

Государственное бюджетное профессиональное образовательное учреждение города
Москвы "Московский колледж управления, гостиничного бизнеса и информационных
технологий "Царицыно" Отделение управления и информационных технологий.

Технические средства информатизации

Индивидуальный проект по теме 1:

Прогнозирование тенденций в изменении технологического процесса производства микропроцессоров

Студент: Уколов Алексей

Группа: ПЗ-3

Преподаватель: к.ф.-м.н. Мещеряков В.В.

Москва 2019

Оглавление

Введение:.....	4
1 Прогнозирование технического процесса на языках Matlab и Pythone.	6
1.1 Прогнозирование технического процесса на языках Matlab и Pythone до 2013.....	7
1.2 Прогнозирование технического процесса на языках Matlab и Pythone до 2017.....	9
1.3 Прогнозирование технического процесса на языках Matlab и Pythone до 2021.....	11
Заключение:	13
Литература и интернет источники:	14

Аннотация:

Прогнозирование тенденций в изменение технологического процесса производства микропроцессоров производится для того , чтобы понять , когда человечество достигнет предела изготовления по старой технологии производства микропроцессоров и перейдёт к новой.

Технические средства:

- процессоре Intel(r) Core(tr) i3-3210 CPU @ 3.20 GHZ.
- Windows 64x home версия
- ST1000DM003-1ER1
- B75MA-E31 (MS-7808) MSI

Введение:

Основная часть нашего прогнозирования основана на методе наименьших квадратов, с помощью которого можно отследить динамику развития микропроцессоров. **Метод наименьших квадратов (МНК)** — математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомым переменных. Он может использоваться для переопределенных систем уравнений для поиска решения в случае обычных нелинейных систем уравнений, для аппроксимации точечных значений некоторой функции. МНК является одним из базовых методов регрессионного анализа для оценки неизвестных параметров регрессионных моделей по выборочным данным.

Матричная формулировка:

$$\sum_{j=1}^p X_{ij} B_j = y_i (i = 1, 2, \dots, n),$$

Из n линейных уравнений по p неизвестных коэффициентов B_1, B_2, \dots, B_p при $n > p$.

Это можно записать в матричной форме как: $XB = y$, где

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}, B = \begin{pmatrix} B_1 \\ B_2 \\ \dots \\ B_p \end{pmatrix}, y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$

Такая система обычно не имеет точного решения, поэтому вместо этого нужно найти коэффициенты B которые соответствуют уравнениям «лучшим», в смысле решения квадратичной минимизации проблемы.

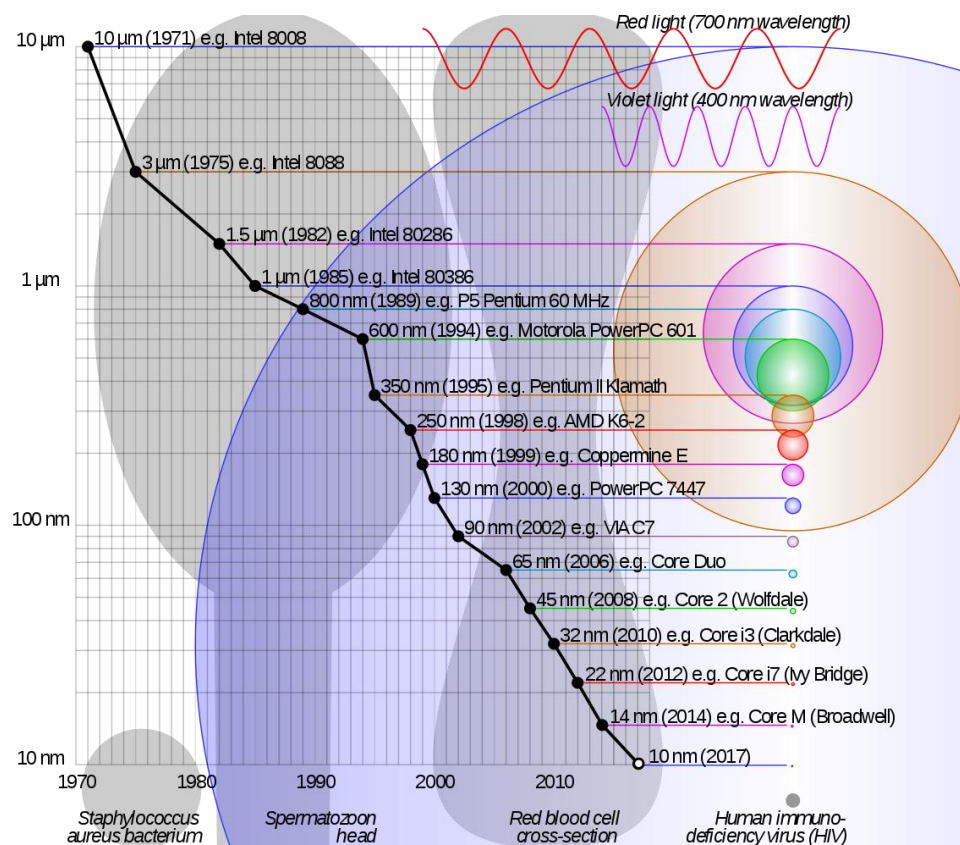
Обоснование выбора этого критерия приведено в разделе свойства ниже. Эта проблема минимизации имеет единственное решение при условии, что p столбцов матрицы X являются линейно независимыми. Данное решение нормальных уравнений.

$$(X^T X)B = X^T y$$

Матрицы $X^T X$ известен как определитель Грама из X , который обладает несколькими хорошими свойствами, такими как положительная полуопределённая матрица и матрица $X^T y$ известна матрица моментов регрессии и регрессоров. Наконец, является вектором коэффициента гиперплоскости наименьших квадратов, выраженной как

$$B = (X^T X)^{-1} X^T y$$

Динамика изменений.



1 Прогнозирование технического процесса на языках Matlab и Python.

MATLAB — это высокоуровневый язык и интерактивная среда для программирования, численных расчетов и визуализации результатов. С помощью MATLAB можно анализировать данные, разрабатывать алгоритмы, создавать модели и приложения.

Язык, инструментарий и встроенные математические функции позволяют вам исследовать различные подходы и получать решение быстрее, чем с использованием электронных таблиц или традиционных языков программирования, таких как C/C++ или Java.

MATLAB широко используется в таких областях, как:

1. обработка сигналов и связь,
2. обработка изображений и видео,
3. системы управления,
4. автоматизация тестирования и измерений,
5. финансовый инжиниринг,
6. вычислительная биология и т.п.

Python— высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций.

Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектноориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

NumPy — библиотека с открытым исходным кодом для языка программирования Python.

Возможности:

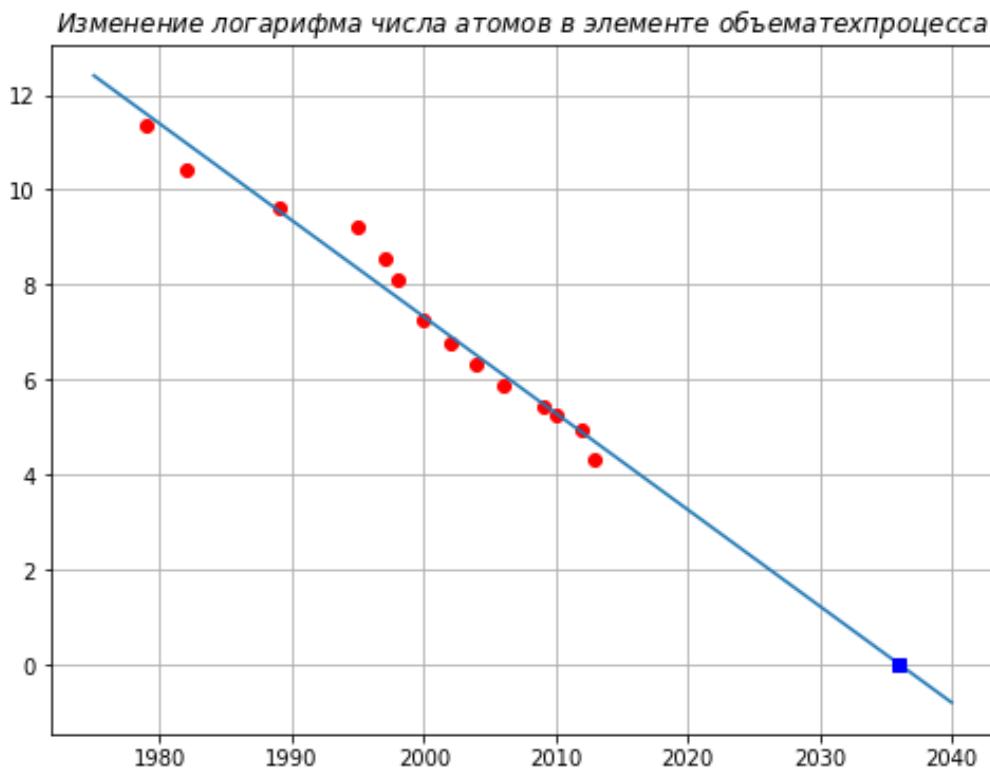
- поддержка многомерных массивов (включая матрицы);
- поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами.

1.1 Прогнозирование технического процесса на языках Matlab и Python до 2013.

Код листинга в python:

```
import matplotlib.pyplot as plt
import numpy as np
T = np.array([1979, 1982, 1989, 1995, 1997, 1998, 2000, 2002, 2004, 2006,
2009, 2010, 2012, 2013]);
L = np.array([3000, 1500, 800, 600, 350, 250, 130, 90, 65, 45, 32, 28, 22,14]);
a = 0.5
NL = L/a
N = NL**3;
Y = np.log10(N);
plt.figure(figsize=(8,6))
plt.plot(T,Y, 'ro')
lengthT = len(T);
X = np.array([np.ones(lengthT), T]).T
print('X = ');
print(X)
b = np.linalg.solve(np.dot(X.T,X), np.dot(X.T,Y.T));
print('b = ');
print (b);
Tend = 2040
t = np.linspace(1975,Tend,10)
F = b[0] + b[1]*t
plt.plot(t,F)
plt.plot(2036.0,0.0,'bs')
plt.grid(True)
plt.title('$Изменение \ логарифма \ числа \ атомов \ в \ элементе \ объема \
техпроцесса$')
plt.show()
```

РИС 1.1



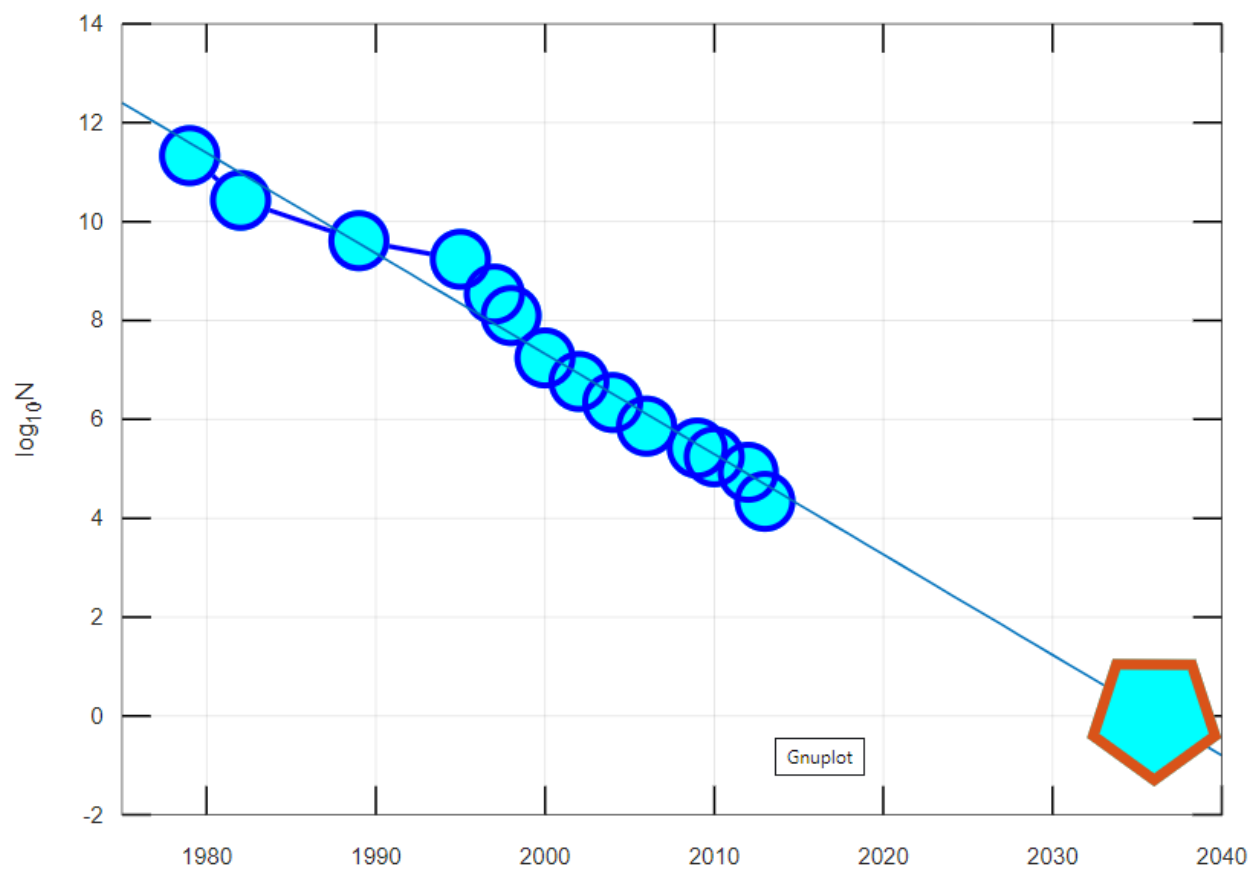
Код листинга в OCTAVE:

```

T = [1979 1982 1989 1995 1997 1998 2000 ... 2002 2004 2006 2009 2010
2012 2013]';
L = [3000 1500 800 600 350 250 130 ... 90 65 45 32 28 22 14]';
a = 0.5; NL = L/a; N = NL.^3;
Y = log10(N); transpY = Y'
plot(T,Y,'b.-','Marker','o',...
      'MarkerSize',8,'MarkerFaceColor','c','LineWidth',2)
Tend = 2040;
axis([1975 Tend -2 14]), hold on grid on, xlabel('T'),
ylabel('log_{10}N')
% Аппроксимация МНК
X = [ones(length(T),1) T];
b = (X'*X)^-1*X'*Y t = 1975:Tend;
F=b(1)+b(2)*t;
plot(t,F,'LineWidth',1)
plot(2036,0,'.','Marker','p',...
      'MarkerSize',14,'MarkerFaceColor','c','LineWidth',1)

```

РИС 1.2

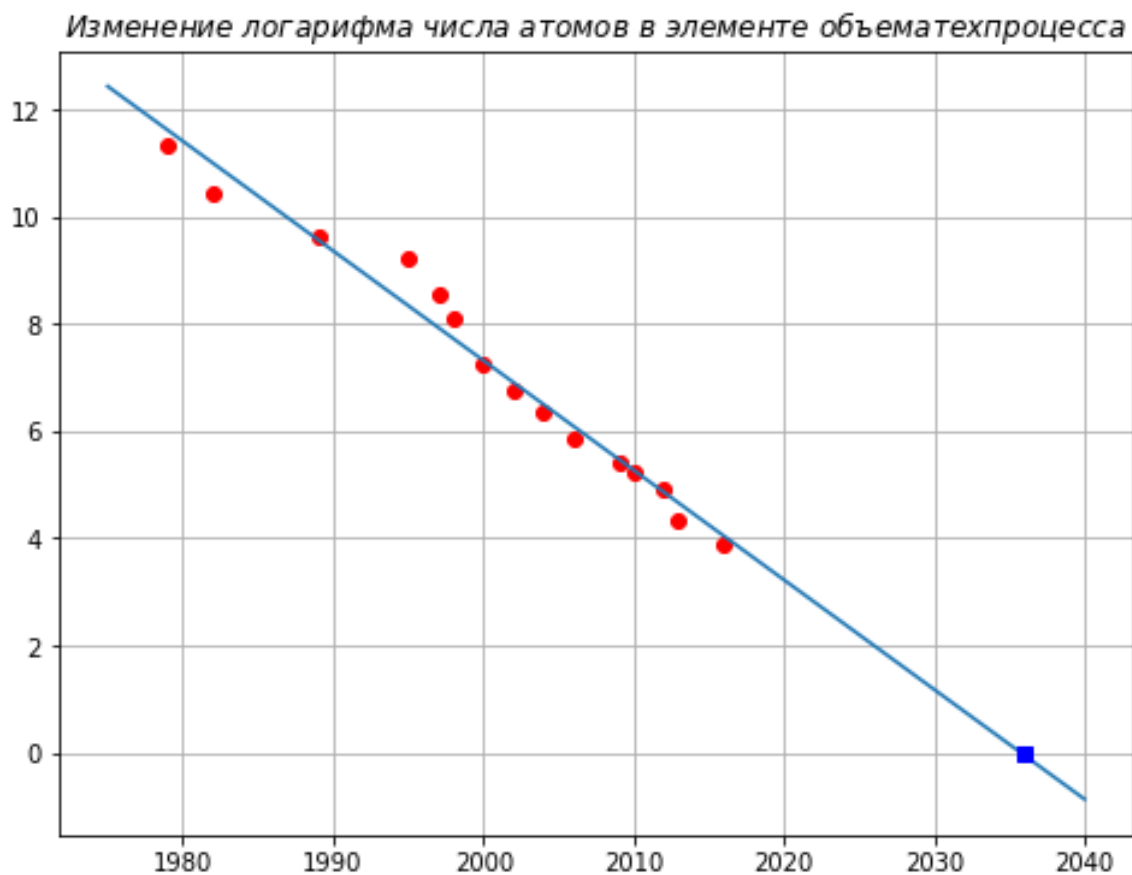


1.2 Прогнозирование технического процесса на языках Matlab и Python до 2017.

Код листинга в python:

```
import matplotlib.pyplot as plt
import numpy as np
T = np.array([1979, 1982, 1989, 1995, 1997, 1998, 2000, 2002, 2004, 2006,
2009, 2010, 2012, 2013, 2016]);
L = np.array([3000, 1500, 800, 600, 350, 250, 130, 90, 65, 45, 32, 28, 22, 14, 10]);
a = 0.5
NL = L/a
N = NL**3;
Y = np.log10(N);
plt.figure(figsize=(8,6))
plt.plot(T,Y, 'ro')
lengthT = len(T);
X = np.array([np.ones(lengthT), T]).T
print('X = ');
print(X)
b = np.linalg.solve(np.dot(X.T,X), np.dot(X.T,Y.T));
print('b = ');
print(b);
Tend = 2040
t = np.linspace(1975,Tend,10)
F = b[0] + b[1]*t
plt.plot(t,F)
plt.plot(2036.0,0.0,'bs')
plt.grid(True)
plt.title('$Изменение\логарифма\числа\атомов\в\элементе\объема\техпроцесса$')
plt.show()
```

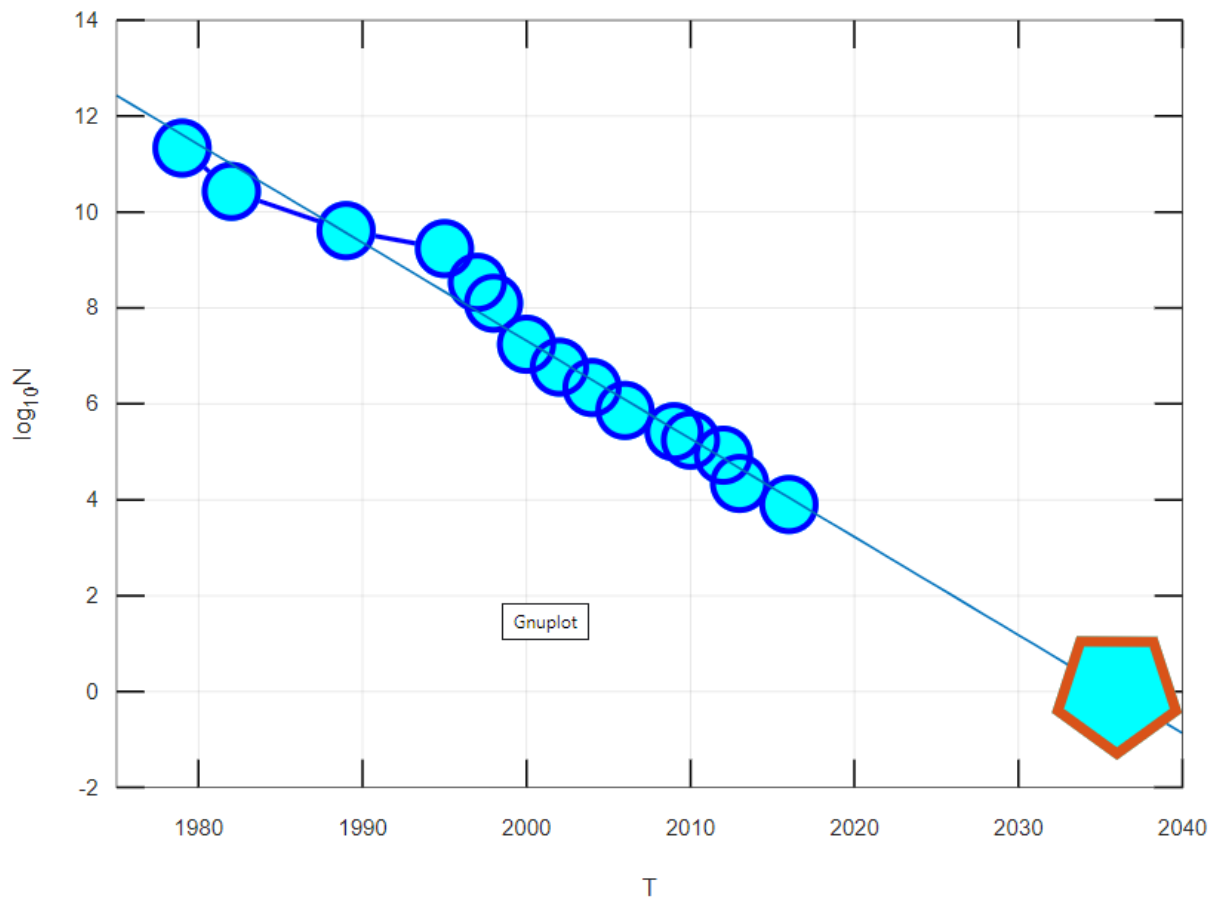
рис.2.1



Код листинга в OCTAVE:

```
T = [1979 1982 1989 1995 1997 1998 2000 ... 2002 2004 2006 2009 2010
2012 2013 2016]';
L = [3000 1500 800 600 350 250 130 ... 90 65 45 32 28 22 14 10]';
a = 0.5; NL = L/a; N = NL.^3;
Y = log10(N); transpY = Y'
plot(T,Y,'b.-','Marker','o',...
      'MarkerSize',8,'MarkerFaceColor','c','LineWidth',2)
Tend = 2040;
axis([1975 Tend -2 14]), hold on grid on, xlabel('T'),
ylabel('log_{10}N')
% Аппроксимация МНК
X = [ones(length(T),1) T];
b = (X'*X)^-1*X'*Y; t = 1975:Tend;
F=b(1)+b(2)*t;
plot(t,F,'LineWidth',1)
plot(2036,0,'.','Marker','p',...
      'MarkerSize',14,'MarkerFaceColor','c','LineWidth',1)
```

рис.2.2



1.3 Прогнозирование технического процесса на языках Matlab и Python до 2021.

Код листинга в python:

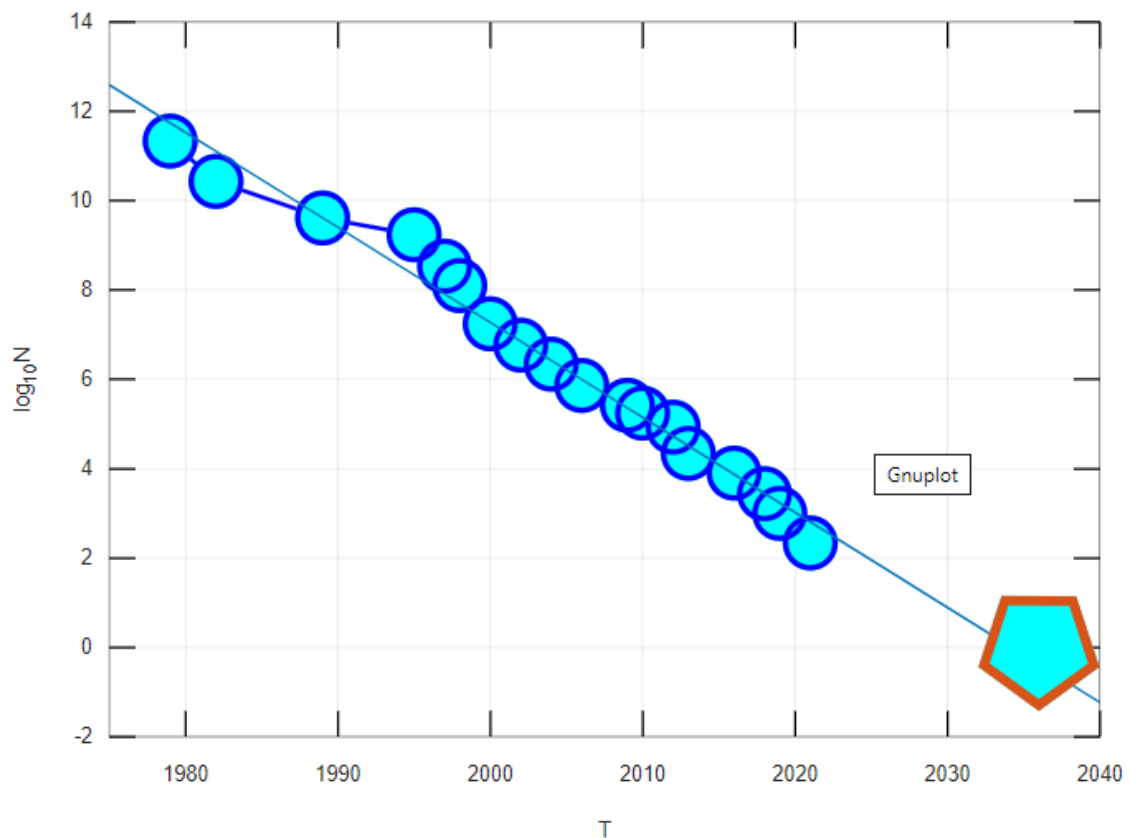
```
import matplotlib.pyplot as plt
import numpy as np
T = np.array([1979, 1982, 1989, 1995, 1997, 1998, 2000, 2002, 2004, 2006,
2009, 2010, 2012, 2013, 2016, 2018, 2019, 2021]);
L = np.array([3000, 1500, 800, 600, 350, 250, 130, 90, 65, 45, 32, 28, 22, 14,
10, 7, 5, 3]);
a = 0.5
NL = L/a
N = NL**3;
Y = np.log10(N);
plt.figure(figsize=(8,6))
plt.plot(T,Y, 'ro')
lengthT = len(T);
X = np.array([np.ones(lengthT), T]).T
print('X = ');
print(X)
b = np.linalg.solve(np.dot(X.T,X), np.dot(X.T,Y.T));
print('b = ');
print (b);
Tend = 2040
t = np.linspace(1975,Tend,10)
F = b[0] + b[1]*t
plt.plot(t,F)
plt.plot(2036.0,0.0, 'bs')
plt.grid(True)
plt.title('$Изменение\логарифма\числа\атомов\в\элементе\объема\техпроцесса$')
plt.show()
рис.3.1
```



Код листинга в OCTAVE:

```
T = [1979 1982 1989 1995 1997 1998 2000 ... 2002 2004 2006 2009 2010
2012 2013 2016 2018 2019 2021]';
L = [3000 1500 800 600 350 250 130 ... 90 65 45 32 28 22 14 10 7 5 3]';
a = 0.5; NL = L/a; N = NL.^3;
Y = log10(N); transpY = Y'
plot(T,Y,'b.-','Marker','o',...
      'MarkerSize',8,'MarkerFaceColor','c','LineWidth',2)
Tend = 2040;
axis([1975 Tend -2 14]), hold on grid on, xlabel('T'),
ylabel('log_{10}N')
% Аппроксимация МНК
X = [ones(length(T),1) T];
b = (X'*X)^-1*X'*Y t = 1975:Tend;
F=b(1)+b(2)*t;
plot(t,F,'LineWidth',1)
plot(2036,0,'.','Marker','p',...
      'MarkerSize',14,'MarkerFaceColor','c','LineWidth',1)
```

рис.3.2



Заключение:

Новая технология создания микропроцессоров появится примерно в 2035 году.

Литература и интернет источники:

1. https://en.wikipedia.org/wiki/Semiconductor_device_fabrication
2. https://en.wikipedia.org/wiki/Ordinary_least_squares
3. https://en.wikipedia.org/wiki/Microprocessor_chronology
4. <https://pythonworld.ru/numpy/4.html>
5. <https://numpy.org/>