Due Thursday, May 20th at 11:59pm

Problem 1

The United States census estimates the population of the US every decade. These population estimates are recorded in the file `population.csv`. You can load this file with the code

```
data = readmatrix('population.csv');
t = data(1, :);
N = data(2, :);
```

in MATLAB, or

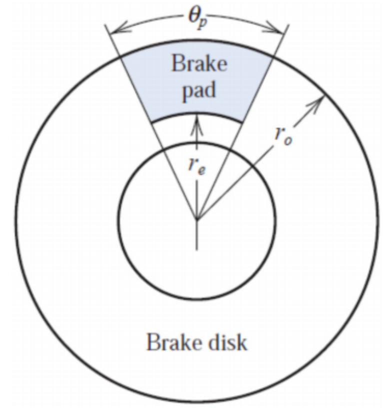
```
data = np.genfromtxt('population.csv', delimiter=',')
t = data[0, :]
N = data[1, :]
```

in python. The variables `t` and `N` will be 1×24 vectors in MATLAB and 1D arrays with 24 entries in python. The entries of `t` represent time in years since 1900. For example, $t = 0$ represents the year 1900, $t = -110$ represents the year 1790 and $t = 120$ represents the year 2020. The times are all evenly spaced. The entries of `N` represent the population of the United States in millions at the corresponding year. (That is, the first entry in `N` is the population at the first time in `t`, the second entry in `N` is the population at the second time in `t`, etc.)

Presumably, the population of the US is a function of time $N(t)$, but we do not know the formula. All we know is the data in `N`. In this problem, we are going to estimate the growth rate of the US population by approximating $\mathrm{d}N/\mathrm{d}t$ at different times. **You should use a second order approximation for all of these derivatives.** In some cases, you will have to use a second order forward or backward difference scheme, but **whenever possible you should use a second order central difference scheme**. You should be able to determine the correct Δt from the data.

(1) Estimate $\frac{\mathrm{d}N}{\mathrm{d}t}$ at the year 2020 and save your approximation in a variable named `A1`.
(2) Estimate $\frac{\mathrm{d}N}{\mathrm{d}t}$ at the year 1880 and save your approximation in a variable named `A2`.
(3) Estimate $\frac{\mathrm{d}N}{\mathrm{d}t}$ at the year 1790 and save your approximation in a variable named `A3`.
(4) Estimate $\frac{\mathrm{d}N}{\mathrm{d}t}$ at every year in `t`. Save your answers in a 1×24 row vector (don't forget to use reshape in python) named `A4`. Your answers should be in the same order as the data, so the kth entry of `A4` should be the derivative at the time in the kth entry of `t`.
(5) The per capita growth rate at time t is given by $\frac{N'(t)}{N(t)}$. Use your approximations of the derivative to calculate the per capita growth rate at every year in `t`. Save your answers in a 1×24 row vector (don't forget to use reshape in python) named `A5`. Your answers should be in the same order as the data, so the kth entry of `A5` should be the growth rate at the time in the kth entry of `t`.
(6) Find the average of all of these per capita growth rates and save it in a variable named `A6`.

Problem 2



Brake pads tend to get extremely hot and it is important to be able to accurately predict their temperature. As a first approximation, we can assume that a brake pad is a section of an annulus, as shown in the following diagram:

Here, r_e represents the radius at which the pad-disc contact begins, r_o represents the radius at which the contact ends and θ_p represents the angular width of the brake pad.

The disc (attached to your vehicle's wheel) is moving faster farther from the center, and the brake pad therefore gets hotter as you move toward the outside edge. If we let $T(r)$ denote the temperature of the brake pad at radius r, then the average temperature of the brake pad is given by

$$\overline{T} = \frac{T_{\text{total}}}{A} = \frac{\int_{r_e}^{r_o} rT(r)\theta_p \, \mathrm{d}r}{\int_{r_e}^{r_o} r\theta_p \, \mathrm{d}r}.$$

The exact formula for $T(r)$ is difficult to obtain, but it is easy to measure the temperature at various points on any given brake pad. In particular, the file `brake_pad.csv` contains data from such a set of measurements. You can load it with the code

```
data = readmatrix('brake_pad.csv');
r = data(1, :);
T = data(2, :);
```

in MATLAB, or

```
data = np.genfromtxt('brake_pad.csv', delimiter=',')
r = data[0, :]
T = data[1, :]
```

in python. The variables `r` and `T` will be 1×11 vectors in MATLAB and 1D arrays with 11 entries in python. The variable `r` contains evenly spaced radius values (in feet) and the vector T contains corresponding temperatures (in degrees Fahrenheit). The brake pad that this data came from has $r_e = 0.308$, $r_o = 0.478$ and $\theta_p = 0.7051$. You should be able to determine the appropriate value of Δr from the data.

In this problem, you will use several different methods to approximate the two integrals in this formula.

(1) Calculate the integral T_{total} using the left hand rule. Save this approximation in a variable `A7`. Then calculate the integral A using the left hand rule, and approximate $\overline{T}$ by dividing your approximations for T_{total} and A. Save the resulting approximation of $\overline{T}$ in a variable named `A8`.

(2) Repeat part (a) using the right hand rule for both integrals. Save your approximation for T_{total} in a variable named `A9` and save your approximation for $\overline{T}$ in a variable named `A10`.

(3) Repeat part (a) using the trapezoidal method for both integrals. Save your approximation for T_{total} in a variable named `A11` and save your approximation for $\overline{T}$ in a variable named `A12`.

1. Problem 3

The following integral arises in the study of reaction-diffusion equations. It describes the width of repeating patterns in periodic steady state solutions.

$$T(\mu) = \int_0^1 \frac{\mu}{\sqrt{F(\mu) - F(\mu z)}} \, \mathrm{d}z.$$

Here, $0 < \mu < 1$ and $F(x) = x^2/2 - x^3/3$. (Notice that the variable of integration is z, not μ, so μ is just a fixed constant in the integrand.)

(1) Find $T(0.95)$ using the function `integral` (in MATLAB) or `quad` (in python). Save your answer in a variable named `A13`.

(2) Find $T(0.5)$ using the function `integral` (in MATLAB) or `quad` (in python). Save your answer in a variable named `A14`.

(3) Find $T(0.01)$ using the function `integral` (in MATLAB) or `quad` (in python). Save your answer in a variable named `A15`.