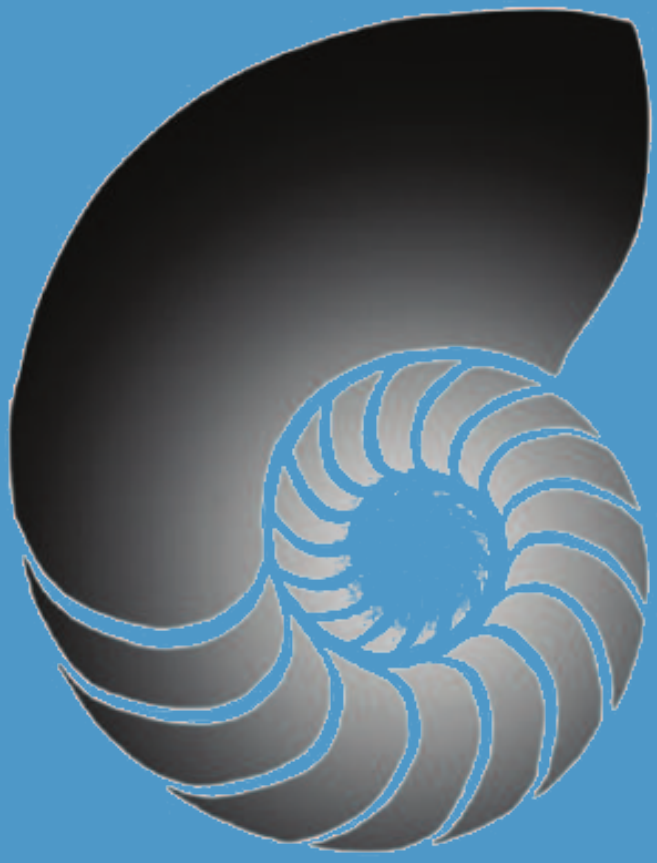
AÑO VII - NÚMERO 7
DICIEMBRE 2021

SMCCA



BOLETÍN

SOCIEDAD MEXICANA DE COMPUTACIÓN
CIENTÍFICA Y SUS APLICACIONES

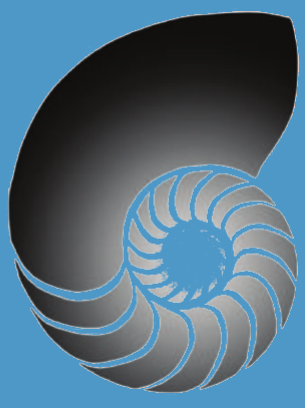
BOLETÍN

SOCIEDAD MEXICANA DE COMPUTACIÓN CIENTÍFICA
Y SUS APLICACIONES

AÑO VII - NÚMERO 7

DICIEMBRE 2021

SMCCA



Comité Editorial

Pablo Barrera Sánchez,	UNAM
Pedro Flores Pérez,	UNISON
Gerardo Tinoco Guerrero,	UMSNH, UVAQ (editor responsable)
José Gerardo Tinoco Ruíz,	UMSNH
Justino Alavez Ramirez,	UJAT

Editores Técnicos

José Alberto Guzmán Torres,	UMSNH

El boletín de la Sociedad Mexicana de Computación Científica y sus Aplicaciones publica artículos de investigación originales y de alta calidad en las áreas de matemáticas aplicadas y computación científica, así como artículos de difusión científica. Todos los artículos son sometidos a una revisión por expertos en estas áreas de instituciones nacionales e internacionales.

El Boletín de la Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C. (SMCCA), Año VII, No. 7, Diciembre 2021, es una publicación oficial anual editada por la Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C., calle Luis Horacio Salinas, 545, Col. Valle de Morelos, Saltillo, Coahuila, C.P. 25013, Tel. (844) 410 1242, www.smcca.org.mx.

Editor responsable: Gerardo Tinoco Guerrero. Reserva de Derechos al Uso Exclusivo No. 04 - 2017 - 103114330600 - 203, ISSN: 2594-0457, ambos otorgados por el Instituto Nacional de Derechos de Autor. Responsable de la última actualización de este Número, Gerardo Tinoco Guerrero, Avenida Francisco J. Mújica S/N, Ciudad Universitaria, Edificio B, Morelia, Michoacán, C.P. 58030, fecha de la última modificación, 27 de diciembre de 2021.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización de la Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C.

Suscripciones al Boletín
smcca@smcca.org.mx
https://www.scipedia.com/sj/smcca

CONTENIDO

Carta de bienvenida 1
Reseña de la XXIX Escuela Nacional de Optimización y Análisis Numérico 2

Artículos

Método de Newton para búsquedas en Línea en el espacio de Hilbert (L^2)^3 9
Solución de la ecuación de Poisson en 2D usando algoritmos en paralelo en entornos multicore 18
La exclusión social de las personas mayores: una forma de mirar la pobreza 32
Análisis estocástico del flujo vehicular para la red de ejes viales de la Ciudad de México mediante cadenas de Markov discretas.....46
El potencial del aprendizaje hebbiano en a clasificación supervisada.....57

Información

¿Quieres publicar artículos, información sobre eventos o noticias en el boletín?.....72
Sociedad Mexicana de Computación Científica y sus Aplicaciones.....73

Carta de bienvenida

La Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C. (SMCCA) y el Comité Editorial, les damos una cordial bienvenida a la séptima edición del Boletín electrónico anual de la SMCCA, el cual tiene como objetivo mantenerlos informados de las actividades realizadas por la SMCCA y sus asociados. En el Boletín se publican noticias, eventos, artículos de divulgación y de investigación de alto nivel en las áreas de Matemáticas Aplicadas y Cómputo Científico, así como resúmenes de las mejores tesis de Licenciatura en Matemáticas Aplicadas.

En esta séptima edición del boletín se presentan: una breve semblanza de la XXIX Escuela Nacional de Optimización y Análisis Numérico, llevada a cabo en modalidad virtual del 2 al 4 de agosto del presente año en el Departamento de Matemáticas de la Facultad de Ciencias de la UNAM, Ciudad Universitaria, Cd. de México, e incluye los ganadores de la décima novena edición del premio Mixbaal para la mejor tesis de Licenciatura en Matemáticas Aplicadas, y cinco artículos de investigación, tres de ellos por solicitud y los otros dos corresponden a los ganadores del premio Mixbaal.

La SMCCA agradecerá que ante el interés que surja en los lectores en los temas que se presenten en nuestra publicación, éstos se conviertan en usuarios asiduos, así como en miembros activos de nuestra Sociedad. La información del registro de membresías a la SMCCA la pueden consultar en el Módulo de Registro de nuestra página www.smcca.org.mx.

Justino Alavez Ramírez
Presidente
Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C.

Reseña de la XXIX Escuela Nacional de Optimización y Análisis Numérico

Justino Alavez Ramírez
Rina Betzabeth Ojeda Castañeda
María Lourdes Velasco Arregui

Del 2 al 4 de agosto de 2021 se celebró con gran éxito la XXIX edición de la Escuela Nacional de Optimización y Análisis Numérico (ENOAN 2021). En esta ocasión y debido a las restricciones impuestas por la pandemia se llevó a cabo en modalidad virtual, siendo la sede el Departamento de Matemáticas de la Facultad de Ciencias, de la Universidad Nacional Autónoma de México, en la Ciudad Universitaria, de la Cd. de México.

El comité organizador estuvo constituido por un Comité Local formado por profesores-investigadores del Departamento de Matemáticas y un Comité Nacional conformado por profesores-investigadores miembros en activo de la Sociedad Mexicana de Computación Científica y sus Aplicaciones (SMCCA).

Como en anteriores ediciones, las actividades académicas realizadas en esta ENOAN 2021 comprendieron: cursos cortos (6), conferencias plenarias (5) y conferencias invitadas a cargo de reconocidos investigadores tanto nacionales (8) como internacionales (3), la presentación de trabajos en forma de cartel (11), la presentación de trabajos en forma de comunicación oral (25) agrupados en 5 sesiones temáticas, y adicionalmente 2 sesiones especiales, el taller "Sección MEX-SIAM", y la sección "Homenaje al Dr. David Romero". Al ser un evento totalmente virtual, todas estas actividades se llevaron a cabo a través de aulas virtuales de Zoom, así como los canales de Facebook de la Facultad de Ciencias y del Departamento de Matemáticas de la UNAM, y el Facebook de la ENOAN de la SMCCA.

En cuanto a los cursos cortos, estos fueron 6, de diferentes niveles: básico, intermedio y avanzado, y fueron impartidos por profesores-investigadores de diferentes Instituciones de Educación Superior y Centros de Investigación del país. Se ofrecieron de forma gratuita a alumnos de licenciatura y posgrado, así como a profesores y profesionales interesados de distintas instituciones educativas del país y del extranjero, y con tópicos específicos, los cuales generalmente no son abordados con regularidad en sus instituciones. El nombre de los cursos impartidos, así como la asistencia promedio en cada uno de ellos por Zoom se presenta en la tabla 1.

Los contenidos de los cursos, bibliografía e información más detallada de los instructores se puede consultar en la página de la ENOAN en la liga www.smcca.org.mx/enoan2021.

En cuanto a las conferencias plenarias (5), la conferencia inaugural "Diego Bricio", estuvo a cargo del reconocido Profesor-Investigador, el Dr. Pedro Miramontes Vidal, de la Facultad de Ciencias de la Universidad Nacional Autónoma de México, la conferencia "Cátedra Humberto Madrid", que se estableció por primera vez en esta edición de la ENOAN, estuvo a cargo del reconocido Profesor-Investigador, el Dr. Pablo Barrera Sánchez, de la Facultad de Ciencias de la Universidad Nacional Autónoma de México y las tres conferencias restantes estuvieron a cargo de los reconocidos profesores investigadores: el Dr. Gerardo Chowell-Puente, de la School of Public Health, Georgia State University, el Dr. Damien Pham Van Bang, del Institut National de la Recherche Scientifique, Quebec, Canadá y la Dra. Graciela del Socorro Herrera Zamarrón, del Instituto de Geofísica de la UNAM. El título y resumen de las conferencias, así como la información más detallada de los conferencistas se puede consultar en la página de la ENOAN en la liga www.smcca.org.mx/enoan2021/conferencias.html.

CURSOS NIVEL BÁSICO	Asistencia promedio por Zoom
Introducción al Cómputo Científico Gerardo Tinoco Guerrero. Universidad Michoacana de San Nicolás de Hidalgo. Universidad Vasco de Quiroga.	45
Julia + Linux + USB Pablo Barrera Sánchez, Iván Méndez Cruz y Mario Rafael Nolasco Estrada. Universidad Nacional Autónoma de México.	29
CURSOS NIVEL INTERMEDIO:	**Asistencia Promedio por Zoom**
Simulación y control de tráfico urbano utilizando redes de Petri Joel Antonio Trejo Sánchez. Investigador Cátedras CONACyT. CIMAT Unidad Mérida.	18
Herramientas de productividad para Ciencia de Datos Juan Pablo Soto Barrera y Julio Waissman Vilanova. Universidad de Sonora.	44
CURSOS NIVEL AVANZADO	**Asistencia Promedio por Zoom**
Análisis numérico de estabilidad lineal de sistemas espacio-temporales Josué Manik Nava Sedeño. Universidad Nacional Autónoma de México.	30
Programación en paralelo Miguel Ángel Pérez León. Universidad Nacional Autónoma de México.	41

Tabla 1: Cursos Impartidos en la XXIX ENOAN.

La presentación de trabajos en forma de cartel se realizó a través de la exposición de los 11 carteles sometidos y aprobados en la página de la SMCCA, liga www.smcca.org.mx/enoan2021/carteles y la plataforma de Youtube, https://www.youtube.com/channel/UCzCEINZBl_qSTCF5uARhQ1Q. Actualmente se puede acceder a ellos en el canal de Youtube de la SMCCA, https://www.youtube.com/channel/UC0rIXGJ1Qx7OVURphWTykXQ. La presentación de trabajos en forma de comunicación oral (25) agrupados en 6 sesiones temáticas, se llevó a cabo a través de las aulas virtuales de la plataforma Zoom. Los títulos y resúmenes de los trabajos, así como la información de los ponentes se pueden consultar en la página de la SMCCA, liga www.smcca.org.mx/enoan2021/horarioponencias.html.

En la sesión especial “Sección MEX-SIAM", se expusieron 5 conferencias invitadas, a cargo de miembros de esta asociación: el Dr. Ismael Herrera Revilla, del Instituto de Geofísica de la UNAM, el Dr. Miguel Ángel Uh, del Centro de Investigación en Matemáticas, A.C. CIMAT-Mérida, el Dr. Francisco Javier Domínguez Mota, de la Universidad Michoacana de San Nicolás de Hidalgo y el Dr. Martín Alberto Díaz-Viera, del Instituto Mexicano del Petróleo. Los títulos y resúmenes de las conferencias, así como la información de los

expositores se pueden consultar en la página de la SMCCA, liga www.smcca.org.mx/enoan2021/mexsiam.html.

La sesión especial "Homenaje a David Romero", coordinada por los Dres. Gilberto Calvillo Vives, del Instituto de Matemáticas de la UNAM, Campus Cuernavaca y Jesús López Estrada de la Facultad de Ciencias de la UNAM, contó con la participación de investigadores de distintas Instituciones y Centros de Investigación, quienes trabajaron muy de cerca con el Dr. David Romero, y durante la sesión presentaron remembranzas de su relación académica y personal. El programa de esta sesión se puede consultar en la liga http://smcca.org.mx/enoan2021/img/cartel/CartelDavidRomero_may.pdf.

Ganadores del décimo noveno Premio Mixbaal

Desde 2002, dentro del marco de la ENOAN, se instituyó el "Premio Mixbaal a la Mejor Tesis de Licenciatura en Matemáticas Aplicadas", cuya convocatoria está dirigida a los egresados de carreras de matemáticas y áreas afines. Los ganadores de la décima novena edición de este premio, evaluados en la ENOAN 2021, son los siguientes:

Ganador del Premio Mixbaal

Métodos de ajuste paramétrico e hiperparamétrico de Redes Neuronales con algoritmos biológicamente inspirados.

Fernando Javier Aguilar Canto

Lic. en Matemáticas
Universidad Autónoma de Yucatán
Director: Dr. Carlos Francisco Brito-Loeza

Mención Honorífica

Estudio del tráfico vehicular mediante redes complejas y cadenas de Markov discretas.

Jessica Pereda Méndez

Lic. en Modelación Matemática
Universidad Autónoma de la Ciudad de México
Director: Dr. Luis Agustín Olivares Quiroz

Ambos ganadores expusieron sus trabajos como ponentes invitados.

Concurso de Carteles

Continuando con la tradición de las anteriores ediciones de la ENOAN, se realizó el concurso de carteles, en el cual una vez que los participantes exponen ante el jurado y el público su trabajo, el jurado lleva a cabo una evaluación para decidir la premiación. En esta ocasión, al ser una edición virtual, el procedimiento se modificó como sigue: los participantes enviaron su cartel en formato PDF junto con un video grabado de 10 minutos exponiendo su trabajo. El jurado, integrado por el Dr. Justino Alavez Ramírez (Coordinador), Dr. Francisco Javier Domínguez Mota, Dr. Pedro Flores Pérez, Dr. Marco Arieli Herrera Valdez y Dr. Jaime Vázquez Alamilla, revisó cada uno de los videos y los carteles en formato PDF, y con base en la Convocatoria emitida para tal fin y bajo los Criterios de Evaluación: Calidad Científica, Impacto Visual, Creatividad y Originalidad, los Miembros del Jurado revisaron reservada y libremente la totalidad de los carteles presentados, dictaminando lo siguiente:

Mejor Cartel Nivel Licenciatura

Estudio del COVID-19 en México: Modelos de crecimiento limitado y tasa de infección como función del tiempo.

José Alberto Mendoza González
Universidad Nacional Autónoma de México

Mejor Cartel Nivel Maestría

Parameter estimation of a mathematical model using two distinct breast cancer cell lines under chemotherapy treatment.

María Eliza Antunes
Instituto de Biociência de Botucatu Brasil

Mejor Cartel Nivel Doctorado

Un problema de máxima cobertura para la ubicación de pruebas para la detección de la COVID-19.

Salvador de Jesús Vicencio Medina
School of Engineering and Sciences

Cobertura de las actividades

De acuerdo al registro oficial del Certificado de Inscripción a la ENOAN 2021, se contó con la asistencia de 88 personas, de las cuales un 27% pertenecía al género femenino y un 73% al género masculino. Los intervalos de edades que se establecieron en la cédula de registro fueron de los 18 años a más de 64 años, que eran las edades probables de la audiencia para este tipo de evento (jóvenes, adultos y adultos mayores), y los resultados que se obtuvieron con respecto a la edad se presentan en la tabla 2 y figura 1.

Intervalos de edades	Número de asistentes
18 a 24	18
25 a 34	32
35 a 44	12
45 a 54	13
55 a 64	8
> 64	5
TOTAL	88

Tabla 2: Número de asistentes al evento por intervalos de edades.

Se puede observar en la tabla 2 que el mayor número de personas se dio en el rango de 25 a 54 años (57, 64.8%), siendo el menor número en las edades adultas de 55 a 65 y mayores a 64. Es interesante ver que hay una participación de aproximadamente del 20.5% de jóvenes que desde sus primeros semestres de la licenciatura estén decidiendo participar en la ENOAN.

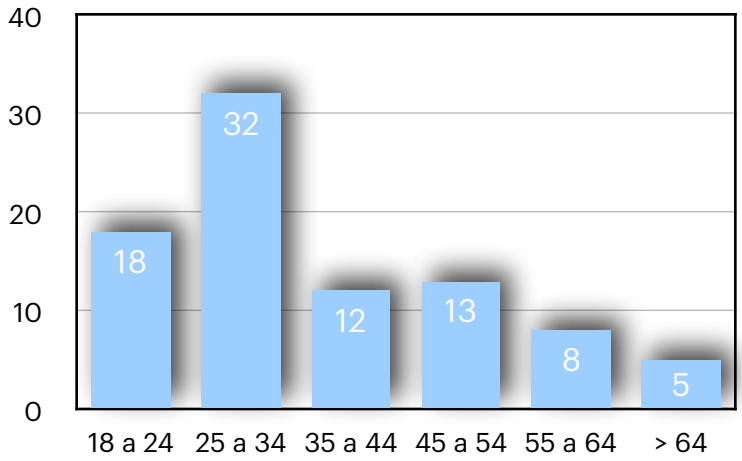


Figura 1: Gráfica de barras de frecuencias por intervalo de edades de los participantes.

Se tuvieron asistentes de diversos países de América (Norte y Sur) y Europa como se muestra en la tabla 3.

País	Número de asistentes	Porcentaje de asistentes
México	78	88.63%
Canadá	1	1.13%
Perú	4	4.54%
Colombia	1	1.13%
Estados Unidos	1	1.13%
Alemania	1	1.13%
Brasil	1	1.13%

Tabla 3: Países de procedencia de los asistentes.

Es importante señalar que gracias a que el evento se realizó en modalidad vía remota, en esta edición de la ENOAN se tuvo asistencia virtual de un alumno de Brasil, uno de Colombia, cuatro de Perú, uno de Alemania y uno de Estados Unidos. Por otro lado, se puede observar que como era de esperarse el mayor número de asistentes tenía nacionalidad mexicana, 78 en total que representa el 88.6% de los participantes.

Consideraciones finales

Por último es de gran importancia señalar que el gran esfuerzo de trabajo realizado tanto por el Comité Local (Departamento de Matemáticas de la Facultad de Ciencias de la UNAM) como el Comité Nacional (aglutinados dentro de la SMCCA) en la organización, y contando con el importante apoyo financiero de

Instituciones, Dependencias y Centros de Investigación como: CONACYT, la Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C., la Facultad de Ciencias de la Universidad Nacional Autónoma de México, el Departamento de Matemáticas de la Universidad de Sonora, y el Centro de Investigación en Matemáticas Aplicadas de la Universidad Autónoma de Coahuila; permitió obtener un conjunto de resultados a beneficio de una comunidad científica conformada por alumnos, profesores, investigadores y profesionales interesados en la Computación Científica y las Matemáticas Aplicadas, que incidieron en indicadores de impacto como los que se presentan en la tabla 4.

Indicador	Cantidad
Total de asistentes registrados:	88
Programas Académicos beneficiados por el evento:	28
Cuerpos Académicos o Grupos de Investigación beneficiados:	12
Cursos cortos impartidos:	6
Conferencias (Diego Bricio, Humberto Madrid, Plenarias e Invitadas):	12
Ponencias por solicitud:	26
Carteles expuestos en smcca.org.mx/enoan2021/carteles/:	11
Estudiantes beneficiados:	47 registrados
Investigadores y Docentes beneficiados:	41 registrados
Número de Instituciones Nacionales participantes:	37
Número de Instituciones Internacionales participantes:	13
Integrantes del Comité Nacional:	13
Integrantes del Comité Local:	4
Personal de Apoyo Sede:	9
Sala virtual para inauguración del evento:	1
Salas virtuales para conferencias plenarias:	5
Salas virtuales para secciones especiales y conferencias invitadas:	4
Salas virtuales para impartición de cursos:	3
Salas virtuales para presentación de trabajos:	3
Sala virtual para la clausura del evento:	1

Tabla 4: Indicadores de impacto.

Finalmente, en la figura 2, se muestran algunos banners de difusión del evento publicados en Facebook, así como capturas de pantalla del curso de Herramientas de productividad para Ciencias de Datos, y la conferencia plenaria Diego Bricio.

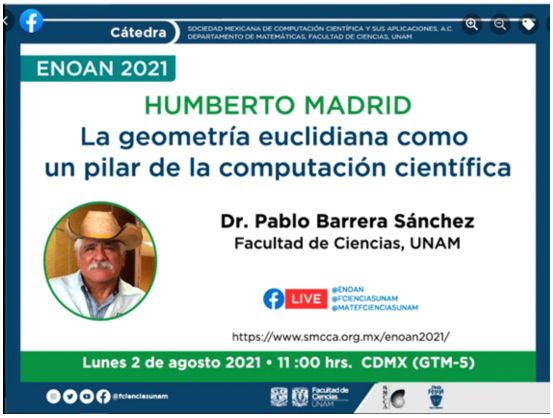








Figura 2: Banners de difusión del evento y capturas de pantalla de curso y conferencia plenaria Diego Bricio.

Método de Newton para búsquedas en Línea en el espacio de Hilbert $(L^2)^3$

Jorge López López
División Académica de Ciencias Básicas
Universidad Juarez Autónoma de Tabasco
jorge.lopez@ujat.mx

Resumen

Este trabajo trata sobre la aplicación del método de Newton para resolver un problema de búsqueda en línea asociado con la minimización de un funcional definido en el espacio de Hilbert $(L^2(0,T))^3$, con T un tiempo final dado. El funcional es parte de un problema de control de un circuito de 3 juntas de Josephson modelado por un sistema de tres ecuaciones diferenciales ordinarias no lineales dependientes del tiempo. El problema de control se puede resolver con el método de gradiente conjugado, dentro del cual se deben resolver problemas de búsqueda en línea como el descrito en este trabajo. Se describe tanto el caso continuo como su versión discreta para lo cual, las funciones en $(L^2(0,T))$ se aproximan por funciones lineales por pedazos y los sistemas diferenciales ordinarios se resuelven con el método de Euler Explícito.

Palabras Clave: Método de Newton; Búsqueda en línea; Espacio de Hilbert; Euler Explícito.

1 Introducción

En optimización, la estrategia de búsqueda en línea es uno de los enfoques iterativos básicos para encontrar un mínimo local v^* de una función objetivo $f : \mathbb{R}^n \to \mathbb{R}$. El enfoque de búsqueda en línea primero supone conocida una aproximación previa u para v^* y encuentra una dirección de descenso w a lo largo de la cual la función objetivo f será minimizada y entonces esto determinará qué tan lejos u debe moverse a lo largo de esa dirección. Este tamaño de paso está asociado con un mínimo local de la restricción de f a la recta de búsqueda. Al problema de encontrar este tamaño de paso se le llama un problema de búsqueda en línea, es decir, es un problema de minimización con restricciones de la forma

$$\begin{cases} Min\ f(v) \\ \text{sujeto a } v = u - \rho w, \forall \rho \in \mathbb{R}; u, w \in \mathbb{R}^n, \end{cases}$$

que también se puede escribir como

$$Min_{\rho \in \mathbb{R}}\ f(u - \rho w). \tag{1}$$

Aquí la recta de búsqueda es la recta que pasa por u y tiene (por conveniencia, para nuestras posteriores aplicaciones) la dirección $-w$. Como ya se dijo, una de las áreas donde surgen problemas de búsqueda en línea son los métodos iterativos para minimizar localmente una función $f : \mathbb{R}^n \to \mathbb{R}$ sin restricciones. En cada iteración i debe resolverse un problema de búsqueda en línea de la forma (1), para u y w conocidos. Si denotamos con ρ_i a la solución del problema de búsqueda en línea de la iteración i, se toma a $v_i = u - \rho_i w$ como la nueva aproximación para el mínimo de f. La solución de los problemas de búsqueda en línea se puede aproximar por una variedad de métodos numéricos [2], entre ellos el método de Newton. Si definimos $g(\rho) = f(u - \rho w)$ entonces la solución del problema (1) es equivalente a la solución del problema en $\mathbb{R}$

$$Min_{\rho \in \mathbb{R}}\ g(\rho).$$

Para resolver este problema buscamos los valores donde la derivada de g se hace cero, es decir, resolvemos (por el método de Newton) la ecuación

$$g'(\rho^*) = 0.$$

Tenemos que

$$g'(\rho) = -Df(u - \rho w; w) = -\nabla f(u - \rho w) \cdot w,$$

donde $Df(v; w)$ denota la derivada direccional de f en v en la dirección de w. Así que por comodidad definimos

$$H(\rho) = -g'(\rho) = Df(u - \rho w; w) = \nabla f(u - \rho w) \cdot w$$

y resolvemos la ecuación

$$H(\rho^*) = 0,$$

para lo cual, dada una aproximación inicial ρ_0, se construyen sucesivas aproximaciones de acuerdo a

$$\rho_{i+1} = \rho_i - \frac{H(\rho_i)}{H'(\rho_i)}, i = 0, 1, 2, \ldots$$

En este trabajo extenderemos estas ideas para aplicar el método de Newton a un problema de búsqueda en línea pero donde los elementos u y w aunque conocidos, pertenecen al espacio de Hilbert $(L^2)^3$ y no a $\mathbb{R}^n$. En este contexto, este artículo es una continuación o extensión de lo expuesto en [6].

2 Descripción del Problema

Los problemas de búsqueda en línea en que estamos interesados aquí son del tipo

$$\begin{cases} \text{Encontrar } \rho^* \in \mathbb{R}, \text{ tal que} \\ J(\mathbf{u} - \rho^* \mathbf{w}) \leq J(\mathbf{u} - \rho \mathbf{w}), \forall \rho \in \mathbb{R}, \end{cases} \tag{2}$$

con $\mathbf{u}$ y $\mathbf{w}$ fijos en $(L^2(0,T))^3$, y donde el funcional de nuestro interés $J : (L^2(0,T))^3 \longrightarrow \mathbb{R}$ es

$$J(\mathbf{v}) = \frac{\eta}{2} \int_0^T \|\mathbf{v}\|^2 dt + \frac{k}{2} ||\mathbf{y}(T) - \mathbf{y}_T||^2, \tag{3}$$

donde $k, \eta > 0$ y $||.||$ la norma euclideana canónica, donde $\mathbf{v}$ es una función vectorial, $\mathbf{v} = (v_1(t), v_2(t), v_3(t))^T$, $\|\mathbf{v}\|^2 = v_1^2 + v_2^2 + v_3^2$ y la función vectorial $\mathbf{y} = (y_1, y_2, y_3)^T$ es la solución del siguiente problema de valor inicial que modela la dinámica de un circuito de tres juntas de Josephson acopladas inductivamente, [1]:

$$\begin{cases} \gamma_1 \frac{dy_1}{dt} + \kappa_1(y_1 - y_2) + \sin y_1 = i_1 + v_1, \text{ en } (0,T), \\ \gamma_2 \frac{dy_2}{dt} + \kappa_1(y_2 - y_1) + \kappa_2(y_2 - y_3) + \sin y_2 = i_2 + v_2, \text{ en } (0,T), \\ \gamma_3 \frac{dy_3}{dt} + \kappa_2(y_3 - y_2) + \sin y_3 = i_3 + v_3, \text{ en } (0,T), \\ \mathbf{y}(0) = \mathbf{y}_0. \end{cases} \tag{4}$$

En (3)-(4), $\mathbf{y}_0$ y $\mathbf{y}_T$ son estados inicial y final conocidos, respectivamente. En [1] y en [7] se dan los siguientes valores factibles para los parámetros involucrados en este sistema y con los cuales se harán los experimentos numéricos mas adelante:

$$\gamma_1 = 0.7, \ \gamma_2 = 1.1, \ \gamma_3 = 0.7, \ i_1 = 1, \ i_2 = 0.8, \ i_3 = -1,$$

2.1. El problema de minimización global y el diferencial de J

El problema de minimización global o sin restricciones asociado con los problemas de búsqueda en línea descritos es

$$\begin{cases} \text{Encontrar } v^* \in (L^2(0,T))^3, \text{tal que} \\ J(v^*) \leq J(v), \forall v \in (L^2(0,T))^3, \end{cases}$$

el cual, (junto con (3)-(4) corresponde a un problema de control cuyo objetivo es llevar la dinámica del sistema del estado $\mathbf{y}_0$ al estado $\mathbf{y}_T$. Este tipo de problemas de control pueden resolverse usando un algoritmo de gradiente conjugado como los discutidos en el Capítulo 2 de [5] y el Capítulo 3 de [4], donde también se menciona el concepto de Frechet-diferenciabilidad adecuado para minimizar funcionales en espacios de Hilbert. A continuación damos una definción y un teorema básicos para resolver el problema que nos ocupa.

Definición 1. *Sea V un espacio de Hilbert. Un funcional J sobre V es Frechet-diferenciable si, para todo $v, w \in V$, existe $DJ(v) \in V'$, la derivada o el diferencial de J en v, tal que*

$$J(v+w) - J(v) = \langle DJ(v), w\rangle + \|w\|\epsilon(v,w),$$

con $\langle .,.\rangle$ denotando par de dualidad, y $\epsilon(v,w)$ tendiendo a cero cuando $\|w\|$ tiende a cero.

Teorema 1. *Si J está definido como en (3) entonces es Frechet-diferenciable y su diferencial $DJ(\mathbf{v})$ es*

$$DJ(\mathbf{v}) = \eta\mathbf{v} + \mathbf{p}, \tag{5}$$

donde $\mathbf{p}$ se obtiene al resolver el sistema (versión matricial de (4)) :

$$\begin{cases} \Gamma\dfrac{d\mathbf{y}}{dt} + K\mathbf{y} + \sin\mathbf{y} = \begin{pmatrix} i_1+v_1 \\ i_2+v_2 \\ i_3+v_3 \end{pmatrix} \ en\ (0,T), \\ \mathbf{y}(0) = \mathbf{y}_0, \end{cases} \tag{6}$$

y después el sistema adjunto

$$\begin{cases} -\Gamma\dfrac{d\mathbf{p}}{dt} + K\mathbf{p} + C(\mathbf{y})\,\mathbf{p} = 0 \ en\ (0,T), \\ \Gamma\mathbf{p}(T) = k(\mathbf{y}(T) - \mathbf{y}_T), \end{cases} \tag{7}$$

donde $C(\mathbf{y})$ es una matriz diagonal de 3x3 para la cual la entrada iésima de la diagonal es $cos(y_i)$.

De este teorema resulta que $DJ(\mathbf{u}-\rho\mathbf{w}) = \eta(\mathbf{u}-\rho\mathbf{w}) + \mathbf{p}$, con $\mathbf{p}$ solución de (7) una vez que $\mathbf{y}(t)$ es solución del sistema (6) tomando $\mathbf{v} = \mathbf{u} - \rho\mathbf{w}$.

La demostración de este teorema, y de los resultados que se mencionan a continuación pueden consultarse en [3].

3 Metodología de solución

Para resolver por el método de Newton nuestro problema de búsqueda en línea (2) definimos

$$g(\rho) = J(\mathbf{u} - \rho\mathbf{w}), \tag{8}$$

entonces el problema se reduce a encontrar raíces para g' (un problema de $\mathbb{R}$ en $\mathbb{R}$), las cuales aproximaremos con la iteración de Newton

$$\rho_{i+1} = \rho_i - \frac{g'(\rho_i)}{g''(\rho_i)}; g''(\rho_i) \neq 0,$$

para lo cual debemos conocer g' y g''. Para describir $g'(\rho)$ en términos de DJ necesitamos el siguiente teorema:

Teorema 2. *. Sea J un funcional Frechet-diferenciable sobre V y g como en (8). Entonces*

$$g'(\rho) = \frac{d}{d\rho}J(\mathbf{u}-\rho\mathbf{w}) = -\langle DJ(\mathbf{u}-\rho\mathbf{w}), \mathbf{w}\rangle.$$

Así que la ecuación que tenemos que resolver para ρ es

$$g'(\rho) = -\langle DJ(\mathbf{u}-\rho\mathbf{w}), \mathbf{w}\rangle = 0.$$

Como la iteración de Newton es

$$\rho_{i+1} = \rho_i - \frac{g'(\rho_i)}{g''(\rho_i)},$$

aún nos falta encontrar una expresión para $g''(\rho)$. Como estamos trabajando en $(L^2(0,T))^3$, los pares de dualidad coinciden con el producto interno gracias al teorema de representación de Riesz. Dado que ya conocemos el representante de la transformación $DJ(v)$ (ver (5)), podemos escribir

$$g'(\rho) = -(\eta(\mathbf{u} - \rho\mathbf{w}) + \mathbf{p}, \mathbf{w}) = -\int_0^T (\eta(\mathbf{u} - \rho\mathbf{w}) + \mathbf{p}) \cdot \mathbf{w} \ \mathrm{dt}, \tag{9}$$

donde $\mathbf{p}$ se obtiene resolviendo

$$\begin{cases} \begin{array}{c} \gamma_1 \frac{dy_{1\rho}}{dt} + \kappa_1(y_{1\rho} - y_{2\rho}) + \sin y_{1\rho} = i_1 + u_1 - \rho w_1 \text{ en } (0,T), \\ \gamma_2 \frac{dy_{2\rho}}{dt} + \kappa_1(y_{2\rho} - y_{1\rho}) + \kappa_2(y_{2\rho} - y_{3\rho}) + \sin y_{2\rho} = i_2 + u_2 - \rho w_2 \text{ en } (0,T), \\ \gamma_3 \frac{dy_{3\rho}}{dt} + \kappa_2(y_{3\rho} - y_{2\rho}) + \sin y_{3\rho} = i_3 + u_3 - \rho w_3 \text{ en } (0,T), \\ \mathbf{y}(0) = \mathbf{y}_0. \end{array} \end{cases}$$

y luego el problema

$$\begin{cases} \begin{array}{c} -\Gamma \frac{d\mathbf{p}_\rho}{dt} + K\mathbf{p}_\rho + \begin{pmatrix} \cos y_{1\rho} & 0 & 0 \\ 0 & \cos y_{2\rho} & 0 \\ 0 & 0 & \cos y_{3\rho} \end{pmatrix} \mathbf{p}_\rho = 0 \text{ en } (0,T), \\ \Gamma \mathbf{p}_\rho(T) = k(\mathbf{y}_\rho(T) - \mathbf{y}_T). \end{array} \end{cases}$$

Para calcular $g''(\rho) = \dfrac{dg'}{d\rho}$, por la regla de Leibniz para la diferenciación bajo el signo integral tenemos, a partir de (9) que:

$$\frac{dg'(\rho)}{d\rho} = -\int_0^T \frac{\partial}{\partial\rho}(\eta(\mathbf{u} - \rho\mathbf{w}) + \mathbf{p}) \cdot \mathbf{w} \ \mathrm{dt} \ ,$$

lo cual nos da

$$g''(\rho) = \int_0^T [\eta\mathbf{w} - \dot{\mathbf{p}}_\rho] \cdot \mathbf{w} \ \mathrm{dt} \ ,$$

que denotaremos como

$$g''(\rho) = \langle D^2 J(\mathbf{u} - \rho\mathbf{w})\mathbf{w}, \mathbf{w}\rangle$$

y donde $\dot{\mathbf{p}}_\rho$ se obtiene resolviendo

$$\begin{cases} \begin{array}{c} \Gamma \frac{d\dot{\mathbf{y}}_\rho}{dt} + K\dot{\mathbf{y}}_\rho + \begin{pmatrix} \cos y_{1\rho} & 0 & 0 \\ 0 & \cos y_{2\rho} & 0 \\ 0 & 0 & \cos y_{3\rho} \end{pmatrix} \dot{\mathbf{y}}_\rho = \begin{pmatrix} -w_1 \\ -w_2 \\ -w_3 \end{pmatrix} \text{ en } (0,T), \\ \dot{\mathbf{y}}_\rho(0) = 0 \end{array} \end{cases}$$

y luego el problema

$$\begin{cases} \begin{array}{c} -\Gamma \frac{d\dot{\mathbf{p}}_\rho}{dt} + K\dot{\mathbf{p}}_\rho + \begin{pmatrix} \cos y_{1\rho} & 0 & 0 \\ 0 & \cos y_{2\rho} & 0 \\ 0 & 0 & \cos y_{3\rho} \end{pmatrix} \dot{\mathbf{p}}_\rho = \\ \begin{pmatrix} \sin y_{1\rho}\ \dot{y}_{1\rho} & 0 & 0 \\ 0 & \sin y_{2\rho}\ \dot{y}_{2\rho} & 0 \\ 0 & 0 & \sin y_{3\rho}\ \dot{y}_{3\rho} \end{pmatrix} \mathbf{p}_\rho \text{ en } (0,T), \\ \Gamma \dot{\mathbf{p}}_\rho(T) = k\dot{\mathbf{y}}_\rho(T). \end{array} \end{cases}$$

Con ésto el método de Newton para aproximar ρ^* queda

$$\rho_{i+1} = \rho_i - \frac{g'(\rho_i)}{g''(\rho_i)}$$

$$\rho_{i+1} = \rho_i - \frac{-\langle DJ(\mathbf{u} - \rho\mathbf{w}), \mathbf{w}\rangle}{\langle D^2 J(\mathbf{u} - \rho\mathbf{w})\mathbf{w}, \mathbf{w}\rangle}$$

$$\rho_{i+1} = \rho_i + \frac{\langle DJ(\mathbf{u} - \rho\mathbf{w}), \mathbf{w}\rangle}{\langle D^2 J(\mathbf{u} - \rho\mathbf{w})\mathbf{w}, \mathbf{w}\rangle}.$$

4 Discretización del problema de búsqueda en línea

La idea principal es sustituir el espacio $L^2(0,T)$ por un espacio de funciones más práctico. A grandes rasgos este espacio será el de las funciones lineales por pedazos en (0,T), que además ofrecen la ventaja de poder ser representadas computacionalmente por eneadas de números.

Para describir esto con mayor precisión, comenzamos haciendo una partición uniforme del intervalo $(0,T)$ con

- N número de subintervalos de la partición uniforme.
- $h = \frac{T}{N}$ tamaño de cada subintervalo.
- tt un vector de $N+1$ puntos espaciados uniformemente en el intervalo $(0,T)$ de la forma

$$tt = [0, \ldots, tt_N].$$

Con $tt_i = i * h, i = 0, 1, 2, \ldots, N$, esto significa que en el arreglo tt se almacena la partición en el tiempo.

Dada esta partición, una función $f(t)$ en $L^2(0,T)$ puede ser aproximada por la función lineal por pedazos $f_h(t)$ cuyo valor en t_i es $f_h(t_i) = f(t_i)$. En la figura 1 se muestra una f y su aproximación lineal f_h por pedazos, con $N = 10$. Note que entonces una función f puede ser representada computacionalmente por dos $(N+1)$-eadas:

$$f \approx \left\{ \begin{array}{c} (tt_0 = 0, \ldots, tti, \ldots, tt_N) \\ (f(0), \ldots, f(tt_i), \ldots, f(tt_N)). \end{array} \right.$$

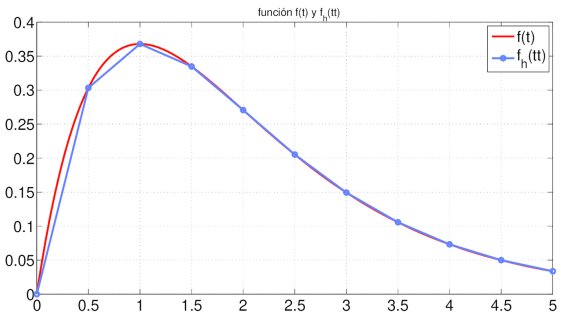


Figura 1: Función $f(t) = t\exp(-t)$ y $f_h(tt) = tt\exp(-tt)$, con $N = 10, T = 5$ y $h = 0.5$

Con esto, está claro que podemos aproximar las funciones $\mathbf{v}$ en $(L^2(0,T))^3$ por triadas de funciones lineales por pedazos, es decir por matrices de la forma

$$\mathbf{v}(t) \approx (v_1, v_2, v_3)(tt) = \begin{pmatrix} v_1(0) & \ldots & v_1(tt_N) \\ v_2(0) & \ldots & v_2(tt_N) \\ v_3(0) & \ldots & v_3(tt_N) \end{pmatrix},$$

mientras que las funciones $\mathbf{u}$ de nuestra teoría quedarían aproximada o representadas por la matriz

$$\mathbf{u} = \begin{pmatrix} u_1(0) & \ldots & u_1(tt_N) \\ u_2(0) & \ldots & u_2(tt_N) \\ u_3(0) & \ldots & u_3(tt_N) \end{pmatrix},$$

y las funciones $\mathbf{w}$ de nuestra teoría quedarían aproximadas o representadas por la matriz

$$\mathbf{w} = \begin{pmatrix} w_1(0) & \ldots & w_1(tt_N) \\ w_2(0) & \ldots & w_2(tt_N) \\ w_3(0) & \ldots & w_3(tt_N) \end{pmatrix}.$$

También a partir de ahora, estaremos usando la notación $\mathbf{v}^n$ para denotar el valor de la función $\mathbf{v}$, lineal por pedazos, en el tiempo nh. Esta notación también aplica para las funciones $\mathbf{y}$ y $\mathbf{p}$, es decir, con $\mathbf{y}^n$ denotamos el valor de la función $\mathbf{y}$ en el tiempo nh y similarmente para $\mathbf{p}$. Esto se resume en decir que en este capítulo damos los valores de $\mathbf{u}$ y $\mathbf{w}$ en la partición tt y debemos minimizar una version discreta del funcional, para lo cual es necesario encontrar los valores aproximados de

$$\mathbf{y} \approx (y_1, y_2, y_3)(tt) = \begin{pmatrix} y_1(0) & \dots & y_1(tt_N) \\ y_2(0) & \dots & y_2(tt_N) \\ y_3(0) & \dots & y_3(tt_N) \end{pmatrix},$$

y de

$$\mathbf{p} \approx (p_1, p_2, y_3)(tt) = \begin{pmatrix} p_1(0) & \dots & p_1(tt_N) \\ p_2(0) & \dots & p_2(tt_N) \\ p_3(0) & \dots & p_3(tt_N) \end{pmatrix}$$

en la partición tt. En lo que sigue detallamos cómo hacemos todas estas discretizaciones.

4.1. Discretización del Funcional

El funcional J de (3) lo aproximamos por

$$J^h(\mathbf{v}) = \frac{\eta h}{2} \sum_{n=1}^{N} ||\mathbf{v}^n||^2 + \frac{k}{2} ||\mathbf{y}^N - \mathbf{y}_T||^2,$$

con $||\mathbf{v}^n||^2 = |v_1^n|^2 + |v_2^n|^2 + |v_3^n|^2$. Aquí $\mathbf{y}^N$ es la aproximación de $\mathbf{y}$ en el tiempo N ($tt_N = Nh$), que más adelante especificamos cómo calcular.

4.2. Discretización del SEDO

Aproximamos el sistema (6) por un esquema de Euler explícito que luce como sigue:

$$\begin{cases} \text{Dado } \mathbf{y}^0 = \mathbf{y}_0, \\ \\ \text{para } n = 0, ..., N-1 \text{ resolver} \\ \\ \mathbf{y}^{n+1} = \mathbf{y}^n - h\Gamma^{-1} K \mathbf{y}^n - h\Gamma^{-1} \begin{pmatrix} \sin y_1^n \\ \sin y_2^n \\ \sin y_3^n \end{pmatrix} + h\Gamma^{-1} \begin{pmatrix} i_1 + v_1^n \\ i_2 + v_2^n \\ i_3 + v_3^n \end{pmatrix}. \end{cases}$$

4.3. Discretización del SEDO Adjunto

El sistema adjunto (7) lo aproximamos por

$$\begin{cases} \mathbf{p}^N = k\Gamma^{-1}(\mathbf{y}^N - \mathbf{y}_T), \\ \\ \text{para } n = N-1, ..., 0 \text{ resolver} \\ \\ \mathbf{p}^n = \mathbf{p}^{n+1} - h\Gamma^{-1} K \mathbf{p}^{n+1} - h\Gamma^{-1} \begin{pmatrix} \cos y_1^n & 0 & 0 \\ 0 & \cos y_2^n & 0 \\ 0 & 0 & \cos y_3^n \end{pmatrix} \mathbf{p}^{n+1}. \end{cases}$$

4.4. Discretización del Producto interno

Las discretizaciones mencionadas implican que estamos considerando el siguiente producto interno

$$(\mathbf{u}, \mathbf{w})_h = h \sum_{n=1}^{N} \mathbf{u}^n \cdot \mathbf{w}^n, \forall \mathbf{u}, \mathbf{w} \in (\mathbb{R}^3)^N.$$

4.5. Discretización de $g'(\rho)$ y $g''(\rho)$

Para resolver la ecuación $g(\rho^*) = 0$ con el método de Newton se requiere el conocimiento de $g'(\rho)$ que estará dada en los nuevos espacios vectoriales por

$$g'(\rho) = -h\eta \sum_{n=1}^{N} [\mathbf{u}^n - \rho \mathbf{w}^n + \mathbf{p}_\rho^n] \cdot \mathbf{w}^n,$$

donde $\mathbf{p}_\rho^n$ se obtiene resolviendo

$$\begin{cases} \text{Dado } \mathbf{y}_\rho^0 = \mathbf{y}_\rho(0), \\ \\ \text{para } n = 0, ..., N-1 \text{ resolver} \\ \\ \mathbf{y}_\rho^{n+1} = \mathbf{y}_\rho^n - h\Gamma^{-1} K \mathbf{y}_\rho^n - h\Gamma^{-1} \begin{pmatrix} \sin y_{\rho 1}^n \\ \sin y_{\rho 2}^n \\ \sin y_{\rho 3}^n \end{pmatrix} + h\Gamma^{-1} \begin{pmatrix} i_1 + u_1^n - \rho w_1^n \\ i_2 + u_2^n - \rho w_2^n \\ i_3 + u_3^n - \rho w_3^n \end{pmatrix}. \end{cases}$$

y luego resolviendo

$$\begin{cases} \mathbf{p}_\rho^N = k\Gamma^{-1}(\mathbf{y}_\rho^N - \mathbf{y}_T), \\ \\ \text{para } n = N-1, ..., 0 \text{ resolver} \\ \\ \mathbf{p}_\rho^n = \mathbf{p}_\rho^{n+1} - h\Gamma^{-1} K \mathbf{p}_\rho^{n+1} - h\Gamma^{-1} \begin{pmatrix} \cos y_{\rho 1}^{n+1} & 0 & 0 \\ 0 & \cos y_{\rho 2}^{n+1} & 0 \\ 0 & 0 & \cos y_{\rho 3}^{n+1} \end{pmatrix} \mathbf{p}_\rho^{n+1}. \end{cases}$$

y para

$$g''(\rho) = h \sum_{n=1}^{N} [\eta \mathbf{w}^n - \dot{\mathbf{p}}_\rho^n] \cdot \mathbf{w}^n,$$

con $\dot{\mathbf{p}}_\rho^n$, obtenido al resolver

$$\begin{cases} \dot{\mathbf{y}}_\rho^0 = \mathbf{0}, \\ \text{para } n = 0, ..., N-1 \text{ resolver} \\ \dot{\mathbf{y}}_\rho^{n+1} = \dot{\mathbf{y}}_\rho^n - h\Gamma^{-1} K \dot{\mathbf{y}}_\rho n - h\Gamma^{-1} \begin{pmatrix} \cos y_{1\rho}^n \\ \cos y_{2\rho}^n \\ \cos y_{3\rho}^n \end{pmatrix} \dot{\mathbf{y}}^n{}_\rho - h\Gamma^{-1} \begin{pmatrix} w_1^n \\ w_2^n \\ w_3^n \end{pmatrix}, \end{cases}$$

y luego resolver

$$\begin{cases} \dot{\mathbf{p}}_\rho^N = k\Gamma^{-1} \dot{\mathbf{y}}_\rho^N, \\ \text{para } n = N-1, ..., 0 \text{ resolver} \\ \dot{\mathbf{p}}_\rho^n = \dot{\mathbf{p}}_\rho^{n+1} + h\Gamma^{-1} K \dot{\mathbf{p}}_\rho^{n+1} \\ \\ + h\Gamma^{-1} \begin{pmatrix} \cos y_{1\rho}^n & 0 & 0 \\ 0 & \cos y_{2\rho}^n & 0 \\ 0 & 0 & \cos y_{3\rho}^n \end{pmatrix} \dot{\mathbf{p}}_\rho^{n+1} \\ \\ - \begin{pmatrix} \dot{y}_{1\rho}^n \sin y_{1\rho}^n & 0 & 0 \\ 0 & \dot{y}_{2\rho}^n \sin y_{2\rho}^n & 0 \\ 0 & 0 & \dot{y}_{3\rho}^n \sin y_{3\rho}^n \end{pmatrix} \mathbf{p}_\rho^{n+1}. \end{cases}$$

Con esto podemos ya aplicar el método de Newton para resolver la versión discreta de $g'(\rho) = 0$: Dado ρ^0, iterar con

$$\rho^{m+1} = \rho^m - \frac{g'(\rho^m)}{g''(\rho^m)}.$$

Usamos el criterio de paro $\dfrac{|\rho^{m+1} - \rho^m|}{|\rho^{m+1}|} < \varepsilon$, para ε pequeño dado.

5 Experimentación computacional

5.1. Ejemplo 1

En este ejemplo tomamos $u_1(t) = t\exp(-t)$, $u_2(t) = t^3$, $u_3(t) = 0$ y $w_1(t) = \exp(-t)/10$, $w_2(t) = 3t - t^3$ y $w_3(t) = 0$. Se tomaron los valores siguientes para los parámetros en la iteración de Newton y en la discretización de los problemas involucrados:

- $T = 16$; $k = 1.0e-3$; $\rho^0 = 0$; $h = T/500$; $\eta = 1$.
- $\mathbf{y}_0 = [1.2514; 0.7456; -0.9753]$; $\mathbf{y}_T = [7.4207; 6.4958; -0.3236]$.

Dadas las funciones $\mathbf{u} = (t\exp(-t), t^3, 0)$ y $\mathbf{w} = (\exp(-t)/10, 3t - t^3, 0)$, minimizamos con el método de Newton el funcional $J(\mathbf{u}(t) - \rho\mathbf{w}(t))$ sobre ρ y en 3 iteraciones se obtuvo el valor $\rho^* = -1.017$. La gráfica del funcional J restringido a la recta $\mathbf{v} = \mathbf{u} - \rho\mathbf{w}$ se muestra en la Figura 2.

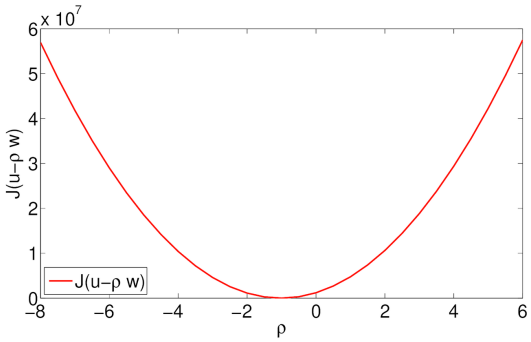


Figura 2: Funcional $J(\mathbf{u}(t) - \rho\mathbf{w}(t))$ para las funciones $\mathbf{u}$ y $\mathbf{w}$ del ejemplo 1.

5.2. Ejemplo 2

En este ejemplo tomamos $u_1(t) = -(t-2)\exp(-t)$, $u_2(t) = 3t^2 + 1$, $u_3(t) = (t-1)^2 + 1$ y $w_1(t) = t^3 + t - 1$, $w_2(t) = \dfrac{1}{3}t^3 - t$, $w_3(t) = \exp(-t)/10$. Tanto para la discretización de los problemas involucrados como para las iteraciones de Newton se tomaron los valores siguientes:

- $T = 12$; $k = 1.0e-3$; $\rho^0 = 0.5$; $h = T/500$; $\eta = 1$.
- $\mathbf{y}_0 = [0.1992; 0.1187; -0.1552]$; $\mathbf{y}_T = [0.1810; 0.0338; -0.0515]$.

Dadas las funciones $\mathbf{u} = (-(t-2)\exp(-t), 3t^2 + 1, (t-1)^2 + 1)$ y $\mathbf{w} = (t^3 + t - 1, \frac{1}{3}t^3 - t, \exp(-t)/10)$, minimizamos con el método de Newton el funcional $J(\mathbf{u}(t) - \rho\mathbf{w}(t))$ sobre ρ y en 6 iteraciones se obtuvo el valor $\rho^* = 0.084$. La gráfica del funcional J restringido a la recta $\mathbf{v} = \mathbf{u} - \rho\mathbf{w}$ se muestra en la Figura 3.

6 Conclusiones

No contamos con un ejemplo para el cual se conozca la solución exacta, así que nuestra validación se basa en el desempeño del método en ejemplos como los dos mostrados, dado que es posible tener una visualización suficientemente detallada del funcional J restringido a la "recta" que pasa por $\mathbf{u}$ y tiene dirección $\mathbf{w}$. De acuerdo a estos ejemplos se puede concluir que el método de Newton produce resultados excelentes. En este trabajo el objetivo no es comparar el método de Newton con otros métodos, sino verificar que este tipo de problemas se pueden resolver aceptablemente utilizando el método de Newton; sin embargo en trabajos futuros esperamos considerar otros métodos y hacer tal comparación, por ejemplo, con los métodos que no utilizan derivadas. También podemos mencionar que los resultados presentados aquí se aplicarán próximamente para resolver el problema de

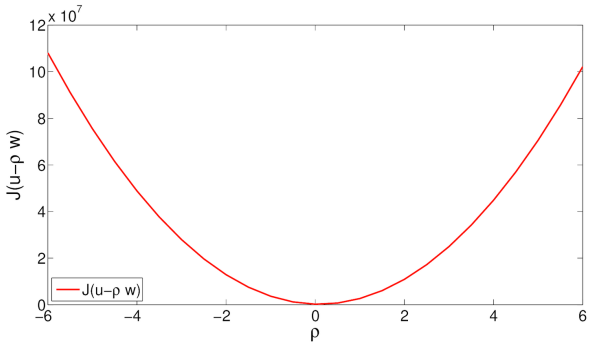


Figura 3: Funcional $J(\mathbf{u}(t) - \rho\mathbf{w}(t))$ para las funciones $\mathbf{u}$ y $\mathbf{w}$ del ejemplo 2.

minimización global asociado con el problema de control, el cual, como se conocen el estado inicial del que se parte y el estado final al que se quiere llegar, podrá considerarse como un caso de validación de la metodología expuesta en el presente trabajo.

Referencias

[1] Y. Braiman, B. Neschke, N. I. N. Nair, and R. Glowinski. Memory states in small arrays of josephson junctions. *Physical Review E*, 94(5), 2016.

[2] R. P. Brent. *Algorithms for minimization without derivatives*. Dover Publications, 2002.

[3] C. N. Cortazar-Cortazar. Método de newton para búsquedas en línea en el espacio $(l^2)^3$. Master's thesis, Universidad Juárez Autónoma de Tabasco, 2020.

[4] R. Glowinski. *In Handbook of Numerical Analysis*, volume 9, chapter Finite element methods for incompressible viscous flow, pages 3–1176. Elsevier, 2003.

[5] R. Glowinski. *Variational Methods for the Numerical Solution of Nonlinear Elliptic Problems*. SIAM, 2015.

[6] J. López-López and L. H. Juárez-Valencia. Método de newton para problemas de búsqueda en línea en l^2. *Journal of Basic Sciences*, 5(13), 2019.

[7] J. D. Rezac, N. Imam, and Y. Braiman. Parameter optimization for transitions between memory states in small arrays of josephson junctions. *Physica A: Statistical Mechanics and its Applications*, 474(C), 2017.

Solución de la ecuación de Poisson en 2D usando algoritmos en paralelo en entornos multicore

Miguel Uh Zapata
Josué Pinzón Vivas
Centro de Investigación en Matemáticas
CONACYT - CIMAT Unidad Mérida
angeluh@cimat.mx

Resumen

En este artículo se propone una implementación en paralelo usando un entorno multicore con MPI a la solución del sistema lineal resultante de la discretización de la ecuación de Poisson en 2D usando diferencias finitas y el método iterativo de Jacobi. El tamaño del dominio y su correspondiente discretización resultan en un sistema de ecuaciones lineales donde el número de variables puede alcanzar los miles de millones. La misma magnitud del problema permite que el algoritmo sea muy escalable paralelamente; esto significa, que al aumentar el número de procesadores disponibles para resolver el sistema, el tiempo de ejecución mejorará considerablemente. Sin embargo, al aumentar el número de procesadores, el trabajo de comunicación que realiza el algoritmo es demasiado y esto detiene su desempeño. Por lo tanto, en este artículo se propone una re-ingeniería del algoritmo paralelo enfocado en el manejo de la memoria para poder acelerar su ejecución y mejorar su efectividad.

Palabras Clave: Poisson 2D; MPI; Overlap domain decomposition; Finite difference method

1 Introducción

Uno de los grandes desafíos en muchas ramas de la ciencia, es la obtención de simulaciones numéricas precisas, de mayor alcance, rápidas y de forma accesible. En particular, la complejidad de los problemas en dinámica de fluidos es tal, que las simulaciones numéricas requieren mucho tiempo de ejecución, cuyos resultados pueden durar varios días, incluso hasta meses. En la mayoría de estos problemas, la solución de la ecuación de Poisson representa el mayor costo computacional, inclusive más del 90 % de la simulación total del problema [5, 14, 19, 21]. Parte del problema, es que su discretización proporciona un sistema de ecuaciones lineales donde el número de variables puede alcanzar los miles de millones.

Existen diferentes métodos para la solución de la ecuación de Poisson en dos dimensiones (2D). Aunque cada método tiene sus propias fortalezas y limitaciones. Debido a sus características y al éxito que ha tenido al ser implementado en numerosos fenómenos físicos, el método de diferencias finitas [2, 10, 13] es el método numérico implementado en este trabajo.

La discretización de la ecuación de Poisson usando diferencias finitas proporciona un sistema de ecuaciones lineales donde el número de incógnitas es proporcional al número de puntos usados en el mallado. Tradicionalmente los sistemas lineales son resueltos usando métodos iterativos, los cuales aproximan la solución mediante un número de iteraciones hasta que cierta tolerancia se satisface. Métodos como el Gradiente Conjugado [3, 8] y el "Generalized Minimal Residual" (GMRES) [7, 17] pueden ser considerados. Sin embargo, estos últimos tienen una mayor dificultad en el momento de implementar y especialmente al tratar de obtener un programa más eficiente en paralelo. Por otro lado, métodos clásicos como Jacobi, Gauss-Seidel y "Succesive-Over Relaxation" (SOR) pueden ser usados con cierto grado de dificultad de implementación y convergencia razonable [6, 18, 20]. Como primera

aproximación, este trabajo se concentrará en el método de Jacobi, el cual es el más sencillo de implementar pero es base para solución de otros algoritmos más complejos.

Varios algoritmos en paralelo han sido desarrollados para resolver grandes sistemas lineales en distintos campos de la ciencia, incluyendo diversas plataformas como los son MPI("Message Passing Interface"), GPU "Graphics Processing Unit"), OpenMP("Open Multi-Processing"), entre otros. Por ejemplo, se han utilizado para algoritmos involucrados con imágenes médicas [16], Finanzas Computacionales [1], Bioinformática [12] y Dinámica de Fluidos [11], por citar algunos.

Cada metodología en paralelo tiene sus diferentes grados de dificultad y eficiencia, sin embargo, algunas de las ventajas de usar MPI [4, 9, 15] es su grado de universalidad. El modelo de paso de mensajes se adapta bien a la mayoría de las actuales super-computadoras debido a su arquitectura de procesadores separados pero conectados por una red de comunicación. En cuanto a su desempeño y rendimiento, con el modelo de paso de mensajes uno puede asociar explícitamente un espacio de memoria a un proceso lo cual otorga un mejor uso de las caches y jerarquías de memoria, lo cual se ve reflejado directamente en el rendimiento. Aún más, en sistemas de memoria compartida se puede mejorar el rendimiento al tener más control sobre la localidad de los datos en la memoria.

El objetivo de este trabajo es diseñar e implementar un algoritmo en paralelo para obtener la solución de manera más rápida de sistemas de ecuaciones lineales obtenidas del método de diferencias finitas y el método de descomposición de dominios. Más aún en este trabajo se propone mejorar los algoritmos en paralelo usando diferentes técnicas de cómputo como el manejo de la forma de acceso a la memoria. Las siguientes secciones están organizadas como se explica a continuación. En la segunda sección se introduce la discretización de la ecuación de Poisson usando diferencias finitas. En la tercera sección se presenta el algoritmo en paralelo. La cuarta sección presenta los resultados numéricos. Finalmente, en la última sección se dan las conclusiones finales de este trabajo.

2 Discretización de la ecuación de Poisson en 2D usando diferencias finitas

En esta sección se describirá brevemente el método de diferencias finitas usado para discretizar la ecuación de Poisson en dos dimensiones. Además, una vez obtenido el sistema de ecuaciones lineales resultantes, se presentará el método iterativo para su correspondiente solución.

Considérese la siguiente ecuación de Poisson en dos dimensiones:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = f(x,y), \quad (x,y) \in \Omega, \tag{1}$$

sobre un dominio $\Omega = \{(x,y) | a < x < b, c < y < d\}$. Además, se considera condiciones de frontera de Dirichlet en la frontera $\partial\Omega$, es decir, la solución es conocida de manera exacta en todas las fronteras del dominio.

2.1. El método de diferencias finitas

La forma discreta del dominio viene dada por una malla estructurada rectangular. Se considera los índices $i = 1, 2, \ldots, N_x$ y $j = 1, 2, \ldots, N_y$, donde N_x y N_y son el número de puntos en la dirección x y y, respectivamente. Los tamaños de paso están definidos como $\Delta x = (b-a)/(N_x - 1)$ y $\Delta y = (d-c)/(N_y - 1)$. Finalmente los puntos donde la solución es aproximada están dados por $(x_i, y_j) = (a + i\Delta x, c + j\Delta y)$. Para mayor detalles véase la Figura 1.

Como se tiene un dominio rectangular y una malla regular, la ecuación diferencial puede ser discretizada usando diferencias finitas centradas para la segunda derivada parcial. Ésta es obtenida usando desarrollos de series de Taylor. Así, las aproximaciones están dadas por

$$\left.\frac{\partial^2 p}{\partial x^2}\right|_{i,j} = \frac{p_{i-1,j} - 2p_{i,j} + p_{i+1,j}}{\Delta x^2} + O(\Delta x^2), \tag{2}$$

$$\left.\frac{\partial^2 p}{\partial y^2}\right|_{i,j} = \frac{p_{i,j-1} - 2p_{i,j} + p_{i,j+1}}{\Delta y^2} + O(\Delta y^2), \tag{3}$$

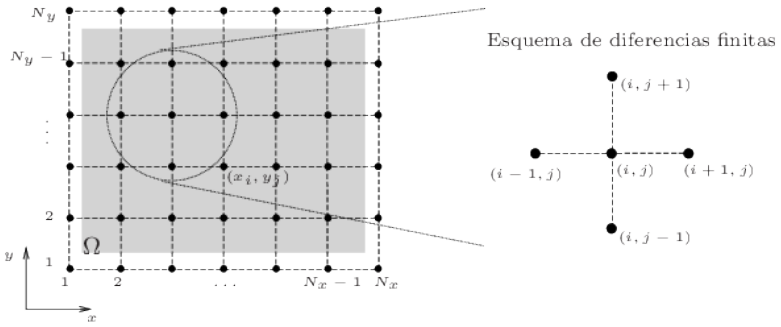


Figura 1: Dominio y esquema de diferencias finitas empleados. La zona sombreada es donde los valores son aproximados.

donde $p_{i,j} = p(x_i, y_j)$. Sustituyendo (8)(2)-(3) en la ecuación (1), se obtiene

$$\frac{1}{\Delta x^2}\left(p_{i-1,j} - 2p_{i,j} + p_{i+1,j}\right) + \frac{1}{\Delta y^2}\left(p_{i,j-1} - 2p_{i,j} + p_{i,j+1}\right) = f_{i,j},$$

donde $f_{i,j} = f(x_i, y_j)$. Finalmente la discretización en diferencias finitas de la ecuación de Poisson (1) está dada por

$$p_{i+1,j} + p_{i-1,j} - 2(1+\beta^2)p_{i,j} + \beta^2(p_{i,j+1} + p_{i,j-1}) = \Delta x^2 f_{i,j}, \tag{4}$$

donde $\beta = \Delta x/\Delta y$. Se puede observar que el esquema resultante relaciona un valor $p_{i,j}$ con sus cuatro vecinos más cercanos, como es ilustrado en la Figura 1.

2.2. El sistema lineal resultante

Todos los valores de la frontera son conocidos, es decir, ya no es necesario calcular $p_{1,j}$, $p_{N_x,j}$, $p_{i,1}$ y p_{i,N_y}. Así, sólo se aproximan los valores $p_{i,j}$ donde $i = 2, 3, \ldots, N_x - 2, N_x - 1$ y $j = 2, 3 \ldots, N_y - 2, N_y - 1$. Se establece un indexado "natural" para resolver el sistema. Este consiste en ordenar todas las variables en un vector, enumerando las variables por columna, así, el vector de incógnitas a resolver es

$$\mathbf{p} = \left[\mathbf{p}_2, \mathbf{p}_3, \cdots, \mathbf{p}_{N_x-2}, \mathbf{p}_{N_x-1}\right]^T, \tag{5}$$

donde cada elemento $\mathbf{p}_i$ $(i = 2, 3, ..., N_x - 1)$ es un vector columna dada por

$$\mathbf{p}_2 = \begin{bmatrix} p_{2,2} \\ p_{2,3} \\ \vdots \\ p_{2,N_y-2} \\ p_{2,N_y-1} \end{bmatrix}, \mathbf{p}_3 = \begin{bmatrix} p_{3,2} \\ p_{3,3} \\ \vdots \\ p_{3,N_y-2} \\ p_{3,N_y-1} \end{bmatrix}, \ldots, \mathbf{p}_{Nx-2} = \begin{bmatrix} p_{N_x-2,2} \\ p_{N_x-2,3} \\ \vdots \\ p_{N_x-2,N_y-2} \\ p_{N_x-2,N_y-1} \end{bmatrix}, \mathbf{p}_{Nx-1} = \begin{bmatrix} p_{N_x-1,2} \\ p_{N_x-1,3} \\ \vdots \\ p_{N_x-1,N_y-2} \\ p_{N_x-1,N_y-1} \end{bmatrix}.$$

Así, se tiene el sistema lineal

$$A\mathbf{p} = \mathbf{f}, \tag{6}$$

donde

$$A = \begin{bmatrix} T & I & 0 & & \cdots & 0 \\ I & T & I & 0 & \cdots & 0 \\ 0 & I & T & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & I & T & I \\ 0 & \cdots & \cdots & 0 & I & T \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \\ \vdots \\ \mathbf{f}_{N_x-2} \\ \mathbf{f}_{N_x-1} \end{bmatrix}.$$

Notar que la matriz A es una matriz tridiagonal a bloques, donde los bloques T son a su vez unas matrices tridiagonales dadas por

$$T = \begin{bmatrix} -2(1+\beta^2) & \beta^2 & 0 & & \cdots & 0 \\ \beta^2 & -2(1+\beta^2) & \beta^2 & 0 & \cdots & 0 \\ 0 & \beta^2 & -2(1+\beta^2) & \beta^2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \beta^2 & -2(1+\beta^2) & \beta^2 \\ 0 & \cdots & \cdots & 0 & \beta^2 & -2(1+\beta^2) \end{bmatrix},$$

y las matrices I corresponden a la matriz identidad. Los valores del lado derecho están dados por los valores correspondientes f_{ij} mas los valores conocidos de la frontera.

2.3. Solución del sistema por el método de Jacobi

Es bien conocido que el método de Jacobi converge lentamente a la solución del problema. Sin embargo, éste tiene la notoria característica que para calcular una nueva iteración sólo se requieren de todos los elementos en el paso anterior. Esta propiedad es raramente compartida por otros solucionadores de sistemas lineales. Por ejemplo, en los métodos de Gauss-Seidel y SOR, la actualización de un elemento depende de otros elementos ya calculados en la misma iteración. Eso implica que el método se realice de a uno por uno en un orden estricto, imposibilitando la paralelización por bloques de dicho método. Existen técnicas en las cuales se puede realizar la paralelización; sin embargo, en este trabajo únicamente se propone un algoritmo en paralelo para el método iterativo de Jacobi, dejando las otras técnicas para trabajos futuros.

Para obtener el método iterativo de Jacobi se considera la descomposición de la matriz $A = D - E - F$ en donde D , $-E$ y $-F$ son las matrices diagonal, tridiagonal inferior y triangular superior de A, respectivamente. Con esta descomposición podemos sustituir en (6) para obtener $(D-E-F)\mathbf{p} = \mathbf{f}$ de la cual se obtiene la iteración para el método de Jacobi

$$\mathbf{p}^{(k+1)} = D^{-1}\mathbf{f} + D^{-1}(E+F)\mathbf{p}^{(k)}, \tag{7}$$

en donde $\mathbf{p}^k$ y $\mathbf{p}^{(k+1)}$ son las aproximaciones obtenidas mediante k y $k+1$ iteraciones, respectivamente. Finalmente, el método iterativo de Jacobi resulta de iterar repetidamente

$$p_s^{k+1} = \frac{1}{a_{ss}}\left[f_s - \sum_{r=1, s\neq r}^{N} a_{sr}p_s^k\right], \tag{8}$$

hasta que el residuo del método sea menor o igual a una tolerancia ε dada. Notar que los índices se han cambiado por s y r para recalcar que estos no son los mismos de los índices i, j correspondientes a la malla, sino que estos representan la posición de los elementos en la matriz global. En otras palabras, un elemento $p_s = p_{i,j}$ para cierto indice i y j.

El método iterativo de Jacobi (8) puede ser fácilmente aplicable al problema, dado que cada fila de la matriz tiene a lo más 5 componentes, así fácilmente se puede ver que

$$p_{i,j}^{k+1} = \frac{1}{-2(1+\beta^2)}\left[\Delta x^2 f_{ij} - \left(p_{i+1,j}^k + p_{i-1,j}^k + \beta^2(p_{i,j+1}^k + p_{i,j-1}^k)\right)\right]. \tag{9}$$

Notar que los índices en (9) han sido cambiados a su notación original para mayor facilidad de identificación del esquema.

Para poder calcular el criterio de paro del método iterativo es necesario obtener el residuo, éste viene definido por $r = ||\mathbf{f} - A\mathbf{p}||$. En este artículo usaremos la norma L_2. Finalmente, para el cálculo más eficiente del residuo y de la solución, el método iterativo (9) se modifica levemente como:

$$\begin{aligned} r_{i,j} &= \Delta x^2 f_{ij} - \left(p_{i+1,j}^k + p_{i-1,j}^k - 2(1+\beta^2)p_{i,j}^k + \beta^2(p_{i,j+1}^k + p_{i,j-1}^k)\right), \\ p_{i,j}^{k+1} &= p_{i,j}^k + \frac{1}{-2(1+\beta^2)}r_{ij}. \end{aligned} \tag{10}$$

3 Algoritmo en paralelo en entornos multicore

En esta sección se describe el método de descomposición de dominios para la programación en paralelo del sistema lineal resultante usando el método iterativo de Jacobi en un entorno "multicore", incluyendo las técnicas de comunicación entre procesos.

Aunque existen diversos modelos de computación paralela, en este trabajo se emplea el paso de mensajes. Este modelo se caracteriza principalmente en que cada proceso tiene su propia memoria local, pero son capaces de comunicarse entre sí enviando y recibiendo mensajes. MPI ("Message Passing Interface") es una especificación de dicho modelo. Este no es un lenguaje de programación, MPI es una librería que especifíca los nombres y las llamadas de funciones o subrutinas. Dichos programas pueden ser compilados con los compiladores ordinarios de cada lenguaje y únicamente necesitan ser vinculados con la librería MPI [9].

3.1. Método de descomposición del dominio

La estrategia del algoritmo en paralelo planteada en este artículo requiere que el dominio sea dividido en varios sub-dominios. Cada sub-dominio será asignado a un proceso, y por medio del método iterativo de Jacobi cada proceso resolverá su propio sistema en su sub-dominio asignado. En la Figura 2 se ejemplifica la descomposición de un dominio estructurado en tres sub-dominios. Se puede observar que a cada sub-dominio se le han incorporado los elementos adicionales de los sub-dominios vecinos.

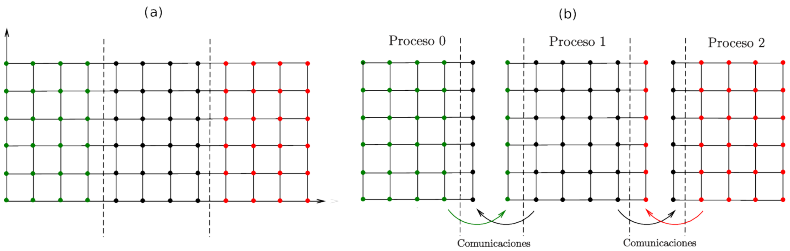


Figura 2: División del dominio con un mallado rectangular en sub-dominios y sus comunicaciones.

El esquema numérico de cinco puntos en cruz proporcionado por el método de Jacobi (10), ver Figura 1, combinado con el método de descomposición de dominio requiere que cada elemento del sub-dominio recopile información de sus elementos vecinos por cada iteración. Esto da como resultado que los elementos del borde (o de frontera artificial) de cada sub-dominio requieran información de elementos que pertenecen al sub-dominio vecino. Lo anterior representa la primera dificultad técnica del algoritmo en paralelo dado que los procesos no tienen acceso directo a la memoria de los demás procesos. Para solucionar este problema, en este artículo, los sub-dominios son creados de tal manera que estén sobrepuestos. Así los sub-dominios incluyen los elementos vecinos colindantes entre ellos.

Como hemos mencionado anteriormente, el método de Jacobi (10) sólo requiere de los elementos de la iteración anterior para actualizar todos los elementos del siguiente paso. Sin embargo, al inicio de la nueva iteración, los elementos adicionales en los bordes no han sido actualizados. Así, los procesos deben comunicarse enviando y recibiendo los resultados de sus elementos frontera después de cada iteración. De este modo, cada proceso tendrá la información actualizada que requiere para generar los nuevos resultados. En la siguiente sección profundizaremos un poco más en el tema de la comunicación entre procesadores.

3.2. Comunicación entre procesos

Una de las partes más importantes a considerar a la hora de realizar un algoritmo en paralelo en MPI es la comunicación entre los procesos, ya que en la mayoría de los casos, varios procesos estarán haciendo una misma operación u operaciones parecidas sobre una sección del problema global. Así, será necesario que los procesos

comuniquen sus resultados a otros procesos antes de continuar o empezar de nuevo un ciclo de operaciones.

Un sólo algoritmo debe ser programado para diferentes números de procesos. Así la comunicación debe hacerse de forma dinámica. Es decir, las órdenes de envío y recepción de mensajes se deben ajustar automáticamente al número de procesos. No es recomendable escribir un código para cada caso particular ya que las instrucciones condicionales para elegir el caso correspondiente también consumen un tiempo valioso de ejecución.

Para poder enviar simultáneamente múltiples mensajes y que se envíen al destinatario correcto sin tener que especificar para cada proceso un caso particular, basta con que se conozca desde un principio quiénes son los vecinos de cada sub-dominio. Por la forma en que se organizó la malla, esa información es conocida desde el inicio del programa. A cada sub-dominio se le asignó un proceso, dicho proceso identificado con el rank obtenido de la sub-rutina MPI_Comm_rank. Si pensamos que la división de dominios es únicamente unidimensional, entones a su vecino derecho le corresponde el identificador rank+1. De la misma manera, a su vecino izquierdo le corresponde rank-1. De tal modo que al usar como parámetro los identificadores correspondientes a sus vecinos, dicho parámetro se vuelve un parámetro "dinámico". Para cada proceso que ejecute la sub-rutina de envío de mensaje, el parámetro tendrá un valor distinto que será correspondiente al proceso vecino de a quién se enviará.

A pesar de que la técnica anterior de programación simplifica mucho la programación del código, aún hay que considerar ciertos casos que pueden hacer que el programa no funcione correctamente. Para esto se implementan otros índices que ayudan a localizar en dónde se encuentra el proceso asignado a un sub-dominio con respecto al dominio global. Dado que el dominio es de forma rectangular, la malla aplicada es estructurada con rectángulos del mismo tamaño, la asignación de los procesos es de orden ascendente, primero a lo ancho y luego a lo largo. Mas aún, se conoce el número de sub-dominios a lo ancho (numx) y el número de sub-dominios a lo largo (numy). Así

- Se puede conocer en que columna de sub-dominios se encuentra el proceso (idx) al obtener el valor del identificador del proceso módulo 2 (idx = modulo(rank,numx)).
- Igualmente se puede obtener en qué fila de sub-dominios se encuentra el proceso (idy) al aproximar al entero inferior más cercano el resultado de dividir el valor de su identificador entre el número de sub-dominios a lo ancho (idy = rank/numx).

En "Fortran 90" dado que los operadores están definidos como INTEGER, automáticamente el valor toma sólo la parte entera, lo que resulta equivalente a obtener el entero inferior más cercano. En el código de la Figura 3 se presenta la implementación de la comunicación a un programa con un dominio regular rectangular y mallado estructurado.

```
SUBROUTINE edge_communication(B,My,Mx,newcolumn,rank,idy,idx, &
    numy,numx,count,MPI_COMM_WORLD,status,ierr)
INTEGER, intent(in) :: rank,idy,idx,numy,numx,count,MPI_COMM_WORLD
INTEGER, intent(inout) :: status(*), ierr
real*8, intent(inout) :: B(My+2,Mx+2)
! Enviar
if (idx.ne.(numx-1)) then ! Lado derecho - borde derecho no envia
    call MPI_Send(B(2,Mx+1), 1, newcolumn, rank+1, count*4+1, &
        MPI_COMM_WORLD, ierr)
endif
if (idx.ne.0) then ! Lado izquierdo - borde izquierdo no envia
    call MPI_Send(B(2,2), 1 , newcolumn, rank-1, count*4+2, &
        MPI_COMM_WORLD, ierr)
endif
! Recibir
if (idx.ne.0) then        ! Lado derecho - borde izquierdo no recibe
    call MPI_Recv(B(2,1), 1, newcolumn, rank-1, count*4+1, &
        MPI_COMM_WORLD, status, ierr)
endif
if (idx.ne.(numx-1)) then ! Lado izquierdo - borde derecho no recibe
    call MPI_Recv(B(2,Mx+2), 1, newcolumn, rank+1, count*4+2, &
        MPI_COMM_WORLD, status, ierr)
endif
. . .
END SUBROUTINE
```

Figura 3: Código correspondiente a la subrutina de comunicación entre sub-dominios.

4 Simulaciones Numéricas

En esta sección se presentan las simulaciones numéricas usando el algoritmo propuesto en paralelo. Inicialmente se propone un ejemplo donde se analiza el desempeño de la subdivisión de dominios, la comunicación y el manejo de memoria. Posteriormente se analiza la solución de la ecuación de Poisson en 2D y su correspondiente desempeño en paralelo. Los resultados son reportados usando una máquina estándar de 20 cores 3.0 GHz Intel Xeon. Todos los códigos fueron implementados en el lenguaje "Fortran 90".

4.1. Ejemplo 1: Pruebas de desempeño

Con el objetivo de medir el desempeño del algoritmo en paralelo propuesto, se desarrolló este ejemplo que consiste en aplicar un esquema en el que cada elemento es sustituido por el promedio de sí mismo y sus cuatro vecinos (superior, inferior, derecho e izquierdo). Dicho proceso es repetido 10 veces. La matriz global $A(:,:)$ es de dimensiones 4×232792562. Con este número (equivalente a $20 \cdot 19 \cdot 18 \cdot 17 \cdot 13 \cdot 7 \cdot 11 \cdot 2 + 2$) el dominio es divisible exactamente por todos los enteros del 1 al 20. La suma del número 2 es porque al dividir el dominio no se consideran los elementos del borde.

4.1.1. Prueba 1. División en sub-dominios

En esta prueba el dominio únicamente es dividido a lo ancho, es decir, que los sub-dominios están únicamente uno a lado del otro, sin tener sub-dominios vecinos superiores e inferiores. De esta manera, cada procesador únicamente tiene que comunicar sus bordes a sus vecinos derecho e izquierdo, evitando así el tener que crear comunicaciones entre vecinos superiores e inferiores.

El tiempo de acceso a la memoria en "Fortran" depende mucho del orden de acceso de los índices de una matriz. Así, puede haber diferencias de tiempo si se accede a elementos contiguos horizontales $A(i,j)$ y $A(i,j+1)$ que a elementos contiguos verticales $A(i,j)$ y $A(i+1,j)$. Dado lo anterior, se realizó una segunda prueba invirtiendo las dimensiones de la matriz y dividiendo el dominio a lo vertical en lugar de a lo horizontal. La Figura 4 muestra las medianas de los tiempos de cómputo y de la eficiencia usando ambas configuraciones.

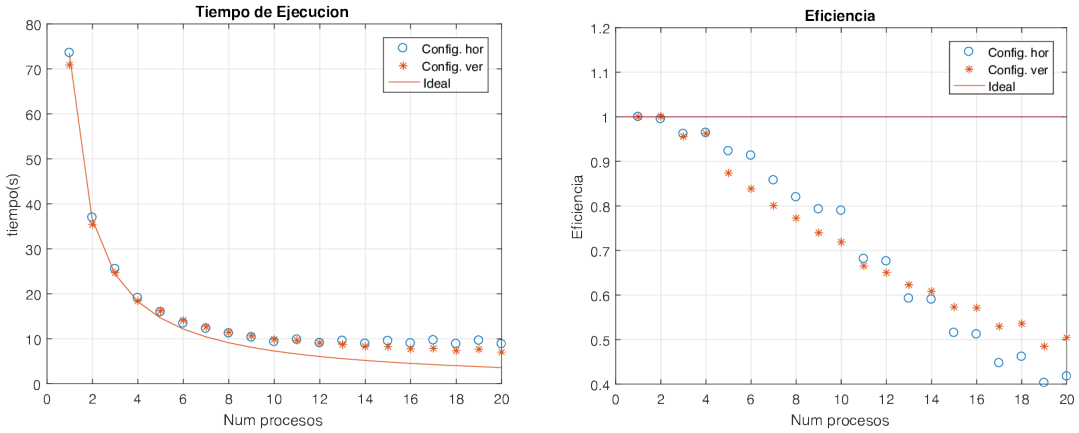


Figura 4: Tiempo de ejecución (en segundos) y eficiencia comparando la configuración horizontal contra configuración vertical usando diferente número de procesos.

Se puede observar que la eficiencia tiende a alejarse del ideal mientras aumenta el número de procesos en paralelo. Los tiempos de ejecución de la configuración vertical fueron reducidos en comparación a los datos obtenidos en la horizontal. Sin embargo, la mejora en la eficiencia no es consistente para todos los números de procesos. Más aún, la caída de rendimiento en el algoritmo en ambos casos es más de lo que se espera. Aunque los resultados no son mostrados aquí, los tiempos de comunicación no varían se mantienen estables con el cambio

del número de procesos. Por lo tanto, la comunicación de los bordes no está contribuyendo a reducir de manera significativa el rendimiento del algoritmo.

4.1.2. Prueba 2. Mejoras en el acceso a memoria

Como mencionamos anteriormente, el orden de acceder a la memoria por medio de los índices de las matrices en "Fortran" puede variar la velocidad de dicho acceso. Así en esta segunda prueba se presentan los resultados de cambiar el orden en cómo se ejecuta el esquema numérico, invirtiendo el orden de los índices de los ciclos do, como se muestra en el extracto de los códigos en las Figuras 5 y 6.

```
! Aplicando el esquema
do i=2,My+1
    do j=2,Mx+1
        C(i,j) = (B(i,j)+B(i-1,j)+B(i,j+1)+B(i+1,j)+B(i,j-1))/5.0d0
    enddo
enddo
! Actualizando matrices locales
B(2:My+1,2:Mx+1) = C(2:My+1,2:Mx+1)
```

Figura 5: Código de la configuración del stencil original.

```
! Aplicando el esquema
do j=2,Mx+1
    do i=2,My+1
        C(i,j) = (B(i,j)+B(i-1,j)+B(i,j+1)+B(i+1,j)+B(i,j-1))/5.0d0
    enddo
enddo
! Actualizando matrices locales
B(2:My+1,2:Mx+1) = C(2:My+1,2:Mx+1)
```

Figura 6: Código de la configuración del stencil con acceso invertido.

En el código 5, siguiendo el orden de los ciclos do de la configuración original, la matriz se recorre haciendo barridos horizontales a través del índice j mientras avanza a lo vertical con el índice i después de cada barrido horizontal. Dicha configuración fue cambiada de tal manera que se invirtiera el orden del acceso, como se ve en el código 6. En dicho código se recorre la matriz haciendo barridos verticales a través del índice i mientras avanza a lo horizontal con el índice j después de cada barrido vertical.

Los resultados con estos cambios se presentan en la Figura 7. Se puede observar que el cambiar el orden de acceso a la matriz al momento de aplicar el esquema numérico mejora significativamente el tiempo de ejecución y consecuentemente el "speed-up" y la eficiencia. También se puede observar que la configuración vertical obtiene mejores resultados consiguiendo hasta un 72 % de eficiencia para 20 procesos en paralelo, comparado con el primer código que se usó en el que sólo se obtuvo un 41 %, ver Figura 4.

4.1.3. Prueba 3. Implementación de punteros

Aunque se logró mejorar, aún existe una caída significativa del rendimiento del algoritmo. Notar que en los Códigos 5 y 6, después de aplicar el esquema numérico a la matriz B, es necesario actualizar dicha matriz. Una matriz C almacena temporalmente los nuevos resultados y posteriormente es copiada a B. Este proceso de actualización de la matriz local consume relativamente mucho tiempo, ya que los procesos de lectura y escritura son procesos computacionales que se consideran "costosos", ya que son lentos si los comparamos con los procesos lógico-aritméticos.

Para solucionar el problema planteado, se consideró hacer uso de punteros como medio alternativo para actualizar las matrices locales luego de aplicar el esquema numérico. En "Fortran", el uso de punteros no supone mucha complicación, ya que basta con editar y agregar unas cuantas líneas para hacer uso de ellos. Como se observa en el código de la Figura 8, una vez declarados los punteros y las matrices locales B y C, sólo es necesario apuntar con el puntero pB a la matriz local B y con el puntero pC a la matriz C.

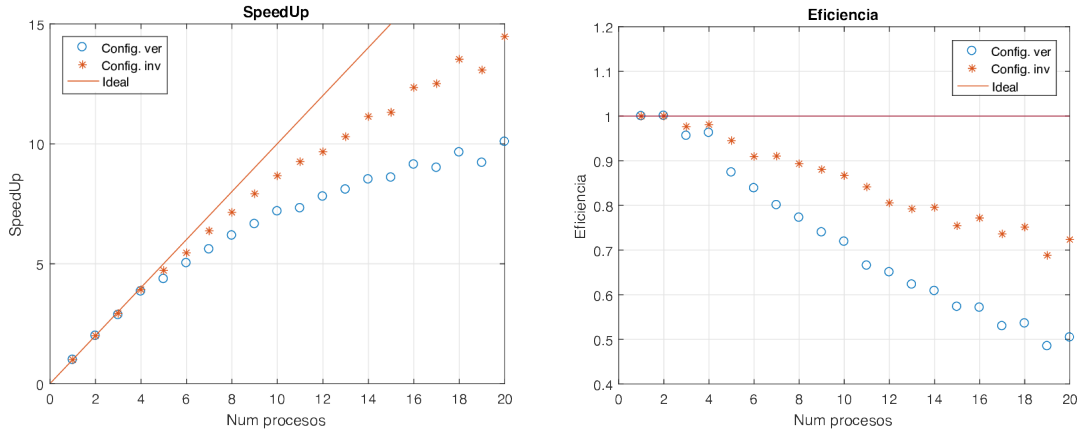


Figura 7: "Speed-up" y eficiencia de la configuración vertical usando diferente número de procesos y cambiando el orden de acceso a la memoria (versión invertida)

```
pB => B
pC => C
. . .
DO count=1,10
!Aplicando el esquema
do j=2,Mx+1
   do i=2,My+1
      pC(i,j) = (pB(i,j)+pB(i-1,j)+pB(i,j+1)+pB(i+1,j)+pB(i,j-1))/5.0d0
   enddo
enddo
! Actualizando matrices locales y re-direccionando punteros
pSwap => pB
pB => pC
pC => pSwap
. . .
```

Figura 8: Código de la implementación de punteros para actualizar matriz local.

En la sección del código en el que se aplica el esquema, es necesario remplazar el uso de las matrices locales B y C por sus respectivos punteros. De esta manera, la actualización de las matrices locales consistirá en hacer que los punteros pB y pC intercambien sus direcciones a las que apuntan. Para ello es necesario hacer que el puntero pB apunte a lo que apuntaba pC y que el puntero pC apunte a lo que apuntaba pB, para este proceso es necesario utilizar un puntero auxiliar, pSwap.

En la Tabla 1 se encuentran los mejores resultados obtenidos, los cuales se lograron con la configuración vertical y acceso a matriz con barridos horizontales. En la Figura 9 se observan los tiempos de ejecución y la eficiencia, en las que se aprecia el nivel de rendimiento obtenido de la configuración vertical con acceso a la matriz por barridos horizontales e implementando punteros. Notar que inclusive con 20 procesadores podemos obtener una eficiencia de 97 %.

Procesos	Tiempo(s)	Speed-up	Eficiencia
1	58.173	1.000	1.000
2	29.199	1.992	0.996
3	19.590	2.969	0.989
4	14.572	3.991	0.997
5	11.766	4.943	0.988
6	9.846	5.908	0.984
7	8.433	6.897	0.985
8	7.404	7.856	0.982
9	6.608	8.802	0.978
10	6.030	9.647	0.964
11	5.499	10.577	0.961
12	5.040	11.542	0.961
13	4.710	12.349	0.949
14	4.362	13.336	0.952
15	4.188	13.890	0.926
16	3.766	15.445	0.965
17	3.818	15.235	0.896
18	3.476	16.731	0.929
19	3.471	16.759	0.882
20	2.975	19.547	0.977

Tabla 1: Datos obtenidos con configuración vertical e implementando punteros.

4.2. Ejemplo 2: Solución de la ecuación de Poisson en paralelo

En este ejemplo se muestran los resultados obtenidos con el programa desarrollado para resolver sistemas lineales por el método de Jacobi provenientes de la ecuación de Poisson y el método de diferencias finitas aplicado a un dominio rectangular con mallado estructurado. Considérese la ecuación de Poisson en 2D dada por (1) donde la solución analítica y la función del lado derecho están dadas por

$$p(x,y) = \sin(2\pi x)\sin(2\pi y), \qquad \text{y} \qquad f(x,y) = -8\pi^2 \sin(2\pi x)\sin(2\pi y), \tag{11}$$

respectivamente. Con estas funciones se aplican condiciones de frontera Dirichlet nulas sobre el dominio dado por $\Omega = (0,1) \times (0,1)$.

Para esta prueba se eligió un mallado de dimensiones 362×362. Dicho tamaño se escogió pensando en generar un tiempo considerable de ejecución y una malla que tenga una gran cantidad de divisores de manera que, sin contar sus fronteras, es exactamente divisible entre la cantidad de procesos que se ejecuten en paralelo. Por último se escogió una tolerancia de 10^{-5} para determinar la condición de paro del método iterativo. En la Figura 10 se puede observar la solución exacta y numérica, así como los errores absolutos de este ejemplo.

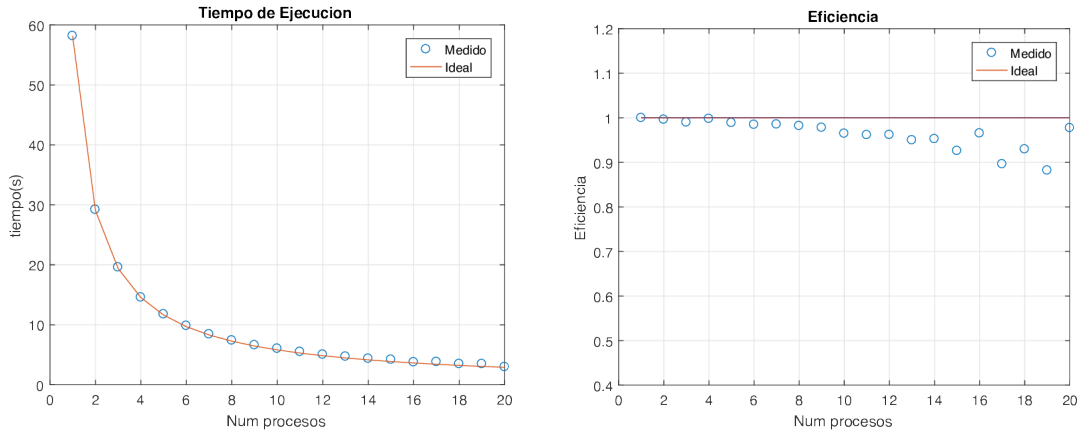


Figura 9: Tiempo de ejecución (en segundos) y eficiencia usando la configuración horizontal e implementando punteros con diferente número de procesos.

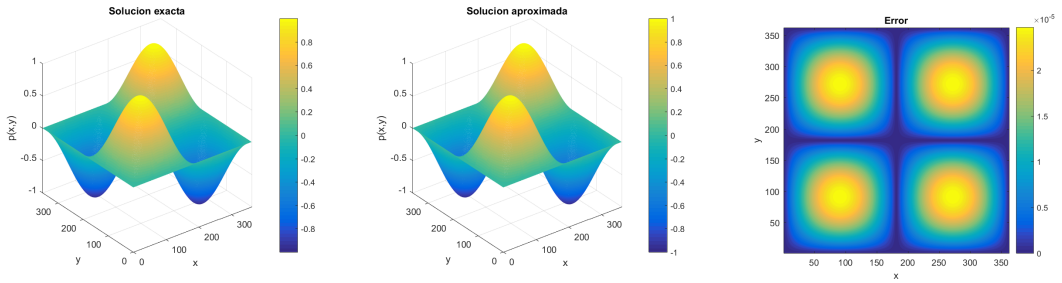


Figura 10: Solución exacta, solución numérica y error absoluto de la ecuación de Poisson en 2D.

El algoritmo propuesto resuelve de manera correcta el problema y otorga una solución precisa a la solución exacta.

A continuación se presentan los resultados obtenidos de los tiempos de ejecución logrados en las configuraciones con distintos número de procesos. Hay que recordar que la máxima eficiencia del programa se logra si la dimensión de la malla (sin fronteras) es divisible exacto entre el número de procesos en paralelo ejecutándose. El mallado de 362×362 es divisible entre 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 18 y 20. La Tabla 2 muestra la mediana de los tiempos de ejecución después de ejecutar varias veces el programa. Los resultados de "speed-up" y eficiencia también pueden ser observados en sus representaciones gráficas en la Figura 12.

Procesos	Tiempo(s)	Speed-up	Eficiencia
1	198.256	1.000	1.000
2	102.978	1.925	0.962
3	68.377	2.899	0.966
4	50.172	3.951	0.987
5	41.049	4.829	0.965
6	33.978	5.834	0.972
8	25.112	7.894	0.986
9	22.041	8.994	0.999
10	20.177	9.825	0.982
12	17.763	11.160	0.930
15	13.689	14.482	0.965
18	11.323	17.508	0.972
20	10.289	19.268	0.963

Tabla 2: Eficiencia de la aproximación de la ecuación de Poisson en 2D con una malla de 362×362

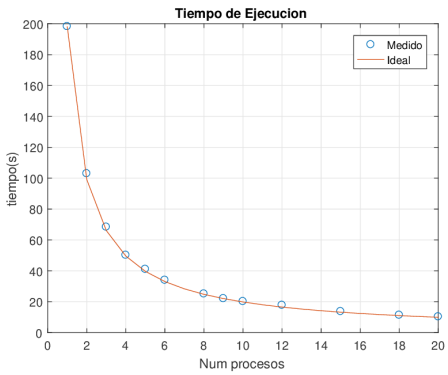


Figura 11: Tiempos de ejecución usando diferente número de procesos.

Como se puede observar, los resultados muestran altos niveles de eficiencia. Por ejemplo, la menor eficiencia del programa se obtiene usando 12 procesadores con un 93 %. Más aún, todas las configuraciones mantienen un nivel parecido de eficiencia, es decir, que no se logra apreciar una pérdida de rendimiento conforme más procesos se ejecuten. Lo anterior da como consecuencia que se logre hasta un "speed-up" de 19.2 con 20 procesos, muy cercano a lo ideal que sería 20. En tiempos de cómputo reales, una simulación en serie que duraba más de tres minutos, puede ser ahora obtenida en tan sólo 10 segundos con la misma precisión.

5 Conclusiones

En este trabajo se presentó el método iterativo de Jacobi en paralelo para resolver el sistema de ecuaciones lineales resultante de discretizar la ecuación de Poisson en 2D usando el método de diferencias finitas. Aunque este

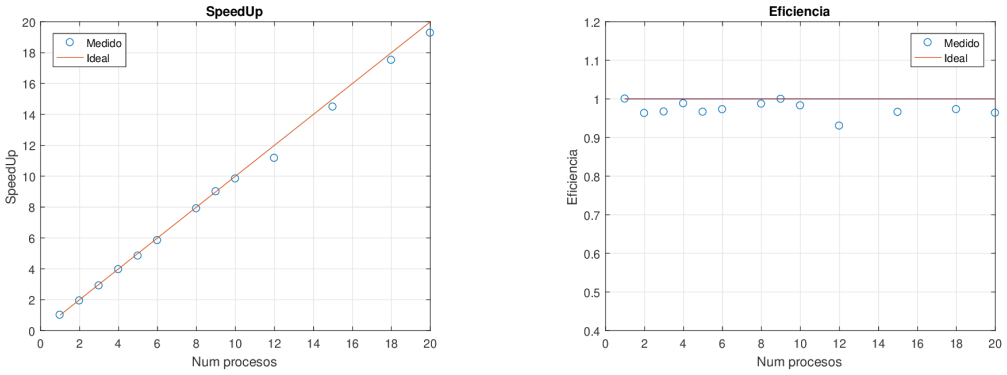


Figura 12: Rendimiento de la aproximación de la ecuación de Poisson en 2D empleando una malla de 362×362.

método tiene la desventaja de una lenta convergencia en comparación con otros métodos iterativos más avanzados, el tiempo de ejecución se compensa con la fácil implementación del algoritmo en paralelo. Se plantearon varias alternativas para implementar el algoritmo incluyendo diferentes opciones de acceso a la memoria. La configuración con pivotes mostró los resultados con mayor rendimiento. Se tuvo una eficiencia por arriba del 90 % y un "speed-up" de hasta 19.2 con 20 procesos en paralelo. En trabajos futuros se implementarán el mismo análisis para aproximaciones de la ecuación de Poisson en 2D usando volúmenes finitos y mallados triangulares no estructurados. También se implementará métodos iterativos con mayor velocidad de convergencia, lo cual ayudará a reducir aún más el tiempo de ejecución de las simulaciones numéricas.

Agradecimientos

Se agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo para la realización de este trabajo con el proyecto de "Investigadoras e Investigadores por México".

Referencias

[1] L. Abbas-Turki, S. Vialle, B. Lapeyre, and P. Mercier. Pricing derivatives on graphics processing units using monte carlo simulation. *Concurrency and Computation: Practice and Experience*, 26(9):1679 – 1697, 2012.

[2] G. T. Balls and P. Colella. A finite difference domain decomposition method using local corrections for the solution of poisson's equation. *Journal of Computational Physics*, 180(1):25 – 53, 2002.

[3] A. Basermann, B. Reichel, and C. Schelthoff. Preconditioned cg methods for sparse matrices on massively parallel machines. *Parallel Computing*, 23(3):381 – 398, 1997.

[4] D. J. Becker, T. Sterling, D. Savarese, J. E. Dorband, U. A. Ranawake, and C. V. Packer. Beowulf: A parallel workstation for scientific computation. *In Proceedings of the 24th International Conference on Parallel Processing*, pages 11 – 14, 1995.

[5] A. Bolis, C. D. Cantwell, D. Moxay, D. Serson, and S. J. Shwerwin. An adaptable parallel algorithm for the direct numerical simulation of incompressible turbulent flows using a fourier spectral/hp element method and mpi virtual topologies. *Computer Physics Communications*, 206:17 – 25, 2016.

[6] D. J. Evans. Parallel s.o.r. iterative methods. *Parallel Computing*, 1(1):3 – 18, 1984.

[7] P. Ghysels, T. J. Ashby, K. Meerbergen, and W. Vanroose. Hiding global communication latency in the gmres algorithm on massively parallel machines. *SIAM Journal on Scientific Computing*, 35(1):C48 – C71, 2013.

[8] D. Gordon and R. Gordon. Carp-cg: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated pdes. *Parallel Computing*, 36(9):495 – 515, 2010.

[9] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: portable parallel programming with the message-passing interface*, volume 1. MIT Press, 1999.

[10] M. M. Gupta and J. Kouatchou. Symbolic derivation of finite difference approximations for the three-dimensional poisson equation. *Numerical Methods for Partial Differential Equations*, 14(5):593 – 606, 1998.

[11] O. Lehmkuhl, C. D. Perez-Segarra, R. Borrell, M. Soria, and A. oliva. *Parallel Computational Fluid Dynamics*, volume 67, chapter TermoFluids: A new Parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC Cluster, pages 275 – 282. Springer, 2007.

[12] Y. Liu, B. Schmidt, and D. L. Maskell. Decgpu: distributed error correction on massively parallel graphics processing units using cuda and mpi. *BMC Bioinformatics*, 12(85):1 – 13, 2011.

[13] A. Nicholls and B. Honing. A rapid finite difference algorithm, utilizing successive over?relaxation to solve the poisson-boltzmann equation. *Journal of Computational Chemestry*, 12(4):435 – 445, 1991.

[14] Y. Notay and A. Napov. A massively parallel solver for discrete poisson-like problems. *Journal of Computational Physics*, 281(C):237 – 250, 2015.

[15] E. Schikuta. *MPI: A message-passing interface standard*. University of Tennessee, 1994.

[16] M. Smelyanskiy, D. Holmes, J. Chhugani, A. Larson, D. M. Carmean, D. Hanson, P. Dubey, K. Augustine, D. Kim, A. Kyker, V. W. Lee, A. D. Nguyen, L. Seiler, and R. Robb. Mapping high-fidelity volume rendering for medical imaging to cpu, gpu and many-core architectures. *IEEE Trans Vis Comput Graph*, 15(6):1563 – 1570, 2009.

[17] E. D. Sturler and H. A. van der Vorst. Reducing the effect of global communication in gmres(m) and cg on parallel distributed memory computers. *Applied Numerical Mathematics*, 18(4):441 – 459, 1995.

[18] T. Tang, W. Liu, and J. M. McDonough. Parallelization of linear iterative methods for solving the 3-d pressure poisson equation using various programming languages. *Procedia Engineering*, 61(1):136 – 143, 2013.

[19] W. Zahng, M. U. Zapata, X. Bai, D. P. V. Bang, and K. D. Nguyen. Three-dimensional simulation of horseshoe vortex and local scour around a vertical cylinder using an unstructured finite-volume technique. *International Journal of Sediment Research*, 35(3):295 – 306, 2020.

[20] M. U. Zapata, D. P. V. Bang, and K. D. Nguyen. Parallel simulations for a 2d x/z two-phase flow fluid-solid particle model. *Computers & Fluids*, 173(1):103 – 110, 2018.

[21] M. U. Zapata, W. Zahng, D. P. V. Bang, and K. D. Nguyen. A parallel second-order unstructured finite volume method for 3d free-surface flows using a σ coordinate. *Computers & Fluids*, 190:15 – 29, 2019.

La exclusión social de las personas mayores: una forma de mirar la pobreza

María del Pilar Alonso Reyes
Universidad Nacional Autónoma de México
sarpar@ciencias.unam.mx

Resumen

Exclusion allows reviewing and analyze situations of poverty, deprivation and inequality. Houghton and Khandker (2009) consider poverty as the deprivation of well-being in a pronounced way and the lack of adequate income to face education, health and safety needs; as well as access to the exercise of empowerment and basic rights. In order to identify the level of poverty, the objective is to review elements of social exclusion in the population and in particular of the elderly. Through two logistic models, which are supported by the information of the ENIGH 2016, which differ in number of variables and have or not as heads of older adults, it is sought to estimate the probability of being below the Welfare Line, in order to identify the level of poverty in which the population is located. The models show that households where older people are more susceptible to poverty.

Palabras Clave: Pobreza; Adulto mayor; Línea de bienestar-lb; Exclusión social

1 Introducción

La exclusión y la pobreza están ligadas por diversas razones, una de ellas es la propia estructura económica, porque ésta elimina a las personas de diversos ámbitos y con ello las excluye de las oportunidades para el desarrollo. Además, como mencionan Sabater y Giró (2015) la exclusión social es un proceso que no sólo afecta a grupos predeterminados concretos, sino que aqueja a personas y colectivos. La exclusión social adquiere diversas formas de manifestación en función de grado de vulnerabilidad y el riesgo que presenten tanto las personas como los colectivos, siempre transformándose en la forma de hacerlo, esto de acuerdo al grado de vulnerabilidad que presentan de acuerdo a los riesgos que enfrentan.

Subirats, *et al.* (2005) señalan que la exclusión social puede entenderse como una situación concreta resultado de la acumulación, superposición y/o combinación de diversos factores de desventaja, y genera entre las personas dificultad de acceder a los mecanismos de desarrollo personal, de inserción socio-comunitaria y sobre todo a los sistemas de protección social.

Tezanos (1999, citados en Sánchez *et. al*, 2013) agrega a la definición anterior, el que la exclusión social engloba situaciones que "apartan" a las personas de la sociedad, como por ejemplo la marginación en lo cultural o en la vivienda; la pobreza, en el ámbito económico; y generan enajenación social por consecuencia de procesos económicos, sociales que obstaculizan la manifestación de las capacidades en todas las áreas de desarrollo de las personas.

De acuerdo con Subirats (en Sabater, *et al.*, 2015) la exclusión social tiene siete dimensiones, las cuales indica que son: económica, laboral, formativa, socio-sanitaria, residencial, política y contexto espacial. Dichas dimensiones que se unen a ejes de desigualdad como el género, la edad, la raza/etnia y la clase social, (mencionado en Sabater, *et al.*, 2015). Además, la exclusión afecta a las personas en diversas áreas de su vida; dentro del ámbito social, donde los lazos se destruyen y no se desarrollan las relaciones; en lo económico, con salarios precarios,

limitación en el consumo y la generación de pobreza; en lo político, con desinterés hacia la participación política y sobre todo el impacto de la exclusión se aprecia en la educación, en la salud, la vivienda, en el trabajo, entre otras actividades de desarrollo humano.

De lo anterior se deduce que la pobreza se vincula con el aspecto económico de la exclusión social. Por su parte, a la *pobreza* se le conceptualiza como la carencia de recursos para cubrir las necesidades de subsistencia, no es permanente, y por lo general, empeora y vulnera la calidad de vida de las personas. La pobreza es dinámica, como lo es la sociedad, responde precisamente a los cambios que en ella se dan, afecta de acuerdo a diversos ámbitos de la vida e impacta diferencialmente en función de las necesidades del mercado y sobre todo responde a la exclusión social, que en la sociedad se presenta hacia las personas de diversas edades y sexo.

La pobreza en las sociedades es estructural, es decir, responde al conjunto de las relaciones de producción que generan las desigualdades en la sociedad, que además se une a las características de escasez de recursos y la forma precaria del empleo. Cabe señalar que la precariedad en el empleo se asocia con la inseguridad, la incertidumbre y la falta de garantías económicas y sociales mínimas que garanticen la supervivencia de la persona o de la familia.

El enfoque económico, es la concepción que se está dando a la pobreza en este trabajo, se caracteriza por bajos niveles de salario o renta, muy poco consumo o gasto; pero en el mismo tenor, la pobreza también está relacionada con la falta de capacidades individuales para alcanzar un nivel básico de bienestar humanos, (Prialé, s.f.).

Hay que señalar que en este binomio exclusión-pobreza, también interviene un tercer elemento que es la inclusión o inserción, conceptualizada como una serie de medidas políticas y de movilización de recursos para organizar la incorporación de personas, en este caso, al acceso de recursos.

La exclusión es concebida por la inclusión, porque en su carácter multidimensional la exclusión genera situaciones complejas debido a las condiciones económicas adversas a los individuos; pero también se expresa en desigualdades sociales debido a las pocas oportunidades económicas. De esta forma el desempleo, ingresos reducidos, generan situaciones críticas en la familia que las conduce a la ausencia de redes o a combinaciones de riesgos que llevan a la ruina a la familia.

García, Malo y Rodríguez (en Pérez Moya, 2006) indican que se dan siete ámbitos donde las personas se mueven de acuerdo a exclusión-pobreza-inclusión:

1. La erosión de las redes sociales;
2. Pobreza integrada[1], donde las personas tienen bajos ingresos, pero redes sociales sólidas;
3. Pobreza económica con erosión en las redes sociales no familiares;
4. La exclusión social, las personas sobreviven gracias a la economía sumergida[2] y tienen problemas en las redes sociales familiares;
5. La exclusión social severa en la que sobreviven gracias a la mendicidad y a actos delictivos.
6. La marginación y muerte social de los individuos.

Las sociedades actuales siempre presentan un grado de exclusión, la cual se da en diversos niveles de afección, Arroyo (2016) señala que la exclusión se presenta en tres niveles de acuerdo con la gravedad de la situación de las personas y el cómo éstas se recuperan, es decir, el grado de resiliencia. El primer nivel se muestra cuando la persona se encuentra en pobreza extrema[3], donde la vida de la persona está en peligro; el segundo grado es cuando las personas, aunque pueden provenir de la pobreza extrema, pueden corregir algunas características de

[1]La pobreza integrada es la considerada como la que sufren las personas de forma endémica; se da en países poco desarrollados. Las personas que están en esta categoría no están estigmatizadas y se piensa que con el desarrollo económico desaparece esta pobreza.

[2]Esta economía hace referencia a la economía ilegal, aunque también se incluye la economía informal.

[3]Estar en pobreza extrema, según CONEVAL, es cuando tres de seis derechos sociales se incumplen (rezago educativo, carencias de alimentación, de servicios de salud, de calidad y espacios de la vivienda, de servicios básicos de la vivienda o de seguridad social) y además, las personas tienen los ingresos por familia son menores a $1,060.24 en zonas rurales o 1,477.31 en las urbanas (a precios de agosto de 2017).

la pobreza, se diferencia de la de primer grado en la gravedad y en magnitud del riesgo para la supervivencia; de acuerdo con Arrollo (2016) en este nivel se incluyen a los ancianos, los desempleados, los indígenas, los discapacitados, los enfermos de SIDA, inmigrantes pobres, etcétera. Por último, el tercer nivel es la exclusión en donde la actitud y acciones del individuo rompen las reglas sociales, morales o legales, incluyen en este caso a los reclusos, prostitutas, alcohólicos, homosexuales, etcétera.

Estos niveles de exclusión se dan más por la vulnerabilidad que ciertos grupos poblacionales presentan, Hernández (2008) señala que en los diversos ámbitos de la vida se tienen poblaciones que se vuelven más susceptibles a la exclusión: a) en el ámbito económico: personas en pobreza y precariedad relativa; b) en lo laboral: los desempleados y aquellas personas que por sus capacidades les es difícil ingresar al mercado laboral; c) en la parte formativa: las personas analfabetas o aquellas que tienen un nivel básico de instrucción, las personas discapacitadas y a las que el sistema sanitario las vulnera; d) en el ámbito socio-sanitario: las personas drogodependientes y con enfermedades relacionadas; e) en el ámbito relacional, socio-sanitario y económico: las personas mayores; y f) en los ámbitos relacional, residencial, económico y ciudadanía: personas sin casa y las que no pueden cubrir sus necesidades básicas.

Por ello estudiar la exclusión y la pobreza en una población específica, resulta importante para determinar algunos factores que pudieran incidir en reducir o, idealmente, eliminar estas vulnerabilidades, y sobre todo para visibilizar una población que necesita el actuar de políticas públicas orientadas a la dignificación de su vida para reducir los dos niveles de coexistencia pobreza y exclusión social.

2 Material y Método

Para este trabajo se utilizó la tabla de información llamada "concentradohogar" que tiene CONEVAL vinculada a la Encuesta Nacional de Ingreso y Gasto en Hogares 2016. Esta tabla tiene 127 variables que se tomaron de las otras tablas de la base de datos de la ENIGH 2016 y que el propio INEGI integra. Se concentra por hogar, ingresos y gastos en toda modalidad posible, se emiten resultados trimestrales.

La variable dependiente fue el ingreso corriente que es considerado como la suma de los ingresos por trabajo, los provenientes de rentas, de transferencias, de estimación del alquiler y de otros ingresos. No se tomó tal cual, fue transformada considerando la Línea de Bienestar (LB), que CONEVAL propone para la medición de la pobreza. Este consejo mide la pobreza a través de dos líneas de bienestar, la mínima que equivale al valor de la canasta alimentaria por persona al mes, y la línea de bienestar, que equivale al valor total de la canasta alimentaria y de la canasta no alimentaria por persona al mes.

La variable de ingreso corriente, fue dividida entre 3 para tener el valor mensual, luego este valor se dividió entre el número de integrantes por hogar para obtener el ingreso de cada miembro de cada hogar. Para considerar esta LB se tomó el valor de $2,686.2, con él se siguió el siguiente algoritmo, este valor de la canasta es referente a peso de 2016 y el que da el porcentaje de pobreza que CONEVAL calcula:

$$\text{Y_ linbie} = \begin{cases} 1 \text{ si ingreso corriente } \leq \$2,686.2 \\ 0 \text{ si ingreso corriente } > \$2,686.2 \end{cases}$$

Es así que se clasifican los hogares de acuerdo a los que están debajo de la línea de bienestar y los que se sitúan por arriba de la misma.

A continuación, en el cuadro se muestran las variables que se escogieron de la tabla "concentradohogar" y que se usarán en el modelo logístico para explicar la probabilidad de caer en pobreza. Las variables independientes consideradas fueron:

Nombre de la variable	Descripción
EST_SOCIO	Estrato socioeconómico que se refiere a la clasificación de las viviendas del país de acuerdo a ciertas características socioeconómicas de las personas que las habitan, así como características físicas y el equipamiento de las mismas expresadas por medio de 24 indicadores construidos con información del Censo de Población y Vivienda 2010. Los códigos corresponden a: 1 Bajo, 2 Medio bajo, 3 Medio alto y 4 Alto
CLASE_HOGAR	Clase hogar que corresponde a la diferenciación de los hogares a partir del tipo de relación consanguínea, legal, de afinidad o de costumbre entre el jefe(a) y los otros integrantes del hogar, sin considerar a los trabajadores domésticos y a los familiares de éstos ni a los huéspedes. Se clasifican en: 1 unipersonal: hogar formado por una sola persona que es el jefe(a); 2 nuclear: hogar constituido por un solo grupo familiar primario; 3 ampliado: hogar formado por el jefe(a) y su grupo familiar primario más otros grupos familiares u otros parientes; 4 compuesto: hogar formado por un hogar nuclear o ampliado con personas sin parentesco con el jefe(a); 5 corresidente: hogar formado por dos o más personas que no tienen parentesco con el jefe(a).
SEXO_JEFE	Distinción biológica que clasifica al jefe del hogar en hombre o mujer. 1 Hombre y 2 Mujer.
EDAD_JEFE	Años transcurridos entre la fecha de nacimiento del jefe del hogar y la fecha de la entrevista.
EDUCA_JEFE	Educación formal del jefe del hogar. Valor etiqueta: 01 Sin instrucción, 02 Preescolar, 03 Primaria incompleta, 04 Primaria completa, 05 Secundaria incompleta, 06 Secundaria completa, 07 Preparatoria incompleta, 08 Preparatoria completa, 09 Profesional incompleta, 10 Profesional completa, 11 Posgrado.
TOT_INTEG	Número de personas pertenecientes a este hogar, sin considerar a los trabajadores domésticos y a los familiares de éstos ni a los huéspedes.
P65MAS	Integrantes del hogar que tienen 65 o más años de edad.
PERC_OCUP	Número de personas perceptoras ocupadas que tienen trabajo y reciben ingreso corriente monetario.
PERCEP_ING	Número de personas que perciben ingreso corriente monetario
TRANSFER	Las transferencias son las entradas en efectivo o en especie recibidas por los integrantes del hogar y por las cuales el proveedor o donante no demanda retribución de ninguna naturaleza
ESTIMA_ALQ	Estimación del alquiler. Es el valor estimado del alquiler que habría de pagar en el mercado por contar con un alojamiento del mismo tamaño, calidad y ubicación. La realiza el propio informante con base en su apreciación del valor de mercado de la renta de su vivienda.
GASTO_MON	Gasto corriente monetario. Es la suma de los gastos regulares que directamente hacen los hogares en bienes y servicios para su consumo.
MENORES	Integrantes menores. Integrantes del hogar que tienen 11 o menos ¡12 años de edad.

Para el modelo matemático se implementó un diseño logístico, el cual se basa en encontrar una regresión: $E\left(y|x_i\right) = \beta_0 + \beta^{'}{}_1 x_i$ para $X = x_i$, se considera p_i la probabilidad cuando $X = x_i$ toma el valor 1, esto es:

$$E\left(y|x_i\right) = P\left(y = 1|x_i\right) * 1 + P\left(y = 0|x_i\right) * 0 = p_i = \beta_0 + \beta^{'}{}_1 x_i$$

Esto indica que la predicción de y_i estima la probabilidad de que un individuo con características definidas por $X = x_i$ pertenezca a la población correspondiente a $y = 1$.

Es importante garantizar que la respuesta prevista esté entre cero y uno. Esto se logra cuando $p_i = F\left(\beta_0 + \beta^{'}{}_1 x_i\right)$, y obligando que F tenga esa propiedad, para lo cual se propone a F como la función logística: $p_i = \frac{1}{1+e^{-\left(\beta_0+\beta^{'}{}_1 x_i\right)}}$

, considerando $1 - p_i$, tomando el cociente $\frac{p_i}{1-p_i}$ y renombrando al logaritmo de este cociente como g, el modelo queda:

$$g_i = log\left(\frac{p_i}{1-p_i}\right) = log\left(\frac{1}{e^{-\left(\beta_0+\beta'_i x_i\right)}}\right) = \beta_0 + \beta'_i x_i$$

A este modelo lineal se le llama logit, la variable g representa en una escala logarítmica la diferencia entre las probabilidades de pertenecer a ambas poblaciones.

Puede utilizarse como parámetros e^{β_0} y e^{β_1}, llamados cociente de momios o cociente de probabilidades. Estos valores indican cuánto se modifican las probabilidades por unidad de cambio en las variables X. De esta forma el modelo queda: $O_i = \frac{p_i}{1-p_i} = e^{\beta_0} e^{\beta_1 x_i}$; extendiendo para el conjunto de las variables x, el modelo se expresa como: $O_i = e^{\beta_0} \prod_{j=1}^{k} \left(e^{\beta_j}\right)^{x_j}$.

Con este modelo logístico se medirá la probabilidad de estar por abajo de la LB, ya que es el valor de 1 que se le dio a la variable dependiente en la clasificación de la tabla de datos de la ENIGH 2016.

3 Resultados

3.1. Modelo general

No. de observaciones: 65, 917; $LR\chi^2_{(10)} = 41,488.93$; Prob $> \chi^2 = 0.0000001$; Log(verosimilitud) $= -24,940.45$; Pseudo$R^2 = 0.4541$.

Y_LB 2016	β	Cociente de momios	Error estándar	z	P>|z|	Intervalo de confianza al 95%	
EST_SOCIO	-0.503	0.605	0.010	-28.97	0.000	0.585	0.626
CLASE_HOGAR	0.193	1.213	0.029	8.18	0.000	1.158	1.271
SEXO_JEFE	0.237	1.267	0.033	8.96	0.000	1.203	1.334
EDAD_JEFE	-0.003	0.997	0.001	-2.78	0.000	0.995	0.999
EDUCA_JEFE	-0.103	0.902	0.006	-16.35	0.000	0.892	0.914
TOT_INTEG	1.124	3.077	0.041	83.52	0.000	2.996	3.159
P65MAS	0.207	1.230	0.033	7.55	0.000	1.166	1.298
PERCEP_ING	0.263	1.301	0.021	16.17	0.000	1.260	1.343
PERC_OCUPA	-1.133	0.322	0.006	-59.41	0.000	0.310	0.334
TRANSFER	-0.00013	0.99987	0.000002	-51.68	0.000	0.99986	0.99987
ESTIMA_ALQU	-0.00028	0.99972	0.000005	-51.35	0.000	0.99971	0.99973
GASTO_MON	-0.00008	0.99992	0.000001	-68.39	0.000	0.99991	0.99992
CONSTANTE	1.406	4.080	0.342	16.76	0.000	3.462	4.809

Tabla 1: Resultados del modelo logístico, considerando 65,917 hogares y 12 variables independientes
Fuente: Obtenida por los autores usando la tabla concentrado hogar de CONEVAL y a través del paquete estadístico STATA 12.

$$\begin{aligned}\tilde{O}_i = {} & 4.08 * 0.605^{EST_SOCIO} * 1.231^{CLASE_HOGAR} * 1.267^{SEXO_JEFE} * 0.997^{EDAD_JEFE} * \\ & 0.902^{EDUCA_JEFE} * 3.077^{TOT_INTEG} * 1.23^{P65MAS} * 1.301^{PERCEP_ING} * 0.322^{PERC_OCUPA} \\ & * 0.9998^{TRANSFER} * 0.99997^{ESTIMA_ALQU} * 0.99999^{GASTO_MON}\end{aligned}$$

Este primer modelo señala que para estar por abajo o por arriba de la línea de bienestar, las 12 variables consideradas en el modelo son altamente significantes, ya que el valor $P > |z|$ es cero, lo que señala que las 12 variables explican la probabilidad de caer por abajo o encima de la línea de bienestar. Los cocientes de momios de la tabla 1 cuantifican la participación de cada variable a la probabilidad de caer por abajo o no de la línea de bienestar.

Variables independientes	β	Cociente de momios	Interpretación
EST_SOCIO	-0.503	0.605	La variable estrato socioeconómico tiene un coeficiente negativo, lo que indica que disminuye la probabilidad de estar por debajo de la LB, por cada nivel de cambio en el estrato socioeconómico, hacia arriba, la disminución es del orden de 0.5.
CLASE_HOGAR	0.193	1.213	La clase de hogar tiene signo positivo en su coeficiente, por lo que aumenta la probabilidad de caer por debajo de la LB, por cada unidad que aumente en la clase de hogar el logit de estar por debajo de LB aumenta 0.193.
SEXO_JEFE	0.237	1.267	La variable sexo del jefe del hogar aumenta (valor positivo del coeficiente) el logit de estar por abajo de la LB. Cambiar de jefe hombre a jefa mujer un 26 % la probabilidad de estar por debajo de la LB.
EDAD_JEFE	-0.003	0.997	Esta variable es negativa en β por lo que influye disminuyendo la probabilidad de estar por debajo de la LB.
EDUCA_JEFE	-0.103	0.902	El nivel educativo del jefe de familia presenta en β un signo negativo lo que influencia disminuyendo la probabilidad de estar por debajo de la LB.
TOT_INTEG	1.124	3.077	El total de integrantes influye de manera positiva, es decir, incrementa la probabilidad de estar por debajo de la LB. Así el riesgo de estar en pobreza se triplica por cada una persona que aumente en el hogar.
P65MAS	0.207	1.230	La población de 65 años o más influye de manera positiva, aumenta la probabilidad de caer por debajo de la LB, así por cada persona que aumente en el hogar de esta edad, el riesgo de caer por debajo de la LB es 0.23.
PERCEP_ING	0.263	1.301	El número de perceptores de ingresos en el hogar influye de manera positiva, así que por cada aumento de un perceptor en el hogar aumenta la probabilidad de estar por debajo de la LB.
PERC_OCUPA	-1.133	0.322	El número de perceptores ocupados disminuye la probabilidad de estar por abajo de la LB.
TRANSFER	-0.00013	0.99987	Las transferencias de recursos influyen de manera inversa a la probabilidad de estar por abajo de la LB, pero como su coeficiente es casi cero, deja igual la situación de estar por arriba o por debajo de la LB.
ESTIMA_ALQU	-0.00028	0.99972	La estimación del gasto en el alquiler de vivienda es negativa por lo que disminuye la probabilidad de estar por debajo de la LB.
GASTO_MON	-0.00008	0.99992	La influencia de esta variable es inversa, así que, a un aumento del gasto monetario, disminuye la probabilidad de caer por debajo de la línea de bienestar.

Tabla 2: Interpretación de cada variable al modelo general

3.2. Modelo donde la jefatura es de AM (60 y más años) y los hogares tienen al menos 1 AM

No. de observaciones $= 12,528$; $LR\chi^2_{(9)} = 8,421.47$; Prob $> \chi^2 = 0.0000$;
Log(verosimilitud) $= -4,451.14$; Pseudo$R^2 = 0.4861$.

Y_LB 2016	β	Cociente de momios	Error estándar	z	P> |z|	Intervalo de confianza al 95%	
EST_SOCIO	-0.546	0.579	0.025	-12.56	0.000	0.532	0.631
CLASE_HOGAR	0.722	2.058	0.115	12.93	0.000	1.845	2.296
EDUCA_JEFE	-0.078	0.925	0.015	-4.79	0.000	0.896	0.955
TOT_INTEG	1.026	2.790	0.138	20.72	0.000	2.532	3.075
P65MAS	0.613	1.846	0.113	10.05	0.000	1.638	2.081
PERCEP_ING	0.131	1.140	0.059	2.55	0.011	1.031	1.261
PERC_OCUPA	-1.127	0.324	0.014	-25.12	0.000	0.297	0.354
TRANSFER	-0.0002	0.99980	0.000006	-33.63	0.000	0.99979	0.99981
ESTIMA_ALQU	-0.0004	0.99960	0.000137	-25.34	0.000	0.99962	0.99968
GASTO_MON	-0.0001	0.99990	0.000004	-25.14	0.000	0.99990	0.99991
MENORES	0.253	1.288	0.094	3.48	0.001	1.117	1.486
CONSTANTE	1.198	3.314	0.411	9.66	0.000	2.599	4.225

Tabla 3: Resultados del modelo logístico, considerando 12,528 hogares cuyo jefe de hogar es de 60 o más años y al menos hay una persona mayor en el hogar, se consideran 9 variables independientes, ya que PERCEP_ING no contribuye al modelo.
Fuente: Obtenida por los autores usando la tabla concentrado hogar de CONEVAL y a través del paquete estadístico STATA 12.

$$\begin{aligned}\hat{\mathrm{O}}_i =& 3.314 * 0.579^{EST_SOCIO} * 2.058^{CLASE_HOGAR} * 0.925^{EDUCA_JEFE} * 0.138^{TOT_INTEG} * \\ & 1.846^{P65MAS} * 0.14^{PERCEP_ING} * 0.324^{PERC_OCUPA} * 0.9998^{TRANSFER} * 0.9996^{ESTIMA_ALQU} * \\ & 0.9999^{GASTO_MON} * 1.288^{MENORES}\end{aligned}$$

$$\hat{\mathrm{p}}_i = \frac{\hat{\mathrm{O}}_i}{1 + \hat{\mathrm{O}}_i}$$

Este modelo señala que para estar por abajo o por arriba de la línea de bienestar, las 11 variables consideradas son altamente significantes, ya que el valor $P > |z|$ es cero. Los cocientes de momios de la tabla 4 cuantifican la participación de cada variable a la probabilidad de caer por abajo o no de la línea de bienestar.

Variables independientes	β	Cociente de momios	Interpretación
EST_SOCIO	-0.546	0.579	La variable estrato socioeconómico presenta un coeficiente negativo lo que significa que al aumentar de estrato la probabilidad de estar por debajo de la LB disminuye.
CLASE_HOGAR	0.722	2.058	La variable clase de hogar influye de manera directa sobre estar debajo de la LB, es decir, al aumentar el tipo de clase de hogar aumenta la probabilidad de caer debajo de LB. Así el riesgo se duplica al cambio del tipo de hogar.
EDUCA_JEFE	-0.078	0.925	La educación del jefe de familia influye de manera negativa, es decir, al aumentar el nivel educativo la probabilidad de estar debajo de la LB disminuye.
TOT_INTEG	1.026	2.790	El total de integrantes es una variable de influencia positiva, al aumentar un integrante en la familia la probabilidad de estar por debajo de la LB aumenta y el riesgo es del doble.
P65MAS	0.613	1.846	Al aumentar una persona de 65 años o más en el hogar aumenta la probabilidad de estar por debajo de la LB.
PERCEP_ING	0.131	1.140	Si aumentan los perceptores de ingresos aumenta la probabilidad de estar debajo de la LB
PERC_OCUPA	-1.127	0.324	Si aumentan los perceptores ocupados disminuye la probabilidad de estar por debajo de la LB.
TRANSFER	-0.0002	0.99980	Las transferencias tienen una influencia inversa en la probabilidad de estar por debajo de LB, así si aumentan las transferencias, disminuye la posibilidad de estar debajo de dicha línea.
ESTIMA_ALQU	-0.0004	0.99960	La estimación del gasto en alquiler influye de manera inversa a la posibilidad de estar por debajo de la LB.
GASTO_MON	-0.0001	0.99990	El gasto monetario tiene un comportamiento parecido a la estimación del gasto en alquiler como de las transferencias.
MENORES	0.253	1.288	La variable "menores" al tener un coeficiente positivo influye directamente en la probabilidad de caer por debajo de la LB, así al aumentar un menor en el hogar aumenta la probabilidad de estar por debajo de la LB.

Tabla 4: Interpretación de cada variable al modelo jefatura mayor o igual a 60 años y al menos un AD en el hogar

3.3. Comparación de parámetros de los dos modelos

Al obtener los dos modelos, a simple vista parecen semejantes, lo que implicaría que no hay diferencias entre tener jefatura joven o vieja y considerar hogares de adultos mayores exclusivamente o compuestos. Es de esta forma en la tabla 5 se muestra estas diferencias o semejanzas, cabe resaltar que el tener en el hogar adultos

mayores es importante para estar por debajo de la LB, pero es más probable estar por debajo de dicha línea en los hogares de adultos mayores y cuando hay más de uno de ellos en él. Los gastos y las transferencias son importantes en el hogar, en ambos modelos logísticos, pero no determinan estar por arriba o por debajo de la LB, porque los cocientes de momios son materialmente uno.

Variables independientes	Modelo jefe 65 y +, con al menos un AM		Modelo general, todos los hogares		Diferencias o similitudes
	Coeficiente β	Cociente de momios	Coeficiente β	Cociente de momios	
EST_SOCIO	0.546	0.579	-0.503	0.605	En ambos modelos el signo de β es negativo, influencia de manera inversa por estar debajo de la LB. Los cocientes de momios son parecidos.
CLASE_HOGAR	0.722	2.058	0.193	1.213	Tiene β signo positivo para ambos modelos, pero el riesgo de estar en pobreza es mayor en el modelo de jefe de 65 años y más.
EDUCA_JEFE	-0.078	0.925	-0.103	0.902	En ambos modelos el signo de β es negativo, influencia de manera inversa por estar debajo de la LB. Los cocientes de momios son parecidos.
TOT_INTEG	1.026	2.790	1.124	3.077	Tiene β signo positivo para ambos modelos, pero el riesgo de estar en pobreza es mayor en el modelo general.
P65MAS	0.613	1.846	0.207	1.230	Tiene β signo positivo para ambos modelos, pero el riesgo de estar en pobreza es mayor en el modelo de jefe de 65 años y más.
PERCEP_ING	0.131	1.140	0.263	1.301	Tiene β signo positivo para ambos modelos, pero el riesgo de estar en pobreza es mayor en el modelo general.
PERC_OCUPA	-1.127	0.324	-1.133	0.322	En ambos modelos el signo de β es negativo, influencia de manera inversa por estar debajo de la LB. Los cocientes de momios son parecidos.
TRANSFER	-0.0002	0.9998	-0.00013	0.99987	En ambos modelos el signo de β es negativo, influencia de manera inversa por estar debajo de la LB. Los cocientes de momios son parecidos.
ESTIMA_ALQU	-0.0004	0.9996	-0.00028	0.99972	En ambos modelos el signo de β es negativo, influencia de manera inversa por estar debajo de la LB. Los cocientes de momios son parecidos.
GASTO_MON	-0.0001	0.9999	-0.00008	0.99992	En ambos modelos el signo de β es negativo, influencia de manera inversa por estar debajo de la LB. Los cocientes de momios son parecidos.

Tabla 5: Comparaciones de los signos y valores de los parámetros en los dos modelos logísticos

3.4. Ejemplos de probabilidades

En la tabla 6 se muestran ejemplos usando el modelo general para calcular probabilidades de quedar debajo de la LB, así los dos primeros muestran un jefe de familia joven, pero uno es hombre y el otro es mujer, con un adulto mayor en el hogar, cuatro integrantes, un perceptor de ingresos, con $3000 de gastos y con $500 de transferencias, $2000 de gasto de alquiler de vivienda y estrato socioeconómico bajo; se observa que en el caso de la jefatura masculina, la probabilidad de estar por abajo de la LB es de 0.9743, mientras que si la jefatura es femenina esa misma probabilidad es 0.9796. Al cambiar la edad del jefe de familia (65 años de edad) y considerando los mismos supuestos de los ejemplos anteriores, la probabilidad de estar por arriba de la LB con la característica de que el jefe de familia es hombre de 65 años es 0.9720; mientras que, si es mujer de 65 años la jefa de familia, la probabilidad es .9778. Cuando se aumenta el nivel escolar del jefe de familia, se sube de estrato socioeconómico, se tiene propiedad de la vivienda y se disminuyen los integrantes de los hogares las probabilidades de caer por debajo de la LB disminuyen considerablemente.

Y_ LB 2016 Modelo general	x_j A	x_j B	x_j C	x_j D	x_j E	x_j F	x_j G	x_j H
EST_SOCIO	1	1	1	1	3	3	3	3
CLASE_HOGAR	2	2	2	2	2	2	2	2
SEXO_JEFE	0	1	0	1	0	1	0	1
EDAD_JEFE	35	35	65	65	35	35	65	65
EDUCA_JEFE	5	5	5	5	9	9	9	9
TOT_INTEG	4	4	4	4	2	2	2	2
P65MAS	1	1	1	1	0	0	1	1
PERCEP_ING	1	1	1	1	1	1	1	1
PERC_OCUPA	1	1	1	1	1	1	1	1
TRANSFER	500	500	500	500	0	0	500	500
ESTIMA_ALQU	2000	2000	2000	2000	0	0	0	0
GASTO_MON	3000	3000	3000	3000	3000	3000	3000	3000
CONSTANTE	1	1	1	1	1	1	1	1
Probabilidad de estar por debajo de la LB	0.9743	0.9796	0.9720	0.9778	0.5961	0.6515	0.6085	0.6632

Tabla 6: Ejemplos de cálculos de probabilidades de estar por abajo de la LB del modelo general

En la tabla 7 se muestran 4 ejemplos del modelo donde sólo se consideró hogares donde el jefe de familia tiene 60 o más años y en el hogar existe al menos una persona mayor; así cuando se tienen las mismas características del modelo general donde el jefe de familia es hombre de 60 años y más, variación de integrantes del hogar, con un o dos adultos mayores, gastos de $3000 y $500 o $1000 de transferencias y tener o no propiedad de la vivienda, la probabilidad de estar por debajo de la LB es en promedio mayor que en los hogares con jefe joven.

Y_ LB 2016 Modelo general	x_j A	x_j B	x_j C	x_j D	x_j E	x_j F	x_j G	x_j H
EST_SOCIO	1	1	1	1	3	3	3	3
CLASE_HOGAR	2	2	2	2	2	2	2	2
SEXO_JEFE	0	1	0	1	0	1	0	1
EDAD_JEFE	35	35	65	65	35	35	65	65
EDUCA_JEFE	5	5	5	5	9	9	9	9
TOT_INTEG	4	4	4	4	2	2	2	2
P65MAS	1	1	1	1	0	0	1	1
PERCEP_ING	1	1	1	1	1	1	1	1
PERC_OCUPA	1	1	1	1	1	1	1	1
TRANSFER	500	500	500	500	0	0	500	500
ESTIMA_ALQU	2000	2000	2000	2000	0	0	0	0
GASTO_MON	3000	3000	3000	3000	3000	3000	3000	3000
CONSTANTE	1	1	1	1	1	1	1	1
Probabilidad de estar por debajo de la LB	0.9743	0.9796	0.9720	0.9778	0.5961	0.6515	0.6085	0.6632

Tabla 7: Ejemplos de cálculos de probabilidades de estar por debajo de la LB del modelo con jefatura de personas con 60 años y más y en el hogar al menos un adulto mayor

4 Conclusión

El binomio exclusión-pobreza de la población adulta mayor que Hernández (2008) clasifica en el ámbito de la exclusión en relacional, socio-sanitario y económico, es decir, tres áreas que hacen más vulnerables a esta población. Así en el ámbito relacional intervienen el deterioro de las redes familiares (donde hay cuadros de violencia o de conflictos), escasez de las redes (hogares de una sola persona), rechazo o estigmatización social (en la vejez la incapacidad, la dependencia); en el área socio-sanitaria que implica el no acceso al sistema, adicciones y enfermedades de alto costo, enfermedades infecciosas, enfermedades mentales, crónicas, degenerativas y las que provocan gran dependencia; por último el ámbito económico que trae consigo la propia pobreza, las dificultades financieras, la dependencia a las prestaciones sociales o bien la desprotección social.

Este binomio pobreza-exclusión se sustenta en todas las poblaciones en lo cultural, económico y social, donde los ejes de la desigualdad son el género, la edad, la etnia y la clase social. En este trabajo revisar la exclusión por edad es importante porque en las personas mayores de antemano se establece una desigualdad en la sociedad, cuando ya no es posible contratar a las personas en el mercado laboral y por tanto son excluidos de cualquier beneficio económico que implica el trabajo. Por otra parte, a determinada edad, el abandono, la orfandad de amigos y familiares sitúan a las personas mayores en un aislamiento que los conduce a ser una población frágil y vulnerable, si además se junta que en edades mayores a los 70 las enfermedades de alto impacto económico son recurrentes y son el pan nuestro de cada día, la segregación de estas personas es mayor.

Arroyo (2016) señala que además a lo anterior hay que agregar que la exclusión social tiene mucho que ver con los territorios, donde condiciones como las ambientales, las sociales, las culturales y hasta las políticas influyen en que los grupos sociales sean excluidos de manera diferenciada, no es lo mismo ser adulto mayor urbano con familiares y redes de amigos, que ser indígena adulto mayor, monolingüe, sin familiares y vivir en comunidades aisladas.

Estudios muestran que hay variables de exclusión que están fuertemente relacionadas con la pobreza, la dinámica del mercado donde la edad es una variable importante para discriminar a las personas; la rotación en el empleo debido a las características macroeconómicas o bien por el género, donde las mujeres por su papel de cuidadoras tienen que abandonar los empleos, el número de hijos en los hogares puede llevar a caer en un círculo vicioso de la pobreza y sobre todo residir en zonas rurales, media urbana o urbana puede hacer la diferencia en la pobreza.

El PNUD (2016) señala que aunado a lo anterior las trayectorias familiares son esenciales para la exclusión-pobreza, así como los problemas de salud; pone especial cuidado en la población indígena o afrodescendiente, donde la probabilidad de salir de la pobreza de ingresos está sujeta a esa condición étnica – racial.

Tener bienes duraderos que sirven para tener efectivo ante las contingencias o como sistema de protección social permite a las familias salir de la pobreza; al igual si el jefe de familia tiene empleo formal. PNUD (2016) agrega, que no sólo el problema económico incide en la pobreza, factores demográficos y educativos son muy importantes, ya que pueden generar grandes desigualdades que no les permite a las personas acceder a un mejor trabajo o a que éste se encuentre en el mercado formal. El empleo formal genera mecanismos de protección social que evitan un retroceso en el bienestar económico de las personas.

Según datos de CONEVAL (2016) 53.4 millones de personas (43.6 %) en México en el año 2016 se encontraban en pobreza y 9.4 millones (7.6 %) estaban considerados en la pobreza extrema, varios de los factores ya mencionados son los que influyen para que las personas permanezcan en la situación de pobreza o bien caigan en ella. El mismo CONEVAL (2016) señala que una desigualdad importante en el país, es la pobreza en los hogares encabezados por mujeres porque se asocian a vulnerabilidad sociodemográfica, porque tienen mayor número de niños y adultos mayores, poblaciones frágiles y sesgadas enfocadas a la exclusión. Es importante señalar que la jefatura del hogar, la composición, el ingreso, la educación, la salud, la vivienda y la seguridad social son elementos importantes para analizar al binomio pobreza – exclusión.

En los estudios de pobreza en España, que realiza Llano (2016), proporciona elementos importantes que pueden extenderse a varios países en el mundo: los hombres en jefatura tienen más altas tasas de pobreza (22.5, en 2015 comparado con 21.8 % para las mujeres), la pobreza es significativa en la población con menor nivel de estudios (29.9 % tienen el primer nivel de secundaria), 44.8 % de los desempleados y 21 % de los inactivos caen en esta situación, el 13 % de las familias monoparentales son las que presentan pobreza.

Considerando estos resultados se elaboraron dos modelos logísticos teniendo como respuesta el estar por abajo de la LB y como variables independientes el sexo de la jefatura, la edad del jefe de familia, el número de integrantes del hogar, los que tienen 65 y más años, el número de ocupados, el gasto total, las transferencias, el total de perceptores en el hogar, el estrato socioeconómico, el tipo de hogar, tener menores en el hogar y el nivel educativo de los jefes de familia. Los dos modelos se hicieron para comparar hogares en general y hogares de adultos mayores. Los resultados muestran lo que se sabe de la pobreza-exclusión, esas variables son muy importantes para calcular la probabilidad de estar por arriba de la LB, efectivamente hay diferenciales por ser jefe de familia hombre o mujer, tener adultos mayores en el hogar o ser jefe adulo mayor, a medida que hay mayor número de perceptores de ingresos, la probabilidad de estar por arriba de la LB.

Con los ejemplos que se calcularon a través de los dos modelos logísticos, puede apreciarse que los hogares de los adultos mayores tienen mayor probabilidad de caer por debajo de la LB, efectivamente la jefatura del adulto mayor y con mayor número de personas mayores en el hogar sitúan a estos hogares en gran vulnerabilidad y mayor exclusión.

Agradecimientos

Los autores agradecen a la Dirección General de Asuntos del Personal Académico de la Universidad Nacional Autónoma de México, a través del proyecto PAPIIT IG300517 el financiamiento de la investigación.

Referencias Bibliográficas

Arroyo Bovea, M. (2016). Exclusión social y pobreza en la Unión Europea. Grupo Interdisciplinario de Estudios Críticos y de América Latina (GIECRYAL) de la Universidad de Alicante, 7(85), 104-131.

CONEVAL. (2016). Pobreza en México. Recuperado el agosto de 2018, de `https://www.coneval.org.mx/Medicion/MP/Paginas/Pobreza_2016.aspx`

Feres, J.C. y Mancero, X. (s.f.). Enfoques para la medición de la pobreza. Breve de la revisión de la literatura. Santiago de Chile: CEPAL.

García Luque, O. y Losa Carmona, A. (2008). Capítulo 13. Pobreza, condiciones de vida y privación. En N. Hernández Pedreño, Exclusión social y desigualdad (págs. 317-342). Murcia: Universidad de Murcia, Servicio de Publicaciones.

Hernández Pedreño, M. (2008). Exclusión social y desigualdad. Murcia: Universidad de Murcia, Servicio de Publicaciones.

Hernández Pedreño, M. (2008). Capítulo 1. Pobreza y exclusión en las sociedades del conocimiento. En Hernández Pedreño, M, Exclusión social y desigualdad (págs. 15-58). Murcia: Universidad de Murcia, Servicio de Publicaciones.

Hernández Pedreño, M. (2010). El estudio de la pobreza y la exclusión social. Aproximación cuantitativa y cualitativa. Revista Interuniversitaria de Formación del Profesorado, 69(24,3), 25-46.

INEGI. (2016). Encuesta Nacional de Ingresos y Gastos de los Hogares 2016, ENIGH. Nueva serie Descripción de la base de datos. Cd. de México: INEGI.

Llano Ortiz, J. (2016). El estado de la pobreza. Seguimiento del indicador de riesgo de pobreza y exclusión social en España 2009-2015. Resumen Ejecutivo. España: EAPN.

Martínez, J. (2008). Capítulo 3. Educación social para la igualdad. En Hernández Pedreño, M., Exclusión social y desigualdad (págs. 79-104). Murcia: Universidad de Murcia, Servicio de Publicaciones.

Mira Albert, S. (2008). Capítulo 11. Indicadores sociales y valoración de la exclusión. En Hernández Pedreño, M., Exclusión social y desigualdad (págs. 281-293). Murcia: Universidad de Murcia, Servicio de Publicaciones.

Moreno Mínguez, A. y Acebes Valentín, R. (s.f.). Estado de bienestar, cambio familiar, pobreza y exclusión social en España en el marco comparado europeo. Revista del Ministerio de Trabajo y Migración, 75, 31-49.

Moya, E. (2013). El concepto de exclusión social, una mirada crítica. Buenos Aires, Argentina: X Jornadas de Sociología. Facultad de Ciencias Sociales, Universidad de Buenos Aires.

Pérez Moya, C. (2016). La pobreza y el nuevo contexto de la exclusión social. Publicaciones didácticas, 479-483.

PNUD. (2013). Seguridad ciudadana con rostro humano: diagnóstico y propuestas para América Latina. Informe regional de Desarrollo Humano 2013-2014. Nueva York, estado Unidos: PNUD.

PNUD. (2016). Informe Regional sobre Desarrollo Humano para América Latina y el Caribe Progreso multidimensional: bienestar más allá del ingreso. Nueva York: Programa de Naciones Unidas para el Desarrollo.

Prialé, M. (s.f.). Exclusión y pobreza: Realidades Conectadas. (S. compartidos, Ed.) Recuperado el marzo de 2018, de `www.sabercompartidos.pe` 04.03.18

Sabater Fernández, C. y Giró Miranda, J. (2015). La nueva pobreza. El desahucio como proceso de exclusión. Ehquidad International Welfare Policies and Social Work Journal(3), 77-106.

Sánchez Alías, A. y Jiménez Sánchez, M. (2013). Exclusión social: Fundamentos teóricos de la intervención. Trabajo social global, 3(4), 133-156.

Sánchez Mora Molina, M.I. y Olaz Capitán, A. (2008). Capítulo 7. Dependencia y riesgo social . En Hernández

Pedreño, M, Exclusión social y desigualdad (págs. 169-184). Murcia: Universidad de Murcia, Servicio de Publicaciones.

Subirats, J.; Giménez, M y Obradors, A. (2005). Exclusión social y políticas de inclusión. Algunos elementos sobre la realidad europea y española. Revista do Serviço Público, 56(1), 23-35.

Tezanos Vázquez, S.; Quiñones Montellano, A.; Gutierrez Sobrao, D. y Madrueño Aguilar, R. (2013). Manuales sobre cooperación y desarrollo. Desarrollo humano, pobreza y desigualdades. Santander, España: Cátedra de Cooperación Internacional y con Iberoamérica - Universidad de Cantabria.

Vicente Jiménez, T. (2008). Capítulo 5. Derechos sociales e integración. En Hernández Pedreño, M., Exclusión social y desigualdad (págs. 131-145). Murcia: Universidad de Murcia, Servicio de Publicaciones.

Análisis estocástico del flujo vehicular para la red de ejes viales principales de la Ciudad de México mediante cadenas de Markov discretas

Jessica Pereda Méndez
Luis Olivares-Quiroz
Departamento de Física and Posgrado en Ciencias de la Complejidad
Universidad Autónoma de la Ciudad de México
luis.olivares@uacm.edu.mx

Resumen

En este trabajo presentamos un análisis del flujo vehicular para un conjunto de 32 ejes viales principales dentro la Ciudad de México basado en la implementación de un proceso estocástico de cadenas de Markov discretas (CMD) definido sobre el grafo $\mathcal{G}$ de la red vial que constituyen los ejes viales y sus intersecciones. En particular, y basados en información estadística del INEGI determinamos la matriz de probabilidad de transición $\mathbf{P}$, la matriz de primer visita promedio (*mean first passage time*) $\mathbf{M}$ y la constante de Kemeny K, las cuales nos permiten establecer un análisis cuantitativo de la dinámica de la red. Con base en esta información, analizaremos el promedio del flujo vehicular tanto en la red completa como en los ejes viales principales. Asimismo, utilizaremos este formalismo para explorar el impacto que tiene el cierre de un eje vial particular $n^{\dagger}$ dentro de la red sobre el flujo en los demás ejes viales y sobre el flujo vehicular global.

Palabras Clave: Teoría de Grafos; matriz de probabilidad de transición; vector de distribución de probabilidad estacionaria; matriz de primer visita promedio; constante de Kemeny

1 Introducción

El tráfico vehicular es uno de los problemas más importantes que enfrentan actualmente las grandes urbes en función del número creciente de vehículos que circulan y del limitado espacio existente [13]. Los efectos producidos por el incremento del flujo vehicular van más allá del impacto económico, alcanzando la esfera social, el medio ambiente y la calidad de vida de los habitantes en sí misma [6]. Por tanto, cada vez es más necesario contar con herramientas formales y cuantitativas que permitan el diseño de redes de flujo vehicular (avenidas, ciclovías, sistemas de transporte público) que permitan una comunicación más eficiente y efectiva entre los distintos puntos de una ciudad [19]. Ésta no es una preocupación reciente. Los primeros estudios sobre el tráfico vehicular se remontan a la década de 1930-1940 con los trabajos de B D Greenshield, quien aplicó por primera vez métodos fotográficos y matemáticos para la medición de diferentes variables relacionadas con el tráfico vehicular y la descripción de su comportamiento [12].

Diferentes modelos matemáticos se han aplicado para analizar el problema del transporte eficiente y del flujo vehicular en redes urbanas. Dependiendo del nivel de detalle, los modelos de tráfico vehicular generalmente se clasifican como microscópicos, mesoscópicos y macroscópicos [16]. Existen modelos matemáticos basados en ecuaciones diferenciales parciales (EDP) [8], autómatas celulares (AC) [1], modelos basados en agentes [10], ecuaciones diferenciales estocásticas, entre otros. En este trabajo abordaremos el análisis del flujo vehicular mediante el formalismo de los procesos estocásticos en tiempo discreto. Más específicamente, definiremos una cadena de Markov discreta (CMD) sobre un grafo dual $\mathcal{G}$ el cual representará la dinámica del flujo vehicular a tiempo discreto sobre un conjunto de nodos $\{n_j\}$ y enlaces $\{e_j\}$ los cuales representan a los ejes viales y sus intersecciones. Basaremos

este análisis en el trabajo de Crisostomi *et al* [9], en donde se discute ampliamente las ventajas del enfoque Markoviano para este tipo de análisis. Tal como mostraremos en este trabajo, el enfoque estocástico mediante CMD nos permitirá extraer información relevante de conectividad y flujo dentro de la red [27, 25].

En este trabajo se analizará el conjunto de 32 ejes viales principales de la Ciudad de México, el cual en conjunto con sus intersecciones conforma un grafo dual $\mathcal{G}$. A partir de este grafo dual, definimos un proceso estocástico de tipo cadena de Markov a tiempo discreto con matriz de probabilidad $\mathbf{P}$. Para determinar los elementos p_{ij} de la matriz $\mathbf{P}$ utilizaremos información estadística obtenida a partir de las matrices origen destino (OD) del INEGI y la Secretaría de Movilidad de la Ciudad de México [2]. Con esta información evaluaremos tanto el vector de distribución de probabilidad estacionaria $\vec{\pi}$ de la cadena de Markov como la matriz de primer visita promedio $\mathbf{M}$ y la constante de Kemeny K. Las cadenas de Markov constituyen uno de los formalismos estocásticos más utilizados en la modelación de un proceso probabilístico[21]. En esencia, este tipo de procesos estocásticos establece que el estado actual del proceso sólo depende del resultado *inmediato anterior* y no del conjunto de estados previos [5]. La propiedad de Markov menciona que la distribución de probabilidad del valor futuro de una variable aleatoria depende únicamente de su valor presente, siendo independiente de la historia de dicha variable, por lo que podemos expresarla como:

$$P\left(X_{n+1}|X_n\right) \tag{1}$$

es decir, sólo depende del valor de la variable estocástica en el paso inmediato anterior. Tal proceso se denomina proceso de Markov o cadena de Markov discreta (CMD) y fue propuesto por Andrei Markov .

Este trabajo se presenta de la siguiente manera. En la Sección II discutiremos los elementos básicos de la teoría de los procesos de Markov discretos. A continuación presentaremos la aplicación de estos resultados a la red de 32 ejes viales principales de la Ciudad de México. Posteriormente efectuaremos un análisis cuantitativo detallado de las cantidades mencionadas previamente y analizaremos el impacto del cierre de un eje vial dado a la circulación mediante el uso del bloqueo de un eje vial $n^{\dagger}$ dentro de la red y el cálculo de este efecto en las cantidades anteriormente descritas. Finalmente presentaremos un conjunto de conclusiones y perspectivas que pueden considerarse en un análisis posterior a futuro de este trabajo.

2 Cadenas de Markov discretas. Un proceso estocástico clásico

En este trabajo representaremos la conectividad de los ejes viales bajo consideración y el tráfico vehicular dentro de la Ciudad de México mediante una cadena de Markov discreta (CMD). Una CMD representa un proceso estocástico discreto definido por un conjunto de estados $\{S_i\}$, para los cuales existe una probabilidad de transición p_{ij} de pasar del estado S_i al estado S_j. Esta probabilidad p_{ij} cumple la propiedad de Markov la cual establece que si el estado S_n al tiempo n y el conjunto de estados discretos a tiempos previos $S_1, ..., S_{n-1}$ son conocidos, la probabilidad del estado futuro S_{n+1} no depende de los estados anteriores $S_1, ..., S_{n-1}$, sino solo depende del estado actual S_n. Esto es, [26]:

$$\begin{aligned} &P\left(S_{n+1} = s_{n+1}|S_1 = s_1, S_2 = s_2, ..., S_n = s_n\right) \\ &= P\left(S_{n+1} = s_{n+1}|S_n = s_n\right) \end{aligned} \tag{2}$$

la Figura (1) muestra una representación de una cadena de Markov discreta con un conjunto de estados dado por el conjunto $\{S_1, S_2, S_3, S_4\}$ con probabilidades de transición p_{ij} determinadas. Una CMD está por tanto determinada por una matriz de tamaño $N \times N$ llamada matriz de probabilidades de transición $\mathbf{P}$ cuyas entradas p_{ij} denotan la probabilidad de pasar del estado S_i al estado S_j en una unidad de tiempo dada. Esto es,

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots \\ p_{21} & p_{22} & p_{23} & \cdots \\ p_{31} & p_{32} & p_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{3}$$

Los elementos p_{ij} satisfacen que $p_{ij} \geq 0$ y $\sum_j p_{ij} = 1$. Es decir, las filas de la matriz $\mathbf{P}$ suman la unidad.

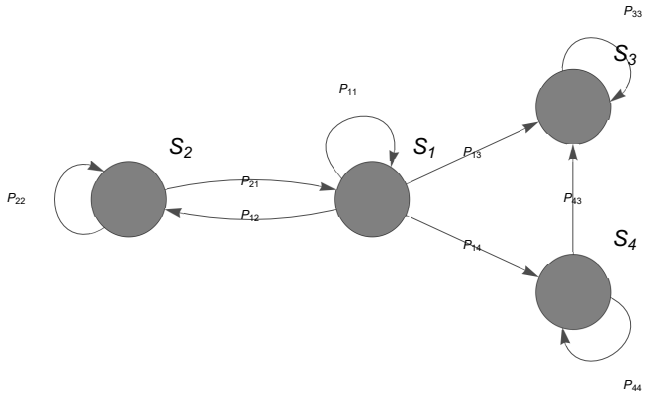


Figura 1: Representación de una cadena de Markov con espacio de estados S_1, S_2, S_3 y S_4 para los cuales existe una probabilidad de transición probabilidad p_{ij} de que el estado S_i pueda pasar al estado S_j. Adicionalmente, dentro de una cadena de Markov pueden existir *loops* en los cuales puede existir una probabilidad de transición $p_{ii} \neq 0$, tal como se muestra en la representación.

Dada la matriz de probabilidades estacionarias $\mathbf{P}$ determinaremos el vector $\vec{\pi}$ de distribución de probabilidad estacionaria, $\vec{\pi} = \{\pi_1, ..., \pi_N\}$ el cual es una distribución de probabilidad invariante para una cadena de Markov con matriz de probabilidades de transición $\mathbf{P} = \{p_{ij}\}$ si cumple que [7]:

$$\vec{\pi}_j = \sum_i \pi_i p_{ij} \tag{4}$$

Lo anterior es equivalente a $\vec{\pi} = \vec{\pi} \cdot \mathbf{P}$ y significa que si una variable aleatoria inicial X_0 tiene una distribución π entonces la distribución de X_n también es π, es decir, esta distribución no cambia con el paso del tiempo [18]. Para CMD ergódicas, i.e, aquellas en las que es posible alcanzar cualquier estado S_k en un número finito de pasos, determinamos la matriz $\mathbf{W}$ y la matriz fundamental $\mathbf{Z}$ dadas por

$$\lim_{n\to\infty} \mathbf{P}^n = \mathbf{W} \tag{5}$$

donde $\mathbf{W}$ es una matriz con todas sus filas idénticas al vector de distribución de probabilidad estacionaria $\vec{\pi}$ discutido anteriormente. La matriz fundamental $\mathbf{Z}$ de esta CMD ergódica está dada por [22]

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P} + \mathbf{W})^{-1} \tag{6}$$

donde $\mathbf{I}$ es la matriz identidad, $\mathbf{P}$ la matriz de probabilidades de transición y $\mathbf{W}$ la matriz discutida anteriormente. Es importante mencionar que la matriz fundamental $\mathbf{Z}$ para cadenas ergódicas desempeña un papel muy importante, pues puede usarse para identificar un conjunto de parámetros relevantes, como la matriz de primer visita promedio (*mean first passage time matrix*, en inglés) [23], la cual nos será fundamental en este trabajo como una cantidad relevante para calcular la relevancia de los distintos ejes viales de la Ciudad de México.

La matriz de primer visita promedio, en inglés (MFPT) denotada como $\mathbf{M}_{ij}$, es el número promedio de pasos para llegar *por primera vez* al estado S_j partiendo de S_i. Se puede calcular de acuerdo con la siguiente expresión:

$$\mathbf{M}_{ij} = \frac{\mathbf{Z}_{jj} - \mathbf{Z}_{ij}}{\vec{\pi}_j}, \qquad i \neq j \tag{7}$$

donde la diagonal $\mathbf{M}_{ii} = 0$ con $i = 1, \ldots, N$ de acuerdo a la definición dada. Sin embargo, es posible extender esta definición para incluir los elementos diagonales de $\mathbf{M}$. Con esa generalización, los elementos diagonales son inversamente proporcionales a los tiempos de recurrencia r_i. Esto es, $M_{ii} = r_i^{-1}$ [24]. Finalmente, en este trabajo calcularemos también la constante de Kemeny, la cual nos permite cuantificar la eficiencia de flujo de información dentro de la red, de acuerdo a [17]:

$$K = \sum_{j=1}^{n} \mathbf{M}_{ij} \vec{\pi}_j \tag{8}$$

donde hemos tomando la convención de $\mathbf{M}_{ii} = 0$, la constante K es la suma de las entradas del vector de tamaño 32×1 que resulta del producto de la matriz $\mathbf{M}_{ij}$ por el vector de distribución de probabilidad estacionaria $\vec{\pi}_j$. La constante de Kemeny proporciona una cantidad interesante para las cadenas finitas de Markov ergódicas, que es independiente del estado inicial de la cadena de Markov como se puede ver en [17]. Por lo tanto, la constante de Kemeny es una medida intrínseca para este proceso estocástico, y si la matriz de transición $\mathbf{P}$ tiene valores propios $\lambda_1 = 1, \lambda_2, \lambda_3, ..., \lambda_n$ entonces otra forma de calcular K es:

$$K = \sum_{j=2}^{N} \frac{1}{1 - \lambda_j} \tag{9}$$

donde la suma inicia en el índice $j = 2$. La ecuación (9) enfatiza el hecho de que K solo depende en realidad de $\mathbf{P}$ y que aumenta si uno o más valores propios de $\mathbf{P}$ es real y cercano a 1. [17].

3 Aplicaciones a redes de transporte. Caso de la Ciudad de México

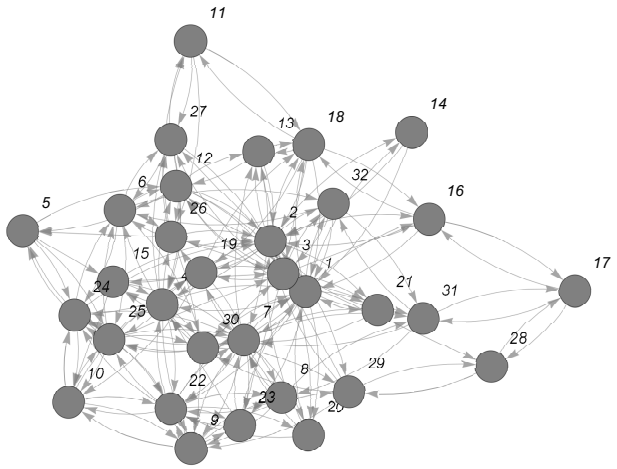
Con base en las herramientas anteriores, definiremos una cadena de Markov discreta (CMD) para analizar cuantitativamente las propiedades del flujo de tráfico dentro de un conjunto de ejes viales de la Ciudad de México. Para ello, seleccionaremos un conjunto de avenidas o ejes viales principales y sus respectivos puntos de cruce como los elementos del grafo G. En esta representación G, a la cual definimos como *grafo primario*, las intersecciones son los nodos de la red y los enlaces son los ejes viales mismos. Para este grafo G se define el grafo dual $\mathcal{G}$ como aquel en el cual los nodos serán ahora los ejes viales y las intersecciones de los ejes viales corresponden a los enlaces del grafo $\mathcal{G}$. Esto es, los enlaces e_j y los nodos n_j intercambian su papel [4]. La Figura (2) muestra la red de ejes viales de la Ciudad de México que analizaremos en este trabajo, la cual consta de los 32 principales ejes viales de la Ciudad de México [14].



Figura 2: Mapa vial de la Ciudad de México en donde se muestra los 32 principales ejes viales de la ciudad.

El flujo vehicular, es decir, el número de vehículos que circulan en los ejes principales es información no fácilmente accesible. Sin embargo, es factible conocer datos estadísticos a partir de la información en las bases de datos del INEGI (Instituto Nacional de Estadística, Geografía e Informática) [15]. Para determinar la matriz $\mathbf{P}$ se requiere conocer el número total de viajes V_{ij} que parten del eje vial E_i y se dirigen al eje vial E_j. Esto es una cantidad compleja de determinar, sin embargo, se considerará la información disponible para una ventana de tiempo y espacio restringido. En este caso, consideraremos inicialmente un día en particular de la semana durante un lapso de tiempo de 2 horas, específicamente consideramos una ventana temporal entre 8 y 10 horas de la mañana. Por tanto entonces, la probabilidad p_{ij} se determina por

$$p_{ij} = \frac{V_{ij}}{\sum_{j=1}^{n} V_{ij}} \tag{10}$$



(a)

N. eje vial	Eje vial	N. eje vial	Eje vial
1	Eje central	17	Eje 4A norte
2	Eje 1 oriente	18	Eje 5 norte
3	Eje 2 oriente	19	Eje 6 norte
4	Eje 3 oriente	20	Eje 1 sur
5	Eje 4 oriente	21	Eje 1A sur
6	Eje 5 oriente	22	Eje 2 sur
7	Eje 1 poniente	23	Eje 2A sur
8	Eje 2 poniente	24	Eje 3 sur
9	Eje 3 poniente	25	Eje 4 sur
10	Eje 4 poniente	26	Eje 5 sur
11	Eje 5 poniente	27	Eje 6 sur
12	Eje 1 norte	28	Eje 7 sur
13	Eje 2 norte	29	Eje 7A sur
14	Eje 3 norte	30	Eje 8 sur
15	Eje 3A norte	31	Eje 9 sur
16	Eje 4 norte	32	Eje 10 sur

(b)

Figura 3: Representación del grafo dual $\mathcal{G}$ para el conjunto de 32 ejes viales principales y sus intersecciones en la Ciudad de México. En el lado izquierdo se muestra la representación de G' asociado a la matriz de adyacencia correspondiente. En la parte derecha se muestra una tabla de referencia que permite identificar el número del nodo n_j con el eje vial correspondiente.

donde p_{ij} son las entradas de la matriz de probabilidad de transición. Con esta información es posible determinar $\mathbf{P}$ y el grafo correspondiente, el cual se muestra en la Figura (3).

A partir de esta información, determinamos el vector de distribución de probabilidad estacionaria $\vec{\pi}$, el cual nos indica el estado de equilibrio de la cadena de Markov definida sobre la red. La figura (4) muestra el vector de distribución de probabilidad estacionaria $\vec{\pi}$ en función del nodo, es decir, de los ejes viales considerados. El vector de distribución de probabilidad estacionaria $\vec{\pi}$ indica aquellos ejes viales con mayor probabilidad de ocupación para un día particular de la semana, en un horario de 8 a 10 horas de la mañana. Los ejes viales que muestran una mayor probabilidad de ocupación, *a tiempos largos de evolución de la CMD* son los correspondientes a los ejes viales 1, 22, 4, 30 y 2, correspondientes al eje Central, eje 2 sur, eje 3 oriente, eje 8 sur y eje 1 oriente.

Uno de los objetivos centrales de este trabajo es determinar la matriz de primer visita promedio, (*mean first passage time*) $\mathbf{M} = \{M_{ij}\}$ la cual nos indica, en promedio, el número de pasos requerido para pasar de un nodo n_i a un nodo n_j dentro de la red. En este caso, dado que el conjunto de nodos n_k se refiere a los ejes viales en cuestión, el elemento M_{ij} se refiere al tiempo promedio, considerando todas las rutas posibles, en el cual partiendo del eje vial i se alcanza el eje vial j por *primera vez*. Es fundamental considerar que este tiempo (o número de pasos) se refiere a la primera vez que se alcanza el nodo n_j dentro de la red. En principio es perfectamente posible construir trayectorias que lleguen a n_j muchas más de una vez, pero no son las que se consideran en el proceso estocástico aquí descrito.

La figura (5) muestra una representación tridimensional de $\mathbf{M}$, en donde los ejes x e y representan los ejes viales de origen y destino, mientras que el eje z se refiere al valor correspondiente de M_{ij}. Los valores más grandes de las entradas M_{ij} representan los tiempos *promedio* más grandes de recorrido entre dos ejes viales determinados.

Finalmente, se ha determinado también la llamada constante de Kemeny K, la cual en teoría de grafos corresponde al número de pasos promedio requeridos para alcanzar cualquier nodo n_j dentro de la red a partir de un nodo inicial n_i. Interesantemente y como ha sido demostrado, el valor de la constante de Kemeny no depende de que nodo n_i se elija [11, 23]. En la literatura se ha utilizado como una medida de la eficacia de la red, pues redes caracterizadas por pequeños valores en la constante de Kemeny generalmente tienen un mayor grado de eficiencia [11]. Para nuestro caso, el valor calculado es $K = 34.4376$, el cual es un indicador de desempeño vial de la red en

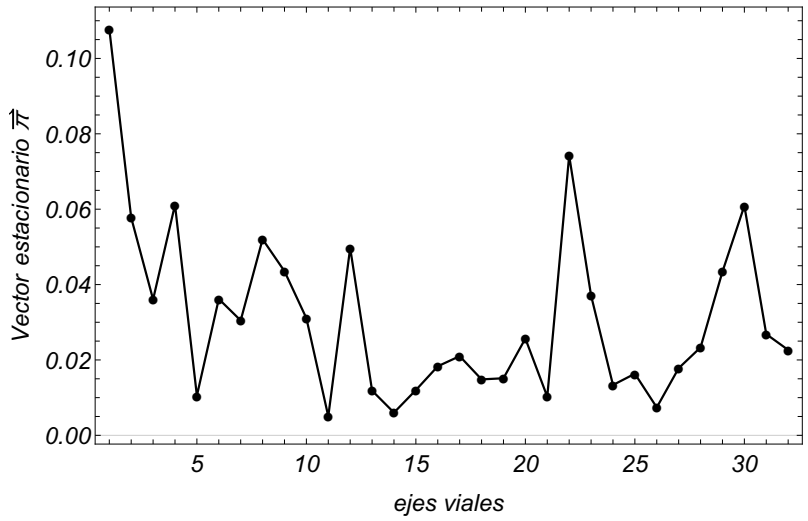


Figura 4: Vector de distribución de probabilidad estacionaria correspondiente $\vec{\pi}$ de la red de ejes viales de la CDMX calculada mediante la ecuación (4) en donde $\vec{\pi} = \pi \cdot \mathbf{P}$, contiene información de la probabilidad de ocupación de cada eje vial.

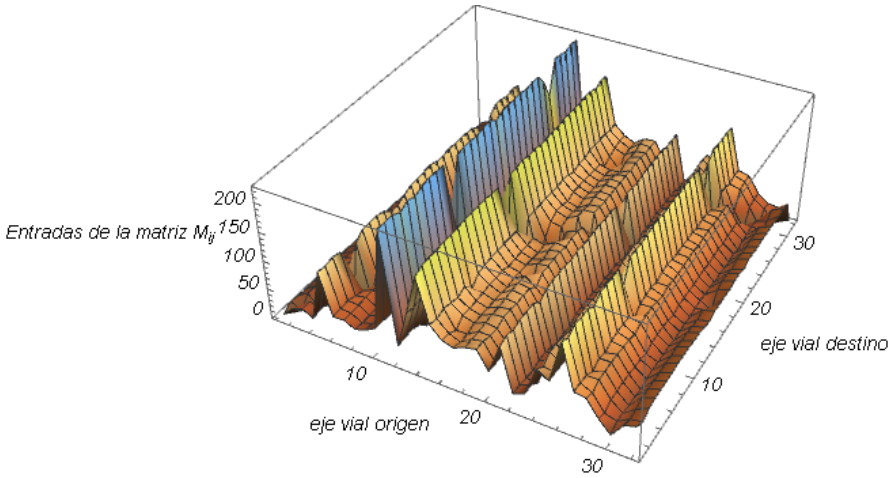


Figura 5: Representación de la matriz de primer visita promedio $\mathbf{M}$ correspondientes a la red de ejes viales considerados de la Ciudad de México. Los ejes x e y contienen los ejes viales de origen y destino, mientras que el eje z representa el tiempo promedio de primera visita.

general. Para comprender mejor el significado de K considere un conductor al azar que se mueve dentro de la red de acuerdo a las probabilidades de transición p_{ij} dadas de la matriz $\mathbf{P}$. En algún momento, nuestro conductor está perdido y no sabe el estado (eje vial) en que se encuentra y hacia dónde se dirige. En este contexto, la constante de Kemeny puede interpretarse como el número promedio de pasos que seguirá el conductor al azar antes de llegar a su destino [20].

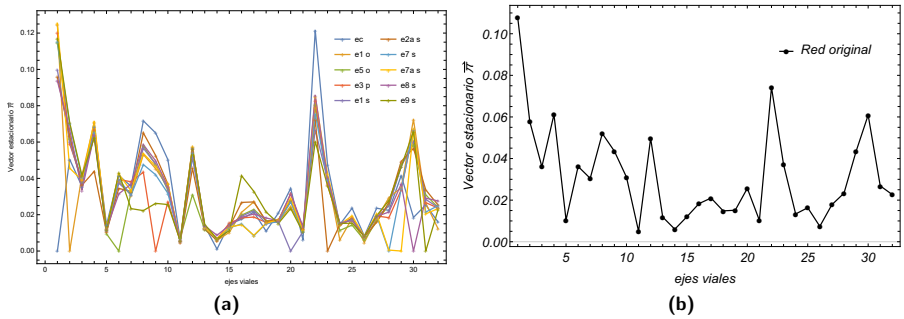


Figura 6: Vector de distribución de probabilidad estacionaria de ocupación $\vec{\pi}^{\star}$ para 10 diferentes redes viales, considerando en cada caso el cierre de un eje vial en particular, es decir, $n_i^{\dagger}$ el nodo asociado al eje vial que se encuentra bloqueado o inaccesible al flujo vehicular Panel(a). En la derecha, se muestra el vector de distribución de probabilidad estacionaria $\vec{\pi}$ para la red original en donde todos los ejes viales están abiertos para la circulación.

3.1. Identificación de ejes viales importantes

Una pregunta interesante en el diseño de redes viales es, qué le sucede a la red con respecto a las ocupaciones de los ejes viales y el tiempo promedio dentro de ella [3]. En particular, un tema central que discutiremos aquí es tratar de estimar que sucede cuando uno de los ejes viales se cierra o es bloqueado al flujo vehicular, debido a desastres naturales, obras viales, manifestaciones, entre otras posibilidades. Para realizar esta estimación, consideraremos la red descrita anterior y llamaremos $n^{\dagger}$ al nodo asociado al eje vial que se encuentra bloqueado o inaccesible al flujo vehicular en función de las razones descritas. Para analizar el efecto de la inaccesibilidad del nodo $n^{\dagger}$ para el flujo de vehicular dentro de la red, realizaremos lo siguiente. Para el nodo $n^{\dagger}$ sustituiremos por ceros los valores de la columna de la matriz $\mathbf{P}$ asociada al nodo $n^{\dagger}$. Esto indicará que eje vial † no está disponible para poder circular y así sucesivamente. Es decir, se construyen 32 redes diferentes en donde se considera el cierre de cada nodo, la información de las nuevas entradas de cada matriz de transición se obtiene de acuerdo a la ecuación (10), como estamos considerando que V_{ij} el número total de viajes que parten del eje vial E_i y se dirigen al eje vial E_j y el eje vial E_j permanece cerrado (denotado como $n^{\dagger}$), entonces su valor será 0, es decir no se van a considerar los viajes a ese destino aunque existan, este cambio o restricción hace que en la parte del denominador de la ecuación (10) se considere un viaje menos del total que parten del eje vial E_i y así se vuelven a distribuir las probabilidades para que las filas sumen 1.

Posteriormente calculamos el nuevo vector de distribución de probabilidad estacionaria $\vec{\pi}^{\star}$para cada una de las redes. Cada nueva red se refiere al bloque del eje vial asociado al nodo $n^{\dagger}$, el cual recorre todos los índices de la red.

Para cada valor del vector de distribución de probabilidad estacionaria $\dagger n$, el cual en este caso recorrerá todos los 32 nodos de los ejes viales de la CDMX, se calculará el vector de distribución de probabilidad estacionaria $\vec{\pi}^{\star}$ de acuerdo a la prescripción anterior. A partir de estos resultados se han calculado las constantes de Kemeny k_i para cada caso. La Figura (6) muestra el vector de distribución de probabilidad estacionaria $\vec{\pi}^{\star}$, el cual indica el valor más probable de la distribución estacionaria de ocupación de cada eje vial n_j, una vez que se ha bloqueado un eje particular, en cada caso se bloquea el $n_i^{\dagger}$ el nodo asociado al eje vial que se encuentra bloqueado o inaccesible al flujo vehicular. Uno de los resultados más interesantes de este análisis es que el cierre de un eje vial en particular tiene como consecuencia el aumento de la ocupación de los demás. Asimismo se puede observar que en algunos casos, el grado de ocupación de algunos ejes viales disminuye cuando algún otro se cierra implicando entonces una mejora en la circulación.

La relevancia de un eje vial dado dentro de una red de comunicación puede definirse de acuerdo a diferentes criterios, pero en este trabajo consideraremos que un eje vial posee un alto grado de relevancia si el tiempo promedio de viaje dentro de la red aumentan cuando se cierra este eje vial. Así, entonces diremos que $n^{\blacktriangle}$ es

un eje vial crítico dentro de la red si se cumple que la constante de Kemeny $k^{\blacktriangle}$ es mayor que la constante de Kemeny K de la red sin perturbación. Para determinar aquellos ejes viales cuya relevancia es mayor dentro de la red, calcularemos las constantes de Kemeny k_i asociadas al bloqueo de cada uno de los ejes viales, tal como se realizó anteriormente. La Figura (7) muestra los valores de k_i en función del bloqueo del nodo n_j. El cálculo de k_i muestra que existen al menos dos ejes viales para los cuales el tiempo promedio de recorrido dentro de la red se incrementa sustancialmente. Éstos son n_{18} y n_{32} los cuales corresponden a los ejes 5 norte y 10 sur. Por otro lado, se observa que es posible se presente el fenómeno inverso. Es decir, que el bloqueo de un eje $\dagger n$ determinado pueda disminuir sustancialmente el flujo de la red. Es importante mencionar que estamos discutiendo fenómenos *promedio* dentro de la red en un proceso estocástico. De forma que de un cierto conjunto de realizaciones del proceso estocástico, *en promedio* se observan estas características. No siempre se observará esta disminución o aumento de la constante de Kemeny pero resulta un indicador de la estructura de conectividad de la red. Los valores más grandes del conjunto $\{k_j\}$ se obtienen si se cierran los ejes viales 31, 17 y 2, correspondientes a eje 9 sur, eje 4A norte, eje 1 oriente; estos ejes viales pueden considerarse las más críticas de la red. En tanto que los menos críticos son los ejes viales 28, 30 y 29, correspondientes a eje 7 sur, eje 8 sur y eje 7A sur, como se muestra en la Figura (7), en donde a medida que se cierra algún eje vial, las demás son afectadas o beneficiadas.

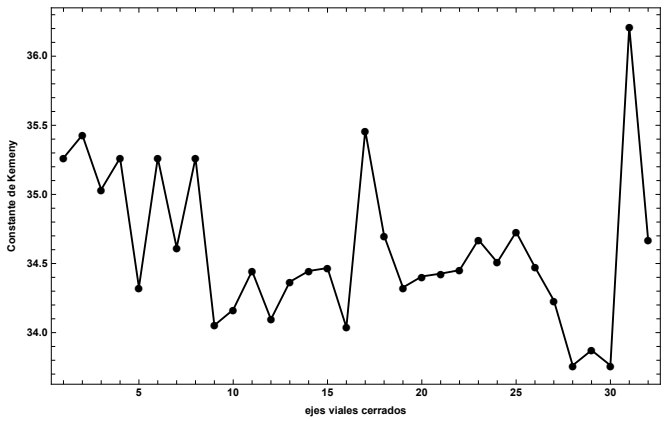


Figura 7: Cálculo del conjunto de constantes de Kemeny $\{k_j\}$ el cual es una estimación cuantitativa de la eficiencia de comunicación dentro de la red. En este cálculo, se bloqueó cada uno de los 32 nodos correspondientes a cada eje vial y se re-calculó el vector de distribución de probabilidad estacionaria $\vec{\pi}^{\star}$ y a partir de el valor de la constante de Kemeny correspondiente.

Tal como se mencionó anteriormente,la determinación de la matriz de probabilidad $\mathbf{P}$ y la matriz $\mathbf{M}$ se realizaron para un horario específico. Se extendió este cálculo considerando la información de las bases de datos del INEGI correspondientes a tres horarios diferentes [15]. Con base en la información de las matrices origen-destino (OD) correspondientes a tres horarios diferentes se calcularon las matrices de probabilidad de transición $\mathbf{P}$ y la matriz de primer visita promedio $\mathbf{M}$ en los tres casos. La Figura (8) muestra una comparación del vector de distribución de probabilidad estacionaria $\vec{\pi}$ en cada caso, correspondientes a 3 diferentes horarios para un día entre semana, de 8 a 10 Hrs. de la mañana, 2 a 4 Hrs. de la tarde y 8 a 10 Hrs. de la noche. Se observa que esencialmente los tres vectores de distribución de probabilidad estacionaria tienen esencialmente el mismo comportamiento. Éste resultado es interesante, dado que en principio, se podría pensar que cada horario podría tener un vector de distribución de probabilidad estacionaria $\vec{\pi}$ completamente diferente. Sin embargo éste no es el caso.

Finalmente, se realizó el cálculo de las constantes de Kemeny con la red completa y funcional para cada uno de los horarios descritos. Tal como se ha mencionado, la constante de Kemeny es una estimación cuantitativa del tiempo promedio de flujo o tránsito dentro de la red. Los cálculos mostrados en la Tabla 1 muestran que K sí tiene una variación dependiendo del horario. Es decir, en tanto que el vector de distribución de probabilidad estacionaria $\vec{\pi}$ es esencialmente constante, el valor de K sí depende del horario. El valor mayor de K corresponde

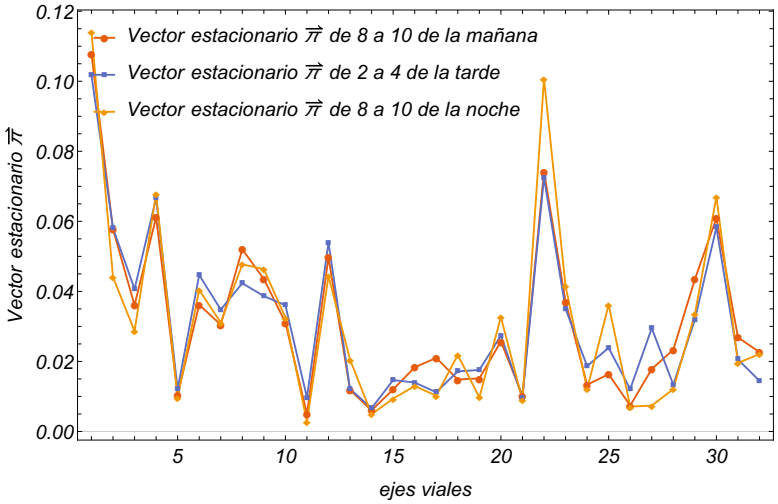


Figura 8: Vector de distribución de probabilidad estacionaria $\vec{\pi}$ para la red de ejes viales considerada dentro de la CDMX en 3 horarios diferentes basados en información del INEGI. De 8 a 10 de la mañana(rojo), 2 a 4 horas de la tarde(azul) y de 8 a 10 de la noche(naranja)

al horario matutino, posteriormente el horario de la tarde y finalmente el horario nocturno. Así, el flujo dentro de la red es mucho más lento por la mañana que la tarde o noche.

Horario	Constante de Kemeny K
8 a 10 hrs. de la mañana	34.4376
2 a 4 hrs. de la tarde	34.2413
8 a 10 hrs. de la noche	34.1795

Tabla 1: Comparación de las variaciones del valor de la Constante de Kemeny donde, todos los ejes viales están disponibles para poder circular, la constante de Kemeny contiene información acerca del tiempo promedio dentro de la red en 3 diferentes horarios, de 8 a 10 de la mañana, 2 a 4 horas de la tarde y de 8 a 10 de la noche, estimada por Wolfram Mathematica mediante la ecuación (8) donde, $K = \sum_{j=1}^{n} \mathbf{M}_{ij}\vec{\pi}_j$.

Los valores de las constantes de Kemeny obtenidas son bastantes cercanas, por lo que no hay un aumento significativo en el tiempo promedio de viaje de un eje vial arbitrario a un destino elegido al azar de acuerdo con la distribución estacionaria en estos horarios. Los tiempos de viaje son constantes para todas los ejes viales en los tres horarios (por lo tanto, no hay tráfico, o al menos se distribuye uniformemente). En cuanto a la eficiencia de la red, la red vial correspondiente al horario de 8 a 10 Hrs. de la noche tiene el menor valor de constante de Kemeny, y redes caracterizadas por valores pequeños de la constante de Kemeny deberían ser más eficientes en términos de flujo de tráfico.

4 Conclusiones

En este trabajo se propuso un modelo de cadena de Markov discreta (CMD) para analizar el flujo de tráfico dentro de la red de treinta y dos ejes viales principales de la Ciudad de México. La red de los 32 principales ejes viales de la Ciudad de México y sus intersecciones se definieron mediante un grafo dual $\mathcal{G}$ en donde los nodos n_j corresponden a los ejes viales y los enlaces a sus intersecciones. Sobre esta red, se define una cadena de Markov discreta (CMD) con matriz de probabilidades de transición $\mathbf{P} = \{p_{ij}\}$ la cual se obtiene a partir de las matrices

de origen-destino (OD) de las bases de datos del INEGI. A partir del cómputo de $\mathbf{P}$ es posible determinar otros cuantificadores como el vector de distribución de probabilidad estacionario $\vec{\pi}$, la matriz de primer visita promedio $\mathbf{M}$ y la constante de Kemeny.

A partir de los cuantificadores anteriores, se calculó el vector de distribución de probabilidad estacionaria $\vec{\pi}$ para la red de ejes viales en completa funcional, observándose que existe algunos ejes con mayor probabilidad de ocupación en el horario considerado inicialmente de 8 a 10 Hrs AM, que son los ejes viales 1, 22, 4, 30 y 2, correspondientes al eje Central, eje 2 sur, eje 3 oriente, eje 8 sur y eje 1 oriente. Uno de los resultados centrales de este trabajo es que al considerar tres diferentes horarios de flujo vehicular, de 8 a 10 Hrs am, de 14 a 16 Hrs y de 20 a 22 Hrs no se observa una diferencia sustancial para el vector de distribución de probabilidad estacionaria $\vec{\pi}$. Existen diferencias muy pequeñas, pero que en realidad no son significativas, indicando entonces que en promedio los tres horarios considerados tienen una probabilidad extremadamente similar de ocupación. Posteriormente se analizó la influencia del bloqueo de un eje vial particular $n^{\dagger}$ dentro de la red para evaluar la relevancia de este nodo para el flujo de tráfico vehicular. Para hacer esta valoración, se generaron 32 redes viales diferentes en cada una de las cuales se bloqueó uno de los ejes viales. Se calculó cada uno de los vectores de distribución de probabilidad estacionaria para cada una de estas redes y se observa que dado el cierre de un eje vial en particular tiene como consecuencia el aumento de la ocupación de los demás, de igual forma se aprecia que en algunos casos, el grado de ocupación de algunos ejes viales disminuye cuando alguno otro se cierra implicando entonces una mejora en la circulación.

En una segunda fase de este trabajo se evalúo la constante de Kemeny K tanto para la red vial inicial completa y funcional, como para cada una de sus 32 variantes en donde existía un eje bloqueado $n^{\dagger}$. El valor de K para la red vial inicial es de $K = 34.43$ el cual es nuestro valor de referencia. Se observa que al generar diferentes bloqueos en distintos nodos $n^{\dagger}$ de la red la constante de Kemeny $k^{\dagger}$ puede incrementarse o disminuir su valor, incluso por debajo del valor de referencia. Éste es el caso por ejemplo que ocurre en el bloqueo de los ejes que tienen un valor de K menor que el de referencia, que son el 28, 29 y 30 que corresponden a los ejes viales 7 sur, 7A sur y 8 sur. En forma contraintuitiva, el bloqueo de estos ejes viales reduce los tiempos promedios de viaje dentro de la red dado que el flujo vehicular se distribuirá en ejes de mayor capacidad. En general, el fenómeno que se observa es que la intuición indicaría. El bloqueo de un eje vial $n^{\dagger}$ induce un incremento en la constante de Kemeny correspondiente. En general, esto sucede en todos los casos, pero es particularmente notorio en el caso de los ejes viales 31, 17 y 2 correspondientes a los ejes viales 9 sur, 4A norte y 1 oriente.

Referencias

[1] Laboratorio nacional de ciencias de la sostenibilidad. http://lancis.ecologia.unam.mx/iai/modelacion/agentes.

[2] Secretaría de movilidad de la ciudad de méxico. https://www.semovi.cdmx.gob.mx/.

[3] S. Alaswad and A. Dhabi. Urban road network crisis response management: time-sensitive decision optimization. *Proceedings of the 2017 Industrial and Systems Engineering Conference*, 2017.

[4] R. Albert and A.-L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47 – 97, 2002.

[5] M. E. Caballero. *Cadenas de Markov: un enfoque elemental*. Aportaciones Matemáticas 29. Sociedad Matemática Mexicana, 2004.

[6] R. Cal and M. Reyes. *Ingeniería de Tránsito: Fundamentos y aplicaciones*. Alfa Omega Grupo Editor, S.A. de C.V., 8 edition, 2007.

[7] G. E. Cho and C. D. Meyer. Comparison of perturbation bounds for the stationary distribution of a markov chain. *Linear Algebra and its Applications*, 335(1):137 – 150, 2001.

[8] S. Çolak, A. Lima, and M. C. González. Understanding congested travel in urban areas. *Nature Communications*, 7(10793), 2016.

[9] E. Crisostomi, S. Kirkland, and R. Shorten. A google-like model of road network dynamics and its application to regulation and control. *International Journal of Control*, 84(3):633 – 651, 2011.

[10] J. Delgado, P. Saavedra, and R. M. Velasco. *Modelación matemática del flujo vehicular*. Publicaciones C.B.I UAM, 2012.

[11] P. Doyle. The kemeny constant of a markov chain. https://arxiv.org/pdf/0909.2636.pdf.

[12] B. Greenshields, J. R. Bibbind, W. S. Channing, and H. H. Miller. A study of traffic capacity. *Highway Research Board*, 14(1):448 – 477, 1935.

[13] J. Harri, F. Filali, and C. Bonnet. Mobility models for vehicular ad hoc networks. *IEEE Communications Surveys & Tutorials*, 11(4):19 – 41, 2009.

[14] G. Inc. Google maps. http://maps.google.com/.

[15] INEGI. Mapa digital de méxico. http://gaia.inegi.org.mx/mdm6/?v.

[16] A. introduction to stochastic modeling. *Howard M. Taylor and Samuel Karlin*. Academic Press, 2nd edition, 1994.

[17] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Princeton: Van Nostrand, 1960.

[18] S. Kirkland. Fastest expected time to mixing for a markov chain on a directed graph. *Linear Algebra and its Applications*, 433(11-12):1988 – 1996, 2010.

[19] R. D. Kuhne. Greenshields' legacy: Highway traffic. *Transportation Research E-Circular*, pages E–C149, 2011.

[20] M. Levene and G. Loizou. Kemeny constant and the random surfer. *The American Mathematical Monthly*, 109(8):741 – 745, 2002.

[21] T. L. Magnati and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1 – 93, 1984.

[22] A. A. Markov. An example of statistical analysis of the text of eugene onegin illustrating the association of trials into a chain. *Science in Context*, 19(4):591 – 600, 2006.

[23] R. P. Norris. *Markov chains*. Cambridge University Press, 1997.

[24] S. Redner. *A guide to first-passage processes*. Cambridge University Press, 2001.

[25] J. Reiter. Application of a markov chain traffic model to the greater philadelphia region. *CONCEPT*, 38, 2015.

[26] L. Rincón. *Introducción a los procesos estocásticos*. Departamento de Matemáticas UNAM, 2012.

[27] S. M. Ross. *Stochastic Processes*. John Wiley & Sons., 2nd edition, 1996.

El potencial del aprendizaje hebbiano en la clasificación supervisada

Fernando Aguilar-Canto
Carlos Brito-Loeza
pherjev@gmail.com

Resumen

A pesar de tener un sustento biológico, la regla de Hebb ha encontrado un espacio reducido en las implementaciones de redes neuronales artificiales en las últimas décadas, siendo los métodos basados en el gradiente más utilizados. Esto supone una controversia sobre hasta qué grado entendemos los aspectos computacionales del aprendizaje a nivel neuronal. En este trabajo, abordamos el problema del aprendizaje hebbiano en términos de la existencia de una arquitectura conveniente, tanto a nivel teórico como experimental. En el plano teórico, se demuestra la existencia de un red hebbiana capaz de lograr minimizar el error de clasificación a un mínimo global. Si bien esta arquitectura es poco práctica por la explosión combinatoria, se prueba que el algoritmo de m-celdas con entrenamiento hebbiano logra, en buena medida, resolver sus principales problemas. Finalmente se verificó experimentalmente que el método de m-celdas alcanza a los métodos basados en el gradiente en un problema de clasificación de baja dimensionalidad (base de datos Iris) con 0.967 de exactitud para $s = 2$, lo cual, a pesar de sus limitaciones, indica que el aprendizaje hebbiano puede derivar a un algoritmo de clasificación competente en el contexto de la Inteligencia Artificial moderna.

Palabras Clave: Regla de Hebb; Redes Neuronales; Modelos de plasticidad sináptica

1 Introducción

La Regla de Hebb es un modelo teórico de plasticidad sináptica propuesto originalmente el trabajo de Donald Hebb [13], aunque, como observa [22], autores como Konorski [17] y Ramón y Cajal [26] ya habían concebido ideas semejantes. Unas décadas después, las ideas de Hebb recibieron confirmación experimental al observarse que su conjetura modelaba al fenómeno de Potenciación a Largo Plazo (*Long-Term Potentiation*, LTP), el cual fue descrita por primera vez en los trabajos de Terje Lmo y Timothy Bliss [21, 2].

Computacionalmente, la regla de Hebb señala un método paramétrico de aprendizaje de redes neuronales que ha recibido cierto sustento biológico. Posterior al planteamiento original de la regla de Hebb simple se han propuesto otras reglas de aprendizaje más próximas a experimentos más recientes, como es el caso de la regla BCM [1] o la regla STDP (véase [9] para una formulación), o tratan de resolver problemas computacionales concretos como es el caso de la regla de Oja [23]. Sin embargo, a pesar de los avances, los algoritmos de redes neuronales (incluyendo las profundas) usualmente se entrenan utilizando métodos basados en el gradiente y retropropagación.

¿Qué ventajas proporciona disponer de un aprendizaje basado en el conocimiento disponible de la plasticidad frente a los métodos "artificiales"? ¿Por qué en la actualidad se han descartado a los métodos basados en la regla de Hebb frente a los métodos basados en el gradiente? Estas preguntas forman parte de la discusión general existente entre hasta qué punto se debe tratar de emular a los sistemas biológicos para crear inteligencia artificial, generalmente enmarcado en la discusión del conexionismo contra el enfoque simbólico de la Inteligencia Artificial (véase, por ejemplo [14] para una breve discusión de la situación actual del debate en un área concreto). En este caso, sin embargo, tenemos un caso donde dos enfoques conexionistas entran en debate sobre la forma en que deben aprender las neuronas, siendo en apariencia el modelo basado en el gradiente el de mayor éxito hasta ahora.

No es la primera vez que dos modelos conexionistas entran en conflicto sobre el nivel de plausibilidad biológica que se necesita para alcanzar los resultados. Por ejemplo, en cuestiones relacionadas con la arquitectura, los modelos de redes neuronales profundas han sido señalados por tener un alto grado de abstracción con respecto a las neuronas biológicas [7], siendo modelos como HMAX (véase [20]) preferidos por algunos autores por tener una construcción más relacionada con los experimentos en la Corteza Visual.

La discusión que se plantea en este artículo es diferente: dadas redes neuronales generales, ¿es preferible disponer de aprendizaje hebbiano sobre aprendizaje basado en el gradiente? Una de las acusadas ventajas que disponen las reglas basadas en Hebb se refieren a la capacidad de implementarse en tiempo real [4], pero su principal desventaja es que por lo general no alcanzan el desempeño otorgado por las reglas basadas en el gradiente y en algunos casos la diferencia es muy aguda [5].

¿Por qué un algoritmo biológicamente inspirado no alcanza los resultados de clasificación de un algoritmo aparentemente artificial? Divisamos cinco posibles razones principales por las que esto ocurra:

1. *Los modelos de aprendizaje por plasticidad están incompletos*. Como se ha mencionado anteriormente, los experimentos y otros datos nos han permitido perfeccionar la regla de aprendizaje sináptica que se tiene. No obstante, la misma puede no estar suficientemente bien comprendida. Un posible elemento faltante es el tercer factor que es mencionado por [19] referente a los moduladores de la plasticidad (como es el caso de los neurotransmisores). Este aspecto ha sido abordado por trabajos como [11] y ha sido explorado computacionalmente en problemas de IA por [15] para la base de datos MNIST, obteniendo resultados mejores que la aplicación directa de la regla de Hebb. El problema es que tales intentos no muestran mejores resultados que los trabajos que utilizan retropropagación. Esto puede deberse a que las reglas de aprendizaje tratan de diseñar una forma de aproximarse a los pesos adecuados, mientras que los métodos basados en el gradiente directamente buscan dichos pesos óptimos.

2. *La retropropagación es biológicamente inspirada*. Como si fuera una convergencia evolutiva, es posible que hayamos partido de una definición matemática para encontrar un algoritmo presente en la naturaleza. Este tema es ampliamente discutido en [24]. Sin embargo, podemos ver de manera rápida que el aprendizaje de redes neuronales artificiales actual difiere del aprendizaje de seres humanos, debido a que por lo regular no se implementa en tiempo real.

3. *El modelo de actividad neuronal es simple*. Es posible que no se haya tomado un modelo conveniente para la actividad de las neuronas y dependamos de incluir este factor (y otros como las células gliales, la maduración de las neuronas, etcétera). Una solución en este sentido es el desarrollo de las redes neuronales pulsantes *Spiking Neural Networks* o SNN, véase por ejemplo [18]). Sin embargo, incluso la adición de reglas de plasticidad avanzadas con STDP (como [9]) no logran alcanzar los resultados logrados por otros modelos como los basados en el gradiente.

4. *La evolución juega un gran papel*. ¿Y si los valores de pesos como por ejemplo los kernels de convolución de la corteza visual están mediados por mecanismos diferentes a la plasticidad para su desarrollo? Esta pregunta requiere de una respuesta en el área de biología pero la cuestión que se plantea aquí es que la evolución puede ser determinante para el aprendizaje hebbiano pueda ajustarse adecuadamente. Sin embargo, a nivel algorítmico, implementar un esquema de evolución para los pesos parece ser excesivo.

5. *La arquitectura es inadecuada*. Posiblemente el éxito del aprendizaje profundo reside en su arquitectura compuesta por múltiples capas convolucionales y millones de parámetros, que se acercan de alguna manera a la realidad del cerebro, el cual está compuesto por un número astronómico de neuronas. Al parecer, incluso un modelo simple pero con una gran cantidad de parámetros suena suficiente e incluso mejor que un modelo pequeño pero con aspectos más basados en la biología, puesto que obvian el aspecto fundamental que es la inclusión de millones de neuronas. Para resolver este problema, en conjunto con los puntos anteriores 2 y 4, en [4] planteamos una red convolucional con pesos pre-entrenados y aprendizaje hebbiano en la capa final, obteniendo resultados cercanos a los logrados utilizando algoritmos recientes basados en el gradiente. Sin embargo, si bien los resultados se acercaron a lo logrado por los métodos basados en el gradiente, no fue posible superarlos.

En apariencia, los esfuerzos por generar algoritmos biológicamente más plausibles para modelos conexionistas parecen acercarse asintóticamente a los resultados logrados vía descenso del gradiente pero sin lograr aún un resultado contundente que los supere. Esto no quiere decir que las propuestas estén mal fundadas, ya que pueden incluir otras ventajas como el aprendizaje en tiempo real. Sin embargo, la pregunta sigue abierta y no creemos que una solución no exista puesto que los seres humanos somos capaces de realizar tales actividades de reconocimiento de imágenes sin dificultad. No obstante, incluso esta idea puede estar mal fundada ya que los modelos de redes profundas pueden superar a la exactitud humana como clama el artículo de [12]. De esta manera, es posible que los algoritmos puedan estar sobrecalificados para tareas específicas y disponer de una buena aproximación a los mismos con arquitecturas convenientes puede llegar a ser una solución conveniente y no tratar de aspirar a entender cómo las redes biológicas logran una mayor capacidad de clasificación.

En este artículo, trataremos de aproximarnos a las preguntas anteriormente planteadas. La aproximación que haremos será referente a la última posible respuesta que se planteó, indicando que la arquitectura que se maneja es insuficiente para resolver los problemas de clasificación. Hemos visto que dada una arquitectura arbitraria, el descenso del gradiente suele obtener mejores resultados. Sin embargo, ¿existirá una arquitectura apropiada para la regla de Hebb con la que se puedan obtener los resultados que se observan con métodos basados en el gradiente? Este acercamiento que se dará será de corte principalmente teórico, pero se mostrará una implementación de las ideas abordadas.

2 Preliminares teóricos

La descripción matemática de la Regla de Hebb estará principalmente basada en [8], salvo donde se indique. Por simplicidad, partiremos del modelo de tasa de disparo para representar a la actividad de cada neurona individual. De manera alternativa, se pueden utilizar *spikes* (potenciales de acción) para modelar la actividad neuronal, como se realiza en [18]. Sea $y \in D \subset \mathbb{R}^+$ la actividad de una neurona (usualmente postsináptica) que recibe un vector de señales aferentes $\mathbf{x} = (x_1, \cdots, x_m)$, las cuales pueden tener origen sensorial (datos de sensores o de bases) o de otras neuronas. El conjunto D puede ser discreto ($\{0, 1\}$), un intervalo continuo ($[0, 1]$), o bien el mismo conjunto de los números reales no negativos. La actividad (tasa de disparo) está modelada por la siguiente ecuación diferencial

$$\tau_y \frac{dy}{dt} = -y + a(\mathbf{w} \cdot \mathbf{x}),$$

donde $\mathbf{w} = (w_1, \ldots, w_m)$ representa el vector de pesos de las entradas, y la constante satisface $\tau_y \approx 0$. Para linealizar la ecuación, se suele tomar $\tau_y = 0$, lo cual lo convierte en el modelo estándar de red neuronal artificial y de esta forma se trabajará en este artículo. $a : \mathbb{R} \to D$ representa a la función de activación y en este caso se tomará por la función identidad cuando utilicemos $D = \mathbb{R}^+$ o sigmoide si $D = [0, 1]$. Para fines prácticos, en este artículo se tomará a la función de activación como la identidad.

2.1. Regla de Hebb Simple

Un modelo sencillo de la Regla de Hebb ha sido llamado como Simple o Básico y tiene como objetivo encapsular a las propiedades esenciales de la conjetura hebbiana, resumidas por Shatz [3] como "*cells that fire together, wire together*". La regla de Hebb Simple está dada por

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{x}y.$$

Es regla lograr representar bien la idea básica, puesto que si $x_i, y > 0$, entonces w_i aumenta (y el aumento es mayor si ambos tienen alta actividad), pero si uno es 0, entonces no se produce ningún cambio. Por ejemplo, si $D = \{0, 1\}$, entonces $x_i = y = 1$ representan encendidos simultáneos que se traducen en un aumento.

Utilizando el método de Euler podemos discretizar a la regla de Hebb para su implementación:

$$\mathbf{w}(t + 1) = \mathbf{w}(t) + \alpha \mathbf{x}(t)y(t), \tag{1}$$

donde $\alpha > 0$ es la tasa de aprendizaje y $\mathbf{w}(0)$ puede tomarse como $\mathbf{0}$, aunque también puede recibir una inicialización aleatoria, la cual es una opción que evita la inactividad total que podría tener una interneurona si los pesos fueran todos inicializados en cero.

3 Resultados teóricos

En esta sección, probaremos algunos resultados sobre la regla de Hebb relacionados con la clasificación supervisada, la cual es un tema relevante puesto que en general dada una red arbitraria y de inicialización en ceros, una red de entrenamiento hebbiano no supera a una red neuronal con entrenamiento por descenso de gradiente como se muestra en [5]. Esto quiere decir, que si definimos a nuestro *dataset* como $\{(\mathbf{x}(i), y(i))\}_{i=1}^{n}$ $(y(i) \in \{0, \ldots, k\})$ y a $\hat{y} = NN_{\mathbf{w}}(\mathbf{x})$ como la salida de una red neuronal de una capa, el entrenamiento hebbiano no converge a algún mínimo de la función $L(\mathbf{w}) = \sum_{i=1}^{n} \mathbf{1}_{y(i) \neq NN_{\mathbf{w}}(\mathbf{x}(i))}$, como sí lo hacen los enfoques basados en el gradiente. No obstante, eso no significa que no exista una arquitectura competente capaz de efectuar adecuadamente el aprendizaje hebbiano.

De esta forma, de la ecuación (1) observamos que, con la base de datos definida,

$$\mathbf{w}_j(n) = \sum_{i=1}^{n} \mathbf{x}(n) y_j(n).$$

En este caso, redefinimos $y_j = 1$ si $y = j$ y $y_j = 0$ si $y \neq j$, teniendo una capa de salida de k neuronas. $\mathbf{w}_j$ se refiere a los pesos de la j-ésima neurona de salida y las entradas. Asimismo, es posible definir $\mathbf{w}_j = \mathbf{w}_j(n)$, que se refiere a los pesos después de un entrenamiento supervisado. Adicionalmente, añadiremos una capa de clasificación, de tal forma que

$$NN_{\mathbf{w}}(\mathbf{x}) = \arg \max_{j \in \{1,\ldots,k\}} y_j$$

3.1. Red hebbiana infinita

Comenzaremos nuestra descripción con una red hebbiana construida de la siguiente forma: a cada valor del espacio vectorial $\mathbf{z} \in D^m \subset \mathbb{R}^m$, le daremos una representación geométrica de la siguiente forma: consideremos una nueva capa de neuronas $r_{\mathbf{z}}$ dada por

$$r_{\mathbf{z}}(\mathbf{x}) = \begin{cases} 1, & |\mathbf{z} - \mathbf{x}| = 0 \\ 0, & |\mathbf{z} - \mathbf{x}| \neq 0 \end{cases}$$

Esta capa debe distinguirse de la capa de entrada, consistente en m neuronas. En su lugar, está conformada por todos los valores posibles que $\mathbf{x}$ puede tener. Esto quiere decir que la capa de neuronas tiene $|D|^m$ valores posibles, lo cual es un número potencialmente grande incluso si $|D| = 2$, e infinito si D lo es. A pesar de lo anterior, dicha capa infinita nos servirá como punto de partida y como un modelo teórico de la importancia de la Regla de Hebb. En términos especulativos, dicha capa puede ser imaginada como la inteligencia de un ser infinito. No obstante, en algún momento tendremos que reducir la dimensionalidad de capa infinita, puesto que no nos interesa estudiar la inteligencia de los seres metafísicos sino de los existentes en el mundo discreto.

La capa de salida para la red hebbiana infinita, la podemos definir expandiendo el concepto de producto punto:

$$y_j = \sum_{\mathbf{z}} w_{j,r_{\mathbf{z}}} r_{\mathbf{z}}(\mathbf{x}).$$

Primeramente, demostraremos la existencia de un minimizador global para posteriormente probar que la regla de Hebb induce a uno.

Proposición 1: Sea $g : D^m \to \{0, \ldots, k\}$ definida para cada valor $\mathbf{a} = (a_1, \ldots, a_m)$ como

$$g(\mathbf{a}) = \arg \max_{j \in \{1,\ldots,k\}} \sum_{i=1}^{n} v(x_1(n) = a_1, \ldots, x_m(i) = a_m, y_j(n) = 1),$$

donde v es una función de verdad binaria. Sea $\mathcal{F} = \{f | f : D^m \to \{0, \ldots, k\}\}$. Entonces

$$g = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathbf{1}_{y(i) \neq f(\mathbf{x}(i))}$$

Notemos que $\sum_{i=1}^{n} \mathbf{1}_{y(i)\neq f(\mathbf{x}(i))} \in \mathbb{N}$. Por lo tanto, existe g^* minimizador global. Supongamos que $g^* \neq g$. Entonces existe $\mathbf{a} = (a_1, \ldots, a_m)$ tal que $g^*(\mathbf{a}) = l$ y

$$\sum v(x_1(i) = a_1, \ldots, x_m(i) = a_m, y_j(i) = j) \leq \sum v(x_1(i) = a_1, \ldots, x_m(i) = a_m, y_l(i) = 1)$$

Hagamos $g^{\star\star}(\mathbf{x}) = g^*(\mathbf{x})$ para $\mathbf{x} \neq \mathbf{a}$ y para $\mathbf{x} = \mathbf{a}$, tomemos $g^{\star\star}(\mathbf{a}) = \sum v(x_1(i) = a_1, \ldots, x_m(i) = a_m, y_j = 1)$. Entonces la función cuenta con menor error que g^*, lo cual contradice su minimalidad. □

Finalmente probaremos que la red hebbiana infinita $H : D^m \to \{0, \ldots, k\}$ es un minimizador global para la función de costo previamente definida:

Teorema 1: Sea $H : D^m \to \{0, \ldots, k\}$ una red hebbiana infinita con $D = \{0, 1\}$. Entonces

$$H = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathbf{1}_{y(i)\neq f(\mathbf{x}(i))}$$

Sea $\mathbf{a} \in D^m$. Entonces

$$\begin{aligned} w_{j,r_{\mathbf{a}}} &= \sum_{i=1}^{n} r_{\mathbf{z}}(\mathbf{a}(i)) y_j(i) \\ &= \sum_{i=1}^{n} v(x_1(i) = a_1, \ldots, x_m(i) = a_m, y_j(i) = 1). \end{aligned}$$

de donde vemos que

$$\begin{aligned} H(\mathbf{a}) &= \arg\max_{j \in \{1,\ldots,k\}} y_j \\ &= \arg\max_{j \in \{1,\ldots,k\}} \sum_{\mathbf{z}} w_{j,\mathbf{z}} r_{\mathbf{z}}(\mathbf{a}) \\ &= \arg\max_{j \in \{1,\ldots,k\}} w_{j,\mathbf{a}}, \end{aligned}$$

siguiéndose el resultado por la Proposición 1. □

A pesar de que la prueba de los anteriores resultados es relativamente directa, las consecuencias son importantes en el contexto de la regla de Hebb simple. A *grosso modo*, establece que si disponemos de recursos ilimitados, es posible diseñar una red neuronal de aprendizaje hebbiano que minimice global la suma de los errores totales que se cometen. La idea básica resultó ser en crear una capa de representación con todos los valores posibles de entrada y la regla de Hebb cuenta cuántos elementos de la base de datos arrojan dichos valores y realiza una actualización hebbiana cuando encuentra uno.

Por situar un ejemplo, podemos tomar una base de datos consistente en los siguientes datos:

$$\mathcal{B} = \{(0,0), (1,0), (2,0), \cdots, (127,0), (128,1), \cdots, (254,1), (255,1)\}.$$

El primer elemento de cada vector representa a la entrada x_i y el segundo a la etiqueta y_i. Para dar una interpretación a esta base de datos hipotética, podemos entender a la entrada como una tonalidad en escala de grises y la salida como negro si $y_i = 0$ y blanco si $y_i = 1$[1]. Este problema de clasificación puede resolverse de manera simple con cualquier algoritmo y el la red hebbiana infinita no es la excepción. La segunda capa de la red (representación) estará formada por cada posible valor de x, en este caso 255 neuronas. Finalmente el aprendizaje hebbiano se da en la última capa, compuesta por una neurona y_0 que clasifica el negro y y_1 que clasifica el blanco. Notemos que $w_{0,z} = 1$ (y $w_{1,z} = 0$) para $z = 0, \ldots, 127$ porque para los primeros datos tenemos $r_z(x) = 1$ y $y_0 = 1$ para $x \in \{0, \ldots, 127\}$ pero a partir de 127 observamos $y_1 = 1$, por lo que $w_{0,z} = 1$ para $z > 127$ pero $w_{1,z} = 1$. De esta forma, el error el conjunto de entrenamiento será 1, ya que cada valor posible que puede tomar x está perfectamente clasificado. Aún más, si suministramos datos difusos, por ejemplo $(127, 0)$ con $(127, 1)$ repetidos

[1]También podemos distinguir los estados iluminado / poco iluminado para evitar pensar en el gris.

varias veces, la red tenderá por el valor más repetido.

No obstante, este ejemplo de juguete muestra algunas de las desventajas propias del método. La primera es la acusada falta de datos, que tiene repercusiones sobre el conjunto de prueba. Si, por ejemplo utilizamos un conjunto de entrenamiento con los datos pares de $\mathcal{B}$ y un conjunto de prueba con los impares, el error en el conjunto de entrenamiento será 0 pero cerca de la mitad de los datos del conjunto de prueba serán incorrectamente clasificados, ya que no se suministran los ejemplos concretos y por lo tanto todos los pesos están en 0.

Este problema puede pensarse que se puede resolver suministrando aún más datos, pero aparece otro problema en la práctica, que es la explosión combinatoria en la capa de representación. Supongamos ahora que queremos clasificar 7 colores RGB. Entonces tenemos $256^3 = 16777216$ valores posibles, lo cual es una gran cantidad de neuronas incluso para algunos organismos vivos. Una posible solución reside en compactar a la representación reduciendo la dimensión al redondear a unos cuantos valores, lo cual también reduce el problema del conjunto de prueba mencionado previamente. Sin embargo, el problema persiste: si quisiéramos clasificar 10 imágenes en un espacio de 7×7 que es un caso reducido de MNIST y de entrada binaria, necesitamos un total de 5.62×10^{14} neuronas de representación, es decir, más neuronas que las existentes en los cerebros humanos (8.6×10^{10}). Reducir imágenes de caracteres a un tamaño de 7×7 es incluso excesivo y una reducción mayor simplemente no se puede clasificar fácilmente. Este caso se complica cuando consideramos imágenes RGB de objetos. Por lo tanto, este enfoque es irrealizable en la práctica computacional y en la biología misma, donde formas más eficientes toman lugar.

A pesar de las dos desventajas agudas, el objetivo del Teorema 1 es probar que la existencia de una arquitectura en el que el aprendizaje hebbiano logre una minimización perfecta (global) de la función de costo suma de los errores. Esto responde de alguna manera a la pregunta fundamental planteada en este artículo, señalando que el principal problema del aprendizaje hebbiano recae en la necesidad de hallar una arquitectura conveniente para su ejecución. Al menos sabemos que existe una arquitectura donde opera perfectamente, quizá exista una arquitectura intermedia que resuelva el problema de clasificación de manera efectiva y eficiente.

3.2. m-celdas

Una posible forma en que se pueden tratar de minimizar los efectos de la red hebbiana infinita en términos prácticos (los seres infinitos pueden ejecutarla sin problemas) consiste en crear dividir a los datos de la capa de representación en m-celdas (mejor conocidas en la literatura como k-celdas, pero en esta caso $k = m$), que es el enfoque que se trató en el artículo de [5], logrando resultados experimentales comparables con otros métodos de clasificación estándares o incluso mejores para datos sintéticos. En esta sección, intentaremos brindar un mejor sustento matemático sobre por qué el método funciona y finalmente lo aplicaremos en datos no sintéticos.

En general la idea de las m-celdas se reduce a dividir al espacio $\mathbb{R}^m$ primero en un espacio compacto donde se asume que viven los datos y después se subdividen en m-celdas de radio δ. Cada celda se puede entender como una neurona que recibe datos y se activa si los datos de entrada pertenecen a la m-celda. Estas celdas forman la capa de representación que se describió en la red hebbiana infinita. Posteriormente, estas celdas se conectan con las neuronas de salida y en la conexión de las celdas con las neuronas de salida se aplica aprendizaje hebbiano.

La siguiente proposición establece que es posible aproximarnos a la medida de Lebesgue μ de un conjunto abierto y acotado utilizando un número finito de m-celdas. La notación que se usará para las celdas es $I_i(\delta)$ si tiene el índice i y radio δ, o bien $I_{\mathbf{x}}(\delta) = \prod_{i=1}^{m}(x_i - \delta, x_i + \delta)$.

Proposición 2: Sea $U \subset \mathbb{R}^m$ un conjunto abierto y acotado, y $\varepsilon > 0$. Entonces existe $\delta > 0$ tal que existen d m-celdas de radio δ tales que

$$\Big|\mu(U) - \mu\big(\bigcup_{i=1}^{d} I_i(\delta)\big)\Big| < \varepsilon.$$

El conjunto de m-celdas abiertas de tamaño arbitrario forman una base del espacio $\mathbb{R}^m$ con la topología de los conjuntos abiertos.

Consideremos el conjunto

$$A = \{\mathbf{x} \in U : x_i \in \mathbb{Q}, i = 1, \ldots, m\}$$

Para cada $\mathbf{x} \in A$ existe $I_{\mathbf{x}}(\rho) = \prod_{i=1}^{m}(x_i - \rho, x_i + \rho)$ tal que $I_x(\rho) \subset U$. Consideremos $I_{\mathbf{x}}(\rho) = \emptyset$ si $I_{\mathbf{x}}(\rho) \not\subset U$. Entonces $U = \bigcup_{\mathbf{x}\in A_r} \bigcup_{\rho\in\mathbb{Q}^+} I_{\mathbf{x}}(\rho)$. Así, tenemos una unión contable de k-celdas. En particular, los elementos $I_i(\delta_i)$ pueden tomarse disjuntos ya que si $I_i(\delta_i) \cap I_j(\delta_j) \neq \emptyset$, podemos definir $I_i(\delta_i) \cap I_j(\delta_j)$ como la unión de otras m-celdas, o aproximarse tan bien como se desee, según el procedimiento que se expone al final de esta prueba. Reindexando y ordenando de mayor a menor δ_i:

$$U = \bigcup_{i=1}^{\infty} I_i(\delta_i).$$

Por lo tanto,

$$\begin{aligned}\mu(U) &= \sum_{i=1}^{\infty} \mu(I_i(\delta_i)) \\ &= \sum_{i=1}^{\infty} (2\delta_i)^m \\ &= \lim_{q\to\infty} \sum_{i=1}^{q} 2^m \delta_i^m.\end{aligned}$$

Entonces, como U es acotado, $\mu(U) < \infty$ y para $\varepsilon/2 > 0$ existe $c \in \mathbb{N}$ tal que

$$\Big|\mu(U) - \mu\big(\bigcup_{i=1}^{c} I_i(\delta_i)\big)\Big| < \frac{\varepsilon}{2}.$$

Ahora, falta aproximar $\bigcup_{i=1}^{c} I_i(\delta_i)\big)$. Para cada δ_i tomemos una aproximación racional $\hat{\delta_i} = \frac{a_i}{b_i}$ de $2^m\delta_i^m$ con radio $\frac{\varepsilon}{2c}$. Notemos que si $\delta = \frac{1}{\prod_{i=1}^{c} b_i}$ entonces

$$\hat{\delta_j} = a_j \prod_{i\neq j} b_i \delta,$$

de donde, para $d_j = a_j \prod_{i\neq j} b_i$

$$I_i(\hat{\delta_j}) = \bigcup_{i=1}^{d_j^m} I_{\mathbf{x}_i}(\delta).$$

donde $\{I_{\mathbf{x}_i}(\delta)\}$ es una partición de $I_i(\hat{\delta_j})$ de tamaño δ. De esta forma,

$$\Big|\mu(I_i(\delta_j)) - \mu\big(\bigcup_{i=1}^{d_j^m} I_{\mathbf{x}_i}(\delta)\big)\Big| < \frac{\varepsilon}{2c}$$

Así, por la desigualdad del triángulo y tomando $d = \sum_{i=1}^{c} d_j^m$

$$\Big|\mu(U) - \mu\big(\bigcup_{i=1}^{d} I_i(\delta)\big)\Big| < \varepsilon.$$

□

De manera similar al último argumento, podemos fragmentar por completo al espacio $\mathbb{R}^m$ de tal forma que se aproxime tanto como se puedan a las m-celdas encontradas en la prueba anterior. De este modo, al reducir δ hemos de encontrar una clasificación más fina que permita obtener resultados más precisos.

Para poder realizar clasificación apropiadamente, debemos asumir que los datos se pueden separar en vecindades o conjuntos abiertos, lo cual siempre es posible para cualquier dispersión, pero idealmente los datos deben no estar intercalados entre sí para su correcto funcionamiento en el conjunto de prueba. Este problema también se

observa si las distribuciones de los datos coinciden y se superponen, pero en general el conjunto de entrenamiento puede sobreajustarse y resolver el problema en el mismo.

Dado que estamos considerando espacios continuos como $\mathbb{R}^m$, utilizando el supuesto de separación de datos en conjuntos abiertos, definiremos el error como

$$L(\mathbf{w}) = |\mu(U) - \mu(V_{\mathbf{w}})|$$

donde U es el conjunto abierto donde cada $\mathbf{x} \in U$ si y sólo si pertenece a la clase j y $V_{\mathbf{w}}$ es el conjunto tal que para toda $\mathbf{x} \in V_{\mathbf{w}}$, $NN_{\mathbf{w}}(\mathbf{x}) = j$. Este error se puede reducir tanto como se desee:

Teorema 2: Supongamos que el problema de clasificación se puede separar en conjuntos abiertos y acotados. Para cada $\varepsilon > 0$, existe una división del espacio E^m en m-celdas de tamaño δ tales que $L(\mathbf{w}) < \varepsilon$.

Usando la Proposición 1, es posible hallar una partición del espacio en k-celdas de tamaño δ tal que

$$\Big|\mu(U) - \mu\Big(\bigcup_{i=1}^{d} I_i(\delta)\Big)\Big| < \frac{\varepsilon}{k}.$$

Como cada $I_i(\delta) \subset U$, para cada $\mathbf{x} \in I_i(\delta)$, $NN_{\mathbf{w}}(\mathbf{x}) = j$, por lo que $L(\mathbf{w}) < \frac{\varepsilon}{k}$. Tomando las k clases, tenemos que $L(\mathbf{w}) < \varepsilon$. (En principio cada δ de cada partición puede ser diferente, pero podemos seleccionar uno usando una técnica similar a la que aparece en la prueba de la Proposición 2). □

En la práctica, la bondad de este resultado (poder minimizar al costo tanto como se quiera) está afectado por varios factores. Primeramente, la superposición de los datos supone un problema, como ya se ha mencionado previamente. Además, cuando tomamos celdas con menor δ si bien aumentamos la precisión el conjunto de clasificación, se produce un sobreajuste al aparecer celdas sin asignación que pueden aparecer en el conjunto de entrenamiento. Sin embargo, nuevamente este problema puede resolverse con una mayor cantidad de datos. El otro problema recae en la definición de m-celdas para dimensiones grandes de m, requiere de definir una cantidad creciente de celdas. Estos dos problemas son análogos a los dos problemas de la red hebbiana infinita, pero al menos aparecen atenuados y en bajas dimensiones son perfectamente implementables, como se verá a continuación.

4 Resultados experimentales

En [5] se implementó el algoritmo de m-celdas para datos sintéticos y se comparó con métodos estándares como la regresión logística y kNN. En esta ocasión, probaremos su desempeño frente a redes neuronales con métodos basados en el gradiente y utilizando un *dataset* no sintético, el cual es la base de datos Iris para la clasificación de flores del género *Iris* a partir de datos biométricos [10]. En este caso, para entrenamiento de las m-celdas se utilizarán únicamente dos dimensiones en cierta medida separables: longitud y ancho de pétalo (véase figura 1), longitud de sépalo y ancho de pétalo (figura 2), y finalmente longitud de sépalo y longitud de pétalo (figura 3).

Como se puede ver, las tres clases del *dataset* son en gran medida separables, aunque existe cierto traslape entre las clases *I. versicolor* e *I. virginica*. Se tiene un total de 150 datos, de los cuales 75 se utilizaron para entrenamiento, 30 para validación y 45 para prueba.

4.1. Descenso de Gradiente

Para contrastar al descenso de gradiente con el aprendizaje hebbiano, se utilizó un esquema de redes neuronales de alimentación hacia adelante (*Feedforward Neural Networks* o FNNs). El entrenamiento se realizó con la función de costo de entropía cruzada categórica a 15 épocas y se utilizó al optimizador Adam [16], uno de los métodos basados en el gradiente más recientes y ampliamente utilizados en la actualidad para redes neuronales. Para este caso, se utilizaron las cuatro dimensiones de los datos. Por ende, se tomarán 4 unidades de entrada y 3 de salida, correspondiente a cada clase.

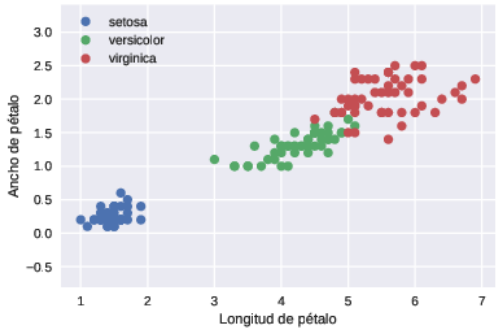


Figura 1: Gráfico de las variables longitud y ancho de pétalo del dataset Iris.

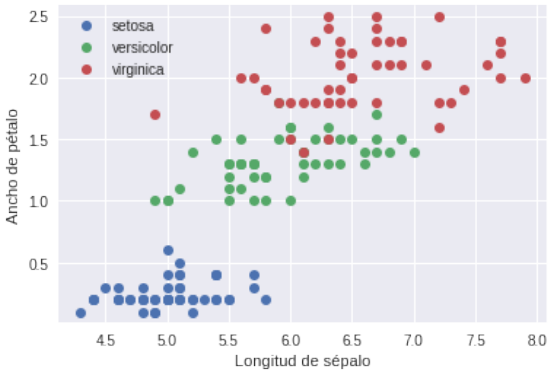


Figura 2: Gráfico de las variables longitud de sépalo y ancho de pétalo del dataset Iris.

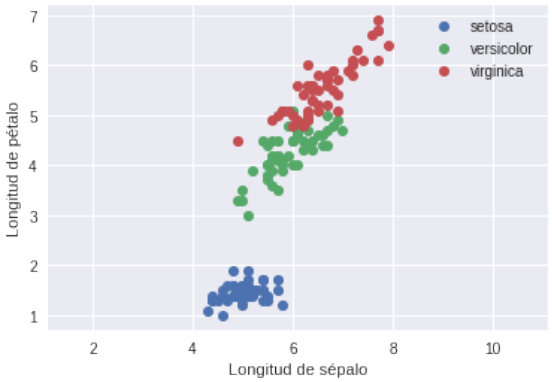


Figura 3: Gráfico de las variables longitud de sépalo y longitud de pétalo del dataset Iris.

Ese enfoque se estudió de dos maneras: por medio de redes neuronales con arquitectura fija y mediante una optimización hiperparamétrica. Las arquitecturas fijas fueron (50), (50, 50) y (50, 50, 50), donde cada vector representa el número de neuronas en cada capa intermedia (oculta): es decir la primera arquitectura (Modelo 1) cuenta con 50 neuronas en una capa intermedia, la segunda 50 en la primera capa oculta y 50 en la segunda (Modelo 2), mientras que el Modelo 3 cuenta con 50 neuronas en tres capas intermedias.

El modelo 1 tardó 1.586 s y alcanzó una exactitud sobre el conjunto de prueba de 0.967, 0.853 en el conjunto de entrenamiento y 0.822 en el conjunto de validación. Por su parte, el modelo 2 logró la misma exactitud sobre el conjunto de prueba, 0.987 en el conjunto de prueba y 0.911 en el conjunto de validación durando 1.489 segundos, mientras que el modelo 3 obtuvo 0.933 en el conjunto de prueba, 0.973 en el conjunto de entrenamiento y 0.866 en la validación, durando 1.733 s. La figura 4 representa las curvas de aprendizaje de los tres modelos.

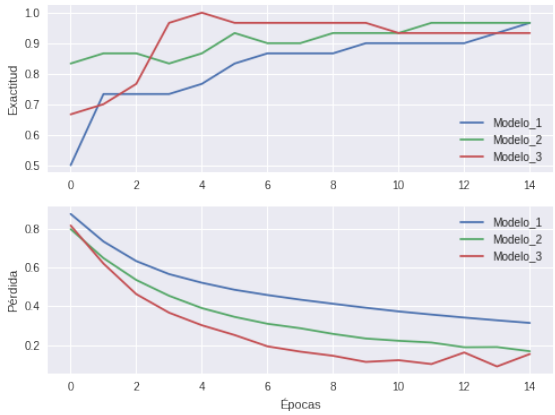


Figura 4: Evolución de la exactitud de los tres modelos base en 15 épocas.

Para realizar una exploración sobre el espacio de las FNNs (optimización hiperparamétrica), se utilizó la heurística de evolución descrita en [6], utilizando la variante A con constantes $\mu = 9$, $M = 4$, $T = 3$ y $k = 10$, así como quince generaciones.

La optimización hiperparamétrica arrojó como arquitectura con máxima validación a $(50, 10, 10, 10)$, la cual logró 0.967 de exactitud en el conjunto de prueba, 0.853 en el conjunto de entrenamiento y 0.889 en el conjunto de validación. El esquema de evolución queda patente en la figura 5. En la figura 6 podemos apreciar la evolución del número de redes neuronales por cada generación que participa en el proceso de evolución.

4.2. Algoritmo de m-celdas

En el caso de las m-celdas, se tomaron celdas de tamaño $\frac{1}{s}$ y únicamente se tomaron dos dimensiones por simplicidad. Los resultados principales se resumen en la tablas 1, 2, 3 y se obtuvieron al variar s. Los mejores resultados se obtuvieron al utilizar las variables de los pétalos (longitud y ancho), donde si seleccionamos la máxima exactitud sobre el conjunto de validación, obtenemos 0.967 de exactitud sobre el conjunto de prueba, aunque para $s = 3$ se obtiene una clasificación perfecta. Los tiempos de ejecución fueron considerablemente más bajos: 0.00072 para la primera combinación de variables, 0.00055 para la segunda y 0.001 para la tercera. Las otras combinaciones de variables en general no resultaron lo suficientemente apropiadas, pero el uso del conjunto de validación nos puede ayudar a seleccionar el par de variables más significativo, aunque una opción más apropiada sería considerar a todas las variables.

Por otro lado, en las tablas observamos cómo al reducir el tamaño de δ, obtenemos mejores resultados de clasificación en el conjunto de entrenamiento, dando confirmación experimental del Teorema 2, pero al reducirse el tamaño de las vecindades también dejamos fuera a los datos que no son de entrenamiento, por lo que las

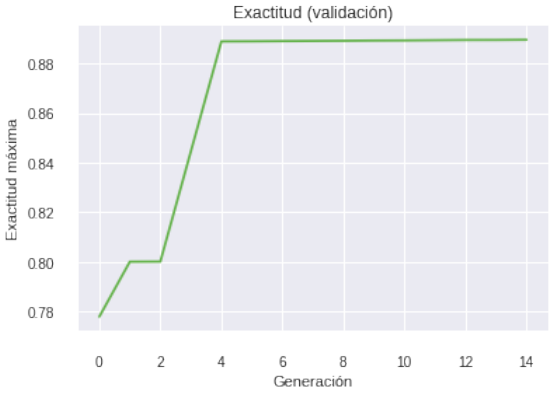


Figura 5: Evolución de la exactitud con 15 generaciones.

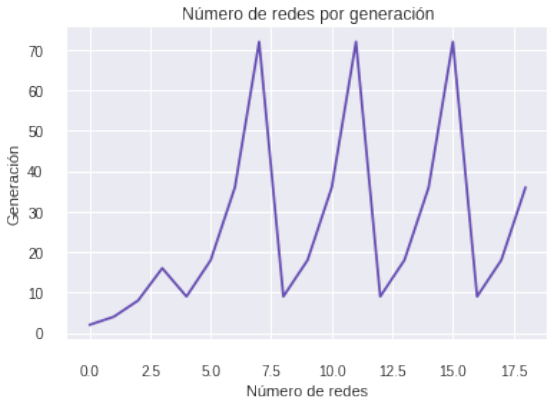


Figura 6: Número de individuos por generación. Se incluyen generaciones donde se da un decremento.

s	Entrenamiento	Validación	Prueba
1	0.893	0.933	0.9
2	0.947	0.956	0.967
3	0.973	0.933	1
4	0.987	0.933	0.966
5	0.973	0.8	0.933
6	0.986	0.911	0.867
7	0.987	0.867	0.867
8	0.987	0.8	0.633
9	0.987	0.756	0.756
10	0.987	0.756	0.6

Tabla 1: Resultados de clasificación utilizando el algoritmo de m-celdas con aprendizaje hebbiano y diferentes valores de s con las variables longitud y ancho de pétalo.

s	Entrenamiento	Validación	Prueba
1	0.84	0.867	0.833
2	0.947	0.867	0.933
3	0.973	0.867	0.9
4	0.973	0.689	0.767
5	0.973	0.667	0.667
6	0.973	0.644	0.7
7	0.973	0.667	0.633
8	0.973	0.511	0.533
9	0.973	0.511	0.533
10	0.973	0.511	0.533

Tabla 2: Resultados de clasificación utilizando el algoritmo de m-celdas con aprendizaje hebbiano y diferentes valores de s con las variables longitud de sépalo y ancho de pétalo.

s	Entrenamiento	Validación	Prueba
1	0.813	0.822	0.9
2	0.88	0.867	1
3	0.96	0.911	0.767
4	0.96	0.933	0.966
5	0.96	0.556	0.667
6	0.986	0.644	0.633
7	0.987	0.733	0.667
8	0.987	0.511	0.567
9	1.0	0.422	0.533
10	1.0	0.4	0.533

Tabla 3: Resultados de clasificación utilizando el algoritmo de m-celdas con aprendizaje hebbiano y diferentes valores de s con las variables longitud de pétalo y longitud de sépalo.

exactitudes sobre los conjuntos de prueba y validación se reducen, tendiendo a $\frac{1}{3}$. La primera tabla también está representada en la figura 7.

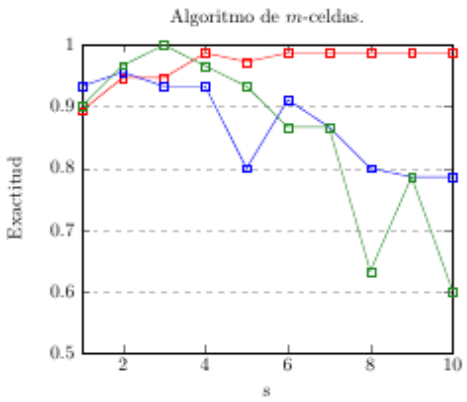


Figura 7: Gráfico de las exactitudes del algoritmo de m-celdas con las variables longitud y ancho de pétalo del dataset Iris.

5 Conclusiones

Los resultados observados con las m-celdas dan sustento a la hipótesis de que la arquitectura tiene una influencia crucial en las redes neuronales con aprendizaje hebbiano, a pesar de que sea un método poco práctico de implementar para dimensiones mayores. En este caso se implementó para $m = 2$, siendo posible ya que se puede no utilizar toda la información de una base de datos, aunque no siempre parece posible lograr resultados de clasificación correctos utilizando pocas dimensiones o reduciéndolas.

En este caso, para la base de datos de iris, se pudieron observar resultados similares pero significativamente más rápidos que los que arroja el optimizador Adam (una versión moderna del descenso del gradiente), incluso utilizando un esquema de evolución. Esto supone una primera victoria de los métodos basados en Hebb frente al enfoque predominante y es respaldado por los resultados teóricos. Dichos resultados empíricos también son comparables con otros algoritmos de clasificación modernos, como los aplicados en [25].

No obstante, el optimismo debe tomarse con cautela. Para mayores dimensiones, parece complicada la ejecución del algoritmo de m-celdas y el refinamiento del radio δ requiere de una mayor capacidad de cómputo. De esta forma, quizá sólo se pueda aspirar a celdas burdas para dimensiones medianas, pero parece imposible manejar $m = 20 \times 20$, que es el caso de las imágenes.

Sin embargo, a pesar de las limitaciones del método propuesto debemos señalar que el logro obtenido tanto a nivel teórico como práctico señala la existencia de una red en donde el aprendizaje hebbiano opera apropiadamente. Esto puede darnos esperanzas para sospechar la existencia de una red óptima en el que el aprendizaje hebbiano no sólo funcione adecuadamente sino que la red sea eficiente para su implementación práctica en dimensiones altas. Tal red puede tener una estructura convolucional y disponer de millones de parámetros. En el trabajo previo de [4] se ha tratado de dar dicha aproximación. Quizá otros componentes como el tercer factor sean necesarios para lograr el delicado balance entre eficacia y eficiencia.

Disponibilidad de código y datos

El código principal se puede obtener de `https://github.com/Pherjev/hebbian-m-cells/blob/master/HebbMNIST-2.ipynb`. La base de datos se obtuvo directamente de la librería sklearn.

Referencias

[1] E. L. Bienenstock, L. N. Cooper, and P. W. Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 1:32 – 48, 1982.

[2] T. V. P. Bliss and T. Lomo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology*, 232(2):331 – 356, 1973.

[3] T. V. P. Bliss and T. Lomo. The developing brain. *Sci. Am.*, 267, 2010.

[4] F. J. A. Canto. Convolutional neural networks with hebbian-based rules in online transfer learning. *Mexican International Conference on Artificial Intelligence*, pages 35 – 49, 2020.

[5] F. J. A. Canto. Eficacia de diferentes reglas hebbianas en el aprendizaje supervisado: Efficacy of different hebbian rules in supervised learning. *Tecnología Educativa Revista CONAIC*, 7(1):92 – 97, 2020.

[6] F. J. A. Canto. Redes neuronales evolutivas con modelos de lotka-volterra. *Research in Computing Science*, 149(8):1179 – 1194, 2020.

[7] R. M. Cichy, , and D. Kaiser. Deep neural networks as scientific models. *Trends in cognitive sciences*, 23(4):305 – 317, 2019.

[8] P. Dayan and L. F. Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational Neuroscience Series, 2001.

[9] P. U. Diehl and M. Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9(99), 2015.

[10] R. A. Fisher and M. Marshall. Iris data set. *RA Fisher, UC Irvine Machine Learning Repository*, 440, 1936.

[11] A. Foncelle, A. Mendes, J. Jedrzejewska-Szmek, S. Valtcheva, H. Berry, K. T. Blackwell, and L. Venance. Modulation of spike-timing dependent plasticity: towards the inclusion of a third factor in computational models. *Frontiers in Computational Neuroscience*, 12(49), 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, pages 1026 – 1034, 2015.

[13] D. O. Hebb. *The Organisation of Behaviour: a neuropsychological theory*. John Wiley & Sons-Chapman & Hall, 1949.

[14] P. Hitzler, F. Bianchi, M. Ebrahimi, and M. K. Sarker. Neural-symbolic integration and the semantic web. *Semantic Web – Interoperability, Usability, Applicability an IOS Press Journal*, 11, 2020.

[15] R. Holca-Lamarre, J. Lücke, and K. Obermayer. Models of acetylcholine and dopamine signals differentially improve neural representations. *Frontiers in Computational Neuroscience*, 2017(54), 11.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] J. Konorski. *Conditioned Reflexes and Neuron Organizaion*. Cambridge University Press, 1948.

[18] K. Kozdon and P. Bentley. The evolution of training parameters for spiking neural networks with hebbian learning. *Artificial Life Conference Proceedings*, pages 276 – 283, 2018.

[19] U. Kumierz, T. Isomura, and T. Toyoizumi. Learning with three factors: modulating hebbian plasticity with errors. *Current opinion in neurobiology*, 46:170 – 177, 2017.

[20] C. Liu and F. Sun. Hmax model: A survey. *IEEE. 2015 International Joint Conference on Neural Networks*, pages 1 – 7, 2015.

[21] T. Lomo. Frequency potentiation of excitatory synaptic activity in dentate area of hippocampal formation. *Blackwell Science*, 128, 1966.

[22] H. Markram, W. Gerstner, and P. J. Sjöström. Spike-timing-dependent plasticity: A comprehensive overview. *Frontiers in Synaptic Neuroscience*, 4, 2012.

[23] E. Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267 – 273, 1982.

[24] L. Timothy P, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21:335 – 346, 2020.

[25] Y. Wu, J. He, Y. Ji, G. Huang, H. Yao, P. Zhang, W. Xu, M. Guo, and Y. Li. Enhanced classification models for iris dataset. *Procedia Computer Science*, 162:946 – 954, 2019.

[26] S. R. y Cajal. The croonian lecture: la fine structure des centres nerveux. *Proceedings of the Royal Society of London*, 55(331-335):444 – 468, 1894.

¿Quieres publicar artículos, información sobre eventos o noticias en el boletín?

La Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C. (SMCCA), convoca a toda la comunidad interesada en el área de la Computación Científica y sus Aplicaciones, a presentar noticias, información sobre eventos, artículos de divulgación e investigación de alta calidad en el era, así como reportes de trabajos de tesis de nivel licenciatura y posgrado en Matemáticas Aplicadas.

Requisitos para la colaboración en el Boletín

I. Artículos de Divulgación e Investigación.
   a) Los artículos que se envíen para ser publicados deberán ser inéditos y no haber sido ni ser sometidos simultáneamente a la consideración en otras publicaciones.
   b) Todos los artículos son sometidos a una revisión por expertos en estas áreas de instituciones nacionales e internacionales.
   c) Los artículos a presentarse deben de ser enviados por medio de la página del Boletín: https://www.scipedia.com/sj/smcca
   d) En la página de la sociedad se puede encontrar la plantilla de LaTeX para la correcta escritura de artículos.

II. Información sobre eventos.
   a) Los eventos cuya información quiera ser publicada para promocionarlos, deberán estar relacionados con el área de las Matemáticas Aplicadas y la Computación Científica.
   b) La información debe enviarse en un archivo de imagen: PDF, JPG, PNG.
   c) La información no deberá exceder una cuartilla.
   d) Enviar la información con al menos 6 meses de anticipación a la fecha en que se llevaría a cabo.

III. Noticias.
   a) Las noticias a ser publicadas en el Boletín deben ser noticias relevantes de actividades de la SMCCA, Socios, Comunidad Científica interesada en las Matemáticas y Computación Científica.
   b) La información de las noticias debe enviarse en un archivo de imagen: PDF, JPG, PNG.
   c) La información no deberá exceder una cuartilla.

El material de colaboración, noticias e información de eventos, deberán ser dirigidos al Dr. Gerardo Tinoco Guerrero al correo electrónico de la SMCCA: smcca@smcca.org.mx

Todos los artículos son sometidos a evaluación por especialistas de instituciones nacionales e internacionales y su publicación estará sujeta a la disponibilidad de espacio en cada número. Las demás colaboraciones se someterán a corrección de estilo y su publicación estará sujeta a la disponibilidad de espacio en cada número. Sólo se aceptará el material enviado que cumpla con todos los requisitos anteriormente señalados.

El envío de cualquier colaboración al Boletín implica no solo la aceptación de lo establecido en este documento, sino también la autorización al Comité Editorial del Boletín de la SMCCA para incluirlo en su página electrónica, reimpresiones, colecciones y cualquier otro medio que permita lograr una mayor y mejor difusión.

Sociedad Mexicana de Computación Científica y sus Aplicaciones

Consejo directivo de la Sociedad Mexicana de Computación Científica y sus Aplicaciones 2020-2022

Presidente
Dr. Justino Alavez Ramírez.
Vicepresidente
Dra. Rina Betzabeth Ojeda Castañeda.
Secretaria de actas y acuerdos
Dra. Ma. Luisa Sandoval Solís.
Tesorero
Dr. Jorge López López.
Secretario General
Dr. Pedro Flores Pérez.
Vocal
Dr. Gerardo Tinoco Guerrero.
Vocal
Dr. Miguel Ángel Uh Zapata.

La Sociedad Mexicana de Computación Científica y sus Aplicaciones fue fundada el 16 de Mayo de 2013, para realizar actividades de investigación científica o tecnológica inscritas en el RENIECyT (Registro Nacional de Instituciones y Empresas Científicas y Tecnológicas), prestadas únicamente a los socios y asociados. Es una Asociación sin fines de lucro. Entre sus tareas fundamentales destacan: Conjuntar acciones e intereses comunes en los investigadores, profesores y estudiantes interesados en la Computación Científica y sus Aplicaciones, con el fin de fomentar la investigación de calidad, promover la actualización y el perfeccionamiento para el desarrollo científico, tecnológico y social; promover la creación, organización, acumulación y difusión de conocimientos referidos a la Computación Científica y sus Aplicaciones; promover la formación e interacción de redes y grupos de trabajo orientados hacia el desarrollo disciplinar, interdisciplinar y temático de la investigación; fomentar el desarrollo de la investigación sobre la Computación Científica y sus Aplicaciones en la República Mexicana; contribuir al mejoramiento de la enseñanza de la Computación Científica y sus Aplicaciones en la República Mexicana; promover y organizar toda clase de encuentros y eventos académicos orientados a la comunicación y discusión entre investigadores y profesores, así como también a la difusión del conocimiento hacia sectores interesados en la integración de la Computación Científica y sus Aplicaciones en los problemas de su sector.

smcca@smcca.org.mx
http://www.smcca.org.mx
https://www.scipedia.com/sj/smcca