

5.68J/10.652J Feb 13, 2003

Numerical Solution to Kinetic Equations

Lecture 1

The Problem

We want to solve

$$\frac{1}{V} \frac{dn_i}{dt} = \sum v_i^l r_l(\bar{n}, T) - \text{Transport of 'i' out of } V \quad \text{Eq.(1)}$$

Where n_i is the number of moles of species 'i', v is the matrix of stoichiometric coefficients, and

$$r_l = \text{Rate of Step } l' \text{ (units: moles/volume-second)}$$

For a typical bimolecular elementary step reaction $A+B = C+D$:

$$r_l = k_{\text{forward}}(T)[A][B] - k_{\text{reverse}}(T)[C][D]$$

For a typical gas-phase dissociation reaction $A = C + D$:

$$r_l = k_{\text{forward}}(T, P)[A] - k_{\text{reverse}}(T, P)[C][D]$$

For a typical gas-phase isomerization reaction $A = C$:

$$r_l = k_{\text{forward}}(T, P)[A] - k_{\text{reverse}}(T, P)[C]$$

The rate constants also usually depend (weakly) on the chemical environment (e.g. dielectric constant), so they strictly are not really constant but (very weak) functions of all the n 's. Note that the forward and reverse rates are normally related by the equilibrium constant, related to the free energy of reaction (thermochemistry).

It is important to appreciate that except in the simplest cases, there is no analytical solution to this system of differential equations, so we MUST use the computer to solve them numerically. Approximations like quasi-steady-state extend the range of systems where analytical (approximate) solutions are possible, but you don't have to get very complicated before even the steady-state-approximation solutions are intractable.

More About the Fundamental Equation

Chemistry textbooks usually give instead of Eq. (1) the equation:

$$\frac{d[i]}{dt} = \sum V_i^l r_i(\bar{n}, T) \quad \text{Eq.(2)}$$

But Eq.(2) is true only if $dV/dt = 0$ (“isochoric”) and there is no transport (i.e. perfectly homogeneous, no convective flow). These textbook approximations are good for dilute reactions in solution or for constant volume apparatus (e.g. bomb calorimeter, shock tube). In many important cases the system is isobaric, i.e. $dP/dt = 0$, so if the temperature increases or there is a change in the number of moles of gas V will change.

It is convenient to express the equation in terms of intensive quantities rather than the extensive variables n and V . The intensive variables that work best are mass fractions y_i :

$$y_i = \frac{n_i W_i}{M_{total}} = \frac{[i] W_i}{\rho}$$
$$\frac{dy_i}{dt} = \frac{W_i}{\rho} \left(\frac{1}{V} \frac{dn_i}{dt} \right) - \text{Transport of 'i' out of } V$$

You can plug in the same expression as above for the term in parentheses. Note that mole fractions x_i are not such a good choice for kinetics as mass fractions, since the number of moles can change in chemical reactions, but the mass does not.

However one writes the chemical reaction equations, one must be explicit about the model for dT/dt , since the kinetics invariably depends strongly on T . The common textbook assumption is isothermal ($dT/dt = 0$) which is okay if the reaction is very slow and the reagents dilute. In many cases of interest the system is closer to being adiabatic, so dT/dt is given by energy conservation. (For the energy conservation equation around a small control volume with uniform T , P , and composition, and an elementary discussion of how ODE's are solved numerically, see:

http://web.mit.edu/10.10/www/Study_Guide/DiffEq.html). Note that to accurately compute dT/dt one must have a way of computing the enthalpies and heat capacities of all the species at any T .

Thermochemical Data

Because one needs the thermochemical data both to compute dT/dt and also to compute the reverse reaction rate constants, it is usual to build up a library file containing all this data for all the species of interest. For (stupid) historical reasons the data are usually represented as the coefficients in a piecewise fitting form known as “the NASA polynomials”. For some of the arcane details about NASA polynomials, see the CHEMKIN manual and http://www.me.berkeley.edu/gri_mech/data/nasa_plnm.html. Beware that there is more than one format for the NASA polynomials! The NIST Webbook gives thermochemical data in a different “Shomate” format which is better

behaved than the NASA form. The other popular form is the discrete values of Benson (used also by THERM and the NIST S&P program) discussed later in the course. All forms take advantage of the fact that the T-dependence of the enthalpy and the entropy is easily computed (often analytically) from the heat capacity. So the main issues are whether one knows H and S exactly at any T for the species of interest and then how well the T-dependence of the heat capacity (usually C_p , measured at constant pressure) can be determined and represented. Very often S and C_p are calculated using statistical mechanics rather than measured, you need to read carefully to determine whether the numbers reported in a table are measured or just computed, and if they are computed what assumptions were made. The reported S and H are usually for an 'ideal gas state at 1 atm at 298 K', even if the molecules are actually liquids or solids at 298 K and 1 atm. So the reported H(298) is often calculated, perhaps from an experimentally measured H at some different temperature and the computed heat capacities.

In many real-world applications of kinetics, the estimated barrier heights of the slow steps are strongly correlated with the estimates of their endothermicities, and many of the fast reactions are nearly equilibrated. Hence one is more sensitive to uncertainties in the thermochemical data than to the (usually much larger) uncertainties in the individual rate constants. So a kineticist MUST be concerned about the quality of the thermochemical data file employed in the simulations. There is plenty to worry about. For example, in 2001 it was discovered that the heat of formation of OH was computed incorrectly many years ago, and that therefore the many species whose thermochemistry was inferred relative to OH were also incorrect. Presumably many similar errors exist in the databases, waiting for some bright graduate student to identify them. Note that you will get observably different results if you use the default therm.dat file shipped with CHEMKIN or the GRI-MECH thermo file.

Beware: Inhomogeneity can develop Spontaneously

If dT/dt is nonzero, often gradients will develop in the system even if it was initially homogeneous due to heat losses at the boundaries. In some cases these temperature gradients can become extreme (e.g. in a flame). Pressure equilibrates at about the speed of sound, much faster than species concentration or temperature gradients can equilibrate, so very often it is a good approximation that P is uniform in a system. However, in some combustion systems the reaction is so fast that significant pressure gradients develop; this can cause loud noises or in an extreme cases detonations and other types of explosions. These inhomogeneities are reduced by using a smaller control volume; for reacting flow simulations people often use control volumes less than a millimeter across so that their approximation that the system has a single temperature and composition is accurate.

Required Inputs

What inputs do we need for a Kinetic Simulation?

- 1) A Thermo File (with data for all the species in the simulation)
- 2) Reaction Mechanism: A list of the reactions and forward rate constants
- 3) Reaction conditions (e.g. initial concentrations, T, P, timescale, adiabatic or isothermal, flow situation, etc.) to fully specify physical situation.

- 4) Simulation details: Specify outputs required, tolerances, numerical methods, initial guesses, etc.

In CHEMKIN, input 1 is called something like therm.dat and usually resides in the /data/ directory, input 2 is called something like chem.inp (and if there is surface chemistry an additional file surf.inp), and inputs 3 and 4 are combined in a single file, e.g. aurora.inp.

How are ODE's solved numerically? (the short short version)

If we define $Y = \{y_i, T\}$ and there is no transport, then Eq.(1) is of the form:

$$dY/dt = F(Y) \quad \text{and we usually know the initial conditions: } Y(t_0) = Y_0$$

The general procedure is to step forward in time with some formula

$$Y(t+\Delta t) = Y(t) + G(Y) \Delta t \quad \text{Eq.(3)}$$

where G is our best estimate of the average of $F(Y(t'))$ over the trajectory from $Y(t'=t)$ to $Y(t'=t+\Delta t)$, using lots of little timesteps Δt until we reach t_{final} . In the simplest approximation called Forward (or Explicit) Euler $G=F(Y(t))$. This turns out to be pretty inaccurate (just like the rectangle rule is not a very accurate way to compute numerical integrals) and also [numerically unstable](#) unless Δt is very small. The fundamental problem is that we don't know the trajectory $Y(t')$ – all we can do is extrapolate to estimate $Y(t')$ at future times. The extrapolation at each timestep introduces a little bit of error, and then the extrapolation at the next timestep is built on an incorrect starting point and an incorrect estimate of the slope $G(Y)$. If one is not careful, the errors can cumulate in a very unfavorable way, making the whole procedure numerically unstable (even if the ODEs and the physical situation they describe is perfectly stable).

There are two general ways of coping with this. One is to use relatively simple estimates of G and just use tiny timesteps Δt much smaller than any of the physical timescales in the system. This general approach is called “explicit”, the most famous algorithms that work this way are called “Runge-Kutta” methods. The alternative approach is to use very complicated methods for estimating G that are guaranteed to be numerically stable, these are called “implicit” methods because in these methods G is a function of $Y(t+\Delta t)$, so Eq.(3) becomes an implicit (usually) nonlinear system of equations, very difficult to solve. The hope is that by improving the numerical stability, one could use large Δt 's and so reduce the number of timesteps required to reach t_{final} .

The difficult aspect of chemical kinetics is that one often has reactive intermediates with lifetimes 10^{12} times shorter than the relatively inert starting materials. The short lifetimes of the reactive species force Δt in the explicit methods to be extremely small, so then a huge number of timesteps (and a correspondingly huge amount of CPU time) are required before the starting materials are consumed significantly. When the range of timescales is very large, the ODE system is called “stiff”. [Stiffness](#) introduces many numerical problems, even for implicit ODE solvers; several algorithmic tricks must

be used simultaneously in order to solve stiff systems without introducing huge numerical errors.

The first algorithm that could correctly handle stiff ODE systems was discovered by William Gear in the 1970's. The best programs now available for solving large stiff ODE systems are VODE, DASL/DASPK, and DAEPACK. I believe the current version of CHEMKIN calls VODE and DASL.