§9. Parameter estimation for ODE models

* Ref. Chapter 6 of P. Englezos & N. Kalogerakis, Applied Parameter Estimation for Chemical Engineers. Marcel Dekker (2000)

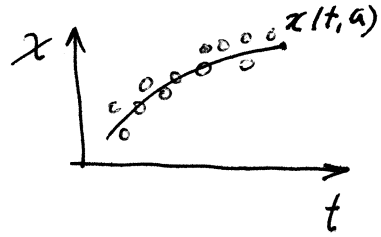
(a) Single equation, single parameter.

Consider ODE model

$$\frac{dx}{dt} = f(x, t, a), \quad x(t_0) = x_0$$

with parameter a, which will be chosen to best approximate measured data set (x_i, t_i), $i = 1, 2, \cdots, N$.

- Recall what we did with algebaic model $x = x(t, a)$:



  * Construct merit function

$$\chi^2 = \sum_{i=1}^{N} \frac{[x_i - x(t_i, a)]^2}{\sigma_i^2}$$

  * Minimize χ^2 with respect to a:

$$\frac{d\chi^2}{da} = 0 : G = -2 \sum_{i=1}^{N} \frac{x_i - x(t_i, a)}{\sigma_i^2} \cdot \left(\frac{\partial x}{\partial a}\right)\Big|_{t_i} = 0$$

Nonlinear algebraic equation, to be solved by Newton–Raphson ($\beta = -G/2$, $\alpha = -\partial\beta/\partial a$):

$$\alpha \, \Delta a = +\beta; \quad \alpha = \frac{-d\beta}{da} \approx \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(\frac{\partial x}{\partial a}\right)^2\Big|_{t_i}$$

with $\frac{\partial^2 x}{\partial a^2}$ term ignored in Hessian matrix. Now start with guessed a, improve it iteratively by correction Δa.

- But with ODE model, we don't have $x(t_i, a)$, and we don't have $\partial x/\partial a|_{t_i}$. What to do?

  * Get both through integrating/solving ODE's.

Take $\frac{\partial}{\partial a}$ of model $\frac{dx}{dt} = f(x, t, a)$:

$$\frac{\partial}{\partial a}\left(\frac{dx}{dt}\right) = \frac{\partial f}{\partial a} + \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial a}$$

$$\frac{d}{dt}\left(\frac{\partial x}{\partial a}\right) = \frac{\partial f}{\partial a} + \frac{\partial f}{\partial x} \cdot \left(\frac{\partial x}{\partial a}\right)$$

Calling $\boxed{\frac{\partial x}{\partial a} = y(t, a)}$ a new unknown function, we have:

$$\underline{z} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \frac{d\underline{z}}{dt} = \begin{pmatrix} f \\ \frac{\partial f}{\partial a} + \frac{\partial f}{\partial x} \cdot y \end{pmatrix}$$

$$z(t_0) = \begin{bmatrix} x_0 \\ 0 \end{bmatrix} \quad \text{since } \frac{d}{da}(x(t_0, a)) = \frac{d}{da}(x_0) = 0.$$

Therefore, we can get $x(t)$ and $\frac{\partial x}{\partial a} = y(t)$ at all t_i values, based on a guessed a values, and then solve the Newton-Raphson problem for Δa. This is the <u>Gauss-Newton</u> Method.

- Algorithm:

(i) Guess a value

(ii) Integrate ODE's for x and y. At each t_i, evaluate $x(t_i, a)$ and $y(t_i, a)$

(iii) Set up linear equation for Δa:

$$\alpha \cdot \Delta a = \beta,$$

$$\alpha = \sum_{i=1}^{N} \frac{1}{\sigma_i^2}\left(\frac{\partial x}{\partial a}\right)^2\Big|_{t_i} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} y_i^2$$

$$\beta = +\sum_{i=1}^{N} \frac{x_i - x(t_i, a)}{\sigma_i^2} \cdot y_i \qquad \text{[Left out factor of 2]}$$

(iv) Update a by $a + \Delta a \rightarrow a$

refinement $\begin{cases} * \text{ Check } \chi^2(a+\Delta a) \text{ to see if it's smaller than } \chi^2(a). \text{ Requires solving for } x(t, a+\Delta a) \\ * \text{ If not half } \Delta a: \ a + \frac{1}{2}\Delta a. \text{ Repeat.} \end{cases}$

(v) Iterate to convergence: $|\Delta a / a| < \varepsilon$.

(b) Single equation, multiple parameters.

$$\frac{dx}{dt} = f(x, t, a_1, a_2, \cdots, a_M), \quad x(t_0) = x_0.$$

To match N data points (x_i, t_i), $i = 1, \cdots, N$.

Following previous procedure: $\underline{a} = (a_1, \cdots, a_M)$

$$\beta_k = -\frac{1}{2}\frac{\partial \chi^2}{\partial a_k} = \sum_{i=1}^{N} \frac{x_i - x(t_i, \underline{a})}{\sigma_i^2} \cdot \frac{\partial x(t_i)}{\partial a_k}$$

$$\alpha_{kl} = -\frac{\partial \beta_k}{\partial a_l} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \frac{\partial x(t_i)}{\partial a_k} \frac{\partial x(t_i)}{\partial a_l} \quad \left[\text{ignored } \partial^2 x / \partial a_k \partial a_l\right]$$

$$\Rightarrow \quad \underline{\underline{\alpha}} \cdot \Delta \underline{a} = \underline{\beta}, \quad M \times M \text{ linear system.}$$

But first need to get $x(t_i, \underline{a})$ and $y_k = \dfrac{\partial x(t_i, \underline{a})}{\partial a_k}$:

$$\underline{z} = \begin{pmatrix} x \\ y_1 \\ \vdots \\ y_M \end{pmatrix}; \quad \frac{dz}{dt} = \begin{pmatrix} f \\ \frac{\partial f}{\partial a_1} + \frac{\partial f}{\partial x} y_1 \\ \vdots \\ \frac{\partial f}{\partial a_M} + \frac{\partial f}{\partial x} y_M \end{pmatrix}, \quad M+1 \text{ ODE's, with initial condition } z(t_0) = [x_0, 0, 0, \cdots, 0]^T.$$

Same algorithm as before, except that $\Delta \underline{a}$ now involves a linear system, and the ODE solution now for $M+1$ equations.

Example: Consider estimation of rate constants K_1, K_2 in Bodenstein-Linder model for homogeneous gas phase reaction $2NO + O_2 \longleftrightarrow 2NO_2$:

$$\frac{dx}{dt} = K_1(\alpha - x)(\beta - x)^2 - K_2 x^2, \quad x(0) = 0,$$

with $\alpha = 126.2$, $\beta = 91.9$, x being concentration of NO_2.
Choose K_1 and K_2 to match data set:

t_i	0	1	2	.....	24	29	39
x_i	0	1.4	6.3	.....	41.6	43.5	45.3

Solution: Now $f = K_1(\alpha - x)(\beta - x)^2 - K_2 x^2$ has 2 parameters. Note that α, β are 2 numbers, different from our Hessian matrix.

$$Z = \begin{pmatrix} x \\ \partial x/\partial K_1 \\ \partial x/\partial K_2 \end{pmatrix}, \quad \frac{\partial f}{\partial x} = -K_1(\beta - x)^2 - 2K_1(\alpha - x)(\beta - x) - 2K_2 x$$

$$\frac{\partial f}{\partial K_1} = (\alpha - x)(\beta - x)^2, \quad \frac{\partial f}{\partial K_2} = -x^2$$

$$\frac{dZ}{dt} = \begin{bmatrix} K_1(\alpha - x)(\beta - x)^2 - K_2 x^2 \\ (\alpha - x)(\beta - x)^2 + \frac{\partial f}{\partial x} \cdot y_1 \\ -x^2 + \frac{\partial f}{\partial x} \cdot y_2 \end{bmatrix}; \quad y_1 = \frac{\partial x}{\partial K_1}, \quad y_2 = \frac{\partial x}{\partial K_2}.$$

Write code to do this problem $\Longrightarrow$ get best estimation for K_1, K_2. [New assignment?].

(c) Multiple equations with multiple parameters.

$\underline{x} = (x_1, \cdots, x_n) \rightarrow$ index $j = 1, 2, \cdots, n$

$\underline{a} = (a_1, \cdots, a_m) \rightarrow$ index $k = 1, 2, \cdots, m$

$(\underline{x}_i, t_i)$: N data points $\rightarrow$ index $i = 1, 2, \cdots, N$.

$$\frac{d\underline{x}}{dt} = \underline{f}(\underline{x}, t, \underline{a}), \quad n \text{ equations}; \quad \underline{x}(t_0) = \underline{x}_0$$

$$\chi^2 = \sum_{i=1}^{N} \frac{\| \underline{x}_i - \underline{x}(t_i, \underline{a}) \|^2}{\sigma_i^2} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} [\underline{x}_i - \underline{x}(t_i)]^T \cdot [\underline{x}_i - \underline{x}(t_i)]^T$$

$$\frac{\partial \chi^2}{\partial a_k} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \sum_{j=1}^{n} 2 [x_i^j - x^j(t_i)] \left(- \frac{\partial x^j}{\partial a_k} \right)$$

Define $n \times m$ matrix $\underline{\underline{G}}_{jk} = \frac{\partial x^j}{\partial a_k}$, "Sensitivity matrix"

Then $\left[\frac{\partial \chi^2}{\partial a_k} \right] = \sum_{i=1}^{N} \frac{-2}{\sigma_i^2} [\underline{x}_i - \underline{x}(t_i)] \cdot \underline{\underline{G}}$; $1 \times m$ vector.

Now $\underline{\beta}_k = -\frac{1}{2} \frac{\partial \chi^2}{\partial a_k} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} [\underline{x}_i - \underline{x}(t_i)] \cdot \underline{\underline{G}}$

$\alpha_{kl} = -\frac{\partial \beta_k}{\partial a_l} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \underline{\underline{G}}^T \underline{\underline{G}}$, $m \times m$ matrix.

Again we're going to solve for $\Delta \underline{a}$ by

$$\underline{\underline{\alpha}} \cdot \Delta \underline{a} = \underline{\beta}$$

But first, we need to integrate ODE's for $\underline{x}(t, \underline{a})$ and $\underline{\underline{G}}(t, \underline{a})$:

$$\frac{\partial \underline{\underline{G}}}{\partial t} = \frac{\partial \underline{f}}{\partial \underline{a}} + \frac{\partial \underline{f}}{\partial \underline{x}} \cdot \frac{\partial \underline{x}}{\partial \underline{a}} = \frac{\partial \underline{f}}{\partial \underline{a}} + \frac{\partial \underline{f}}{\partial \underline{x}} \cdot \underline{\underline{G}}$$

$$\underline{\underline{G}}(t_0) = \underline{0}.$$

Writing $\underline{\underline{G}} = [\, g_1 \,|\, g_2 \,|, \cdots |\, g_m \,] = [\, \frac{\partial \underline{x}}{\partial a_k} \,]$ into m columns of n-dimensional vectors, then append $\underline{x}$ to the left:

$$\underline{\underline{Z}} = [\, \underline{x} \,|\, g_1 \,| \cdots |\, g_m \,]$$

Such that

$$\begin{cases} \dfrac{d\underline{\underline{Z}}}{dt} = \left[\underline{f} \,\middle|\, \dfrac{\partial \underline{f}}{\partial a_1} + \dfrac{\partial \underline{f}}{\partial \underline{x}} \cdot g_1, \cdots, \dfrac{\partial \underline{f}}{\partial a_k} + \dfrac{\partial \underline{f}}{\partial \underline{x}} \cdot g_k, \cdots, \dfrac{\partial \underline{f}}{\partial a_m} + \dfrac{\partial \underline{f}}{\partial \underline{x}} g_m \right] \\ \underline{\underline{Z}}(t_0) = [\, \underline{x}_0, \underline{0}, \cdots, \underline{0} \,] \end{cases}$$