Exercise 1: Population Growth and Numerical Differentiation
Consider the U.S. population for years 1810-2010 listed below:

Calendar Year	Years since 1900, t	Population in millions, $N(t)$
1810	-90	7.24
1820	-80	9.64
1830	-70	12.87
1840	-60	17.07
1850	-50	23.19
1860	-40	31.44
1870	-30	38.56
1880	-20	50.19
1890	-10	62.98
1900	0	76.21
1910	10	92.23
1920	20	106.02
1930	30	123.20
1940	40	132.16
1950	50	151.33
1960	60	179.32
1970	70	203.30
1980	80	226.54
1990	90	248.71
2000	100	281.42
2010	110	307.75

```
N = [ 7.24; 9.64; 12.87; 17.07; 23.19; 31.44; 38.56; 50.19; 62.98; ...
      76.21; 92.23; 106.02; 123.20; 132.16; 151.33; 179.32; 203.30; ...
      226.54; 248.71; 281.42; 307.75];
```

Estimate the growth rate $\frac{dN}{dt}$ (millions/year) for years 1810, 1820, ... 2010 using finite differences of *second order* accuracy. Save

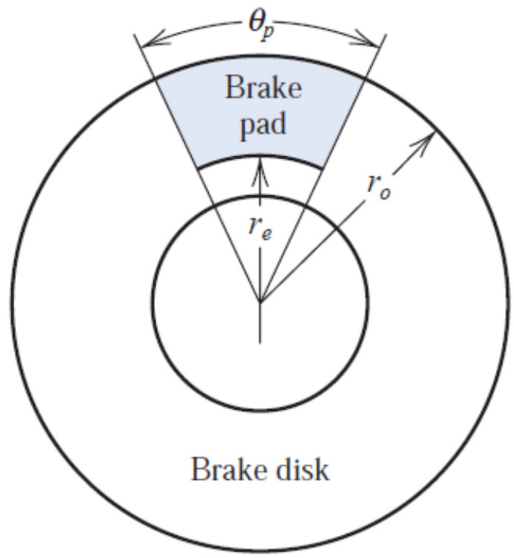
$$\left[\frac{dN}{dt}(t=1810), \frac{dN}{dt}(t=1820), \ldots, \frac{dN}{dt}(t=2010)\right]^T$$

as a column vector in **A1.dat**.
Hint: Second-order forward/backward differences must be used at the end points.

Exercise 2: Brake Pads and Numerical Integration

To simulate the thermal characteristics of disk brakes, D. A. Secrets and R. W. Hornbeck needed to approximate numerically the area-averaged lining temperature, $\overline{T}$ of the brake pad defined by:

$$\overline{T} = \frac{\int_{r_\ell}^{r_0} rT(r)\theta_p \, dr}{\int_{r_\ell}^{r_0} r\theta_p \, dr}$$



where r_ℓ represents the radius at which the pad-disk contact begins, r_0 represents the outside radius of the pad-disk contact, θ_p represents the angle subtended by the sector brake pads, and $T(r)$ is the temperature at each point of the pad, obtained numerically from analyzing the heat equation. Suppose $r_\ell = 0.308$ ft, $r_0 = 0.478$ ft, and $\theta_p = 0.7051$ radians, and the temperatures at equidistant points on the disk are listed in the table below.

r (ft)	T (°F)
.308	640
.325	794
.342	885
.359	943
.376	1034
.393	1064
.410	1114
.427	1152
.444	1204
.461	1222
.478	1239

(a) Calculate $\overline{T}$ using the composite Simpson rule for both the integral in the numerator and the denominator. Save this number as **A2.dat**.

(b) Calculate $\overline{T}$ using Matlab's function `trapz` for both the integral in the numerator and the denominator. Save this number as **A3.dat**.

Exercise 3: Chemical Differential Equations
Consider the chemical reaction

$$X \xrightarrow{k_1[C]} Y.$$

The molecule X is converted to Y. There is a catalyst $[C]$ that controls the rate of the reaction, such that the rate of conversion of X to Y is $k_1[C]$ molecules/second. The catalyst itself, however, is affected by the amount of X present. Namely,

$$[C] = \frac{[X]}{k_2 + [X]}.$$

Finally, suppose we have an external machine that is adding or removing X periodically, where the time-dependent rate is given by $k_3 \sin(\pi t)$. Then the differential equation governing the concentration of $[X]$ is:

$$\frac{d[X]}{dt} = -k_1[C][X] + k_3 \sin(\pi t).$$

Suppose $k_1 = 2$, $k_2 = 0.25$, and $k_3 = 1.5$. Suppose also that we begin with $[X] = 1$ at $t = 0$. We wish to compute $[X]$ as a function of t from $t = 0$ to $t = 1$.

(a) Integrate the differential equation to $t = 1$ using the forward Euler's method with a time step of $h = 0.1$. Save the resulting values

$$([X](t = 0), [X](t = 0.1), \dots, [X](t = 1))^T$$

as a column vector in **A4.dat**.

(b) Integrate the differential equation to $t = 1$ using the fourth order Runge-Kutta method with a time step of $h = 0.1$. Save the resulting values

$$([X](t = 0), [X](t = 0.1), \dots, [X](t = 1))^T$$

as a column vector in **A5.dat**.

(c) Repeat exercises (a) and (b) with a time step of $h = 0.01$. Save the results in a two-column matrix where the first column contains the solution from the forward Euler's method and the second column contains the solution from the fourth order Runge-Kutta method. Save this matrix in **A6.dat**

(d) Integrate the differential equation to $t = 1$ using the Matlab function `ode45` with `tspan=[0;1]` and default settings otherwise. Save the output t and $[X]$, in that order, in a two-column matrix in **A7.dat**.

Try plotting your results on the same plot to see how they compare. No need to submit comments, but explore a bit!

Exercise 4: A Second Order Differential Equation

Consider the damped pendulum equation, which describes the motion of a pendulum that has some impediment hindering its movement (the hinge may need oil, or there might be friction in the air, etc.). The motion of this pendulum is described by:

$$x''(t) = \frac{g}{L}\sin(x(t)) - \delta x'(t),$$

where $x(t)$ is the angle of displacement, g is the acceleration due to gravity, L is the length of the pendulum, and δ is a damping coefficient. Note that this equation is non-linear. However, if we assume that $x \ll 1$, then we can approximate $\sin(x(t))$ with $x(t)$ (think about the Taylor series for $\sin(x)$). This gives us:

$$x''(t) = \frac{g}{L}x(t) - \delta x'(t).$$

Now that the equation is linear, it can be written in terms of a linear system by introducing $v(t) = x'(t)$, i.e.

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} x(t) \\ v(t) \end{bmatrix}.$$

(a) Find the matrix A if $g = -10$, $L = 8$, and $\delta = 3$. Save it in **A8.dat**.

(b) Recall that the forward Euler method for a linear system of differential equations can be written as an iteration:

$$\begin{bmatrix} x_{n+1} \\ v_{n+1} \end{bmatrix} = (I + \Delta t A)\begin{bmatrix} x_n \\ v_n \end{bmatrix},$$

and that the stability of this method depended upon the spectral radius (maximum size of the eigenvalues) of $M = I + \Delta t A$. Use these concepts, and the matrix A found above, to determine the maximum time step $\Delta t_{\max}$ we can take and still have the forward Euler method be stable. You should do this by hand. Save the value you find in **A9.dat**.

(c) Solve the linear system of differential equations, with the matrix A above, using the forward Euler method from $t = 0$ to $t = 50$. For initial conditions, use $[x(0), v(0)]^T = [1, 0]^T$, and use a timestep of $0.8\Delta t_{\max}$. Plot the displacement $x(t)$ to verify that your solution is stable (i.e. it does not blow up to very large values). Store the resulting $x(t)$ in a row vector. Save this as **A10.dat**.

(d) Now solve the same linear system with the same initial conditions using forward Euler again, but use $\Delta t = 1.05\Delta t_{\max}$ instead (only five percent above the maximum). Plot the results to verify that the scheme is no longer stable. Store the resulting $x(t)$ in a row vector. Save this as **A11.dat**.

(e) Solve the same problem with the same initial conditions using the built-in solver `ode45`. As before, solve this from $t = 0$ to $t = 50$. Use default settings otherwise. Save the computed $x(t)$ values as a column vector in **A12.dat**. Save the computed $v(t)$ values as a column vector in **A13.dat**. Try comparing these result with your results from (c) and (d). Does your forward Euler scheme perform better if you reduce the time step even further? Explore further, but do not worry about submitting commentary.