# Application of Excel Macro Programming to Core Chemical Engineering Subjects

By: John Barford, Kelvin Wong

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Motivation

- Our current chemical engineering curricula requires students to learn C++ in computer science.
- However, C++ will not be used further in their chemical engineering courses or their future working environment.
- Most chemical engineering graduates spend at least half of their workday at computer.
- Graduates work mostly involves the use of user-friendly commercial software (e.g. Excel).
- It is believed that it is more suitable to equip chemical engineering graduates with skills to use those commercial available software.

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Project Aims

- This project aims to initiate the teaching of Excel VBA programming in chemical engineering.
- Teaching will be emphasized on the practical use of Excel VBA programming to solve chemical engineering problems.

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Course Development (1)

- This course would initially be taught to first year students in the winter section.
- Teaching of Excel VBA programming will include self-learning, tutorial and practice.
- It is based on using practical examples in core courses, where students is required to write Excel VBA programs to solve practical chemical engineering problems.
- The use of Excel VBA was demonstrated to the students and compare with other methods of solution which could use (e.g. Excel spreadsheet or Polymath).

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Course Development (2)

- Those examples and excises emphasis the power of using programming to replace time-consuming hand calculation, and the use of variable input / output as a generic solution to the problem.

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Current Progress

- Course materials, examples are developed and test run at CENG 364 (Biomolecular Engineering) during the Spring semester 2007 and CENG 361 (Introduction to Biochemical Engineering) during the Fall semester 2007.
- The course is going to give to first year students in coming winter section / semester.

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Example Program to Solve Chemical Engineering Problem

- This example is to demonstrate the advantage of writing own program as generic solution to chemical engineering problem.

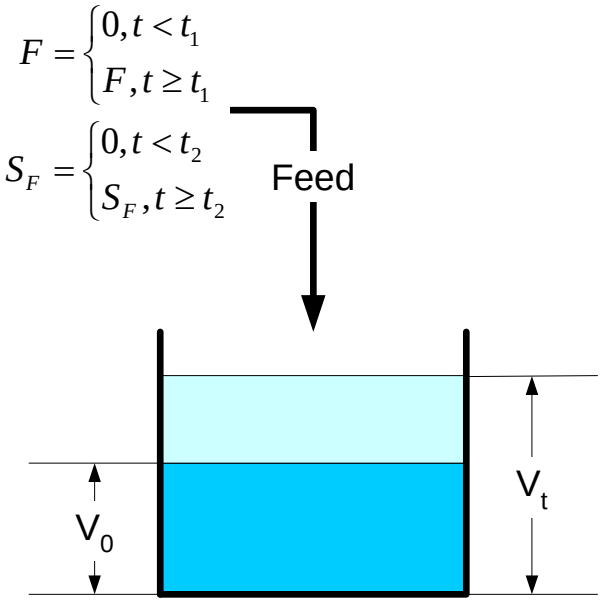
- Fed-batch bioreactor is an example used in the Excel VBA programming course. It can demonstrate the benefit of using programming to replace inflexible solutions such as Polymath or spreadsheet.

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Fed-Batch Bioreactor (1)

- A fed-batch bioreactor is a reactor that initially runs as a batch reactor, with volume $V_0$.
- After a certain period of time, a feed stream is introduced to input more substrate to maintain bacteria growth in the bioreactor.
- As the feed stream is introduced, the reactor volume changes from $V_0$ to $V_t$ with time.




with permission © Prof John Barford & Mr Kelvin Wong

# Fed-Batch Reactor (2)

- Governing equations:

Specific growth rate: $$\mu_t = \mu_{\max} \frac{S_t}{K_S + S_t}$$

Reactor volume: $$\frac{dV_t}{dt} = F_t$$

Biomass: $$\frac{d(X_t \cdot V_t)}{dt} = X_t \cdot \frac{dV_t}{dt} + V_t \cdot \frac{dX_t}{dt} = X_t \cdot F_t + V_t \cdot (\mu \cdot X_t)$$

Substrate: $$\frac{d(S_t \cdot V_t)}{dt} = S_t \cdot \frac{dV_t}{dt} + V_t \cdot \frac{dS_t}{dt} = S_t \cdot F_t + V_t \cdot \left( F_t \cdot S_{F,t} - \frac{\mu \cdot X_t}{Y_{XS}} \right)$$

Product: $$\frac{d(P_t \cdot V_t)}{dt} = P_t \cdot \frac{dV_t}{dt} + V_t \cdot \frac{dP_t}{dt} = P_t \cdot F_t + V_t \cdot \left( \frac{\mu \cdot X_t \cdot Y_{PS}}{Y_{XS}} \right)$$

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Fed-Batch Reactor (3)

- Governing equations (cont')

Feed rate: $$F = \begin{cases} 0, & 0 \le t < t_1 \\ F, & t \ge t_1 \end{cases}$$

Feed substrate conc.: $$S_F = \begin{cases} 0, & 0 \le t < t_1 \\ S_F, & t \ge t_1 \end{cases}$$

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Fed-Batch Reactor (4)

- Initial condition:

Reactor volume: $V(t=0)=V_0$

Biomass level: $X(t=0)=X_0$

Substrate conc.: $S(t=0)=S_0$

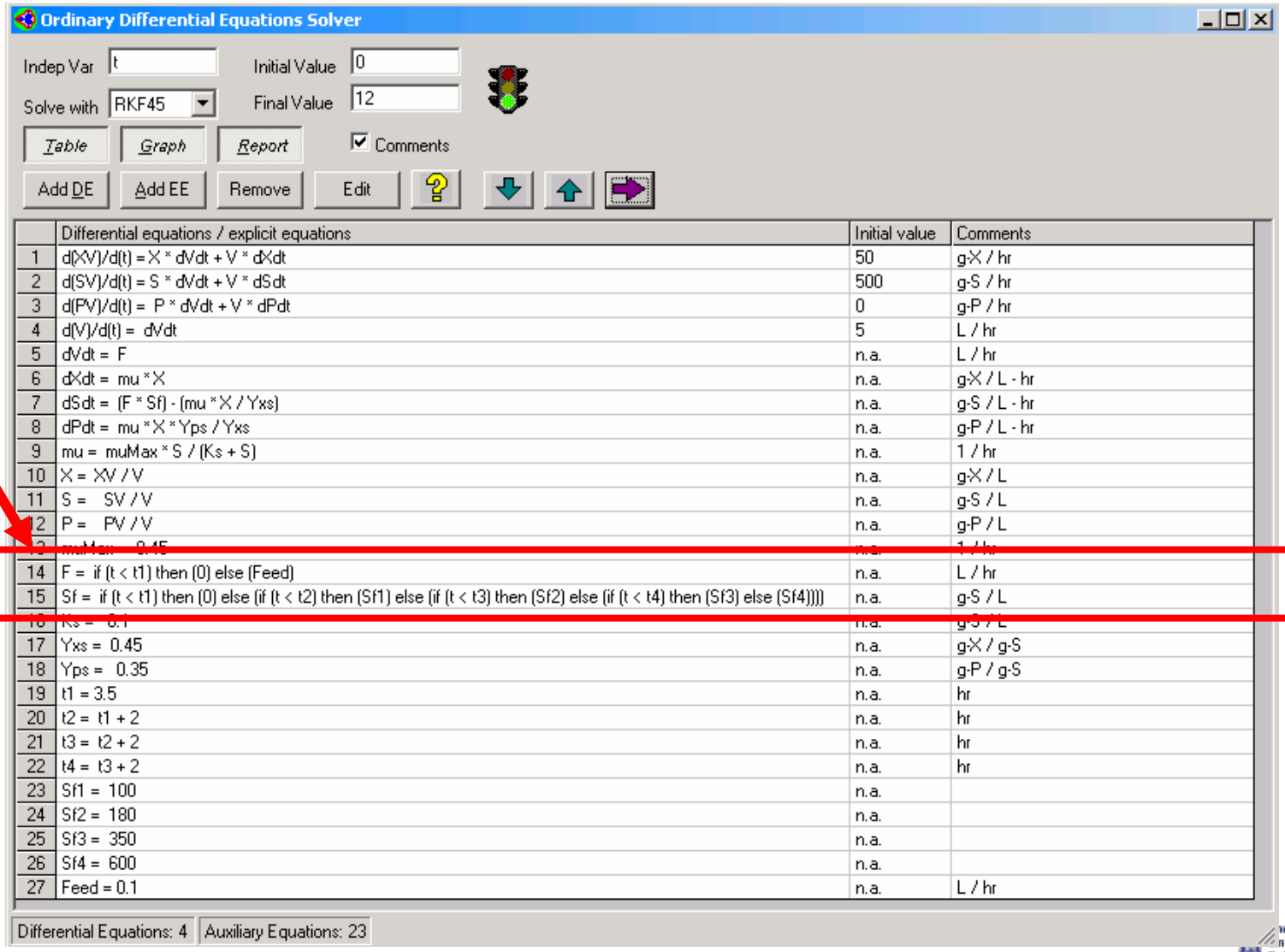
Product conc.: $P(t=0)=P_0$

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Using Polymath (1)

Polymath file



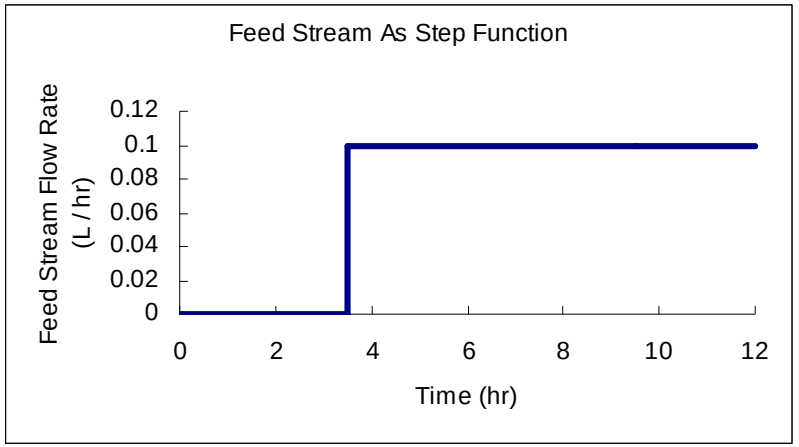

| | Differential equations / explicit equations | Initial value | Comments |
|---|---|---|---|
| 1 | d(XV)/d(t) = X * dVdt + V * dXdt | 50 | g-X / hr |
| 2 | d(SV)/d(t) = S * dVdt + V * dSdt | 500 | g-S / hr |
| 3 | d(PV)/d(t) = P * dVdt + V * dPdt | 0 | g-P / hr |
| 4 | d(V)/d(t) = dVdt | 5 | L / hr |
| 5 | dVdt = F | n.a. | L / hr |
| 6 | dXdt = mu * X | n.a. | g-X / L - hr |
| 7 | dSdt = (F * Sf) - (mu * X / Yxs) | n.a. | g-S / L - hr |
| 8 | dPdt = mu * X * Yps / Yxs | n.a. | g-P / L - hr |
| 9 | mu = muMax * S / (Ks + S) | n.a. | 1 / hr |
| 10 | X = XV / V | n.a. | g-X / L |
| 11 | S = SV / V | n.a. | g-S / L |
| 12 | P = PV / V | n.a. | g-P / L |
| 13 | muMax = 0.45 | n.a. | 1 / hr |
| 14 | F = if (t < t1) then (0) else (Feed) | n.a. | L / hr |
| 15 | Sf = if (t < t1) then (0) else (if (t < t2) then (Sf1) else (if (t < t3) then (Sf2) else (if (t < t4) then (Sf3) else (Sf4)))) | n.a. | g-S / L |
| 16 | Ks = 0.1 | n.a. | g-S / L |
| 17 | Yxs = 0.45 | n.a. | g-X / g-S |
| 18 | Yps = 0.35 | n.a. | g-P / g-S |
| 19 | t1 = 3.5 | n.a. | hr |
| 20 | t2 = t1 + 2 | n.a. | hr |
| 21 | t3 = t2 + 2 | n.a. | hr |
| 22 | t4 = t3 + 2 | n.a. | hr |
| 23 | Sf1 = 100 | n.a. | |
| 24 | Sf2 = 180 | n.a. | |
| 25 | Sf3 = 350 | n.a. | |
| 26 | Sf4 = 600 | n.a. | |
| 27 | Feed = 0.1 | n.a. | L / hr |

Differential Equations: 4 | Auxiliary Equations: 23

with permission © Prof John Barford & Mr Kelvin Wong

# Using Polymath (2)

- Step functions could be done by using the `IF-THEN-ELSE` statement to change value of constants according to simulation time:

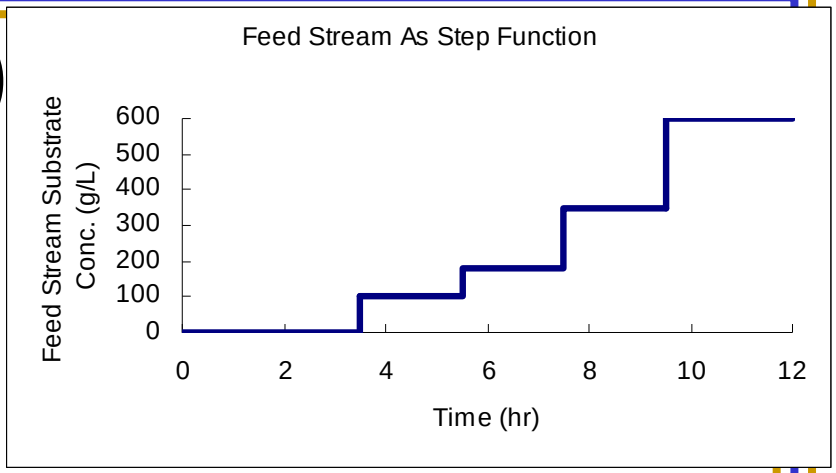



```
F = IF ( time < 3.5 hr ) THEN (
            F = 0 L/hr
    ) ELSE (
            F = 0.1 L/hr
    )
```

with permission © Prof John Barford & Mr Kelvin Wong

# Using Polymath (3)

- The multiple steps of feed substrate concentration could be done by using nested IF-THEN-ELSE statements:




- $S_{Feed}$ = IF ( time < 3.5 hr ) THEN (
  $S_{Feed}$ = 0 g/L
  ) ELSE ( IF ( time < 5.5 hr ) THEN (
  $S_{Feed}$ = 100 g/L
  ) ELSE ( IF ( time < 7.5 hr ) THEN (
  $S_{Feed}$ = 180 g/L
  ) ELSE ( IF ( time < 9.5 hr ) THEN
  $S_{Feed}$ = 350 g/L
  ) ELSE (
  $S_{Feed}$ = 600 g/L
  )
  )
  )
  )

with permission © Prof John Barford & Mr Kelvin Wong

# Using Polymath (4)

- **Advantages:**
  - Polymath already built-in numerical integration methods;
  - Users only need to input differential equations, etc.
  - Step functions could be input by using the `IF ... THEN ... ELSE ...` statement provided by Polymath.

with permission © Prof John Barford & Mr Kelvin Wong

# Using Polymath (5)

- **Disadvantages:**
  - Number of steps in step functions should be known before setting up the equations;
  - The numbers of nested `IF-THEN-ELSE` statements = number of steps - 1
  - If user wants to add more steps, he / she needs to modify the `IF-THEN-ELSE` statement, i.e. **no flexibility**;
  - Complicated `IF-THEN-ELSE` statements are not easy to read when there are many steps.
  - Polymath has limitation on the number of equations in a problem, for differential equation problems (Polymath version 6.10):

| Version | Educational | Professional |
|---|---|---|
| Max. # of simultaneous differential equations | 30 | 300 |
| Max. # of simultaneous explicit equations | 40 | 300 |
| Max. # of intermediate data points | 152 | 1200 |

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Using Excel (1)

Spreadsheet file

- It is possible to set up an Excel spreadsheet with numerical integration methods.
- E.g. Reactor volume with feed stream:

| Step | Time | Vol | Feed |
|---|---|---|---|
| 0 | $t_0$ = 0 | $V_0$ | IF ( ( Time < 3.5 hr ), ( F = 0 ), ( F = 0.1 ) ) |
| 1 | $t_1$ = $t_0$ + h | $V_1$ = $V_0$ + F x h | IF ( ( Time < 3.5 hr ), ( F = 0 ), ( F = 0.1 ) ) |
| 2 | $t_2$ = $t_1$ + h | $V_2$ = $V_1$ + F x h | IF ( ( Time < 3.5 hr ), ( F = 0 ), ( F = 0.1 ) ) |
| 3 | $t_3$ = $t_2$ + h | $V_3$ = $V_2$ + F x h | IF ( ( Time < 3.5 hr ), ( F = 0 ), ( F = 0.1 ) ) |
| 4 | $t_4$ = $t_3$ + h | $V_4$ = $V_3$ + F x h | IF ( ( Time < 3.5 hr ), ( F = 0 ), ( F = 0.1 ) ) |

with permission © Prof John Barford & Mr Kelvin Wong

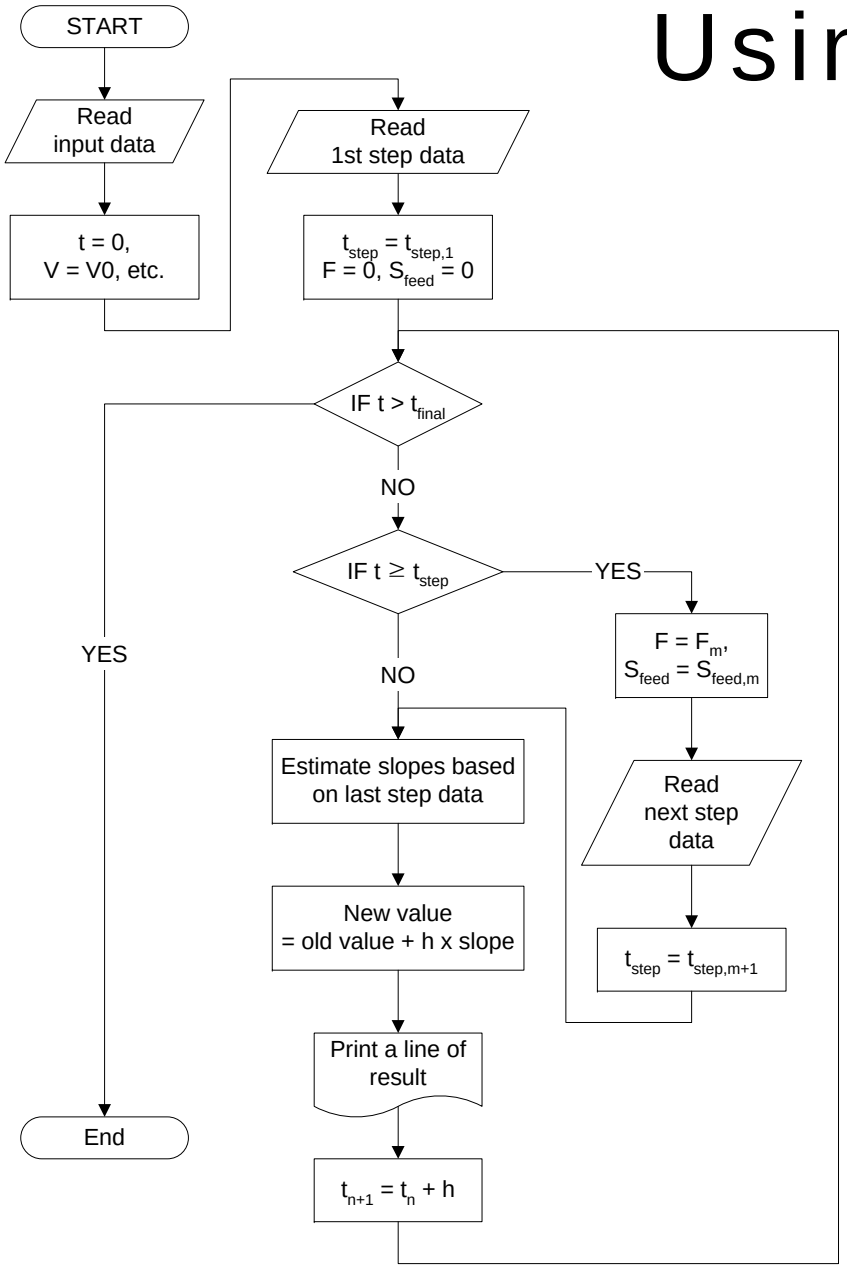
# Using Excel (2)

- **Costs:**
  - Users need to set up their own numerical integration.
- **Advantages:**
  - Possible to perform numerical integration when no mathematical software package is available.
- **Disadvantages:**
  - The same style to input step functions as using Polymath, i.e. **no flexibility**.
  - Number of steps for numerical integration (i.e. number of rows in the spreadsheet) must be changed when the ending simulation time changes.

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Using Excel VBA (1)




- Numerical integration should be set up by user as VBA code.
- An automatic feed data reading code is introduced;
- It is possible to handle any number of steps.

Excel VBA file

with permission © Prof John Barford & Mr Kelvin Wong

# Using Excel VBA (2)

- **Costs:**
  - Users need to write their own VBA program code.
- **Disadvantages:**
  - Need more time to set up the program;
  - Programming is not easy for novices.
- **Advantages:**
  - It is more flexible and can handle various kinds of decisions.
  - Although it requires time to set up the program, it is a one-off cost and this program can serve as generic solution towards this problem.

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong

# Excel VBA Project for Students

- In CENG 361 (Introduction to Biochemical Engineering), a project is given to student to let them to appreciate the benefit of the flexibility brought by write their own program.
- They are asked to write a program to solve sterilisation problem in the course.
- They are asked to develop an Excel spreadsheet solution and then write an Excel VBA program.  And to compare the benefit / advantages of either solution.

香港科技大學 The Hong Kong University of Science & Technology

with permission © Prof John Barford & Mr Kelvin Wong