

CONDICIÓN DE ESTABILIDAD NUMÉRICA PARA UN OPERADOR EN DIFERENCIAS FINITAS UTILIZADO PARA EL PROBLEMA INVERSO DE CONDUCCIÓN DE CALOR

NUMERICAL STABILITY CONDITION OF A FINITE DIFFERENCE OPERATOR USED FOR THE INVERSE HEAT CONDUCTION PROBLEM

Fernando B. Sanchez-Sarmiento

*Laboratorio de Investigación, Desarrollo y Transferencia de la Universidad Austral (LIDTUA),
Facultad de Ingeniería, Universidad Austral, Mariano Acosta 1611, B1629WWA Pilar, Argentina*

Palabras clave: Problemas inversos, Estabilidad numérica, Buen condicionamiento.

Resumen. Se suele afirmar que los problemas inversos de conducción de calor son problemas mal condicionados. Esto es, que el problema no tiene solución, que su solución no es única o que la solución no es estable. Para discutir esta afirmación, el artículo presenta un análisis de la estabilidad numérica de un operador en diferencias finitas utilizado para el problema inverso. El análisis se realiza para el problema transitorio unidimensional con simetría de revolución en el cual se quiere conocer la condición de contorno (curva de temperatura o de flujo de calor) cuando se conoce la temperatura en el centro del sólido. Este problema corresponde al ensayo ISO 9950 utilizado para determinar la característica de enfriamiento de los aceites industriales templantes. Se concluye que los operadores son condicionalmente estables. Anteriormente se demostró la existencia y unicidad de la solución. Con este trabajo, se demuestra que este problema inverso de conducción de calor puede resultar bien condicionado, dependiendo de los parámetros numéricos que se utilicen.

Keywords: IHCP, Numerical stability, well-posed problems.

Abstract. Inverse heat conduction problems are known as ill-posed problems. That is, the problem has no solution, its solution is not unique or the solution is not stable. To discuss this assumption, this work presents an analysis of the numerical stability of a finite difference operator used for the inverse problem. The analysis is carried out for the one-dimensional transient problem with symmetry of revolution in which we want to know the boundary condition (temperature or heat flow curve) when the temperature at the center of the solid is known. This problem corresponds to the ISO 9950 test used to determine the cooling characteristic of tempering industrial oils. It is concluded that the operators are conditionally stable. The existence and uniqueness of the solution was previously demonstrated. With this work, it is shown that this inverse heat conduction problem can be well posed, depending on the numerical parameters.

1. INTRODUCCIÓN

El DHCP (*Direct Heat Conduction Problem* - Problema directo de conducción de calor) consiste en obtener la temperatura en la probeta cuando se conoce la temperatura inicial, las propiedades térmicas del material y las condiciones de contorno. Para el caso de la probeta con la termocupla centrada la condición de contorno corresponde a su pared exterior ($r = r_p$), ver figura 1a. En cambio, el IHCP (*Inverse Heat Conduction Problem* - Problema Inverso de Conducción de Calor) consiste en obtener la condición de contorno en la superficie, cuando se conoce la historia térmica en al menos un punto interno. En este caso se conoce la curva del enfriamiento en el centro de la probeta y se quiere encontrar la temperatura y el flujo de calor en la pared exterior, ver figura 1 b.

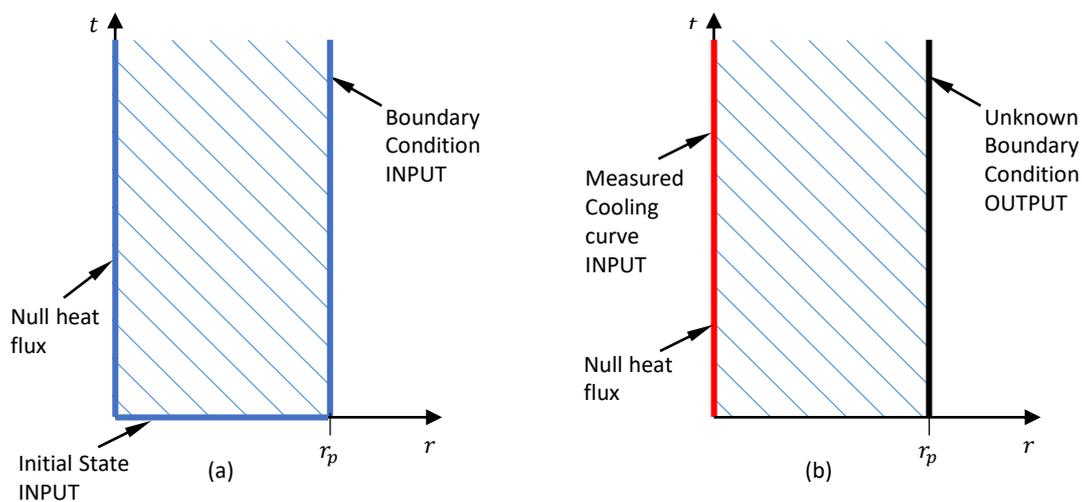


Figura 1: Dominios y condiciones de contorno del problema directo (a) y del problema inverso (b) de conducción de calor de la probeta con la termocupla centrada (Sanchez-Sarmiento, 2023).

Los problemas de conducción del calor se pueden clasificar en bien y mal condicionados (Hadamard, 2003). Se dice que un problema es bien condicionado (*Well-Posed*) si cumple las siguientes condiciones (Grysa, 2011):

1. Existe solución
2. La solución es única.
3. La solución es estable con respecto a perturbaciones en los datos.

La tercera condición se la puede redefinir de forma que la solución sea continua con los datos de entrada (Akrami y Erjaee, 2015).

Los problemas DHCP son bien condicionados porque cumplen las tres condiciones. Respecto a estos problemas IHCP las condiciones de existencia y unicidad han sido demostradas en (Weber, 1981) y (Beck et al., 1985), respectivamente. Por otra parte, Beck demuestra que la condición de estabilidad no queda garantizada ya que una pequeña variación en la curva de enfriamiento puede generar una diferencia arbitrariamente grande en la temperatura superficial (Beck et al., 1985). Para hacer la demostración equipara el problema inverso a un problema de optimización. Sin embargo, existen otras técnicas de resolución del IHCP que no consisten

en resolver el problema de optimización, entre otros, el método de DSouza (D´Souza, 1975) y el método de reconstrucción de temperaturas (Sanchez-Sarmiento y Cavaliere, 2021). En este trabajo se utiliza esta última técnica, aplicada al problema unidimensional y lineal, utilizando dos esquemas, el explícito centrado y el implícito centrado.

2. DEFINICIÓN DEL PROBLEMA

Sea el campo de temperatura de un cuerpo cilíndrico una función de la posición radial y del tiempo:

$$T = T(r, t), \quad (1)$$

en el cual solo hay transferencia de calor por difusión en la coordenada radial. A su vez, las propiedades del material, k , ρ y c_p , son constantes y conocidas. La función temperatura responde a:

$$\frac{k}{r} \frac{\partial T}{\partial r} + k \frac{\partial^2 T}{\partial r^2} = \frac{\partial T}{\partial t} \rho c_p, \quad (2)$$

donde r es la posición radial desde el centro de la probeta hasta su radio, r_p , y t el tiempo en el cual transcurre el enfriamiento:

$$\begin{aligned} 0 < r < r_p, \\ 0 < t. \end{aligned} \quad (3)$$

Al asumir la condición de simetría de revolución el centro debe tener un flujo de calor nulo:

$$q = -k \left. \frac{\partial T}{\partial r} \right|_{r=0} = 0, \quad (4)$$

se obtiene la condición de contorno en el interior, dada por

$$\left. \frac{\partial T}{\partial r} \right|_{r=0} = 0. \quad (5)$$

En el centro de la probeta ($r = 0$) se aplica el enfriamiento medido con la termocupla, dato del problema, que es una función del tiempo:

$$T(0, t) = T_m(t). \quad (6)$$

Para resolver este problema se propone el método de reconstrucción de temperaturas, el cual consiste en reconstruir las temperaturas en toda la probeta usando la ecuación 2, partiendo de las ecuaciones 5 y 6 para luego calcular las curvas flujo de calor en la superficie con las temperaturas de los últimos nodos.

3. ESQUEMA EN DIFERENCIAS FINITAS

Los esquemas utilizados aquí para calcular las temperaturas surgen de reemplazar a las derivadas parciales por los operadores en diferencias finitas de la ecuación diferencial

$$\frac{\partial T}{\partial r} \cong \frac{T_{j+1}^p - T_{j-1}^p}{2\Delta r}, \quad (7)$$

$$\frac{\partial^2 T}{\partial r^2} \cong \frac{T_{j+1}^p - 2T_j^p + T_{j-1}^p}{\Delta r^2}. \quad (8)$$

La función derivada temporal de la temperatura se la obtiene de la ec. 2:

$$f(t, r) = \frac{\partial T}{\partial t} = \frac{\alpha}{r} \frac{\partial T}{\partial r} + \alpha \frac{\partial^2 T}{\partial r^2}, \quad (9)$$

donde la difusividad es $\alpha = \frac{k}{\rho c_p}$.

En los dos esquemas se utiliza la misma discretización radial:

$$r_j = j\Delta r \quad (10)$$

Por lo tanto, al aplicar función derivada temporal en en el instante p se reemplazan las ecuaciones 7, 8 y 10 en 9, obteniendo:

$$f(t^p, r_j) = \frac{\alpha}{j\Delta r} \frac{T_{j+1}^p - T_{j-1}^p}{2\Delta r} + \alpha \frac{T_{j+1}^p - 2T_j^p + T_{j-1}^p}{\Delta r^2}, \quad (11)$$

Ordenando se obtiene:

$$f(t^p, r_j) = \frac{F_o}{\Delta t} \left(1 - \frac{1}{2j}\right) T_{j-1}^p + \frac{F_o}{\Delta t} (-2) T_j^p + \frac{F_o}{\Delta t} \left(1 + \frac{1}{2j}\right) T_{j+1}^p, \quad (12)$$

donde el número de Fourier, $F_o = \frac{\alpha \Delta t}{\Delta r^2}$.

En los dos casos los datos son las temperaturas en el centro de la probeta para todo instante de tiempo, T_0^{p+1} . Con esquemas de arranque se obtienen las temperaturas en el segundo nodo y luego con los esquemas presentados a continuación se calculan las temperaturas de todos los nodos siendo de interés solo las últimas, correspondientes a la superficie de la probeta.

3.1. Esquema explícito centrado

El esquema explícito centrado (EEC) consiste en calcular la temperatura en un instante $p+1$ con la función derivada evaluada en el instante anterior:

$$T_j^{p+1} = T_j^p + \Delta t f(t^p) \quad (13)$$

Por lo tanto, el operador directo, figura 2 a, resulta:

$$T_j^{p+1} = F_o \left(1 - \frac{1}{2j}\right) T_{j-1}^p + (1 - 2F_o) T_j^p + F_o \left(1 + \frac{1}{2j}\right) T_{j+1}^p \quad (14)$$

Despejando T_{j+1}^p , se obtiene el operador inverso, figura 2 b, obteniendo:

$$T_{j+1}^p = \frac{1 - 2j}{1 + 2j} T_{j-1}^p + \frac{4jF_o - 2j}{2jF_o + F_o} T_j^p + \frac{2j}{2jF_o + F_o} T_j^{p+1} \quad (15)$$

El operador de arranque es:

$$T_1^p = \frac{1}{2F_o} T_0^{p+1} + \frac{2F_o - 1}{2F_o} T_0^p \quad (16)$$

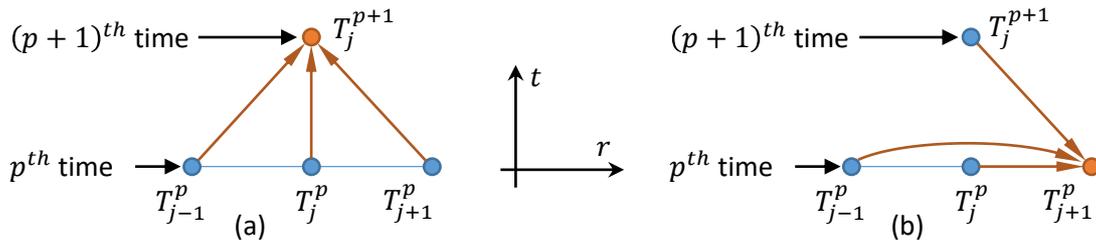


Figura 2: Molécula de cálculo del esquema explícito centrado (EEC) para el problema directo (a) y para el problema inverso (b), las flechas indican el sentido del cálculo (Sanchez-Sarmiento, 2023).

3.2. Esquema implícito centrado

El esquema implícito centrado (EIC) consiste en calcular la temperatura en un instante $p + 1$ con la función derivada evaluada en el mismo instante:

$$T_j^{p+1} = T_j^p + \Delta t f(t^{p+1}) \tag{17}$$

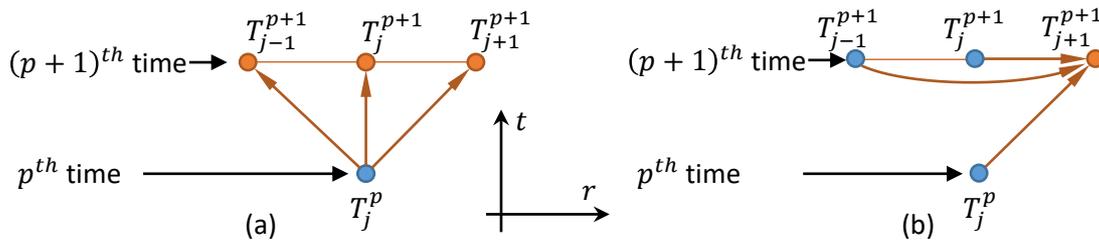


Figura 3: Molécula de cálculo del esquema de implícito centrado (EIC) para el problema directo (a) y para el problema inverso (b), las flechas indican el sentido del cálculo.

Por lo tanto, el operador directo implícito, figura 3 a, resulta:

$$F_o \left(1 - \frac{1}{2j} \right) T_{j-1}^{p+1} + (-2F_o - 1) T_j^{p+1} + F_o \left(1 + \frac{1}{2j} \right) T_{j+1}^{p+1} = -T_j^p \tag{18}$$

Usualmente se lo resuelve mediante un sistema de ecuaciones donde las incógnitas son las temperaturas en el instante $p + 1$ y los datos son las temperaturas en el instante p . Para el IHCP se despeja T_{j+1}^{p+1} , y ordenando se obtiene el operador inverso, figura 3 b:

$$T_{j+1}^{p+1} = \frac{1 - 2j}{2j + 1} T_{j-1}^{p+1} + \frac{2j + 4jF_o}{2jF_o + F_o} T_j^{p+1} + \frac{-2j}{2jF_o + F_o} T_j^p \tag{19}$$

El operador de arranque es:

$$T_1^{p+1} = \frac{1 + 2F_o}{2F_o} T_0^{p+1} - \frac{1}{2F_o} T_0^p \tag{20}$$

4. PROPAGACIÓN DE PERTURBACIONES

Se aplica una perturbación d para analizar si crece o disminuye durante la resolución del problema inverso. Debido a que es un problema lineal se aplica solamente la perturbación y en todos los demás nodos se deja el cero. En el caso del EEC la perturbación se aplica al primer nodo en el penúltimo instante (figura 4a).

$$EEC \begin{cases} d_j^p = 1 & p = n - 2 ; j = 0 \\ d_j^p = 0 & p < n - 2 ; j = 0 \\ d_j^p = 0 & p = n - 1 ; \forall j \end{cases} \quad (21)$$

En el caso del EIC la perturbación se aplica al primer nodo en el segundo instante (figura 4b).

$$EIC \begin{cases} d_j^p = 1 & p = 1 ; j = 0 \\ d_j^p = 0 & p > 1 ; j = 0 \\ d_j^p = 0 & p = 0 ; \forall j \end{cases} \quad (22)$$

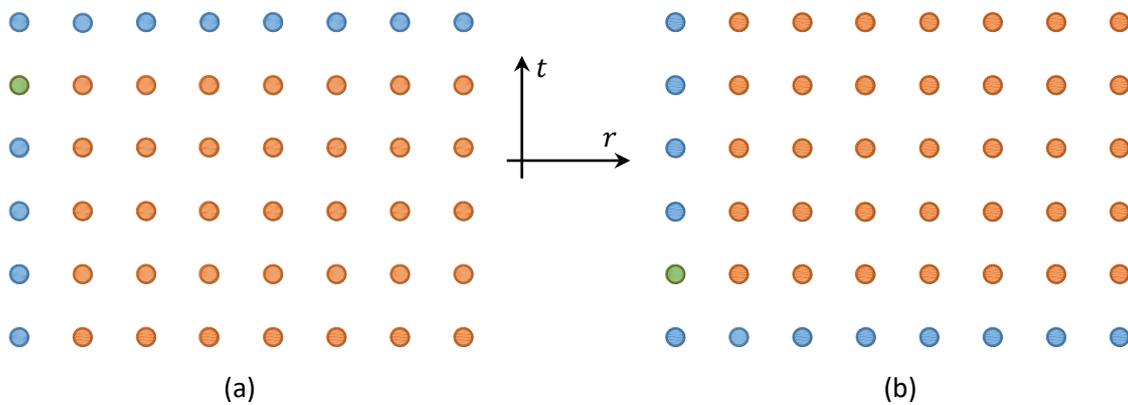


Figura 4: Representación de las corridas para el esquema explícito centrado (a) y esquema implícito centrado (b). La perturbación se aplica en el nodo verde, en azul las condiciones de contorno y finales (a) o iniciales (b) o y en naranja las temperaturas calculadas.

Se analiza la propagación de una perturbación d para analizar si la perturbación crece o disminuye con el factor de amplificación:

$$g(F_o, j) = \frac{\max_{\forall p} d_j^p}{\max_{\forall p} d_{j-1}^p} \quad (23)$$

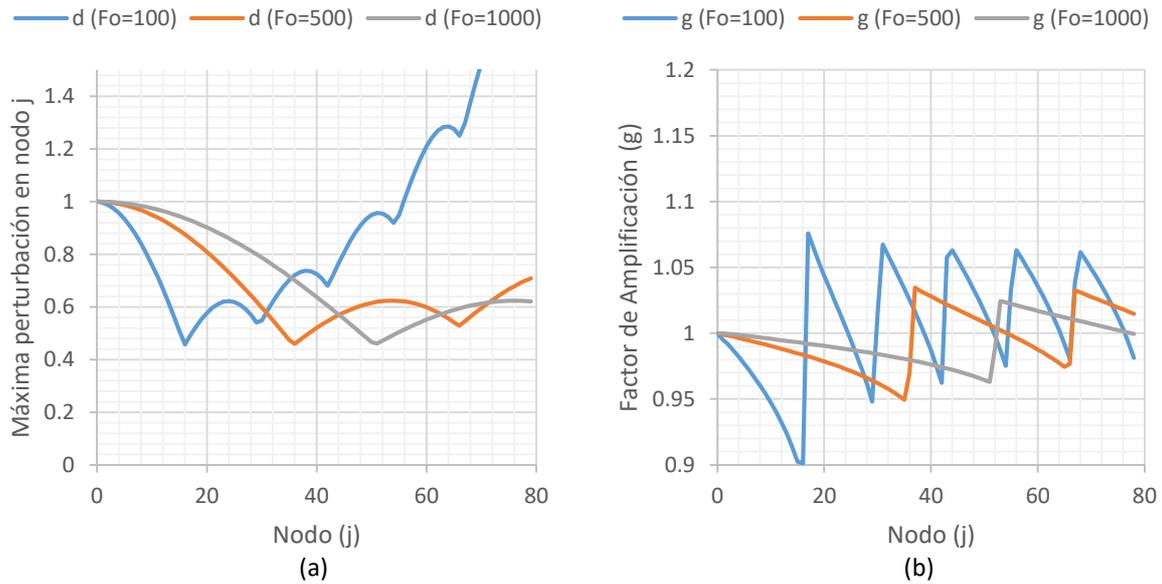


Figura 5: Propagación de una perturbación para tres números de Fourier (a) y su factor de amplificación (b) utilizando el esquema explícito centrado.

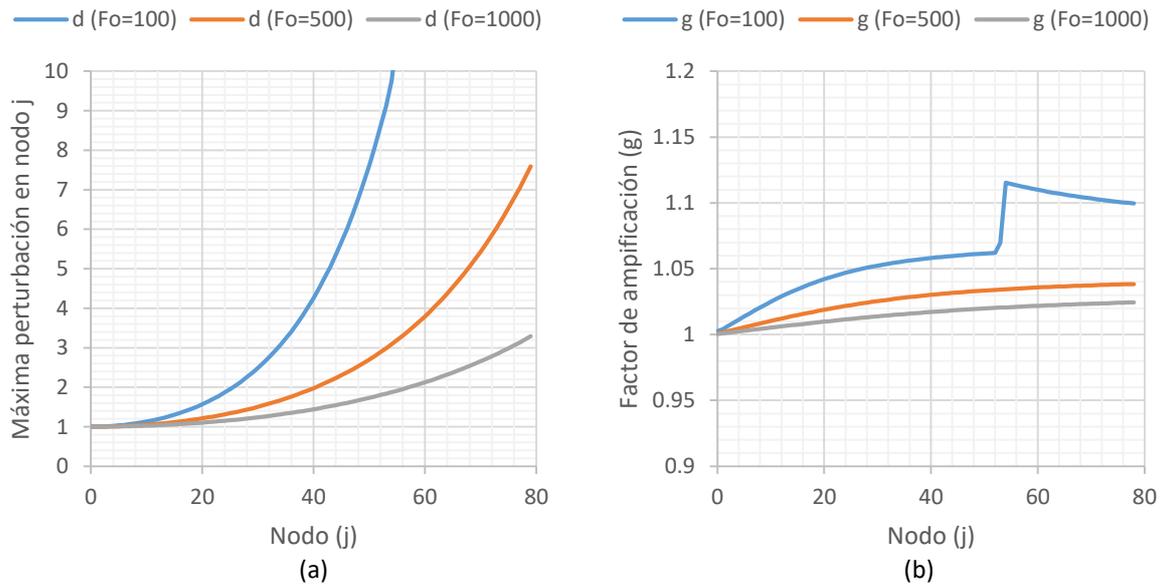


Figura 6: Propagación de una perturbación para tres números de Fourier (a) y su factor de amplificación (b) utilizando el esquema implícito centrado.

De esta manera, un factor de amplificación mayor que uno implica que la perturbación crece en el siguiente nodo, y por el contrario si es menor que uno la perturbación decrece al avanzar un nodo. Se hicieron corridas con los números de Fourier 100, 500 y 1000 con los dos esquemas. En la figura 5a se muestra la máxima perturbación para cada nodo obtenida con las tres corridas utilizando el EEC. Las corridas se realizaron en Python con el código del Anexo, utilizando la función *FactorAmplitudEEC*. Puede notarse que en las tres corridas para los primeros nodos la perturbaciones decrecen y a posterior crecen. En la figura 5b se grafican los correspondientes factores de amplificación. Puede notarse que en las tres corridas en algún nodo el factor es ma-

yor que uno y que el nodo donde ocurre crece con el número de Fourier.

Se realizaron corridas con los mismos números de Fourier con el esquema EEI. En este caso se utilizó la función *FactorAmplitudEEI* del código del Anexo. Puede notarse en la figura 6a que las perturbaciones crecen siempre, aunque en menor proporción con un mayor número de Fourier. En consecuencia, el factor de amplificación es siempre mayor que uno (figura 6b).

5. ESTABILIDAD NUMÉRICA

La estabilidad numérica se cumple cuando (Marshall, 1985)

$$g < 1. \quad (24)$$

El crecimiento de la perturbación depende del nodo (j) y del número de Fourier. Si el crecimiento de la perturbación, g, es menor que 1 entonces estamos en una zona estable, por el contrario, con un crecimiento mayor que 1 estamos en una zona inestable.

Se realizaron múltiples corridas variando el número de Fourier desde 10 hasta 1000, y se determinó el primer nodo cuyo g es mayor a uno (figura 7). De esta manera, en los nodos anteriores a dicho nodo se cumple la condición de estabilidad, y en los nodos posteriores se cumple la condición de inestabilidad. Cada corrida corresponde a ejecutar la función *Estabilidad* del código en una oportunidad.

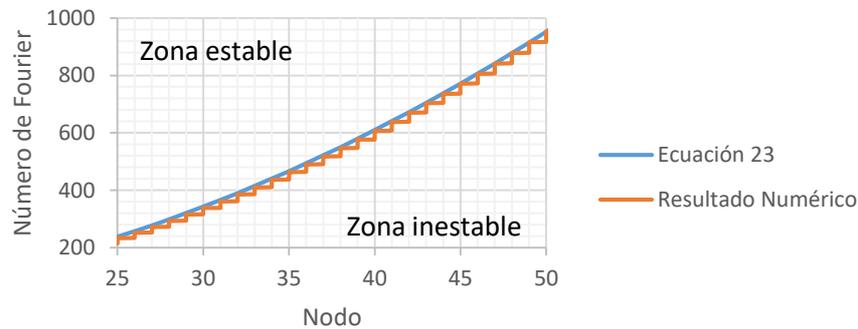


Figura 7: Zona estable ($g < 1$) y zona inestable ($g > 1$) para el EEC.

Mediante el método de cuadrados mínimos se obtiene la curva que ajusta a los puntos de la figura 7. Por lo tanto la condición de estabilidad del esquema EEC es:

$$F_o > \left(\frac{j}{1,62} \right)^2 \quad (25)$$

En el caso del EIC el factor de amplitud (g) es siempre mayor que uno, por lo tanto es incondicionalmente inestable.

6. CONCLUSIONES

Se concluye que estos problemas inversos al cumplir las tres condiciones puede resultar bien condicionados. Esto se logra con el método de reconstrucción de temperaturas, utilizando el esquema explícito centrado y los parámetros numéricos oportunos, dados por el número de Fourier.

A diferencia de los problemas directos, para estos problemas, con los dos esquemas utilizados las perturbaciones crecen más rápido cuanto menor sea el número de Fourier. Esto se debe a que la estabilidad de los operadores de diferencias finitas utilizados en problemas inversos es distinta que para los mismos operadores utilizados en problemas directos. El esquema explícito centrado es condicionalmente estable, a su vez, el esquema implícito centrado es incondicionalmente inestable.

La estabilidad del esquema explícito centrado depende del nodo y del número de Fourier que resulte de las propiedades térmicas del material y los parámetros numéricos utilizados (ecuación 25). Dado que la condición de estabilidad depende del nodo, puede ocurrir que la corrida sea estable en los primeros nodos y luego se desestabilice.

AGRADECIMIENTOS

Este proyecto se financió con fondos de la Universidad Austral.

REFERENCIAS

- Akrami M. y Erjaee G. Existence uniqueness and well-posed conditions on a class of fractional differential equations with boundary condition. *Journal of Fractional Calculus and Applications*, 6(2):171–185, 2015.
- Beck J.V., Blackwell B., y Clair Jr C.R.S. *Inverse heat conduction: Ill-posed problems*. Wiley, New York, 1985. ISBN 0471083194.
- D´Souza N. Numerical solution of one-dimensional inverse transient heat conduction by finite difference method. En *American Society of Mechanical Engineers, Winter Annual Meeting, Houston, Tex.* Houston, Texas, 1975.
- Grysa K. Inverse heat conduction problems. En V.S. Vikhrenko, editor, *Heat Conduction*, capítulo 1. IntechOpen, Rijeka, 2011. doi:10.5772/26575.
- Hadamard J. *Lectures on Cauchy's problem in linear partial differential equations*. Courier Corporation, 2003.
- Marshall G. *Solución numérica de ecuaciones diferenciales Tomo 1*. Editorial Reverté SA, Buenos Aires, 1985.
- Sanchez-Sarmiento F.B. *Caracterización sistemática de la transferencia de calor asociada a los tratamientos térmicos*. Phd thesis, Universidad Austral, Pilar, Buenos Aires, Argentina, 2023. Available at <https://rii.austral.edu.ar/handle/123456789/2141>.
- Sanchez-Sarmiento F.B. y Cavaliere M.A. Numerical Calculation of the Heat Transfer Coefficients in Multi-Thermocouple Probes by the Reconstruction Temperature Method. *Materials Performance and Characterization*, 10(1):420–442, 2021. ISSN 2379-1365. doi: 10.1520/MPC20200144.
- Weber C.F. Analysis and solution of the ill-posed inverse heat conduction problem. *International Journal of Heat and Mass Transfer*, 24(11):1783–1792, 1981. ISSN 0017-9310. doi:[https://doi.org/10.1016/0017-9310\(81\)90144-7](https://doi.org/10.1016/0017-9310(81)90144-7).

ANEXO: CÓDIGO IMPLEMENTADO EN PYTHON

```

import numpy as np
def FactorAmplitudEEC(F,n):
    T=np.zeros([n,n])
    T[n-2,0]=1
    for p in range(0,n-1):
        T[p,1]=T[p+1,0]/(2*F)+T[p,0]*(2*F-1)/(2*F)
    for j in range(1,n-1):
        for p in range(0,n-1):
            T[p,j+1]=(1-2*j)/(1+2*j)*T[p,j-1]
            T[p,j+1]+=(4*j*F-2*j)/(2*j*F+F)*T[p,j]
            T[p,j+1]+=(2*j)/(2*j*F+F)*T[p+1,j]
    g=[1]
    for j in range(1,n):
        g.append(max(T[:,j])/max(T[:,j-1]))
    return g
def FactorAmplitudEIC(F,n):
    T=np.zeros([n,n])
    T[1,0]=1
    for p in range(0,n-1):
        T[p+1,1]=T[p+1,0]*(1+2*F)/(2*F)-T[p,0]/(2*F)
    for j in range(1,n-1):
        for p in range(0,n-1):
            T[p+1,j+1]=(1-2*j)/(1+2*j)*T[p+1,j-1]
            T[p+1,j+1]+=(4*j*F+2*j)/(2*j*F+F)*T[p+1,j]
            T[p+1,j+1]+=(-2*j)/(2*j*F+F)*T[p,j]
    g=[]
    for j in range(n):
        g.append(max(T[:,j])/max(T[:,j-1]))
    return g
def Estabilidad(g):
    s=1
    for j in range(1,len(g)):
        if g[j]<=1:
            s=s+1
        else:
            return j

```