Lab Assignment #12
Due 4/22/2016 at 4pm on bCourses

This assignment will be partially graded by an autograder which will check the values of the required variables, so it is important that your function and variable names are exactly what they are supposed to be and the input and output arguments are in the correct order (names ARE case-sensitive). Instructions for submitting your assignment are included at the end of this document.

1 Thermal Dynamics of a Building

When designing a building, it is important to be able to model how the outdoor ambient temperature influences the temperature inside the building. To save electricity, energy engineers will model several proposed building designs and decide which design would be the most efficient for a given climate. The most important design aspects are often the selection of the HVAC (heating, ventilation, and air conditioning) equipment, the building materials to be used, the orientation of the building, and the window to wall ratio. In this problem, we consider only the building materials and the outdoor ambient temperature to determine the temperature inside a building. Today, well-designed, highly controlled, well-integrated buildings like these are often referred to as "Smart Buildings".

In this question, we will use a simplified model of a building, represented in Figure 1.

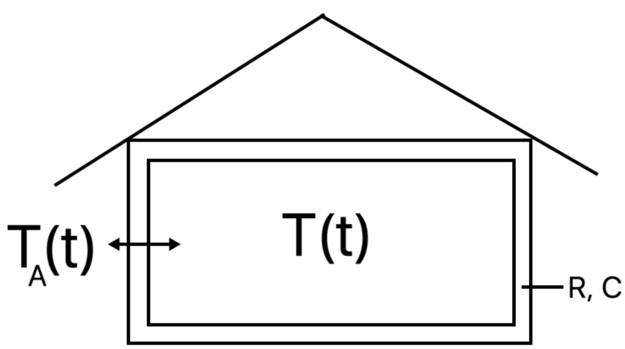

Figure 1: Schematic of Building

Here, t is the time, $T_A(t)$ is the outdoor ambient temperature in degrees Fahrenheit, given by equation 1, $T(t)$ is the indoor temperature in degrees Fahrenheit, R is the thermal resistance of the wall, in $\frac{^\circ\text{F hr}}{\text{Btu}}$, and C is the thermal capacitance of the wall, in $\frac{\text{Btu}}{^\circ\text{F}}$. Btu is a unit of work and stands for British Thermal Unit, as commonly used in the energy industry. For this problem we will represent the diurnal cycle of the ambient outdoor temperature with a Gaussian function:

$$T_A(t) = 60^\circ\text{F} + \frac{10^\circ\text{F hr}}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(t-\mu)^2}{2\sigma^2}\right) \tag{1}$$

where T_A is the ambient temperature in degrees Fahrenheit, t is the time of day in hours (between 0 and 24), $\sigma = 1$ hr, and $\mu = 12$ hr. Figure 2 shows T_A versus time of day.

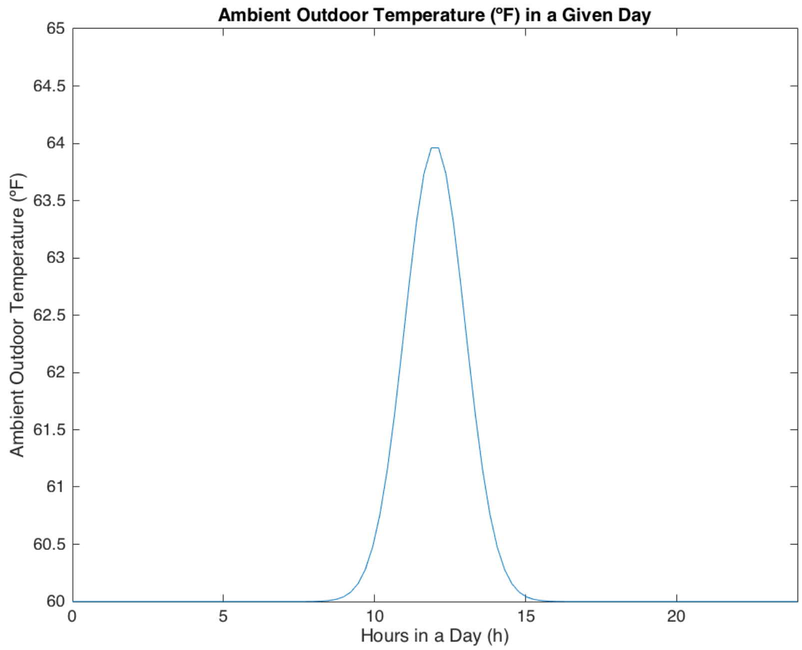

Figure 2: Daily profile of the ambient outdoor air temperature, $T_A(t)$.

We model the thermal dynamics of the building given by the interaction between the ambient temperature and the indoor temperature with the following governing equation:

$$\frac{dT(t)}{dt} = \frac{1}{RC}(T_A(t) - T(t)) \tag{2}$$

given the initial temperature $T(0) = T_0$. We assume that no heating or air conditioning is running, so in this problem the only controllable design aspects are the properties of the building materials, R and C.

You do not have to do any unit conversion for this problem (1.1 and 1.2) as all the quantities will be given with consistent units.

1.1 Euler's Method

In this question, we will use Euler's method to calculate an approximate solution to Equation 2. Using Euler's method, one can estimate the solution of Equation 2 at time t_{i+1} knowing the solution at time t_i:

$$T(t_{i+1}) \approx T(t_i) + \left.\frac{dT}{dt}\right|_{t_i} \Delta t \tag{3}$$

where $\Delta t = t_{i+1} - t_i$ is the time step and $\frac{dT}{dt}|_{t_i}$ is given by Equation 2. In this problem we use a constant time step when using Euler's method (i.e. $\Delta t = t_3 - t_2 = t_2 - t_1 = t_{k+1} - t_k$).

Write a function with the following header:

```
function [T_euler, t_euler] = myEulerApprox(delta_t, T_0, R, C)
```

where `delta_t`, `T_0`, `R`, and `C` are scalars of class `double`. `delta_t` (Δt in Equation 3) is the time step, in units of hours. `T_0` is the initial condition (i.e. temperature $T(t = 0)$ at time zero) in °F inside the building. `R` is the thermal resistance of the wall in $\frac{^{\circ}\mathrm{F\ hr}}{\mathrm{Btu}}$. `C` is the thermal capacitance of the wall in $\frac{\mathrm{Btu}}{^{\circ}\mathrm{F}}$. Your function should use Euler's method to solve Equation 2 over the time period 0 to 24 hours, using a constant time step of `delta_t` hours. The ambient outdoor air temperature T_A is given by Equation 1. The function's outputs `T_euler` and `t_euler` represent the temperature in °F in the building estimated using Euler's method, and the time of day in hours at which temperature is estimated. `T_euler` and `t_euler` should be arrays of class `double` and size $(\frac{24}{\Delta t} + 1) \times 1$. You can assume that `delta_t` is such that the array `0:delta_t:24` contains the value 24.

Your function should match the following test case:

```
>> [T_euler, t_euler] = myEulerApprox(1, 70, 2, 10)

T_euler =

   70.0000
   69.5000
   69.0250
   68.5738
   68.1451
   67.7378
   67.3509
   66.9834
   66.6342
   66.3026
   65.9896
   65.7172
   65.5523
   65.4741
   65.3214
   65.0823
   64.8304
   64.5890
   64.3595
   64.1416
   63.9345
   63.7378
   63.5509
   63.3733
   63.2047
```

```
t_euler =

 0
 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
```

1.2 Matlab's ODE Solver

In this question, we will solve the same ordinary differential equation as in the previous question, but this time using Matlab's `ode45` solver, which uses a higher-order Runge-Kutta formula. Write a function with the following header:

```
function [T, t, range] = myODESolver(delta_t, T_0, R, C)
```

where `delta_t`, `T_0`, `R`, and `C` are the same as for function `myEulerApprox`. Your function should use Matlab's `ode45` solver to solve Equation 2 over the time period 0 to 24 hours, using a constant time step specified by `delta_t`. The function's outputs `T` and `t` represent the temperature in °F in the building estimated by Matlab's `ode45` solver, and the time of day in hours at which temperature is estimated. `T` and `t` should be arrays of class `double` and size $(\frac{24}{\Delta t} + 1) \times 1$. You can assume that `delta_t` is such that the array `0:delta_t:24` contains the value 24. The function's output `range` should be an array of class `double` and size 1×2 which contains the minimum and maximum temperatures (in that order) in °F experienced in the building during the 24-hour simulated time window.

Your function should match the following test case:

```
>> [T, t, range] = myODESolver(2, 65, 2, 10)

T =

   65.0000
   64.5242
   64.0937
   63.7041
   63.3516
   63.0426
   62.9856
   62.9252
   62.6571
   62.4044
   62.1756
   61.9686
   61.7812

t =

     0
     2
     4
     6
     8
    10
    12
    14
    16
    18
    20
    22
    24

range =

   61.7812   65.0000
```

We can compare the result of Matlab's ode45 solver with our results from Euler's method by plugging in the same test values into both functions and plotting them against each other (we will look more into assessing the discrepancies between solution methods in the next problem):

```
>> [T_euler, t_euler] = myEulerApprox(1, 70, 2, 10);
>> [T, t, range] = myODESolver(1, 70, 2, 10);
>> figure;
>> plot(t_euler, T_euler, t, T);
```

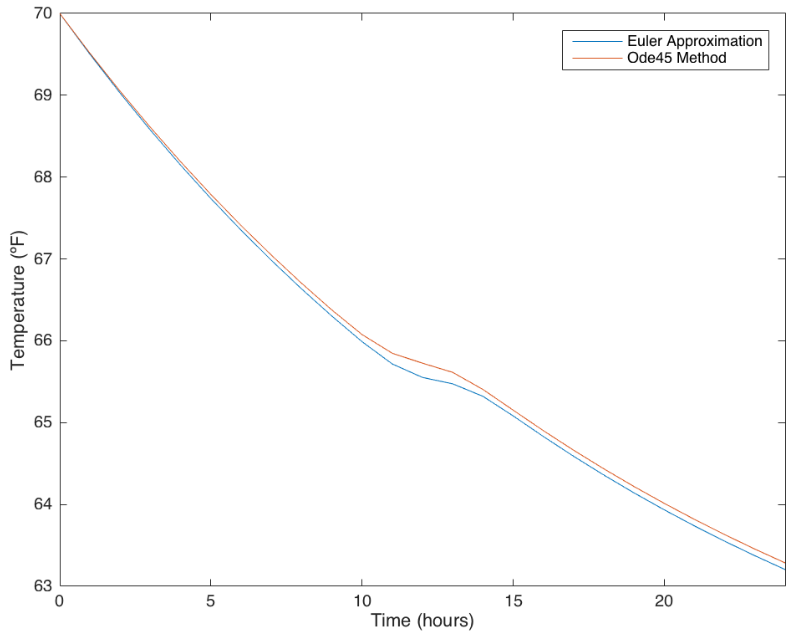

Figure 3: Euler's Method vs. Matlab ODE Solver for problem 1

2 Indoor Air Quality

Particulate matter air pollution is associated with adverse health effects, including respiratory and cardiovascular diseases and cancer. Airborne particulate matter consists of small (from about 1 nm to about 100 μm in diameter) liquid and/or solid particles suspended in air. There are many sources of particulate matter, both indoor and outdoor. Examples of outdoor sources of particulate matter include diesel internal combustion engines and power plants. Examples of indoor sources of particulate matter include cooking and lighted candles.

The particle mass concentration is the mass of particles per unit volume of air. In this problem, we will only consider particles whose diameter is smaller than 2.5 μm. The mass concentration of these particles is often referred to as $PM_{2.5}$ (which stands for "particulate matter $<$ 2.5 μm"). In what follows, we use the term "particle mass concentration" to refer to $PM_{2.5}$ (although in the general case $PM_{2.5}$ is only a part of the total particle mass concentration). In this problem, we will estimate the particle mass concentration inside a house before, during, and after indoor emission events such as cooking.

In the model that we will use, the particle mass concentration in the house depends on: the volume of the house, the ventilation system of the house, the particle mass concentration of the air brought in by the ventilation system, the initial particle mass concentration in the house, the rate of deposition of particles onto surfaces (e.g. onto walls), and the rate of emissions of particles inside the house. Figure 4 represents the different processes that influence the particle mass concentration inside the house.

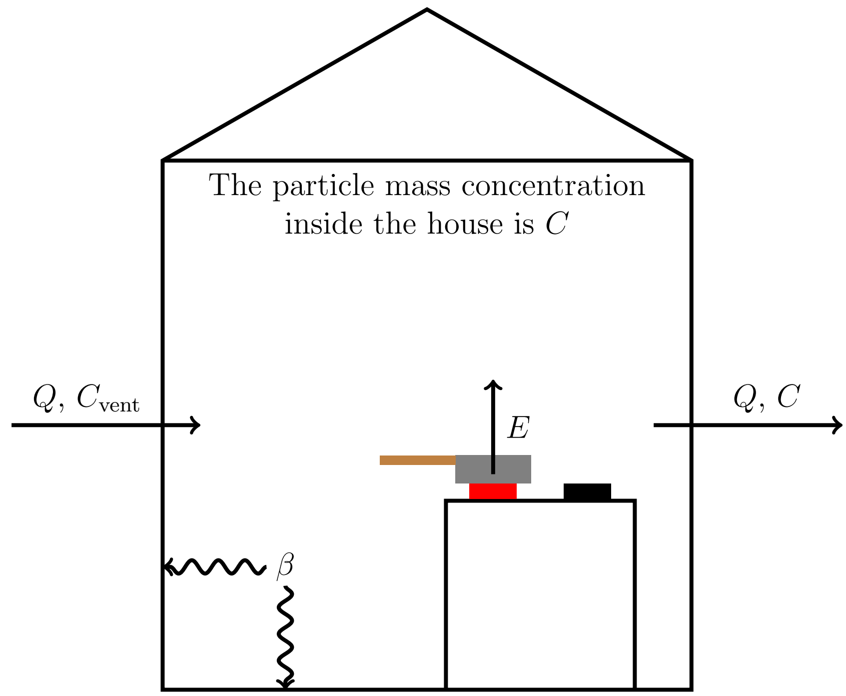

Figure 4: Model representation of a house for analyzing indoor air quality.

We make the following assumptions:

- The air inside the house is well-mixed. A consequence of this assumption is that, at any given time t, the value of the particle mass concentration $C(t)$ is the same everywhere inside the house.
- The ventilation system of the house brings in air from the outside at a constant volumetric flow rate of Q, and expels air from the house to the outside at the same constant volumetric flow rate Q. Volumetric flow rate is expressed as volume/time, and in this problem you can think of this flow as the volume of air entering or leaving the house per minute (in units of m^3/min).
- The particle mass concentration C_{vent} of the air brought into the house by the ventilation system is constant with time.
- The emission rate of particles E from cooking can vary with time t.
- The rate of deposition of particles onto the surfaces (e.g. onto walls) can be described by a first-order loss process with rate constant β, which is assumed constant with time.
- We neglect any process not mentioned above. For example we neglect particle coagulation.
- The volume of air V inside the house is constant.
- The particle mass concentration at the initial time t_0 is C_0.

Using the principle of conservation of mass and the assumptions listed above, one can write the following first-order ordinary differential equation, which describes the time rate of change of particle mass concentration C in the house:

$$\frac{dC}{dt} = \frac{Q}{V}C_{\text{vent}} + \frac{E}{V} - \frac{Q}{V}C - \beta C \tag{4}$$

with the following initial condition:

$$C(t_0) = C_0 \tag{5}$$

Note that in the ordinary differential equation above (Equation 4) and its initial condition (Equation 5):

- C is the unknown function. C is a function of time t;
- Q, V, β, C_{vent}, t_0, and C_0 are known constants; and
- E is a known function of time t.

You do not have to do any unit conversion for this problem (2.1 and 2.2) as all the quantities will be given with consistent units.

2.1 Without indoor emissions

First, we assume that there are no indoor emissions of particulate matter. In other words, we assume that $E(t) = 0$ for all times t. In this case, the analytical solution of Equation 4 given the initial condition (Equation 5) is given by:

$$C(t) = \frac{Q}{V}\tau C_{\text{vent}} + (C_0 - \frac{Q}{V}\tau C_{\text{vent}})e^{-(t-t_0)/\tau} \tag{6}$$

with:

$$\tau = \frac{V}{Q + \beta V} \tag{7}$$

Write a function with the following header:

```
function [t_out, C_ode45, C_analytic, ME, RMSE] = ...
myAirQualityNoEmissions(V, Q, beta, C_vent, t_span, C0)
```

Your function should calculate the particle mass concentration inside the house over a specified period of time. The input and output arguments of the function are described below:

- `V` is a scalar of class `double`. `V` (V in Equation 4) is the volume of the house. The units of `V` will be m^3.
- `Q` is a scalar of class `double`. `Q` (Q in Equation 4) is the volumetric flow rate of air through the house imposed by the ventilation system. The units of `Q` will be $\text{m}^3\ \text{min}^{-1}$.
- `beta` is a scalar of class `double`. `beta` (β in Equation 4) is the rate constant of the first-order deposition of particles onto surfaces (e.g. onto walls). The units of β will be min^{-1}.
- `C_vent` is a scalar of class `double`. `C_vent` (C_{vent} in Equation 4) is the particle mass concentration in the air brought into the house by the ventilation system. The units of `C_vent` will be $\mu\text{g m}^{-3}$.
- `t_span` is a 1×2 vector of class `double`. The two values of `t_span` indicate the initial and final times (in that order) of the period of time over which the particle mass concentration must be calculated. The units of the values of `t_span` will be minutes. Matlab's `ode45` solver will determine at which specific values of times the particle mass concentration in the house will be calculated.
- `C0` (the second character in the variable name is a zero) is a scalar of class `double`. `C0` (C_0 in Equation 5) is the particle mass concentration inside the house at time `t_span(1)`. The units of `C0` will be $\mu\text{g m}^{-3}$.
- `t_out`, `C_ode45`, and `C_analytic` are column vectors of class `double`. These three output arguments should have the same size. `t_out` and `C_ode45` are the vector of times and the corresponding solutions, respectively, returned by the Matlab `ode45` solver when used to solve Equation 4 (with $E = 0$) over the time span specified by `t_span` and with the initial condition specified by `C0`. `C_analytic` is the particle mass concentration inside the house calculated using the analytical solution (Equation 6) for each time specified in the vector `t_out`. The units of `t_out` will be min and the units of `C_ode45` and `C_analytic` will be $\mu\text{g m}^{-3}$.
- `ME` and `RMSE` are scalars of class `double` that contain the mean error (ME) and the root mean squared error ($RMSE$), respectively, between the numerical solution (`C_ode45`) and the analytical solution (`C_analytic`).

$$\text{ME} = \frac{1}{m}\sum_{i=1}^{m}\left(C_{\text{numerical}}(t_i) - C_{\text{analytical}}(t_i)\right) \tag{8}$$

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(C_{\text{numerical}}(t_i) - C_{\text{analytical}}(t_i)\right)^2} \tag{9}$$

In Equations 8 and 9, m is the number of points (in time) for which we have values of the solutions. $C_{\text{numerical}}(t_i)$ and $C_{\text{analytical}}(t_i)$ are the values of the numerical (from `ode45`) and analytical (from Equation 6) solutions at time t_i.

You can test your function using the following test cases, which should produce a figure similar to Figure 5. In these test cases, we consider a 20 m^2 ($\approx$ 215 ft^2) studio with a 2.5 m ($\approx$ 8.2 ft) ceiling height, which is not ventilated very well ($Q = 0.25$ m^3 min^{-1}). We set the value of C_{vent} to 15 μg m^{-3}. We set the initial particle mass concentration inside the studio to a fairly elevated level (45 μg m^{-3}) and see how long it takes for the indoor particle mass concentration to reach equilibrium (i.e. how much time it takes to flush out the elevated indoor particle mass concentration). We set $\beta = 0.01$ min^{-1} and simulate a period of 300 minutes.

```
>> V = 50; Q = 0.25; C_vent = 15; beta = 0.01;
>> t_span = [0, 300]; C0 = 45;
>> [t_out, C_ode45, C_analytic, ME, RMSE] = ...
    myAirQualityNoEmissions(V,Q,beta,C_vent,t_span,C0);

% Check the numerical values in your outputs against these:
>> t_out(1:5)'
    0     3.7678     7.5357    11.3035    15.0713
>> C_ode45(1:5)'
    45.0000    42.8020    40.7248    38.7617    36.9065
>> C_analytic(1:5)'
    45.0000    42.8020    40.7248    38.7617    36.9065
>> ME
    -2.6665e-06
>> RMSE
    9.9684e-05

>> plot(t_out, C_analytic, 'b-', t_out, C_ode45, 'ro')
>> xlabel('Time (minutes)')
>> ylabel('Particle mass concentration ({\mug} m^{-3})')
```

We now increase the ventilation rate in the studio ($Q = 2$ m^3 min^{-1}) to simulate a well-ventilated studio, and see how this change influences the time it takes to flush out the elevated indoor particle mass concentration.

```
>> Q = 2;
>> [t_out, C_ode45, C_analytic, ME, RMSE] = ...
    myAirQualityNoEmissions(V,Q,beta,C_vent,t_span,C0);

% Check the numerical values in your outputs against these:
>> t_out(1:5)'
    0     1.3701     2.7402     4.1104     5.4805
>> C_ode45(1:5)'
```

```
   45.0000    42.8150    40.7746    38.8694    37.0904
>> C_analytic(1:5)'
   45.0000    42.8150    40.7747    38.8694    37.0904
>> ME
   -6.5186e-05
>> RMSE
    0.0023

>> hold on
>> plot(t_out, C_analytic, 'b--', t_out, C_ode45, 'rs')
```

Finally, we set the initial particle mass concentration in the studio to zero. In this scenario, the particle mass concentration in the studio increases with time.

```
>> C0 = 0;
>> [t_out, C_ode45, C_analytic, ME, RMSE] = ...
    myAirQualityNoEmissions(V,Q,beta,C_vent,t_span,C0);

% Check the numerical values in your outputs against these:
>> t_out(1:5)'
    1.0e-03 *
         0    0.0837    0.1675    0.2512    0.3349
>> C_ode45(1:5)'
    1.0e-03 *
         0    0.0502    0.1005    0.1507    0.2009
>> C_analytic(1:5)'
    1.0e-03 *
         0    0.0502    0.1005    0.1507    0.2009
>> ME
    2.6542e-05
>> RMSE
    3.4706e-04

>> plot(t_out, C_analytic, 'b:', t_out, C_ode45, 'r^')
```

You do not have to submit any answer to the following questions to complete this assignment, but you should think about the answers to improve your understanding of first-order ordinary differential equations:

- On which parameter(s) does the final particle mass concentration (i.e. $C(t)$ as $t \to +\infty$) depend? To answer this question, you can take the limit of Equation 6 as $t \to +\infty$. You can also check your answer with Figure 5.
- Which parameter(s) control(s) how rapidly the particle mass concentration converges to its final value?

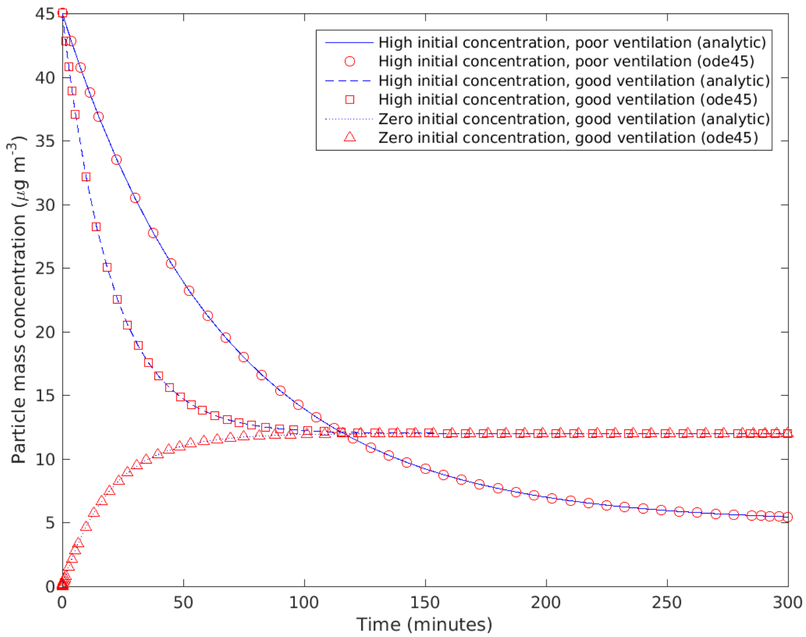

Figure 5: Figure created by the test cases of Problem 2.1 (plus a legend box).

2.2 With indoor emissions

We now wish to analyze the impact of indoor emissions of particulate matter on the particle mass concentration inside the house. We therefore no longer require that $E(t) = 0$ for all times t. Depending on the form of the function E, the analytical solution of Equation 4 together with its initial condition (Equation 5) might not be known. Numerical methods are often used to find approximate solutions of ordinary differential equations when the analytical solution is unknown. Here we will model the emission rate function as a top-hat function, defined by the three parameters t_{start}, t_{end}, and E_{value} as (see Figure 6 for an illustration):

$$E(t) = \begin{cases} E_{\text{value}} & \text{if } \; t_{\text{start}} < t \leq t_{\text{end}} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Write a function with the following header:

```
function [t_out, C_ode45] = myAirQuality(V, Q, beta, C_vent, t_span, ...
    C0, t_start, t_end, E_value)
```

where:

- `V`, `Q`, `beta`, `C_vent`, `t_span`, and `C0` are the same as in Problem 2.1.

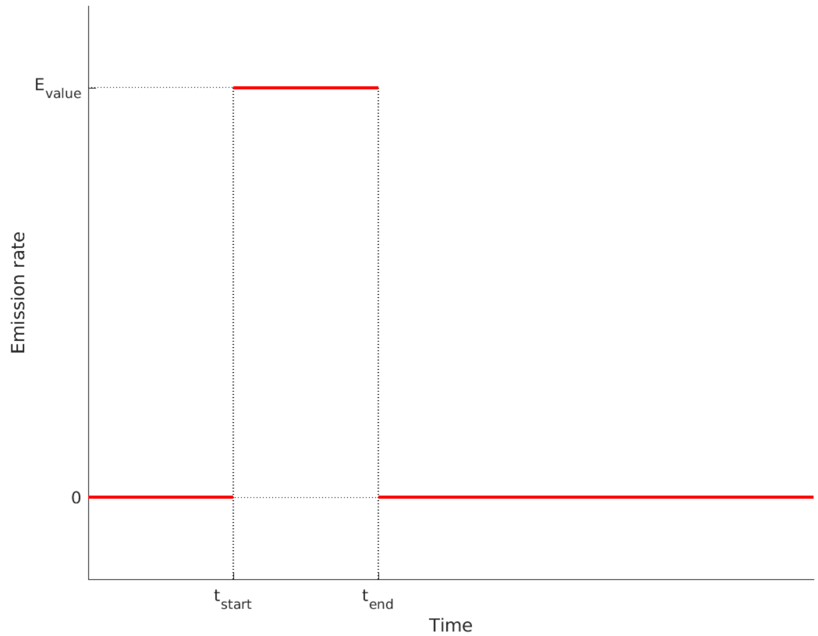

Figure 6: An illustration of the top-hat emission function used in Problem 2.2.

- `t_start`, `t_end`, and `E_value` are scalars of class `double` that represent t_{start}, t_{end}, and E_{value} in Equation 10, respectively. The value of `E_value` will be given in units of μg min^{-1}.
- `t_span` can have one of two formats (your function should work for either):
  - The same as in Problem 2.1. In this case, Matlab's `ode45` solver will determine at which specific values of times the particle mass concentration in the house will be calculated.
  - A $N \times 1$ array of class `double` with $N > 2$ that contains the values of the times (in minutes) at which Matlab's `ode45` solver should calculate the particle mass concentration.
- `t_out` and `C_ode45` are column vectors of class `double`. These two output arguments should have the same size. `t_out` and `C_ode45` are the vector of times and the corresponding solutions, respectively, returned by the Matlab function `ode45` when used to solve Equation 4 for times specified by `t_span`, with the initial condition specified by `C0`, and with the emission rate defined as the top-hat function discussed above.

Hint: you may want to define a (sub)-function to calculate the emission rate $E(t)$ for a given time t and parameters t_{start}, t_{end}, and E_{value}.

You can find the function `myAirQualityAnalytic.m` (on bCourses) that calculates the analytical solution of Equation 4 together with its initial condition (Equation 5) when E is the top-hat function defined in Equation 10, with the additional assumption that $t_{\text{start}} > t_0$. See the comments in `myAirQualityAnalytic.m` and the test cases below for more information about how to use this function, if you decide to use it (optional). It is provided to you if you want to further investigate the performance of Matlab's `ode45` solver.

You can test your function using the following test cases, which should produce a figure similar to Figure 7. We start with the same studio apartment as in the first test case of Problem 2.1 when it is not ventilated very well ($Q = 0.25$ m^3 min^{-1}). We set C_{vent} to the same value as before. We set the initial particle mass concentration inside the house equal to C_{vent}. We add an emission event with $t_{\text{start}} = 30$ min, $t_{\text{end}} = 45$ min, and $E_{\text{value}} = 100$ μg min^{-1} to simulate a 15-minute cooking event.

```
>> V = 50; Q = 0.25; C_vent = 15; beta = 0.01; t_span = [0, 300];
>> C0 = C_vent; E_start = 30; E_end = 45; E_value = 100;

>> [t_out, C_ode45] = myAirQuality(V, Q, beta, C_vent, t_span, ...
    C0, E_start, E_end, E_value);

>> t = linspace(t_span(1), t_span(2), 1000)';
>> C_analytic = myAirQualityAnalytic(V, Q, beta, C_vent, t_span(1), ...
    C0, E_start, E_end, E_value, t);

% Check the numerical values in your outputs against these:
>> t_out(10:15)'
    26.7506    27.7471    28.7437    29.7403    30.7369    31.7335
>> C_ode45(10:15)'
    11.6948    11.5954    11.4976    11.4012    12.3755    14.3667

>> plot(t, C_analytic, 'b-', t_out, C_ode45, 'ro')
>> ylim([0, 40])
>> xlabel('Time (minutes)')
>> ylabel('Particle mass concentration ({\mu}g m^{-3})')
```

We now reduce the duration of the emission episode to 10 minutes.

```
>> E_end = 40;
>> [t_out, C_ode45] = myAirQuality(V, Q, beta, C_vent, t_span, ...
    C0, E_start, E_end, E_value);
>> C_analytic = myAirQualityAnalytic(V, Q, beta, C_vent, t_span(1), ...
    C0, E_start, E_end, E_value, t);

% Check the numerical values in your outputs against these:
>> t_out(10:15)'
    57.5951    65.0951    72.5951    80.0951    87.5951    95.0951
>> C_ode45(10:15)'
    9.2150     8.7665     8.3658     8.0077     7.6876     7.4016

>> hold on
```

```
>> plot(t, C_analytic, 'b--', t_out, C_ode45, 'rs')
```

Note that for the above test case, Matlab's ode45 solver fails remarkably in approximating the solution to the differential equation (see Figure 7). We can partially fix this issue here by forcing Matlab's `ode45` solver to use more time steps, which helps the solver pick up the abrupt change in emissions due to the top hat function used:

```
>> t_ode45 = linspace(t_span(1), t_span(2), 301)';
>> [t_out, C_ode45] = myAirQuality(V, Q, beta, C_vent, t_ode45, ...
    C0, E_start, E_end, E_value);
>> plot(t_out, C_ode45, 'r.')

% Check the numerical values in your outputs against these:
>> t_out(40:45)'
    39    40    41    42    43    44
>> C_ode45(40:45)'
    27.7756   29.1628   29.9508   30.2649   30.2211   29.9219
```

We now increase the ventilation rate in the studio ($Q = 2$ m^3 min^{-1}) to simulate a well-ventilated studio, and see how this change affects the particle mass concentration in the studio.

```
>> Q = 2;
>> [t_out, C_ode45] = myAirQuality(V, Q, beta, C_vent, t_span, ...
    C0, E_start, E_end, E_value);
>> C_analytic = myAirQualityAnalytic(V, Q, beta, C_vent, t_span(1), ...
    C0, E_start, E_end, E_value, t);

% Check the numerical values in your outputs against these:
>> t_out(10:15)'
    28.7770   29.1623   29.5475   29.9327   30.3180   30.7032
>> C_ode45(10:15)'
    12.7125   12.6989   12.6856   12.6725   13.0723   13.8622

>> plot(t, C_analytic, 'b:', t_out, C_ode45, 'r^')
```

Figure 7 illustrates how the solution calculated by Matlab's `ode45` solver can differ from the analytical solution, especially when the solution and/or its derivative vary rapidly with time. You should try to understand the results that are shown on Figure 7.

You do not have to submit any answer to the following questions to complete this assignment, but you should think about the answers to improve your understanding of first-order ordinary differential equations:

- How does the peak concentration in the studio change when the strength of the ventilation system (Q) changes? What about when the duration or strength of the emission

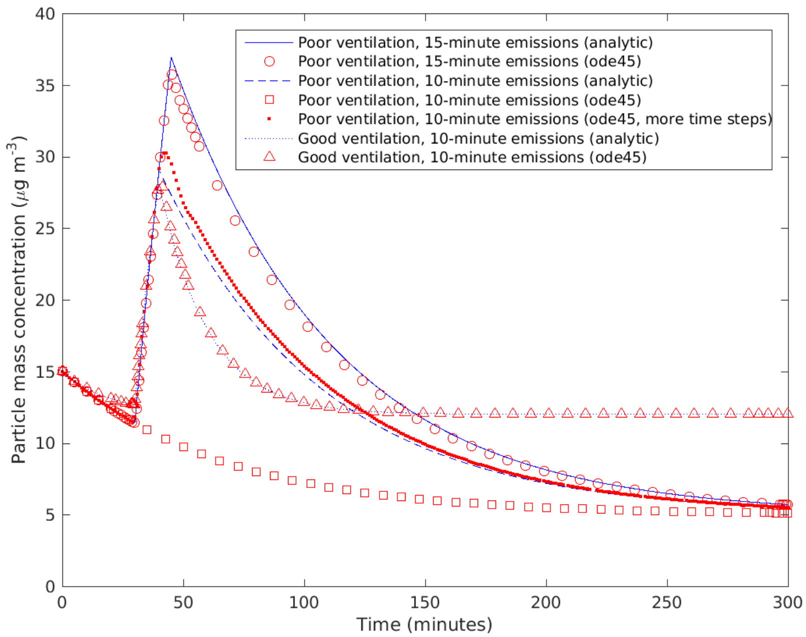

Figure 7: Figure created by the test cases of Problem 2.2 (plus a legend box).

episode vary?

- The solution of Equation 4 together with its initial condition (Equation 5) with E defined as a top-hat function can be derived analytically. Try to derive it. **Hint:** write three separate solutions: a solution that works for the time interval $t \leq t_{\text{start}}$, then a solution that works for the time interval $t_{\text{start}} < t \leq t_{\text{end}}$, and then a solution that works for the time interval $t > t_{\text{end}}$.

Submission Instructions

Your submission should include the following function files in a single zip file named Lab12.zip

- `myEulerApprox.m`
- `myODESolver.m`
- `myAirQualityNoEmissions.m`
- `myAirQuality.m`