

' This program was written by Craig Keim for use
' in ABE580. 1/12/00

Public numEq As Integer 'Just some variable declarations.
Public stepSize, stopTime, startTime 'Move onto the constants subroutine below
Public O(), I() 'and enter the relevant information

Sub constants()
'Enter the values for the step size and the number of equations here.
stepSize = 0.05
numEq = 2

'*****
ReDim I(numEq), O(numEq) ' Just leave this line alone. It dimensions the
' input and output arrays
'*****

' Enter the initial conditions
startTime = 0 'Time value at which the initial conditions are known
O(1) = 0.1 'Remark for Output 1
O(2) = 50 'Remark for Output 2

'Enter the time at which you would like the simulation to end
stopTime = 125

End Sub

Function inputs(eqnNumber As Integer, timeValue As Variant, outputs() As Variant)
'*****
Dim j As Integer 'Just leave this code alone.
'
For j = 1 To numEq
O(j) = outputs(j)
Next j

t = timeValue 't is the time
'*****

'Enter the constants here
Pi = 3.14
F = 75
S0 = 50
X0 = 0
Ks = 0.719
mumax = 0.134
V = 800
D = F / V
Ysx = 12.698

'Constraints
If O(1) <= 0 Then O(1) = 0
If O(2) <= 0 Then O(2) = 0

'Enter the input equations here
I(1) = ((mumax * O(2) / (Ks + O(2))) - D) * O(1)
I(2) = D * (S0 - O(2)) - ((mumax * O(2) / (Ks + O(2))) * O(1) * Ysx)

'*****
inputs = I(eqnNumber)

End Function

```

Sub magic()
    Static j As Integer, k As Integer

    'Get all of the information about the constants
    constants

    'Clear out the old data from the spreadsheet
    clearData

    'Determine the number of iterations
    iterations = Int((stopTime - startTime) / stepSize)

    t = startTime

    'Print the initial time label
    Sheets("Data").Cells(numEq + 5, 1).Value = "Time"
    Sheets("Data").Cells(numEq + 6, 1).Value = t

    'Print the time variables used (step size, start and stop times)
    Sheets("Data").Cells(2, 1).Value = "Step Size"
    Sheets("Data").Cells(2, 2).Value = stepSize
    Sheets("Data").Cells(3, 1).Value = "Start Time"
    Sheets("Data").Cells(3, 2).Value = startTime
    Sheets("Data").Cells(4, 1).Value = "Stop Time"
    Sheets("Data").Cells(4, 2).Value = stopTime

    For k = 1 To numEq
        'List the Initial Conditions
        Sheets("Data").Cells(numEq + 5, k + 1).Value = "O(" & LTrim(Str(k)) & ")"
        Sheets("Data").Cells(numEq + 6, k + 1).Value = O(k)
    Next k

    For j = 1 To iterations
        RK4 (t)
        For k = 1 To numEq
            Sheets("Data").Cells(j + numEq + 6, k + 1).Value = O(k)
        Next k
        t = t + stepSize
        Sheets("Data").Cells(j + numEq + 6, 1).Value = t
    Next j

End Sub

Sub RK4(timeValue As Variant)
    Static k1(), k2(), k3(), k4(), oldOutput(), temp1(), temp2(), temp3(), newOutput()
    ReDim k1(numEq), k2(numEq), k3(numEq), k4(numEq)
    ReDim oldOutput(numEq), temp1(numEq), temp2(numEq), temp3(numEq), newOutput(numEq)
    Dim t, j As Integer

    t = timeValue

    For j = 1 To numEq
        oldOutput(j) = O(j)
    Next j

    'Calculate the 4 Runge-Kutta constants
    For j = 1 To numEq
        k1(j) = inputs(j, t, oldOutput())
        temp1(j) = oldOutput(j) + 0.5 * stepSize * k1(j)
    Next j

```

```
For j = 1 To numEq
    k2(j) = inputs(j, t + 0.5 * stepSize, temp1())
    temp2(j) = oldOutput(j) + 0.5 * stepSize * k2(j)
Next j

For j = 1 To numEq
    k3(j) = inputs(j, t + 0.5 * stepSize, temp2())
    temp3(j) = oldOutput(j) + stepSize * k3(j)
Next j

For j = 1 To numEq
    k4(j) = inputs(j, t + stepSize, temp3())
Next j

For j = 1 To numEq
    'Calculate the new y Value
    newOutput(j) = oldOutput(j) + stepSize / 6 * (k1(j) + 2 * k2(j) + 2 * k3(j) + k4(j))
    O(j) = newOutput(j)
Next j

End Sub

Sub clearData()
    'Clears all of the old data prior to putting in the new data
    Static lastCell As String

    Sheets("Data").Activate
    Sheets("Data").Cells(numEq + 6, 1).End(xlToRight).Select
    ActiveCell.End(xlDown).Select
    lastCell = ActiveCell.Address()
    Sheets("Data").Range("A1:" & lastCell).Clear
    Sheets("Data").Range("A1").Select
End Sub

Sub viewCode()
    'Sheets("Module1").Activate
    Application.Goto Reference:="constants"
End Sub
```