

# Determinación de parámetros

Supongamos un modelo dinámico

$$\frac{dx}{dt} = f(x, \theta, u, t) \quad x(t=0) = x_0$$

$$y = g(x, \theta, u, t)$$

$x$  = variables de estado  
 $y$  = variables de salida  
 $u$  = variables de entrada  
 $\theta$  = parámetros

Para un conjunto dado de mediciones (con su correspondiente ruido de medición experimental), estimar los parámetros desconocidos del modelo (vector  $\theta$ ).

1

2

La identificabilidad *estructural* de un modelo puede definirse de la siguiente manera: dada la estructura de un modelo y datos perfectos (en el sentido de que no tienen error) de las variables ¿es posible llegar a un único conjunto de valores de parámetros?

P.ej. supongamos un CSTR con una reacción de primer orden  $A \rightarrow B$ . Del balance de masa

$$\frac{dC_A}{dt} = D(C_{Ain} - C_A) - kC_A$$

Los parámetros a determinar serían  $D$  y  $k$ . Supongamos que tenemos medidas de  $C_A$  y  $C_{Ain}$ . Entonces, por ejemplo haciendo un escalón en la entrada tendremos una respuesta

$$C_A(t) = C_{A0} + \frac{D}{D+k} \Delta C_{in} (1 - e^{-(D+k)t})$$

A  $t = \infty$  la concentración de salida habrá aumentado  $\frac{D}{D+k} \Delta C_{in}$  y como conocemos  $\Delta C_{in}$  sabremos también  $\frac{D}{D+k}$ . Considerando otro punto, p.ej. el 63% de la respuesta corresponde a  $t = D+k$  y por lo tanto de ambos puntos podemos calcular ambos parámetros  $D$  y  $k$ . El modelo es identificable.

3

4

Pero supongamos ahora que la concentración se mide con un sensor lineal  $y = y_C C_A$  siendo el parámetro  $y_C$  desconocido. Ahora tenemos tres parámetros y el salto de la respuesta a  $t$  infinito es  $y_C \frac{D}{D+k} \Delta C_{in}$  con lo cual no se pueden identificar los tres parámetros. Si se pueden identificar las combinaciones  $\theta_1 = D+k$  y  $\theta_2 = y_C D$

Si el modelo es **lineal**

$$\frac{dx}{dt} = A(\theta)x(t, \theta) + B(\theta)u(t), \quad x(0, \theta) = x_0(\theta)$$

$$y(t, \theta) = C(\theta)x(t, \theta)$$

Se puede caracterizar por su función de transferencia:

$$H(s, \theta) = C(\theta)(sI - A(\theta))^{-1}B(\theta)$$

En el ejemplo anterior  $H(s) = \frac{b}{s+a}$  con  $a = D+k$   
 $b = D$  or  $y_C D$

Como generalización, si la función de transferencia es de orden  $n$  se pueden determinar  $2n$  parámetros.

Para sistemas no lineales lo anterior funciona como condición suficiente, pues en un entorno adecuado cualquier sistema se puede aproximar a uno lineal.

5

6

Alternativas para la estimación de parámetros

- Ensayo y error
- Métodos estadísticos:
  - Frecuentista (máxima verosimilitud, mínimos cuadrados, regresión no lineal, etc.).
  - Bayesiano (Metropolis-Hasting, Markov Chain Monte Carlo (MCMC), muestreo de importancia, etc.).
  - Pragmático/híbrido (que emplea algunos elementos de las dos escuelas de pensamiento anteriores, por ejemplo, el método bootstrap, métodos Montecarlo, etc.).

7

La estimación puede interpretarse desde un punto de vista estadístico: p.ej. si las observaciones son independientes y con distribución normal, entonces es un estimador de máxima probabilidad

$$\bar{y}_n = \frac{1}{N} \sum_{i=1}^N y_i$$

También puede verse como un problema de minimización con la siguiente función objetivo:

$$J(\bar{a}) = \sum_{i=1}^N (y_i - \bar{a})^2$$

Derivando e igualando a cero  $\frac{\partial J(\bar{a})}{\partial \bar{a}} = 2 \sum_{i=1}^N (y_i - \bar{a}) = 2 \sum_{i=1}^N y_i - 2N\bar{a} = 0$

se obtiene 
$$\bar{a} = \bar{y}_n = \frac{1}{N} \sum_{i=1}^N y_i$$

9

Pasos preliminares en la estimación de parámetros

- Selección de subconjuntos de datos para estimación y validación. Un primer subconjunto se utiliza para obtener los parámetros y el otro para validar el modelo con esos parámetros. Es fundamental que los subconjuntos sean independientes.
- Selección de los parámetros a ser estimados. A veces el análisis de la estructura del modelo nos lleva a identificar los parámetros o grupos de parámetros a estimar. P.ej. si tenemos una combinación de parámetros como  $(1-y)\frac{\mu x}{y}$  elegimos uno de ellos para estimar (p.ej.  $\mu$ ) y fijamos los valores de los otros dos. En caso de cambiar estos últimos no sería necesario recalcular  $\mu$ .
- Estudios de sensibilidad
- Problemas numéricos puede obligar a reestructurar el problema de estimación de parámetros subdividiéndolo en problemas menores (p.ej. estimar primero coeficientes estequiométricos y luego coeficientes cinéticos).

11

Un ejemplo introductorio: estimar el valor medio de un set de datos

En bloque, se consideran todos los datos a la vez  $\bar{y}_n = \frac{1}{N} \sum_{i=1}^N y_i$

Método recursivo, se va expandiendo el set de datos durante el cómputo; las estimaciones intermedias convergen hacia la solución que se obtiene en bloque cuando se tienen todos los datos

$$\hat{a}_0 = 0$$

$$\hat{a}_i = \hat{a}_{i-1} - \frac{1}{k}(\hat{a}_{i-1} - y_i)$$

Método iterativo, se dispone de todos los datos pero se va actualizando la estimación en pasos sucesivos

8

Supongamos un modelo no lineal  $y = x^\theta$

donde  $\theta$  es el parámetro a ser estimado

Planteamos la función objetivo  $J(\theta) = \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \sum_{i=1}^N (y_i - x_i^\theta)^2$

$$\frac{\partial J(\theta)}{\partial \theta} = -2 \sum_{i=1}^N (y_i - x_i^\theta) \ln(x_i) x_i^{\theta-1} = 0$$

Como no es fácil de despejar puede usarse un método iterativo:

$$\sum_{i=1}^N y_i \ln(x_i) x_i^\theta = \sum_{i=1}^N \ln(x_i) x_i^{2\theta}$$

10

Método frecuentista

- En el problema de estimación de parámetros solemos definir estimadores de parámetros,  $\hat{\theta}$ , para distinguirlos de los verdaderos parámetros del modelo,  $\theta$ . En el contexto de la estimación estadística los parámetros se consideran desconocidos, y se utilizan métodos estadísticos para inferir su valor real.
- Se trata de encontrar estimadores,  $\hat{\theta}$ , a partir de un conjunto dado de mediciones,  $y$ .
- Los parámetros del modelo  $\theta$  se tratan como verdaderos y fijos, pero sus correspondientes estimadores,  $\hat{\theta}$ , se tratan como variables aleatorias.

$$y = f(\theta) + \varepsilon \quad \text{con} \quad \varepsilon \sim N(0, \sigma)$$

12

Los errores de medida son definidos con una distribución de probabilidad, normalmente la distribución normal N, con media 0 y desviación estándar  $\sigma$ . Con estas suposiciones la función de máxima verosimilitud ("likelihood") es

$$L(y, \theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y - f(\theta))^2}{2\sigma^2}\right)$$

La estimación más probable de  $\theta$  se encuentra en aquellos valores de los parámetros que minimizan la función de verosimilitud

$$\hat{\theta}: \min_{\theta} L(y, \theta)$$

La solución a este problema se suele encontrar mediante algoritmos de optimización, como el simplex, algoritmos genéticos, simulated annealing, etc.

13

El método de los mínimos cuadrados es un caso especial del método de máxima verosimilitud en el que se supone que las mediciones son independientes e idénticamente distribuidas con una desviación gaussiana típica conocida,  $\sigma$ . La función de verosimilitud equivale a minimizar la siguiente función de costo (o función objetivo):

$$F(y, \theta) = \sum \frac{(y - f(\theta))^2}{\sigma^2}$$

14

La estimación de parámetros habitualmente se formula como un problema de minimización, siendo la función objetivo a minimizar:

$$J(\theta) = \sum_{i=1}^N (y_i(\hat{\theta}) - y_i)^T Q_i (y_i(\hat{\theta}) - y_i)$$

en la que  $y_i$  e  $y_i(\hat{\theta})$  son vectores de  $N$  valores medidos y predicciones del modelo en tiempos  $t_i$  ( $i = 1$  a  $N$ ) respectivamente, y  $Q_i$  es una matriz cuadrada con coeficientes de ponderación.

El valor esperado de la función objetivo para un conjunto de parámetros ligeramente diferente del óptimo viene dado por:

$$E[J(\theta + \delta\theta)] \cong \delta\theta^T \left[ \sum_{i=1}^N \left(\frac{\partial y}{\partial \theta}(t_i)\right)^T Q_i \left(\frac{\partial y}{\partial \theta}(t_i)\right) \delta\theta + \sum_{i=1}^N \text{tr}(C_i Q_i) \right]$$

en la que  $C_i$  representa la matriz de covarianza del error de medición ( $Q_i$  se suele elegir como  $C_i^{-1}$  y el segundo término se reduce a un escalar).

15

$$E[J(\theta + \delta\theta)] \cong \delta\theta^T \left[ \sum_{i=1}^N \left(\frac{\partial y}{\partial \theta}(t_i)\right)^T Q_i \left(\frac{\partial y}{\partial \theta}(t_i)\right) \delta\theta + \sum_{i=1}^N \text{tr}(C_i Q_i) \right]$$

Maximizar el término entre [ ] maximiza la diferencia entre  $J(\theta + \delta\theta)$  y  $J(\theta)$  o, en otras palabras, garantiza que el ajuste de un conjunto de parámetros ligeramente diferente del mejor conjunto de parámetros sea significativamente peor.

Matriz de información de Fisher  $F = \sum_{i=1}^N \left(\frac{\partial y}{\partial \theta}(t_i)\right)^T Q_i \left(\frac{\partial y}{\partial \theta}(t_i)\right)$

Esta matriz es, de hecho, la inversa de la matriz de covarianza del error de estimación del parámetro del mejor estimador lineal insesgado:

$$V = F^{-1} = \left( \sum_{i=1}^N \left(\frac{\partial y}{\partial \theta}(t_i)\right)^T Q_i \left(\frac{\partial y}{\partial \theta}(t_i)\right) \right)^{-1}$$

16

Otra alternativa es la estimación Bayesiana que tiene en cuenta las funciones de densidad de probabilidad

$$p(\hat{\theta}|y) = \frac{p(y|\hat{\theta})}{p(y)} p(\hat{\theta})$$

También pueden utilizarse otras funciones objetivo, por, ej., la desviación absoluta, minimizar el máximo error absoluto, el número de cambios de signos en los residuales, la autocorrelación, etc.

17

Estimación de parámetros lineales - Cuando el modelo es lineal en los parámetros, en general se puede escribir

$$y_i = \phi_i^T \theta$$

Si planteamos WLS  $J(\theta) = \sum_{i=1}^N w_i (y_i - \hat{y}_i(\theta))^2$

Y la estimación de parámetros  $\hat{\theta} = \left[ \sum_{i=1}^N w_i \phi_i \phi_i^T \right]^{-1} \sum_{i=1}^N w_i y_i \phi_i^T$

Ejemplo:  $\frac{dS}{dt} = DS_n - DS - \frac{1}{V} \mu XS - mX$

$$y_i = \frac{dS}{dt} \Big|_{t_i} = DS_{n,i} + DS_i, \quad \phi_i = \begin{bmatrix} \frac{1}{V} X_i S_i \\ X_i \end{bmatrix}, \quad \theta = \begin{bmatrix} \mu \\ m \end{bmatrix}$$

18

Las no linealidades complican la estimación de parámetros. Pej.

$$\frac{dS}{dt} = DS_0 - DS - \frac{1}{Y} \mu_{max} X$$

$$\frac{dX}{dt} = -DX + \mu_{max} X$$

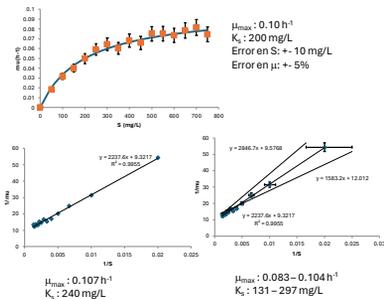
es no lineal en los parámetros  $Y$  y  $\mu_{max}$

Pero redefiniendo  $\theta_1 = \frac{1}{Y} \mu_{max}$   $\theta_2 = \mu_{max}$

el sistema queda escrito en forma lineal respecto a los parámetros, más allá de que se pierda el significado físico.

19

20



21

22

A veces se transforma el modelo para que quede lineal (Obs. No es una linealización en el sentido de Taylor). La más conocida es la transformación de Lineweaver-Burk para la expresión de Monod:

$$\mu = \mu_{max} \frac{S}{K_s + S} \Rightarrow \frac{1}{\mu} = \frac{K_s}{\mu_{max} S} + \frac{1}{\mu_{max}}$$

La nueva expresión es lineal en las nuevas variables  $1/S$  y  $1/\mu$ . Pero estas nuevas variables ya no mantienen la característica de ser "independientes e igualmente distribuidas en forma normal"

Optimización no lineal usando el solver de Excel, partiendo de  $\mu_{min} = 0.15$  y  $K_s = 300$

```

mu_min = 0.1021913104
Ks = 200.20213176
mu         modelo   [mu - modelo]^2
S          0         0
50         0.021843803 0.01965833 1.548846E-06
100        0.021794868 0.022926109 1.818356E-06
150        0.040007241 0.042952154 6.271518E-06
200        0.049802327 0.048951647 1.075705E-05
250        0.024804118 0.05484187 1.218792E-05
300        0.0646313 0.065052793 2.03721E-05
350        0.062381796 0.06378851 1.234737E-05
400        0.068141224 0.06692937 1.469329E-06
450        0.064582624 0.06995195 1.388935E-05
500        0.074028223 0.07188689 1.013746E-05
550        0.075121123 0.07384796 2.170246E-06
600        0.077193995 0.07558424 4.180166E-06
650        0.077188771 0.07711846 1.632138E-06
700        0.081560024 0.07848196 9.417428E-06
750        0.074949488 0.07979712 2.746794E-05
sum = 0.000124181
    
```

También se puede trabajar con el sistema  $\mu = \frac{S}{\theta_1 S + \theta_2}$

donde los parámetros tienen el mismo significado que en L-B pero la optimización es no lineal

```

mu_min = 0.1021913104
Ks = 200.20213176
mu         modelo   [mu - modelo]^2
S          0         0
50         0.021843803 0.01965833 1.548846E-06
100        0.021794868 0.022926109 1.818356E-06
150        0.040007241 0.042952154 6.271518E-06
200        0.049802327 0.048951647 1.075705E-05
250        0.024804118 0.05484187 1.218792E-05
300        0.0646313 0.065052793 2.03721E-05
350        0.062381796 0.06378851 1.234737E-05
400        0.068141224 0.06692937 1.469329E-06
450        0.064582624 0.06995195 1.388935E-05
500        0.074028223 0.07188689 1.013746E-05
550        0.075121123 0.07384796 2.170246E-06
600        0.077193995 0.07558424 4.180166E-06
650        0.077188771 0.07711846 1.632138E-06
700        0.081560024 0.07848196 9.417428E-06
750        0.074949488 0.07979712 2.746794E-05
sum = 0.000124181
    
```

En general para las estimaciones numéricas se parte de un valor inicial. Es importante contar con una buena aproximación inicial para evitar errores numéricos y acelerar el cálculo. En la medida de que sea posible considerar conocimiento previo.

- Si se cuenta con una estimación de los parámetros no lineales no es importante tener una buena aproximación para los que si son lineales. Pej. si hay que ajustar  $y(t) = \theta_1 e^{-\theta_2 t}$ . Y se sabe que  $\theta_1 = 6$ , podemos obtener una aproximación de  $\theta_2$ , usando mínimos cuadrados con  $J = \sum [y_i - \theta_1 e^{-\theta_2 t}]^2$

- Recurrir a una reparametrización para dejar el sistema en forma lineal
- Estimar en etapas. Pej. si hay que estimar los parámetros de  $r = k e^{-\frac{E}{RT}} S$  y tenemos valores de  $r$  para diferentes temperaturas  $r = k_1 S$  tomando logaritmos tenemos un sistema lineal  $\log(k_1) = \log(k) - \frac{E}{RT}$  y pueden extraerse valores de  $k$  y de  $E$  para la aproximación inicial

- Simplificar el modelo

23

24

Restricciones de desigualdad

- Funciones de penalización

$$J_{con \text{ restricción}} = J_{sin \text{ restricción}} + \sum J_{penalización}$$

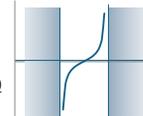
Donde en general la función de penalización es del tipo  $J_{penalización} = \frac{\alpha}{\epsilon} f(\theta)$

con  $\alpha$  un valor pequeño respecto a  $J_{sin \text{ restr.}}$ . Para asegurar que la penalización sea pequeña cuando estamos en la región permitida

- Transformación de parámetros, p.ej.

$$\phi = \tan\left(\frac{\pi}{2} \frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}}\right)$$

$$\theta = \frac{1}{2}(\theta_{min} + \theta_{max}) + (\theta_{max} - \theta_{min}) \frac{\arctan(\phi)}{\pi}$$



Cuando solo se necesitan valores positivos se puede transformar mediante la función exponencial/logaritmo

Estimación de parámetros no lineales – En general se parte de una estimación inicial, se integran las ecuaciones y se verifica si se cumple o no cierto criterio de ajuste; en caso de que no se cumpla se vuelve a reiniciar la iteración con un nuevo valor. El mayor peligro es caer en óptimos locales.

Algoritmos que usan la información de la derivada:

- Steepest descent
- Gauss-Newton
- Levenberg-Marquardt
- Quasi-Newton

Algoritmos que son independientes de la derivada:

- Rosenbrok
- Brent
- Simplex (Nelder-Mead)

Algoritmos de minimización global:

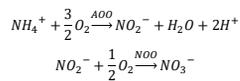
- Grilla
- Simulated Annealing
- Genetic Algorithms
- Controled Random Search

25

## Ejemplo práctico de determinación de parámetros

26

Estimación de los parámetros de los procesos de oxidación de amonio y nitrito a partir de datos de ensayos batch. –



AOO = Ammonia Oxidizer Organisms  
NOO = Nitrite Oxidizer Organisms

27

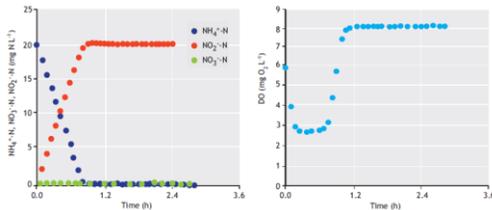
Table 5.1 The two-step nitrification model structure using matrix representation (adopted from Sin et al., 2008)

Variables—C	S <sub>in</sub>	S <sub>i</sub>	S <sub>ii</sub>	S <sub>iii</sub>	X <sub>in</sub>	X <sub>ii</sub>	Rates
Processes j,j	mg N L <sup>-1</sup>	mg O <sub>2</sub> L <sup>-1</sup>	mg N L <sup>-1</sup>	mg N L <sup>-1</sup>	mg COD L <sup>-1</sup>	mg COD L <sup>-1</sup>	q
AOO growth	-1	1 - $\frac{2.43}{Y_{AOO}}$	$\frac{1}{Y_{AOO}}$		1		$\mu_{max}^{AOO} \cdot M_{NH_4} \cdot M_{O_2} \cdot X_{AOO}$
AOO decay					-1		$b_{AOO} \cdot X_{AOO}$
NOO growth		1 - $\frac{1.14}{Y_{NOO}}$	$\frac{-1}{Y_{NOO}}$	$\frac{1}{Y_{NOO}}$		1	$\mu_{max}^{NOO} \cdot M_{NO_2} \cdot M_{O_2} \cdot X_{NOO}$
NOO decay						-1	$b_{NOO} \cdot X_{NOO}$
Aeration		1					$KL_a \cdot (S_{in}^{O_2} - S_i)$

$$M_{NH_4} = \frac{S_{NH_4}}{S_{NH_4} + K_{s,NH_4}}, M_{O_2, AOO} = \frac{S_{O_2}}{S_{O_2} + K_{s,O_2, AOO}}, M_{O_2, NOO} = \frac{S_{O_2}}{S_{O_2} + K_{s,O_2, NOO}}, M_{NO_2} = \frac{S_{NO_2}}{S_{NO_2} + K_{s,NO_2}}$$

28

Se realizan ensayos batch aeróbicos. Se realizaron dos ensayos por lotes de la siguiente manera: (i) ensayo 1 con adición de amonio de 20 mg NH<sub>4</sub>-N L<sup>-1</sup> y un inhibidor (azida sódica) para suprimir la actividad de NOO. (ii) ensayo 2 con amonio añadido de 20 mg NH<sub>4</sub>-N L<sup>-1</sup> sin ningún inhibidor. En ambos ensayos, tanto el pH como la temperatura se controlan a 7,5 y 25 °C respectivamente. Durante ambos ensayos por lotes, el amonio, el nitrito y el nitrato se miden cada 5 minutos, mientras que el oxígeno disuelto se mide cada minuto. Los datos recogidos se muestran en la figura 5.2.



29

Utilizaremos primero los datos de la prueba 1 (sin actividad de NOO)

El ensayo no fue diseñado para medir el decaimiento por lo que exceptuaremos el parámetro b<sub>AOO</sub>

Conjunto de medidas: Y = [NH<sub>4</sub> NO<sub>2</sub> DO]

Subconjunto de parámetros: θ = [Y<sub>AOO</sub> μ<sub>max</sub><sup>AOO</sup> K<sub>s, AOO</sub> K<sub>s, AOO</sub>]

Vamos a minimizar la suma de cuadrados de los errores, por ejemplo, con un método de búsqueda directa (Nelder-Mead)

30

Lo primero que hacemos es cargar los datos experimentales y parámetros ya conocidos:

```

1 % estimación de primer set de parámetros Yaob, mum_aob, Ks_aob, Ko_aob
2 clear all;clf
3 % carga datos experimentales
4 pkg load io
5 z = xlsread('am_nit_oxid.xlsx','Hoja1','A2:D18');
6
7 % parámetros conocidos
8 b_aob = 0.1; % d-1
9 kLa = 360; % d-1
10 So_sat = 8; % mgO2 L-1
11 p1 = [b_aob kLa So_sat];

```

31

Si conocemos algún valor de literatura o estimación previa podemos usarlos como valores semilla y con ellos determinar los límites superior e inferior de los parámetros

```

12
13 %valores semilla
14 %Yaob_init = 0.15; % gCOD gN-1
15 %mum_aob_init = 0.8; % d-1
16 %Ks_aob_init = 0.4; % mgN L-1
17 %Ko_aob_init = 0.5; % mgO2 L-1
18 %p_init = [Yaob_init mum_aob_init Ks_aob_init Ko_aob_init];
19 % limite inferior de los parámetros
20 p_inf = [0.11 0.50 0.14 0.10];
21 % limite superior de los parámetros
22 p_sup = [0.21 2.10 1.00 1.45];
23 p_init = (p_inf + p_sup)/2;

```

32

Cargamos las condiciones iniciales:

```

25 % condiciones iniciales
26 Snh0 = 20; % mgN/L
27 So0 = 8; % mgO2/L
28 Sno20 = 0; % mgN/L
29 Xaob0 = 75; % mgVSS/L
30 x0 = [Snh0 So0 Sno20 Xaob0];

```

33

Definimos la función que queremos optimizar: se integra con el set de parámetros la función que contiene las derivadas en el tiempo,

```

1 function y = der_f(x,t,p,p1)
2   Snh=x(1); So=x(2); Sno2=x(3); Xaob=x(4);
3
4   Yaob=p(1); mum_aob=p(2); Ks_aob=p(3); Ko_aob=p(4);
5   b_aob = p1(1); kLa = p1(2); So_sat = p1(3);
6
7   % Petersen matrix
8   K = [-1/Yaob (1-3.43/Yaob) 1/Yaob 1;
9         0 0 0 -1;
10        0 1 0 0];
11
12 % rates
13 r = [mum_aob*Xaob*Snh/(Snh+Ks_aob)*So/(So+Ko_aob);
14      b_aob*Xaob;
15      kLa*(So_sat - So)];
16
17 y = K'*r;
18 endfunction

```

34

se calcula el error respecto a los datos experimentales y la función a optimizar en este caso es mínimos cuadrados del error:

```

1 function J = cost(p,z,p1,x0)
2
3   t_exp = z(:,1)/60/24; % t en d (el original está en minutos)
4
5   Yaob=p(1); mum_aob=p(2); Ks_aob=p(3); Ko_aob=p(4);
6   b_aob = p1(1); kLa = p1(2); So_sat = p1(3);
7
8   x = lsode(@(x,t) der_f(x,t,p,p1),x0,t_exp);
9
10  e = x(:,1:3) - z(:,2:4); e = e(:);
11  J = e'*e;
12 endfunction

```

35

Y ahora estamos en condiciones de realizar una búsqueda directa

```

32 % búsqueda directa con fminsearch
33 p_opt = fminsearch(@cost(p,z,p1,x0),p_init)
34

```

```

p_opt =
    0.1348    1.3475    0.4832    0.7030

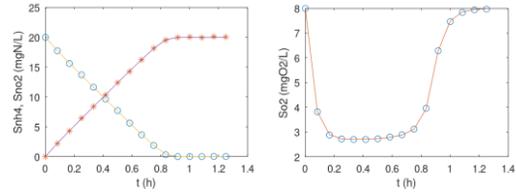
```

36

Una vez hallado el set de parámetros óptimos podemos graficar:

```

35 % Gráficas
36 t = z(:,1)/60/24; % d (el original está en minutos)
37 x = lsode(@(x,t) der_f(x,t,p_opt,p1),x0,t);
38
39 t_e = z(:,1)/60; % h
40 Snh_exp = z(:,2); % Snh4 en mgN/L
41 Sno2_exp = z(:,4); % Sno2 en mgN/L
42 So_exp = z(:,3); % So2 en mgO2/L
43
44 figure(1)
45 plot(t_e, Snh_exp, 'o', t_e, Sno2_exp, '+', t*24, x(:,1), t*24, x(:,3))
46 xlabel('t (h)'); ylabel('Snh4, Sno2 (mgN/L)')
47 figure(3)
48 plot(t_e, So_exp, 'o', t*24, x(:,2))
49 xlabel('t (h)'); ylabel('So2 (mgO2/L)')
    
```



37

38

Otra alternativa es usar optimización cuadrática no lineal, ya que el comando lsqnonlin nos permite obtener más información, además del vector óptimo.

```

51 % también se puede usar lsqnonlin:
52 pkg load optim
53 p_opt2 = lsqnonlin(@ (p) cost2(p,z,p1,x0),p_init,p_inf,p_sup)
54
    p_opt2 =
        0.1348
        1.3475
        0.4632
        0.7030
    
```

La solución al problema de optimización anterior proporciona la *mejor estimación* de los valores de los parámetros. El siguiente paso es evaluar la *calidad* de los estimadores de los parámetros. Para esto se requiere la estimación del *intervalo de confianza* de los valores de los parámetros y la *correlación lineal* por pares entre los parámetros.

$$cov(\hat{\theta}) = s^2(F' \cdot F)^{-1} \quad \text{donde} \quad F = \left. \frac{\partial f(\theta)}{\partial \theta} \right|_{\theta=\hat{\theta}}$$

Donde  $s^2$  es la estimación no sesgada de  $\sigma^2$  obtenida a partir de los residuos de la estimación de parámetros:

$$s^2 = \frac{S_{min}(y, \theta)}{n - p}$$

$n$  = número total de mediciones  
 $p$  = número de parámetros estimados  
 $n-p$  = grados de libertad  
 $S_{min}(y, \theta)$  = valor mínimo de la función objetivo  
 $F$  = matriz Jacobiana, evaluada en  $\theta = \hat{\theta}$ .

39

40

La matriz de covarianza es una matriz cuadrada de  $(p \times p)$  dimensiones. Los elementos diagonales de la matriz son la varianza de los estimadores de los parámetros, mientras que los elementos no diagonales son la covarianza entre cualquier par de estimadores de parámetros.

Los intervalos de confianza (la diferencia entre los estimadores y los verdaderos valores de los parámetros), siguen una distribución t de Student

$$\hat{\theta}_{1-\alpha} = \hat{\theta} \pm t_{n-p}^{\alpha/2} \sqrt{diag\ cov(\hat{\theta})}$$

$$R_{ij} = \frac{cov(\hat{\theta}_i, \hat{\theta}_j)}{\sigma_{\hat{\theta}_i} \sigma_{\hat{\theta}_j}}$$

Esta correlación lineal oscilará entre [-1,1] e indica si el estimador del parámetro es identificable de forma única (en el caso de que el coeficiente de correlación sea bajo) o si está correlacionado (si el coeficiente de correlación es alto).

41

42

Para la estimación de incertidumbres nos sirve lsqnonlin, porque nos devuelve el residuo normalizado y el jacobiano, con los cuales podemos calcular la varianza de los errores y la covarianza de los parámetros:

```

54
55 % estimación de incertidumbres
56 % se usa lsqnonlin con 0 iteraciones:
57 options = optimset('MaxFunEvals',0);
58 [X, RESNORM, RESIDUAL, EXITFLAG, OUTPUT, LAMBDA, JACOBIAN] = ...
59 lsqnonlin(@ (p) cost2(p,z,p1,x0),p_opt,p_inf,p_sup,options);
60 re = RESNORM; j = JACOBIAN;
61 n = length(t_e); m = 3; % número de variables medidas
62 dof = (n - 1)*m;
63 s = re/dof % varianza de los errores
64 pcov = s*inv(j'*j) % covarianza de los parámetros
65 psigma = sqrt(diag(pcov)) % desviación estándar de los parámetros
66 pcor = pcov./(psigma*psigma') % matriz de correlación
67
    
```

```
s = 1.9561e-03
pcov =
    8.6198e-05    8.8238e-04    1.0100e-05    1.1479e-04
    8.8238e-04    9.6576e-03    1.9322e-05    2.9727e-03
    1.0100e-05    1.9322e-05    4.6637e-04    -4.1017e-04
    1.1479e-04    2.9727e-03    -4.1017e-04    5.4117e-03

psigma =
    9.2843e-03
    9.8273e-02
    2.1596e-02
    7.3564e-02

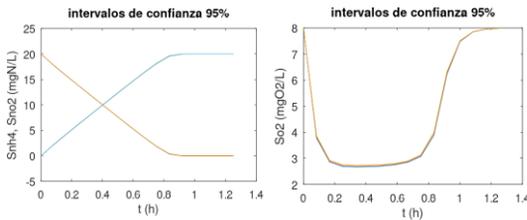
pcor =
    1.0000e+00    9.6710e-01    5.0376e-02    1.6806e-01
    9.6710e-01    1.0000e+00    9.1042e-03    4.1120e-01
    5.0376e-02    9.1042e-03    1.0000e+00    -2.5818e-01
    1.6806e-01    4.1120e-01    -2.5818e-01    1.0000e+00
```

43

También podemos calcular los márgenes de error:

```
68 pkg load statistics
69 alfa = 0.025;
70 tcr = tinvt(1-alfa,dof); % distribución t al valor crítico de alfa
71 p95 = [p_opt2-tcr*psigma p_opt2+tcr*psigma];
72 p95(p95(:)>0) = 0; % sustituye por 0 los elementos no positivos
73
74 ycov = j*pcov*j';
75 ysigma = sqrt(diag(ycov));
76 ys = reshape(ysigma,n,m);
77 xl = x(:,1:3);
78 y95 = [xl-tcr*ys xl+tcr*ys]
79
80 figure(3)
81 plot(t_e,[y95(:,1),xl(:,1),y95(:,4)],t_e,[y95(:,3),xl(:,3),y95(:,6)])
82 xlabel('t (h)');ylabel('SmH4, SmO2 (mg/L)');title('Intervalos de confianza 95%')
83
84 figure(4)
85 plot(t_e,[y95(:,2),xl(:,2),y95(:,5)])
86 xlabel('t (h)');ylabel('So2 (mgO2/L)');title('Intervalos de confianza 95%')
```

44



45

### El método “bootstrap”

- El método de método de estimación de máxima verosimilitud supone que la distribución de los errores sigue una distribución normal (gaussiana). Sin embargo, en muchas aplicaciones prácticas esta condición rara vez se cumple.
- El método bootstrap trabaja con la distribución real de los errores de medición, que se propagan a los errores de estimación de los parámetros utilizando un esquema Monte Carlo.
- Se utiliza el conjunto de datos original D(0) con sus N puntos de datos, para generar cualquier número de conjuntos de datos sintéticos D<sup>s</sup>(1);D<sup>s</sup>(2);..., también con N puntos de datos. El procedimiento consiste simplemente en extraer N puntos de datos con sustituciones a partir del conjunto D(0).

46

Supongamos que en nuestro modelo  $y_i = f_i(\theta) + \varepsilon_i$  donde  $\varepsilon_i \propto F$  donde F es una distribución cualquiera.

Estimamos los parámetros minimizando  $\hat{\theta} : \min_{\theta} \|y - f(\theta)\|^2$

Definimos  $\hat{F}$  como la distribución de probabilidad de muestreo de  $\hat{\varepsilon}$  así:

$$\hat{F} = \frac{1}{n} \text{ en } \varepsilon_i = (y_i - f_i(\theta)) \quad i = 1, 2, \dots, n$$

La muestra bootstrap,  $y^*$ , se genera según

$$y_i^* = f_i(\hat{\theta}) + \varepsilon_i^* \text{ donde } \varepsilon_i^* \propto \hat{F}$$

El error de medición,  $\varepsilon^*$ , se simula mediante un muestreo aleatorio con sustitución a partir de los residuos originales.

47

Repetiendo el procedimiento de muestreo anterior M veces, se generan M conjuntos de datos: D<sup>s</sup>(1), D<sup>s</sup>(2), D<sup>s</sup>(3), ... D<sup>s</sup>(M). Cada conjunto de datos sintéticos, D<sup>s</sup>(j), permite obtener un nuevo estimador del parámetro  $\hat{\theta}(j)$  mediante el mismo método de minimización:

$$\hat{\theta}_j : \min_{\theta} \|D^s(j) - f(\theta)\|^2 \text{ con } j = 1, 2, \dots, M$$

Obtenemos una matriz de estimadores de parámetros  $\hat{\theta}(M \times p)$  que puede representarse como un histograma e interpretarse utilizando parámetros frecuentistas comunes como la media, la desviación típica y el percentil del 95 %.

La matriz de covarianza y correlación puede calcularse utilizando la propia  $\hat{\theta}(M \times p)$ . Esto proporciona toda la información necesaria sobre la calidad de los parámetros.

48

Los primeros pasos son igual que antes; calculamos además los residuos:

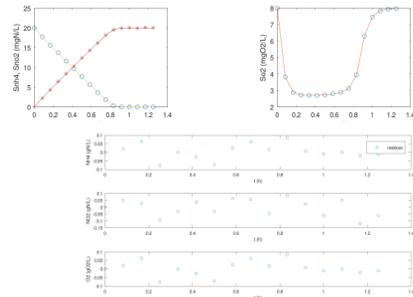
```

52 % residuos
53 resid = x(:,1:3) - z(:,2:4);
54 figure(3)
55 subplot(311)
56 plot(t_e,resid(:,1),'o')
57 xlabel('t (h)');ylabel('NH4 (gN/L)');legend("residuos")
58 subplot(312)
59 plot(t_e,resid(:,3),'o')
60 xlabel('t (h)');ylabel('NO2 (gN/L)')
61 subplot(313)
62 plot(t_e,resid(:,1),'o')
63 xlabel('t (h)');ylabel('O2 (gO2/L)')
64

```

49

50



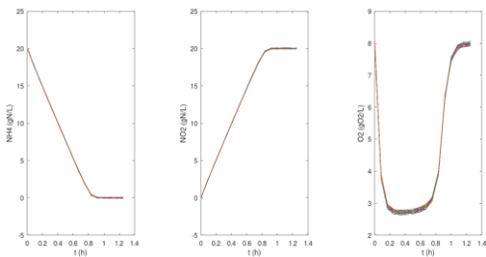
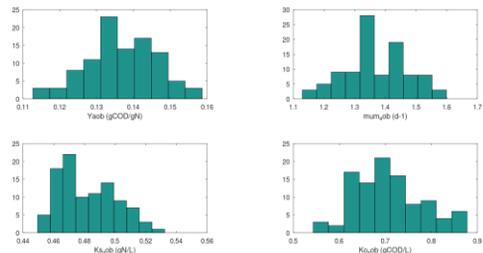
```

65 pkg load optim
66
67 [p_opt, RESNORM, res, EXITFLAG, OUTPUT, LAMBDA, JACOBIAN] = ...
68 lsqnonlin(@p) cost2(p,z,pi,x0),p_opt,p_inf,p_sup);
69 n = length(t_e); m = 3; % número de variables medidas
70 x1 = x;
71 nboot = 100; % número de simulaciones
72 for i = 1:nboot
73     onesam = cell(n*rand(n,m)); % genera una matriz "desordenada" de nxm
74     rcam = res(onesam); % reordena la matriz de residuos según la matriz anter
75     x = x1(:,1:3) + rcam; % genera los pseudo datos
76     pp(i,:) = lsqnonlin(@p) cost2(p,[t*60*24 x],pi,x0),p_opt); % busca los pa
77     bootsam(:,i) = x; % guarda los pseudo datos
78 endfor
79
80 p_mean = mean(pp)
81 p_std = std(pp)
82 p_cov = corr(pp)
83

```

51

52



53

54

