Chemical Reaction Engineering - Part 4 - Integration

Richard K. Herz, rherz@ucsd.edu, www.ReactorLab.net

Integrating the component balance for a batch reactor

For a single reaction in a batch reactor, the component balance is an ordinary differential equation (ODE). We can write this for any component in the stoichiometric equation. Here, as an example, we will write it for the limiting or key reactant A.

$$\frac{dN_A}{dt} = r_A V$$

We only need to write one component balance because we can use the stoichiometric table to compute the concentration of any component in the stoichiometric equation knowing the concentration of one component and the initial composition.

The rate term r_A will be a function of T and the concentration of components in the mixture. For gases, partial pressures may appear rather than concentration.

$$\frac{dN_A}{dt} = r_A V = f(T, C_A, C_B, etc.)V$$

$$\frac{dN_A}{dt} = r_A V = f(T, p_A, p_B, etc.)V$$

On the left side of the balance we have moles and on the right we have concentrations or partial pressures.

To do before we can integrate

Before we can integrate, we need to write all of these moles, concentrations and pressures in terms of one dependent variable. This one dependent variable can be any one of the following: conversion X_A, concentration C_A, partial pressure p_A, or stoichiometric extent of reaction ξ. Any will work. Some might save you a step or two in writing. The independent variable is time t.

We do this by writing a stoichiometric table. If we have gases at relatively low pressure, we can write the ideal gas law and include any inert gases in N_{tot}.

Simple cases

Some of these may be easy to integrate from memory. An example is the balance for an essentially irreversible, first-order reaction in a constant volume reactor from Part 1 of these notes.

$$V\frac{dC_A}{dt} = -kC_A V \quad ; \quad \text{in terms of conversion} \quad \frac{dX_A}{dt} = k(1 - X_A)$$

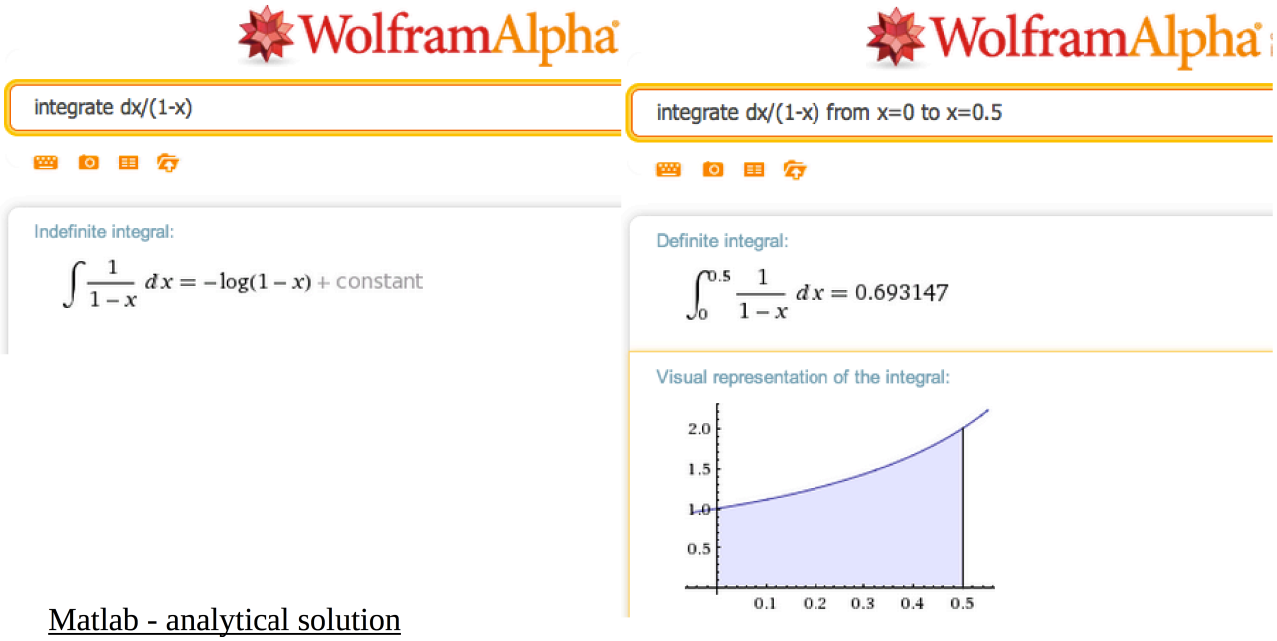
$$\frac{dC_A}{C_A} = -k\,dt \quad ; \quad t=0\,;\, C_A(0)=C_{A,0};\, X_A(0)=0 \quad ; \quad \frac{dX_A}{(1-X_A)} = k\,dt$$

$$-\ln(C_A/C_{A,0}) = k\,t_{rxn} \qquad ; \text{ in terms of conversion} \quad -\ln(1-X_A) = k\,t_{rxn}$$

Other cases will be more complex, so we will discuss several ways to obtain an exact analytical solution or an approximate numerical solution.

WolframAlpha - analytical solution

If you have access to the Web, you can get analytical solutions using WolframAlpha.com. The natural logarithm ("ln") in WolframAlpha is "log." Log to the base 10 is "log10."



Matlab - analytical solution

Matlab's symbolic toolbox (function library) can be used to do analytical integration.

```
>> syms x
>> int(1/(1-x))

ans =

-log(x - 1)
```

We can also get the definite integral for integration between initial and final values of x

```
>> syms x
>> int(1/(1-x),0,0.5)

ans =

log(2)
```

This is the symbolic answer from the symbolic toolbox. We can then use the standard function log to evaluate this result. In Matlab, as in WolframAlpha, the natural logarithm ("ln") is "log." Log to the base 10 is "log10."

```
>> log(2)

ans =

    0.6931
```

Numerical approximation by Euler's method

The basic idea is that we start at the initial conditions (X = 0 at t = 0) and take "steps" in the independent variable, t here, and estimate the new value of the dependent variable (X here) at each new step.

$$\frac{dX_A}{dt} = k(1 - X_A) \quad ; \quad t = 0 \; ; \; C_A(0) = C_{A,0}$$

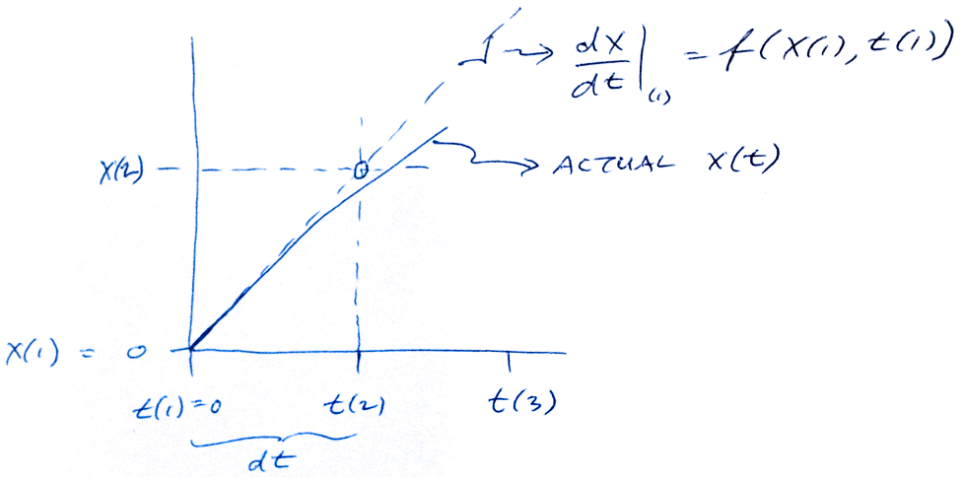
The easiest way to show this is with Matlabs

```
% first-order, essentially irreversible reaction
% by Euler's method
clear
k = 1; % (1/s), rate coefficient
i = 1; % initial array index
x(i) = 0; % d'less, initial conversion
t(i) = 0; % s, initial time
dt = 0.1; % s, step size
trxn = 5; % s, final time
while t(i) <= trxn
    dxdt = k*(1-x(i)); % variable dxdt = dx/dt (not dx*dt)
    dx = dxdt * dt; % change in x
    x(i+1) = x(i) + dx;
    t(i+1) = t(i) + dt;
    i = i+1; % increment array index
end
ta = 0:0.001:trxn; % compute analytical solution for comparison
xa = 1 - exp(-k*ta);
plot(t,x,'bx',ta,xa,'r')
tt = sprintf('X vs. T, dt = %4.2f, blue x = Eulers, red = exact',dt);
title(tt)
ylabel('X'),xlabel('t')
```

At each time, the current value of dx/dt is computed (dxdt is variable name in code, since can't use /) and that value is used to extrapolate to the value of x at the next time. Since the value of dx/dt changes continuously with time, the value extrapolated to the next time is an approximation of the real value of x at the new time. The smaller the time step, the better approximation we will get at the cost of additional computation time. If the steps are too big, then the numerical solution can become unstable and exhibit oscillations and nonphysical values.

Euler's method, e.g., for $\frac{dx}{dt} = f(x,t)$

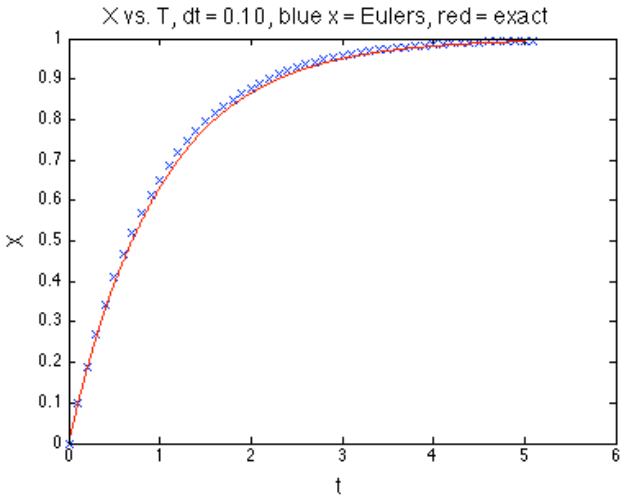
$$dx(i) = f(x(i), t(i)) * dt$$
$$x(i+1) = x(i) + dx$$
$$t(i+1) = t(i) + dt$$

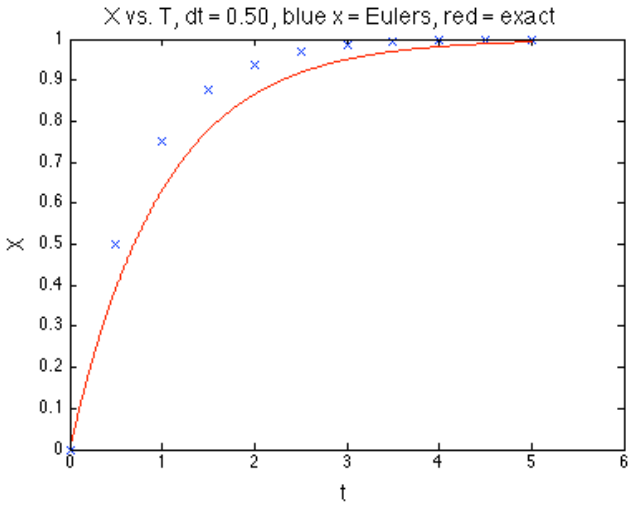


These are example variables. You can use this with any ODE and variables, e.g., concentration and time, u and v, x and y, etc.

This example is simple so we can compare the numerical approximation to the exact analytical solution. This may not be possible for more complex problems, such as multiple ODE's for multiple reaction systems or non-isothermal systems.

On the left, we can see that we get a reasonable numerical solution with a time step of 0.10. That value is specific to this example. We get a pretty bad solution with a time step of 0.50 on the right.





On an exam, when you have a complex rate equation and are given the reaction time and asked to estimate the conversion, you can use Euler's method, taking large steps. On the other hand, when you are given the desired final conversion and asked to estimate the reaction time, a Levenspiel plot (inverse rate plot) may be faster. Those plots are discussed below.

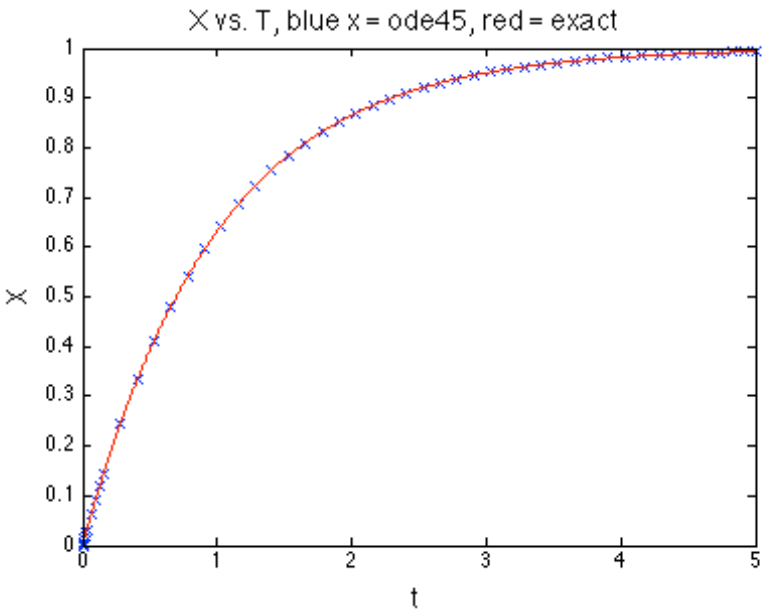
Numerical approximation by Matlab's standard function ode45

Euler's method is easy to understand and will be good enough for most of our work. There are more sophisticated numerical algorithms, such as Runge-Kutta methods. Runge-Kutta methods give more accurate estimates for the same computational cost. Matlab's standard function `ode45` uses such algorithms. Remember to use Matlab's help facilities when writing programs.

```
% first-order, essentially irreversible reaction
% using Matlab's standard function ode45
% uses user-written function myodeDeriv
clear
global k % share k value with main and function (quick & dirty method)
k = 1; % (1/s), rate coefficient
x0 = 0; % d'less, initial conversion
t0 = 0; % s, initial time
trxn = 5; % s, final time
[t x] = ode45('myodeDeriv',[t0 trxn],x0);
ta = 0:0.001:trxn; % compute analytical solution for comparison
xa = 1 - exp(-k*ta);
plot(t,x,'bx',ta,xa,'r')
title('X vs. T, blue x = ode45, red = exact')
ylabel('X'),xlabel('t')
```

Listing of file myodeDeriv.m

```
function dxdt = myodeDeriv(t,x)
    global k
    dxdt = k*(1-x);
end
```



Numerical approximation by Levenspiel plots or inverse rate plots

Euler's method and ode45 can be use for single ODE's. In addition, they can be used with multiple, coupled ODE's, such as those we will encounter later with multiple reaction systems and nonisothermal systems.

Levenspiel plots, named in honor of Prof. Octave Levenspiel, can be used for solution of single ODE's only. Single ODE's are what we are discussing now with single, isothermal reactors, so let's take a look at them. We may also refer to them as inverse rate plots.

Levenspiel plots can be used when we are given a table of reaction rate values at specified conversions, or can compute such a table. This will be the case when

1. we know the rate equation and the values of the rate coefficients
2. we are given a table in homework or exam problems
3. we do experiments using a steady-state, continuous stirred-tank reactor, or CSTR

Start with the component balance for a batch reactor.

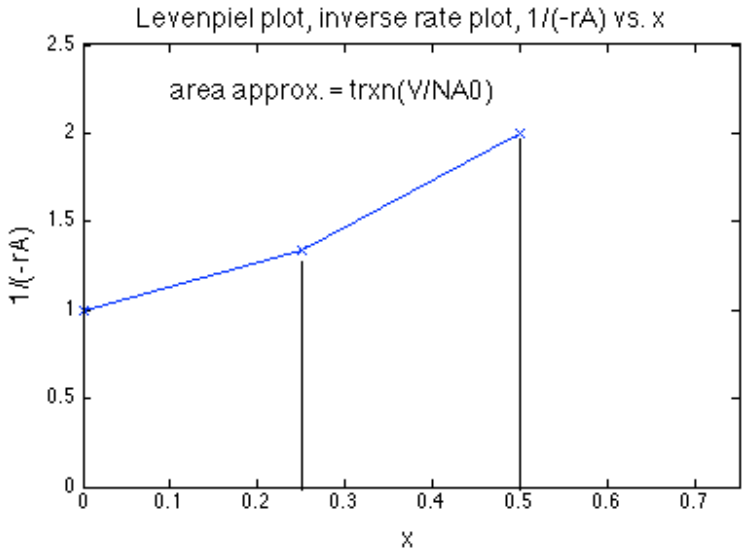
$$\frac{dN_A}{dt} = r_A V$$

$$-N_{A0}\frac{dX_A}{dt} = r_A(X_A)V \qquad \text{where the rate is a known function of conversion}$$

$$\int_0^{X_{A,final}} \frac{dX_A}{-r_A(X_A)} = \frac{1}{N_{A0}}\int_0^{t_{rxn}} V\,dt$$

Plot ($1/-r_A(X_A)$) values (inverse rate values) vs. a few X_A values. The points will define a curve.

X_A	$-r_A$	$1/(-r_A)$
0.00	1.00	1.00
0.25	0.75	1.33
0.50	0.50	2.00



The area under the curve out to the final conversion will equal the value of $t_{rxn}(V/N_{A0}) = t_{rxn}/C_{A0}$ for a constant volume batch reactor (or the integral shown for a variable volume reactor).

Drawing straight lines between the points on the plot defines trapezoids. The area of each trapezoid can be computed and the areas summed to get the total area.

$$\text{area} = 0.25\left(\frac{1+1.33}{2}\right) + 0.25\left(\frac{1.33+2}{2}\right) = 0.71$$

For $C_{A0} = 1$, $k = 1$, and $r_A = -kC_{A0}(1-X_A)$, the exact value for t_{rxn}/C_{A0} is 0.69 so the plot gives us a good estimate for this case.

A Levenspiel plot is just a method of graphically integrating an ODE.

You don't need to exactly have 1/-rA on the y-axis. For example, you could bring the rate constant or other constants over to the right-hand side of the equation.

You could also leave the equation in terms of concentration.

For example, for constant volume and a 1st-order, essentially irreversible reaction:

$$\int_{C_A}^{C_{A0}} \frac{dC_A}{-r_A} = t_{rxn}$$

Note that the integration limits have been reversed from the usual order to convert $+r_A$ to $-r_A$ on the left side.

Here are the cases listed above again

1. we know the rate equation and the values of the rate coefficients
2. we are given a table in homework or exam problems
3. we do experiments using a steady-state, continuous stirred-tank reactor, or CSTR

In case 1, where we know the rate equation, we can integrate using any of the methods discussed above.

For cases 2 and 3, we don't know the rate equation, so we should use a Levenspiel plot.

On an exam, a Levenspiel plot is useful for case 1 when we are asked to estimate the reaction time to reach a specified conversion. We can compute rate values at the final conversion and one or two intermediate conversions and use a Levenspiel plot. We could use Euler's method but we don't know what size of time steps to take in order to get a reasonable approximation in a small number of steps.

On the other hand for case 1, when we are asked to find the conversion at a given final time on an exam, then Euler's method may be faster than a Levenspiel plot. With Euler's method, we can take a time couple steps to the final time. With a Levenspiel plot, we don't know the final conversion to use, so we would have to guess and iterate.

Question: How long would we need to react to get to 100% conversion? What happens to the value of the inverse rate when the concentration approaches zero? What happens to the area under the curve?

Summary

With Euler's method, you step in time, so it's easy to use when given a final time

$$\frac{dX_A}{dt} = -r_A(X_A)(V/N_{A0}) \quad \text{step in t to get } X_A \text{ at specified } t_{rxn}$$

Alternatively, when given a final conversion, you can keep stepping in time and checking until you reach that conversion. Euler's method and related methods also can be used with multiple, coupled ODE's.

With Levenspiel plots, you essentially step in conversion, it's so easy to use when given final conversion

$$\int_0^{X_{A,final}} \frac{dX_A}{-r_A(X_A)} = t_{rxn}(V/N_{A0}) \quad \text{to get } t_{rxn} \text{ at given } X_A$$

Alternatively, when given a final time, you can try different final conversions until the area under the curve matches the final time (iterate). Levenspiel plots can only be used for single ODE's.

Important new terms

ODE
Euler's method
Runge-Kutta methods
Levenspiel plot or inverse rate plot
trapezoid area

Additional resources

Schmidt's book - on library reserve & online via `http://roger.ucsd.edu/record=b7179459~S9`
Chapter 3 - section 7 [note: plots $1/-r_A$ vs. $(C_{A0}-C_A)$, where $(C_{A0}-C_A) = C_{A0} X_A$ for constant V]

Levenspiel's book - on library reserve
Chapter 5 - section 1