

Universidad Católica de Temuco Facultad de Ingeniería Departamento de Ciencias Matemáticas y Física

# Simulación numérica del crecimiento de sistemas híbridos silicio poroso-metal

Por

Cristian Mancilla Vargas

Profesor Guía

Dr. Gonzalo Recio Sánchez

Actividad Formativa Equivalente, para optar al grado de Magíster en Matemáticas Aplicadas (Profesional)

Temuco - 21 de agosto de 2017

Universidad Católica de Temuco Facultad de Ingeniería Departamento de Ciencias Matemáticas y Física

# COMISIÓN EVALUADORA

Profesor Guía:

Dr. Gonzalo Recio

Profesor informante:

Dr. Ramón Bécar

Profesor informante:

Dr. Stefan Berres

Director del Programa (Ministro de fe):

Dr. Emilio Cariaga

Temuco ·····

# Perfil de Egreso

Magíster en Matemáticas Aplicadas. Universidad Católica de Temuco.

El egresado del Magíster en Matemáticas Aplicadas es un profesional posgraduado que posee la competencia de aplicar la matemática al análisis de sistemas y procesos complejos en el ámbito de los fenómenos de transporte. Específicamente

Formula ecuaciones diferenciales como modelos matemáticos, en el ámbito de los fenómenos de transporte, para obtener una relación cuantitativa entre las variables relevantes del sistema.

Resuelve ecuaciones diferenciales como modelos matemáticos, utilizando técnicas numéricas y analíticas, para obtener valores cuantitativos de la variable respuesta del sistema.

Utiliza programas computacionales en la resolución, análisis y aplicación de ecuaciones diferenciales al mejoramiento de sistemas complejos en el ámbito de los fenómenos de transporte.

#### Abstract

In the present work, a matemathical model and its numerical simulations for the growth of porous silicon (SiP)-metal hybrid systems are presented for the specific case of the nucleation of copper ions on SiP surface by the experimental technique photoassisted immersion deposition. The growth of Cu nanoparticles can be represented by Hamilton-Jacobi equations, due to its behavior is similar as a front spreading. This front depends on concentration profiles of copper ions into the dissolutions which can be modeled by the diffusion equation. Level set method is used to model the front spreading of the nanoparticles growth by level curves. The front modifies the domain in which copper ions can move, involving a moving boundary problem. A bidimensional and evolutionary mathematical model is presented. This model is solved numerically by finite volume method for the moving boundary and finite differences for the diffusion of coper ions, both methods in its explicit form. The numerical methods are implemented computationally to obtain simulations of the growth. Representative parameters are used to analyze hybrid systems growth rate both, qualitatively and quantitatively, and computational results are compared with experimental results.

**Keywords:** Photoassisted deposition, Levet set method, bidimensional Hamilton-Jacobi, Godunov scheme, Semidiscrete central-upwind, Bidimensional diffusión, Reinitialization.

#### Resumen

En el presente trabajo se muestra el modelo matemático y sus respectivas simulaciones numéricas para el crecimiento de sistemas híbridos silicio poroso (SiP)-metal, tomando el caso específico de la nucleación de iones de cobre (Cu) en la superficie del SiP mediante la técnica experimental de deposición fotoasistida por inmersión. El crecimiento de las nanopartículas de cobre se puede representar mediante ecuaciones de Hamilton-Jacobi, ya que se comportan como un frente de propagación que depende de los perfiles de concentración de los iones de cobre en la disolución los cuales vienen modelados por la ecuación de difusión. Se introduce el método level set para capturar, mediante curvas de nivel, el frente de propagación del crecimiento. Este frente modifica el dominio donde se ubican los iones de cobre, conllevando a fronteras móviles en el tiempo en el problema a resolver. Se presenta un modelo bidimensional y evolutivo que se resuelve numéricamente con esquemas de volúmenes finitos para el borde móvil y diferencias finitas para la difusión de los iones de cobre, ambos casos de forma explícita. Los métodos numéricos utilizados se implementan computacionalmente para obtener simulaciones del fenómeno. Se utilizan parámetros representativos para un análisis cualitativo y cuantitativo de las velocidades de crecimiento de los sistemas híbridos y son comparados con resultados experimentales.

Palabras claves: Deposición fotoasistida, Método level set, Hamilton-Jacobi bidimensional, Esquema Godunov, Semidiscrete central-upwind, Difusión Bidimensional, Reinicialización.

# Índice

1	Introducción 1							
<b>2</b>	Objetivos							
3	Modelo conceptual         3.1       Concentración de cobre en disolución	<b>4</b> 6 6						
4	Modelo matemático         4.1       Ecuación level set para el crecimiento de nanopartículas de Cu         4.2       Difusión de iones de cobre en disolución         4.3       Condición inicial y de bordes         4.3.1       Borde móvil         4.4       Dominio del problema         4.5       Problema a resolver	8 9 10 10 10 11						
5	Modelo numérico       5.1         Esquema numérico para ecuación level set en volúmenes finitos       5.1.1         5.1.1       Discretización espacial       5.1.2         5.1.2       Flujo numérico esquema Godunov       5.1.3         5.1.3       Flujo numérico Lax-Friedrichs       5.1.3         5.2       Reinicialización función level set       5.1.4         5.3       Esquema numérico para la difusión de cobre en diferencias finitas       5.4         5.4       Mallado de esquema volumen finito v/s diferencia finita       5.5         5.5       Discretización temporal y estabilidad numérica       5.4	<b>12</b> 12 12 12 14 14 16 17 18						
6	Implementación computacional	20						
7	Resultados y discusión27.1Estudio del crecimiento de nanopartículas de $Cu$ en función del parámetro $F$ 77.2Estudio de la difusión de $Cu$ 77.3Estudio de regiones con distintas dimensiones77.4Estudio del crecimiento de $Cu$ en función de la condición inicial	22 22 25 27 29						
8	Conclusiones 31							
Α	Anexos          A.1       Códigos computacionales función level set         A.1.1       Reconstrucción espacial de v         A.1.2       Velocidades de propagación y Hamiltonianos         A.1.3       Flujos numéricos         A.1.4       Función de reinicialización         A.2       Códigos computacionales ecuación de difusión         A.2.1       Caracterización de las zonas de cobre depositado         A.2.3       Flujo numérico ecuación de difusión	<b>35</b> 35 37 38 39 42 42 42 42 44						

# 1 Introducción

La ciencia de la nanotecnología y sus aplicaciones se presentan en gran parte de las industrias, como la aeroespacial, electrónica, fotovoltaica, computacional, medicina, entre otras, aportando avances en la obtención de productos cada vez más pequeños, rápidos y eficientes, como es el caso de los microprocesadores, medicamentos, dispositivos tanto ópticos y biomedicos, sensores, células fotovoltaicas y una amplia gama de dispositivos electrónicos como fotónicos. Cada caso exige una mejora continua en los materiales utilizados, con respecto a sus rendimientos y propiedades. Una de las innovaciones más recientes es el desarrollo de nuevas combinaciones de materiales, integrando las principales características de cada uno, dando entrada a los llamados sistemas híbridos, correspondiente a la combinación de materiales. La adopción de nuevas características de un material se deben realizar en base a procesos simples de obtención y de bajo costo, recurriendo a materiales que sostengan un uso factible.

La técnica de deposición fotoasistida por inmersión de un sustrato cumple con los criterios antes mencionados, ya que es una técnica que no necesita idear condiciones especiales de proceso, sino que depende del sustrato a recubrir y la solución empleada, aprovechando las reacciones químicas entre ambos elementos del sistema para la formación del recubrimiento requerido en el sustrato. Por estas razones es un método recurrente en las industrias, ya sea en la preparación de recubrimientos cerámicos, vítreos, o bien para los recubrimientos metálicos como los procesos de galvanización o de cromado. Lo relevante de este proceso es poder seleccionar un sustrato ideal, según sea la aplicación, que sea compatible con esta técnica de fabricación de sistemas híbridos. Bajo esta línea se utiliza como sustrato el silicio poroso (SiP) por su compatibilidad a esta técnica de fabricación.

La versatilidad del cobre (Cu) lo convierte en el material de mayor demanda por su amplia aplicación en sistemas eléctricos y electrónicos, por su conductividad, ductilidad, maleabilidad y ser unos de los mejores conductores eléctricos, después de la plata. También considerar el oro y el platino que son de baja resistencia al paso de la electricidad, pero más costosos. Es por ello que la creación de sistemas híbridos basado en SiP y Cu, se ha hecho relevante con el paso de los años, ya que la combinación de ambos promueve materiales con propiedades avanzadas con respeto a cada uno en su forma natural.

Con respecto a la deposición de Cu en SiP que es una técnica innovadora que aprovecha la pequeña diferencia de energía entre el potencial de reducción de los iones de Cu y el potencial de oxidación del silicio (Si), proceso llamado reacciones de oxido-reducción (redox), promovida en la superficie del SiP por medio de la formación de portadores de carga ante la presencia de luz. Entonces la reducción de Cu sobre la superficie del sustrato ocurre por una reacción acoplada entre fotoexitación y reducción [1]. Tras esto, la simplicidad del proceso radica en ser una reacción espontánea, desde el momento en que la luz incida en el sustrato. Recio et. al. [1] describen el proceso experimental del fenómeno de deposición de Cu sobre el SiP mediante la inmersión del sustrato en una solución acuosa de sales de cobre (50 mM de  $CuSO_4$  y 1 mM de  $H_2SO_4$  para estabilizar el pH). La reacción inicia ante el encendido de una lampara halógena de Xenón (100 W) durante el tiempo de exposición requerida. Las reacciones de desplazamiento que dominan este proceso es catalizado por la luz empleada en donde los fotones, que favorecen el proceso, crean pares de electrones-huecos [1] que se encuentran en el sustrato y donde se reducirá el Cu. Los iones de  $Cu^{2+}$  recopilan estos electrones generando semillas en la superficie del sustrato, los siguientes iones de cobre se reducirán sobre estas semillas formando grupos metálicos en la superficie del sustrato. Posteriormente los grupos de Cu crecen por la incorporación de nuevos iones de  $Cu^{+2}$ reducidos ante la oxidación del Cu ya depositado.

El estudio experimental y seguimiento del fenómeno presentado posee elevados costos de producción y caracterización, ya que las aglomeración de iones Cu sobre la superficie del SiP son de tamaño nanométrico, éstos llamándose nanopartículas, que no se pueden apreciar con microscopia óptica. Para ello se requiere de otros tipos de microscopias de mayor tecnología, como la de fuerza atómica (STM o AFM) o electrónica (SEM o TEM), que hacen costoso hacer un seguimiento constante a estos procesos. Los riesgos de manipulación de las nanopartículas, que pueden traspasar las membranas de la piel, inhaladas o ingeridas, lo cual hace que estos procesos no se pueda manipular con facilidad. Estas complicaciones dan pie a la elaboración de estos sistemas (SiP - Cu), con el fin de dar predicción y optimizar dicho proceso frente a la variaciones de parámetros que contenga el sistema de fabricación.

# 2 Objetivos

# **Objetivo** general

Simular numéricamente el crecimiento de sistemas híbridos silicio poroso-metal.

# Objetivos específicos

- 1. Formular el modelo matemático que describe el crecimiento de sistemas híbridos en dos dimensiones.
- 2. Resolver numéricamente el modelo en dos dimensiones acoplando métodos de volúmenes finitos y diferencias finitas.
- 3. Interpretar físicamente los resultados obtenidos en las simulaciones a partir del estudio de parámetros relevantes.

#### 3 Modelo conceptual

Los sistemas híbridos SiP - Cu se obtienen por reacciones de desplazamiento asistida por luz. A partir de la inmersión de una placa de SiP en disolución acuosa de sulfato de cobre  $(CuSO_4)$ , la capa porosa del SiP actúa como agente reductor, permitiendo que los iones de Cu nucleen en su superficie. Este fenómeno está dado bajo las siguientes reacciones:

$$Si + 2H_2O \longrightarrow SiO_2 + 4H^+ + 4e^-$$
 (1)

$$SiH_x + 2H_2O \longrightarrow SiO_2 + (4+x)H^+ + (4+x)e^-$$
 (2)

$$Cu^{+2} + 2e^{-} \longrightarrow Cu \tag{3}$$

El potencial de reducción del Cu es más positivo que el de oxidación del Si, lo cual hace que los silicios cedan electrones hacia los iones de  $Cu^{+2}$  para que nuclee en forma metálica sobre la superficie del SiP.



Figura 1: Secuencia de la reacción de deposición fotoasistada por inmersión de la deposición de Cu en SiP. Una vez que el sustrato SiP se coloca en la solución de Cu (a), La luz se enciende para iluminar el sustrato y los fotones comienzan a crear los pares de electrones-hueco en el sustrato, que migran a la interfaz y serán recopilado por los  $Cu^{2+}$  (b); Una vez que los electrones están disponibles en la interfase, la alta diferencia de potencial redox entre el  $Cu^{2+}$  y el SiP permite la reducción directa del Cu en la superficie, formando las semillas del racimo (c); El proceso continúa y los núcleos de Cu forman grupos metálicos en la superficie (y parcialmente dentro de la matriz porosa), donde la formación de nuevos clusters compite con el crecimiento de los anteriores. En esta etapa, la reducción de Cu puede estar acoplada por la oxidación de la superficie del Cu (d); La deposición de fotoasistancia de Cu sobrepasa la unión de los cúmulos en una película de Cu continua sobre el sustrato SiP (e). Extraído de [1]

En los primeros segundos de proceso, la aglomeración de Cu en la superficie del SiP es de forma lenta, generando pequeños núcleos de Cu. Posteriormente se forman aglomeraciones de iones de Cu de tamaño nanométrico, a los cuales se denominan nanopartículas, en general de forma esférica. Esta forma geométrica nace de la familiarización del ion de Cu en depositarse en aquellas zonas de la nanopartícula donde hay mayor presencia de Cu depositado (ver figura 2). Para tiempos ya elevados en segundos, el número de nanopartículas se eleva sustancialmente, cubriendo la mayor parte de la superficie del sustrato, pero aun existiendo segmentos en donde la superficie del sustrato no han sido cubiertos. Dependiendo de las condiciones, este proceso puede tardar 60 segundos aproximadamente (ver figura 3).



Figura 2: Imágenes una lámina de Cu depositada sobre la superficie de silicio mediante reacciones de desplazamiento asistidas por luz. Extraído de [17].

La formación de nanopartículas de Cu, depende del mecanismo de Volmer-Weber [2], el cual indica que los iones de Cu al difundirse sobre la superficie del sustrato tienden a unirse para formar pequeños núcleos. El crecimiento se inicia a partir de estos núcleos en forma de aglomerados o islas que coalescen para acabar formando una capa continua [23]. La forma de nucleación se relaciona a la poca interacción química entre el Si y el Cu, que al comenzar la nucleación en alguna zona de la superficie, el resto de iones Cu se depositarán, o bien se reducen preferentemente sobre el Cu ya depositado en la superficie, ya que el Cu depositado se estará oxidando.



Figura 3: Imágenes del proceso de formación y crecimiento de una nanopartícula Cu en las superficie del SiP a distintos tiempos: 0s (a), 10s (b), 20s (c) y 60s (d). Extraído de [1]

#### 3.1 Concentración de cobre en disolución

Schacham-Diamand et al. [3] demostraron que bajo ciertas condiciones de proceso, la deposición de los iones de Cu sobre SiP depende de la concentración de éstos en la solución acuosa. De igual forma Recio et al. [1] en su estudio experimental sugiere la presencia de un perfil de difusión, típico en fabricaciones de sistemas híbridos con ausencia de potenciales eléctricos externos.



Figura 4: Esquema de la presencia de una capa de difusión a partir de un gradiente de concentración de iones de cobre. Extraído y adaptado de [21]

La difusión es el mecanismo más importante sobre el movimiento de iones, produciendo movimiento de los iones por la presencia de un gradiente de concentración de éstos. El proceso de deposición depende de que los iones de cobre en disolución puedan llegar a la superficie del sustrato y reducirse. Los iones reducidos se incorporan a la superficie del SiP, generando un transporte de masa, esta zona del fenómeno se le denomina capa de difusión. En esta capa es donde la concentración de cobre varía y posee una concentración menor a la zona que esta sobre la capa de difusión (ver figura 4).

#### 3.2 Crecimiento de nanopartículas de cobre

El aumento de tamaño de las nanopartículas de Cu sobre la superficie el SiP, se puede describir como un modelo de propagación que va a depender de la tasa de deposición del cobre. Para los problemas que involucran la evolución de curvas o superficies se utiliza una técnica geométrica denominada "Level set method".

El propósito inicial de esta técnica fue el de resolver problemas que involucren el movimiento de curvas o superficies, por ejemplo, la propagación de frentes de fuego y el procesamiento de imágenes [4]. Este método comienza con una función arbitraria (v = v(x, y) para superficies en 3D), que se propaga para recuperar formas y curvas de interés en una dimensión menor a través de sus lineas de nivel, estas curvas se propagan con una velocidad, en una dirección normal a la curva, a través de la evolución de la superficie en 3D [18].

El Método de level set (curvas de nivel) se basa en una representación implícita del frente por aproximaciones numéricas, comenzando con una función arbitraria v definida para el conjunto de nivel cero v = 0 correspondiente a la evolución de la curva  $\Gamma(t)$  que se mueve con respecto al tiempo t y definida implícitamente. La aproximación de estos tipos de problemas dependen del tiempo y un valor inicial para evolucionar la función de level set asociada, en donde el conjunto de nivel cero corresponde a la ubicación del frente de propagación [4] en cierto instante de tiempo.

Hsiu-Chuan [6] define p = v(x, y, t) para (x, y) en el frente, donde p es un valor prescrito relacionado con la densidad del cobre, de manera que la tasa de deposición para cualquier punto en la superficie de nucleación definida en [5] es:

$$\frac{Fc^r}{p} \tag{4}$$

con F,  $p \ge r$  parámetros del sistema.  $Fc^r$  determina la velocidad de deposición del ion de Cu que depende de la concentración y de un parámetro desconocido F que determina la velocidad con que las nanopartículas crecerán, que se relaciona con las condiciones del sistema. El movimiento del frente depende del campo de velocidad  $\vec{u}$  donde:

$$\vec{u} = -\frac{Fc^r}{p\vec{n}} \tag{5}$$

el signo negativo indica que el crecimiento va en dirección opuesta a la difusión del cobre, el vector normal depende de la relación entre el gradiente de velocidad y la norma de éste

$$\vec{n} = \frac{\nabla v}{|\nabla v|} \tag{6}$$

Los elementos descritos anteriormente son necesario para determinar la función level a utilizar.

# 4 Modelo matemático

Se presenta el problema a resolver en versión bidimensional, considerando una región de estudio  $[0, a] \times [0, b]$  correspondiente a una porción del sistema, donde se encuentra la disolución de iones de cobre difundiéndose (capa de difusión). El sustrato se encuentra al fondo de la disolución de forma horizontal en y = 0.



Figura 5: Región de estudio (A) a escala nanométrica.

#### 4.1 Ecuación level set para el crecimiento de nanopartículas de Cu

La ecuación level set, corresponde a una ecuación diferencial parcial de Hamilton-Jacobi [4] y [7], la cual representa la velocidad de propagación del frente de crecimiento de los sistemas híbridos silicio poroso-metal y viene dado como:

$$v_t + \vec{u} \cdot \nabla v = 0 \tag{7}$$

donde, v = v(x, y, z, t) es la velocidad con que se propaga el frente que depende de las dimensiones espaciales y del tiempo, también del campo de velocidad  $\vec{u} = (\mu, \nu)$  con  $\mu = x_t$  y  $\nu = y_t$ . Lo anterior para un curva cerrada  $\Gamma(t)$  moviéndose en dos dimensiones en el instante de tiempo t. La solución de este frente se expresa implícitamente por la linea de nivel cero v = 0. Sobre ésta linea de nivel (v > 0) se encuentra los iones de cobre en suspensión y bajo ella (v < 0) el cobre está depositado (ver figura 6).

Si se introduce las ecuaciones (5) y (6) en la ecuación descrita anteriormente, ésta se puede expresa de la siguiente manera

$$v_t = \frac{Fc^r}{p} \left| \nabla v \right| \tag{8}$$

correspondiente a la ecuación level set [4] del fenómeno presentado.



Figura 6: Frente del crecimiento de una nanopartícula de cobre. Se ilustra el vector velocidad en cada punto a la que se mueve el frente y su descomposición. Si indica v = 0 para el frente de propagación (curva  $\Gamma(t)$ ), v < 0 como la zona interior de la nanopartículas y v > 0 para la zona donde hay iones de cobre en disolución.

#### 4.2 Difusión de iones de cobre en disolución

La evidencia de perfiles de concentración sugiere un transporte de masa por parte de los iones de Cu, esta situación se gobierna por las ecuaciones de Fick. Precisamente se utiliza la segunda ley de Fick cuando la difusión varía con respecto al tiempo, vale decir en un estado no estacionario, y con un coeficiente de difusión independiente del tiempo. Proponiendo que la razón de cambio entre la concentración y el tiempo depende del coeficiente difusivo y la velocidad con que cambia el gradiente de concentración, y que el modelo de propagación depende de estos perfiles de concentración en la capa de difusión del problema, se presenta la ecuación diferencial parcial de difusión bidimensional:

$$c_t = D(c_{xx} + c_{yy}) \tag{9}$$

donde, c es la concentración de iones de cobre en la disolución que depende de la posición x, y, z y el tiempo t; c = c(x, y, z, t). D es el coeficiente de difusión.

La ecuación (9) representa la difusión de iones de cobre en disolución. Pero solamente existirá difusión en las zonas donde el cobre aún está en suspensión. Lo anterior se representa con valores positivos de la velocidad de propagación v > 0. Para el caso v < 0 indica que no hay difusión ya que se encuentra en la zona donde el cobre ya fue depositado. Entonces, con respecto a la posición se define v de la siguiente forma:

$$v = \begin{cases} > 0 & \text{zona de iones de Cu en suspension} \\ < 0 & \text{zona de iones de Cu depositados} \\ = 0 & \text{interfase (Superficie de la nanopárticula de Cu)} \end{cases}$$
(10)

El flujo de deposición de los iones de Cu sobre la superficie de la nanopartícula esta dado por:

$$D\frac{\partial c}{\partial n} = Fc^r \tag{11}$$

#### 4.3 Condición inicial y de bordes

Para t = 0 se tendrá una concentración inicial igual a cero en toda la región de estudio, c(x, y, 0) = 0. Pero en el borde superior y = b hay una concentración  $c_0$  que permanece fija a lo largo del tiempo (ver figura 7). Lo anterior se debe a que sobre este límite hay una cantidad ilimitada de iones de Cu asegurando una concentración permanente, y solamente existiendo difusión en la región de estudio.

En los bordes x = 0 y x = a los iones de cobre no se pueden depositar, más bien no habrá flujo de deposición porque no existe sustrato en donde se puedan depositar, representado por

$$\frac{\partial c}{\partial n} = 0 \tag{12}$$

#### 4.3.1 Borde móvil

El borde inferior y = 0, se encuentra el sustrato donde se desarrolla la deposición, generando el crecimiento de las nanopartículas en aquella superficie. Este borde se moverá con respecto al tiempo ya que se irá cubriendo la superficie con la aglomeración de Cu metálico en la superficie. Este borde representa las zonas donde los iones de Cu se podrán depositar, representado por el flujo de deposición, ecuación (11). El borde móvil que representa implícitamente el crecimiento de la superficie de la nanopartícula de Cu condicionará el dominio de la difusión de cobre en disolución, por lo que la ecuación (9) dependerá de este borde móvil.

El cobre tiene familiaridad en depositarse en los sectores donde ya se encuentran los iones depositados en forma metálica, ya que éstos últimos se oxidarán generando una reducción del ion de Cu más favorable con respecto a una reducción en el Si. Por lo tanto, en los sectores que no familiariza se considera la siguiente condición de borde  $\partial c/\partial n = 0$  indicando la ausencia de flujo de deposición en aquellas zonas. La forma en que se desplaza el borde móvil esta representado mediante la ecuación level set (8).

#### 4.4 Dominio del problema

En base a lo expuesto en [6] se define el dominio  $\Omega(t)$  considerando una región cuadrada  $E = [0, a] \ge [0, b]$ . Se define el contorno de E de la siguiente forma:

$$\Gamma_{1} = \{(x, y) \in E \mid y = 0; \ 0 \leq x \leq a\}$$
  
$$\Gamma_{2} = \{(x, y) \in E \mid y = b; \ 0 \leq x \leq a\}$$
  
$$\Gamma_{3} = \{(x, y) \in E \mid x = 0; \ x = a; \ 0 \leq y \leq b\}$$

En el proceso de deposición, los iones de cobre se reducen y se depositan sobre la superficie de la nanopartícula. El tamaño de éstas dependen del tiempo t, por lo cual se define el borde móvil como

$$\Gamma_{1,i}(t) \subset E, i = 1, 2, ..., N$$
 (13)

con N el número de zonas donde habrá deposición sobre el sustrato al tiempot. Entonces se define el conjunto

$$\Gamma_{1,N+1}(t) = \Gamma_1 \setminus \{(x,0) \mid (x,y_0) \in \Gamma_{1,i}(t), i = 1, 2, ..., N \text{ para algunos } y_0 \in [0,b] \}$$

 $\Gamma_{1,N+1}(t)$  representa las zonas en el fondo de la solución que no están cubiertas por las partículas de cobre. El dominio que esta en función del tiempo  $\Omega(t)$  esta formado por los contornos  $\bigcup_{i=1}^{N} \Gamma_{1,i}(t) \cup \Gamma_2 \cup \Gamma_3$  (ver figura 7).



Figura 7: Dominio del problema con respecto al crecimiento de una nanopartículas de cobre  $\Gamma_{1,1}(t)$ , para t > 0.

#### 4.5 Problema a resolver

Finalmente, se tiene la concentración de iones de cobre c en el punto  $(x, y) \in \Omega(t)$  y el tiempo ten el que evoluciona el fenómeno que actúa sobre el borde móvil  $\bigcup_{i=1}^{N} \Gamma_{1,i}(t)$ . La concentración esta dada por:

$$c_t = D(c_{xx} + c_{yy}), \qquad \text{en } \Omega(t), \quad t > 0.$$
(14)

Se tiene la condición inicial para t = 0:

$$c(x, y, 0) = 0 \quad \forall \quad (x, y) \in \Omega(0) \tag{15}$$

en donde no habrá semillas en el sustrato para dar inicio al proceso  $\Gamma_{1,i}(0), i = 1, 2, ..., N$ . Sujeto a las condiciones de borde:

$$c = c_0, \quad \text{en } \Gamma_2, \quad t > 0 \tag{16}$$

$$D\frac{\partial c}{\partial n} = Fc^r, \quad \text{en } \bigcup_{i=1}^N \Gamma_{1,i}, \quad t > 0$$
(17)

$$\frac{\partial c}{\partial n} = 0, \quad \text{en } \Gamma_{1,N+1} \cup \Gamma_3 \quad t > 0$$
(18)

con el borde móvil  $\Gamma_{1,i} \subset E, i = 1, 2, ..., N$ , descrito por la ecuación level set (8)

# 5 Modelo numérico

Se desarrolla un esquema numérico para dar solución numérica y simulación computacional del fenómeno correspondiente al crecimiento de nanopartículas de cobre en SiP, que depende de la concentración de iones cobre en solución, utilizando un esquema de volúmenes finitos para la ecuación level set (8) y de diferencias finitas para la ecuación de difusión (14).

#### 5.1 Esquema numérico para ecuación level set en volúmenes finitos

#### 5.1.1 Discretización espacial

El dominio se define en una malla cuadrada y equidistante en sus dos dimensiones espaciales x e y. Para los puntos  $x_i e y_j$ , y los puntos intermedios  $x_{i\pm\frac{1}{2}} e y_{j\pm\frac{1}{2}}$  se utiliza la siguiente denotación:

$$\begin{aligned} x_i &:= i \Delta x, & x_{i \pm \frac{1}{2}} &:= (i \pm 1/2) \Delta x, & i = 1, 2, ..., N. \\ y_j &:= j \Delta y, & y_{j \pm \frac{1}{2}} &:= (j \pm 1/2) \Delta y, & j = 1, 2, ..., N. \end{aligned}$$

donde  $\Delta x = \Delta y$ . El valor del centro de cada volumen  $C_{i,j}$  formado se define de la siguiente forma  $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ .

Los puntos de contorno del dominio se incluirán en el mallado, por medio de la ampliación del dominio en ambas direcciones. Se incrementa  $\frac{\Delta x}{2}$  y  $\frac{\Delta y}{2}$  según sea el caso, generando un dominio ampliado.

#### 5.1.2 Flujo numérico esquema Godunov

Kurganov et al. [8] desarrollaron "Semidiscrete central-upwind scheme" que es un esquema de segundo orden tipo Godunov para ecuaciones Hamilton-Jacobi en 1-D y 2-D, que se utilizará para resolver el frente de propagación de la ecuación level set (8) que se puede generalizar a la siguiente expresión,

$$v_t + H(v_x, v_y) = 0,$$
 (19)

donde el Hamiltoniano  $H(v_x, v_y) = -\frac{Fc^r}{p} |\nabla v|.$ 

Se comienza asumiendo que se ha llegado al tiempo  $t = t^n$ , n = 1, 2, ..., N el número de pasos en el tiempo, con una solución aproximada  $v_{i,j}^n \approx v(x_i, y_j, t^n)$  y se proyecta la evolución de esta solución en el tiempo  $t^{n+1}$  [8]. Se realiza una reconstrucción alrededor de cada punto  $(x_i, y_j)$  con respecto a cuatro direcciones (ver figura 8). Lo anterior se realiza mediante un polinomio interpolante de variable conservada  $\tilde{v}(x, y, t^n)$  y se estima la velocidad local de propagación  $a_{i,j}^{\pm}$  y  $b_{i,j}^{\pm}$ , para ambas direcciones  $x \in y$  respectivamente.



Figura 8: La reconstrucción se define sobre los cuatro triángulos, NW, NE, SW y SE alrededor de cada uno de los puntos  $(x_i, y_j)$  de la malla. Esquema extraído de [8].

El flujo numérico que representa la tasa de evolución de v al tiempo  $t^{n+1}$  es el siguiente:

$$\frac{d}{dt}v_{i,j}(t) = -\frac{a_{i,j}^- b_{i,j}^- H(v_x^+, v_y^+) - a_{i,j}^- b_{i,j}^+ H(v_x^+, v_y^-) - a_{i,j}^+ b_{i,j}^- H(v_x^-, v_y^+) + a_{i,j}^+ b_{i,j}^+ H(v_x^-, v_y^-)}{(a_{i,j}^+ - a_{i,j}^-)(b_{i,j}^+ - b_{i,j}^-)} - \frac{a_{i,j}^+ a_{i,j}^- (v_x^+ - v_x^-)}{a_{i,j}^+ - a_{i,j}^-} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i,j}^+ - b_{i,j}^-} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i,j}^+ - b_{i,j}^- (v_y^+ - v_y^-)} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i,j}^+ - b_{i,j}^- (v_y^+ - v_y^-)} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i,j}^+ - b_{i,j}^- (v_y^+ - v_y^-)} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i,j}^+ - b_{i,j}^- (v_y^+ - v_y^-)} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i,j}^+ - b_{i,j}^- (v_y^+ - v_y^-)} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i,j}^+ - b_{i,j}^- (v_y^+ - v_y^-)} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i,j}^+ - b_{i,j}^- (v_y^+ - v_y^-)} - \frac{b_{i,j}^+ b_{i,j}^- (v_y^+ - v_y^-)}{b_{i$$

donde  $v_x^{\pm}$  y  $v_y^{\pm}$  son las derivadas con respecto a  $x \in y$  respectivamente. Las velocidades locales de propagación se pueden estimar de la siguiente manera:

$$\begin{split} a_{i,j}^+ &:= \max_{\pm} \left\{ H_u(v_x^+, v_y^+), H_u(v_x^+, v_y^-), H_u(v_x^-, v_y^+), H_u(v_x^-, v_y^-), 0 \right\} \\ a_{i,j}^- &:= \min_{\pm} \left\{ H_u(v_x^+, v_y^+), H_u(v_x^+, v_y^-), H_u(v_x^-, v_y^+), H_u(v_x^-, v_y^-), 0 \right\} \\ b_{i,j}^+ &:= \max_{\pm} \left\{ H_w(v_x^+, v_y^+), H_w(v_x^+, v_y^-), H_w(v_x^-, v_y^+), H_w(v_x^-, v_y^-), 0 \right\} \\ b_{i,j}^- &:= \min_{\pm} \left\{ H_w(v_x^+, v_y^+), H_w(v_x^+, v_y^-), H_w(v_x^-, v_y^+), H_w(v_x^-, v_y^-), 0 \right\} \end{split}$$

donde se tiene las derivadas de H de la ecuación (19), representadas de la siguiente forma,

$$H_u(v_x^{\pm}, v_y^{\pm}) = -\frac{Fc^r}{p} \left( \frac{v_x^{\pm}}{\sqrt{(v_x^{\pm})^2 + (v_y^{\pm})^2}} \right), \qquad H_w(v_x^{\pm}, v_y^{\pm}) = -\frac{Fc^r}{p} \left( \frac{v_y^{\pm}}{\sqrt{(v_x^{\pm})^2 + (v_y^{\pm})^2}} \right). \tag{20}$$

Por último, se utiliza una interpolación cuadrática continua por partes para calcular  $v_x^{\pm} \ge v_y^{\pm}$ , en las dimensiones espaciales  $x \in y$ . De la siguiente forma:

$$v_x^{\pm} = \frac{(\Delta v)_{i\pm\frac{1}{2}}^n}{\Delta x} \mp \frac{(\Delta v)_{i\pm\frac{1}{2}}'}{2\Delta x}, \qquad v_y^{\pm} = \frac{(\Delta v)_{j\pm\frac{1}{2}}^n}{\Delta y} \mp \frac{(\Delta v)_{j\pm\frac{1}{2}}'}{2\Delta y}.$$
 (21)

donde

$$(\Delta v)_{i+\frac{1}{2}}^{n} := v_{i+1}^{n} - v_{i}^{n}, \qquad (\Delta v)_{j+\frac{1}{2}}^{n} := v_{j+1}^{n} - v_{j}^{n}$$

$$\tag{22}$$

El valor de  $(\Delta v)'_{i+\frac{1}{2}}$  se estima utilizando el parámetro limitador de pendientes *minmod* propuesto en [8], que resguarda las oscilaciones que pueda tener  $\tilde{v} = (x, y, t^n)$ , limitando las derivadas en la evolución de v de la siguiente forma:

$$(\Delta v)'_{i+\frac{1}{2}} = \operatorname{minmod}\left(\theta\left[(\Delta v)^{n}_{i+\frac{3}{2}} - (\Delta v)^{n}_{i+\frac{1}{2}}\right], \frac{1}{2}\left[(\Delta v)^{n}_{i+\frac{3}{2}} - (\Delta v)^{n}_{i-\frac{1}{2}}\right], \theta\left[(\Delta v)^{n}_{i+\frac{1}{2}} - (\Delta v)^{n}_{i-\frac{1}{2}}\right]\right)$$
(23)

donde  $\theta \in [1, 2]$  representa la disipación numérica del esquema, para  $\theta$ 's cercanos a 2 se generan aproximaciones más confiables, pero menos exactas, para  $\theta$ 's cercanos a 1 presenta el caso opuesto. En [8] se siguiere utiliza  $\theta = 1,5$ . Finalmente, la función multivariable *minmod* se define por

$$\operatorname{minmod}(x_1, x_2, \ldots) = \begin{cases} \operatorname{min}_i \{x_i\} & \text{si } x_i > 0 \ \forall i, \\ \operatorname{max}_i \{x_i\} & \text{si } x_i < 0 \ \forall i, \\ 0 & \text{otro caso.} \end{cases}$$
(24)

Para el caso de  $(\Delta v)'_{j+\frac{1}{2}}$  se calcula de forma similar a lo expuesto anteriormente en las expresiones (23) y (24), pero en el sentido de la dirección y.

De forma general, se fija cualquier y para obtener  $v_x^{\pm}$  y se fija x para  $v_y^{\pm}$ .

#### 5.1.3 Flujo numérico Lax-Friedrichs

Si la diferencia entre las velocidades locales  $a_{i,j}^+$  y  $a_{i,j}^-$ , o  $b_{i,j}^+$  y  $b_{i,j}^-$  son igual a cero, se produce una indefinición en el flujo numérico presentado anteriormente, luego es necesario utilizar un flujo numérico alternativo que reemplace al anterior. En estos casos se utiliza el flujo Lax-Friedrich de primer orden.

$$\frac{1}{4} \left( H(v_x^+, v_y^+) + H(v_x^+, v_y^-) + H(v_x^-, v_y^+) + H(v_x^-, v_y^-) \right) - \frac{1}{2} \alpha^x (v_x^+ - v_x^-) - \frac{1}{2} \alpha^y (v_y^+ - v_y^-)$$
(25)

donde

$$\alpha^{x} = \max\{a_{i,j}^{+}, a_{i,j}^{-}\}, \qquad \alpha^{y} = \max\{b_{i,j}^{+}, b_{i,j}^{-}\}.$$
(26)

#### 5.2 Reinicialización función level set

Según Osher y Sethian [10] es común que las funciones level set en su evolución presenten irregularidades por errores numéricos que pueden llevar a disminuir la estabilidad del esquema numérico. Para ello, la función level set v(x, y, t) en cada paso del tiempo en que evoluciona se introduce en un cálculo numérico, en base a una función de reinicialización que corrige estas irregularidades, manteniendo estable su evolución. Se comienza los cálculos numéricos con  $v = 1 \pm d^2$  donde d es una función de distancia, a la cual se degrada v, el signo  $\pm$  es negativo en  $\Omega^-$  (interior del frente) y positivo en  $\Omega^+$  (exterior del frente) [10].

Sussman et. al. [11] indican la ecuación de reinicialización

$$\phi_t + S(\phi)(|\nabla \phi| - 1) = 0 \tag{27}$$

donde

$$S(\phi) = \begin{cases} 1 & \text{si } \Omega^+, \\ -1 & \text{si } \Omega^-, \\ 0 & \text{en la interfase.} \end{cases}$$
(28)

Para el cálculo de  $S(\phi)$  se sugiere en [10] la utilización de la siguiente expresión

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla \phi|^2 (\Delta x)^2}}.$$
(29)

Se introduce la ecuación (29) en (27) y particularizando al problema propuesto se obtiene la siguiente expresión

$$v_t + \frac{v(\sqrt{v_x^2 + v_y^2} - 1)}{\sqrt{v^2 + (v_x^2 + v_y^2)(\Delta x)^2}} = 0$$
(30)

en donde tenemos un nuevo Hamiltoniano  $H(v, v_x, v_y)$  y sus respectivas derivadas  $H_x$  y  $H_y$ .

$$H(v, v_x, v_y) = \frac{v(\sqrt{v_x^2 + v_y^2} - 1)}{\sqrt{v^2 + (v_x^2 + v_y^2)(\Delta x)^2}},$$
(31)

$$H_x(v, v_x, v_y) = \frac{v(v^2v_x + v_x\Delta x^2\sqrt{v_x^2 + v_y^2})}{\sqrt{v_x^2 + v_y^2}\left[v^2 + (v_x^2 + v_y^2)\Delta x^2\right]^{3/2}}, \quad H_y(v, v_x, v_y) = \frac{v(v^2v_y + v_y\Delta x^2\sqrt{v_x^2 + v_y^2})}{\sqrt{v_x^2 + v_y^2}\left[v^2 + (v_x^2 + v_y^2)\Delta x^2\right]^{3/2}}.$$
 (32)

Estos elementos son introducidos en un esquema numérico auxiliar, creado en base al esquema central-upwind propuesto, reemplazando el Hamiltoniano de éste por los de (31) y (32). Cada evolución de v, en el esquema central-upwind, se introduce en el esquema auxiliar en cada paso de tiempo, entregando un v que genere curvas más suaves en el frente de crecimiento.



Figura 9: Gráficas de superficie y de nivel, del resultado numérico del problema 4.6 de [8]. Resultados no reinicializados (A) y resultados reinicializados (B).

#### 5.3 Esquema numérico para la difusión de cobre en diferencias finitas

Se presenta un método numérico con la principal idea de no implementar de forma explicita la condición de borde móvil, ya que el movimiento y ubicación de este borde está definido implícitamente por la función level set v, con respecto a la posición en el dominio que describe la ecuación (10).

La discretización por diferencias finitas de (14) queda del siguiente modo:

$$c_t = D \frac{c_{i+1,j} - 2c_{i,j} + c_{i-1,j}}{(\Delta x)^2} + D \frac{c_{i,j+1} - 2c_{i,j} + c_{i,j-1}}{(\Delta y)^2}$$
(33)

y se puede reestructurar de la siguiente manera:

$$c_{t} = \frac{D(c_{i+1,j} - c_{i,j}) - D(c_{i,j} - c_{i-1,j})}{(\Delta x)^{2}} + \frac{D(c_{i,j+1} - c_{i,j}) - D(c_{i,j} - c_{i,j-1})}{(\Delta y)^{2}}$$
(34)

Con respecto a la literatura presente y lo expuesto por Hsiu-Chun Wei en [9], se identifica un término de reacción R que representa el flujo de deposición, que se utilizará para acoplar el método de volúmenes finitos con el de diferencias finitas. Este término no solo depende de los coeficientes de la tasa de deposición en (4), sino también de la ubicación de deposición en (x, y). Agregando lo anterior en (34) y con  $\Delta x = \Delta y$ , se formula la versión semidiscreta del problema

$$\frac{dc}{dt} = \frac{d_{i+1/2,j}(c_{i+1,j} - c_{i,j}) + d_{i-1/2,j}(c_{i-1,j} - c_{i,j}) + d_{i,j+1/2}(c_{i,j-1} - c_{i,j}) + d_{i,j-1/2}(c_{i-1,j} - c_{i,j})}{(\Delta x)^2} + \frac{R_{i,j}}{\Delta x}$$

donde  $R_{i,j}$  representa el flujo de deposición y los coeficientes  $d_{i,j}$  se estiman de la siguiente forma:

$$d_{i+1/2,j} = \chi(\min(v_{i+1,j}, v_{i,j}))D, \qquad d_{i-1/2,j} = \chi(\min(v_{i-1,j}, v_{i,j}))D$$
(35)

$$d_{i,j+1/2} = \chi(\min(v_{i,j+1}, v_{i,j}))D, \qquad d_{i,j-1/2} = \chi(\min(v_{i,j-1}, v_{i,j}))D$$
(36)

tomando como ejemplo el coeficiente  $d_{i+1/2,j}$ , en que si uno de los valores  $v_{i+1,j}$  o  $v_{i,j}$  es positivo, entonces el coeficiente es igual a D, ya que para un v > 0 nos encontramos en la capa de difusión. Caso contrario, el coeficiente es igual a 0 para v < 0, donde no hay difusión ya que se encuentra en la zona donde el cobre está depositado. Para representar esta situación se utiliza la función  $\chi = \chi(v)$ , la cual se define

$$\chi(v) = \begin{cases} 1 & \text{si } v > 0, \\ 0 & \text{si } v < 0. \end{cases}$$
(37)

El flujo de deposición  $R_{i,j}$  esta modelado por

$$R_{i,j} = \operatorname{sign}(v_{i,j}) \Big( \chi(-v_{i,j}v_{i+1,j}) F c_{i+1/2,j}^r + \chi(-v_{i-1,j}v_{i,j}) F c_{i-1/2,j}^r + \chi(-v_{i,j}v_{i,j+1}) F c_{i,j+1/2}^r + \chi(-v_{i,j-1}v_{i,j}) F c_{i,j-1/2}^r \Big) \Big/ D$$

$$(38)$$

en donde se verifica si hay flujo en el borde, por medio de un cambio de signo en todos los bordes de un volumen. El termino  $\operatorname{sign}(v_{i,j})$  representa el signo del flujo, tal que  $v_{i,j} < 0$  indica concentraciones decrecientes y para  $v_{i,j} > 0$  concentraciones crecientes. Para obtener  $c_{i+1/2,j}$  de (38) se utiliza mín $(c_{i,j}, c_{i+1,j})$  que elige el valor que se encuentre más cercano al borde, ya que existen 4 posibilidades de ubicación con respecto al borde móvil, cobre en el interior del dominio de difusión, cobre suelto en el borde, cobre depositado en el borde y cobre depositado en el interior del borde.

#### 5.4 Mallado de esquema volumen finito v/s diferencia finita

Se requiere alinear el mallado de diferencias finitas al de volúmenes finitos. Por lo cual, cada nodo de la discretización por diferencia finta se sobrepone en cada centro  $C_{i,j}$  de los volúmenes del mallado inicial. Para ello se define:

$$c_{i+1/2,j} = \frac{c_{i,j} + c_{i+1,j}}{2}, \qquad c_{i-1/2,j} = \frac{c_{i-1,j} + c_{i,j}}{2}.$$
 (39)

y de forma análoga para  $c_{i,j+1/2}$  y  $c_{i,j-1/2}$ .

#### 5.5 Discretización temporal y estabilidad numérica

Para la discretización temporal de v del flujo numérico de la ecuación level set, y de c en la difusión del cobre, se utiliza el método de Euler, donde se define el número de niveles de tiempo  $n\Delta t := t^n$ , con n = 1, 2, ..., N. De este modo:

$$\frac{d}{dt}v_{i,j}(t) \approx \frac{v_{i,j}^{n+1} - v_{i,j}^{n}}{\Delta t_{VF}}, \qquad \frac{d}{dt}c_{i,j}(t) \approx \frac{c_{i,j}^{n+1} - c_{i,j}^{n}}{\Delta t_{DF}}$$
(40)

Tanto  $\Delta t_{VF}$  del esquema de volúmenes finitos y  $\Delta t_{DF}$  de diferencias finitas, se calculan de forma diferente con tal asegurar la estabilidad numérica en la evolución de las funciones. Ya que en un método estable los errores debido a las aproximaciones disminuyen a lo largo de la computación [14].

Para asegurar la estabilidad numérica en la evolución de v, su intervalo de tiempo  $\Delta t_{VF}$  se irá ajustando conforme v evoluciona, de acuerdo a lo expuesto en [12] se debe cumplir la siguiente relación:

$$\Delta t < \frac{\min(\Delta x, \Delta y)}{a_{i,j}^+} \tag{41}$$

Con lo anterior y lo expresado en [8], se utiliza el número de CFL (Courant-Friedrich-Levy), que es una relación entre el intervalo de tiempo y el tiempo de residencia en un volumen finito. De este modo, el intervalo de tiempo debe ser menor al tiempo que tarda en pasar la propagación del frente por una malla discreta,

$$CFL > \frac{v_x \Delta t_{VF}}{\Delta x} + \frac{v_y \Delta t_{VF}}{\Delta y} \tag{42}$$

este número según se sugiere en [8] debe ser igual 0,475. Por lo tanto, el intervalo de tiempo se ajusta en la evolución de v debe ser igual a

$$\Delta t_{VF} = 0,475 \left( \frac{\min(\Delta x, \Delta y)}{a_{i,j}^+} \right)$$
(43)

Para el caso de  $\Delta t_{DF}$ , se logra estabilidad numérica cuando  $\alpha \leq 0, 5$  [14], en donde

$$\alpha = \frac{D\Delta t_{DF}}{(\Delta x)^2} \tag{44}$$

Por lo tanto, el intervalo de tiempo para la difusión del cobre se calcula de la siguiente forma

$$\Delta t_{DF} = \frac{1}{2} \frac{(\Delta x)^2}{D} \tag{45}$$

Ahora,  $v \neq c$  evolucionarán simultáneamente con igual  $\Delta t$ , para ello en cada intervalo de tiempo se escoge el  $\Delta t$  menor de ambos esquemas:

$$\Delta t = \min\left(\Delta t_{VF}, \Delta t_{DF}\right) \tag{46}$$

En consecuencia, si se introduce un error en alguna etapa del proceso, este error se va amortiguando hasta prácticamente desaparecer, garantizando aproximaciones razonablemente precisas [14] de v(x, y, t) y de c(x, y, t).

# 6 Implementación computacional

Se desarrollan códigos computacionales para resolver el problema propuesto y realizar simulaciones computacionales del fenómeno en estudio. Este código es de elaboración propia e implementado en la plataforma MATLAB que posee un lenguaje basado en matrices. Los códigos se basan en los modelos numéricos propuestos en la sección anterior, que se presentan en la sección de anexos.

En primera instancia se implementa la reconstrucción espacial de v sujeta a las condiciones iniciales del problema, calculando los valores de la ecuación (22) y (23), necesarios para utilizar el polinomio interpolante (21), (véase anexo A.1.1). Se extiende el dominio en todas las direcciones por medio de una extrapolación lineal en los nodos fronterizos.

Seguidamente se estiman las derivadas del Hamiltoniano de la ecuación (19) y velocidades de propagación de ambos flujos numéricos (véase anexo A.1.2). Estos elementos son introducidos en la implementación de los flujos numéricos (véase anexo A.1.3). Se calcula el intervalo de tiempo a utilizar, finalizando con la evolución de v en  $t^{n+1}$ . De forma posterior, el v estimado se introduce en un código auxiliar (véase anexo A.1.4), en el cual se reinicializa v. Esto último se emplea en cada intervalo de tiempo.

En cuanto a la implementación de la ecuación de difusión del cobre (14), se comienza con un código que caracterice la región del dominio donde hay difusión de iones de cobre en suspensión y la región donde no hay difusión por haber cobre depositado (véase anexo A.2.1). Posteriormente se implementa el flujo de deposición R (véase anexo A.2.2), él cual es parte del flujo numérico de la ecuación de difusión. Finalmente, la concentración c evoluciona al tiempo  $t^{n+1}$  (véase anexo 2.2.3) actualizando c. Este último, ingresa a la función level set en cada nueva concentración, propagando la superficie de la nanopartículas de cobre.

Por lo tanto, la implementación del método numérico mantiene el siguiente ciclo de cálculo:



Figura 10: Secuencia de calculo de implementación computacional del modelo numérico.



Figura 11: Diagrama de flujo del programa de simulación de crecimiento de sistemas híbridos silicio porosometal.

# 7 Resultados y discusión

La evolución del crecimiento de Cu en las placas de SiP, depende del tiempo de proceso. Este crecimiento se calcula en cada paso de tiempo durante este recubrimiento. Para las simulaciones de este fenómeno se fija un  $\Delta x$  aproximadamente a 1 nm, y el coeficiente de difusión corresponde a  $D = 0.7 \times 10^{-5} cm^2/s$  [16] equivalente a  $0.7 \times 10^9 nm^2/s$ . Estas condiciones implican que el cálculo del intervalo de tiempo con respecto a  $\Delta x^2/(2D)$  sea cercano a cero, más bien  $\Delta t = 7.14 \times 10^{-10}s$ . Bajo este intervalo de tiempo se asegura la estabilidad numérica de las simulaciones .

Pero no es práctico utilizar un  $\Delta t$  tan pequeño en las simulaciones numéricas, ya que tomaría tiempos extremadamente prolongados para obtener resultados. Tomando estas consideraciones, se realizan simulaciones cualitativas con parámetros que permitan cortos tiempos de simulación, estabilidad numérica y que los resultados se ajusten a la información experimental.

Se simula computacionalmente el crecimiento de una nanopartícula de Cu cuya semilla se ubica en el centro del sustrato. Se considera un dominio cuadrado  $R = [0, 100] \times [0, 100]$  y la semilla ubicada en (50,0). Se utilizan los siguientes parámetros:

$$D = 250,$$
  $c_0 = 0.05,$   $r = 0.8,$   $p = 1.4$   
 $\Delta x = \Delta y = 1.$ 

# 7.1 Estudio del crecimiento de nanopartículas de Cu en función del parámetro F

Se desarrolla un set preliminar de simulaciones numéricas para distintos valores de F a distintos tiempos. Las simulaciones entregan curvas de nivel como solución al problema propuesto que representa la superficie de la nanopartícula (ver figura 12). El parámetro F es un valor desconocido que depende de las condiciones del sistema, pero es el que determina la velocidad con que crecerán las nanopartículas, por lo cual se hace un barrido de valores para este parámetro. La idea es encontrar aquel parámetro F que se ajuste de mejor forma al crecimiento desarrollado experimentalmente [1].

A modo de ejemplo se presentan los resultados para F = 30, F = 40 y F = 50 en la figura 12 y 13, donde se aprecia las curvas de nivel (v = 0) a distintos tiempos. En ambas figuras se aprecia la diferencia de crecimiento al a cambiar F, se puede distinguir que mientras aumenta el valor de F la velocidad de crecimiento aumenta sustancialmente. Entonces a mayor F la tasa de crecimiento será mayor.



Figura 12: Crecimiento representado por curvas de nivel para F = 30, F = 40 y F = 50. De la curva inferior hacia la superior los tiempos t = 10, t = 20, t = 30 y t = 40.



Figura 13: Crecimiento representado por curvas de nivel para t = 10, t = 20 y t = 40. De la curva inferior hacia la superior los parámetros F = 20, F = 30, F = 40 y F = 50.

A modo de resumen se construye el cuadro 1 considerando los resultados más significativos, indicando el crecimiento horizontal a escala nanométrica de la nanopartícula a diferentes F y tiempos, e información experimental. El cuadro se muestra a continuación:

Tiempo	F = 20	F = 30	F = 40	F = 50	F = 60	Experimental
10	5,9906	9,0090	12,0438	15,0921	$18,\!1548$	20
20	26,966	$40,\!6592$	$54,\!5086$	$68,\!5366$	$82,\!8254$	50
30	$55,\!9592$	$84,\!8452$	> 100	>> 100	>>> 100	88
Tasa	1,3483	2,0330	2,7254	3,4268	4,1413	2.5

Cuadro 1: Resultados del crecimiento de nanopartículas a escala namométrica (nm) y respectiva tasa de crecimiento.

La tasa de crecimiento expresada en el cuadro 1, se obtiene al ajustar linealmente los datos. Con esto se puede determinar que los resultados con parámetro F = 30 se asemejan a los datos experimentales [1] a diferencia de F = 20, F = 50 y F = 60 que se alejan a éstos datos con mayores velocidades de crecimiento e incluyendo crecimiento fuera de la región de estudio (ver figura 12), y un crecimiento menor para F = 20. Se considera a F = 30 como el parámetro que mejor se ajusta al comportamiento experimental. Para F = 40 no se asemeja de igual forma a los resultados experimentales, pero tiene una tasa de crecimiento cercana a éstos. En consecuencia las siguientes simulaciones se realizan con respecto a F = 30. Lo anterior se detalla en las siguientes gráficas



Figura 14: Gráfica de crecimiento con respecto a curva experimental: (A) F = 20, F = 40 - (B) F = 30.

En la gráfica 14.A se muestra el comportamiento experimental del crecimiento de nanopartículas en un intervalo de tiempo de [0,30] segundos. Se puede apreciar la similitud del crecimiento con F = 20, pero de forma más lenta. La curva F = 40 se ajusta a la velocidad para un determinado tiempo, porque sobre los 30*s* tiene una velocidad mayor. En la gráfica 14.B, el crecimiento con F = 30 es el que más se adecua a la curva teórica de crecimiento, dentro del intervalo indicado, la diferencia radica en los primeros instantes de tiempo donde la nucleación comienza de forma más tardía.

#### 7.2 Estudio de la difusión de Cu

Se presentan los siguientes esquemas que representan la relación entre el crecimiento de nanopartículas y la concentración de iones de Cu, para F = 30 a los tiempos t = 1, t = 10, t = 20, t = 30, y t = 40. Se ilustra el congelamiento del dominio de la difusión de iones de Cu, que representa la región donde los iones de cobre se encuentran depositados.



Figura 15: Difusión de los iones de cobre en t = 1 con F = 30.

En la figura 15 se ilustra la distribución de concentración de iones Cu para 1s de proceso, a este instante aún no se inicia la deposición en el sustrato, ya que los iones se deben difundir hasta la superficie del SiP para dar inicio a la nucleación.

En la figura 16 se gráfica perfiles de concentración para tiempos mayores a 10s en donde se evidencia la nucleación. A los 40s se puede apreciar que los iones de cobre ya se han difundido completamente por todo el dominio, proporcionando un aceleramiento al crecimiento de las nanopartículas. Con esto se puede determinar que el tamaño de las nanopartículas de Cu van a depender de la concentración de iones de Cu que se encuentre en cada punto de la región.



Figura 16: Crecimiento y difusión de los iones de cobre en los tiempos t = 10, t = 20, t = 30 y t = 40 para F = 30.

#### 7.3 Estudio de regiones con distintas dimensiones

Con respecto a la baja tasa de crecimiento para los primeros instantes de tiempo ilustrados en la figura 14, se puede suponer que se debe a una lenta difusión de los iones de cobre, porque la deposición comienza cuando los iones de Cu llegan a la superficie del SiP. Entonces se puede interpretar que aquellos primeros segundos representan el tiempo necesario en que llegan los iones de Cu a la superficie del sustrato. A continuación se presenta una simulación para una región mayor a los resultados anteriores, considerando una región  $R = [0, 220] \times [0, 220]$ , F = 30 y manteniendo los parámetros anteriores, y a varios tiempos. Se presenta los resultados en el siguiente cuadro:

Tiempo	t = 10	t = 20	t = 30	t = 40
F = 30	0,0454	2,8432	11,6373	26,6478
Experimental	20	50	88	> 100

Cuadro 2: Resultados del crecimiento de nanopartículas (nm) de una región de  $[0, 220] \times [0, 220]$ .



Figura 17: Crecimiento en una región de  $220 \times 220$  nm con F = 30 para t = 10, t = 20 y t = 30, con respecto a la curva experimental.

Ante el presente cuadro 2 y la gráfica 17, se ilustra que para el mismo parámetro F = 30 se obtiene un crecimiento notoriamente más lento. Se puede inferir que al hacer esta ampliación de la región, el cobre deberá difundirse el doble de distancia para llegar al sustrato y comenzar la nucleación. Según el gráfico, la nucleación comienza pasados los 10s en comparación a los resultados anteriores que lo hace antes de los 5s. Para fundamentar esta idea se presentan los perfiles de difusión en la figura 18 y las lineas de nivel de crecimiento, figura 19, ambos a distintos tiempos. Donde se puede apreciar la relación que hay entre el tiempo que tarda en difundirse los iones de Cu con respecto al inicio de la nucleación:



Figura 18: Distribución de concentración de iones de Cu con F = 30 a t = 10, t = 20, t = 30 y t = 40.

Para el primer gráfico de la figura 18, se evidencia que transcurrido 10s los iones de Cu no han llegado a la superficie del sustrato, como se había mencionado antes los iones de Cu debe recorrer una mayor distancia. En la vista ampliada del segundo gráfico de la figura 19 solamente se reflejan 3 lineas de nivel, por el mismo motivo en que a los primeros segundos no hay nucleación, por ende no hay nanopartícula.



Figura 19: Lineas de nivel del crecimiento de nanopartículas de Cu con F = 30 a t = 10, t = 20, t = 30 y t = 40, y la derecha una vista ampliada de estas lineas de nivel.

#### 7.4 Estudio del crecimiento de Cu en función de la condición inicial

Con lo anterior, se infiere que el inicio de la nucleación depende de que tan rápido pueda llegar el ion de Cu al sustrato, vale decir que el tiempo que se demora en iniciar la nucleación depende de la distancia que hay desde el sustrato hasta la concentración  $c_0$ . Bajo este análisis se genera un nuevo set de simulaciones manteniendo los parámetros anteriores a distintos F, pero se cambia la condición inicial.

Las simulaciones antes presentadas ejecutan la condición inicial expresada en la ecuación (16), que indica que la concentración  $c_0$  solamente estará disponible en el borde superior del dominio, para luego difundirse a la superficie del SiP. Ahora se proponen simulaciones cambiando esta condición inicial en donde la concentración inicial  $c_0$  estará distribuida uniformemente en todo el dominio del problema,  $c(x, y, 0) = c_0$  donde  $(x, y) \in \Omega(0)$ . Con esta nueva condición inicial se debería evitar el tiempo necesario en que el ion de Cu llegue a la superficie del sustrato. Bajo esta idea en las simulaciones realizadas se utilizó F = 5, F = 10, F = 20, F = 30 y F = 40, pero solamente se ilustra los resultados en el cuadro 3 y en la figura 20 las simulaciones con respecto a F = 10, ya que es el que más se ajusta al crecimiento experimental.

Tiempo	t = 10	t = 20	t = 30
F = 10	18,4144	36,8932	55,434
Experimental	20	50	88

Cuadro 3: Resultados del crecimiento de nanopartículas a escala namométrica (nm) para F = 10.



Figura 20: Curva de crecimiento para F = 10 con respecto a la curva experimental.

Para los F mayores a 10, el crecimiento fue excesivamente rápido sobrepasando la curva experimental y para F = 5 la tasa de crecimiento es baja en comparación a F = 10. Según refleja el gráfico de la figura 20 y los datos del cuadro 3, la nucleación comienza a los primeros segundos al igual que lo experimental, y a diferencia de los resultados anteriores (ver figura 14.B) se asemeja al crecimiento experimental. Pero el crecimiento que se tiene bajo la nueva condición inicial es de un comportamiento lineal, lo que es completamente distinto al comportamiento de la curva experimental. Es probable que al mantener un nivel de concentración constante en el dominio a lo largo del tiempo causaría que el crecimiento de las nanopartículas se mantenga constante, ya que según el modelo planteado este crecimiento depende de la concentración que se encuentra en cada punto del dominio. Por último se presenta las curvas de nivel para este caso en particular, a distintos tiempos y valores de F.



Figura 21: Curvas de nivel para F = 20 y F = 30 a los tiempos t = 10 y t = 20



Figura 22: Curvas de nivel para F = 10 para t = 10, t = 20, t = 30 y t = 40.

# 8 Conclusiones

El método numérico utilizado para simular el crecimiento de nanopartículas de Cu en la superficie de SiP representa de buena forma el modelo matemático propuestos, correspondiente a la ecuación de difusión con un borde móvil. Este borde se logra implementar implícitamente por medio de una función level que se comporta como frente de propagación y correspondiente a una ecuación Hamilton-Jacobi. Con los resultados obtenidos se evidencia aproximaciones al comportamiento de los fenómenos presentado. Las dos características del problema se discretizan mediante diferencias finitas y volúmenes finitos respectivamente que se acoplan mediante un termino de reacción equivalente al flujo de deposición, esta metodología es una alternativa a lo ya realizado para las aproximaciones numérica de los problemas con borde móvil.

Se hace un estudio con simulaciones cualitativas del problema con parámetros relevantes por la inviabilidad de generar simulaciones cuantitativas con respecto a parámetros teóricos. Tras los primeros resultados se evidencia que el parámetro F = 30 es el que se ajuste de mejor manera a los datos experimentales disponibles. En primera instancia los resultados obtenidos modelan una curva de crecimiento semejante a la información experimental, pero con la diferencia en que el crecimiento a los primeros segundos es más lento en las simulaciones. Se genera otro set de simulaciones para determinar que provoca aquella diferencia, con estos nuevos resultados se infiere que esta diferencia puede depender de la rapidez con que llegue el ion de Cu a la superficie del SiP, ya que se realizaron simulaciones con diferentes distancias entre el sustrato y la ubicación de la concentración inicial.

Las simulaciones computacionales sirven para encontrar las curvas de crecimiento que se adapten de mejor forma las curvas experimentales, pero falta información con respecto al sistema para escoger parámetros más representativos, como saber a que distancia se encuentra la concentración inicial al sustrato, mayor cantidad de datos experimentales para poder considerar tiempos de simulación apropiados, entre otros. Por medio de la información experimental presentada se hace necesario revisar el modelo matemático, por medio de una descripción detalla del experimento y del modelo conceptual.

Se expresó la poca factibilidad de realizar simulaciones con información experimental, porque conllevan largos tiempos de simulación, en especial los métodos numéricos utilizados que necesitan pequeños intervalos de tiempos para mantener la estabilidad numérica y también por el cálculo explícito que se realiza. La mayor restricción aparece por la parte parabólica del problema en donde se podría resolver implícitamente, asegurando estabilidad y disminución de los tiempos de simulación. En este trabajo los parámetros se ajustaron para no tener intervalos de tiempos pequeños y poder mantener la estabilidad numérica. En particular el código generado ocupa elevados tiempo de procesamiento por la cantidad de cálculos que debe realizar. Según la literatura disponible el uso de métodos implícitos tampoco aseguran el uso de datos experimentales esto por la naturaleza del problema y los coeficientes que lo gobiernan. Las representaciones de las lineas de nivel, no tiene completa similitud a la realidad, ya que el modelo no contempla que la deposición depende del tamaño de la partícula y la afinidad que tiene los iones en depositarse en las zonas donde existe mayor cantidad de Cu generando nanopartículas de apariencia esférica, pero en el programa los iones de cobre se van depositando de forma uniforme en toda la superficie de la nanopartícula simulada.

#### Trabajos futuros

Queda pendiente en este trabajo el poder realizar ajustes al programa computacional, como la inclusión de una nueva variable que determine en donde se depositará el ion de cobre, generar un estudio de conservación de masa en las simulaciones entre el Cu depositado y el Cu en suspensión, desarrollo de un esquema numérico semi-implícito para evitar la restricción del intervalo del tiempo en la ecuación de difusión, recopilar mayor información de los aspectos físicos que tenga el problema y mayor cantidad de datos experimentales a varios tiempos, generar un análisis de la precisión en la aproximaciones utilizando distintos mallados incluyendo las finas. Finalmente, el método numérico implementado computacionalmente se puede tomar como una herramienta más para el estudio del crecimiento de sistemas híbridos en base al estudio de parámetros significativos para determinar ciertas condiciones y factores que influyen en el fenómeno, pero no se garantiza simulaciones exactas a las realidad, para ello se debe mejorar el modelo matemático, numérico y computacional.

## Referencias

- Recio, G., Gallach, D., Manso Silvan, M., Fukami, K., Martin Palma, R. J., Castro, G. R., and Muñoz-Noval, A. (2014). Photoassisted Immersion Deposition of Cu Clusters onto Porous Silicon: A Langmuir-Hill Ligand-Locus Model Applied to the Growth Kinetics. The Journal of Physical Chemistry C, 118(27), 14905-14912.
- [2] Budevski, E. B., Staikov, G. T., and Lorenz, W. J. (2008). Electrochemical phase formation and growth: an introduction to the initial stages of metal deposition. John Wiley and Sons.
- [3] Shacham-Diamand, Y., Dubin, V., and Angyal, M. (1995). Electroless copper deposition for ULSI. Thin Solid Films, 262(1-2), 93-103.
- [4] Sethian, J. A., and Smereka, P. (2003). Level set methods for fluid interfaces. Annual review of fluid mechanics, 35(1), 341-372.
- [5] Smy, T., Tan, L., Dew, S. K., Brett, M. J., Shacham?Diamand, Y., and Desilva, M. (1997). Simulation of electroless deposition of Cu thin films for very large scale integration metallization. Journal of the Electrochemical Society, 144(6), 2115-2122.
- [6] Wei, H. C. (2011). Qualitative simulation of the growth of electrolessly deposited Cu thin films. Mathematical Problems in Engineering, 2011.
- [7] Osher, S., and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of computational physics, 79(1), 12-49.
- [8] Kurganov, A., Noelle, S., and Petrova, G. (2001). Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton–Jacobi equations. SIAM Journal on Scientific Computing, 23(3), 707-740.
- [9] Shu, C. W. (2007). High order numerical methods for time dependent Hamilton-Jacobi equations. Mathematics and Computation in Imaging Science and Information Processing, Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap, 11, 47-91.
- [10] Osher, S. and Sethian, J., Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, J. Comput. Phys. 79, 12-49 (1988).
- [11] Sussman, M., Smereka, P., and Osher, S., A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, J. Comput. Phys. 114, 146-159 (1994).
- [12] Coronel, A., Cumsille, P., & Quezada, R. (2010). Un método numérico híbrido para capturar los choques en leyes de conservación escalares. Revista Integración, 28(2), 111-132.
- [13] Kurganov, A., and Tadmor, E. (2000). New high-resolution central schemes for nonlinear conservation laws and convection?diffusion equations. Journal of Computational Physics, 160(1), 241-282.
- [14] Mathews, J. H., Mathews, K. D. J. H., and Fink, K. D. (2000). Métodos numéricos con Matlab.
- [15] Osher, S., & Fedkiw, R. (2006). Level set methods and dynamic implicit surfaces (Vol. 153). Springer Science & Business Media.
- [16] Fukami, K., Chourou, M. L., Sakka, T., and Ogata, Y. H. (2011). Numerical simulation of copper filling within mesoporous silicon by electrodeposition. physica status solidi (a), 208(6), 1407-1411.
- [17] Recio, G., (2013). Aplicaciones fotónicas y biomédicas de silicio poroso nanoestructurado (Doctoral dissertation, Universidad Autónoma de Madrid).
- [18] Torres-Ulloa, C. (2015). Predicción del frente espuma-petróleo en coordenadas Eulerianas (Masters thesis, Universidad Católica de Temuco).

- [19] Faires, J. D., and Burden, R. (1998). Numerical Methods, Brooks. Cole, second ed., Pacific Grove, CA, USA.
- [20] Martín, M. Y. C. (2003). Recubrimientos protectores obtenidos por deposición electroforética (EPD) a partir de suspensión sol-gel (Doctoral dissertation, Universidad Autónoma de Madrid).
- [21] Serrano, Á. L. (2012). Electrodeposición de nanohilos magnéticos: aleaciones de FeNi y óxidos de hierro y de cobalto (Doctoral dissertation, Universidad Complutense de Madrid).
- [22] Lazo Acuña, P. A. (2014). Estudio de la nucleación y crecimiento de aleaciones obtenidas por electro co deposición de molibdeno y níquel en solución de citrato amonio, Universidad de Chile.
- [23] Martín, J. M. A. (Ed.). (2003). Láminas delgadas y recubrimientos: preparación, propiedades y aplicaciones (Vol. 11). Editorial CSIC-CSIC Press.

## A Anexos

#### A.1 Códigos computacionales función level set

Se presenta la implementación computacional de la función level set, que representa el crecimiento de nanopartículas de cobre en silicio poroso.

#### A.1.1 Reconstrucción espacial de v

• Calculo de  $(\Delta v)_{i+\frac{1}{2},j}^{n}$ ,  $(\Delta v)_{i-\frac{1}{2},j}^{n}$ ,  $(\Delta v)_{i+\frac{3}{2},j}^{n}$ ,  $(\Delta v)_{i-\frac{3}{2},j}^{n}$ ,  $(\Delta v)_{i,j+\frac{1}{2}}^{n}$ ,  $(\Delta v)_{i,j-\frac{1}{2}}^{n}$ ,  $(\Delta v)_{i,j+\frac{3}{2}}^{n}$  y  $(\Delta v)_{i,j-\frac{3}{2}}^{n}$ , mediante el siguiente código:

```
for i=1:N
               %i+1/2
    for j=1:N+1 %nodo frontera
        x1(i,j)=v(i+1,j,k)-v(i,j,k);
                                         %i+1/2
        x1(N+1,j)=v(N+1,j,k)-v(N,j,k); %nodo frontera
    end
end
for i=2:N+1
               %i-1/2
    for j=1:N+1
        x2(i,j)=v(i,j,k)-v(i-1,j,k);
                                          %i-1/2
        x2(1,j)=v(2,j,k)-v(1,j,k);
    end
end
for i=1:N-1
               %i+3/2
    for j=1:N+1
        x3(i,j)=v(i+2,j,k)-v(i+1,j,k);
                                          %i+3/2
        x3(N,j)=v(N+1,j,k)-v(N,j,k);
        x3(N+1,j)=v(N+1,j,k)-v(N,j,k);
    end
end
for i=3:N+1
               %i-3/2
    for j=1:N+1
        x4(i,j)=v(i-1,j,k)-v(i-2,j,k);
                                           %i-3/2
        x4(2,j)=v(2,j,k)-v(1,j,k);
        x4(1,j)=v(2,j,k)-v(1,j,k);
    end
end
for i=1:N+1
               %j+1/2
    for j=1:N
        y1(i,j)=v(i,j+1,k)-v(i,j,k);
                                           %j+1/2
        y1(i,N+1)=v(i,N+1,k)-v(i,N,k);
    end
end
for i=1:N+1
               %j-1/2
    for j=2:N+1
        y2(i,j)=v(i,j,k)-v(i,j-1,k);
                                           %j-1/2
        y2(i,1)=v(i,2,k)-v(i,1,k);
    end
end
for i=1:N+1 %j+3/2
    for j=1:N-1
        y3(i,j)=v(i,j+2,k)-v(i,j+1,k);
                                           %j+3/2
        y3(i,N)=v(i,N+1,k)-v(i,N,k);
        y3(i,N+1)=v(i,N+1,k)-v(i,N,k);
    end
end
for i=1:N+1
               %j-3/2
   for j=3:N+1
```

```
\begin{array}{c} y4(i,j)=v(i,j-1,k)-v(i,j-2,k)\,; \qquad \mbox{\sc k}j-3/2\\ y4(i,2)=v(i,2,k)-v(i,1,k)\,;\\ y4(i,1)=v(i,2,k)-v(i,1,k)\,;\\ end\\ end \end{array}
```

• Código para estimar los valores de  $(\Delta v)'_{i+\frac{1}{2}}$ ,  $(\Delta v)'_{i-\frac{1}{2}}$ ,  $(\Delta v)'_{j+\frac{1}{2}}$  y  $(\Delta v)'_{j-\frac{1}{2}}$ , mediante la función minmod:

```
%minmod para x
th=1.5;
for i=1:N+1
   for j=1:N+1
        if
               th*(x3(i,j)-x1(i,j))>0 && 0.5*(x3(i,j)-x2(i,j))>0 && th*(x1(i,j)-x2(i,j))>0
                minmodx1(i,j)=min(min(th*(x3(i,j)-x1(i,j)), 0.5*(x3(i,j)-x2(i,j))), th*(x1(i,j)-x2(i,j)));
        elseif th*(x3(i,j)-x1(i,j))<0 && 0.5*(x3(i,j)-x2(i,j))<0 && th*(x1(i,j)-x2(i,j))<0
                minmodx1(i,j)=max(max(th*(x3(i,j)-x1(i,j)), 0.5*(x3(i,j)-x2(i,j))), th*(x1(i,j)-x2(i,j)));
        else
                minmodx1(i,j)=0;
        end
    end
end
for i=1:N+1
   for j=1:N+1
               th*(x1(i,j)-x2(i,j))>0 && 0.5*(x1(i,j)-x4(i,j))>0 && th*(x2(i,j)-x4(i,j))>0
        if
                minmodx2(i,j)=min(min(th*(x1(i,j)-x2(i,j)), 0.5*(x1(i,j)-x4(i,j))), th*(x2(i,j)-x4(i,j)));
        elseif th*(x1(i,j)-x2(i,j))<0 && 0.5*(x1(i,j)-x4(i,j))<0 && th*(x2(i,j)-x4(i,j))<0
                minmodx2(i,j)=max(max(th*(x1(i,j)-x2(i,j)), 0.5*(x1(i,j)-x4(i,j))), th*(x2(i,j)-x4(i,j)));
        else
                minmodx2(i,j)=0;
        end
    end
end
%minmod para y
for i=1:N+1
   for j=1:N+1
               th*(y3(i,j)-y1(i,j))>0 && 0.5*(y3(i,j)-y2(i,j))>0 && th*(y1(i,j)-y2(i,j))>0
        if
                minmody1(i,j)=min(min(th*(y3(i,j)-y1(i,j)), 0.5*(y3(i,j)-y2(i,j))), th*(y1(i,j)-y2(i,j)));
        elseif th*(y3(i,j)-y1(i,j))<0 && 0.5*(y3(i,j)-y2(i,j))<0 && th*(y1(i,j)-y2(i,j))<0
                minmody1(i,j)=max(max(th*(y3(i,j)-y1(i,j)), 0.5*(y3(i,j)-y2(i,j))), th*(y1(i,j)-y2(i,j)));
        else
                minmody1(i,j)=0;
        end
    end
end
for i=1:N+1
   for j=1:N+1
        if
               th*(y1(i,j)-y2(i,j))>0 && 0.5*(y1(i,j)-y4(i,j))>0 && th*(y2(i,j)-y4(i,j))>0
                minmody2(i,j)=min(min(th*(y1(i,j)-y2(i,j)), 0.5*(y1(i,j)-y4(i,j))), th*(y2(i,j)-y4(i,j)));
        elseif th*(y1(i,j)-y2(i,j))<0 && 0.5*(y1(i,j)-y4(i,j))<0 && th*(y2(i,j)-y4(i,j))<0
                minmody2(i,j)=max(max(th*(y1(i,j)-y2(i,j)), 0.5*(y1(i,j)-y4(i,j))), th*(y2(i,j)-y4(i,j)));
        else
                minmody2(i,j)=0;
        end
    end
end
```

• Polinomio interpolante:

vx1=(x1./dx)-(minmodx1./(2\*dx)); %vx1+ vx2=(x2./dx)+(minmodx2./(2\*dx)); %vx2vy1=(y1./dy)-(minmody1./(2\*dy)); %vy1+ vy2=(y2./dy)+(minmody2./(2\*dy)); %vy2-

#### A.1.2 Velocidades de propagación y Hamiltonianos

• El presente código estima las derivadas  $H_u ext{ y } H_w$  (ecuación (19)), correspondiente al Hamiltoniano de la ecuación (19), posteriormente se obtienen las velocidades de propagación del esquema Godunov  $a_{i,j}^+$ ,  $a_{i,j}^-$ ,  $b_{i,j}^+$  y  $b_{i,j}^-$ , también los de Lax-Friedrich  $\alpha^x ext{ y } \alpha^y$ . Finalmente se obtiene los Hamiltonianos necesarios para el flujo numérico.

```
for i=1:N+1
   for j=1:N+1
        Hx1(i,j)=(U(i,j)*vx1(i,j))/(sqrt(vx1(i,j)^2 + vy1(i,j)^2));
                                                                           %px+,py+
        Hx2(i,j)=(U(i,j)*vx1(i,j))/(sqrt(vx1(i,j)<sup>2</sup> + vy2(i,j)<sup>2</sup>));
                                                                           %px+,py-
        Hx3(i,j)=(U(i,j)*vx2(i,j))/(sqrt(vx2(i,j)<sup>2</sup> + vy1(i,j)<sup>2</sup>));
                                                                           %px-,py+
        Hx4(i,j)=(U(i,j)*vx2(i,j))/(sqrt(vx2(i,j)^2 + vy2(i,j)^2));
                                                                           %px-,py-
        Hy1(i,j)=(U(i,j)*vy1(i,j))/(sqrt(vx1(i,j)^2 + vy1(i,j)^2));
                                                                            %px+,py+
        Hy2(i,j)=(U(i,j)*vy2(i,j))/(sqrt(vx1(i,j)^2 + vy2(i,j)^2));
                                                                           %px+,py-
        Hy3(i,j)=(U(i,j)*vy1(i,j))/(sqrt(vx2(i,j)<sup>2</sup> + vy1(i,j)<sup>2</sup>));
                                                                           %px-,py+
        Hy4(i,j)=(U(i,j)*vy2(i,j))/(sqrt(vx2(i,j)<sup>2</sup> + vy2(i,j)<sup>2</sup>));
                                                                           %px-,py-
                 a1(i,j)=max(max(Mx1(i,j),Hx2(i,j)), max(Hx3(i,j),Hx4(i,j))), 0);
                 a2(i,j)=min(min(Hx1(i,j),Hx2(i,j)), min(Hx3(i,j),Hx4(i,j))), 0);
                 b1(i,j)=max(max(max(Hy1(i,j),Hy2(i,j)), max(Hy3(i,j),Hy4(i,j))), 0);
                 b2(i,j)=min(min(My1(i,j),Hy2(i,j)), min(Hy3(i,j),Hy4(i,j))), 0);
                      alphax(i,j)=max(a1(i,j),a2(i,j));
                      alphay(i,j)=max(b1(i,j),b2(i,j));
```

%calculo hamiltoniano esquema semidiscreto y Lax-friedrichs flux

H1(i,j)=U(i,j)\*sqrt((vx1(i,j)^2+vy1(i,j)^2)); %px+,py+ H2(i,j)=U(i,j)\*sqrt((vx1(i,j)^2+vy2(i,j)^2)); %px+,py-H3(i,j)=U(i,j)\*sqrt((vx2(i,j)^2+vy1(i,j)^2)); %px-,py+ H4(i,j)=U(i,j)\*sqrt((vx2(i,j)^2+vy2(i,j)^2)); %px-,py-

end end

#### A.1.3 Flujos numéricos

• Se implementa los flujos numéricos propuestos, bajo las condiciones de uso expresadas anteriormente. Se calcula el intervalo de tiempo a utilizar. Finalmente evoluciona v en el tiempo  $t^{n+1}$ , se reinicializa y actualiza v:

```
%flujo numerico
for i=1:N+1
   for j=1:N+1
            P1(i,j)= a1(i,j)-a2(i,j);
            Q1(i,j)= b1(i,j)-b2(i,j);
            if P1(i,j)==0 || Q1(i,j)==0 % cualquiera de las dos se puede cumplir, desarrolla
                %Lax-friedrichs flux
               Fv(i,j)=(1/4)*(H4(i,j)+H2(i,j)+H3(i,j)*H1(i,j)) - (1/2)*alphax(i,j)*(vx1(i,j)-vx2(i,j))
                - (1/2)*alphay(i,j)*(vy1(i,j)-vy2(i,j));
            else
                O(i,j)= a2(i,j)*b2(i,j)*H1(i,j) - a2(i,j)*b1(i,j)*H2(i,j) - a1(i,j)*b2(i,j)*H3(i,j)
                  + a1(i,j)*b1(i,j)*H4(i,j);
                P(i,j)= (a1(i,j)*a2(i,j))*(vx1(i,j)-vx2(i,j));
                Q(i,j)= (b1(i,j)*b2(i,j))*(vy1(i,j)-vy2(i,j));
                %flujo numerico upwind
                Fv(i,j)= -O(i,j)/(P1(i,j)*Q1(i,j)) - (P(i,j)/P1(i,j)) - (Q(i,j)/Q1(i,j));
            end
    end
end
```

 $dt(a) = \min(0.475 * dx/max(max(max(max(abs(a1))), max(max(abs(a2)))), max(max(abs(b1))), max(max(abs(b2))))), 0.25 * (dx^2/D));$ 

```
if sum(dt)>=t
    dt(a)=t-sum(dt(1:end-1));
end
```

end

```
v(:,:,k+1)=v(:,:,k)+dt(a).*Fv;
v(:,:,k)=v(:,:,k+1);
v(:,:,k+1)=Reint_LevelSet(N,dx,dy,v(:,:,k+1));
```

#### A.1.4 Función de reinicialización

• Se presenta el código de la función de reinicialización de v con el nuevo Hamiltoniano y sus derivadas, correspondientes a las ecuaciones (31) y (32):

```
function [ phiR ] = Reint_LevelSet( N,dx,dy,v)
k=1;
for i=1:N
               %i+1/2
   for j=1:N+1 %nodo frontera
        x1(i,j)=v(i+1,j,k)-v(i,j,k);
                                         %i+1/2
        x1(N+1,j)=v(N+1,j,k)-v(N,j,k); %nodo frontera
    end
end
               %i-1/2
for i=2:N+1
    for j=1:N+1
        x2(i,j)=v(i,j,k)-v(i-1,j,k);
                                          %i-1/2
        x2(1,j)=v(2,j,k)-v(1,j,k);
    end
end
for i=1:N-1
               %i+3/2
   for j=1:N+1
        x3(i,j)=v(i+2,j,k)-v(i+1,j,k);
                                          %i+3/2
        x3(N,j)=v(N+1,j,k)-v(N,j,k);
        x3(N+1,j)=v(N+1,j,k)-v(N,j,k);
    end
end
for i=3:N+1
               %i-3/2
   for j=1:N+1
        x4(i,j)=v(i-1,j,k)-v(i-2,j,k);
                                           %i-3/2
        x4(2,j)=v(2,j,k)-v(1,j,k);
        x4(1,j)=v(2,j,k)-v(1,j,k);
    end
end
for i=1:N+1
               %j+1/2
   for j=1:N
        y1(i,j)=v(i,j+1,k)-v(i,j,k);
                                           %j+1/2
        y1(i,N+1)=v(i,N+1,k)-v(i,N,k);
    end
end
for i=1:N+1
               %j-1/2
    for j=2:N+1
        y2(i,j)=v(i,j,k)-v(i,j-1,k);
                                           %j-1/2
        y2(i,1)=v(i,2,k)-v(i,1,k);
    end
end
for i=1:N+1 %j+3/2
   for j=1:N-1
        y3(i,j)=v(i,j+2,k)-v(i,j+1,k);
                                           %j+3/2
        y3(i,N)=v(i,N+1,k)-v(i,N,k);
        y3(i,N+1)=v(i,N+1,k)-v(i,N,k);
    end
end
for i=1:N+1
               %j-3/2
    for j=3:N+1
        y4(i,j)=v(i,j-1,k)-v(i,j-2,k);
                                           %j-3/2
        y4(i,2)=v(i,2,k)-v(i,1,k);
        y4(i,1)=v(i,2,k)-v(i,1,k);
    end
end
```

```
%minmod para x
```

```
th=1.5;
for i=1:N+1
    for j=1:N+1
        if
                th*(x3(i,j)-x1(i,j))>0 && 0.5*(x3(i,j)-x2(i,j))>0 && th*(x1(i,j)-x2(i,j))>0
                 minmodx1(i,j)=min(min(th*(x3(i,j)-x1(i,j)), 0.5*(x3(i,j)-x2(i,j))), th*(x1(i,j)-x2(i,j)));
        elseif th*(x3(i,j)-x1(i,j))<0 && 0.5*(x3(i,j)-x2(i,j))<0 && th*(x1(i,j)-x2(i,j))<0
                 minmodx1(i,j)=max(max(th*(x3(i,j)-x1(i,j)), 0.5*(x3(i,j)-x2(i,j))), th*(x1(i,j)-x2(i,j)));
        else
                minmodx1(i,j)=0;
        end
    end
end
for i=1:N+1
    for j=1:N+1
        if
                th*(x1(i,j)-x2(i,j))>0 && 0.5*(x1(i,j)-x4(i,j))>0 && th*(x2(i,j)-x4(i,j))>0
                 minmodx2(i,j)=min(min(th*(x1(i,j)-x2(i,j)), 0.5*(x1(i,j)-x4(i,j))), th*(x2(i,j)-x4(i,j)));
        elseif th*(x1(i,j)-x2(i,j))<0 && 0.5*(x1(i,j)-x4(i,j))<0 && th*(x2(i,j)-x4(i,j))<0
                minmodx2(i,j)=max(max(th*(x1(i,j)-x2(i,j)), 0.5*(x1(i,j)-x4(i,j))), th*(x2(i,j)-x4(i,j)));
        else
                minmodx2(i,j)=0;
        end
    end
end
%minmod para y
for i=1:N+1
    for j=1:N+1
                th*(y3(i,j)-y1(i,j))>0 && 0.5*(y3(i,j)-y2(i,j))>0 && th*(y1(i,j)-y2(i,j))>0
        if
                minmody1(i,j)=min(min(th*(y3(i,j)-y1(i,j)), 0.5*(y3(i,j)-y2(i,j))), th*(y1(i,j)-y2(i,j)));
        elseif th*(y3(i,j)-y1(i,j))<0 && 0.5*(y3(i,j)-y2(i,j))<0 && th*(y1(i,j)-y2(i,j))<0
                minmody1(i,j)=max(max(th*(y3(i,j)-y1(i,j)), 0.5*(y3(i,j)-y2(i,j))), th*(y1(i,j)-y2(i,j)));
        else
                minmody1(i,j)=0;
        end
    end
end
for i=1:N+1
    for j=1:N+1
        if
                th*(y1(i,j)-y2(i,j))>0 && 0.5*(y1(i,j)-y4(i,j))>0 && th*(y2(i,j)-y4(i,j))>0
                minmody2(i,j)=min(min(th*(y1(i,j)-y2(i,j)), 0.5*(y1(i,j)-y4(i,j))), th*(y2(i,j)-y4(i,j)));
        elseif th*(y1(i,j)-y2(i,j))<0 && 0.5*(y1(i,j)-y4(i,j))<0 && th*(y2(i,j)-y4(i,j))<0
                 minmody2(i,j)=max(max(th*(y1(i,j)-y2(i,j)), 0.5*(y1(i,j)-y4(i,j))), th*(y2(i,j)-y4(i,j)));
        else
                minmody2(i,j)=0;
        end
    end
end
%calculo de forma vectorial del polinomio interpolante (derivadas de v)
vx1=(x1./dx)-(minmodx1./(2*dx)); %phix1+
vx2=(x2./dx)+(minmodx2./(2*dx)); %phix2-
vy1=(y1./dy)-(minmody1./(2*dy)); %phiy1+
vy2=(y2./dy)+(minmody2./(2*dy)); %phiy2-
%velocidades de propagación
for i=1:N+1
   for j=1:N+1
    %cambiar las velocidades que están en funcion de un nuevo hamiltoniano
     h11(i,j)=sqrt(vx1(i,j)^2+vy1(i,j)^2);
                                               h21(i,j)=(v(i,j)<sup>2</sup> + (vx1(i,j)<sup>2</sup> + vy1(i,j)<sup>2</sup>)*dx<sup>2</sup>)<sup>(3/2)</sup>; %px+,py+
     h12(i,j)=sqrt(vx1(i,j)^2+vy2(i,j)^2);
                                               h22(i,j)=(v(i,j)^2 + (vx1(i,j)^2 + vy2(i,j)^2)*dx^2)^(3/2); %px+,py-
     h13(i,j)=sqrt(vx2(i,j)^2+vy1(i,j)^2);
                                               h23(i,j)=(v(i,j)<sup>2</sup> + (vx2(i,j)<sup>2</sup> + vy1(i,j)<sup>2</sup>)*dx<sup>2</sup>)<sup>(3/2)</sup>; %px-,py+
```

```
40
```

h24(i,j)=(v(i,j)<sup>2</sup> + (vx2(i,j)<sup>2</sup> + vy2(i,j)<sup>2</sup>)\*dx<sup>2</sup>)<sup>(3/2)</sup>; %px-,py-

h14(i,j)=sqrt(vx2(i,j)^2+vy2(i,j)^2);

Hx1(i,j)= (v(i,j)\*((v(i,j)^2)\*vx1(i,j) + vx1(i,j)\*(dx^2)\*h11(i,j)))/(h11(i,j)\*h21(i,j)); %px+,py+ Hx2(i,j)= (v(i,j)\*((v(i,j)^2)\*vx1(i,j) + vx1(i,j)\*(dx^2)\*h12(i,j)))/(h12(i,j)\*h22(i,j)); %px+,py-Hx3(i,j)= (v(i,j)\*((v(i,j)^2)\*vx2(i,j) + vx2(i,j)\*(dx^2)\*h13(i,j)))/(h13(i,j)\*h23(i,j)); %px-,py+ Hx4(i,j)= (v(i,j)\*((v(i,j)^2)\*vx2(i,j) + vx2(i,j)\*(dx^2)\*h14(i,j)))/(h14(i,j)\*h24(i,j)); %px-,py-Hy1(i,j)= (v(i,j)\*((v(i,j)^2)\*vy1(i,j) + vy1(i,j)\*(dx^2)\*h11(i,j))/(h11(i,j)\*h21(i,j)); %px+,py+ Hy2(i,j)= (v(i,j)\*((v(i,j)^2)\*vy2(i,j) + vy2(i,j)\*(dx^2)\*h12(i,j)))/(h12(i,j)\*h22(i,j)); %px+,py-Hy3(i,j)= (v(i,j)\*((v(i,j)^2)\*vy1(i,j) + vy1(i,j)\*(dx<sup>2</sup>)\*h13(i,j)))/(h13(i,j)\*h23(i,j)); %px-,py+ a1(i,j)=max(max(Mx1(i,j),Hx2(i,j)), max(Hx3(i,j),Hx4(i,j))), 0); a2(i,j)=min(min(Hx1(i,j),Hx2(i,j)), min(Hx3(i,j),Hx4(i,j))), 0); b1(i,j)=max(max(Hy1(i,j),Hy2(i,j)), max(Hy3(i,j),Hy4(i,j))), 0); b2(i,j)=min(min(min(Hy1(i,j),Hy2(i,j)), min(Hy3(i,j),Hy4(i,j))), 0); alphax(i,j)=max(a1(i,j),a2(i,j)); alphay(i,j)=max(b1(i,j),b2(i,j)); %calculo hamiltoniano esquema semidiscreto y Lax-friedrichs flux

> H1(i,j)=(v(i,j)\*(h11(i,j)-1))/sqrt(v(i,j)<sup>2</sup> + (vx1(i,j)<sup>2</sup> + vy1(i,j)<sup>2</sup>)\*dx<sup>2</sup>); %px+,py+ H2(i,j)=(v(i,j)\*(h12(i,j)-1))/sqrt(v(i,j)<sup>2</sup> + (vx1(i,j)<sup>2</sup> + vy2(i,j)<sup>2</sup>)\*dx<sup>2</sup>); %px+,py-H3(i,j)=(v(i,j)\*(h13(i,j)-1))/sqrt(v(i,j)<sup>2</sup> + (vx2(i,j)<sup>2</sup> + vy1(i,j)<sup>2</sup>)\*dx<sup>2</sup>); %px-,py+ H4(i,j)=(v(i,j)\*(h14(i,j)-1))/sqrt(v(i,j)<sup>2</sup> + (vx2(i,j)<sup>2</sup> + vy2(i,j)<sup>2</sup>)\*dx<sup>2</sup>); %px-,py-

end

```
end
```

#### %flujo numerico

dt=0.475\*dx/max(max(max(max(abs(a1))),max(max(abs(a2)))),max(max(max(abs(b1))),max(max(abs(b2)))));

for i=1:N+1 for j=1:N+1

P1(i,j)= a1(i,j)-a2(i,j); Q1(i,j)= b1(i,j)-b2(i,j);

if  $P1(i,j)==0 \mid\mid Q1(i,j)==0$  % cualquiera de las dos se puede cumplir, desarrolla

%Lax-friedrichs flux

Fv(i,j)=(1/4)\*(H4(i,j)+H2(i,j)+H3(i,j)\*H1(i,j)) - (1/2)\*alphax(i,j)\*(vx1(i,j)-vx2(i,j))
- (1/2)\*alphay(i,j)\*(vy1(i,j)-vy2(i,j));

else

end

```
O(i,j)= a2(i,j)*b2(i,j)*H1(i,j) - a2(i,j)*b1(i,j)*H2(i,j) - a1(i,j)*b2(i,j)*H3(i,j)
+ a1(i,j)*b1(i,j)*H4(i,j);
```

P(i,j)= (a1(i,j)\*a2(i,j))\*(vx1(i,j)-vx2(i,j));

Q(i,j)= (b1(i,j)\*b2(i,j))\*(vy1(i,j)-vy2(i,j));

#### %flujo numerico upwind

end

end

#### A.2 Códigos computacionales ecuación de difusión

Se presenta la implementación computacional de la ecuación de difusión, que representa la concentración y difusión de iones de cobre en suspensión:

#### A.2.1 Caracterización de las zonas de cobre depositado

• El presente código caracteriza las regiones del dominio en donde hay cobre en suspensión y cobre depositado, mediante la variable  $\chi$  (Ecuación (37)). Se utiliza extrapolación lineal para los nodos fronterizos.

```
for i=1:N
            for j=1:N+1
                if -v(i,j)*v(i+1,j)>0
                   X1(i,j)=1; %cobre suelto
                else
                   X1(i,j)=0; %cobre depositado/congelado
                end
            end
        end
        for j=1:N+1
                if -v(N+1,j)*(2*v(N+1,j)-v(N,j))>0
                   X1(N+1,j)=1; %cobre suelto
                else
                   X1(N+1,j)=0; %cobre depositado/congelado
                end
        end
%----
        for i=2:N+1
            for j=1:N+1
                if -v(i-1,j)*v(i,j)>0
                   X2(i,j)=1; %cobre suelto
                else
                   X2(i,j)=0; %cobre depositado/congelado
                end
            end
        end
        for j=1:N+1
                if -(2*v(1,j)-v(2,j))*v(1,j)>0
                   X2(1,j)=1; %cobre suelto
                else
                   X2(1,j)=0; %cobre depositado/congelado
                end
        end
%----
        for i=1:N+1
            for j=1:N
                if -v(i,j)*v(i,j+1)>0
                   X3(i,j)=1; %cobre suelto
                else
                   X3(i,j)=0; %cobre depositado/congelado
                end
            end
        end
        for i=1:N+1
                if -v(i,N+1)*(2*v(i,N+1)-v(i,N))>0
                   X3(i,N+1)=1; %cobre suelto
```

```
else
                   X3(i,N+1)=0; %cobre depositado/congelado
                 end
        end
%----
        for i=1:N+1
            for j=2:N+1
                if -v(i,j-1)*v(i,j)>0
    X4(i,j)=1; %cobre suelto
                 else
                   X4(i,j)=0; %cobre depositado/congelado
                 end
            end
        end
        for i=1:N+1
                if -(2*v(i,1)-v(j,2))*v(i,1)>0
                   X4(i,1)=1; %cobre suelto
                else
                   X4(i,1)=0; %cobre depositado/congelado
                 end
        end
```

#### A.2.2 Flujo de deposición

• Se implementa el código del flujo de deposición en base a la variable  $\chi$ , parámetros y la concentración de cobre en cada punto del dominio. Descrito en la ecuación (38). Se utiliza extrapolación lineal para los nodos fronterizos.

```
for i=1:N
                        for j=1:N+1
                            u1(i,j)=min(u(i,j),u(i+1,j));
                                                                                %u(i+(1/2))
                            u1(N+1,j)=min(u(N+1,j),(2*u(N+1,j)-u(N,j)));
                        end
                    end
                    for i=2:N+1
                        for j=1:N+1
                                                                               %u(i-(1/2))
                             u2(i,j)=min(u(i-1,j),u(i,j));
                             u2(1,j)=min((2*u(1,j)-u(2,j)),u(1,j));
                        end
                    end
                    for i=1:N+1
                        for j=1:N
                                                                               %u(j+(1/2))
                             u3(i,j)=min(u(i,j),u(i,j+1));
                             u3(i,N+1)=min(u(i,N+1),(2*u(i,N+1)-u(i,N)));
                        end
                    end
                    for i=1:N+1
                        for j=2:N+1
                              u4(i,j)=min(u(i,j-1),u(i,j));
                                                                               %u(j-(1/2))
                             u4(i,1)=min((2*u(i,1)-u(i,2)),u(i,1));
                        end
                    end
for i=1:N+1
   for j=1:N+1
             R(i,j)=sign(phi(i,j))*(X1(i,j)*(u1(i,j)^r) + X2(i,j)*(u2(i,j)^r) + X3(i,j)*(u3(i,j)^r)
                        + X4(i,j)*(u4(i,j)^r))*(F/D);
    end
end
```

#### A.2.3 Flujo numérico ecuación de difusión

• Se implementa el flujo numérico con respecto a la concentración de cobre en cada punto del dominio, el flujo de deposición y los coeficientes descritos en las ecuaciones (35) y (36). Estos coeficientes tomarán el valor de D si  $v_{i,j}$  o  $v_{i+1,j}$  son positivo, caso contrario tomarán el valor cero. Se utiliza extrapolación lineal para los nodos fronterizos. El valor de c se hace evolucionar en el tiempo  $t^{n+1}$  y se actualiza este valor:

```
for i=1:N
     for j=1:N+1
         if min(v(i+1,j),v(i,j))>0 %d(i+0.5,j)
            d1(i,j)=D;
         else
             d1(i,j)=0;
         end
      end
   end
         if min((2*v(N+1,j)-v(N,j)),v(N+1,j))>0
             d1(N+1,j)=D;
         else
             d1(N+1,j)=0;
         end
   for i=2:N+1
       for j=1:N+1
         if min(v(i-1,j),v(i,j))>0
                                      %d(i-0.5,j)
             d2(i,j)=D;
         else
             d2(i,j)=0;
         end
       end
   end
         if min((2*v(1,j)-v(2,j)),v(1,j))>0
             d2(1,j)=D;
         else
             d2(1,j)=0;
         end
   for i=1:N+1
       for j=1:N
         if min(v(i,j+1),v(i,j))>0
                                      %d(i,j+0.5)
             d3(i,j)=D;
         else
             d3(i,j)=0;
         end
       end
   end
         if min((2*v(i,N+1)-v(i,N)),v(i,N+1))>0
             d3(i,N+1)=D;
         else
             d3(i,N+1)=0;
         end
   for i=1:N+1
       for j=2:N+1
         if min(v(i,j-1),v(i,j))>0
                                      %d(i,j-0.5)
             d4(i,j)=D;
         else
             d4(i,j)=0;
         end
```

```
end
             end
                                if min((2*v(i,1)-v(i,2)),v(i,1))>0 %d(i,j-0.5)
                                             d4(i,1)=D;
                                else
                                            d4(i,1)=0;
                                end
%-----
% FLUJO NUMERICO SEMIDISCRETO 2D
            for i=1:N
                         for j=1:N+1
                                                du1(i,j)=u(i+1,j)-u(i,j);
                                                du1(N+1,j)=(2*u(N+1,j)-u(N,j))-u(N+1,j);
                          end
             end
             for i=2:N+1
                         for j=1:N+1
                                               du2(i,j)=u(i-1,j)-u(i,j);
                                                du2(1,j)=(2*u(1,j)-u(2,j))-u(1,j); %=u(1,j)-u(2,j)
                          end
             end
            for i=1:N+1
                          for j=1:N
                                                du3(i,j)=u(i,j+1)-u(i,j);
                                                du3(i,N+1)=(2*u(i,N+1)-u(i,N))-u(i,N+1);
                          end
             end
             for i=1:N+1
                         for j=2:N+1
                                                du4(i,j)=u(i,j-1)-u(i,j);
                                                du4(i,1)=(2*u(i,1)-u(i,2))-u(i,1);
                          end
             end
for i=1:N+1
             for j=1:N+1
                               Fu(i,j)=d1(i,j)*du1(i,j) + d2(i,j)*du2(i,j) + d3(i,j)*du3(i,j) + d4(i,j)*du4(i,j);
             end
end
for i=2:N
             for j=2:N
                          u(i,j,k+1)=u(i,j,k) + dt(a).*(Fu(i,j)/(dx^2) + R(i,j)/dx);
             end
end
u(1,:,k+1)=u(2,:,k+1);u(:,1,k+1)=u(:,2,k+1); u(:,N+1,k+1)=u0;u(N+1,:,k+1)=u(N,:,k+1); u(1,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1); u(1,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1,k+1)=u(N,1,1
u(:,:,k)=u(:,:,k+1);
```