Previous Next

8.8.2 Example: Forward Euler simulation of coffee cooling

MO2.1 MO2.3 MO2.4 MO2.7

The script below is a Forward Euler simulation of coffee cooling and is available for download here. The resulting plot is shown in Figure 8.14.

```
import matplotlib.pyplot as plt
import math
from coffee_model_rev2 import coffeeIVP

def solve_coffee(coffee_IVP, dt):
    # Sets initial condition
    t0 = coffee_IVP.get_tI()
    v0 = coffee_IVP.get_uI()

    t = [t0]
    v = [v0[0]]

    # Get final time
    tF = coffee_IVP.get_tF()

    # Loop from from t=t0 to t>=tF
    tn = t[0]
    vn = v[0]

    while (tn<tF):
        # Calculate forcing
        fn = coffee_IVP.evalf([vn],tn)

        # Update solution and time
        vn1 = vn + dt*fn[0]
        tn1 = tn + dt

        # Append to v and t
        v.append(vn1)
        t.append(tn1)

        # Set vn and tn for next iteration
        vn = vn1
        tn = tn1

    return t, v

mc   = 0.35 # kg
cc   = 4200.0 # J / (kg C)
h    = 5.0 # W/(m^2 C)
A    = 0.04 # m^2
Tout = 25.0 # C

TcI   = 85.0 # Initial temperature of coffee
(C)
tFmin = 700.0 # final time to simulate to
(min)
dtmin = 2.5e1 # time increment to give
solutions at (min)

# Convert times to seconds
```

Discussions

All posts sorted by recent activity

```
tF = tFmin*60
dt = dtmin*60

# Initialize CoffeeIVP object
p = {}
p['h']    = h
p['A']    = A
p['mc']   = mc
p['cc']   = cc
p['Tout'] = Tout

coffeeIVP_hotday = coffeeIVP([TcI], 0.0, tF,
p)

# Solve coffee IVP
t, Tc = solve_coffee(coffeeIVP_hotday, dt)

# Calculate exact solution
u = []

lam  = -h*A/(mc*cc)

for n in range(len(t)):
    ts = t[n]
    t[n] = t[n]/60.0 # convert to minutes
    un = Tout + (TcI-Tout)*math.exp(lam*ts) #
this is the exact solution
    u.append(un)

# Plot
fig, ax = plt.subplots()
ax.scatter(t,Tc,marker='o',label='numerical')
ax.set_xlabel('t (min)')
l
)

ax.scatter(t,u,marker='x',label='exact')
```

Previous Next

Your Privacy Choices

Connect

Blog
Contact Us
Help Center
Security
Media Kit