

Modele systemów dynamicznych

Sprawozdanie z laboratorium 4

Igor Lis

Nr albumu: **284053**

Kierunek: **Inżynieria systemów**

MODEL WZROSTU LOGISTYCZNEGO

1. Wstęp teoretyczny

Celem niniejszego sprawozdania jest implementacja i analiza modelu wzrostu logistycznego populacji. Model ten stanowi klasyczne zagadnienie w ekologii matematycznej i opisuje, jak wielkość populacji zmienia się w czasie w środowisku o ograniczonych zasobach. W ramach pracy porównane zostanie rozwiązanie numeryczne, uzyskane za pomocą biblioteki scipy, z dokładnym rozwiązaniem analitycznym, wyprowadzonym przy użyciu pakietu sympy.

Model wzrostu logistycznego jest rozwinięciem prostszego modelu wykładniczego. Jego kluczowym założeniem jest istnienie tzw. pojemności środowiska (K), która reprezentuje maksymalną liczbę osobników, jaką dane środowisko jest w stanie utrzymać. W przeciwieństwie do nieograniczonego wzrostu wykładniczego, w modelu logistycznym tempo wzrostu populacji zwalnia w miarę zbliżania się do tej granicy, co prowadzi do charakterystycznej krzywej w kształcie litery "S".

Podstawowe wzory i definicje:

Podstawą modelu jest nieliniowe równanie różniczkowe zwyczajne pierwszego rzędu:

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{K}\right)$$

gdzie:

$N(t)$ – wielkość populacji w chwili czasu (t),

r – współczynnik wzrostu wewnętrznego, określający potencjalne tempo reprodukcji,

K – pojemność środowiska, czyli maksymalna wielkość populacji,

$\frac{dt}{dN}$ – chwilowa szybkość zmiany populacji.

Rozwiązaniem analitycznym powyższego równania, przy zadanym warunku początkowym $N(0)=N_0$:

$$N(t) = \frac{K}{1 + \left(\frac{K}{N_0} - 1\right) e^{-rt}}$$

gdzie:

N_0 – początkowa wielkość populacji w czasie $t = 0$.

W celu ilościowej oceny dokładności rozwiązania numerycznego (N_{num}) względem rozwiązania analitycznego (N_{dok}), wykorzystano dwie standardowe metryki błędów: średni błąd bezwzględny (MAE) oraz średni błąd kwadratowy (MSE). Wzory na te błędy przedstawiono poniżej:

$$MAE = \frac{1}{n} \sum_{i=1}^n |N_{num,i} - N_{dok,i}|$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (N_{num,i} - N_{dok,i})^2$$

gdzie:

n – liczba punktów pomiarowych w czasie,

$N_{num,i}$ – wartość populacji uzyskana z symulacji numerycznej w i -tym kroku,

$N_{dok,i}$ – wartość populacji obliczona ze wzoru analitycznego w i -tym kroku.

2. Opis rozwiązania

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
import sympy as sp
import os
import pandas as pd
import time # Dodajemy do pomiaru czasu

# --- Krok 1: Utworzenie folderu na wyniki ---
output_dir = "lista_4/wyniki"

# --- Krok 2: Funkcja do symulacji i rysowania ---
def run_and_plot_simulation(params, t_span=(0, 25), num_points=1000):
    """
    Uruchamia symulację dla danych parametrów, rysuje wykresy i zapisuje je do
    plików.
    """
    # Rozpakowanie parametrów
    r, K, N0, label = params['r'], params['K'], params['N0'], params['label']

    filename_safe_label = label.lower().replace(" ", "_").replace(":", "")
```

```

# Definicja równania dla SciPy
def logistic_growth_rhs(t, N, r_val, K_val):
    return r_val * N * (1 - N / K_val)

t_eval = np.linspace(*t_span, num_points)

# --- Rozwiązanie analityczne (SymPy) ---
t_sym, r_sym, K_sym, N0_sym = sp.symbols('t r K N0')
N_sym = sp.Function('N')
ode = sp.Eq(N_sym(t_sym).diff(t_sym), r_sym * N_sym(t_sym) * (1 - N_sym(t_sym)
/ K_sym))
exact_solution_expr = sp.dsolve(ode, ics={N_sym(0): N0_sym}).rhs

# Podstawienie wartości parametrów (r, K, N0) do wyrażenia
expr_with_params = exact_solution_expr.subs({r_sym: r, K_sym: K, N0_sym: N0})

start_time = time.time() # Pomiar czasu

N_exact_list = []
for t_val in t_eval:
    # Podstawiamy konkretną wartość czasu 't_val' do symbolu 't_sym'
    # i konwertujemy wynik (który jest obiektem SymPy) na typ float.
    value = float(expr_with_params.subs(t_sym, t_val))
    N_exact_list.append(value)

# Konwersja listy wyników z powrotem na tablicę NumPy
N_exact = np.array(N_exact_list)

end_time = time.time()
print(f" Czas obliczeń analitycznych: {end_time - start_time:.4f} s")

# --- Rozwiązanie numeryczne (SciPy) ---
sol = solve_ivp(
    fun=logistic_growth_rhs, t_span=t_span, y0=[N0], args=(r, K),
    t_eval=t_eval, rtol=1e-8
)
N_numeric = sol.y[0]

# --- Obliczenie błędów i wizualizacja ---
abs_error = np.abs(N_numeric - N_exact)
mae = np.mean(abs_error)
mse = np.mean((N_numeric - N_exact)**2)

# Wykres 1: Porównanie
fig1, ax1 = plt.subplots(figsize=(10, 6))
ax1.plot(t_eval, N_numeric, 'k-', label='Rozwiązanie numeryczne (SciPy)',
linewidth=4)
ax1.plot(t_eval, N_exact, 'r--', label='Rozwiązanie analityczne (SymPy)',
linewidth=2)
ax1.set_title(f"Model wzrostu logistycznego: {label}")
ax1.set_xlabel('Czas [t]')

```

```

ax1.set_ylabel('Wielkość populacji [N(t)]')
ax1.legend()
ax1.grid(True)
fig1.tight_layout()
fig1.savefig(os.path.join(output_dir, f"porownanie_{filename_safe_label}.png"))
plt.close(fig1)

# Wykres 2: Błąd
fig2, ax2 = plt.subplots(figsize=(10, 6))
ax2.plot(t_eval, abs_error, label='Błąd bezwzględny', color='blue')
ax2.set_title(f"Błąd bezwzględny dla: {label}")
ax2.set_xlabel('Czas [t]')
ax2.set_ylabel('Błąd bezwzględny')
ax2.grid(True)
ax2.set_yscale('log')
ax2.legend()
fig2.tight_layout()
fig2.savefig(os.path.join(output_dir, f"blad_{filename_safe_label}.png"))
plt.close(fig2)

return {"Opis": label, "MAE": mae, "MSE": mse}

# --- Scenariusze i uruchomienie ---
scenarios = [
    {'label': '1. Scenariusz bazowy', 'r': 0.5, 'K': 1000, 'N0': 50},
    {'label': '2. Szybki wzrost (r=2.0)', 'r': 2.0, 'K': 1000, 'N0': 50},
    {'label': '3. Spadek populacji (N0 > K)', 'r': 0.5, 'K': 1000, 'N0': 1500},
    {'label': '4. Niska pojemność środowiska (K=500)', 'r': 0.5, 'K': 500, 'N0': 50},
    {'label': '5. Start blisko granicy (N0=950)', 'r': 0.5, 'K': 1000, 'N0': 950}
]

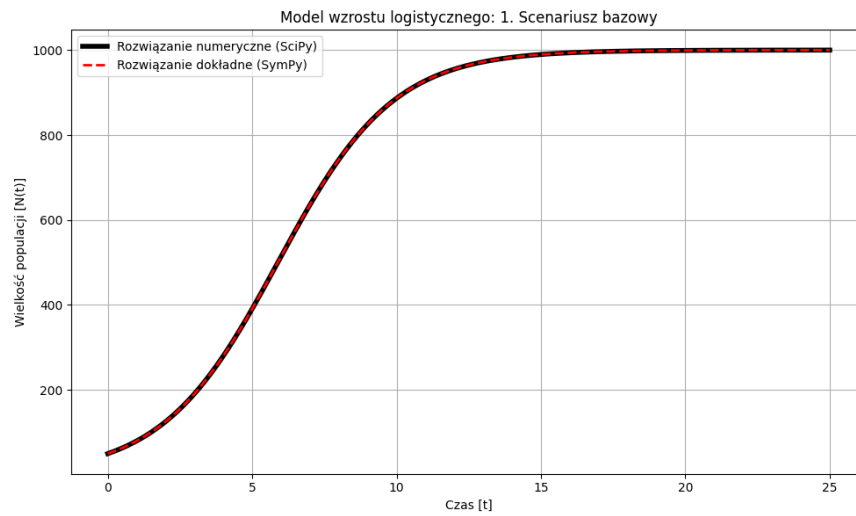
results = []
for params in scenarios:
    print(f"Uruchamiam symulację dla: '{params['label']}'...")
    result = run_and_plot_simulation(params)
    results.append(result)

df_results = pd.DataFrame(results)
print("\nPorównanie błędów dla różnych scenariuszy:")
print(df_results.to_string(index=False))

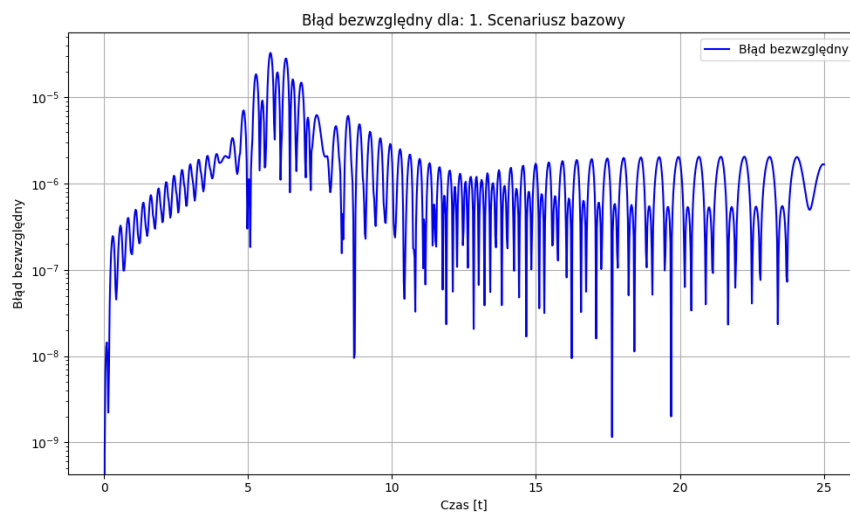
```

3. Wyniki obliczeń

3.1. Scenariusz bazowy ($r=0.5$, $K=1000$, $N_0=50$)

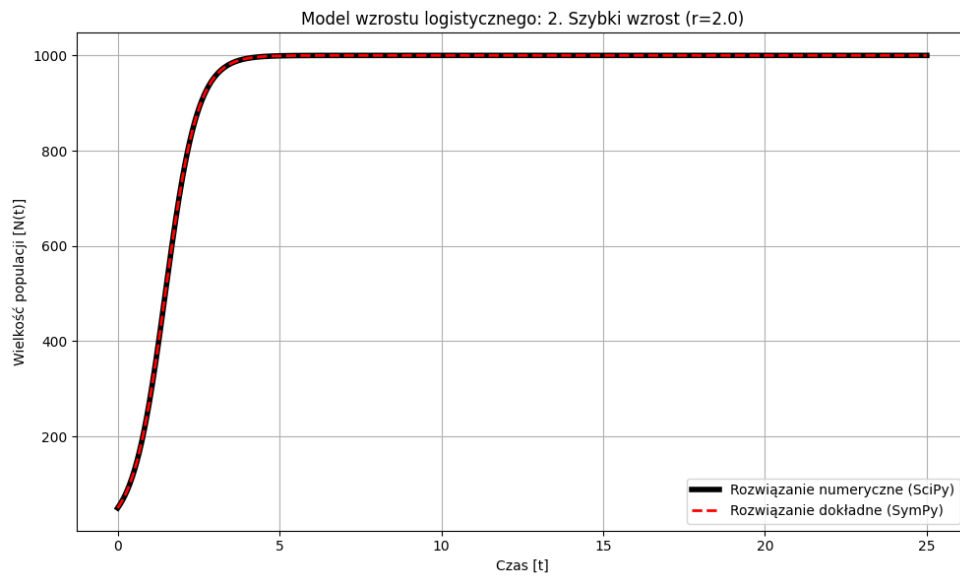


Rysunek 1. Porównanie trajektorii numerycznej i analitycznej (dokładnej) dla scenariusza bazowego.

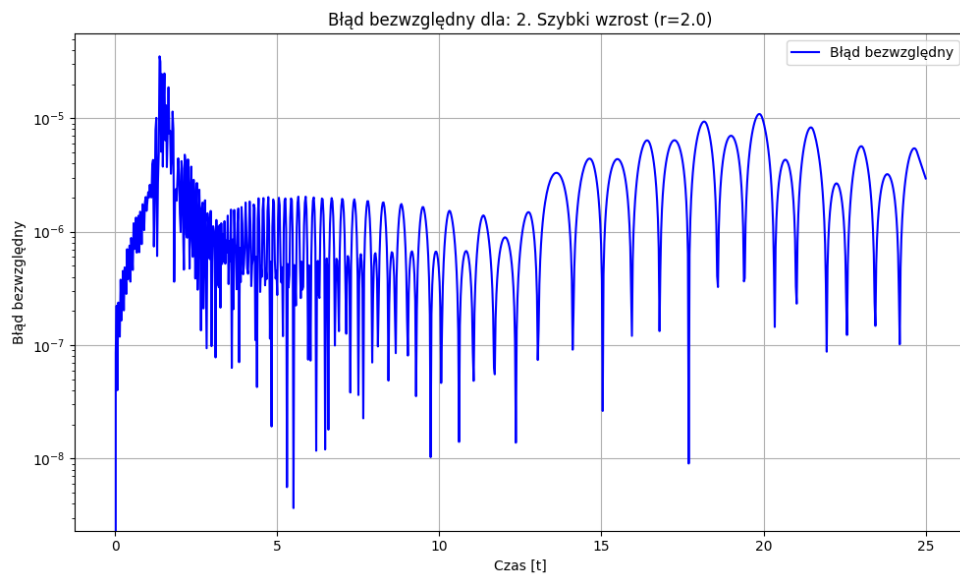


Rysunek 2. Wykres błędu bezwzględnego dla scenariusza bazowego.

3.2. Szybki wzrost ($r=2.0$):

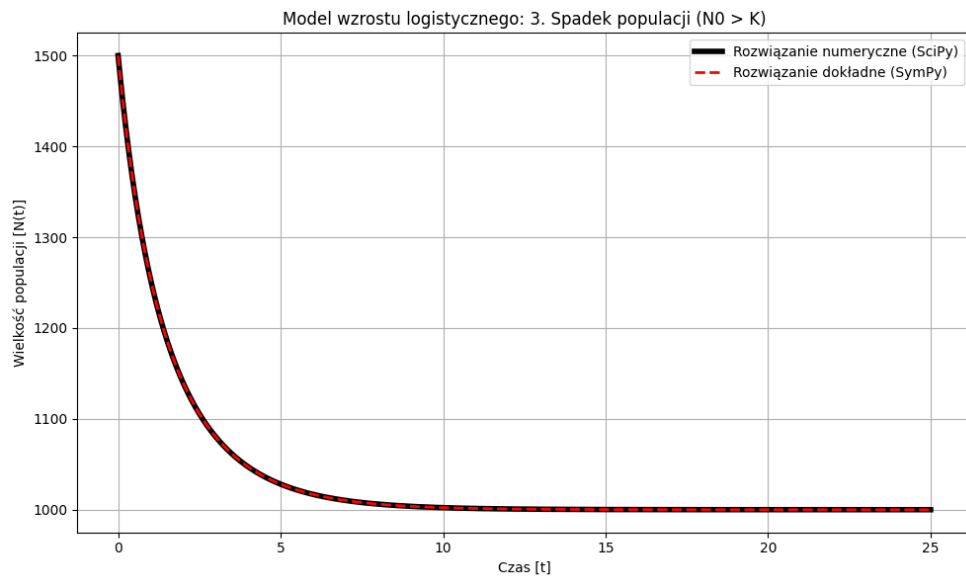


Rysunek 3. Porównanie trajektorii numerycznej i analitycznej (dokładnej) dla scenariusza z wysokim współczynnikiem wzrostu ($r=2.0$).

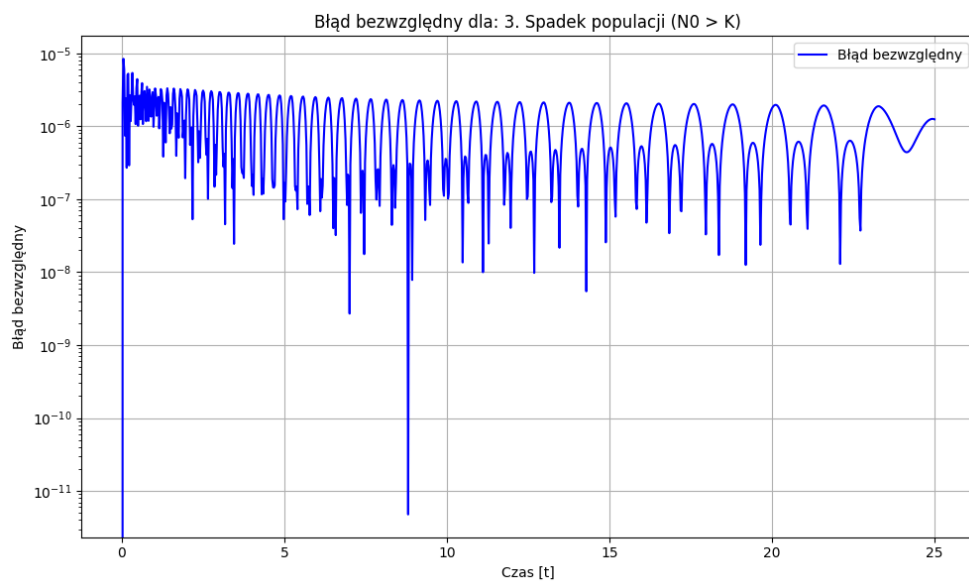


Rysunek 4. Wykres błędu bezwzględnego w funkcji czasu dla scenariusza z wysokim współczynnikiem wzrostu.

3.3. Spadek populacji ($N_0 > K$):

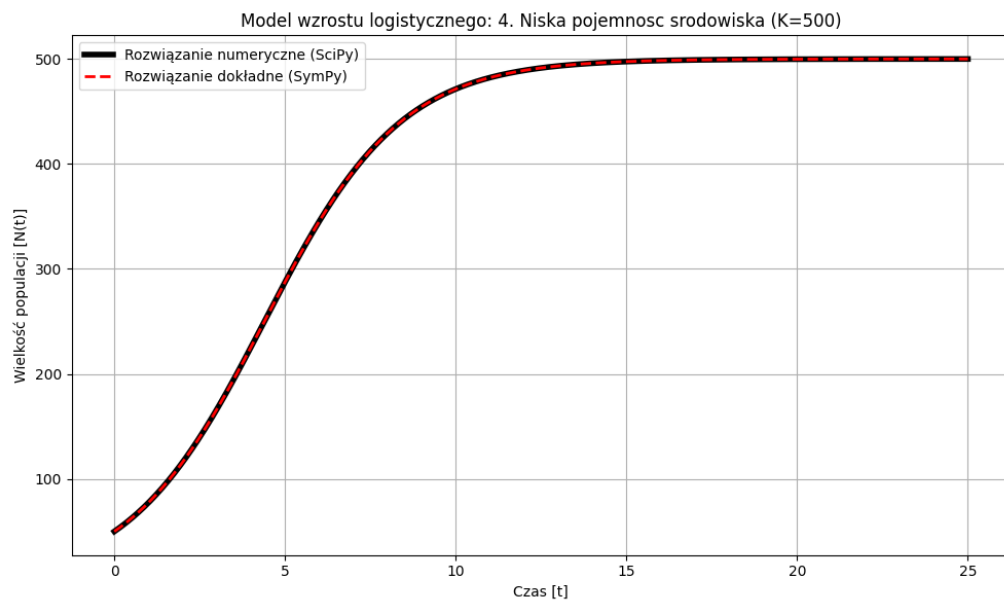


Rysunek 5. Porównanie trajektorii numerycznej i analitycznej (dokładnej) dla scenariusza spadku populacji ($N_0 > K$).

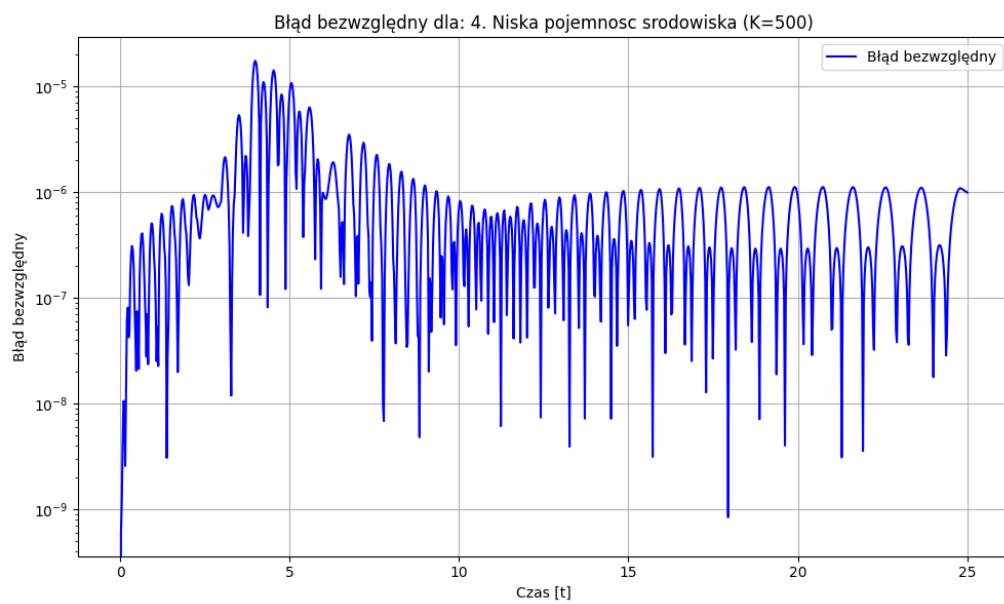


Rysunek 6. Wykres błędu bezwzględnego w funkcji czasu dla scenariusza spadku populacji.

3.4. Niska pojemność środowiska ($K=500$):

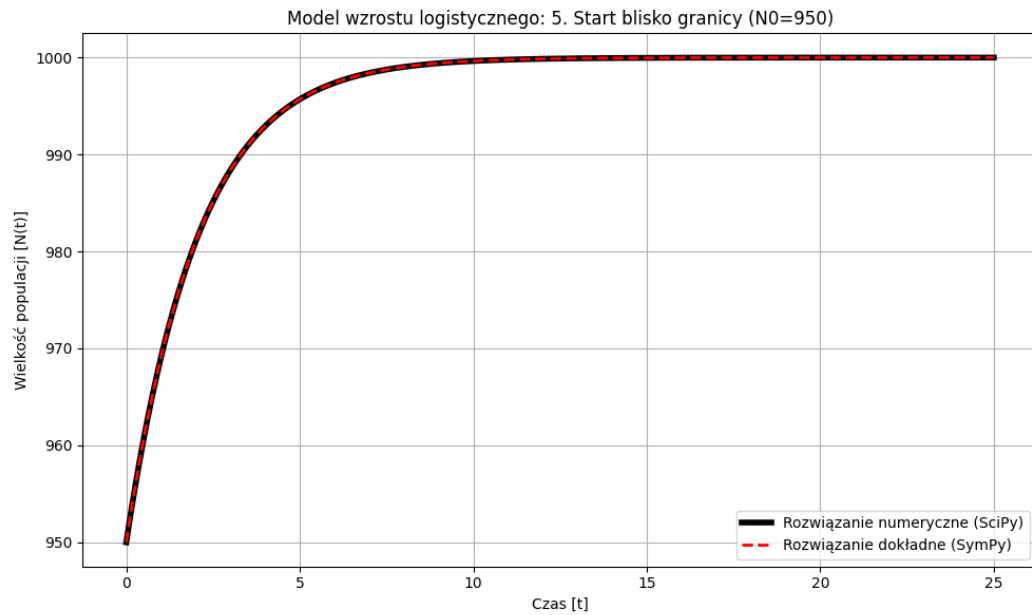


Rysunek 7. Porównanie rozwiązania numerycznego z analitycznym (dokładnym) dla scenariusza z niską pojemnością środowiska ($K=500$).

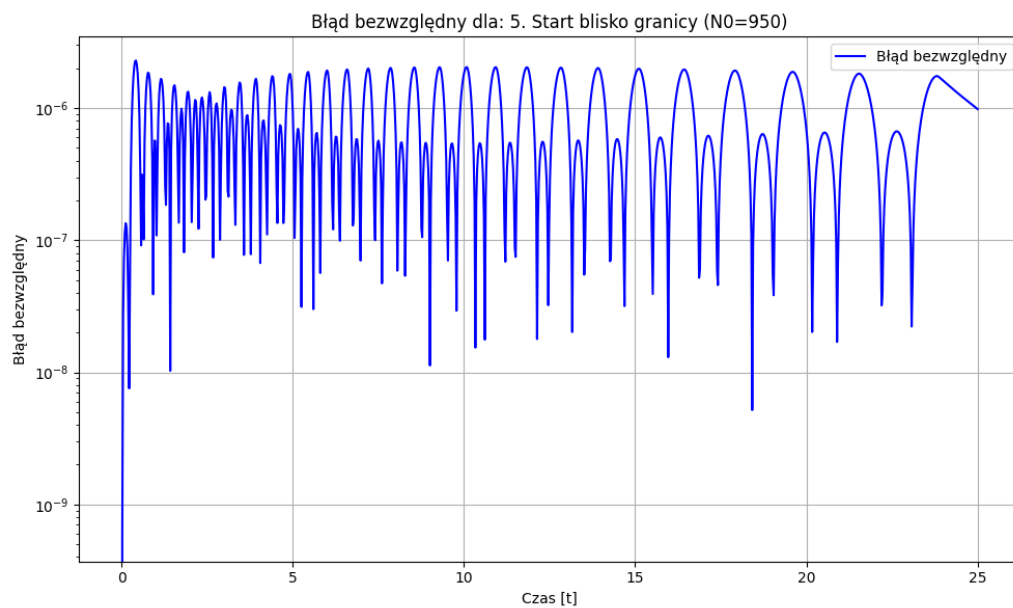


Rysunek 8. Wykres błędu bezwzględnego w funkcji czasu dla scenariusza z niską pojemnością środowiska.

3.5. Start blisko granicy ($N_0=950$):



Rysunek 9. Porównanie trajektorii numerycznej i analitycznej (dokładnej) dla scenariusza ze startem blisko pojemności środowiska ($N_0=950$).



Rysunek 10. Wykres błędu bezwzględnego w funkcji czasu dla scenariusza ze startem blisko pojemności środowiska.

Podsumowanie dla każdego z przypadków

1. **Scenariusz bazowy:** Symulacja wiernie odwzorowuje klasyczną, esowatą dynamikę wzrostu logistycznego, przechodząc przez fazę powolnego startu, gwałtownej ekspansji i końcowego nasycenia przy osiągnięciu pojemności środowiska.
2. **Szybki wzrost:** Zwiększony współczynnik wzrostu r prowadzi do znacznie bardziej agresywnej i skróconej fazy ekspansji, w wyniku której populacja osiąga stan równowagi w znacznie krótszym czasie, co ilustruje wrażliwość modelu na ten parametr.
3. **Spadek populacji:** Ten przypadek demonstruje samoregulujący charakter modelu; populacja startująca powyżej pojemności środowiska w sposób stabilny maleje, dążąc do poziomu równowagi K , co potwierdza jego rolę jako atraktora.
4. **Niska pojemność środowiska:** Obniżona pojemność środowiska K skutecznie ogranicza maksymalną wielkość populacji, która stabilizuje się na niższym poziomie, co dowodzi, że zasoby środowiskowe są kluczowym czynnikiem limitującym wzrost.
5. **Start blisko granicy:** Startując blisko pojemności środowiska, system pomija fazę gwałtownego wzrostu i od razu wchodzi w etap nasycenia, co pokazuje zachowanie modelu w warunkach bliskich stanu równowagi i skutkuje najniższymi błędami numerycznymi.

	MAE	MSE
1. Scenariusz bazowy	1.986482e-06	1.865908e-11
2. Szybki wzrost ($r=2.0$)	2.474823e-06	1.471049e-11
3. Spadek populacji ($N_0 > K$)	1.138523e-06	2.124529e-12
4. Niska pojemność środowiska ($K=500$)	1.037557e-06	5.227425e-12
5. Start blisko granicy ($N_0=950$)	8.899362e-07	1.157025e-12

Tabela 1. Obliczone średnie bezwzględne i kwadratowe błędy

4. Wnioski i podsumowanie

- Przeprowadzona analiza i symulacje komputerowe pozwoliły na dokładne zbadanie modelu wzrostu logistycznego populacji oraz weryfikację poprawności jego implementacji numerycznej. Głównym wnioskiem jest potwierdzenie wysokiej zgodności rozwiązania numerycznego z analitycznym, co udokumentowano za pomocą niskich wartości błędów MAE i MSE (Tabela 1.) we wszystkich analizowanych scenariuszach.
- Badanie pięciu różnych scenariuszy w czytelny sposób zilustrowało kluczowy wpływ parametrów modelu na jego dynamikę. Wykazano, że współczynnik wzrostu r determinuje tempo osiągnięcia stanu równowagi, pojemność środowiska K wyznacza stabilny poziom maksymalnej populacji, a warunek początkowy N_0 decyduje o początkowym zachowaniu systemu – wzroście lub spadku w kierunku punktu równowagi.
- Szczegółowa analiza wykresów błędów bezwzględnych dostarczyła głębszych wniosków na temat działania solwera numerycznego (ode). Potwierdzono, że największe odchylenia od rozwiązania dokładnego występują w fazach najszybszych zmian symulowanego systemu, gdzie pochodna funkcji osiąga ekstremum. Dodatkowo, zaobserwowano, że częstotliwość oscylacji błędów jest odwrotnie proporcjonalna do dynamiki systemu. W okresach dużej dynamiki, jak na początku scenariusza spadku populacji, solwer używa małych kroków czasowych, co skutkuje gęstymi oscylacjami błędów. Natomiast w miarę stabilizacji systemu, solwer adaptacyjnie zwiększa krok czasowy, co wizualnie objawia się jako "rozciągnięcie" i spadek częstotliwości oscylacji widocznych na wykresie.
- Podsumowując, zrealizowany projekt nie tylko zweryfikował poprawność modelu logistycznego, ale również pozwolił na praktyczne zbadanie charakterystyki i zachowania nowoczesnych metod numerycznych.