KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Skaitiniai metodai ir algoritmai (P170B115)

4 Laboratorinio darbo ataskaita

Atliko:

IFF-1/5 gr. student

Jurgis Andziulis

Priėmė:

Doc. Andrius Kriščiūnas

KAUNAS 2023

TURINYS

1. Užduotis

3 Uždavinys variantams 1-10

Sujungti m_1 ir m_2 masių objektai iššaunami vertikaliai į viršų pradiniu greičiu v_0. Oro pasipriešinimo koeficientas sujungtiems kūnams lygus k_s. Praėjus laikui t_s, objektai pradeda judėti atskirai. Oro pasipriešinimo koeficientai atskirai judantiems objektams atitinkamai yra k_1 ir k_2. Oro pasipriešinimas proporcingas objekto greičio kvadratui. Raskite, kaip kinta objektų greičiai nuo 0 s iki t_{max}. Kada kiekvienas objektas pasieks aukščiausią tašką ir pradės leistis?

1 Lentelė. Uždavinyje naudojami dydžiai.

Varianto numeris	m_1, kg	m_2, kg	v_0, m/s	k_s, kg/m	t_s, s	k_1, kg/m	k_2, kg/m	t_{max}, s
1	0,2	0,4	80	0,015	1	0,02	0,005	15

2. Teorija

2.1. Diferencialinė lygtis

$$\frac{dv}{dt} = -g - \left(\frac{kv^2 sgn(v)}{m}\right)$$

● dv/dt – tai pagreitis, kuris parodo, kaip greitis kinta laikui bėgant.
● -g - tai pagreitis dėl gravitacijos, kuris yra neigiamas, nes gravitacija visada traukia žemyn.
● k*v^2 - čia k reiškia oro pasipriešinimą. v^2 rodo, kad oro pasipriešinimas didėja su greičio kvadratu.
● sgn(v) - naudojama siekiant užtikrinti, kad pasipriešinimas visada veiktų priešinga greičiui kryptimi.
● m - Tai objekto masė. Kuo sunkesnis objektas, tuo mažesnis pagreitis jį veikia tam tikra jėga (pagal antrąjį Niutono dėsnį F = ma).

```
def equations(t, y):
    if t < t_separate:
        dv_dt = -g - ks * y[0] ** 2 * np.sign(y[0]) / (m1 + m2)
        return [dv_dt, dv_dt]
    else:
        dv1_dt = -g - k1 * y[0] ** 2 * np.sign(y[0]) / m1
        dv2_dt = -g - k2 * y[1] ** 2 * np.sign(y[1]) / m2
        return [dv1_dt, dv2_dt]
```

2.2. Eulerio metodas

● Pradedame nuo taško, kurio padėtį ir greitį žinome.
● Apskaičiuojame pagreitį tame taške.
● Žengiame nedidelį žingsnį į priekį, reguliuodami greitį ir padėtį pagal apskaičiuotą pagreitį.
● Toliau kartojame šį procesą.

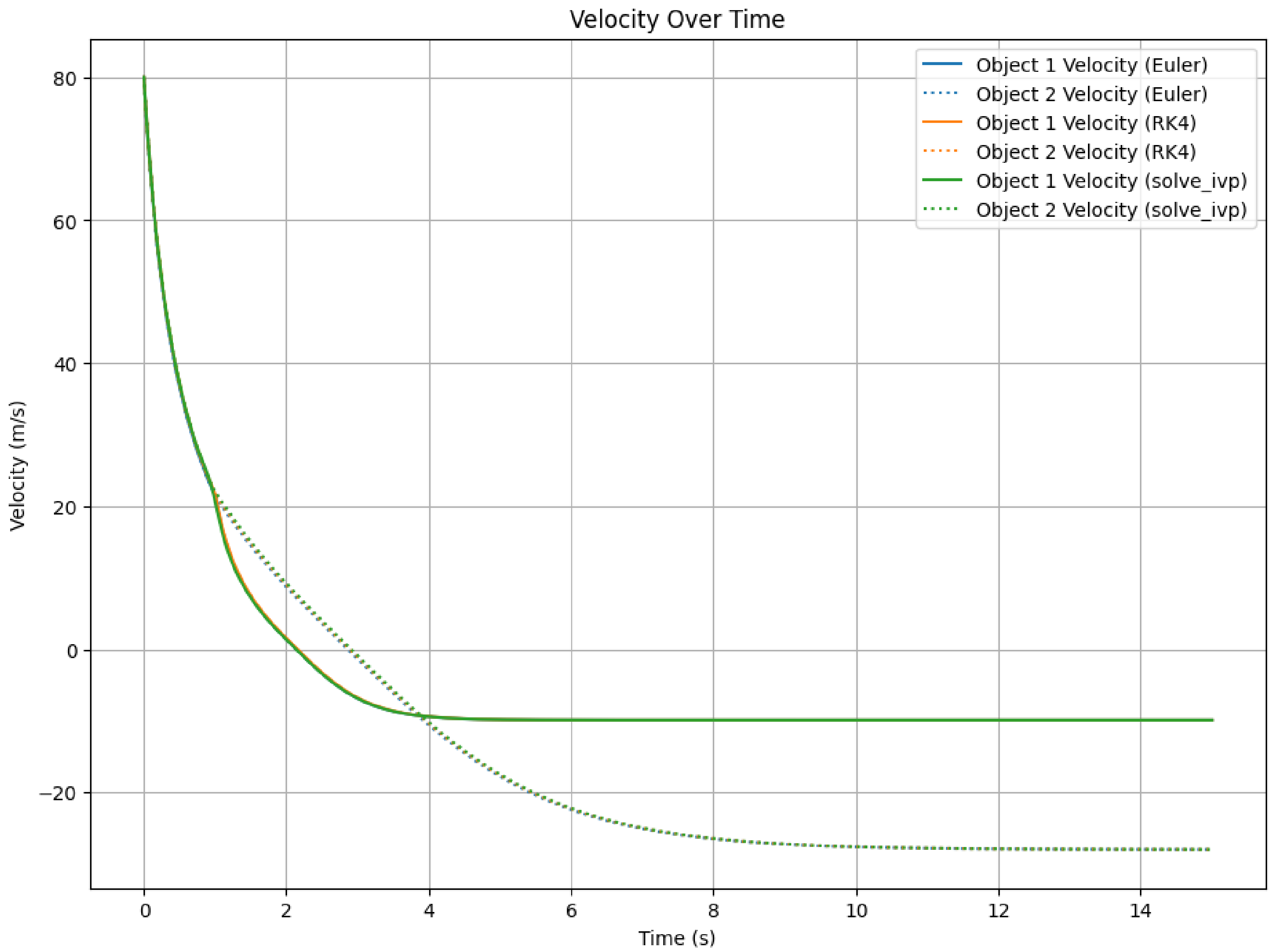
```python
def euler_method(f, t_span, y0, steps):
    t0, tf = t_span
    h = (tf - t0) / steps
    t_values = np.linspace(t0, tf, steps + 1)
    y_values = np.zeros((2, steps + 1))
    y_values[:, 0] = y0
    for i in range(steps):
        a = f(t_values[i], y_values[:, i])
        y_values[0, i + 1] = y_values[0, i] + h * a[0]
        y_values[1, i + 1] = y_values[1, i] + h * a[1]
    return t_values, y_values
```

2.3. *IV eilės Rungės ir Kutos metodas*

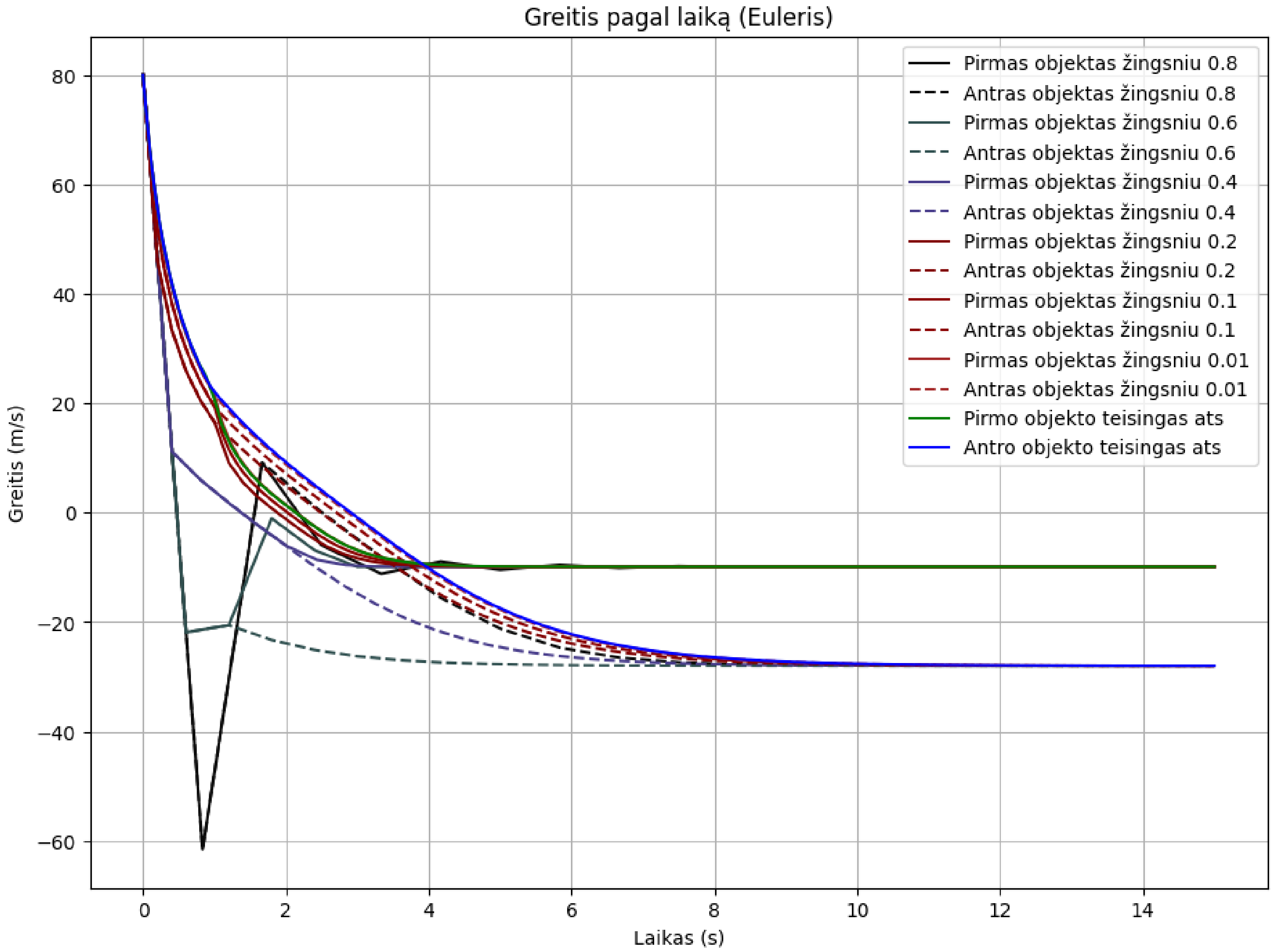
- Apskaičiuojame pagreitį intervalo pradžioje (kaip ir taikant Eulerio metodą).
- Įvertiname pagreitį intervalo viduryje, naudodami pirmajame etape apskaičiuotą pagreitį.
- Dar kartą įvertiname pagreitį įpusėjus intervalui, tačiau dabar naudojame antrajame etape apskaičiuotą pagreitį.
- Apskaičiuojame pagreitį intervalo pabaigoje, naudodami trečiojo žingsnio pagreitį.
- Apskaičiuojame šių pagreičių svertinį vidurkį.
- Pagal šį svertinį vidurkį atnaujiname padėtį ir greitį.

```python
def runge_kutta_4th_order(f, t_span, y0, steps):
    t0, tf = t_span
    h = (tf - t0) / steps
    t_values = np.linspace(t0, tf, steps + 1)
    y_values = np.zeros((2, steps + 1))
    y_values[:, 0] = y0
    for i in range(steps):
        k1 = np.array(f(t_values[i], y_values[:, i]))
        k2 = np.array(f(t_values[i] + h / 2, y_values[:, i] + h / 2 * k1))
        k3 = np.array(f(t_values[i] + h / 2, y_values[:, i] + h / 2 * k2))
        k4 = np.array(f(t_values[i] + h, y_values[:, i] + h * k3))
        y_values[:, i + 1] = y_values[:, i] + h / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
    return t_values, y_values
```

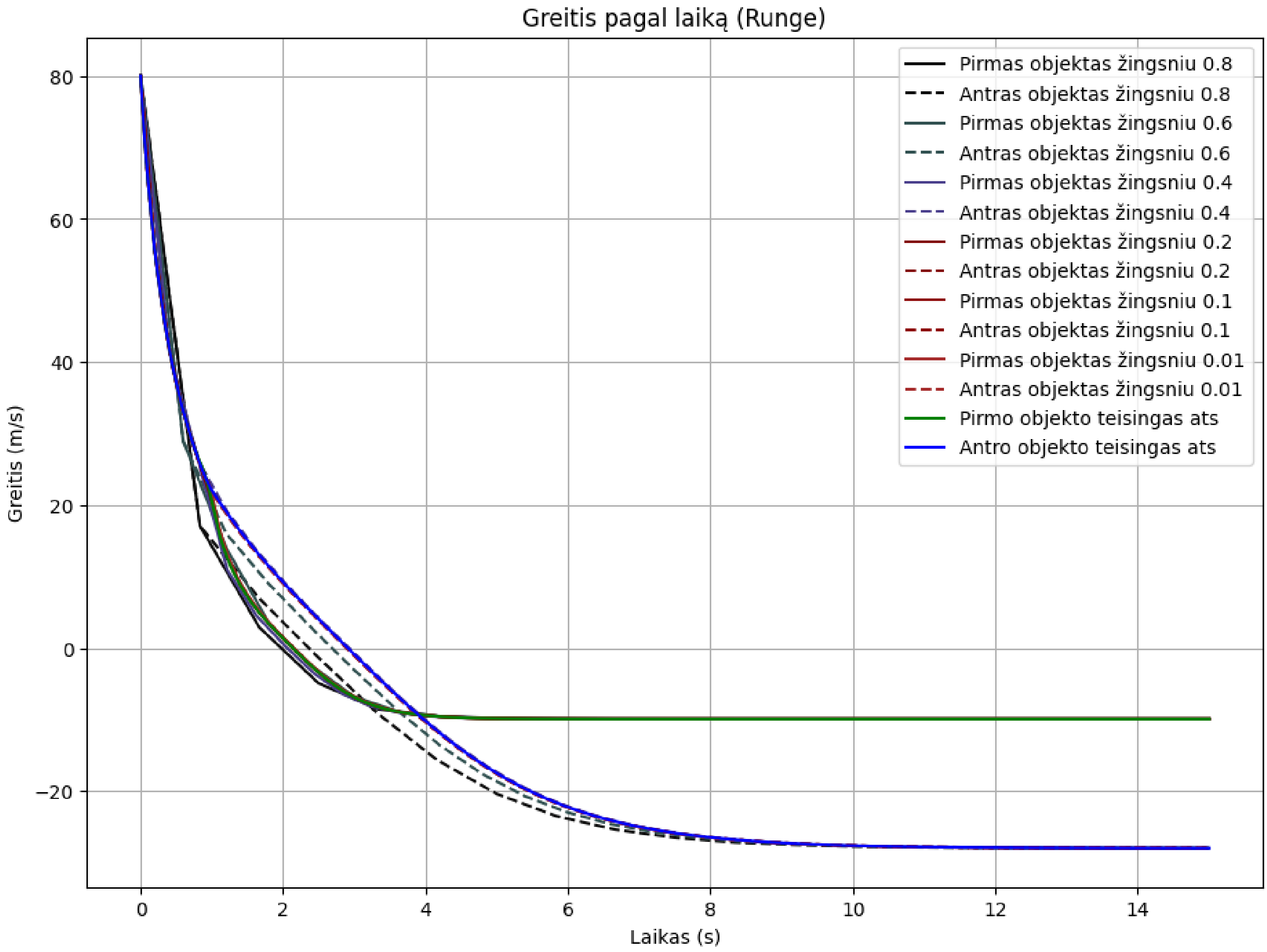
3. Rezultatai

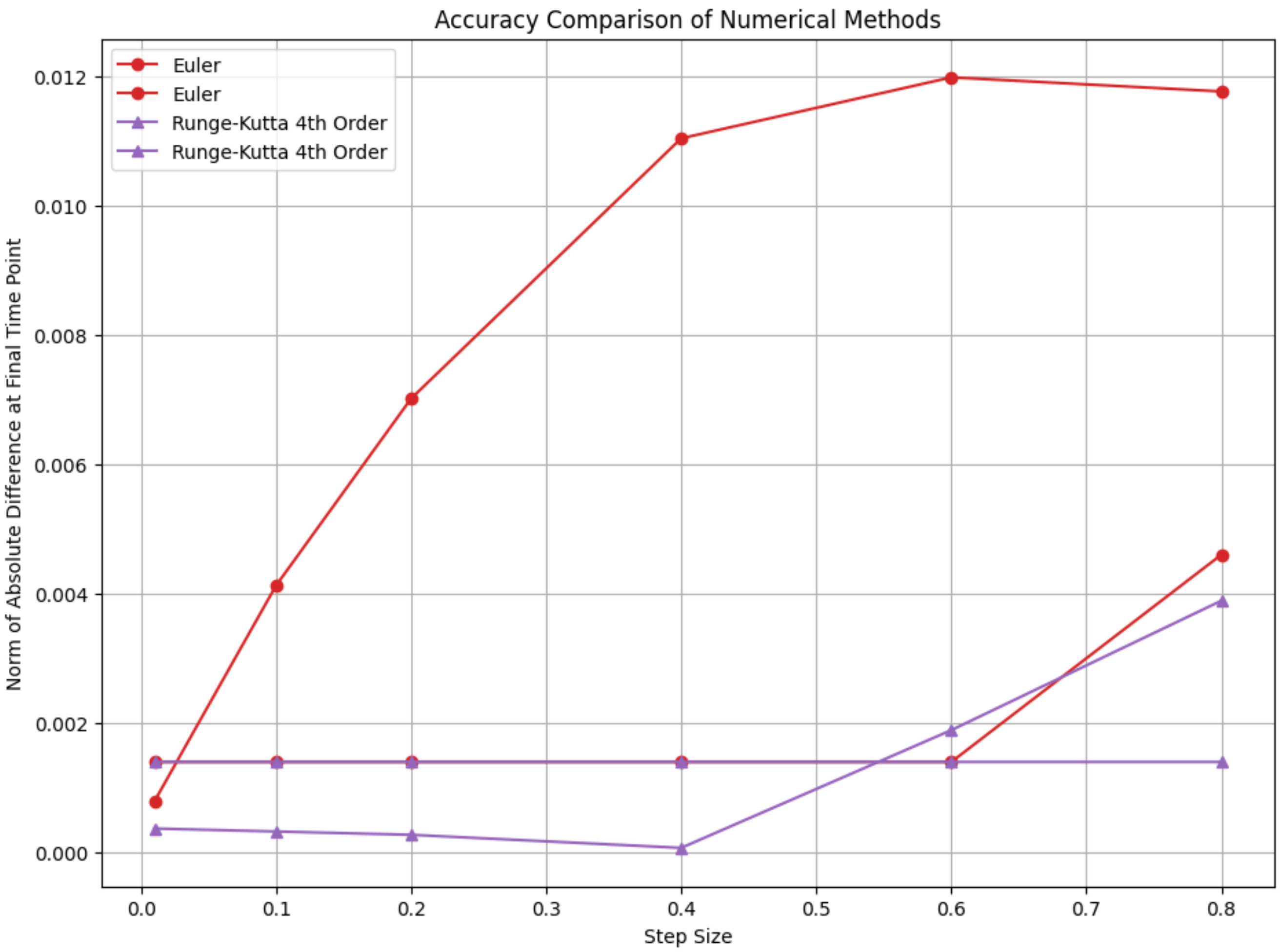


1 objektas pasiekia aukščiausią tašką (žingsnis 0.8) laiku 2.5s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.8) laiku 2.5s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.6) laiku 1.7999999999999998s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.6) laiku 1.2s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.4) laiku 1.2162162162162162s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.4) laiku 1.2162162162162162s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.2) laiku 1.8s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.2) laiku 2.4000000000000004s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.1) laiku 2.0s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.1) laiku 2.7s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.01) laiku 2.14s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.01) laiku 2.87s
1 objektas pasiekia aukščiausią tašką (tikslus) laiku 2.1321321321321323s
2 objektas pasiekia aukščiausią tašką (tikslus) laiku 2.9129129129129128s



1 objektas pasiekia aukščiausią tašką (žingsnis 0.8) laiku 1.6666666666666667s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.8) laiku 2.5s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.6) laiku 2.4s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.6) laiku 2.4s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.4) laiku 2.027027027027027s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.4) laiku 2.837837837837838s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.2) laiku 2.2s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.2) laiku 3.0s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.1) laiku 2.1s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.1) laiku 2.9000000000000004s
1 objektas pasiekia aukščiausią tašką (žingsnis 0.01) laiku 2.15s
2 objektas pasiekia aukščiausią tašką (žingsnis 0.01) laiku 2.89s
1 objektas pasiekia aukščiausią tašką (tikslus) laiku 2.1321321321321323s
2 objektas pasiekia aukščiausią tašką (tikslus) laiku 2.9129129129129128s


Greitis pagal laiką (Runge)
Greitis (m/s)
Laikas (s)
Pirmas objektas žingsniu 0.8
Antras objektas žingsniu 0.8
Pirmas objektas žingsniu 0.6
Antras objektas žingsniu 0.6
Pirmas objektas žingsniu 0.4
Antras objektas žingsniu 0.4
Pirmas objektas žingsniu 0.2
Antras objektas žingsniu 0.2
Pirmas objektas žingsniu 0.1
Antras objektas žingsniu 0.1
Pirmas objektas žingsniu 0.01
Antras objektas žingsniu 0.01
Pirmo objekto teisingas ats
Antro objekto teisingas ats


Accuracy Comparison of Numerical Methods
Norm of Absolute Difference at Final Time Point
Step Size
Euler
Euler
Runge-Kutta 4th Order
Runge-Kutta 4th Order