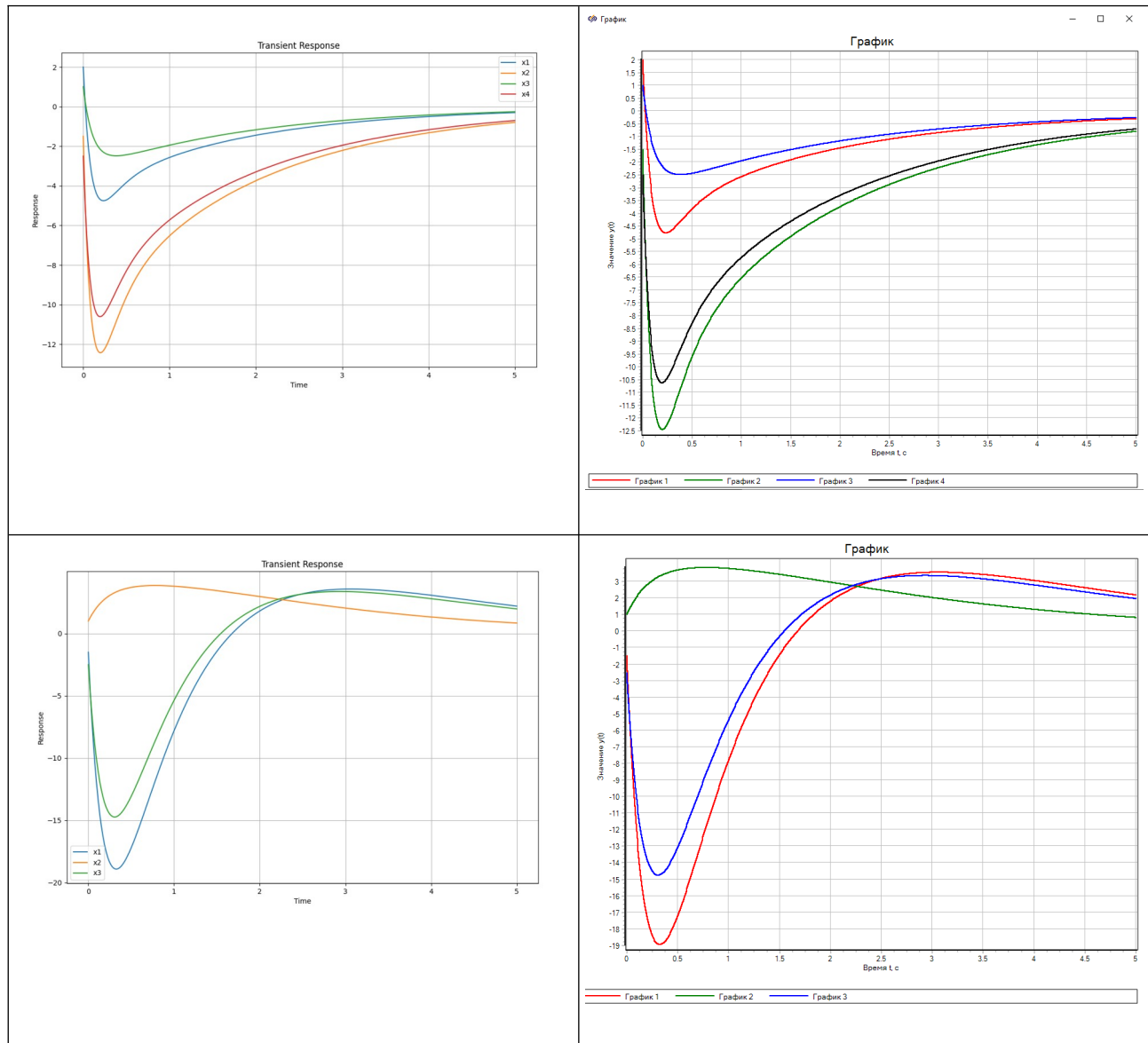
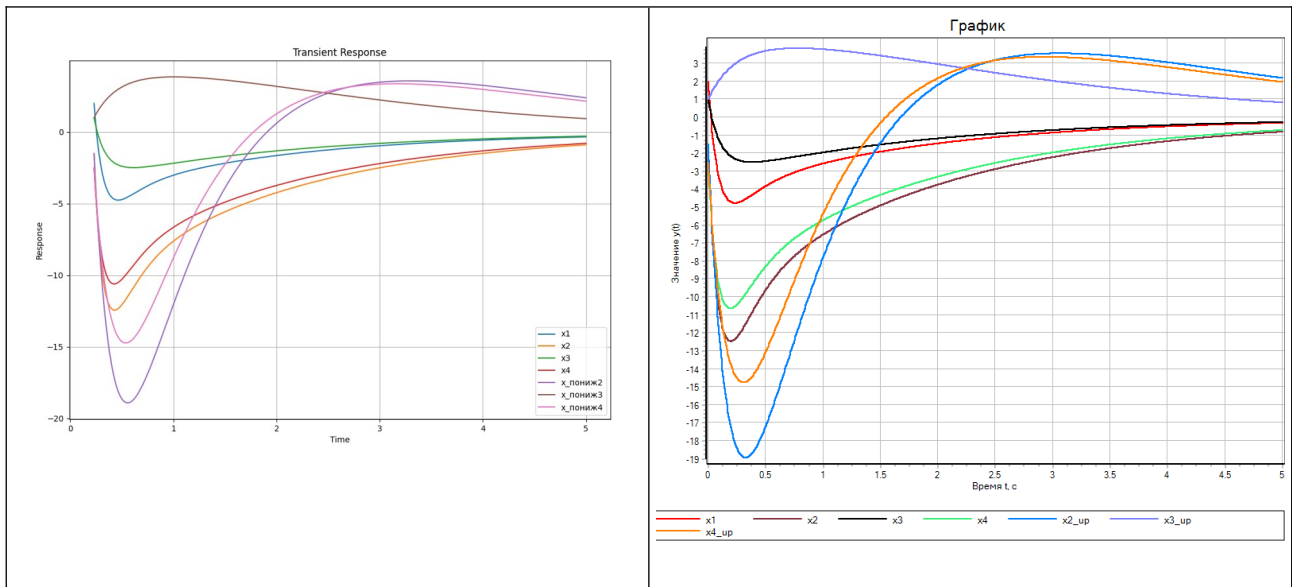


Ниже представлено сравнение графиков переходных процессов из SimInTech и из программы составленной мною.

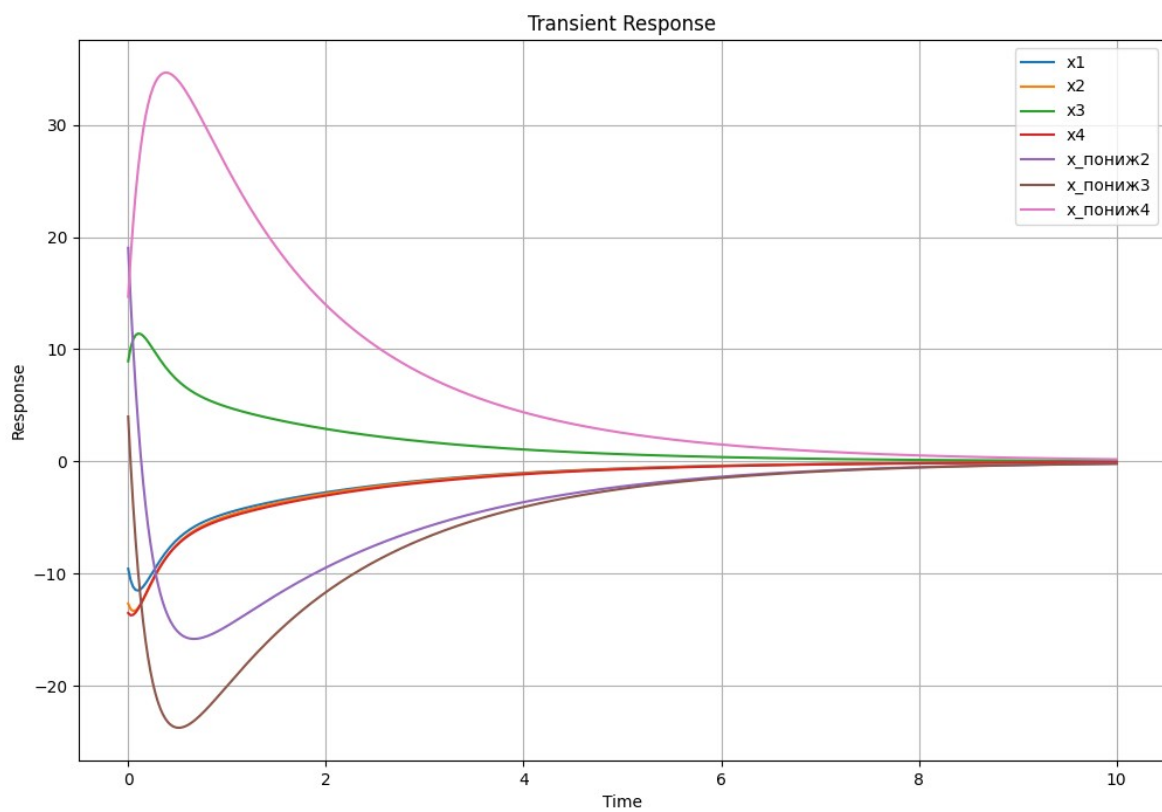
Из сравнения этих графиков можно сделать вывод, что программа работает корректно.





Далее рассмотрим графики решения системы, которые были построены по формуле

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & C_4 \end{pmatrix} (\vec{\alpha}_1 \vec{\alpha}_2 \vec{\alpha}_3 \vec{\alpha}_4) * \begin{pmatrix} e^{\lambda_1} \\ e^{\lambda_2} \\ e^{\lambda_3} \\ e^{\lambda_4} \end{pmatrix} = C_1 * \vec{\alpha}^1 * e^{\lambda_1} + C_2 * \vec{\alpha}^2 * e^{\lambda_2} + C_3 * \vec{\alpha}^3 * e^{\lambda_3} + C_4 * \vec{\alpha}^4 * e^{\lambda_4}$$



В статье «МЕТОДИКА ДЕКОМПОЗИЦИИ МОДЕЛЕЙ МНОГОТЕМПОВЫХ ДИНАМИЧЕСКИХ СИСТЕМ» было указано, как посчитать левую и правую границу интервалов голономных связей

$$\Gamma_v = \frac{-1}{\alpha_v} \ln(\delta^0)$$

Рассчитаем  $\Gamma_1$  и  $\Gamma_4$ . Пусть  $\delta^0 = 0.3$ , тогда:

$$\Gamma_1 = \frac{1}{-10} * \ln(0.3) = 0.120397$$

$$\Gamma_4 = \frac{1}{-0.5} * \ln(0.3) = 2.40795$$

Для упрощенной системы общий интервал справедливости  $t \in [\Gamma_1; \Gamma_n]$

В коде это реализовано так

```
err = 0.3 # ошибка для расчета границ интервалов

def calc_gran(self): # Функция для расчета границ, без «-» потому что в
self.eigenvalues[i] уже лежат отрицательные числа
    for i in range(self.size):
        self.gran.append((1 / self.eigenvalues[i]) * m.log(self.err))

values = np.array([
    [-3.0504, -44.6371, -14.7046, 56.9748],
    [-1.4677, -89.0323, -25.8790, 110.2661],
    [-3.4798, -10.1452, -6.7681, 15.2601],
    [-1.2984, -66.4516, -19.4315, 82.3508]
])
initial_conditions = np.array([[2.0000, -1.5000, 1.0000, -2.5000]])
# Создание матрицы
a = Matrix(4, values, initial_conditions, err)
a.calc()

# Создание новой матрицы без первой строки и первого столбца
new_values = a.matr_resh
b = Matrix(3, new_values, initial_conditions[:, 1:], err)
b.calc()
```

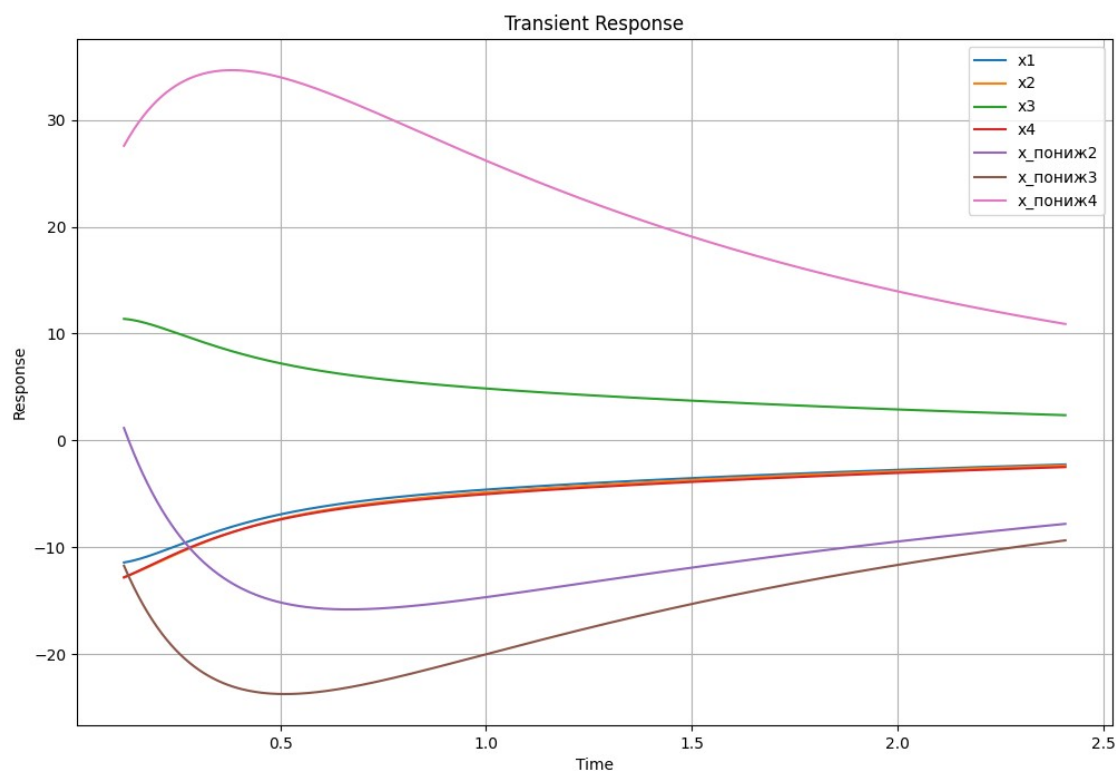
- значения границ  
0.12039416861751517  
0.24079229740064917  
1.2043844144863232

```
2.4077708501668678  
min(a.gran) = 0.12039416861751517
```

```
max(a.gran) = 2.4077708501668678
```

```
plot_transient_response2(a, b, min(a.gran), max(a.gran))
```

В результате получаем график решения систем, исходной и с пониженным порядком.



Если ввести эти временные интервалы в построение переходного процесса, получим следующее.

