\[ [t, y] = \text{ode45('bangladesh', [0 \ t \text{lim}], [4 \ 4])}; \]

\textbf{function} \ dydt = \text{bangladesh}(t, y)

Defines your system of equations for \( \frac{dH_a}{dt} \) & \( \frac{dH_p}{dt} \)
[t,y]=ode45('bangladesh',[0 tlim], [4 4]);

function dydt=bangladesh(t,y)

Hr=(2.5*\sin(pi+(t/210)*pi))+4;

dydt=[kp*(y(2)-y(1))-Cp*Eo;
     (1/S)*(kp*fp*(y(1)-y(2))+kr*fr*(Hr-y(2))-fag*Cag*Eo)];

dydt @ t=0

ode45
ode45 chooses \( \Delta t \) based on magnitude of dydt

\[
\Delta H_a = \frac{dH_a}{dt} \cdot \Delta t
\]

Update \( t \to t+\Delta t \)

\( H_a @ t=0 = 4 \)

Update \( H_a \to H_a + \Delta H_a \)

repeat starting with the new values of \( t \) and \( H_a \), by inputing them back into the function and calculating new values of dydt
ode45 now inputs the **new values of t and y** into the function to get the **new dydt**

```matlab
function dydt=bangladesh(t,y)

Hr=(2.5*sin(pi+(t/210)*pi))+4;

dydt=[kp*(y(2)-y(1))-Cp*Eo;
     (1/S)*(kp*fp*(y(1)-y(2))+kr*fr*(Hr-y(2))-fag*Cag*Eo);
     dydt @ t=0+Δt

ode45 now calculates the next Δt, updates t and y (i.e. H_a, H_p) ... etc until t=tlim
function dydt=bangladesh(t,y)
Hr=(2.5*sin(pi+(t/210)*pi))+4;

If t < 180
Eo = 30
else Eo=50
end

dydt=[kp*(y(2)-y(1))-Cp*Eo;
  (1/S)*(kp*fp*(y(1)-y(2))+kr*fr*(Hr-y(2))-fag*Cag*Eo);

You can define your parameters for dydt in terms of t

ode45
New Δt
New t
New Ha
So, \textit{ode45} outputs $t$ values (your independent variable) \textcolor{red}{VECTOR}.

AND the solutions for $y$ (your dependent variable) for each $t$ value. \textcolor{red}{y} has two components, $H_a$ and $H_p$. \textcolor{red}{MATRIX}.
By clicking on WORKSPACE in the Matlab window, you can see the dimensions of your ode45 outputs.

y is a MATRIX, to refer to it in Matlab, use:

- `plot(t, y)`

or

- `plot(t, y(row#, column#))`
NOTE that \( t \) and \( y \) have the same number of rows. If not, something’s wrong. So, the rows in \( y \) are the solutions at the different \( t \) values.

\[
\text{plot}(t, \ y(\text{row#},\text{column#}))
\]

- To plot all time values use:
  \[
  \text{plot}(t, \ y(1:\text{n},\text{column#}))
  \]
  or simply
  \[
  \text{plot}(t, \ y(:,\text{column#}))
  \]
Syntax for *if* statements

```plaintext
if  t<180
    qi=0;
elseif t>190
    qi=5;
else
    qi=20;
end
```

`NEED` to make sure the parameter values for *all possible* `t`'s are covered

For example:

If `t<180`

```plaintext
    qi=0;
elseif t>180
    qi=5;
end
```

`WON’T WORK`  
`WHAT IF t=180??`
Subtracting and Adding vectors in Matlab

As long as the vectors have the same lengths, Matlab will add or subtract them, element by element, and the result will have the same length.

Example: \( A = y(:,1)-y(:,2) \)

then \( A \) will have the same number of rows as \( y \), and 1 column.
Vector 1 + Vector 2 = Result

Operation (+ or -)
Similarly if you multiply a constant by a vector

\[ C \times \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix} \]

\[ = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix} \]

OR, result = \( \sin(\text{vector1}) \)
FluxI

FluxI = 0 outside irrigating period, and equal to Transpiration within the irrigating period.

FluxI depends on t, but is not the result of an operation on t.
So to fill in the values of FluxI

You need to step through each element of $t$ one by one, check if it falls within the irrigating period using an if statement, and then assign the corresponding value of FluxI.
Use a `for` loop

\[ n = \text{size}(t) \]

\[ \text{for } i = 1 : n \]

`check t(i) and assign Flux(i) using an if statement`

\[ \text{end} \]

Because you’re assigning a value for `FluxI` for every value of `t`, `FluxI` and `t` should end up having the same lengths.
You didn’t need to do this when assigning \texttt{qi} values for \texttt{dydt} in the \textit{function M-file}, because \texttt{ode45} did the stepping through time and checked each value of \textit{t} before calculating the corresponding value of \textit{dydt}.
Tips

• Save your commands to an M-file
• To plot on top of a figure: `hold on`
• `hold off`
• To start a new figure: `figure`
• To plot in red: `plot(t,V,’r’)`
• To define a vector that starts at 1, ends at 500, with an increment of 5
  \[
  V=[1:5:500];
  \]
• `size(V)` ill give you vector with 2 components: #rows and #columns
• `help size`
Annual Aquifer Water Budget

- **Aquifer Discharge**
  - **Pumping**
  - **Field Infiltration**
  - **Pond Exchange**
  - **River Exchange**
  - **Aquifer Storage**

- **Recharge**
  - **With pumping**
  - **Without pumping**

**Residence Times:**
- With pumping - ~40 years
- No pumping - ~80 years
- If no ponds, much longer
Hydraulic Head at Intensive Study Site
(Sampled every 30 minutes)

- Shallow Aquifer
- Deep Aquifer
- Onset of Irrigation Pumping

Depth below land (m)

Dec. 1 2001
Jan. 1 2002
Additional Discussion Questions

• How are the recharge and discharge flows to and from the aquifer different between the two cases (with and without pumping)?

• How do you think the change in the hydrology of the aquifer produced by pumping for irrigation affects groundwater chemistry?