KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

Skaitiniai metodai ir algoritmai (P170B115)

4 laboratorinis darbas

17 variantas

Dėstytojas:

lekt. Andrius Kriščiūnas

Darbą atliko:

IFF – 8/13 Mykolas Paulauskas

KAUNAS, 2020

Turinys

Užduotis

T_1 temperatūros kūnas patalpinamas į aplinką, kurios temperatūra T_{A1}. Tariama, kad aplinkos temperatūra yra palaikoma išorinių šaltinių ir kūno temperatūra neturi įtakos aplinkos temperatūrai. Praėjus laikui t_s aplinkos temperatūra pradeda kisti pagal nurodytą dėsnį $T_A(t)$ ir pakinta iki T_{A2}, kuri yra palaikoma likusį laiką. Žinoma, kad Niutono temperatūros kaitos dėsnyje taikomas proporcingumo koeficientas priklauso nuo kūno temperatūros pagal dėsnį $k(T)$. Raskite, kaip kinta kūno temperatūra nuo pradinio laiko momento iki t_{max}. Kada kūno temperatūra pasiekia aplinkos temperatūrą?

Išspręskite tą pačią lygtį, jeigu pradinė kūno temperatūra lygi T_2. Kaip skiriasi su skirtingomis pradinėmis kūno temperatūromis gauti sprendiniai ir jų savybės (stabilumo ir tikslumo žingsniai)?

Varianto numeris	T_1, K	T_{A1}, K	t_s, s	T_{A2}, K	t_{max}, s	T_2, K
17	400	320	30	460	80	270

pav. 1 Duomenys

Varianto numeris	$T_A(t)$, K	$k(T)$
17	$T_A(t) = T_{A1} + \frac{(T_{A2} - T_{A1})}{2}\left(1 - \cos\left(\frac{\pi}{20}(t - t_s)\right)\right)$	$k(T) = -0{,}01 - 0{,}16\left(\frac{T - 273}{100}\right) - 0{,}04\left(\frac{T - 273}{100}\right)^2$

pav. 2 Formulės

Diferencialinė lygtis

Pagal duotą Niutono temperatūros kaitos dėsnį mes žinome, kad temperatūros kitimo greitis yra proporcingas kūno ir jo aplinkos temperatūrų skirtimui, kadangi greitis iš principo yra išvestinė mes galim susidaryti diferencialinę lygtį : $dTdt = k \cdot (T - T_A)$.

Pirmame etape mes tokią diferencialinę lygtį ir naudosime, tačiau atėjus kambario šildymo etapui, T_A bus skaičiuojamas pagal duotą kambario temperatūros kitimo formulę.

Kode realizuotos funkcijos atrodo taip:

Konstantos skaičiavimas:

```
def get_k(T):

    return -0.01 - 0.16 * (T - 273) / 100 - 0.04 * ((T - 273) / 100) ** 2
```

Kambario temperatūros didinimo funkcija:

```
if t >= ts and np.abs(TA - TA2) > margin:
            TA = TA1 + (TA2 - TA1) / 2 * (1 - np.cos(np.pi / 20 * (t - ts)))
```

Eulerio metodas

```
def euler_method(T1, TA1, ts, TA2, tmax, t0, dt):

    print("Euler method calculating...")

    x_over_time = [t0]
    y_subject_temp = [T1]
    y_env_temp = [TA1]
    TA = TA1
    T = T1
    margin = 1e-2
    subject_env_temp_is_equal = False

    for t in np.arange(t0, tmax, dt):

        # Praėjus duotam laikui temperatūros kitimo dėsnis keičiamas į TA(t)
        if t >= ts and np.abs(TA - TA2) > margin:
            TA = TA1 + (TA2 - TA1) / 2 * (1 - np.cos(np.pi / 20 * (t - ts)))

        # Eulerio formule. 18 skaidre
        k = get_k(T)
        dTdt = k * (T - TA)
        T = T + dt * dTdt

        if np.abs(T - TA) < margin and not subject_env_temp_is_equal:
            print(f"Kūno ir aplinkos temperatūros tapo vienodos praėjus: {t}s, te
mperatūra = {T}T")
            subject_env_temp_is_equal = True
            plt.scatter(t, T, color="red", zorder=3, label='Kūno temperatūra pasi
ekia aplinkos temperatūrą')

        x_over_time.append(t)
        y_subject_temp.append(T)
        y_env_temp.append(TA)

    plt.plot(x_over_time, y_subject_temp, zorder=1, label='Kūno temperatūra')
    plt.plot(x_over_time, y_env_temp, zorder=2, label='Aplinkos temperatūra')
```

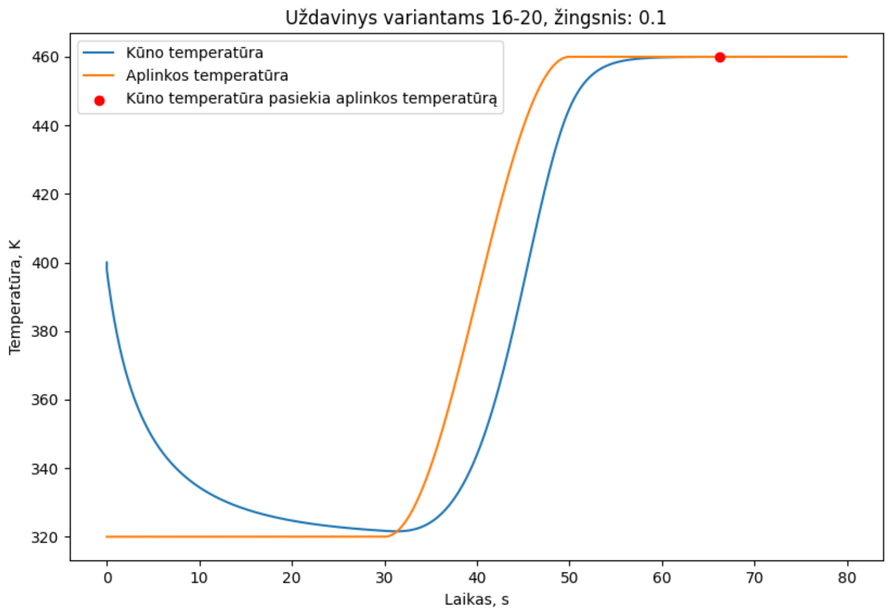
```
    plt.xlabel('Laikas, s')
    plt.ylabel('Temperatūra, K')
    plt.title(f"Uždavinys variantams 16-20, žingsnis: {dt}")
    plt.legend()
    plt.show()
```

IV eilės Rungės ir Kutos metodas

```
def runge_kutta_method(T1, TA1, ts, TA2, tmax, t0, dt):

    print("Runge-Kutta method calculating...")

    x_over_time = [t0]
    y_subject_temp = [T1]
    y_env_temp = [TA1]
    TA = TA1
    T = T1
    margin = 1e-2
    subject_env_temp_is_equal = False

    for t in np.arange(t0, tmax, dt):

        # Praėjus duotam laikui temperatūros kitimo dėsnis keičiamas į TA(t)
        if t >= ts and np.abs(TA - TA2) > margin:
            TA = TA1 + (TA2 - TA1) / 2 * (1 - np.cos(np.pi / 20 * (t - ts)))

        k = get_k(T)
        dTdt = k * (T - TA)
        f1 = dTdt

        # Pirmas Runge skaiciavimo etapas
        T1 = T + dt / 2 * f1
        k = get_k(T1)
        dTdt = k * (T1 - TA)
        f2 = dTdt

        # Antras Runge skaiciavimo etapas
        T2 = T + dt / 2 * f2
        k = get_k(T2)
        dTdt = k * (T2 - TA)
        f3 = dTdt
```

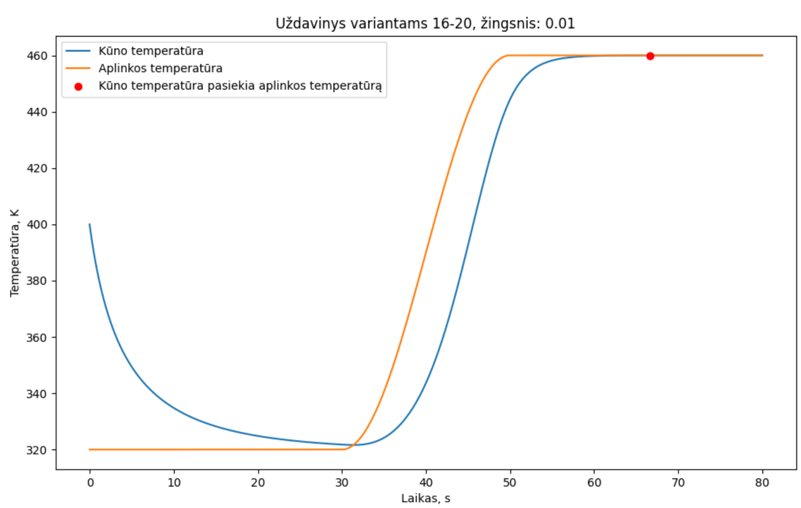
```
        # Trecias Runge skaiciavimo etapas
        T3 = T + dt * f3
        k = get_k(T3)
        dTdt = k * (T3 - TA)
        f4 = dTdt

        # Apjungimas. 30 skaidre
        T = T + dt / 6 * (f1 + 2 * f2 + 2 * f3 + f4)

        if np.abs(T - TA) < margin and not subject_env_temp_is_equal:
            print(f"Kūno ir aplinkos temperatūros tapo vienodos praėjus: {t}s, temperatūra = {T}T")
            subject_env_temp_is_equal = True
            plt.scatter(t, T, color="red", zorder=3, label='Kūno temperatūra pasiekia aplinkos temperatūrą')

        x_over_time.append(t)
        y_subject_temp.append(T)
        y_env_temp.append(TA)

    plt.plot(x_over_time, y_subject_temp, zorder=1, label='Kūno temperatūra')
    plt.plot(x_over_time, y_env_temp, zorder=2, label='Aplinkos temperatūra')
    plt.xlabel('Laikas, s')
    plt.ylabel('Temperatūra, K')
    plt.title(f"Uždavinys variantams 16-20, žingsnis: {dt}")
    plt.legend()
    plt.show()
```

Rezultatai pirmos dalies



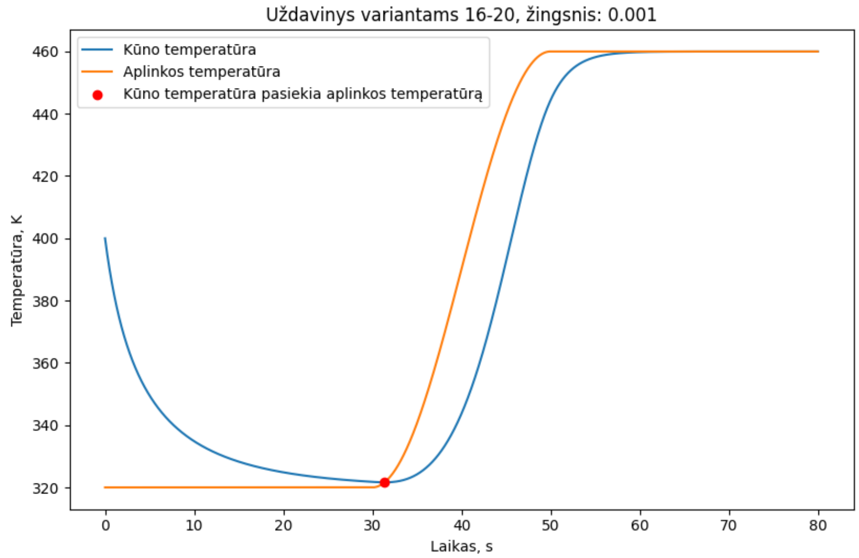
pav. 3 Eulerio metodo rezultatai, kai T = T1 ir žingsnis = 0.1

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 66.2s, temperatūra = 459.9815221341439T



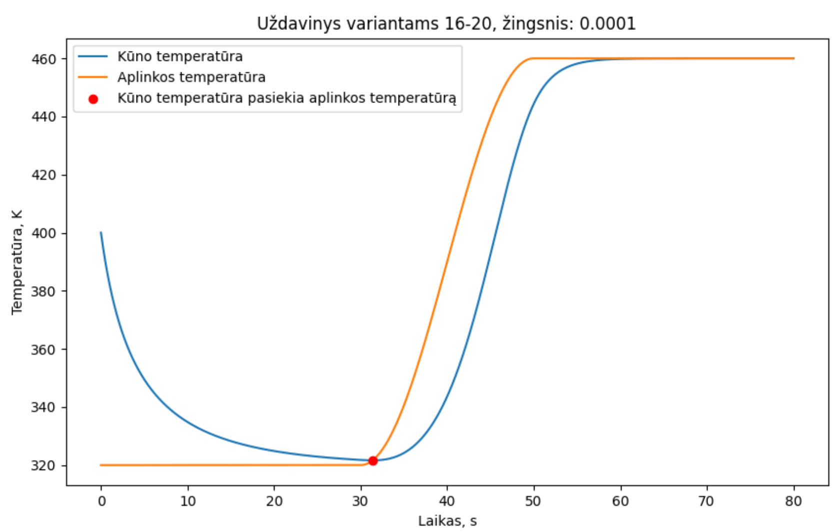
pav. 4 Eulerio metodo rezultatai, kai T = T1 ir žingsnis = 0.01

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 66.6s, temperatūra = 459.9813943146658T



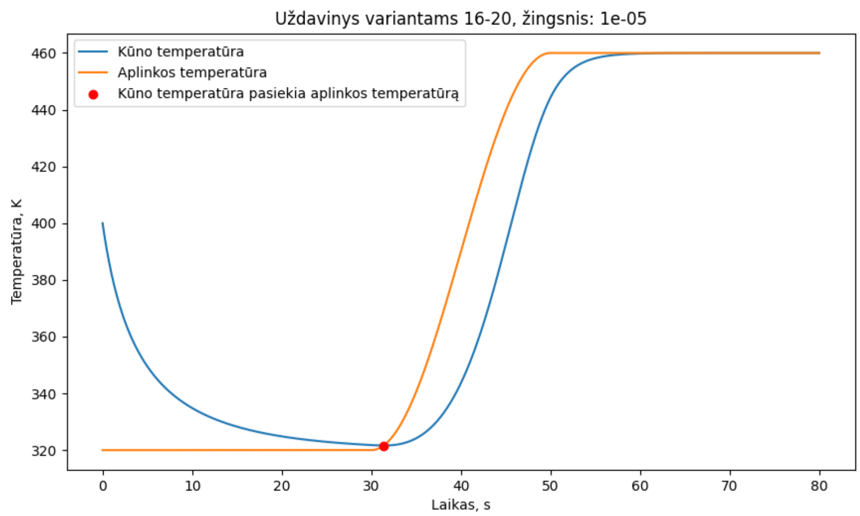
pav. 5 Eulerio metodo rezultatai, kai T = T1 ir žingsnis = 0.001

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 31.373s, temperatūra = 321.6296871510557T



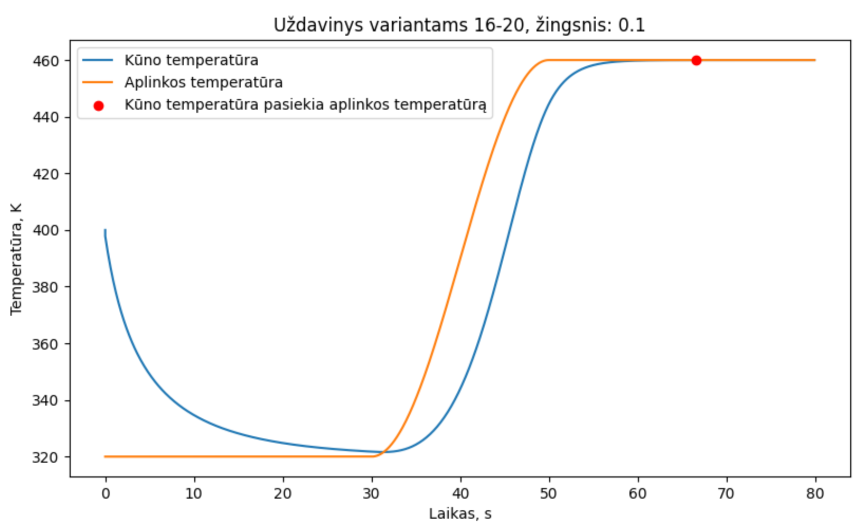
pav. 6 Eulerio metodo rezultatai, kai T = T1 ir žingsnis = 0.0001

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 31.372400000000003s, temperatūra = 321.6300937796677T



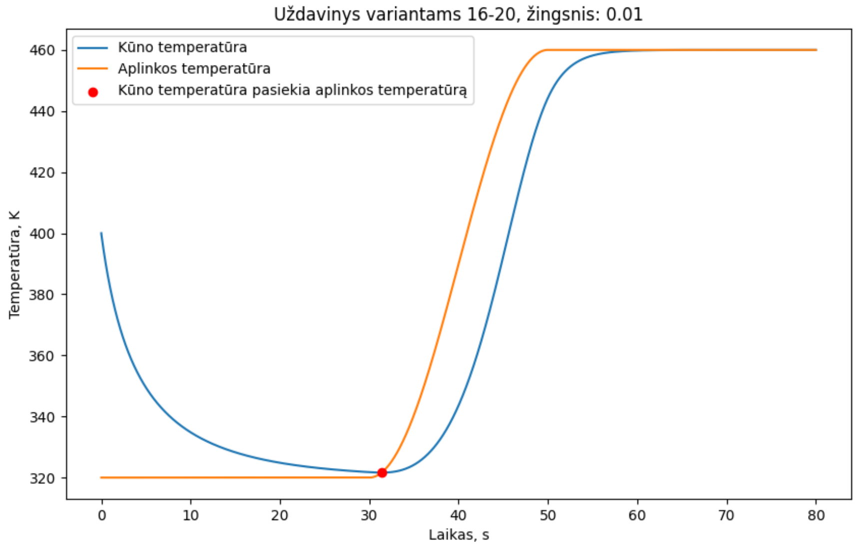
pav. 7 Eulerio metodo rezultatai, kai T = T1 ir žingsnis = 1e-05

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 31.37235s, temperatūra = 321.6301344528633T



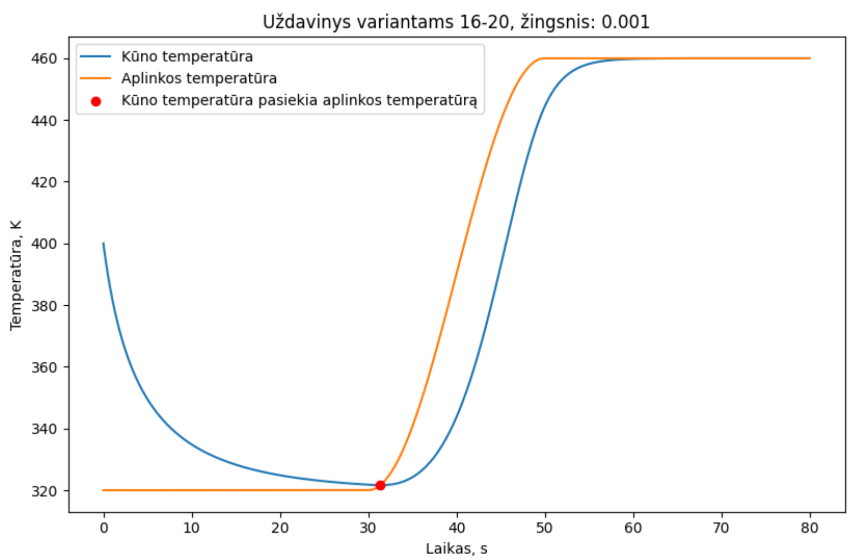
pav. 8 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T1 ir žingsnis = 0.1

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 66.60000000000001s, temperatūra = 459.98140531911633T



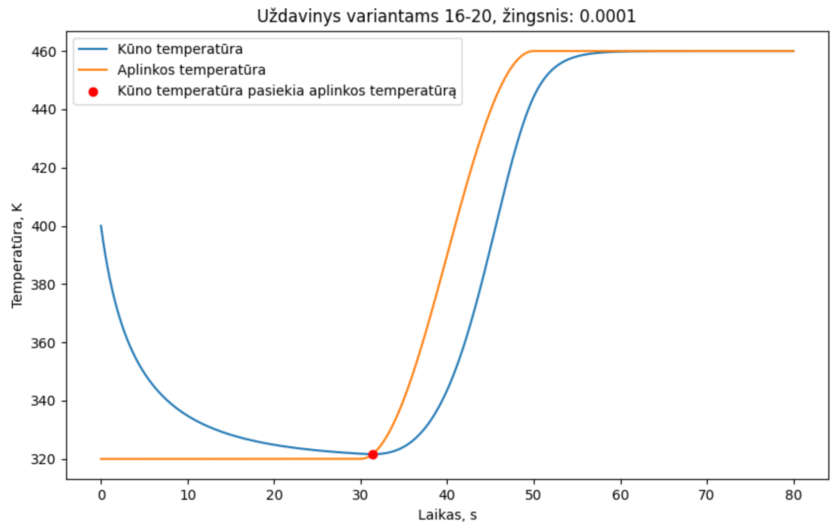
pav. 9 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T1 ir žingsnis = 0.01

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 31.38s, temperatūra = 321.62938631838955T



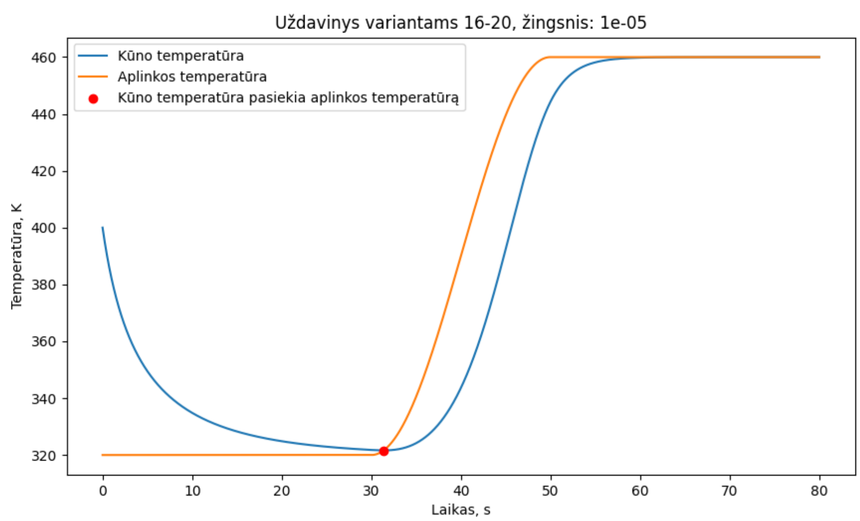
pav. 10 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T1 ir žingsnis = 0.001

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 31.373s, temperatūra = 321.6300621195401T



pav. 11 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T1 ir žingsnis = 0.0001

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 31.372400000000003s, temperatūra = 321.63013128270245T



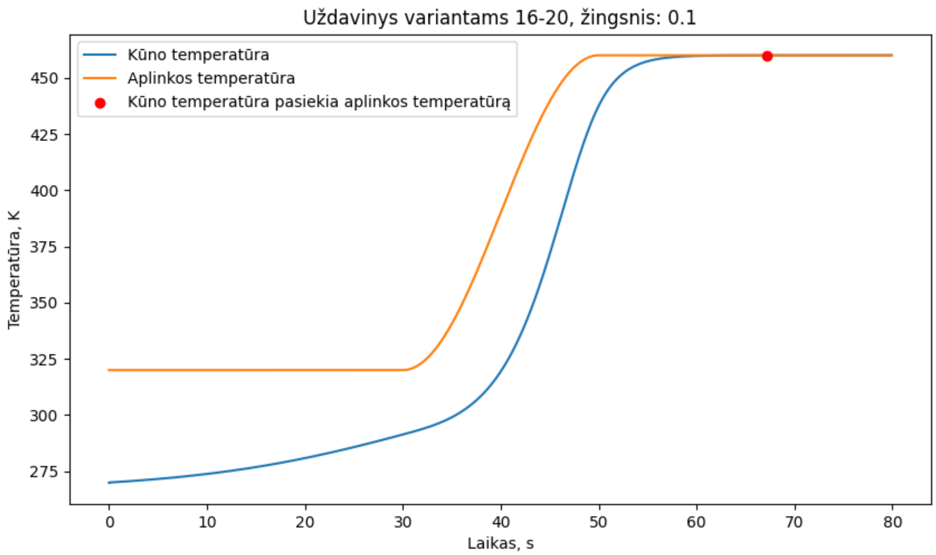
pav. 12 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T1 ir žingsnis = 1e-05

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 31.37235s, temperatūra = 321.63013820321015T

Pirmos dalies išvados

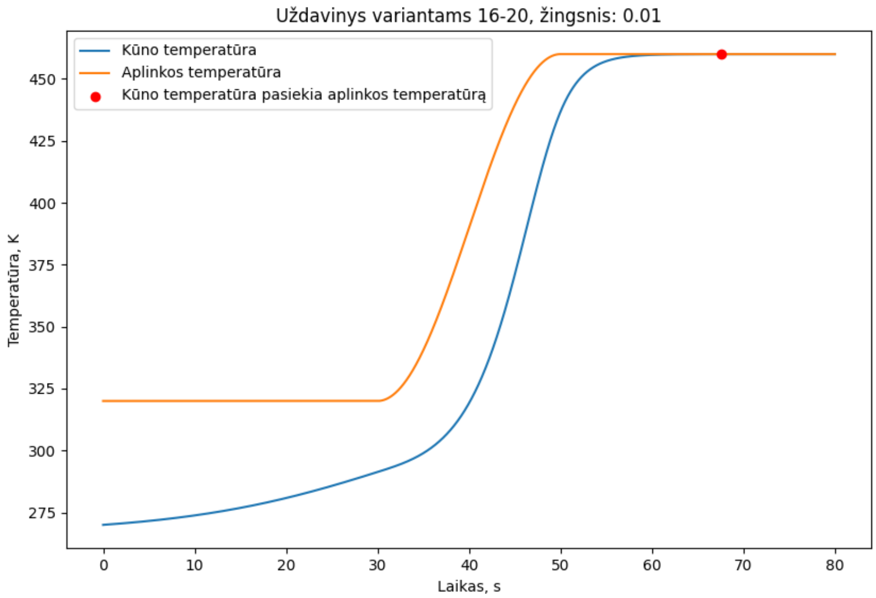
Iš gautų rezultatų matome, kad IV eilės Rungės ir Kutos metodas greičiau nustato kada kūnas pasiekia aplinkos temperatūra ir kad laikas, kada tai įvyksta, yra ~ 31.37 sekundę.

Rezultatai antros dalies



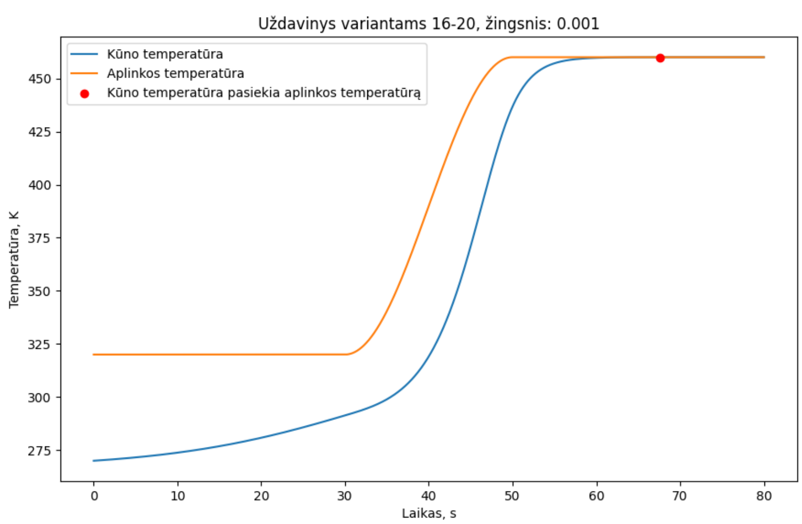
pav. 13 Eulerio metodo rezultatai, kai T = T2 ir žingsnis = 0.1

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 67.2s, temperatūra = 459.98145812496705T



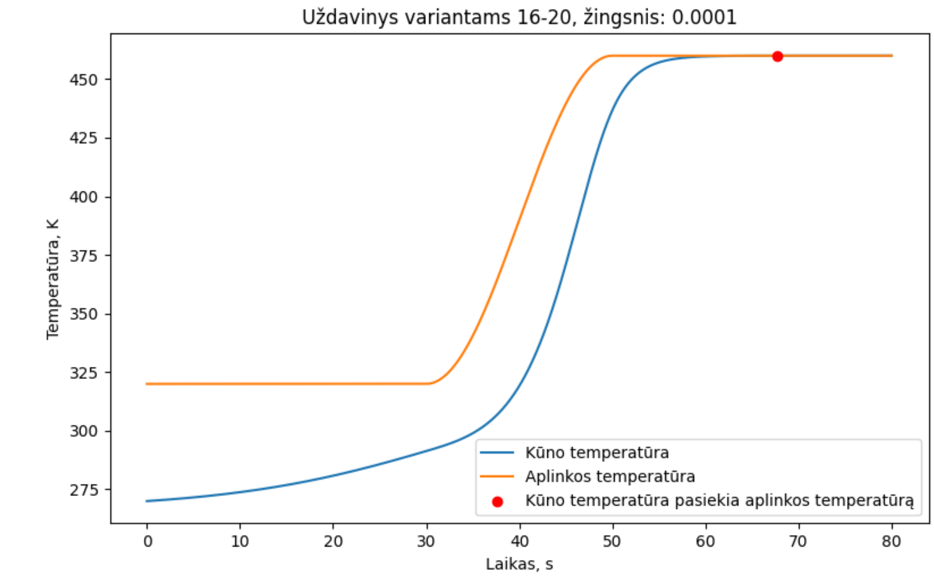
pav. 14 Eulerio metodo rezultatai, kai T = T2 ir žingsnis = 0.01

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 67.6s, temperatūra = 459.98137535695435T



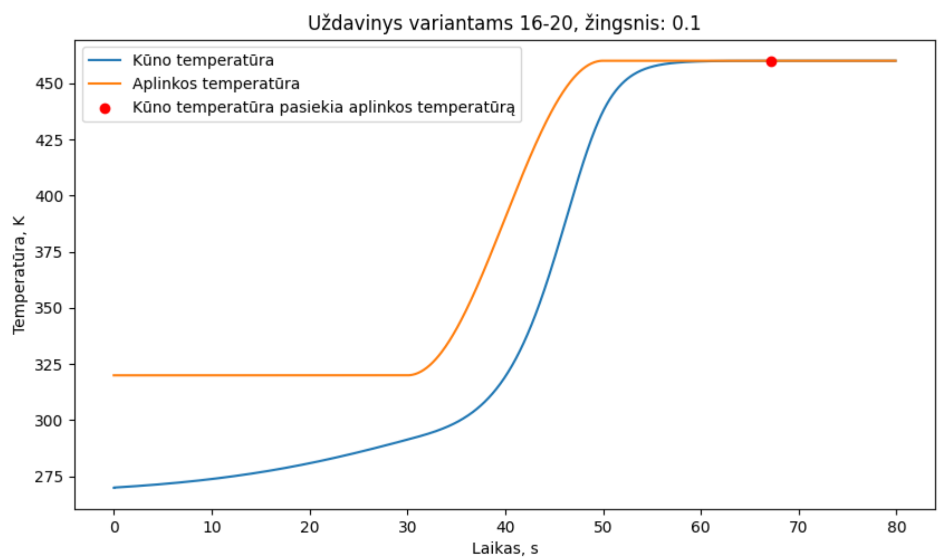
pav. 15 Eulerio metodo rezultatai, kai T = T2 ir žingsnis = 0.001

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 67.64s, temperatūra = 459.98011680866875T



pav. 16 Eulerio metodo rezultatai, kai T = T2 ir žingsnis = 0.0001

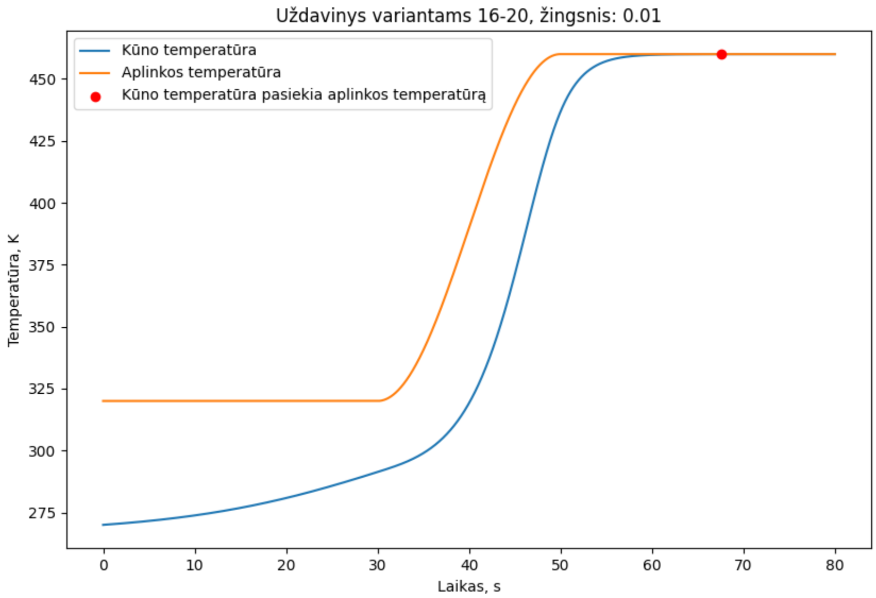
Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 67.6434s, temperatūra = 459.98000220971284T



pav. 17 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T2 ir žingsnis = 0.1

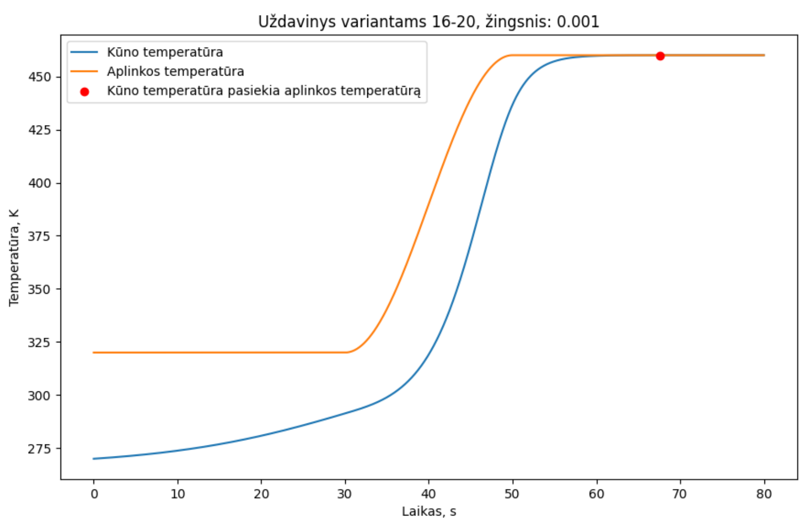
Kūno temperatūra pasiekė aplinkos temperatūrą, kai

laikas = 67.60000000000001s, temperatūra = 459.9814094993552T



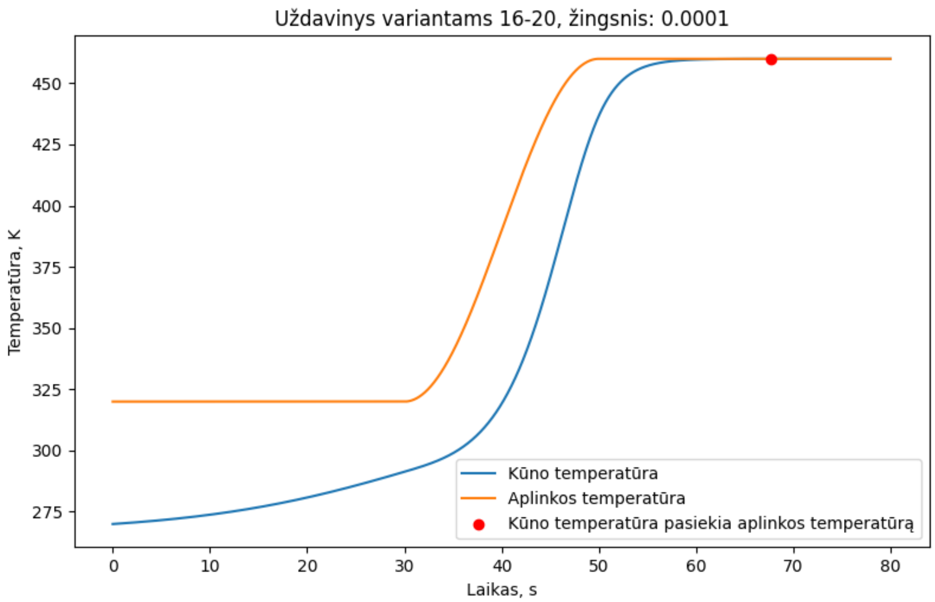
pav. 18 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T2 ir žingsnis = 0.01

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 67.64s, temperatūra = 459.9813716365492T



pav. 19 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T2 ir žingsnis = 0.001

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 67.644, temperatūra = 459.9801164477151T



pav. 20 IV eilės Rungės ir Kutos metodo rezultatai, kai T = T2 ir žingsnis = 0.0001

Kūno temperatūra pasiekė aplinkos temperatūrą, kai laikas = 67.6438, temperatūra = 459.9800021737776T

Antros dallies išvados

Iš gautų rezultatų matome, kad IV eilės Rungės ir Kutos metodas greičiau nustato kada kūnas pasiekia aplinkos temperatūra ir kad laikas, kada tai įvyksta, yra ~ 67.64 sekundę. Žingsnių kiekis yra mažesnis, lyginant su pirmąja temperatūra, kad galėtume nustatyti, ties kuria reikšme *nusistovi* gautas rezultatas.