# Практическое занятие №2. Исследование модели Солоу-Свана. Вариант -

## Формулировка задания

1. Рассчитать значение капиталовооруженности в стационарной точке k\* и точке, соответствующей золотому правилу накопления капитала  (для этого необходимо использовать соотношения:  и ).
2. Построить графики функций y(t) , k(t), i(t), c(t).
3. Построить график движения экономики к стационарной точке k\* и графически определить ее.
4. Построить графики, позволяющие графически определить точку , соответствующую золотому правилу накопления капитала.
5. Построить графики, описывающие темпы прироста капиталовооруженности при разных нормах сбережения.
6. Построить графики, описывающие темпы прироста производительности труда при разных нормах сбережения.

## Ход решения

Для каждого метода была разработанна отдельная программа.

Ее словесный алгоритм приведен ниже:

1. Запросить начало отрезка
2. Запросить конец отрезка
3. Запросить шаг отрезка
4. Запросить начальное значение Y0

Алгоритмы методов решения не был принципиально изменен, за исключением того, что начальные значения передаются в функции напрямую, а не запрашиваются у пользователя индивидуально для каждого прохода. Также результаты работы выводятся после окончания расчетов. В связи с этим словесное описание опущено, блок схемы приведены ниже.

## Блок-схемы функций программы

Основная функция.

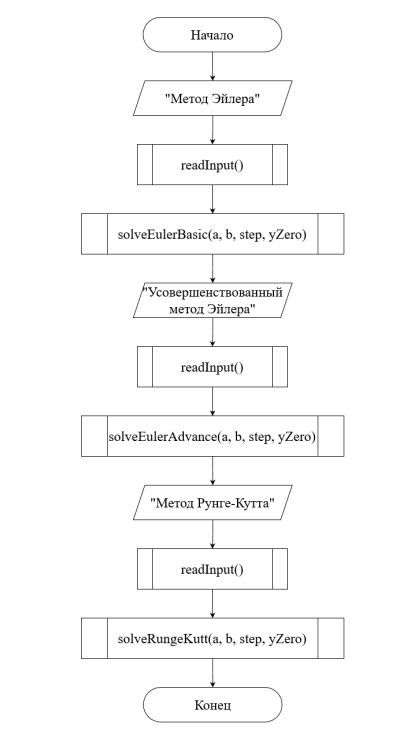


Рисунок 1. Функция считывания начальных значений

## Программа

import numpy as np

from scipy import stats

import matplotlib.pyplot as plt

import math as mth

plt.legend(fontsize=14)

# %УДЕЛЬНАЯ МОДЕЛЬ СОЛОУ БЕЗ НТП ДИСКРЕТНЫЙ ВАРИАНТ

#ЗАДАНИЕ ИСХОДНЫХ ДАННЫХ

#горизонт прогноза

T:int=100

#норма амортизации капитала

d:float=0.3

#норма сбережения

s=0.8

# темп роста труда

n=0.01

#коэффициент эластичности капитала

a=0.7

#РАСЧЕТ СТАЦИОНАРНОЙ ТОЧКИ

b:float=a-1

h:float=(n+d)/s

ks= h \*\* (1/b)

#РАСЧЕТ НОРМЫ СБЕРЕРЕЖЕНИЯ ДЛЯ ЗОЛОТОГО ПРАВИЛА

h1=(n+d)/a

kg= h1 \*\* (1/b)

sg=(kg\*(n+d))/ kg \*\* a

#МОДЕЛИВАНИЕ ДИНАМИКИ ЭКОНОМИКИ ПО ДИСКРЕТНОЙ МОДЕЛИ СОЛОУ И ЕЕ ГРАФИЧЕСКИЙ АНАЛИЗ

#начальное значение капиталовооруженности

k = np.zeros(T+1)

c = np.zeros(T)

inv = np.zeros(T)

y = np.zeros(T)

k[T]=0

t=0

k[t]=0.1

#Расчет эндогенных переменных модели для t

#выпуск

y[t]= k[t] \*\* a

#потребление

c[t]=(1-s)\*y[t]

#инвестиции

inv[t]=s\*y[t]

#прирост капиталовооруженности

k[t+1]=s\*y[t]+(1-(d+n))\*k[t]

for t in range(1,T):

    y[t]= k[t] \*\* a

    c[t]=(1-s)\*y[t]

    inv[t]=s\*y[t]

    k[t+1]=s\*y[t]+(1-(d+n))\*k[t]

k = k[:-1]

X = np.arange(T)

#ГРАФИКИ ДВИЖЕНИЯ ВО ВРЕМЕНИ ПАРАМЕТРОВ ЭКОНОМИКИ СОЛЛОУ

plt.figure(1)

plt.subplot(221)

plt.axis([1,T, 0, max(y)])

plt.plot(X, y,label='y(t)')

plt.xlabel('t')

plt.ylabel('y')

plt.subplot(222)

plt.axis([1,T, 0, max(k)])

plt.plot(X, k,label='k(t)')

plt.xlabel('t')

plt.ylabel('k')

plt.subplot(223)

plt.axis([1,T, 0, max(c)])

plt.plot(X, c,label='c(t)')

plt.xlabel('t', fontsize=16)

plt.ylabel('c', fontsize=16)

plt.subplot(224)

plt.axis([1,T, 0, max(inv)])

plt.plot(X, inv, label='inv(t)')

plt.xlabel('t', fontsize=16)

plt.ylabel('inv', fontsize=16)

#ГРАФИЧЕСКОЕ ОПРЕДЕЛЕНИЕ СТАЦИОНАРНОЙ ТОЧКИ ЭКОНОМИКИ СОЛОУ

plt.figure(2)

y1=np.array(y)\*s

y2=np.array(k)\*(n+d)

plt.plot(k,y, k, y1,'-r',k,y2,'-g','LineWidth',2 )

k1=max(k)

yy=max(y1)

ym=max(y)

plt.plot(k1,yy)

plt.title('СТАЦИОНАРНОЕ СОСТОЯНИЕ ЭКОНОМИКИ СОЛОУ')

plt.xlabel('k-капиталовооруженность')

plt.ylabel('у, sy, (d+n)k')

plt.text(0.1\*ks,0.9\*ym,'k\*расчет =')

plt.text(0.4\*ks,0.9\*ym, str(ks))

#ЗОЛОТОЕ ПРАВИЛО НАКОПЛЕНИЯ КАПИТАЛА

s0:float=sg-0.1

s1:float=sg

s2:float=sg+0.1

y11:float=np.array(y) \* s0

y12:float=np.array(y) \* s1

y13:float=np.array(y) \* s2

plt.figure(3)

plt.plot(k,y)

plt.plot(k, y2)

plt.plot(k,y, k, y2,k,y11,k,y12,k,y13)

plt.plot (kg,kg\*(n+d))

#Формирование надписей

plt.title('ЗОЛОТОЕ ПРАВИЛО ЭКОНОМИКИ СОЛОУ')

plt.xlabel('k-капиталовооруженность')

plt.ylabel('у, s0y, s1y, s2y, (d+n)k')

plt.text(0.1\*ks,0.88\*ym,'kg расчет =')

plt.text(0.4\*ks,0.88\*ym, str(kg))

#ДОЛГОСРОЧНАЯ ДИНАМИКА КАПИТАЛОВООРУЖЕННОСТИ И ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА

t=1

nd = [0 for i in range(T)]

nd[t]=(n+d)

for t in range(1, T):

    nd[t]= (n+d)

kk0=np.array(y)/np.array(k) \* s0

kk1=np.array(y)/np.array(k) \* s1

kk2=np.array(y)/np.array(k) \* s2

yy0=a\*(kk0-nd)

yy1=a\*(kk1-nd)

yy2=a\*(kk2-nd)

kkm=max(kk2)

yym=max(yy2)

plt.figure(4)

plt.axis([min(k),max(k), min(min(kk0), min(kk1), min(kk2)), max(max(kk0), max(kk1), max(kk2))])

plt.plot(k,kk0)

plt.plot(k,kk1)

plt.plot(k,kk2)

plt.plot(k,nd)

plt.title('ТЕМП РОСТА КАПИТАЛОВООРУЖЕННОСТИ ЭКОНОМИКИ СОЛОУ')

plt.xlabel('k-капиталовооруженность')

plt.ylabel('sy/k, (d+n)')

plt.text(0.1\*ks,0.8\*kkm,'s =')

plt.text(0.2\*ks,0.8\*kkm, str(s0))

plt.text(0.3\*ks,0.8\*kkm, str(s1))

plt.text(0.4\*ks,0.8\*kkm, str(s2))

plt.figure(5)

plt.axis([min(k),max(k), min(min(yy0), min(yy1), min(yy2)), max(max(yy0), max(yy1), max(yy2))])

plt.plot(k,yy0)

plt.plot(k,yy1)

plt.plot(k,yy2)

plt.title('ТЕМП РОСТА ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА ЭКОНОМИКИ СОЛОУ')

plt.xlabel('k-капиталовооруженность')

plt.ylabel('yy-темп роста производительности')

plt.text(0.1\*ks,0.8\*yym,'s =')

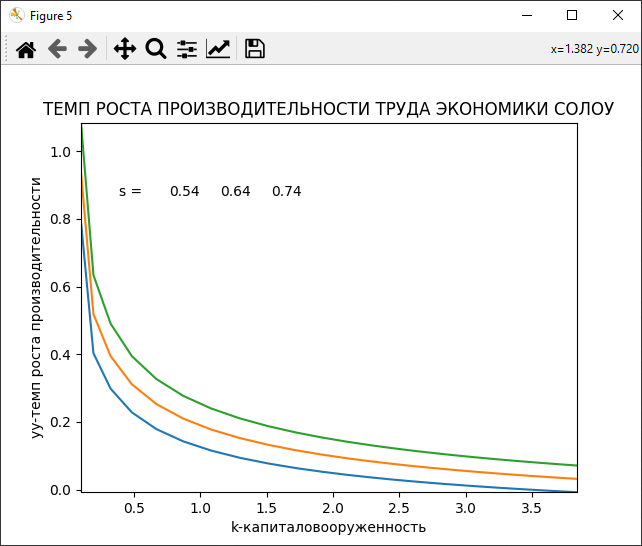
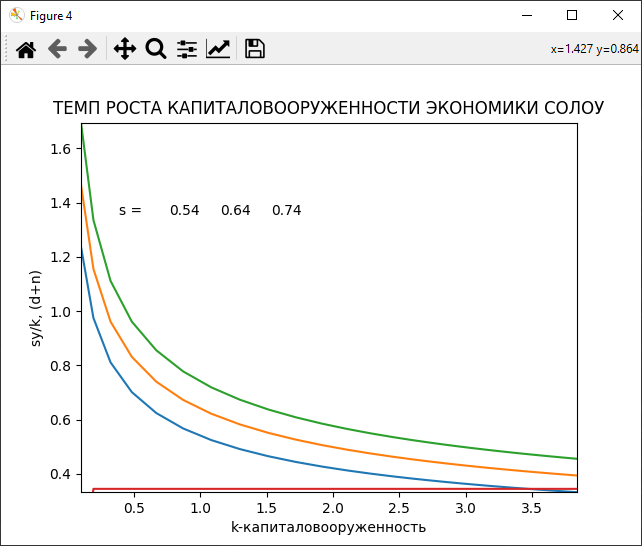
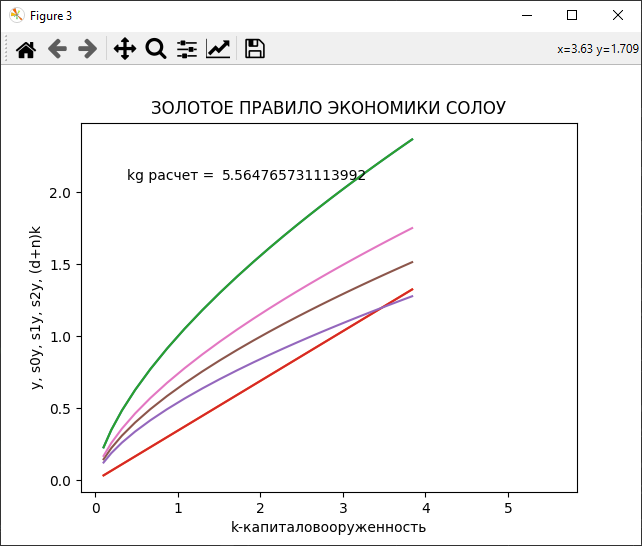
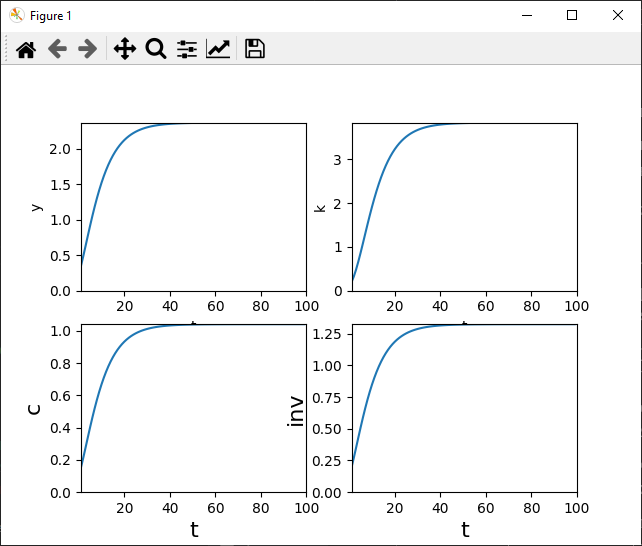
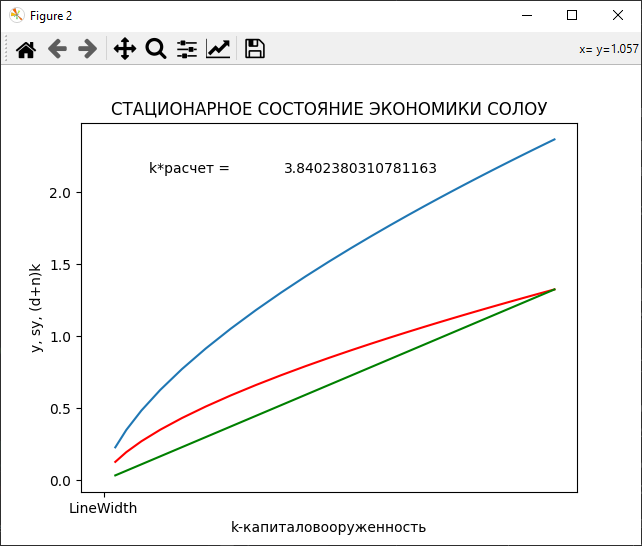
plt.text(0.2\*ks,0.8\*yym, str(s0))

plt.text(0.3\*ks,0.8\*yym, str(s1))

plt.text(0.4\*ks,0.8\*yym, str(s2))

plt.show()

## Скриншоты



## Вывод

В ходе работы были изучены модели для исследования дискретных моделей. Они были реализованы на языке программирования Python.