ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Профессор		С.И. Колесникова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

Нелинейное программирование. Вариационный принцип.

по дисциплине: Компьютерное моделирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. 4134к		Костяков Н.А.
	подпись, дата	инициалы, фамилия

Санкт-Петербург
2024

Цель работы.

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

Постановка задачи

Вариант 4

На складах w1, w2, w3 хранятся соответственно 15, 25, 20 кроватей, должны быть распределены по четырем магазинам m1, m2, m3, m4, где требуется 20, 12, 5 и 9 кроватей. Пусть стоимость перевозки одной кровати со склада в магазин задается следующей таблицей в условных единицах:

Склад	Магазин			
	M1	M2	M3	M4
W1	2	2	2	4
W2	3	1	1	3
W3	3	6	3	4

Как следует планировать перевозку для минимизации стоимости?

Формализованная постановка задачи

Обозначим переменные (x{ij}) как количество кроватей, которые нужно перевезти с склада (Wi) в магазин (Mj).

1. (x{11}): количество кроватей из склада (W1) в магазин (M1).
2. (x{12}): количество кроватей из склада (W1) в магазин (M2).
3. И так далее для всех комбинаций (i) и (j) (где (i \in {1, 2, 3}) и (j \in {1, 2, 3, 4})).

Целевая функция

Цель — минимизировать общую стоимость транспортировки, которая выражается как сумма произведений количества кроватей и стоимости транспортировки для каждой пары склад-магазин:

[\text{Minimize } Z = 2x_{11} + 2x_{12} + 2x_{13} + 4x_{14} + 3x_{21} + 1x_{22} + 1x_{23} + 3x_{24} + 3x_{31} + 6x_{32} + 3x_{33} + 4x_{34}]

Ограничения

1. **Ограничения на потребности магазинов**:

2. (x_{11} + x_{21} + x_{31} = 20) (для (M1))
3. (x_{12} + x_{22} + x_{32} = 12) (для (M2))
4. (x_{13} + x_{23} + x_{33} = 5) (для (M3))
5. (x_{14} + x_{24} + x_{34} = 9) (для (M4))

6. **Ограничения на запасы складов**:

7. (x{11} + x_{12} + x_{13} + x_{14} <= 15) (для (W1))
8. (x_{21} + x_{22} + x_{23} + x_{24} <= 25) (для (W2))
9. (x_{31} + x_{32} + x_{33} + x_{34} <= 20) (для (W3))
10. **Ограничения на неотрицательность**:

11. (x_{ij} >= 0) для всех (i) и (j).

Итоговая формулировка задачи

Мы должны минимизировать функцию стоимости (Z), удовлетворяя ограничениям по потребностям магазинов и доступным запасам на складах.

Скриншоты решения Excell

Я использовал LibreOffice Calc так как писал работу из под дистрибутива на linux

условие	m1	m2	m3	m4	на складе		целевая F(x)	2x11+2x12+2x13+4x13+3x21+x22+x23+3x24+3x31+6x32+3x33+4x34
w1	2	2	2	4	15			
w2	3	1	1	3	25			
w3	3	6	3	4	20			
требуется	20	12	5	9				

	m1	m2	m3	m4	всего	ограничение	предел
w1	15	0	0	0	15	<=	15
w2	0	12	5	8	25	<=	25
w3	5	0	0	1	6	<=	20
всего	20	12	5	9			
ограничение	>=	>=	>=	>=			
Предел	20	12	5	9			
Затраты	90						

Рисунок 1 – Результаты решения

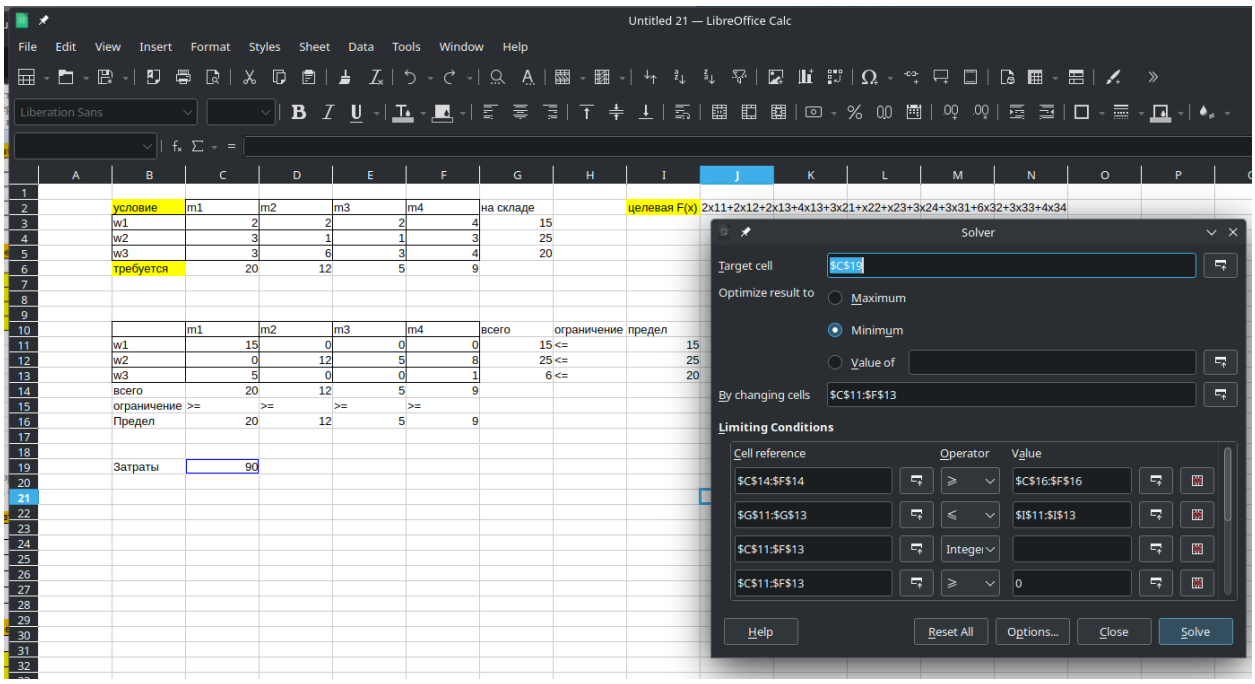


Рисунок 2 – настройка для решения задачи

Скриншоты работы программы на python

```
Оптимальное распределение кроватей:
Склад W1 -> Магазин M1: 15.0 кроватей
Склад W1 -> Магазин M2: 0.0 кроватей
Склад W1 -> Магазин M3: 0.0 кроватей
Склад W1 -> Магазин M4: 0.0 кроватей
Склад W2 -> Магазин M1: 0.0 кроватей
Склад W2 -> Магазин M2: 12.0 кроватей
Склад W2 -> Магазин M3: 5.0 кроватей
Склад W2 -> Магазин M4: 8.0 кроватей
Склад W3 -> Магазин M1: 5.0 кроватей
Склад W3 -> Магазин M2: 0.0 кроватей
Склад W3 -> Магазин M3: 0.0 кроватей
Склад W3 -> Магазин M4: 1.0 кроватей
Минимальная стоимость перевозки: 90.0
(.venv) kasi@kasi-kubu:~/Documents/GitHub/Vyzovskoe3-4/7 сем/Комп моделирование/лр1$
```

Листинг программы

```python
import pulp

supply = [15, 25, 20]  # w1, w2, w3
warehouses = ['W1', 'W2', 'W3']

demand = [20, 12, 5, 9]  # m1, m2, m3, m4
stores = ['M1', 'M2', 'M3', 'M4']
```

```
cost = {
    ('W1', 'M1'): 2, ('W1', 'M2'): 2, ('W1', 'M3'): 2, ('W1', 'M4'): 4,
    ('W2', 'M1'): 3, ('W2', 'M2'): 1, ('W2', 'M3'): 1, ('W2', 'M4'): 3,
    ('W3', 'M1'): 3, ('W3', 'M2'): 6, ('W3', 'M3'): 3, ('W3', 'M4'): 4,
}

problem = pulp.LpProblem("Transportation_Problem", pulp.LpMinimize)

x = pulp.LpVariable.dicts("x", [(w, s) for w in warehouses for s in stores],
lowBound=0, cat='Continuous')

problem += pulp.lpSum(cost[(w, s)] * x[(w, s)] for w in warehouses for s in
stores), "Total_Cost"

for i, w in enumerate(warehouses):
    problem += pulp.lpSum(x[(w, s)] for s in stores) <= supply[i],
f"Supply_Constraint_{w}"

for j, s in enumerate(stores):
    problem += pulp.lpSum(x[(w, s)] for w in warehouses) >= demand[j],
f"Demand_Constraint_{s}"

problem.solve()


print("Оптимальное распределение кроватей:")
for w in warehouses:
    for s in stores:
        print(f"Склад {w} -> Магазин {s}: {x[(w, s)].varValue} кроватей")
    print()

print("Минимальная стоимость перевозки:", pulp.value(problem.objective))
```

Часть 2 – решение вариационной задачи

Вариант 4

4	$V[y(x)]=\int_0^{\pi}\left((y'(x)+y(x))^2+2y(x)\sin x\right)dx, \quad y(0)=0, \quad y(\pi)=1;$

Скриншот решения на python

```
PS L:\Vyzovskoe3-4\7 сем\КомпМод\лр1> python .\lag.py
y(x)**2 + 2*y(x)*sin(x) + Derivative(y(x), x)
Eq(y(x), -sin(x))
PS L:\Vyzovskoe3-4\7 сем\КомпМод\лр1>
```

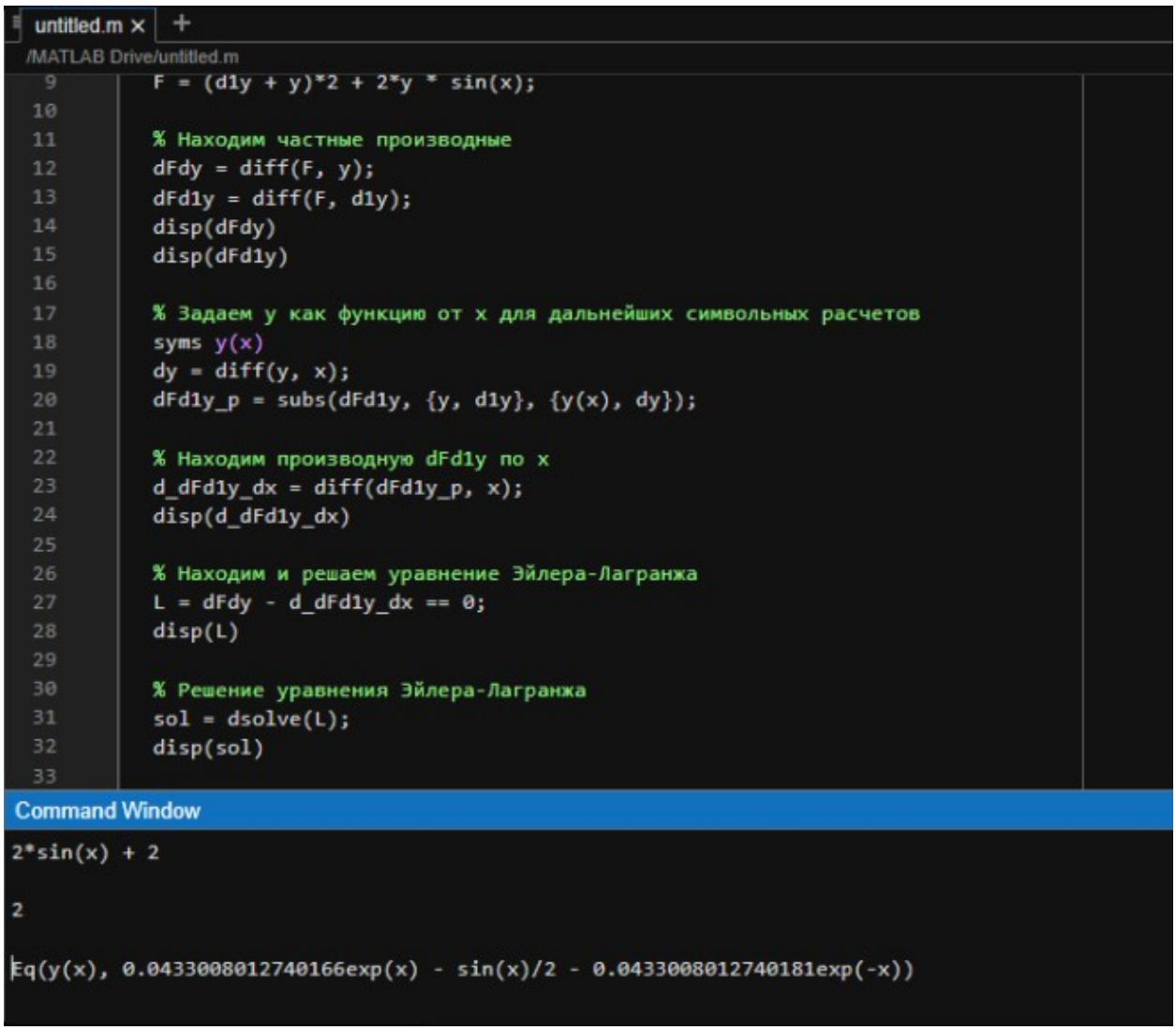
Листинг программы

```
# this code requires SymPy v1.4
# следующие две строки не нужны при запуске кода на live.sympy.org
from sympy import init_printing, sin
import numpy as np

init_printing()

from sympy import Symbol, Function, Derivative, dsolve, solve
x = Symbol('x')
y = Function('y')(x)
dy = Derivative(y)

F = (y-(1/2)*y**2)*sin(x)
F.doit() # выводим выражение в человекочитаемом формате ...
print(F) # ... и в машиночитаемом виде
dFdy = Derivative(F, y)
dFd1y = Derivative(F, dy)
dFdy.doit()
dFd1y.doit()
L = dFdy - Derivative(dFd1y, x)
sol = dsolve(L)
eq1 = sol.subs({x:np.pi/4, y:-np.log(np.sqrt(2))})
eq2 = sol.subs({x:np.pi/2, y:0})
coeffs = solve([eq1, eq2])
res = sol.subs(coeffs)
print(res.doit())
```



Результат на матлабе, сошлось