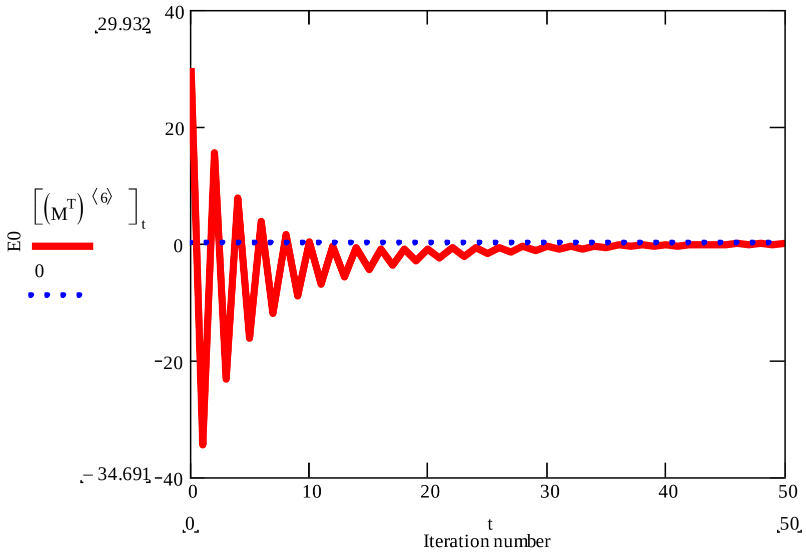

Рис. 2. Динамика стабилизации избыточного спроса в зависимости от времени (номера итерации)

$$c0 := \begin{pmatrix} 1.8 \\ 1.85 \\ 1.9 \\ 1.95 \\ 2 \\ 2.05 \\ 2.1 \\ 2.15 \\ 2.2 \\ 2.25 \end{pmatrix} \quad p0 := \begin{pmatrix} 1.04 \\ 1.012 \\ 0.985 \\ 0.96 \\ 0.936 \\ 0.913 \\ 0.892 \\ 0.877 \\ 0.907 \\ 0.948 \end{pmatrix}$$

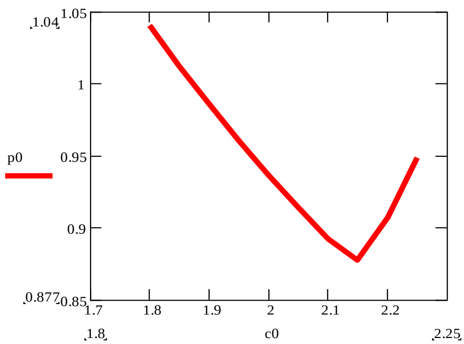

Рис. 3. Изменение равновесной цены на первый продукт p_0 при увеличении объема производства (изменение коэффициента c_0): подъем кривой означает, что равновесие не устанавливается при слишком большом значении c_0

5. Задания

1. Дополните недостающий текст программы, используя формулы (A) – (H).

2. Проанализируйте полученные данные: как изменяются величины избыточных спросов, цен на продукты, функции полезности, объёмы выпусков в зависимости от номера итерации – времени функционирования рынка. Постройте соответствующие графики.

3. Изменяя последовательно параметры $c_1, c_2, a_1, a_2, b_1, b_2$, найдите для каждого параметра границы, в которых итерационная процедура сходится.

Можно ли на основе проведенных испытаний сделать вывод о том, какая из моделей – общего равновесия или двухсекторная, ведет себя более устойчиво при изменении входных данных?

4. Установите исходные (заданные в программе) значения параметров. Выведите на экран значения прибылей отраслей. Постройте графики изменения прибылей в зависимости от номера итерации.

НЕОКЛАССИЧЕСКАЯ МОДЕЛЬ РОСТА

1. Формулировка модели

В любой экономике валовой выпуск делится на две основные составляющие. Первая идет на обеспечение текущего спроса (потребление), а вторая на обеспечение будущего спроса (капитальные вложения). Разделение на две части предполагает наличие проблемы выбора: в какой пропорции осуществлять деление. Увеличение объемов текущего спроса более благоприятно в настоящем, но это приводит к уменьшению капитальных вложений, а следовательно, к уменьшению потребления в будущем. Желательно выработать стратегию разделения валового выпуска такую, которая приводила бы к оптимальному потреблению в настоящем и обеспечивала его рост в будущем. Возникают задачи об экономическом росте и оптимальном экономическом росте [1, 2, 11].

В неоклассической модели роста на примере агрегированной, замкнутой экономики исследуется динамика выпуска во времени при наличии проблемы разделения выпуска на две составляющие: потребление и капитальные вложения. Агрегированность означает, что экономика выпускает один однородный продукт. Замкнутость – что ни выпуск, ни затраты не импортируются и не экспортируются.

Пусть в момент времени t экономика использует объем капитала $K(t)$ и объем труда $L(t)$. Пусть валовой выпуск за единицу времени равен $Y(t)$. Обозначим уровень текущего потребления за единицу времени $C(t)$, а объем инвестиций – $I(t)$. При равенстве доходов и расходов имеем:

$$Y(t)=C(t)+I(t). \tag{A}$$

Введем в рассмотрение малый временной интервал dt. Величина инвестиций за dt составляет $I(t)\cdot dt$, но часть их идет на замещение изношенного капитала (амортизация). Будем считать, что амортизация наличного капитала пропорциональна его величине с коэффициентом μ $(0 < \mu < 1)$, который носит название – норма амортизации. Тогда, при амортизационных отчислениях за промежуток dt равных $\mu\cdot K(t)\cdot dt$ величина изменения капитала $dK(t)$ составит

$$dK(t)=I(t)\cdot dt-\mu\cdot K(t)\cdot dt.$$

Разделив на dt обе части равенства, получаем величину чистых капитальных вложений, которые измеряются скоростью изменения наличного капитала $\dot{K}(t)=dK(t)/dt$:

$$\frac{dK(t)}{dt}=I(t)-\mu\cdot K(t).$$

Из последнего соотношения получаем тождество для валовых инвестиций:

$$I(t)=\dot{K}(t)+\mu K(t). \tag{B}$$

Уравнение (А) в этом случае приводится к виду

$$Y(t) = \dot{K}(t) + \mu \cdot K(t) + C(t). \qquad (1)$$

В дальнейших выкладках точка над переменной будет обозначать взятие производной от этой переменной по времени.

Можно продвинуться дальше в детализации полученного уравнения (1), если в него наряду с капиталом ввести второй фактор – величину используемого труда как функцию времени *L(t).* Ее можно интерпретировать как изменяющееся во времени количество рабочих, занятых в экономике. Для этого введем в рассмотрение производственную функцию

$$F(K(t), L(t)) = Y(t),$$

которая характеризует возможности производства в зависимости от капитала и труда. Будем считать, что производственная функция является однородной функцией аргументов K и L, т.е. для любого положительного числа α изменение аргументов K и L в α раз приводит к изменению функции тоже в α раз:

$$F(\alpha K, \alpha L) = \alpha F(K, L) = \alpha Y.$$

Таким свойством обладает, например, известная производственная функция Кобба-Дугласа [2, 11].

В частности, полагая $\alpha = 1/L$, получим

$$\frac{Y}{L} = F\left(\frac{K}{L}, 1\right) = f\left(\frac{K}{L}\right).$$

Перейдем в уравнении (1) от абсолютных величин к относительным, нормированным на L. Для этого введем обозначения:

$k(t) = K(t)/L(t)$ – величина капитала, приходящаяся на одного рабочего (капиталовооруженность);

$f(k(t)) = Y(t)/L(t)$ – выпуск продукции, приходящийся на одного рабочего (производительность труда);

$i(t) = I(t)/L(t)$ – капитальные вложения, приходящиеся на одного рабочего;

$c(t) = C(t)/L(t)$ – величина потребления на одного рабочего.

Разделив левую и правую части уравнения (1) на $L(t)$, получаем

$$f(k(t)) = \frac{\dot{K}(t)}{L(t)} + \mu \cdot k(t) + c(t). \qquad (2)$$

Необходимо выразить производную $\dot{K}(t)$ через $k(t)$, чтобы окончательно перейти от абсолютных значений к относительным. Скорость изменения величины капиталовооруженности рабочего:

$$\dot{k}(t) = \frac{d}{dt}\left(\frac{K(t)}{L(t)}\right)$$

$$= \frac{\dot{K}(t)\cdot L(t) - K(t)\cdot \dot{L}(t)}{L^2(t)}$$

$$= \frac{\dot{K}(t)}{L(t)} - k(t)\cdot\frac{\dot{L}(t)}{L(t)},$$

откуда следует:

$$\frac{\dot{K}(t)}{L(t)} = \dot{k}(t) + k(t)\cdot\frac{\dot{L}(t)}{L(t)}. \qquad (3)$$

В стабильных экономических условиях (без внутренних и внешних потрясений) население стран с течением времени возрастает по экспоненциальному закону e^{nt} с некоторым коэффициентом $n > 0$. Можно с большой достоверностью принять, что функция $L(t)$ также будет следовать этому закону:

$$L(t) = L(0)e^{nt}, \qquad (C)$$

где $L(0)$ и n – коэффициенты, характерные для рассматриваемой экономики. В этом случае

$$\frac{\dot{L}(t)}{L(t)} = \frac{n\cdot L(0)\cdot e^{nt}}{L(0)\cdot e^{nt}} = n \qquad (4)$$

отражает темп роста численности рабочей силы.

Подставив (4) в (3), а (3) в (2), получаем

$$f(k(t)) = \dot{k}(t) + n\cdot k(t) + \mu\cdot k(t) + c(t). \qquad (5)$$

Величина $n\cdot k(t)$ соответствует затратам, идущим на создание новых рабочих мест, связанных с ростом населения. Второе и третье слагаемые в уравнении (5) имеют одинаковую структуру. Их можно объединить, введя новый коэффициент $\lambda = n + \mu$. Тогда из уравнения (5) получаем ***основное дифференциальное уравнение неоклассической модели экономического роста***

$$\boxed{f(k(t)) = \dot{k}(t) + \lambda k(t) + c(t).} \qquad (D)$$

Его можно интерпретировать следующим образом: выпуск продукции, приходящийся на одного рабочего, раскладывается на три составляющие – расходы на поддержание капиталовооруженности рабочего на прежнем уровне $\lambda\cdot k(t)$; чистое изменение капиталовооруженности рабочего $\dot{k}(t)$; потребление на одного рабочего $c(t)$.

Если перейти к случаю с дискретным временем, где $t=t,t+1,t+2,\ldots$ то основное дифференциальное уравнение преобразуется к виду

$$f(k(t)) = (k(t+1) - k(t)) + \lambda k(t) + c(t). \qquad (6)$$

Уравнения (A), (B), (C) и (D) – базовые уравнения неоклассической модели роста.

2. Анализ модели

2.1. Графики функций

Рассмотрим, какие особенности динамики экономики следуют из этой модели. Начнем с того, что преобразуем уравнение (D) к виду

$$f(k(t)) - \lambda \cdot k(t) = \dot{k}(t) + c(t).$$

Функция $f(k)$ – нормированная на $L(t)$ производственная функция. Это возрастающая функция с убывающей первой производной (убывающая эффективность, убывающая предельная производительность). Расходы на поддержание капиталовооруженности – линейная функция k, (рис. 1, а).

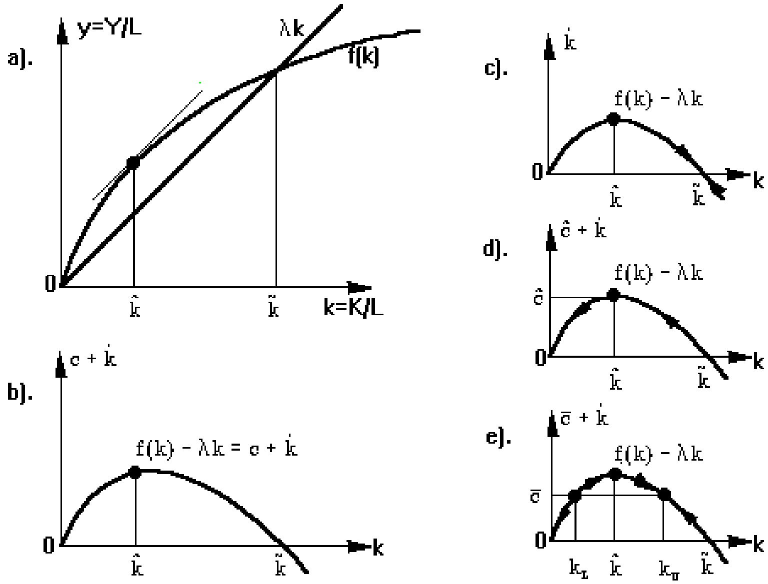

Рис. 1. Графики для основного дифференциального уравнения неоклассической модели роста:

a) графики функций $f(k)$ и $\lambda \cdot k$;

b) график разности $f(k(t)) - \lambda \cdot k(t)$;

c) нулевой уровень потребления *(c=0);*

d) максимальный уровень потребления *(c=* $\hat{c}$ *);*

e) промежуточный уровень потребления $(c = \bar{c},\ 0 < \bar{c} < \hat{c})$.

Обозначения в тексте.

В точке $\tilde{k}$ графики этих функций пересекаются и для $k > \tilde{k}$ расходы на поддержание капиталовооруженности рабочего $\lambda \cdot k$ превышают выпуск продукции, приходящейся на одного рабочего *f(k).* В этой области выпуска продукции недостаточно, чтобы обеспечить поддержание капиталовооруженности рабочего (обеспечить расширение рабочих мест в связи с ростом населения и восстановить изношенный капитал). На рис. 2, б представлена разность $f(k(t)) - \lambda \cdot k(t)$: кривая после точки $\tilde{k}$ лежит в области отрицательных значений функции.

В диапазоне $0 < k < \tilde{k}$ выпуск продукции *f(k)* больше необходимых отчислений $\lambda \cdot k$ и, следовательно, остаток от разности $f(k(t)) - \lambda \cdot k(t)$ направляется на потребление и увеличение капиталовооруженности.

Динамика экономики в этой области зависит от уровня потребления. Рассмотрим эту область отдельно и подробно.

2.2. Отсутствие потребления

Рассмотрим абстрактную ситуацию, когда потребление отсутствует: $c(t) = 0$. Этот вариант прост для интерпретации и позволит понять более реальные случаи.

Если рассмотрение состояний экономики начать с некоторой начальной капиталовооруженности, например с $\hat{k}$ (рис. 1, с), то здесь $\dot{k} > 0$. Это чистое увеличение капиталовооруженности. Следовательно, следующий экономический цикл через некоторый промежуток времени начнется с увеличенной капиталовооруженностью: текущая точка капиталовооруженности передвинется по оси абсцисс в направлении точки $\tilde{k}$. В новом положении чистое изменение капиталовооруженности также положительно, хотя и будет несколько меньше, чем в предыдущем случае. Поэтому движение во времени будет продолжаться с уменьшающейся скоростью в направлении точки $\tilde{k}$ до тех пор, пока k не станет равным $\tilde{k}$.

В точке $\tilde{k}$ величина $\dot{k} = 0$ и, следовательно, движение остановится.

Напротив, если начальная величина капиталовооруженности k находится в области $k > \tilde{k}$, то для этого положения чистое изменение капиталовооруженности отрицательно ($\dot{k} < 0$), капиталовооруженность уменьшается. Следующий экономический цикл начнется с меньшей капиталовооруженностью. Направление движения текущей точки k на графике к точке $\tilde{k}$ с остановкой в этой точке.

В реальной экономике это может соответствовать следующей ситуации: при очень большой капиталовооруженности велики амортизационные отчисления и расходы на расширение рабочих мест. Если выпуск не в состоянии обеспечить уровень необходимых отчислений, то капиталовооруженность рабочего будет уменьшаться.

Точка $\tilde{k}$ является ***точкой устойчивого экономического равновесия***: при отклонении в ту или другую сторону происходит возвращение капиталовооруженности в точку $\tilde{k}$.

2.3. Максимальное потребление

Если установить максимально возможный для данной экономической ситуации уровень потребления $c = \hat{c} = max$ (рис. 1, d), то во всей области $k > 0$ изменение капиталовооруженности идет в сторону уменьшения $(\dot{k} < 0)$. Действительно, при $c = \hat{c}$, как видно из графика

$$f(k) - \lambda \cdot k < \hat{c},$$

а в соответствии уравнением (D)

$$f(k) - \lambda \cdot k = \hat{c} + \dot{k},$$

поэтому

$$\hat{c} + \dot{k} < \hat{c}.$$

Исключение составляет лишь точка $k = \hat{k}$, в которой изменение капиталовооруженности нулевое $(\dot{k} = 0)$. Если начальное состояние экономики характеризуется значением $k > \hat{k}$, то с течением времени капиталовооруженность будет уменьшаться с уменьшающейся скоростью, пока k не станет равной $\hat{k}$ и движение прекратится.

Состояние экономики с $k = \hat{k}$ и $c = \hat{c}$ является ***равновесным, но неустойчивым***: если по каким-либо причинам (природные катастрофы, войны и т.д.) произойдет уменьшение капиталовооруженности, то при неизменном уровне потребления $c = \hat{c}$ движение будет продолжаться до точки $k = 0$.

2.4. Промежуточный уровень потребления

Для промежуточного уровня потребления $0 < \bar{c} < \hat{c}$ имеются две точки равновесия (рис. 1, е): в точке $k = k_U$ состояние экономики ***равновесное устойчивое***, а в точке $k = k_L$ – ***равновесное неустойчивое***. Анализ ситуации с динамикой капиталовооруженности аналогичен приведенному в разделах 2.2 и 2.3.

3. Контрольные вопросы

1. Перечислите уравнения, составляющие основу неоклассической модели роста.
2. Почему капиталовооруженность начинает уменьшаться, если начальный уровень ее слишком высок?
3. Почему состояние $k = k_U$ (рис. 1, е) является устойчивым равновесным, а состояние $k = k_L$ – неустойчивым равновесным?
4. Какой случай из представленных на рис. 1 является наиболее благоприятным для текущего потребления?
5. Какие действия необходимо предпринять, чтобы предотвратить уменьшение капиталовооруженности до нуля и, напротив, обеспечить ее увеличение в дальнейшем, если при $0 < c < \hat{c}$ капиталовооруженность оказалась в области $0 < k < k_L$?

4. Программы

4.1. Алгоритм вычислений

Алгоритм вычислений довольно прост и заключается в следующем:

a) используя определенное для момента времени t значение $k(t)$ (начальное значение $k(t=0)$ задается с клавиатуры по программному запросу), вычисляется величина

$$\boxed{DK = f(k(t)) - \lambda k(t) = b \cdot k(t)^a - \lambda k(t)}, \qquad \text{(I)}$$

где в качестве производственной функции используется функция Кобба-Дугласа: $f(t) = b \cdot k^a$.

b) Ставится точка на графике зависимости $f(k(t)) - \lambda k(t)$ от k.

c) Вычисляется значение $k(t+1)$ для следующего момента времени согласно уравнению (5):

$$k(t+1) = H \cdot \left[f(k(t)) - \lambda k(t) - c \right] + k(t).$$

(величина H < 1 введена для увеличения количества точек на графике) или

$$\boxed{k(t+1) = H \cdot [DK - c] + k(t)}. \qquad \text{(II)}$$

d) Происходит возврат к пункту a), если не выполнены условия окончания итерационной процедуры.

Обозначения:

A, B – значения параметров в производственной функции Кобба-Дугласа;
C – величина потребления;
EP – некоторое малое число для остановки итерационной процедуры;
H – величина H < 1 введена для увеличения количества точек на графике;
LAM – норма необходимых отчислений (LAM = λ).

4.2. QBASIC

```
' НЕОКЛАССИЧЕСКАЯ МОДЕЛЬ РОСТА[1]
SCREEN 2: CLS: KEY OFF
X0 = 100: Y0 = 110
LINE (X0, 16)-(X0, 199)
LINE (0, Y0)-(600, Y0)
LOCATE 1, X0 / 8 – 4: PRINT "dk/dt + c"
LOCATE Y0 / 8 – 6, X0 / 8 – 2: PRINT "+"
LOCATE Y0 / 8 + 5, X0 / 8 – 2: PRINT "-"
LOCATE Y0 / 8 – 1, 75: PRINT "k=K/L"
READ A, B
READ LAM, C
READ H, EP
LOCATE 18, 30: PRINT "c="; USING "##.###"; C
LOCATE 1, 28: PRINT "НЕОКЛАССИЧЕСКАЯ МОДЕЛЬ РОСТА"
XX = 20: YY =.3
SX = XX / (600 – X0): SY = YY / (Y0 – 8)
100 LOCATE 2, 54: PRINT SPC(8);
LOCATE 2, 28: INPUT "k=K/L НАЧАЛЬНОЕ ЗНАЧЕНИЕ ="; K
LOCATE 3, 28: PRINT SPC(52);
LOCATE 4, 28: PRINT SPC(52);
DK = 0
200 DK0 = DK
....                    'Вычисление значения dk(t)/dt + c
PSET (K / SX + X0, Y0 – DK / SY)
....                    'Новое k, увеличенное с множителем H
IF K < 0 THEN 300
LOCATE 20, 30: PRINT "k="; USING "##.######"; K
LOCATE 22, 18: PRINT "f(k)-lambda*k="; USING "##.######"; DK
IF ABS((DK0 – DK) / DK) > EP THEN 200
300 LOCATE 3, 28: PRINT "ПРОЦЕСС СОШЕЛСЯ! ИЗМЕНИТЕ k! "
LOCATE 2, 28: PRINT SPC(25);
GOTO 100
'****ДАННЫЕ****
DATA 0.5, 0.3
DATA 0.1, 0.225
DATA 0.1,0.00000001
```

4.3. C++

```
//Неоклассическая модель роста
/*Программная оболочка Borland C++ 5.x
```

При формировании Результирующего Объекта (Target) необходимо выбрать Application DOS(Standard) Large и включить кнопку BGI. Сведения о функциях BGI и файлы примеров расположены:
Help => Keyword search => Book Shelf => Borland C++ DOS Reference => Borland Grafic Interface => About.... */

```
#include <iostream.h>
#include <graphics.h>          //Для функций графики
#include <conio.h>             //Для функции getch()
#include <stdio.h>             //Для функции sprintf()()
#include <math.h>              //Для функции pow()
#include <dos.h>               //Для функции delay()
main()
{ double a=0.5, b=0.3;
  double lambda=0.1,c=0.225;
  double h=0.1;
  double epsilon=0.000001;
  double k=0,dk,dk0;
  char p[40],ch;
//Открытие графического режима. Экран размером 640 х 480 пикселей.
//Начало координат в верхнем левом углу.
//Последний параметр в initgraph – путь на BGI в Вашем PC.
//Далее две обязательных строки, чтобы открыть графический режим.
  int graphdriver = DETECT, graphmode;
  initgraph(&graphdriver, &graphmode,"D:\\bc5\\BGI");
  float xc=120,  yc=400;       //Координаты центра осей f-lambda*k, k
//Начертить линию от точки с координатами (xc,yc+160) до точки (xc,yc-
320).
  line(xc,yc+160,xc,yc-320);
//----------------------------------
  line(xc-40,yc,xc+370,yc);
//Вывести на экран "0" в точку с координатами (xc+10,yc+10).
  outtextxy(xc+10,yc+10,"0");
//----------------------------------
  outtextxy(xc+10,yc-310,"f-lambda*k");
  outtextxy(xc+360,yc+10,"k");
  outtextxy(xc+90,yc-390,"NEOCLASSIC GROWTH MODEL");
  outtextxy(xc+90,yc-370,"f = dk/dt-lambda*k+c");
  outtextxy(xc+90,yc-355,"c = ");
//Переформатирование и вывод значения с в графическом режиме
  sprintf(p,"%5.2f",c);
  outtextxy(xc+120,yc-355,p);
//----------------------------------
  outtextxy(xc+190,yc-355,"lambda = ");
```

```
  sprintf(p,"%5.2f",lambda);
  outtextxy(xc+260,yc-355,p);
//Масштабирование выводимых данных
double x_unit=30, y_unit=1000;
//Основной цикл
  for(;;)
  {
//Стирание предыдущей записи (вывод черным цветом)
     setcolor(0);
     outtextxy(xc+120,yc-320,"Stop!  ");
     outtextxy(xc+120,yc-305,"Good accuracy in iteration.   ");
     outtextxy(xc+120,yc-290,"k(fin) = ");
     sprintf(p,"%5.2f",k);
     outtextxy(xc+185,yc-290,p);
     setcolor(15);
//-----------------------------------
     cout<<"Input k: ";          cin>>k;
     dk=0;
     do
     {  dk0 = dk;
      .........
           putpixel(k*x_unit+xc, yc-dk*y_unit, 15);
      .........
       if(k<0) break;
      delay(1.0);          //Задержка.Подберите величину параметра
     }  while (fabs((dk0-dk)/dk)>epsilon);
//Вывод записи об окончании итерационной процедуры
     outtextxy(xc+120,yc-320,"Stop!  ");
     outtextxy(xc+120,yc-305,"Good accuracy in iteration.   ");
     outtextxy(xc+120,yc-290,"k(fin) = ");
     sprintf(p,"%5.2f",k);
     outtextxy(xc+185,yc-290,p);
//-----------------------------------
     cout<<"Next? <Y/N>";          cin>>ch;
     if(ch=='Y' || ch=='y') continue;
       else break;
  }
//Графики на экране, ожидается нажатие любой клавиши
    cout<<"Press any key";
    getch();
//Закрытие графического режима, обязательная строка.
   closegraph();
   return 0;
```

```
}
```

4.4. Borland C++ Builder

```
//В форму помещены следующие компоненты: кнопка для запуска
//программы из закладки Standard и Chartfx из ActiveX для построения
//графиков
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include "economics_growth_BCB.h"
#include <math.h>            //Для функции pow()
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{  double a=0.5, b=0.3;
   double lambda=0.1,c=0.2;
   double h=0.1;
   double epsilon=0.00000001;
   double k,dk,dk0;
   double x[10000],y[10000];
   int t=0;
   k=10.;
   dk=0.;
//Основной цикл
   do
   { dk0 = dk;
     …….
     x[t]=k;
     y[t]=dk;
     …….
     t++;
     if (t==9999) break;
     if(k<0) break;
   } while (fabs((dk0-dk)/dk)>epsilon);
```

```
//Открытие данных для графика типа scatter
   Chartfx1->OpenDataEx(COD_VALUES,2,10000);
   Chartfx1->OpenDataEx(COD_XVALUES,2,10000);
//Начертим линию уровня потребления
   Chartfx1->ThisSerie=1;
   for(int i=0; i<=100; i++)
   { Chartfx1->XValue[i]=0.1*i;
     Chartfx1->Value[i]=c;
   }
//Начертим график dk/dt + c
   Chartfx1->ThisSerie=0;
   for(int i=0; i<t; i++)
   { Chartfx1->Value[i]=y[i];
     Chartfx1->XValue[i]=x[i];
   }
//Закрытие данных графика
   Chartfx1->CloseData(COD_VALUES);
   Chartfx1->CloseData(COD_XVALUES);
}
//---------------------------------------------------------------------------
```

4.5. MathCAD

Программа неоклассической модели роста

```
M := | a ← 0.5
     | b ← 0.3
     | λ ← 0.1
     | c ← 0.1
     | h ← 0.4
     | ε ← 1 · 10^-5
     | k ← 1
     | dk ← 1
     | dk0 ← 0
     | t ← 0
     | while |(dk0 − dk)/dk| > ε
     |    | ▪
     |    | ▪
     |    | ▪
     |    | M^⟨t⟩ ← (dk, k)
     |    | break  if  k < 0
     |    | t ← t + 1
     | M
```

Результаты расчетов в среде MathCAD и шаблон для анимации графика приведены на рисунках.

15	0.221	1.715

Capital

Рис. 1. Результаты расчетов и график зависимости $f(k) - \lambda k$ от k для промежуточного уровня потребления: короткий пунктир – уровень потребления

$$j := \frac{FRAME}{20} \cdot tmax \qquad i := 0..j$$

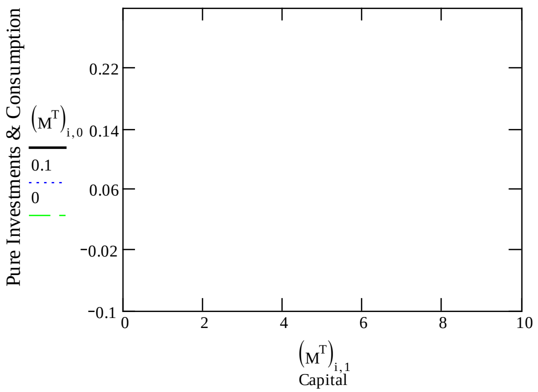

Рис. 2. Шаблон для анимации графика. Анимация позволяет увидеть направление изменения капиталовооруженности с течением времени

5. Задания

Необходимо построить графики динамик экономик для трех случаев уровней потребления: нулевого, промежуточного и максимального. Пользователь может выбрать приемлемый для него вариант работы в QBASIC[6], C++[7], Borland C++ Builder[8] или MathCAD[9]. Удобны для работы оболочки C++ Builder и MathCAD. Они избавляют пользователя от необходимости детальной прорисовки графиков: операторы для построения графиков в QBASIC и C++ в значительной степени перегружают текст программ.

1. Введите текст выбранного варианта программы в компьютер, дописав необходимые операторы. Задайте величину c=0. Исследуйте сходимость к равновесной точке $\tilde{k}$ при различных начальных значениях k. Попробуйте изменить показатель степени в производственной функции. Как изменяется поведение кривой и положение равновесной точки $\tilde{k}$?

2. Задайте величину $c = \bar{c}$ (рис. 1, e). Исследуйте поведение функции $f(k) - \lambda k$ в этом случае. Определите значения капиталовооруженности в равновесных точках.

3. Определите $\hat{c}$ и, задав $c = \hat{c}$ (рис. 1, d), поэкспериментируйте с функцией $f(k) - \lambda k$ в этом варианте.

МОДЕЛЬ ДЕЛОВОГО ЦИКЛА

1. Базовые положения модели

Развитие экономики демонстрирует в долгосрочном плане тенденцию к росту, на которую накладываются подъемы и спады конъюнктуры. Уже в XIX веке была введена классификация: сезонные колебания, краткосрочные циклы деловой конъюнктуры в 3–3,5 года, торгово-промышленные циклы в 7–11 лет. В XX веке эта классификация была дополнена обнаруженной экономистом Н.В. Кондратьевым долговременной периодичностью экономической конъюнктуры в 50–60 лет [13].

Можно построить математическую модель (Самуэльсон-Хикс, [1]), которая демонстрирует подъемы и спады валового выпуска $Y(t)$, если задать определенные зависимости от времени t для инвестиций $I(t)$ и потребительского спроса $C(t)$, связанных уравнением равенства спроса и предложения

$$\boxed{Y(t) = C(t) + I(t)}. \qquad \text{(A)}$$

Разобъем временной интервал на дискретные участки, введем лаг во времени $\Delta t = 1$ и будем фиксировать величины введенных в рассмотрение функций в моменты времени …, $t-2$, $t-1$, t, $t+1$, $t+2$, …. Величина инвестиций устанавливается в зависимости от ***величины изменения*** (роста или спада) валового выпуска в предыдущие моменты времени: инвестиции в момент времени t пропорциональны с коэффициентом v $(v>0)$ величине изменения объема выпуска на интервале $[t\text{-}2,\ t\text{-}1]$

$$\boxed{I(t) = v \cdot (Y(t-1) - Y(t-2))}. \qquad \text{(B)}$$

Как следует из (B), в период подъема конъюнктуры, когда $Y(t-1)>Y(t-2)$, величина инвестиций имеет положительный знак, а в моменты спада – отрицательный. Масштабный коэффициент v называется фактором акселерации.

Таким образом, ***объем инвестиций зависит от прироста и темпов изменения спроса на конечную продукцию***. По трактовке Хикса "… смысл этой формулы состоит в том, что новые инвестиции являются результатом изменений выпуска, имевших место в период t-1. Эти инвестиции планируются в период t-1, но из-за наличия временного лага в действительности осуществляются в период t "[1].

Потребительский спрос $C(t)$ определяется в данной модели следующим образом: он ***пропорционален валовому выпуску $Y(t-1)$, имевшему место в предыдущий рассматриваемому момент времени,***

$$\boxed{C(t) = aY(t-1) + b}. \qquad \text{(C)}$$

Коэффициент пропорциональности a $(0<a<1)$ называется склонностью к потреблению, а константа b $(b>0)$ – базовым потреблением.

Уравнения (A), (B) и (C) являются основой модели делового цикла. Подставляя (B) и (C) в (A), получаем, так называемое, разностное уравнение

$$\boxed{Y(t)=(a+v)Y(t-1)-vY(t-2)+b}. \qquad (D)$$

Функцию $Y(t)$ можно найти, решив уравнение (D). Это означает, что можно прогнозировать объемы валового выпуска для различных моментов времени, оставаясь, конечно, в рамках базовых положений (B) и (C) модели.

Можно усложнить модель, введя в нее государственное потребление

$$G(t)=(1+R)G(t-1),$$

где R – константа, равная темпу роста государственного потребления. В этом случае уравнение (D) перепишется в виде

$$\boxed{Y(t)=(a+v)Y(t-1)-vY(t-2)+b+G(t)}. \qquad (E)$$

2. Решение уравнений

2.1. Численное решение

Найти дискретные значения функции $Y(t)$ из уравнения (D) с помощью численных методов можно достаточно просто, если известны два значения валового выпуска в моменты, например, t и $t+1$. В этом случае значение $Y(t+2)$ сразу определяется из уравнения (D):

$$Y(t)=(a+v)Y(t-1)-vY(t-2)+b.$$

Подставив найденное значение вновь в уравнение (D), получаем

$$Y(t)=(a+v)Y(t-1)-vY(t-2)+b,$$

и т. д.

Другими словами, для нахождения дискретных значений функции $Y(t)$ применяется итерационная процедура. На языке программирования высокого уровня (C++, Fortran, Pascal, Basic) организация такой процедуры сводится к составлению тривиального цикла. Для визуализации полученных данных строятся графики зависимостей Y как функции t.

Задавая различные значения фактора акселерации v и склонности к потреблению a, т. е. изменяя вклады потребительского спроса и инвестиций в уравнении (A), можно исследовать поведение функции $Y(t)$ для каждого конкретного случая.

2.2. Аналитическое решение

Особенности поведения функции $Y(t)$ разностного уравнения (D) в зависимости от значений коэффициентов a и v станут более наглядны и понятны, если получить аналитическое решение разностного уравнения (D).

Получение аналитического решения упростится, если привести уравнение (D) к однородному виду. Для этого, во-первых, введем равновесное решение $Y(t)=Y(t-1)=Y^*$, соответствующее состоянию с неизменным выпуском. Его можно получить достаточно просто: если в уравнение (D) подставить Y^* вместо *Y(t), Y(t-1), Y(t-2),* получим

$$Y^*=b/(1-a). \tag{1}$$

Во-вторых, введем в рассмотрение новую функцию $y(t)=Y(t)-Y^*$, отражающую поведение валового выпуска, отсчитываемого от равновесного уровня Y^*. Подставляя в (D) вместо *Y(t), Y(t-1)* и *Y(t-2)* соответствующие им $y(t)+Y^*$, $y(t-1)+Y^*$ и $y(t-2)+Y^*$, получим однородное уравнение

$$y(t)=ay(t-1)+v(y(t-1)-y(t-2)),$$

или

$$y(t)-(a+v)\cdot y(t-1)+v\cdot y(t-2)=0. \tag{2}$$

Будем искать частные решения уравнения (2) в виде [14]

$$y(t)=q^t, \tag{3}$$

где q – число, подлежащее определению. Подставляя (3) в (2) получим квадратное уравнение

$$q^2-(a+v)\cdot q+v=0, \tag{4}$$

которое называется *характеристическим уравнением, соответствующим разностному уравнению* (2).

В зависимости от знака дискриминанта $(a+v)^2-4v$ могут представиться три различных случая. Если $(a+v)^2>4v$, то корни

$$q_1=\frac{(a+v)+\sqrt{(a+v)^2-4v}}{2}, \quad q_2=\frac{(a+v)-\sqrt{(a+v)^2-4v}}{2} \tag{5}$$

уравнения (4) вещественны и различны. В этом случае разностное уравнение (2) имеет частные решения

$$y_1(t)=q_1^t, \quad y_2(t)=q_2^t. \tag{6}$$

Если $(a+v)^2<4v$, то корни комплексно сопряжены:

$$q_1=\frac{(a+v)+i\sqrt{4v-(a+v)^2}}{2}, \quad q_2=\frac{(a+v)-i\sqrt{4v-(a+v)^2}}{2}, \tag{7}$$

где $i=\sqrt{-1}$. Функции (6) и в этом случае являются решениями разностного уравнения (2), однако удобнее представить их в тригонометрической форме $q_{1,2}=r(\cos\varphi\pm i\sin\varphi)$, где

$$r=\sqrt{\frac{(a+v)^2}{4}+\frac{4v-(a+v)^2}{4}}=\sqrt{v}\,,\ sin\,\varphi=\frac{\sqrt{4v-(a+v)^2}}{2\sqrt{v}}\,,\ cos\,\varphi=\frac{a+v}{2\sqrt{v}}\,. \quad (8)$$

В качестве решений уравнения (2) можно взять функции

$$y_1(t)=r^t\cos(t\varphi)\,,\qquad y_2(t)=r^t\sin(t\varphi)\,. \quad (9)$$

Наконец, если $(a+v)^2=4v$, то уравнение (4) имеет кратный корень $q=(a+v)/2$, а разностное уравнение имеет частные решения

$$y_1(t)=q^t\,,\qquad y_2(t)=t\cdot q^t\,. \quad (10)$$

Построим теперь решение *задачи Коши* (решение уравнения при заданных начальных условиях):

$$y(t)-(a+v)\cdot y(t-1)+v\cdot y(t-2)=0\,,\quad t=1,2,..., \quad (11)$$

$$y(0)=\mu_1\,,\qquad y(1)=\mu_2\,, \quad (12)$$

исходя из найденных частных решений (6). В силу линейности и однородности уравнения (2) любая линейная комбинация

$$y(t)=\alpha_1 q_1^t+\alpha_2 q_2^t \quad (13)$$

также является его решением. Подберем параметры α_1 и α_2 таким образом, чтобы удовлетворялись начальные условия (12):

$$\alpha_1+\alpha_2=\mu_1\,,\ \alpha_1 q_1+\alpha_2 q_2=\mu_2\,. \quad (14)$$

Решая систему (14), находим

$$\alpha_1=\frac{\mu_1 q_2-\mu_2}{q_2-q_1}\,,\qquad \alpha_2=\frac{\mu_2-\mu_1 q_1}{q_2-q_1}\,. \quad (15)$$

Подставляя (15) в (13) и собирая коэффициенты при μ_1, μ_2, получим, что решение задачи Коши (11), (12) в случае $(a+v)^2>4v$ имеет вид

$$\boxed{y(t)=\frac{q_1q_2(q_1^{t-1}-q_2^{t-1})}{q_2-q_1}\mu_1+\frac{q_2^t-q_1^t}{q_2-q_1}\mu_2\qquad t=0,1,2,...,} \quad (F)$$

где $q_1=\dfrac{(a+v)+\sqrt{(a+v)^2-4v}}{2}$, $q_2=\dfrac{(a+v)-\sqrt{(a+v)^2-4v}}{2}$.

В таком же виде представляется и решение задачи Коши (11), (12), когда $(a+v)^2<4v$. Заметим, что в этом случае, подставляя вместо q_1 и q_2 их выражения $q_{1,2}=r(\cos\varphi\pm i\sin\varphi)$, получим

$$\frac{q_1q_2(q_1^{t-1}-q_2^{t-1})}{q_2-q_1}=-r^t\,\frac{\sin((t-1)\varphi)}{\sin\varphi}\,,\qquad \frac{q_2^t-q_1^t}{q_2-q_1}=r^{t-1}\,\frac{\sin(t\varphi)}{\sin\varphi}$$

и решение задачи Коши

$$\boxed{y(t)=-r^t\frac{sin((t-1)\varphi)}{sin\varphi}\mu_1+r^{t-1}\frac{sin(t\varphi)}{sin\varphi}\mu_2, \quad t=0,1,2,...} \qquad (G)$$

где $r=\sqrt{\frac{(a+v)^2}{4}+\frac{4v-(a+v)^2}{4}}=\sqrt{v}, \quad \varphi=arctg(\frac{\sqrt{4v-(a+v)^2}}{(a+v)})$.

В случае $(a+v)^2=4v$, используя частные решения (10), можно представить решение задачи Коши (11),(12) в виде

$$\boxed{y(t)=-(t-1)q^t\mu_1+tq^{t-1}\mu_2, \quad q=(a+v)/2,\ t=0,1,2,...} \qquad (H)$$

Области изменения значений фактора акселерации v, в которых применимы решения (F), (G), (H), определяются из уравнения $(a+v)^2=4v$.

Перепишем его в виде

$$v^2-2(2-a)v+a^2=0$$

и введем величину *предельной склонности к сбережениям* $s=1-a$. Тогда уравнение

$$v^2-2(1+s)v+(1-s)^2=0$$

будет иметь два решения:

$$v_{1,2}=(1+s)\pm\sqrt{(1+s)^2-(1-s)^2}=(1+s)\pm2\sqrt{s} \qquad \text{или}$$

$$v_1=(1-\sqrt{s})^2 \quad \text{и} \quad v_2=(1+\sqrt{s})^2.$$

Таким образом, решения (F), (G), (H) определяют четыре вида динамики в следующих областях изменения фактора акселерации

значение ν: 0	$(1-\sqrt{s})^2$	1	$(1+\sqrt{s})^2$	∞
вид динамики:	затухающий, колебаний нет, решение (F)	затухающий, колебания, решение (G)	растущий, колебания, решение (G)	растущий, колебаний нет решение (F)

Формула (H) определяет решение в точках $v_1=(1-\sqrt{s})^2$ и $v_2=(1+\sqrt{s})^2$.

На рис. 1 и 2 приведены графики динамик валовых выпусков $Y(t)$, когда значение ν находится в диапазоне $(1-\sqrt{s})^2 < v < 1$, без государственного потребления и с ним. Расчеты выполнены в среде MathCAD.

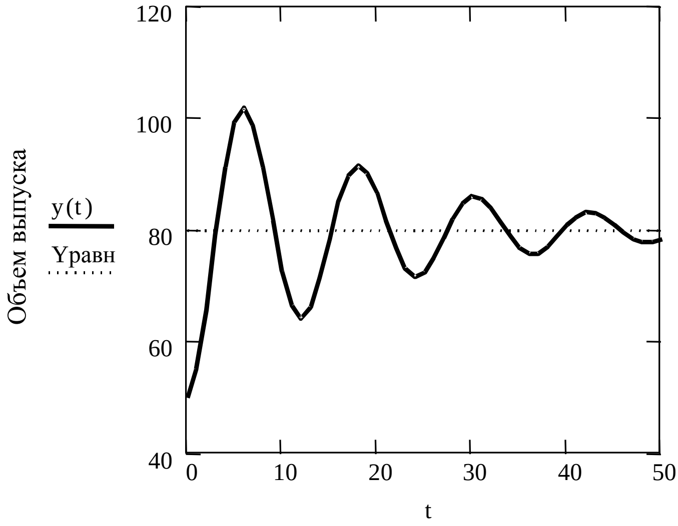

Рис. 1. Динамика валового выпуска без учета *G(t)*. Yравн соответствует Y^*

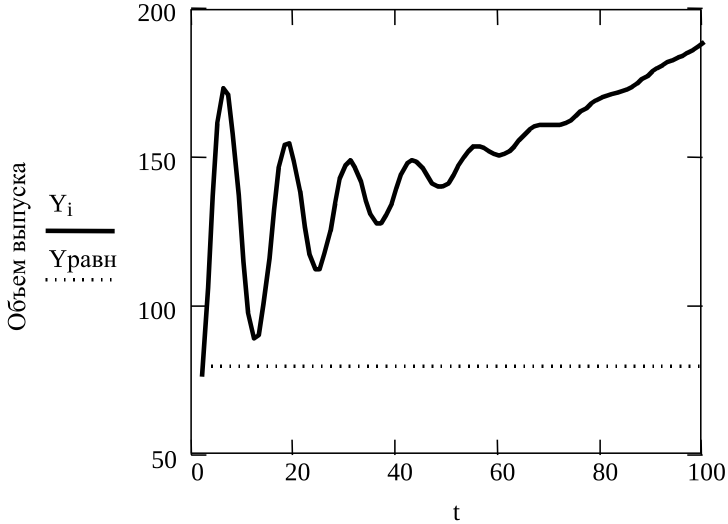

Рис. 2. Динамика валового выпуска с учетом *G(t)*. Yравн соответствует Y^*.

3. Контрольные вопросы

1. Перечислите основные, базовые положения модели делового цикла.
2. Как формулируется задача Коши для этой модели?
3. С помощью какого простого приема можно получить численное решение этого уравнения? Какие данные для этого необходимы?
4. Как влияет государственное потребление на валовой выпуск?
5. Какой вид динамики конъюнктуры, на Ваш взгляд, наиболее приемлем?

4. Программы

Ниже приведены заготовки программ для определения функций Y(t) из уравнений (D) и (E) с помощью итерационной процедуры, а также на основе аналитического решения – (F), (G), (H). Программы строят графики динамик национального дохода Y = Y(t). Пользователю предлагаются на выбор работы на одном из языков программирования – QBASIC[6],C++[7], Borland C++Builder[8] или проводить расчеты в среде MathCAD[9]. Последнее наиболее удобно, так как MathCAD позволяет работать с комплексными числами, выкладки просты, имеются широкие возможности для построения графиков.

Список переменных:
Y1 = Y(0) или Y(t-2), Y2 = Y(1) или Y(t-1);
A = a – склонность к потреблению;
B = b – базовое потребление;
V = v – фактор акселерации, или коэффициент инвестиций.

4.1. QBASIC

```
'МОДЕЛЬ ДЕЛОВОГО ЦИКЛА,  БЕЗ G(t) [2]
SCREEN 2: CLS
X0 = 50: Y0 = 168
LINE (X0, 0)-(X0, 399)
LINE (0, Y0)-(420, Y0)
LOCATE Y0 / 8 – 1, X0 / 8 – 1: PRINT "0"
LOCATE 1, X0 / 8 – 1: PRINT "Y"
LOCATE Y0 / 8 – 1, 52: PRINT "t";
LOCATE 1, 15: PRINT "МОДЕЛЬ ДЕЛОВОГО ЦИКЛА"
LOCATE 2, 15: PRINT "Y(t)=(a+v)*Y(t-1)-v*Y(t-2)+b"
LOCATE 6, 56: PRINT "Y(t)=C(t)+I(t)"
```

```
LOCATE 8, 56: PRINT "C(t)=a*Y(t-1)+b"
LOCATE 10, 56: PRINT "I(t)=v*(Y(t-1)-Y(t-2))"
LINE (432, 22)-(623, 91), 1, B
XX = 30: YY = 200
SX = XX / (420 – X0): SY = YY / (Y0 – 8)
A =.75: B = 20
LOCATE 15, 55: PRINT "a ="; A
LOCATE 16, 55: PRINT "b ="; B

LINE (X0, Y0 – B / (1 – A) / SY)-(420, Y0 – B / (1 – A) / SY)
320 Y1 = 50: Y2 = 55
LOCATE 13, 55: PRINT "Y(0)="; Y1
LOCATE 14, 55: PRINT "Y(1)="; Y2
LOCATE 17, 55: PRINT "v ="; V
LOCATE 3, 15: PRINT "Y(t)="; A + V; "*Y(t-1)-"; V; "*Y(t-2)+"; B; " "
LINE (X0, Y0 – Y1 / SY)-(X0 + 1 / SX, Y0 – Y2 / SY), 1
....
LINE -(X0 + T / SX, Y0 – Y / SY), 1
....
LOCATE 19, 57: INPUT "НАЖМИТЕ ENTER "; X$
LOCATE 17, 60: PRINT SPC(15);
LOCATE 17, 60: INPUT V
GOTO 320
```

```
'МОДЕЛЬ ДЕЛОВОГО ЦИКЛА,  БЕЗ G(t),ТОЧНОЕ РЕШЕНИЕ
SCREEN 2: CLS
X0 = 50: Y0 = 168
LINE (X0, 0)-(X0, 399)
LINE (0, Y0)-(420, Y0)
LOCATE Y0 / 8 – 1, X0 / 8 – 1: PRINT "0"
LOCATE 1, X0 / 8 – 1: PRINT "Y"
LOCATE Y0 / 8 – 1, 52: PRINT "t";
LOCATE 1, 15: PRINT "МОДЕЛЬ ДЕЛОВОГО ЦИКЛА"
LOCATE 2, 15: PRINT "Y(t)=(a+v)*Y(t-1)-v*Y(t-2)+b"
LOCATE 6, 56: PRINT "Y(t)=C(t)+I(t)"
LOCATE 8, 56: PRINT "C(t)=a*Y(t-1)+b"
LOCATE 10, 56: PRINT "I(t)=v*(Y(t-1)-Y(t-2))"
LINE (432, 22)-(623, 91), 1, B
XX = 30: YY = 200
SX = XX / (420 – X0): SY = YY / (Y0 – 8)
A =.75: B = 20
YEQ = B / (1 – A)
```

```
LOCATE 15, 55: PRINT "a ="; A
LOCATE 16, 55: PRINT "b ="; B
LINE (X0, Y0 – YEQ / SY)-(420, Y0 – YEQ / SY)
320 Y1 = 50: Y2 = 55
MU1 = Y1 – YEQ: MU2 = Y2 – YEQ
LOCATE 13, 55: PRINT "Y(0)="; Y1
LOCATE 14, 55: PRINT "Y(1)="; Y2
LOCATE 17, 55: PRINT "v ="; V
LOCATE 3, 15: PRINT "Y(t)="; A + V; "*Y(t-1)-"; V; "*Y(t-2)+"; B; " "
LINE (X0, Y0 – Y1 / SY)-(X0 + 1 / SX, Y0 – Y2 / SY), 1
....
400 LINE -(X0 + T / SX, Y0 – Y / SY), 1
'400 PSET (X0 + T / SX, Y0 – Y / SY)
....
LOCATE 19, 57: INPUT "НАЖМИТЕ ENTER "; X$
LOCATE 17, 60: PRINT SPC(15);
LOCATE 17, 60: INPUT V
GOTO 320
100 Q1 =
Q2 =
YA =
YB =
Y =
GOTO 400
200 R =
PHI =
YA =
YB =
Y =
GOTO 400
300 Q =
Y =
GOTO 400
```

4.2. C++

```
//Модель делового цикла, без G(t)
/*Программная оболочка Borland C++ 5.x
При формировании Результирующего Объекта (Target) необходимо
выбрать Application  DOS(Standard) Large  и включить кнопку  BGI.
Сведения о функциях BGI и файлы примеров расположены:
```

```
Help => Keyword search => Book Shelf => Borland C++ DOS Reference =>
Borland Grafic Interface => About.... */
#include <iostream.h>
#include <graphics.h>
#include <conio.h>
#include <stdio.h>

main()
{  float a=0.75, b=20, v;
   float Y,Y1,Y2;
   char p[40],ch;
//Открытие графического режима. Экран размером 640 x 480 пикселей.
//Начало координат в верхнем левом углу.
//Последний параметр в initgraph – путь на BGI в Вашем PC.
//Далее две обязательных строки, чтобы открыть графический режим.
   int graphdriver = DETECT, graphmode;
   initgraph(&graphdriver, &graphmode,"D:\\bc5\\BGI");
   float xc=120,  yc=400;                    //Координаты центра осей Y,t
//Начертить линию от точки с координатами (xc,yc+160) до точки (xc,yc-
320).
   line(xc,yc+160,xc,yc-320);
   line(xc-40,yc,xc+370,yc);
//Вывести на экран "0" в точку с координатами (xc+10,yc+10).
   outtextxy(xc+10,yc+10,"0");
   outtextxy(xc+10,yc-310,"Y");
   outtextxy(xc+360,yc+10,"t");
   outtextxy(xc+90,yc-390,"BUSINESS CYCLE MODEL");
   outtextxy(xc+60,yc-370,"Y(t)=(a+v)*Y(t-1)-v*Y(t-2)+b");
   outtextxy(xc+90,yc-355,"a = ");
   sprintf(p,"%5.2f",a);
   outtextxy(xc+120,yc-355,p);
   outtextxy(xc+190,yc-355,"b = ");
   sprintf(p,"%5.2f",b);
   outtextxy(xc+220,yc-355,p);
//Масштабирование выводимых данных
   int xx=45, yy=200;
   float sx=xx/(450-xc), sy=yy/(yc-8);
//Линия равновесного выпуска
   line(xc,yc-b/(1-a)/sy, xc+370, yc-b/(1-a)/sy);
   for(;;)
  {  Y1=50;  Y2=55;
     cout<<"Input v: ";          cin>>v;
     line(xc,yc-Y1/sy, xc+1/sx, yc-Y2/sy);
```

```
        moveto(xc+1/sx, yc-Y2/sy);
        for(…)
        {  ……
           lineto(xc+t/sx,yc-Y/sy);
           ……
        }
        cout<<"Next? <Y/N>";          cin>>ch;
          if(ch=='Y' || ch=='y') continue;
          else break;
   }
//Графики на экране, ожидается нажатие любой клавиши
    cout<<"Press any key";
    getch();
//Закрытие графического режима, обязательная строка.
   closegraph();
   return 0;
}
```

```
 //Модель делового цикла, без G(t), точное решение
/*Программная оболочка Borland C++ 5.x
При формировании Результирующего Объекта (Target) необходимо
выбрать Application  DOS(Standard) Large  и включить кнопку  BGI. */
#include <iostream.h>
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>
//Прототипы функций
double fun1(float t,float a,float v,float yeq,float mu1,float mu2);
double fun2(float t,float a,float v,float yeq,float mu1,float mu2);
double fun3(float t,float a,float v,float yeq,float mu1,float mu2);
main()
{ float a=0.75, b=20, v, mu1, mu2;
   float Y,Y1,Y2,yeq=b/(1-a);
   char p[40],ch;
//Открытие графического режима и вывод надписей
   int graphdriver = DETECT, graphmode;
   initgraph(&graphdriver, &graphmode,"D:\\bc5\\BGI");
   float xc=120,  yc=400;
   line(xc,yc+160,xc,yc-320);
   line(xc-40,yc,xc+370,yc);
   outtextxy(xc+10,yc+10,"0");
```

```
   outtextxy(xc+10,yc-310,"Y");
   outtextxy(xc+360,yc+10,"t");
   outtextxy(xc+90,yc-390,"BUSINESS CYCLE MODEL");
   outtextxy(xc+60,yc-370,"Y(t)=(a+v)*Y(t-1)-v*Y(t-2)+b");
   outtextxy(xc+90,yc-355,"a = ");
   sprintf(p,"%5.2f",a);
   outtextxy(xc+120,yc-355,p);
   outtextxy(xc+190,yc-355,"b = ");
   sprintf(p,"%5.2f",b);
   outtextxy(xc+220,yc-355,p);
//Масштабирование выводимых данных
   int xx=45, yy=200;
   float sx=xx/(450-xc), sy=yy/(yc-8);
//Линия равновесного выпуска
   line(xc,yc-b/(1-a)/sy, xc+370, yc-b/(1-a)/sy);
   Y1=50;  Y2=55;
   mu1=Y1-yeq;            mu2=Y2-yeq;
   for(;;)
   {  cout<<"Input v: ";          cin>>v;
      line(xc,yc-Y1/sy, xc+1/sx, yc-Y2/sy);
      moveto(xc+1/sx, yc-Y2/sy);
      for(…)
      {  …..
       …..
       lineto(xc+t/sx,yc-Y/sy);
      }
     cout<<"Next? <Y/N>";          cin>>ch;
     if(ch=='Y' || ch=='y') continue;
       else break;
   }
//Графики на экране, ожидается нажатие любой клавиши
   cout<<"Press any key";
   getch();
//Закрытие графического режима, обязательная строка.
  closegraph();
  return 0;
}
//Описание функций
double fun1(float t,float a,float v,float yeq,float mu1,float mu2)
{ double ya,yb,q1,q2;
   q1=
   q2=
   ya=
```

```
  yb=
  return ya+yb+yeq;
}
double fun2(float t,float a,float v,float yeq,float mu1,float mu2)
{ double ya,yb,phi,r=sqrt(v);
  phi=
  ya=
  yb=
  return ya+yb+yeq;
}
double fun3(float t,float a,float v,float yeq,float mu1,float mu2)
{ …….
  return …..;
}
```

4.3. Borland C++Builder

```
// Модель делового цикла, без G(t).
//На форму вызваны два компонента: кнопка из Standart для запуска
//программы и Chartfx из ActiveX для изображения графиков
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "b_cycle_BCB_tch.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
   : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{  float a=0.75, b=20, v;
   float Y1=50,Y2=55,Y;
   v=1.05;
//Открытие данных для графика
  Chartfx1->OpenDataEx(COD_VALUES,2,51);
//Начертим линию равновесного выпуска
  for(int t=0; t<=50; t++)
```

```
  { Chartfx1->ThisSerie=1;
    Chartfx1->Value[t]=b/(1-a);
  }
//Зададим положение двух начальных точек
  Chartfx1->ThisSerie=0;
  Chartfx1->Value[0]=Y1;
  Chartfx1->Value[1]=Y2;
//Начертим кривую выпуска
  for(……)
  { ……
     Chartfx1->ThisSerie=0;
     Chartfx1->Value[t]=Y;
    ……
  }
//Закрытие данных графика
  Chartfx1->CloseData(COD_VALUES);
}
//---------------------------------------------------------------------------


//Модель делового цикла, без G(t), точное решение
//На форму вызваны два компонента: кнопка из Standart для запуска
//программы и Chartfx из ActiveX для изображения графиков
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "b_cycle_accuracy_BCB_tch.h"
#include <math.h>
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
   : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{  float a=0.75,b=20,v;
  float Y1=50,Y2=55,Y,yeq=b/(1-a);
  float mu1=Y1-yeq, mu2=Y2-yeq;
```

```
  v=0.8;
//Открытие данных для графика
  Chartfx1->OpenDataEx(COD_VALUES,2,51);
//Начертим линию равновесного выпуска
  for(int t=0; t<=50; t++)
  { Chartfx1->ThisSerie=1;
    Chartfx1->Value[t]=b/(1-a);
  }
//Зададим положение двух начальных точек
  Chartfx1->ThisSerie=0;
  Chartfx1->Value[0]=Y1;
  Chartfx1->Value[1]=Y2;
//Начертим кривую выпуска
  for(……)
  {   if(pow(a+v,2) > 4*v)  Y=Form1->fun1(t,a,v,yeq,mu1,mu2);
      ……
      ……
    Chartfx1->ThisSerie=0;
    Chartfx1->Value[t]=Y;
  }
//Закрытие данных графика
  Chartfx1->CloseData(COD_VALUES);
}
//---------------------------------------------------------------------------
double __fastcall TForm1::fun1(float t,float a,float v,float yeq,float mu1,float mu2)
{  double ya,yb,q1,q2;
  q1= ……;
  q2= ……;
  ya= ……;
  yb= ……;
  return ya+yb+yeq;
}
double __fastcall TForm1::fun2(float t,float a,float v,float yeq,float mu1,float mu2)
{  double ya,yb,phi,r=sqrt(v);
  phi= ……;
  ya= ……;
  yb= ……;
  return ya+yb+yeq;
}
double __fastcall TForm1::fun3(float t,float a,float v,float yeq,float mu1,float mu2)
```

```
{  double q=(a+v)/2;
   return ……;
}

//---------------------------------------------------------------------------
#ifndef b_cycle_accuracy_BCB_tchH
#define b_cycle_accuracy_BCB_tchH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <chartfx3.hpp>
#include <OleCtrls.hpp>
//---------------------------------------------------------------------------
class TForm1: public TForm
{
__published:  // IDE-managed Components
        TButton *Button1;
        TChartfx *Chartfx1;
        void __fastcall Button1Click(TObject *Sender);
private:          // User declarations
public:           // User declarations
        __fastcall TForm1(TComponent* Owner);
       double __fastcall fun1(float t,float a,float v,float yeq,float mu1,float
mu2);
       double __fastcall fun2(float t,float a,float v,float yeq,float mu1,float
mu2);
       double __fastcall fun3(float t,float a,float v,float yeq,float mu1,float
mu2);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm1 *Form1;
//---------------------------------------------------------------------------
#endif
```

4.4. MathCAD

Приведено только начало страницы расчетов.

Модель делового цикла

Точное решение

Основное уравнение для совокупного выпуска имеет вид

$$Y(t) = (a+\nu)Y(t-1) - \nu Y(t-2) + b$$

Пусть

$a := 0.75 \quad b := 20 \quad \nu := 0.9 \quad m1 := 50 - \dfrac{b}{1-a} \quad m2 := 55 - \dfrac{b}{1-a}$

$$q1 := \frac{(a+\nu) + \sqrt{(a+\nu)^2 - 4\cdot\nu}}{2}$$

……

5. Задания

5.1. QBASIC, C++, Borland C++Builder

1. Введите в компьютер первую программу, дополнив ее необходимыми операторами, обеспечивающими выполнение итерационной процедуры в соответствии с формулой (D).
2. Задавая различные значения фактора акселерации ν, исследуйте поведение функции Y(t) для всех видов динамик, приведенных на с. 89.
3. Создайте копию первой программы. Внесите в нее необходимые изменения для выполнения расчетов по формуле (E). Выполните пункт 2 после отладки второй программы.
4. Создайте программу, реализующую получение точных решений в этой модели по формулам (F), (G) и (H).

5.2. MathCAD

1. Поскольку MathCAD способен производить вычисления с комплексными числами, то для рассмотрения всех возможных вариантов (формулы (F), (G) и (H)) достаточно производить вычисления лишь по формуле (F).
2. Задавая различные значения фактора акселерации ν, исследуйте поведение функции y(t) (формула (F)) для всех видов динамик, приведенных на с. 89. Постройте графики зависимости y = y(t) для всех видов динамик.
3. Решите эту же задачу численно, основываясь на разностном уравнении (D). Сравните соответствующие графики.
4. Включите в уравнение государственное потребление (формула (G)). Что изменилось в поведении функции Y(t)?
5. Постройте графики для функций C(t) и I(t) на одном рисунке для случая ν=1. Почему максимумы функции C(t) сдвинуты относительно максимумов функции I(t)?

ЛИТЕРАТУРА

1. Математическая экономика на персональном компьютере: Пер. с яп. / М.Кубонива, М.Табата, С.Табата, Ю.Хасэбэ; Под ред. М.Кубонива; Под ред. и с предисл. Е.З.Демиденко. – М.: Финансы и статистика, 1991. – 304 с.: ил.

2. Колемаев В.А. Математическая экономика: Учебник для вузов. – М.: ЮНИТИ, 1998. – 240 с.

3. Замков О.О., Толстопятенко А.В., Черемных Ю.Н. Математические методы в экономике: Учебник. – М.: МГУ им. М.В. Ломоносова, Издательство «ДИС», 1997. – 368 с.

4. Paul H. Douglas, The Theory of Wages. – New York, 1934. – P. 121-127.

5. Брандт З. Статистические методы анализа наблюдений / Пер. с англ. – М.: Мир, 1975. – 312 с.

6. Михайлов В.Ю., Степанников В.М. Современный Бейсик для IBM PC. Среда, язык, программирование. – М.: Изд-во МАИ, 1993. – 288 с.

7. Подбельский В.В. Язык С++: Учеб. пособие. 2-е изд., перераб. и доп. – М.: Финансы и статистика, 1996. – 560 с.

8. Рейсдорф К., Хендерсон К. Borland C++Builder. Освой самостоятельно / Пер. с англ. – М.: Изд-во БИНОМ, 1998 г. – 704 с.

9. Очков В.Ф. MathCAD 7 PRO для студентов и инженеров. – М.: КомпьютерПресс, 1998. – 384 с.

10. Шикин А.В., Боресков А.В. Компьютерная графика. Полигональные модели. – М.: ДИАЛОГ-МИФИ, 2000. – 464 с.

11. Интрилигатор М. Математические методы оптимизации и экономическая теория / Пер. с англ. – М.: Прогресс, 1975. – 607 с.

12. Турчак Л.И. Основы численных методов: Учеб. пособие. – М.: Наука. Гл. ред. физ.-мат. лит., 1987. – 320 с.

13. Лопатников Л.И. Экономико-математический словарь. – М.: Наука, 1993. – 448 с.

14. Самарский А.А., Гулин А.В. Численные методы: Учеб. пособие для вузов. – М.: Наука. Гл. ред. физ-мат. лит., 1989. – 432 с.

СОДЕРЖАНИЕ