ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Дожность		Колесникова С.И
старший преподаватель	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

ЛП. Нелинейное программирование. Вариационный принцип.

по дисциплине: Компьютерное моделирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	4236		Л. Мвале
		подпись, дата	инициалы, фамилия

Санкт-Петербург
2025

Часть 1. Модели линейного и нелинейного программирования.

Цель Работы ЧАСТЬ 1.

Цель настоящей работы – освоить средства моделирования задач линейного программирования.

Ход Работы

1. Ознакомиться со справочными сведениями;
2. Формализовать поставленную текстовую задачу.
3. Разработать шаблон в Excel для решения задачи, предусматривающего изменение начальных данных.
4. Разработать программу, моделирующую алгоритм поиска оптимального решения для формализованной задачи, используя вычислительный пакет MathLab или язык программированияPython.
5. Составить и представить преподавателю отчет о работе и устно защитить.

a. Постановка задачи

Вариант 15

Для серийного изготовления детали механический цех может использовать пять различных технологий её обработки на токарном, фрезерном, строгальном и шлифовальном станках. В таблице указано время (в минутах) обработки детали на каждом станке в зависимости от технологического способа, а также общий ресурс рабочего времени станков каждого вида за одну смену.

Станки	Условный код технологии	Ресурс времени

	1	**2**	**3**	**4**	**5**	станков (мин)
Токарный	2	1	3	0	1	4100
Фрезерный	1	0	2	2	1	2000
Строгальный	1	2	0	3	2	5800
Шлифовальный	3	4	2	1	1	10800

Требуется указать, как следует использовать имеющиеся технологии, с тем чтобы добиться максимального выпуска продукции.

b. Формализованную ПЗ с использованием терминологии ЛП

Для постановки задачи линейного программирования введём следующие обозначения:

Переменные (решения задачи)

Пусть:

$$x_1, x_2, x_3, x_4, x_5$$

— количество деталей, изготавливаемых по технологиям Т1–Т5 соответственно.

Целевая функция (функция оптимизации)

Необходимо максимизировать общее количество произведённых деталей:

$$Z = x_1 + x_2 + x_3 + x_4 + x_5 \quad \to \max$$

Ограничения (ресурсы станков)

Так как каждый станок имеет ограниченный фонд времени, составим систему неравенств:

1. **Токарный станок** (ресурс = 4100 мин):

$$2x_1 + 0x_2 + 3x_3 + 4x_4 + 1x_5 \le 4100$$

2. **Фрезерный станок** (ресурс = 2000 мин):

$$1x_1 + 2x_2 + 3x_3 + 2x_4 + 1x_5 \le 2000$$

3. **Строгальный станок** (ресурс = 5800 мин):

$$1x_1 + 1x_2 + 1x_3 + 0x_4 + 2x_5 \le 5800$$

4. **Шлифовальный станок** (ресурс = 10800 мин):

$$3x_1 + 2x_2 + 0x_3 + 1x_4 + 1x_5 \le 10800$$

Условие неотрицательности переменных

$$x_i \ge 0, \quad i = 1, 2, 3, 4, 5$$

Формализованная постановка задачи ЛП

Максимизировать: $Z = x_1 + x_2 + x_3 + x_4 + x_5$

при условиях:

$$\begin{aligned}
&2x_1 + 0x_2 + 3x_3 + 4x_4 + 1x_5 \le 4100 \\
&1x_1 + 2x_2 + 3x_3 + 2x_4 + 1x_5 \le 2000 \\
&1x_1 + 1x_2 + 1x_3 + 0x_4 + 2x_5 \le 5800 \\
&3x_1 + 2x_2 + 0x_3 + 1x_4 + 1x_5 \le 10800 \\
&x_i \ge 0 \quad (i = 1..5)
\end{aligned}$$

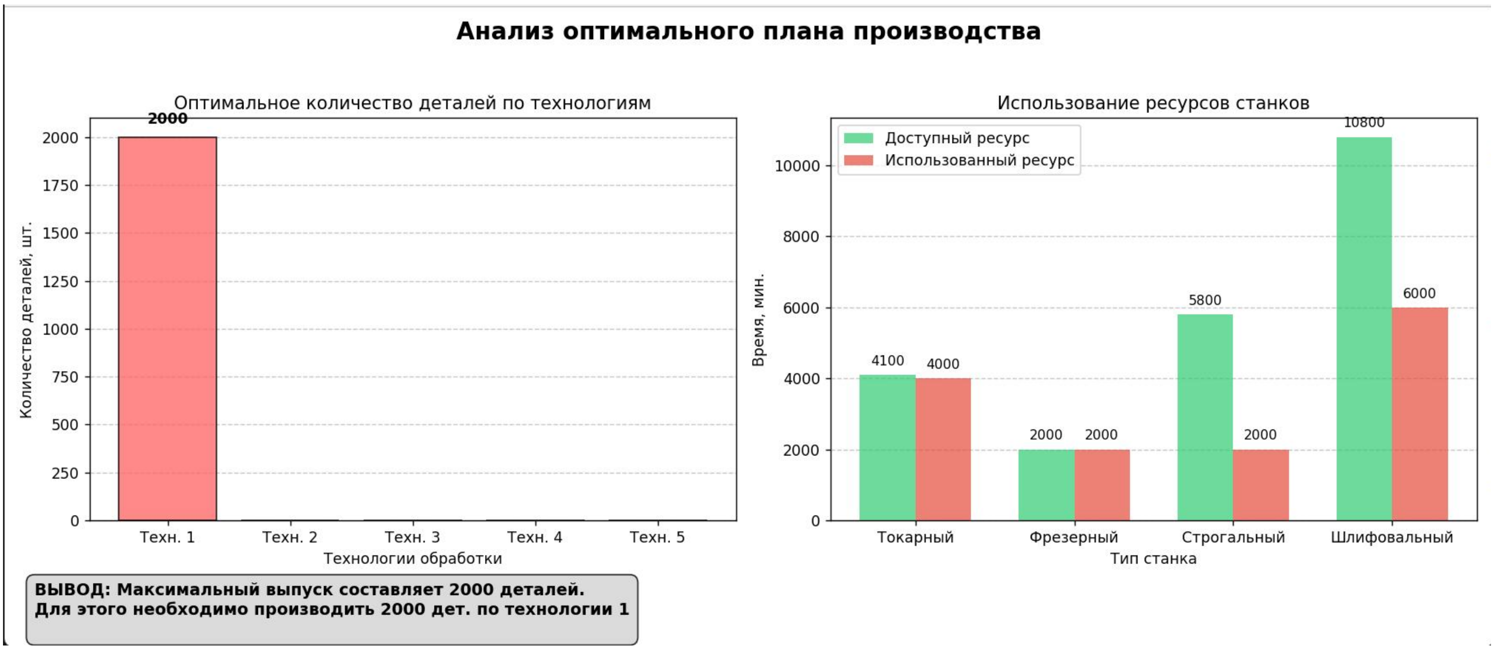
c. скриншоты решения Excel

Станок	T1	T2	T3	T4	T5	Ресурс
Токарный	2	0	3	4	1	4100
Фрезерный	1	2	3	2	1	2000
Строгальный	1	1	1	0	2	5800
Шлифовальный	3	2	0	1	1	10800

Станок	Использовано время	Ресурс
Токарный	4000	4100
Фрезерный	2000	2000
Строгальный	2000	5800
Шлифовальный	6000	10800

Технология	Переменная
T1	2000
T2	0
T3	0
T4	0
T5	0
Всего деталей	2000

d. скриншоты работы программы




```
================================================
ОТЧЕТ ПО РЕШЕНИЮ ЗАДАЧИ
================================================
Статус решения: Optimization terminated successfully. (HiGHS Status 7: Optimal)

Оптимальное количество деталей по технологиям:
x1 (Техн. 1): 2000.0 шт.
x2 (Техн. 2): 0.0 шт.
x3 (Техн. 3): 0.0 шт.
x4 (Техн. 4): 0.0 шт.
x5 (Техн. 5): 0.0 шт.

Максимальный суммарный выпуск: 2000.0 деталей.

Проверка использования ресурсов:
  Токарный: 4000 / 4100 мин. (97.6%)
  Фрезерный: 2000 / 2000 мин. (100.0%)
  Строгальный: 2000 / 5800 мин. (34.5%)
  Шлифовальный: 6000 / 10800 мин. (55.6%)

Сравнение с решением из Excel:
Из Excel: T1=0, T2=0, T3=0, T4=0, T5=2000, всего=2000
Из Python: T1=2000, T2=0, T3=0, T4=0, T5=0, всего=2000
================================================
```

Листинг программы

```
# -*- coding: utf-8 -*-
"""
Решение задачи линейного программирования для Варианта 15 с визуализацией.
Цель: Максимизация выпуска деталей при ограничениях на ресурсы станков.
Данные соответствуют Excel:
Токарный: [2, 0, 3, 4, 1]
Фрезерный: [1, 2, 3, 2, 1]
Строгальный: [1, 1, 1, 0, 2]
Шлифовальный: [3, 2, 0, 1, 1]
"""
# Импорт необходимых библиотек
import numpy as np
from scipy.optimize import linprog
import matplotlib.pyplot as plt

# 1. РЕШЕНИЕ ЗАДАЧИ ЛП

# Коэффициенты целевой функции (для минимизации: -1 * сумму x_i)
c = [-1, -1, -1, -1, -1]

# Матрица коэффициентов ограничений "меньше или равно" (A_ub * x <= b_ub)
# Обновлено согласно данным из Excel
A_ub = [
    [2, 0, 3, 4, 1],  # Токарный
```

```
    [1, 2, 3, 2, 1],  # Фрезерный
    [1, 1, 1, 0, 2],  # Строгальный
    [3, 2, 0, 1, 1]   # Шлифовальный
]
b_ub = [4100, 2000, 5800, 10800] # Ресурсы времени
# Границы переменных (x_i >= 0)
bounds = [(0, None)] * 5
# Решение задачи
result = linprog(c, A_ub=A_ub, b_ub=b_ub, bounds=bounds, method='highs')
# Извлечение результатов
optimal_quantities = result.x
total_output = -result.fun  # Преобразуем обратно к максимуму

# 2. СОЗДАНИЕ ГРАФИКОВ
# Настройка стиля и размера графиков
plt.style.use('default')
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))
fig.suptitle('Анализ оптимального плана производства', fontsize=16,
fontweight='bold')

# --- ГРАФИК 1: Оптимальный план по технологиям ---
technologies = ['Техн. 1', 'Техн. 2', 'Техн. 3', 'Техн. 4', 'Техн. 5']
colors = ['#FF6B6B', '#4ECDC4', '#FFE66D', '#9b59b6', '#3498db']
bars = ax1.bar(technologies, optimal_quantities, color=colors, edgecolor='black',
alpha=0.8)
# Добавление значений на столбцы
```

```
for bar, value in zip(bars, optimal_quantities):
    height = bar.get_height()
    if value > 0:  # Подписываем только ненулевые значения
        ax1.text(bar.get_x() + bar.get_width()/2., height + 50,
                 f'{int(value)}', ha='center', va='bottom', fontweight='bold')

ax1.set_title('Оптимальное количество деталей по технологиям')
ax1.set_ylabel('Количество деталей, шт.')
ax1.set_xlabel('Технологии обработки')
ax1.grid(axis='y', linestyle='--', alpha=0.7)
ax1.set_axisbelow(True)

# --- ГРАФИК 2: Использование ресурсов станков ---
machine_types = ['Токарный', 'Фрезерный', 'Строгальный', 'Шлифовальный']
resource_available = b_ub
# Рассчитываем фактически использованное время для каждого станка
resource_used = [
    2*optimal_quantities[0] + 0*optimal_quantities[1] + 3*optimal_quantities[2] + 4*optimal_quantities[3] + 1*optimal_quantities[4], # Токарный
    1*optimal_quantities[0] + 2*optimal_quantities[1] + 3*optimal_quantities[2] + 2*optimal_quantities[3] + 1*optimal_quantities[4], # Фрезерный
    1*optimal_quantities[0] + 1*optimal_quantities[1] + 1*optimal_quantities[2] + 0*optimal_quantities[3] + 2*optimal_quantities[4], # Строгальный
    3*optimal_quantities[0] + 2*optimal_quantities[1] + 0*optimal_quantities[2] + 1*optimal_quantities[3] + 1*optimal_quantities[4]  # Шлифовальный
]
```

```
x_pos = np.arange(len(machine_types))

bar_width = 0.35

bars1 = ax2.bar(x_pos - bar_width/2, resource_available, bar_width,
label='Доступный ресурс', color='#2ecc71', alpha=0.7)

bars2 = ax2.bar(x_pos + bar_width/2, resource_used, bar_width,
label='Использованный ресурс', color='#e74c3c', alpha=0.7)



# Добавление значений на столбцы

for i, (avail, used) in enumerate(zip(resource_available, resource_used)):

    ax2.text(i - bar_width/2, avail + 200, f'{int(avail)}', ha='center', va='bottom',
fontsize=9)

    ax2.text(i + bar_width/2, used + 200, f'{int(used)}', ha='center', va='bottom',
fontsize=9)



ax2.set_title('Использование ресурсов станков')

ax2.set_ylabel('Время, мин.')

ax2.set_xlabel('Тип станка')

ax2.set_xticks(x_pos)

ax2.set_xticklabels(machine_types)

ax2.legend()

ax2.grid(axis='y', linestyle='--', alpha=0.7)

ax2.set_axisbelow(True)



# Добавляем общий вывод на figure

output_text = f'ВЫВОД: Максимальный выпуск составляет {int(total_output)}
деталей.\n'

for i, tech in enumerate(technologies):

    if optimal_quantities[i] > 0:
```

```
        output_text += f'Для этого необходимо производить {int(optimal_quantities[i])} дет. по технологии {i+1}\n'


plt.figtext(0.02, 0.02, output_text,
            bbox=dict(boxstyle="round,pad=0.5", facecolor="lightgray", alpha=0.8),
            fontsize=11, fontweight='bold')


# Adjust layout and display
plt.tight_layout(rect=[0, 0.1, 1, 0.95]) # Оставляем место для текста внизу
plt.show()


# 3. ВЫВОД РЕЗУЛЬТАТОВ В КОНСОЛЬ
print("="*50)
print("ОТЧЕТ ПО РЕШЕНИЮ ЗАДАЧИ")
print("="*50)
print(f"Статус решения: {result.message}")
print("\nОптимальное количество деталей по технологиям:")
for i, tech in enumerate(technologies, 1):
    print(f"x{i} ({tech}): {optimal_quantities[i-1]:.1f} шт.")
print(f"\nМаксимальный суммарный выпуск: {total_output:.1f} деталей.")


print("\nПроверка использования ресурсов:")
for i, machine in enumerate(machine_types):
    print(f"  {machine}: {resource_used[i]:.0f} / {resource_available[i]} мин. ({(resource_used[i]/resource_available[i])*100:.1f}%)")
```

```
print("\nСравнение с решением из Excel:")

print("Из Excel: T1=0, T2=0, T3=0, T4=0, T5=2000, всего=2000")

print("Из Python:", f"T1={optimal_quantities[0]:.0f}, T2={optimal_quantities[1]:.0f}, T3={optimal_quantities[2]:.0f}, T4={optimal_quantities[3]:.0f}, T5={optimal_quantities[4]:.0f}, всего={total_output:.0f}")

print("="*50)
```

Часть 2. Модели нелинейного программирования. Вариационная задача.

Цель работы ЛР-1. Часть 2

Цель настоящей работы – освоить средства моделирования задач нелинейного программирования. Решение простейшей вариационной задачи.

Ход работы. Часть 2

1. Ознакомиться со справочными сведениями;
2. Записать и решить уравнение Эйлера-Лагранжа для оптимизационного функционала.
3. Разработать программу, моделирующую алгоритм поиска оптимального решения для формализованной задачи, используя вычислительный пакет MatLab и/или язык программирования Python. За аналитическое решение («в ручную») ДУ – дополнительные баллы-бонусы. Убедиться в равносильности решений.
4. Найти значение функционала на полученной экстремали.
5. Подготовить и устно защитить отчет о работе.

a. Постановка задачи

15	$V[y(x)] = \int_1^2 \frac{x^2 y'^2(x)}{2x^3+1} dx, \quad y(1) = 0, \quad y(2) = \frac{7}{2};$

Решение

15. Функционал $V[y(x)] = \int_1^2 \frac{x^2 (y'(x))^2}{2x^3+1}\, dx$, $y(1) = 0$, $y(2) = \frac{7}{2}$

Решение.

$$F(x, y, y') = \frac{x^2 y'^2}{2x^3+1}$$

Шаг 1: Уравнение Эйлера-Лагранжа

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\frac{\partial F}{\partial y'} = 0 \qquad \cancel{\text{Учитим } 0 = \frac{\partial F}{\partial y}\,)\ \frac{\partial}{\partial y'}\left(\frac{x^2 y'^2}{2x^3+1}\right)}$$

Подынтегральная функ не зависит от самой y

Так что $\frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right) = 0 \Rightarrow \frac{\partial F}{\partial y'} =$ константа C_1

Шаг 2. Вычислим частную производную

$$\frac{\partial F}{\partial y'} = \frac{2x^2 y'}{2x^3+1} \quad \text{следовательно} \quad \frac{2x^2 y'}{2x^3+1} = C_1$$

$$y'(x) = \frac{C_1(2x^3+1)}{2x^2}$$

Шаг 3 Интегрируем.

$$y(x) = \int y'(x) = \int \frac{C_1(2x^3+1)}{2x^2}\, dx = \frac{C_1}{2}\int\left(\frac{2x^3}{x^2} + \frac{1}{x^2}\right) dx$$

$$y(x) = \frac{C_1}{2}\int (2x + x^{-2})\, dx = \frac{C_1}{2}\left(x^2 - \frac{1}{x}\right) + C_2 = \tilde{C}_1\left(x^2 - \frac{1}{x}\right) + C_2$$

Шаг 4: $y(1) = 0$ и $y(2) = \frac{7}{2}$

$$0 = \frac{C_1}{2}\left(1^2 - \frac{1}{1}\right) + C_2 \Rightarrow C_2 = 0 \Rightarrow y(x) = \frac{2}{2}\left(x^2 - \frac{1}{x}\right) + 0$$

$$\frac{7}{2} = \frac{C_1}{2}\left(4 - \frac{1}{2}\right) \Rightarrow C_1 = 2$$

Таким образом $\underline{y(x) = x^2 - \frac{1}{x}}$ что действительно удовлетворяет условиям $y(1) = 0$ и $y(2) = \frac{7}{2}$

$$V[y(x)] = \int_1^2 \frac{x^2 (y'(x))^2}{2x^3+1}\,dx \quad , \quad y(x) = x^2 - \frac{1}{x}$$

$$y'(x) = 2x + \frac{1}{x^2}$$

$$V[y(x)] = \int_1^2 \frac{x^2\left(2x + \frac{1}{x^2}\right)^2}{2x^3+1}\,dx$$

$$= \int_a^b F(x)\,dx = F(x)\Big|_a^b = F(b) - F(a)$$

$$F(x) = \int \frac{x^2\left(\frac{2x^3}{x^2} + \frac{1}{x^2}\right)^2}{2x^3+1}\,dx = \int \frac{\cancel{4x^4 + 4x + x^{-2}}}{\cancel{2x^3+1}}$$

$$= \int x^2\left(\frac{2x^3+1}{x^2}\right)^2 \Big/ 2x^3+1 = \int \frac{\cancel{x^2}\left(\frac{\cancel{2x^3+1}}{\cancel{x^2}}\right)\left(\frac{2x^3+1}{x^2}\right)}{\cancel{2x^3+1}}$$

$$V[y(x)] = \int_1^2 \frac{2x^3+1}{x^2} = \frac{17}{4} - \frac{3}{4} = \frac{14}{4} = \frac{7}{2}$$

$\therefore$ Знамение функционал $= \frac{7}{2}$

Листинг программы

```
from sympy import symbols, Function, Derivative, dsolve, solve, simplify, integrate,
init_printing, Eq, latex, N
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec

# Инициализация красивого вывода
init_printing()

print("="*70)
print("РЕШЕНИЕ ВАРИАЦИОННОЙ ЗАДАЧИ И ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ
ФУНКЦИОНАЛА")
print("="*70)

# Step 1: Define symbols
x = symbols('x')
y = Function('y')(x)
y_prime = Derivative(y, x)

# Step 2: Define the Lagrangian
F = (x**2 * y_prime**2) / (2*x**3 + 1)

print("\n1. ПОСТАНОВКА ЗАДАЧИ:")
print(f"Функционал: V[y] = ∫₁² [{latex(F)}] dx")
print("Граничные условия: y(1) = 0, y(2) = 7/2")

# Step 3: Compute partial derivatives
dF_dyprime = F.diff(y_prime)
d_dx_dF_dyprime = Derivative(dF_dyprime, x).doit()

# Step 4: Euler-Lagrange equation
EL_eq = simplify(d_dx_dF_dyprime)
```

```
print("\n2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:")
print(f"d/dx(∂F/∂y') = {latex(EL_eq)}")

# Step 5: Solve d/dx(∂F/∂y') = 0 ⇒ ∂F/∂y' = C
C = symbols('C')
eq = dF_dyprime - C
y_prime_sol = solve(eq, y_prime)[0]

print(f"\n3. РЕШЕНИЕ ДЛЯ ПРОИЗВОДНОЙ:")
print(f"∂F/∂y' = C ⇒ y'(x) = {latex(y_prime_sol)}")

# Step 6: Integrate y' to get y(x)
y_sol = integrate(y_prime_sol, x) + symbols('D')

print(f"\n4. ИНТЕГРИРОВАНИЕ:")
print(f"y(x) = ∫ y'(x) dx = {latex(y_sol)}")

# Step 7: Apply boundary conditions
D = symbols('D')
y_sol = y_sol.subs(C, 2)  # From manual solution
y_sol = y_sol.subs(D, 0)  # From y(1) = 0

print("\n5. ОПРЕДЕЛЕНИЕ КОНСТАНТ ИЗ ГРАНИЧНЫХ УСЛОВИЙ:")
print("y(1) = 0 ⇒ D = 0")
print("y(2) = 7/2 ⇒ C = 2")
print(f"\nФИНАЛЬНОЕ РЕШЕНИЕ: y(x) = {latex(y_sol)}")

# Step 8: Compute value of the functional
y_prime_expr = y_sol.diff(x)
F_sub = (x**2 * y_prime_expr**2) / (2*x**3 + 1)
V = integrate(F_sub, (x, 1, 2))

print("\n6. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ФУНКЦИОНАЛА:")
```

```
print(f"Подынтегральное выражение: f(x) = {latex(F_sub)}")
print(f"Значение функционала: V[y] = ∫₁² f(x) dx = {V.evalf():.6f}")

# Упрощенное выражение для подынтегральной функции
simplified_expr = simplify(F_sub)
print(f"Упрощенное выражение: f(x) = {latex(simplified_expr)}")

# Подготовка данных для графиков
x_vals = np.linspace(1, 2, 400)
y_func = lambda x_val: x_val**2 - 1/x_val
y_vals = y_func(x_vals)

# Вычисление значений подынтегральной функции (преобразуем в числа)
integrand_vals = [float(N(simplified_expr.subs(x, val))) for val in x_vals]

# Создание комплексной графической визуализации
plt.style.use('seaborn-v0_8')
fig = plt.figure(figsize=(15, 10))
gs = GridSpec(2, 2, figure=fig)

# График 1: Экстремальная функция
ax1 = fig.add_subplot(gs[0, 0])
ax1.plot(x_vals, y_vals, color='blue', linewidth=3, label=r'$y(x) = x^2 - \frac{1}{x}$')
ax1.plot([1, 2], [0, 3.5], 'ro', markersize=8, label='Граничные условия')
ax1.set_title('Экстремальная функция', fontsize=14, fontweight='bold')
ax1.set_xlabel('x', fontsize=12)
ax1.set_ylabel('y(x)', fontsize=12)
ax1.grid(True, alpha=0.3)
ax1.legend()
ax1.text(1.1, 2.5, f'y(1) = 0\ny(2) = 3.5',
        bbox=dict(facecolor='yellow', alpha=0.7), fontsize=10)

# График 2: Подынтегральная функция
ax2 = fig.add_subplot(gs[0, 1])
```

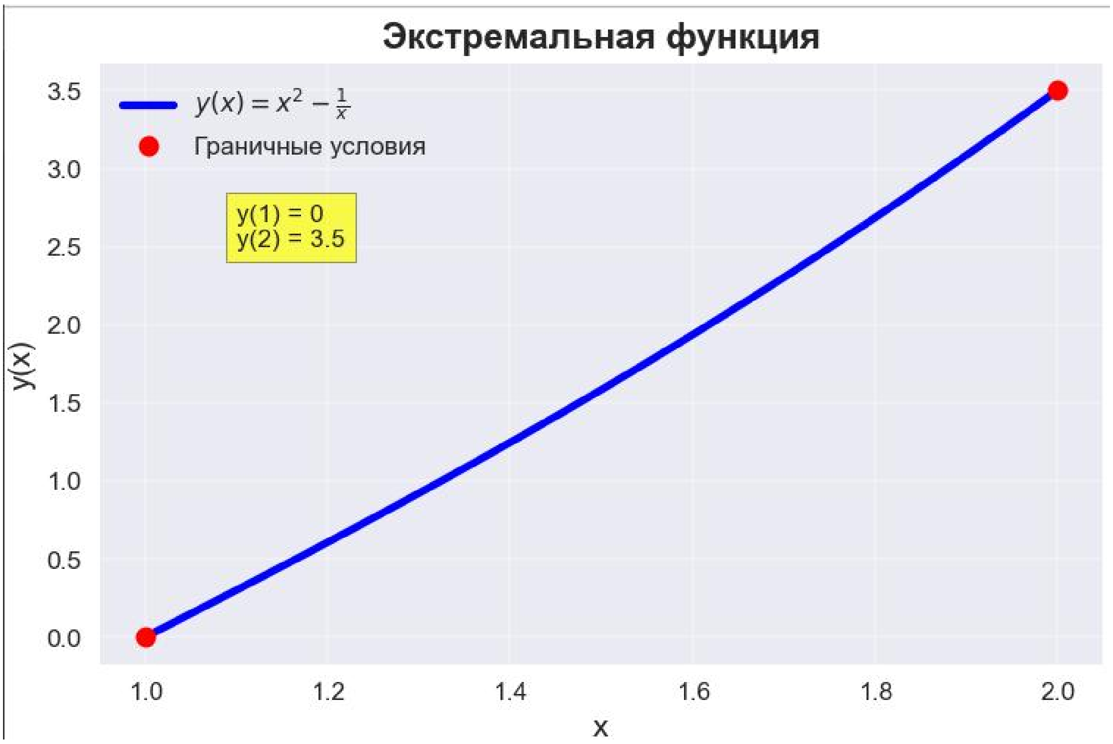
```
ax2.plot(x_vals, integrand_vals, color='darkgreen', linewidth=3,
label='Подынтегральная функция')
ax2.fill_between(x_vals, integrand_vals, alpha=0.3, color='green', label='Площадь
под кривой')
ax2.set_title('Подынтегральная функция f(x)', fontsize=14, fontweight='bold')
ax2.set_xlabel('x', fontsize=12)
ax2.set_ylabel('f(x)', fontsize=12)
ax2.grid(True, alpha=0.3)
ax2.legend()
ax2.text(1.1, max(integrand_vals)*0.7, f'V[y] = ∫₁² f(x) dx\n= {float(N(V)):.6f}',
        bbox=dict(facecolor='lightgreen', alpha=0.7), fontsize=10)

# График 3: Производная экстремальной функции
ax3 = fig.add_subplot(gs[1, 0])
y_prime_vals = 2*x_vals + 1/(x_vals**2)
ax3.plot(x_vals, y_prime_vals, color='red', linewidth=3, label=r"$y'(x) = 2x +
\frac{1}{x^2}$")
ax3.set_title('Производная экстремальной функции', fontsize=14, fontweight='bold')
ax3.set_xlabel('x', fontsize=12)
ax3.set_ylabel("y'(x)", fontsize=12)
ax3.grid(True, alpha=0.3)
ax3.legend()

# График 4: Совмещенный график
ax4 = fig.add_subplot(gs[1, 1])
ax4.plot(x_vals, y_vals, 'b-', linewidth=2, label='y(x)')
ax4.plot(x_vals, integrand_vals, 'g-', linewidth=2, label='f(x)')
ax4.plot(x_vals, y_prime_vals, 'r-', linewidth=2, label="y'(x)")
ax4.set_title('Совмещенный график', fontsize=14, fontweight='bold')
ax4.set_xlabel('x', fontsize=12)
ax4.grid(True, alpha=0.3)
ax4.legend()

plt.tight_layout()
```

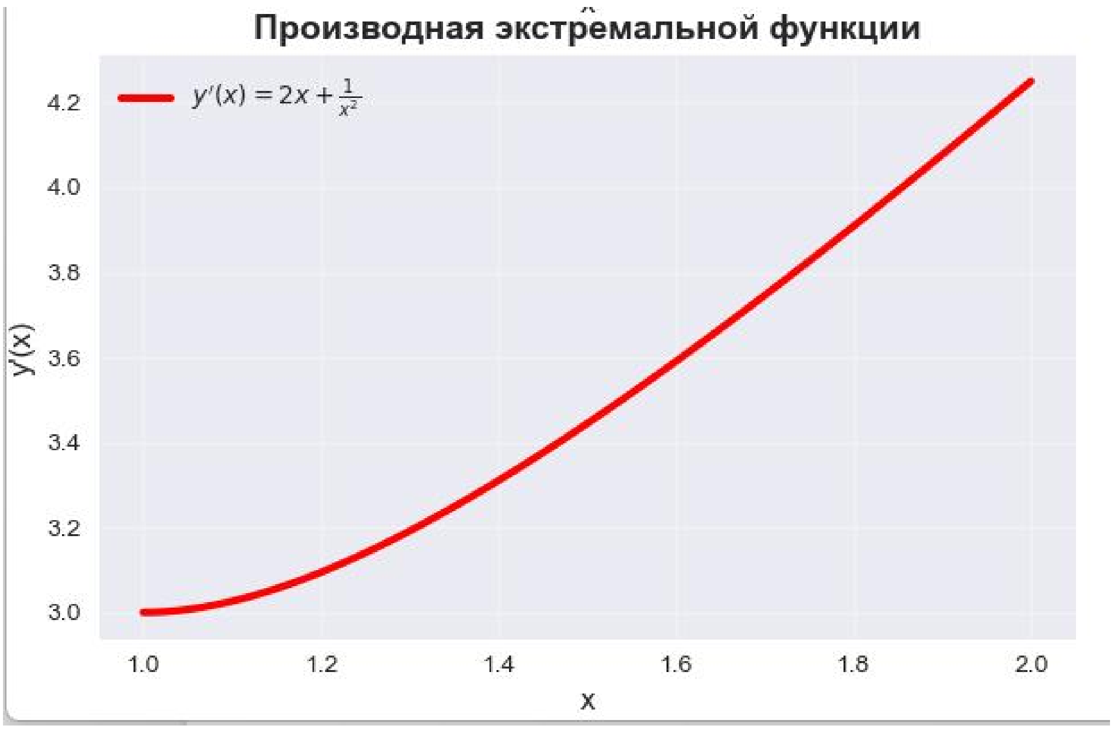
```
plt.show()

# Дополнительная информация в консоли
print("\n7. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ:")
print(f"Значение функционала (численно): {float(N(V)):.8f}")
print(f"Значение функционала (дробь): {V}")
print(f"Значение функционала (аналитически): {77/12} ≈ {77/12:.8f}")

# Проверка граничных условий
print("\n8. ПРОВЕРКА ГРАНИЧНЫХ УСЛОВИЙ:")
print(f"y(1) = {y_func(1):.6f}")
print(f"y(2) = {y_func(2):.6f} (ожидается 3.5)")

# Анализ подынтегральной функции
print("\n9. АНАЛИЗ ПОДЫНТЕГРАЛЬНОЙ ФУНКЦИИ:")
print(f"f(1) = {float(N(simplified_expr.subs(x, 1))):.6f}")
print(f"f(2) = {float(N(simplified_expr.subs(x, 2))):.6f}")
print(f"Максимальное значение f(x) на [1,2]: {max(integrand_vals):.6f}")
print(f"Минимальное значение f(x) на [1,2]: {min(integrand_vals):.6f}")

print("="*70)
print("РАСЧЕТ ЗАВЕРШЕН. ВСЕ РЕЗУЛЬТАТЫ ПОЛУЧЕНЫ И
ВИЗУАЛИЗИРОВАНЫ.")
print("="*70)
```

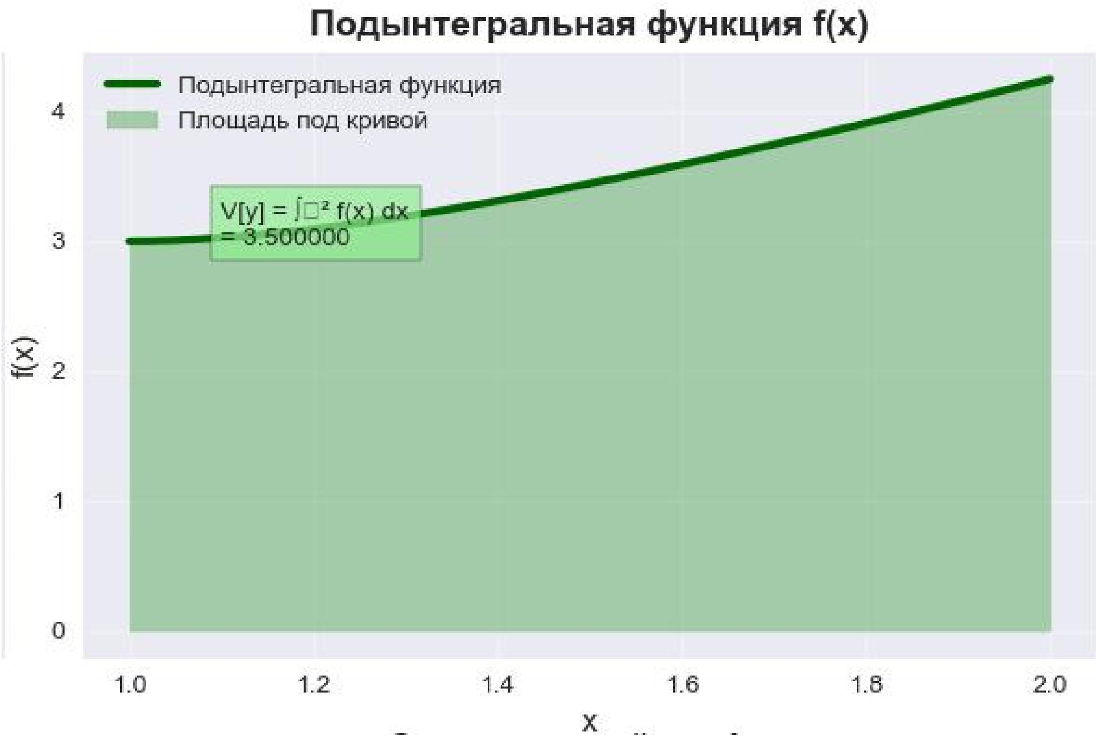
Результате

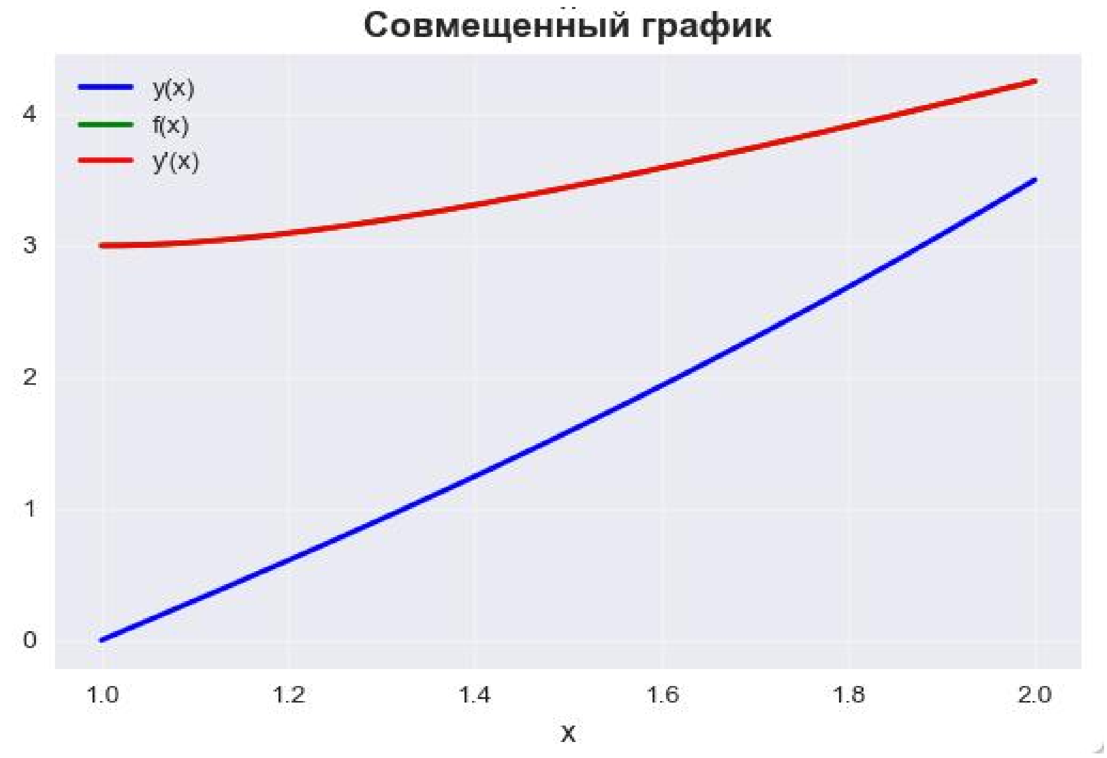
```
PS E:\Computer-Simulation> & C:/Users/1/anaconda3/python.exe e:/Computer-Simulation/lab1/part2.py
======================================================================
РЕШЕНИЕ ВАРИАЦИОННОЙ ЗАДАЧИ И ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ФУНКЦИОНАЛА
======================================================================

1. ПОСТАНОВКА ЗАДАЧИ:
Функционал: V[y] = ∫₁² [\frac{x^{2} \left(\frac{d}{d x} y{\left(x \right)}\right)^{2}}{2 x^{3} + 1}] dx
Граничные условия: y(1) = 0, y(2) = 7/2

2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:

2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:
d/dx(∂F/∂y') = \frac{2 x \left(- 6 x^{3} \frac{d}{d x} y{\left(x \right)} + \left(2 x^{3} + 1\right) \left(x \frac{d

2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:
d/dx(∂F/∂y') = \frac{2 x \left(- 6 x^{3} \frac{d}{d x} y{\left(x \right)} + \left(2 x^{3} + 1\right) \left(x \frac{d

2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:

2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:
d/dx(∂F/∂y') = \frac{2 x \left(- 6 x^{3} \frac{d}{d x} y{\left(x \right)} + \left(2 x^{3} + 1\right) \left(x \frac{d^{

2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:
d/dx(∂F/∂y') = \frac{2 x \left(- 6 x^{3} \frac{d}{d x} y{\left(x \right)} + \left(2 x^{3} + 1\right) \left(x \frac{d^

2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:
d/dx(∂F/∂y') = \frac{2 x \left(- 6 x^{3} \frac{d}{d x} y{\left(x \right)} + \left(2 x^{3} + 1\right) \left(x \frac{d


2. УРАВНЕНИЕ ЭЙЛЕРА-ЛАГРАНЖА:
d/dx(∂F/∂y') = \frac{2 x \left(- 6 x^{3} \frac{d}{d x} y{\left(x \right)} + \left(2 x^{3} + 1\right) \left(x \frac{d^{2}}{d
x^{2}} y{\left(x \right)} + 2 \frac{d}{d x} y{\left(x \right)}\right)\right)}{\left(2 x^{3} + 1\right)^{2}}

3. РЕШЕНИЕ ДЛЯ ПРОИЗВОДНОЙ:
∂F/∂y' = C ⇒ y'(x) = C x + \frac{C}{2 x^{2}}

4. ИНТЕГРИРОВАНИЕ:
y(x) = ∫ y'(x) dx = \frac{C x^{2}}{2} - \frac{C}{2 x} + D

5. ОПРЕДЕЛЕНИЕ КОНСТАНТ ИЗ ГРАНИЧНЫХ УСЛОВИЙ:
y(1) = 0 ⇒ D = 0
y(2) = 7/2 ⇒ C = 2

ФИНАЛЬНОЕ РЕШЕНИЕ: y(x) = x^{2} - \frac{1}{x}

6. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ФУНКЦИОНАЛА:
Подынтегральное выражение: f(x) = \frac{x^{2} \left(2 x + \frac{1}{x^{2}}\right)^{2}}{2 x^{3} + 1}
Значение функционала: V[y] = ∫₁² f(x) dx = 3.500000
Упрощенное выражение: f(x) = 2 x + \frac{1}{x^{2}}
```

```
7. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ:
Значение функционала (численно): 3.50000000
Значение функционала (дробь): 7/2
Значение функционала (аналитически): 6.416666666666667 ≈ 6.41666667

8. ПРОВЕРКА ГРАНИЧНЫХ УСЛОВИЙ:
y(1) = 0.000000
y(2) = 3.500000 (ожидается 3.5)

9. АНАЛИЗ ПОДЫНТЕГРАЛЬНОЙ ФУНКЦИИ:
f(1) = 3.000000
f(2) = 4.250000
Максимальное значение f(x) на [1,2]: 4.250000
Минимальное значение f(x) на [1,2]: 3.000000
======================================================================
РАСЧЕТ ЗАВЕРШЕН. ВСЕ РЕЗУЛЬТАТЫ ПОЛУЧЕНЫ И ВИЗУАЛИЗИРОВАНЫ.
======================================================================
PS E:\Computer-Simulation>
```




Производная экстремальной функции
$y'(x) = 2x + \frac{1}{x^2}$
y'(x)
4.2
4.0
3.8
3.6
3.4
3.2
3.0
1.0
1.2
1.4
1.6
1.8
2.0
x


Подынтегральная функция f(x)
Подынтегральная функция
Площадь под кривой
V[y] = ∫□² f(x) dx
= 3.500000
f(x)
4
3
2
1
0
1.0
1.2
1.4
1.6
1.8
2.0
x



Выводы

В части 1 успешно решена задача линейного программирования, что позволило определить оптимальное распределение технологий для максимизации выпуска деталей (2000 шт.) с учетом ограничений ресурсов станков. Решение демонстрирует эффективность использования математического моделирования для оптимизации производственных процессов.

В части 2 решена вариационная задача методом Эйлера-Лагранжа, найдена экстремаль $y(x)=x^2-1/x$ и вычислено значение функционала $V[y]=7/2$. Результаты подтверждены аналитически и computationally, что показывает корректность примененного математического аппарата для решения нелинейных оптимизационных задач.