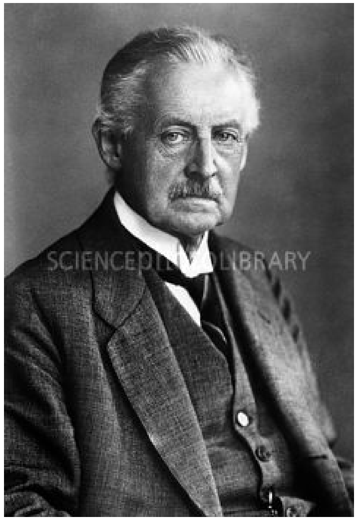
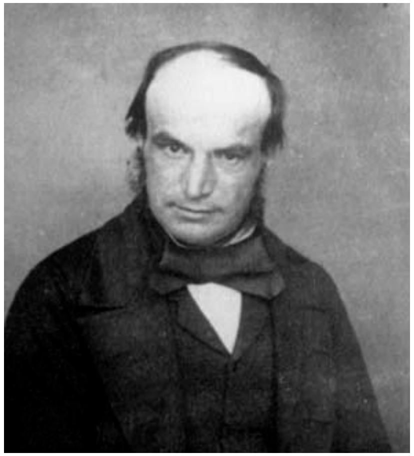
Using SIMBAS

- VisualBASIC for Applications
- 4^{th} Order Runge-Kutta ODE solver
- Enable “macros” in MS Excel
- Need ODEs and <u>initial conditions</u> for all variables!

Runge-Kutta!?



Carl Runge (1856-1927)



Martin Kutta (1867- 1944)

German mathematicians.

Develped numerical methods for solving the differential equations

Runge-Kutta Method

$$\frac{dy}{dx} = f(x, y)$$

The form used by Runge-Kutta to solve ODE:

$$\boldsymbol{y_{i+1} = y_i + \emptyset h}$$

Ø : slope

h: step size

$$dy = f(x, y)dx$$

$$\int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x, y)dx$$

Runge-Kutta General Equation

$$dy = f(x,y)dx$$

$$\int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x,y)dx$$

$$y_{i+1} - y_i = \int_{x_i}^{x_{i+1}} f(x,y)dx$$

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)dx$$

$$\boldsymbol{y_{i+1} = y_i + \emptyset h}$$

Runge-Kutta General Equation

$$\boldsymbol{y_{i+1} = y_i + \emptyset h}$$

$\emptyset = \frac{y_{i+1}-y_i}{x_{i+1}-x_i}$ *gradient*

$h = x_{i+1} - x_i$ *step size*

$$\boldsymbol{\emptyset h} = \frac{y_{i+1}-y_i}{\cancel{x_{i+1}-x_i}} \times \cancel{x_{i+1} - x_i} \longrightarrow \boldsymbol{\emptyset h} = y_i - y_{i+1}$$

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)dx$$

Runge-Kutta General Equation

$$dy = f(x, y)dx$$

$$\boldsymbol{y_{i+1} = y_i + \emptyset h}$$

Requirements:

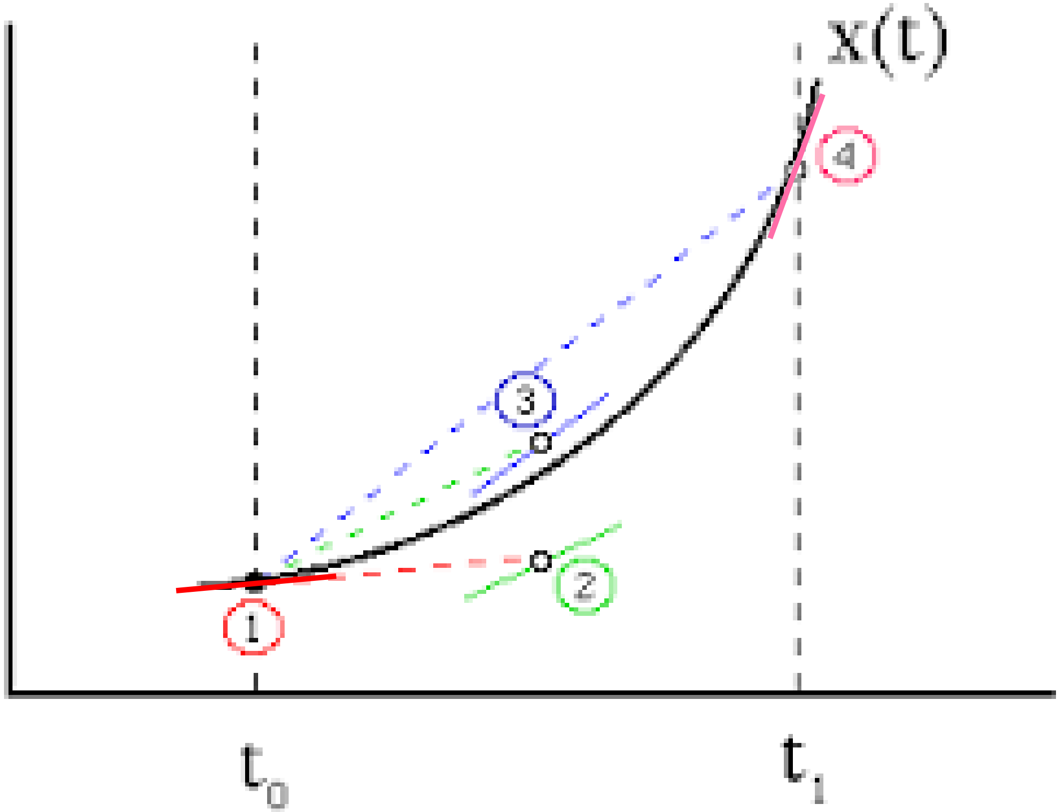
- Initial condition
- Slope [=f(x,y)]

Finds y_{i+1}, then y_{i+2}, … y_n

4th Order Runge-Kutta

Require:

1. ODE that describes curve
2. Initial condition (t_0, y_0)
3. Step size ($h = \Delta t$)



$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_1) \\
k_3 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_2) \\
k_4 &= f(t_n + h, y_n + h \cdot k_3) \\
y_{n+1} &= y_n + \frac{h}{6} \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)
\end{aligned}
$$

SIMBAS

```
I = input
O = output

Syntax
I(x)= element x in vector I
O(x)= element x in vector O
```

SIMBAS

```
I = dX/dt
O = X

Usage
I(1)= d(cells)/dt
O(1)= cells
```

SIMBAS vs MatLAB

```
SIMBAS
O(1) = 5
O(2) = 20
O(3) = 13
```

```
MatLAB
```

O = [5, 20, 13]; or

O = [5; 20; 13];

Microsoft Visual Basic for Applications - HW1-Monod-solution.xlsm - [Module1 (Code)]

```
' 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 relavent 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 intial conditions are known
    O(1) = 1                    'X, g/L
    O(2) = 200                  'S, g/L

    'Enter the time at which you would like the simulation to end
    stopTime = 24

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
    mumax = 0.26
    Ks = 0.315
    ysx = 14.3

    '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)) * O(1)
    I(2) = -ysx * I(1)

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

End Function
```

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

Microsoft Visual Basic for Applications - HW1-Monod-solution.xlsm - [Module1 (Code)]

```vb
' 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 relavent 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 intial conditions are known
    O(1) = 1                     'X, g/L
    O(2) = 200                   'S, g/L

    'Enter the time at which you would like the simulation to end
    stopTime = 24

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
    mumax = 0.26
    Ks = 0.315
    ysx = 14.3

    '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)) * O(1)
    I(2) = -ysx * I(1)

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

End Function
```

```
' Enter the initial conditions
   startTime = 0 'Time value at which the initial conditions are known
   O(1) = 1            'X, g/L
   O(2) = 200      'S, g/L

'Enter the time at which you would like the simulation to end
   stopTime = 24
```

Microsoft Visual Basic for Applications - HW1-Monod-solution.xlsm - [Module1 (Code)]

```
' 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 relavent 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 intial conditions are known
    O(1) = 1                      'X, g/L
    O(2) = 200                    'S, g/L

    'Enter the time at which you would like the simulation to end
    stopTime = 24

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
    mumax = 0.26
    Ks = 0.315
    ysx = 14.3

    '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)) * O(1)
    I(2) = -ysx * I(1)

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

End Function
```

```
'Enter the constants here
   mumax = 0.26
   Ks = 0.315
   ysx = 14.3

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

Microsoft Visual Basic for Applications - HW1-Monod-solution.xlsm - [Module1 (Code)]

```vb
' 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 relavent 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 intial conditions are known
    O(1) = 1                    'X, g/L
    O(2) = 200                  'S, g/L

    'Enter the time at which you would like the simulation to end
    stopTime = 24

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
    mumax = 0.26
    Ks = 0.315
    ysx = 14.3

    '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)) * O(1)
    I(2) = -ysx * I(1)
    '*******************************************
    inputs = I(eqnNumber)

End Function
```

'Enter the input equations here

```
I(1) = mumax * O(2) / (Ks + O(2)) * O(1)
I(2) = -ysx * I(1)
```

I(1) = dX/dt
I(2) = dS/dt

$$Y_{S/X} = \frac{dS}{dX} = \frac{dS/dt}{dX/dt}$$

SIMBAS

- Use Push Buttons to access code and run program
- Make sure to “reset” if it pauses due to a “run time error”
- Check that your number of equations (numeq) matches the number you define!
- Don’t “copy and paste”!!!!!

HW 1, Output result

- NO negative axis
    - glucose content axis starting at -25 g/L is meaningless!!
- Remove unncessary grid lines.
- Get creative with your graph/plot line to make it reader/grader friendly!
- Graphs with no axis label lose all points! ☹