Sociedad Mexicana de Computación Científica
y sus Aplicaciones

Boletín

Sociedad Mexicana de Computación Científica y sus Aplicaciones

Año XI - Número 11
Diciembre 2025

El Boletín 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 Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C. (SMCCA), Año XI, No. 11, diciembre 2025, 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) 133 5647, `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 2025.

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

Índice general

Carta de Bienvenida

La Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C. (SMCCA) y el Comité Editorial les dan la más cordial bienvenida a la edición 2025 del Boletín electrónico anual de la SMCCA. Esta publicación tiene como objetivo mantener informada a nuestra comunidad sobre las actividades de la Sociedad y de sus asociados, así como difundir trabajos y reflexiones relevantes en el ámbito de las Matemáticas Aplicadas y el Cómputo Científico.

La presente edición se publica en un año significativo para nuestra Sociedad. A lo largo de sus páginas se refleja no sólo el dinamismo académico de la SMCCA, sino también el reconocimiento y la gratitud hacia quienes han contribuido de manera decisiva a su construcción y consolidación. En este contexto, el Boletín incluye una nota póstuma en memoria del Dr. Humberto Madrid de la Vega, socio fundador, maestro y referente para generaciones de estudiantes y colegas, cuya visión y compromiso siguen siendo parte fundamental del espíritu de nuestra comunidad.

Asimismo, el año 2025 estuvo marcado por acontecimientos de especial relevancia para la vida institucional de la SMCCA. En esta edición del Boletín se da cuenta de la conmemoración académica en honor al Prof. Jesús López Estrada, figura fundamental en el desarrollo de la computación científica y el análisis numérico en México. De igual forma, este año la SMCCA fue aceptada como *Full Member* del International Council for Industrial and Applied Mathematics (ICIAM), convirtiéndose en la única sociedad mexicana con esta distinción, lo que representa un reconocimiento internacional al trabajo sostenido de nuestra comunidad y fortalece la proyección global de nuestras actividades académicas.

El Boletín 2025 incluye noticias, eventos, artículos de divulgación, docencia e investigación de alto nivel, así como contribuciones que reflejan la diversidad de enfoques y aplicaciones de nuestra área. Como novedad, se incorpora la sección Panorama, concebida como un espacio para ofrecer una visión general y reflexiva sobre temas actuales, tendencias y retos en las Matemáticas Aplicadas y el Cómputo Científico, con el propósito de contextualizar los trabajos presentados y enriquecer la lectura del Boletín.

En esta edición se presentan, entre otros contenidos, una semblanza de la edición más reciente de la Escuela Nacional de Optimización y Análisis Numérico, artículos de investigación seleccionados por invitación y por convocatoria, así como los trabajos distinguidos en la edición correspondiente del Premio Mixbaal, que reconoce las mejores tesis de licenciatura en Matemáticas Aplicadas y áreas afines. Estos aportes reflejan el compromiso permanente de la SMCCA con la formación académica, la excelencia científica y el impulso a las nuevas generaciones.

La SMCCA agradece el interés de sus lectores y los invita a continuar participando activamente en las actividades de la Sociedad, ya sea como lectores habituales, autores o miembros activos. La información sobre el registro de membresías puede consultarse en el Módulo de Registro disponible en la página `www.smcca.org.mx`.

Jonathan Montalvo Urquizo
Presidente
Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C.

Panorama SMCCA

In Memoriam: Prof. Humberto Madrid de la Vega (1946–2025)

Con profunda tristeza y, al mismo tiempo, con gratitud, la Sociedad Mexicana de Computación Científica y sus Aplicaciones recuerda la vida y obra del Dr. Humberto Madrid de la Vega, colega entrañable, profesor y uno de los socios fundadores de nuestra asociación. Su partida representa una pérdida irreparable para la comunidad matemática y de cómputo científico en México, pero también una oportunidad para reconocer y celebrar un legado académico y humano excepcional.

Formado en la Facultad de Ciencias de la Universidad Nacional Autónoma de México, el Dr. Madrid realizó estudios de doctorado en matemáticas en la Universidad de Nuevo México y se especializó en temas de análisis numérico. A su regreso al país, desempeñó un papel fundamental en la construcción y el fortalecimiento de instituciones que hoy son referentes nacionales, en las que siempre impulsó la formación de recursos humanos en matemáticas aplicadas y cómputo científico. Fue fundador de la hoy Facultad de Ciencias Físico Matemáticas y también del Centro de Investigación en Matemáticas Aplicadas (CIMA) de la Universidad Autónoma de Coahuila, fundador de la organización de la Escuela Nacional de Optimización y Análisis Numérico (ENOAN) y de nuestra actual Sociedad Mexicana de Computación Científica y sus Aplicaciones (SMCCA).

Convencido de la importancia de la colaboración académica y del desarrollo de las matemáticas aplicadas fuera de los grandes centros tradicionales, impulsó activamente las Escuelas de Matemáticas fuera de la capital mexicana en el marco de los Congresos Nacionales de la Sociedad Matemática Mexicana. Ese esfuerzo sostenido y su entusiasmo incansable derivaron, años más tarde, en la creación de una Red de Escuelas que fue uno de los orígenes de la hoy llamada Red Mexicana de Instituciones de Matemáticas (ReMIM), formalizada en 2021, la cual continúa promoviendo la comunicación y la cooperación interinstitucional en todo el país, donde el Prof. Madrid de la Vega participó activamente hasta este año.

Más allá de sus aportaciones científicas e institucionales, Humberto Madrid fue, para muchos de nosotros, un maestro cercano y generoso. Numerosos socios de la SMCCA tuvimos el privilegio de ser sus estudiantes, de aprender de su rigor académico, de su claridad conceptual y de su profunda vocación por la enseñanza. Su influencia se refleja no solo en publicaciones y proyectos, sino también en generaciones de profesionistas formados bajo su guía y ejemplo.

Toda la comunidad matemática del país, de la cual la SMCCA forma parte, lamenta profundamente su partida. Agradecemos el impacto indeleble de su trabajo y celebramos el ejemplo de dedicación que nos deja. Extendemos nuestras sinceras condolencias a su familia, amistades, colegas y estudiantes, y honramos su memoria, reafirmando nuestro compromiso con los altos valores académicos y humanos que él ayudó a construir.

¡Descansa en paz, querido Humberto!

La ENOAN 2025: contexto y alcance de nuestro evento emblemático

La Escuela Nacional de Optimización y Análisis Numérico (ENOAN) es uno de los eventos académicos más representativos de la comunidad de Matemáticas Aplicadas y Cómputo Científico en México. Desde su creación a principios de la década de 1990, la ENOAN ha mantenido como objetivo central la formación avanzada de estudiantes, la actualización de investigadores y la promoción del intercambio académico en áreas como la optimización, el análisis numérico, la modelación matemática y disciplinas afines. A lo largo de más de tres décadas, este encuentro se ha consolidado como un espacio de referencia nacional, estrecha-

mente vinculado al desarrollo y al fortalecimiento de la Sociedad Mexicana de Computación Científica y sus Aplicaciones (SMCCA).

La edición **ENOAN 2025** se llevó a cabo en la ciudad de **Guanajuato, Guanajuato**, del **23 al 27 de junio de 2025**, en modalidad híbrida, bajo la organización conjunta de la SMCCA y el **Centro de Investigación en Matemáticas (CIMAT)**. El programa académico de esta **edición XXXIII** incluyó cursos especializados, conferencias plenarias, sesiones de trabajos contribuidos y actividades orientadas a la interacción entre estudiantes, académicos e investigadores provenientes de diversas instituciones nacionales e internacionales. Esta estructura permitió atender tanto a participantes en etapas tempranas de su formación como a especialistas consolidados, fomentando el diálogo intergeneracional y la colaboración interdisciplinaria.

La ENOAN 2025 también destacó por su impacto en términos de participación, diversidad institucional y amplitud temática. Con el propósito de documentar y analizar de manera objetiva el alcance de este evento emblemático, en esta sección se presenta una visión general de los principales aspectos académicos y organizativos asociados a su realización. De manera complementaria, el Boletín incluye una sección posterior dedicada exclusivamente a un análisis detallado de los resultados cuantificables de la ENOAN 2025, donde se ofrece un amplio conjunto de estadísticas sobre asistencia, actividades académicas, contribuciones científicas y otros indicadores relevantes que permiten dimensionar con mayor precisión el impacto y la evolución de este encuentro académico.

Más allá de los indicadores cuantitativos, la ENOAN 2025 sirvió nuevamente como un nodo fundamental para fomentar colaboraciones, fortalecer redes académicas, inspirar vocaciones científicas y presentar soluciones, desde la matemática aplicada y el cómputo científico, a problemas de interés nacional. La realización exitosa de esta edición fue posible gracias al trabajo comprometido del Comité Organizador Nacional y Local, así como al apoyo institucional del CIMAT. La SMCCA expresa un agradecimiento profundo a la Dirección General del CIMAT, a cargo del Dr. Rafael Herrera Guzmán, y a los investigadores responsables de la organización local: Dr. Miguel Ángel Moreles Vázquez y Dr. Salvador Botello Rionda, cuya dedicación, liderazgo y compromiso fueron determinantes para el desarrollo académico y logístico del evento.

Reconociendo a los jóvenes: entrega del Premio Mixbaal 2025

Como cada año, la SMCCA realizó la convocatoria para el Premio Mixbaal a la mejor tesis de nivel de licenciatura en matemáticas aplicadas del país. Desde hace más de dos décadas, este premio reconoce la labor científica de quienes son el origen de nuestros objetivos y metas como sociedad científica: los jóvenes estudiantes.

Este año, la convocatoria atrajo a diversos trabajos sometidos desde las instituciones académicas Instituto Tecnológico Autónomo de México, Universidad Autónoma de la Ciudad de México, Universidad Autónoma de las Américas Puebla, Universidad Autónoma de Tlaxcala, Universidad Autónoma de Zacatecas, Universidad Michoacana de San Nicolás de Hidalgo, Universidad Nacional Autónoma de México y Universidad Tecnológica de la Mixteca. Todas las personas que sometieron sus trabajos de titulación de nivel profesional en temas de matemáticas aplicadas fueron evaluadas de manera estricta por investigadores de primer nivel y basado en estas evaluaciones el Comité consideró que este año se entregaran los siguientes dos premios:

- Premio Mixbaal 2025 a la mejor tesis de licenciatura en matemáticas aplicadas para la Lic. Ana Teresa Calderón Juárez con el trabajo *'Coloración en gráficas de mapas en la Tierra y mapas en la luna'* del Instituto Tecnológico Autónomo de México (ITAM).
- Mención honorífica del Premio Mixbaal 2025 para el Lic. Rodrigo Gonzaga Sierra con el trabajo *'Cuantificación de incertidumbre sobre parámetros en modelos no lineales'* de la Universidad Tecnológica de la Mixteca.

Cabe destacar que se reestructuró la organización del premio, iniciando con la formalización de un Comité del Premio Mixbaal, que a partir de este año estará conformado por 3 reconocidos profesores e investigadores miembros de nuestra asociación, que se mantendrán como miembros de este Comité por una duración máxima de tres años. Las funciones de este Comité consisten en llevar el proceso completo de la entrega del Premio Mixbaal, desde la publicación de la convocatoria y la recepción de trabajos, hasta la estructuración del proceso de revisión y la decisión final e inapelable sobre el otorgamiento anual del Premio.

Para esta edición, los miembros del Comité y su permanencia son: Dra. María del Pilar Alonso Reyes (Coordinadora, de 2025 a 2027), Dr. Miguel Ángel Uh Zapata (de 2025 a 2026), Dr. Francisco Domínguez

Mota (2025). A partir de 2026, cada año habrá un nuevo miembro del Comité del Premio Mixbaal y todos los nuevos integrantes serán invitados a participar durante tres años.

Agradecemos por este medio al comité por su arduo trabajo, aunado a un profundo agradecimiento a todos los colegas que anualmente apoyan a la SMCCA con la revisión de los trabajos sometidos, en una valiosa función que nos ayuda a realizar una excelente selección de los mejores trabajos de cada año.

Voto de confianza: la SMCCA obtiene financiamiento estratégico

La SMCCA se complace en anunciar la aprobación del proyecto **'Divulgación de actividades de matemática aplicada, computación científica e ingeniería para el impulso de las vocaciones de investigación en el país'** por parte de la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI). Este respaldo institucional obtenido en la convocatoria ACADEMIAS 2025 se traduce en un apoyo de $1,620,740.00 M.N. para ser ejercido entre el segundo semestre de 2025 y los años 2026 y 2027. Este financiamiento es fundamental para dar continuidad y fortalecer las actividades centrales de nuestra sociedad.

El proyecto, que cuenta con el Dr. Miguel Ángel Uh Zapata, el Dr. Jonathan Montalvo Urquizo y la Dra. Rina Betzabeth Ojeda Castañeda como responsables ante la SECIHTI, permitirá:

- Fortalecer los encuentros académicos insignia: Dar continuidad a la Escuela Nacional de Optimización y Análisis Numérico (ENOAN) en sus ediciones de 2026 y 2027, e innovar con actividades de divulgación científica en forma de conferencias nacionales durante los primeros meses de 2026.
- Ampliar el impacto de la divulgación: Producir y distribuir materiales accesibles, como infografías interactivas y cápsulas audiovisuales, que acerquen las matemáticas aplicadas y el cómputo científico a estudiantes y al público en general.
- Consolidar la publicación científica: Garantizar la edición y publicación de los Boletines de la SMCCA correspondientes a los años 2025, 2026 y 2027, manteniendo este espacio de difusión arbitrada y de acceso abierto.
- Sembrar para el futuro: Crear y mantener un catálogo nacional de temas de tesis y fomentar redes de mentoría, con el objetivo claro de atraer y guiar a la próxima generación de investigadores.
- Promover la inclusión y la equidad: Todas las actividades se diseñarán con un enfoque transversal de género e inclusión, priorizando la participación de mujeres y jóvenes para reducir las brechas históricas en las áreas STEM.

Este éxito se basa en los resultados del proyecto multianual anterior (2021-2024), donde la SMCCA demostró capacidad de ejecución e impacto. Agradecemos a la SECIHTI por confiar en nuestro trabajo y renovamos nuestro compromiso con la comunidad académica nacional para utilizar estos recursos con transparencia y alto impacto, en beneficio de las vocaciones científicas del país.

55 años de legado: homenaje al Prof. Jesús López Estrada

El día 3 de diciembre de 2025, en el Auditorio 'Yelizcalli' de la Facultad de Ciencias de la UNAM, la comunidad académica se congregó para celebrar una trayectoria excepcional. En un acto lleno de reconocimiento y afecto, la Sociedad Mexicana de Computación Científica y sus Aplicaciones (SMCCA) rindió homenaje al Prof. Jesús López Estrada por sus 55 años de ininterrumpida y fructífera labor académica. Esta sesión especial de la SMCCA fue organizada en colaboración con los colegas Pablo Barrera Sánchez, Guilmer González Flores y Humberto Madrid de la Vega†, y reunió a compañeros, colaboradores y discípulos de diversas generaciones e instituciones.

El evento fue inaugurado por el Dr. Luis Felipe Jiménez García, director de la Facultad de Ciencias de la UNAM, y estuvo compuesto por un sólido programa de conferencias que reflejaron la amplitud e impacto del trabajo del Prof. López Estrada en temas de modelación matemática en medicina y de análisis numérico:

- **Dr. Benito Chen Charpentier** (University of Texas at Arlington, USA) inauguró las ponencias con la conferencia *"Modelos matemáticos sencillos de epidemias con tiempo de infección"*, analizando métodos para incorporar tiempos de incubación en modelos epidemiológicos, un área cercana a la línea de investigación del homenajeado.
- **Dr. Justino Alavez Ramírez** (Universidad Juárez Autónoma de Tabasco) presentó *"Estimación de parámetros de modelos basados en EDO de dinámica viral"*, abordando técnicas para resolver problemas inversos, tema central en la investigación del Prof. López Estrada.
- **Dr. Faustino Sánchez Garduño** (Facultad de Ciencias, UNAM) exploró la *"Dinámica de dos sistemas de EDO en tres dimensiones"*, vinculando modelos de invasión cancerosa con la teoría de ondas viajeras. Además, mostró notas históricas de gran interés personal para el Prof. López Estrada, como notas rigurosas tomadas de sus cursos impartidos cuando el expositor era su estudiante hace varias décadas.
- **Dr. Gilberto Calvillo Vives** (IMATE - UNAM) compartió *"Una aventura inconclusa en el ámbito del Álgebra Lineal Numérica"*, rememorando proyectos conjuntos iniciados hace más de 40 años en el IPN sobre códigos de Programación Lineal.
- **Dr. Pablo Barrera Sánchez** (Facultad de Ciencias, UNAM) cerró el ciclo de conferencias con *"Desarrollo de la Matemática Aplicada y Numérica en la Facultad de Ciencias"*, reflexionando sobre los hitos que llevaron a la consolidación de este campo, donde el homenajeado ha sido una figura clave.

También se proyectó un video realizado por colegas de Cuba. En este país, siempre se hace mención de la influencia formativa y humana en la isla por más de cuatro décadas que tuvo el Prof. López Estrada. Investigadores como la Dra. Valia Guerra Ones (Universidad de La Laguna, España), la Dra. Victoria Hernández Mederos (CIMAF, Cuba) e Isidro Abello Ugalde (Universidad de La Habana, Cuba) son algunos de los colegas internacionales que dan fe de este legado de varias décadas de colaboración internacional.

El homenaje fue más allá de lo académico. Durante la sesión se proyectaron y compartieron **remembranzas** escritas por una constelación de colegas y amigos que han acompañado al profesor a lo largo de los años, entre ellos, los profesores: **Benito Chen Charpentier, Zeferino Parada, Jonathan Montalvo Urquizo, Irma García Calvillo, Francisco Domínguez Mota, Lourdes Velasco Arregui y Pedro Miramontes.**

Este acto no solo celebró a un **pionero de la computación científica** y de la modelación matemática en medicina en México, un **formador excepcional** de decenas de profesionales, un indiscutible **organizador de la ENOAN** en muchas de sus ediciones desde hace varias décadas, y un **miembro fundador** de la SMCCA. Celebró, sobre todo, la integridad, la generosidad y la pasión por el conocimiento de un hombre cuya vida y obra han dejado una huella indeleble en la comunidad matemática mexicana y más allá. Su legado, de 55 años de servicio en la UNAM y de producción académica de calidad, es una piedra angular sobre la cual se puede construir el futuro de las matemáticas aplicadas en nuestro país.

¡Muchas felicidades, Jesús! Te deseamos lo mejor en todos los años venideros.

La SMCCA alcanza un escenario global: adhesión a ICIAM

La SMCCA ha dado un paso histórico en su proyección internacional al ser aceptada formalmente como **Sociedad Miembro del *International Council for Industrial and Applied Mathematics (ICIAM)***.

ICIAM (www.iciam.org) es la organización mundial que reúne a 56 de las sociedades más importantes de matemática aplicada e industrial, con presencia en más de 40 países, y nos enorgullece enormemente formar parte de ella. ICIAM constituye la agrupación más grande e importante a nivel global en temas de aplicaciones de las matemáticas y esta es la primera vez que una asociación científica mexicana es adherida como *Full Member* de este importante consorcio internacional, distinción que hasta ahora solo se tenía otorgada a 25 organizaciones en todo el mundo.

Con este reconocimiento como miembros del ICIAM, logramos:

- **Posicionar a la SMCCA** en el mapa global de las matemáticas aplicadas, reconociendo el trabajo y la relevancia del sector académico y de investigación mexicano a nivel internacional.

- **Conectamos a nuestra comunidad** con una red de primer nivel, facilitando el intercambio de ideas, la colaboración en proyectos y la movilidad de investigadores y estudiantes hacia otras sociedades con intereses similares.
- **Ofrecer una voz para México** en los foros de discusión más importantes sobre el futuro de la disciplina y su papel en la solución de problemas globales en los que tenemos incidencia.
- **Brindar acceso privilegiado** a convocatorias y congresos internacionales (como el Congreso ICIAM, el más grande del mundo en el área, que se celebra cada 4 años), así como a información sobre programas de vinculación academia-industria de nivel mundial.

Este logro es el resultado de años de trabajo consistente, de la organización exitosa de eventos como la ENOAN y de la calidad científica de nuestros miembros. Ser parte de ICIAM es un nuevo punto de partida que nos compromete a fortalecer nuestra sociedad, a incrementar nuestra actividad internacional y a representar con orgullo a la matemática aplicada mexicana en el mundo.

Membresía en expansión: únete a la SMCCA

En los últimos meses de 2024, cuando el actual Consejo Directivo de la SMCCA inició su período, la Sociedad contaba con un total de 28 miembros regulares. A la fecha, la membresía ha crecido hasta alcanzar **40 miembros**, lo que representa un **incremento del 39 % en el número de asociados** en un periodo relativamente corto. Este crecimiento refleja el interés sostenido de la comunidad académica en formar parte de la SMCCA, así como el fortalecimiento de sus actividades, eventos y espacios de participación, lo que consolida a la Sociedad como un referente nacional en Matemáticas Aplicadas y Cómputo Científico.

La comunidad de la SMCCA está creciendo y queremos que muchas más personas formen parte de ella. Nuestra sociedad es el punto de encuentro para estudiantes, profesores, investigadores y profesionales interesados en la computación científica y las matemáticas aplicadas.

¿Por qué ser miembro de la SMCCA?

- **Red y colaboración:** Conecta con una red nacional de expertos y entusiastas de tu área.
- **Información y oportunidades:** Accede de primera mano a convocatorias para eventos (como la ENOAN), becas, premios (como el Mixbaal) y oportunidades de colaboración.
- **Divulgación de tu trabajo:** Participa como autor o revisor en el **Boletín SMCCA**, nuestro medio de difusión, arbitrado y registrado en el ISSN.
- **Incidencia y voz:** Contribuye a definir las actividades y el rumbo de la sociedad que reflejan tus intereses académicos.
- **Proyección internacional:** Sé parte de una sociedad que ahora es miembro de **ICIAM**, el consorcio de sociedades de matemáticas aplicadas más grande del mundo, donde podemos colaborar en la construcción científica de nuestros temas de trabajo.

El proceso para unirse es muy sencillo: visita nuestra página web **www.smcca.org.mx** en la sección de 'Membresía' para consultar los tipos de afiliación (estudiante, individual, institucional). Amplía tus horizontes, fortalece tu red profesional y contribuye al desarrollo de nuestra disciplina en México.

¡Tu talento y pasión tienen un lugar aquí con nosotros!

Indicadores de Impacto de la ENOAN 2025

En junio del presente año, realizamos con éxito la XXXIII Escuela Nacional de Optimización y Análisis Numérico (ENOAN 2025) en organización conjunta con la Escuela de Modelación y Métodos Numéricos (EMMN 2025) del CIMAT. Como parte de las acciones de evaluación y documentación del alcance de la ENOAN–EMMN 2025, se implementó un registro en línea mediante una cédula de inscripción con el propósito de recabar información relevante sobre el perfil personal y académico de las personas asistentes, así como sobre su modalidad y forma de participación en el evento. Esto permitió construir un conjunto de indicadores

que reflejan el interés generado por la ENOAN–EMMN 2025 entre estudiantes, profesores e investigadores provenientes de instituciones de educación superior públicas y privadas, institutos tecnológicos, centros de investigación y otras entidades públicas y privadas, tanto de ámbito nacional como internacional. Asimismo, estos datos permiten identificar tendencias relacionadas con los niveles de formación, las áreas temáticas de interés y los mecanismos de participación académica.

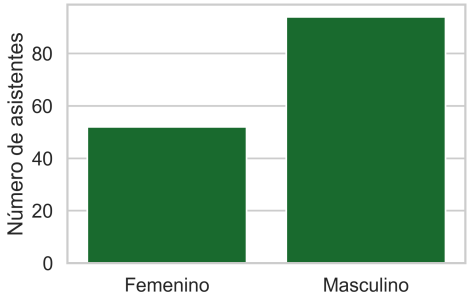
De manera complementaria, el registro proporcionó información sobre quienes presentaron solicitudes para realizar trabajos de investigación o recibir apoyos para su asistencia presencial. El análisis de estos datos constituye una herramienta fundamental para caracterizar la audiencia del evento, evaluar su evolución y orientar la planeación académica y organizativa de futuras ediciones.

Número, género y edad de los asistentes

Como se observa en la Tabla 1, se contó con la asistencia de 146 personas, de las cuales el 36 % (52) correspondió al género femenino y el 64 % (94) al masculino. Los intervalos de edad considerados en el registro abarcaron desde los 18 años hasta más de 64 años, rangos representativos de la audiencia esperada para este tipo de eventos académicos.

Tabla 1: Número y porcentaje del género de los asistentes

Género	Asistentes	Porcentaje (%)
Femenino	52	36
Masculino	94	64
Total general	146	100



Se observa que el mayor número de asistentes se concentró en el rango de 25 a 34 años, seguido por el rango de 18 a 24 años. Los porcentajes más bajos corresponden a los grupos de 55 a 64 años y mayores de 64 años. Un 10.96 % de los asistentes no proporcionó información sobre su edad.

En la Tabla 2 se muestra la distribución cruzada entre género y edad, lo que permite analizar la relación entre ambas variables. Los resultados muestran que en todos los rangos de edad se registró un mayor porcentaje de participación masculina que el femenina.

Modalidad de participación

Dado que la ENOAN–EMMN 2025 se realizó en modalidad híbrida, se registró la preferencia de los asistentes por participar de manera presencial o virtual. Los resultados se presentan en la Tabla 3, donde puede observarse una clara preferencia por la modalidad presencial, que concentró cerca del 77 % de la asistencia total. Además, puede verse que en ambos géneros se observa una preferencia consistente por la modalidad presencial.

Ocupación, nivel de estudios y tipo de participación

La información sobre la ocupación académica de las personas asistentes a la ENOAN–EMMN 2025 da constancia de que el mayor porcentaje de asistentes correspondió a estudiantes, seguidos por profesores–investigadores e investigadores, como puede observarse en la Tabla 4.

En cuanto al nivel de estudios de las personas asistentes, se registró una distribución similar entre los niveles de licenciatura y doctorado, seguidos un poco por debajo por el de maestría. Con esta distribución, puede verse que las personas que acuden a la ENOAN provienen, de manera casi homogénea, de los diferentes niveles universitarios y profesionales.

Los resultados presentados en la Tabla 6 muestran una participación diversa y activa por parte de la comunidad asistente a la ENOAN–EMMN 2025. Destaca que el 33.56 % de las personas registradas participó como asistente en todas las actividades del evento, lo que refleja un alto nivel de interés general. De manera

Tabla 2: Distribución cruzada género–edad de los asistentes

	Femenino		Masculino		Total	
Edad	Número	%	Número	%	Número	%
18 a 24 años	19	45.24	23	54.76	42	28.77
25 a 34 años	17	34.00	33	66.00	50	34.25
35 a 44 años	6	33.33	12	66.67	18	12.33
45 a 54 años	1	12.50	7	87.50	8	5.48
55 a 64 años	2	33.33	4	66.67	6	4.11
Más de 64 años	1	16.67	5	83.33	6	4.11
Sin respuesta	6	37.50	10	62.50	16	10.96
Total general	52	35.62	94	64.38	146	100.00

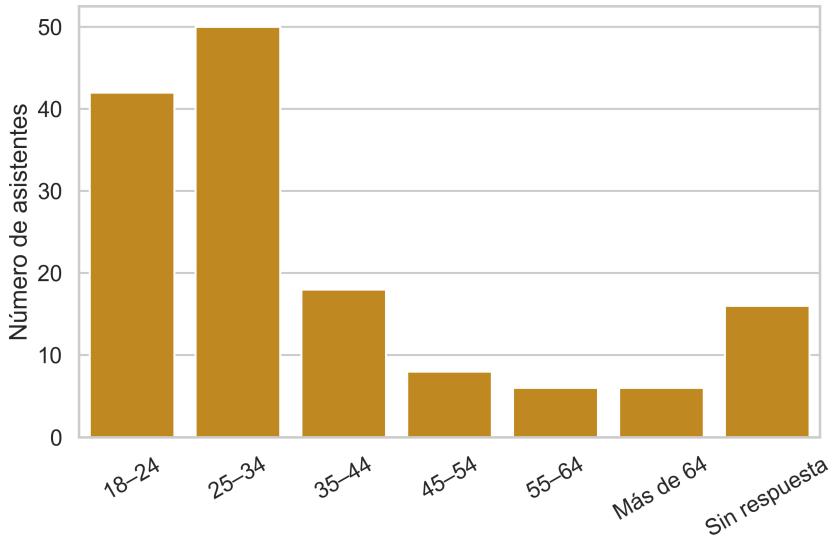


Tabla 3: Distribución cruzada género–modalidad

	Femenino		Masculino		Total	
Modalidad	Número	%	Número	%	Número	%
Presencial	39	26.71	73	50.00	112	76.71
Virtual	13	8.90	21	14.38	34	23.29
Total	52	35.62	94	64.38	146	100.00

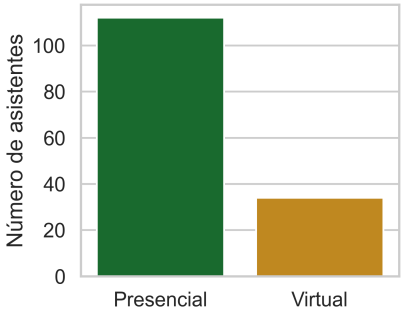


Tabla 4: Ocupación de los asistentes

Ocupación	Asistentes	Porcentaje (%)
Alumno	90	61.64
Analista	1	0.68
Ayudante de profesor	1	0.68
Empleado	4	2.74
Investigador	20	13.70
Pasante	3	2.05
Profesor	9	6.16
Profesor–Investigador	18	12.33
Total general	146	100.00

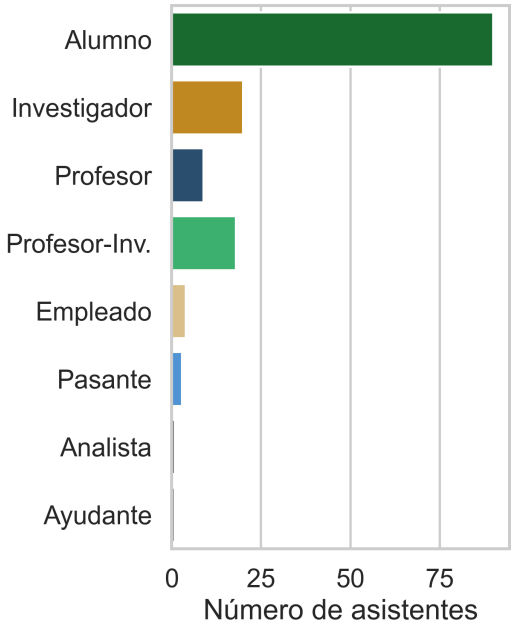
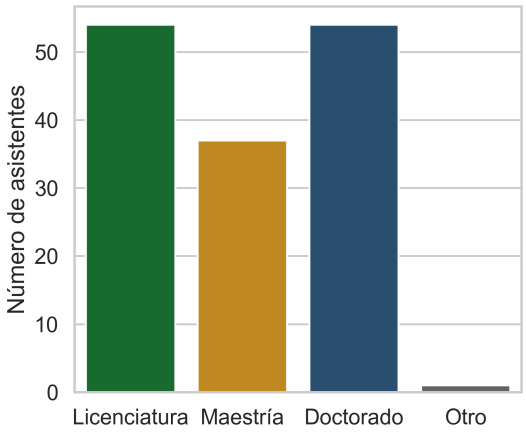


Tabla 5: Nivel de estudios de los asistentes

Nivel de estudios	Asistentes	Porcentaje (%)
Doctorado	54	36.99
Licenciatura	54	36.99
Maestría	37	25.34
Otro	1	0.68
Total general	146	100.00



similar, un 32.19 % correspondió a asistentes que, además de participar en el evento, contribuyeron con la presentación de ponencias por solicitud, lo cual evidencia un involucramiento académico significativo, particularmente entre estudiantes y profesores–investigadores. La participación en modalidades de exposición de carteles fue menor (3.42 %), mientras que el conjunto de conferencistas plenarios e invitados representó cerca del 9 % del total, lo que resalta la presencia de investigadores consolidados en el programa académico. Asimismo, el 5.48 % de los participantes contribuyó como instructor de cursos y el 8.90 % formó parte del equipo organizador, lo que pone de manifiesto la estructura colaborativa necesaria para la realización del evento y el compromiso institucional de la comunidad de la SMCCA.

Estado de nacimiento e identificación étnica

La información relacionada con la entidad federativa o el país de nacimiento de las personas asistentes, así como su identificación, a través de su familia, con algún grupo étnico, con base en los datos proporcionados en la cédula de registro, puede encontrarse en las siguientes tablas. Las Tablas 7 y 8 muestran la información correspondiente a quienes reportaron haber nacido en México y en otro país. Del total de 146 asistentes, 139 (95.02 %) fueron de nacionalidad mexicana y 7 (4.98 %) de nacionalidad extranjera.

Tabla 6: Tipo de participación de los asistentes

Tipo de participación	Asistentes	Porcentaje (%)
Asistente a todo el evento	49	33.56
Asistente únicamente a cursos	10	6.85
Asistente y expositor de cartel	5	3.42
Asistente y expositor de ponencia	47	32.19
Conferencista invitado	6	4.11
Conferencista plenario	7	4.79
Instructor de curso	8	5.48
Organizador del evento	13	8.90
Sin respuesta	1	0.68
Total general	146	100.00

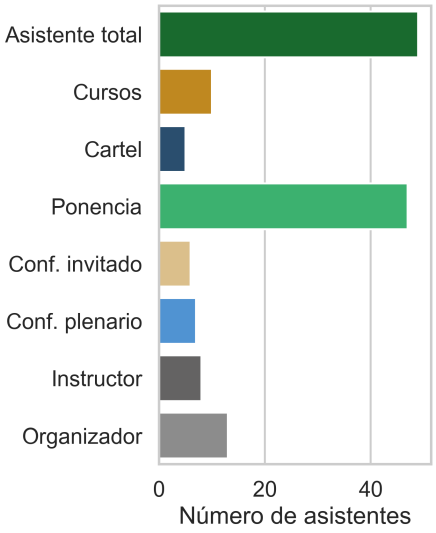


Tabla 7: Estado de nacimiento de los asistentes de nacionalidad mexicana

Estado (México)	Asistentes	Porcentaje (%)
Aguascalientes	1	0.72
Chiapas	3	2.16
Chihuahua	1	0.72
Ciudad de México	40	28.78
Coahuila	5	3.60
Colima	1	0.72
Durango	2	1.44
Estado de México	7	5.04
Guanajuato	3	2.16
Guerrero	3	2.16
Hidalgo	3	2.16
Jalisco	2	1.44
Mérida	4	2.88
Michoacán	15	10.79
Morelos	2	1.44
Nuevo León	6	4.32
Oaxaca	5	3.60
Puebla	7	5.04
Querétaro	1	0.72
San Luis Potosí	2	1.44
Sinaloa	2	1.44
Tabasco	3	2.16
Tlaxcala	2	1.44
Veracruz	3	2.16
Zacatecas	2	1.44
Sin respuesta	14	10.07
Total general	139	100.00

Tabla 8: Estado de nacimiento de los asistentes de nacionalidad extranjera

Estado (otro país)	Asistentes	Porcentaje (%)
Bogotá, Colombia	4	57.14
Putumayo, Colombia	1	14.29
Guayas, Ecuador	2	28.57
Total general	7	100.00

Del total de asistentes de nacionalidad mexicana, la mayor proporción reportó haber nacido en la Ciudad de México (28.78 %), seguida por el estado de Michoacán (10.79 %), así como Puebla y el Estado de México (5.04 % cada uno). En el caso de los asistentes extranjeros, se registró la participación principalmente de Colombia y Ecuador.

La información relacionada con la identificación de los asistentes, o de sus familias, con algún grupo étnico se presenta en la Tabla 9, donde se registra que la mayoría de los asistentes (95.21 %) indicó no identificarse con ningún grupo étnico. Del 3.42 % que respondió afirmativamente, se reportaron identificaciones con los grupos maya, mixteco y afro-mexicano.

Tabla 9: Identificación con algún grupo étnico

Respuesta	Asistentes	Porcentaje (%)
No	139	95.21
Sí	5	3.42
Sin respuesta	2	1.37
Total general	146	100.00

Lugar de residencia e instituciones donde laboran

En las Tablas 10 y 11 se presentan los estados de la República Mexicana y del extranjero en los que actualmente radican los asistentes a la ENOAN. Los resultados indican que hubo asistentes de 21 entidades federativas del país y de dos localidades del extranjero, concentrándose principalmente en la Ciudad de México, Guanajuato y Michoacán.

En la ENOAN–EMMN 2025 se contó con representantes de 37 instituciones educativas y de investigación, de las cuales 34 fueron nacionales y 3 extranjeras. En la Tabla 12 se presenta la distribución de los asistentes por institución nacional. Asimismo, se contó con la participación de asistentes adscritos a instituciones extranjeras, como se muestra en la Tabla 13.

Impacto y perspectivas de la ENOAN

Los resultados presentados en este informe estadístico muestran que la ENOAN–EMMN 2025 se consolidó como un foro académico de alto impacto, capaz de convocar a estudiantes, profesores e investigadores de diversas instituciones nacionales e internacionales. La participación activa en cursos, conferencias plenarias, ponencias y carteles evidencia el interés sostenido de la comunidad por las áreas de optimización, análisis numérico y modelación matemática.

Asimismo, los indicadores analizados reflejan una amplia diversidad en términos de niveles de formación, modalidades de participación y procedencia geográfica, lo que contribuye al fortalecimiento del carácter nacional e internacional del evento. Más allá de los indicadores cuantitativos, la ENOAN–EMMN 2025 propició la interacción académica, el intercambio de ideas y la generación de vínculos que derivaron en proyectos colaborativos con potencial de impacto en la academia, el sector productivo, el ámbito de la salud y la sociedad en general.

La SMCCA seguirá impulsando este evento anual que nos representa como comunidad y constituye la actividad más emblemática para el cumplimiento de nuestros principales objetivos. Entre los retos a futuro

Tabla 10: Estado de la República Mexicana donde radican los asistentes

Estado	Asistentes	Porcentaje (%)
Ciudad de México	37	25.69
Guanajuato	30	20.83
Michoacán	21	14.58
Nuevo León	8	5.56
Estado de México	6	4.17
Morelos	5	3.47
Puebla	5	3.47
Coahuila	4	2.78
Tabasco	4	2.78
Yucatán	4	2.78
Aguascalientes	3	2.08
Chiapas	3	2.08
Chihuahua	2	1.39
Guerrero	2	1.39
Jalisco	2	1.39
Oaxaca	2	1.39
Veracruz	2	1.39
Colima	1	0.69
Durango	1	0.69
Querétaro	1	0.69
San Luis Potosí	1	0.69
Total general	144	100.00

Tabla 11: Lugar de residencia de los asistentes en el extranjero

Lugar	Asistentes	Porcentaje (%)
Antioquia, Colombia	1	50.00
Madison, Wisconsin, EUA	1	50.00
Total general	2	100.00

Tabla 12: Instituciones nacionales de adscripción de los asistentes

Institución	Asistentes	Porcentaje (%)
Centro de Investigación en Matemáticas – Guanajuato	26	17.81
Universidad Nacional Autónoma de México	25	17.12
Universidad Michoacana de San Nicolás de Hidalgo	20	13.70
Universidad Autónoma de la Ciudad de México	8	5.48
Universidad Autónoma Metropolitana – Iztapalapa	6	4.11
Instituto Tecnológico de Estudios Superiores de Monterrey	5	3.42
Universidad Juárez Autónoma de Tabasco	4	2.74
Centro de Investigación en Matemáticas – Mérida	4	2.74
Otras instituciones nacionales	43	29.46
Total general	141	96.58

Tabla 13: Instituciones extranjeras de adscripción de los asistentes

Institución	Asistentes	Porcentaje (%)
Universidad de Pamplona	1	0.68
Universidad Nacional de Colombia	1	0.68
University of Wisconsin–Madison	1	0.68
Total general	3	2.05

del evento se encuentra su consolidación mediante la atracción de nuevos públicos y la inclusión de temas novedosos de interés científico y técnico.

Artículos

Cuantificación de incertidumbre sobre parámetros en modelos no lineales

Rodrigo Gonzaga Sierra, José del Carmen Jiménez Hernández y José Andrés Christen Gracia

Centro de Investigación en Matemáticas

Resumen

En este trabajo se estudia la cuantificación de incertidumbre en parámetros de modelos no lineales mediante el enfoque bayesiano. Se parte del planteamiento clásico de problemas inversos, en los cuales los parámetros del modelo deben inferirse a partir de observaciones ruidosas y de un modelo directo formulado como un sistema de ecuaciones diferenciales. Dado que estos problemas suelen estar mal planteados, se introduce la inferencia bayesiana como estrategia de regularización, permitiendo incorporar información *a priori* y actualizarla con datos mediante la distribución *a posteriori*. Se presentan los fundamentos teóricos del enfoque bayesiano, así como su aplicación al caso particular del modelo de crecimiento logístico, destacando el uso de métodos computacionales para aproximar las distribuciones resultantes de los parámetros del modelo.

Palabras clave: Inferencia bayesiana; Problemas inversos; Cuantificación de incertidumbre; ecuaciones diferenciales; modelo logístico.

1 Introducción

De acuerdo con [3], tradicionalmente se han modelado los problemas de ciencias, ingeniería, medioambiente y otras aplicaciones mediante modelos matemáticos deterministas que describen leyes naturales subyacentes. En la actualidad, se tiende cada vez más a incorporar algún tipo de incertidumbre para representar la falta de conocimiento sobre parámetros y datos físicos relevantes, variaciones aleatorias en las condiciones de operación o pura ignorancia sobre cómo debiera ser el modelo en realidad.

Suponga que se cuenta con observaciones $\mathbf{Y} = (y_1, \ldots, y_n)$, tomadas en los tiempos $\mathbf{t} = (t_1, \ldots, t_n)$, de un fenómeno representado por medio de un sistema de ecuaciones con la siguiente estructura:

$$y_i = \mathcal{H}(X_\theta(t_i)) + \varepsilon_i, \quad i = \overline{1, n}, \tag{1}$$

donde $\mathcal{H}$ es el funcional de observaciones, es común por ejemplo tener $\varepsilon_i \sim_{i.i.d.} \mathcal{N}(0, \sigma^2)$ y X_θ es la solución del siguiente sistema de ecuaciones diferenciales ordinarias, es decir; el regresor o el **modelo directo (forward)**:

$$\frac{dX_\theta}{dt} = F(X_\theta, t, \theta); \quad X_\theta(t_0) = X_0. \tag{2}$$

Conociendo el verdadero valor de θ y las condiciones iniciales, resolver y conocer a X_θ se le llama un **problema directo**.
Sin embargo, en este caso el objetivo es hacer inferencia sobre θ a partir de las observaciones Y, por eso se habla de un **problema inverso** [14].
Se puede considerar este problema como un mapeo:

$$\mathcal{F}_{\mathbf{t}}(\theta) = (\mathcal{H}(X_\theta(t_1)), \ldots, \mathcal{H}(X_\theta(t_n))),$$

este es el "mapeo del modelo directo".

El mapeo inverso está, en general; mal planteado y no tiene mucho sentido:

$$\mathcal{F}_{\mathbf{t}}^{-1}(y_1, \ldots, y_n) = \theta,$$

por eso es necesario realizar una estrategia de regularización, como la cuantificación de la incertidumbre mediante inferencia bayesiana.

Para resolver el problema de inferencia dado en la ecuación (1) se propone que las entradas de θ sean variables aleatorias que siguen alguna distribución de probabilidad. Esto no necesariamente tiene un significado físico o intrínseco, sólo se sabe que se tiene incertidumbre acerca de los valores que pueden tomar estas variables aleatorias, y que la distribución de probabilidad cuantifica su incertidumbre. En este sentido, en el presente trabajo se aborda el problema de cuantificar la incertidumbre desde la perspectiva Bayesiana, en donde se obtiene la distribución *a posteriori* de los parámetros de interés.

2 Teoría Bayesiana

El trabajo de Thomas Bayes, publicado de manera póstuma en 1763, ha tenido una importante consecuencia en la forma de hacer inferencia estadística, este provee una manera formal de combinar el conocimiento *a priori* (o inicial) que se tiene sobre un fenómeno, con el nuevo conocimiento que se adquiere a partir de nuevos datos y mediciones sobre el mismo, obteniendo así un conocimiento *a posteriori* (o final). Es decir, el conocimiento *a priori* se actualiza con la nueva información, y dicho conocimiento *a posteriori* se convertirá en el nuevo conocimiento *a priori*, a la espera, otra vez, de nueva información que lo actualice. En este sección se utilizaron las siguientes bibliografías: [11], [1], [13], [6] y [8].

2.1 Distribución *a priori* y *a posteriori*

En estadística bayesiana, el término común para referirse a la información con la que cuenta el investigador es el de **información subjetiva**, y es importante aclarar, al menos brevemente, qué se entiende en este contexto por el adjetivo "subjetiva", ya que puede tener una connotación distinta a la que se requiere bajo el enfoque bayesiano.
Al hablar de información subjetiva se refiere a toda aquella información *a priori* que se tiene en relación al fenómeno aleatorio de interés, antes de recolectar o realizar nuevas mediciones sobre el mismo, y esto incluye: datos históricos, teorías, opiniones y conjeturas de expertos, conclusiones basadas en estudios previos.

El primer paso en la inferencia estadística bayesiana es traducir todo lo anterior en una distribución de probabilidad **a priori (o inicial)**. El segundo paso consiste en recolectar o realizar nuevas mediciones, y actualizar la distribución de probabilidad a priori, para obtener, mediante el teorema de Bayes, una distribución de probabilidad **a posteriori (o final)**.

Será esta última la mejor descripción posible de la incertidumbre, de acuerdo a toda la información disponible, y por tanto, será la herramienta fundamental a partir de la cual se realiza inferencia estadística.

Para referirse a un **modelo probabilístico paramétrico general** se denota como $p_{X|\Theta}(x|\theta)$, donde la función $p_{X|\Theta}(\cdot|\theta)$ puede ser una función de masa de probabilidades de una variable (o vector) aleatoria discreta o bien una función de densidad de una variable aleatoria continua. El escribir dicha función condicional en el parámetro (o vector de parámetros) θ se debe al hecho de que, una vez dado un valor específico de θ, la función de probabilidad queda totalmente determinada.
Para referirse a **una muestra aleatoria (m.a.)** se utilizará la notación $\mathbf{X} = (X_1, \ldots, X_n)$ y para referirse a una **observación muestral** se utilizará $\mathbf{x} = (x_1, \ldots, x_n)$. Por **espacio paramétrico** se entenderá como el conjunto Θ de todos los valores que puede tomar θ, y por familia paramétrica se entenderá como un conjunto $P = \{p_{X|\Theta}(x|\theta) : \theta \in \Theta\}$.

Siguiendo a [8], la estadística bayesiana modela la incertidumbre que se tiene sobre θ probabilísticamente. Esto es, considere el valor de θ como una variable (o vector) aleatoria con **una distribución de probabilidad *a priori* (o inicial)** $p(\theta)$, de la misma forma se denotará solo como $p(\cdot)$, sin importar si θ es una variable

aleatoria discreta o continua. Se trata de una distribución basada en experiencia previa (experiencia de especialistas, datos históricos, etc.) antes de obtener datos. Luego se procede a observar los nuevos datos (obtención de la muestra) $\mathbf{x} = (x_1, \ldots, x_n)$ y combina esta información con la distribución *a priori* mediante el teorema de Bayes y se obtiene una **distribución de probabilidad *a posteriori* (o final)**:

$$p_{\Theta|\mathbf{X}}(\theta|\mathbf{x}) = \frac{p_{\mathbf{X},\Theta}(\mathbf{x},\theta)}{p_{\mathbf{X}}(\mathbf{x})} = \frac{p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta) \cdot p_{\Theta}(\theta)}{\int p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta) \cdot p_{\Theta}(\theta)\, d\theta}. \tag{3}$$

Note que $p_{\Theta|\mathbf{X}}(\theta|\mathbf{x})$ es también una distribución de probabilidad de θ, pero a diferencia de la distribución a priori $p_{\Theta}(\theta)$ toma en cuenta tanto la información contemplada en $p_{\Theta}(\theta)$ como la información contenida en los datos observados $\mathbf{x} = (x_1, \ldots, x_n)$. La distribución *a posteriori* de la variable aleatoria Θ es la base para hacer inferencias sobre θ.

Tenga presente que, por un lado, la función de verosimilitud $p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)$ y $p_{\Theta}(\theta)$ son distribuciones de probabilidad, y por otro:

$$p_{\mathbf{X}}(\mathbf{x}) = \int p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta) \cdot p_{\Theta}(\theta)\, d\theta,$$

es la probabilidad (o densidad) conjunta de la muestra $\mathbf{x} = (x_1, \ldots, x_n)$ observada a partir del vector aleatorio $\mathbf{X} = (X_1, \ldots, X_n)$. Pero hay que estar consciente de que $p(\mathbf{x})$ es constante respecto a θ, por lo que se puede escribir:

$$p_{\Theta|\mathbf{X}}(\theta|\mathbf{x}) \propto p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta) \cdot p_{\Theta}(\theta), \tag{4}$$

note que $p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta) = p_{X|\Theta}(x_1, x_2, \ldots, x_n|\theta)$, es la probabilidad conjunta de la muestra condicional en θ, llamada función de *verosimilitud*, denotada también por $L(\theta|\mathbf{x})$. En el caso particular de que los componentes del vector aleatorio $\mathbf{X} = (X_1, \ldots, X_n)$ resulten ser independientes, se tiene que:

$$p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta) = \prod_{j=1}^{n} p_{X|\Theta}(x_j|\theta).$$

Puede proponer como estimador puntual de θ alguna medida de tendencia central, por ejemplo la mediana o la esperanza:

$$\hat{\theta} := E(\theta) = \int \theta\, p_{\Theta|X}(\theta|x)\, d\theta. \tag{5}$$

Y aún en el caso de que no se cuente con información muestral se puede calcular $\hat{\theta}$ utilizando $p_{\Theta}(\theta)$ en lugar de $p_{\Theta|\mathbf{X}}(\theta|\mathbf{x})$.

En algunos casos en los que, en vez de conocer el vector de parámetros θ, lo que interesa es describir el comportamiento de observaciones futuras del fenómeno aleatorio en cuestión, esto es, hacer **predicción**.
Dado un valor de θ, la distribución que describe el comportamiento de la observación futura X_n es $p_{X_n|\Theta}(x|\theta)$. El problema es que por lo general el valor de θ es desconocido. Por lo regular, la estadística frecuentista aborda este problema estimando puntualmente a θ con base en la muestra observada, y dicho estimador $\hat{\theta}$ es sustituido en $p_{X_n|\Theta}(x|\hat{\theta})$. Desde la perspectiva bayesiana, el modelo $p_{X_n|\Theta}(x|\theta)$ junto con la distribución a priori $p_{\Theta}(\theta)$ inducen una distribución conjunta para el vector aleatorio (X_n, Θ) mediante el concepto de probabilidad condicional:

$$p_{X_n,\Theta}(x,\theta) = p_{X_n|\Theta}(x|\theta) p_{\Theta}(\theta).$$

Si se marginaliza la distribución de probabilidad conjunta anterior se obtiene:

$$p_{X_n}(x) = \int_{\Theta} p_{X_n,\Theta}(x,\theta)\, d\theta.$$

De los dos resultados anteriores, se tiene:

$$p_{X_n}(x) = \int_{\Theta} p_{X_n|\Theta}(x|\theta) p_{\Theta}(\theta)\, d\theta. \tag{6}$$

A $p_{X_n}(x)$ se le denomina **distribución predictiva a priori (o inicial)** y describe el conocimiento acerca de una observación futura X_n basado únicamente en la información contenida en $p_\Theta(\theta)$. Nótese que $p_{X_n}(x)$ no depende ya de θ.

Para hacer estimación por regiones, por ejemplo, si desea calcular la probabilidad de que el vector de parámetros θ pertenezca a una región $A \subset \Theta$:

$$P(\theta \in A) = \int_A p_{\Theta|\mathbf{X}}(\theta|\mathbf{x})d\theta,$$

o bien, dado un valor $\delta \in (0,1)$, se busca un $A \subset \Theta$ tal que $P(\theta \in A) = \delta$. Con frecuencia la solución para A no es única. Cabe aclarar que si $\dim(\Theta) = 1$ las regiones son subconjuntos de $\mathbb{R}$ y que un caso particular de estas regiones son los intervalos. En este sentido, la estimación por regiones en estadística bayesiana es más general que la estimación por intervalos de la estadística frecuentista.
Una vez obtenida la muestra, el modelo $p_{X|\Theta}(x|\theta)$ y la distribución *a posteriori* $p_{\Theta|\mathbf{X}}(\theta|\mathbf{x})$ inducen una distribución conjunta para $(\mathbf{X}, \Theta)$ condicional en los valores observados $\mathbf{x} = (x_1, \ldots, x_n)$:

$$\begin{aligned} p_{X,\Theta|\mathbf{X}}(x,\theta|\mathbf{x}) &= \frac{p_{X,\Theta,\mathbf{X}}(x,\theta,\mathbf{x})}{p_{\mathbf{X}}(\mathbf{x})} \\ &= \frac{p_{X|\Theta,\mathbf{X}}(x|\theta,\mathbf{x})p_{\Theta,\mathbf{X}}(\theta,\mathbf{x})}{p_{\mathbf{X}}(\mathbf{x})} \\ &= p_{X|\Theta,\mathbf{X}}(x|\theta,\mathbf{x})p_{\Theta|\mathbf{X}}(\theta|\mathbf{x}) \\ &= p_{X,\Theta}(x|\theta)p_{\Theta|\mathbf{X}}(\theta|\mathbf{x}). \end{aligned}$$

En lo anterior, $p_{X|\Theta,\mathbf{X}}(x|\theta,\mathbf{x}) = p_{X,\Theta}(x|\theta)$ se justifica por la independencia condicional de X y $\mathbf{X} = (X_1, ..., X_n)$ dado θ. Si se marginaliza la distribución conjunta condicional anterior:

$$p_{X|\mathbf{X}}(x|\mathbf{x}) = \int p_{X,\Theta|\mathbf{X}}(x,\theta|\mathbf{x})\, d\theta.$$

Combinando los dos resultados anteriores:

$$p_{X,\mathbf{X}}(x|\mathbf{x}) = \int p_{X|\Theta}(x|\theta)p_{\Theta|\mathbf{X}}(\theta|\mathbf{x})\, d\theta. \tag{7}$$

A $p_{X|\mathbf{X}}(x|\mathbf{x})$ se le llama **distribución predictiva a posteriori (o final)**, y describe el conocimiento acerca de una observación futura X basado tanto en la información contenida en $p_\Theta(\theta)$ como en la información muestral $\mathbf{x} = (x_1, ..., x_n)$. Nótese nuevamente que $p_{X|\mathbf{X}}(x|\mathbf{x})$ no depende de θ.

Así que para hacer predicción sobre observaciones futuras del fenómeno aleatorio que esté modelando se usa $p_X(x)$ o bien $p_{X|\mathbf{X}}(x|\mathbf{x})$, según sea el caso. Y de manera análoga a lo mencionado sobre inferencia bayesiana, una manera simple de hacer predicción puntual, por ejemplo, de una observación futura X podría ser mediante alguna medida de tendencia central, como la mediana o la esperanza:

$$\hat{x} = E(X) = \int_{\mathrm{R}(X)} x\, p_{X|\mathbf{X}}(x|\mathbf{x})\, dx,$$

donde $\mathrm{R}(X)$ es el rango de la v.a. X. También, una manera de calcular la probabilidad de que una observación futura se encuentre en un conjunto $A \subseteq \mathrm{R}(X)$ sería:

$$P(\{X \in A\}) = \int_A p_{X|\mathbf{X}}(x|\mathbf{x})dx.$$

Las ecuaciones (3), (6) y (7) constituyen *el modelo general de la estadística bayesiana*. Cualquier problema estadístico tratado bajo el enfoque bayesiano implica la obtención y utilización de las distribuciones correspondientes.

3 Análisis bayesiano del problema inverso

Una vez conocidas algunas bases sobre probabilidad y estadística, estadística bayesiana, se puede realizar inferencia bayesiana para resolver problemas inversos. El caso particular que se abordará, será el del modelo de crecimiento logístico. Para el desarrollo de este capítulo se utilizaron las siguientes bibliografías: [4], [8], [15], [10] y [16].

3.1 Aproximación bayesiana a inferencia

Se considera un fenómeno representado por medio de un sistema de ecuaciones diferenciales ordinarias. Suponga que se cuenta con observaciones $\mathbf{y} = (y_1, \ldots, y_n)$ tomadas en los tiempos $t = (t_1, \ldots, t_n)$, con la siguiente estructura:

$$y_i = \mathcal{H}(X_\theta(t_i)) + \varepsilon_i, \quad i = \overline{1,n} \tag{8}$$

Como se mencionó en la introducción, el objetivo es hacer inferencia sobre θ a partir de las observaciones. Para resolver el problema de inferencia (8) se propone que las entradas de θ sean variables aleatorias que siguen alguna distribución de probabilidad. La incertidumbre se cuantifica con una medida de probabilidad. El agente interesado en conocer el parámetro θ, establece una variable aleatoria Θ con su **densidad de probabilidad**

$$p_\Theta(\cdot),$$

Los valores que Θ toma son los posibles valores para los parámetros, en este caso $\Phi = (\Theta, \Sigma)$ toma valores (θ, σ). Esta medida de probabilidad cuantifica la incertidumbre que tiene el agente respecto a los posibles valores de los parámetros en el modelo, $P_\Phi(\theta, \sigma)$ es la distribución a priori .

En presencia de datos $\mathbf{Y} = \mathbf{y}$, y suponiendo un modelo para la distribución conjunta:

$$p_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta, \sigma),$$

para $\mathbf{Y}$, donde $E(y_i \mid \theta, \sigma) = \mathcal{H}(X_\theta(t_i))$. Al observar los datos Y, interesa inferir el valor de θ. La teoría bayesiana prescribe que calcular la distribución condicional de las incógnitas de interés dados los datos, se calcula utilizando el teorema de Bayes para variables aleatorias:

$$p_{\Phi|\mathbf{Y}}(\theta, \sigma|\mathbf{y}) = \frac{p_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta, \sigma) p_\Phi(\theta, \sigma)}{p_{\mathbf{Y}}(\mathbf{y})},$$

$p_{\Phi|\mathbf{Y}}(\theta, \sigma|\mathbf{y})$ es la distribución *a posteriori*; además $p_\Phi(\theta, \sigma)$ es la distribución *a priori* de (Θ, Σ) y

$$p_{\mathbf{Y}}(\mathbf{y}) = \int p_{\mathbf{Y},\Phi}(\mathbf{y}, \theta, \sigma) d\theta d\sigma,$$

$$p_{\mathbf{Y}}(\mathbf{y}) = \int p_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta, \sigma) p_\Phi(\theta, \sigma) d\theta d\sigma,$$

es la constante de normalización, también llamada **verosimilitud marginal**.

En el caso en que el error de cada observación representa un ruido aditivo gaussiano, la función de verosimilitud es:

$$p_{Y|\Phi}(\mathbf{y}|\theta, \sigma) = \sigma^{-n}(2\pi)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mathcal{H}(X_\theta(t_i)))^2\right).$$

Note que, cada vez que se evalué $p_{Y|\Phi}(\mathbf{y}|\theta, \sigma)$ debe resolverse a X_θ, lo cual se hace de forma aproximada por medio de un método numérico.

Como consecuencia, el sistema de ecuaciones diferenciales ordinarias se resuelve utilizando un método numérico y la inferencia se realiza, no en el modelo exacto anterior, sino en un modelo aproximado, a saber:

$$y_i = \mathcal{H}(X_\theta^h(t_i)) + \varepsilon_i, \quad \varepsilon_i \underset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$$

donde X_θ^h denota la solución aproximada proporcionada por el método numérico. La nueva verosimilitud derivada del modelo es:

$$p^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta, \sigma) = \sigma^{-n}(2\pi)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mathcal{H}(X_\theta^h(t_i)))^2\right).$$

Para calcular $p_{\Phi|\mathbf{Y}}(\theta,\sigma|\mathbf{y})$ comúnmente se usan métodos tipo Monte Carlo vía cadenas de Markov (MCMC). Este cálculo a su vez va a estar afectado por la precisión del método numérico usado para calcular a X_θ. La distribución *a posteriori* numérica es:

$$p^h_{\Phi|Y}(\theta,\sigma|\mathbf{y}) = \frac{p^h_{Y|\Phi}(\mathbf{y}|\theta,\sigma)p_\Phi(\theta,\sigma)}{p^h_{\mathbf{Y}}(\mathbf{y})},$$

donde $p(\theta,\sigma)$ es la distribución a priori en (θ,σ) y

$$p^h_{\mathbf{Y}}(\mathbf{y}) = \int p^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)p_\Phi(\theta,\sigma)d\theta d\sigma,$$

es la constante de normalización, también llamada **verosimilitud marginal**. Note que dado que no hay otra alternativa que utilizar la distribución *a posteriori* numérica, hay una necesidad real de comprender y controlar el error incurrido al trabajar con $p_{\Phi|\mathbf{Y}}(\theta,\sigma|\mathbf{y})$ y la aproximada numéricamente $p^h_{\Phi|\mathbf{Y}}(\theta,\sigma|\mathbf{y})$.

En el anexo A se menciona que un método numérico es de orden p, si $\varepsilon_h(\theta) = \mathcal{O}(h^p)$, es decir, $\varepsilon_h(\theta) \leq Kh^p$, con K una constante global que no depende de h. [2] demuestran que bajo un tamaño de paso h, se garantiza que prácticamente no existe diferencia entre la distribución a posteriori teórica $p_{\Phi|\mathbf{Y}}(\theta,\sigma|\mathbf{y})$ y la aproximada numéricamente $p^h_{\Phi|\mathbf{Y}}(\theta,\sigma|\mathbf{y})$. Con lo cual se le da validez al análisis de inferencia que se realice a partir de la distribución *a posteriori* aproximada numéricamente.

Los métodos estadísticos tradicionales se centran en la estimación puntual. El estimador de máxima *a posteriori* (MAP) se considera como una versión regularizada del estimador de máxima verosimilitud o el estimador de mínimos cuadrados.

$$\begin{aligned}\log(p^h_{\Phi|\mathbf{Y}}(\theta,\sigma|\mathbf{y})) &= C + \log p^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma) + \log p_\Phi(\theta,\sigma)\\ &= C - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathcal{H}(X^h_\theta(t_i)))^2 + \log p_\Phi(\theta,\sigma),\end{aligned}$$

donde C es constante respecto a Φ. Si se supone que hay independencia entre las variables θ y σ, se tiene que:

$$\log(p^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)) = C - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathcal{H}(X^h_\theta(t_i)))^2 + \log p_\Theta(\theta) + \log p_\Sigma(\sigma) \tag{9}$$

Esta última ecuación se llamará **objetivo**, y se utilizará más adelante.

3.2 El modelo logístico

De acuerdo con [9, 15], una **población** es un grupo de organismos vivos (plantas, animales, microorganismos, etc.) que está compuesto por individuos con un comportamiento dinámico similar. Las poblaciones cambian de tamaño (crecen o disminuyen) debido al nacimiento, muerte y migración.
La **dinámica de poblaciones** estudia las leyes que rigen los cambios de la población en el espacio y el tiempo. Se centra en cómo las poblaciones cambian con el tiempo. Además, una población se describe por su número de individuos.

En 1798, Thomas Robert Malthus, propuso su modelo bajo las siguientes hipótesis:

1. La población es homogénea (todos los individuos son idénticos).
2. El medio es homogéneo, es decir, las características físicas, biológicas, etcétera, son las mismas en el hábitat.
3. No hay limitaciones ni de espacio ni de alimento para el crecimiento de la población (la tasa de cambio de la población en el tiempo t es proporcional a la población en ese instante de tiempo).
4. La población está aislada (no hay migración).

5. Las tasas de natalidad y de mortalidad son constantes.

Denote por:

- $X(t)$ el número de individuos en el tiempo t,
- β la tasa de natalidad, y
- μ la tasa de mortalidad,

con β y μ positivos, entonces, por las hipótesis dadas, se tiene el siguiente modelo:

$$\frac{dX(t)}{dt} = \beta X(t) - \mu X(t),$$

o bien,

$$\frac{dX(t)}{dt} = rX(t),$$

donde $r = \beta - \mu > 0$ y se le conoce como tasa de crecimiento instantáneo o tasa de crecimiento per cápita.

Cuarenta años más tarde, en 1838, el matemático belga Pierre François Verhulst (1804-1849) modificó el modelo de Malthus, cambiando la hipótesis 3:

- Los recursos (alimentos o tamaño del medio) son finitos.

Con esta nueva hipótesis habrá competencias entre la misma especie, así Verhulst propuso el modelo de crecimiento logístico:

$$\frac{dX(t)}{dt} = rX(t)\left(1 - \frac{X(t)}{K}\right),$$

o bien,

$$\frac{dX(t)}{dt} = LX(t)\left(K - X(t)\right), \tag{10}$$

donde $L = \frac{r}{K}$ y se llama **capacidad de carga**.

Dada una condición inicial $X(0) = X_0$, la ecuación (10) tiene como solución:

$$X(t) = \frac{KX_0e^{LKt}}{K + X_0\left(e^{LKt} - 1\right)}, \tag{11}$$

o bien,

$$X(t) = \frac{K}{1 + \left(\frac{K}{X_0} - 1\right)e^{-rt}}.$$

A la ecuación (10) se le llama **ecuación logística**. A pesar de que el modelo de crecimiento logístico tiene a esta como solución explícita, se utilizará el programa *odeint*, que se encuentra en la paquetería *scipy.integrate* implementado en Python; pues realizarlo de esta manera podrá ser replicable para cualquier otro modelo no lineal al que se pueda solucionar con un método numérico.

3.3 Simulación estocástica y aplicación

De acuerdo con [4], las evaluaciones de incertidumbre también pueden adoptar la forma de intervalos o regiones de credibilidad, similares a los intervalos de confianza utilizados en la estadística clásica. En general, para realizar inferencias sobre las incógnitas en el modelo y finalmente responder preguntas relevantes de investigación, se debe ser capaces de analizar la distribución *a posteriori*.
Conforme a [7], aunque no es prudente intentar analizar directamente las propiedades de la distribución posterior, existen métodos indirectos que puede proporcionarnos información considerable. En lugar de pensar en la distribución posterior como una función, se puede utilizar el hecho de que es una distribución de probabilidad y, por lo tanto, puede analizarse mediante métodos estadísticos, siempre que exista una manera de obtener una muestra.
Para eso, se utilizará un método de **Monte Carlo vía Cadenas de Markov** (MCMC). Los métodos de MCMC representan un conjunto de algoritmos que permiten obtener muestras aleatorias de una determinada distribución de probabilidad objetivo, de la cual es difícil muestrear directamente. Estos métodos se basan en construir una cadena de Markov cuya distribución de equilibrio, es la distribución objetivo. De esta forma, los estados de la cadena de Markov después de que esta ha alcanzado el estado estacionario representan muestras de la distribución objetivo.
La principal ventaja de los métodos de Monte Carlo es que podemos muestrear de una medida de probabilidad solo conocida hasta una constante de normalización. La principal limitación de este enfoque es que los métodos de Monte Carlo se deterioran con el aumento de la dimensión del parámetro.

El algoritmo base para realizar MCMC, es el **algoritmo Metropolis-Hasting**, este tiene la siguiente estructura:

Algorithm 1 Algoritmo Metropolis -Hasting

Sea f la distribución de interés. Como pre-proceso se genera el valor incial $X_{(0)} \sim \mu_0$ (con μ_0 una distribución de probabilidad con el mismo soporte que f). Para generar los valores $t = 1, 2, \ldots$ de la cadena de Markov, se hace:

1: Siendo $X_t = x_t$ el estado actual de la cadena de Markov se propone como candidato para el siguiente elemento de la cadena a $y_t \sim q(\cdot|x_t)$, con q una distribución de probabilidad instrumental.

2: Se hace $X_{(t+1)} = \begin{cases} y_t & \text{con probabilidad } \rho(x_t, y_t) \\ x_t & \text{con probabilidad } 1 - \rho(x_t, y_t) \end{cases}$

con $\rho(x_t, y_t) = \text{mín}\left\{1, \dfrac{f(y)q(x|y)}{f(x)q(y|x)}\right\}$.

En general, los métodos de MCMC son muy complejos, requiriendo una calibración cuidadosa por parte de un experto, tanto para optimizar la velocidad de convergencia , como para identificar cuanto tiempo debe simularse la cadena antes de extraer la muestra. Una solución a este problema es el uso de una biblioteca de Python escrita por [5], llamada el *t-walk*, disponible en `http://www.cimat.mx/~jac/twalk/`. El *t-walk* utiliza un tipo especial de algoritmo Metropolis-Hastings de propósito general que se ajusta automáticamente para muestrear de cualquier distribución cuando se le proporcionan las funciones de soporte y el logaritmo de la función objetivo, en este caso la ecuación (9). Para iniciar el MCMC se debe añadir la cantidad de muestras a obtener y dos puntos iniciales.

Simulando un conjunto de datos sintético con la ecuación (11), con el modelo de error $y_i = X(t_i) + \varepsilon_i$, donde $\varepsilon_i \sim N(0, \sigma^2)$, y los siguientes parámetros:

$$X(0) = 100, \quad L = 1/1000, \quad K = 1000, \quad \sigma = 30.$$

Se consideran 26 observaciones en los tiempos t_i distribuidos regularmente entre 0 y 10.

Para realizar la cuantificación de incertidumbre para esta simulación, se tiene:

$$y_i = \mathcal{H}(X_\theta(t_i)) + \varepsilon_i, \quad i \in \{1, 2, \ldots, n\},$$

donde $\mathcal{H}(x) = x$ y $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, además el modelo directo es,

$$\frac{dX_\theta}{dt} = F(X_\theta, t, \theta),$$

$$F(X_\theta, t, \theta) = LX(K - X),$$

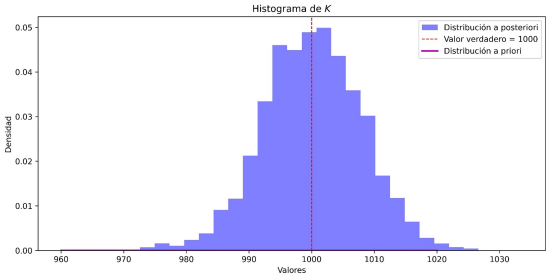
donde $\theta = (L, K)$. Suponiendo independencia *a priori* sobre los parámetros K, L y σ, las distribuciones *a priori* propuestas son:

$$K \sim Gamma(2, 1/0{,}001),$$

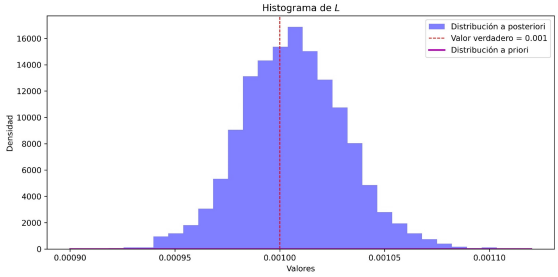
$$L \sim Gamma(2, 1/0{,}001),$$

$$\sigma \sim Gamma(2, 1/0{,}001),$$

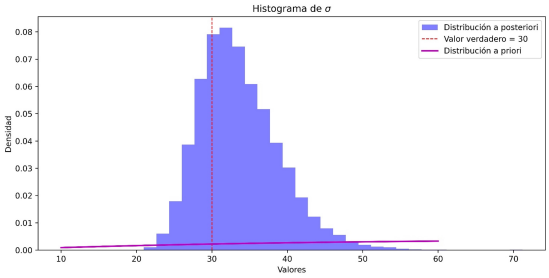
dado que lo único que se "conoce" sobre estos parámetros es que son positivos, por esto se tomarán valores de la distribución gamma, la cual toma solo valores positivos.



(a) Distribución *a posteriori* del parámetro K.



(b) Distribución *a posteriori* del parámetro L.



(c) Distribución *a posteriori* del parámetro σ.

Figura 1: Distribuciones *a posteriori* de los parámetros del modelo.

En las Figuras 1a, 1b y 1c, se tienen los histogramas de 100 000 valores de la distribución *a posteriori* de K, L y σ. Se puede observar que el valor verdadero, con el que se realizó la simulación está dentro de la distribución *a posteriori*, además de estar cercano al punto con mayor credibilidad. Además, note que la distribución *a priori* queda muy por debajo de la distribución *a posteriori*.

En la Figura 2 se muestran los datos simulados, la solución del modelo logístico con los parámetros reales, la curva que mejor se ajusta la cual se forma utilizando los valores de mayor credibilidad en cada distribución a posteriori. Note que la curva con los parámetros reales y el mejor ajuste está una encima de la otra, salvo un pequeño error que es provocado por el error del método numérico. Por último, la parte sombreada son curvas solución a partir de valores de las distribución *a posteriori* para cada parámetro.

3.4 Aplicación: Saccharomyces cerevisiae

Las levaduras son hongos que forman sobre los medios de cultivo colonias pastosas, constituidas en su mayor parte por células aisladas que suelen ser esféricas, ovoideas, elipsoideas o alargadas. Los genetistas [12] realizaron un estudio para comparar la capacidad de sobrevivencia de cepas haploides, diploides y tetraploides

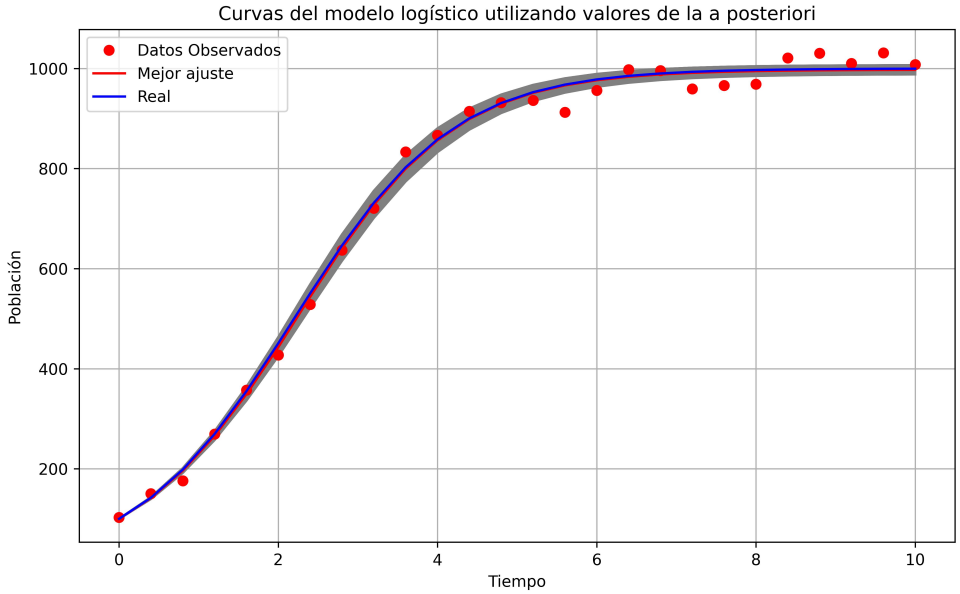
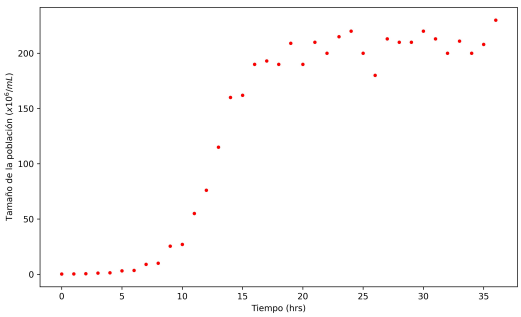


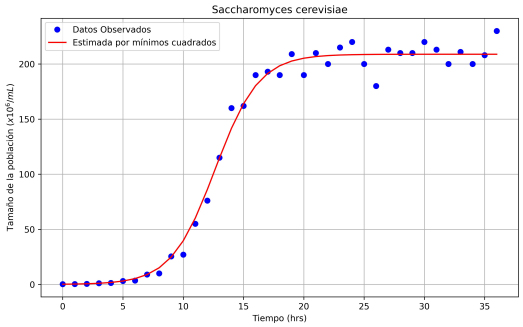
Figura 2: Incertidumbre del ejemplo de simulación.

de la levadura *Saccharomyces cerevisiae* al ser expuestas a mutaciones inducidas por metanosulfonato de etilo, evaluando las ventajas del enmascaramiento en niveles altos de ploidía, así como la eliminación de mutaciones en células haploides.
[15] recopilaron los datos del crecimiento de esta población de levaduras en diferentes horas, los datos se muestran en la Tabla 1.

En la figura 3a se puede observar el crecimiento de la levadura durante las primeras 36 horas, donde es claro observar que siguen el comportamiento de un modelo de crecimiento logístico donde su capacidad de carga llega aproximadamente a 250.



(a) Crecimiento de la levadura Saccharomyces cerevisiae.



(b) Crecimiento de Saccharomyces cerevisiae, con el ajuste por mínimos cuadrados.

Figura 3: Análisis del crecimiento de Saccharomyces cerevisiae.

[15] sugieren el modelo logístico y muestran que las estimaciones de los parámetros son:

$$K = 211{,}538461, \quad L = 0{,}0026,$$

como los mejores para el ajuste de los datos al modelo. Cabe mencionar que no especifican el método utilizado

Tiempo (hrs)	**Tamaño ($\times 10^6$/mL)**	**Tiempo (hrs)**	**Tamaño ($\times 10^6$/mL)**
0	0.200	19	209
1	0.330	20	190
2	0.500	21	210
3	1.10	22	200
4	1.40	23	215
5	3.10	24	220
6	3.50	25	200
7	9.00	26	180
8	10.0	27	213
9	25.4	28	210
10	27.0	29	210
11	55.0	30	220
12	76.0	31	213
13	115	32	200
14	160	33	211
15	162	34	200
16	190	35	208
17	193	36	230
18	190		

Tabla 1: Crecimiento de la levadura Saccharomyces cerevisiae.

para dichas estimaciones.
[17] presentó las siguientes estimaciones de los mismos parámetros calculados por mínimos cuadrados:

$$K = 208{,}855224, \quad L = 0{,}00263224.$$

La figura 3b muestra la gráfica del crecimiento y la curva con los parámetros estimados en [17].

Usando estos datos, se realiza el procedimiento de cuantificación de la incertidumbre suponiendo el modelo de crecimiento logístico como el regresor. Este proceso se realiza de manera similar al ejemplo de simulación, es decir, se toman las mismas distribuciones *a priori.*

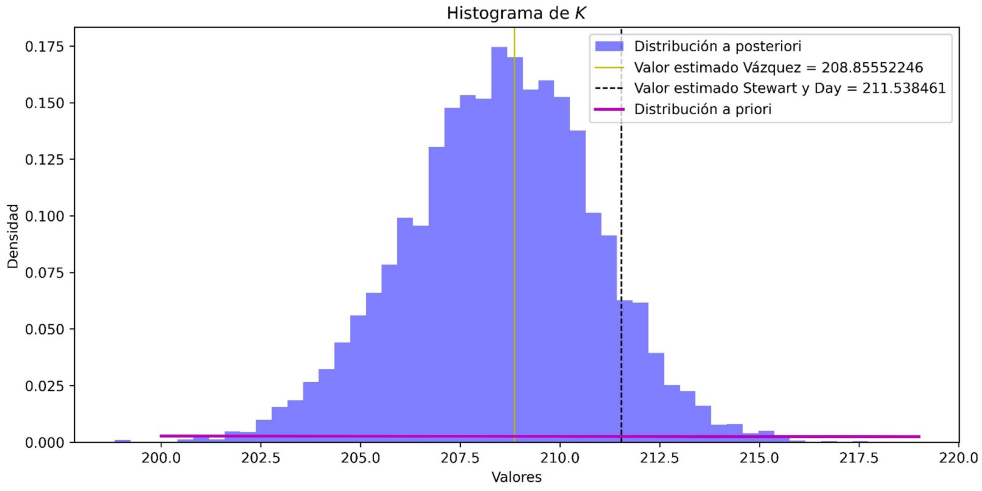


Figura 4: Distribución *a posteriori* del parámetro K, *Saccharomyces cerevisiae.*

En las Figuras 4, 5 y 6, se tienen los histogramas de 100 000 valores de la distribución *a posteriori* de K, L y σ. Se puede observar que los valores estimados por [15], [17] están dentro de la distribución *a posteriori*,

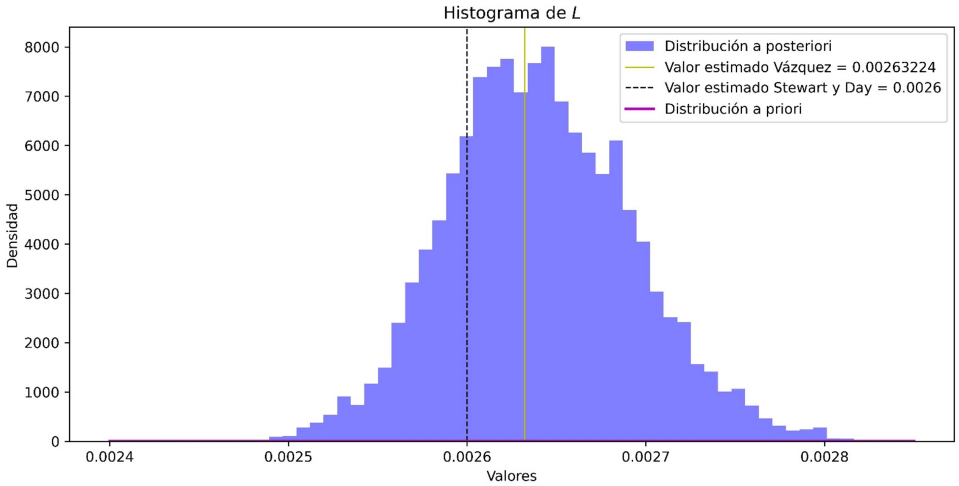


Figura 5: Distribución *a posteriori* del parámetro L, *Saccharomyces cerevisiae*.

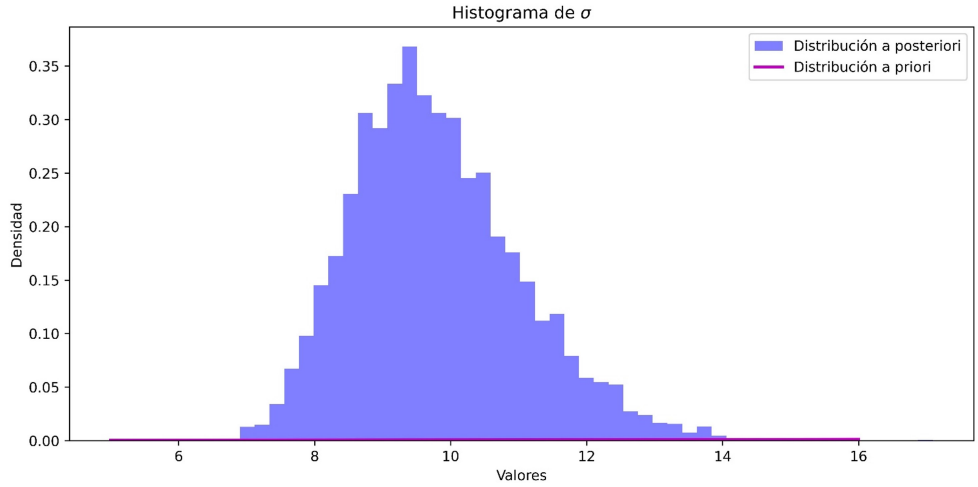


Figura 6: Distribución *a posteriori* del parámetro σ, *Saccharomyces cerevisiae*.

además de estar cercanos al punto con mayor credibilidad, puede parecer que los valores estimados son los mismos pero se debe a la escala, estos si difieren un poco. Además, note que la distribución *a priori* queda muy por debajo de la distribución *a posteriori*. Por último, se puede ver la distribución *a priori*, donde hay un gran cambio entre la distribución *a priori* y la distribución *a posteriori*.

En la Figura 7 se muestran los datos recopilados de la Tabla 1, la solución del modelo logístico con los parámetros de [17], [15] y la curva que mejor se ajusta, esta última se realizó utilizando los valores de mayor credibilidad en cada distribución *a posteriori*. Note que la curva con los parámetros dados por Vázquez se aleja un poco de las otras dos curvas, además la curva de mejor ajuste con la de Stewart y Day coinciden, por ese motivo se colocó de manera punteada, para distinguir entre cada una.
La Figura 7 es un ejemplo de una aplicación de la distribución *a posteriori*. Además, como se tiene varios valores de esta distribución se pueden encontrar valores de interés como las medidas de localización, medidas de variabilidad, coeficiente de asimetría y coeficiente de curtosis o algún otro cálculo dependiendo del interés del investigador.

El área sombreada se formó tomando curvas solución a partir de valores de las distribución *a posteriori* para cada parámetro. El uso de un área sombreada alrededor de los parámetros que mejor se ajustan en las gráficas de modelos es una técnica común en el análisis de datos. Esta área sombreada suele representar la

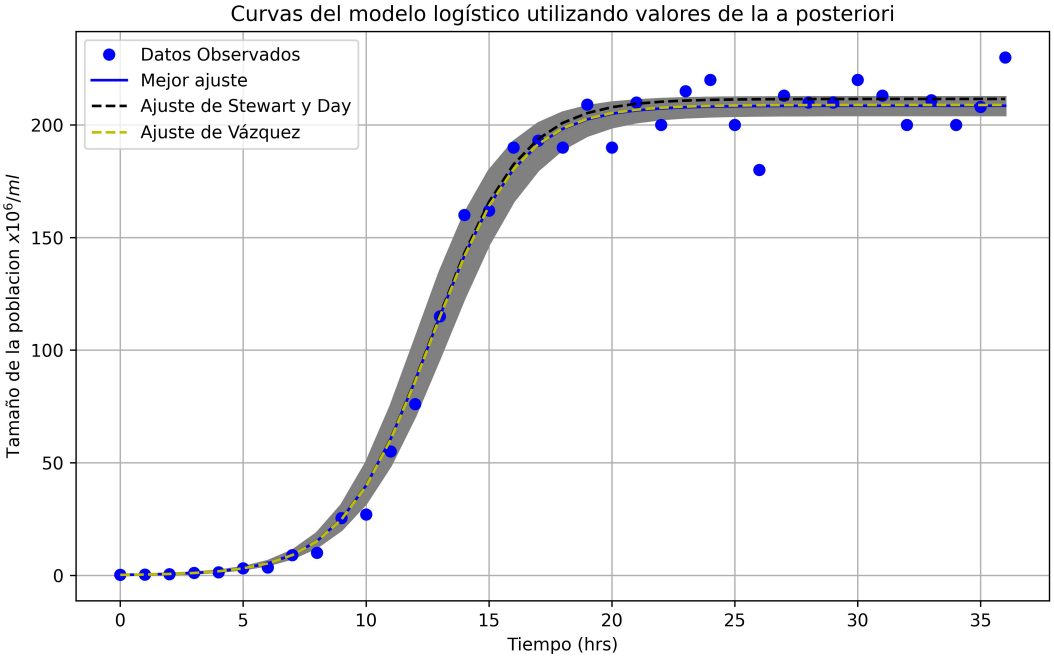


Figura 7: Incertidumbre de *Saccharomyces cerevisiae*.

incertidumbre o la variabilidad de los parámetros del modelo, en este caso, obtenida de la distribución *a posteriori*.
Esta área sombreada proporciona una visualización clara de la incertidumbre en las estimaciones de los parámetros del modelo. Muestra cómo varían las predicciones del modelo debido a la variabilidad en los datos. Esto es crucial en el enfoque bayesiano, donde la incertidumbre en los parámetros es una parte integral del análisis, por ejemplo, permite comunicar de manera efectiva la precisión y la confianza en las predicciones del modelo. El investigador puede ver, **no sólo una línea de mejor ajuste**, sino también cuánto pueden variar las predicciones. Esto es importante en contextos de toma de decisiones, donde entender la variabilidad puede influir en las decisiones basadas en los resultados del modelo. Además, facilita la comparación entre diferentes modelos o ajustes. Al superponer áreas sombreadas de diferentes modelos, se puede ver rápidamente cuál modelo proporciona predicciones más precisas o con menos incertidumbre.
Las áreas sombreadas pueden ayudar a identificar puntos de datos que se encuentran fuera de las predicciones esperadas, señalando posibles anomalías o la necesidad de ajustar el modelo. Por todo esto, es mejor optar en el futuro por resolver problemas inversos utilizando estadística bayesiana, no sólo usar métodos estadísticos y numéricos clasicos.

4 Conclusiones

El análisis de cuantificación de incertidumbre bayesiana de problemas inversos continua siendo un tema de investigación desafiante. El presente artículo es una introducción a realizar inferencia en el aspecto del análisis bayesiano de sistemas de ecuaciones diferenciales ordinarias en el contexto de problemas inversos, en particular el caso del modelo logístico.

Los problemas inversos surgen en una variedad de aplicaciones científicas y de las ingenierías, donde los parámetros del modelo deben ser estimados a partir de datos observacionales. Estos problemas se caracterizan por errores observacionales, errores de modelo y problemas de mal planteamiento que generan incertidumbres en los parámetros del modelo. Los enfoques estadísticos bayesianos permiten realizar simulaciones y predicciones

con incertidumbres cuantificadas.

Se ha destacado la dificultad añadida que presentan estos problemas debido a la incapacidad de tratar analíticamente la función del regresor, lo que nos obliga a recurrir a aproximaciones numéricas. A pesar de que comúnmente se pasa por alto la sustitución de la solución teórica por una aproximación numérica. Se ha señalado el trabajo de investigación que [2] ha llevado sobre realizar inferencia sobre la distribución *a posteriori* númerica, en lugar de la téorica.

Como se mencionó, la distribución *a posteriori* puede tomar la forma de un área sombreada alrededor de los ajustes del modelo, esta área sombreada muestra la incertidumbre en las estimaciones de los parámetros del modelo. Muestra cómo varían las predicciones del modelo debido a la variabilidad en los datos. El investigador al resolver el problema inverso utilizando este método no solo obtiene los mejores parámetros que se ajustan, sino toda una distribución para ellos, lo que ayuda a ver cuánto pueden variar las predicciones.

Uno de los logros destacables de este trabajo es presentar de manera detallada un código realizado en Python, para la obtención de la distribución *a posteriori* para los parámetros de un modelo no lineal. Además, que el mismo genera histogramas de esta distribución, lo que ayuda a comprender a la misma y analogizar entre distribuciones más conocidas. Este código no lo muestran en trabajos del mismo tema, con este nivel de detalle. Los códigos se encuentran disponibles en `https://github.com/RodGon22/RepositarioTesisRodrigo` o escaneando el código QR de la Figura 8.

Por último, la aplicación como la que se mostró en la sección 3 ilustra cómo un enfoque probabilístico para codificar errores en el proceso de modelado puede conducir a simulaciones predictivas con medidas de incertidumbre confiables en problemas del mundo real. Sin embargo, aplicaciones más rigurosas conllevan más desafíos relacionados con la computación de alto rendimiento, alta dimensionalidad en datos y parámetros, predicción y otros que no se abordaron en este trabajo que se pueden considerar problemas a futuro.

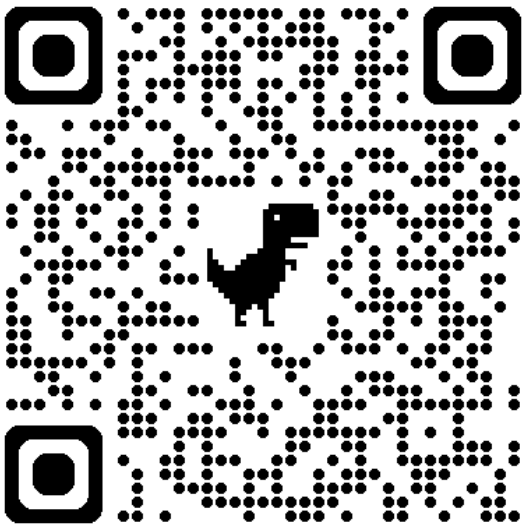


Figura 8: QR del repositario, generado con Google.

Referencias

[1] J. M. Bernardo and A. F. Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

[2] M. Capistran, J. Christen, M. Daza, H. Flores Arguedas, and J. Montesinos-López. Error control of the numerical posterior with bayes factors in bayesian uncertainty quantification. *Bayesian Analysis*, -1, 01 2021.

[3] A. Carpio. Incertidumbre y problemas inversos, 2023. `https://www.ucm.es/doctorado/doctorado_inv_mat/incertidumbre-y-problemas-inversos` [Accedido: 19 de noviembre del 2023].

[4] J. A. Christen. Cuantificación de incertidumbre bayesiana (bayesian uq), 2020. Escuela de Probabilidad y Estadística 2020, CIMAT, Guanajuato, México, 2 de noviembre del 2020.

[5] J. A. Christen and C. Fox. A general purpose sampling algorithm for continuous distributions (the t-walk). 2010.

[6] J. C. Correa Morales and C. J. Barrera Causil. Introducción a la estadística bayesiana. *Textos Académicos*, 2018.

[7] K. K. N. Elio. Improvements in chronology building from 14c measurements using bayesian inference on autoregressive gamma processes. 2014.

[8] A. Erdely and E. Gutiérrez-Peña. Monografía de estadística bayesiana. *arXiv preprint arXiv:2309.06601*, 2023.

[9] M. Iannelli and A. Pugliese. *An introduction to mathematical population dynamics: along the trail of volterra and lotka*, volume 79. Springer, 2015.

[10] N. E. Kuschinski Kathmann. Improvements in chronology building from 14c measurements using bayesian inference on autoregressive gamma processes. 2014.

[11] P. M. Lee. *Bayesian statistics: an introduction. 3rd*. Wiley, New York, 2009.

[12] B. K. Mable and S. P. Otto. Masking and purging mutations following ems treatment in haploid, diploid and tetraploid yeast (saccharomyces cerevisiae). *Genetics Research*, 77(1):9–26, 2001.

[13] M. Mendoza and P. Regueiro. Estadística bayesiana. *Instituto Teconológico de México*, 2011.

[14] M. J. D. Molina. Cuantificación de la incertidumbre desde el enfoque bayesiano en el contexto de edo, 2023. Accedido: 20 de noviembre del 2023.

[15] J. Stewart and T. Day. *Biocalculus: Calculus for Life Sciences*. Cengage Learning, 2015.

[16] M. L. D. Torres. *Numerical Solution of the Inverse Scattering Problem using High Level Representations*. PhD thesis, 2018.

[17] V. Vázquez. Notas de modelación matemática. 2023.

Coloración en gráficas de mapas en la Tierra y mapas en la Luna

Ana Teresa Calderón Juárez

Resumen

La *coloración de mapas* es un problema clásico en la *Teoría de Grafos*, donde cada país se modela como un vértice y las fronteras entre países como aristas. El *Teorema de los Cuatro Colores* establece que cualquier mapa plano puede colorearse con cuatro colores sin que dos regiones adyacentes compartan el mismo color. En este artículo, exploramos la generalización del problema de coloración de mapas al caso de la Tierra y la Luna, conocido como el **Earth Moon Problem**, propuesto por Ringel. Este problema busca determinar el número mínimo de colores necesarios para colorear un mapa donde cada país en la Tierra y su colonia lunar deben recibir el mismo color, respetando la restricción de que las regiones adyacentes en cualquiera de los dos cuerpos celestes deben tener colores distintos.

Nuestro principal aporte es demostrar que el problema de la 3-*coloración* de la Tierra-Luna es *NP-completo*, mediante una reducción desde 3-*SAT*, lo que implica que no existe un algoritmo eficiente para resolverlo en general (suponiendo $P \neq NP$). Además, complementamos demostraciones previas que aparecían incompletas en la literatura y modelamos el problema como un *problema de satisfacción de restricciones* (CSP), lo que permite un análisis más profundo de su complejidad computacional.

Este trabajo no solo aporta una nueva demostración de que el problema de coloración de la Tierra-Luna con 3 colores es NP-completo, sino que también abre la puerta a futuros estudios sobre su dificultad para diferentes números de colores.

Por último, describir el problema de coloración de la Tierra-Luna a través de grafos, un caso abierto en la coloración de grafos que extiendel problema de la coloración de mapas planos. En términos de grafos, esto se puede reformular como la búsqueda del **número cromático máximo** de un grafo G que es la unión de dos grafos planares (sobre el mismo conjunto de vértices). Se demuestra mediante inducción que G es 12-coloreable, como observó Heawood. Ringel conjeturó que el Problema de la Tierra-Luna era 8-coloreable pero Sulanke reportó un ejemplo que requiere 9 colores, aún no se conoce si existen configuraciones que requieran 10, 11 o 12 colores.

Palabras clave: Problema de la Tierra-Luna; Coloración en grafos; Teorema de los cinco colores; Teorema de los cuatro colores; Complejidad computacional; Problemas de Satisfacción de restricciones; 3SAT; Reducción de problemas; Clases de problemas.

1 Introducción

La coloración de mapas de la tierra es uno de los problemas de optimización más estudiados en la Teoría de Grafos. El problema de la coloración de un mapa consiste en asignar un color a cada vértice (país) de tal manera que dos vértices conectados por una arista obtengan colores diferentes, minimizando el número total de colores utilizados.

Ahora, la siguiente afirmación no es verdadera:

> "Es posible asignar uno de cuatro colores a cada país en cualquier mapa, de manera que ningún par de países que compartan una frontera tengan el mismo color." [6]

Sin embargo, este no es el enunciado del famoso *Teorema de los Cuatro Colores*, que fue demostrado hace más de 30 años utilizando numerosas comprobaciones por computadora.

La figura 1 muestra un pequeño ejemplo de un mapa que necesita cinco colores si cada par de países adyacentes debe recibir colores diferentes. La característica importante es que un país (#5) es desconectado y está compuesto por dos regiones.

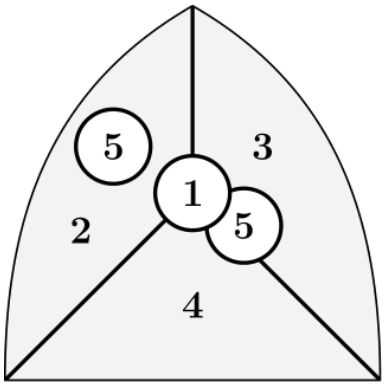


Figura 1: Un mapa plano que necesita cinco colores. Mapa tomado de [6].

Los mapas que incluyen países desconectados son posibles; un ejemplo es el mapa de América del Norte. Esto plantea la siguiente pregunta: ¿Es posible colorear el mapa actual del mundo con cuatro colores de manera que todas las partes de cada país reciban el mismo color y que dos países diferentes con un arco fronterizo en común no reciban el mismo color? Para dar una respuesta afirmativa a dicha pregunta nos apoyaremos en el siguiente teorema.

Teorema de los Cuatro Colores: Este teorema establece que cualquier mapa dibujado en el plano puede colorearse con solo cuatro colores, de tal manera que cada par de regiones conectadas que comparten un borde reciban colores diferentes.

Escrito en términos de grafos:

Teorema 1.1. *[7]*

Cualquier grafo planar simple G es 4-coloreable.

Donde:

Definición 1.2. Un **grafo** es una pareja ordenada $G = (V, E)$, donde V es un conjunto no vacío de objetos llamados **vértices** (nodos) y E es un conjunto de **aristas** (líneas) entre los pares de vértices de G.

Antes de continuar estableceremos la siguiente notación.

- Sea V el conjunto de vértices. Es habitual utilizar letras minúsculas para representar dichos vértices, es decir $V = \{a, b, c, \dots\}$. También se usan letras enumeradas con subíndices, esto es $V = \{v_1, v_2, v_3, \dots\}$.
- El conjunto de aristas $E \subseteq V \times V$, describe una relación entre los vértices de G. Comúnmente se describe a E etiquetando cada arista, es decir, $E = \{e_1, e_2, e_3, \dots\}$, donde la arista $e_i := (u_i, v_i)$ o (v_i, u_i) denota la conexión entre los vértices u_i y v_i. También usaremos la siguiente manera más simplificada $E = \{u_1v_1, u_2v_2, \dots\}$.

Definición 1.3. Una **coloración** (o **coloración propia**) de vértices en grafos es una asignación de colores a los vértices de un grafo G. De tal manera que cualquiera vértices adyacentes tienen distintos colores. Es decir, se busca una función

$$\varphi : V \longrightarrow Colores = \{1, 2, \dots, k\}$$

de tal forma que si $uv \in E$ entonces $\varphi(u) \neq \varphi(v)$.

- Una coloración usando exactamente s colores se llama ***s*-coloración**.

- Un grafo es ***k*-coloreable** si existe una s-coloración de G para algún entero $s \leq k$.

Este resultado fue propuesto por primera vez en 1852 por Francis Guthrie. Una demostración fue publicada en 1879 por Kempe, pero 11 años después, Heawood encontró un error en la prueba. Finalmente, en 1976, el teorema fue demostrado por Appel y Haken, aunque de una manera inusual. Brevemente, la demostración de Appel y Haken consiste en mostrar que cada mapa plano contiene solo una de una lista de al menos 1,800 configuraciones, y que cada configuración admite una reducción, permitiendo una demostración por inducción. Aunque los números involucrados son inusualmente grandes, la parte más inusual de la prueba es que se utilizaron aproximadamente 1,200 horas de tiempo de computadora para generar la lista de 1,800 configuraciones y verificar que las coloraciones de estas admitían la reducción necesaria [6].

El mapa de la figura 1 también podría representar límites políticos en los que el área #5 represente, por ejemplo, un imperio. Cuando Heawood encontró tanto un error en el argumento de Kempe como descubrió que no podía corregirlo, inventó generalizaciones sobre la coloración de mapas que, hasta cierto punto, pudo resolver. Su investigación dio origen al campo de la *Teoría de Grafos Topológicos* tal como se estudia hoy en día. Primero investigó el problema de la coloración de imperios: si los mapas están formados por países unidos en imperios, ¿cuántos colores se necesitan para colorear tales mapas, siempre que todos los países en un imperio reciban el mismo color y que los imperios con una frontera común reciban colores diferentes?

Ringel sugirió una variación del problema de la coloración de imperios. Supongamos que la Luna fuera colonizada y quisiéramos colorear un mapa de la tierra y la luna con el mínimo número de colores de tal forma que:

1. Las regiones adyacentes en la tierra o en la luna reciban colores diferentes, y
2. Un país en la tierra y su colonia lunar reciban el mismo color.

A este problema se le conoce como el **Problema de la Tierra-Luna**. Este problema ha estado abierto durante más de 20 años.

Por último se añade un teorema y un corolario que usaremos más adelante:

Teorema 1.4. *Sea G un grafo de orden n y tamaño m, cuyo conjunto de vértices es $V = \{v_1, v_2, \ldots, v_n\}$. Entonces se satisface que;*

$$\sum_{i=1}^{n} deg(v_i) = 2m.$$

Corolario 1.5. *Si G es un grafo planar de orden $n \geq 3$ y tamaño m, entonces se cumple la siguiente desigualdad:*

$$m \leq 3n - 6.$$

2 Complejidad computacional de problemas que involucran coloración en grafos.

Comenzaremos con algunas definiciones antes de pasar a los resultados principales. Tanto en las matemáticas como en la teoría de la informática se han estudiado diferentes aspectos de los *problemas de decisión*. Algunos de estos estudios se enfocan en determinar si existen *algoritmos* eficientes para resolverlos y en qué medida algunos problemas de decisión son más difíciles de resolver que otros. Al analizar estos problemas, encontramos que algunos presentan mayor complejidad que otros. Un problema de decisión importante dentro de la teoría de grafos, que será el principal objeto de estudio en esta sección, es el llamado *3-coloración de grafos planares*. Este problema consiste en:

Determinar si dado cualquier grafo planar no dirigido G, es posible colorear los vértices de G con tres colores, digamos (Azul, Rojo y Verde) de tal manera que ningún par de vértices adyacentes tenga el mismo color.

El problema de determinar si es posible colorear un grafo planar con 3 colores se convierte en un problema de decisión con el siguiente formato:

Problema: 3-coloración de grafos planares *G*.

Entrada: Un grafo planar no dirigido. ¿Es 3-coloreable?

Salida: Sí o No.
Sí, nos dice que el grafo es 3-coloreable; No, nos dice que no lo es.

La segunda pregunta que nos hacemos en este caso es:

¿Qué tan difícil es este problema desdel punto de vista computacional?

Para la elaboración de esta sección se consultaron los libros [9] [1] [8] y los artículos [3] [11].

Definición 2.1. Un **problema de decisión** es aquel que admite únicamente dos posibles respuestas: "Sí" o "No", para cualquier entrada.

Para resolver problemas de decisión y otros tipos de problemas matemáticos, a menudo recurrimos a algoritmos. Estos son fundamentales para la solución de problemas tanto en computación como en matemáticas. Su definición es la siguiente:

Definición 2.2. Un **algoritmo** es un conjunto finito de instrucciones matemáticas bien definidas, que, cuando se ejecutan correctamente, produce un resultado específico a partir de una entrada dada.

Los algoritmos suelen ser vistos como una secuencia de instrucciones matemáticas para resolver problemas.

Un problema de decisión que puede ser resuelto a través de un algoritmo en un tiempo finito se llama **decidible**.

Entender los algoritmos es fundamental para resolver problemas computacionales. Sin embargo, no solo es importante encontrar una solución, sino también evaluar cuan eficiente es el algoritmo que utilizamos. Aquí es donde entra en juego el concepto de *complejidad* de un algoritmo, que a continuación definimos.

Definición 2.3. La **complejidad** de un algoritmo se refiere al tiempo y la memoria requeridos para ejecutar el algoritmo en función del tamaño de la entrada.

Para cuantificar la complejidad, utilizamos la notación ***Big-O***, denotada como $O(g(n))$, que describe el crecimiento de una función f en términos del tamaño de la entrada. La comprensión de la notación Big-O es esencial para evaluar y comparar algoritmos en términos de su eficiencia.

Definición 2.4. (Big-O) Sean f, $g : \mathbb{N} \to \mathbb{R}$ dos funciones definidas en los números enteros positivos, $\mathbb{N}$. Decimos que g es una **cota superior asintótica** para f si existe un número real $C \in \mathbb{R}$ y un entero positivo $n_0 \in \mathbb{N}$ tal que:

$$|f(n)| \leq C|g(n)| \quad \text{para todo número entero positivo } n \geq n_0.$$

En tal caso, se escribe

$$f = O(g) \text{ o } f \in O(g).$$

La cota superior asintótica $g(n)$ se elige típicamente de manera que sea lo más simple y pequeña posible. Decimos que $f(n)$ tiene un crecimiento **lineal, cuadrático, cúbico o polinomial** en n si $f(n)$ pertenece a $O(n)$, $O(n^2)$, $O(n^3)$ o $O(n^k)$ para algún $k \in \mathbb{N}$, respectivamente. Cuando hablamos de la *eficiencia* de un algoritmo, nos referimos al tiempo que este tarda en ejecutarse en función del tamaño de la entrada. *Los algoritmos con tiempo de ejecución polinomial*, como la suma o multiplicación de números, se consideran **eficientes**. Por otro lado, si un algoritmo necesita iterar sobre cada instancia de un conjunto con 2^n elementos, *su complejidad es exponencial en n*. Un problema se considera **difícil** si no existe un algoritmo eficiente, es decir, de tiempo polinomial, que pueda resolverlo.

Existen clases distintas de problemas; aquellos cuyos algoritmos tienen tiempo de ejecución polinomial, o no.

3 Clases de problemas

A continuación definimos las clases de problemas más usados, para preparar esta sección nos basamos en las notas del curso *Design and Analysis of Algoritms* del Profesor Erick Demaine [3].

Definición 3.1.

- Un algoritmo es polinomial si para algún k, su tiempo de ejecución sobre las entradas de tamaño n es $O(n^k)$. La clase de **problemas *tractables***, denotada por **P**, comprende los problemas de decisión que se pueden resolver mediante un algoritmo de tiempo polinomial. Estos problemas se consideran eficientes.
- La clase de problemas **NP**, denominada **tiempo polinomial no determinista** es la clase de problemas de decisión en los que se permite adivinar y *verificar* su solución en tiempo polinomial. El hecho de que una solución pueda adivinarse a partir de muchas opciones polinomiales en tiempo constante se le conoce como no-determinista.

Definición 3.2. Dentro de la clase de problemas NP hay una clase de problemas denominados **NP-completos**, que en términos generales son considerados difíciles de resolver y si existe un algoritmo que pueda resolver un problema NP-completo en tiempo polinomial, entonces también puede resolver cualquier otro problema NP en tiempo polinomial. En otras palabras, un problema X es NP-completo si $X \in$ NP-*hard* (véase definición abajo).

En la práctica, verificar si una prueba es válida parece ser más sencillo que encontrar la solución a un problema. Sin embargo, en el campo de la informática, persiste una pregunta fundamental sin resolver: ¿Es la clase NP estrictamente más grande que la clase P? Esta pregunta es una de las incógnitas más importantes en la teoría de algoritmos.

Definición 3.3. Una **reducción** del problema A al problema B en tiempo polinomial, denotada por $A \leq_p B$, es una transformación tal que existe un algoritmo que convierte las entradas del problema A en entradas equivalentes del problema B en tiempo polinomial. Equivalente significa que para cualquier entrada, el problema A y el problema B producen la misma respuesta (sí o no). Sea $A \leq_p B$:

- Si $A \in NP - hard$, entonces $B \in NP - hard$.
- Si $B \in P$, entonces $A \in P$.
- Si $B \in NP$, entonces $A \in NP$.

Definición 3.4. Un problema X es **NP-*hard*** o **NP-duro** si cada problema $Y \in NP$ se reduce a X. Es decir, un problema es NP-*hard* si es al menos tan dificil como todos los problemas en NP.

Con lo anterior podemos definir formalmente los problemas *NP-completos*.

Definición 3.5. Un problema X es **NP-completo** si $X \in NP$ y X es NP-hard.

Un ejemplo muy famoso de problemas NP-completos es el problema denominado 3SAT, con el fin de hacer este trabajo auto contenido y definir formalmente un problema 3SAT introduciremos las siguientes definiciones.

Definición 3.6. Una **fórmula booleana** es una expresión formada con las variables $u_1, \ldots, u_n$ con $u_i \in \{0, 1\}$ para $i \in \{1, 2, \ldots, n\}$, junto con los operadores lógicos **y** ($\wedge$), **no** ($\neg$) y **o** ($\vee$). Sea φ una fórmula booleana y $z \in \{0, 1\}^n$, entonces $\varphi(z)$ denota el valor de φ cuando a las variables de φ se les asignan los valores z. Se dice que una fórmula φ satisface (o tiene solución) si existe alguna asignación z tal que $\varphi(z)$ sea verdadera.

Una fórmula booleana está en **forma CNF** (Forma Normal Conjuntiva) si es una conjunción (y) de disyunciones (o) de variables o sus negaciones. Es decir, una fórmula CNF tiene la forma

$$\bigwedge_i \left(\bigvee_j v_{ij} \right),$$

donde cada v_{ij} es un literal que puede ser la variable u_k o su negación $\neg u_k$. Los términos v_{ij} se llaman **literales** y las expresiones $(\bigvee v_{ij})$ se llaman **cláusulas**. Un k**CNF** es una fórmula CNF en la cual todas las cláusulas contienen a lo sumo k literales.

Con esta definición podemos dar paso a una definición formal de un tipo de fórmula booleana denominada 3SAT, para después demostrar que el problema de decidir si una fórmula booleana que está en 3CNF es o no válida se reduce a un problema de 3-coloración en grafos en tiempo polinomial y viceversa. Así que son equivalentemente difíciles y concluiremos que el problema de 3-coloración en grafos es NP-completo pues 3SAT lo es.

Definición 3.7. 3**SAT:** El problema de decisión 3SAT se puede plantear como sigue. Dada una fórmula booleana de la forma:

$$(x_1 \vee x_3 \vee x_6) \wedge (\overline{x_2} \vee x_3 \vee x_7) \wedge \ldots,$$

¿Existe una asignación de variables verdadero (1) y falso (0), tal que toda la fórmula evalúe a verdadero (1)?

Fue probado que el problema 3SAT resulta ser NP-completo por Cook en 1971 [1]. A continuación, enunciamos el teorema y definimos formalmente el problema:

Teorema 3.8. (Teorema de Cook-Levin [1]) *Sea SAT el lenguaje de todas las fórmulas CNF que tienen solución y* 3*SAT el lenguaje de todas las fórmulas 3CNF que tienen solución. Entonces:*

1. *SAT es NP-completo.*
2. 3*SAT es NP-completo.*

En el problema de decisión de 3-coloración, se nos da un grafo G y nos preguntamos si existe una forma de colorear los vértices de dicho grafo utilizando tres colores, a saber: rojo, verde o amarillo, de tal manera que ningún par de vértices adyacentes comparta el mismo color.

El objetivo de esta sección es demostrar una reducción en tiempo polinomial del problema 3SAT al problema de 3-coloración. Es decir, demostraremos el siguiente teorema:

Teorema 3.9. $3SAT \leq_p 3-Coloración.$

Antes de demostrar este teorema, esbozaremos la estrategia de la prueba. Dada una entrada del problema 3SAT (una fórmula booleana), debemos construir un grafo que será 3-coloreable si y solamente si la fórmula booleana tiene solución. La idea de la siguiente prueba fue tomada de [11].

Demostración. La demostración se ejecutará en varios pasos.

- **Paso 1:** Comenzamos construyendo un grafo que contiene 3 vértices etiquetados como T, F, y S, conectados formando un triángulo. Coloreamos estos vértices con tres colores diferentes. Sin pérdida de generalidad, podemos colorearlos como se muestra en la figura 2:

  En esta construcción, el color verde indica los valores verdaderos, el color rojo indica los valores falsos y el color de S no representa ningún valor.

- **Paso 2:** En nuestra fórmula, tenemos variables y negaciones de variables, conocidas como literales. Debemos asignar valores a los literales, por lo que basta asignar valores a su variable correspondiente. Para asegurar esto, hacemos lo siguiente:

  Para cada varibale x_i y su negación $\overline{x_i}$, creamos dos vértices en nuestro grafo y conectamos estos vértices al vértice S, como se muestra en la figura 3.

  Esto asegura que los literales x_i y $\overline{x_i}$ no reciban el mismo color que el vértice S, ya que están conectadas a S y, por lo tanto, deben ser coloreadas de rojo o verde. Para garantizar que x_i y $\overline{x_i}$ no reciban el mismo color (es decir, que si x_i es verdadero, entonces $\overline{x_i}$ sea falso, y viceversa), conectamos el vértice x_i con el vértice $\overline{x_i}$, como se muestra en la figura 3.

  De esta manera, si coloreamos el grafo de una forma válida, obtendremos valores de verdad para las variables y sus negaciones que son consistentes.

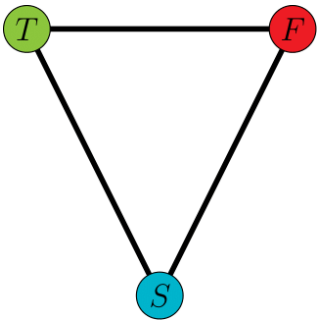


Figura 2: Primer paso

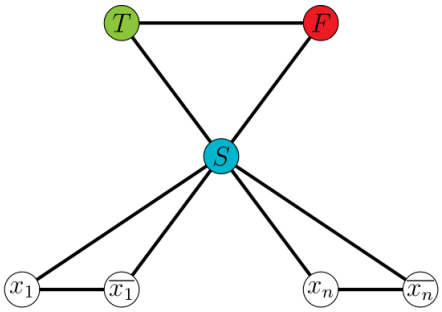


Figura 3: Segundo paso

- **Paso 3:** Finalmente, necesitamos representar cada cláusula de la fórmula booleana a través de un grafo. Para ello, representaremos el operador booleano OR mediante una construcción que llamaremos "gadget" (véase figura 4). Este gadget es 3-coloreable si y solo si al menos una de las literales en la cláusula es asignada o coloreada con T (verdadero), como veremos a continuación.

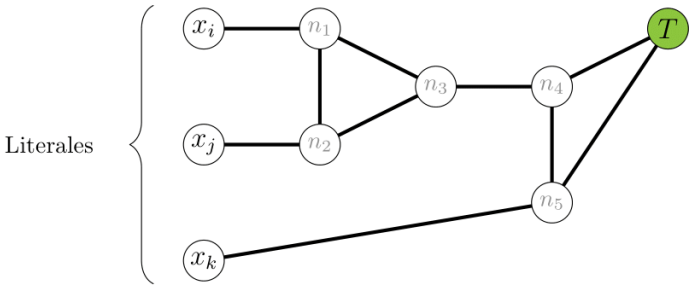


Figura 4: Tercer paso

En el gadget de la figura 4, si todas las literales se colorean como Falso (rojo), el vértice n_5 debe ser coloreado de azul, ya que está conectado a T (verde) y a x_k (rojo). A continuación, el vértice n_4 debe ser coloreado de rojo, ya que está conectado a T (verde) y a n_5 (azul), como se muestra en la figura 5:

Esto implica que ni el vértice n_3, ni n_1, ni n_2 pueden ser coloreados de rojo. Por lo tanto, solo pueden tener los colores verde o azul. Sin embargo, al tratarse de tres vértices y solo dos colores disponibles, necesariamente habrá un par de vértices que compartan el mismo color. Como estos tres vértices están conectados entre sí, se produciría una contradicción, ya que dos vértices conectados tendrían el mismo color. De este modo, si las tres literales son falsas, es imposible colorear el gadget.

- **Paso 4:** El paso final consiste en intersectar los gadgets, es decir, se crea un gadget por cada cláusula y se busca empatar esos gadgets. Aunque el resultado es grande, está bien estructurado, como se muestra en la figura 6. De este modo, el grafo resultante es 3-coloreable si y solo si la fórmula booleana se satisface.

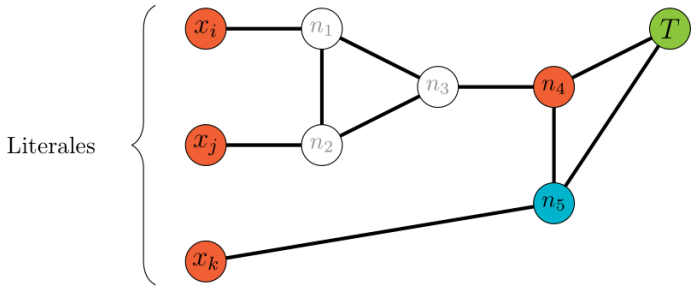


Figura 5: Coloración del tercer paso

En el caso de que la fórmula booleana representada por la figura no se satisfaga, es decir, si x_1, x_2, y $\overline{x_3}$ son coloreadas de rojo, entonces al aplicar el proceso descrito en el paso 3, se llega a la conclusión de que es imposible colorear el grafo.

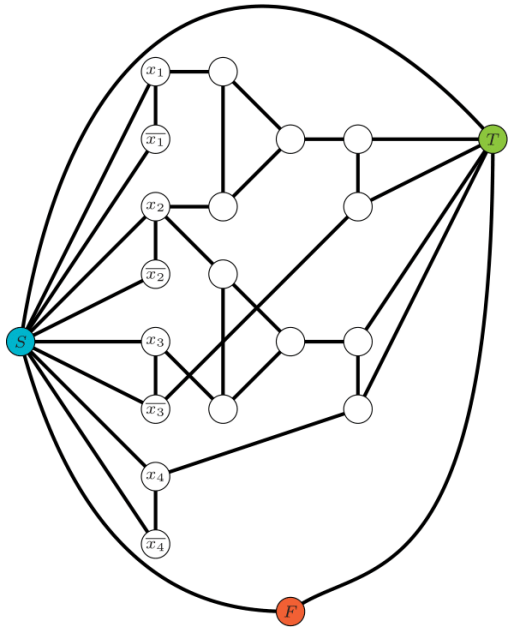


Figura 6: Reducción completa de la fórmula 3SAT
$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee x_4)$ a una 3-coloración

- **Paso 5:** Para demostrar que esta reducción se realiza en tiempo polinomial, supongamos que la fórmula tiene n variables y m cláusulas. Entonces, el número de vértices y aristas se calcula de la siguiente manera:

  Vértices:

  - 3 vértices para el triángulo formado por los vértices T, F y S (figura 2).
  - 2 vértices por cada variable: se requieren $2n$ vértices extra (figura 3).
  - 5 vértices por cada cláusula: se requieren $5m$ vértices extra (figura 4).

  Aristas:

- 3 aristas para el triángulo formado por los vértices T, F y S (figura 2).
- Aristas entre pares $(x_i, \overline{x_i})$ opuestos a las literales: n aristas (figura 3).
- Aristas de las literales al vértice S: Se tienen $2n$ aristas (figura 3).
- 10 aristas por cláusula (gadget): Se tienen $10m$ aristas (figura 4).

De este modo, el grafo generado tiene $2n + 5m + 3$ vértices y $3n + 10m + 3$ aristas. La construcción del grafo se realiza en un número de pasos que es polinomial respecto a n y m, garantizando así que la reducción se lleva a cabo en tiempo polinomial.

Hasta ahora hemos probado que un problema 3SAT se reduce a un problema de 3-coloración en grafos en general. Como 3SAT es un problema NP-completo por la reducción la 3-coloración en grafos también lo es.

□

Otra forma de demostrar que los problemas de 3-coloración en grafos son NP es modelar este problema como un problema de satisfacción de restricciones, dado que estos últimos son NP. En general, aunque en 2017 se demostró que, en el caso de que P $\neq$ NP [12], los problemas de satisfacción de restricciones pertenecen a la clase de problemas en los que se satisface la dicotomía; es decir, son P o son NP.

4 Problemas de Satisfacción de Restricciones.

Los Problemas de Satisfacción de Restricciones (CSP, por sus siglas en inglés) constituyen un área fundamental en la teoría de la computación. Estos se definen como conjuntos de objetos que deben satisfacer una serie de restricciones o limitaciones.

A continuación, los definiremos formalmente:

Definición 4.1. Un **Problema de Satisfacción de Restricciones CSP** se define como una tripleta $\langle X, D, C\rangle$, donde:

- $X = \{x_1, x_2, x_3, \ldots\}$ es un **conjunto de variables**, que puede o no ser infinito.
- $D = \{d_1, d_2, d_3, \ldots\}$ es **el conjunto de valores** que pueden tomar las variables o dominio.
- $C = \{c_1, c_2, \ldots, c_k\}$ es un **conjunto finito de restricciones**, donde cada restricción $c_i = \langle t_i, R_i\rangle$ para $i \in \{1, 2, \ldots, k\}$ consiste de una n_i**-tupla de variables** $t_i = (x_{i1}, x_{i2}, \ldots, x_{in_i})$ y una **relación** n_i**-aria** R_i sobre D.

Definición 4.2. Una **evaluación** de las variables es una función

$$\varphi : X \to D.$$

Decimos que una evaluación φ **satisface** una restricción $c_i = \langle t_i, R_i\rangle$ con $t_i = (x_{i1}, x_{i2}, \ldots, x_{in_i})$ si la tupla de valores $(\varphi(x_{i1}), \varphi(x_{i2}), \ldots, \varphi(x_{in_i}))$ pertenece a la relación R_i.

Una **solución del CSP** es una evaluación que satisface todas las restricciones del problema.

Muchos problemas prácticos pueden modelarse como problemas de satisfacción de restricciones. Un ejemplo clásico de esto es el *problema de satisfacibilidad booleana (SAT)* visto en la sección anterior. En esencia, este problema implica encontrar asignaciones de valores a variables que hagan que una fórmula lógica proposicional sea verdadera. A continuación mostraremos como un problema SAT se modela como un problema CSP. El siguiente ejemplo se construyó tomando como base el ejemplo del libro [8].

Ejemplo 4.3. El **problema de satisfacibilidad booleana** (SAT) consiste en determinar si existen valores de ceros y unos para las variables de una fórmula lógica proposicional como la siguiente, que hagan verdadera la fórmula. Consideremos la siguiente fórmula:

$$\begin{aligned} t(x_1, x_2, x_3, x_4) :=& (x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge \\ & (\overline{x_3} \vee \overline{x_4} \vee x_1) \wedge (\overline{x_3} \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}). \end{aligned}$$

El problema consiste en asignar valores de 0 o 1 para a_1, a_2, a_3, a_4 de tal manera que al evaluar la fórmula t, esta resulte verdadera, es decir, $t(a_1, a_2, a_3, a_4) = 1$. Este tipo de problemas se puede expresar como un CSP, en específico, de la siguiente manera:

- Sea $V = \{x_1, x_2, x_3, x_4\}$.
- Sea $D = \{0, 1\}$.
- Sea C el conjunto de restricciones definido por:

  $$C := \{c_1, c_2, c_3, c_4, c_5\}$$

  donde:

  - $c_1 := ((x_1, x_2, x_3, x_4), D^4 \setminus \{(0, 0, 0, 0)\})$.
  - $c_2 := ((x_1, x_2, x_3), D^3 \setminus \{(1, 1, 0)\})$.
  - $c_3 := ((x_3, x_4, x_1), D^3 \setminus \{(1, 1, 0)\})$.
  - $c_4 := ((x_3, x_2, x_4), D^3 \setminus \{(1, 1, 0)\})$.
  - $c_5 := ((x_1, x_3), D^2 \setminus \{(1, 1)\})$.

Por lo tanto, la solución del ejemplo anterior consiste en encontrar las asignaciones de valores $\varphi : X \to D$ que satisfagan todas las restricciones del problema. En este caso, las soluciones son las siguientes:

	x_1	x_2	x_3	x_4
φ_1	0	0	0	1
φ_2	0	0	1	0
φ_3	0	1	0	0
φ_4	0	1	0	1
φ_5	1	0	0	0
φ_6	1	0	0	1

Los CSP abarcan una amplia gama de problemas, desdel famoso Problema de las Ocho Reinas hasta el desafiante Teorema de los Cuatro Colores en la coloración de mapas. Numerosos juegos y rompecabezas, como por ejemplo el Sudoku, pueden modelarse como problemas de satisfacción de restricciones.

A continuación, formularemos el problema de coloración de mapas como un problema CSP, donde la tarea consiste en determinar si un mapa dado puede ser coloreado con tres colores distintos de manera que ningún par de regiones adyacentes tenga el mismo color.

Definición 4.4. El **Problema de Coloración** de un grafo (véase teorema 1.3). Sea un grafo $G = (V, E)$ donde $V = \{v_1, v_2, \ldots, v_n\}$ es el conjunto de vértices y $E = \{e_1, e_2, \ldots, e_m\}$ es el conjunto de aristas. El problema consiste en determinar si es posible colorear los vértices de G con k colores de tal manera que vértices adyacentes tengan colores diferentes.

El problema anterior lo formularemos como un CSP, donde:

- $X = \{v_1, v_2, \ldots, v_n\}$ es el conjunto de vértices de G.
- $D = \{d_1, d_2, \ldots, d_k\}$ es el conjunto de colores que pueden asignarse a los vértices.
- $C = \{c_1, c_2, \ldots, c_m\}$ es el conjunto de restricciones, donde cada restricción $c_i = \langle e_i, \neq_D \rangle$ para $i \in \{1, 2, \ldots, m\}$, con $e_i = (u_i, v_i)$ representando las aristas de G y $\neq_D$ la relación de desigualdad sobre D, definida como:
$$\neq_D = \{(d_i, d_j) \in D^2 \mid d_i \neq d_j\}.$$

Por lo tanto, una solución al problema de colorear el mapa consiste en encontrar una función

$$\varphi : X \longrightarrow D$$

que asigne un color a cada vértice en $v_i \in X$ con $i \in \{1, 2, \ldots, n\}$ tal que

$$\text{para todo } e_i = (u_i, v_i) \in E \text{ entonces } (\varphi(u_i), \varphi(v_i)) \in \neq$$

Lo anterior equivale a ver si

$$\text{para todo } e_i = (u_i, v_i) \in E \text{ entonces } \varphi(u_i) \neq \varphi(v_i)$$

Finalizaremos esta sección modelando un problema de 3-coloración como un problema 3SAT.

Ejemplo 4.5. Modelando un problema de 3-coloración de grafos como un problema 3SAT.
Sea un grafo $G = (V, E)$. Formularemos este problema como un 3SAT de la siguiente manera:

- Las variables del conjunto $u \in \{u_{1,i}, u_{2,i}, \ldots, u_{n,i}\}$ con $i \in \{1, 2, 3\}$ representan los colores asignados a los vértices en G. Hay 3 variables por cada vértice en G, una por cada color posible.
- El dominio $D = \{0, 1\}$ indica los valores que puede tomar cada variable u, donde 1 significa que el color es asignado y 0 que no lo es.
- Para cada vértice $v_j \in V$ con $j \in \{1, 2, \ldots, n\}$, asignamos las siguientes restricciones:
  - Cada vértice debe ser coloreado con al menos un color, es decir, se debe satisfacer la siguiente fórmula:
  $$u_{j,1} \vee u_{j,2} \vee u_{j,3}$$
  para esta restricción se necesita una sola cláusula por cada vértice, es decir n cláusulas en total para el grafo.
  - En nuestro problema de coloración no se pueden asignar dos colores al mismo tiempo al mismo vértice y esto se logra con la siguiente expresión lógica:
  $$(\overline{u_{j,1}} \vee \overline{u_{j,2}}) \wedge (\overline{u_{j,1}} \vee \overline{u_{j,3}}) \wedge (\overline{u_{j,2}} \vee \overline{u_{j,3}})$$
  para esta restricción se necesitan $3 \times n$ cláusulas.
  - Cada par de vértices adyacentes son de diferente color. Para todo $e_j = (v_j, v_k) \in E$ se tiene la restricción:
  $$(\overline{u_{j,1}} \vee \overline{u_{k,1}}) \wedge (\overline{u_{j,2}} \vee \overline{u_{k,2}}) \wedge (\overline{u_{j,3}} \vee \overline{u_{k,3}})$$
  para esta restricción se necesitan $3 \times m$ cláusulas

Así, el problema de 3-coloración se reduce a encontrar una asignación a las variables $z \in \{0, 1\}^{3n}$ que haga que la fórmula booleana descrita por la intercesión de todas las clausulas anteriores sea verdadera.

Con este ejemplo, observamos que para cualquier grafo $G = (V, E)$ se puede escribir como un problema 3SAT utilizando un total de $n + 3n + 3m = 4n + 3m$ restricciones. De este modo hay una reducción de un problema de coloración en grafos a un problema 3SAT en tiempo polinomial.

Para demostrar que el problema de decidir si un grafo planar se puede colorear con 3 colores es NP-completo, se realiza una reducción en tiempo polinomial desdel problema de colorear cualquier grafo con 3 colores. Es decir, se demuestra el siguiente teorema:

Teorema 4.6. *El problema de decidir si un grafo es* 3*-coloreable se reduce en tiempo polinomial al problema de decidir si un grafo planar es* 3*-coloreable.*

$$\textit{Grafo 3-coloreable} \leq_p \textit{Grafo planar 3-coloreable}$$

La estrategia consiste en comenzar con un grafo no planar que tenga cruces, y reemplazar cada cruce por un *gadget planar*: un subgrafo de 11 vértices que puede ser coloreado con 3 colores. Este proceso se realiza para cada cruce, transformando así el grafo original en un grafo planar que es 3-coloreable si, y solo si, el grafo original no planar también lo es. El lector interesado en los detalles de la reducción puede consultar [5].

A partir del teorema anterior y de la reducción del problema 3-SAT al problema de 3-coloración, obtenemos que:

$$\text{3-SAT} \leq_p \text{Grafo planar 3-coloreable.}$$

Y, dado que 3-SAT es NP-completo, concluimos que el problema de 3-coloración en grafos planares también es NP-completo.

5 El Problema de la Tierra-Luna (Earth-Moon Problem).

El problema de la Tierra-Luna es un problema que permanece abierto dentro del ámbito de la coloración de grafos, planteado por Gerhard Ringel en 1959 [6]. Como vimos en la introducción este problema es una extensión del problema de la coloración de mapas planos, cuya solución se obtiene a través del Teorema de los Cuatro Colores (véase teorema 1.1).

De manera intuitiva, el problema puede enunciarse de la siguiente forma:

¿Cuántos colores se necesitan para colorear los mapas políticos de la tierra y la luna en un futuro hipotético, donde cada país de la tierra tiene una colonia en la luna que debe recibir el mismo color?

Figura 7: Territorio de países en tierra y luna.

En la figura 7 se muestra un ejemplo de una coloración para la tierra y la luna. Nótese que cada país debe ser coloreado con el mismo color tanto en la tierra como en la luna, aunque la disposición geográfica es diferente en ambos cuerpos celestes.

La pregunta anterior se puede formular de la siguiente manera: ¿Cuál es el número mínimo de colores necesarios para colorear un conjunto de países, de tal forma que no haya dos países que compartan una frontera común y estén coloreados con el mismo color, bajo la condición de que cada país consiste en una región en la tierra y una región en la luna?

En términos de teoría de grafos, esta pregunta se puede reformular como sigue:

¿Cuál es el número cromático máximo de un grafo G que es la unión de dos grafos planares (sobre el mismo conjunto de vértices)?

Nos gustaría establecer algunas cotas sobre el número de colores necesarios. Comenzaremos con un ejemplo dado por Thom Sulanke en 1974, que refuta la conjetura de Ringel, la cual postulaba que 8 colores serían suficientes para resolver la pregunta.

Antes de continuar, es importante señalar que para la preparación de esta sección se consultaron los artículos [10], [6] y [4]. La mayoría de las demostraciones aquí expuestas se completaron o se hizo una demostración alternativa.

Ejemplo 5.1. Supongamos que tenemos la coloración del mapa en "la tierra" donde cada letra $\{A, B, C, D, E, F, P, Q, R, S, T\}$ representa un país diferente.

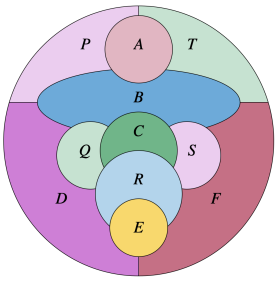


Figura 8: Coloración en "la tierra".

El grafo correspondiente G_1 (figura 9) para este mapa tiene 11 vértices (uno por cada país) y 26 aristas (una por cada colindancia), cumpliendo la desigualdad de Euler para grafos planos, $m = 26 < 3 \times 11 - 6$, donde m es el número de aristas y 11 es el número de vértices. Así,

$$G_1 = (V_1, E_1)$$

queda definido por

$$V_1 = \{A, B, C, D, E, F, P, Q, R, S, T\}$$

$$E_1 = \left\{ \begin{array}{l} AT, AB, AP, BC, BD, BF, BT, BP, BS, BQ, \\ CQ, CS, CR, DP, DQ, DR, DE, DF, ER, EF, FT, \\ FS, FR, PT, QR, RS \end{array} \right\}$$

El grafo G_1 es isomorfo al grafo de la figura 10, que a simple vista no parece ser plano; sin embargo, como se observa en la figura 9, sí posee una representación plana. Más adelante usaremos este grafo para simplicidad visual.

A continuación, analizamos el mapa en la luna. Supongamos que los 11 países del mapa terrestre se mantienen en la luna, pero en este caso, cada país está dividido en regiones diferentes a las del mapa terrestre. Esta variación en la división regional altera las colindancias entre las regiones en la luna, en comparación con las del mapa de la tierra.

Denotemos por G_2 el grafo que representa el mapa lunar (figura 12). En G_2 las aristas indican las nuevas colindancias entre las regiones lunares.

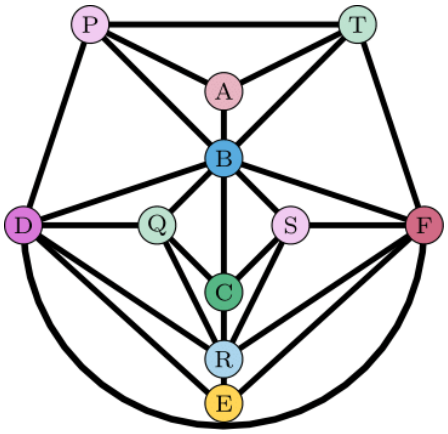


Figura 9: Grafo de mapa en "la tierra". G_1

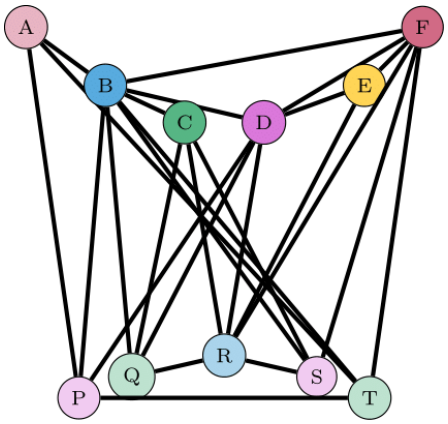


Figura 10: Grafo de mapa en "la tierra".

El grafo G_2 correspondiente al mapa lunar tiene 11 vértices (uno por cada región) y 24 aristas. Así, $G_2 = (V_2, E_2)$ queda definido por

$$V_2 = \{A, B, C, D, E, F, P, Q, R, S, T\}$$

$$E_2 = \left\{ \begin{array}{l} AC, AQ, AF, AD, AS, AE, AR, \\ BR, BE, CF, CD, CT, CE, CP, \\ DS, DT, ET, ES, EQ, EP, \\ FQ, FP, PQ, ST \end{array} \right\}$$

Reordenando los vértices alfabéticamente para facilitar el análisis, obtenemos el grafo de la figura 13.

Para garantizar que las colindancias sean consistentes entre los dos mapas, debemos considerar la unión de los grafos G_1 y G_2. En dondel conjunto de vértices $\{A, B, C, D, E, F, P, Q, R, S, T\}$ es el mismo, pero el de aristas será la unión dellos. Al grafo resultante, que representa la combinación de las regiones de ambos mapas, le llamamos $G_3 = (V_3, E_3)$ donde:

$$V_3 = V_1 = V_2 = \{A, B, C, D, E, F, P, Q, R, S, T\}$$

$$E_3 = E_1 \cup E_2 = \left\{ \begin{array}{l} AC, AQ, AF, AD, AS, AE, AR, BR, BE, CF, \\ CD, CT, CE, CP, DS, DT, ET, ES, EQ, EP, \\ FQ, FP, PQ, ST, AT, AB, AP, BD, BF, BT, \\ BP, BS, BQ, CQ, CS, CR, DP, DQ, DR, DE, \\ ER, EF, FT, FS, FR, PT, QR, RS \end{array} \right\}$$

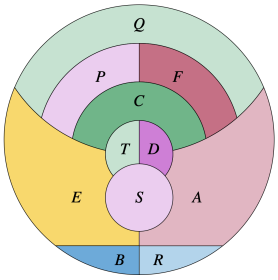


Figura 11: Coloración en "la luna".

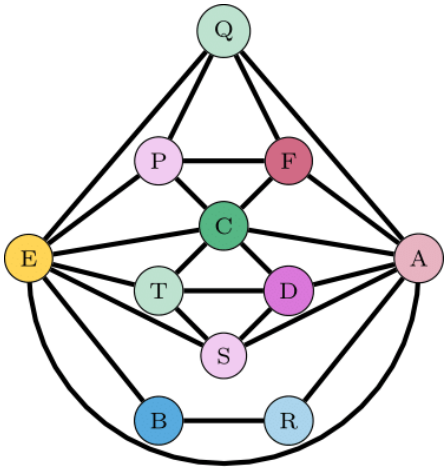


Figura 12: Grafo de mapa en "la luna". G_2

El grafo final G_3 combina las coloraciones de las regiones de la Tierra y la Luna, respetando las adyacencias especificadas en ambos mapas. La pregunta clave es: ¿cuál es el número mínimo de colores necesarios para colorear este grafo de manera que no haya dos regiones adyacentes con el mismo color?

En este caso, hemos determinado que se requieren al menos 9 colores, y este número es óptimo. Esto se debe a que el subgrafo formado por los vértices A a F y las aristas que los conectan es un grafo completo K_6, donde cada vértice está conectado con todos los demás. Por lo tanto, se necesitan al menos 6 colores para colorear K_6.

Por otro lado, el subgrafo formado por los vértices P a T es un ciclo C_5, que también requiere colores distintos de los usados en K_6. Para colorear un ciclo C_5, se necesitan al menos 3 colores adicionales. Dado que G_3 es la unión de K_6 y C_5, la cota mínima de colores necesarios es 9. Además, este número es suficiente, como se ha demostrado en el ejemplo, donde se logra una coloración correcta con 9 colores.

Como se puede observar, el Problema de la Tierra-Luna se puede formular como un grafo $G = (V, E)$ cuyo conjunto de vértices es el mismo, pero cuyo conjunto de aristas se puede dividir para formar dos grafos planares G_1 y G_2. A la operación inversa, es decir, construir un grafo de Tierra-Luna la llamaremos *unión*.

Definición 5.2. Sean $G_1 = (V, E_1)$ y $G_2 = (V, E_2)$ dos grafos planos definidos sobre el mismo conjunto de vértices V, definimos la **unión** de estos grafos como el grafo $G = (V, E)$, donde $E = E_1 \cup E_2$. Es decir, el grafo unión mantiene el mismo conjunto de vértices y su conjunto de aristas corresponde a la unión de las aristas de G_1 y G_2.

Por el corolario 1.5, sabemos que un grafo planar con $n \geq 3$ vértices tiene a lo sumo $m \leq 3n - 6$ aristas. De este modo, el Problema de la Tierra-Luna tendrá como máximo $3n - 6$ aristas en un grafo plano G_1 y $3n - 6$ aristas en el grafo plano G_2.

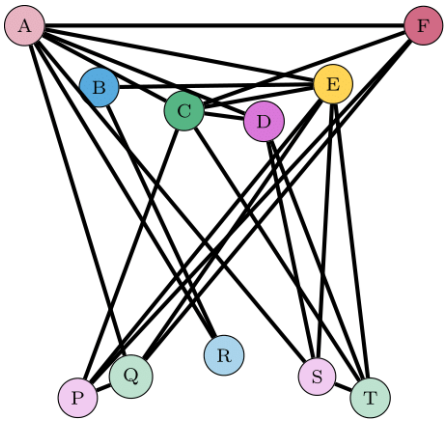


Figura 13: Grafo de mapa en "la luna".

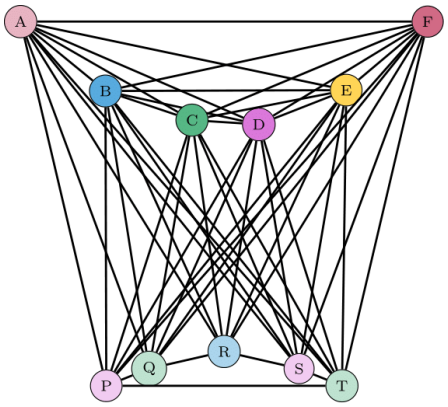


Figura 14: Grafo Tierra-Luna. G_3

Por lo tanto, tenemos el siguiente corolario.

Corolario 5.3. *Si G es un grafo con $n \geq 3$ vértices que es la unión de dos grafos planares, entonces G tendrá como máximo $|E| \leq 2(3n-6) = 6n-12$ aristas.*

Corolario 5.4. *Si G es un grafo que es la unión de dos grafos planares, existe al menos un vértice con grado no mayor a 11.*

Demostración. Si $n \leq 2$ el resultado es trivial, ya que todos los vértices tendrán grado no mayor a 11. Supongamos que $n \geq 3$.

El resultado se obtendrá por contradicción. Supongamos, que tenemos un grafo $G = (V, E)$ unión de dos planares de orden n y tamaño m en el cual todos sus vértices tienen un grado mayor o igual a 12.
Usando el corolario anterior (5.3) y el teorema 1.4, podemos afirmar que:

$$\sum_{i=1}^{n} deg(v_i) = 2m \leq 2(6n-12) = 12n-24.$$

Sin embargo, bajo nuestra hipótesis de que todos los vértices tienen un grado mayor o igual a 12, y aplicando nuevamente el teorema 1.4, obtenemos:

$$12n \leq \sum_{i=1}^{n} deg(v_i) \ = 2m.$$

Combinando las dos desigualdades anteriores obtenemos:

$$12n \leq 2m \leq 12n - 24.$$

Esta última desigualdad es una contradicción. Por lo tanto, llegamos a la conclusión de que debe existir al menos un vértice en G con un grado a lo más de 11. □

Teorema 5.5. *[6] El grafo G del Problema de la Tierra-Luna es* 12*-coloreable.*

Demostración. Probaremos este teorema usando inducción matemática sobre el orden de G. Sea n el orden de G.

Base de inducción: Supongamos que $n \leq 12$. Si G tiene $n \leq 12$ vértices, entonces existe una 12-coloración para G, ya que, basta con asignar un color distinto a cada vértice, y por construcción, no habrá dos vértices adyacentes con el mismo color. Por lo tanto, el resultado se cumple para $n \leq 12$.

Hipótesis de inducción: Supongamos que todo grafo planar de orden n es 12-coloreable.

Paso de inducción: Por demostrar que cualquier grafo $G = (V, E)$ de orden $n + 1$ es 12-coloreable. Por el colorario 5.4 sabemos que en cualquier grafo que es la unión de dos grafos planares existe al menos un vértice v con $deg(v) \leq 11$. Sea $G' = (V', E)$ el subgrafo de G, donde $V' = V \setminus \{v\}$ y E' incluye todas las aristas que no son incidentes con v. En otras palabras, G' es el grafo resultante de "eliminar" v y todas las aristas que lo conectan. Entonces, el orden de G' es n, y por la hipótesis de inducción, obtenemos que G' es 12-coloreable. De lo anterior se sigue que existe una s-coloración para G' con $s \leq 12$. Utilizamos esta 12-coloración para G con los colores $\{c_1, c_2, \ldots, c_{12}\}$, y solo falta asignar un color a v. Sabemos que v tiene a lo más 11 aristas incidentes a él. Supongamos sin pérdida de generalidad que estas aristas están conectadas a lo más a 11 vértices diferentes $\{v_1, v_2, \ldots, v_{11}\}$. Asignamos el color no utilizado para $\{v_1, v_2, \ldots, v_{11}\}$ a v. De esta manera, se conservan las características de la 12-coloración en G, y G de orden $n + 1$ es 12-coloreable. □

Hasta aquí, el ejemplo 14 demuestra que se requieren al menos 9 colores para resolver el Problema de la Tierra-Luna. Por otro lado, el teorema 5.5 establece que es posible utilizar hasta 12 colores. Esto plantea la pregunta: ¿es factible lograr una coloración con menos de 12 colores? Por lo que sabríamos que la respuesta al mínimo número de colores necesarios se encuentra en el conjunto:

$$\{9, 10, 11, 12\}$$

Dejando abiertas las posibilidades de exploración en torno a la existencia de configuraciones que requieran 10 u 11 colores. Esta incertidumbre resalta la complejidad del problema y nos surge la pregunta de cual es la complejidad computacional del problema de n-coloración de la Tierra-Luna con $3 \leq n \leq 11$.

Para concluir esta sección, modelaremos el Problema de la Tierra-Luna mostrado en la figura 7 como un Problema de Satisfacción de Restricciones (CSP).

Con el objetivo de analizar su complejidad computacional, en la siguiente sección presentaremos la modelación del Problema de la Tierra-Luna como un CSP, con el propósito de demostrar que es un problema de complejidad NP-hard.

6 Problema de la Tierra-Luna a través de los CSP.

Empezamos esta sección construyendo el siguiente ejemplo. Tenemos una porción de países europeos: Alemania, Austria, Bélgica, Francia, Países Bajos, Luxemburgo, Polonia, República Checa y Suiza (véase figura 7). El problema consiste en colorear los países utilizando el mínimo número de colores de tal manera que no haya dos países colindantes que compartan el mismo color.

Primero, describiremos el mapa como dos grafos, donde cada vértice representará un país y una arista unirá dos países si estos colindan, i.e. Sea $G_t = (V, E)$ el grafo en la tierra donde:

- $V_{=}\{Al, Au, Be, Fr, Pb, Lu, Po, Re, Su\}$ representa los países de Alemania, Austria, Bélgica, Francia, Pblanda, Luxemburgo, Polonia, República Checa y Suiza respectivamente.
- E_t representa colindancias entre los países en la tierra: $E_t = \{AlPo, AlRe, AlAu, AlSu, AlFr, AlLu, AlBe, AlPb, AuRe, AuSu, BePb, BeLu, BeFr, FrLu, FrSu, PoRe\}$

Además, sea $G_l = (V, E)$ el grafo en la luna donde:

- $V_{=}\{Al, Au, Be, Fr, Pb, Lu, Po, Re, Su\}$ representa los países de Alemania, Austria, Bélgica, Francia, Países Bajos, Luxemburgo, Polonia, República Checa y Suiza respectivamente.
- E_l representa colindancias entre los países en la luna: $E_l = \{AlPb, AlLu, AlAu, AuLu, AuSu, BeFr, BeLu, BePb, FrLu, FrRe, PbLu, LuPo, LuRe, LuSu, PoRe, PoSu\}$

Por lo tanto, los grafos planares que representa los mapas en la tierra y en la luna están representados en la figura 15. Una vez dada la representación gráfica, formularemos este problema como una instancia de CSP. El Problema **Tierra-Luna** de arriba se puede plantear como un CSP de la siguiente manera:

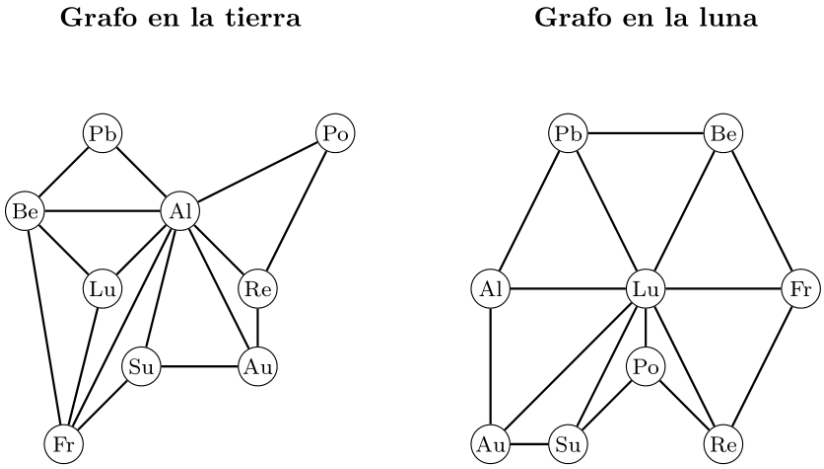


Figura 15: Representación gráfica del Problema de la Tierra-Luna.

- Sea $X = \{Al, Au, Be, Fr, Pb, Lu, Po, Re, Su\}$ nuestro conjunto de variables, representando cada país.
- Sea $D = \{V, Az, R, Am\}$ el dominio de los valores de las variables, que representan los colores (por ejemplo, Verde, Azul, Rosa y Amarillo) que se pueden asignar a los vértices.
- Sea C el conjunto de restricciones sobre las variables. Definimos las restricciones de colindancia de la siguiente manera:
  - **Tierra:** Los países adyacentes deben tener colores diferentes. Las colindancias son:

$$C_t = \left\{ \begin{array}{l} \langle AlPo, \neq\rangle, \langle AlRe, \neq\rangle, \langle AlAu, \neq\rangle, \\ \langle AlSu, \neq\rangle, \langle AlFr, \neq\rangle, \langle AlLu, \neq\rangle, \\ \langle AlBe, \neq\rangle, \langle AlPb, \neq\rangle, \langle AuRe, \neq\rangle, \\ \langle AuSu, \neq\rangle, \langle BePb, \neq\rangle, \langle BeLu, \neq\rangle, \\ \langle BeFr, \neq\rangle, \langle FrLu, \neq\rangle, \langle FrSu, \neq\rangle, \\ \langle PoRe, \neq\rangle \end{array} \right\}$$

- **Luna:** Similarmente, las colindancias para el grafo en la luna son:

$$C_l = \left\{ \begin{array}{l} \langle AlPb, \neq \rangle, \langle AlLu, \neq \rangle, \langle AlAu, \neq \rangle, \\ \langle AuLu, \neq \rangle, \langle AuSu, \neq \rangle, \langle BeFr, \neq \rangle, \\ \langle BeLu, \neq \rangle, \langle BePb, \neq \rangle, \langle FrLu, \neq \rangle, \\ \langle FrRe, \neq \rangle, \langle PbLu, \neq \rangle, \langle LuPo, \neq \rangle, \\ \langle LuRe, \neq \rangle, \langle LuSu, \neq \rangle, \langle PoRe, \neq \rangle, \\ \langle PoSu, \neq \rangle \end{array} \right\}$$

En este contexto, la relación binaria $\neq$ establece que los colores asignados a países adyacentes deben ser distintos tanto en la tierra como en la luna.

Por lo tanto, una solución al problema de colorear el mapa en la tierra consiste en encontrar una función

$$\varphi : X \longrightarrow D$$

que asigne un color a cada vértice en $v_i \in X$ con $i \in \{1, 2, 3, \ldots, 9\}$ tal que

$$\text{para todo } (v_i, v_j) \in E_t \text{ entonces } (\varphi(v_i), \varphi(v_j)) \in \neq$$

Esto último es equivalente a decir que $\varphi(v_i) \neq \varphi(v_j)$.

De manera similar, para el problema de la Luna, buscamos la misma función

$$\varphi : X \longrightarrow D$$

que asigne un color a cada vértice en $v_i' \in X$ tal que

$$\text{para todo } (v_i', v_j') \in E_l \text{ entonces } (\varphi(v_i'), \varphi(v_j')) \in \neq$$

Ahora, dado que buscamos satisfacer simultáneamente las restricciones tanto en la tierra como en la luna, el CSP que representa la unión de estas condiciones implica que X es el conjunto de variables definido anteriormente, y D representa los mismos colores. El conjunto de restricciones es la unión de las restricciones de colindancia para ambos contextos:

- **Tierra-Luna:** Los países adyacentes deben tener colores diferentes, tanto en la tierra como en la luna. La unión de las colindancias es:

$$C_{tl} = \left\{ \begin{array}{l} \langle AlPo, \neq \rangle, \langle AlRe, \neq \rangle, \langle AlAu, \neq \rangle, \\ \langle AlSu, \neq \rangle, \langle AlFr, \neq \rangle, \langle AlLu, \neq \rangle, \\ \langle AlBe, \neq \rangle, \langle AlPb, \neq \rangle, \langle AuRe, \neq \rangle, \\ \langle AuSu, \neq \rangle, \langle BePb, \neq \rangle, \langle BeLu, \neq \rangle, \\ \langle BeFr, \neq \rangle, \langle FrLu, \neq \rangle, \langle FrSu, \neq \rangle, \\ \langle PoRe, \neq \rangle, \langle PbLu, \neq \rangle, \langle LuPo, \neq \rangle, \\ \langle LuRe, \neq \rangle, \langle LuSu, \neq \rangle, \langle PoSu, \neq \rangle \end{array} \right\}$$

Por lo tanto, una solución al problema de colorear el mapa en la Tierra-Luna consiste en encontrar una función $\varphi : X \longrightarrow D$ que asigne un color a cada vértice $v_i \in X$ con $i \in \{1, 2, 3, \ldots, 9\}$, de manera que

$$\text{para todo } (v_i, v_j) \in (E_t \cup E_l) \text{ entonces } (\varphi(v_i), \varphi(v_j)) \in \neq$$

Esto es equivalente a afirmar que $\varphi(v_i) \neq \varphi(v_j)$.

Una posible solución para el Problema de la Tierra-Luna es:

$$\begin{array}{lll} \varphi(Al) = Am, & \varphi(Au) = Az, & \varphi(Be) = V, \\ \varphi(Fr) = Az, & \varphi(Pb) = Az, & \varphi(Lu) = R, \\ \varphi(Po) = Az, & \varphi(Re) = V, & \varphi(Su) = V. \end{array}$$

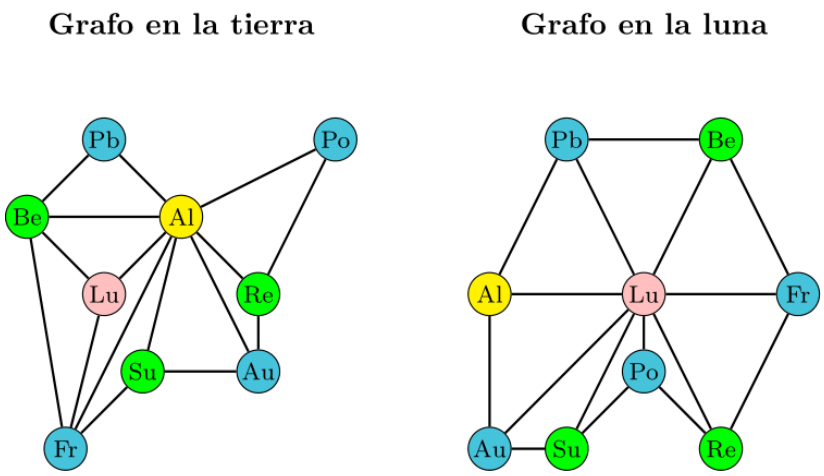


Figura 16: Una solución gráfica del Problema de la Tierra-Luna.

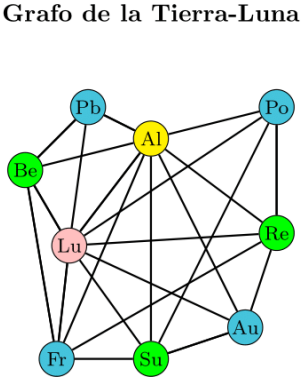


Figura 17: Grafo combinado de la Tierra-Luna.

Estas soluciones se ilustran en la figura 16, que corresponde a cada uno de los mapas.

Finalmente, en la figura 16 se presenta los grafos que modelan la tierra y la luna, asociados a esta solución. Adicionalmente, en la figura 17 se representa el grafo que es la unión de estos dos grafos planares, y sería el que se modeló al final.

Hasta el momento, hemos demostrado que el problema de coloración del modelo Tierra-Luna presenta desafíos significativos, con un mínimo de 9 colores necesarios y la posibilidad de utilizar hasta 12. Este análisis nos lleva a considerar la complejidad intrínseca de la coloración de grafos y abre la puerta a la exploración de la complejidad computacional para la n-coloración del Problema de la Tierra-Luna con $3 \leq n \leq 8$. Debra Boutin, Ellen Gethner y Thom Sulanke en [2] proporcionaron un catálogo de 40 nuevos ejemplos (configuraciones) del Problema de la Tierra-Luna que pueden ser coloreados con 9 colores. Además, presentan una nueva metodología para construir configuraciones adicionales, lo que permite crear una familia infinita de ejemplos de problemas de la Tierra-Luna que son 9-coloreables.

Debido a que el Problema de la Tierra-Luna puede ser modelado como un problema de satisfacción de restricciones (CSP), sabemos que este problema tiene una complejidad en NP-hard. Sin embargo, para $n \geq 12$, el teorema 5.5 establece que el problema de decidir si un grafo que representa un problema de la Tierra-Luna es n-coloreable siempre tiene una solución afirmativa. Esto implica que dicho problema puede resolverse en tiempo polinomial. El problema de la 3-coloración en la Tierra es NP-completo, ya que se redujo del problema 3-SAT a la 3-coloración de grafos. Por lo tanto, el Problema de la Tierra-Luna con 3 colores también será NP-completo, dado que en la tierra y la luna lo es.

Para concluir esta sección, construiremos una reducción en tiempo polinomial del problema 3SAT al problema

de 3-coloración de la Tierra-Luna. De esta manera, probaremos de forma independiente, y sin hacer uso del hecho de que el problema de 3-coloración en la Tierra es NP-completo, que el problema de 3-coloración de la Tierra-Luna también es NP-completo.

Teorema 6.1. $3SAT \leq_p$ *Problema de la Tierra − Luna.*

Antes de demostrar este teorema, esbozaremos la estrategia de la prueba. Dada una entrada del problema $3SAT$ (una fórmula booleana), debemos construir un mapa en la tierra y un mapa en la luna que será 3-coloreable si y solamente si la fórmula booleana tiene solución.

Demostración. La demostración se ejecutará en varios pasos. Sea una entrada del problema $3SAT$ con n variables y m cláusulas.

- **Paso 1:** Empezamos construyendo un mapa de la tierra que incluye tres países adyacentes, etiquetados como T, F, y S. Aquí, T es un cuadrado, F es un $(m+2)$-ágono, y S es $(n+2)$-ágono. Asignamos a cada país un color distinto utilizando tres colores diferentes. Sin pérdida de generalidad, los colores se distribuyen como se muestra en la figura 18:

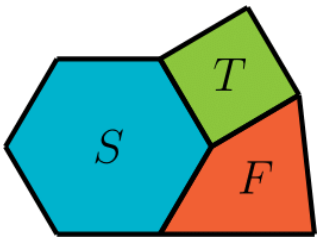


Figura 18: Primer paso.

  En esta construcción, el color verde representa los valores verdaderos, el rojo representa los valores falsos y el color de S no indica ningún valor específico.

- **Paso 2:** En nuestra fórmula, se incluyen variables y sus negaciones, denominadas literales. Asignamos valores a cada literal de manera que si una literal x_i es verdadera su negación $\overline{x_i}$ sea falsa. Para garantizar esta condición, procedemos de la siguiente forma:

  Para cada literal x_i y su negación $\overline{x_i}$, agregamos dos países (cuadrados) en nuestro mapa de la tierra. Estos países se conectan entre sí y a uno de los lados disponibles del país S (disponibles n lados, uno para cada variable y su negación), como se ilustra en la figura 19.

  Esto asegura que las literales x_i y $\overline{x_i}$ no reciban el mismo color que el país S, ya que sus países están conectadas a S y, por lo tanto, deben ser coloreadas de rojo o verde. Para garantizar que x_i y $\overline{x_i}$ no reciban el mismo color (es decir, que si x_i es verdadero, entonces $\overline{x_i}$ sea falso, y viceversa), conectamos el país x_i con el país $\overline{x_i}$, como se muestra en la figura 19.

  De esta manera, si coloreamos el mapa de una forma válida, obtendremos valores de verdad para las variables y sus negaciones que son consistentes.

- **Paso 3:** Finalmente, debemos representar cada cláusula de la fórmula booleana en el mapa. Para esto, utilizaremos una construcción denominada "gadget" para simular el operador booleano OR (véase la figura 20). Este gadget es 3-coloreable si y solo si al menos uno de los países representado por las literales en la cláusula se asigna o colorea con T (verdadero), como se explicará a continuación.

  El gadget que construiremos es análogo al descrito en el paso 3 de la demostración del Teorema 3.9, con la diferencia de que en esta ocasión se conecta al país que representa el valor *Falso*. Una parte del gadget se sitúa en la "tierra" y consiste en varios países conectados a uno de los lados del país correspondiente a F. Dado que el polígono que representa a F tiene $m+2$ lados, podemos utilizar un lado distinto para cada gadget asociado a cada cláusula de la fórmula booleana.

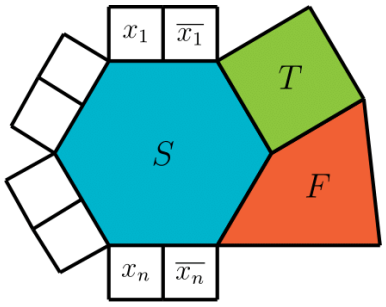


Figura 19: Segundo paso.

Cada uno de estos gadgets incluye también países en la "luna", los cuales representan los literales de la cláusula correspondiente. Para ello, cada literal se modela mediante un polígono en la luna: específicamente, utilizamos un m-ágono para cada literal, o un triángulo en caso de que $m < 3$. Esta construcción permite representar adecuadamente la estructura lógica de cada cláusula dentro del grafo, respetando las restricciones de adyacencia y colorabilidad necesarias para la reducción.

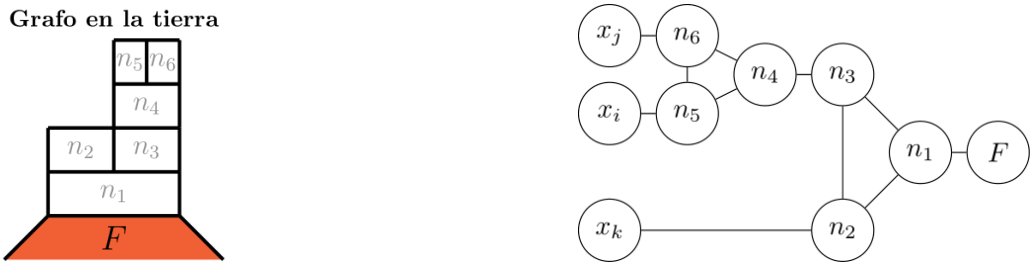


Figura 20: Tercer paso (tierra).

Estos países de las literales en la luna no serán colindantes entre sí. En este modelo, para cada cláusula, adicional se agregarán algunos de los países del gadget en uno de los lados de los m-ágonos que representan las literales utilizadas en esa cláusula.

Grafo en la luna

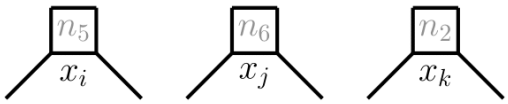


Figura 21: Tercer paso (luna).

En el gadget mostrado en la figura 20 que se representa con los grafos de las figura 20 y la figura 21, si todos los países asociados a literales se colorean como Falsas (rojo), ninguno de los países n_1, n_5, n_6 o n_2 puede ser coloreado de rojo, ya que colindan con los países correspondientes a las literales y F. Además, dado que en la tierra el país n_5 colinda con n_6, si n_5 es verde, n_6 debe ser azul, y viceversa. Esto implica que el país n_4 debe ser coloreado de rojo, ya que colinda con n_5 (verde/azul) y con n_6 (azul/verde).

Por lo tanto, el país n_3 debe ser coloreado de verde o azul, dado que está conectado a n_4 (rojo). Por lo tanto, los países n_2, n_1 y n_3 solo pueden tener los colores verde o azul. Sin embargo, al tratarse de tres países y solo dos colores disponibles, necesariamente habrá un par de vértices que compartan el mismo color. Como estos tres países están conectados entre sí, se produciría una contradicción, ya que dos países conectados tendrían el mismo color. De este modo, si los países de las tres literales son falsas,

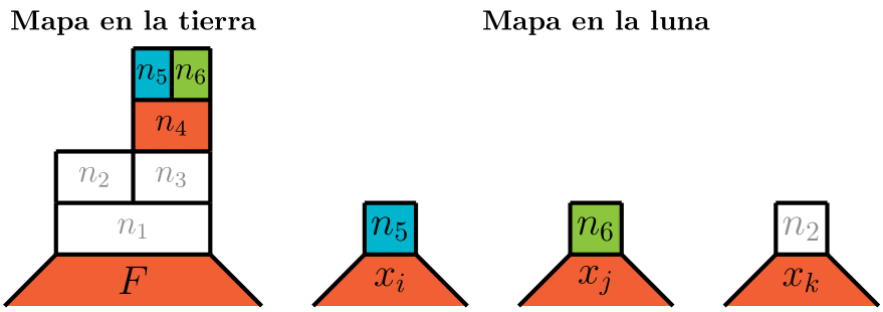


Figura 22: Tercer paso (coloración).

es imposible colorear el gadget. Adicionalmente, es sencillo ver que si alguno es verde, se puede colorear el gadget.

Paso 4: El paso final consiste en intersectar los gadgets, de manera que no colinden con los países ya existentes, es decir, se crean los mapas por cada clausal y se busca empatar esos gadgets. Aunque el resultado es grande, está bien estructurado, como se muestra en la figura 23. De este modo, el mapa resultante es 3-coloreable si y solo si la fórmula booleana se satisface.

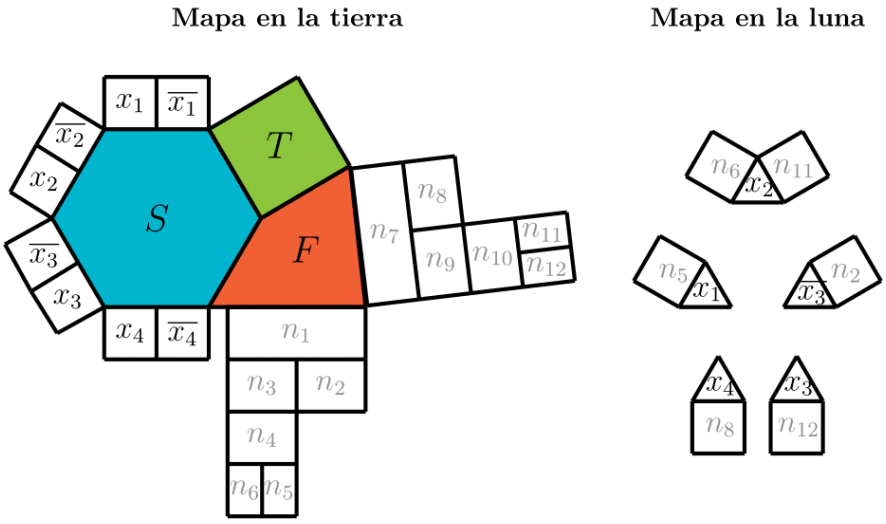


Figura 23: Reducción completa de la fórmula $3SAT$ $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee x_4)$ a una 3-coloración.

En el caso de que la fórmula booleana representada por la figura no se satisfaga, es decir, si x_1, x_2, y $\overline{x_3}$ son coloreadas de rojo, entonces al aplicar el proceso descrito en el paso 3, se llega a la conclusión de que es imposible colorear el mapa.

- **Paso 6:** Para demostrar que esta reducción se realiza en tiempo polinomial, supongamos que la fórmula tiene n literales y m cláusulas. Entonces, el número de países se calcula de la siguiente manera:

  Países en la tierra:

  - 3 países para T, F y S (figura 18).
  - 2 países por cada variable: se requieren $2n$ países extra (figura 19).
  - 6 países por cada cláusula: se requieren $6m$ países extra (figura 21).

  Países en la luna:

  - 1 país por cada literal: se requieren a lo más $2n$ países extra (figura 21).

- 3 países conectados a las literales por cada clausal: se requieren $3m$ países adicionales (figura 21).

De este modo, los mapas generados tienen en total $4n + 9m + 3$ países.

Hasta ahora, hemos demostrado que un problema 3SAT puede reducirse al Problema de la Tierra-Luna de forma general. Dado que 3SAT es un problema NP-completo, la reducción implica que el Problema de la Tierra-Luna también es NP-completo. □

7 Conclusión

Este trabajo ha ofrecido una revisión sobre los problemas de coloración en grafos. Que se profundiza más en la tesis de la cual se deriva este artículo.

Se permitió profundizar en la complejidad computacional de los problemas de decisión, destacando la relevancia de la clasificación de problemas en P, NP, NP-completo y NP-hard. Además, la modelación de la 3-coloración coo un Problema de Satisfacción de Restricciones (CSP) proporcionó un marco útil para abordar estos problemas desde distintos enfoques. Adicionalmente, se presentó un importante resultado: 3SAT $\leq_p$ Grafo 3-coloreable (esto significa que cualquier instancia del problema de 3-satisfacibilidad booleana puede transformarse en tiempo polinomial en una instancia equivalente de un grafo 3-coloreable), acompañado de dos reducciones significativas.

$$\text{Grafo 3-coloreable} \leq_p \text{3SAT}$$

$$\text{Grafo 3-coloreable} \leq_p \text{Grafo planar 3-coloreable}$$

Estas reducciones permiten concluir que la 3-coloración de mapas en la tierra es un problema NP-completo, dado que 3SAT lo es.

Finalmente, se abordó el problema de colorear mapas de la Tierra-Luna, que ilustra la evolución del pensamiento en la teoría de grafos y la necesidad de seguir investigando áreas abiertas para avanzar en nuestra comprensión de la coloración de grafos. Se concluye con la demostración 3SAT $\leq_p$ Problema Tierra-Luna, es decir, que se el problema 3SAT se puede reducir en tiempo polinómico a una instancia del Problema Tierra-Luna, de dicha reducción se puede concluir que este problema último es NP-completo. La demostración se hizo en colaboración con mi asesora Edith Vargas y el profesor Mike Behrisch, hasta la fecha dicha demostración no ha sido encontrada en alguna bibliografía. Se continuó investigando y podemos afirmar que el problema de colorear mapas en la Tierra-Luna es un problema NP-completo para $n = 4$, $n = 5$ y $n = 6$ colores. Las pruebas de estas afirmaciones se encontrarán en un futuro artículo.

Referencias

[1] S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[2] D. L. Boutin, E. Gethner, and T. Sulanke. Thickness-two graphs part one: New nine-critical graphs, permuted layer graphs, and catlin's graphs. *Journal of Graph Theory*, 57(3):198–214, 2008.

[3] E. Demain, J. Ku, and J. Solomon. Lecture 19: Complexity. *MIT OpenCourse*, 2020.

[4] G. Gonthier et al. Formal proof–the four-color theorem. *Notices of the AMS*, 55(11):1382–1393, 2008.

[5] D. Gorrie. Scribe notes: Planar graphs and space complexity. `https://people.eecs.berkeley.edu/~gorrie/cs170/notes/planar.pdf`, 2014. Lecture notes.

[6] J. P. Hutchinson. Coloring ordinary maps, maps of empires, and maps of the moon. *Mathematics Magazine*, 66(4):211–226, 1993.

[7] A. B. Kempe. On the geographical problem of the four colours. *American journal of mathematics*, 2(3):193–200, 1879.

[8] D. Lau. *Function algebras on finite sets: Basic course on many-valued logic and clone theory*. Springer Science & Business Media, 2006.

[9] R. J. McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

[10] G. Ringel. *Map color theorem*, volume 209. Springer Science & Business Media, 2012.

[11] A. Tsiatas. Phase transitions in boolean satisfiability and graph coloring. *Department of Computer Science, Cornell University*, 2008.

[12] D. Zhuk. A proof of the csp dichotomy conjecture. *Journal of the ACM (JACM)*, 67(5):1–78, 2020.

Hybrid Discontinuous Galerkin method for perturbations of the modified Helmholtz equation

Danalie de los Angeles Azofeifa and Miguel Angel Moreles*

Centro de Investigación en Matemáticas

Abstract

The application of the Discontinuous Galerkin Method to elliptic problems usually leads to underdetermined linear systems, and penalization or suitable constraints are necessary. In this work, we address this issue for the modified Helmholtz equation. For this elliptic problem, we propose a hybrid numerical flux in the Discontinuous Galerkin method to introduce unknowns on the edges of the mesh, yielding a well-determined linear system. Performance is tested as a Poisson solver. Additionally, accurate approximations are presented for certain Helmholtz problems in Coastal Ocean Modeling.

Palabras clave: Modified Helmholtz equation; Hybrid Discontinuous Galerkin; Numerical flux.

1 Introduction

The modified Helmholtz equation appears in a common physical model, and its numerical solution remains an active line of research. See the discussion in Yaman & Özdemir [8] and the recent Yaman et al [6]. In the former, an integral equation method is proposed for the numerical solution. In this work, a Galerkin approach is preferred for the numerical solution of perturbations of the modified Helmholtz equation.

The classical Galerkin approach is the continuous Galerkin, Finite Element Method (FEM). A recent and attractive alternative is the Discontinuous Galerkin (DG) method. Unlike FEM, it is locally conservative and H^p-adaptative. On the downside, the application of the Discontinuous Galerkin Method to elliptic problems usually leads to underdetermined linear systems, and penalization or suitable constraints are required. Moreover, Discontinuous Galerkin methods are more expensive because of the need for numerical fluxes at the edges of the mesh elements, thus yielding more coupled unknowns than FEM. See Rivière [7] and Di Pietro & Ern [3] for a thorough exposition.

These shortcomings of the DG method can be addressed in part by its descendant, the Hybrid Discontinuous Galerkin Method (HDG), Bui-Tahn [1]. The numerical flux is *hybridized*, introducing additional unknowns on the edges of the mesh, which reduces the coupling between elements. The global solution is then obtained by solving small and independent linear systems on each element.

For the elliptic problem under study, we use a simple, physically motivated hybrid numerical flux that yields a well-posed linear system. This is guaranteed by the dominant reaction term in the modified Helmholtz equation.

The outline is as follows. In Section 2, we follow the HDG methodology and introduce a first-order system associated with the perturbed modified Helmholtz equations. Then, we delve into the discretization of the first-order system and show that the resulting finite-dimensional problem is well posed. Numerical tests are presented in Section 3. First, we illustrate its performance as a Poisson solver in a preliminary comparison with

*moreles@cimat.mx

a leading method in the literature. Accurate approximations are also shown for some benchmark Helmholtz problems in Coastal Ocean Modeling. In all examples, we highlight approximations on coarse meshes. We close our exposition with conclusions and future work.

2 Numerical solution of the modified Helmholtz equation

In what follows, we build on the theoretical and numerical aspects of the finite element method (FEM) as presented, for instance, in the text [5].

2.1 A first order system for perturbations of the modified Helmholtz equation

For a bounded Lipschitz domain Ω in $\mathbb{R}^d$, let us consider the diffusion-advection-reaction equation for the unknown function u,

$$-\nabla \cdot (\mathbf{D}\nabla u) + \mathbf{v} \cdot \nabla u + cu = b. \tag{1}$$

The diffusion term is uniformly elliptic, that is $\mathbf{D}(x)$ is symmetric and there exist constants $0 < \lambda < \Lambda$ such that for almost all $x \in \Omega$

$$\lambda|\xi|^2 \leq \xi^T \mathbf{D}(x)\xi \leq \Lambda|\xi|^2, \quad \xi \in \mathbb{R}^d. \tag{2}$$

Also, assume that $\mathbf{v} \in (L^\infty(\Omega)))^d$, and c, $b \in L^\infty(\Omega)$.

For $\mathbf{v} = 0$ and $c > 0$, equation (1) is known as the modified Helmholtz equation. To construct a Discontinuous Galerkin approximation, a first step is to construct a *hyperbolic-like* first-order system. A small advection term is allowed; the smallness condition is made precise below.

Define the new variable (flux) $\mathbf{z} = -\mathbf{D}\nabla u$. Since the matrix $\mathbf{D}$ is invertible, $\nabla u = -\mathbf{D}^{-1}\mathbf{z}$, we obtain the following system

$$\begin{aligned} \nabla u + \mathbf{D}^{-1}\mathbf{z} &= \mathbf{0} \\ \nabla \cdot \mathbf{z} - \mathbf{v} \cdot \left(\mathbf{D}^{-1}\mathbf{z}\right) + cu &= b \end{aligned}$$

Let $m = d + 1$, and $\mathbf{e}^i$ be the $i-$th canonical vector. Define

$$\mathcal{K} = \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ -\mathbf{v}^T\mathbf{D}^{-1} & c \end{pmatrix},$$

$$\mathcal{A}_i = \begin{pmatrix} \mathbf{0} & \mathbf{e}^i \\ \left(\mathbf{e}^i\right)^T & 0 \end{pmatrix}, \quad i = 1, 2, \ldots, d,$$

$$\mathbf{u} := \begin{pmatrix} \mathbf{z} \\ u \end{pmatrix}, \quad \mathbf{f} := \begin{pmatrix} \mathbf{0} \\ b \end{pmatrix}.$$

The system becomes

$$\sum_{i=1}^{d} \mathcal{A}_i \partial_i \mathbf{u} + \mathcal{K}\mathbf{u} = \mathbf{f}. \tag{3}$$

2.2 A Hybrid Discontinuous Galerkin Method

Here we introduce the essentials of HDG, for full details see Bui-Thanh [1].

For any matrix $\mathbf{M}$, we denote by $\mathbf{M}_i$, $\mathbf{M}^j$ its row i and column j respectively.

Let us define

$$F_i(\mathbf{u}) := ((\mathcal{A}_1)_i\mathbf{u}, \ldots, (\mathcal{A}_d)_i\mathbf{u}), \quad i = 1, 2, \ldots, m,$$

and

$$F(\mathbf{u}) := \begin{bmatrix} F_1(\mathbf{u}) \\ \vdots \\ F_m(\mathbf{u}) \end{bmatrix}.$$

We apply operators component-wise. For instance, for the divergence operator, we have,

$$\nabla \cdot F(\mathbf{u}) := \begin{bmatrix} \nabla \cdot F_1(\mathbf{u}) \\ \vdots \\ \nabla \cdot F_m(\mathbf{u}) \end{bmatrix}.$$

We can write system (3) in the form

$$\nabla \cdot F(\mathbf{u}) + \mathbf{B}\mathbf{u} = \mathbf{f}, \tag{4}$$

Assume we have a valid element partition of the domain Ω. In DG, functions are approximated locally in each element by polynomials, then coupled with others in adjacent elements by means of a numerical flux. The basic process is as follows.

Let τ be an element in the partition. Compute the $(L^2)^m$ inner product on τ of each side of (4) with a test function $\mathbf{v}$, to obtain

$$(\nabla \cdot F(\mathbf{u}), \mathbf{v})_\tau + (\mathbf{B}\mathbf{u}, \mathbf{v})_\tau = (\mathbf{f}, \mathbf{v})_\tau. \tag{5}$$

Denoting the inner product in th eboundary of τ by $\langle \cdot, \cdot \rangle_{\partial\tau}$ we obtain after integrating by parts,

$$-(F(\mathbf{u}), \nabla\mathbf{v})_\tau + \langle F(\mathbf{u}) \cdot \mathbf{n}, \mathbf{v} \rangle_{\partial\tau} + (B\mathbf{u}, \mathbf{v})_\tau = (\mathbf{f}, \mathbf{v})_\tau. \tag{6}$$

Continuity is not enforced at the boundary of adjacent elements. Therefore, the boundary term $F(\mathbf{u})$ is replaced with a boundary numerical flux $F^*(\mathbf{u}^*)$ where $\mathbf{u}^* \equiv \mathbf{u}^*(\mathbf{u}^-, \mathbf{u}^+)$ solves a Riemann problem with Cauchy data $\mathbf{u}^-$, $\mathbf{u}^+$. As is standard, the $-$ superscript denotes limits from the interior of e, and the $+$ superscript, limits from the exterior. In this context, element τ is denoted by τ^- and the outer normal $\mathbf{n}$ by $\mathbf{n}^-$.

To hybridize the flux, and break the coupling, $\mathbf{u}^*$ is regarded as an extra unknown to be solved on the skeleton (the set of element edges $\mathcal{F}$) of the mesh. Renaming $\mathbf{u}^*$ as $\hat{\mathbf{u}}$ and F^* as $\hat{F}$, the problem is to propose a suitable hybrid numerical flux $\hat{\mathbf{F}}(\mathbf{u})$.

Let us apply this scheme to equation (1) under the following condition

$$c - \frac{1}{2}\mathbf{v} \cdot \left(\mathbf{D}^{-1}\mathbf{v}\right) \geq 0. \tag{7}$$

This condition implies a dominant reaction term, which yields a tamed advection. Consequently, the diffusive flux is assumed to be continuous across any interface. Instead of the Riemann problem solution approach, we set the hybrid flux

$$\hat{\mathbf{F}} \cdot \mathbf{n} = \begin{pmatrix} \hat{u}\mathbf{n} \\ \mathbf{z} \cdot \mathbf{n} + u - \hat{u} \end{pmatrix}. \tag{8}$$

Notice the dependence only in the unknown $\hat{u}$.

For each element τ, the DG local unknown $\mathbf{u}$ and the extra *trace* unknown $\hat{u}$ need to satisfy

$$-(F(\mathbf{u}), \nabla \mathbf{v})_\tau + \langle \hat{F} \cdot \mathbf{n}, \mathbf{v} \rangle_{\partial\tau} + (\mathbf{B}\mathbf{u}, \mathbf{v})_\tau = (\mathbf{f}, \mathbf{v})_\tau, \tag{9}$$

This is complemented by a weak jump condition in the skeleton. Namely, for each edge e

$$([\![\mathbf{z} \cdot \mathbf{n} + u - \hat{u}]\!], w)_e = 0, \tag{10}$$

where $\mathbf{v}$ in (9), w in (10) are polynomial test functions defined on elements τ and edges e respectively.

Here, $[\![\cdot]\!]$ is the jump operator,

$$[\![(\cdot)]\!] = (\cdot)^- + (\cdot)^+.$$

On each element, we are led to solving the equations (9) and (10). Let us show that this finite-dimensional problem is well-posed by showing that under null data, the solution is the trivial one.

Lemma. Assume condition (7) holds. If $f(x) = 0$ and $\hat{u} = 0$ on $\mathcal{F}$, then $u = 0, \mathbf{z} = \mathbf{0}$ in Ω.
Proof.

Let us split the test function in the form $\mathbf{v} = (\mathbf{w}, w)$. The equation (9) using the hybrid flux can be written as

$$\begin{aligned} -(u, \nabla \cdot \mathbf{w})_\tau + (\hat{u}, \mathbf{w} \cdot \mathbf{n})_{\partial\tau} + (\mathbf{D}^{-1}\mathbf{z}, \mathbf{w})_\tau &= 0 & (11)\\ -(\mathbf{z}, \nabla w)_\tau + (\mathbf{z} \cdot \mathbf{n} + u - \hat{u}, w)_{\partial\tau} + (cu - \mathbf{v} \cdot \left(\mathbf{D}^{-1}\mathbf{z}\right), w)_\tau &= (f, w)_\tau & (12) \end{aligned}$$

By integration by parts in (12)

$$(\nabla \cdot \mathbf{z}, w)_\tau + (u, w)_{\partial\tau} + (cu - \mathbf{v} \cdot \left(\mathbf{D}^{-1}\mathbf{z}\right), w)_\tau = 0. \tag{13}$$

Let $\mathbf{w} = \mathbf{z}, w = u$, and add equations (11) and (13) to obtain

$$(\mathbf{D}^{-1}\mathbf{z}, \mathbf{z})_\tau + (u, u)_{\partial\tau} + (cu - \mathbf{v} \cdot \left(\mathbf{D}^{-1}\mathbf{z}\right), u)_\tau = 0$$

.

Notice that in the boundary term, u is the inner limit u^-.

We are led to

$$\begin{aligned} 0 &= (\mathbf{D}^{-1}\mathbf{z}, \mathbf{z})_\tau + (u^-, u^-)_{\partial\tau} + (cu - \mathbf{v} \cdot \left(\mathbf{D}^{-1}\mathbf{z}\right), u)_\tau \\ &\geq \frac{1}{2}(\mathbf{D}^{-1}\mathbf{z}, \mathbf{z})_\tau + (u^-, u^-)_{\partial\tau} + (c - \frac{1}{2}\mathbf{v} \cdot \left(\mathbf{D}^{-1}\mathbf{v}\right), u^2)_\tau \geq 0 \end{aligned}$$

Hence, $u^- = 0$ on $\partial\tau$, and $\mathbf{z} = \mathbf{0}$ in τ. Equation (11) becomes

$$-(u, \nabla \cdot \mathbf{w})_\tau = 0.$$

Integrating by parts

$$-(u^-, \mathbf{w} \cdot \mathbf{n})_{\partial\tau} + (\nabla u, \mathbf{w})_\tau = 0.$$

$$(\nabla u, \mathbf{w})_I = 0.$$

Since $\mathbf{w}$ is arbitrary, $\nabla u = 0$, and consequently $u \equiv 0$ in each element u.

We conclude that $u = 0, \mathbf{z} = \mathbf{0}$ in Ω. ■

Remark. Notice that the result is valid for Poisson problems, $c = 0$, $\mathbf{v} = \mathbf{0}$.

3 Numerical results

Here, we illustrate some numerical results on a variety of problems, where the perturbed modified Helmholtz equation HDG solver serves as the computational engine.

The full code of the numerical implementation is available on GitHub:
https://github.com/Danalie/Hibrid-Discontinuos-Galerkin-DG2

We will test benchmark problems in rectangular geometries and regular meshes for simplicity.

We use the classical element functions on the reference segment $[-1,1]$ for 1D applications. We denote by HDG_p a p-order bilinear approximation. In 2D, HDG_2, HDG_4 refer to basis functions on the reference square $[-1,1]\times[-1,1]$, as products of first-order and second-order one-dimensional 1D basis functions, respectively.

To report the accuracy of the approximation, we use the Root Mean Square Error (RMSE). Sometimes, we normalize by the corresponding norm of the exact solution to obtain the Normalized Root Mean Square Error (NRMSE).

3.1 Poisson problems

For $c=0$ and $\mathbf{v}=0$ in (1) we have the pure diffusion equation

$$-\nabla\cdot(\mathbf{D}\nabla u)=b.$$

To illustrate the HDG method with continuous diffusion numerical flux, as a Poisson solver, we consider $\mathbf{D}$ as the identity matrix.

First a smooth Dirichlet Problem for Poisson's equation on the domain $\Omega=(0,1)\times(0,1)$,

$$\begin{aligned} -\Delta p &= 0,(x,y)\in\Omega,\\ p(x,y) &= 1+x+y+xy,(x,y)\in\partial\Omega. \end{aligned}$$

The exact solution is $p(x,y)=1+x+y+xy$. The numerical solution using HDG_2 on a 10×10 grid yields $RMSE=0$ to machine precision; no further refinement is necessary.

The DG method is H^p-adaptative. Local high order is straightforward, and accurate approximations are possible on coarse meshes. For instance, we use HDG_4 for the Dirichlet problem

$$\begin{aligned} -\Delta p &= -6,\quad (x,y)\in\Omega=(0,1)\times(0,1),\\ p(x,y) &= 1+x^2+2y^2,\quad (x,y)\in\partial\Omega. \end{aligned}$$

On a 10×10 grid the $RMSE$ is again zero to machine precision.

A preliminary comparison with the method in [4]. The latter relies on an internal-penalty numerical flux requiring a penalty parameter chosen heuristically, see (34) therein. Accuracy is shown in an exhaustive list of test problems. The simplest of such is the Dirichlet problem

$$\begin{aligned} -\Delta q &= 2\pi^2\sin(\pi x)\sin(\pi y),\mathbf{x}\in\Omega=(0,1)^2,\\ q &= 0,\mathbf{x}\in\partial\Omega, \end{aligned}$$

with analytic solution is $q_e=\sin(\pi x)\sin(\pi y)$. Our solution is in Table 1.

By inspection of Figure 7 in [4], it is apparent that our results compare favorably. A full comparison is to be carried out.

Elements by dimension	HDG_2	$\mathcal{P}$	HDG_4	$\mathcal{P}$
20	0.0051	1.97	5.71e-5	2.94
30	0.0022	2.07	1.7e-5	2.98
50	0.00079	2.00	3.7e-6	2.98
70	0.00041	1.94	1.34e-6	3.01
100	0.00022	1.74	4.57e-7	3.01
150	8.93e-5	2.22	1.34e-7	3.02

Table 1: Convergence of the two-dimensional Poisson problem using the Root means square error (RMSE). $\mathcal{P}$ is the order of approximation

3.2 Helmholtz equation

We consider two benchmark problems in Coastal Ocean Modeling (COM) that lead to Helmholtz equations. A comparison is made with the Finite Volume (FVCOM) as presented in Chen et al [2]. Therein, FVCOM is applied for the modeling of tidal simulation in a semienclosed basin with tidal forcing at the open boundary under near-resonance conditions.

Here we apply HDG to illustrate the accurate simulation of the troublesome near resonance case. We remark that for the Helmholtz equation, condition (7) is not valid. Nevertheless, the numerical results are highly satisfactory.

3.2.1 A Rectangular Channel

Consider a fluid layer of uniform density that propagates along a channel aligned with the $x-$direction. More precisely, a semienclosed narrow channel with length L and variable depth $H(x)$ and a closed boundary at $x = L_1$ and an open boundary at $x = L$.

Neglecting Coriolis force and advection of momentum, the governing equations modeling tidal waves propagation in the semienclosed channel (see Figure 1) are given as

$$\frac{\partial u}{\partial t} + g\frac{\partial \zeta}{\partial x} = 0; \quad \frac{\partial \zeta}{\partial t} + g\frac{\partial uH}{\partial x} = 0; \quad (x,t) \in (L_1, L) \times (0, T).$$

Here, $H(x)$ is the total water depth, g is acceleration due to gravity, u is speed in the $x-$direction, and ζ is sea-level elevation.

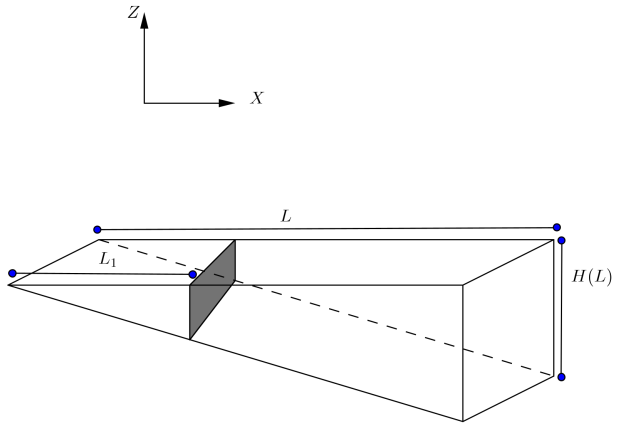


Figure 1: *Configuration of the semienclosed channel.*

Assuming harmonic solutions,

$$\zeta = \zeta_0(x)e^{-i\sigma t}, \quad u = u_0(x)e^{-i\sigma t},$$

we obtain the one dimensional Helmholtz problem for ζ_0

$$(H\zeta_0')' + \frac{\sigma^2}{g}\zeta_0 = 0. \tag{14}$$

We specify a periodic tidal forcing with amplitude A at the mouth of the channel,

$$\zeta_0(L) = A,$$

and a no-flux boundary condition at the wall,

$$H(L_1)\zeta_0'(L_1) = 0.$$

Let the channel depth decrease linearly toward the end of the channel, so that $H(x)$ can be written as

$$H(x) = \frac{xH(L)}{L}.$$

It is readily seen that (14) is a Bessel's equation. With the given boundary conditions, the analytic solution is given by

$$\zeta_0(x) = A\frac{Y_0'(2k\sqrt{L_1})J_0(2k\sqrt{x}) - J_0'(2k\sqrt{L_1})Y_0(2k\sqrt{x})}{Y_0'(2k\sqrt{L_1})J_0(2k\sqrt{L}) - J_0'(2k\sqrt{L_1})Y_0(2k\sqrt{L})}$$

where

$$k := \frac{\sigma\sqrt{L}}{\sqrt{gH(L)}},$$

J_0, Y_0 are the Bessel's functions of degree zero and one respectively.

To compare with the HDG approximation, we solve the system,

$$\begin{aligned} \zeta_0' + H^{-1}z &= 0 \\ z' - \frac{\sigma^2}{g}\zeta_0 &= 0. \end{aligned}$$

Here we have defined $z = -H\zeta_0'$ in (14).

The following parameters are considered for a channel very close to resonance.

$$L = 300km,\ L_1 = 10km,\ H(L) = 20.1m,\ \sigma = \frac{2\pi}{12.42 \cdot 3600s},\ A = 1cm.$$

The HDG method is applied using nodal polynomials of degree 2 (HDG_2). It is compared with the FV method and the analytic solution. For consistency with the FV method, we consider the approximation at the middle point x_i of the element $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. As illustrated in Table 4, a second-order approximation with HDG in a coarse resolution is of greater quality than FV.

The analytic solution describes a standing wave with a node point near the closed side of the channel. As expected, the reproduction of these features by HDG is accurate. See Figure 2

Remark. As pointed out in Chen et al [2], regardless of the numerical method, a proper selection of horizontal resolution recover accurately this tidal resonance problem. We argue that the accuracy attained by HDG in coarse meshes yields a better choice.

Elements	NRMSE(HDG_2)	NRMSE(FV)
10	0.180784	22.3052
20	0.0748017	0.585413
40	**0.0235224**	0.553751
80	0.00684013	0.43336
160	0.00187985	0.297118
320	0.000495726	0.181964
640	0.000127432	0.102475
1280	0.0000324	**0.0546902**

Table 2: Relative root mean square error for HDG_2 and FV.

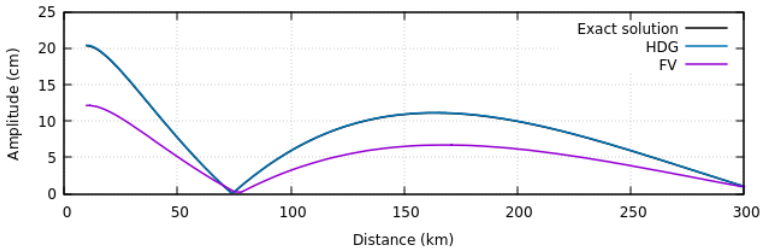


Figure 2: Solution for the rectangular channel near resonance case, with HDG nodal with polynomials of degree 2 and 80 elements.

3.2.2 A Sector Channel

Now consider a flat bottom channel in the form of a semicircular section, which in polar coordinates is defined from 0 to L in the radial direction and from $-\alpha/2$ to $\alpha/2$ in the angular direction.

The semicircular line of radius L corresponds to an open border, while along the semicircular line of radius L_1 and the two sides, they are closed, (Figure 3).

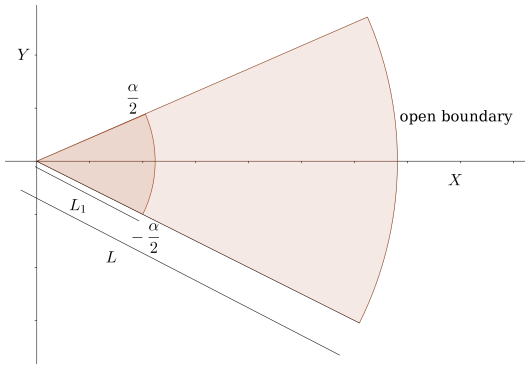


Figure 3: Semienclosed sector channel in the polar region $L_1 \leq r \leq L$, $-\frac{\alpha}{2} \leq \theta \leq \frac{\alpha}{2}$. The sector is open at $r = L$ and closed elsewhere.

The following equations govern the nonrotating tidal oscillation,

$$\frac{\partial V_r}{\partial t} = -g\frac{\partial \eta}{\partial r}, \tag{15}$$

$$\frac{\partial V_\theta}{\partial t} = -g\frac{\partial \eta}{r\partial \theta}, \tag{16}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial r V_r H_0}{r\partial r} + \frac{\partial V_0 H_0}{r\partial \theta} = 0. \tag{17}$$

H_0 is the constant water depth, V_r, V_θ are the radial and angular r, θ velocity components, and η is the free surface water elevation.

Assuming a harmonic solution,

$$\eta = Re(\eta_0(r,\theta)e^{-it(\omega t - \frac{\pi}{2})}),$$

we can reduce the equations (15-17) to an elliptic equation

$$\frac{\partial^2 \eta_0}{\partial r^2} + \frac{1}{r}\frac{\partial \eta_0}{\partial r} + \frac{1}{r^2}\frac{\partial^2 \eta_0}{\partial \theta^2} + \frac{\omega^2}{gH_0}\eta_0 = 0. \tag{18}$$

The physical boundary conditions are as follows

1. At the open mouth of the channel, a harmonic tidal forcing is assumed,

$$\eta_0(r,\theta) = \overline{A}\cos\left(m\pi\frac{\theta + \frac{\alpha}{2}}{\alpha}\right), \quad \{L\} \times \left(-\frac{\alpha}{2}, \frac{\alpha}{2}\right),$$

2. On the solid walls, null flux is prescribed,

$$-K\nabla\eta_0(r,\theta) = 0, \quad \{L_1\} \times \left(\frac{-\alpha}{2}, \frac{\alpha}{2}\right) \bigcup (L_1, L) \times \left\{-\frac{\alpha}{2}\right\} \bigcup (L_1, L) \times \left\{\frac{\alpha}{2}\right\}.$$

The analytic solution of this boundary-value problem is:

$$\eta_0(r,\theta) = \overline{A}\frac{Y_v'(L_1\kappa)J_v(r\kappa) - J_v'(L_1\kappa)Y_v(r\kappa)}{Y_v'(L_1\kappa)J_v(L\kappa) - J_v'(L_1\kappa)Y_v(L\kappa)}\cos\left(\frac{m\pi}{\alpha}(\theta + \frac{\alpha}{2})\right)$$

where

$$v = \frac{m\pi}{\alpha}, \kappa = \frac{\omega}{\sqrt{gH_0}},$$

J_v, Y_v are the v th-order Bessel function of the first and second types.

Let us show the HDG solution in the rectangular domain $(L_1, L] \times (-\alpha/2, \alpha/2)$.

Let ∇ denote the gradient with respect to (r, θ) and let

$$K := \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{r^2} \end{pmatrix}.$$

Equation (18) becomes a perturbation of the two-dimensional Helmholtz equation,

$$-\nabla \cdot (K\nabla\eta_0) - (r^{-1}, 0) \cdot \nabla\eta_0 - \frac{\omega^2}{gH_0}\eta_0 = 0. \tag{19}$$

We apply the fourth order (HDG_4) scheme developed above for a near-resonance case. The geometric parameter values are:

$$H_0 = 1m,\ \alpha = \frac{\pi}{4},\ L_1 = 90km,\ m = 1.0,\ \omega = \frac{2\pi}{12.42 \cdot 3600}\frac{1}{s},\ L = 158km, \overline{A} = 1cm.$$

Figure (4) compares the analytic and numerical solutions. The relative mean square error is shown in Table 3.

Remark. Noteworthy, the 10×10 HDG_4 solution is of grater quality that the 40×40 FV solution. In regards to execution time, the former is attained in half a second on a personal laptop. A solution of the same quality would require in the order of minutes with the Finite Volume Method in a much finer mesh.

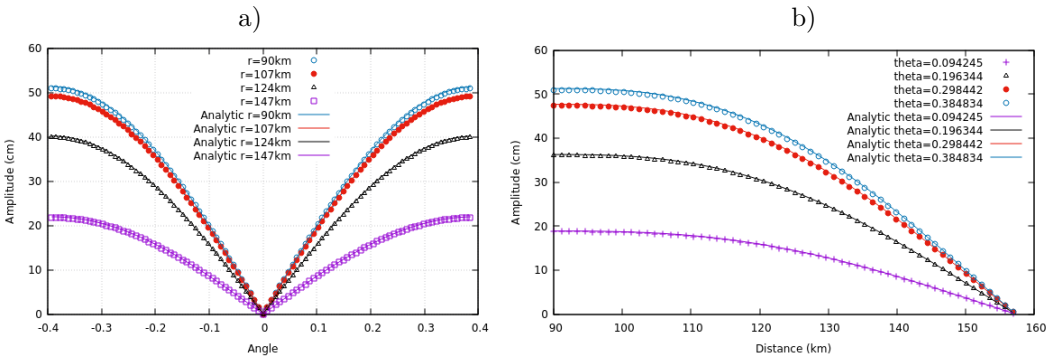


Figure 4: Comparison between the analytic solution to the sector problem with the HDG_4 solution using 30×50 elements. a) Solution for some r values, b) Solution for some θ values.

Elements	NRMSE(HDG_4)	NRMSE(FV)
5×5	0.441213	0.920939
10×10	**0.009879**	0.887757
35×35	0.0026408	0.803039
40×40	0.00115041	**0.794262**

Table 3: Relative root mean square error for HDG_4 and FV.

4 Conclussions

We have introduced a first-order system formulation for a regularly perturbed Modified Helmholtz equation. A discretization is derived by means of the Hybrid Discontinuous Galerkin method with an ad-hoc numerical flux. The well-posedness of the finite-dimensional HDG discrete linear system follows from a dominant reaction condition.

The preliminary comparison with a leading Poisson solver shows promise as a competitive alternative. Research on this is ongoing.

The proposed HDG discrete method is also applied to Helmholtz's problems arising from coastal ccean modeling. It outperforms the Finite Volume method commonly used in this setting.

The application to irregular regions with more general meshes is more elaborate but straightforward. Also, a parallel computing implementation is desired. Both tasks are left for future work.

References

[1] T. Bui-Thanh. From godunov to a unified hybridized discontinuous galerkin framework for partial differential equations. *Journal of Computational Physics*, 295:114–146, 2015.

[2] C. Chen, H. Huang, R. C. Beardsley, H. Liu, Q. Xu, and G. Cowles. A finite volume numerical approach for coastal ocean circulation studies: Comparisons with finite difference models. *Journal of Geophysical Research: Oceans*, 112(C3), 2007.

[3] D. A. Di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69. Springer Science & Business Media, 2011.

[4] N. L. Fischer and H. P. Pfeiffer. Unified discontinuous galerkin scheme for a large class of elliptic equations. *Physical Review D*, 105(2):024034, 2022.

[5] M. S. Gockenbach. *Understanding and implementing the finite element method*. SIAM, 2006.

[6] O. Ivanyshyn Yaman and G. Özdemir. An interior inverse generalized impedance problem for the modified helmholtz equation in two dimensions. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 105(1):e202300711, 2025.

[7] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*. SIAM, 2008.

[8] O. I. Yaman and G. Özdemir. Numerical solution of a generalized boundary value problem for the modified helmholtz equation in two dimensions. *Mathematics and Computers in Simulation*, 190:181–191, 2021.

A simple overview of least squares

Reymundo Itzá Balam [*1,2], L. Héctor Juárez Valencia [†3], and Miguel Uh Zapata [‡1,2]

[1]Secretaria de Ciencia, Humanidades, Tecnología e Innovación, Mexico
[2]Centro de Investigación en Matemáticas A.C, Unidad Mérida, Mexico
[3]Departamento de Matemáticas, UAM–Iztapalapa, México

Abstract

In this work we aim to give an overview of least squares for curve fitting. The idea is to illustrate, for a broad audience, the mathematical foundations and practical methods used to solve this simple problem. We will consider four methods: the normal equations method, the QR factorization, the singular value decomposition (SVD), as well as a new approach based on neural networks. The last approach is not as common as the others, but it is very interesting because, in modern days, it has become a very important tool in many branches of modern knowledge, like data science (DS), machine learning (ML) and artificial intelligence (AI).

Palabras clave: Least squares; normal equations; QR; SVD; neural network.

1 Introduction

In mathematics, the term '*least squares*' refers to an approach for "solving" overdetermined linear or nonlinear systems of equations. A common problem in science is to fit a model to noisy measurements or observations. Instead of solving the equations exactly, which in many problems is not possible, we seek only to minimize the sum of the squares of the residuals.

The algebraic procedure of the method of least squares was first published by Legendre in 1805 [16]. It was justified as a statistical procedure by Gauss in 1809 [8], where he claimed to have discovered the method of least squares in 1795 [4]. Robert Adrian had already published a work in 1808, according to [18]. After Gauss, the method of least squares quickly became the standard procedure for analysis of astronomical and geodetic data. There are several good accounts of the history of the invention of least squares and the dispute between Gauss and Legendre, as shown in [4] and references therein. Gauss gave the method a theoretical basis in two memoirs [9], where he proves the optimality of the least squares estimate without any assumptions that the random variables follow a particular distribution. In an article by Yves [23] there is a survey of the history, development, and applications of least squares, including ordinary, constrained, weighted, and total least squares, where he includes information about fitting curves and surfaces from ancient civilizations, with applications to astronomy and geodesy.

The basic modern numerical methods for solving linear least squares problems were developed in the late 1960s. The QR decomposition by Householder transformations was developed by Golub and published in 1965 [5]. The implicit QR algorithm for computing the singular value decomposition (SVD) was developed by Kahan, Golub, and Wilkinson, and the final algorithm was published in 1970 [12]. Both fundamental

*reymundo.itza@cimat.mx
†hect@xanum.uam.mx
‡angeluh@cimat.mx

matrix decompositions have since been developed and generalized to a high level of sophistication. Since then great progress has been made in methods for generalized and modified least squares problems in direct, and iterative methods for large sparse problems. Methods for total least squares problems, which allow errors also in the system matrix, have been systematically developed.

In this work we aim to give a simple overview of least squares for curve fitting. The idea is to illustrate, for a broad audience, the mathematical foundations and practical methods to solve this simple problem. Particularly, we will consider four methods: the normal equations method, the QR approach, the singular value decomposition (SVD), as well as a more recent approach based on neural networks. The last one has not been used as frequently as the classical ones, but it is very interesting because in modern days it has become a very important tool in many fields of modern knowledge, like data science (DS), machine learning (ML) and artificial intelligence (AI).

2 Linear least squares for curve fitting and the normal equations

There are many problems in applications that can be addressed using the least squares approach. A common source of least squares problems is curve fitting. This is the one of the simplest least squares problems, but still it is a very fundamental problem, which contains all important ingredients of commonly ill posed problems and, even worse, they may be ill conditioned and difficult to compute with good precision using finite (inexact) arithmetic in modern computer devices. We start with the linear least squares problem.

Let's assume that we have m noisy experimental observations (points)

$$(t_1, \hat{y}_1), (t_2, \hat{y}_2), \ldots, (t_m, \hat{y}_m),$$

which relate two real quantities, as shown in figure 2. We want to fit a curve, represented by a real scalar function $y(t)$, to the given data. A linear model for the unknown curve can be represented as a linear combination of given (known) base functions ϕ_1, ϕ_1,...,ϕ_n:

$$y(t) = c_1\phi_1(t) + c_2\phi_2(t) + \ldots + c_n\phi_n(t), \tag{1}$$

where $c_1, c_2, \ldots . c_n$ are unknown coefficients. A first naive approach to compute those coefficients is assuming that $\hat{y}_i = y(t_i)$ for each $i = 1, 2, \ldots, m$. This assumption yields the linear system $\hat{y}_i = c_1\phi_1(t_i) + c_2\phi_2(_i) + \ldots + c_n\phi_n(t_i)$, $i = 1, 2, \ldots, m$, which can be represented as $A\,\mathbf{x} = \mathbf{b}$, where

$$A = \begin{bmatrix} \phi_1(t_1) & \phi_2(t_1) & \ldots & \phi_n(t_1) \\ \phi_1(t_2) & \phi_2(t_2) & \ldots & \phi_n(t_2) \\ \vdots & & \ddots & \vdots \\ \phi_1(t_m) & \phi_2(t_m) & \ldots & \phi_n(t_m) \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix},$$

Depending on the problem or application, the base functions ϕ_j, $1 \leq j \leq n$, may be polynomials $\phi_j(t) = t^{j-1}$, exponentials $\phi_j(t) = e^{\lambda_j\, t}$, log-linear $\phi = K\, e^{\lambda\, t}$, among many others.

There are some drawbacks and difficulties with the previous approach: vector $\mathbf{b}$ must belong to the column space of A, denoted by $\mathrm{Col}(A)$, in order to get solution(s) of the linear system. The rank of A is $r = \dim \mathrm{Col}(A)$, and plays a important role. When $m > r$, most likely $\mathbf{b} \notin \mathrm{Col}(A)$, and the system has no solution; if $m < r$, the undetermined linear system has infinite many solutions; finally, when $m = r$, if the system has a solution, the computed curve produces undesirable oscillations, specially near the far right and left points, a well known phenomenon in approximation theory and numerical analysis, known as the Runge's phenomenon [27], demonstrating that high degree interpolation does not always produce better accuracy. The least squares approach considers the residuals, which are the differences between the observations and the model values:

$$r_i = \hat{y}_i - \sum_{j=1}^{n} c_j\, \phi_j(t_i) \quad \text{or} \quad \mathbf{r} = \mathbf{b} - A\,\mathbf{x}.$$

Ordinary least squares to find the best fitting curve $y(t)$, consists in finding $\mathbf{x}$ that minimizes the sum of squared residuals

$$\|\mathbf{r}\|^2 = \sum_{j=1}^{n} r_i^2 = \|\mathbf{b} - A\,\mathbf{x}\|^2.$$

The least squares criterion has important statistical interpretations, since the residual r_i in

$$\hat{y}_i = y(t_i) + r_i,$$

may be considered as a measurement error with a given probabilistic distribution. In fact, least squares produces what is known as the maximum-likelihood estimate of the parameter estimation of the given distribution. Even if the probabilistic assumptions are not satisfied, years of experience have shown that least squares produces useful results.

2.1 The normal equations

The quadratic function $f(\mathbf{x}) = \|\mathbf{b} - A\,\mathbf{x}\|^2 = \mathbf{x}^T A^T A\,\mathbf{x} - 2\mathbf{x}^T A^T \mathbf{b} + \mathbf{b}^T \mathbf{b}$ has gradient and Hessian given by $\nabla f(x) = 2\,A^T A\,\mathbf{x} - 2\,A^T \mathbf{b}$ and $H_f(\mathbf{x}) = 2\,A^T A$, respectively. Assuming that the design matrix A has full rank, then the Hessian is positive definite, thus invertible, because $A^T A$ is an $n \times n$ symmetric matrix, and positive definite since $\mathbf{x}^T A^T A\,\mathbf{x} = ||A\,\mathbf{x}||^2 > 0$ when $\mathbf{x} \neq \mathbf{0}$. Therefore, its minimum $\widehat{\mathbf{x}}$ is the unique solution of the so called **normal equations**:

$$A^T A\,\mathbf{x} = A^T \mathbf{b}, \quad \text{i.e.} \quad \widehat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}, \tag{2}$$

and the best fitting curve, of the form (1), is obtained with the coefficients $\widehat{\mathbf{x}} = [\widehat{c}_1, \ldots, \widehat{c}_n]^T$. The linear system (2) can be solved computationally using the Cholesky factorization or conjugate gradient iterations (for large scale problems).

Example 2.1. The best fitting polynomial of degree $n-1$, say $y(t) = c_1 + c_2 t + \cdots + c_n t^{n-1}$, to a set of m data points $(t_1, \hat{y}_1)$, $(t_2, \hat{y}_2), \ldots,$ $(t_m, \hat{y}_m)$ is obtained solving the normal equations (2), where the design matrix is the Vandermonde matrix

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{bmatrix}.$$

A sufficient condition for A to be full rank is that $t_1, \ldots, t_m$ be all different, which may be proved using mathematical induction.

Remark 2.2. Rank deficient least squares problems, where the design matrix A has linearly dependent columns, can be solved with specialized methods, like truncated singular value decomposition (SVD), regularization methods, QR decomposition with pivoting, and data filtering, among others. These difficulties are studied and understood more clearly when we start from basic principles. So, in order to keep the discussion easy we first consider the simplest case, where matrix A is full rank, although it may be very ill-conditioned or near singular.

2.2 An interpretation with orthogonal projections

The least squares solution (2) satisfies

$$A\,\widehat{\mathbf{x}} = P_{_A}\,\mathbf{b}, \tag{3}$$

where the $m \times m$ square matrix $P_{_A} = A(A^T A)^{-1} A^T$ defines an orthogonal projection, since

- $P_{_A}^2 = P_{_A}$. It projects $\mathbb{R}^m$ onto $\mathrm{Col}(A)$.
- $P_{_A}^T = P_{_A}$ and $P_{_A}\,\mathbf{b} \;\perp\; (\mathbf{b} - P_{_A}\,\mathbf{b})$, with $\mathbf{b} - P_{_A}\,\mathbf{b}$ the residual with minimum norm.

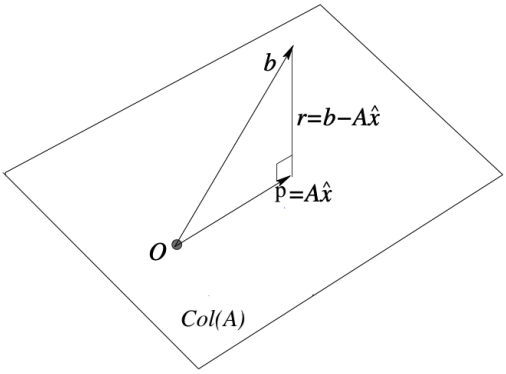


Figure 1: Col(A) is orthogonal to the minimum residual $\widehat{\mathbf{r}} = \mathbf{b} - A\,\widehat{\mathbf{x}} = \mathbf{b} - P_{_A}\,\mathbf{b}$.

Therefore, the vector $\mathbf{p} = P_{_A}\,\mathbf{b}$ is the orthogonal projection of $\mathbf{b}$ onto the column space of A, as illustrated in Figure 1. Additionally, relation (3) defines a well posed problem (a consistent linear problem), with unique solution, since A is full rank. This unique solution is the least squares solution obtained from the normal equations.

2.3 Instability of the normal equations method

The normal equations approach is a very simple procedure to solve the linear least squares problem. It is the most used approach in the scientific and engineering community, and very popular in statistical software. However, it must be used with precaution, specially when the design matrix A is ill-conditioned (or it is rank deficient) and finite precision arithmetic, in digital conventional devices, is employed. In order to understand this phenomenon, it is convenient to show an example and then discuss the results.

Example 2.3. The National Institute of Standards and Technology (*NIST*) is a branch of the U.S. Department of Commerce responsible for establishing national and international standards. *NIST* maintains reference data sets for use in the calibration and certification of statistical software. On its website [1] we can find the *Filip* data set, which consists of 82 observations of a variable y for different t values. The aim is to model this data set using a 10th-degree polynomial. This is part of exercise 5.10 in Cleve Molers' book [20].

For this problem we have $m = 82$ data points $(t_i, \hat{y}_i)$, and we want to compute $n = 11$ coefficients c_j for the 10th-degree polynomial. The $m \times n$ design matrix A has coefficients $a_{ij} = t_i^{j-1}$. In order to given an idea of the complexity of this matrix, we observe that its minimum coefficient is 1 and its maximum coefficient is a bit greater than 2.7×10^9, while its condition number is $\kappa(A) \approx \mathcal{O}(10^{15})$. The matrix of the normal equations, $A^T A$, is a much smaller matrix of size $n \times n$, but *more singular*, since its minimum and maximum coefficients (in absolute value) are close to 82 and 5.1×10^{19}, respectively, with a very high condition number $\kappa(A^T A) \approx \mathcal{O}(10^{30})$. The matrix of the normal equations is highly ill-conditioned in this case because there are some clusters of data points very close to each other with almost identical t_i values.

The computed coefficients $\widehat{c}_j$ using the normal equations are shown in Table 1, along with the certified values provided by *NIST*. The *NIST* certified values were found solving the normal equations, but with multiple precision of 500 digits (which represents an idealization of what would be achieved if the calculations were made without rounding error). Our calculated values differ significantly from those of *NIST*, even in the sign, the relative difference $\|\widehat{\mathbf{c}} - \mathbf{c}_{nist}\|/\|\mathbf{c}_{nist}\|$ is about 118%. This dramatic difference is mainly because we are using finite arithmetic with 16-digit standard *IEEE double precision*, and solving the normal equations with the Cholesky factorization yields a relative error amplified proportionately to the product of the condition number times the machine epsilon. The computed residual keeps reasonable, though. Figure 2 shows *Filip* data along with the certified curve and our computed curve. The difference is most visible at the extremes, where our computed curve shows some pronounced oscillations.

Polynomial coefficients	*NIST* ($\times 10^3$)	Normal equations ($\times 10^2$)
$\widehat{c}_1$	-1.467489614229800	3.397167285217155
$\widehat{c}_2$	-2.772179591933420	5.276500833542165
$\widehat{c}_3$	-2.316371081608930	3.545138197108058
$\widehat{c}_4$	-1.127973940983720	1.345510235823048
$\widehat{c}_5$	-0.354478233703349	0.316966659871258
$\widehat{c}_6$	-0.075124201739376	0.047864845714924
$\widehat{c}_7$	-0.010875318035534	0.004604461850533
$\widehat{c}_8$	-0.001062214985889	0.000269285797556
$\widehat{c}_9$	-0.000067019115459	0.000008526963608
$\widehat{c}_{10}$	-0.000002467810783	0.000000107514812
$\widehat{c}_{11}$	-0.000000040296253	0.000000000044407
Relative difference	0	118%
Norm of residual	0.028400823094900	0.041260859317660

Table 1: Comparison of numerical results with the *NIST*'s certified values.

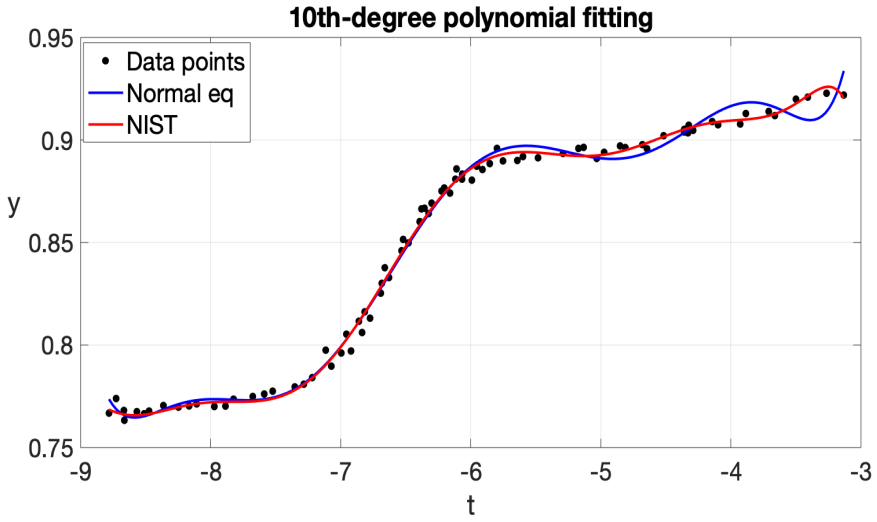


Figure 2: Computed polynomial curve along with *NIST*'s certified one.

3 Orthogonal projections and the QR factorization

The previous numerical results for solving a least square problem have shown instability for the normal equations approach, when the design matrix is ill-conditioned. However the normal equations approach usually yields good results when the problem is of moderate size as well as well-conditioned. For the cases where the design matrix is ill-conditioned the QR factorization method is an excellent alternative. The SVD factorization is convenient when the design matrix is rank deficient, as will be discussed below.

3.1 The QR factorization

We begin with the following theorem in reference [27].

Theorem 3.1. *Each $A \in \mathbb{C}^{m\times n}(m \geq n)$ of full rank has a unique reduced QR factorization $A = \widehat{Q}\widehat{R}$ with $r_{jj} > 0$.*

For simplicity we keep the discussion for the case $A \in \mathbb{R}^{m\times n}$. In this case $\widehat{Q}$ is the same size than A and $\widehat{R} \in \mathbb{R}^{n\times n}$ is upper triangular. Actually this factorization is a matrix version of the Gram-Schmidt

orthogonalization algorithm. More precisely, let $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \in \mathbb{R}^m$ be the linear independent column vectors of A

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ | & | & & | \end{bmatrix},$$

then the orthonormal vectors $\mathbf{q}_1$, $\mathbf{q}_2$, ... , $\mathbf{q}_n$ en $\mathbb{R}^m$ obtained from the Gram-Schmidt orthogonalization gives the following matrix

$$\widehat{Q} = \begin{bmatrix} | & | & & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \\ | & | & & | \end{bmatrix},$$

which has the same column space that A. These vectors are constructed sequentially, starting with $\mathbf{q}_1 = \mathbf{a}_1/\|\mathbf{a}_1\|$, and they satisfy

$$\mathbf{q}_j = \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|_2} \quad \text{con} \quad \mathbf{v}_j = \mathbf{a}_j - (\mathbf{q}_1^T\mathbf{a}_j)\mathbf{q}_1 - \ldots - (\mathbf{q}_{j-1}^T\mathbf{a}_j)\,\mathbf{q}_{j-1} = \mathbf{a}_j - \sum_{i=1}^{j-1}(\mathbf{q}_i^T\mathbf{a}_j)\,\mathbf{q}_i \quad 1 \le j \le n.$$

Using the notation $r_{ij} \equiv \mathbf{q}_i^T\mathbf{a}_j$ for $i > j$ and $r_{jj} = \|\mathbf{v}_j\|_2$, we obtain

$$\begin{array}{rcl} \mathbf{q}_1 &=& \dfrac{\mathbf{a}_1}{r_{11}}\,, \\ \mathbf{q}_2 &=& \dfrac{\mathbf{a}_2 - r_{12}\,\mathbf{q}_1}{r_{22}}\,, \\ &\vdots& \\ \mathbf{q}_n &=& \dfrac{\mathbf{a}_n - \sum_{i=1}^{n-1} r_{in}\,\mathbf{q}_i}{r_{nn}}\,. \end{array} \qquad \text{and} \qquad \begin{array}{rcl} \mathbf{a}_1 &=& r_{11}\,\mathbf{q}_1\,, \\ \mathbf{a}_2 &=& r_{12}\,\mathbf{q}_1 + r_{22}\,\mathbf{q}_2\,, \\ &\vdots& \\ \mathbf{a}_n &=& r_{1n}\,\mathbf{q}_1 + r_{2n}\,\mathbf{q}_2 + \cdots + r_{nn}\,\mathbf{q}_n\,. \end{array}$$

This set of equations leads to the so called *reduced QR factorization*:

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ 0 & r_{22} & \ldots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & r_{nn} \end{bmatrix} = \widehat{Q}\,\widehat{R}.$$

This factorization allows another way to solve the overdetermined system $A\mathbf{x} = \mathbf{b}$, that arise in linear least square problems. The key property is that $\widehat{Q}^T\widehat{Q} = I_n$, where I_n is the identity matrix of size $n \times n$. Then

$$\widehat{Q}\widehat{R}\,\mathbf{x} = \mathbf{b} \quad \text{is equivalent to} \quad \widehat{R}\,\mathbf{x} = \widehat{Q}^T\mathbf{b}, \tag{4}$$

and this triangular system is solved easily using backward substitution. Furthermore, the obtained solution is the least squares solution, since the following set of relations are equivalent

$$\widehat{R}\mathbf{x} = \widehat{Q}^T\mathbf{b}, \quad \widehat{Q}\widehat{R}\mathbf{x} = \widehat{Q}\widehat{Q}^T\mathbf{b}, \quad A\mathbf{x} = P_{\widehat{Q}}\mathbf{b}, \quad A\mathbf{x} = P_{A}\mathbf{b},$$

where the projection matrices satisfy $P_{\widehat{Q}}\mathbf{b} = P_{A}\mathbf{b}$ because the column space of A is equal to the column space of $\widehat{Q}$.

A complete QR factorization of A goes further by adding $m-n$ orthonormal columns to $\widehat{Q}$, and adding $m-n$ rows of zeros to $\widehat{R}$, obtaining an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ and an upper triangular matrix $R \in \mathbb{R}^{m\times n}$ as shown in Figure 3. In the complete factorization the additional columns $\mathbf{q}_j$, $j = n+1, \ldots, m$, are orthogonal to the column space of A. Of course, the matrix Q is an orthogonal matrix, since $Q^TQ = I_m$, so $Q^{-1} = Q^T$.

Theorem 3.2. *Any matrix $A \in \mathbb{R}^{m\times n}$ $(m \ge n)$ has a complete factorization QR, given by $A = QR$ with $Q \in \mathbb{R}^{m\times m}$ an orthogonal matrix and $R \in \mathbb{R}^{m\times n}$ an upper triangular matrix.*

Warning. The Gram-Schmidt algorithm is numerically unstable (sensitive to rounding errors). Stabilization methods can be used by changing the order in which the operations are performed. Fortunately, there is an stable algorithm to compute the QR factorization which relies on Householder reflections.

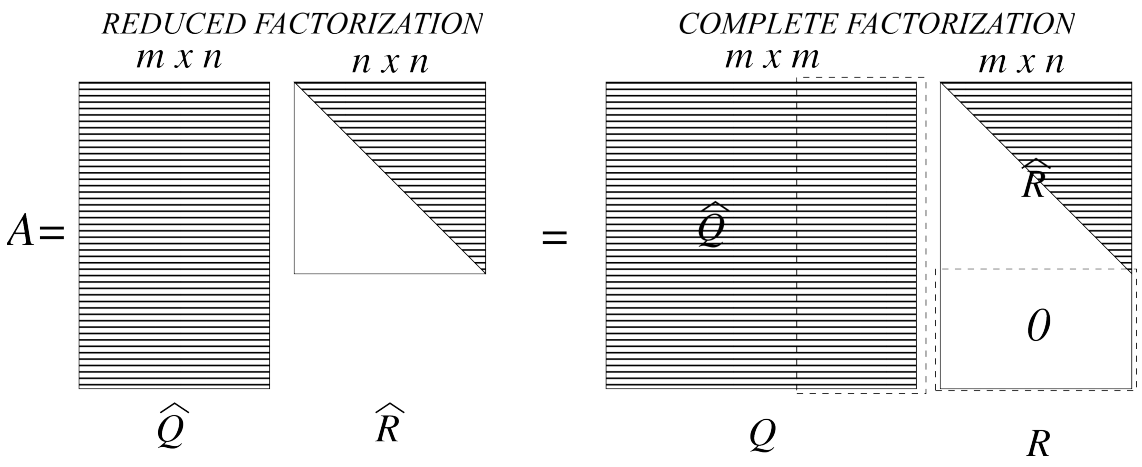


Figure 3: The reduced and complete QR factorizations of A

3.2 Householder reflections

A *Householder reflection* is a linear transformation with matrix H, which is constructed from a given fixed vector $\mathbf{x}$ in a Euclidean space $\mathbb{R}^p$ ($p \geq 2$), seeking its reflected vector to be $H\,\mathbf{x} = ||\mathbf{x}||\,\mathbf{e}_1 = (||\mathbf{x}||, 0, \ldots, 0)^T$, as shown in Figure 4. This reflection reflects on a hyperplane H^+ with normal unitary vector $\mathbf{u}$, and is given

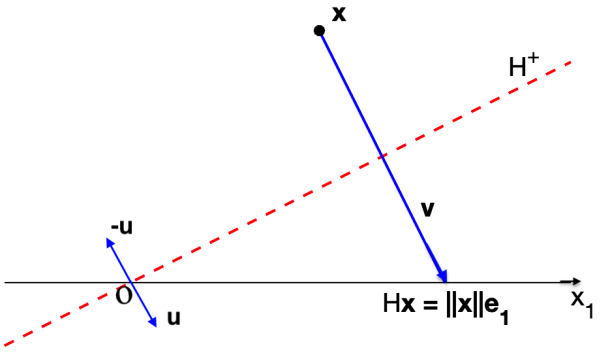


Figure 4: Householder reflection with vector $\mathbf{v} = ||\mathbf{x}||\,\mathbf{e}_1 - \mathbf{x}$.

by

$$H = I - 2\mathbf{u}\mathbf{u}^T, \quad \text{with} \quad \mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \quad \mathbf{v} = \|\mathbf{x}\|\,\mathbf{e}_1 - \mathbf{x}, \tag{5}$$

where the outer (or external) product $\mathbf{u}\mathbf{u}^T$ gives rise to a rank one symmetric matrix. We emphasize that, given the vector $\mathbf{x}$, the projection H performs the following transformation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \quad \longmapsto \quad H\,\mathbf{x} = \begin{bmatrix} ||\mathbf{x}|| \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|\mathbf{x}\|\,\mathbf{e}_1.$$

This matrix is a symmetric orthogonal matrix, i.e. it satisfies $H^T H = I_p$, $H^T = H$.

Matrix A is transformed into an upper triangular matrix R by successively applying Householder matrix transformations H_k

$$H_n \cdots H_2\, H_1\, A = R.$$

Each H_k matrix is chosen to introduce zeros below the diagonal in the k-th column. For example, for a matrix A of $m \times n = 5 \times 3$, the H_k operations are applied as shown below:

$$\overbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \end{bmatrix}}^{A}_{A} \rightarrow \overbrace{\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} \\ 0 & a_{42}^{(1)} & a_{43}^{(1)} \\ 0 & a_{52}^{(1)} & a_{53}^{(1)} \end{bmatrix}}^{A^{(1)}}_{H_1 A} \rightarrow \overbrace{\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(2)} \\ 0 & 0 & a_{43}^{(2)} \\ 0 & 0 & a_{5}^{(2)} \end{bmatrix}}^{A^{(2)}}_{H_2 H_1 A} \rightarrow \overbrace{\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}^{R}_{H_3 H_2 H_1 A},$$

where each $H_k \in \mathbb{R}^{m \times m}$ is of the form

$$H_k = \begin{bmatrix} I_k & \mathbb{O}^T \\ \mathbb{O} & H \end{bmatrix},$$

which is a symmetric orthogonal matrix (see [15] for more details), I_k is the identity matrix of size $k \times k$, and $\mathbb{O}$ is the zero matrix of size $(m-k) \times k$. Actually, for any vector $\mathbf{x}$ there are two Householder refletions, as shown in Figure 5, and each Householder matrix H is constructed with the election $\mathbf{v} = \text{sign}(x_1)\|\mathbf{x}\|\,\mathbf{e}_1 + \mathbf{x}$. It is evident that this election allows $\|\mathbf{v}\|$ to never be smaller than $\|\mathbf{x}\|$, avoiding cancellation by subtraction when dividing by $\|\mathbf{v}\|$ to find $\mathbf{u}$ in (5), thus ensuring stability of the method.

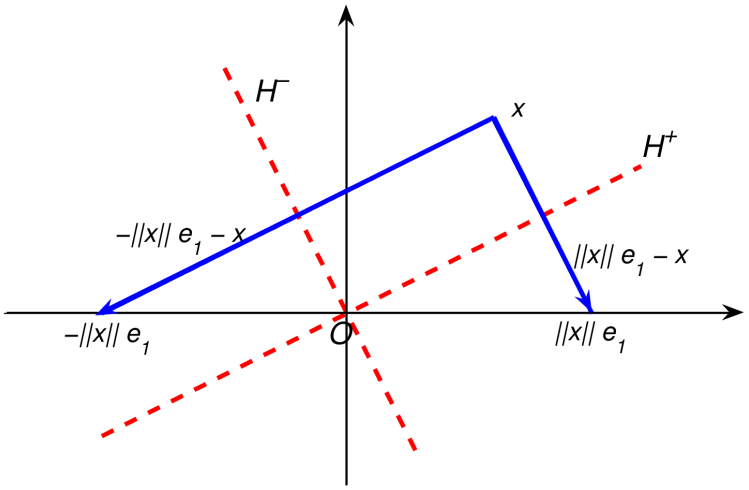


Figure 5: Two Huoseholder reflections, constructed from $\mathbf{x}$.

The above process is called *Householder triangularization* [14], and currently it is the most widely used method for finding the QR factorization. There are two procedures to construct the reflection matrices: *Givens rotations* and *Householder reflections*. Here we have described only Householder reflections. For further insight, we refer the reader to [11], [15] and [27].

We may compute the factorization $A = \widehat{Q}\widehat{R}$, with $\widehat{Q} = (H_n \cdots H_2 H_1)^T$. However, if we are interested only in the solution of the least squares problem, we do not have to compute explicitly either matrices H_k or $\widehat{Q}$. We just find the factor R and store it in the same memory space occupied by A, and $\widehat{Q}^T\mathbf{b}$ and store it in the same memory location occupied by $\mathbf{b}$. At the end, we solve the triangular system with backward substitution, as shown bellow.

Householder triangularization algorithm for solving $A\,\mathbf{x} = \mathbf{b}$ with QR

for $k = 1, \ldots, n$ ** Triangularization **

. $\mathbf{x} = A(k:m,\, k)$

. $\mathbf{v} = sign(x_1)\|\mathbf{x}\|\,\mathbf{e}_1 + \mathbf{x}$

. $\mathbf{v} = \mathbf{v}/\|\mathbf{v}\|$

. $A(k:m,\, k:n) = A(k:m,\, k:n) - 2\mathbf{v}\,(\mathbf{v}^T\, A(k:m,\, k:n))$

. $\mathbf{b}(k:m) = \mathbf{b}(k:m) - 2\mathbf{v}(\mathbf{v}^T\,\mathbf{b}(k:m))$

end

$\mathbf{x}(n) = \mathbf{b}(n)/A(n,n)$ ** Backward sustitution **

for $k = n-1:-1:1$

$\vdots$ $\mathbf{x}(k) = (\ \mathbf{b}(k) - A(k, k+1:n)\cdot\mathbf{b}(k+1:n)\ )/A(k,k)$

end

Notation. We have used the *MATLAB* notation for arrays. For instance, $\mathbf{x} = A(k:m,\, k)$ represents the vector constructed with coefficients a_{ik}, $k \le i \le m$ and k fixed; $A(k:m,\, k:n)$ represents the submatrix with coefficients $\{a_{ij}\}_{i=k,j=k}^{m,n}$; $\mathbf{b}(k:m)$ represents the subvector $\{b_i\}_{i=k}^{m}$.

The most important steps in the previous algorithm are the last two lines in the ** Triangularization ** loop. The main idea is that it is not necessary to construct H to compute a product $H\,\mathbf{y}$, since

$$H\,\mathbf{y} = (I - 2\mathbf{u}\mathbf{u}^T)\mathbf{y} = \mathbf{y} - \mathbf{u}(\mathbf{u}^T\mathbf{y}) \quad \text{with} \quad \mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|},$$

so we only need the vectors $\mathbf{v}$ and $\mathbf{y}$ at each step of the process. Numerical results are shown in Section 6.

4 The singular value decomposition (SVD)

4.1 Symmetrizing

The key idea to achieving the SVD of a matrix A is *symmetrizing*. That is, if $A \in \mathbb{R}^{m\times n}$, we can consider the symmetric positive semidefinite matrices $A^T A \in \mathbb{R}^{n\times n}$ and $A\,A^T \in \mathbb{R}^{m\times m}$. By the spectral theorem for symmetric matrices, these matrices are diagonalizable. For instance, if $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $A^T A$ with orthonormal eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, then

$$A^T A = V\,D\,V^T, \quad \text{with} \quad V = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix}, \quad D = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.$$

Each eigenvalue λ_j is real and non-negative, because

$$A^T A\,\mathbf{v}_j = \lambda_j\,\mathbf{v}_j \quad \Rightarrow \quad \mathbf{v}_j^T A^T A\,\mathbf{v}_j = \lambda_j\,\mathbf{v}_j^T\mathbf{v}_j \quad \Rightarrow \quad \lambda_j = \frac{\|A\,\mathbf{v}_j\|^2}{\|\mathbf{v}_j\|^2} \ge 0. \tag{6}$$

Therefore, we can order the eigenvalues. Without loss of generality, we assume that

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n \ge 0.$$

We remark that some eigenvalues may be repeated. Furthermore, each eigenvalue λ_j of $A^T A$ is also an eigenvalue of $A\,A^T$ since

$$A^T A\,\mathbf{v}_j = \lambda \mathbf{v}_j \quad \Rightarrow \quad (A\,A^T)A\,\mathbf{v}_j = \lambda_j\,A\mathbf{v}_j.$$

4.2 Reduced SVD

We have found that matrices A^TA and AA^T have the same eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ with corresponding eigenvectors

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^n \quad \text{and} \quad A\mathbf{v}_1, A\mathbf{v}_2, \ldots, A\mathbf{v}_n \in \mathbb{R}^m,$$

respectively. The eigenvectors $\mathbf{v}_j$ of A^TA are orthonormal. However, the eigenvectors of AA^T are only orthogonal ($(A\mathbf{v}_i)^TA\mathbf{v}_j = \mathbf{v}_i^TA^TA\mathbf{v}_j = \mathbf{v}_i^T\lambda_j\mathbf{v}_j = \lambda_j\delta_{ij}$), so we normalize them to get an orthonormal set $\mathbf{u}_1, \ldots, \mathbf{u}_n$ in $\mathbb{R}^m$

$$\mathbf{u}_j = \frac{A\mathbf{v}_j}{\|A\mathbf{v}_j\|} = \frac{A\mathbf{v}_j}{(\lambda_j)^{1/2}}, \quad j = 1, 2, \ldots, n. \tag{7}$$

Definition 4.1. Non-negative values

$$\sigma_1 = \sqrt{\lambda_1} \geq \sigma_2 = \sqrt{\lambda_2} \geq \ldots \geq \sigma_n = \sqrt{\lambda_n} \geq 0,$$

are called the *singular values* of the matrix A.

Therefore, according to (7), the following relationship is obtained between the two sets of orthonormal vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\} \subset \mathbb{R}^m$ and $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^n$:

$$A\mathbf{v}_j = \sigma_j\mathbf{u}_j, \quad j = 1, 2, \ldots, n. \tag{8}$$

These relationships can be expressed as the matrix product:

$$A \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \sigma_1\mathbf{u}_1 & \sigma_2\mathbf{u}_2 & \cdots & \sigma_n\mathbf{u}_n \\ | & | & & | \end{bmatrix},$$

which leads to

$$A = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix}}_{\widehat{U}\ (m \times n)} \underbrace{\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}}_{\widehat{\Sigma}\ (n \times n)} \underbrace{\begin{bmatrix} — & \mathbf{v}_1^T & — \\ — & \mathbf{v}_2^T & — \\ & \vdots & \\ — & \mathbf{v}_n^T & — \end{bmatrix}}_{V^T\ (n \times n)}. \tag{9}$$

This factorization can also be expressed as **sum of rank one matrices**:

$$A = \sigma_1\mathbf{u}_1\mathbf{v}_1^T + \sigma_2\mathbf{u}_2\mathbf{v}_2^T + \cdots + \sigma_n\mathbf{u}_n\mathbf{v}_n^T, \quad \text{with} \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0. \tag{10}$$

Note. In the matrix equation $AV = \widehat{U}\widehat{\Sigma}$ or $A = \widehat{U}\widehat{\Sigma}V^T$, the matrix $\widehat{U}$ is a rectangular matrix of size $m \times n$ with orthonormal columns in $\mathbb{R}^m$, $\widehat{\Sigma}$ is an $n \times n$ diagonal matrix with singular values, and V is an $n \times n$ orthogonal matrix (i.e. $V^{-1} = V^T$). The reduced SVD is also valid for matrices with complex entries or coefficients, but now V is Hermitian, so V^* (the complex conjugate) replaces V^T in the matrix factorization. For the interested reader, reference [26] is an extraordinary paper that surveys the contributions of five mathematicians who were responsible for establishing the existence of the SVD and developing its theory.

4.3 Full SVD

In most applications, the reduced SVD decomposition is employed. However, in textbooks and many publications, the '**full**' SVD decomposition is used. The reduced and full SVD are the same for $m = n$. We illustrate two cases: $m > n$ and $m < n$.

- **Case $m > n$: the columns of the matrix $\widehat{U}$ do not form a basis of $\mathbb{R}^m$.** We augment $\widehat{U} \in \mathbb{R}^{m\times n}$ to an orthogonal matrix $U \in \mathbb{R}^{m\times m}$ by adding $m-n$ orthonormal columns and replacing $\widehat{\Sigma} \in R^{n\times n}$ by $\Sigma \in \mathbb{R}^{m\times n}$ adding $m-n$ null rows $\widehat{\Sigma}$:

$$A = \underbrace{\begin{bmatrix} | & | & & | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n & \mathbf{u}_{n+1} & \cdots & \mathbf{u}_m \\ | & | & & | & | & & | \end{bmatrix}}_{U\ (m\times m)} \underbrace{\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\Sigma\ (m\times n)} \underbrace{\begin{bmatrix} — & \mathbf{v}_1^T & — \\ — & \mathbf{v}_2^T & — \\ & \vdots & \\ — & \mathbf{v}_n^T & — \end{bmatrix}}_{V^T\ (n\times n)}. \tag{11}$$

- **Case $m < n$: the reduced decomposition is of the form $A = U\widehat{\Sigma}\widehat{V}^T$.** The rows of $\widehat{V}^T$ does not form a basis of $\mathbb{R}^n$ so we must add $n-m$ orthonormal rows to obtain an orthogonal matrix V^T and adding $n-m$ null columns to $\widehat{\Sigma}$ to get Σ:

$$A = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m \\ | & | & & | \end{bmatrix}}_{U\ (m\times m)} \underbrace{\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_m & 0 & \cdots & 0 \end{bmatrix}}_{\Sigma\ (m\times n)} \underbrace{\begin{bmatrix} — & \mathbf{v}_1^T & — \\ — & \mathbf{v}_2^T & — \\ & \vdots & \\ — & \mathbf{v}_m^T & — \\ — & \mathbf{v}_{m+1}^T & — \\ & \vdots & \\ — & \mathbf{v}_n^T & — \end{bmatrix}}_{V^T\ (n\times n)}. \tag{12}$$

The previous results are summarized in the following theorem.

Theorem 4.2. *Every matrix $A \in \mathbb{R}^{m\times n}$ ($\mathbb{C}^{m\times n}$, in the complex case) has a singular value decomposition of the form*

$$A = U\Sigma V^T, \quad (A = U\Sigma V^*, \text{ in the complex case}) \tag{13}$$

with the orthogonal matrices U, V (or unitary, in the complex case), and the matrix Σ as indicated in the previous development.

4.4 Computing the SVD

As stated in [27] (Lecture 23), the SVD of $A \in \mathbb{C}^{m\times n}$ $(m > n)$, $A = U\,\Sigma\,V^*$ is related to the eigenvalue decomposition of the covariance matrix $A^*A = V\,\Sigma^*\Sigma\,V^*$ and mathematically it may be calculated doing the following:

1. Form A^*A;
2. Compute the eigenvalue decomposition $A^*A = V\,\Lambda\,V^*$;
3. Let Σ be the non negative diagonal square root of Λ;
4. Solve the system $U\,\Sigma = A\,V$ for unitary U (e.g. via QR factorization).

But the problem with this strategy is that the algorithm is not stable, mainly because it relies on the covariance matrix A^*A, which we have found before in the normal equations for least squares problems.

Additionally, the eigenvalue problem in general is very sensitive to numerical perturbations in computer's finite precision arithmetic

An alternative stable way to compute the SVD is to reduce it to an eigenvalue problem by considering a $2m \times 2m$ Hermitian matrix and the corresponding eigenvalue system.

$$H = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix} \quad \Longrightarrow \quad H \begin{bmatrix} V \\ U \end{bmatrix} = \begin{bmatrix} V\,\Sigma \\ U\,\Sigma \end{bmatrix},$$

since $A = U\Sigma V^*$ implies $A\,V = U\Sigma$ and $A^*U = V\,\Sigma$. Thus the singular values of A are the absolute values of the eigenvalues of H, and the singular vectors of A can be extracted from the eigenvectors of H, which can be done in an stable way, contrary to the previous strategy. This Hermitian eigenvalue problems are usually solved by a two-phase computation: first reduce the matrix to tridiagonal form, then diagonalize the tridiagonal matrix. The reduction is done by similarity unitary transformations, so the diagonal matrix contains the information about the singular values.

Actually, this strategy has been standard for computing the SVD since the work of Golub and Kahan in the 1960s [10]. The method involves in phase 1 applying Householder reflections alternately from the left and right of the matrix to reduce it to an upper bidiagonal form. In phase 2, the SVD of the bidiagonal matrix is determined with a variant of the QR algorithm. More recently, divide-and-conquer algorithms [21] have become the standard approach for computing the SVD of dense matrices in practice. These strategies overcome the computational difficulties associated with ill-conditioned or rank-deficient matrices during the SVD calculations.

4.5 Least squares with SVD

Most of the software environments, like *MATLAB* and *Phyton*, incorporate very efficient algorithms and state of the art tools related to SVD. So, using those routines provide reasonable accurate results in most of the cases.

Concerning linear least squares problems, we know that this often leads to an inconsistent overdetermined system $A\mathbf{x} = \mathbf{b}$ with $A \in \mathbb{R}^{m\times n}$, $m \geq n$. Thus, we seek the minimum of the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$. We know that if A is of full rank $r = n$, then A^TA is positive definite symmetric and the least squares solution is given by

$$\widehat{\mathbf{x}} = \left(A^TA\right)^{-1} A^T\mathbf{b}.$$

Via SVD, $A = U\Sigma V^T$, and using that $V^{-1} = V^T$, $U^{-1} = U^T$, $\Sigma^T\Sigma$ invertible, we have

$$\left(A^TA\right)^{-1} A^T = \left(V\Sigma^TU^TU\Sigma V^T\right)^{-1} V\,\Sigma^TU^T = V\,\left(\Sigma^T\Sigma\right)^{-1}\Sigma^TU^T = V\Sigma^\dagger U^T = A^\dagger,$$

and the least squares solution is given by

$$\widehat{\mathbf{x}} = A^\dagger\mathbf{b}. \tag{14}$$

What is remarkable is that the solution given by (14) is still valid, even if A is rank deficient. The following formal definition of the pseudoinverse corroborate our claim.

Definition 4.3. Let $A = U\Sigma V^T$ a real $m \times n$ matrix with rank $r \leq n$, then its ***pseudoinverse*** is the $n \times m$ matrix, denoted by $A^\dagger$, given by

$$A^\dagger = V\Sigma^\dagger U^T \quad \text{with} \quad \Sigma^\dagger = \text{diag}(1/\sigma_1, \ldots, 1/\sigma_r, 0, \ldots, 0) \in \mathbb{R}^{n\times m}. \tag{15}$$

With this definition $A^\dagger$ is well defined and it has the same size as A^T. If A is full rank, then $A^\dagger$ is called the left inverse of A since $A^\dagger A = I_n$, and $P_{_A} = AA^\dagger$ defines the projection onto the column space of A. When A is an invertible square matrix $A^\dagger = A^{-1}$.

Fitting data to a polynomial curve. Given the point set $(t_1, \hat{y}_1), \ldots, (t_m, \hat{y}_m)$. The algorithm for calculating $\widehat{\mathbf{x}} \in \mathbb{R}^{n+1}$, with the coefficients of the polynomial of degree n, consists of the following steps:

1. Form the $m \times (n+1)$ design matrix A, with coefficients $a_{ij} = t_i^{j-1}$.
2. Compute the SVD of $A = U\,\Sigma V^T$. Actually, the reduced SVD, $A = \widehat{U}\widehat{\Sigma}V^T$, is sufficient.
3. Calculate the generalized inverse $A^\dagger = V\,\Sigma^\dagger U^T$.
4. Calculate $\widehat{\mathbf{x}} = A^\dagger \mathbf{b}$, with $\mathbf{b} = \{\hat{y}_j\}_{j=1}^m$ being the vector of observations.

In Section 5 we present numerical results and compare them with the results obtained with QR and normal equations algorithms.

5 Numerical comparisons of QR and SVD

As before we consider the *Filip* data set, which consists of 82 observations of a variable y for different t values. The aim is to model these data set using a 10th-degree polynomial, using both, the QR factorization and SVD, to solve the associated least squares problem. In Section 2 we gave a description of the data and showed numerical results with the normal equations approach. Here we use the QR algorithm with Householder reflections, introduced in Section 3, and the SVD, described in Section 4.

Table 2 shows the coefficient values of the polynomial obtained with both algorithms. The coefficients obtained with the QR algorithm are very close to the certified values of *NIST* (shown in Table 1), while the coefficients obtained with SVD are far from the certified ones, with two or three orders of magnitude apart and different signs for most of them. In fact, the relative difference $\|\widehat{\mathbf{c}} - \mathbf{c}_{nist}\|/\|\mathbf{c}_{nist}\|$ of the coefficients obtained with the stable QR is insignificant, while the relative difference is as high as 100% when the SVD is employed. However, the polynomial obtained with the SVD shows that the data still fit fairly well to the obtained curve, as shown in Figure 6. Again, the main differences between the accurate curve (red line) with respect to the less accurate (blue line) is more evident at the left and right extremes of the interval.

A better measure for accuracy is the norm of the residual $\|\mathbf{b} - A\,\widehat{\mathbf{x}}\|$, since the algorithms are designed to minimize this quantity. We observe that the residual obtained with the QR algorithm is very close to the certified one (shown in Figure 2) and, surprisingly this residual is slightly lower than the certified one, while the residual obtained with the SVD is higher than the certified one but lower than the one obtained with the normal equations. So, we conclude that the best method for this particular problem is QR, followed by SVD and the less accurate is obtained with the normal equations.

Polynomial coefficients	QR ($\times 10^3$)	SVD
$\widehat{c}_1$	-1.467489624841714	8.443047022531269
$\widehat{c}_2$	-2.772179612867669	1.364997532790476
$\widehat{c}_3$	-2.316371099847143	-5.350747822573923
$\widehat{c}_4$	-1.127973950228995	-3.341901399544638
$\widehat{c}_5$	-0.354478236724762	-0.406458058717373
$\widehat{c}_6$	-0.075124202404921	0.257727453320758
$\widehat{c}_7$	-0.010875318135669	0.119771677097139
$\widehat{c}_8$	-0.001062214996057	0.023140894524175
$\widehat{c}_9$	-0.000067019116127	0.002403995388431
$\widehat{c}_{10}$	-0.000002467810808	0.000131618846926
$\widehat{c}_{11}$	-0.000000040296253	0.000002990001355
Relative difference	7.68×10^{-7}%	100%
Norm of residual	0.028210838088578	0.032726981836403

Table 2: Comparison of polynomial coefficients: QR and SVD.

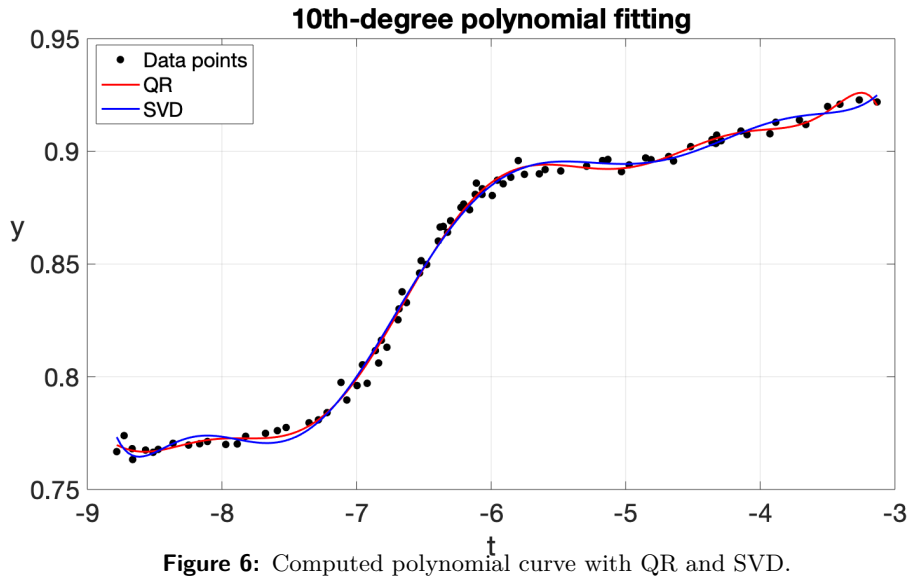


Figure 6: Computed polynomial curve with QR and SVD.

6 Neural network approach

Finally, we present a neural network (NN) framework to address the same fitting problem analyzed in the preceding sections. While NNs have historically been less common than classical methods, they have recently emerged as powerful tools across numerous scientific disciplines. Our goal is to develop a NN that can be used together with the known data for curve fitting. If we have m observations

$$(t_1, \hat{y}_1), (t_2, \hat{y}_2), \ldots, (t_m, \hat{y}_m)\,, \tag{16}$$

where $\hat{y}_i$, $i = 1, 2, \ldots, m$, are measurements of $y(t_i)$. The idea is to model $y(t)$ as a NN of the form:

$$y(t) \approx \widetilde{y}(t, \mathbf{W}, \mathbf{b}), \tag{17}$$

where $\mathbf{W}$ and $\mathbf{b}$ are two sets of parameters of the neural network, which must be determined. This NN model consists of an input layer, L hidden layers, each one containing N_ℓ neurons, and an additional output layer. The received input signal propagates through the network from the input layer to the output layer, through the hidden layers. When the signals arrive in each node, an activation function $\phi : \mathbb{R} \to \mathbb{R}$ is used to produce the node output [17, 24, 28]. Neural networks with many layers (two or more) are called multi-layer neural networks.

Example 6.1. The model corresponding to a neural network with a single hidden layer consisting of five neurons, each activated by the hyperbolic tangent function, and a scalar output obtained through a linear combination of the hidden activations, can be expressed as

$$\widetilde{y}(t, \mathbf{W}, \mathbf{b}) = W^2 \phi(W^1 t + b^1) + b^2,$$

where $\phi(x) = \tanh(x)$. Explicitly, in this case, the neural network can be written as a functional representation of the input x in the following form:

$$\begin{aligned} \widetilde{y}(t, \mathbf{W}, \mathbf{b}) \quad = \quad & W_1^2 \tanh(W_1^1 t + b_1^1) + W_2^2 \tanh(W_2^1 t + b_2^1) + \\ & W_3^2 \tanh(W_3^1 t + b_3^1) + W_4^2 \tanh(W_4^1 t + b_4^1) + \\ & W_5^2 \tanh(W_5^1 t + b_5^1) + b^2. \end{aligned}$$

Hence, the complete model involves 16 unknown parameters, which entirely determine the behavior of the neural network. The unknown parameters are optimized using an appropriate optimization algorithm (e.g., gradient descent) based on the given training dataset (16). The goal is to minimize a loss function that quantifies the discrepancy between the network's predictions and the true target values, which is described below.

Remark 6.2. It is known that any continuous, non-constant function mapping $\mathbb{R}$ to $\mathbb{R}$ can be approximated arbitrarily well by a multilayer neural network, see [6, 13, 25]. This result establishes the expressive power of feedforward neural networks. Specifically, it shows that even a network with a single hidden layer, containing a sufficient number of neurons and an appropriate activation function, can approximate any continuous function on compact subsets of $\mathbb{R}$.

6.1 General NN architecture

In this work, the neural network is described in terms of the input $t \in \mathbb{R}$, the output $\widetilde{y} \in \mathbb{R}$, and an input-to-output mapping $t \mapsto \widetilde{y}$. For any hidden layer ℓ, we consider the pre-activation $T^\ell \in \mathbb{R}^{N_\ell}$ and post-activation $Y^\ell \in \mathbb{R}^{N_{\ell+1}}$ vectors as

$$T^\ell(t) = \left[T_1^\ell(t), \ldots, T_{N_\ell}^\ell(t)\right]^T \qquad \text{and} \qquad Y^\ell(t) = \left[Y_1^\ell(t), \ldots, Y_{N_\ell+1}^\ell(t)\right]^T, \tag{18}$$

respectively. Thus, the activation in the ℓ-th hidden layer of the network for $j = 1, \ldots, N_{\ell+1}$, is given by [19]:

$$Y_j^\ell(t) = b_j^\ell + \sum_{k=1}^{N_\ell} W_k^\ell \, T_k^\ell(t), \qquad \ell = 1, \ldots, L, \tag{19}$$

where

$$T_k^1(t) = t, \qquad T_k^\ell(t) = \phi(Y_k^{\ell-1}(t)), \qquad \ell = 2, \ldots, L, \tag{20}$$

for $k = 1, \ldots, N_\ell$. Here, W_k^ℓ and b^ℓ are the weights and bias parameters of layer ℓ. Activation functions ϕ must be chosen such that the differential operators can be readily and robustly evaluated using reverse mode automatic differentiation [7]. Throughout this work, we have been using relatively simple feedforward neural networks architectures with hyperbolic tangent and sigmoidal activation functions. Results show that these functions are robust for the proposed formulation. It is important to remark that as more layers and neurons are incorporated into the NN the number of parameters significantly increases. Thus the optimization process becomes less efficient.

Figure 7 shows an example of the computational graph representing a NN as described in equations (18)–(20). When one node's value is the input of another node, an arrow goes from one to another. In this particular example, we have

$$\text{layers} = \left(N_\ell\right)_{\ell=1}^{L+2} = (1, 4, 4, 4, 4, 1).$$

That is, the total number of hidden layers is 4. The first entry corresponds to the input layer ($\ell = 1$) and contains a single neuron. The next four entries correspond to the hidden layers, each one with $N_\ell = 4$ neurons. Finally, the last layer is the output layer and contains one neuron, corresponding to a single solution value. Bias is also considered (light grey nodes), there is a bias node in each layer, which has a value equal to the unit and is only connected to the nodes of the next layer. Although, the number of nodes for each layer can be different; the same number has been employed in this paper for simplicity.

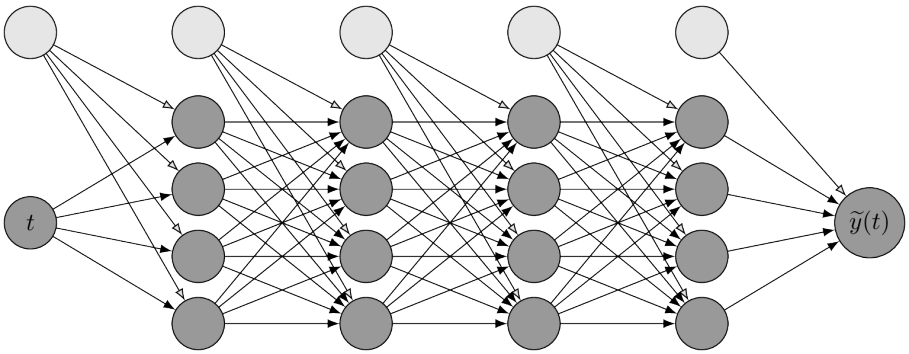


Figure 7: Neural network with 4 hidden layers, one input and one output.

6.2 Optimization Algorithm

The parameters $\mathbf{W}$ and $\mathbf{b}$ in (17) are determined using a finite set of training points $\{t_i, \hat{y}_i\}_{i=1}^{m}$ corresponding to the dataset in (16). Here, m denotes the number of training points, which can be arbitrarily selected. The parameters are estimated by minimizing the mean squared error (MSE) loss

$$E = \frac{1}{N_u} \sum_{i=1}^{N_u} |\widetilde{y}(t_i, \mathbf{W}, \mathbf{b}) - \hat{y}_i|^2 , \tag{21}$$

where E represents the error over the training dataset $\{t_i, \hat{y}_i\}_{i=1}^{m}$. The neural network defined in (18)–(20) is trained iteratively, updating the neuron weights by minimizing the discrepancy between the target values $\hat{y}_i$ and the outputs $\widetilde{y}(t_i, \mathbf{W}, \mathbf{b})$.

Formally, the optimization problem can be written as

$$\min_{[\mathbf{W},\mathbf{b}]} E([\mathbf{W}, \mathbf{b}]), \tag{22}$$

where the vector $[\mathbf{W}, \mathbf{b}]$ collects all unknown weights and biases. Several optimization algorithms can be employed to solve the minimization problem (21) and (22), and the final performance strongly depends on the residual loss achieved by the chosen method. In this work, the optimization process is performed using gradient-based algorithms, such as Stochastic Gradient Descent (SGD) or Adam [2]. These methods iteratively adjust the network parameters in the direction that minimizes the loss function. Starting from an initial guess $[\mathbf{W}^0, \mathbf{b}^0]$, the algorithm generates a sequence of iterates $[\mathbf{W}^1, \mathbf{b}^1]$, $[\mathbf{W}^2, \mathbf{b}^2]$, $[\mathbf{W}^3, \mathbf{b}^3], \ldots$, converging to a (local) minimizer as the stopping criterion is satisfied.

The required gradients are computed efficiently using automatic differentiation [3], which applies the chain rule to propagate derivatives through the computational graph. In practice, this process is known as backpropagation [22]. All computations were carried out in Python using Pytorch [2], a widely used and well-documented open-source library for machine learning.

6.3 Results

To illustrate the capability of the NN, we consider the same curve-fitting problem based on the *Filip* dataset, which consists of 82 observations of a variable y at different values of t. The computed solution is shown in Figures 8 and 9. The predicted values of y are obtained by training all the parameters of a five-layer neural network: the first layer contains a single neuron, while each hidden layer consists of twenty neurons. The hyperbolic tangent and sigmoidal activation functions are used throughout the network. It is worth noting that the NN solution closely resembles the one obtained using a 10th-degree polynomial fit (*NIST*). Therefore, this example shows its potential for generalization and robustness in more challenging scenarios.

7 Conclusions

Fitting a curve to a given set of data is one of the most simple of the so called '*ill-posed*' problems. This is an example of a broad set of problems called least squares problems. This simple problem contains many of the ingredients, both theoretical and computational, of modern challenge and complex problems that are of great importance in computational modelling and applications, specially when computer solutions are obtained using finite precision machines. Commonly there is no '*best computational algorithm*' for general problems, but for a particular problem, like the one considered in this article, we can compare results obtained with different approaches or algorithms.

It is clear that the best fit to a 10th-degree polynomial is obtained with the QR algorithm, as it produces the smallest residual when compared to algorithms based on *the normal equations* and SVD. It is noteworthy that each method yields entirely different coefficients for this polynomial. Not only the sign of the coefficients but also the scale of the values differ drastically. These results demonstrate that even simple ill-posed problems must be studied and numerically solved with extreme care, employing stable state-of-the-art algorithms and tools that avoid the accumulation of rounding errors due to the finite arithmetic precision of computers.

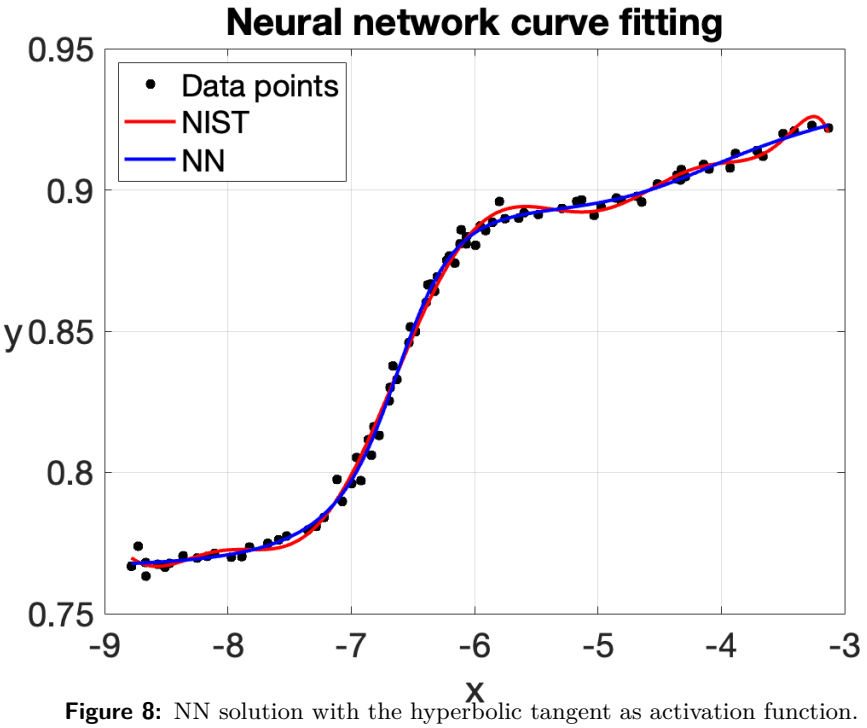


Figure 8: NN solution with the hyperbolic tangent as activation function.

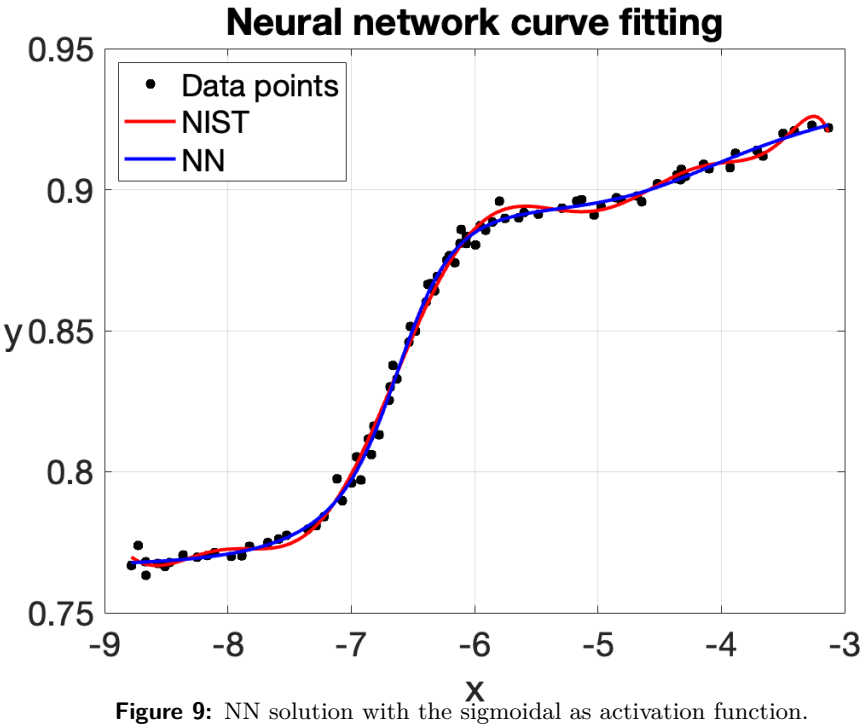


Figure 9: NN solution with the sigmoidal as activation function.

Concerning the neural network approach, we obtained qualitatively excellent numerical results. The resulting fitted curve is smooth and provides an accurate representation of the data, and it appears to offer a slightly improved approximation when compared to the QR-based fit, while maintaining sufficient flexibility to capture the overall behavior. These results suggest that the multi-layer neural networks constitute an effective and robust framework for curve fitting. Is the NN approach better than the QR algorithm for curve fitting?

Again, this general question depends of what you are looking for. But if you are able to construct with NN a 10th-degree polynomial that fits the given experimental data, then you are able to answer this particular question. A diligent reader may put their hands on the problem in order to give an answer.

Acknowledgements. We would like to express our sincere gratitude to the Department of Mathematics at Universidad Autónoma Metropolitana–Iztapalapa for their valuable support of this research work. The authors also gratefully acknowledge partial support from the Secretaría de Ciencia, Humanidades, Tecnología e Innovación (Secihti) through the Investigadores e Investigadoras por México program and the Ciencia de Frontera Project No. CF-2023-I-2639.

References

[1] ˙Statistical reference datasets: Filip data. `https://www.itl.nist.gov/div898/strd/lls/data/Filip.shtml`. Dataset web · no author specified.

[2] ˙Pytorch. `https://pytorch.org`, 2025. Página web del proyecto.

[3] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey, 2015.

[4] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[5] P. Businger and G. H. Golub. Linear least squares solutions by householder transformations. *Numerische Mathematik*, 7:269–276, 1965.

[6] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[7] D. A. Fournier, H. J. Skaug, J. Ancheta, J. Ianelli, A. Magnusson, M. N. Maunder, A. Nielsen, and J. Sibert. Ad model builder: Using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods & Software*, 27(2):233–249, 2012.

[8] C. F. Gauss. *Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections*. Dover, New York, 1963. First published in 1809.

[9] C. F. Gauss. *Theory of the Combination of Observations Least Subject to Errors. Part I, Part II, Supplement*. SIAM, Philadelphia, 1995.

[10] G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM Journal on Numerical Analysis*, 2(2):205–224, 1965.

[11] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 1983. Ediciones: 1983, 1989, 1996, 2013.

[12] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14:403–420, 1970.

[13] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[14] A. S. Householder. A class of methods for inverting matrices. *Journal of the Society for Industrial and Applied Mathematics*, 6(2):189–195, 1958.

[15] L. H. Juárez. Resultados relevantes del álgebra lineal en modelos y aplicaciones. *Revista Metropolitana de Matemáticas Mixba'al*, 16(1), 2025.

[16] A. M. Legendre. *Nouvelles methodes pour la determination des orbites des cometes*. Courcier, Paris, 1805.

[17] S. Marsland. *Machine Learning: An Algorithmic Perspective*. CRC Press, 2015.

[18] M. Merriman. Note on the history of the method of least squares. *The Analyst*, 4(5):140–143, 1877.

[19] C. Michoski, M. Milosavljevic, T. Oliver, and D. Hatch. Solving irregular and data-enriched differential equations using deep neural networks, 2019.

[20] C. B. Moler. *Numerical Computing with MATLAB*. SIAM, 2004. Segunda edición según tu bibitem.

[21] Y. Nakatsukasa and N. Higham. Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the svd. *SIAM Journal on Scientific Computing*, 35(3):A1325–A1349, 2013.

[22] M. A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, San Francisco, CA, 2015.

[23] Y. Nievergelt. A tutorial history of least squares with applications to astronomy and geodesy. *Journal of Computational and Applied Mathematics*, 121:37–72, 2000.

[24] S. Pattanayak. *Pro Deep Learning with TensorFlow: A Mathematical Approach to Advanced Artificial Intelligence in Python*. Apress, New York, NY, 2017.

[25] A. Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.

[26] G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.

[27] L. N. Trefethen and D. B. III. *Numerical Linear Algebra*. SIAM, 1997.

[28] G. Zaccone and R. Karim. *Deep Learning with TensorFlow: Explore Neural Networks and Build Intelligent Systems with Python*. Packt Publishing Ltd, 2018.

Hacia un Método de Tractografía Basado en Información Microestructural por Medio de Optimización Convexa

Ramón Aranda[*1,2], Gabriel A. Rocha[1], Ángel Díaz-Pacheco[3] y Miguel Á. Álvarez-Carmona[1,2]

[1]Centro de Investigación en Matemáticas, A.C., México.
[2]Secretaría de Ciencia, Humanidades, Tecnología e Innovación (Secihti), México.
[3]Departamento de Ingeniería Electrónica, Campus Irapuato-Salamanca, Universidad de Guanajuato, México.

Resumen

En este trabajo se presenta un método para estimar la estructura de la materia blanca (haces de axones) integrando información microestructural mediante optimización convexa. El enfoque valida localmente cada segmento utilizando un modelo físico de difusión que asigna pesos a las posibles trayectorias, reduciendo conexiones espurias desde etapas tempranas del proceso. El método se evalúa frente a algoritmos clásicos mediante métricas como LiFE, correlación de conectividad y el área bajo la curva ROC. Los resultados muestran una mayor coherencia estructural y una reducción de falsos positivos, con un desempeño robusto ante el ruido. El estudio evidencia la viabilidad de incorporar información microestructural en las estimaciones, aunque también revela una mayor cantidad de falsos negativos y una alta demanda computacional.

Palabras clave: Tractografía; Optimización; Información microestructural; Haces de axones; Estructura cerebral.

1 Introducción

La complejidad que presenta la estimación de las conexiones neuronales en el cerebro humano ha impulsado el desarrollo de diversas herramientas para su adquisición y estudio. Entre ellas, la tractografía cerebral, basada en imágenes por resonancia magnética de difusión (dMRI), se ha convertido en un método clave para estimar estas conexiones estructurales. Esta técnica permite estimar las trayectorias de las fibras axonales que conectan distintas regiones cerebrales, proporcionando información importante sobre la organización y la integridad del conectoma humano [18]. Sin embargo, a pesar de los avances en esta técnica, persisten desafíos importantes que limitan su precisión, como la alta incidencia de trayectorias erróneas (falsos positivos) y la incapacidad de resolver de manera adecuada la complejidad estructural en zonas donde las fibras se cruzan o bifurcan [5, 12, 17]. En este contexto, la presente investigación propone abordar dichas limitaciones mediante la implementación de un enfoque de optimización convexa basado en microestructura, inspirado en el algoritmo *Convex Optimization Modeling for Microstructure Informed Tractography* (COMMIT) [6]. Esta propuesta integra información microestructural detallada en el proceso de la estimación de trayectorias de haces axonales, reduciendo tanto los falsos positivos como las incertidumbres en áreas de complejidad anatómica.

La relevancia de esta investigación es significativa, ya que un avance en la precisión de la tractografía cerebral no solo aportaría un mayor entendimiento sobre la conectividad estructural en individuos sanos, sino que

[*]arac@cimat.mx

también abriría nuevas posibilidades en el estudio de enfermedades neurológicas, donde el análisis detallado de las vías neuronales es crucial para la comprensión de los mecanismos patológicos [10]. Además, este desarrollo podría impactar directamente en áreas como la planificación quirúrgica y la medicina personalizada, donde la exactitud en la reconstrucción de los tractos neuronales es esencial para evitar errores clínicos [8, 15].

Este trabajo se enfoca en las limitaciones actuales de los métodos de tractografía para mejorar la reconstrucción de las conexiones cerebrales integrando información microestructural mediante optimización convexa, ofreciendo un marco más robusto para futuras aplicaciones clínicas y de investigación.

2 Trabajos relacionados

La tractografía es un enfoque que permite estimar las trayectorias de haces de axones mediante curvas tridimensionales (3D), a partir de las orientaciones derivadas de las dMRI [7]. En general, la construcción de una trayectoria comienza con la selección de un punto inicial, denominado semilla. A partir de este punto, la trayectoria se propaga utilizando la información proporcionada por las dMRI, generando así una curva que representa el recorrido estimado de las fibras axonales. El movimiento de la trayectoria puede describirse mediante la siguiente ecuación de actualización:

$$y_{t+1} = y_t + \Delta d_t, \tag{1}$$

donde y_t representa la posición tridimensional (3D) de la partícula en el tiempo t, d_t corresponde a la dirección de propagación y Δ es el tamaño de paso. Así, una trayectoria, s, está formada por $s = \{y_1, y_2, \ldots, y_m\}$. En términos generales, la principal diferencia entre los distintos métodos de tractografía radica en la manera en que se estima la dirección d_t en cada paso de la trayectoria [9, 13].

Una clase de métodos que han demostrado un gran desempeño son aquellos que logran integrar información de vecindades para estimar d_t [10]. Entre estos métodos se encuentran: la Tractografía basada en Comportamiento Colectivo (TCC) [1], la Tractografía por Filtro de Partículas (TFP) [11] y la Tractografía por Transporte Paralelo (TTP) [2].

TCC modela la propagación de las trayectorias inspirándose en el comportamiento colectivo de bandadas (*flocking*). Cada trayectoria sigue reglas locales de alineamiento, cohesión y separación, lo que permite que las trayectorias se ajusten mutuamente mientras avanzan. Esto genera un comportamiento colaborativo que reduce rutas aisladas y guía a las fibras hacia patrones anatómicamente plausibles. El método incorpora información de vecindades del conjunto de trayectorias, y no únicamente del punto local de difusión. Como resultado, produce tractogramas más coherentes y con menos falsos positivos.

TFP tiene como objetivo reducir la cantidad de trayectorias que terminan prematuramente dentro de la materia blanca o en el líquido cefalorraquídeo. La idea central de este método es realizar un retroceso en los tiempos $t - i$ en aquellas trayectorias que han experimentado una terminación anticipada. De esta manera, se aplica una corrección a la trayectoria estimada, permitiendo que alcance adecuadamente un punto de terminación válido.

Finalmente, TTP es un método que define un marco ortonormal a lo largo de cada trayectoria para evitar rotaciones arbitrarias en la orientación de las fibras. Utiliza el concepto geométrico de *transporte paralelo* [4] para actualizar las direcciones locales de manera suave y consistente. Esto permite que la dirección de propagación se adapte fielmente a los cambios en la estructura de la materia blanca. A diferencia de los métodos clásicos, evita giros abruptos y mantiene la coherencia direccional. Como resultado, se obtiene una tractografía más estable, precisa y menos afectada por artefactos de orientación.

Al igual que los métodos previamente descritos, en este trabajo también proponemos la integración de vecindades de información; sin embargo, incorporamos información microestructural mediante el algoritmo COMMIT [6] directamente durante el proceso de generación de la tractografía. Esta integración se realiza de forma local, aplicando COMMIT a pequeñas vecidades alrededor del punto y_t. Con ello, no solo se optimiza la selección de las trayectorias más plausibles, sino que además se posibilita la recuperación de falsos negativos, mejorando la precisión de las conexiones anatómicas estimadas.

3 Algoritmo propuesto

Nuestra contribución principal radica en la integración del algoritmo COMMIT [6] en el proceso de generación de tractografías, aplicándolo a pequeñas vecindades durante la estimación de las trayectorias, lo cual nos permite calcular d_t. A diferencia de las implementaciones tradicionales de COMMIT, que se aplican de manera posterior a la generación de las tractografías, nuestra propuesta explora su aplicación en tiempo real, reduciendo la dependencia de ajustes posteriores y preservando una mayor cantidad de información durante el proceso de reconstrucción.

3.1 COMMIT

Para abordar la problemática de la presencia de falsos positivos en las reconstrucciones de las tractografías, Daducci et al. (2015) [6] propusieron el algoritmo COMMIT, el cual utiliza un marco de optimización convexa para refinar tractografías mediante la eliminación de trayectorias inconsistentes con los datos de difusión. A diferencia de otros métodos de posprocesamiento, COMMIT incorpora información microestructural del tejido cerebral, ajustando la contribución de cada fibra en la señal medida y filtrando aquellas que no se justifican a partir del modelo físico subyacente.

Dado un tractograma, el modelo matemático de COMMIT se basa en la siguiente ecuación lineal:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \eta, \tag{2}$$

donde $\mathbf{y} \in \mathbb{R}^M$ representa la señal de difusión medida en cada vóxel, $\mathbf{A} \in \mathbb{R}^{M \times N}$ es la matriz de diseño que describe la contribución de cada trayectoria a la señal medida, $\mathbf{x} \in \mathbb{R}^N$ es el vector de pesos asociado a cada trayectoria candidata y $\eta \in \mathbb{R}^M$ corresponde al ruido de medición.

El objetivo de COMMIT es encontrar el vector $\mathbf{x}$ que mejor explica la señal $\mathbf{y}$, imponiendo restricciones de no negatividad y promoviendo la dispersión en $\mathbf{x}$ para eliminar trayectorias irrelevantes. Esto se logra mediante la siguiente formulación de optimización convexa:

$$\min_{\mathbf{x} \geq 0} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1, \tag{3}$$

donde el primer término minimiza el error de reconstrucción de la señal, mientras que el término de regularización $\|\mathbf{x}\|_1$ promueve la selección de un subconjunto reducido de fibras, eliminando aquellas que no contribuyen significativamente a la señal observada. El parámetro λ controla el grado de regularización [6].

COMMIT incorpora modelos de microestructura basados en principios biológicos, como el modelo Stick–Tensor [20] (que representa axones como cilindros delgados con difusión restringida) y el modelo de compartimentos mixtos [6] (que considera la presencia de múltiples tipos de tejidos, como fibras y espacio extracelular). Estos modelos, codificados en la matriz $\mathbf{A}$, permiten ajustar la contribución de cada trayectoria de acuerdo con su compatibilidad con la estructura neuronal, lo que ayuda a eliminar conexiones falsas [3].

Tras la resolución del problema de optimización presentado en la ecuación (3) (donde la matriz $\mathbf{A}$ es de una escala muy grande), se obtiene un conjunto refinado de fibras, en el cual muchas de las trayectorias originales han sido eliminadas debido a su baja contribución a la señal medida. El criterio de eliminación se basa en el valor de $\mathbf{x}$ obtenido para cada fibra. Si $x_i \approx 0$, la trayectoria i se considera inconsistente con la señal de difusión y es eliminada. De lo contrario, si $x_i > 0$, la trayectoria i es retenida en la tractografía final. Este proceso permite mejorar la especificidad de la tractografía y reducir significativamente la cantidad de falsos positivos, dando lugar a estimaciones de conectividad más precisas [6, 19].

3.2 Tractografía basada en COMMIT

Dado un tractograma base, calculado con cualquier algoritmo de tractografía, con $S = \{s_1, s_2, \ldots, s_K\}$, en cada paso del proceso de generación de las estimaciones de los haces de axones se obtienen trayectorias candidatas aplicando criterios de filtrado basados en un cono direccional (vecindad) alrededor de la trayectoria central estimada. Sea $\hat{s}_i$ el segmento de la trayectoria s_i contenido dentro del cono. Formalmente, si $\mathbf{c}$ denota

la dirección del cono y $\mathbf{u}_{\hat{s}_i,j}$ representa la dirección del segmento j de la trayectoria $\hat{s}_i$, dicho segmento se considera válido si todos sus subsegmentos satisfacen un umbral angular máximo θ_{tol}:

$$\max_j \angle\big(\mathbf{u}_{\hat{s}_i,j}, \mathbf{c}\big) \leq \theta_{\text{tol}}. \tag{4}$$

o equivalentemente en términos del producto escalar,

$$\frac{|\mathbf{u}_{\hat{s}_i,j} \cdot \mathbf{c}|}{\|\mathbf{u}_{\hat{s}_i,j}\|\|\mathbf{c}\|} \geq \cos(\theta_{\text{tol}}), \tag{5}$$

para todo segmento. Este criterio garantiza que las trayectorias en el cono no se alejen demasiado de la dirección predominante en la vencindad del cono.

Para cada trayectoria $\hat{s}_i$ en el cono, se calcula la dirección promedio de todos sus segmentos. Si la trayectoria $\hat{s}_i$ tiene $N_{\hat{s}_i}$ segmentos con vectores de dirección $\mathbf{u}_{\hat{s}_i,j}$, su dirección promedio se define como

$$\mathbf{d}_{\hat{s}_i} = \frac{1}{N_{\hat{s}_i}} \sum_{j=1}^{N_{\hat{s}_i}} \mathbf{u}_{\hat{s}_i,j}. \tag{6}$$

A continuación se normaliza este vector promedio para obtener la dirección unitaria:

$$\hat{\mathbf{d}}_{\hat{s}_i} = \frac{\mathbf{d}_{\hat{s}_i}}{\|\mathbf{d}_{\hat{s}_i}\|}. \tag{7}$$

De esta manera cada trayectoria válida queda representada por su dirección promedio unitaria $\hat{\mathbf{d}}_{\hat{s}_i}$, lo cual facilita la comparación entre fibras.

La similitud direccional entre dos trayectorias se cuantifica mediante el coseno del ángulo entre sus direcciones promedio. Para dos vectores unitarios $\hat{\mathbf{d}}_{\hat{s}_i}$ y $\mathbf{d}_t$, la métrica de similitud direccional se define como su producto escalar:

$$\sigma_i(\hat{s}_i, t) = \big|\hat{\mathbf{d}}_{\hat{s}_i} \cdot \mathbf{d}_t\big| = \cos\big(\angle(\hat{\mathbf{d}}_{\hat{s}_i}, \mathbf{d}_t)\big). \tag{8}$$

Este valor se encuentra en el rango $[0, 1]$ y mide cuán alineadas están las direcciones de dos trayectorias. En particular, la similitud de cada trayectoria s puede evaluarse respecto a la dirección predominante o frente a otras fibras candidatas, penalizando aquellas que desvían significativamente su orientación.

Finalmente, cada trayectoria candidata se puntúa con una función compuesta que pondera su peso dado por COMMIT y su similitud direccional. Si $x_{\hat{s}_i}$ es el peso asignado por COMMIT al segmento de trayectoria $\hat{s}_i$, se puede definir el peso normalizado como $\tilde{x}_{\hat{s}_i} = x_{\hat{s}_i} / \sum_k x_k$, donde x_k son los pesos de todos los segmentos en el cono. Con un parámetro $\lambda \in [0, 1]$ que regula la importancia relativa, se define la puntuación compuesta:

$$\kappa_{\hat{s}_i} = (1 - \lambda)\, \tilde{x}_{\hat{s}_i} \; + \; \lambda\, \sigma_i(\hat{s}_i, t). \tag{9}$$

De esta forma, si $\lambda = 0$ se toma en cuenta solo el criterio COMMIT y si $\lambda = 1$ solo la similitud direccional. Finalmente, se selecciona la trayectoria $\hat{s}^*$ con la mayor puntuación compuesta:

$$\hat{s}^* \; = \; \arg\max_{\hat{s}_i} \; \kappa_{\hat{s}_i}. \tag{10}$$

Posteriormente, podemos usar $\hat{s}^*$ para calcular d_{t+1} usando ecuación (7).

4 Experimentos y resultados

Con el objetivo de evaluar el desempeño del método propuesto de tractografía informada por microestructura mediante optimización convexa, esta sección detalla los experimentos y resultados.

Se emplean dos conjuntos de datos sintéticos ampliamente usados en la comunidad científica: el ISMRM Tractography Challenge 2015 (actualización 2023) [16] y DiSCo (Diffusion-Simulated Connectivity) [10]. El primero se usará para análisis espaciales y estructurales. El segundo, para evaluación cuantitativa de

conectividad. Su uso complementario ofrece un marco sólido para validar el nuevo método de tractografía propuesto.

En ambos casos se generarán tractografías con el algoritmo propuesto y con métodos tradicionales ampliamente utilizados. Posteriormente se aplicarán métricas de evaluación adecuadas al tipo de datos: para el primer conjunto se emplea la técnica Linear Fascicle Evaluation (LiFE) [14] para estimar la contribución predictiva de cada trayectoria. Para el segundo se usan métricas de correlación, exactitud y área bajo la curva ROC (AUC), tal como lo establece el protocolo de evaluación del DiSCo Challenge [10].

Para la validación anatómica cualitativa y cuantitativa basada en trayectorias con el conjunto ISMRM, primero se seleccionará una semilla en el cuerpo calloso, una región con conectividad interhemisférica bien definida que resulta ideal para evaluar la coherencia geométrica de las tractografías generadas. A partir de esta semilla se obtendrán tractografías mediante algoritmos determinísticos convencionales (como FACT, TEND, SD-Stream o iFOD2) y mediante el algoritmo propuesto, que integra restricciones microestructurales mediante COMMIT durante la generación de las trayectorias. Cada tractografía se evaluará con el algoritmo LiFE, que estima la contribución predictiva de cada trayectoria sobre la señal dMRI original, permitiendo filtrar trayectorias no justificadas y evaluar el ajuste global de cada modelo. Finalmente se comparará el rendimiento del algoritmo propuesto con el de los métodos convencionales considerando su coherencia anatómica respecto de las fibras de referencia, la puntuación de predicción bajo el modelo LiFE y el número de líneas filtradas por LiFE como no justificadas.

La validación cuantitativa de conectividad estructural entre regiones cerebrales utilizará el conjunto de datos del DiSCo Challenge. En este caso se evaluará solo el desempeño del algoritmo propuesto para contrastar sus resultados con los obtenidos por otros métodos participantes del challenge original. Se usará el subconjunto de alta resolución del DiSCo1, en sus versiones con ruido Rician a niveles de Relación señal-ruido (SNR por sus siglas en inglés) de 10, 20, 30 y 50. A partir de las tractografías generadas, se estimarán matrices de conectividad ponderadas entre las 16 regiones de interés (ROIs) definidas en el conjunto de datos. Las conexiones se cuantificarán según el número de trayectorias entre cada par de ROIs y, cuando corresponda, según el peso estimado en función de la microestructura. La calidad de estas matrices se evaluará mediante las métricas del DiSCo Challenge: correlación de Pearson con la matriz de referencia (peso continuo), área bajo la curva ROC (comparando con la matriz binaria de referencia) y exactitud de clasificación (pares correctamente identificados como conectados o no conectados). Finalmente, se analizará el comportamiento del algoritmo frente a diferentes niveles de ruido para evaluar su robustez y sensibilidad a condiciones adversas.

4.1 Resultados

La Figura 1 muestra una comparación visual de tres tractografías generadas a partir de una misma semilla localizada en el cuerpo calloso, utilizando diferentes algoritmos. Todas las tractografías fueron generadas con la misma cantidad de trayectorias, lo que permite una observación cualitativa más equitativa entre métodos.

A partir de esta visualización inicial pueden identificarse algunas diferencias notables entre las tres reconstrucciones. El algoritmo iFOD2 (Figura 1a) presenta una mayor dispersión de trayectorias hacia regiones laterales, algo que puede relacionarse con su naturaleza probabilística, propensa a explorar recorridos menos restringidos. SD-Stream (Figura 1b) exhibe una estructura más centralizada y simétrica en la región del cuerpo calloso; las trayectorias aparecen más alineadas, aunque ello podría limitar la cobertura en zonas periféricas. El algoritmo propuesto (Figura 1c) parece mantener un equilibrio entre coherencia espacial y extensión: las trayectorias se distribuyen dentro de un rango anatómicamente plausible y sin una dispersión excesiva. Estos resultados corresponden únicamente a una inspección visual preliminar.

Aunque el comportamiento del algoritmo propuesto resulta alentador, es necesario complementar este análisis con métricas cuantitativas en las siguientes secciones para validar su desempeño de forma objetiva. Para complementar el análisis visual anterior se aplicó el modelo LiFE a cada una de las tractografías generadas. Esta herramienta filtra las fibras que no contribuyen de manera significativa a explicar la señal de difusión y, de este modo, ofrece un estimador indirecto de la fidelidad anatómica de las trayectorias reconstruidas.

La Tabla 1 muestra los resultados obtenidos al aplicar LiFE. Aunque todos los métodos partieron del mismo número total de fibras (780), se observa una diferencia en la cantidad de trayectorias que LiFE considera

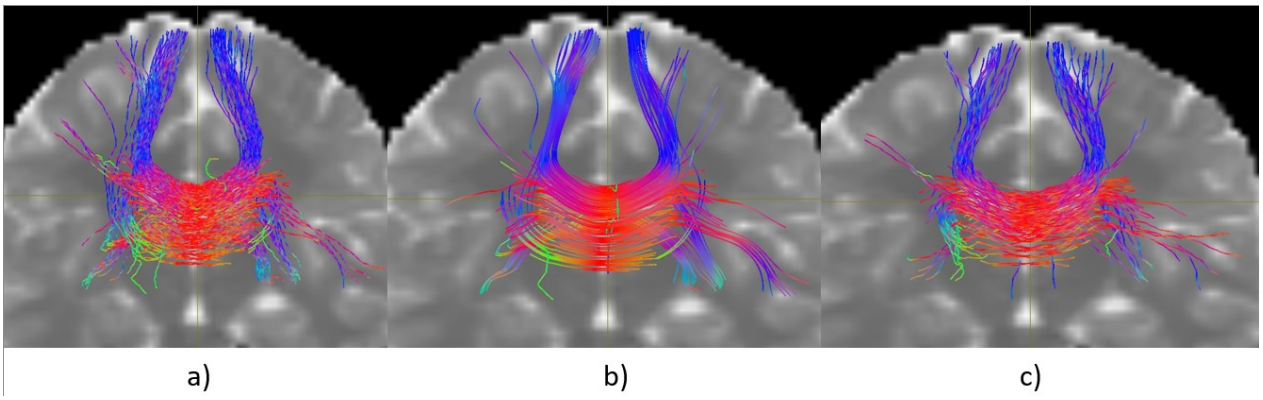


Figura 1: Comparación visual de tractografías generadas desde una semilla en el cuerpo calloso: a) iFOD2, b) SD-Stream, c) Algoritmo propuesto.

Tabla 1: Número de fibras retenidas por LiFE en cada tractografía (semilla en el cuerpo calloso).

Método	**# Fibras**	**Fibras retenidas**
iFOD2	780	597
SD-Stream	780	571
Propuesto	780	**642**

significativas. En particular, la tractografía generada con el algoritmo propuesto retuvo una mayor proporción de fibras, lo cual podría indicar una mejor coherencia con los datos de difusión.

Adicionalmente, se calcularon métricas cuantitativas para evaluar la similitud entre las matrices de conectividad estimadas y la matriz de referencia proporcionada por el DiSCo Challenge. Las métricas consideradas incluyen la correlación de Pearson (r), el AUC y la exactitud, esta última evaluada considerando un umbral de conectividad del 5 %.

Los resultados en la Tabla 2 muestran una correlación positiva significativa en todos los niveles de ruido, con valores de r superiores a 0.83, lo cual indica una correspondencia estructural consistente entre las matrices estimadas y la matriz de referencia. Asimismo, los valores de AUC cercanos a 0.95 en todos los casos sugieren una buena capacidad del algoritmo para discriminar entre conexiones presentes y ausentes. En cuanto a la exactitud, los valores absolutos son moderados.

Un aspecto central en la evaluación del algoritmo propuesto es determinar su comportamiento frente a distintas condiciones de ruido y la sensibilidad respecto a los parámetros de construcción de fibras, en particular el tamaño del cono (cone_size) y el ángulo de apertura (θ_{tol}). La Tabla 3 resume los valores de las métricas de desempeño en función de dichos parámetros, considerando diferentes niveles de SNR. Los resultados muestran que, de manera general, tanto la correlación r como el AUC presentan valores más altos para configuraciones intermedias de parámetros, mientras que valores extremos tienden a deteriorar el rendimiento.

5 Conclusiones

El presente trabajo se centró en el desarrollo y la evaluación preliminar de un algoritmo de tractografía cerebral que integra el marco de optimización convexa COMMIT de manera temprana en la construcción de fibras. El objetivo principal fue explorar si esta estrategia permite reducir falsos positivos y recuperar falsos negativos desde la fase inicial de la generación de trayectorias, en lugar de aplicar la validación únicamente como un paso posterior. Los resultados presentados demuestran el potencial del enfoque planteado. Si bien el desempeño aún no alcanza niveles competitivos, especialmente en términos de exactitud, las mejoras observadas en correlación y AUC respaldan la hipótesis de que la validación temprana con COMMIT puede convertirse en una herramienta prometedora para incrementar la precisión de las tractografías.

Tabla 2: Métricas de evaluación cuantitativa por nivel de SNR.

SNR	**Correlación de Pearson (r)**	**AUC**	**Exactitud (umbral 5 %)**
10	0.8460	0.9505	0.2833
20	0.8325	0.9406	0.2917
30	0.8715	0.9499	0.2833
50	**0.8819**	**0.9543**	**0.3160**

Tabla 3: Sensibilidad del algoritmo propuesto según parámetros de construcción (cone_size y θ_{tol}) en diferentes niveles de SNR. Se muestran los valores de correlación r, AUC y exactitud.

		cone_size = 5				cone_size = 7				cone_size = 10			
Métrica	SNR	90°	120°	150°	170°	90°	120°	150°	170°	90°	120°	150°	170°
r	10	0.854	0.853	0.840	0.847	0.865	0.834	0.817	0.808	0.834	0.827	0.759	0.777
r	20	0.844	0.834	0.820	0.811	0.850	0.816	0.790	0.798	0.833	0.806	0.758	0.762
r	30	0.885	0.881	0.867	0.868	0.866	0.881	0.858	0.847	0.870	0.872	0.862	0.853
r	50	0.901	0.893	0.887	0.888	0.893	0.900	0.902	0.892	0.887	0.894	0.886	0.880
AUC	10	0.959	0.951	0.948	0.947	0.968	0.967	0.944	0.929	0.952	0.944	0.905	0.912
AUC	20	0.964	0.975	0.978	0.970	0.977	0.972	0.972	0.976	0.959	0.950	0.961	0.952
AUC	30	0.946	0.947	0.936	0.938	0.946	0.948	0.941	0.936	0.946	0.939	0.946	0.935
AUC	50	0.968	0.967	0.965	0.966	0.967	0.967	0.976	0.974	0.969	0.971	0.979	0.972
exactitud	10	0.333	0.342	0.350	0.333	0.350	0.383	0.383	0.392	0.383	0.417	0.442	0.467
exactitud	20	0.358	0.383	0.375	0.375	0.358	0.408	0.383	0.400	0.408	0.442	0.450	0.483
exactitud	30	0.325	0.342	0.342	0.342	0.325	0.350	0.417	0.408	0.400	0.408	0.442	0.492
exactitud	50	0.383	0.383	0.400	0.400	0.417	0.417	0.417	0.433	0.375	0.425	0.475	0.492

El método propuesto presenta como principal limitación su dependencia de una tractografía inicial, lo cual puede afectar directamente el desempeño global. No obstante, los hallazgos obtenidos sientan las bases para futuras líneas de investigación orientadas a refinar el método, optimizar su implementación y ampliar su validación en escenarios más cercanos a la práctica clínica.

Referencias

[1] R. Aranda, M. Rivera, and A. Ramirez-Manzanares. A flocking based method for brain tractography. *Medical Image Analysis*, 18(3):515–530, 2014.

[2] D. B. Aydogan and Y. Shi. Parallel transport tractography. *IEEE Transactions on Medical Imaging*, 40(2):635–647, 2021.

[3] T. E. Behrens, H. J. Berg, S. Jbabdi, M. F. Rushworth, and M. W. Woolrich. Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *Neuroimage*, 34(1):144–155, 2007. Epub 2006 Oct 27.

[4] R. L. Bishop. There is more than one way to frame a curve. *The American Mathematical Monthly*, 82(3):246–251, Mar. 1975.

[5] M. Boudreau, A. Karakuzu, J. Cohen-Adad, E. Bozkurt, M. Carr, M. Castellaro, L. Concha, M. Doneva, S. A. Dual, A. Ensworth, A. Foias, V. Fortier, R. E. Gabr, G. Gilbert, C. K. Glide-Hurst, M. Grech-Sollars, S. Hu, O. Jalnefjord, J. Jovicich, K. Keskin, P. Koken, A. Kolokotronis, S. Kukran, N. G. Lee, I. R. Levesque, B. Li, D. Ma, B. Mädler, N. G. Maforo, J. Near, E. Pasaye, A. Ramirez-Manzanares, B. Statton, C. Stehning, S. Tambalo, Y. Tian, C. Wang, K. Weiss, N. Zakariaei, S. Zhang, Z. Zhao, N. Stikov, and the ISMRM Reproducible Research Study Group and the ISMRM Quantitative MR Study Group . Repeat it without me: Crowdsourcing the t1 mapping common ground via the ismrm reproducibility challenge. *Magnetic Resonance in Medicine*, 92(3):1115–1127, 2024.

[6] A. Daducci, A. Dal Palù, A. Lemkaddem, and J.-P. Thiran. Commit: Convex optimization modeling for microstructure informed tractography. *IEEE Transactions on Medical Imaging*, 34(1):246–257, 2015.

[7] F. Dell'Acqua, M. Descoteaux, and A. Leemans. *Handbook of Diffusion MR Tractography: Imaging Methods, Biophysical Models, Algorithms and Applications*. Elsevier, Academic Press, Amsterdam, Netherlands, 2024.

[8] J. C. Fernández-Miranda, S. Pathak, J. Engh, K. Jarbo, T. Verstynen, F.-C. Sí, Y. Wang, A. Mintz, F. Boada, W. Schneider, and R. Friedlander. Tractografía de fibra de alta definición del cerebro humano: Validación neuroanatómica y aplicaciones neuroquirúrgicas. *Neurocirugía*, 71(2):430–453, Aug 2012.

[9] G. Girard, D. B. Aydogan, F. Dell'Acqua, A. Leemans, M. Descoteaux, and S. N. Sotiropoulos. Chapter 14 - probabilistic tractography. In F. Dell'Acqua, M. Descoteaux, and A. Leemans, editors, *Handbook of Diffusion MR Tractography*, pages 257–274. Academic Press, 2025.

[10] G. Girard, J. Rafael-Patiño, R. Truffet, D. B. Aydogan, N. Adluru, and et al. Tractography passes the test: Results from the diffusion-simulated connectivity (disco) challenge. *NeuroImage*, 277:120231, 2023.

[11] G. Girard, K. Whittingstall, R. Deriche, and M. Descoteaux. Towards quantitative connectivity analysis: Reducing tractography biases. *NeuroImage*, 98:266–278, 2014.

[12] D. K. Jones, T. R. Knösche, and R. Turner. White matter integrity, fiber count, and other fallacies: The do's and don'ts of diffusion mri. *NeuroImage*, 73:239–254, 2013.

[13] A. Leemans, F. Dell'Acqua, and M. Descoteaux. Chapter 13 - deterministic fiber tractography. In F. Dell'Acqua, M. Descoteaux, and A. Leemans, editors, *Handbook of Diffusion MR Tractography*, pages 241–255. Academic Press, 2025.

[14] F. Pestilli, J. D. Yeatman, A. Rokem, K. N. Kay, and B. A. Wandell. Evaluation and statistical inference for human connectomes. *Nature Methods*, 11(10):1058–1063, 2014.

[15] L. B. Reid, M. I. Cespedes, and K. Pannek. How many streamlines are required for reliable probabilistic tractography? solutions for microstructural measurements and neurosurgical planning. *NeuroImage*, 211:116646, 2020.

[16] E. Renauld, A. Théberge, L. Petit, et al. Validate your white matter tractography algorithms with a reappraised ismrm 2015 tractography challenge scoring system. *Scientific Reports*, 13:2347, 2023.

[17] K. G. Schilling, F. Rheault, L. Petit, C. B. Hansen, V. Nath, F.-C. Yeh, G. Girard, M. Barakovic, J. Rafael-Patino, T. Yu, E. Fischi-Gomez, M. Pizzolato, M. Ocampo-Pineda, S. Schiavi, E. J. Canales-Rodríguez, A. Daducci, C. Granziera, G. Innocenti, J.-P. Thiran, L. Mancini, S. Wastling, S. Cocozza, M. Petracca, G. Pontillo, M. Mancini, S. B. Vos, V. N. Vakharia, J. S. Duncan, H. Melero, L. Manzanedo, E. Sanz-Morales, Ángel Peña-Melián, F. Calamante, A. Attyé, R. P. Cabeen, L. Korobova, A. W. Toga, A. A. Vijayakumari, D. Parker, R. Verma, A. Radwan, S. Sunaert, L. Emsell, A. De Luca, A. Leemans, C. J. Bajada, H. Haroon, H. Azadbakht, M. Chamberland, S. Genc, C. M. Tax, P.-H. Yeh, R. Srikanchana, C. D. Mcknight, J. Y.-M. Yang, J. Chen, C. E. Kelly, C.-H. Yeh, J. Cochereau, J. J. Maller, T. Welton, F. Almairac, K. K. Seunarine, C. A. Clark, F. Zhang, N. Makris, A. Golby, Y. Rathi, L. J. O'Donnell, Y. Xia, D. B. Aydogan, Y. Shi, F. G. Fernandes, M. Raemaekers, S. Warrington, S. Michielse, A. Ramírez-Manzanares, L. Concha, R. Aranda, M. R. Meraz, G. Lerma-Usabiaga, L. Roitman, L. S. Fekonja, N. Calarco, M. Joseph, H. Nakua, A. N. Voineskos, P. Karan, G. Grenier, J. H. Legarreta, N. Adluru, V. A. Nair, V. Prabhakaran, A. L. Alexander, K. Kamagata, Y. Saito, W. Uchida, C. Andica, M. Abe, R. G. Bayrak, C. A. G. Wheeler-Kingshott, E. D'Angelo, F. Palesi, G. Savini, N. Rolandi, P. Guevara, J. Houenou, N. López-López, J.-F. Mangin, C. Poupon, C. Román, A. Vázquez, C. Maffei, M. Arantes, J. P. Andrade, S. M. Silva, V. D. Calhoun, E. Caverzasi, S. Sacco, M. Lauricella, F. Pestilli, D. Bullock, Y. Zhan, E. Brignoni-Perez, C. Lebel, J. E. Reynolds, I. Nestrasil, R. Labounek, C. Lenglet, A. Paulson, S. Aulicka, S. R. Heilbronner, K. Heuer, B. Q. Chandio, J. Guaje, W. Tang, E. Garyfallidis, R. Raja, A. W. Anderson, B. A. Landman, and M. Descoteaux. Tractography dissection variability: What happens when 42 groups dissect 14 white matter bundles on the same dataset? *NeuroImage*, 243:118502, 2021.

[18] J. Shapey, S. Vos, T. Vercauteren, R. Bradford, S. Saeed, S. Bisdas, and S. Ourselin. Clinical applications for diffusion mri and tractography of cranial nerves within the posterior fossa: A systematic review. *Frontiers in Neuroscience*, 13:23, Feb. 2019.

[19] R. E. Smith, J.-D. Tournier, F. Calamante, and A. Connelly. Sift: Spherical-deconvolution informed filtering of tractograms. *Neuroimage*, 67:298–312, 2013.

[20] H. Zhang, T. Schneider, C. A. Wheeler-Kingshott, and D. C. Alexander. NODDI: practical in vivo neurite orientation dispersion and density imaging of the human brain. *NeuroImage*, 61(4):1000–1016, 2012.

Información General

¿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 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

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