

An introductory overview of process modeling

James B. Rawlings

Department of Chemical and Biological Engineering



September 28, 2009

The Theory of Process Engineering
Summer School of the IMPRS
Magdeburg, Germany

- 1 Overview of this Lecture
- 2 Ordinary Differential Equation Models
- 3 Random Variables and Stochastic Models
- 4 Parameter Estimation with Differential Equation Models
- 5 Summary and Further Reading

What you will learn in this lecture

- 1 Differential equation models
 - ▶ Writing them
 - ▶ Solving them numerically with software (MATLAB or Octave)
- 2 Modeling experimental data
 - ▶ Random variables, probability, and measurement noise
 - ▶ Normal distribution
 - ▶ Confidence intervals
- 3 Estimating parameters from data
 - ▶ Objective function and constraints
 - ▶ Optimization algorithm
 - ▶ Approximate confidence intervals
- 4 Good software is important.
The software used in this lecture can be downloaded here:
<http://cbe255.che.wisc.edu>

What we will *not* cover in this lecture

- 1 Differential algebraic equation models
 - ▶ Professors Mehrmann and Biegler
- 2 Partial differential equation models
 - ▶ Professors Eigenberger and Mazzotti
- 3 Empirical models from data
 - ▶ Professors Carrondo and Oliveira
- 4 Optimization methods
 - ▶ Professors Wright, Biegler, Bock and Schlöder
- 5 Estimating states from data
 - ▶ Professor Findeisen
- 6 Feedback controller design based on models
 - ▶ Professor Findeisen

Material balance for the well-mixed, batch reactor

The batch reactor is assumed to be well stirred, so there are no concentration gradients anywhere in the reactor volume. In this case it is natural to consider the entire reactor contents to be the reactor volume element as in Figure 1, and $V = V_R$. The statement of conservation of moles for component j is

$$\frac{d(c_j V_R)}{dt} = R_j V_R \quad j = 1, \dots, n_s \quad (1)$$

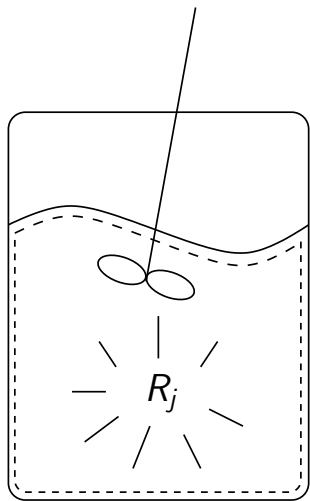


Figure 1: Batch reactor volume element.

Constant volume assumption

Equation 1 applies whether the reactor volume is constant or changes during the course of the reaction. If the reactor volume is constant, which is sometimes a good approximation for liquid-phase reactions, V_R can be divided out of both sides of Equation 1 to give

$$\frac{dc_j}{dt} = R_j \quad j = 1, \dots, n_s \quad (2)$$

Be sure to use Equation 1 rather than Equation 2 if the reactor volume changes significantly during the course of the reaction.

Now, where do we get R_j ?

Consider the following reaction in CVD chemistry



The **reaction rate**, r , is defined as the number of times this reaction event takes place per time per volume.

One can imagine turning SiH_4 , SiH_2 and Si_2H_6 molecules loose in a box of some fixed volume V as depicted in Figure 2.

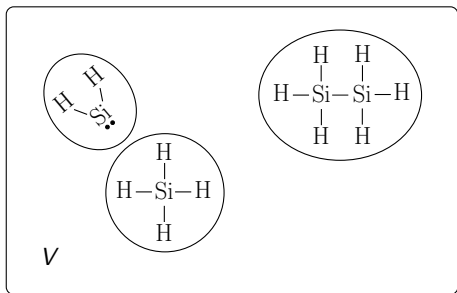


Figure 2: Defining the reaction rate, r , for the reaction $\text{SiH}_2 + \text{SiH}_4 \rightleftharpoons \text{Si}_2\text{H}_6$.

Imagine that we could somehow count up the net number of times an SiH_4 molecule hit an SiH_2 molecule and turned into an Si_2H_6 molecule during a short period of time. The change in the reaction extent, $\Delta\varepsilon$, is the net number of reaction events that occur in the time interval Δt . The reaction rate is then

$$r = \frac{\Delta\varepsilon}{\Delta t V} \quad (4)$$

It is important to connect the reaction rate to the rate of change of the concentrations of the various species in the reactor, which are the quantities we usually care about in a commercial reactor. We define **production rate**, R_j , as the rate at which the j th species is produced (moles/(time·volume)) due to the chemical reactions taking place.

It is clear looking at the stoichiometry in Reaction 3 that each time the forward reaction event occurs, an Si_2H_6 molecule is produced. Each time the reverse reaction occurs, an Si_2H_6 molecule is consumed. The production rate of Si_2H_6 , $R_{\text{Si}_2\text{H}_6}$, is therefore directly related to the reaction rate,

$$R_{\text{Si}_2\text{H}_6} = r$$

Notice that if r is positive $R_{\text{Si}_2\text{H}_6}$ is positive as we expect because Si_2H_6 is being produced. Similar arguments lead to relating the other production rates to the reaction rate,

$$R_{\text{SiH}_4} = -r$$

$$R_{\text{SiH}_2} = -r$$

Notice that we have three production rates, one for each species, but only one reaction rate, because there is only a single reaction. If we now introduce the production rate vector, \mathbf{R} ,

$$\mathbf{R} = \begin{bmatrix} R_{\text{SiH}_4} \\ R_{\text{SiH}_2} \\ R_{\text{Si}_2\text{H}_6} \end{bmatrix}$$

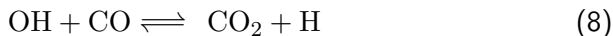
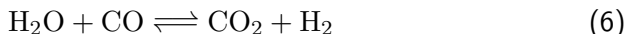
we can summarize the connection between the three production rates and the single reaction rate by

$$\mathbf{R} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} r \quad (5)$$

Notice that the column vector in Equation 5 is just the transpose of the row vector that comprises $\nu = [-1 \ -1 \ 1]$, which follows from Reaction 3

Another example

Consider what happens to the relationship between the production and reaction rates if there is more than one reaction. Recall the water gas shift reaction,



Three reaction rates are required to track all three reactions. Let r_i denote the reaction rate for the i th reaction.

What production rate of atomic hydrogen, H, results from these three reactions? We notice that H does not take part in the first reaction, is consumed in the second reaction, and is produced in the third reaction. We therefore write

$$R_H = (0) r_1 + (-1) r_2 + (1) r_3 = -r_2 + r_3$$

Consider the second species, H₂. It is produced in the first and second reactions and does not take part in the third reaction. Its production rate can therefore be expressed as

$$R_{H_2} = (1) r_1 + (1) r_2 + (0) r_3 = r_1 + r_2$$

You should examine the remaining four species and produce the following matrix equation,

$$\begin{bmatrix} R_H \\ R_{H_2} \\ R_{OH} \\ R_{H_2O} \\ R_{CO} \\ R_{CO_2} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & -1 \\ -1 & -1 & 0 \\ -1 & 0 & -1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad (9)$$

Batch reactor example



The full set of differential equations for a well-mixed reactor are

$$\begin{aligned}\frac{dc_A}{dt} &= -r_1 = -k_1 c_A \\ \frac{dc_B}{dt} &= r_1 - r_2 = k_1 c_A - k_2 c_B \\ \frac{dc_C}{dt} &= r_2 = k_2 c_B\end{aligned}$$

- 1 Solve the full model and plot c_A, c_B, c_C versus time using for the following parameters

$$k_1 = 1 \quad k_2 = 10 \quad c_{A0} = 1 \quad c_{B0} = 0 \quad c_{C0} = 0$$

- 2 Solve the QSSA reduced model for the same parameter values. Note however that you need to find a value of c_{B0} that is consistent with the algebraic equation at $t = 0$ or the DAE solver may fail.

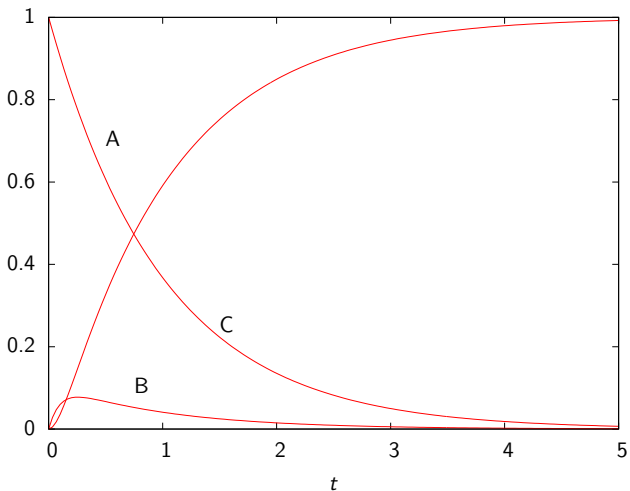


Figure 3: The solution to the model for the series reaction $A \rightarrow B \rightarrow C$; ODE model.

Figure 3 shows the solution to the model. We use the following call to the ODE solver `ode15s` and the following function `rates` to evaluate the differential equations.

```
function dcdt = rates(t, x)
    global k1 k2
    ca = x(1); cb = x(2); cc = x(3);
    r1 = k1*ca; r2 = k2*cb;
    dcdt = [-r1; r1-r2; r2];
```

```
[tout, x] = ode15s (@rates, time, x0, opts);
```

```
global k1 k2
k1 = 1;
k2 = 10;

x0 = [1; 0; 0];
npts = 150;
tfin = 5;
time = linspace(0, tfin, npts)';

[tout, x] = ode15s (@rates, time, x0, opts);

plot(tout, x)
```

Figure 4 shows the solution to the QSSA model. Notice that the intermediate B is highly reactive and present only in small concentration compared to the reactant A and main product B. Also, we see that the full model with three ODEs shows the rapid approach of species B to its quasi-steady-state value. Using the reduced QSSA model removes this transient and species B jumps instantly to its quasi-steady-state value. The small difference in B's dynamics has no noticeable impact on the behavior of A and C. For these rate constant values ($k_1 = 1, k_2 = 10, k_1 \ll k_2$), the QSSA provides an accurate reduced model.



The QSSA reduced model is obtained by setting $dc_B/dt = 0$ giving a set of DAEs

$$\begin{aligned}\frac{dc_A}{dt} &= -r_1 &&= -k_1 c_A \\ 0 &= r_1 - r_2 &&= k_1 c_A - k_2 c_B \\ \frac{dc_C}{dt} &= r_2 &&= k_2 c_B\end{aligned}$$

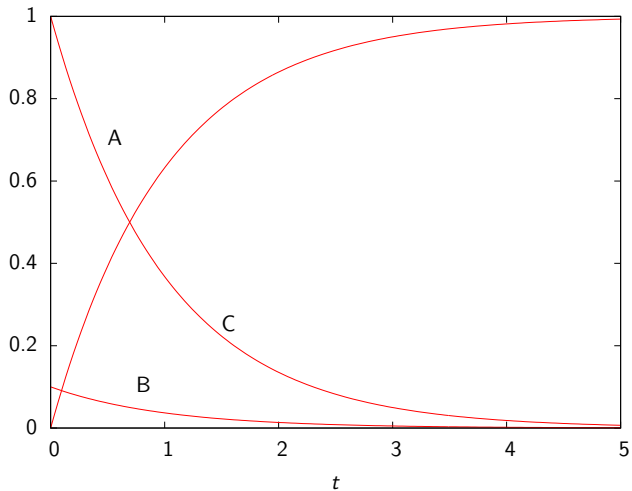


Figure 4: The solution to the reduced QSSA model; DAE model.

We use the following call to the DAE solver `ode15i` and the following function `qssa` to evaluate the residuals of the DAEs. Notice the similarities and differences between the functions `qssa` and `rates`.

```
function resid = qssa(t, x, xdot)
    global k1 k2
    ca = x(1); cb = x(2); cc=x(3);
    cadot = xdot(1); cbdot = xdot(2); ccdot = xdot(3);
    r1 = k1*ca; r2 = k2*cb;
    resid = [-cadot - r1; r1 - r2; -ccdot + r2];
end
```

```
[tout, x] = ode15i (@qssa, time, x0, xdot0, opts);
```

```
global k1 k2
k1 = 1;
k2 = 10;

npts = 150;
tfin = 5;
time = linspace(0, tfin, npts)';

x0 = [1; 0; 0];
% set initial b concentration
x0(2) = k1/k2*x0(1);
% set the initial time derivatives
xdot0 = qssa(0, x0, zeros(3,1));
xdot0(2) = 0;

[tout, x] = ode15i (@qssa, time, x0, xdot0, opts);

plot(tout, x)
```

Rescaling B to remove k_2 from the model

The only problem with the DAEs is that I still need to know k_2 to solve them. I might as well solve the ODEs if I know k_2 .

So let's introduce a new state $z = k_2 c_B$. We then have the model



This QSSA reduced model is again obtained by setting $dc_B/dt = 0$ giving a set of DAEs

$$\frac{dc_A}{dt} = -k_1 c_A$$

$$0 = k_1 c_A - z$$

$$\frac{dc_C}{dt} = z$$

Removing k_2 from the DAE model

- And using $z = k_1 c_A$, the initial condition is simply

$$z(0) = k_1 c_{A0} \quad \dot{z}(0) = k_1 \dot{c}_A(0) = -k_1^2 c_{A0}$$

and we have removed k_2 completely from the model.

- Instead of requiring $c_B(0)$ and k_2 , which are difficult to measure, we require only $c_A(0)$, $c_C(0)$, and k_1 , which are easy to measure.
- Here's the new code

No k_2 or $c_B(0)$ in this m-file

```
global k1
k1 = 0.5;

function resid = qssa_z(t, x, xdot)
    global k1
    ca = x(1); z = x(2); cc=x(3);
    cadot = xdot(1); zdot = xdot(2); ccdot = xdot(3);
    r1 = k1*ca;
    resid = [-cadot - r1; r1 - z; -ccdot + z];
endfunction

x0 = [1; 0; 0];
% set initial z value
x0(2) = k1*x0(1);
% set the initial time derivatives
xdot0 = qssa_z(0, x0, zeros(3,1));
xdot0(2) = k1*xdot0(1);

[tout, x] = ode15i (@qssa, time, x0, xdot0, opts);

plot(tout, x)
```

Material balance for the well-mixed, continuous reactor

The continuous-stirred-tank reactor (CSTR) is also a well-stirred reactor so there are no concentration gradients anywhere in the reactor volume.

The difference between the CSTR and batch reactor is the flow streams shown in Figure 5.

The statement of conservation of moles for component j is

$$\frac{d(c_j V_R)}{dt} = Q_f c_{jf} - Q c_j + R_j V_R,$$
$$j = 1, \dots, n_s \quad (10)$$

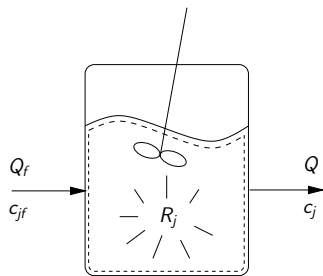


Figure 5: CSTR volume element.

Constant volume assumption

If the reactor volume is constant and the volumetric flowrates of the inflow and outflow streams are the same, Equation 10 reduces to

$$\frac{dc_j}{dt} = \frac{1}{\tau}(c_{jf} - c_j) + R_j \quad j = 1, \dots, n_s \quad (11)$$

in which

$$\tau = V_R/Q_f$$

is called the **mean residence time** of the CSTR.

We refer to this balance as the constant-density case. It is often a good approximation for liquid-phase reactions.

Continuous reactor. Energy balance

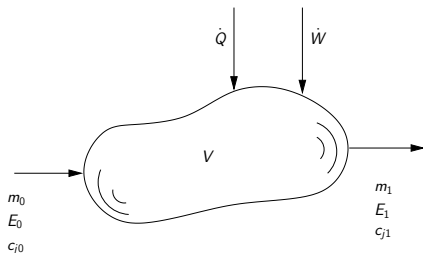


Figure 6: Reactor volume element.

The statement of conservation of energy for this system takes the form,

$$\left\{ \begin{array}{l} \text{rate of energy} \\ \text{accumulation} \end{array} \right\} = \left\{ \begin{array}{l} \text{rate of energy} \\ \text{entering system} \\ \text{by inflow} \end{array} \right\} - \left\{ \begin{array}{l} \text{rate of energy} \\ \text{leaving system} \\ \text{by outflow} \end{array} \right\} + \left\{ \begin{array}{l} \text{rate of heat} \\ \text{added to system} \end{array} \right\} + \left\{ \begin{array}{l} \text{rate of work} \\ \text{done on system} \end{array} \right\} \quad (12)$$

CSTR energy balance (simplified)

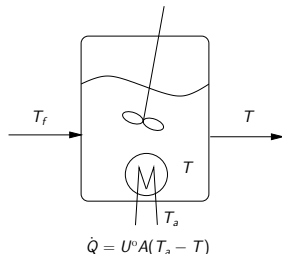


Figure 7: CSTR with heat exchange.

$$\frac{dT}{dt} = \frac{U^o A}{V_R \rho \hat{C}_P} (T_a - T) + \frac{T_f - T}{\tau} - \sum_i \frac{\Delta H_{Ri}}{\rho \hat{C}_P} r_i$$

Oscillations. Coupled mass and energy balances.

Consider again the irreversible reaction



The mass and energy balances for a continuous-stirred-tank reactor (CSTR) are given by (Rawlings and Ekerdt, 2002, p. 299)

$$\frac{dc_A}{dt} = \frac{c_{Af} - c_A}{\tau} - kc_A \quad (13)$$

$$\frac{dT}{dt} = \frac{U^o A}{V_R \rho \hat{C}_P} (T_a - T) + \frac{T_f - T}{\tau} - \frac{\Delta H_R}{\rho \hat{C}_P} kc_A \quad (14)$$

$$k(T) = k_m e^{-E(1/T - 1/T_m)} \quad (15)$$

in which c_A is the reactor A concentration and T is the reactor temperature. Notice the rate constant k also depends on the reactor temperature. Including the energy balance to describe the reactor temperature allows the reactor to exhibit complex behavior.

We wish to solve the model for the following parameter values

Param.	Value	Units
T_f	298	K
T_a	298	K
T_m	298	K
\hat{C}_P	4.0	kJ/(kg K)
c_{Af}	2.0	kmol/m ³
$k_m(T_m)$	0.004	min ⁻¹
E	1.5×10^4	K
ρ	10^3	kg/m ³
ΔH_R	-2.2×10^5	kJ/kmol
$U^o A/V_R$	340	kJ/(m ³ min K)
τ	73.1	min

Solve the two differential equations describing the reactor from the following initial condition using MATLAB's default parameters

$$c_{A0} = 0.36 \text{ mol/L} \quad T_0 = 315 \text{ K}$$

Plot c_A , T versus t for this solution out to $t = 20\tau$.

Next plot c_A versus T for this solution. This type of plot is called a phase plot or phase portrait. Notice it gives you a clear indication when a system is approaching a persistent oscillation known as a limit cycle.

Next we tighten the absolute and relative error tolerance for `ode15s` using the `odeset` function. Learn more about this function by typing at the command line

```
help odeset
```

The machine precision is stored in the variable `eps`. Try setting both the absolute and relative error tolerances to the square root of the machine precision. The `odeset` command is

```
opts = odeset ('AbsTol', sqrt (eps), 'RelTol', sqrt (eps));
```

After setting `opts`, you pass this extra argument to the ODE solver to use the tighter tolerances with

```
[tout, x] = ode15s (@oderhs, tout, x0, opts);
```

Resolve the ODEs with the same initial conditions and the tighter tolerances. Plot the solution with both loose and tight tolerances on the same graph. What do you conclude about MATLAB's default error tolerances?

The following function `rhs.m` defines the right-hand sides for the differential equations given in Equations 13–15

```
%%                                rhs.m

function retval = rhs(t,x)
global k_m T_m E c_Af theta C_ps T_f DeltaH_R U T_a

c_A = x(1);
T = x(2);
k = k_m*exp(-E*(1/T - 1/T_m));
retval = zeros (2, 1);
retval(1) = (c_Af - c_A)/theta - k*c_A;
retval(2) = U/C_ps*(T_a - T) + (T_f - T)/theta - k*c_A*DeltaH_R/C_ps;
```

This function is passed to the ODE solver in the following script.

```
%%                                exofirst.m
%%
%% limit cycle in A->B CSTR with heat release
%%
%% jbr, 10/15/2007
%%
global k_m T_m E c_Af theta C_ps T_f DeltaH_R U T_a

x0 = [(1-0.82)*c_Af;    315];
tfinal = 20*theta;
ntimes = 200;
tout = linspace(0, tfinal, ntimes)';

[tsolver, x] = ode15s (@rhs, tout, x0);
conv = (c_Af - x(:,1)) / c_Af;
unstabilim = [tsolver x conv];

figure(1)
plot (tsolver, x(:,1))
figure(2)
plot (tsolver, x(:,2))
figure(3)
plot (x(:,1), x(:,2))
```

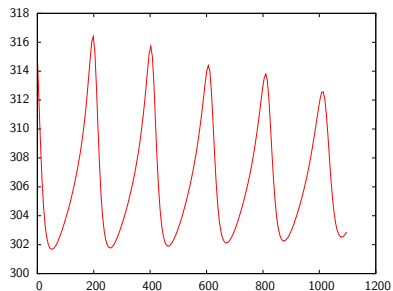
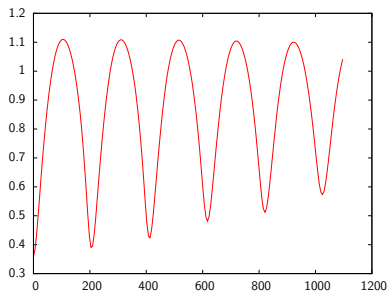


Figure 8: Concentration of A and temperature versus time with MATLAB default error tolerances.

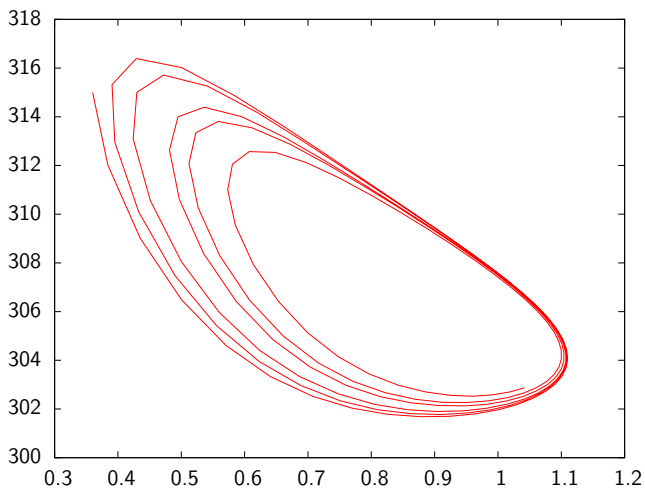


Figure 9: Phase portrait of c_A versus T with MATLAB default error tolerances.

The call to the ODE solver `ode15s` was changed to the following

```
opts = odeset ('AbsTol', sqrt (eps), 'RelTol', sqrt (eps));  
[tsolver, x] = ode15s (@rhs, tout, x0, opts);
```

The results for the tighter error tolerance are displayed in Figures 10 and 11. We see that the solution is not decaying to a steady state, but is approaching a limit cycle. Using the default error tolerances changed not only the quantitative details but also the qualitative features of the solution.

The moral of the story is to not use the `MATLAB` defaults, but to use $\sqrt{\epsilon}$ for both relative and absolute error tolerances. Always call `odeset` and tighten the error tolerances before calling `ode15s`.

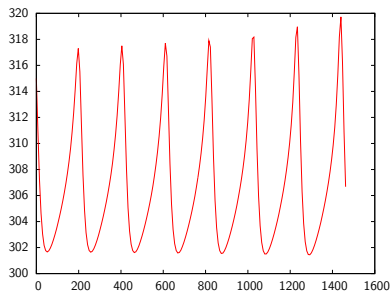
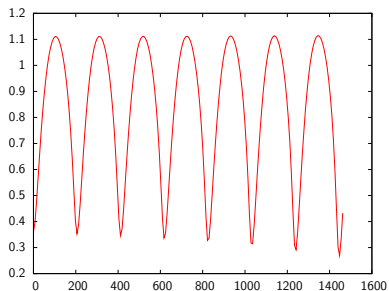


Figure 10: Concentration of A and temperature versus time with $\sqrt{\epsilon}$ error tolerances.

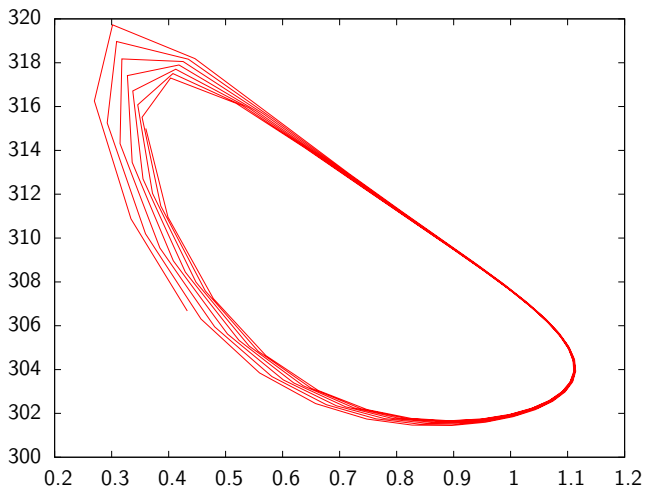


Figure 11: Phase portrait of c_A versus T with $\sqrt{\varepsilon}$ error tolerances.

Random variables

Let x be a *random* variable taking real values.

We define the **probability density function**, denoted $p(x)$, with the following properties

$$p(x) \geq 0 \quad \text{all } x$$

$$\int_{-\infty}^{\infty} p(x) dx = 1$$

and the interpretation in terms of probability

$$\Pr(x_1 \leq x \leq x_2) = \int_{x_1}^{x_2} p(x) dx$$

Discrete or continuous valued

We can define the density function also for discrete (integer valued) as well as continuous random variables. The random variable may be a coin toss or a dice game, which takes on values from a discrete set contrasted to a temperature or concentration measurement, which takes on a values from a continuous set.

Mean and Variance

The **mean** or **expectation** of a random variable x is defined as

$$\bar{x} = \int_{-\infty}^{\infty} xp(x)dx \quad (16)$$

The **variance** is defined as

$$\text{var}(x) = \int_{-\infty}^{\infty} (x - \bar{x})^2 p(x)dx$$

The standard deviation is the square root of the variance

$$\sigma(x) = \sqrt{\text{var}(x)}$$

Review of the Normal Distribution

Probability and statistics provide one useful set of tools to model the uncertainty in experimental data.

The normal or Gaussian distribution is ubiquitous in applications. It is characterized by its mean, m , and variance, σ^2 , and is given by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x - m)^2}{\sigma^2}\right) \quad (17)$$

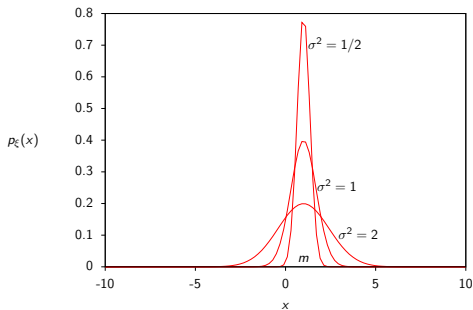


Figure 12: Normal distribution, $p_{\xi}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x - m)^2}{\sigma^2}\right)$. Mean is one and variances are 1/2, 1 and 2.

We adopt the following notation to write Equation 17 more compactly

$$x \sim N(m, \sigma^2)$$

which is read “the random variable x is distributed as a normal with mean m and variance σ^2 .” Equivalently, the probability density $p(x)$ for random variable x is given by Equation 17.

Vectors of random variables

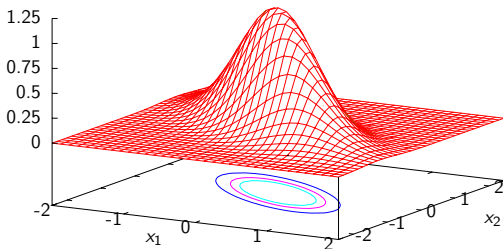
For distributions in more than one variable, we let \mathbf{x} be an n_p -vector and the generalization of the normal is

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{n_p/2} |\mathbf{P}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \mathbf{m})' \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m}) \right]$$

in which the n_p -vector \mathbf{m} is the mean and the $n_p \times n_p$ -matrix \mathbf{P} is called the covariance matrix. The notation $|\mathbf{P}|$ denotes determinant of \mathbf{P} . We also can write for the random variable \mathbf{x} vector

$$\mathbf{x} \sim N(\mathbf{m}, \mathbf{P})$$

The matrix \mathbf{P} is a real, *symmetric* matrix.



A multivariate normal with

$$\mathbf{P}^{-1} = \begin{bmatrix} 3.5 & 2.5 \\ 2.5 & 4.0 \end{bmatrix}$$

Lines of constant probability in the multivariate normal are lines of constant

$$(\mathbf{x} - \mathbf{m})' \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m})$$

Eigenvalues and Eigenvectors

To understand the geometry of lines of constant probability (ellipses in two dimensions, ellipsoids or hyperellipsoids in three or more dimensions) we examine the eigenvalues and eigenvectors of the \mathbf{P} matrix.

An eigenvector of a matrix \mathbf{A} is a nonzero vector \mathbf{v} such that when multiplied by \mathbf{A} , the resulting vector points in the same direction as \mathbf{v} , and only its magnitude is rescaled. The rescaling factor is known as the corresponding eigenvalue λ . Therefore the eigenvalues and eigenvectors satisfy the relation

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \quad \mathbf{v} \neq \mathbf{0}$$

We normalize the eigenvectors so

$$\mathbf{v}'\mathbf{v} = \sum_i v_i^2 = 1$$

The eigenvectors show us the orientation of the ellipse given by the normal distribution. Consider the ellipse in the two-dimensional \mathbf{x} coordinates given by the quadratic

$$\mathbf{x}'\mathbf{A}\mathbf{x} = b$$

If we march along a vector \mathbf{x} pointing in the eigenvector \mathbf{v} direction, we calculate how far we can go in this direction until we hit the ellipse $\mathbf{x}'\mathbf{A}\mathbf{x} = b$. Substituting $\alpha\mathbf{v}$ for \mathbf{x} in this expression yields

$$(\alpha\mathbf{v}')\mathbf{A}(\alpha\mathbf{v}) = b$$

Using the fact that $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ for the eigenvector gives

$$\alpha^2\lambda\mathbf{v}'\mathbf{v} = b$$

because the eigenvectors are of unit length, we solve for α and obtain

$$\alpha = \sqrt{\frac{b}{\lambda}}$$

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = b$$

$$\mathbf{A} \mathbf{v}_i = \lambda_i \mathbf{v}_i$$

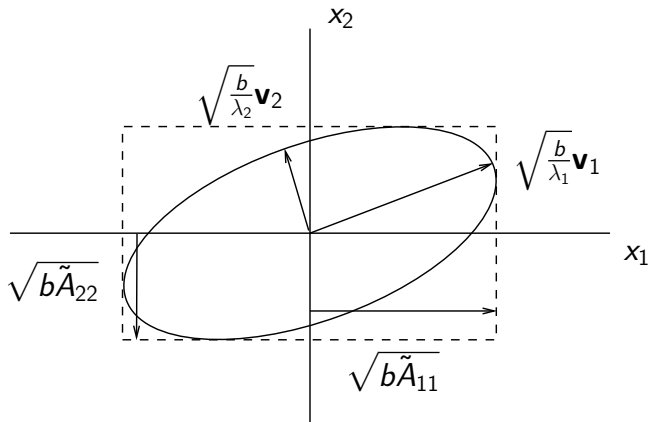


Figure 13: The geometry of quadratic form $\mathbf{x}'\mathbf{A}\mathbf{x} = b$.

Bounding box

If we want to put simple bounds on the ellipse, then we draw a box around it as shown in Figure 13. Notice the box contains much more area than the corresponding ellipse and we have lost the correlation between the elements of \mathbf{x} . This loss of information means we can put different tangent ellipses of quite different shapes inside the same box. The size of the bounding box is given by

$$\text{length of } i\text{th side} = \sqrt{b\tilde{A}_{ii}}$$

in which

$$\tilde{A}_{ii} = (i, i) \text{ element of } \mathbf{A}^{-1}$$

Figure 13 displays these results: the eigenvectors are aligned with the ellipse axes and the eigenvalues scale the lengths. The lengths of the sides of the box that is tangent to the ellipse are proportional to the square root of the diagonal elements of \mathbf{A}^{-1} .

Sampling random variables

Often we do not know the probability density of a random variable, but we have collected samples of the random variable. Given enough samples, we can sometimes approximate the probability density or any of the properties of the probability density such as the mean and variance. For example, the `MATLAB` function `rand` generates samples of a uniformly distributed random variable on the interval $[0, 1]$, and `randn` generates samples of a normally distributed random variable with unit variance and zero mean. The `MATLAB` function `hist` plots a histogram of the samples.

Histogram of samples from a normal density

Use `MATLAB` to generate 10,000 samples of a normally distributed random variable with zero mean and unit variance and plot the histogram of the samples. How does the histogram compare to Figure 12? Compare this result to the histogram with 10,000 samples.

The two commands:

```
x = randn(10000,1);  
hist(x,50)
```

generates the 10,000 samples and plots the histogram using 50 bins. The default is 10 bins if we leave off the second argument to `hist`. Type `help hist` and `help randn` to learn more about these functions.

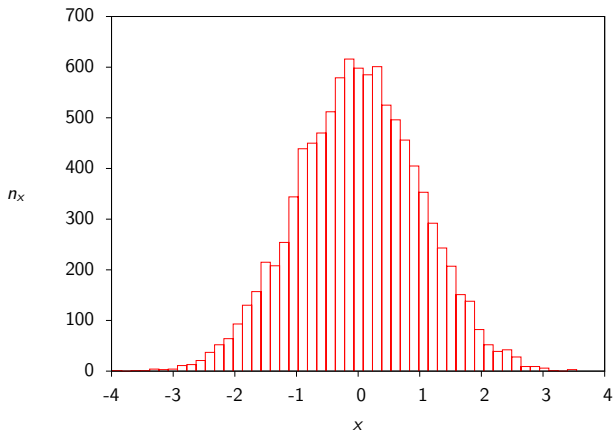


Figure 14: Histogram of 10,000 samples of MATLAB's `randn` function.

Sample mean and variance

We can use the samples to generate an approximation of the mean and variance of the random variable. These are known as the *sample mean* and *sample variance*, respectively, to distinguish the approximation computed from samples from the true mean and variance. Let the samples of x be denoted $x_i, i = 1, 2, \dots, n$ in which we have n samples. The sample mean and variance are given by

$$m = \frac{1}{n} \sum_{i=1}^n x_i$$
$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - m)^2$$

Notice the sample mean is the formula you always use to compute an average of a collection of numbers. The sample variance sums the squares of the distances of the samples from the mean, and then divides by $n - 1$ (rather than n). The division by $n - 1$ reflects the fact that we cannot compute a variance or spread when we have only a single sample.

We can extend the sample mean and variance when \mathbf{x} is a vector of random variables. Let \mathbf{x}_i denote the i th sample of the random \mathbf{x} vector. The formulas are

$$\mathbf{m} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$
$$\mathbf{P} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})'$$

Notice in the sample variance, we take the (outer) product of two vectors $\mathbf{x}\mathbf{x}'$, which is an $n \times n$ matrix, and not the usual (inner) product $\mathbf{x}'\mathbf{x}$, which is a scalar. The `MATLAB` functions `mean` and `cov` compute the sample mean and covariance for samples of a vector-valued random variable

Sample mean and covariance for three different measurement types

Say we measure three constant variables in an experimental system: the temperature, pressure, and concentration of some key component. Say the mean temperature is 20°C , the mean pressure is 1 bar, and the mean concentration is 2 mol/L. Say we know the measurement error standard deviations are 0.5°C , 0.2 bar, and 0.1 mol/L, respectively. We also know these measurement errors are independent random variables.

Generating measurements with noise

Use `randn` to generate 15 samples of these three measurements and use `mean` and `cov` to compute the sample mean and covariance. How close are the sample mean and covariance to the true mean and covariance used to generate the samples?

Sample mean and variance

We generate the samples and compute the sample mean and covariance using

```
nsam = 15;

Tsam = 20 + 0.5*randn(nsam,1);
Psam = 1 + 0.2*randn(nsam,1);
Csam = 2 + 0.1*randn(nsam,1);

Measmat = [Tsam, Psam, Csam]

smean = mean(Measmat)
sP = cov(Measmat)
plot(Measmat)
```

which generates the following output

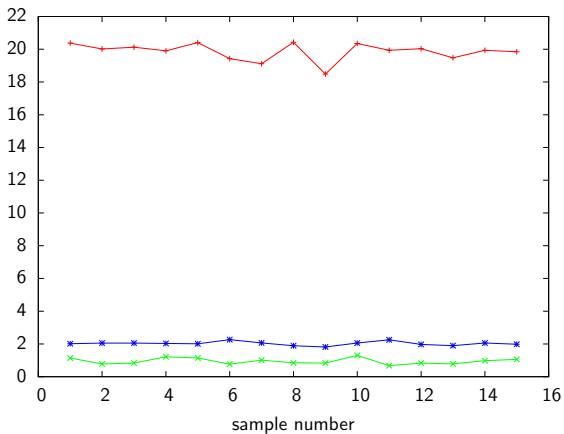


Figure 15: The 15 samples of measurement, pressure, and concentration.

Sample mean and variance

Measmat =

20.15075	0.92840	2.12553
19.37189	1.31131	2.00060
19.40432	1.31351	1.82824
20.16828	0.76685	1.88849
20.34552	0.91807	1.94029
20.41108	0.89530	2.03677
20.16328	0.96543	1.86885
18.86462	1.00509	1.94699
20.35392	1.03841	2.14621
20.17987	0.96683	1.84220
19.98065	0.64298	1.87331
20.07565	0.94577	2.06438
20.67500	0.93168	2.11631
19.57186	0.94377	2.03733
20.21466	0.98150	1.84896

smean =

19.99542	0.97033	1.97096
----------	---------	---------

sP =

2.3235e-01	-3.8051e-02	1.3868e-02
-3.8051e-02	2.8680e-02	7.6420e-05
1.3868e-02	7.6420e-05	1.2442e-02

Notice the mean is fairly accurate. The true covariance matrix is

$$\mathbf{P} = \begin{bmatrix} (0.5)^2 & 0 & 0 \\ 0 & (0.2)^2 & 0 \\ 0 & 0 & (0.1)^2 \end{bmatrix}$$

The covariance matrix is diagonal because the measurement errors are not correlated with each other. Notice the sample covariance and the true covariance are fairly different from each other. The diagonal elements are reasonably close but the off-diagonal elements in the sample covariance are not very close to zero. It is generally true that sample means are accurate with a small number of samples, but sample variances require a much larger number of samples before they are accurate.

Linear transformation of a random variable

If we multiply a scalar random variable x by a constant a and define that to be a new random variable y , then we alter both the mean and variance of y compared to x . The formulas are given as follows

$$y = ax$$

$$m_y = am_x \quad \text{var}(y) = a^2\text{var}(x)$$

You should test this result by generating some samples of a uniform or normal distribution and multiplying by a scalar.

Vector of random variables

The corresponding formulas for vector-valued random variables are as follows.

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$
$$\mathbf{m}_y = \mathbf{A}\mathbf{m}_x \quad \text{var}(\mathbf{y}) = \mathbf{A}\text{var}(\mathbf{x})\mathbf{A}' \quad (18)$$

Why is the normal distribution so important? Central limit theorem

The central limit theorem states that if the random variables x_i are independent, then under general conditions the density $p(x)$ of their sum

$$y = x_1 + x_2 + \cdots + x_n$$

tends to a normal curve as $n \rightarrow \infty$. (Papoulis, 1984, p. 194).

Notice we have not said *anything* about how the x_i are distributed.

Adding ten uniformly distributed random variables.

Consider ten uniformly and independently distributed random variables, x_1, x_2, \dots, x_{10} . Consider a new random variable y , which is the sum of the ten x random variables

$$y = x_1 + x_2 + \dots + x_{10}$$

Even though the ten x random variables are uniformly distributed, and their probability distribution looks nothing like a normal distribution, let's explore the probability distribution of the resulting y random variable. According to the central limit theorem, it may appear to be normally distributed.

Mean and variance of x

The x random variables are distributed as

$$x \sim U(0, 1)$$

Computing the mean and variance gives

$$\bar{x} = \int_0^1 x dx = \frac{x^2}{2} \Big|_0^1 = \frac{1}{2}$$
$$\text{var}(x) = \int_0^1 (x - \bar{x})^2 dx = \frac{1}{3} (x - 1/2)^3 \Big|_0^1 = \frac{1}{12}$$

Linear transformation

If we stack the x variables in a vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{10} \end{bmatrix}$$

we can write the y random variable as the linear transformation of the x 's

$$y = Ax \quad A = [1 \quad 1 \quad \cdots \quad 1]$$

Mean and variance of y

Using the previous results on linear transformations we know that y 's mean and variance are given by

$$\begin{aligned}\bar{y} &= A\bar{x} & \bar{y} &= 5 \\ \text{var}(y) &= A\text{var}(x)A' & \text{var}(y) &= \frac{5}{6}\end{aligned}$$

So, if the central limit theorem is in force with only ten random variables in the sum, we might expect y to be distributed as

$$y \sim N(5, 5/6)$$

Histogram of x

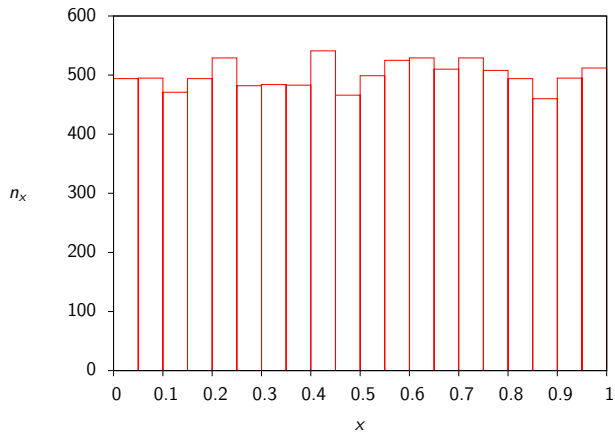


Figure 16: Histogram of 10,000 samples of uniformly distributed x .

Histogram of y

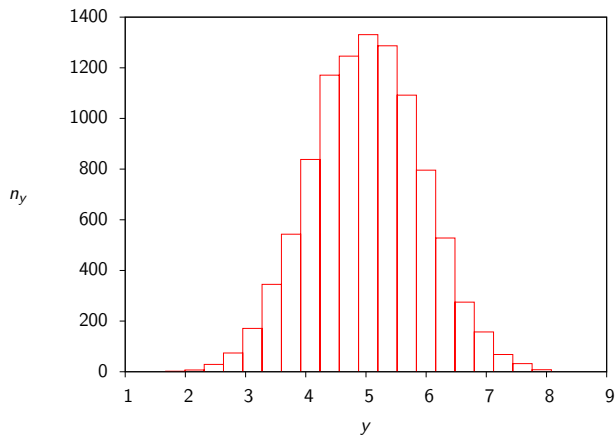


Figure 17: Histogram of 10,000 samples of $y = \sum_{i=1}^{10} x_i$.

We use the following Octave code (MATLAB code is similar) to generate 10,000 samples of the 10 uniformly distributed x random variables and add them together to make y .

```
nsam = 10000;
nsum = 10;
x = rand(nsam, nsum);
y = sum(x,2);
figure(1); hist(x,50)
figure(2); hist(y,50)
mx = mean(x(1,:))
varx = var(x(1,:))
my = mean(y)
vary = var(y)
```

Executing this code and then examining the means and variances of the first x , x_1 , and y gives

```
mx = 0.48859  
varx = 0.12250  
my = 4.9949  
vary = 0.80278
```

A histogram of the 10,000 samples of x_1 and y are shown in Figures 16 and 17. It is clear that even ten uniformly distributed x random variables produce nearly a normal distribution for their sum y .

Disturbances and measurement noise

So if we have a deterministic model for the state x (mass and energy balances), and we can measure y (concentration, temperature)

$$\begin{aligned}\frac{dx}{dt} &= f(x) \\ x(0) &= x_0 \\ y &= h(x) + v\end{aligned}$$

We may want to characterize the measurement error v . This measurement “noise” is often the combined effect of *many different kinds* of random variables.

The central limit theorem says that the zero mean normal distribution may be a good representation of random variable v in many applications.

Confidence intervals for the normal distribution

Assume \mathbf{x} is normally distributed with mean \mathbf{m} and covariance \mathbf{P} .
Figure 18 shows 1000 samples for

$$\mathbf{m} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 2 & 0.75 \\ 0.75 & 0.5 \end{bmatrix}$$

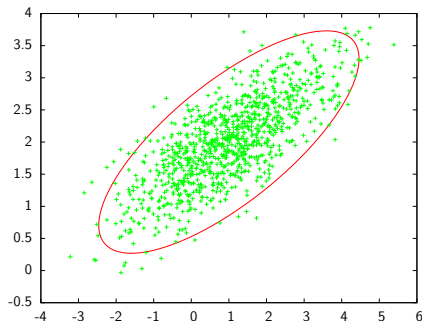


Figure 18: One thousand samples of the random variable $\mathbf{x} \sim N(\mathbf{m}, \mathbf{P})$ and 95% probability contour.

The samples were generated with the following code

```
m = [1;2];
n = length(m);
P = [2, 0.75; 0.75, 0.5];
nsam = 1000;

%% generate the samples x is N(m, P)
x = repmat(m, 1, nsam) + sqrtm(P)*randn(n,nsam);

%% compute the 95% confidence interval ellipse
alpha = 0.95;
A = inv(P);
b = chi2inv(alpha, n);
[xe, ye] = ellipse(A, b, 100, m);

%% plot samples and 95% confidence ellipse
plot(x(1,:), x(2,:), 'o', xe, ye)
```

The α -level probability ellipse for the normal is given by

$$(\mathbf{x} - \mathbf{m})' \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m}) \leq \chi^2(n, \alpha)$$

Notice from the code sample that the MATLAB function to return the value of $\chi^2(n, \alpha)$ is¹

```
chi2inv(alpha, n)
```

For $\alpha = 0.95$ and $n = 2$, we have $\chi^2(n, \alpha) = 5.99$, and each sample of \mathbf{x} has 95% probability of lying inside this ellipse. We use the function `ellipse(A, b, 100, m)` to generate the ellipse corresponding to $(\mathbf{x} - \mathbf{m})' \mathbf{A} (\mathbf{x} - \mathbf{m}) \leq b$. Notice we can make this plot only when $n = 2$. In higher dimensions we report the box containing the ellipse as shown in Figure 13.

¹In particular, notice that the order of the arguments (n, α) is reversed.

Figure 18 shows 1000 samples and we see that the samples indeed generate an elliptical region. For 1000 samples and $\alpha = 0.95$, we would expect that $1000(1 - 0.95) = 50$ samples would lie outside the 95% probability region. In Figure 18, the actual number is 44.

You can compute the number outside the ellipse with the code

```
e = x - repmat(m, 1, nsam);  
sum( diag(e'*inv(P)*e)> chi2inv(alpha, n) )
```

The first command moves the mean of the samples to zero and the second command measures how far each sample is from zero. The ones outside the ellipse have values of $\mathbf{e}'\mathbf{P}^{-1}\mathbf{e}$ greater than $\chi^2(n, \alpha)$ and the “greater than” test generates 1 for these samples and 0 for the others. The `sum` command then counts how many lie outside the ellipse.

In parameter estimation problems, the α -level probability ellipse is also known as an α -level confidence interval. It is customary to report 95% confidence intervals, although reporting several values of α may be appropriate depending on the application.

Least-Squares Estimation

Consider again the problem of fitting a straight line to data

$$y_i = mx_i + b$$

in which y_i is the measurement at x_i , $i = 1, \dots, n_d$ and n_d is the number of data points. Using matrix vector notation, we can write the equation for all the data as

$$\mathbf{y} = \mathbf{A}\boldsymbol{\theta}$$

in which the parameters to be estimated are placed in the $\boldsymbol{\theta}$ vector

$$\boldsymbol{\theta} = \begin{bmatrix} m \\ b \end{bmatrix}$$

and the \mathbf{y} vector and \mathbf{A} matrix are given by

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_d} \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_{n_d} & 1 \end{bmatrix}$$

Error as random variable

We do not expect the best fit line to pass through all the data points, so we modify the model to account for measurement error

$$\mathbf{y} = \mathbf{A}\boldsymbol{\theta}_0 + \mathbf{e} \quad (19)$$

in which $\boldsymbol{\theta}_0$ is the (unknown!) true parameter value vector and \mathbf{e} is a random variable. We model the measurement error as a normal distribution with mean $\mathbf{0}$ and variance $\sigma^2\mathbf{I}$.

$$\mathbf{e} \sim N(\mathbf{0}, \sigma^2\mathbf{I}) \quad (20)$$

The best estimate of $\boldsymbol{\theta}$ in a least-squares sense is given by

$$\hat{\boldsymbol{\theta}} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{y} \quad (21)$$

The distribution of parameter estimates

Imagine we create replicate datasets by drawing measurement errors \mathbf{e} from the distribution given in Equation 20. For each dataset we apply Equation 21 and produce a parameter estimate. The distribution of measurement errors creates a distribution of parameter estimates. In fact, for models linear in the parameters, we can show the parameter estimates also are normally distributed

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}_0, \mathbf{P})$$

in which the mean is the true value of the parameters and the covariance is

$$\mathbf{P} = \sigma^2(\mathbf{A}'\mathbf{A})^{-1}$$

Parameter confidence intervals

We also can calculate the parameter “confidence intervals.” We merely compute the size of the ellipse containing a given probability of the multivariate normal. As discussed previously, that can be shown to be the chi-square probability function (Box and Tiao, 1973, p. 116). Given the number of estimated parameters, n_p , and the confidence level, α , then

$$\frac{(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})' \mathbf{A}' \mathbf{A} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})}{\sigma^2} \leq \chi^2(n_p, \alpha) \quad (22)$$

The χ^2 distribution is tabulated in many statistics handbooks Box et al. (1978) and is available in many computing environments²

²The MATLAB command is `chi2inv(alpha, n)`.

Parameter Estimation with Differential Equation Models

Now we turn to the single most important parameter estimation problem in chemical reactor modeling: determining reaction-rate constants given dynamic concentration measurements.

$$\frac{dx}{dt} = f(x; \theta) \quad (23)$$

$$x(0) = g(x_0; \theta) \quad (24)$$

$$y = h(x) \quad (25)$$

in which x is the single material balance of interest, θ are the unknown model parameters, x_0 is the initial condition, and y is the experimentally measurable quantity.

Parameter Estimation with Differential Equation Models

$$\frac{dx}{dt} = f(x; \theta) \quad (26)$$

$$x(0) = g(x_0; \theta) \quad (27)$$

$$y = h(x) \quad (28)$$

For simplicity let us assume here that x itself is measured, in which case $h(x) = x$. It may be necessary to include some of the initial conditions also as unknown parameters.

Often t is time, but in steady-state tubular PFRs, reactor volume or length can take the place of time without changing the structure of the parameter-estimation problem.

Least squares objective

As before, we define a least-squares objective to measure our fit to the data

$$\Phi(\boldsymbol{\theta}) = \sum_i (\tilde{x}_i - x_i)^2 \quad (29)$$

in which \tilde{x}_i is the experimental measurement at time t_i , and x_i is the solution to the model at time t_i , $x_i = x(t_i; \boldsymbol{\theta})$. Note x_i is the only part of the objective function that depends on the model parameters. Again, we minimize this objective function to obtain our parameter estimates

$$\min_{\boldsymbol{\theta}} \Phi(\boldsymbol{\theta}) \quad (30)$$

subject to Equations 26–28.

Parameter estimation algorithm:

- ① Guess initial parameter values.
- ② Using an appropriate ODE solver, solve the model given the current parameter values. Compute x_j .
- ③ Evaluate Φ using Equation 29.
- ④ Update parameter values to minimize Φ . This step and the next are usually controlled by an optimization package.
- ⑤ Check convergence criteria. If not converged, go to ②.
- ⑥ On convergence, set $\hat{\theta}$ to current parameter values. Calculate approximate confidence intervals.

Estimating two rate constants in reaction $A \longrightarrow B \longrightarrow C$

Consider the irreversible series reactions



We wish to estimate the two rate constants k_1 and k_2 from the measurements shown in Figure 19. We would also like to know how much confidence to place in these estimated parameters.

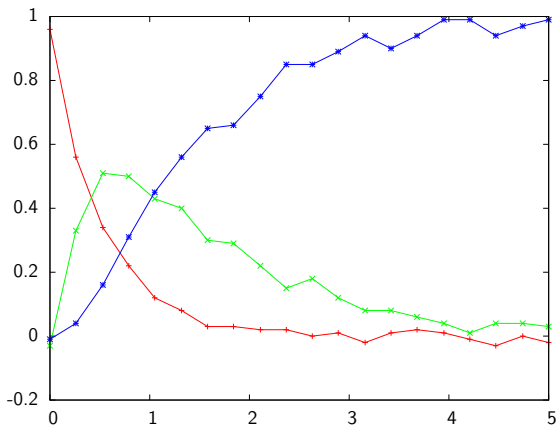


Figure 19: Measurements of species concentrations in Reactions 31 versus time.

Estimated parameters and confidence intervals

Running the `ABC.m` file produces the following parameter estimates and confidence intervals.

$$\hat{\theta} = \begin{bmatrix} 2.03 \\ 0.989 \end{bmatrix} \pm \begin{bmatrix} 0.229 \\ 0.0823 \end{bmatrix} \quad \theta_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

The estimates are close to the true values θ_0 used to generate the data. The confidence intervals are reasonably tight given the three species measurements with the noise levels in the data.

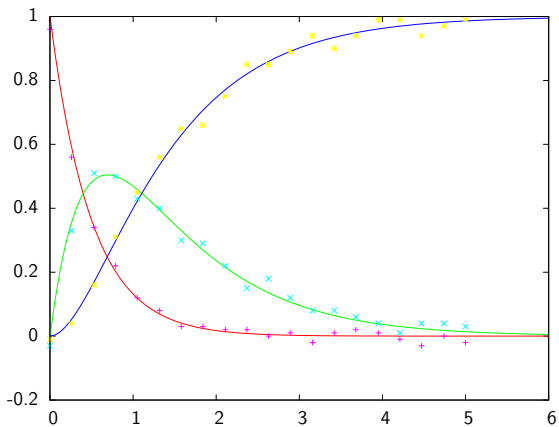
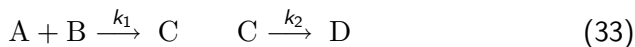
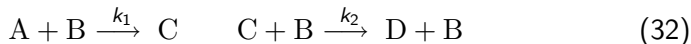


Figure 20: Fit of model to measurements using estimated parameters.

Does B catalyze the second reaction?

We have collected the laboratory data shown in Figure 21 and are considering the following two proposed kinetic mechanisms to explain these data



We have measurements of species A, C, and D, but not species B.

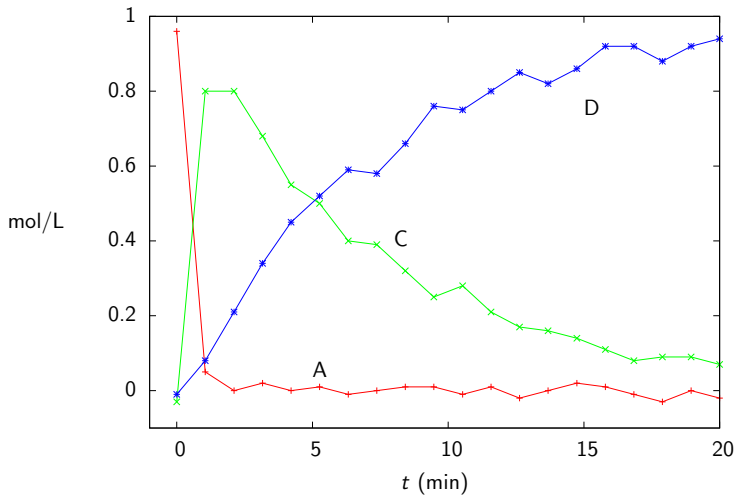


Figure 21: Measurements of species concentrations versus time.

Although we cannot measure species B, we do know that we charged the reactor with

$$\begin{aligned}c_A(0) &= 1 \text{ mol/L} & c_B(0) &= 10 \text{ mol/L} \\c_C(0) &= 0 \text{ mol/L} & c_D(0) &= 0 \text{ mol/L}\end{aligned}$$

We wish to find out whether or not species B catalyzes the second reaction.

- 1 Estimate the kinetic parameters for Reactions 32 and show the best parameter estimates and their confidence intervals. Plot the fit to the experimental data using the estimated parameters.
- 2 Repeat for Reactions 33.
- 3 Which mechanism better explains the data? Explain your answer.

The estimated parameters with B catalyzing the second reaction are

Estimated Parameters and Bounding
Boxes =

2.6853e-01	1.0642e-01
1.5807e-02	9.4527e-04

The fit to the data is shown in Figure 22. The model fit is good, and the parameter estimates have reasonably small confidence intervals.

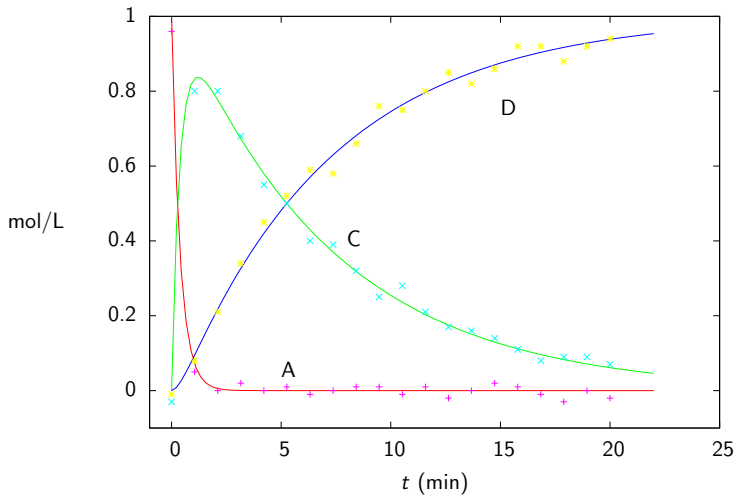


Figure 22: Fit of model to measurements for catalyzed second reaction.

Model discrimination fails

The estimated parameters with uncatalyzed second reaction are

Estimated Parameters and Bounding Box

ans =

0.2687392 0.1069366

0.1427191 0.0086426

The fit to the data is shown in Figure 23. Again we see the fit to the data is good, and the parameter estimates have reasonably small confidence intervals.

Neither mechanism is better. The fits are comparable and the confidence intervals are comparable. These experimental data do not tell us whether or not B catalyzes the second reaction.

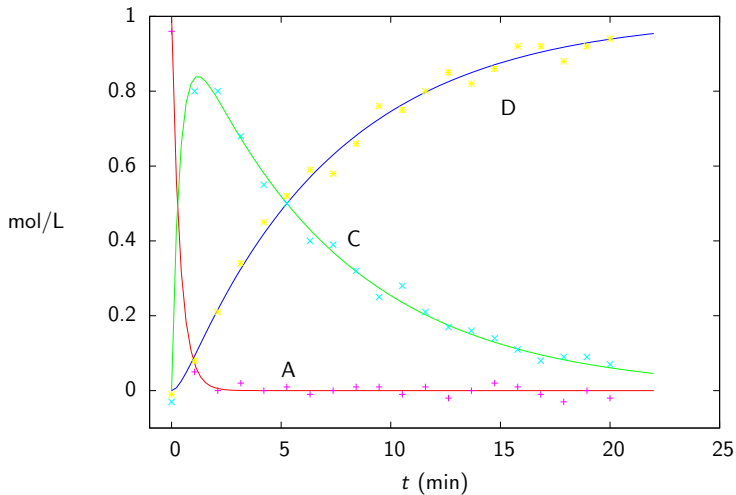
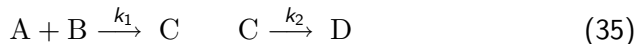
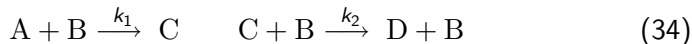


Figure 23: Fit of model to measurements for uncatalyzed second reaction.

Can you hear me now?

Consider again the two kinetic mechanisms



We have measurements of species A, C, and D, but not species B.

- 1 Estimate the parameter again, but with the experimental data shown in Figure 24 that were collected after charging the reactor with

$$\begin{aligned}c_A(0) &= 4 \text{ mol/L} & c_B(0) &= 3 \text{ mol/L} \\c_C(0) &= 1 \text{ mol/L} & c_D(0) &= 0 \text{ mol/L}\end{aligned}$$

- 2 Make a plot of the residuals of the model fits for the catalyzed and uncatalyzed mechanisms. Use `help parest` to see how the residuals are returned by `parest` so you can plot them.
What do you conclude from these residual plots? Does your conclusion change about which mechanism better explains these data?
- 3 Explain any differences in your conclusions from the two different data sets. Although we are measuring the same species in both experiments, do the two experiments contain different information? Explain why or why not.

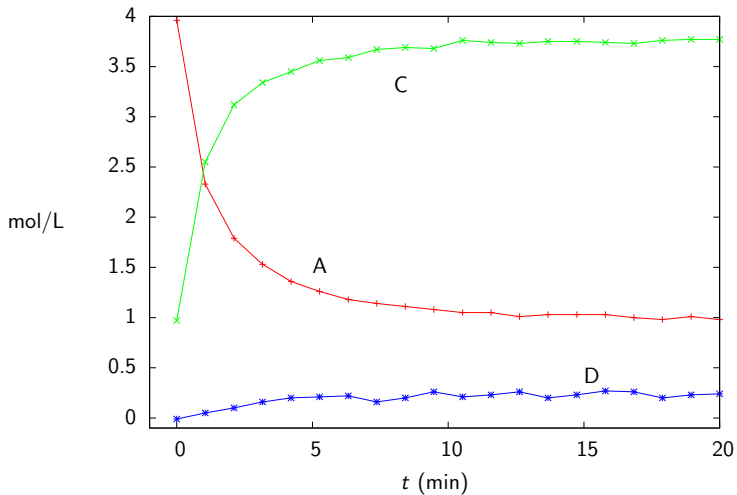


Figure 24: Measurements of species concentrations versus time.

The fit to this experiment using the catalyzed second reaction is shown in Figure 25.

The estimated parameters and confidence intervals are given by

Estimated Parameters and Bounding Box

ans =

0.2503297	0.0121540
0.0157937	0.0016184

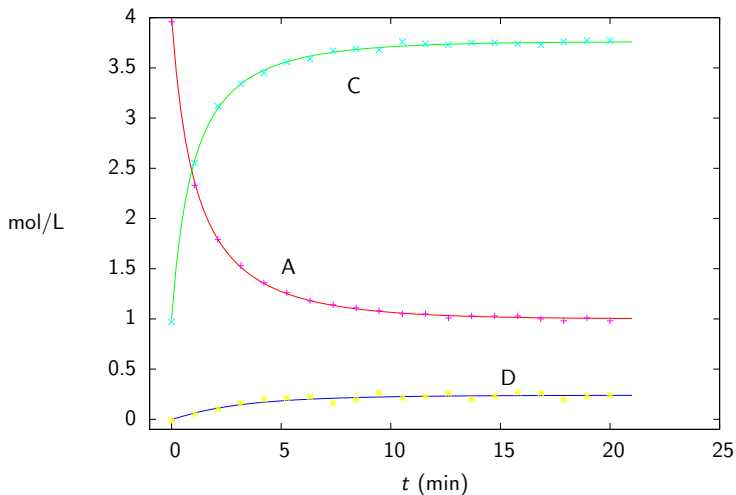


Figure 25: Fit of model to measurements for catalyzed second reaction.

The fit using the uncatalyzed second reaction is shown in Figure 26. The estimated parameters and confidence intervals are given by

Estimated Parameters and Bounding Box

ans =

0.2313643 0.0303810

0.0045569 0.0012070

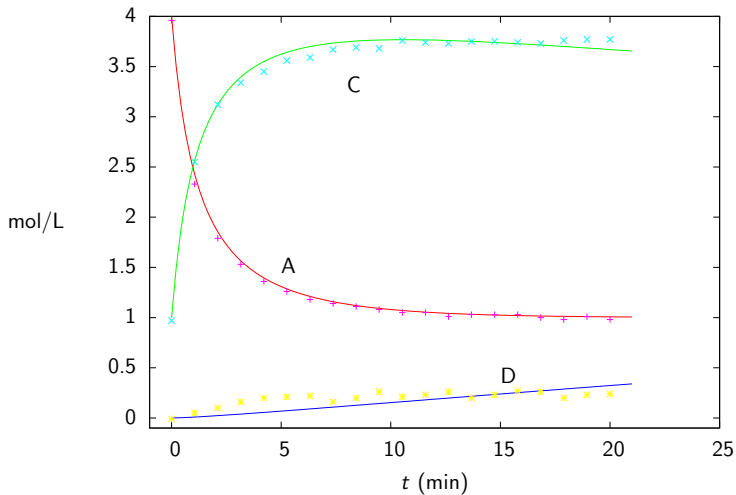


Figure 26: Fit of model to measurements for uncatalyzed second reaction.

The parameter confidence intervals are comparable for the two mechanisms, so we next look at the model residuals. Figures 27–28 show the model residuals for the catalyzed and uncatalyzed second reaction, respectively.

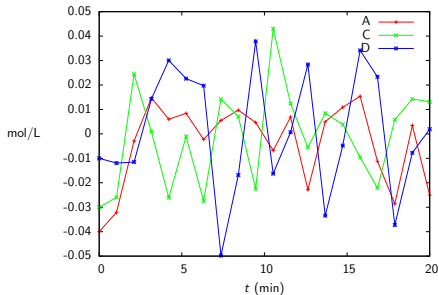


Figure 27: Residuals for catalyzed second reaction.

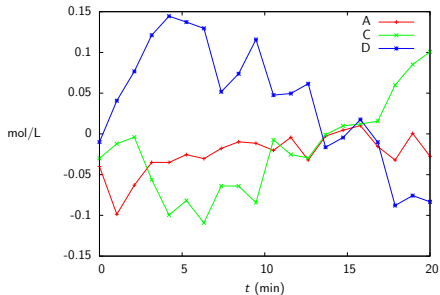


Figure 28: Residuals for uncatalyzed second reaction.

Model discrimination succeeds

Notice the residuals for the catalyzed second reaction are reasonably randomly distributed about zero. The residuals for the uncatalyzed second reaction, however, show long runs of positive followed by negative residuals. These long runs indicate some mismatch between model and data.

Therefore, we conclude based on the residuals that the first mechanism provides a better fit to the data compared to the second mechanism. We conclude the second reaction is catalyzed by B.

Experimental design for model discrimination

- Because we have used different initial conditions for the experiment, we do indeed have different information. The B concentration in the first experiment changes from 10 to 9 mol/L during the experiment, which is only a 10% change.
- The B changes from 3 to 0 mol/L in the second experiment, which is a large change. This wider change of B concentration allows us to determine better whether B is catalyzing the second reaction.
- We can now pose the question: what is the “best” experiment to discriminate between two candidate models. This leads into the subject of optimal experimental design.

What did we learn in this lecture?

- 1 Differential equation models
 - ▶ Writing them
 - ▶ Solving them numerically with software (MATLAB or Octave)
- 2 Modeling experimental measurements
 - ▶ Random variables and probability
 - ▶ Normal distribution
 - ▶ Confidence intervals
- 3 Estimating parameters from data
 - ▶ Objective function and constraints
 - ▶ Optimization algorithm
 - ▶ Approximate confidence intervals

Further reading

- G. E. P. Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison–Wesley, Reading, Massachusetts, first edition, 1973.
- G. E. P. Box, W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters*. John Wiley & Sons, New York, 1978.
- A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc., second edition, 1984.
- J. B. Rawlings and J. G. Ekerdt. *Chemical Reactor Analysis and Design Fundamentals*. Nob Hill Publishing, Madison, WI, 2002.
- J. B. Rawlings, C. T. Maravelias, and R. E. Swaney. CBE 255. Introduction to chemical process modeling. Online course instructional materials. <http://cbe255.che.wisc.edu>, 2009.