Instructions for coding FluxI(t) into an M-file

To define your vector \( \text{FluxI} \), that contains the values of \( \text{FluxI} \) for each time point in the time vector \( t \) produced by the \textit{ode45} function, you need to fill in the values of \( \text{FluxI} \) one by one. The time vector \( t \) is the vector of time points for which \textit{ode45} finds solutions for \( H_a \) and \( H_p \). Your \( \text{FluxI} \) vector will have the same size as the \( t \) vector.

Remember that your conditions for \( \text{FluxI} \) are the following:

- \( \text{FluxI} = 0 \), outside the irrigation period
- \( \text{FluxI} = E_o \cdot C_{ag} \cdot f_{ag} \), within the irrigation period

Use a "for loop" to march through each element of \( \text{FluxI} \) and specify its value, according to the value of the corresponding element in the time vector \( t \).

An example code for defining \( \text{FluxI} \) follows (the sentences following a % are comments):

```matlab
n=size(t)          %  determines the size of the t vector.
for i=1:n
    if t(i)>=t1 & t(i)<=t2    % t1 is the day at which irrigation begins and t2 is the day at which the irrigating season ends.
        FluxI(i)= Eo*Cag*fag;
    else
        FluxI(i)= 0;
    end
end
```

Since \( E_o \), \( C_{ag} \), and \( f_{ag} \) are inputs for computing \( \text{FluxI} \), you need to have defined them before using them in the \( \text{FluxI} \) equation. \( C_{ag} \) and \( f_{ag} \) are easily defined as constants. \( E_o \), however, may be a function of time, depending on how you modeled it in part (f-1). So, once again, you need to specify the value for each element of the \( E_o \) vector depending on the value of the \( t \) element to which it corresponds. Just as in the example code for \( \text{FluxI} \), use a “for loop” that steps through all the elements of \( E_o \) from \( i=1 \) to \( n=\text{size}(t) \), checks the value of \( t(i) \), and correspondingly assigns a value for \( E_o(i) \).