

SUB-ESTIMACIÓN LINEAL A TROZOS APLICADO A UN MODELO INVERSO

José Meza

UNIVERSIDAD CENTROCCIDENTAL "LISANDRO ALVARADO"

Decanato de Ciencias y Tecnología

Departamento de Investigación de Operaciones Y Estadística

Barquisimeto, Mayo de 2010

SUB-ESTIMACIÓN LINEAL A TROZOS APLICADO A UN MODELO INVERSO

Por

José Meza

Trabajo de Ascenso presentado como requisito parcial para optar
a la categoría de Asociado en el escalafón del personal
docente e investigación de la UCLA.

Barquisimeto, Mayo de 2010

TRABAJO DE ASCENSO

**SUB-ESTIMACIÓN LINEAL A TROZOS APLICADO A
UN MODELO INVERSO**

Autor: José Meza

Jurado Calificador

Secretario: Prof. Alí Duin

Principal: Dr. Hugo Lara

Principal: Dr. Javier Hernández Benítez

Barquisimeto, Mayo de 2010

SUB-ESTIMACIÓN LINEAL A TROZOS APLICADO A UN MODELO INVERSO

RESUMEN

Para el diseño de reactores trifásico tipo columna de burbujeo (CBT) es un problema inverso en el cual, es necesario tener la distribución de sólidos dentro del reactor. En este trabajo se considera el problema inverso (CBT) formulado por J. Hernández [6]. Esta distribución viene dada por la solución de una ecuación diferencial ordinaria (EDO) de orden dos con condiciones de frontera desarrollada por Cova [3], Smith y Ruether [12]. Algunos elementos de esta ecuación están dados por correlaciones que dependen de ciertos parámetros que son desconocidos, estos parámetros son determinada mediante un ajuste de parámetros a partir de datos experimentales. La metodología utilizada para determinar dichos parámetros es el método de sub-estimación lineal a trozos desarrollado por Mangasarian, Rosen y Thompson [10].

ÍNDICE GENERAL

ÍNDICE DE TABLAS	III
ÍNDICE DE FIGURAS	IV
INTRODUCCIÓN	1
1. CONSIDERACIÓN DEL MODELO INVERSO	4
1.1. MODELO DE COLUMNAS DE BURBUJEO TRIFÁSICOS	4
1.2. ECUACIONES DEL MODELO	6
1.3. FORMULACIÓN DEL PROBLEMA INVERSO	15

ÍNDICE GENERAL	II
1.4. ANTECEDENTES DEL MODELO	18
2. METODOLOGÍA DE RESOLUCIÓN	20
2.1. RESOLUCIÓN DEL PROBLEMA INVERSO	20
2.2. SUB-ESTIMACIÓN LINEAL A TROZOS	22
2.3. ALGORITMO QUE DETERMINA LA SUB-ESTIMACIÓN LINEAL A TROZOS	36
2.4. ALGORITMO DE INVERSION	42
3. RESULTADOS NUMÉRICOS	44
4. CONCLUSIONES	49
A. CÓDIGOS USADOS	53
BIBLIOGRAFÍA	60

ÍNDICE DE TABLAS

3.1. Datos experimentales utilizados en el ajuste de los parámetros	45
3.2. Aplicado a una distribución de 100 puntos en Λ	47

ÍNDICE DE FIGURAS

1.1. Esquema del reactor	5
1.2. Parte cónica del reactor	11
2.1. Sub-estimador lineal a trozos con $\ell = 3$	23
2.2. Gráfica de $f(x, y)$	40
2.3. Sub-estimación Lineal a Trozos	41

INTRODUCCIÓN

El diseño de reactores trifásicos tipo columna de burbujeo (CBT) es un modelo inverso en el cual se requiere determinar de la concentración de sólidos en función de la posición dentro de un reactor. Esta distribución generalmente se calcula usando un modelo de espacio de sedimentación-dispersión [7].

En estos reactores, el sólido tiende en promedio a sedimentar, pero es impulsado hacia arriba por un líquido, que a su vez es agitado por las burbujas gaseosas. Por lo tanto hay un decrecimiento gradual de la concentración de sólidos desde el fondo hasta el tope del lecho.

Esta concentración de sólidos a lo largo del eje del reactor, se puede determinar mediante la resolución de una ecuación diferencial ordinaria no lineal (EDO) de orden 2 con condiciones de frontera o también por la solución de una ecuación diferencial ordinaria

no lineal (EDO) de orden 1 con condición inicial. Además, el coeficiente de dispersión axial y la velocidad de sedimentación impedida, que forman parte de la EDO, están dados por correlaciones tomadas de la literatura y son dependientes de parámetros a ajustar, los cuales a su vez se expresan en función de datos experimentales.

En este trabajo se considera el problema inverso (CBT) formulado por J. Hernández [6], el cual está representado por la ecuación diferencial ordinaria no lineal de orden 2 descrita por Cova [3], Smith y Ruether [12] la cual está expresada por:

$$\frac{d}{dz} \left[A(z)(1 - \epsilon_g(z))(C_s(z) \left(\frac{\bar{u}_{sl}^*(z)}{1 - \epsilon_g(z)} - \psi_l(z)u_t^*(z) \right) - E_s(z) \frac{d}{dz} C_s(z)) \right] = 0, \quad (0.0.1)$$

con las condiciones de frontera:

$$A(z)(1 - \epsilon_g(z))(C_s(z) \left(\frac{\bar{u}_{sl}^*(z)}{1 - \epsilon_g(z)} - \psi_l(z)u_t^*(z) \right) - E_s(z) \frac{d}{dz} C_s(z))|_{z=0} = M_s \quad (0.0.2)$$

$$C_s(z)|_{z=L} = \gamma \cdot C_s^0 \quad (0.0.3)$$

donde

M_s : caudal másico de sólido,

L : altura total de la columna (m),

C_s^0 : concentración de sólido en la entrada de la columna (alimentación) $\left(\frac{kg}{m^3}\right)$,

γ : parámetro a ajustar.

Además, $C_s(z)$ es la concentración de sólido en el lodo y los coeficientes de la ecuación son funciones que dependen del coeficiente de dispersión axial y la velocidad de sedimentación impedida, que a su vez, éstas dependen implícitamente de los parámetros a ajustar.

En el Capítulo 1, se describe el problema inverso de reactores trifásicos tipo columna de burbujeo (CBT) considerado en este trabajo, formulado por Hernández [6] indicando todas las consideraciones teóricas que permiten modelar el problema inverso como el problema de fronteras antes mencionado el cual depende de correlaciones desconocidas, además se formula el problema que va a permitir ajustar estos parámetros con datos experimentales conocidos. Por tal motivo el objetivo del trabajo es ajustar los parámetros o correlaciones que están presentes en los coeficientes de (0.0.1), usando el método de sub-estimación lineal a trozos. En el Capítulo 2, se da una explicación detallada del método de Sub-Estimación Lineal a Trozos desarrollado por Mangasarian O. L., Rosen J.B., y Thompson [10] empleado y se describe la metodología utilizada para obtener el ajustar de estos parámetros. En el Capítulo 3, se presentan los resultados obtenidos al implementar la metodología usada. En el Capítulo 4, se indican las conclusiones que se obtienen de los resultados y en el apéndice A se muestran los códigos en MATLAB, que se utilizaron para obtener los resultados.

CAPÍTULO 1

CONSIDERACIÓN DEL MODELO INVERSO

1.1. MODELO DE COLUMNAS DE BURBUJEO TRI- FÁSICOS

Los reactores trifásicos se pueden dividir en dos tipos; reactores de lecho fijo en los que las fases líquida y gaseosa se desplazan a través de un lecho fijo de partículas de catalizador, y reactores de lecho móvil en donde el catalizador se mantiene en movimiento agitado por las fases líquida y/o gaseosa o en forma mecánica. Los reactores de lecho móvil donde el catalizador es agitado por el gas y el líquido se denominan reactores *slurry* o de burbujeo.

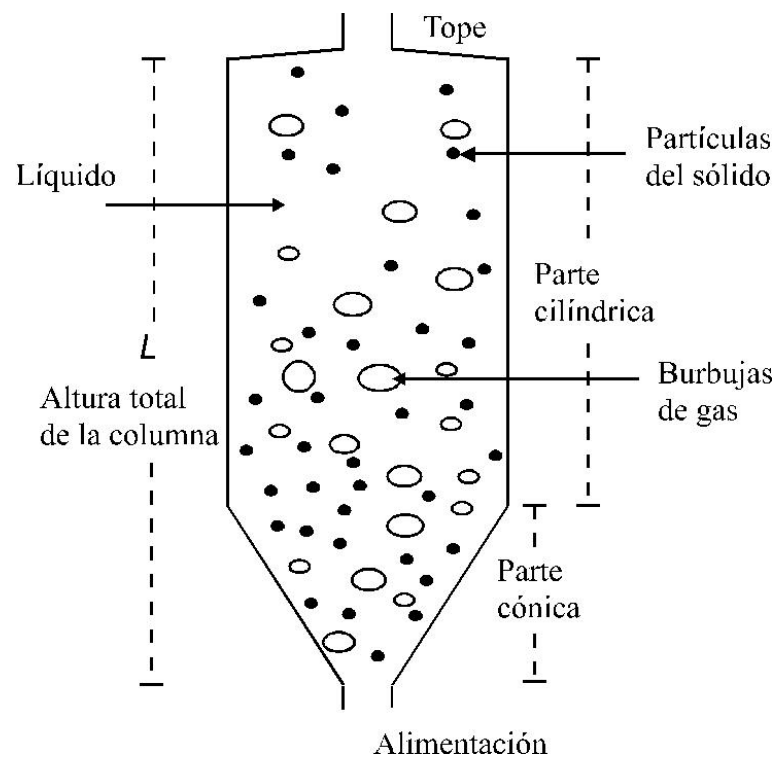


Figura 1.1: Esquema del reactor

Los reactores tipo columna de burbujeo están constituidos por una cámara de reacción de forma cilíndrica en su parte superior y de forma cónica en la parte inferior, un diseño típico para estos reactores se pueden ver en la figura (1.1).

El modelo inverso considerado en este trabajo, consiste en el diseño de reactores trifásicos tipo columna de burbujeo (CBT) el cual requiere del conocimiento de la distribución de sólidos en función de la posición dentro del reactor. Esta distribución generalmente se calcula usando un modelo de sedimentación-dispersión [7].

En estas columnas, el sólido tiende a sedimentar, pero es impulsado hacia arriba por el líquido, que a su vez es agitado por las burbujas gaseosas. Por lo tanto hay un decrecimiento gradual de la concentración de sólidos desde el fondo hacia el tope del reactor. El líquido y el gas operan en forma continua y las partículas sólidas son suspendidas por el movimiento del líquido inducido por el gas.

Para conocer el flujo neto del sólido en el área transversal de la columna en una determinada posición, medida desde la boca de alimentación del reactor (ver fig. 1.1), partimos de la siguiente igualdad

$$\frac{d}{dz} [F_s(z)A_l(z)] = 0 \quad (1.1.1)$$

donde

$F_s(z)$ es el flujo másico de sólido por unidad de área transversal de lodo, $A_l(z)$ es el área transversal de lodo (m^2) y z es la dirección axial en el reactor.

1.2. ECUACIONES DEL MODELO

En 1966, Cova [3] modela el flujo $F_s(z)$ mediante la ecuación: el flujo,

$$F_s(z) = C_s(z) \frac{F_s(z) + F_l(z)}{\rho_l} - C_s(z) u_t(z) - E_s(z) \frac{d}{dz} C_s(z) \quad (1.2.1)$$

donde

$C_s(z)$: concentración de sólido en el lodo, dependiente de la posición z (m) en la columna $\left(\frac{kg}{m^3}\right)$,

$F_l(z)$: flujo másico de líquido por unidad de área transversal de lodo $\left(\frac{kg}{m^2 \cdot s}\right)$,

ρ_l : densidad de lodo $\left(\frac{kg}{m^3}\right)$,

$u_t(z)$: velocidad de sedimentación referida a la sección transversal del lodo $\left(\frac{m}{s}\right)$,

$E_s(z)$: coeficiente de dispersión del sólido referido al lodo.

Se puede definir la velocidad superficial de lodo basada en la sección transversal del lodo como

$$\bar{u}_{lodo}(z) = \frac{F_s(z) + F_l(z)}{\rho_l}.$$

Luego, la ecuación (1.2.1) queda:

$$F_s(z) = C_s(z) \bar{u}_{lodo}(z) - C_s(z) u_t(z) - E_s(z) \frac{d}{dz} C_s(z) \quad (1.2.2)$$

y sustituyendo en (1.1.1),

$$\frac{d}{dz} \left[\left(C_s(z) \bar{u}_{lodo}(z) - C_s(z) u_t(z) - E_s(z) \frac{d}{dz} C_s(z) \right) A_l(z) \right] = 0. \quad (1.2.3)$$

Esta ecuación puede ser reformulada haciendo

$$\bar{u}_{lodo}(z) = \frac{\bar{u}_{sl}^*(z)}{1 - \epsilon_g(z)} \quad (1.2.4)$$

$$A_l(z) = A(z) (1 - \epsilon_g(z)) \quad (1.2.5)$$

$$u_t(z) = \psi_l(z) \cdot u_t^*(z) \quad (1.2.6)$$

donde

$A(z)$: sección transversal de la columna (m^2),

$\epsilon_g(z)$: *hold up* gaseoso,

ϵ_l : *hold up* líquido,

$\bar{u}_{sl}^*(z)$: velocidad superficial de lodo basada en la sección transversal de la columna ($\frac{m}{s}$),

$\psi_l(z)$: fracción volumétrica de líquido en el lodo,

$u_t^*(z)$: velocidad de sedimentación referida a la sección transversal de la columna ($\frac{m}{s}$).

Las ecuaciones (1.2.4-1.2.6) permiten reformular el problema basado en la sección

transversal de la columna. Sustituyendo estas ecuaciones en (1.2.3), se obtiene la ecuación diferencial de orden 2 dada por:

$$\frac{d}{dz} \left[A(z)(1 - \epsilon_g(z)) \left(C_s(z) \left(\frac{\bar{u}_{sl}^*(z)}{1 - \epsilon_g(z)} - \psi_l(z) u_l^*(z) \right) - E_s(z) \frac{d}{dz} C_s(z) \right) \right] = 0, \quad (1.2.7)$$

La cual puede ser resuelta directamente como un problema con las condiciones de frontera

$$A(z)(1 - \epsilon_g(z)) \left(C_s(z) \left(\frac{\bar{u}_{sl}^*(z)}{1 - \epsilon_g(z)} - \psi_l(z) u_l^*(z) \right) - E_s(z) \frac{d}{dz} C_s(z) \right) \Big|_{z=0} = M_s \quad (1.2.8)$$

$$C_s(z) \Big|_{z=L} = \gamma \cdot C_s^0 \quad (1.2.9)$$

También puede resolverse como una ecuación diferencial de orden 1 dada por:

$$A(z)(1 - \epsilon_g(z)) \left(C_s(z) \left(\frac{\bar{u}_{sl}^*(z)}{1 - \epsilon_g(z)} - \psi_l(z) u_l^*(z) \right) - E_s(z) \frac{d}{dz} C_s(z) \right) = M_s \quad (1.2.10)$$

con la condición inicial:

$$C_s(z) \Big|_{z=L} = \gamma \cdot C_s^0 \quad (1.2.11)$$

donde

M_s : caudal másico de sólido,

L : altura total de la columna (m),

C_s^0 : concentración de sólido en la entrada de la columna (alimentación) $\left(\frac{kg}{m^3}\right)$,

γ : parámetro a ajustar.

El parámetro γ a ajustar en la condición de frontera o inicial, representa una relación de la concentración en la alimentación con concentración en el tope de la columna. Esta relación fue usada en [12]. Se hace este ajuste porque no hay forma exacta de conocer la concentración en el tope.

Las soluciones de estos problemas requieren de expresiones que permitan calcular $\psi_l(z)$, $A(z)$, $\epsilon_g(z)$, $u_t^*(z)$ y $E_s(z)$, estas expresiones se describen a continuación.

La fracción volumétrica de líquido en el lodo viene dada por

$$\psi_l(z) = 1 - \frac{C_s^0}{\rho_s} C_s(z) \quad (1.2.12)$$

Al sustituir esta expresión en las expresiones (1.2.7) y (1.2.10) se obtiene que tales ecuaciones diferenciales ordinarias son no lineales.

El valor de $A(z)$ se calcula a partir de la geometría de la columna (fig. 1.2); si la posición z se encuentra en la parte del cono ($0 \leq z \leq h_{cono}$) se tiene que

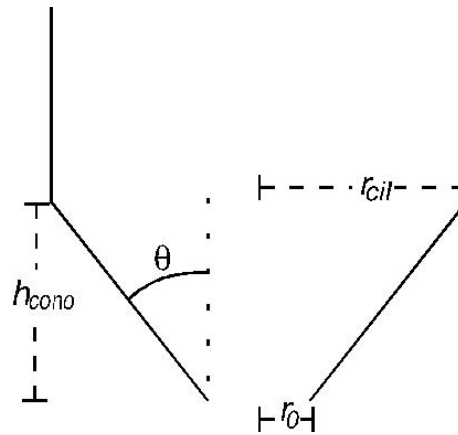


Figura 1.2: Parte cónica del reactor

$$A(z) = A_0 \left(1 + \frac{z \tan \theta}{r_0} \right)^2, \quad (1.2.13)$$

Si z está por encima del cono ($h_{cono} \leq z \leq L$) se tiene

$$A(z) = A_0 \left(\frac{r_{cil}}{r_0} \right)^2, \quad (1.2.14)$$

donde

r_0 : radio de la entrada a la columna (m),

A_0 : área de la sección transversal de la entrada a la columna (m^2),

r_{cil} : radio de la parte cilíndrica de la columna,

θ : ángulo del cono inferior de la columna.

La altura del cono, h_{cono} , es

$$h_{cono} = (r_{cil} - r_0) \cot \theta.$$

El *hold up* gaseoso, ϵ_g , puede ser calculado a través de varias correlaciones publicadas en la literatura. Para este caso se siguió la elección de Smith y Ruether [12] quienes utilizaron la correlación de G. A. Hughmark [5]

$$\epsilon_g(z) = \left[2 + \left(\frac{0,35}{u_g(z)} \right) \left(\frac{\rho_l \cdot \sigma}{72} \right)^{\frac{1}{3}} \right]^{-1}$$

donde

$u_g(z)$: velocidad superficial de gas $\left(\frac{m}{s}\right)$,

σ : tensión superficial $\left(\frac{kg}{s^2}\right)$.

La velocidad superficial de gas puede calcularse a partir del caudal molar de gas, Ω , mediante la fórmula

$$u_g(z) = \frac{\Omega RT}{A(z) [P_a + (L - z) g (\rho_l \epsilon_l + \rho_s \epsilon_s)]}$$

donde

R : constante universal de los gases,

T : temperatura absoluta (K),

P_a : presión atmosférica,

ρ_s : densidad del sólido $\left(\frac{kg}{m^3}\right)$,

g : aceleración de la gravedad $\left(\frac{m}{s^2}\right)$,

ϵ_s : *hold up* sólido.

La velocidad de sedimentación referida a la sección transversal de la columna se puede obtener mediante la siguiente correlación (véase [12, 13])

$$u_t^*(z) = \alpha_1 \cdot u_g(z)^{\alpha_2} \cdot u_{t_\infty}^{\alpha_3} \cdot \psi_l(z)^{\alpha_4} \quad (1.2.15)$$

donde u_{t_∞} es la velocidad de sedimentación de una partícula en medio líquido puro, y α_1 , α_2 , α_3 y α_4 son los parámetros de la correlación.

El coeficiente de dispersión axial, $E_s(z)$, se correlaciona de la forma (véase [13])

$$E_s(z) = \frac{u_g(z)D}{Pe_p(z)}$$

donde

$$Pe_p(z) = \beta_1 \left(\frac{Fr_g(z)^{\beta_2}}{Re_g(z)} \right)^{\beta_3} + \beta_4 Re_p(z)^{\beta_5} \quad (1.2.16)$$

$$Re_g(z) = \frac{u_g(z) D \rho_l}{\mu_l}$$

$$Fr_g(z) = \frac{u_g(z)}{(gD)^{\frac{1}{2}}}$$

$$Re_p(z) = \frac{d_p \rho_l u_{t\infty}}{\mu_l}$$

D : diámetro de la columna (m),

μ_l : viscosidad $\left(\frac{kg}{m.s}\right)$,

Pe_p : número de Peclet,

Fr_g : número de Froude,

Re_g : número de Reynolds del gas,

Re_p : número de Reynolds de la partícula,

β_i : parámetros de la correlación ($i = 1, \dots, 5$).

1.3. FORMULACIÓN DEL PROBLEMA INVERSO

En este trabajo consideraremos el modelo inverso de (CBT) formulado por la ecuación diferencial ordinaria no lineal de orden 2 expresada por (1.2.7) con las condiciones de frontera dadas por (1.2.8) y (1.2.9) desarrollado por J. Hernández [6], los parámetros que intervienen en las expresiones (1.2.9), (1.2.15) y (1.2.16) son desconocidos. Estos se determinan haciendo un ajuste de la solución de la ecuación diferencial (1.2.7) en función de datos experimentales. Estos datos se conforman de una cierta cantidad de experimentos, que dependen de las velocidades $u_g(z)$ y $\bar{u}_{sl}^*(z)$ y la concentración en la alimentación del reactor C_s^0 , y consisten en mediciones de la concentración de sólidos a diferentes alturas del reactor (ver tabla 3.1).

El ajuste del modelo (CBT) a los datos experimentales permitirá tener la correlación final para la distribución de sólidos en las columnas de burbujeo que constituyen los reactores del proceso *HDH*¹. Esto ayudará al mejoramiento de las herramientas del diseño de un reactor en dicho proceso. Los resultados de este trabajo se usarán para predecir el comportamiento en un reactor de tamaño comercial.

Obsérvese que la ecuación diferencial de orden 2 (1.2.7) puede ser vista de la siguiente manera:

¹*HDH* es un proceso catalítico para la conversión de los residuos, provenientes de las destiladoras atmosféricas y de vacío, en productos de mayor valor comercial

$$G_0(z, C_s, C'_s)C_s + G_1(z, C_s) + G_2(z)C_s'' = 0$$

donde

$$G_0(z, C_s, C'_s) = \frac{d}{dz} \left(A_l(z) \left(\frac{\bar{u}_{sl}^*(z)}{1 - \epsilon_g(z)} - \alpha_1 \cdot u_g(z)^{\alpha_2} \cdot u_{t\infty}^{\alpha_3} \cdot \left(1 - \frac{C_s^0}{\rho_s} C_s(z) \right)^{\alpha_4+1} \right) \right),$$

$$G_1(z, C_s) = A_l(z) \left(\frac{\bar{u}_{sl}^*(z)}{1 - \epsilon_g(z)} - \alpha_1 \cdot u_g(z)^{\alpha_2} \cdot u_{t\infty}^{\alpha_3} \cdot \left(1 - \frac{C_s^0}{\rho_s} C_s(z) \right)^{\alpha_4+1} \right) - \frac{d}{dz} (A_l(z) E_s(z))$$

con

$$E_s(z) = \frac{u_g(z)D}{\beta_1 \left(\frac{Fr_g(z)\beta_2}{Re_g(z)} \right)^{\beta_3} + \beta_4 Re_p(z)\beta_5}, \quad A_l(z) = A(z) (1 - \epsilon_g(z))$$

y

$$G_2(z) = -A_l(z)E_s(z)$$

Denotemos al número de experimentos por n_{exp} y al número de muestras o mediciones que se realizan en el experimento j por m_j para $j = 1, \dots, n_{\text{exp}}$.

Sean $\tilde{F}_j = (\tilde{f}_{1j}, \dots, \tilde{f}_{m_j j})$ el vector de mediciones realizadas para el experimento j para $j = 1, \dots, n_{\text{exp}}$, $x \in \mathbb{R}^{10}$ es el vector de parámetros del modelo (CBT) dado por

$$x = (\alpha_1, \dots, \alpha_4, \beta_1, \dots, \beta_5, \gamma)^t = (x_1, \dots, x_{10})^t.$$

y $F_j(x) = (f_{1j}(x), \dots, f_{m_j j}(x))$ representa las concentraciones de sólidos obtenidas en cada posición del reactor donde se realizó el experimento j .

El propósito es hacer que la distancia entre \tilde{F}_j y $F_j(x)$ sea lo más pequeña posible para cada $j = 1, \dots, n_{\text{exp}}$. Es por ello que consideramos la función

$$f(x) = \frac{1}{n_{\text{exp}}} \sum_{j=1}^{n_{\text{exp}}} \|F_j(x) - \tilde{F}_j\|_2 \quad (1.3.1)$$

que representa el promedio de las distancias entre los datos experimentales y los valores obtenidos al modelar la concentración.

Por tal motivo, el objetivo es resolver el problema dado por:

$$\min_{x \in \Lambda} f(x) \quad \text{con} \quad \Lambda = \prod_{i=1}^{10} [a_i, b_i]. \quad (1.3.2)$$

Obsérvese que la función objetivo $f(x)$ no se conoce explícitamente aunque se tenga la expresión (1.3.1), ya que para la evaluación de la función $f(x)$ se requiere resolver n_{exp} veces, la ecuación diferencial ordinaria no lineal de orden 2 (1.2.7) con las condiciones de frontera dadas por (1.2.8) y (1.2.9), lo cual el cálculo numérico y computacional es costoso. Por otro lado, es posible que se encuentre más de una solución $x \in \Lambda$ que satisfaga la condición (1.3.2). También podríamos estar en presencia de puntos donde $f'(x) = 0$, pudiendo haber un mínimo local o un punto silla de la función $f(x)$ en dicho punto.

Resumiendo lo anterior, estaríamos en presencia de un problema de optimización global donde la función objetivo no se conoce en forma explícita, y el cálculo de su valor en un punto puede ser costoso en términos de tiempo y de recursos de computación.

1.4. ANTECEDENTES DEL MODELO

Smith & Ruether [12] realizaron un estudio de las columnas de burbujeo trifásicas (CBT), bajo el modelo de sedimentación-dispersión; ellos presentaron un modelaje del comportamiento de la concentración en un reactor sin considerar la parte cónica del reactor y basándose en una ecuación de orden 1. Ellos dan correlaciones que permiten calcular el coeficiente de dispersión axial y la velocidad de sedimentación, que son coeficientes de la EDO.

Jean, Tang & Fan [7] consideraron tres modos de operación para las CBT tomando en cuenta el sólido y el líquido que influyen en el reactor y llegaron a un modelo parecido a (1.2.9).

G. Dassori & S. Buitrago [4] consideraron la una ecuación de orden 1 en con una condición inicial y para ésto tomaron en cuenta la concentración en el tope de la columna del reactor. Sin embargo, ésta no es fácil de determinar con exactitud. También resolvieron la ecuación de orden 2, pero no obtuvieron una buena aproximación de los datos experimentales.

J. Hernández en su trabajo [6] resolvió el modelo inverso de (CBT) formulado por la ecuación diferencial ordinaria no lineal de orden 2 expresada por (1.2.7) con las condiciones de frontera dadas por (1.2.8) y (1.2.9) utilizando el paquete COLSYS [2], el cual resuelve ecuaciones diferenciales de orden menor o igual a cuatro, sujetas a un sistema no lineal de ecuaciones (condiciones) de frontera, usando un método de colocación en puntos Gaussianos [1].

CAPÍTULO 2

METODOLOGÍA DE RESOLUCIÓN

2.1. RESOLUCIÓN DEL PROBLEMA INVERSO

El problema que se quiere resolver esta dado por:

$$\min_{x \in \Lambda} f(x) \quad \text{con} \quad \Lambda = \prod_{i=1}^{10} [a_i, b_i]. \quad (2.1.1)$$

donde

$$f(x) = \frac{1}{n_{\text{exp}}} \sum_{j=1}^{n_{\text{exp}}} \|F_j(x) - \tilde{F}_j\|_2 \quad (2.1.2)$$

$\tilde{F}_j = (\tilde{f}_{1j}, \dots, \tilde{f}_{m_jj})$ es el vector de mediciones realizadas para el experimento j y $F_j(x) = (f_{1j}(x), \dots, f_{m_jj}(x))$, representa las concentraciones de sólidos, obtenidas del modelo inverso de (CBT) descrito en la Sección 1.3, en cada posición del reactor donde

se realizó el experimento j para $j = 1, \dots, n_{\text{exp}}$.

Notase que para $j = 1, \dots, n_{\text{exp}}$, $F_j(x) = (f_{1j}(x), \dots, f_{m_jj}(x))$, es la solución del modelo inverso de (CBT) formulado por la ecuación diferencial ordinaria no lineal de orden 2 expresada por (1.2.7) con las condiciones de frontera dadas por (1.2.8) y (1.2.9) en los puntos z_1, \dots, z_{m_j} a lo largo del reactor.

Hay que resaltar que el aporte del trabajo no se encuentra en la formulación del modelo (CBT), debido a que como se ha señalado anteriormente estamos usando el desarrollo por J. Hernández en su trabajo [6] en el cual resolvió el problema (CBT), evaluando directamente la función (2.1.2) lo cual es muy costoso computacionalmente, debido a que se debe estar resolviendo el problema de frontera, n_{exp} veces para poder obtener el valor de la función.

En este trabajo se utiliza la misma formulación del modelo (CBT) presentada por J. Hernández, pero utilizando la teoría de sub-estimación lineal a trozos para resolverlo. En las secciones siguientes, se describe la teoría, se presenta el algoritmo que permite encontrar la sub-estimación y el algoritmo que resuelve el problema inverso (CBT).

2.2. SUB-ESTIMACIÓN LINEAL A TROZOS

El problema que estamos interesados en resolver es encontrar el mínimo global de una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$, dados m valores de la función f

$$f(x^k) = y^k, k = 1, \dots, m.$$

Definición 2.2.1 (Sub-estimador lineal a trozos) [10] *Una función lineal a trozos viene dada por:*

$$p(x) = \alpha + c^t x + \|Ax + b\|_1, \quad (2.2.1)$$

donde $\alpha \in \mathbb{R}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{\ell \times n}$ y $b \in \mathbb{R}^\ell$, donde ℓ es el número de funciones lineales que generan esta sub-estimación lineal a trozos.

p debe cumplir:

$$p(x^k) \leq y^k$$

Así el problema de encontrar un sub-estimador de f está dado por el siguiente problema de minimización

$$\begin{aligned} \min_{\alpha, c, A, b} \sum_{k=1}^m (y^k - (\alpha + c^t x^k + \|Ax^k + b\|_1)) \\ \text{s.a.} \quad \alpha + c^t x^k + \|Ax^k + b\|_1 \leq y^k, \quad k = 1, \dots, m. \end{aligned} \quad (2.2.2)$$

O. L. Mangasarian y R. R. Meyer (1979) [9] probaron que el problema (2.2.2) tiene los mismos puntos Karush-Kuhn-Tucker que el problema

$$\begin{aligned} \min_{\alpha, c, A, b, s^k} \sum_{k=1}^m (y^k - (\alpha + c^t x^k + e^t |Ax^k + b|) + \epsilon e^t s^k) \\ \text{s.a.} \quad \alpha + c^t x^k + e^t s^k \leq y^k, \quad k = 1, \dots, m \\ -s^k \leq Ax^k + b \leq s^k, \quad k = 1, \dots, m, \end{aligned} \quad (2.2.3)$$

donde $\epsilon \in (0, \bar{\epsilon}]$ para algún $\bar{\epsilon} > 0$ y $s^k \in \mathbb{R}^\ell$, $k = 1, \dots, m$, y $e \in \mathbb{R}^\ell$ es un vector de unos.

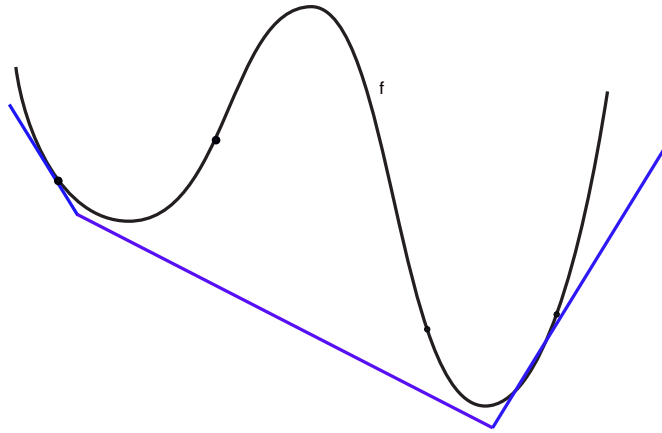


Figura 2.1: Sub-estimador lineal a trozos con $\ell = 3$

Lema 2.2.2 [11] *Sea f una función convexa y sea C un conjunto poliédrico convexo no vacío contenido en el dominio de f . Supongamos que C no tiene líneas que van al infinito y que f es acotada sobre C entonces el supremo de f en C es alcanzado en uno de los extremos de C .*

Proposición 2.2.3 (Solución de la existencia de un vértice) [10] *Supongamos que la región factible del problema de programación lineal (2.2.3) no tiene líneas rectas que van al infinito en ambas direcciones, entonces existe una solución vértice al problema.*

Demostración:

Sabemos que la función objetivo de (2.2.3) es cóncava y acotada en la región factible, en efecto, del conjunto tenemos

$$-s^k \leq Ax^k + b \leq s^k$$

Por propiedad de valor absoluto y multiplicando por e^t se obtiene,

$$e^t |Ax^k + b| \leq e^t s^k, \quad s^k \geq 0 \quad (2.2.4)$$

Luego sustituyendo (2.2.4) en la región factible vemos que esta acotada inferiormente

$$y^k - (\alpha + c^t x^k + e^t |Ax^k + b|) + \epsilon e^t s^k \geq y^k - (\alpha + c^t x^k + e^t s^k) + \epsilon e^t s^k$$

Y esto es equivalente a,

$$y^k - (\alpha + c^t x^k + e^t |Ax^k + b|) + \epsilon e^t s^k \geq y^k - \alpha - c^t x^k - (1 - \epsilon) e^t s^k \quad (2.2.5)$$

Ahora veamos que la región factible de (2.2.3) es un conjunto poliédrico no vacío, para esto tomemos, $c = 0$, $A = 0$, $b = 0$, $s^k = 0$ para $k = 1, \dots, m$ y $\alpha = \min_{k=1, \dots, m} y^k$.

tenemos que

$$\alpha + c^t x^k + e^t s^k \leq y^k, \quad k = 1, \dots, m$$

Luego, sustituimos valores,

$$\alpha + 0x^k + e^t 0 \leq y^k$$

Así,

$$\alpha = \min_{k=1, \dots, m} y^k \leq y^k$$

Y de

$$s^k \leq Ax^k + b \leq s^k$$

Sustituyendo valores,

$$0 \leq 0x^k + 0 \leq 0$$

Por tanto la región factible de (2.2.3) es un conjunto no vacío.

Por otro lado, el problema matemático (2.2.3) es equivalente a

$$\begin{aligned} & \max_{\alpha, c, A, b, s^k} - \sum_{k=1}^m (y^k - (\alpha + c^t x^k + e^t |Ax^k + b|) + \epsilon e^t s^k) \\ & \text{s.a.} \quad \alpha + c^t x^k + e^t s^k \leq y^k, \quad k = 1, \dots, m \\ & \quad -s^k \leq Ax^k + b \leq s^k, \quad k = 1, \dots, m \end{aligned} \tag{2.2.6}$$

la cual cumple que la función objetivo de (2.2.6) es convexa y es acotada superiormente ya que (2.2.3) esta acotada inferiormente y la región factible es un conjunto poliédrico convexo. Ahora aplicando el Lema (1) tenemos que existe una solución vértice del problema minimización (2.2.3).

■

Algoritmo de linealización sucesiva (SLA).

Denotemos $\theta(z) = \sum_{k=1}^m \left(y^k - (\alpha + c^t x^k + e^t |Ax^k + b|) + \epsilon e^t s^k \right)$ que es la función objetivo del problema (2.2.3) y su región factible por

$$Z = \left\{ z \in \mathbb{R}^{1+n+\ell+ln+\ell+lm} \left| \begin{array}{l} \alpha + c^t x^k + e^t s^k \leq y^k, \quad k = 1, \dots, m \\ -s^k \leq Ax^k + b \leq s^k, \quad k = 1, \dots, m, \end{array} \right. \right\}$$

entonces (2.2.3) puede ser escrita como

$$\min_{z \in Z} \theta(z). \quad (2.2.7)$$

Para el estudio del algoritmo SLA debemos evaluar el supergradiente de la función objetivo (2.2.3). Primero representaremos a las variables de optimización de (2.2.3) por $z \in \mathbb{R}^{1+n+\ell+ln+\ell+lm}$,

$$z = \begin{bmatrix} \alpha \\ c \\ A_i^t \\ b \\ s^k \end{bmatrix} \quad (2.2.8)$$

donde $\alpha \in \mathbb{R}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{\ell \times n}$, $A_i = (a_{i1} \ a_{i2} \ \dots \ a_{in})$ y $b \in \mathbb{R}^\ell$, donde ℓ es el número de funciones lineales que generan la sub-estimación lineal a trozos.

Un supergradiente de $\theta(z)$ con respecto a z es dado por,

$$\partial\theta(z) = \sum_{k=1}^m - \begin{bmatrix} 1 \\ x^k \\ \partial|A_i x^k + b_i| x^k, \quad i = 1, 2, \dots, \ell \\ \partial|A_i x^k + b_i|, \quad i = 1, 2, \dots, \ell \\ -\frac{1}{m} \epsilon e \end{bmatrix} \quad (2.2.9)$$

donde $e \in \mathbb{R}^{\ell m}$, para $i = 1, 2, \dots, \ell$ y $k = 1, 2, \dots, m$:

$$\partial|A_i x^k + b_i| = \begin{cases} 1 & \text{si } A_i x^k + b_i > 0 \\ \in [-1, 1] & \text{si } A_i x^k + b_i = 0 \\ -1 & \text{si } A_i x^k + b_i < 0 \end{cases} \quad (2.2.10)$$

Algoritmo 1 *Algoritmo de linealización sucesiva (SLA).* [10]

Comienza con un $z^0 \in \mathbb{R}^{1+n+\ell n+\ell+ \ell m}$ al azar. Si tenemos z^i , determinar z^{i+1} como una solución vértice del programa lineal:

$$\min_{z \in Z} \partial\theta(z^i)(z - z^i) \quad (2.2.11)$$

es decir: $z^{i+1} \in \arg \min_{z \in Z} \partial\theta(z^i)(z - z^i)$.

Parar cuando $\partial\theta(z^i)(z^{i+1} - z^i) = 0$.

Demostraremos que este algoritmo termina en un número finito de pasos de la siguiente manera.

Lema 2.2.4 (Condición necesaria de optimalidad de principio mínimo) [8]

Sea \bar{z} una solución local de $\min_{z \in Z} \theta(z)$ donde Z es un conjunto convexo en \mathbb{R}^n y θ es una función cóncava en \mathbb{R}^n . Entonces \bar{z} satisface el siguiente principio mínimo

$$\partial\theta(\bar{z})(z - \bar{z}) \geq 0, \quad \forall z \in Z. \quad (2.2.12)$$

Demostración:

Sea \bar{z} un mínimo local, por definición tenemos que,

$$\theta(\bar{z}) \leq \theta(y), \quad \forall y \in B(\bar{z}) \cap Z, \quad (2.2.13)$$

donde $B(\bar{z})$ es una vecindad de \bar{z} . Por otro lado tenemos por hipótesis que Z es convexa, entonces para algún $z \in Z$ que no pertenece a $B(\bar{z})$, se deduce que, existe $\lambda \in (0, 1)$ tal que

$$(1 - \lambda)\bar{z} + \lambda z \in B(\bar{z}),$$

Por lo tanto

$$\theta(\bar{z}) \leq \theta((1 - \lambda)\bar{z} + \lambda z) \tag{2.2.14}$$

En consecuencia

$$0 \leq \theta((1 - \lambda)\bar{z} + \lambda z) - \theta(\bar{z})$$

Luego, por definición, si $\partial\theta(\bar{z})$ es un supergradiente de θ en \bar{z} se tiene

$$0 \leq \theta((1 - \lambda)\bar{z} + \lambda z) - \theta(\bar{z}) \leq \partial\theta(\bar{z})((1 - \lambda)\bar{z} + \lambda z) - \bar{z})$$

Así nos queda que

$$0 \leq \theta((1 - \lambda)\bar{z} + \lambda z) - \theta(\bar{z}) \leq \partial\theta(\bar{z})(\lambda z - \lambda\bar{z})$$

Entonces

$$0 \leq \lambda\partial\theta(\bar{z})(z - \bar{z})$$

Y como $\lambda > 0$, se concluye que,

$$0 \leq \partial\theta(\bar{z})(z - \bar{z})$$

■

Teorema 2.2.5 (Terminación finita SLA) [8] Sea $\theta(z)$ una función cóncava en \mathbb{R}^n , acotada inferiormente en Z . El SLA genera una sucesión finita de iterados $\{z^1, z^2, \dots, z^{\bar{i}}\}$ estrictamente decreciente de los valores de la función objetivo: $\{\theta(z^1) > \theta(z^2) > \dots > \theta(z^{\bar{i}})\}$ tal que $z^{\bar{i}}$ satisface la condición necesaria de optimalidad de principio mínimo:

$$\partial\theta(z^{\bar{i}})(z - z^{\bar{i}}) \geq 0, \quad \forall z \in Z \quad (2.2.15)$$

Demostración:

Por hipótesis tenemos que θ es acotada inferiormente en Z , es decir,

$$-\infty < \inf_{z \in Z} \theta(z) \leq \theta(z)$$

sumando $-\theta(z^i)$ y aplicando definición de supergradiente obtenemos lo que sigue,

$$-\infty < \inf_{z \in Z} \theta(z) - \theta(z^i) \leq \theta(z) - \theta(z^i) \leq \partial\theta(z^i)(z - z^i), \quad \forall z \in Z \quad (2.2.16)$$

Esto se cumple para $\forall z^i \in \mathbb{R}^n$, incluso para algún $z^i \notin Z$, que $\partial\theta(z^i)(z - z^i)$ es acotado inferiormente en Z , esto es por (2.2.16). Por lo tanto el programa lineal (2.2.15) tiene una solución y es un vértice, digamos z^{i+1} . Esto se debe a que Z es un conjunto poliédrico convexo (Politopo). Se sigue para $i = 1, 2, \dots$, que

$$\forall z \in Z : \partial\theta(z^i)(z - z^i) \geq \min_{z \in Z} \partial\theta(z^i)(z - z^i) = \partial\theta(z^i)(z^{i+1} - z^i) \quad (2.2.17)$$

Veamos que para $z^i \in \mathbb{R}^n$ para $i = 1, 2, \dots$, tenemos que $\partial\theta(z^i)(z^{i+1} - z^i) \leq 0$, en efecto, si definimos $g(z) = \partial\theta(z^i)(z - z^i)$ y tenemos que

$$g(z^{i+1}) = \min_{z \in Z} g(z)$$

Luego,

$$g(z^{i+1}) \leq g(z), \forall z \in Z$$

Entonces se tiene,

$$g(z^{i+1}) \leq g(z^i) = \partial\theta(z^i)(z^i - z^i) = 0, \quad z^i \in Z$$

Por tanto $g(z^{i+1}) = \partial\theta(z^i)(z^{i+1} - z^i) \leq 0$. Ahora se nos presentan dos casos.

Caso (a):

$\partial\theta(z^i)(z^{i+1} - z^i) < 0$, el algoritmo dado no se para en la iteración i , por la definición de supergradiente y la desigualdad estricta del caso (a) se tiene

$$\theta(z^{i+1}) - \theta(z^i) \leq \partial\theta(z^i)(z^{i+1} - z^i) < 0$$

Así,

$$\theta(z^{i+1}) \leq \theta(z^i) + \partial\theta(z^i)(z^{i+1} - z^i) < \theta(z^i)$$

Por lo tanto,

$$\theta(z^{i+1}) < \theta(z^i), \quad \text{para } i = 1, 2, \dots,$$

Caso (b):

$\partial\theta(z^i)(z^{i+1} - z^i) = 0$, por el algoritmo tenemos que z^i es solución de $\min_{z \in Z} \theta(z)$ y Z es un conjunto convexo, aplicando el Lema (2) tenemos que

$$\forall z \in Z : \partial\theta(z^i)(z - z^i) \geq 0 \tag{2.2.18}$$

Debido que $z^i \in Z$ establecemos $\bar{i} = i$. El punto $z^{\bar{i}}$ por lo tanto cumple con la condición necesaria de optimalidad de principio mínimo con $z^{\bar{i}} = z^i$ y z^i posiblemente puede ser solu-

ción global. Puesto que $\{\theta(z^i)\}_1^{\bar{i}}$ es estrictamente decreciente, θ esta acotada inferiormente en Z y Z es un conjunto poliédrico convexo, entonces la sucesión de los z^i es finita (ya que los z^i son vértices de Z).

■

Corolario 2.2.6 (Terminación finita de SLA) [10] *Supongamos que la región factible Z del programa lineal a trozos (2.2.3) no tiene líneas rectas que van al infinito en ambas direcciones, la SLA genera una sucesión finita de vertices factibles $\{z^1, z^2, \dots, z^{\bar{i}}\}$ de Z estrictamente decrecientes de los valores de la función objetivo: $\{\theta(z^1), \theta(z^2), \dots, \theta(z^{\bar{i}})\}$, tal que $\theta(z^{\bar{i}})$ satisface la condición necesaria de optimalidad de principio mínimo:*

$$\partial\theta(z^{\bar{i}})(z - z^{\bar{i}}) \geq 0, \quad \forall z \in Z \quad (2.2.19)$$

Demostración:

Sabemos que $\theta(z)$ es cóncava y acotada inferiormente, luego aplicando el teorema (1) tenemos que la SLA genera una sucesión finita de vertices factibles $\{z^1, z^2, \dots, z^{\bar{i}}\}$ de Z estrictamente decrecientes de los valores de la función objetivo: $\{\theta(z^1), \theta(z^2), \dots, \theta(z^{\bar{i}})\}$, tal que $\theta(z^{\bar{i}})$ satisface el mínimo principio de la condición necesaria de optimalidad:

$$\partial\theta(z^{\bar{i}})(z - z^{\bar{i}}) \geq 0, \quad \forall z \in Z. \quad (2.2.20)$$

■

Una vez terminado el algoritmo SLA, obtenemos un sub-estimador lineal a trozos de la función original. Dado que nuestra meta es encontrar el mínimo global de la función original, buscamos el mínimo global del sub-estimador para llegar a dicha meta. En otras palabras, una vez obtenido el sub-estimador lineal a trozos debemos resolver el siguiente problema

$$\min_{x \in \mathbb{R}^n} \alpha + c^t x + \|Ax + b\|_1. \quad (2.2.21)$$

El problema (2.2.21) es equivalente a resolver el siguiente problema de programación lineal

$$\begin{aligned} \min_{x,y} \quad & \alpha + c^t x + e^t y \\ \text{s.a.} \quad & -y \leq Ax + b \leq y, \\ & a^1 \leq x \leq a^2, \end{aligned}$$

donde a^1, a^2 son las cotas inferior y superior conocidas del vector x .

Ejemplo 2.2.7 *Ahora nosotros generaremos una síntesis de una función cuadrática a trozos no convexa como sigue:*

$$y(x) = \min_{j \in \{1, \dots, r\}} h_j(x), \quad (2.2.22)$$

donde $h_j(x)$, $j = 1, \dots, r$ son funciones arbitrarias cuadráticas estrictamente convexas, tales como:

$$h_j(x) = \beta^j + d^j x + \frac{1}{2} x^t (0,5I + M^j M^j) x, \quad j = 1, \dots, r, \quad (2.2.23)$$

donde $\beta^j \in \mathbb{R}$, $d^j \in \mathbb{R}^n$ y $M^j \in \mathbb{R}^{n \times n}$ son escogidos.

Algo interesante de la función cuadrática a trozos generada como ya vimos, es que su solución exacta del mínimo global puede ser computado como sigue:

Proposición 2.2.8 (Solución exacta del mínimo global de (2.2.22) y (2.2.23)) [10] Un mínimo global exacto de (2.2.22) y (2.2.23) viene dado por:

$$\min_{j \in \{1, \dots, r\}} \min_{x \in \mathbb{R}^n} h_j(x). \quad (2.2.24)$$

Más específicamente,

$$\min_{x \in \mathbb{R}^n} y(x) = \min_{j \in \{1, \dots, r\}} h_j(x^j), \quad (2.2.25)$$

donde:

$$x^j = (-0,5I + M^j M^j)^{-1} d^j, \quad j \in \{1, \dots, r\}. \quad (2.2.26)$$

Demostración:

Dado que h_j , $j = 1, \dots, r$ son funciones cuadráticas estrictamente convexas, podemos asegurar la existencia de mínimos de las funciones h_j , $j = 1, \dots, r$. Denotemos por \bar{x}^j el

mínimo global de h_j , $j = 1, 2, \dots, r$. Entonces,

$$\min_{x \in \mathbb{R}^n} y(x) = \min_{x \in \mathbb{R}^n} \min_{j \in \{1, \dots, r\}} h_j(x)$$

Aplicando el mín $h_j(x)$
 $j \in \{1, \dots, r\}$

$$\min_{x \in \mathbb{R}^n} \min_{j \in \{1, \dots, r\}} h_j(x) = \min_{x \in \mathbb{R}^n} \{h_1(x), h_2(x), \dots, h_r(x)\}$$

Tenemos entonces

$$\min_{x \in \mathbb{R}^n} \min_{j \in \{1, \dots, r\}} h_j(x) = h_{\bar{j}}(\bar{x}^{\bar{j}})$$

Y esto es equivalente a

$$\min_{x \in \mathbb{R}^n} \min_{j \in \{1, \dots, r\}} h_j(x) = \min_{x \in \mathbb{R}^n} \{h_1(\bar{x}^1), h_2(\bar{x}^2), \dots, h_r(\bar{x}^r)\}$$

Así

$$\min_{x \in \mathbb{R}^n} \min_{j \in \{1, \dots, r\}} h_j(x) = \min_{x \in \mathbb{R}^n} \{ \min_{x \in \mathbb{R}^n} h_1(x), \min_{x \in \mathbb{R}^n} h_2(x), \dots, \min_{x \in \mathbb{R}^n} h_r(x) \}$$

En consecuencia se tiene que,

$$\min_{x \in \mathbb{R}^n} \min_{j \in \{1, \dots, r\}} h_j(x) = \min_{j \in \{1, \dots, r\}} \min_{x \in \mathbb{R}^n} h_j(x)$$

donde $\bar{j} \in \{1, 2, \dots, r\}$, es tal que $\bar{j} = \operatorname{argmin}\{h_1(\bar{x}^1), h_2(\bar{x}^2), \dots, h_r(\bar{x}^r)\}$, es decir

$h_{\bar{j}}(\bar{x}^{\bar{j}}) = \min_{j \in \{1, \dots, r\}} h_j(\bar{x}^j)$. Por lo tanto

$$\min_{x \in \mathbb{R}^n} \min_{j \in \{1, \dots, r\}} h_j(x) = \min_{j \in \{1, \dots, r\}} \min_{x \in \mathbb{R}^n} h_j(x) \quad (2.2.27)$$

Veamos donde $h_j(x)$ toma su valor mínimo, para esto veamos para que puntos $x^{\bar{j}}$ la derivada se anula o no existe.

Derivando $h_j(x^{\bar{j}})$ tenemos,

$$h_j(x^{\bar{j}})' = d^j + x^{\bar{j}}(0,5I + M^j M^j), \quad , j = 1, \dots, r, \quad (2.2.28)$$

Notamos que el único punto que anula a (2.2.28) es:

$$x^{\bar{j}} = -(0,5I + M^j M^j)^{-1} d^j, \quad , j = 1, \dots, r, \quad (2.2.29)$$

■

2.3. ALGORITMO QUE DETERMINA LA SUB-ESTIMACIÓN LINEAL A TROZOS

En esta sección se redescrive el algoritmo de linealización sucesiva (SLA) presentado en la sección anterior.

Dados x^1, x^2, \dots, x^m en \mathbb{R}^n y los valores de la función f

$$f(x^k) = y^k, k = 1, \dots, m.$$

Denotemos $f_k(z) = (\partial\theta(z^k))^t z$. Entonces en el Algoritmo (SLA)

$$\partial\theta(z^k)^t(z - z^k) = f_k(z) - f_k(z^k) \quad (2.3.1)$$

Además, $f_k(z)$ esta expresada por:

$$\begin{aligned} f_k(z) = & -m\alpha - \left(\sum_{h=1}^m x^h \right)^t c - \sum_{i=1}^{\ell} \left(\sum_{h=1}^m \alpha |A_i^k x^h + b_i^k| x^h \right)^t A_i^t \\ & - \sum_{i=1}^{\ell} \sum_{h=1}^m \alpha |A_i^k x^h + b_i^k| b_i^k + \epsilon \sum_{h=1}^m \sum_{i=1}^{\ell} s_i^h \end{aligned}$$

obsérvese que para $z \in \mathbb{R}^{1+n+\ell n+\ell+m}$,

$$z = \left(\alpha, \quad c^t, \quad A_1, \quad \dots, \quad A_{\ell}, \quad b^t, \quad s^1, \quad \dots, \quad s^{mt} \right)^t \quad (2.3.2)$$

donde $\alpha \in \mathbb{R}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{\ell \times n}$ siendo $A_i = (a_{i1} \ a_{i2} \ \dots \ a_{in})$ su i -ésima fila, $b, s^i \in \mathbb{R}^{\ell}$ y ℓ es el número de funciones lineales que generan la sub-estimación lineal a trozos.

Notase que $z \in \mathbb{R}^{1+n+\ell n+\ell+\ell m}$ se puede expresar como,

$$z = (z_1, z_2, \dots, z_{1+n+\ell n+\ell+\ell m})^t$$

donde cada componente esta determinada de la siguiente forma:

si $I = 1$, entonces $z(I) = \alpha$.

si $2 \leq I \leq n + 1$, entonces $z(I) = c_{I-1}$

si $I = (in + 1) + j$ para $i = 1, \dots, \ell$ y $j = 1, \dots, n$ entonces $z(I) = a_{ij}$

si $I = 1 + n + \ell n + i$ para $i = 1, \dots, \ell$ entonces $z(I) = b_i$

si $I = (1 + n + \ell n + \ell + (h - 1)\ell) + i$ para $i = 1, \dots, \ell$ y $h = 1, \dots, m$ entonces
 $z(I) = s_i^h$

Por lo indicado anteriormente, la función $f_k(z)$ es lineal y esta se puede expresar como

$$f_k(z) = (p^k)^t z$$

teniendo que las componentes del vector p^k están dadas por:

$$p_I^k = -m \quad \text{si } I = 1$$

$$p_I^k = -\sum_{h=1}^m X_{I-1}^h \quad \text{si } I = 2, \dots, n + 1$$

$$p_I^k = -\sum_{h=1}^m \alpha |A_i^k X^h + b_i^k| X^h \quad \text{si } I = (in + 1) + j \quad \text{para } i = 1, \dots, \ell \text{ y } j = 1, \dots, n$$

$$p_I^k = -\sum_{h=1}^m \alpha |A_i^k x^h + b_i^k| \quad \text{si } I = 1 + n + \ell n + i \quad \text{para } i = 1, \dots, \ell$$

$$p_I^k = \epsilon \quad \text{si } I = (1 + n + \ell n + \ell + (h - 1)\ell) + i \quad \text{para } i = 1, \dots, \ell \text{ y } h = 1, \dots, m$$

Para $i = 1, 2, \dots, \ell$ y $k = 1, 2, \dots, m$ se tiene que:

$$\partial|A_i x^k + b_i| = \begin{cases} 1 & \text{si } A_i x^k + b_i > 0 \\ \in [-1, 1] & \text{si } A_i x^k + b_i = 0 \\ -1 & \text{si } A_i x^k + b_i < 0 \end{cases} \quad (2.3.3)$$

Como en cada iteración del algoritmo (SLA) minimiza en z a la función definida por (2.3.1) la cual es una función afín, por lo tanto es suficiente minimizar en cada iteración a la función $f_k(z)$, por tal motivo el algoritmo (SLA) se puede reescribir de la siguiente manera:

Algoritmo 2 *Algoritmo de linealización sucesiva (SLA).* [10]

Dados un punto azar $z_0 \in \mathbb{R}^{1+n+\ell n+\ell+\ell m}$, $l \in \mathbb{Z}^+$ y $\epsilon > 0$.

Calcular H y t .

Para $k = 1, 2, \dots$

Calcular p^k .

Determinar z_k resolviendo el problema de programación lineal dado por

$$\begin{aligned} \min \quad & f_{k-1}(z) = (p^{k-1})^t z \\ \text{s.a} \quad & Hz \leq t \\ & z \text{ sin restriccin} \end{aligned} \quad (2.3.4)$$

hasta que se cumpla $|f_{k-1}(z^k) - f_{k-1}(z^{k-1})| < tol$

teniendo que la matriz $H \in \mathbb{R}^{(m+2\ell m) \times (1+n+\ell n+\ell+\ell m)}$ donde cada componente esta definida de la siguiente manera:

$$h_{IJ} = 1 \quad \text{si} \quad I = 1, \dots, m \quad \text{y} \quad J = 1$$

$$h_{IJ} = X_{J-1}^I \quad \text{si} \quad I = 1, \dots, m \quad \text{y} \quad J = 2, \dots, n + 1$$

$$h_{IJ} = X_j^h \quad \text{si} \quad I = m + (h - 1)\ell + i \quad \text{y} \quad J = in + 1 + j \quad \text{para} \quad h = 1, \dots, m,$$

$$j = 1, \dots, n \quad \text{y} \quad i = 1, \dots, \ell$$

$$h_{IJ} = -X_j^h \quad \text{si} \quad I = m + (m + h - 1)\ell + i \quad \text{y} \quad J = in + 1 + j \quad \text{para} \quad h = 1, \dots, m,$$

$$j = 1, \dots, n \quad \text{y} \quad i = 1, \dots, \ell$$

$$h_{IJ} = 1 \quad \text{si} \quad I = m + (h - 1)\ell + i \quad \text{y} \quad J = 1 + n + \ell n + i \quad \text{para} \quad h = 1, \dots, m \quad \text{y}$$

$$i = 1, \dots, \ell$$

$$h_{IJ} = -1 \quad \text{si} \quad I = m + (m + h - 1)\ell + i \quad \text{y} \quad J = 1 + n + \ell n + i \quad \text{para} \quad h = 1, \dots, m \quad \text{y}$$

$$i = 1, \dots, \ell$$

$$h_{IJ} = -1 \quad \text{si} \quad I = m + (h - 1)\ell + i \quad \text{y} \quad J = 1 + n + \ell n + \ell + (h - 1)\ell + i \quad \text{para}$$

$$h = 1, \dots, m \quad \text{y} \quad i = 1, \dots, \ell$$

$$h_{IJ} = -1 \quad \text{si} \quad I = m + (m + h - 1)\ell + i \quad \text{y} \quad J = 1 + n + \ell n + \ell + (h - 1)\ell + i \quad \text{para}$$

$$h = 1, \dots, m \quad \text{y} \quad i = 1, \dots, \ell$$

$$h_{IJ} = 0 \quad \text{En cualquier otro caso}$$

y el vector $t \in \mathbb{R}^{(m+2\ell m)}$ cuyas componentes están determinadas por:

$$t_I = \left\{ \begin{array}{ll} y_I & \text{si} \quad I = 1, \dots, m \\ 0 & \text{si} \quad I = (m + 1), \dots, (m + 2\ell m) \end{array} \right\}$$

A continuación se muestra un ejemplo aplicando el algoritmo 2.

Exemplo: Consideremos la función $f(x, y) = x^2 + y^2 + \text{sen}(\pi x) + \text{sen}(\pi y)$ en la region

$\Lambda = [-2, 2] \times [-2, 2]$ cuya gráfica esta dada por:

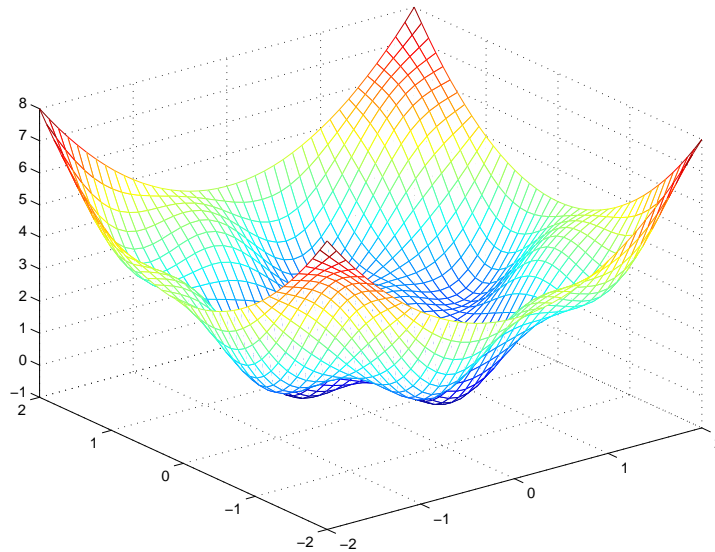


Figura 2.2: Gráfica de $f(x, y)$

Tomando 30 puntos (x_i, y_i) distribuidos uniformemente en $\Lambda = [-2, 2] \times [-2, 2]$ con $z_i = f(x_i, y_i)$, $\epsilon = 10^{-10}$ y $l = 10$ aplicando el algoritmo 2 se obtiene la siguiente sub-estimación lineal a trozos dada por:

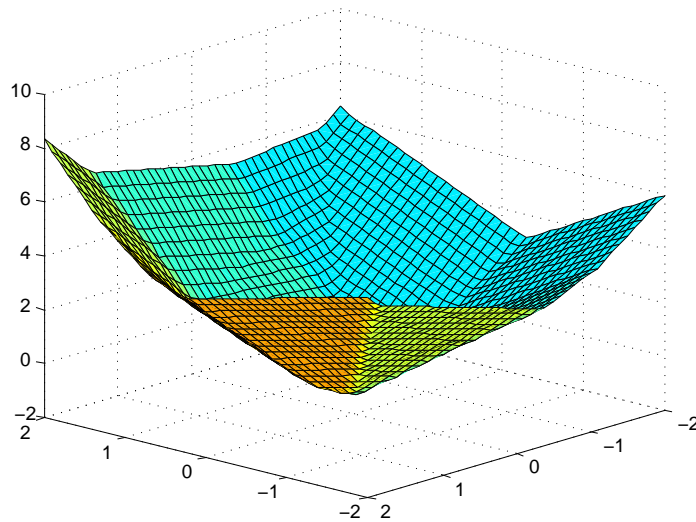


Figura 2.3: Sub-estimación Lineal a Trozos

2.4. ALGORITMO DE INVERSION

En esta sección se describe un esquema general del algoritmo que resuelve el problema inverso (CBT).

Consideremos en $\Lambda = \prod_{i=1}^{10} [a_i, b_i]$ los puntos x^1, x^2, \dots, x^m y el funcional $f : \mathbb{R}^{10} \rightarrow \mathbb{R}$ definido por:

$$f(x) = \frac{1}{n_{\text{exp}}} \sum_{j=1}^{n_{\text{exp}}} \|F_j(x) - \tilde{F}_j\|_2 \quad (2.4.1)$$

donde $\tilde{F}_j = (\tilde{f}_{1j}, \dots, \tilde{f}_{m_jj})$ es el vector de mediciones realizadas para el experimento j , para $j = 1, \dots, n_{\text{exp}}$

Hay que resaltar que para el experimento j la función $F_j(x) = (f_{1j}(x), \dots, f_{m_jj}(x))$ es la solución de la ecuación diferencial ordinaria no lineal de orden 2 (1.2.7) con las condiciones de frontera dadas por (1.2.8) y (1.2.9), fijado el parámetro $x \in \Lambda$. Es decir, para $x \in \Lambda$ se tiene que $F_j(x) = (C_s^x(z_1), \dots, C_s^x(z_{m_j}))$ donde $0 \leq z_1 \leq \dots \leq z_{m_j} \leq L$ son puntos a lo largo del reactor de altura L y C_s^x es la solución del problema del frontera que representa al modelo inverso (CBT) que se obtiene a fijar el parámetro x .

En siguiente bosquejo se presenta el algoritmo que resuelve el problema inverso considerado en este trabajo:

Paso 1: Para $k = 1, \dots, m$ obtener $y^k = f(x^k)$. Esto se obtiene resolviendo m veces el problema de frontera que representa el modelo inverso (CBT) al fijar el parámetro x^k para evaluar la función (2.4.1).

Paso 2: Se determina la sub-estimación lineal a trozos para obtener el vector z aplicando el algoritmo 2 de donde se extraen los valores α , c , A y b que determinan la sub-estimación.

Paso 3: Se resuelve el problema programación lineal

$$\begin{aligned} \text{mín} \quad & \alpha + c^t x + e^t y \\ \text{s.a.} \quad & -y \leq Ax + b \leq y, \\ & x \in \Lambda, y \geq 0 \end{aligned} \tag{2.4.2}$$

donde α , c , A y b se obtienen del Paso 2 y $\Lambda = \prod_{i=1}^{10} [a_i, b_i]$ son las cotas inferior y superior conocidas del vector de parámetros x .

Obsérvese que la función objetivo presente en el Paso 3 es una función afín, por tal motivo es suficiente minimizar la función lineal $f = c^t x + e^t y$.

El punto x que se obtiene en el Paso 3, es la solución óptima del funcional (2.4.1), es decir este va a representar los parámetros que ajustan al problema inverso (CBT).

CAPÍTULO 3

RESULTADOS NUMÉRICOS

Todas las corridas se hicieron en un computador Intel Core 2 Duo de 2.4 GHz con 2 GB de memoria RAM, en el lenguaje MATLAB R2008a para utilizar la librería que este posee. El modelo inverso (CBT) considerado para aplicar la nueva metodología es el mismo modelo que consideró J. Hernández [6] en su trabajo, este esta definido de la siguiente manera: $n_{\text{exp}} = 32$, para $j = 1, \dots, n_{\text{exp}}$ se considera $m_j = 13$, la altura del reactor es de $L = 1$ mts, $\Lambda = [2,0 ; 3,0] \times [0,05 ; 0,25] \times [0,1 ; 0,8] \times [1,1 ; 1,8] \times [5,0 ; 7,0] \times [7,0 ; 9,2] \times [0,05 ; 0,2] \times [0,003 ; 0,005] \times [1,0 ; 1,5] \times [1,5 ; 2,0]$ y los datos experimentales considerados son los mismos utilizados por Hernández estos fueron proporcionados en su momento por la empresa INTEVEP S. A. y fueron realizados en un

caso	u_g m/s	u_{sl} m/s	C_s^0 Kg/m ³	Alturas (m)												
				0.041	0.124	0.207	0.29	0.374	0.455	0.529	0.603	0.677	0.751	0.825	0.899	0.973
1	0.0297	0.0040	48.	470.	446.	373.	300.	217.	171.	137.	129.	104.	103.	88.	79.	70.
2	0.0572	0.0039	23.	178.	125.	119.	97.	71.	54.	63.	75.	52.	53.	50.	43.	41.
3	0.0267	0.0020	74.	731.	741.	711.	630.	557.	504.	478.	418.	360.	325.	292.	207.	190.
4	0.0583	0.0040	139.	666.	640.	673.	596.	527.	491.	443.	462.	446.	435.	398.	360.	341.
5	0.0903	0.0050	148.	572.	606.	562.	499.	498.	419.	391.	368.	342.	324.	331.	291.	225.
6	0.0943	0.0034	198.	674.	663.	630.	588.	527.	465.	466.	461.	443.	431.	394.	395.	371.
7	0.1646	0.0019	111.	685.	650.	539.	462.	470.	409.	396.	381.	347.	328.	312.	267.	275.
8	0.1581	0.0038	215.	618.	635.	597.	551.	549.	450.	427.	402.	402.	392.	380.	333.	309.
9	0.0569	0.0040	68.	317.	258.	216.	193.	171.	146.	140.	132.	126.	124.	116.	112.	106.
10	0.0970	0.0028	62.	425.	350.	255.	240.	193.	160.	164.	156.	147.	130.	125.	124.	109.
11	0.1593	0.0040	19.	165.	109.	103.	78.	70.	66.	50.	48.	57.	46.	43.	36.	34.
12	0.1571	0.0041	36.	195.	143.	130.	142.	137.	96.	87.	92.	92.	72.	77.	73.	70.
13	0.1607	0.0040	61.	387.	332.	258.	228.	248.	154.	146.	144.	149.	131.	129.	121.	115.
14	0.0149	0.0023	35.	633.	652.	582.	461.	354.	253.	157.	111.	85.	77.	58.	59.	52.
15	0.0139	0.0048	36.	498.	624.	618.	518.	410.	353.	207.	106.	78.	59.	59.	50.	49.
16	0.0269	0.0039	46.	544.	547.	502.	386.	295.	261.	220.	183.	165.	142.	107.	97.	87.
17	0.0574	0.0035	48.	544.	474.	394.	348.	287.	261.	236.	210.	217.	182.	164.	170.	135.
18	0.0569	0.0040	108.	552.	507.	454.	371.	332.	296.	264.	250.	237.	222.	195.	188.	186.
19	0.0930	0.0038	80.	409.	427.	373.	347.	279.	225.	222.	208.	200.	188.	172.	167.	161.
20	0.1710	0.0016	70.	482.	385.	313.	296.	266.	222.	204.	202.	190.	178.	170.	164.	155.
21	0.0140	0.0041	65.	651.	812.	900.	911.	800.	758.	755.	745.	734.	734.	703.	700.	661.
22	0.0328	0.0039	143.	834.	874.	853.	807.	743.	696.	687.	675.	679.	646.	633.	596.	533.
23	0.0266	0.0038	248.	812.	880.	897.	846.	824.	787.	759.	736.	742.	697.	688.	667.	634.
24	0.0962	0.0037	509.	813.	833.	801.	790.	736.	687.	702.	651.	659.	627.	643.	624.	592.
25	0.0974	0.0014	326.	825.	851.	831.	804.	782.	700.	720.	687.	638.	646.	629.	597.	631.
26	0.1525	0.0033	290.	786.	807.	772.	713.	697.	643.	633.	612.	612.	607.	575.	543.	530.
27	0.0140	0.0040	12.	395.	303.	189.	148.	87.	70.	62.	57.	46.	41.	41.	33.	35.
28	0.0145	0.0046	37.	391.	410.	349.	251.	167.	152.	116.	95.	79.	71.	62.	42.	49.
29	0.0148	0.0044	37.	447.	472.	418.	304.	227.	194.	147.	118.	98.	77.	61.	58.	59.
30	0.0274	0.0037	18.	203.	136.	116.	88.	71.	47.	48.	45.	40.	36.	32.	38.	33.
31	0.0272	0.0021	23.	262.	180.	118.	113.	57.	55.	53.	45.	47.	49.	42.	42.	41.
32	0.0281	0.0044	61.	439.	373.	334.	263.	188.	164.	143.	125.	124.	109.	106.	88.	81.

TABLA 3.1: Datos experimentales utilizados en el ajuste de los parámetros

reactor experimental en la Universidad Simón Bolívar. estos datos experimentales están descritos en la tabla 3.1.

En la metodología utilizada por J. Hernández [6], obtuvo como resultado que los parámetros x^* que mejor se ajustan al modelo (CBT) considerado son los siguientes:

$$\begin{aligned}
 \alpha_1 &= 2,34 & \beta_1 &= 6,44 & \gamma &= 1,923 \\
 \alpha_2 &= 0,05 & \beta_2 &= 9,061 \\
 \alpha_3 &= 0,751 & \beta_3 &= 0,149 & & (3.0.1) \\
 \alpha_4 &= 1,142 & \beta_4 &= 0,0037 \\
 & & \beta_5 &= 1,145
 \end{aligned}$$

Estos valores son el punto de comparación de los resultados obtenidos con la metodología implementada.

Para obtener los resultados se utilizaron 4 funciones m-file en MATLAB los cuales implementan el Algoritmo 2, estos son:

1. SLA2.m : determina la matriz H y el vector t en (2.3.4).
2. coef2.m : determina los coeficientes p^k de la función objetivo en 2.3.4).
3. subestlineatrazos.m : es la implantación del Algoritmo 2 para obtener la sub-estimación lineal a trozos, donde se utiliza la instrucción linprog.m el cual implementa el método simplex.
4. parametros2.m : resuelve el problema (2.4.2), usando nuevamente la instrucción linprog.m para obtener los parámetros que ajustan al modelo inverso (CBT) considerado.

En el apéndice A, se muestran los códigos de estos archivos m-file, que se implementaron para obtener los resultados.

	$l = 30$ $\epsilon = 10^{-5}$ $tol = 10^{-5}$	$l = 30$ $\epsilon = 10^{-10}$ $tol = 10^{-10}$	$l = 30$ $\epsilon = 10^{-5}$ $tol = 10^{-10}$	$l = 45$ $\epsilon = 10^{-10}$ $tol = 10^{-10}$	$l = 50$ $\epsilon = 10^{-5}$ $tol = 10^{-5}$	$l = 50$ $\epsilon = 10^{-8}$ $tol = 10^{-8}$	$l = 50$ $\epsilon = 10^{-5}$ $tol = 10^{-10}$	$l = 50$ $\epsilon = 10^{-10}$ $tol = 10^{-5}$
α_1	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
α_2	0.05	0.05	0.25	0.05	0.25	0.05	0.25	0.25
α_3	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.793
α_4	1.80	1.80	1.80	1.10	1.10	1.5284	1.2084	1.10
β_1	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00
β_2	9.20	9.20	9.20	9.20	9.2	9.20	9.20	9.20
β_3	0.1967	0.1010	0.1069	0.1950	0.20	0.1413	0.164	0.20
β_4	0.0030	0.0045	0.0050	0.0049	0.0030	0.0030	0.0030	0.0050
β_5	1.00	1.00	1.50	1.50	1.50	1.3396	1.50	1.3871
γ	1.50	1.5094	2.00	2.00	2.00	1.7607	2.00	1.50
$\tilde{f}(x)$	510.46	543.25	558.49	524.67	548.82	505.10	542.67	507.58
<i>tiempo</i> (min)	57.18	144.59	110.45	288.07	116.44	197.98	117.62	218.14
<i>Niter</i>	18	52	25	55	11	29	14	20
$\frac{\ x-x^*\ }{\ x^*\ }$	0.0894	0.0891	0.0882	0.0657	0.0679	0.0699	0.0679	0.0734

TABLA 3.2: Aplicado a una distribución de 100 puntos en Λ

Para la implementación de los programas, se consideró una distribución de 100 puntos en Λ , los resultados obtenidos se presentan en la Tabla 3.2, donde la primera fila se indican los parámetros $x = (\alpha_1, \dots, \alpha_4, \beta_1, \dots, \beta_5, \gamma)$ obtenidos en cada corrida, seguido una aproximación valor del funcional ($\tilde{f}(x)$) en estos valores, el tiempo (*tiempo*) que se tardó en obtener la solución, el número de iteraciones (*Niter*) que se necesitaron para obtener la sub-estimación lineal a trozos y el error relativo de los resultados con res-

pecto a la solución (3.0.1) obtenida por Hernández [6]. El punto inicial para obtener la sub-estimación y los subgradientes se generaron aleatoriamente.

Al observar los resultados obtenidos, hay que resaltar que el tiempo de CPU que se necesita para obtener la solución con la nueva metodología, en el peor de los casos fue de 218 minutos lo que equivale a 3.63 horas, lo cual es muy bueno en comparación con los resultados obtenidos por Hernández [6], ya que estos utilizaban un tiempo promedio de CPU de 15 horas. Es evidente que el tiempo de CPU se consume encontrando el sub-estimador lineal a trozos, mientras mas ajuste se le agregue a los parámetros l , ϵ y tol del método, este tiende a ajustar mejor la solución, pero aumenta el tiempo de CPU, una ventaja del método es que no depende del valor inicial para obtener la sub-estimación.

Como el menor error relativo en los resultados obtenidos es de 0,0657, entonces los parámetros que mejor se ajustan a los datos experimentales con la nueva metodología son:

$$\alpha_1 = 2,00 \quad \beta_1 = 7,00 \quad \gamma = 2,00$$

$$\alpha_2 = 0,05 \quad \beta_2 = 9,20$$

$$\alpha_3 = 0,80 \quad \beta_3 = 0,1950$$

$$\alpha_4 = 1,10 \quad \beta_4 = 0,0049$$

$$\beta_5 = 1,5$$

CAPÍTULO 4

CONCLUSIONES

- Los parámetros que mejor se ajustan a los datos experimentales con la nueva metodología son:

$$\alpha_1 = 2,00 \quad \beta_1 = 7,00 \quad \gamma = 2,00$$

$$\alpha_2 = 0,05 \quad \beta_2 = 9,20$$

$$\alpha_3 = 0,80 \quad \beta_3 = 0,1950$$

$$\alpha_4 = 1,10 \quad \beta_4 = 0,0049$$

$$\beta_5 = 1,5$$

- Se propone una metodología eficiente para ajustar los parámetros desconocidos en el modelo de la concentración de sólidos en columnas de burbujeo trifásicas
- Dicha metodología consiste en la implantación de un método de optimización glo-

bal, es decir se obtiene el mínimo global del problema para el ajuste de dichos parámetros

- Además es eficiente en el tiempo de CPU que se necesita para obtener el resultado.
- Hay que destacar que el algoritmo implementado no depende del valor inicial para obtener la solución , esto significa que la convergencia es global.
- Como las implementaciones se hicieron con el software MATLAB, se presento limitaciones al momento de aumentar la cantidad de puntos considerados en la region Λ del problema y al hacer un mayor ajuste en l debido a que se presentaba un problema en la memoria en el software.
- Es evidente que ajustando los parámetros en el algoritmo la solución obtenida se aproximaría mas a la solución optima (3.0.1) del problema.
- Hacer mayor ajuste en los parámetros en el algoritmo trae como consecuencia, que se utiliza mas tiempo de CPU para obtener la solución como se invidencia en los resultados obtenidos.

Hay que señalar de que existe la posibilidad de que el modelo de la concentración considerada no represente de la manera más precisa el problema físico, para verificar esto seria necesario realizar un estudio más a fondo sobre el modelaje del problema y verificar la confiabilidad de los datos experimentales. Sin embargo, verificar esta posibilidad no es

parte de este trabajo.

Recomendaciones para trabajos futuros:

1. Implementar los algoritmos utilizados en otro software por ejemplo FORTRAN, con la mismos datos utilizados para ver si disminuye el Tiempo de CPU. Además hacer mas ajustes en los datos de entrada para ver si se obtiene mayor aproximación en los resultados.
2. Como en el algoritmo de la sub-estimación lineal a trozos, el punto inicial y el sub-gradiente se escogen aleatoriamente, seria importante ver si se puede implementar otra estrategia para escoger estos valores para ver si se pueden mejorar los resultados.

APÉNDICES

APÉNDICE A

CÓDIGOS USADOS

En esta función se determina la matriz H y el vector t del problema (2.3.4).

```
function [H,t]=SLA2(X,y,l)

[n,m]=size(X); Id=eye(l,l); H=zeros(m+2*l,1+n+l*n+l*m);

H(1:m,1)=ones(m,1); H(1:m,2:n+1)=X';

for i=1:m

    H(i,1+n+l*n+l+(i-1)*l+1:1+n+l*n+l+i*l)=ones(1,l);

end

for h=1:m

    for i=1:l

        H(m+(h-1)*l+i,1+n+(i-1)*n+1:1+n+i*n)=X(:,h)';

        H(m+(h-1)*l+i,1+n+l*n+1:1+n+l*n+l)=Id(i,:);

    end

end
```

```

    H(m+(h-1)*l+i, 1+n+l*n+l+(h-1)*l+1:1+n+l*n+l+h*l)=-Id(i, :);
end
end
H(m+l*m+1:m+2*l*m, n+1:1+n+l*n+l)=-H(m+1:m+l*m, n+1:1+n+l*n+l);
j=1+n+l*n+l+1:1+n+l*n+l+l*m;
H(m+l*m+1:m+2*l*m, j)=H(m+1:m+l*m, j);  t=[y;zeros(2*l*m, 1)];

```

Esta función se utiliza para determinar los coeficientes p^k de la función objetivo en (2.3.4) y además se calcula la matriz de subgradientes.

```

function [P, sGradAiXbi]=coef2(Z1,X,l,E,tol)
[n,m]=size(X); alfa=Z1(1);
for I=2:(n+1)
    c(I-1)=Z1(I);
end
for i=1:l
    for j=1:n
        A(i, j)=Z1(i*n+1+j);
    end
end
for i=1:l
    bb(i)=Z1(1+n+l*n+i);

```

```
end

b=bb';

for h=1:m %CALCULAMOS LA MATRIZ DE SUBGRADIENTE de orden nx1

    for i=1:l

        if (A(i,:)*X(:,h)+b(i))> tol sGradAiXbi(i,h)=1;

        elseif (A(i,:)*X(:,h)+b(i))<-tol sGradAiXbi(i,h)=-1;

            else sGradAiXbi(i,h)=-1+2*rand;

        end

    end

end

end

pP=X*(sGradAiXbi'); P(1,1)=-m;

for I=2:(n+1)

    P(I,1)=-sum(X(I-1,:));

end

for i=1:l

    P(n+1+((i-1)*n+1):1+n+i*n,1)=-pP(:,i);

end

for i=1:l

    P(1+n+1*n+1:1+n+1*n+1,1)=-sum(sGradAiXbi');

end
```

```
P(1+n+l*n+l+1:1+n+l*n+l+l*m,1)=E;
```

Esta función se programó el Algoritmo 2, para obtener la sub-estimación lineal a trozos.

```
function [alfa,c,A,b,Nroiter]=subestlineatrozos(X,y,l,E,tol,ITERMAX)
```

```
%Algoritmo para estimar la sub-estimación lineal a trozos
```

```
%Datos de entrada: los puntos (X1,y1),..., (Xm,ym) donde yi=f(xi)
```

```
%estos puntos estan almacenados en la matrizX=[X1 X2... Xm] cuyas
```

```
%columnas estan compuestas por los ptos Xi y y=[y1;y2;...;ym], l es
```

```
%el numero de funciones lineales que genera la sub-estimacion lineal a
```

```
%trozos y tol>0 es la tolerancia y el parametro E (es epsilon)
```

```
%ITERMAX es el numero maximo de iteraciones en la algoritmo de la
```

```
%sub-estimación, esta se obtiene resolviendo el problema
```

```
%Para k=1,2,..., hasta que se cumple: P^(k-1)'*( Zk - Z(k-1) ) < tol
```

```
%      Zk= arg min fk(Z) =(P^(k-1))' * Z
```

```
%      sa      H*Z<=t
```

```
%      Z sin rest.
```

```
%Salida: alfa, c, A y b son los parámetros de la subestimaron lineal a
```

```
%trozos obtenida
```

```
[n,m]=size(X); Zo=randn(1+n+l*n+l+l*m,1); % se genera valor inicial
```

```
[H,t]=SLA2(X,y,l); %se obtiene la matriz H y el vector t
```



```
Po=coef2(Zo,X,l,E,tol); %se obtiene el vector de coeficientes de fo(Z)
nz=length(Po);
for i=1:nz
    lb(i)=-inf; ub(i)=inf;
end
options = optimset('LargeScale','off','Simplex','on');
%se obtiene el vector Zk usando el método simplex de Matlab
Z1=linprog(Po,H,t,[],[],lb,ub,[],options); k=0;
while (abs(Po'*(Z1-Zo))>=tol)& (k < ITERMAX)
Zo=Z1; Po=coef2(Zo,X,l,E,tol); %vector de coeficientes de fk(Z)
options = optimset('LargeScale','off','Simplex','on');
Z1=linprog(Po,H,t,[],[],lb,ub,[],options); %se obtiene el vector Zk
k=k+1;
end
Nroiter=k; alfa=Z1(1);
for I=2:(n+1)
    c(I-1,1)=Z1(I);
end
for i=1:l
    for j=1:n
```

```
        A(i,j)=Z1(i*n+1+j);
    end
end
for i=1:l
    bb(i)=Z1(1+n+l*n+i);
end
b=bb';
```

Esta es la función principal donde se ejecutan todas las funciones anteriores, con esta se resuelve el problema (2.4.2) con el cual se obtienen los parámetros que ajustan al modelo inverso (CBT) considerado.

```
function [alfa,beta,gamma,fval,exitflag,tiempo,Nroiter]=
        parametros2(X,y,lb,ub,l,E,tol,ITERMAX)
%En esta funcion se determinan los parametros del problema inverso
tic
[alfa,c,A,b,Nroiter]=subestlineatrozos(X,y,l,E,tol,ITERMAX);
cc=[c' ones(1,l)]'; H=[A -eye(l);-A -eye(l)];
bb=[-b;b]; nc=length(c); nb=length(b);
for i=1:nc
    llb(i)=lb(i); uub(i)=ub(i);
end
```

```
for i=(nc+1):(nc+nb)
    llb(i)=0; uub(i)=inf;
end

options = optimset('LargeScale','off','Simplex','on');

[xy,fval,exitflag]=linprog(cc,H,bb,[],[],llb',uub',[],options);

fval=fval+alfa;

for i=1:5
    beta(i)=xy(i);
end

for i=1:4
    alfa(i)=xy(5+i);
end

gamma=xy(10);

tiempo=toc;

tiempo=tiempo/60;
```

BIBLIOGRAFÍA

- [1] U. Ascher, J. Christiansen, and R.D. Russell, *A collocation solver for mixed order systems of boundary value problems*, Mathematics of Computations **33** (1979), no. 146, 659–679.
- [2] U. Ascher, J. Christiansen, and R.D. Russell, *Collocation software for boundary value odes*, ACM Transaction on Mathematical Software **7** (1981), no. 2, 209–222.
- [3] Cova D. R., *Catalyst suspension in gas-agitated tubular reactor*, Ind. Eng. Chem. Proc. Des. Dev. **5** (1966), no. 20.
- [4] G. Dassori and S. Buitrago, *Modelo en frío del reactor HDH: distribución de sólidos*, Tech. Report INT-02973.94, INTEVEP, S. A., Los Teques, 1994.
- [5] Hughmark G. A., *Holdup and mass transferring bubble columns*, Ind. Eng. Chem. Proc. Des. Dev. **6** (1967), no. 218.

-
- [6] J. Hernández, *Métodos numéricos en el modelado de columnas de burbujeo trifásicas*, Master's thesis, Universidad Simon Bolivar, 1997.
- [7] R. H. Jean, W. T. Tang, and L. S. Fan, *The sedimentation-dispersion model for slurry bubble columns*, *AIChE J.* **35** (1989), no. 4, 662–665.
- [8] O. Mangasarian, *Solution of general linear complementarity problems via nondifferentiable concave minimization*, *Acta Mathematica Vietnamita* **22** (1997), no. 1, 199–205.
- [9] O. Mangasarian and R. R. Meyer, *Nonlinear perturbation of linear programs*, *SIAM Journal on Control and Optimization* **17** (1979), no. 6, 745–752.
- [10] Mangasarian O. L., Rosen J.B., and Thompson. M.E., *Global optimization via piecewise-linear underestimation.*, *Journal of Global Optimization* **32** (2004), 1–9.
- [11] R. T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, New Jersey, 1970.
- [12] D. N. Smith and Ruether J. A., *Dispersed solid dynamics in a slurry bubble column*, *Chemical Engineering Science* **40** (1985), no. 5, 741–754.
- [13] D. N. Smith, J. A. Ruether, T. Shah, and M. N. Badgujar, *Modified sedimentation - dispersion model for solids in a three phase slurry column*, *AIChE J.* **32** (1986), 426–436.