HW8

March 22, 2024

0.1 8.3

[1]:
```
%%capture
from scipy.optimize import root_scalar
import matplotlib.pyplot as plt
import numpy as np
```

[53]:
```
class DriverConfig:
  def __init__(self, driver_temp, driver_gamma, driver_pressure,␣
 ↪driven_pressure, driver_molar_mass):
    self.driver_temp = driver_temp
    self.driver_gamma = driver_gamma
    self.driver_pressure = driver_pressure
    self.driven_pressure = driven_pressure
    self.driver_molar_mass = driver_molar_mass

    self.shock_mach = 0
    self.u_contact = 0
    self.p_two = 0

  def driver_R(self):
    return 8.31446261815324 / self.driver_molar_mass
```

[5]:
```
driven_gamma = 1.4
driven_temp = 295
driven_R = 287.052874
```

0.1.1 i-iii

[3]:
```
driver_configs = []
driver_configs.append(DriverConfig(295, 5/3, 1.2e6, 50e3, 4.002602e-3))
driver_configs.append(DriverConfig(2500, 1.5, 650e3, 100, 7.5e-3))
driver_configs.append(DriverConfig(4600, 5/3, 110e6, 100e3, 5.6e-3))
```

0.1.2 b

[34]:
```
for ind, conf in enumerate(driver_configs):
  print("config", "".join(["i"] * (ind + 1)))
  lhs = conf.driver_pressure / conf.driven_pressure

  a_ratio = (driven_gamma / conf.driver_gamma * driven_R / conf.driver_R() *␣
 ↪driven_temp / conf.driver_temp) ** 0.5

  def rhs(shock_mach):
    out = 1 + 2 * driven_gamma / (driven_gamma + 1) * (shock_mach ** 2 - 1)
    out /= (1 - (conf.driver_gamma - 1) / (driven_gamma + 1) * a_ratio *␣
 ↪(shock_mach ** 2 - 1) / shock_mach) ** (2 * conf.driver_gamma / (conf.
 ↪driver_gamma - 1))
    return out

  def equation(shock_mach):
      return rhs(shock_mach) - lhs




  result = root_scalar(equation, bracket=[0.1, 25], x0 = 10, method='secant')
  shock_mach_value = result.root

  print(shock_mach_value)
  conf.shock_mach = shock_mach_value
```

```
config i
2.5714086448254747
config ii
12.738545450126754
config iii
11.528272193820246
```

0.1.3 c

[56]:
```
for ind, conf in enumerate(driver_configs):
  print("config", "".join(["i"] * (ind + 1)))

  a_one = (driven_gamma * driven_R * driven_temp) ** 0.5
  a_four = (conf.driver_gamma * conf.driver_R() * conf.driver_temp) ** 0.5

  u_contact = a_one * 2 * (conf.shock_mach ** 2 - 1) / (driven_gamma + 1) /␣
 ↪conf.shock_mach

  conf.u_contact = u_contact
```

```
  mach_exp = ((a_four / u_contact) - 1) / ((conf.driver_gamma - 1) / 2)

  print("Mach of Three", mach_exp, "which is to the", "left" if mach_exp < 0␣
 ↪else "right")
```

```
config i
Mach of Three 1.8413916032575262 which is to the right
config ii
Mach of Three -1.754814871866512 which is to the left
config iii
Mach of Three 0.08310866625922707 which is to the right
```

0.1.4 d

[54]:
```
from __init__ import NormalShockRatio, normal_shock

for ind, conf in enumerate(driver_configs):
  print("config", "".join(["i"] * (ind + 1)))

  shock_ratio = normal_shock(conf.shock_mach, driven_gamma)

  pressure = conf.driven_pressure * shock_ratio.pressure_ratio
  temp = driven_temp * shock_ratio.temp_ratio

  conf.p_two = pressure

  a_two = (driven_gamma * driven_R * temp) ** 0.5

  mach_two = u_contact / a_two

  print("Pressure: {:.3g}, Temperature: {:.3g}, Mach Number: {:.3g}".
 ↪format(pressure, temp, mach_two))
```

```
config i
Pressure: 1.19e+05, Temperature: 510, Mach Number: 7.25
config ii
Pressure: 278, Temperature: 762, Mach Number: 5.93
config iii
Pressure: 2.77e+05, Temperature: 752, Mach Number: 5.97
```

0.1.5 e

[55]:
```
for ind, conf in enumerate(driver_configs):
  print("config", "".join(["i"] * (ind + 1)))

  pressure_ratio = 1 + 2 * (conf.p_two - conf.driven_pressure) / conf.
 ↪driven_pressure
```

```
    pressure_ratio *= (1 + (3 * driven_gamma - 1) / 2 / (driven_gamma + 1) *␣
    ↪(conf.shock_mach ** 2 - 1))
    pressure_ratio /= 1 + (driven_gamma - 1) / (driven_gamma + 1) * (conf.
    ↪shock_mach ** 2 - 1)

    temp_ratio = 1 + 4 * (driven_gamma - 1) / (driven_gamma + 1) * (conf.
    ↪shock_mach ** 2 - 1) / conf.shock_mach
    temp_ratio *= 1 + (3 * driven_gamma - 1) / 2 / (driven_gamma + 1) * (conf.
    ↪shock_mach ** 2 - 1)

    pressure = conf.driven_pressure * pressure_ratio
    temp = driven_temp * temp_ratio

    print("Pressure: {:.3g}, Temperature: {:.3g}".format(pressure, temp))
```

```
config i
Pressure: 4.61e+05, Temperature: 3.43e+03
config ii
Pressure: 1.78e+03, Temperature: 3.02e+05
config iii
Pressure: 1.75e+06, Temperature: 2.26e+05
```