Dynamic Pressure with Changing Altitude

Author: Alex Kenan

Date: 26 April 2017

Imports

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

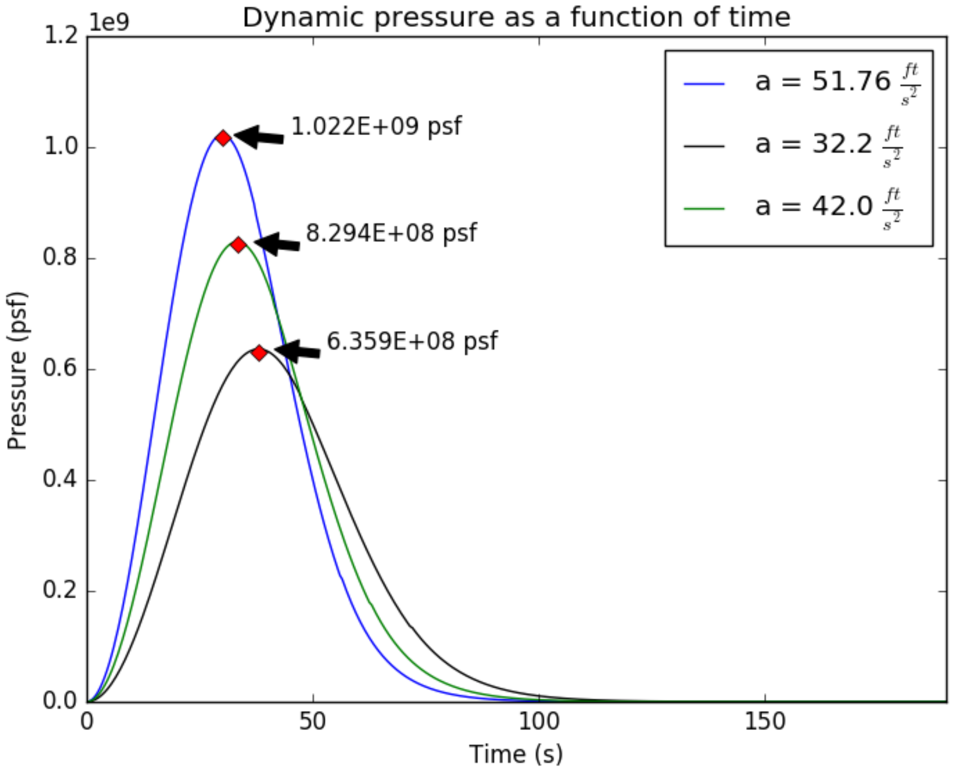
Equations

```
def density(height: float) -> float:
    """
    Returns the air density in slug/ft^3 based on altitude
    Equations from https://www.grc.nasa.gov/www/k-12/rocket/atmos.html
    :param height: Altitude in feet
    :return: Density in slugs/ft^3
    """
    if height < 36152:
        T = 59 - 0.00356 * height
        rho = 2116 * ((T + 459.7)/518.6)**5.256
    elif 36152 <= height < 82345:
        rho = 473.1*np.exp(1.73 - 0.000048*height)
    else:
        T = -205.05 + 0.00164 * height
        rho = 51.97*((T + 459.7)/389.98)**-11.388

    return rho


def velocity(time: float, acceleration: float) -> float:
    """
    Convert time to velocity using Vf = Vi + at
    (where Vf = final velocity,
    Vi = initial velocity, [0 in this case]
     a = acceleration, t = time
    :param time: time in seconds
    :param acceleration: acceleration in ft/s^2
    :return: velocity in ft/s
    """
    return acceleration*time


def altitude(time: float, acceleration: float) -> float:
    """
    Convert time to altitude using the constant acceleration equation
    x = vi*t + 0.5*a*t^2, where vi = 0 in this case
    :param time: Time in seconds
    :param acceleration: acceleration in ft/s^2
    :return: Altitude in feet
    """
    return 0.5*acceleration*time**2
```

Inputs

```
if __name__ == '__main__':
    y_values = []
    x_values = np.arange(0, 511, 0.5)
    for value in x_values:
        # Dynamic pressure q = 0.5*rho*V^2
        '''
        Acceleration is the average acceleration
        to go from 0 ft/s to 26,400 ft/s
        (18,000 mph) in 8.5 minutes = 51.764705882 ft/s^2
        '''
        accel = 51.764705882
        alt = altitude(value, accel)
        q = 0.5*density(alt)*velocity(value, accel)**2
        y_values.append(q)

    plt.plot(x_values, y_values, 'b-',
             label=r"a = 51.76 $\frac{ft}{s^2}$")
    max_val = max(y_values)
    ind = y_values.index(max_val)

    # Plot an arrow and text with the max value
    plt.annotate('{:.3E} psf'.format(max_val),
                 xy=(x_values[ind] + 2, max_val),
                 xytext=(x_values[ind] + 15, max_val + 1E5),
                 arrowprops=dict(facecolor='black', shrink=0.05),
                 )
    # Put a dot on the max value
    plt.plot(['{}'.format(x_values[ind])],
             ['{}'.format(max_val - 0.5E7)], 'rD')

    # Now let's make acceleration = 32.2 ft/s^2
    y2_values = []
    for value in x_values:
        accel = 32.2
        alt = altitude(value, accel)
        q = 0.5*density(alt)*velocity(value, accel)**2
        y2_values.append(q)

    plt.plot(x_values, y2_values, 'k-',
             label=r"a = 32.2 $\frac{ft}{s^2}$")
    max_val = max(y2_values)
    ind = y2_values.index(max_val)

    # Plot an arrow and text with the max value
    plt.annotate('{:.3E} psf'.format(max_val),
                 xy=(x_values[ind] + 3, max_val),
                 xytext=(x_values[ind] + 15, max_val + 1E5),
                 arrowprops=dict(facecolor='black', shrink=0.05),
                 )
    # Put a dot on the max value
    plt.plot(['{}'.format(x_values[ind])],
             ['{}'.format(max_val - 0.5E7)], 'rD')
```

```
# Now let's make acceleration the average of the two: 42 ft/s^2
y3_values = []
for value in x_values:
    accel = 42.0
    alt = altitude(value, accel)
    q = 0.5 * density(alt) * velocity(value, accel) ** 2
    y3_values.append(q)

plt.plot(x_values, y3_values, 'g-',
         label=r"a = 42.0 $\frac{ft}{s^2}$")
max_val = max(y3_values)
ind = y3_values.index(max_val)

# Plot an arrow and text with the max value
plt.annotate('{:.3E} psf'.format(max_val),
             xy=(x_values[ind] + 3, max_val),
             xytext=(x_values[ind] + 15, max_val + 1E5),
             arrowprops=dict(facecolor='black', shrink=0.05),
             )
# Put a dot on the max value
plt.plot(['{}'.format(x_values[ind])],
         ['{}'.format(max_val - 0.5E7)], 'rD')

plt.xlim(0, 190)
plt.xlabel(r'Time (s)')
plt.ylabel('Pressure (psf)')
plt.title(r'Dynamic pressure as a function of time')
plt.legend()
plt.show()
```



Published from maxq.py (maxq.py) using Pweave (http://mpastell.com/pweave) 0.25 on 26-04-2017.