Experimental Determination of Steady-State and Transient Behavior of Non-Isothermal and Non-Adiabatic Continuous Stirred-Tank Reactors

Extra Credit Project 2: Cyclical and Chaotic Steady-States

Computer Project

By

Dr. David Keffer

ChE 301
Department of Chemical Engineering
University of Tennessee
Knoxville, TN
Last Updated: February 2000

Purpose:

This document serves as an appendix to computer project 02. It describes an extra-credit project that can be performed in addition to the mandatory components of computer project 02. This computer project involves using the code written for computer project 02 to describe reactors with cyclical and chaotic steady states.

Cyclical and Chaotic Steady-States

Some reactive systems have cyclical steady states. The steady state is a closed trajectory through space. Other reactive systems have stable chaotic "steady" states. Whether you observe multiple steady states depends upon all parameters of the reactor system including inlet conditions, the nature of the reaction, and physical properties. A system that exhibits cyclical or chaotic steady states with an intermediate concentration of reactant in the feed stream may exhibit only an ordinary steady state with low or high concentrations of reactant in the feed stream. The program, sysode.m, can model these systems just as easily as a reactor with a single steady state.

Let's examine a system of reactions that produces cyclical and chaotic steady states that was introduced to me by Dr. Jeff Derby at the University of Minnesota in 1995. There are three liquid-phase irreversible reactions:

$$A + 2B \rightarrow 3B$$
$$B \rightarrow C$$
$$D + 2B \rightarrow 3B$$

We will consider an isothermal reaction so we will completely ignore the energy balance and the solvent mass balance.

Inlet conditions:

These are feed ratios:

$$\alpha_i = \frac{\text{molar feedrate of B}}{\text{molar feedrate of i}}$$

$$\alpha_A = 1.5$$
$$\alpha_B = 1.0$$
$$\alpha_C = 0.001$$
$$\alpha_D = 3.50$$

Reaction conditions:

These are Dahmkohler numbers which contain the isothermal reaction rates constants.

$$D_{a,A} = 18000.0;$$
$$D_{a,B} = 80.0;$$
$$D_{a,D} = 400.0;$$

Initial Conditions

$C_A = C_B = C_C = C_D = 0.0$ mol/liter (initially all unreactive solvent)

Use sysode.m to map a few trajectories, for different values of α_D.

Use $\alpha_D = 3.5,3.9,4.15,4.18,4.2,31$

This change in α_D corresponds to a decrease in the amount of component D fed to system. With the six numbers above, you should observe 2 ordinary steady states, 3 cyclical steady states (a single-cycle, a double-cycle, and a quadruple-cycle), and one chaotic steady state. Describe each system.

Appendix Two has the a sample input code for sysodeinput.m. These are just dimensionless mass balances.

To get you started, this command will give you one trajectory:

sysode(2,1000,0,2.0,[0.0,0.0,0.0,0.0])

You will need very small times steps for this reaction. Otherwise the Runge-Kutta algorithm will blow up.

You will have to integrate the ODEs for various times depending upon the value of α_D. Some values will require more time to reach steady state than others.

Appendix Two. Sample input file, sysodeinput.m

```
function f = sysodeinput(t,x0,nvec)
%
%
%  The Data entered here will correspond to the data of
%  Computer Problem 1, From ChEn 8203, Winter 1995
%  Taught by Prof. Jeff Derby in the Chem E. Dept. at the University of Minnesota
%
%  This system demonstrates cyclical steady states.
%
%   A + 2B --> 3B
%        B --> C
%   D + 2B --> 3B
%
%   Isothermal (so no energy balance is required)
%
%
%  STEP ONE.  INITIALIZE VARIABLES
%
CA = x0(1);
CB = x0(2);
CC = x0(3);
CD = x0(4);
%
%  STEP TWO.  DEFINE PARAMETERS
%
%  Dahmkohler Numbers
%
DaA = 18000.0;
DaB = 80.0;
DaD = 400.0;
%
%  Feed Ratios with respect to flow of component B (actually inverse flow ratios, alphaA = Flow
B/ Flow A)
%
alphaA = 1.5;
alphaB = 1.0;
alphaC = 0.001;
alphaD = 3.50;
%
%  STEP THREE.  Define (non-dimensionalized) equations
%  (these are mass balances)
%
% acc =  in  - out        +/- generated/consumed
dCAdt =  1.0 - CA         - DaA*CA*CB*CB;
dCBdt =  1.0 - (1+DaB)*CB + alphaA*DaA*CA*CB*CB + alphaD*DaD*CD*CB*CB;
dCCdt =  0.0 - CC         + alphaC*DaD*CB;
dCDdt =  1.0 - CD         - DaD*CD*CB*CB;
%
f(1) = dCAdt;
f(2) = dCBdt;
f(3) = dCCdt;
f(4) = dCDdt;
```