MATHEMATICAL SCIENCES

MATHEMATICAL MODELING ONE PROBLEMS FROM ECOLOGY

Asgarli R.
Baku State University , Second year master of the Department of Equations of Mathematical Physics of the Faculty of Applied Mathematica and Cybernetics, Baku, Azerbaijan
Abbasova A.
Baku State University, Associate professor of the Department of Equations of Mathematical Physics of the Faculty of Applied Mathematics and Cybernetics, Baku, Azerbaijan
https://doi.org/10.5281/zenodo.20186443

Abstract

In this work, a particular ecologically significant problem is analytically described using differential equations and contemprorary computer method are used to implement the solution.

Keywords: differential equations, mathematical modeling, Newton's law of cooling, numerical methods.

It should be mentioned, that Newton's law of cooling is offered as a useful tool for similating ecological and biological issues. The problem under consideration is applied to a specific situation: the object's temperature change over time is modeled in order to ascertain when a gazelle carcass discovered in the forest was slain [1],[2].

Consider the following situation:

While walking through a nature reserve, a ranger sees the body of a dead gazelle. Upon examination, it becomes clear that the gazelle was shot by a poacher with a well-aimed shot. Two people are detained nearby, but to determine who committed the crime, it is necessary to determine when the animal was killed.

This problem can be solved using Newton's law of cooling. According to this law, the rate of cooling of an object in air is proportional to the difference in temperature between the object and the air, i.e.

$$\frac{dy}{dt} = -k(y-a), y(0) = y_0. \quad (1)$$

Here y is the temperature of the body at time t , a is the air temperature, and k is the proportionality coefficient, y_0 is the temperature of the body at time $t = 0$. The solution to the problem depends on the study of the solution to the equation. It should be taken into account that the air temperature may not change after the gazelle is killed, but it may change over time. In the first case, the solution to the equation is as follows:

$$ln\frac{y-a}{y_0-a} = -kt, y \neq a, \quad (2)$$

If we fix the moment $t = 0$, when the unknown persons were detained, the gazelle's body temperature was $y = 31^oC$, and an hour later, $y = 29^oC$, we can determine the time of firing if we know that the gazelle's body temperature at the moment of firing was $y = 37^oC$ and the air temperature was $a = 21^0C$. Taking into account the given, we get from the last expression:

$$k = ln\frac{31-21}{29-21} = ln1{,}25 \approx 0{,}223 \ (3)$$

If we substitute (3) for k into (2) and take into account, that $y = 37^oC$, we get

$$t = -\frac{1}{0{,}223}ln\frac{37-31}{31-21} = -\frac{1}{0{,}223}ln1{,}6 \approx -2{,}106 \ ,$$

That is, 2 hours and 6 minutes had passed from the moment the shooting began until the moment the unknown persons were detained. Note that in the considered situation, the air temperature does not change

Consider anothe situation:

When the air temperature changes over time, the cooling law of the body is expressed by a linear non-homogeneous differential equation:

$$\frac{dy}{dt} + ky = kb(t), \quad (4)$$

where $b(t)$ is the air temperature at time t.

Let us assume that the temperature of the gazelle's body at the time the suspects was detained was 30^oC. It is also known that on the day of the incident, the air temperature dropped by 1°C every hour in the afternoon and was 0^oC when the body was found. Let us assume that one hour after the body was found, its temperature was 25^oC and the air temperature was –1°C. Suppose that the moment of the shooting was $t = 0$ and that the gazelle's body temperature at that moment was $y_0 = 37^oC$. If the time of the discovery of the killed gazelle is $t = t_1$, then $b(t) = t_1 - t$.

It is known that (4) is a linear non-homogeneous equation. If we apply the method of variation of a constant, the solution of equation (1) will be as follows:

$$y = \left(y_0 - t - \frac{1}{k}\right)e^{-kt} + t_1 - t + \frac{1}{k}. \ (5)$$

Taking into account, that at $t = t_1$ initial temperature is $y_0 = 30^oC$ and at $t = 1 + t_1$ $y_0 = 25^oC$ (5) bərabərliyindən aşağıdaki münasibətləri alırıq:

$$\left(37 - t_1 - \frac{1}{k}\right)e^{-kt} + \frac{1}{k} = 30,$$

$$\left(37 - t_1 - \frac{1}{k}\right)e^{-k(t_1+1)} + \frac{1}{k} = 26.$$

From the last two equations we obtain a transcendental equation with respect to k :

$$\left(30 - \frac{1}{k}\right)e^{-k} - 26 + \frac{1}{k} = 0 \ (6)$$

This type of equation cannot be solved by algebraic methods, so we apply one of the numerical methods, Newton's successive approximation method, to solve equation (6). For this, we reduce equation (6) to the following form:

$$30k - 1 + (1 - 26k)e^k = 0, \quad (7)$$

The last equation has the form:

$$(ak + b)e^{\alpha k} + ck + d = 0 \ . \quad (8)$$

In our case : a = –26, b = 1, c = 30, a = 1, d = –1.

Based on the results obtained, it was determined that the gazelle was killed approximately 1 hour and 12 minutes before its body was found. [1].

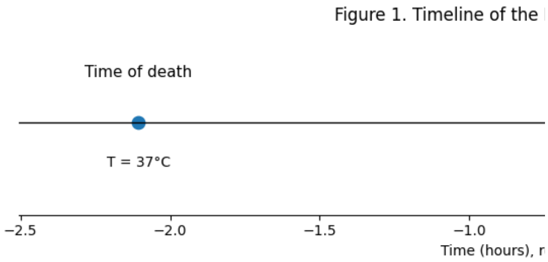
Using Python Programming Tools in the Analytical Solution of the Model

The Python programming language was used to more efficiently and accurately analyze the mathematical expression of the model [3]. The model, constructed based on the specified parameters, took into account the dynamics of temperature changes, and appropriate solution methods were applied for both constant and variable environmental conditions. The Python libraries numpy, math, and matplotlib were used to perform calculations, graphically represent functions, and find roots. In addition to the analytical solution of the model, numerical approaches were also implemented using the software codes presented in the study. In particular, the roots of the functions were determined using classical numerical methods, such as the bisection method, ensuring the reliability of the results.

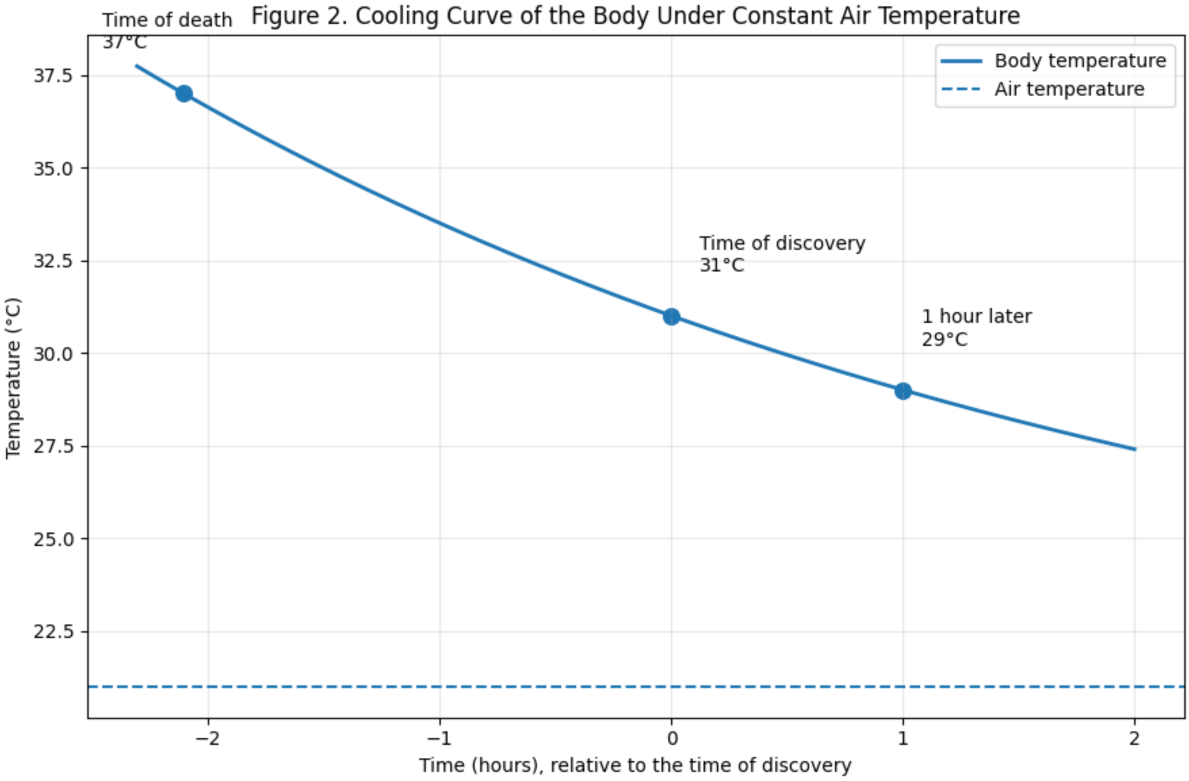
Thus, the mathematical expression of the model was practically implemented using the specified software, and the obtained results were analyzed from both theoretical and applied perspectives.

```
import math
import numpy as np
import matplotlib.pyplot as plt

# ========================
#The main parameters in the article
# ========================

# Under constant air temperature conditions
T_body_initial = 37.0
T_air_const = 21.0
T_found_const = 31.0
T_found_1h_const = 29.0

# In conditions of varying air temperature
T_found_var = 30.0
T_found_1h_var = 25.0

# ========================
# Auxiliary functions
# ========================

def bisection(func, a, b, tol=1e-10, max_iter=300):
 fa = func(a)
 fb = func(b)

 if fa == 0:
 return a
 if fb == 0:
 return b
 if fa * fb > 0:
 raise ValueError("No root has been isolated in the interval.")

 for _ in range(max_iter):
 m = 0.5 * (a + b)
 fm = func(m)
 if abs(fm) < tol or 0.5 * (b - a) < tol:
 return m

 if fa * fm <= 0:
 b = m
 fb = fm
 else:
 a = m
 fa = fm
 return 0.5 * (a + b)

def find_bracket(func, x_min=0.01, x_max=3.0, n=20000):
 xs = np.linspace(x_min, x_max, n)
 vals = [func(x) for x in xs]

 for i in range(len(xs) - 1):
 if vals[i] == 0:
 return xs[i], xs[i]
 if vals[i] * vals[i + 1] < 0:
 return xs[i], xs[i + 1]

 raise ValueError("No interval was found ford the root.")

# ========================
# Under constant temperature conditions
# ========================

k_const = math.log((T_found_const - T_air_const) / (T_found_1h_const - T_air_const))

t_death_relative_to_found = -(1 / k_const) * math.log(
 (T_body_initial - T_air_const) / (T_found_const - T_air_const)
)
def T_const_relative_to_found(t):
 return T_air_const + (T_found_const - T_air_const) * np.exp(-k_const * t)

elapsed_const = abs(t_death_relative_to_found)

# ========================
# Under varying temperature conditions
# ========================

def f_k_var(k, T_found=T_found_var, T_found_1h=T_found_1h_var):
 return (T_found - 1.0 / k) * math.exp(-k) - (T_found_1h + 1.0) + 1.0 / k

a_k, b_k = find_bracket(lambda x: f_k_var(x), x_min=0.01, x_max=2.0)
k_var = bisection(lambda x: f_k_var(x), a_k, b_k)

def g_t1_var(t1, k, T_found=T_found_var):
 return (37.0 - t1 - 1.0 / k) * math.exp(-k * t1) + 1.0 / k - T_found

a_t1, b_t1 = find_bracket(lambda x: g_t1_var(x, k_var), x_min=0.01, x_max=10.0)
```

```
t1_var = bisection(lambda x: g_t1_var(x, k_var), a_t1, b_t1)

def T_air_var(t, t1=t1_var):
 return t1 - t

def T_body_var(t, t1=t1_var, k=k_var):
 return t1 - t + 1.0 / k + (37.0 - t1 - 1.0 / k) * np.exp(-k * t)

print("The time of death under constant conditions:", round(elapsed_const, 3), "hours")
print("The time of death under varying conditions:", round(t1_var, 3), "hours")
```

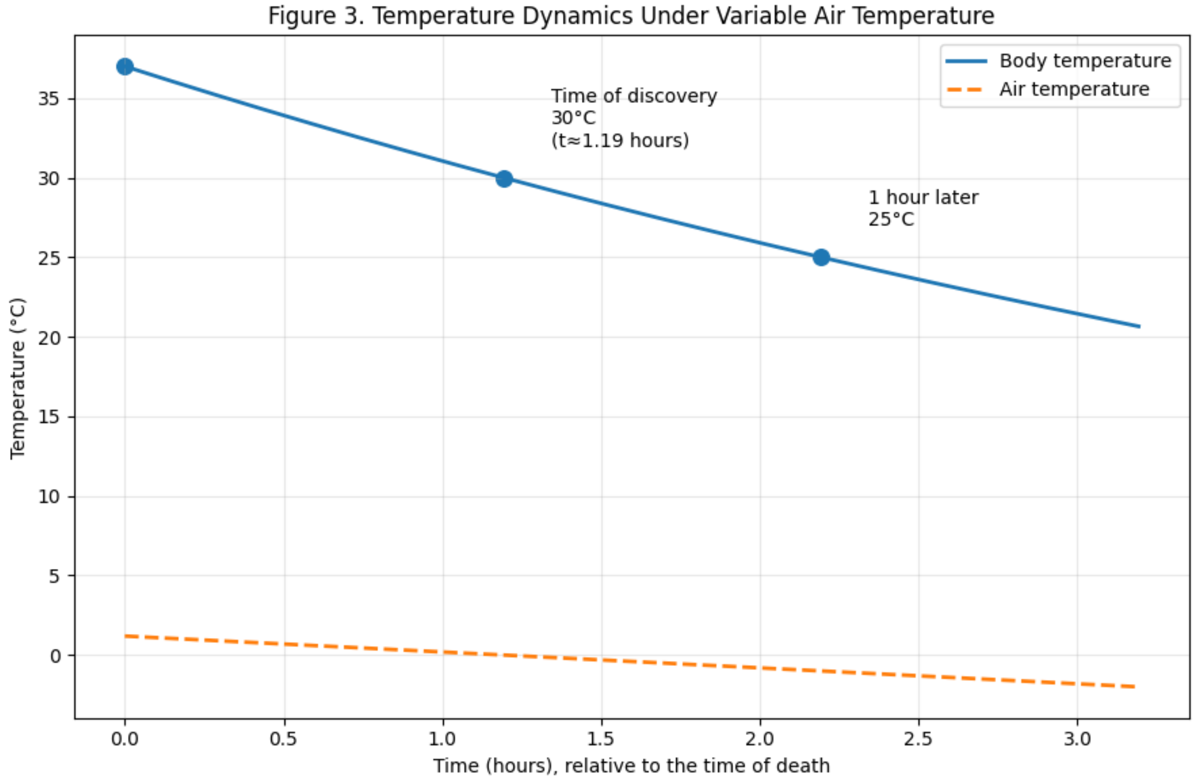



```
fig, ax = plt.subplots(figsize=(12, 2.8))

x1 = t_death_relative_to_found
x2 = 0
x3 = 1

ax.axhline(0, color='black', linewidth=1)
ax.scatter([x1, x2, x3], [0, 0, 0], s=80)

ax.text(x1, 0.12, "Time of death", ha="center", fontsize=11)
ax.text(x2, 0.12, "Time of discovery", ha="center", fontsize=11)
ax.text(x3, 0.12, " 1 hour later", ha="center", fontsize=11)

ax.text(x1, -0.12, "T = 37°C", ha="center", fontsize=10)
ax.text(x2, -0.12, "T = 31°C", ha="center", fontsize=10)
ax.text(x3, -0.12, "T = 29°C", ha="center", fontsize=10)

ax.set_title("Figure1. Timeline of the Event and Temperature Measurements ")
ax.set_xlim(x1 - 0.4, 1.4)
ax.set_ylim(-0.25, 0.25)
ax.set_yticks([])
ax.set_xlabel("Time (hours), relative to the time of discovery")

for spine in ["left", "right", "top"]:
 ax.spines[spine].set_visible(False)

plt.tight_layout()
plt.savefig("fig1_timeline.png", dpi=300, bbox_inches="tight")
plt.show()
plt.close()
```



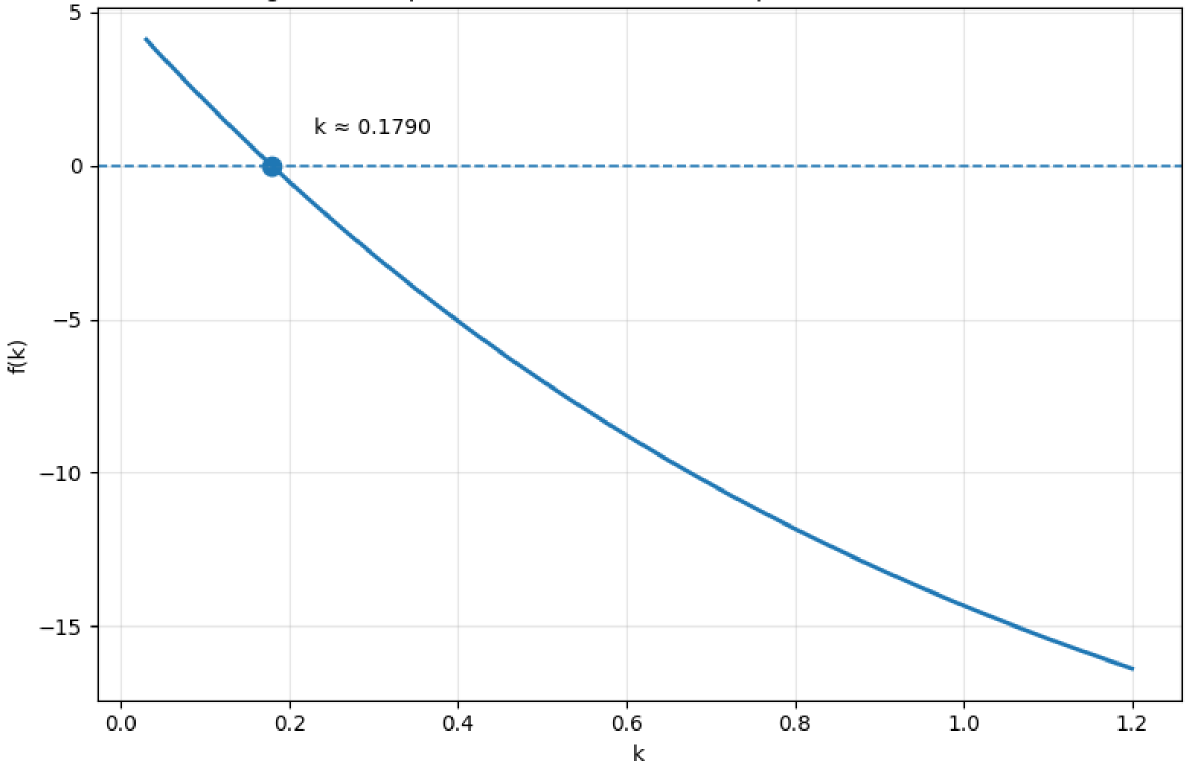

```
t_const = np.linspace(t_death_relative_to_found - 0.2, 2.0, 500)
T_const_vals = T_const_relative_to_found(t_const)

fig, ax = plt.subplots(figsize=(9, 6))

ax.plot(t_const, T_const_vals, linewidth=2, label="Body temperature ")
ax.axhline(T_air_const, linestyle="--", linewidth=1.5, label=" Air temperature ")

ax.scatter(
 [t_death_relative_to_found, 0, 1],
```

```
[T_body_initial, T_found_const, T_found_1h_const],
 s=70
)

ax.text(t_death_relative_to_found - 0.35, T_body_initial + 1.2, " Time of death \n37°C", fontsize=10)
ax.text(0.12, T_found_const + 1.2, " Time of discovery \n31°C", fontsize=10)
ax.text(1.08, T_found_1h_const + 1.2, "1 hour later \n29°C", fontsize=10)
ax.set_title("Figure 2. Cooling Curve of the Body Under Constant Air Temperature")
ax.set_xlabel("Time (hours), relative to the time of discovery ")
ax.set_ylabel("Temperature (°C)")
ax.grid(True, alpha=0.3)
ax.legend()

plt.tight_layout()
plt.savefig("fig2_constant_cooling.png", dpi=300, bbox_inches="tight")
plt.show()
plt.close()
```



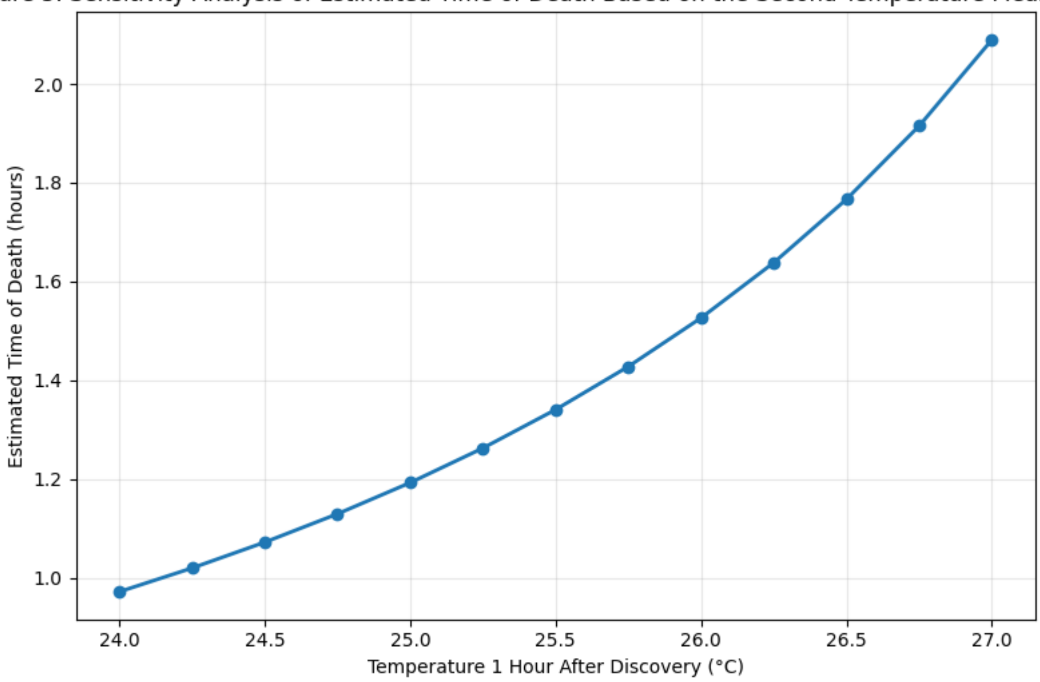

```
t_var = np.linspace(0, t1_var + 2.0, 600)
T_air_vals = T_air_var(t_var)
T_body_vals = T_body_var(t_var)

fig, ax = plt.subplots(figsize=(9, 6))

ax.plot(t_var, T_body_vals, linewidth=2, label=" Body temperature ")
ax.plot(t_var, T_air_vals, linestyle="--", linewidth=2, label=" Air temperature ")

ax.scatter([0, t1_var, t1_var + 1], [37.0, T_found_var, T_found_1h_var], s=70)

ax.text(0.15, 38.2, " Time of death \n37°C", fontsize=10)
ax.text(t1_var + 0.15, T_found_var + 2.0, f" Time of discovery \n30°C\n(t≈{t1_var:.2f} saat)", fontsize=10)
ax.text(t1_var + 1.15, T_found_1h_var + 2.0, "1 hour later \n25°C", fontsize=10)

ax.set_title("Figure 3. Temperature Dynamics Under Variable Air Temperature")
ax.set_xlabel("Time (hours), relative to the time of discovery ")
ax.set_ylabel("Temperature (°C)")
ax.grid(True, alpha=0.3)
ax.legend()

plt.tight_layout()
plt.savefig("fig3_variable_air_temp.png", dpi=300, bbox_inches="tight")
plt.show()
plt.close()
```

Figure 4. Graph of the Transcendental Equation and Root Point




```
k_grid = np.linspace(0.03, 1.2, 800)
f_vals = [f_k_var(k) for k in k_grid]

fig, ax = plt.subplots(figsize=(8, 5.5))

ax.plot(k_grid, f_vals, linewidth=2)
ax.axhline(0, linestyle="--", linewidth=1.2)
ax.scatter([k_var], [0], s=80)

ax.text(k_var + 0.05, 1.0, f"k ≈ {k_var:.4f}", fontsize=10)
ax.set_title("Figure 4. Graph of the Transcendental Equation and Root Point")
ax.set_xlabel("k")
ax.set_ylabel("f(k)")
ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig("fig4_transcendental_root.png", dpi=300, bbox_inches="tight")
plt.show()
plt.close()
```

Figure 5. Sensitivity Analysis of Estimated Time of Death Based on the Second Temperature Measurement




```
def solve_k_and_t1(T_found, T_found_1h):
    def f(k):
```

```
return (T_found - 1.0 / k) * math.exp(-k) - (T_found_1h + 1.0) + 1.0 / k

ak, bk = find_bracket(f, x_min=0.01, x_max=2.0)
k_sol = bisection(f, ak, bk)

def g(t1):
return (37.0 - t1 - 1.0 / k_sol) * math.exp(-k_sol * t1) + 1.0 / k_sol - T_found

at, bt = find_bracket(g, x_min=0.01, x_max=10.0)
t1_sol = bisection(g, at, bt)
return k_sol, t1_sol

T2_values = np.linspace(24.0, 27.0, 13)
estimated_times = []

for T2 in T2_values:
 _, t1_est = solve_k_and_t1(T_found_var, T2)
 estimated_times.append(t1_est)

fig, ax = plt.subplots(figsize=(8, 5.5))

ax.plot(T2_values, estimated_times, marker="o", linewidth=2)
ax.set_title("Figure 5. Sensitivity Analysis of Estimated Time of Death Based on the Second Temperature Measurement")
ax.set_xlabel("Temperature 1 Hour After Discovery (°C)")
ax.set_ylabel("Estimated Time of Death (hours)")
ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig("fig5_sensitivity_analysis.png", dpi=300, bbox_inches="tight")
plt.show()
plt.close()
```

References:

1. В. В. Амелькин. Дифференциальные уравнения в приложениях. Москва , Наука, 169 с., 1987.
2. X.M. Həmzəyev, Q.Q. İsmayılov. Ekoloji problemlərdə riyazi modelləşdirmə. Dərs vəsaiti, Bakı Azərb.DNSU-nun nəşri, 2002, 84 səh
3. V.Ə. Kərimov Ekoloji və iqtisadi sistemlərin analizi üçün bəzi hesablama üsulları. Metodiki vəsait. Bakı. Qanun, 2010, 84 səh
4. Вольтера В. Математическая теория борьбы за существование. М., Наука, 1976, 288 стр.