SMCCA

AÑO VIII - NÚMERO 8
DICIEMBRE 2022

BOLETÍN

SOCIEDAD MEXICANA DE COMPUTACIÓN CIENTÍFICA Y SUS APLICACIONES

BOLETÍN

SOCIEDAD MEXICANA DE COMPUTACIÓN CIENTÍFICA Y SUS APLICACIONES

AÑO VIII - NÚMERO 8

DICIEMBRE 2022

SMCCA

Comité Editorial

Editores Técnicos

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

El Boletín de la Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C. (SMCCA), Año VIII, No. 8, Diciembre 2022, 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 2022.

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

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

CONTENIDO

Artículos

Información

Carta de bienvenida

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

En esta octava edición del boletín se presentan: una breve semblanza de la XXX Escuela Nacional de Optimización y Análisis Numérico, llevada a cabo en conjunto con el Centro de Investigación en Matemáticas, CIMAT-Unidad Mérida, en modalidad híbrida del 4 al 8 de julio del presente año en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, en la Ciudad de Mérida, Yucatán, e incluye tres artículos de investigación, dos de ellos por solicitud y el otro corresponde al ganador de la vigésima edición del premio Mixbaal.

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

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

Reseña de la XXX Escuela Nacional de Optimización y Análisis Numérico

Justino Alavez Ramírez
Rina Betzabeth Ojeda Castañeda
Miguel Ángel Uh Zapata
Carlos Francisco Brito Loeza

XXX Escuela Nacional de Optimización y Análisis Numérico, ENOAN 2022
4 al 8 de Julio
30 años de la ENOAN: celebrando su pasado, presente y futuro
CONACYT

Del 4 al 8 de julio de 2022 se celebró con gran éxito la XXX edición de la Escuela Nacional de Optimización y Análisis Numérico (ENOAN 2022). En esta ocasión y debido a las restricciones impuestas por la pandemia COVID-19 se llevó a cabo en modalidad híbrida, siendo las sedes el Centro de Investigación en Matemáticas Unidad Mérida y la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, en la Ciudad de Mérida, Yucatán.

El comité organizador estuvo constituido por el Comité Local formado por profesores-investigadores, administradores y técnicos del Centro de Investigación en Matemáticas (CIMAT), CIMAT Unidad Mérida y de la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, y el Comité Nacional conformado por profesores-investigadores miembros en activo de la Sociedad Mexicana de Computación Científica y sus Aplicaciones (SMCCA).

Como en anteriores ediciones, las actividades académicas realizadas en esta ENOAN 2022 comprendieron: 10 cursos cortos, 6 conferencias plenarias y 22 conferencias invitadas a cargo de reconocidos investigadores tanto nacionales como internacionales, 6 trabajos de investigación en formato de cartel y 29 en formato de comunicación oral agrupados en 6 sesiones temáticas; adicionalmente se tuvieron 2 sesiones especiales: la sesión “Homenaje al Professor Roland Glowinski” y la sección "Primer Foro Conjunto de Sociedades Científicas". Al ser un evento híbrido, todas las actividades tanto presenciales como virtuales se trasmitieron en línea vía Zoom, salvo la exposición de los carteles en modalidad presencial. Las actividades virtuales también se proyectaron en la sala para los participantes presenciales.

En cuanto a los 10 cursos cortos, estos fueron de diferentes niveles: básico, intermedio y avanzado, y fueron impartidos por profesores-investigadores, de diferentes Instituciones de Educación Superior y Centros de Investigación del País. Se ofrecieron para alumnos de licenciatura y posgrado, así como a profesores y profesionales interesados, de distintas instituciones educativas del país y del extranjero, y con tópicos específicos, los cuales generalmente no son abordados con regularidad en sus instituciones. El nombre de los cursos impartidos, así como la asistencia promedio en cada uno de ellos se presenta en la Tabla 1.

Tabla 1: Cursos impartidos en la XXX ENOAN.
Las grabaciones de estos cursos se encuentran en
https://www.youtube.com/channel/UC0rIXGJ1Qx7OVURphWTykXQ

CURSOS NIVEL BÁSICO:	Asistencia promedio
Programación de un algoritmo de optimización utilizando bibliotecas numéricas en un ambiente Linux Dr. Ricardo Legarda Saenz, UADY.	8
Introducción al aprendizaje de datos: un enfoque de cómputo científico en la nube a través del lenguaje R Dr. José Refugio Reyes Valdés, CIMA-UAdeC. MIA. José Luis Fraga Almanza, UAdeC.	18
Resolviendo rompecabezas de lógica mediante programación matemática Dr. Jonás Velasco Álvarez, CIMAT Aguascalientes.	14
Introductorio de Criptografía Cuántica M. en C. Claudio Francisco Nebbia Rubio, FC-UNAM. M. en C. Karen Elizabeth Galindo Schembri, FC-UNAM.	21
CURSOS NIVEL INTERMEDIO:	**Asistencia Promedio**
Introducción a filtros de Kalman Dr. Arturo Espinosa Romero, UADY.	15
Modelos epidemiológicos sin y con retraso Dr. Benito Chen Charpentier, University of Texas at Arlington.	9
Herramientas de productividad para Ciencias de Datos Dr. Juan Pablo Soto Barrera, UNISON. Dr. Julio Waissman Vilanova, UNISON.	15
CURSOS NIVEL AVANZADO:	**Asistencia Promedio**
Solución numérica de ecuaciones diferenciales con discontinuidades Dr. Reymundo Ariel Itzá Balam, CIMAT - Mérida. Dr. Miguel Ángel Uh Zapata, CIMAT - Mérida.	21

Estimación de parámetros en EDO en dinámica de enfermedades virales Dr. Justino Alavez Ramírez, UJAT.	10
De dinámica de poblaciones, emergencia de patrones y Ondas Viajeras Dr. Faustino Sánchez Garduño, FC-UNAM.	11

En cuanto a las 6 conferencias plenarias, la conferencia inaugural "Diego Bricio", estuvo a cargo del reconocido Profesor-Investigador, el Dr. Pedro Flores Pérez, profesor jubilado de la Universidad de Sonora, la conferencia "Cátedra Humberto Madrid" estuvo a cargo del reconocido Profesor-Investigador, el Dr. Faustino Sánchez Garduño, de la Facultad de Ciencias de la Universidad Nacional Autónoma de México y las cuatro conferencias restantes estuvieron a cargo de los reconocidos profesores investigadores: el Professor Richard A. Tapia de Rice University, Dr. Humberto Madrid de la Vega, profesor jubilado de la Universidad Autónoma de Coahuila, Dr. Jesús López Estrada de la Facultad de Ciencias de la Universidad Nacional Autónoma de México, Dr. Luis B. Morales Mendoza del IIMAS-UNAM Mérida, Dr. Gilberto Calvillo Vives del IMATE Cuernavaca.

La presentación de trabajos en forma de cartel se realizó a través de la exposición de seis carteles sometidos y aprobados por la comisión de evaluación de trabajos, dos en modalidad virtual expuestos en la red social https://www.facebook.com/SMCCA.org.mx y cuatro en modalidad presencial. La presentación de los 29 trabajos en forma de comunicación oral, agrupados en 6 sesiones temáticas, se llevó a cabo en modalidad híbrida y transmitidas a través de las aulas virtuales de la plataforma Zoom.

Las 6 conferencias invitadas de la escuela, estuvieron a cargo de la Dra. Úrsula X. Iturrarán Viveros de la Facultad de Ciencias de la UNAM, Dr. Erick Treviño Aguilar del IMATE Unidad Cuernavaca, Dr. Carlos Francisco Brito Loeza de la Universidad Autónoma de Yucatán, Dr. Marcos Aurelio Capistrán Oampo del CIMAT, Dr. Fernando Brambila Paz de la Facultad de Ciencias de la UNAM y Dr. Jonathan Montalvo Urquizo de Modeling Optimization and Computing Technology SAS de C.V.

En cuanto a la sesión especial "Homenaje al Professor Roland Glowinski", coordinada por el Dr. Héctor Juárez Valencia del Depto. de Matemáticas de la UAM Unidad Iztapalapa, se contó con la participación del Professor Richard A. Tapia de Rice Univesity, Dr. Enrique Fernández-Cara de la Universidad de Sevilla, España, Professor Tsorng-Whay Pan de University of Houston, Dra. Patricia Saavedra Barrera de la UAM Unidad Iztapalapa, Dr. José Jacobo Oliveros Oliveros de la FCFM-BUAP, Dr. Jorge López López de la UJAT, Dra. Susana Gómez Gómez del IIMAS-UNAM, Dr. Francisco Javier Sánchez Bernabé de la UAM Unidad Iztapalapa, Dra. Diana Assaely León Velasco de la UAM Unidad Cuajimalpa y Dr. Héctor Juárez Valencia de la UAM Unidad Iztapalapa.

En la sesión especial "Conferencias del Primer Foro Conjunto de Sociedades Científicas", coordinada por el Dr. Justino Alavez Ramírez de la SMCCA, se contó con la participación de conferencistas de cinco sociedades: Dra. Yasmín Águeda Ríos Solís de la Sociedad Matemática Mexicana, Dr. Luis Fernando Morales Mendoza de la Sociedad Mexicana de Investigación de Operaciones, Dr. Pedro González-Casanova de SIAM Sección México, Dr. Felipe Javier Medina Aguayo de la Asociación Mexicana de Estadística, y la conferencia plenaria del Dr. Gilberto Calvillo Vives de la SMCCA.

Toda la información correspondiente al programa de la XXX ENOAN se puede consultar en la liga https://www.smcca.org.mx/Boletínes/Cuaderno_de_Resumenes_ENOAN_2022.pdf.

Los vídeos de los cursos y conferencias se pueden accesar desde la página https://www.smcca.org.mx/Productos_y_Resultados_2022, dando clic en las ligas de Youtube o Facebook o bien directamente en: https://www.youtube.com/channel/UC0rIXGJ1Qx7OVURphWTykXQ, https://www.facebook.com/enoan.

Ganadores del décimo noveno Premio Mixbaal

Desde 2002, dentro del marco de la ENOAN, se instituyó el "Premio Mixbaal a la Mejor Tesis de Licenciatura en Matemáticas Aplicadas", cuya convocatoria está dirigida a los egresados de carreras de matemáticas y áreas afines. El ganador de la vigésima edición de este premio, evaluado en la ENOAN 2022, es:

Ganador del Premio Mixbaal

Análisis del Proceso Espacio-Temporal de los Incendios Forestales en el Estado de México.

Luis Ramón Munive Hernández
Lic. en Estadística
Universidad Autónoma de Chapingo
Director: Dr. Antonio Villanueva Morales

El ganador expuso su trabajo en la ENOAN 2022 como ponente invitado.

Concurso de Carteles

Continuando con la tradición de las anteriores ediciones de la ENOAN, se realizó el concurso de carteles, en el cual una vez que los participantes exponen ante el jurado y el público su trabajo, el jurado lleva a cabo una evaluación para decidir la premiación. En esta ocasión, al ser una edición híbrida (presencial y virtual), el procedimiento que se estableció fue que los participantes en modalidad presencial expusieron su cartel ante el jurado y el público en forma oral; los participantes en modalidad virtual enviaron su cartel en formato PDF junto con un video grabado de 10 minutos exponiendo su trabajo, mismos que están expuestos en la red social https://www.facebook.com/SMCCA.org.mx. El jurado, integrado por el Dr. Justino Alavez Ramírez (Coordinador), Dr. Pedro Eduardo Miramontes Vidal y el Dr. José Luis Batún Cutz, revisó cada uno de los videos y los carteles en formato PDF de los participantes en modalidad virtual y analizó la exposición de los participantes en modalidad presencial, y con base en la Convocatoria emitida para tal fin y bajo los Criterios de Evaluación: Calidad Científica, Impacto Visual, Creatividad y Originalidad, los Miembros del Jurado revisaron reservada y libremente la totalidad de los carteles presentados, dictaminando lo siguiente:

Mejor Cartel Nivel Licenciatura

Ruteo de entrega de productos para una empresa de e-commerce.

Echeagaray González Juan Pablo
Emily Rebeca Méndez Cruz
Verónica Victoria García de la Fuente
Carolina Longoria Lozano
Instituto Tecnológico y de Estudios Superiores de Monterrey

Mejor Cartel Nivel Maestría

Análisis numérico del rango de operación de un motor de ignición por carga homogénea con recirculación de gas de la combustión.

América Eileen Mendoza Rojas
Juan Manuel García Guendulain
Rodrigo Hernández Alvarado
Universidad Politécnica de Querétaro

Cobertura de las actividades

De acuerdo al registro oficial del Certificado de Inscripción a la ENOAN 2022, se contó con la asistencia de 119 personas, de las cuales un 27.73% pertenecía al género femenino y un 72.27% al género masculino. Cabe aclarar que en la cédula de registro, se consideró la opción de "Prefiero no contestar", pero no se obtuvo ningún registro con esa respuesta, por lo que no aparece en la Gráfica 1.

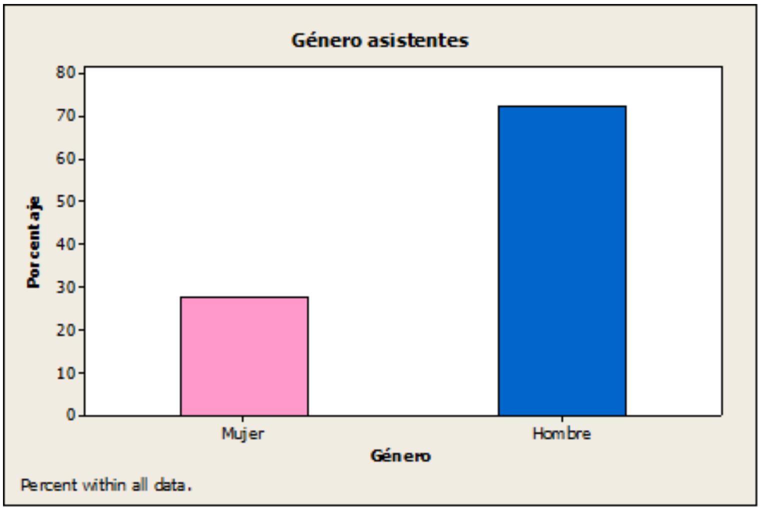

Figura 1. Gráfica de porcentajes del género de los asistentes a la XXX ENOAN

Los intervalos de edades que se establecieron en la cédula de registro fueron de los 18 años a más de 64 años, que eran las edades probables de la audiencia para este tipo de evento (jóvenes, adultos y adultos mayores), y los resultados que se obtuvieron con respecto a la edad se presentan en la Tabla 2 y Figura 2. Se puede observar que el mayor número de personas se dio en el rango de 25 a 34 años (27.74%), seguido por el rango de 35 a 44 años con el 23.54% y el rango de 18 a 24 años con 21.0%, siendo el menor número de asistentes en las edades adultas de 55 a 64 años con 6.72%. Es interesante ver que hay una participación de aproximadamente del 21% de jóvenes que desde sus primeros semestres de la licenciatura estén decidiendo participar en la ENOAN.

Intervalos de edades	Porcentaje de asistentes
18 a 24	21
25 a 34	27.74
35 a 44	23.54
45 a 54	10.08
55 a 64	6.72
> 64	10.92
TOTAL	100

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

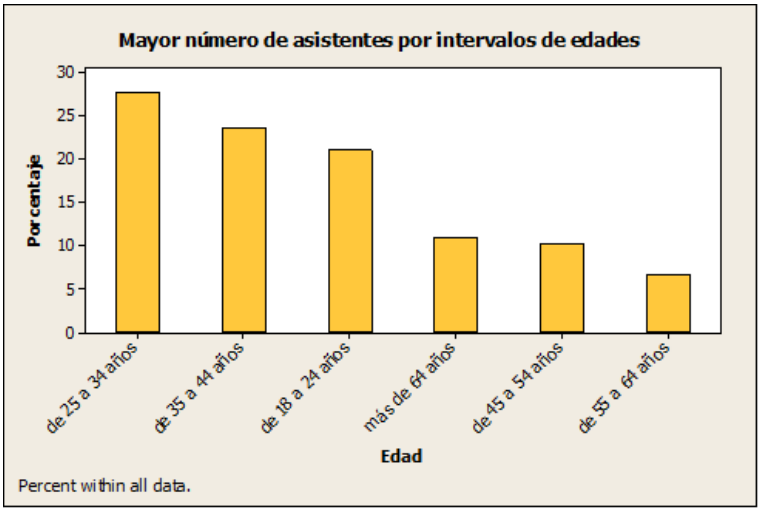

Figura 2: Gráfica de porcentajes por intervalo de edades de mayor a menor de los participantes.

En la Figura 3 se muestran las frecuencias cruzadas edad-género, en la que se muestra que en todos los rangos de edad de 18 a 64 años y más, se tuvo mayor porcentaje de asistencia de hombres que de mujeres, siendo una diferencia menor de participación de género en el intervalo de edades de 18 a 24 años.

A continuación se muestra en la Figura 4, los porcentajes de asistentes en cada una de las dos modalidades (presencial y virtual) y en la Figura 5 se presenta una gráfica cruzada de porcentaje de asistentes con respecto al género y la modalidad seleccionada por género.

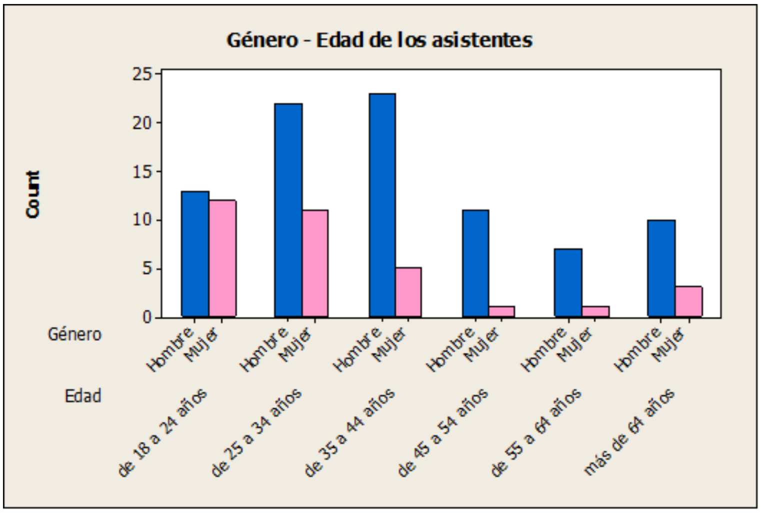

Figura 3: Gráfica cruzada género-edad (por intervalo de edades).

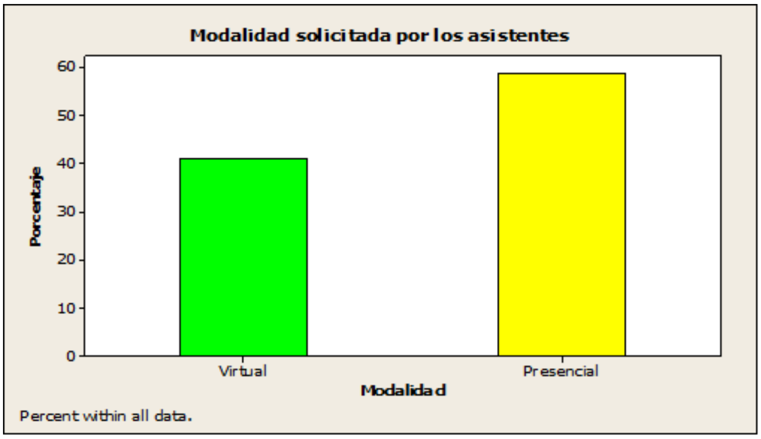

Figura 4: Gráfica de porcentaje de selección de modalidad de los asistentes.

Se tuvieron asistentes de diversos países de América (Norte y Sur) y Europa como se muestra en la Figura 6. El 95.8% manifestó tener nacionalidad mexicana y el resto 4.2% extranjera (2 de USA, 1 de España, 1 de Francia y 1 de Guatemala). Del 95.8% que indicó ser de nacionalidad mexicana, como se puede observar en la Figura 6 y en la Figura 7, que el mayor porcentaje de asistentes 22.7% radica en la Ciudad de México, seguido por el estado de Yucatán, Tabasco, Nuevo León y Coahuila.

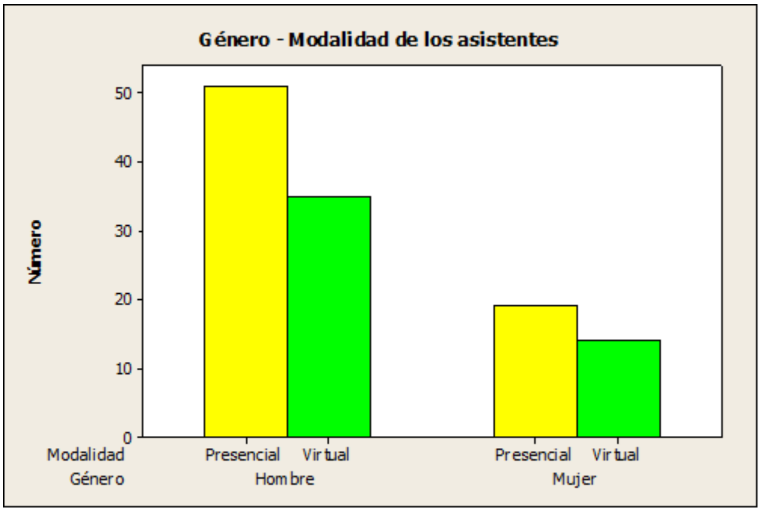

Figura 5: Gráfica cruzada género-modalidad del evento.

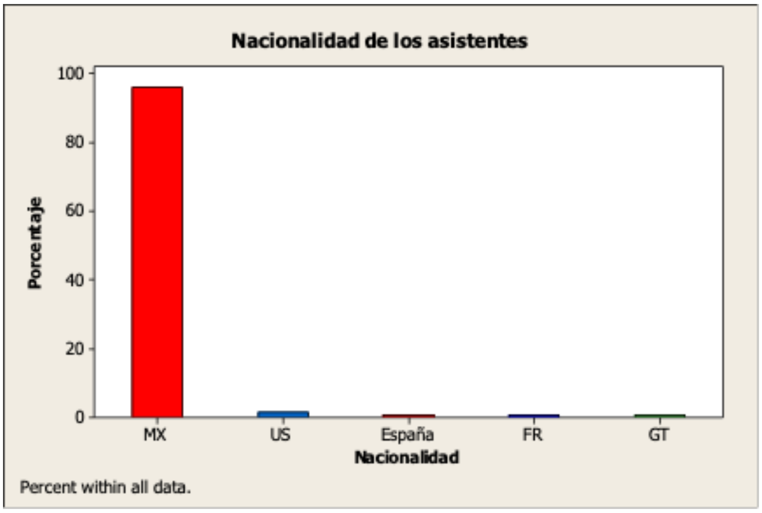

Figura 6: Nacionalidad de los participantes: mexicana: 95.8%, USA: 1.68%, España: 0.84%, Francia: 0.84% y Guatemala: 0.84%.

Por lo que respecta a la identificación de pertenencia de los asistentes a alguna etnia, la respuesta se puede ver en la Figura 8; el 89.2% respondió no identificarse con ninguna, 3.61% con los Mayas, 2.41% con los Zapotecas, 1.20% con Mixtecos, 2.41% con Afromexicano y 1.20% indicó ser Mestizo.

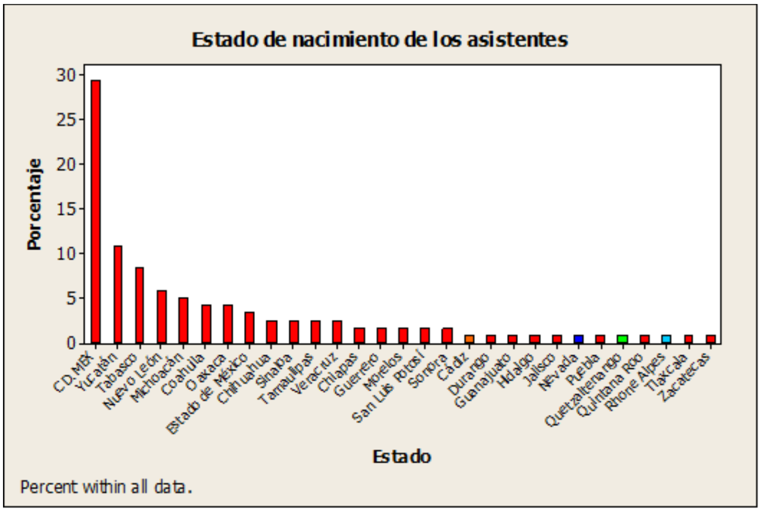

Figura 7: Lugar en el que manifiestan radicar los asistentes.

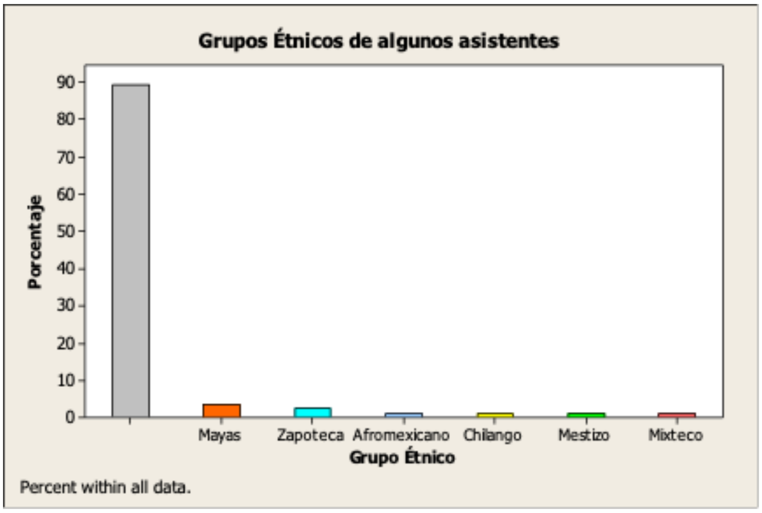

Figura 8: Grupos étnicos con los que se identifican los asistentes.

En la Figura 9 se presenta el porcentaje del tipo de ocupación que tenían los participantes al evento. Se puede observar que el 47.9% eran alumnos, el 31.09% profesores-investigadores, el 10.08% profesores, el 9.24% investigadores y el 1.68% otra ocupación.

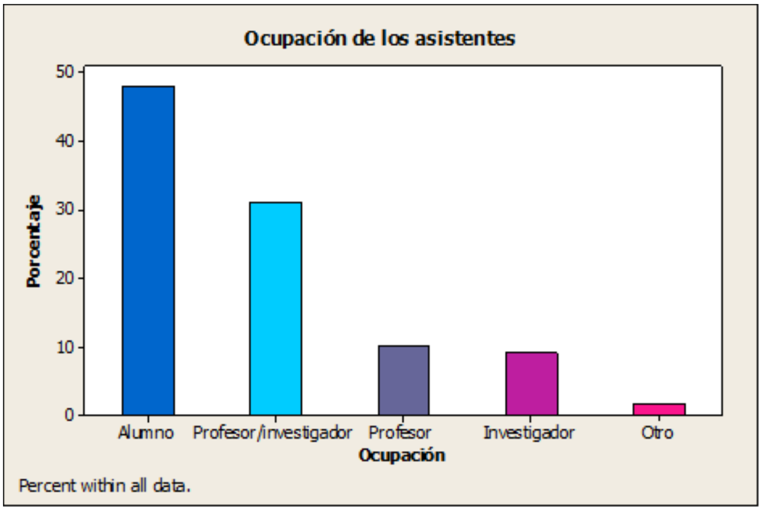

Figura 9: Gráfica de porcentajes de la ocupación de los participantes al evento.

Es importante señalar que gracias a que el evento se realizó en modalidad híbrida (presencial y vía remota), en esta edición de la ENOAN se tuvo una asistencia virtual de más del 40% de los participantes. Por otro lado, se puede observar que como era de esperarse el mayor número de asistentes tenía nacionalidad mexicana (95.8%).

Consideraciones finales

Por último es de gran importancia señalar que el gran esfuerzo de trabajo realizado tanto por el Comité Local (Centro de Investigación en Matemáticas, Centro de Investigación en Matemáticas Unidad Mérida y Universidad Autónoma de Yucatán) como el Comité Nacional (aglutinados dentro de la SMCCA) en la organización, y contando con el importante apoyo financiero de Instituciones, Dependencias y Centros de Investigación como: CONACYT, la Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C., el Centro de Investigación en Matemáticas, el Centro de Investigación en Matemáticas Unidad Mérida, la Facultad de Matemáticas de la Universidad Autónoma de Yucatán y el Departamento de Matemáticas de la Universidad de Sonora; permitió obtener un conjunto de resultados a beneficio de una comunidad científica conformada por alumnos, profesores, investigadores y profesionales interesados en la Computación Científica y las Matemáticas Aplicadas, que incidieron en indicadores de impacto como los que se presentan en la Tabla 3.

Indicador	Cantidad
Total de asistentes registrados:	119
Programas Académicos beneficiados por el evento:	35
Cuerpos Académicos o Grupos de Investigación beneficiados:	15
Cursos cortos impartidos:	10

Indicador	Cantidad
Conferencias (Diego Bricio, Humberto Madrid, Plenarias e Invitadas):	26
Ponencias por solicitud:	29
Carteles expuestos:	6
Estudiantes beneficiados:	57
Investigadores y Docentes beneficiados:	62
Mujeres beneficiadas:	33
Hombres beneficiados:	86
Número de Instituciones Nacionales participantes:	52
Número de Instituciones Internacionales participantes:	6
Integrantes del Comité Nacional:	11
Integrantes del Comité Local:	13
Personal de Apoyo Sede:	14
Aula Magna para inauguración, conferencias plenarias e invitadas:	1
Salas de cómputo equipados con cañón proyector para cursos:	4
Salones equipados con cañón proyector y pizarrón para ponencias:	3
Mamparras para carteles:	7
Licencias de uso de aulas virtuales de la plataforma ZOOM:	1

Tabla 3: Indicadores de impacto.

Finalmente, se muestran algunas imágenes de las actividades desarrolladas por los participantes de la ENOAN 2022.

XXX
ENOAN 2022
Bienvenidos

Zoom Reunión
María Luisa Sandoval Solís
Aula C9
Gerardo Tinoco Guerrero
Roxana Gongora
Aula C9

Stabilizing a Josephson Junction Array Memory around an unstable equilibrium: A control approach using a first-order state model

Jorge López López[1], Lorenzo Héctor Juárez Valencia[2], and Roland Glowinski[3]

[1]Universidad Juárez Autónoma de Tabasco
[2]Universidad Autónoma Metropolitana, Iztapalapa
[3]Department of Mathematics, University of Houston

Abstract

In this paper we consider a system of three nonlinear ordinary differential equations that model a three Josephson Junctions Array (JJA). Our goal is to stabilize the system around an unstable equilibrium employing an optimal control approach. We first define the cost functional and calculate its differential by using perturbation analysis and variational calculus. For the computational solution of the optimality system we consider a conjugate gradient algorithm for quadratic functionals in Hilbert spaces, combined with a finite difference discretization of the involucrated differential equations.

Keywords: Josephson Junction, Cryogenic Memory, Conjugate Gradient Algorithms, Stabilization, Optimal Control.

1 Introduction

In this article we discuss the numerical simulation of a control approach to stabilize a Josephson Junction Array (JJA) around an unstable steady state. Our methodology relies significantly on a conjugate gradient algorithm operating in a well-chosen control space. It has been shown recently that such an array can carry out the functions of a memory cell operations [2, 3, 6, 10, 11], consequently this array is frequently referred as a Josephson junction array memory (JJAM).

A Josephson junction is a quantum interference device that consists of two super-conductors coupled by a weak link that may be, for example, an insulator or a ferromagnetic material. A Josephson junction can carry current without resistance (this current is called supercurrent) and such a device is an example of a macroscopic quantum phenomenon. In 1962, B.D. Josephson proposed the equations governing the dynamics of the Josephson junction effect, namely:

$$\begin{cases} V = \frac{\hbar}{2e}\frac{d\phi}{dt}, \\ I = I_c \sin\phi, \end{cases} \tag{1}$$

where $\hbar = h/2\pi$, h being the Planck constant, $-e$ is the electric charge of the electron, V and I are the voltage and current across the junction, respectively, ϕ is the phase difference across the junction, and, finally, the constant I_c is the critical current across the junction. The Josephson junction technology offers numerous applications and could potentially provide sound alternatives for computer memory devices. In a 2005 US National Security Agency (NSA) report it has been alluded that, as transistors were rapidly approaching their physical limits, their most likely successors would be cryogenic devices based on Josephson junctions [1]. Indeed, (cf.

[7]), "Single flux quantum (SFQ) circuits produce small current pulses that travel at about 1/3 the speed of light. Superconducting passive transmission lines (PTL) are also able to transmit the pulses with extremely low losses". These are currently the fastest switching digital circuits, since (cf. [8]) "Josephson junctions, the superconducting switching devices, switch quickly ($\sim$ 1 ps), dissipate very little energy per switch ($< 10^{-19}J$), and communicate information via current pulses that propagate over superconducting transmission lines nearly without loss". Recently it was suggested that a small array consisting of inductively coupled Josephson junctions possesses all the basic functions (WRITE, READ, RESET) of a memory cell ([3, 11] and the references therein). In the two above articles, stable zero-voltage states were identified and basic memory cell operations based on the transitions between the equilibrium states were identified.

The equations modeling the dynamics of the inductively coupled Josephson junction array can be written in the following dimensionless form ($t = 1$ corresponds to $4.15ps$)

$$\begin{cases} \dfrac{d^2\phi_1}{dt^2} + \gamma_1 \dfrac{d\phi_1}{dt} + \kappa_1(\phi_1 - \phi_2) + \sin\phi_1 = I_1, \\ \dfrac{d^2\phi_2}{dt^2} + \gamma_2 \dfrac{d\phi_2}{dt} + \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin\phi_2 = I_2, \\ \dfrac{d^2\phi_3}{dt^2} + \gamma_3 \dfrac{d\phi_3}{dt} + \kappa_2(\phi_3 - \phi_2) + \sin\phi_3 = I_3, \end{cases} \tag{2}$$

where $I_j = i_j + ad_j$, i_j being a direct current and ad_j an additional energy as a current pulse. Plausible values for the various quantities in the model are:

$$\gamma_1 = 0.7, \gamma_2 = 1.1, \gamma_3 = 0.7, i_1 = 1, i_2 = 0.8, i_3 = -1, \tag{3}$$

$$\kappa_1 = \kappa_2 = 0.1. \tag{4}$$

For those regimes where the second order derivatives can be neglected (meaning the junctions are (relatively) highly dissipative), (2) reduces to

$$\begin{cases} \gamma_1 \dfrac{d\phi_1}{dt} + \kappa_1(\phi_1 - \phi_2) + \sin\phi_1 = I_1, \\ \gamma_2 \dfrac{d\phi_2}{dt} + \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin\phi_2 = I_2, \\ \gamma_3 \dfrac{d\phi_3}{dt} + \kappa_2(\phi_3 - \phi_2) + \sin\phi_3 = I_3. \end{cases} \tag{5}$$

The "Read/Write" operations can be performed by applying appropriate Gaussian pulses ([3, 6, 10]) in order to drive the system from an equilibrium configuration to another one. In [5], we went beyond Gaussian pulses and investigated an optimal controllability approach in L^2 to perform these transitions between equilibrium configurations. These approach is closely related to the methodology discussed in [4] for systems modelled by partial differential equations.

For the driving currents and coupling parameters provided in (3) and (4), the ordinary differential equation systems (5) (and (2) with $\dfrac{d\boldsymbol{\phi}}{dt}(0) = \mathbf{0}$) have several steady state solutions, consisting of phase triplets. In many cases, the steady state phases are near the values $2\pi n_k$, for certain integers n_k, and, according to Braiman, et. al., ([3]), equilibrium junction phases can be defined by their offsets σ_k from the negative cosine function's minima, as shown in the following equation

$$\theta_k = 2\pi n_k + \sigma_k. \tag{6}$$

Following Braiman et al., ([3]), Table 1 includes all the possible triplets of stable steady state offsets, where $2 \geq n_1 \geq n_2 \geq n_3$. For the steady states, we assumed, without loss of generality, that $n_3 = 0$, since all the phases can be shifted by the same multiple of 2π. Also we include in Table 1 three unstable steady states, namely: $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$, $\{2, 1, 0(u)\}$, $\{2, 2, 0(u)\}$. Here the description of some unstable equilibria is artificial (in the sense that the terms $\sigma_i/2\pi$ are not small and its only purpose is to be consistent with the description of stable states).

For the set of dc currents and coupling parameters defined by (3) and (4), no stable steady state exists when the first junction phase is about 2π (respectively 4π) greater than both of the other two junction phases (see row

Table 1: Stable and unstable equilibrium configurations for the JJAM system defined by (2) and (3). The symbols (u) and (s) qualify the unstable and stable equilibria, respectively

n_1	n_2	n_3	$\sigma_1/2\pi$	$\sigma_2/2\pi$	$\sigma_3/2\pi$	θ_1	θ_3	θ_3
0	0	$0(s)$	0.1992	0.1187	-0.1552	1.2517	0.7458	-0.9752
1	0	$0(s)$			no	stable	equil	
1	1	$0(s)$	0.1810	0.0338	-0.0515	7.4207	6.4958	-0.3236
2	0	$0(s)$			no	stable	equil	
2	1	$0(s)$	0.0661	0.1164	-0.0437	12.9821	7.0148	-0.2746
2	2	$0(s)$	0.1671	-0.0443	0.0333	13.6151	12.2884	0.2092
1	0	$0(u)$	0.0908	0.3591	-0.4140	6.8539	2.2566	-2.6015
2	1	$0(u)$	0.1011	0.4539	-0.0125	13.2016	9.1355	-0.0786
2	2	$0(u)$	0.1576	-0.1029	-0.5938	13.5568	11.9196	-3.7436

where $\{n_1, n_2, n_3\} = \{1, 0, 0\}$ (respectively $\{n_1, n_2, n_3\} = \{2, 0, 0\}$)). Following Braiman, et. al., [3], for memory cell demonstration it is sufficient to manipulate the system within the particular sets of states, namely states $\{0, 0, 0\}$ and $\{1, 1, 0\}$. The circuit of three coupled Rapid Single Joint (RSJ) unit circuits associated with model (5) can operate as a memory cell if a set of operators can transition the system to clearly defined states and can output a signal that discriminates memory states. The value n_k as presented in equation (6), will describe the location of the k^{th} junction phase. When all three junction phases are in the same sinusoidal potential well, the system will be considered in the '0' state, $\{n_1, n_2, n_3\} = \{0, 0, 0\}$. When the phases of the first and second junctions are shifted to the next potential well (about 2π greater), the cell will be in the '1' state, $\{n_1, n_2, n_3\} = \{1, 1, 0\}$. These two states correspond to the first and third rows of Table 1. Figure 1 shows the phases of the junctions when the systems starts from zero initial conditions. The junctions settle into the steady state '0' where all phases are close to the same multiple of 2π. For convenience, in several of the following figures the phases are normalized by 2π and in all the graphs that include different colors, color blue, red and green is associated with junction 1, 2 and 3, respectively.

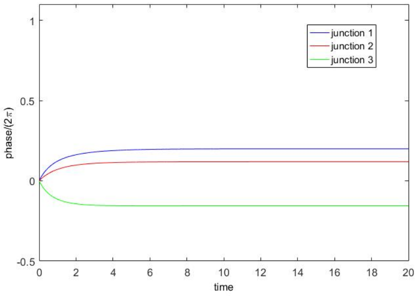

Figure 1: Time series of the phases scaled by 2π, with zero initial conditions.

In this work we investigate a controllability approach in order to stabilize the phase junctions of (5) around an unstable equilibrium. Figures 2 and 3 show the behavior of the system when the initial conditions are approximations of the unstable equilibrium $\{n_1, n_2, n_3\}$= $\{1, 0, 0(u)\}$, $\{2, 1, 0(u)\}$, $\{2, 2, 0(u)\}$.

The organization of this paper is as follows. In Section 2 we focus on a methodology to stabilize the JJAM system around an unstable equilibrium configuration, via a controllability approach. Section 3 is concerning the practical aspects of this methodology. In Section 4 we show the numerical results and in Section 5 we give the conclusions.

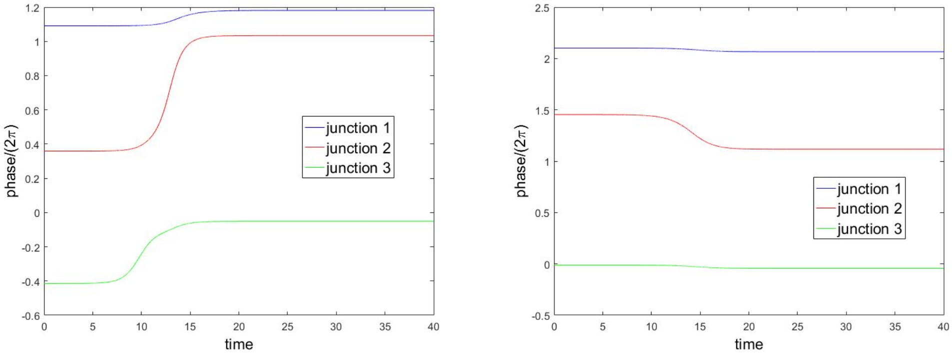

Figure 2: Evolution to state $\{1, 1, 0(s)\}$(left), $\{2, 1, 0(s)\}$(right) from an approximation to the unstable equilibrium $\{1, 0, 0(u)\}$(left), $\{2, 1, 0(u)\}$(right).

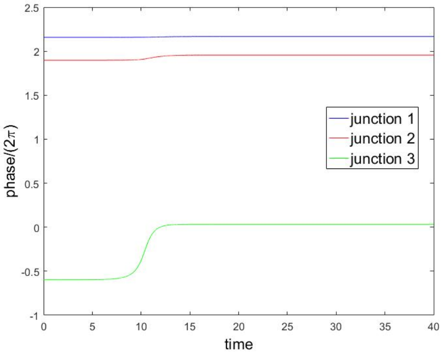

Figure 3: Evolution to $\{2, 2, 0(s)\}$ from an approximation to the unstable equilibrium $\{2, 2, 0(u)\}$.

2 Formulation of the optimal control problem for the stabilization of the JJAM and the conjugate gradient algorithm

2.1 The approach

As we explained in the Introduction, Figures 2 and 3 show the behavior of the system (5) when the initial conditions are approximations of the unstable equilibrium configurations $\{n_1, n_2, n_3\}= \{1, 0, 0(u)\}$, $\{2, 1, 0(u)\}$, $\{2, 2, 0(u)\}$. Our goal in this subsection is to stabilize the system (5), around an unstable equilibrium, controlling it via one, two or three junctions i.e., we want to maintain the phase values in Figures 2 and 3 almost constant along time. This constant value must be the initial value on each case. The approach taken here is the following classical one, namely,

(a) Linearize the model (5) in the neighborhood of an (unstable) equilibrium of the system.

(b) Compute an optimal control for the linearized model.

(c) Apply the above control to the nonlinear system.

Let us consider an unstable state $\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3\}$ of (5) and a small initial variation $\delta\boldsymbol{\theta}$ of $\boldsymbol{\theta}$. The perturbation $\delta\boldsymbol{\phi}$ of the steady-state $\boldsymbol{\theta}$ in $(0, T)$ satisfies approximately the following linear model

$$\left\{\begin{array}{c} \gamma_1 \dfrac{d\delta\phi_1}{dt} + \kappa_1(\delta\phi_1 - \delta\phi_2) + \delta\phi_1 \cos\theta_1 = 0, \\ \gamma_2 \dfrac{d\delta\phi_2}{dt} + \kappa_1(\delta\phi_2 - \delta\phi_1) + \kappa_2(\delta\phi_2 - \delta\phi_3) + \delta\phi_2 \cos\theta_2 = 0, \\ \gamma_3 \dfrac{d\delta\phi_3}{dt} + \kappa_2(\delta\phi_3 - \delta\phi_2) + \delta\phi_3 \cos\theta_3 = 0, \\ \delta\boldsymbol{\phi}(0) = \delta\boldsymbol{\theta}. \end{array}\right. \tag{7}$$

At least one of the eigenvalues of the jacobian of system (7) is positive. This means that the system can develop blow up phenomena (in infinite time). Figures 4, 5 and 6 show the solution of (7) for the three unstable equilibriums given by $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}, \{2, 1, 0(u)\}, \{2, 2, 0(u)\}$ in Table 1, respectively. The used value for the initial perturbation was $\delta\boldsymbol{\theta} = [1e-2, 1e-2, 1e-2]$.

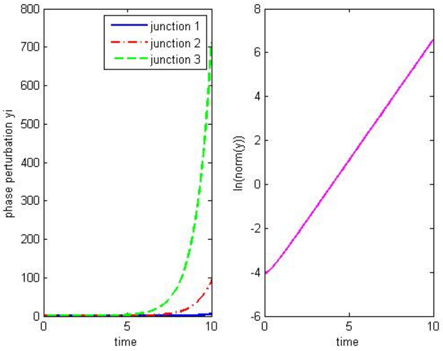

Figure 4: Solution $\mathbf{y} = \delta\boldsymbol{\phi}$ of (7) for the unstable equilibrium given by $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$ and $\delta\boldsymbol{\theta} = [1e-2, 1e-2, 1e-2]$.

It is clear that the model (7) is no longer valid if $\delta\boldsymbol{\theta}$ becomes too large but the idea behind considering the linearized model (7) is to use it to compute a control action preventing $\delta\boldsymbol{\phi}$ from becoming too large (and possibly driving $\delta\boldsymbol{\phi}$ to zero) and hope that the computed control will also stabilize the original nonlinear system (5) with initial condition

$$\boldsymbol{\phi}(0) = \boldsymbol{\theta} + \delta\boldsymbol{\theta}. \tag{8}$$

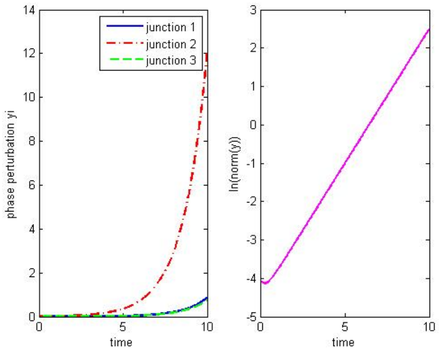

Figure 5: Solution $\mathbf{y} = \delta\boldsymbol{\phi}$ of (7) for the unstable equilibrium given by $\{n_1, n_2, n_3\} = \{2, 1, 0(u)\}$ and $\delta\boldsymbol{\theta} = [1e-2, 1e-2, 1e-2]$.

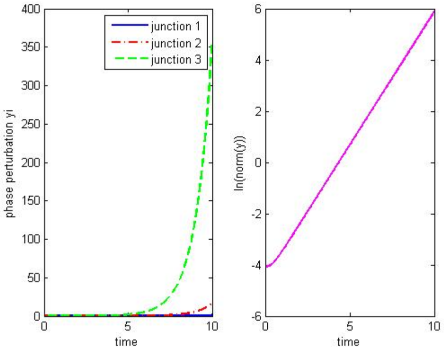

Figure 6: Solution $\mathbf{y} = \delta\boldsymbol{\phi}$ of (7) for the unstable equilibrium given by $\{n_1, n_2, n_3\} = \{2, 2, 0(u)\}$ and $\delta\boldsymbol{\theta} = [1e-2, 1e-2, 1e-2]$.

2.2 The Linear Control Problem

Using the notation $\mathbf{y} = \delta\boldsymbol{\phi}$, we shall (try to) stabilize the controlled variant of system (7) (the three junctions are used to control):

$$\begin{cases} \gamma_1 \dfrac{dy_1}{dt} + \kappa_1(y_1 - y_2) + y_1 \cos\theta_1 = v_1, \\ \gamma_2 \dfrac{dy_2}{dt} + \kappa_1(y_2 - y_1) + \kappa_2(y_2 - y_3) + y_2 \cos\theta_2 = v_2, \\ \gamma_3 \dfrac{dy_3}{dt} + \kappa_2(y_3 - y_2) + y_3 \cos\theta_3 = v_3, \\ \mathbf{y}(0) = \delta\boldsymbol{\theta}. \end{cases} \tag{9}$$

via the following control formulation:

$$\begin{cases} \mathbf{u} \in \mathcal{U}, \\ J(\mathbf{u}) \leq J(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}, \end{cases} \tag{10}$$

where

$$J(\mathbf{v}) = \frac{1}{2}\int_0^T ||\mathbf{v}||^2 dt + \frac{k_1}{2}\int_0^T ||\mathbf{y}||^2 dt + \frac{k_2}{2}||\mathbf{y}(T)||^2, \tag{11}$$

and $||\mathbf{y}||^2 = |y_1|^2 + |y_2|^2 + |y_3|^2$.

2.3 Optimality Conditions and Conjugate Gradient Solution for Problem (10)

2.3.1 Generalities and Synopsis

Let us denote by $DJ(\mathbf{v})$ the differential of J at $\mathbf{v} \in \mathcal{U} = L^2(0, T; 3) = (L^2(0, T))^3$. Since $\mathcal{U}$ is a Hilbert Space for the scalar product defined by

$$(\mathbf{v}, \mathbf{w})_{\mathcal{U}} = \int_0^T \mathbf{v}(t) \cdot \mathbf{w}(t)\ dt,$$

the action $\langle DJ(\mathbf{v}), \mathbf{w}\rangle$ of $DJ(\mathbf{v})$ on $\mathbf{w} \in \mathcal{U}$ can also be written as

$$\langle DJ(\mathbf{v}), \mathbf{w}\rangle = \int_0^T DJ(\mathbf{v})(t) \cdot \mathbf{w}(t)\ dt,\ \forall \mathbf{v}, \mathbf{w} \in \mathcal{U},$$

with $DJ(\mathbf{v})(t) \in \mathcal{U}$. If $\mathbf{u}$ is the solution of problem (10), it is characterized [from convexity arguments (see, e.g., [9])] by

$$DJ(\mathbf{u}) = \mathbf{0}. \tag{12}$$

Since the cost function J is quadratic and since the state model (7) is linear, $DJ(\mathbf{v})$ is in fact an affine function of $\mathbf{v}$, implying in turn from (12) that $\mathbf{u}$ is the solution of a linear equation in the control space $\mathcal{U}$. In abstract form, equation (12) can be written as

$$DJ(\mathbf{u}) - DJ(\mathbf{0}) = -DJ(\mathbf{0}),$$

and from the properties (need to be shown) of the operator $\mathbf{v} \to DJ(\mathbf{v}) - DJ(\mathbf{0})$, problem $DJ(\mathbf{u}) = \mathbf{0}$ could be solved by a (quadratic case)-conjugate gradient algorithm operating in the space $\mathcal{U}$. The practical implementation of the above algorithm would requires the explicit knowledge of $DJ(\mathbf{v})$.

2.3.2 Computing $DJ(v)$

We compute the differential $DJ(\mathbf{v})$ of the cost function J at $\mathbf{v}$ assumming that $\mathcal{U} = L^2(0, T; 3)$. To achieve that goal we will use a perturbation analysis. Let us consider thus a perturbation $\delta\mathbf{v}$ of the control variable $\mathbf{v}$. We have then,

$$\delta J(\mathbf{v}) = \int_0^T DJ(\mathbf{v}) \cdot \delta\mathbf{v} dt = \int_0^T \mathbf{v} \cdot \delta\mathbf{v} dt + k_1 \int_0^T \mathbf{y} \cdot \delta\mathbf{y} dt + k_2 \mathbf{y}(T) \cdot \delta\mathbf{y}(T), \tag{13}$$

where in (13):

(i) We denote by $a \cdot b$ the dot product of two vectors a and b.

(ii) The function $\delta\mathbf{y} = \{\delta y_1, \delta y_2, \delta y_3\}$ is the solution of the following initial value problem, obtained by perturbation of (9):

$$\begin{cases} \gamma_1 \dfrac{d\delta y_1}{dt} + \kappa_1(\delta y_1 - \delta y_2) + C_1 \delta y_1 = \delta v_1, \text{ in } (0,T), \\ \gamma_2 \dfrac{d\delta y_2}{dt} + \kappa_1(\delta y_2 - \delta y_1) + \kappa_2(\delta y_2 - \delta y_3) + C_2 \delta y_2 = \delta v_2, \text{ in } (0,T), \\ \gamma_3 \dfrac{d\delta y_3}{dt} + \kappa_2(\delta y_3 - \delta y_2) + C_3 \delta y_3 = \delta v_3, \text{ in } (0,T), \\ \delta\mathbf{y}(0) = \mathbf{0}, \end{cases} \tag{14}$$

where $C_1 = \cos\theta_1$, $C_2 = \cos\theta_2$, $C_3 = \cos\theta_3$. In matrix-vector form (14) reads as follows:

$$\begin{cases} \Gamma \dfrac{d}{dt}\delta\mathbf{y} + (K + C)\delta\mathbf{y} = \delta\mathbf{v}, \text{ in } (0,T), \\ \delta\mathbf{y}(0) = \mathbf{0}. \end{cases} \tag{15}$$

where C and Γ are diagonal matrices with coefficients C_i and γ_i, $i = 1, 2, 3$, respectively, and

$$K = \begin{pmatrix} \kappa_1 & -\kappa_1 & 0 \\ -\kappa_1 & \kappa_1 + \kappa_2 & -\kappa_2 \\ 0 & -\kappa_2 & \kappa_2 \end{pmatrix}.$$

We introduce now a vector-valued function $\mathbf{p} = \{p_1, p_2, p_3\}$ that we assume differentiable over $(0,T)$; multiplying by $\mathbf{p}$ both sides of the differential equation in (15), and integrating over $(0,T)$ we obtain, after integrating by parts, and taking into account the symmetry of the matrices Γ and K, the following equation

$$\Gamma\mathbf{p}(T) \cdot \delta\mathbf{y}(T) + \int_0^T \left\{ -\Gamma \frac{d}{dt}\mathbf{p} \ + (K + C)\mathbf{p} \right\} \cdot \delta\mathbf{y} \ dt = \int_0^T \mathbf{p} \cdot \delta\mathbf{v} \ dt, \text{ in } (0,T). \tag{16}$$

Now suppose that the vector-valued function $\mathbf{p}$ is solution of the following adjoint system:

$$\begin{cases} -\boldsymbol{\Gamma} \dfrac{d}{dt}\mathbf{p} \ + (K + C)\mathbf{p} \ = k_1 \mathbf{y}, \\ \boldsymbol{\Gamma}\mathbf{p}(T) = k_2 \mathbf{y}(T). \end{cases} \tag{17}$$

It follows from (16), (17) that

$$k_2 \mathbf{y}(T) \cdot \delta\mathbf{y}(T) + k_1 \int_0^T \mathbf{y} \cdot \delta\mathbf{y} \ dt = \int_0^T \mathbf{p} \cdot \delta\mathbf{v} \ dt. \tag{18}$$

Combining (13) and (18) we obtain that

$$\int_0^T DJ(\mathbf{v}) \cdot \delta\mathbf{v} dt = \int_0^T \mathbf{v} \cdot \delta\mathbf{v} dt + k_1 \int_0^T \mathbf{y} \cdot \delta\mathbf{y} dt + k_2 \mathbf{y}(T) \cdot \delta\mathbf{y}(T) = \int_0^T (\mathbf{v} + \mathbf{p}) \cdot \delta\mathbf{v} dt, \tag{19}$$

which implies in turn that

$$DJ(\mathbf{v}) = \mathbf{v} + \mathbf{p}. \tag{20}$$

Remark 1. *So far, we have been assuming that a control v_i is acting on the i-th junction. The method we used to compute DJ would still apply if one considers controlling the transition from $\mathbf{y}(0) = \delta\boldsymbol{\phi}(0) = \delta\boldsymbol{\theta}$ to $\mathbf{y}(T) = \delta\boldsymbol{\phi}(T) = \mathbf{0}$ using only one or only two controls. The same apply for the results in the next sections.*

2.3.3 Optimality conditions for problem (10)

Let $\mathbf{u}$ be the solution of problem (10) and let us denote by $\mathbf{y}$ (respectively, $\mathbf{p}$) the corresponding solution of the state system (9) (respectively, of the adjoint system (17)). It follows from Subsubsection 3.3.2 that $DJ(\mathbf{u}) = \mathbf{0}$ is equivalent to the following (optimality) system:

$$\mathbf{u} = -\mathbf{p}, \tag{21}$$

$$\Gamma\frac{d\mathbf{y}}{dt} + (K+C)\mathbf{y} = \mathbf{u}, \text{ in } (0,T), \tag{22}$$

$$\mathbf{y}(0) = \delta\boldsymbol{\theta}, \tag{23}$$

$$-\boldsymbol{\Gamma}\frac{d\mathbf{p}}{dt} + (K+C)\mathbf{p} = k_1\mathbf{y}, \text{ in } (0,T), \tag{24}$$

$$\boldsymbol{\Gamma}\mathbf{p}(T) = k_2\mathbf{y}(T). \tag{25}$$

Conversely, it can be shown (see, e.g., [9]) that the system (21)-(15) characterizes $\mathbf{u}$ as the solution (necessarily unique here) of the control problem (10). The optimality conditions (21)-(25) will play a crucial role concerning the iterative solution of the control problem (10).

2.3.4 Functional equations satisfied by the optimal control

Our goal here is to show that the optimality condition $DJ(\mathbf{u}) = \mathbf{0}$ can also be written as

$$\mathbf{A}\mathbf{u} = \boldsymbol{\beta}, \tag{26}$$

where the linear operator $\mathbf{A}$ is a strongly elliptic and symmetric isomorphism from $\mathcal{U}$ into itself (an automorphism of $\mathcal{U}$) and where $\boldsymbol{\beta} \in \mathcal{U}$. A candidate for $\mathbf{A}$ is the linear operator from $\mathcal{U}$ into itself defined by

$$\mathbf{A}\mathbf{v} = \mathbf{v} + \mathbf{p}(\mathbf{v}),$$

where $\mathbf{p}(\mathbf{v})$ is obtained from $\mathbf{v}$ via the successive solution of the following two problems:

$$\begin{cases} \Gamma\dfrac{d}{dt}\mathbf{y} + (K+C)\mathbf{y} = \mathbf{v}, \text{ in } (0,T), \\ \mathbf{y}(0) = \mathbf{0}, \end{cases} \tag{27}$$

which is forward in time, and the adjoint system (17) in $(0,T)$ which is backward in time. To see that operator $\mathbf{A}$ is a symmetric and strongly elliptic isomorphism from $\mathcal{U}$ into itself consider $\mathbf{v}^1$, $\mathbf{v}^2$ belonging to $\mathcal{U}$ and define $\mathbf{y}^i$, $\mathbf{p}^i$ by

$$\mathbf{y}^i = \mathbf{y}(\mathbf{v}^i), \ \mathbf{p}^i = \mathbf{p}(\mathbf{v}^i), \ i = 1,2;$$

we have then

$$\int_0^T (A\mathbf{v}^1)\cdot\mathbf{v}^2\, dt = \int_0^T (\mathbf{v}^1 + \mathbf{p}^1)\cdot\mathbf{v}^2\, dt = \int_0^T \mathbf{v}^1\cdot\mathbf{v}^2\, dt + \int_0^T \mathbf{p}^1\cdot\mathbf{v}^2\, dt. \tag{28}$$

On the other hand, we have, starting with the differential equation in (27) and using integration by parts, that

$$0 = \int_0^T \left(\Gamma\frac{d}{dt}\mathbf{y}^2 + (K+C)\mathbf{y}^2 - \mathbf{v}^2\right)\cdot\mathbf{p}^1\, dt$$

$$= \boldsymbol{\Gamma}\mathbf{p}^1(T)\cdot\mathbf{y}^2(T) + \int_0^T \left(-\Gamma\frac{d}{dt}\mathbf{p}^1 + (K+C)\mathbf{p}^1\right)\cdot\mathbf{y}^2 dt - \int_0^T \mathbf{v}^2\cdot\mathbf{p}^1 dt$$

$$= k_2\mathbf{y}^1(T)\cdot\mathbf{y}^2(T) + \int_0^T k_1\mathbf{y}^1\cdot\mathbf{y}^2 dt - \int_0^T \mathbf{v}^2\cdot\mathbf{p}^1 dt.$$

This implies that

$$\int_0^T \mathbf{v}^2\cdot\mathbf{p}^1 dt = k_2\mathbf{y}^1(T)\cdot\mathbf{y}^2(T) + \int_0^T k_1\mathbf{y}^1\cdot\mathbf{y}^2 dt. \tag{29}$$

Combining (28) with (29) we obtain

$$\int_0^T (\mathbf{A}\mathbf{v}^1)\cdot\mathbf{v}^2\ dt = \int_0^T \mathbf{v}^1\cdot\mathbf{v}^2\ dt + k_2\mathbf{y}^1(T)\cdot\mathbf{y}^2(T) + \int_0^T k_1\mathbf{y}^1\cdot\mathbf{y}^2 dt. \tag{30}$$

Relation (30) imply the symmetry of $\mathbf{A}$; we also have

$$\int_0^T (\mathbf{A}\mathbf{v})\cdot\mathbf{v}\ dt \geq \int_0^T |\mathbf{v}|^2\ dt,\ \forall\mathbf{v}\in\mathcal{U},$$

which implies the strong ellipticity of $\mathbf{A}$ over $\mathcal{U}$. The linear operator $\mathbf{A}$, being continuous and strongly elliptic over $\mathcal{U}$, is an automorphism of $\mathcal{U}$. To identify the right hand side $\boldsymbol{\beta}$ of equation (26), we introduce $\mathbf{Y}_0$ and $\mathbf{P}_0$ defined as the solutions of

$$\begin{cases} \Gamma\dfrac{d}{dt}\mathbf{Y}_0 + (K+C)\mathbf{Y}_0 = \mathbf{0},\text{ in } (0,T), \\ \mathbf{Y}_0(0) = \delta\boldsymbol{\theta}, \end{cases} \tag{31}$$

and

$$\begin{cases} -\boldsymbol{\Gamma}\dfrac{d}{dt}\mathbf{P}_0\ + (K+C)\mathbf{P}_0 = k_1\mathbf{Y}_0, \\ \boldsymbol{\Gamma}\mathbf{P_0}(T) = k_2\mathbf{Y}_0(T). \end{cases} \tag{32}$$

Suppose now that $\mathbf{y}$ and $\mathbf{p}$ satisfies the optimality conditions. Define

$$\overline{\mathbf{y}} = \mathbf{y} - \mathbf{Y}_0,$$
$$\overline{\mathbf{p}} = \mathbf{p} - \mathbf{P}_0.$$

Subtracting (31) and (32) from (27) and (17) we obtain

$$\begin{cases} \Gamma\dfrac{d}{dt}\overline{\mathbf{y}} + (K+C)\overline{\mathbf{y}} = \mathbf{u},\text{ in } (0,T), \\ \overline{\mathbf{y}}(0) = \mathbf{0}, \end{cases} \tag{33}$$

and

$$\begin{cases} -\boldsymbol{\Gamma}\dfrac{d}{dt}\overline{\mathbf{p}}\ + (K+C)\overline{\mathbf{p}} = k_1\overline{\mathbf{y}}, \\ \boldsymbol{\Gamma}\overline{\mathbf{p}}(T) = k_2\overline{\mathbf{y}}(T). \end{cases} \tag{34}$$

From the definition of operator $\mathbf{A}$ it follows that

$$\mathbf{A}\mathbf{u} = \mathbf{u}+\overline{\mathbf{p}} = -\mathbf{p}+\overline{\mathbf{p}} = -\mathbf{P}_0; \tag{35}$$

the right hand side of (35) is the vector $\boldsymbol{\beta}$ that we were looking for.

From the properties of $\mathbf{A}$, problem (35) can be solved by a conjugate gradient algorithm operating in the Hilbert space $\mathcal{U}$. This algorithm will be described in the following subsubsection.

2.3.5 Conjugate gradient solution of the control problem (10)

Problem (35) can be written in variational form as

$$\mathbf{u}\in\mathcal{U}\ (= L^2(0,T;3)), \tag{36}$$

From the symmetry and $\mathcal{U}$-ellipticity of the bilinear form

$$\{\mathbf{v},\mathbf{w}\}\to\int_0^T(\mathbf{A}\mathbf{v})\cdot\mathbf{w}\ dt,$$

the variational problem (36) is a particular case of the following class of linear variational problems:

$$\begin{cases} u\in\mathcal{V}, \\ a(u,v) = L(v),\forall v\in\mathcal{V}, \end{cases} \tag{37}$$

where

(i) $\mathcal{V}$ is a real Hilbert space for the scalar product $(\cdot,\cdot)$ and the correspondent norm $\|\cdot\|$;

(ii) $a:\mathcal{V}\times\mathcal{V}\to\mathbb{R}$ is bilinear, continuous, symmetric and $\mathcal{V}$-elliptic, i.e., $\exists\alpha>0$ such that $a(v,v)\geq\alpha\|v\|^2, \forall v\in\mathcal{V}$;

(iii) $L:\mathcal{V}\to\mathbb{R}$ is linear and continuous.

If the above properties hold, then problem (37) has a unique solution (see [4]); in fact, the symmetry of $a(\cdot,\cdot)$ is not necessary in order to have a unique solution to problem (27); however, the symmetry of $a(\cdot,\cdot)$, combined with the other properties, allows problem (37) to be solved by the following conjugate gradient algorithm:

- **Step 1**. $u^0\in\mathcal{V}$ is given.
- **Step 2**. Solve
$$\begin{cases} g^0\in\mathcal{V}, \\ (g^0,v)=a(u^0,v)-L(v),\forall v\in\mathcal{V}, \end{cases} \tag{38}$$
and set
$$w^0=g^0. \tag{39}$$
- **Step 3**. For $q\geq 0$, u^q, g^q and w^q being known, compute u^{q+1}, g^{q+1} and w^{q+1} as follows:
$$\rho^q=\|g^q\|^2/a(w^q,w^q), \tag{40}$$
and take
$$u^{q+1}=u^q-\rho^q w^q. \tag{41}$$
Solve
$$\begin{cases} g^{q+1}\in\mathcal{V}, \\ (g^{q+1},v)=(g^q,v)-\rho^q a(w^q,v),\forall v\in\mathcal{V}, \end{cases} \tag{42}$$
and compute
$$\gamma^q=\left\|g^{q+1}\right\|^2/\left\|g^q\right\|^2, \tag{43}$$
$$w^{q+1}=g^{q+1}+\gamma^q w^q. \tag{44}$$
- **Step 4.** Do n=n+1 and go to (40).

Let us recall that $\mathcal{U}=L^2(0,T;3)$ is a Hilbert space for the inner-product $\{\mathbf{v},\mathbf{w}\}\longrightarrow\int\limits_0^T\mathbf{v}\cdot\mathbf{w}dt$ and the associated norm $\mathbf{v}\longrightarrow\sqrt{\int\limits_0^T|\mathbf{v}|^2dt}$, implying that problem (10)-(36) can be solved applying the conjugate gradient algorithm (38)-(44). The above algorithm takes the following form:

- Suppose
$$\mathbf{u}^0 \text{ is given in } L^2(0,T;3)\ (\mathbf{u}^0=\mathbf{0} \text{ for example}). \tag{45}$$
- Solve
$$\begin{cases} \Gamma\frac{d}{dt}\mathbf{y}^0+(K+C)\mathbf{y}^0=\mathbf{u}^0, \text{ in } (0,T), \\ \mathbf{y}^0(0)=\delta\boldsymbol{\theta}. \end{cases}, \tag{46}$$
and then
$$\begin{cases} -\boldsymbol{\Gamma}\frac{d}{dt}\mathbf{p}^0+(K+C)\mathbf{p}^0=k_1\mathbf{y}^0, \text{ in } (0,T), \\ \boldsymbol{\Gamma}\mathbf{p}^0(T)=k_2\mathbf{y}^0(T). \end{cases} \tag{47}$$
- Set
$$\mathbf{g}^0=\mathbf{u}^0+\mathbf{p}^0. \tag{48}$$
- If $\int\limits_0^T|\mathbf{g}^0|^2dt\leq tol^2\max[1,\int\limits_0^T|\mathbf{u}^0|^2dt]$, take $\mathbf{u}=\mathbf{u}^0$; otherwise, set
$$\mathbf{w}^0=\mathbf{g}^0. \tag{49}$$

For $q \geq 0$, $\mathbf{u}^q$, $\mathbf{g}^q$ and $\mathbf{w}^q$ being known, we compute $\mathbf{u}^{q+1}$, $\mathbf{g}^{q+1}$ and if necessary, $\mathbf{w}^{q+1}$ as follows:

- Solve
$$\left\{ \begin{array}{c} \Gamma \frac{d}{dt}\overline{\mathbf{y}}^q + (K+C)\overline{\mathbf{y}}^q = \mathbf{w}^q \text{ in } (0,T), \\ \mathbf{y}^0(0) = \mathbf{0}. \end{array} \right. \tag{50}$$
and then
$$\left\{ \begin{array}{c} -\boldsymbol{\Gamma} \frac{d}{dt}\overline{\mathbf{p}}^q \ + (K+C)\overline{\mathbf{p}}^q = k_1\overline{\mathbf{y}}^q, \text{ in } (0,T), \\ \boldsymbol{\Gamma}\overline{\mathbf{p}}^q(T) = k_2\overline{\mathbf{y}}^q(T). \end{array} \right. \tag{51}$$
Set
$$\overline{\mathbf{g}}^q = \mathbf{w}^q + \overline{\mathbf{p}}^q, \tag{52}$$
and
$$\rho_q = \|\mathbf{g}^q\|^2 \,/ \int\limits_0^T \overline{\mathbf{g}}^q \cdot \mathbf{w}^q dt. \tag{53}$$
- Set
$$\mathbf{u}^{q+1} = \mathbf{u}^q - \rho_q \mathbf{w}^q. \tag{54}$$
- Set
$$\mathbf{g}^{q+1} = \mathbf{g}^q - \rho_q \overline{\mathbf{g}}^q. \tag{55}$$
- If $\frac{\int\limits_0^T \left|\mathbf{g}^{q+1}\right|^2 dt}{\max[\int\limits_0^T |\mathbf{g}^0|^2 dt, \int\limits_0^T |\mathbf{u}^{q+1}|^2 dt]} \leq tol^2$, take $\mathbf{u} = \mathbf{u}^{q+1}$; otherwise, compute
$$\gamma_q = \frac{\int\limits_0^T \left|\mathbf{g}^{q+1}\right|^2 dt}{\int\limits_0^T \left|\mathbf{g}^{q}\right|^2 dt}, \tag{56}$$
and set
$$\mathbf{w}^{q+1} = \mathbf{g}^{q+1} + \gamma_q \mathbf{w}^q. \tag{57}$$
- Do $q+1 \longrightarrow q$ and return to (50).
- End of algorithm.

The state variable $\mathbf{y}^q$ can be actualized simultaneously with the control $\mathbf{u}^q$ since for all q we have
$$\left\{ \begin{array}{c} \Gamma \frac{d}{dt}\mathbf{y}^q + (K+C)\mathbf{y}^q = \mathbf{u}^q, \text{ in } (0,T), \\ \mathbf{y}^q(0) = \delta\boldsymbol{\theta}. \end{array} \right. \tag{58}$$
so,
$$\left\{ \begin{array}{c} \Gamma \frac{d}{dt}(\mathbf{y}^{q+1} - \mathbf{y}^q) + (K+C)(\mathbf{y}^{q+1} - \mathbf{y}^q) = \mathbf{u}^{q+1} - \mathbf{u}^q = -\rho_q \mathbf{w}^q, \text{ in } (0,T), \\ (\mathbf{y}^{q+1} - \mathbf{y}^q)(0) = \mathbf{0}, \end{array} \right. \tag{59}$$
which, by definition of $\overline{\mathbf{y}}^q$, implies that
$$\mathbf{y}^{q+1} - \mathbf{y}^q = -\rho_q \overline{\mathbf{y}}^q \Longrightarrow \mathbf{y}^{q+1} = \mathbf{y}^q - \rho_q \overline{\mathbf{y}}^q.$$

The practical implementation of algorithm (45)-(57), via a finite difference discretization of problem (10), will be discussed in the following section.

3 Discrete formulation of the optimal control problem

3.1 Finite difference approximation of problem (10)

We approximate (10) when $\mathcal{U} = (L^2(0,T))^3$ by

$$\begin{cases} \mathbf{u}^{\Delta t} \in \mathcal{U}^{\Delta t}, \\ J^{\Delta t}(\mathbf{u}^{\Delta t}) \leq J^{\Delta t}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}^{\Delta t}, \end{cases} \tag{60}$$

where:

- $\Delta t = T/N$ with N a "large" positive integer.
- $\mathcal{U}^{\Delta t} = (\mathbb{R}^3)^N$.

The cost functional $J^{\Delta t}$ is defined by

$$J^{\Delta t}(\mathbf{v}) = \frac{\Delta t}{2}\sum_{n=1}^{N}||\mathbf{v}^n||^2 + \frac{k_1 \Delta t}{2}\sum_{n=1}^{N}||\mathbf{y}^n||^2 + \frac{k_2}{2}||\mathbf{y}^N||^2,$$

with $\mathbf{v} = \{\mathbf{v}^n\}_{n=1}^N$ and $\{\mathbf{y}^n\}_{n=1}^N$ obtained from $\mathbf{v}$ and $\delta\boldsymbol{\theta}$ via the following discrete variant of (9):

$$\mathbf{y}^0 = \delta\boldsymbol{\theta}, \tag{61}$$

and for $n = 1, ..., N$

$$\Gamma\frac{\mathbf{y}^n - \mathbf{y}^{n-1}}{\Delta t} + (K + C)\mathbf{y}^n = \mathbf{v}^n. \tag{62}$$

To compute $\mathbf{y}^n$ we have thus to solve a linear system of the following type:

$$(\Gamma + \Delta t(K + C))\mathbf{y}^n = RHS^n. \tag{63}$$

The matrix $\Gamma + \Delta t(K + C)$ being a 3×3 matrix symmetric and positive definite, solving (63) is easy.

3.2 Optimality Conditions and conjugate gradient solution of (60)

3.2.1 Computing $DJ^{\Delta t}(\mathbf{v})$

Assuming that one wants to use the conjugate gradient algorithm (38)-(44) to solve the discrete problem (60), we compute first $DJ^{\Delta t}(\mathbf{v})$. On $\mathcal{U}^{\Delta t} = (\mathbb{R}^3)^N$ we will use the following inner-product

$$(\mathbf{v}, \mathbf{w})_{\Delta t} = \Delta t \sum_{n=1}^{N} \mathbf{v}^n \cdot \mathbf{w}^n,\ \forall \mathbf{v} = \{\mathbf{v}^n\}_{n=1}^N,\ \mathbf{w} = \{\mathbf{w}^n\}_{n=1}^N \in (\mathbb{R}^3)^N.$$

We have

$$\delta J^{\Delta t}(\mathbf{v}) = \Delta t \sum_{n=1}^{N} \mathbf{v}^n \cdot \delta\mathbf{v}^n + k_1 \Delta t \sum_{n=1}^{N} \mathbf{y}^n \cdot \delta\mathbf{y}^n + k_2 \mathbf{y}^N \cdot \delta\mathbf{y}^N, \tag{64}$$

with $\{\delta\mathbf{y}^n\}_{n=0}^N$ obtained by perturbation of (61), (62), that is

$$\delta\mathbf{y}^0 = \mathbf{0}, \tag{65}$$

and for $n = 1, ..., N$

$$\Gamma\frac{\delta\mathbf{y}^n - \delta\mathbf{y}^{n-1}}{\Delta t} + (K + C)\delta\mathbf{y}^n = \delta\mathbf{v}^n. \tag{66}$$

Let us introduce $\{\mathbf{p}^n\}_{n=1}^N \in (\mathbb{R}^3)^N$. Taking the dot product of $\mathbf{p}^n$ with each side of equation (66), we obtain after summation and multiplication by Δt :

$$\Delta t \sum_{n=1}^{N} \Gamma\frac{\delta\mathbf{y}^n - \delta\mathbf{y}^{n-1}}{\Delta t} \cdot \mathbf{p}^n + \Delta t \sum_{n=1}^{N}(K + C)\delta\mathbf{y}^n \cdot \mathbf{p}^n = \Delta t \sum_{n=1}^{N} \delta\mathbf{v}^n \cdot \mathbf{p}^n. \tag{67}$$

Applying a discrete integration by parts to relation (67), and considering that $\delta \mathbf{y}^0 = \mathbf{0}$, we obtain:

$$\Gamma \mathbf{p}^{N+1} \cdot \delta \mathbf{y}^N + \Delta t \sum_{n=1}^{N} \Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} \cdot \delta \mathbf{y}^n + \Delta t \sum_{n=1}^{N} (K + C)\mathbf{p}^n \cdot \delta \mathbf{y}^n = \Delta t \sum_{n=1}^{N} \delta \mathbf{v}^n \cdot \mathbf{p}^n, \tag{68}$$

or equivalently

$$\Gamma \mathbf{p}^{N+1} \cdot \delta \mathbf{y}^N + \Delta t \sum_{n=1}^{N} \left\{ \Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + (K + C)\mathbf{p}^n \right\} \cdot \delta \mathbf{y}^n = \Delta t \sum_{n=1}^{N} \delta \mathbf{v}^n \cdot \mathbf{p}^n. \tag{69}$$

Suppose that $\{\mathbf{p}^n\}_{n=1}^{N+1}$ verifies the following discrete adjoint system:

$$\Gamma \mathbf{p}^{N+1} = k_2 \mathbf{y}^N, \tag{70}$$

and for $n = N, ..., 1$

$$\Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + (K + C)\mathbf{p}^n = k_1 \mathbf{y}^n. \tag{71}$$

It follows from (64), (69)-(71) that

$$\delta J^{\Delta t}(\mathbf{v}) = \Delta t \sum_{n=1}^{N} \mathbf{v}^n \cdot \delta \mathbf{v}^n + k_1 \Delta t \sum_{n=1}^{N} \mathbf{y}^n \cdot \delta \mathbf{y}^n + k_2 \mathbf{y}^N \cdot \delta \mathbf{y}^N = \Delta t \sum_{n=1}^{N} (\mathbf{v}^n + \mathbf{p}^n) \cdot \delta \mathbf{v}^n, \tag{72}$$

that is

$$DJ^{\Delta t}(\mathbf{v}) = \{\mathbf{v}^n + \mathbf{p}^n\}_{n=1}^{N}. \tag{73}$$

3.2.2 Optimality conditions for (60)

The optimality conditions for the discrete problem (60) are

$$\mathbf{u}^n = -\mathbf{p}^n,\ n = 1, ..., N, \tag{74}$$

$$\mathbf{y}^0 = \delta \boldsymbol{\theta}, \tag{75}$$

$$\Gamma \frac{\mathbf{y}^n - \mathbf{y}^{n-1}}{\Delta t} + (K + C)\mathbf{y}^n = \mathbf{u}^n \text{ in } (0, T),\ \ n = 1, ..., N, \tag{76}$$

$$\Gamma \mathbf{p}^{N+1} = k_2 \mathbf{y}^N, \tag{77}$$

$$\Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + (K + C)\mathbf{p}^n = k_1 \mathbf{y}^n,\ n = N, ..., 1. \tag{78}$$

3.2.3 Functional equation for the discrete control solution of (60)

Following the sketch for the continuous case we can show that the discrete version $\mathbf{A}^{\Delta t}$ of operator $\mathbf{A}$ and the discrete version $\boldsymbol{\beta}^{\Delta t}$ of $\boldsymbol{\beta}$ satisfies the equation

$$\mathbf{A}^{\Delta t} \mathbf{u}^{\Delta t} = \boldsymbol{\beta}^{\Delta t}, \tag{79}$$

where $\mathbf{u}^{\Delta t}$ is the discrete control satisfying the optimality condition (74). Operator $\mathbf{A}^{\Delta t}$ enjoys the same properties than the continuous version: symmetric, strongly elliptic and continuous, allowing us to use a conjugate gradient like (38)-(44) to solve (79).

3.2.4 Conjugate gradient solution of the discrete control problem (60)

Using $\mathbf{y}_q^n = \{y_{iq}^n\}_{i=1}^3$ to denote the discrete value of the vector-valued function $\mathbf{y}$ at time $n\Delta t$ and iteration q; similarly, $\mathbf{u}_q^n$ will denote the discrete value of the control $\mathbf{u}$ at time $n\Delta t$ and iteration q, the conjugate gradient algorithm (38)-(44) to solve the finite dimensional problem (60) reads as follow:

- Suppose

$$\mathbf{u}_0 = \{\{u_{i0}^n\}_{i=1}^3\}_{n=1}^N \text{ is given in } \mathcal{U}_{ad}^{\Delta t} = (\mathbb{R}^3)^N \ (\mathbf{u}_0 = \mathbf{0} \text{ for example}). \tag{80}$$

- Compute $\{\mathbf{y}_0^n\}_{n=0}^N = \{\{y_{i0}^n\}_{i=1}^3\}_{n=0}^N$ and $\{\mathbf{p}_0^n\}_{n=1}^{N+1} = \{\{p_{i0}^n\}_{i=1}^3\}_{n=1}^{N+1}$ via the solution of

$$\left\{\begin{array}{c} \mathbf{y}_0^0 = \delta\boldsymbol{\theta}, \\ \text{for } n = 1, ..., N \text{ solve} \\ \Gamma\dfrac{\mathbf{y}_0^n - \mathbf{y}_0^{n-1}}{\Delta t} + (K + C)\mathbf{y}_0^n = \mathbf{u}_0^n, \end{array}\right. \tag{81}$$

and then

$$\left\{\begin{array}{c} \Gamma\mathbf{p}^{N+1} = k_2\mathbf{y}^N, \\ \text{for } n = N, ..., 1 \text{ solve} \\ \Gamma\dfrac{\mathbf{p}_0^n - \mathbf{p}_0^{n+1}}{\Delta t} + (K + C)\mathbf{p}_0^n = k_1\mathbf{y}_0^n. \end{array}\right. \tag{82}$$

- Set

$$\mathbf{g}_0 = \{\mathbf{g}_0^n\}_{n=1}^N = \{\mathbf{u}_0^n + \mathbf{p}_0^n\}_{n=1}^N. \tag{83}$$

- If

$$\frac{\Delta t\sum_{n=1}^N |\mathbf{g}_0^n|^2}{\max[1, \Delta t\sum_{n=1}^N |\mathbf{u}_0^n|^2]} \leq tol^2, \text{ with } |\mathbf{g}_0^n|^2 = |g_{10}^n|^2 + |g_{20}^n|^2 + |g_{30}^n|^2,$$

take $\mathbf{u}^{\Delta t} = \mathbf{u}_0$; otherwise, set

$$\mathbf{w}_0 = \mathbf{g}_0. \tag{84}$$

For $q \geq 0$, $\mathbf{u}_q$, $\mathbf{g}_q$ and $\mathbf{w}_q$ being known, the last two different from $\mathbf{0}$, we compute $\mathbf{u}_{q+1}$, $\mathbf{g}_{q+1}$ and, if necessary $\mathbf{w}_{q+1}$ as follows:

- Solve

$$\left\{\begin{array}{c} \overline{\mathbf{y}}_q^0 = \mathbf{0}, \\ \text{for } n = 1, ..., N \text{ solve} \\ \Gamma\dfrac{\overline{\mathbf{y}}_q^n - \overline{\mathbf{y}}_q^{n-1}}{\Delta t} + (K + C)\overline{\mathbf{y}}_q^n = \mathbf{w}_q^n, \end{array}\right. \tag{85}$$

and then

$$\left\{\begin{array}{c} \Gamma\overline{\mathbf{p}}_q^{N+1} = k_2\overline{\mathbf{y}}_q^N, \\ \text{for } n = N, ..., 1 \text{ solve} \\ \Gamma\dfrac{\overline{\mathbf{p}}_q^n - \overline{\mathbf{p}}_q^{n+1}}{\Delta t} + (K + C)\overline{\mathbf{p}}_q^n = k_1\overline{\mathbf{y}}_q^n. \end{array}\right. \tag{86}$$

- Set

$$\overline{\mathbf{g}}_q = \mathbf{w}_q + \overline{\mathbf{p}}_q, \tag{87}$$

and

$$\rho_q = \Delta t\sum_{n=1}^N \left|\mathbf{g}_q^n\right|^2 / (\Delta t\sum_{n=1}^N \overline{\mathbf{g}}_q^n \cdot \mathbf{w}_q^n). \tag{88}$$

- Compute

$$\mathbf{u}_{q+1} = \mathbf{u}_q - \rho_q\mathbf{w}_q, \tag{89}$$

and

$$\mathbf{g}_{q+1} = \{\mathbf{g}_{q+1}^n\}_{n=1}^N = \mathbf{g}_q - \rho_q\overline{\mathbf{g}}_q. \tag{90}$$

- If

$$\frac{\Delta t\sum_{n=1}^N \left|\mathbf{g}_{q+1}^n\right|^2}{\max[\Delta t\sum_{n=1}^N |\mathbf{g}_0^n|^2, \Delta t\sum_{n=1}^N \left|\mathbf{u}_{q+1}^n\right|^2]} \leq tol^2,$$

take $\mathbf{u}^{\Delta t} = \mathbf{u}_{q+1}$; otherwise, compute

$$\gamma_q = \frac{\sum_{n=1}^N \left|\mathbf{g}_{q+1}^n\right|^2}{\sum_{n=1}^N \left|\mathbf{g}_q^n\right|^2}, \tag{91}$$

and set

$$\mathbf{w}_{q+1} = \mathbf{g}_{q+1} + \gamma_q\mathbf{w}_q. \tag{92}$$

- Do $q+1 \longrightarrow q$ and return to (85).
- End of algorithm

Similar to the continuous case, we can deduce that if $\mathbf{y}_q = \mathbf{y}(\mathbf{u}_q)$ then

$$\mathbf{y}_{q+1} - \mathbf{y}_q = -\rho_q \overline{\mathbf{y}}_q \Longrightarrow \mathbf{y}_{q+1} = \mathbf{y}_q - \rho_q \overline{\mathbf{y}}_q.$$

4 Numerical Results

In previous sections we have described a methodology and the respective practical algorithms to use a control on each joint in order to stabilize the linear JJAM perturbation system around an unstable equilibrium. It is easy to simplify the procedure and algorithm to the case when we want to control via only (any combina-tion of) two junctions or via only one junction. However, since (according to the experiments) it is necessary to control via at least two junctions in order to stabilize the system around an unstable equilibrium, we show only the results when two and three junctions are used to control the system model. For the calculations we used $tol^2 = 10^{-16}$ for the stopping criteria in conjugate gradient algorithm, and $\Delta t = 10^{-2}$ for solving the differential systems. In the next two subsections we use as θ the equilibrium given by $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$ and $\delta\boldsymbol{\theta} = [1e-2, 1e-2, 1e-2]$; the time interval under consideration being $[0, 2]$. Finally we apply iteratively this control process to stabilize in the longer interval $[0, T]$ using 10 subintervals of length 2.

4.1 Controlling via two junctions

When controlling via two junctions only the case when using junctions 2 and 3 is a successful one (for all values of k_1 and k_2). In Figures 7 and 8 are shown the respective results.

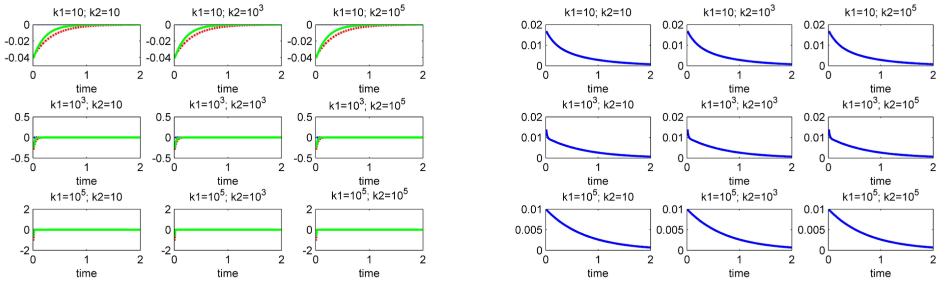

Figure 7: $\mathbf{u}^{\Delta t}$ (left) and $\|\mathbf{y}^{\Delta t}(\cdot)\|$ (right) for several values of k_1 and k_2. The unstable equilibrium $\boldsymbol{\theta}$ is given by $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. The junctions used to control are 2 and 3.

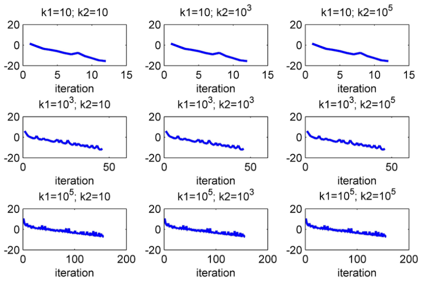

Figure 8: $Ln\|\mathbf{g}_q^{\Delta t}\|$ for several values of k_1 and k_2. The unstable equilibrium $\boldsymbol{\theta}$ is given by $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. The junctions used to control are 2 and 3.

4.2 Controlling via three junctions

Figures 9 and 10 show the results when the three junctions are used to control. As we can see, for all values of the penalty parameters, the linear system is controlled.

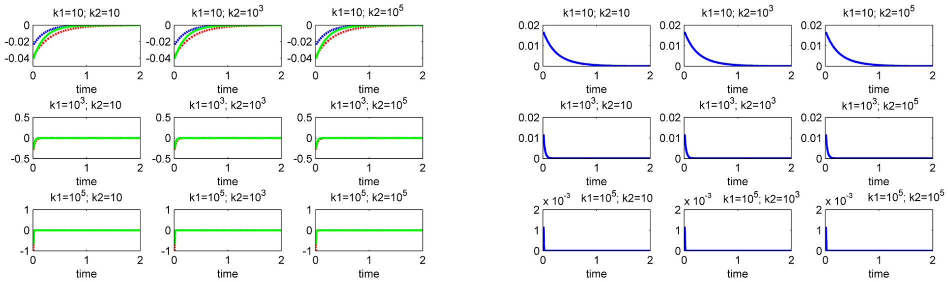

Figure 9: $\mathbf{u}^{\Delta t}$ (left) and $\|\mathbf{y}^{\Delta t}(\cdot)\|$ (right) for several values of k_1 and k_2. The unstable equilibrium $\boldsymbol{\theta}$ is given by $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. The junctions used to control are 1, 2 and 3.

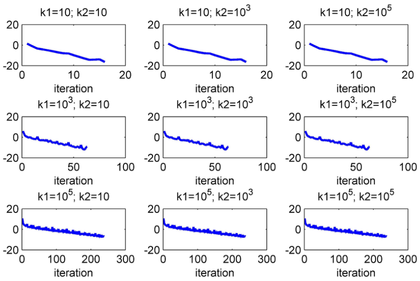

Figure 10: $Ln\|\mathbf{g}_q^{\Delta t}\|$ for several values of k_1 and k_2. The unstable equilibrium $\boldsymbol{\theta}$ is given by $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. The junctions used to control are 1, 2 and 3.

For the particular values $k_1 = 1E+3$ and $k_2 = 1E+3$ of the penalty parameters we show in Figure 11 the controls and the norm of the solution of the nonlinear system (the behavior of the solution of the linear system is shown in Figure 9).

In Figure 12 we show the solution for the linear and nonlinear model, using controls in Figure 11 (we continue from $t = 2$ to $t = 20$ with no control).

4.3 Controlling via three junctions in the interval [0,20]

To control during the whole time interval $(0, 20)$ we have divided the time interval into subintervals of smaller length $\Delta T = T/Q = 2$, and we denote $q\Delta T$ by T_q for $q = 1, ..., Q$; we proceed then as follows:

- For $q = 0$, we denote by $\mathbf{y}_0$ the difference $\boldsymbol{\phi}_0 - \boldsymbol{\theta}$, and we solve the associated linear control problem (9)-(11) in $[0, T_1]$; let us denote by $\mathbf{u}_1$ the corresponding control. This control is injected in (5) with initial condition (8) and $\delta\boldsymbol{\theta} = \mathbf{y}_0$, and we denote by $\mathbf{y}_1$ the difference $\boldsymbol{\phi}(T_1) - \boldsymbol{\theta}$.
- For $q > 0$, we denote by $\mathbf{y}_q$ the difference $\boldsymbol{\phi}(T_q) - \boldsymbol{\theta}$; we solve the associated linear control problem (9)-(11) in $[T_q, T_{q+1}]$, with $\mathbf{y}_0$ replaced by $\mathbf{y}_q$, and we denote by $\mathbf{u}_{q+1}$ the corresponding optimal control. The control $\mathbf{u}_{q+1}$ is injected in (5) with initial condition (8) and $\delta\boldsymbol{\theta} = \mathbf{y}_q$, and we denote by $\mathbf{y}_{q+1}$ the difference $\boldsymbol{\phi}(T_{q+1}) - \boldsymbol{\theta}$.
- We do $q = q + 1$ and we repeat the process.

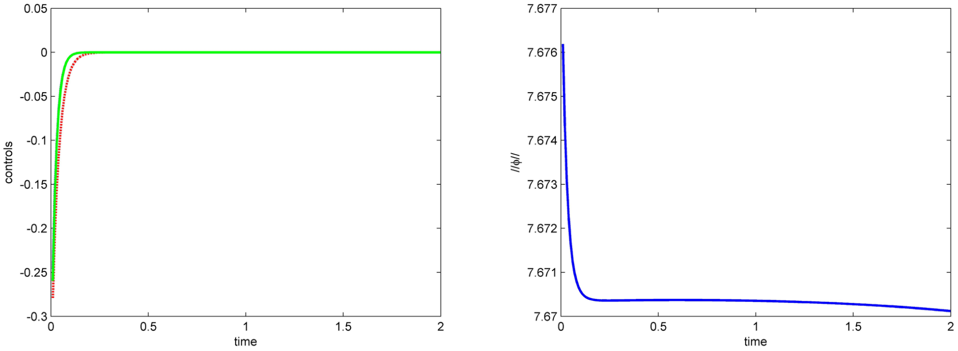

Figure 11: Optimal controls (left), and Euclidean norm of the controlled solution $\boldsymbol{\phi}^{\Delta t}$ of the nonlinear system (right).

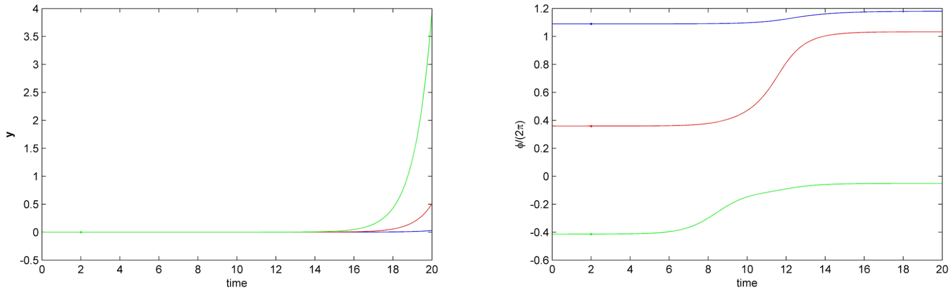

Figure 12: Extended controlled solution $y_i^{\Delta t}$ of the linear perturbation model (left), and extended controlled solution $\phi_i^{\Delta t}$ of the nonlinear system (right). After $t = 2$ the controls are extended as cero.

The above time partitioning method has been applied with $\boldsymbol{\phi}_0 = \boldsymbol{\theta} + \delta\boldsymbol{\theta}$ and $\delta\boldsymbol{\theta} = [1e-2, 1e-2, 1e-2]$, the time interval under consideration being $[0, 20]$; we have used $\Delta T = 2.0$. After $t = 20$, we have taken $\mathbf{v} = \mathbf{0}$ in (9) and in (5) to observe the evolution of the suddenly uncontrolled linear and nonlinear systems. The results are reported in Figure 13. We observe that the system is practically stabilized for $1 \leq t \leq 20$, but if one stops controlling, the small residual perturbations of the system at $t = 20$, are sufficient to destabilize the linear and nonlinear systems and induces the nonlinear one to transition to a stable equilibrium in finite time.

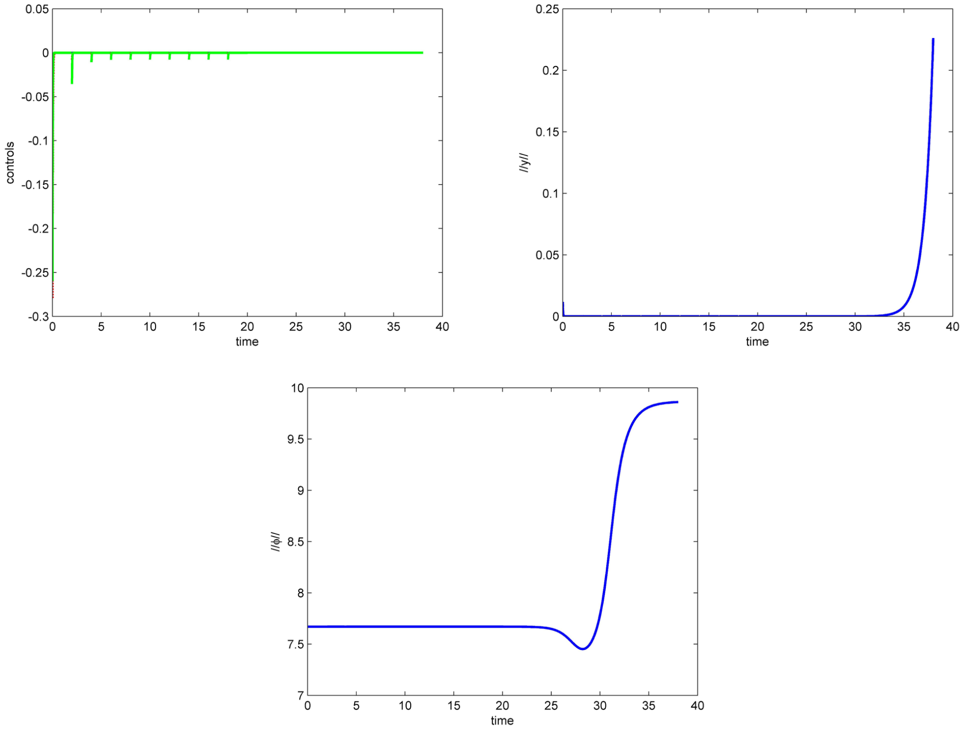

Figure 13: Controls calculated each two seconds (top-left); Euclidean norm of the controlled solution $\mathbf{y}^{\Delta t}$ using the controls in the top (top-right); Euclidean norm of the controlled solution $\boldsymbol{\phi}^{\Delta t}$ using the controls in the top-left (bottom).

5 Conclusions

We have stabilized the phases of a JJAM model, around an unstable equilibrium by using the classical approach: linearize the state model around the unstable equilibrium; control the linear model in order to stabilize it around the unstable equilibrium; apply the linear control to the nonlinear model and hope this control will also stabilize it. Since the time interval is large ($t \in [0, 30]$) in certain applications, and could be that the linear control do not stabilize the nonlinear model, we subdivi-ded the original interval into subintervals and calculate iteratively the linear control on each subinterval, obtaining a piecewise control that stabilize not only the linear model but also the nonlinear one. For an efficient calculation of the control we formulated an operational linear equation satisfied by the control. The associated operator is self-adjoint and elliptic, so a conjugate gradient algorithm for quadratic functional was used.

References

[1] F. Bedard, N. Welker, G. R. Cotter, M. A. Escavage, and J. T. Pinkston. Superconducting technology assessment. Technical report, National Security Agency, Office of Corporate Assessment, 2005.

[2] Y. Braiman, N. Nair, J. D. Rezac, and N. Imam. Memory cell operation based on small josephson junction arrays. *Superconductor Science and Technology*, 129(124003):1–15, 2016.

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

[4] R. Glowinski, J.-L. Lions, and J. He. *Exact and Approximate Controllability for Distributed Parameter Systems*. Cambridge University Press, 2008.

[5] R. Glowinski, J. López-López, H. Juarez-Valencia, and Y. Braiman. On the controllability of transitions between equilibrium states in small inductively coupled arrays of josephson junctions: A computational approach. *Journal of Computational Physics*, 403(109023), 2020.

[6] R. Harvey and Z. Qu. Control of cryogenic memory state transitions in a josephson junction array. In *2018 Annual American Control Conference (ACC)*, pages 5671–5676, 2018.

[7] F. A. Holmes, L. Ripple, and M. A. Manheimer. Energy-efficient superconducting computing-power budgets and requirements. *IEEE Transactions on Applied Superconductivity*, 23(3), 2013.

[8] IARPA. Broad agency announcement: Iarpa-baa-13-05, cryogenic computing complexity (c3) program. Technical report, IARPA, 2013.

[9] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer Verlag, 1971.

[10] N. Nair and Y. Braiman. A ternary memory cell using small josephson junction arrays. *Superconductor Science and Technology*, 31(115012), 2018.

[11] J. D. Rezac, N. Imam, and Y. Braiman. Superconductor science and technology. *PHYSICA A*, 427:267–281, 2017.

Parameter estimation in ODEs. Modelling and computational issues

Lorenzo Héctor Juárez Valencia[1] and Jessica Rojas[1]

[1]Universidad Autónoma Metropolitana, Iztapalapa

Abstract

In this work we discuss a variational approach for the determination of the parameters of systems of ordinary differential equations (ODE). We construct a model for fitting observed noisy data into the given dynamical system. Also we explain in detail the advantage of using the adjoint equation method to compute the derivatives or gradients, which are needed for the application of gradient methods and quasi-Newton algorithms to find the minimum of the cost function. In particular, we consider two classic iterative algorithms: the conjugate gradient (CG) algorithm and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. For educational purposes we try to explain several numerical and computational issues with some detail and illustrate them with the Susceptible, Exposed, Infected, Recovered, and Deceased (SEIRD) epidemiological model.

Keywords: Inverse problem, noisy data, parameter determination, adjoint method, variational approach.

1 Introduction

Systems of ordinary (and partial) differential equations are an important tool to model the physical state of a real phenomenon that arise in many areas of applied sciences and engineering. Predicting the future behavior or allowing control of these processes requires not only accurately describing the system but also finding or improving its parameters. Estimation of unknown or inaccurate parameters in turn requires fitting partially observed noisy data or experimental measurements to the model. More generally, in the statistics and machine learning literature various methods have been employed to fit differential equations to data, from maximum likelihood approaches, [13] to Bayesian sampling algorithms, [4] or traditional deterministic approaches. Thus, parameter estimation needs efficient ODE (forward) solvers, optimization routines, statistical and possible stochastic procedures. There are several stochastic, deterministic and hybrid optimization routines [3]. Contrary to stochastic algorithms, deterministic ones are computationally efficient but they tend to converge to local minima. However, they are the departure point in many applications and in the design of better and efficient procedures. For these methods the gradient is combined with information from line searches or methods involving a Newton, quasi-Newton (low-rank) or Fisher information based curvature estimators to update model parameters, [10]. The main computational bottleneck in these algorithms is the computation of the gradient (or the curvature) of the parametric cost function. Then, efficient methods to evaluate gradients or for parametric sensitivity analysis of differential-algebraic models is important, no only for the determination of parameters, but also in other areas of application like model simplification, data assimilation, optimal control, process sensitivity, uncertainty analysis, and experimental design, among others, for a wide range of scientific and engineering problems.

In this work we concentrate in deterministic optimization procedures, mainly those for quadratic non-linear programming and based on gradient methods and quasi-Newton algorithms. Our purpose is to explain in some detail modelling, algorithmic and computational issues to a wide, and possible non-expert, audience. In Section 2 we introduce the quadratic non-linear model that incorporates noisy data into the cost function and it is adapted to the SEIRD epidemiological deterministic model, which is employed to illustrate the algorithms and their related computational issues. Section 3 is devoted to explaining the adjoint equation method for computing gradients of the given cost function and its advantages over other methods. In section 4 we describe the CG and BFGS optimization algorithms and discuss important computational issues. Numerical results are shown in Section 5 and finally, in Section 6, we give some conclusions and perspectives for future work in order to improve the model and overcome some drawbacks and algorithmic problems.

2 The quadratic model

Let $\mathbf{x}(t) \in \mathbb{R}^d$ be a state variable at time $t \in I = [t_0, t_f]$ of a continuous time ordinary differential equation (ODE) satisfying the following initial value problem:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}), \qquad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{1}$$

where the upper dot on the left hand side denotes derivation with respect to time. The vector function $\mathbf{f} : \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^{np} \mapsto \mathbb{R}^d$ depends on the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{np}$ with np the number of parameters. The solution at time t of this problem is denoted by $\mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta})$ when its dependence of $\mathbf{x}_0$ and $\boldsymbol{\theta}$ is made explicit. Frequently, we use the short notation $\mathbf{x}(t)$ for simplicity, like in equation (1) above. A set of vector measurements at times $t_0 \leq t_1 < \ldots < t_m \leq t_f$ in I are available:

$$\mathbf{x}_i = \mathbf{x}(t_i; \mathbf{x}_0, \boldsymbol{\theta}) + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, m \tag{2}$$

where $\boldsymbol{\epsilon}_i \in \mathbb{R}^d$ are independent random vectors and represent measurement errors with a multivariate Gaussian distribution having zero mean and vector variances $\boldsymbol{\sigma}_i^2 \in \mathbb{R}^d$. In many problems not all components of the state $\mathbf{x}(t)$ are observable, so this vector variable is decomposed in observable variables $\overline{\mathbf{x}}$ and non-observable variables $\underline{\mathbf{x}}$, which can be regarded as orthogonal projections of $\mathbf{x}$ over the coordinates of these observable and non-observable variables. A model to estimate the unknown parameter vector $\boldsymbol{\theta}$ and the initial conditions $\mathbf{x}_0$, from the given measurements, relays on the minimization of the least squares objective function

$$\ell(\mathbf{x}_0, \boldsymbol{\theta}) = \frac{1}{2} \left\| \frac{\mathbf{x}_0 - \mathbf{s}_0}{\boldsymbol{\sigma}_0} \right\|_d^2 + \frac{1}{2} \sum_{i=1}^{m} \left\| \frac{\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta}) - \overline{\mathbf{x}}_i}{\overline{\boldsymbol{\sigma}}_i} \right\|_{no}^2, \tag{3}$$

where the state variable $\mathbf{x}(t)$ is subject to satisfy the ODE (1). The first norm is the euclidean norm in $\mathbb{R}^d$ and the norms into the sum are the euclidean norms in $\mathbb{R}^{no}$, $no =$ number of observable variables. The fixed vector $\mathbf{s}_0$ denotes an experimental measurement of the initial conditions.

Remark 1. *We assume that all components of the state variable are observable at the initial time t_0, otherwise the first term in (3) can be modified accordingly. The most used norms, in the construction of objective functions $\ell(\mathbf{x}_0, \boldsymbol{\theta})$, are those of the form $||\mathbf{x}||_p = (\sum_{j=1}^{d} |x_j|^p)^{1/p}$ with $p = 1$, $p = 2$, $p = \infty$. Choosing the most appropriate norm depends on the particular application and of the properties of the state variable. As mentioned before we use the usual euclidean norm, i.e. $p = 2$.*

Remark 2. *Observe that the quantities in (3) are vectors, so the quotients are computed component-wise. In general, if Σ_i is the covariance matrix at experimental time t_i, then*

$$||(\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta}) - \overline{\mathbf{x}}_i)/(\overline{\boldsymbol{\sigma}}_i)||_{no}^2 = (\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta}) - \overline{\mathbf{x}}_i)^T W_i (\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta}) - \overline{\mathbf{x}}_i)$$

where $W_i = \Sigma_i^{-1}$ is the precision matrix. If the symmetric matrix W is positive definite, then it defines a norm in the corresponding subspace of the observable variables, as $||\overline{\mathbf{x}}||_W = \overline{\mathbf{x}}^T W_i \overline{\mathbf{x}}$. Of course, it is possible to define the cost function (3) taking $\overline{\boldsymbol{\sigma}}_i$ as the unitary vector for every i, however the minimization process using iterative gradient methods converge faster when using the weights $\overline{\boldsymbol{\sigma}}_i$. The attraction ball around the global minimum is also larger in this case.

Example 1. *To illustrate the previous concepts and notation we consider the SEIRD epidemiological deterministic model that describe the dynamics of the propagation of an infections disease, like COVID-19, [12]. If we have a closed constant population of N individuals, at each time t a compartmental model separates this population* *in five segments: susceptible, exposed, infectious, recovered and dead, denoted by* $S(t)$, $E(t)$, $I(t)$, $R(t)$, $D(t)$, *respectively. The system of equations they satisfy is given by*

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\alpha}{N} S\, I \\ \frac{dE}{dt} &= \frac{\alpha}{N} S\, I - \beta\, E \\ \frac{dI}{dt} &= \beta\, E - \frac{1}{T_I}\, I \\ \frac{dR}{dt} &= \frac{1-f}{T_I}\, I \\ \frac{dD}{dt} &= \frac{f}{T_I}\, I \end{aligned} \tag{4}$$

which is complemented with appropriate initial conditions, $S(t_0)$, $E(t_0)$, $I(t_0)$, $R(t_0)$, $D(t_0)$. *In this system* α *is the infection rate,* β *is the incubation rate,* T_I *is the average infectious period and* f *is the fraction of individuals who die. Here the dimension is* $d = 5$ *and the state variable is the vector* $\mathbf{x}(t) = (S(t), E(t), I(t), R(t), D(t))^T$, *the number of parameters is* $np = 4$ *and* $\boldsymbol{\theta} = (\alpha, \beta, T_I, f)^T \sim (\alpha, \beta, \gamma, \mu)^T$ *with* $\gamma = 1/T_I$ *and* $\mu = f/T_1$. *Figure 1 shows the solution* $\mathbf{x}(t)$, *obtained with the standard RK4 solver, of (1)-(5) in the time interval* $[t_0, t_f] = [40, 80]$ *(days), with exact parameters* $\boldsymbol{\theta} = (\alpha, \beta, \gamma, \mu)^T = (1, 1/7, 1/5, 1/70)^T$, *initial condition* $\mathbf{x}_0 = (91647, 4853, 1755, 1620, 125)^T$ *and total population* $N = 10^5$. *The points are experimental measurements at times* $t_i = 40 + i$, $1 \leq i \leq 13$ $(m = 13)$. *Of course, not always all variables are observable, as shown in this figure. Most frequently the observed variables reported by the medical or government agencies are those people in the population that have been infected, recovered and died, i.e.* $\overline{\mathbf{x}_i} = (I_i, R_i, D_i)$ *at times* t_i, *so that the non-observable variables are* $\underline{\mathbf{x}}_i = (S_i, E_i)$.

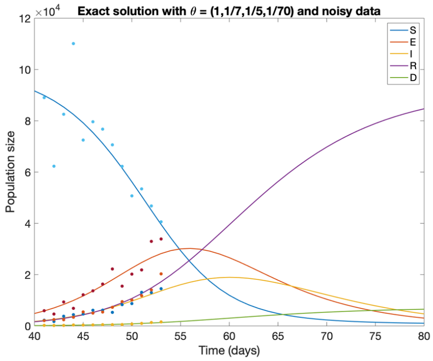

Figure 1: The solution of the SEIRD model and synthetic measurements with white noise.

Assuming that the total population is constant or its change is negligible during the time period, then above system must be complemented by the conservation equation

$$S(t) + E(t) + I(t) + R(t) + D(t) = N, \quad \forall\, t \in [t_0, t_f], \tag{5}$$

so (1)-(5) describe an differential-algebraic system. Accordingly, we may modify the optimization model (6), adding a penalized term, obtaining the extended model:

$$L(\mathbf{x}_0, \boldsymbol{\theta}) = \frac{1}{2} \left\| \frac{\mathbf{x}_0 - \mathbf{s}_0}{\boldsymbol{\sigma}_0} \right\|_5^2 + \frac{k}{2} (\mathbf{x}_0 \cdot \mathbf{1} - N)^2 + \frac{1}{2} \sum_{i=1}^{m} \left\| \frac{\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta}) - \overline{\mathbf{x}}_i}{\overline{\boldsymbol{\sigma}}_i} \right\|_3^2, \tag{6}$$

The penalty parameter k *may be proportional to* N. *The added penalized term in (6) helps stabilizing the optimization process to estimate the initial conditions. The constant vector* $\mathbf{1} \in \mathbb{R}^d$ *has components all equal to one, so the scalar product* $\mathbf{x}_0 \cdot \mathbf{1}$ *is equal to the sum of the components of* $\mathbf{x}_0$.

3 Variational approach

Our goal is to find the initial conditions $\mathbf{x}_0 \in \mathbb{R}^d$ and the parameters $\boldsymbol{\theta} \in \mathbb{R}^{np}$ that minimize the cost function (6) subject to (1) and (5), given that we have a set of noisy experimental measurements (2) at different times. Since the model is deterministic and all the variables and functions are both continuous and smooth, we can employ gradient descent methods or quasi-Newton algorithms and its variants. The most expensive and delicate task, when applying these methods, is the calculation of the gradient or Jacobians of the cost function at each iteration. Some options are available to compute these quantities as shown in [5, 6, 14], and references therein. Here, we are interested on two approaches which are based on variational calculus.

Let $\mathbf{z} = (x_0, \boldsymbol{\theta})^T \in \mathbb{R}^{d+np}$ be the vector which contain the unknown initial conditions and the unknown vector of parameters of the dynamical system, then to first order

$$\nabla L(\mathbf{z}) \cdot \delta\mathbf{z} \approx L(\mathbf{z} + \delta\mathbf{z}) - L(\mathbf{z}) \tag{7}$$

where $\delta\mathbf{z} = (\delta\mathbf{x}_0, \delta\boldsymbol{\theta})^T$ is an small increment of $\mathbf{z}$. If we want to be more specific we can write

$$\nabla_{\mathbf{x}_0} L(\mathbf{x}_0, \boldsymbol{\theta}) \cdot \delta\mathbf{x}_0 + \nabla_\theta L(\mathbf{x}_0, \boldsymbol{\theta}) \cdot \delta\boldsymbol{\theta} \approx L(\mathbf{x}_0 + \delta\mathbf{x}_0, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) - L(\mathbf{x}_0, \boldsymbol{\theta}), \tag{8}$$

where $\nabla_{\mathbf{x}_0}$ and ∇_θ denote the gradients with respect $\mathbf{x}_0$ and $\boldsymbol{\theta}$, receptively and the dot is used to denote the corresponding scalar products. Evaluating directly $L(\mathbf{x}_0 + \delta\mathbf{x}_0, \boldsymbol{\theta} + \delta\boldsymbol{\theta})$ from (6), and simplifying, we obtain

$$\begin{aligned} \nabla_{x_0} L(\mathbf{x}_0, \boldsymbol{\theta}) \cdot \delta\mathbf{x}_0 + \nabla_\theta L(\mathbf{x}_0, \boldsymbol{\theta}) \cdot \delta\boldsymbol{\theta} &\approx \frac{\mathbf{x}_0 - \mathbf{s}_0}{\boldsymbol{\sigma}_0^2} \cdot \delta\mathbf{x}_0 + k(\mathbf{x}_0 \cdot \mathbf{1} - N)\, \mathbf{1} \cdot \delta\mathbf{x}_0 \\ &+ \sum_{i=1}^{m} \frac{\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta}) - \overline{\mathbf{x}}_i}{\overline{\boldsymbol{\sigma}}_i^2} \cdot \left(\frac{\partial\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta})}{\partial\mathbf{x}_0} \delta\mathbf{x}_0 + \frac{\partial\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \delta\boldsymbol{\theta} \right). \end{aligned} \tag{9}$$

At the limit $\delta\mathbf{z} = (\delta\mathbf{x}_0, \delta\boldsymbol{\theta})^T \to \mathbf{0}$ we obtain the gradient $\nabla L(\mathbf{z}) = (\nabla_{x_0} L(\mathbf{x}_0, \boldsymbol{\theta}), \nabla_\theta L(\mathbf{x}_0, \boldsymbol{\theta}))^T$ with the above derivatives distributed accordingly. In (9), matrices $\partial\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta})/\partial\mathbf{x}_0$ and $\partial\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta})/\partial\boldsymbol{\theta}$ are the Jacobians of the observable variables $\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta})$ with respect $\mathbf{x}_0$ and $\boldsymbol{\theta}$, respectively. In the literature, their partial derivatives are commonly called "sensitivities of the observables", and their evaluation may require considerable computational effort. The first matrix is of size $no \times d$ and the second matrix is of size $no \times np$, then the total number of partial derivatives that we must compute in (9) is $m \times no(d + np)$. For instance, in the above example for the SEIRD model, if the number of observable variables is $no = 3$, and there are $m = 13$ experimental measurements, we have to compute $13 \times 3(5 + 4) = 351$ sensitivities at each iteration of a typical gradient or quasi-Newton method. The most basic method to compute these derivatives is the finite difference method (see [14]), but it has some disadvantages. As mentioned before, we concentrate only in variational methods, which may be more sophisticated but they are commonly more efficient.

3.1 Variational method to compute the sensitivities

This method is also known as the "forward approach", because a forward dynamical systems is solved to compute the sensitivities. We consider first the calculation of the sensitivities with respect to the parameter $\boldsymbol{\theta}$, so let us denote the Jacobian $\partial\mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta})/\partial\boldsymbol{\theta}$ by $J(t; \mathbf{x}_0, \boldsymbol{\theta})$ for simplicity. Then

$$J(t; \mathbf{x}_0, \boldsymbol{\theta})\, \delta\boldsymbol{\theta} \approx \mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) - \mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta}).$$

Taking derivatives with respect to t, we obtain

$$\begin{aligned} \dot{J}(t; \boldsymbol{\theta})\, \delta\boldsymbol{\theta} &\approx \dot{\mathbf{x}}(t; \mathbf{x}_0, \boldsymbol{\theta} + \delta\boldsymbol{\theta}) - \dot{\mathbf{x}}(t; \mathbf{x}_0, \boldsymbol{\theta}) \\ &= \mathbf{f}(\mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta} + \delta\boldsymbol{\theta}), \boldsymbol{\theta} + \delta\boldsymbol{\theta}) - \mathbf{f}(\mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta}), \boldsymbol{\theta}) \\ &\approx \mathbf{f}_{\mathbf{x}}(\mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta}), \boldsymbol{\theta}) J(t; \mathbf{x}_0, \boldsymbol{\theta})\, \delta\boldsymbol{\theta} + \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta}), \boldsymbol{\theta})\, \delta\boldsymbol{\theta}. \end{aligned}$$

Then, $\mathbf{x}(t) = \mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta})$ and $J(t) = J(t; \mathbf{x}_0, \boldsymbol{\theta})$ satisfy the following variational system:

$$\begin{aligned} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}), &\quad t_0 < t \le t_f, \\ &\quad \mathbf{x}(t_0) = \mathbf{x}_0, \\ \dot{J}(t) = \mathbf{f}_{\mathbf{x}}(\mathbf{x}(t), \boldsymbol{\theta}) J(t) + \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), \boldsymbol{\theta}), &\quad t_0 < t \le t_f, \\ &\quad J(t_0) = \mathbb{O}. \end{aligned} \tag{10}$$

Last two relations in (10) form a system of matricial differential equations and its initial condition reflects the fact that $\mathbf{x}_0$ does not depend on θ, because

$$J(t_0) = J(t_0; \mathbf{x}_0, \boldsymbol{\theta}) = \partial \mathbf{x}(t_0; \mathbf{x}_0, \boldsymbol{\theta})/\partial \boldsymbol{\theta} = \partial \mathbf{x}_0/\partial \boldsymbol{\theta} = \mathbb{O} \quad \text{(the null matrix)}.$$

A similar variational system is satisfied by the matrix with the sensitivities with respect $\mathbf{x}_0$. Therefore, with this approach, most of the computational work is concentrated in the solution of two matricial systems of differential equations and their evaluation at the experimental times t_i, $1 \leq i \leq m$. The amount of computational work will accumulate at each new iteration of the optimization algorithm, which in some cases may be prohibitive.

3.2 The adjoint method to compute the sensitivities

The total variation of $\mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta})$ with respect to $\mathbf{x}_0$ and $\boldsymbol{\theta}$ is given by

$$\delta \mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta}) = \mathbf{x}(t; \mathbf{x}_0 + \delta \mathbf{x}_0, \boldsymbol{\theta} + \delta \boldsymbol{\theta}) - \mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta}) = \frac{\partial \mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta})}{\partial \mathbf{x}_0} \delta \mathbf{x}_0 + \frac{\partial \mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \delta \boldsymbol{\theta}. \tag{11}$$

Our goal is to avoid the explicit calculation of the Jacobian matrices on the right hand side. Before we proceed further, let us first introduce the following Hilbert spaces:

$$V = L^2([t_0, t_f]; \mathbb{R}^d) = \left\{ \mathbf{v} : [t_0, t_f] \to \mathbb{R}^d \,\middle|\, \int_{t_0}^{t_f} ||\mathbf{v}(t)||^2 \, dt < \infty \right\},$$
$$H = H^1([t_0, t_f]; \mathbb{R}^d) = \{ \mathbf{v} \in V | d\mathbf{v}/dt \in V \},$$

where the $||\mathbf{v}(t)||^2 = \mathbf{v}(t)^T \mathbf{v}(t) = \mathbf{v}(t) \cdot \mathbf{v}(t)$ is the usual scalar product in $\mathbb{R}^d$. Then a natural inner product in V is given by $\langle \mathbf{u}, \mathbf{v} \rangle = \int_{t_0}^{t_f} \mathbf{u}(t) \cdot \mathbf{v}(t) \, dt$ with induced norm given by $\|\mathbf{u}\|_V = \langle \mathbf{u}, \mathbf{u} \rangle^{1/2}$.

Since $\mathbf{x}(t) = \mathbf{x}(t; \mathbf{x}_0, \boldsymbol{\theta}) \in H$ satisfies the state equation (1), then the following inner product is null

$$\int_{t_0}^{t_f} (\dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{x}_0, \boldsymbol{\theta})) \cdot \mathbf{p}(t) \, dt = 0, \quad \forall \mathbf{p} \in H.$$

Differentiating this expression, we get

$$\int_{t_0}^{t_f} \delta \dot{\mathbf{x}}(t) \cdot \mathbf{p}(t) \, dt = \int_{t_0}^{t_f} [\, \mathbf{f}_{\mathbf{x}}(\mathbf{x}(t), \mathbf{x}_0, \boldsymbol{\theta}) \, \delta \mathbf{x}(t) + \mathbf{f}_{\mathbf{x}_0}(\mathbf{x}(t), \mathbf{x}_0, \boldsymbol{\theta}) \, \delta \mathbf{x}_0 + \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), \mathbf{x}_0, \boldsymbol{\theta}) \, \delta \boldsymbol{\theta} \,] \cdot \mathbf{p}(t) \, dt,$$

where $\mathbf{f}_{\mathbf{x}}$, $\mathbf{f}_{\mathbf{x}_0}$ and $\mathbf{f}_{\boldsymbol{\theta}}$ are the Jacobians of $\mathbf{f}$ with respect to $\mathbf{x}$, $\mathbf{x}_0$ and $\boldsymbol{\theta}$, respectively. Integrating by parts the left hand side, and observing that on the right hand side $[\mathbf{f}_{\mathbf{x}} \, \delta \mathbf{x}] \cdot \mathbf{p} = \left[\mathbf{f}_{\mathbf{x}}^T \, \mathbf{p} \right] \cdot \delta \mathbf{x}$, $[\mathbf{f}_{\boldsymbol{\theta}} \, \delta \mathbf{x}_0] \cdot \mathbf{p} = \left[\mathbf{f}_{\boldsymbol{\theta}}^T \, \mathbf{p} \right] \cdot \delta \mathbf{x}_0$, and $[\mathbf{f}_{\boldsymbol{\theta}} \, \delta \boldsymbol{\theta}] \cdot \mathbf{p} = \left[\mathbf{f}_{\boldsymbol{\theta}}^T \, \mathbf{p} \right] \cdot \delta \boldsymbol{\theta}$, we obtain

$$\mathbf{p}(t) \cdot \delta \mathbf{x}(t)|_{t_0}^{t_f} - \int_{t_0}^{t_f} \dot{\mathbf{p}}(t) \cdot \delta \mathbf{x}(t) \, dt = \int_{t_0}^{t_f} \left\{ \left[\mathbf{f}_{\mathbf{x}}^T \, \mathbf{p} \right](t) \cdot \delta \mathbf{x}(t) + \left[\mathbf{f}_{\mathbf{x}_0}^T \, \mathbf{p} \right](t) \cdot \delta \mathbf{x}_0(t) + \left[\mathbf{f}_{\boldsymbol{\theta}}^T \, \mathbf{p} \right](t) \cdot \delta \boldsymbol{\theta} \right\} dt,$$

and, assuming that $\mathbf{f}$ does not depend explicitly of $\mathbf{x}_0$ (it depends only through $\mathbf{x}$, but the variation w.r.t. $\mathbf{x}_0$ is already accounted in $\delta \mathbf{x}$), then

$$\mathbf{p}(t) \cdot \delta \mathbf{x}(t)|_{t_0}^{t_f} - \int_{t_0}^{t_f} \left\{ \dot{\mathbf{p}}(t) + \left[\mathbf{f}_{\mathbf{x}}^T \, \mathbf{p} \right](t) \right\} \cdot \delta \mathbf{x}(t) \, dt = \int_{t_0}^{t_f} \left[\mathbf{f}_{\boldsymbol{\theta}}^T \, \mathbf{p} \right](t) \cdot \delta \boldsymbol{\theta} \, dt, \tag{12}$$

where $\delta \mathbf{x}$ is given by (11) and arise in the last term of (9). One way to avoid the explicit computation of (11) is forcing a relation with (12) by introducing the following adjoint equation

$$-\dot{\mathbf{p}}(t) = \mathbf{f}_{\mathbf{x}}(\mathbf{x}(t), \boldsymbol{\theta})^T \, \mathbf{p}(t) + \sum_{i=1}^{m} \frac{(\overline{\mathbf{x}}(t_i; \mathbf{x}_0, \boldsymbol{\theta}) - \overline{\mathbf{x}}_i)}{\overline{\boldsymbol{\sigma}}_i^2} \boldsymbol{\delta}_{\mathbf{D}}(t - t_i), t_f > t \geq t_0, \tag{13}$$

with $\boldsymbol{\delta}_{\mathbf{D}}(t-t_i)$ the Dirac measure centred at t_i. This adjoint equation (a backward in time differential equation) contains all the information about the experimental data $\{\overline{\mathbf{x}}_i\}_{i=1}^m$, and its variational formulation is obtained multiplying by a differentiable test function $\boldsymbol{\phi}(t)$ and integrating:

$$-\int_{t_0}^{t_f}\left\{\dot{\mathbf{p}}(t)+\left[\mathbf{f}_{\mathbf{x}}^T\,\mathbf{p}\right](t)\right\}\cdot\boldsymbol{\phi}(t)\,dt=\int_{t_0}^{t_f}\sum_{i=1}^m\frac{(\overline{\mathbf{x}}(t_i;\mathbf{x}_0,\boldsymbol{\theta})-\overline{\mathbf{x}}_i)}{\overline{\boldsymbol{\sigma}}_i^2}\,\boldsymbol{\delta}_{\mathbf{D}}(t-t_i)\cdot\boldsymbol{\phi}(t)\,dt. \tag{14}$$

The integral on the right hand side is equal to $\sum_{i=1}^m\frac{(\overline{\mathbf{x}}(t_i;\mathbf{x}_0,\boldsymbol{\theta})-\overline{\mathbf{x}}_i)}{\overline{\boldsymbol{\sigma}}_i}\cdot\boldsymbol{\phi}(t_i)$. Choosing $\boldsymbol{\phi}(t)=\delta\mathbf{x}(t)$ in (14) and substituting the result in (12), we obtain

$$-\mathbf{p}(t_0)\cdot\delta\mathbf{x}(t_0)+\sum_{i=1}^m\frac{(\overline{\mathbf{x}}(t_i;\mathbf{x}_0,\boldsymbol{\theta})-\overline{\mathbf{x}}_i)}{\overline{\boldsymbol{\sigma}}_i^2}\cdot\delta\mathbf{x}(t_i)=\int_{t_0}^{t_f}\left[\mathbf{f}_{\boldsymbol{\theta}}^T\,\mathbf{p}\right](t)\cdot\delta\boldsymbol{\theta}\,dt, \tag{15}$$

where $\mathbf{p}(t)$ is the solution of the adjoint equation. Finally, the substitution of this equation into (9), gives

$$\begin{aligned}&\nabla_{x_0}L(\mathbf{x}_0,\boldsymbol{\theta})\cdot\delta\mathbf{x}_0+\nabla_{\theta}L(\mathbf{x}_0,\boldsymbol{\theta})\cdot\delta\boldsymbol{\theta}\\ \approx\ &\frac{\mathbf{x}_0-\mathbf{s}_0}{\boldsymbol{\sigma}_0^2}\cdot\delta\mathbf{x}_0+k(\mathbf{x}_0\cdot\mathbf{1}-N)\,\mathbf{1}\cdot\delta\mathbf{x}_0+\mathbf{p}(t_0)\cdot\delta\mathbf{x}_0+\left[\int_{t_0}^{t_f}\left[\mathbf{f}_{\boldsymbol{\theta}}^T\,\mathbf{p}\right](t)\,dt\right]\cdot\delta\boldsymbol{\theta}.\end{aligned} \tag{16}$$

Therefore, the gradient of the objective function (6) is $\nabla L(\mathbf{z})=(\nabla_{x_0}L(\mathbf{x}_0,\boldsymbol{\theta}),\nabla_{\boldsymbol{\theta}}L(\mathbf{x}_0,\boldsymbol{\theta}))^T$, with

$$\nabla_{x_0}L(\mathbf{x}_0,\boldsymbol{\theta})=\frac{\mathbf{x}_0-\mathbf{s}_0}{\boldsymbol{\sigma}_0^2}+k(\mathbf{x}_0\cdot\mathbf{1}-N)\,\mathbf{1}+\mathbf{p}(t_0), \tag{17}$$

$$\nabla_{\boldsymbol{\theta}}L(\mathbf{x}_0,\boldsymbol{\theta})=\int_{t_0}^{t_f}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t),\mathbf{x}_0,\boldsymbol{\theta})^T\,\mathbf{p}(t)\,dt, \tag{18}$$

where $\mathbf{x}(t)$ solves the state equation (1) and $\mathbf{p}(t)$ solves the adjoint equation (13).

Remark 3. *Formulas (17)-(18) avoid the explicit calculation of the sensitivities, they only requiere the solution of the state equation (1) and of the adjoint equation (13), regardless of the number of experimental data, m, the number of parameters, np, and of the observable variables, no. Furthermore, these same equations must be solved to compute either the gradient with respect* $\mathbf{x}_0$ *or with respect* $\boldsymbol{\theta}$*, or both gradients simultaneously.*

Remark 4. *The solution of the adjoint equation turns out to be the Lagrange multiplier of the optimization problem, with objective function* $L(\mathbf{x}_0,\boldsymbol{\theta})$*, subject to the constraint (1). To show this property, let us introduce the Lagrangian function*

$$\mathcal{L}(\mathbf{x}_0,\boldsymbol{\theta},\mathbf{p})=L(\mathbf{x}_0,\boldsymbol{\theta})+\int_{t_0}^{t_f}\mathbf{p}(t)^T\left[\mathbf{f}(t,\mathbf{x}(t),\boldsymbol{\theta})-\dot{\mathbf{x}}(t)\right]\,dt, \tag{19}$$

where $\mathbf{p}$ *is the Lagrange multiplier associated to the given constraint, and the last term is the inner product of this multiplier with the restriction. Our goal is to compute* $\partial L/\partial\boldsymbol{\theta}$ *from this expression. Formally, the second term on the right hand side of (4) is zero because the state* $\mathbf{x}(t)$ *solves the ODE, therefore* $\partial\mathcal{L}/\partial\boldsymbol{\theta}=\partial L/\partial\boldsymbol{\theta}$*. However, the differentiation of* $\mathcal{L}$ *reveals additional information and gives more freedom. Doing integration by parts of the term* $-\int_{t_0}^{t_f}\mathbf{p}(t)^T\dot{\mathbf{x}}(t)$ *in (4) first, and then taking the derivative with respect to* $\boldsymbol{\theta}$*, we obtain*

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}}&=\frac{\partial L}{\partial\boldsymbol{\theta}}-\mathbf{p}^T\frac{\partial\mathbf{x}}{\partial\boldsymbol{\theta}}\bigg|_{t_0}^{t_f}+\int_{t_0}^{t_f}\dot{\mathbf{p}}^T\frac{\partial\mathbf{x}}{\partial\boldsymbol{\theta}}dt+\int_{t_0}^{t_f}\mathbf{p}^T\left[\mathbf{f}_{\mathbf{x}}\frac{\partial\mathbf{x}}{\partial\boldsymbol{\theta}}+\mathbf{f}_{\boldsymbol{\theta}}\right]dt\\ &=\sum_{i=1}^m\left(\frac{\overline{\mathbf{x}}(t_i,\mathbf{x}_0,\boldsymbol{\theta})-\overline{\mathbf{x}}_i}{\overline{\boldsymbol{\sigma}}_i^2}\right)^T\frac{\partial\overline{\mathbf{x}}}{\partial\theta}-\mathbf{p}^T\frac{\partial\mathbf{x}}{\partial\boldsymbol{\theta}}\bigg|_{t_0}^{t_f}+\int_{t_0}^{t_f}\left(\dot{\mathbf{p}}+\mathbf{f}_{\mathbf{x}}^T\mathbf{p}\right)^T\frac{\partial\mathbf{x}}{\partial\boldsymbol{\theta}}dt+\int_{t_0}^{t_f}\mathbf{p}^T\mathbf{f}_{\boldsymbol{\theta}}\,dt.\end{aligned} \tag{20}$$

The first term on the right hand side is obtained directly from (6) and can be expressed as

$$\sum_{i=1}^m\left(\frac{\overline{\mathbf{x}}(t_i,\mathbf{x}_0,\boldsymbol{\theta})-\overline{\mathbf{x}}_i}{\overline{\boldsymbol{\sigma}}_i^2}\right)^T\frac{\partial\overline{\mathbf{x}}}{\partial\theta}=\int_{t_0}^{t_f}\sum_{i=1}^m\left(\frac{\overline{\mathbf{x}}(t_i,\mathbf{x}_0,\boldsymbol{\theta})-\overline{\mathbf{x}}_i}{\overline{\boldsymbol{\sigma}}_i^2}\right)^T\frac{\partial\overline{\mathbf{x}}}{\partial\theta}\,\delta_D(t-t_i)dt.$$

Now, if $\mathbf{p}$ *satisfies the adjoint equation (13) then the sum of first and third terms in (20) vanish, and also the boundary term* $\mathbf{p}^T\frac{\partial\mathbf{x}}{\partial\boldsymbol{\theta}}\Big|_{t_0}^{t_f}$*, since* $\mathbf{p}(t_f)=\mathbf{0}$ *and* $\partial\mathbf{x}(t_0)/\partial\boldsymbol{\theta}=\mathbb{O}$*. Therefore*

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}}=\frac{\partial L}{\partial\boldsymbol{\theta}}=\int_{t_0}^{t_f}\mathbf{p}^T\mathbf{f}_{\boldsymbol{\theta}}\,dt\quad\Longrightarrow\quad\nabla_{\boldsymbol{\theta}}L(\mathbf{x}_0,\boldsymbol{\theta})=\int_{t_0}^{t_f}\mathbf{f}_{\boldsymbol{\theta}}(t,\mathbf{x}(t),\boldsymbol{\theta})^T\,\mathbf{p}(t)\,dt.$$

A similar development can be applied to obtain $\partial L/\partial\mathbf{x}_0$*.*

3.3 Solution of the adjoint equation and computation of the gradient

The adjoint equation (13) is a system of ODE with backward in time propagation. We apply a change of variable from the symmetric relation $t_f - \tau = t - t_0$:

$$p(t) = p(t_f - (\tau - t_0)) \equiv p_A(\tau) \quad \Longrightarrow \quad \dot{p}(t) = -\dot{p}_A(\tau), \tag{21}$$

obtaining the equivalent system with forward in time dynamics

$$\dot{\mathbf{p}}_A(\tau) = \mathbf{f}_{\mathbf{x}}(\mathbf{x}_A(\tau), \boldsymbol{\theta})^T \, \mathbf{p}_A(\tau) + \sum_{i=1}^{m} \frac{\overline{\mathbf{x}}_A(\tau_i, \mathbf{x}_0, \boldsymbol{\theta}) - \overline{\mathbf{x}}_{m+1-i}}{\boldsymbol{\sigma}^2_{m+1-i}} \, \delta_D(\tau_i - \tau), \quad t_0 \le \tau < t_f, \tag{22}$$

$$\mathbf{p}_A(\tau_0) = \mathbf{0}, \tag{23}$$

which can be solved with any usual numerical ODE solver like Runge-Kutta methods.

For the SEIRD model, if the observable variables are $\overline{\mathbf{x}}(t) = (I(t), R(t), D(t))^T$, the adjoint equation (22) takes the form

$$\begin{bmatrix} \dot{p}_{A1}(\tau) \\ \dot{p}_{A2}(\tau) \\ \dot{p}_{A3}(\tau) \\ \dot{p}_{A4}(\tau) \\ \dot{p}_{A5}(\tau) \end{bmatrix} = \begin{bmatrix} -\frac{\alpha\, I(\tau)}{N} & \frac{\alpha\, I(\tau)}{N} & 0 & 0 & 0 \\ 0 & -\beta & \beta & 0 & 0 \\ -\frac{\alpha\, S(\tau)}{N} & \frac{\alpha\, S(\tau)}{N} & -\gamma & \gamma - \mu & \mu \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_{A1}(\tau) \\ p_{A2}(\tau) \\ p_{A3}(\tau) \\ p_{A4}(\tau) \\ p_{A5}(\tau) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^m \frac{I(\tau_i) - I_{m+1-i}}{\sigma^2_{i3}} \delta_D(\tau_i - \tau) \\ \sum_{i=1}^m \frac{R(\tau_i) - R_{m+1-i}}{\sigma^2_{i4}} \delta_D(\tau_i - \tau) \\ \sum_{i=1}^m \frac{D(\tau_i) - D_{m+1-i}}{\sigma^2_{i5}} \delta_D(\tau_i - \tau) \end{bmatrix},$$

with $\gamma = 1/T_1$, $\mu = f/T_1$ in (1). After solving this forward adjoint equation we recover the solution $\mathbf{p}$ of the backward adjoint equation (13) with $\mathbf{p}(t_i) = \mathbf{p}_A(\tau_{m+1-i})$, $i = 1, \ldots, m$ or by interpolation and using (21) for other values of t. The last step to compute the gradient is the calculation of the integral in (18). Observe that

$$\begin{aligned} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), \boldsymbol{\theta})^T \, \mathbf{p}(t) &= \begin{bmatrix} -\dfrac{S(t)\, I(t)}{N} & \dfrac{S(t)\, I(t)}{N} & 0 & 0 \\ 0 & -E(t) & E(t) & 0 \\ 0 & 0 & -I(t) & I(t) \end{bmatrix} \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ p_4(t) \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{S(t)\, I(t)}{N} \, (p_2(t) - p_1(t)) \\ E(t)\, (p_3(t) - p_2(t)) \\ I(t)\, (p_4(t) - p_3(t)) \end{bmatrix} \equiv \mathbf{H}(t) \end{aligned}$$

Then, using the Simpson's rule in a set of given times (nodes of the mesh or interpolated times) we have

$$\int_{t_0}^{t_f} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), \boldsymbol{\theta})^T \, \mathbf{p}(t)\, dt = \frac{\Delta t}{6} \sum_{j=1}^{N-1} \{\mathbf{H}(t_{j-1}) + 4\mathbf{H}(t_j) + \mathbf{H}(t_{j+1})\} \tag{24}$$

4 Numerical algorithms for the optimization problem

4.1 Conjugate gradient algorithm

This algorithm is one of the most important algorithms for quadratic optimization problems with positive definite Hessians and for unconstrained continuous convex optimization. It may be considered as a variant of gradient descent where the search directions are generated progressively based on the orthogonality of the residuals and conjugacy of the search directions. The conjugate directions are calculated at each iteration a linear combination of the most recent negative gradient and the last conjugate direction, as indicated in step 9 of Algorithm 1 bellow. Its computational cost is comparable to steepest descent, but it has faster convergence, specially for ill conditioned problems, [10].

Algorithm 1. Conjugate Gradient Algorithm
1. Initial guess $\mathbf{z}^0 = \widehat{\mathbf{z}} = (\widehat{\mathbf{x}}_0, \widehat{\boldsymbol{\theta}})^T$*. 2. Initial gradient* $\mathbf{g}^0 = \nabla L(\mathbf{z}^0)$*. 3. Initial direction* $\mathbf{d}^0 = -\mathbf{g}^0$*. "'Descent"' For* $\ell \geq 0$*, given* $\mathbf{z}^\ell$*,* $\mathbf{g}^\ell$*,* $\mathbf{d}^\ell$*, find* $\mathbf{z}^{\ell+1}$*,* $\mathbf{g}^{\ell+1}$*,* $\mathbf{d}^{\ell+1}$*, doing the following 4. Find* $\rho_\ell = \arg\min_{\rho \geq 0} \varphi(\rho) = L(\mathbf{z}^\ell + \rho\,\mathbf{d}^\ell)$ *5. Update* $\mathbf{z}^{\ell+1} = \mathbf{z}^\ell + \rho_\ell\,\mathbf{d}^\ell$*. 6. Evaluate* $\mathbf{g}^{\ell+1} = \nabla L(\mathbf{z}^{\ell+1})$ *"'Convergence test and new direction"'* $\|\mathbf{g}^{\ell+1}\| \leq \epsilon \|\mathbf{g}^0\|$ *7. Take* $\mathbf{z}^* = \mathbf{z}^{\ell+1}$*. Stop and exit. 8. Evaluate* $\beta_\ell = \dfrac{\mathbf{g}^{\ell+1} \cdot \mathbf{g}^{\ell+1}}{\mathbf{g}^\ell \cdot \mathbf{g}^\ell}$ *9. Update* $\mathbf{d}^{\ell+1} = -\mathbf{g}^\ell + \beta_\ell\,\mathbf{d}^\ell$ *10. Make* $\ell = \ell + 1$ *and go back to 4.*

If $\beta_\ell = 0$ for all ℓ we recover steepest descent. Some variants for computing $\beta_\ell \neq 0$, include:

- Fletcher–Reeves: $\beta_\ell = \dfrac{\mathbf{g}^{\ell+1} \cdot \mathbf{g}^{\ell+1}}{\mathbf{g}^\ell \cdot \mathbf{g}^\ell}$
- Polak–Ribiere: $\beta_\ell = \dfrac{\mathbf{g}^{\ell+1} \cdot (\mathbf{g}^{\ell+1} - \mathbf{g}^\ell)}{\mathbf{g}^\ell \cdot \mathbf{g}^\ell}$
- Hestenes–Stiefel: $\beta_\ell = \dfrac{\mathbf{g}^{\ell+1} \cdot (\mathbf{g}^{\ell+1} - \mathbf{g}^\ell)}{\mathbf{d}^\ell \cdot (\mathbf{g}^{\ell+1} - \mathbf{g}^\ell)}$
- Dai–Yuan: $\beta_\ell = \dfrac{\mathbf{g}^{\ell+1} \cdot \mathbf{g}^{\ell+1}}{\mathbf{d}^\ell \cdot (\mathbf{g}^{\ell+1} - \mathbf{g}^\ell)}$

We use the short notations F-R, P-R, H-S, D-Y, for these variants.

4.2 BFGS algorithm

This is one of the most popular quasi-Newton algorithms for nonlinear optimization. It is usually more effective because it involves information about curvature, besides the information about the gradient. The curvature information is incorporated by approximating the Hessian matrix during the iteration process, which formaly plays the role of preconditioner for the gradient. The general idea about these methods comes from the second order approximation:

$$L(\mathbf{z}^\ell + \Delta\mathbf{z}) \approx L(\mathbf{z}^\ell) + \nabla L(\mathbf{z}^\ell)^T \Delta\mathbf{z} + \frac{1}{2}\Delta\mathbf{z}^T A\,\Delta\mathbf{z},$$

with a matrix A close to the Hessian. Then the so called secant condition is obtained:

$$\nabla L(\mathbf{z}^\ell + \Delta\mathbf{z}) \approx \nabla L(\mathbf{z}^\ell) + A\,\Delta\mathbf{z}.$$

Forcing the gradient to be zero (to look for a minimum), we obtain the Newton step

$$\Delta\mathbf{z} = -H\,\nabla L(\mathbf{z}^\ell), \quad \text{where} \quad H \approx A^{-1}$$

and the following search direction $\mathbf{d}^\ell$ is obtained;

$$\mathbf{d}^\ell = -H\,\mathbf{g}^\ell, \quad \text{with} \quad \mathbf{d}^\ell = \Delta\mathbf{z}, \quad \mathbf{g}^\ell = \nabla L(\mathbf{z}^\ell).$$

The Hessian and its inverse are updated at every iteration adding rank-one or rank-two matrices. The BFGS algorithm updates the approximated Hessian with a rank-two matrix as showing in step 9 of Algorithm 2 bellow.

Algorithm 2. BFGS algorithm
1. Initial guess ans initial Hessian: $\mathbf{z}^0$ *given, and* $H^0 = I$*. 2. Initial gradient* $\mathbf{g}^0 = \nabla L(\mathbf{z}^0)$*. 3. Initial direction* $\mathbf{d}^0 = -\mathbf{g}^0$*. "'Descent"' For* $\ell \geq 0$*, given* $\mathbf{z}^\ell$*,* $\mathbf{g}^\ell$*,* $\mathbf{d}^\ell$*,* H^ℓ *find* $\mathbf{z}^{\ell+1}$*,* $\mathbf{g}^{\ell+1}$*,* $\mathbf{d}^{\ell+1}$*,* $H^{\ell+1}$*, doing the following 4. Find* $\rho_\ell = \arg\min_{\rho \geq 0} \varphi(\rho) = L(\mathbf{z}^\ell + \rho\,\mathbf{d}^\ell)$ *5. Update* $\mathbf{z}^{\ell+1} = \mathbf{z}^\ell + \rho_\ell\,\mathbf{d}^\ell$*. 6. Evaluate* $\mathbf{g}^{\ell+1} = \nabla L(\mathbf{z}^{\ell+1})$ *"'Convergence test and new direction"'* $\|\mathbf{g}^{\ell+1}\| \leq \epsilon \|\mathbf{g}^0\|$ *7. Do* $\mathbf{z}^* = \mathbf{z}^{\ell+1}$*. Stop and exit. 8. Evaluate* $\mathbf{u} = \Delta\mathbf{z}^\ell = \mathbf{z}^{\ell+1} - \mathbf{z}^\ell$ *and* $\mathbf{v} = \Delta\mathbf{g}^\ell = \mathbf{g}^{\ell+1} - \mathbf{g}^\ell$ *9. Update* $H^{\ell+1} = \left(I - \dfrac{\mathbf{v}\mathbf{u}^T}{\mathbf{u}^T\mathbf{v}}\right) H^\ell \left(I - \dfrac{\mathbf{u}\mathbf{v}^T}{\mathbf{u}^T\mathbf{v}}\right) + \dfrac{\mathbf{v}\mathbf{v}^T}{\mathbf{u}^T\mathbf{v}}$ *10. Update* $\mathbf{d}^{\ell+1} = -H^{\ell+1}\mathbf{g}^{\ell+1}$ *11. Make* $\ell = \ell + 1$ *and go back to 4*

Remark 5. *Another very popular method for least squares problems is the Gauss-Newton method and it variant, the Levenverg-Marquadt method (see [10]), where the Jacobian of the residual vector* $\mathbf{r} = \left\{ \left\| \frac{\overline{\mathbf{x}}(t_i;\mathbf{x}_0,\boldsymbol{\theta}) - \overline{\mathbf{x}}_i}{\overline{\boldsymbol{\sigma}}_i} \right\|^2 \right\}_{i=1}^m$ *with respect to* $\mathbf{x}_0$ *and* $\boldsymbol{\theta}$ *must be computed at each iteration. These partial derivatives can also be computed with variational methods.*

4.3 Line Search

The most critical step in Algorithms 1 and 2 is the solution of the one dimensional optimization problem at step 4. This is not a trivial step and requires careful treatment. There are several algorithms for this problem, the most common are line search methods and trust region methods. Some classic references are [9] and [10], or the more recent one [15], while some publications, e,g, [1] and [11], show that this topic is still under development.

Given that we have a efficient way to compute the derivative $\varphi'(\rho) = \nabla L(\mathbf{z}^\ell + \rho\,\mathbf{d}^\ell) \cdot \mathbf{d}^\ell$, we may use the secant method, whose iteration formula is

$$\rho_{k+1} = \rho_k - \frac{\varphi'(\rho_k)(\rho_k - \rho_{k-1})}{\varphi'(\rho_k) - \varphi'(\rho_{k-1})} = \frac{\rho_{k-1}\varphi'(\rho_k) - \rho_k\varphi'(\rho_{k-1})}{\varphi'(\rho_k) - \varphi'(\rho_{k-1})}, \qquad k = 1, 2, \ldots \tag{25}$$

with two initial values $\rho_0 = 0$ and $\rho_1 \approx \epsilon \frac{||\mathbf{z}||}{||\mathbf{d}||}$, $\epsilon < 1$. The initial value ρ_1 takes into account a proper scaling (see step 5). Computing $\varphi'(\rho)$ requires solving the state equation (1) with initial conditions $\mathbf{x}_0^\ell + \rho\mathbf{d}_0^\ell$ and parameter $\boldsymbol{\theta}^\ell + \rho\mathbf{d}_{\boldsymbol{\theta}}^\ell$, obtaining a solution that we call $\mathbf{x}_\rho^\ell$, and then solving the corresponding adjoint equation (13) with $\mathbf{x} = \mathbf{x}_\rho^\ell$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^\ell + \rho\mathbf{d}_{\boldsymbol{\theta}}^\ell$, obtaining the solution $\mathbf{p}_\rho^\ell$. Thus

$$\varphi'(\rho) = \begin{bmatrix} \mathbf{p}_\rho^\ell(t_0) + \dfrac{\overline{\mathbf{x}}_0^\ell + \rho\,\overline{\mathbf{d}}_0^\ell - \overline{\mathbf{y}}_0}{\overline{\boldsymbol{\sigma}}_0^2} + k\left([\mathbf{x}_0^\ell + \rho\mathbf{d}_0^\ell]\cdot\mathbf{1} - N\right)\,\mathbf{1} \\ \displaystyle\int_{t_0}^{t_f} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_\rho^\ell, \boldsymbol{\theta}^\ell + \rho\,\mathbf{d}_\theta^\ell)^T \mathbf{p}_\rho^\ell\,dt \end{bmatrix} \cdot \begin{bmatrix} \mathbf{d}_0^\ell \\ \mathbf{d}_\theta^\ell \end{bmatrix}. \tag{26}$$

Remark 6. *The secant method for line search may be improved, incorporating standard bracketing strategies that keeps track of upper and lower bounds for the location of the root [1]. We will try this enhancement in a future work.*

Remark 7. *Newton's method for line is also possible. However, it is more costly because we must compute $\varphi''(\rho)$, i.e. the derivative of (26) with respect to ρ . This operation involves the solution of another two systems of ODE at each iteration: one forward-in-time problem (related to the state equation) and a backward-in-time problem (related to the adjoint equation), where the Jacobians $\mathbf{f_x}$ and $\mathbf{f}_{\boldsymbol{\theta}}$ must be evaluated at different values. We leave this task for a future work.*

5 Numerical results for the SEIRD model

To validate the fitting model and the proposed optimization algorithms we consider the SEIRD model, described in Section 2. Our base or reference true solution is the one obtained numerically in the time interval is $[t_0, t_f] = [40, 80]$ (days), parameters $\boldsymbol{\theta} = (\alpha, \beta, \gamma, \mu)^T = (1, 1/7, 1/5, 1/70)^T$, initial condition $\mathbf{x}_0 = (91647, 4853, 1755, 1620, 125)^T$ and total population $N = 10^5$, shown in Figure 1. Synthetic data is generated adding white noise to the true solution with the random Gaussian generator of Matlab, with zero mean and standard deviations $\mathbf{s}(t_i) = noise\_level * \mathbf{x}(t_i)$ at the times t_i where we suppose to have experimental measurements. The proposed model and numerical algorithms are tested at three levels of noise, 0.05, 0.1, and 0.2. We will divide the experiments in two parts: 1) only the vector parameter $\boldsymbol{\theta}$ is unknown, 2) both the initial conditions $\mathbf{x}_0$ and the vector parameter $\boldsymbol{\theta}$ are unknown.

Example 2. *Case when $\mathbf{x}_0$ is known and $\boldsymbol{\theta}$ is unknown, $noise\_level = 0.1$.*

In many problems we are interested in recovering the vector of unknown parameters $\boldsymbol{\theta}$, assuming that we are given the exact initial conditions $\mathbf{x}_0$ and the experimental data $\{\overline{\mathbf{x}}_i\}_{i=1}^m$ at the corresponding times t_i. We consider synthetic experimental noisy data with $noise\_level = 0.1$, where the observable variables are $\overline{\mathbf{x}}(t_i) = (I_i, R_i, D_i)^T$ in the time window $t = 40 + i$, $1 \leq i \leq 13$. Table 2 shows the numerical results obtained with the CG-algorithm (F-R variant) and the BFGS-algorithms, with initial guess $\theta^0 = (1.4, 0.09, 0.2, 0.001)^T$ and tolerance $\epsilon = 10^{-8}$ to stop the iterations. The relative error is computed component-wise.

We obtain almost the same numerical value for the computed $\boldsymbol{\theta}$ with both algorithms, the main difference is the number of iterations for each algorithm to achieve convergence to the given tolerance. Overall, the most important

Table 1: Numerical results with the CG and BFGS algorithms.

Method	CG (F-R variant)	BFGS
$\boldsymbol{\theta}^0$	$(1.4, 0.09, 0.2, 0.001)$	$(1.4, 0.09, 0.2, 0.001)$
Data time window	$t_i = 40 + i,\ 1 \le i \le 13$	$t_i = 40 + i,\ 1 \le i \le 13$
ϵ, no. iters.	$\epsilon = 10^{-8}$, 168	$\epsilon = 10^{-8}$, 14
Computed $\boldsymbol{\theta}$	$(1.0769, 0.1425, 0.2180, 0.0143)$	$(1.0768, 0.1426, 0.2179, 0.0143)$
Relative error	$(0.0769, 0.0022, 0.0898, 4.8e(-5))$	$(0.0768, 0.0021, 0.0897, 0.0001)$

feature in this experiment is that the numerical computation is stable and the relative error is smaller than the noise level 0.1 (10%) for each parameter. Figure 2 (left), shows the behaviour of $||\nabla L(\boldsymbol{\theta})||$ (logarithmic scale) and clearly show the faster convergence of the BFGS algorithm. Figure 2 (right) illustrate the convergence of the parameters $\boldsymbol{\theta}^\ell = \left(\alpha^\ell, \beta^\ell, \gamma^\ell, \mu^\ell\right)^T$. Finally, Figure 2 shows the dynamics of the true solution $\mathbf{x}$ (continuous lines) and the numerical solution obtained with the computed parameters (dashed lines), along with the noisy data (points). We want to emphasize that the initial value for parameter $\gamma^0 = 0.2$ is the exact one, since this a parameter is usually assumed to be known. However the numerical algorithms converge for other initial values, like $\gamma^0 = 0.1$ or 0.3 with a similar number of iterations.

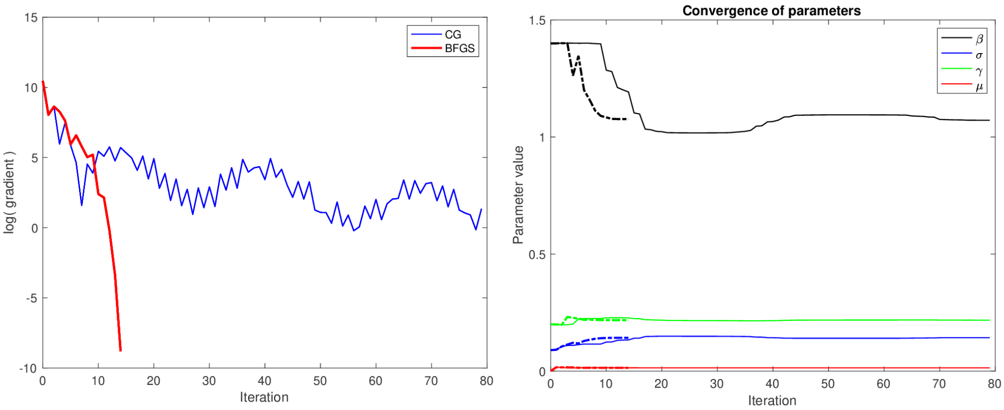

Figure 2: Left: graph of $\log\left(\nabla L(\boldsymbol{\theta})^\ell\right)$ against iteration value ℓ. Right figure: Convergence of $\boldsymbol{\theta}^\ell = \left(\alpha^\ell, \beta^\ell, \gamma^\ell, \mu^\ell\right)^T$ with respect to iteration ℓ for CG (continuous lines) and for BFGS (dashed lines).

Example 3. *Case when both $\mathbf{x}_0$ and $\boldsymbol{\theta}$ are unknown, noise_level* $= 0.1$.

This problem needs a more careful treatment, we now have to compute nine parameters instead of four and the scale of both unknowns is different: $\mathbf{x}_0$ may have components of the order of 10^5 while $\boldsymbol{\theta}$ has components of the order at most $10^0 = 1$. The initial guess $\boldsymbol{\theta}^0$ to start iterations is the same for both algorithms (CG and BFGS). However, the election of the initial guess $\mathbf{x}_0^0$ is more subtle. We first fix the value of $\mathbf{s}_0$ in model (6) with the following formula

$$\mathbf{s}_0 = round(N/\widehat{N})\,\widehat{\mathbf{x}}_0 \quad where \quad \widehat{N} = \sum_{i=1}^{5} \widehat{x}_{i\,0}, \tag{27}$$

where $\widehat{\mathbf{x}}_0$ is obtained from the noisy data at $t = 40$. The adjustment in (3) ensures that the sum of the components of $\mathbf{s}_0$ be equal to $N = 10^5$. Observe that $\widehat{N}$ is not guaranteed to be equal to N, because experimental data have inherent noise. Finally, we choose $\mathbf{x}_0^0 = \mathbf{s}_0$ for the CG algorithm and $\mathbf{x}_0^0 = \widehat{\mathbf{x}}_0$ for the BFGS algorithm. The CG algorithm does not always converge properly with an arbitrary initial value like $\mathbf{x}_0^0 = \widehat{\mathbf{x}}_0$. Table 2 summarize the

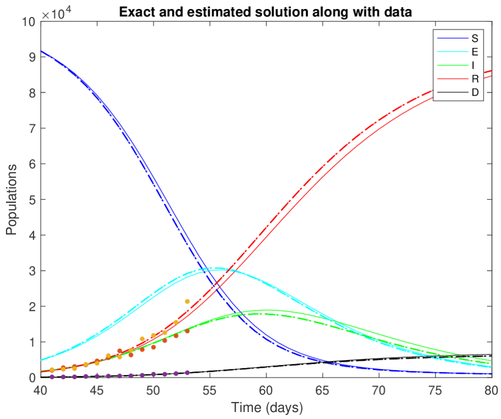

Figure 3: The dynamics of the true solution $\mathbf{x}(t) = (S(t), E(t), I(t), R(t), D(t))^T$ (continuous lines) and the one obtained with computed $\boldsymbol{\theta}$ (dashes lines), along with noisy data for the observable variables (I_i, R_i, D_i)

numerical results.

Table 2: Numerical results for computed $\boldsymbol{\theta}$ and $\mathbf{x}_0$ with CG (P-R) and BFGS algorithms.

Method	CG (P-R variant)	BFGS
$\boldsymbol{\theta}^0$	$(1.40, 0.09, 0.20, 0.001)$	$(1.40, 0.09, 0.20, 0.001)$
$\mathbf{x}_0^0$	$(91503, 4978, 1872, 1643, 132)$	$(91503, 4978, 1872, 1643, 132)$
Data time window	$t_i = 40 + (2i - 1),\ 1 \le i \le 8$	$t_i = 40 + (2i - 1),\ 1 \le i \le 8$
ϵ, no. iters.	10^{-5}, 229	10^{-6}, 41
Computed $\boldsymbol{\theta}$	$(1.0075, 0.1420, 0.2018, 0.0134)$	$(1.0077, 0.1419, 0.2003, 0.0132)$
Relative error	$(0.0075, 0.0058, 0.0090, 0.0590)$	$(0.0077, 0.0064, 0.0016, 0.0758)$
Computed $\mathbf{x}_0$	$(91386, 4972, 1870, 1641, 132)$	$(91340, 4996, 1844, 1681, 139)$
Relative error	$(0.0029, 0.0245, 0.0654, 0.0128, 0.0548)$	$(0.0034, 0.0294, 0.0505, 0.0379, 0.1148)$

This time the CG algorithm does not admit tolerances smaller than $\epsilon = 10^{-5}$ and also the P-R variant to compute β_ℓ at step 8 turn out to be more efficient than F-R. We observe that the window time for experimental data is wider but with less data points for both algorithms. The computed $\boldsymbol{\theta}$ obtained with both algorithms is almost the same, but the computed $\mathbf{x}_0$ exhibits more discrepancy. This lack of stability on the estimation of initial conditions from noisy data is already known by the scientific community. In fact, if $\mathbf{f}$ is Lipschitz continuous with constant $L > 0$, then the sensitivity of the solution of the system (1) with respect to initial conditions is given by $||\mathbf{x}(t;\mathbf{x}_0) - \mathbf{x}(t;\mathbf{x}_0 + \delta\mathbf{x}_0)|| \le e^{L(t-t_0)}||\delta\mathbf{x}_0||$.

Figures 4 and 5 show information about the convergence of both methods like in the previous example. We have only added in Figure 5 (left) a plot that shows convergence of $\mathbf{x}_0^\ell$. The main difference with respect to the previous example is that convergence not only is slower for both methods but also it is not as smooth as before. The convergence curves oscillate a lot more due to a destabilization effect of the unknown initial conditions. However, Figure 5 (right) shows that the overall numerical reproduction of the true solution $\mathbf{x}(t)$ is much better than in the previous example (all curves are very close to each other).

Example 4. *This last example includes numerical results obtained with the BFGS algorithm only, but with perturbations noise_level $= 0.05, 0.2$ for the generation of synthetic noisy measurements. Table 4 summarizes the numerical results. Comparing these results with those for BFGS in Table 2 we observe that the speed of convergence decreases with increasing noise_level for the same tolerance. Also, since the initial guess $\mathbf{x}_0^0$ is closer to exact $\mathbf{x}_0$*

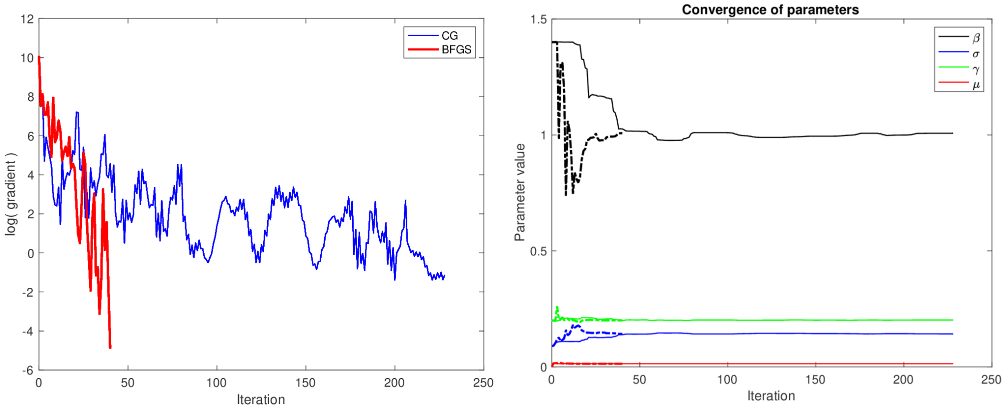

Figure 4: Left: gradient behaviour obtained with the CG (blue line) and the BFGS (red line) algorithms. Right: convergence of $\boldsymbol{\theta}$ with CG (continuous lines) and BFGS (dashed lines) algorithms.

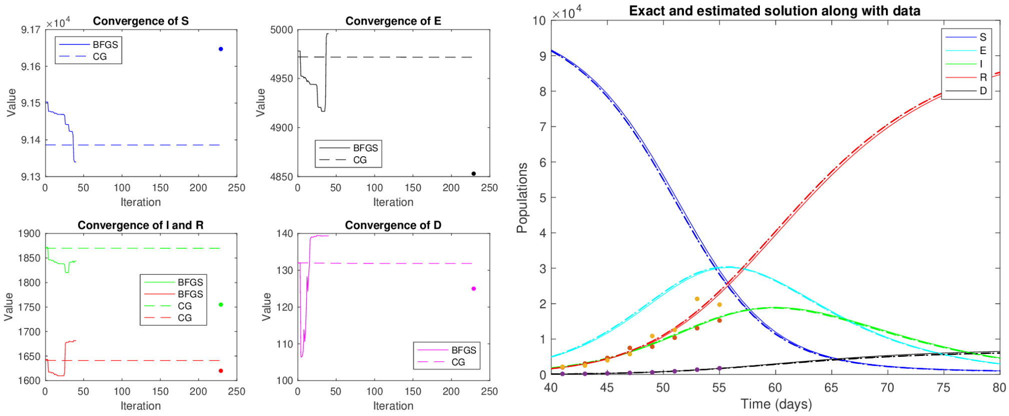

Figure 5: Left: convergence of $\mathbf{x}_0$ with the CG (dashed lines) and BFGS (continuous lines) algorithms. Right: dynamics of the true solution (continuous lines) and the numerical obtained from $(\mathbf{x}_0, \boldsymbol{\theta})$ computed with CG (dased line) and BFGS (dash-pointed line) algorithms, and noisy data for observable variables (I_i, R_i, D_i) (points)

Table 3: Effect of the noisy data on the numerical results for $\boldsymbol{\theta}$ and $\mathbf{x}_0$ computed with the BFGS algorithm.

noise_level	0.05 (5%)	0.15 (15%)
$\boldsymbol{\theta}^0$	$(1.40, 0.09, 0.20, 0.001)$	$(1.40, 0.09, 0.20, 0.001)$
$\mathbf{x}_0^0$	$(91645, 4782, 1844, 1605, 124)$	$(90474, 5782, 1628, 987, 129)$
Data time window	$t_i = 40 + i,\ 1 \leq i \leq 10$	$t_i = 40 + 2i,\ 1 \leq i \leq 8$
ϵ, no. iters.	10^{-8}, 30	10^{-8}, 43
Computed $\boldsymbol{\theta}$	$(1.0256, 0.1356, 0.2055, 0.0143)$	$(0.9953, 0.1385, 0.2126, 0.0153)$
Relative error	$(0.0256, 0.0508, 0.0277, 0.0004)$	$(0.0047, 0.0307, 0.0628, 0.0705)$
Computed $\mathbf{x}_0$	$(91546, 4795, 1886, 1646, 127)$	$(90759, 5780, 1637, 1690, 135)$
Relative error	$(0.0011, 0.0119, 0.0746, 0.0160, 0.0194)$	$(0.0097, 0.1910, 0.0675, 0.0433, 0.0770)$

for the 5% noisy case, then the accuracy of the computed value is better in this case. This sensitivity is not so evident in the calculation of the parameters $\boldsymbol{\theta}$, since the achieved accuracy is comparable. Another feature is that the time window of noisy data is wider the higher the noise_level to achieve convergent results. Figures 8 and 9 illustrate the performance of the BFGS algorithm with respect to noise_level.

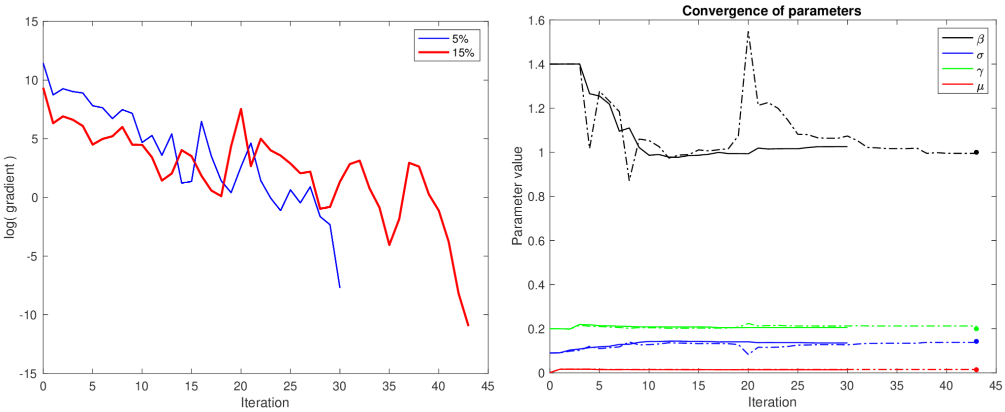

Figure 6: Left: plot of $\log(\nabla L(\mathbf{x}_0, \boldsymbol{\theta}))$ against iteration obtained with the BFGS algorithm for 5% noisy data (blue line) and 15% noisy data (red line). Right: convergence of $\boldsymbol{\theta}$ for 5% noisy data (continuous lines) and 15% noisy data (dashed lines).

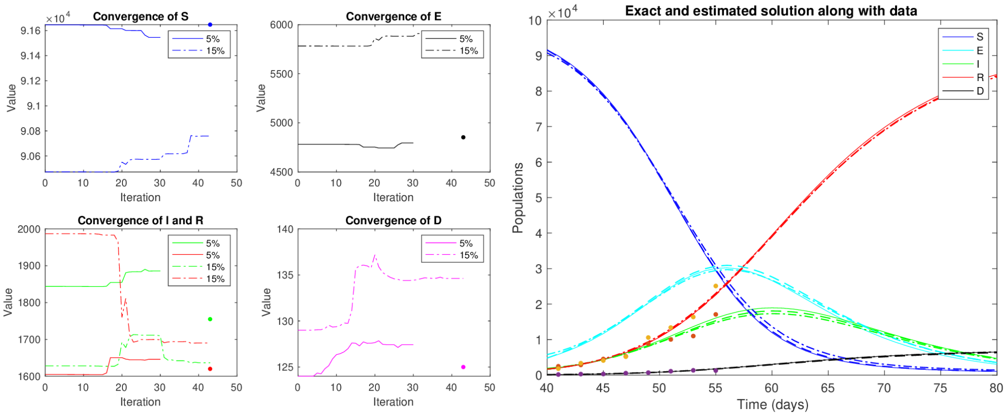

Figure 7: Left: convergence of $\mathbf{x}_0$ for 5% noisy data (continuous lines) and 15% noisy data (dashed lines). Right: dynamics of the true solution (continuous lines) and of the numerical solution obtained with 5% noisy data (dashed lines) and 15% noisy data (dash-pointed lines). Here we only show the points with 15% noisy data (see Table 2).

Another way to enhance convergence of the optimization algorithms is adding observable variables. For instance, adding E as observable variable for the case of 15% noise and using the same numerical parameters in Table 4, the BFGS algorithm converges in 27 iterations for the given tolerance. The best improvement, besides the faster convergence, is the estimation of the initial conditions, as shown in Table 4.

Figures 8 and 9 show the corresponding improvements.

Table 4: Numerical results obtained with the BFGS algorithm, adding E as observable variable.

Parameter	Computed	Relative error
$\boldsymbol{\theta}$	$(1.0106, 0.1415, 0.2131, 0.0153)$	$((0.0106, 0.0092, 0.0657, 0.0731)$
$\mathbf{x}_0$	$(91154, 5381, 1632, 1698, 135)$	$(0.0054, 0.1088, 0.0704, 0.0480, 0.0821)$

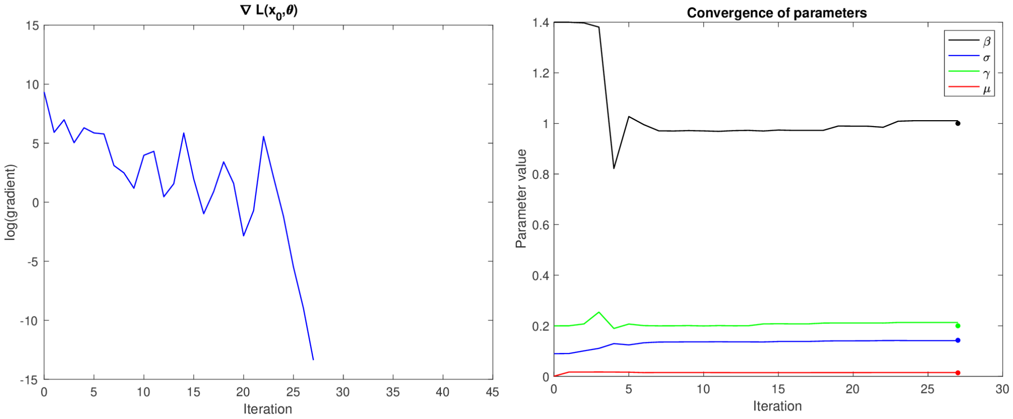

Figure 8: Left: plot of $\log(\nabla L(\mathbf{x}_0, \boldsymbol{\theta}))$ against iteration obtained with the BFGS algorithm for 15% noisy data and observable variables (E, I, R, D). Right: convergence of $\boldsymbol{\theta}$

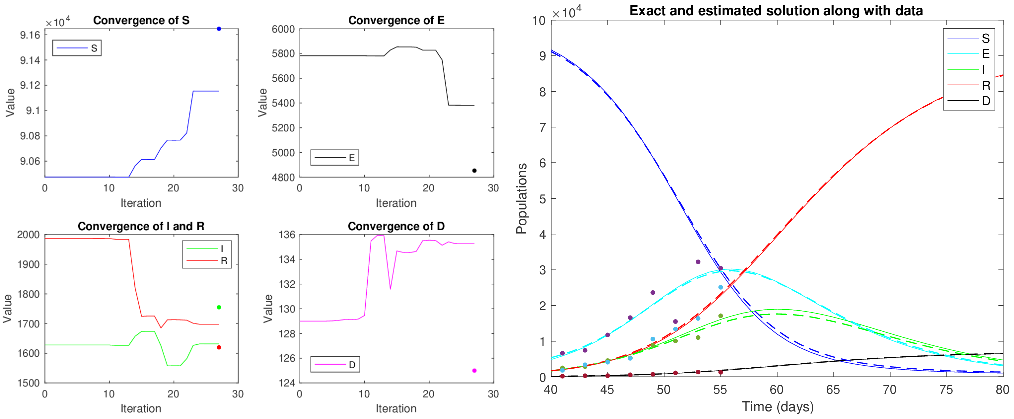

Figure 9: Left: convergence of $\mathbf{x}_0$ for 15% noisy data and observable variables (E, I, R, D). Right: dynamics of the true solution (continuous lines) and of the numerical solution (dashed lines).

6 Conclusions

We have introduced a deterministic model for fitting observed noisy data into a given dynamical system to find initial conditions and the parameters of the associated system of ordinary differential equations. The classical CG and BFGS optimization algorithms are employed to minimize the quadratic non-linear cost function. It is shown the advantage of using the adjoint equation approach to find the derivatives or gradients. We explain with some detail the implementation of this methods and algorithms with the SEIRD epidemiological model. However, this approach can be equally applied to other problems modelled by ODEs.

Similar numerical results are obtained with both algorithms, CG and BFGS using the same tolerance to achieve a given accuracy, but as expected the BFGS algorithm has better convergence properties and it is more robust. Numerical results show that more experimental data points and more observable variables increase the convergence properties of these algorithms. On the other hand, the higher the noise of the experimental data the slower is the convergence of the optimization algorithms. The main drawback of the proposed methodology is that it is sensitive to the location of noisy data and also to the initial guesses for initial conditions. However, if the algorithms converge properly, then the numerical results obtained are more accurate when $\mathbf{x}_0$ is also estimated along with $\boldsymbol{\theta}$.

For future work, we want to overcome some difficulties or deficiencies that arise with the proposed model and numerical algorithms. First, we must include explicitly into the fitting model (6) the positivity constraint of the unknown parameters, $\mathbf{x}_0$ and $\boldsymbol{\theta}$, specially for those that are relatively small and extend the proposed algorithms accordingly, like in [8]. The inherent instability and difficulty to find the initial conditions may be fixed incorporating the technique of multiple shooting, e.g. [7, 2]. Concerning the efficiency of optimization algorithms, we still need to test the Gauss-Newton method and if necessary its variant, the Levenberg-Marquardt algorithm. Finally, as mentioned before, the line search strategy is crucial for gradient descent algorithms. We may improve the performance of the secant method incorporating bracketing strategies like in [1], or trying the Newton's method as mentioned in remark 7.

References

[1] I. K. Argyros, M. A. Hernández-Verón, and M. J. Rubio. *Current Trends in Mathematical Analysis and Its Interdisciplinary Applications*, chapter On the Convergence of Secant-Like Methods. Springer Verlag, 2019.

[2] O. Aydogmus and A. H. TOR. A modified multiple shooting algorithm for parameter estimation in odes using adjoint sensitivity analysis. *Applied Mathematics and Computation*, 390:125644, 2021.

[3] J. R. Banga, C. G. Moles, and A. A. Alonso. Global optimization of bioprocesses using stochastic and hybrid methods. In Springer, editor, *Frontiers in Global Optimization. Nonconvex Optimization and Its Applications*, volume 74, 2004.

[4] B. Calderhead and M. Girolami. Estimating bayes factors via thermodynamic integration and population mcmc. *Computational Statistics & Data Analysis*, 53(12):4028–4045, 2009.

[5] J. Calver and W. Enright. Numerical methods for computing sensitivities for odes and ddes. *Numerical Algorithms*, 74:1101–1117, 2016.

[6] Y. Cao, S. Li, L. Petzold, and R. Serban. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089, 2003.

[7] F. Carbonell, Y. Iturria-Medina, and J. C. Jimenez. Multiple shooting-local linearization method for the identification of dynamical systems. *Communications in Nonlinear Science and Numerical Simulation*, 37:292–304, 2016.

[8] M. V. Chávez-Hernández, L. H. Juárez-Valencia, and Y. Á. Ríos-Solís. Penalization and augmented lagrangian for o-d demand matrix estimation from transit segment counts. *Transportmetrica A Transport Science*, 15(2):915–943, 2019.

[9] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1996.

[10] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[11] I. F. D. Oliveira and R. H. C. Takahashi. An enhancement of the bisection method average performance preserving minmax optimality. *ACM Transactions on Mathematical Software*, 47(1):1–24, 2020.

[12] E. L. Piccolomini and F. Zama. Monitoring italian covid-19 spread by an adaptive seird model. *medRxiv*, 2020.

[13] J. O. Ramsay, G. Hooker, and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007.

[14] B. Sengupta, K. Friston, and W. Penny. Efficient gradient computation for dynamical models. *NeuroImage*, 98:521–527, 2014.

[15] W. Sun and Y.-X. Yuan. *Optimization Theory and Methods: Nonlinear Programming*. Springer New York, 2006.

Spatio-temporal point process analysis of Mexico State wildfires

Luis Ramón Munive-Hernández[1] and Antonio Villanueva-Morales[2]

[1]Departamento de Matemáticas, Universidad Autónoma Metropolitana
[1]Colegio de Ciencias y Humanidades, Universidad Autónoma de la Ciudad de México
[2]Departamento de Estadística, Matemática y Cómputo, Universidad Autónoma Chapingo

Abstract

Wildfires are an example of a phenomenon that can be investigated using point process theory. We analyze public data from the National Forestry Commission. It consists of wildfire records, specifically their coordinates and dates of occurrence in Mexico State from 2010 to 2018. The spatial component was examined, and we found that wildfires tend to cluster. Afterwards, a time series analysis was conducted. This shows that the data comes from a stationary stochastic process. Finally, some spatio-temporal features that demonstrate the point process' regular behavior in space and time were investigated. This research could be a reference to describe wildfire behavior in a specific space and time.

Keywords: Environmental statistics, point processes, spatio-temporal statistics, wildfires.

1 Introduction

Wildfires are complex phenomena with serious socio-environmental consequences, including economic and biodiversit losses, among others. Anthropogenic factors are responsible for nearly all wildfires in Mexico State, according to data from the National Forestry Commission (Conafor, its Spanish acronym) [9] (see Figure 1).

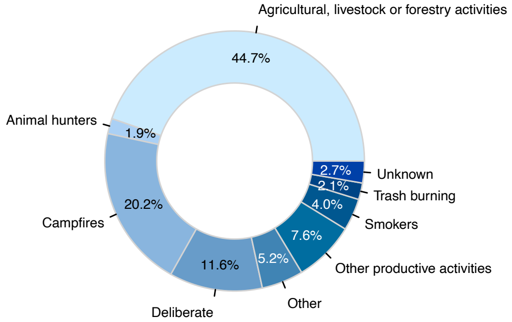

Figure 1: Mexico State wildfire causes (2010-2018).

There is plenty of specialized literature available on wildfires (see [22] and [23]). The authors of [7] use a logistic regression model to assess the risk of wildfire in Puebla, Mexico, taking into account land cover, meteorological, topographic, and social variables. Using two different data sources: Conafor's open data and Modis' (Moderate Resolution Imaging Spectroradiometer) data, the authors of [8] show that wildfire spatial patterns in Mexico tend to cluster. The spatial and temporal relationships between Conafor's wildfire records from 2005 to 2015 and the Standardized Precipitation-Evapotranspiration Index (SPEI) were investigated [19]. Machine learning techniques were used to determine the wildfire propensity in Mexico using Conafor's open data [18].

The spatio-temporal behavior of wildfires could be critical for improving fire management strategies. The point processes approach can be used to model random events in time, space, or space-time, such as wildfires. In this study, we used point processes theory to describe the spatio-temporal behavior of wildfires in Mexico State from 2010 to 2018.

2 Point processes basic theory

A point process is a random set in which the number of points and their locations are both random [2, p. 12]. A point process could occur in any completely separable metric space $\mathcal{S}$, such as d-dimensional Euclidean space $\mathbb{R}^d$.

Definition 1 (Point process)**.** *The point process Y, with state space $\mathcal{S}$, is a measurable mapping from a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ to the measure space of the point process' realizations equipped with the counting measure, $\left(\mathcal{Y}_{\mathcal{S}}^{\#}, \mathscr{B}\left(\mathcal{Y}_{\mathcal{S}}^{\#}\right), \mu_{\#}\right)$.*

Where $\mathcal{Y}_{\mathcal{S}}^{\#} = \{\mu_{\#} : \mathscr{B}(\mathcal{S}) \to \mathbb{N} \mid \ \mu_{\#}(A) < \infty, A \in \mathscr{B}(\mathcal{S})\}$ is the space of all finite counting measures on a σ-algebra $\mathscr{B}(\mathcal{S})$ of subsets of $\mathcal{S}$, $\mathscr{B}\left(\mathcal{Y}_{\mathcal{S}}^{\#}\right)$ is a σ-algebra of subsets of the space $\mathcal{Y}_{\mathcal{S}}^{\#}$ and $\mu_{\#}$ is the counting measure.

The commutative diagram in Figure 2 illustrates the point process definition.

$$\begin{tikzcd}
(\Omega, \mathscr{F}, \mathbb{P}) \arrow[rr, shift left, "Y : \omega \mapsto Y(\omega, \cdot)"] \arrow[dr, bend right, "Y_A : \omega \mapsto Y_A(\omega)"] & & \left(\mathcal{Y}_{\mathcal{S}}^{\#}, \mathscr{B}\left(\mathcal{Y}_{\mathcal{S}}^{\#}\right), \mu_{\#}\right) \arrow[ll, shift left, "Y^{-1}\left(\varphi_A^{-1}(B)\right) \mapsfrom \varphi_A^{-1}(B) : Y^{-1}"] \arrow[dl, bend left, "\mu_{\#}(A) \mapsfrom \mu_{\#} : \varphi_A"'] \\
& (\mathbb{N}, \mathcal{P}(\mathbb{N})) \arrow[ul, bend left=50, "Y_A^{-1}(B) \mapsfrom B : Y_A^{-1}"] \arrow[ur, bend right=50, "\varphi_A^{-1} : B \mapsto \varphi_A^{-1}(B)"'] &
\end{tikzcd}$$

Figure 2: Commutative diagram of point process definition.

The mapping φ_A takes measures $\mu_{\#} \in \mathcal{Y}_{\mathcal{S}}^{\#}$ and maps them into $\mu_{\#}(A)$. As a result, the mapping φ_A in terms of the point process Y is $\varphi_A : Y(\omega, \cdot) \mapsto Y(\omega, A)$.

Furthermore, the commutative diagram reveals the equivalences: $Y(\omega, A) = \varphi_A(Y(\omega, \cdot)) = Y_A(\omega)$ and $Y_A^{-1}(B) = Y^{-1}\left(\varphi_A^{-1}(B)\right)$, for any $B \in \mathcal{P}(\mathbb{N})$, where $\mathcal{P}(\mathbb{N})$ denotes the power set of $\mathbb{N}$, so $(\mathbb{N}, \mathcal{P}(\mathbb{N}))$ is a measurable space, [11, pp. 8–9], [1, p. 13].

The following are some fundamental properties of a point process [1, pp. 7–8]:

i. Is additive, this is
$$Y(\omega, A_1 \cup A_2) = Y(\omega, A_1) + Y(\omega, A_2),$$
whenever $A_1 \cap A_2 = \varnothing$, $A_1, A_2 \subset \mathcal{S}$ and of course
$$Y(\omega, \varnothing) = 0.$$

ii. Is locally finite
$$\mathbb{P}(Y(\omega, A) < \infty) = 1,$$
for any $A \subset \mathcal{S}$.

iii. Is simple
$$\mathbb{P}(Y(\omega, \{\boldsymbol{s}\}) \leq 1) = 1,$$
for any point $\boldsymbol{s} \in \mathcal{S}$.

For simplification, we will write $Y(\omega, A) = Y(A)$ in the foregoing. When the point process Y is observed, we have a point pattern denoted by $\boldsymbol{Y}$.

In order to generate models, some assumptions about a point process must be made. Stationarity and isotropy are the most important assumptions. The former refers to statistical invariance under translations, whereas the latter refers to statistical invariance under rotations [1, p. 16], [3, pp. 146–147]. Nonetheless, some research on non-stationary and anisotropic processes has been conducted (see [14] and [26, ch. 5]).

Definition 2 (Stationary point process)**.** *A point process Y on $\mathcal{S}$ is stationary if, for any fixed $\boldsymbol{s} \in \mathcal{S}$, the distribution of the process $Y + \boldsymbol{s}$ is identical to the distribution of Y.*

2.1 Poisson process

The general Poisson point process in some space $\mathcal{S}$ can be defined as follows [1, p. 12], [3, pp. 300–301].

Definition 3 (Poisson process)**.** *The Poisson process Y on $\mathcal{S}$ with intensity measure Λ is a point process such that:*

i. For every compact set $A \subset \mathcal{S}$, the random variable $Y(A) \sim \text{Poisson}(\Lambda(A))$.

ii. If $A_1, \ldots, A_n \subset \mathcal{S}$ are disjoint compact sets, then $Y(A_1), \ldots, Y(A_n)$ are independent random variables.

Where the intensity measure Λ is defined, for any $A \subset \mathcal{S}$, as $\Lambda(A) = \mathbb{E}(Y(A))$.

If the state space is $\mathcal{S} = \mathbb{R}^2 \times \mathbb{R}_+$ and the expected value of the point process Y in $S \times T$, with $S \subset \mathbb{R}^2$ and $T \subset \mathbb{R}_+$, can be written as follows:

$$\mathbb{E}(Y(S \times T)) = \lambda\, \mu_L(S)\, \mu_L(T),$$

where $\lambda > 0$ and μ_L is the Lebesgue measure, then we have the spatio-temporal homogeneous Poisson point process [15, pp. 9–10].

The simplest stochastic mechanism for generating point patterns is the homogeneous Poisson point process. As a data model, it is almost never plausible. Regardless, it is the fundamental reference or benchmark model of a point process [2, p. 53].

The homogeneous Poisson point process is also known as complete spatial (or spatio-temporal) randomness. Additionally, the Poisson point process is stationary and isotropic [1, p. 16].

Figure 3 depicts a spatial point pattern generated by a homogeneous Poisson point process.

3 Point pattern's data analysis

Distances between points are a straightforward way to examine a point pattern. The most common statistics used in exploratory analysis of a point pattern are as follows.

3.1 Empty-space function F

Let Y be a stationary point process on $\mathcal{S}$. The shortest distance between a given point $\boldsymbol{s} \in \mathcal{S}$ and the nearest observed point $\boldsymbol{y}_i \in \boldsymbol{Y}$ is denoted as $\mathrm{d}(\boldsymbol{s}, \boldsymbol{Y}) = \min_i \{\|\boldsymbol{s} - \boldsymbol{y}_i\|\}$. It is called the empty-space distance, spherical contact distance, or simply contact distance [2, p. 83], [1, pp. 21–22], [3, pp. 261–262].

Note that

$$\mathrm{d}(\boldsymbol{s}, \boldsymbol{Y}) \leq r \Leftrightarrow Y(B_r(\boldsymbol{s})) > 0, \tag{1}$$

where $B_r(\boldsymbol{s})$ is the neighborhood of radius r centered on $\boldsymbol{s}$.

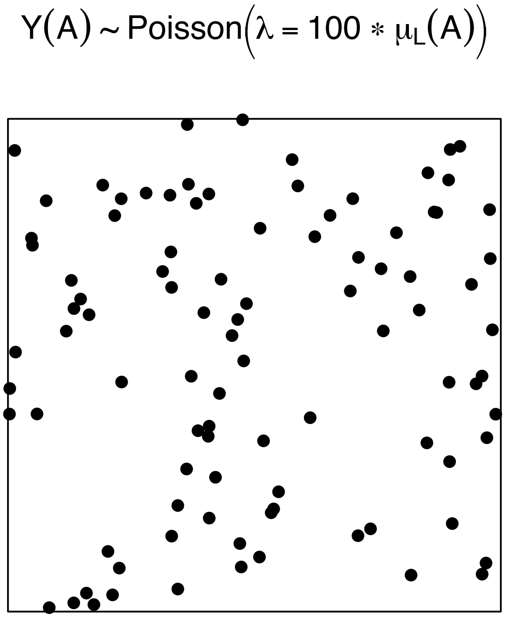

Figure 3: Simulation of a spatial homogeneous Poisson process.

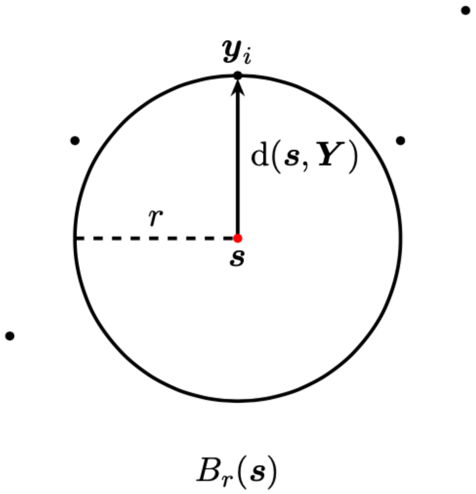

Figure 4: Empty-space distance illustration.

In other words, as shown in Figure 4, the empty-space distance satisfies the logical equivalence of the biconditional (1), $\mathrm{d}(\boldsymbol{s}, \boldsymbol{Y}) > r \Leftrightarrow Y\left(B_r(\boldsymbol{s})\right) = 0$.

Moreover, because $\{Y\left(B_r(\boldsymbol{s})\right) > 0\}$ is measurable, the event $\{\mathrm{d}(\boldsymbol{s}, \boldsymbol{Y}) \leq r\}$ is measurable, implying that the contact distance is a well-defined random element.

Definition 4 (Empty-space function F)**.** *Let Y be a stationary point process on $\mathcal{S}$. The empty-space function F is the cumulative distribution function of the empty-space distance*

$$F(r) = \mathbb{P}\left(\mathrm{d}(\boldsymbol{s}, \boldsymbol{Y}) \leq r\right).$$

If Y is a homogeneous Poisson process on $\mathbb{R}^d$ with intensity λ, then the empty-space function is

$$F(r) = 1 - \exp\left(-\lambda\, \mu_L\left(B_1(\mathbf{0})\right)\, r^d\right),$$

where $r \geq 0$, $\mu_L\left(B_1(\mathbf{0})\right) = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}$ denotes the volume of the unitary d-ball in $\mathbb{R}^d$ and Γ is the usual gamma function.

3.2 Nearest-neighbour function G

The nearest-neighbour distance, denoted by $\mathrm{d}_i = \min_{i \neq j}\left\{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|\right\}$, is the distance between each point $\boldsymbol{y}_i \in \boldsymbol{Y}$ and its nearest neighbour in the set $\boldsymbol{Y}\backslash\{\boldsymbol{y}_i\}$, [2, p. 90], [1, pp. 51–52]. It is worth noting that d_i can also be written as $\mathrm{d}_i = \mathrm{d}\left(\boldsymbol{y}_i, \boldsymbol{Y}\backslash\{\boldsymbol{y}_i\}\right)$, [3, p. 262]. This distance is depicted in Figure 5.

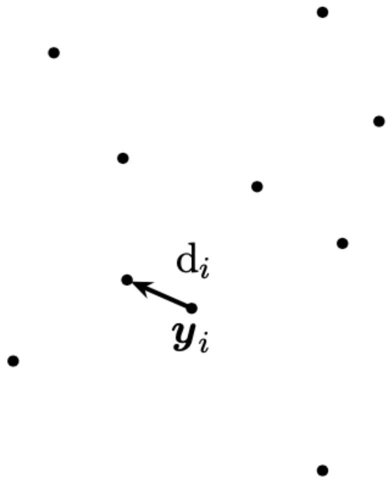

Figure 5: Nearest-neighbour distance illustration.

Definition 5 (Nearest-neighbour function G)**.** *Let Y be a stationary point process on $\mathcal{S}$. The nearest-neighbour function G is the cumulative distribution function of the nearest-neighbour distance*

$$G(r) = \mathbb{P}\left(\mathrm{d}\left(\boldsymbol{s}, \boldsymbol{Y}\backslash\{\boldsymbol{s}\}\right) \leq r \mid \ \boldsymbol{s} \in \boldsymbol{Y}\right),$$

where $r \geq 0$ and $\boldsymbol{s}$ is any location in the state space $\mathcal{S}$.

If Y is a homogeneous Poisson process on $\mathbb{R}^d$ with intensity λ, then the nearest-neighbour function is

$$G(r) = 1 - \exp\left(-\lambda\, \mu_L\left(B_1(\mathbf{0})\right)\, r^d\right).$$

In this case, we have that $F(r) = G(r)$, i.e., under complete spatial randomness, the points of the Poisson process are independent of each other, so conditioning does not affect them. Therefore, F is equivalent to G, [2, p. 91].

3.3 Intensity

The intensity function describes the first-order properties of a point process [10, p. 623], [12, p. 57].

The average number of points per spatial (or spatio-temporal) unit defines the intensity of a point process. In this regard, intensity is analogous to the expected value of a random variable [1, p. 26].

Similarly, we can investigate the analogue of a point process' variance or covariance throughout the second-order properties.

As we will see in the following, the intensity measure Λ of a point process Y is clearly a set function, whereas the "instantaneous" intensity function λ is an atomic function.

Definition 6 (First-order intensity)**.** *Let Y be a point process on $\mathcal{S}$. The first-order intensity is defined as*

$$\lambda(\boldsymbol{s}) = \lim_{\nu(\mathrm{d}\boldsymbol{s})\to 0} \frac{\mathbb{E}\left(Y(\mathrm{d}\boldsymbol{s})\right)}{\nu\left(\mathrm{d}\boldsymbol{s}\right)},$$

where ν is a suitable measure on $(\mathcal{S}, \mathscr{B}(\mathcal{S}))$ and $\mathrm{d}\boldsymbol{s}$ defines a infinitesimally small region around $\boldsymbol{s}$.

If Y is a point process on $\mathbb{R}^d$ with intensity measure Λ, it satisfies

$$\Lambda(A) = \int_A \lambda(\boldsymbol{s})\, \mu_L\left(\mathrm{d}\boldsymbol{s}\right),$$

for some function λ and any $A \subset \mathbb{R}^d$. Then λ is called the intensity function of Y [1, p. 27]. If λ is constant, then Y is said to be homogeneous, otherwise is said to be inhomogeneous [17, p. 40]. Likewise, if the intensity function exists, we can interpret it as follows:

$$\mathbb{P}\left(Y(\mathrm{d}\boldsymbol{s}) > 0\right) \approx \mathbb{E}\left(Y(\mathrm{d}\boldsymbol{s})\right) \approx \lambda(\boldsymbol{s})\, \mu_L\left(\mathrm{d}\boldsymbol{s}\right).$$

The K function and pair correlation are both second-moment properties, so the second-order intensity must be defined [12, p. 57].

Definition 7 (Second-order intensity)**.** *Let Y be a point process on $\mathcal{S}$. The second-order intensity is defined as*

$$\lambda_2(\boldsymbol{s}, \boldsymbol{u}) = \lim_{\substack{\nu(\mathrm{d}\boldsymbol{s})\to 0\\ \nu(\mathrm{d}\boldsymbol{u})\to 0}} \frac{\mathbb{E}\left(Y(\mathrm{d}\boldsymbol{s})\, Y(\mathrm{d}\boldsymbol{u})\right)}{\nu\left(\mathrm{d}\boldsymbol{s}\right)\, \nu\left(\mathrm{d}\boldsymbol{u}\right)}.$$

We already have the fundamental elements for defining the following pair of second-order properties.

3.4 K function

The K function counts the number of locations within a certain radius of a given point (see Figure 6), [3, p. 226], [5, p. 171]. Ripley defined it in [21]. We present the following definition [2, p. 92], [12, pp. 57–58].

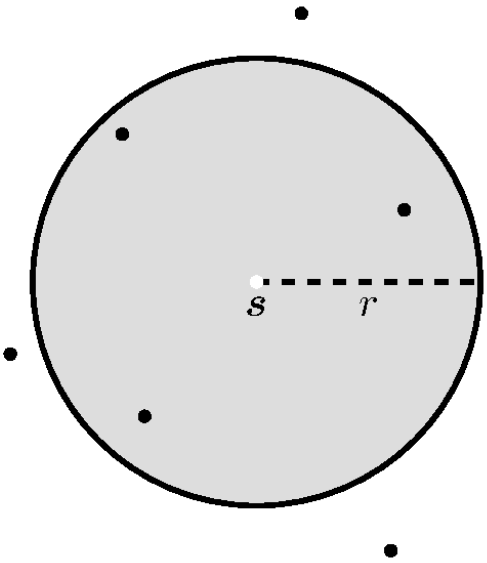

Figure 6: K function illustration.

Definition 8 (K function)**.** *Let Y be a stationary and isotropic point process on $\mathcal{S}$ with intensity λ. The K function is defined as*

$$K(r) = \frac{1}{\lambda}\mathbb{E}\left(Y\left(\boldsymbol{Y} \cap B_r(\boldsymbol{s})\backslash\{\boldsymbol{s}\}\right) \mid \ \boldsymbol{s} \in \boldsymbol{Y}\right),$$

where $r \geq 0$ and $\boldsymbol{s}$ is any location in $\mathcal{S}$.

If $\mathcal{S} = \mathbb{R}^d$ and the point process Y is assumed to be stationary, then hold $\lambda_2\left(\boldsymbol{s}, \boldsymbol{u}\right) = \lambda_2\left(\boldsymbol{s} - \boldsymbol{u}\right)$. Also, if Y is isotropic, hence $\lambda_2(\boldsymbol{s} - \boldsymbol{u}) = \lambda_2(r)$, where $r = \|\boldsymbol{s} - \boldsymbol{u}\|$. These conditions implies that [10, p. 633], [12, p. 58],

$$\lambda \, K(r) = \frac{d \, \mu_L\left(B_1(\mathbf{0})\right)}{\lambda} \int_0^r \lambda_2(z) \, z^{d-1} \, \mathrm{d}z. \tag{2}$$

The above expression provides a relationship between the K function and the second-order intensity under the assumptions of stationarity and isotropy.

If Y is a homogeneous Poisson process on $\mathbb{R}^d$, then the K function is [1, p. 38],

$$K(r) = \mu_L\left(B_1(\mathbf{0})\right) \, r^d.$$

3.5 Pair correlation function g

In general, the pair correlation function is a quotient of probabilities; that is, the probability of observing a pair of points separated by a given distance is divided by the same probability, assuming a Poisson point process [2, p. 94]. In the strictest sense, it is neither a distribution nor a correlation function [12, p. 57].

Some authors consider the pair correlation function to be the most informative second-order property because it provides information more simply than, say, the K function [16, p. 218]. We present the following definition [1, pp. 33–34], [17, p. 41].

Definition 9 (Pair correlation function g)**.** *Let Y be a point process on $\mathcal{S}$ with intensity function λ and second-moment density g_2. The pair correlation function g is defined as*

$$g(\boldsymbol{s},\boldsymbol{u}) = \frac{g_2(\boldsymbol{s},\boldsymbol{u})}{\lambda(\boldsymbol{s})\,\lambda(\boldsymbol{u})},$$

for any $\boldsymbol{s},\boldsymbol{u} \in \boldsymbol{Y}$, where the second-moment density is such that

$$\nu_{[2]}(C) = \int_C g_2(\boldsymbol{s},\boldsymbol{u})\,\nu\left(\mathrm{d}\boldsymbol{s}\right)\nu\left(\mathrm{d}\boldsymbol{u}\right),$$

for any compact set $C \subset \mathcal{S}\times\mathcal{S}$, where ν is a suitable measure on $(\mathcal{S},\mathscr{B}(\mathcal{S}))$ (e.g., if $\mathcal{S}=\mathbb{R}^d$, so $\nu=\mu_L$), and $\nu_{[2]}(A_1\times A_2)=\mathbb{E}\left(Y(A_1)\,Y(A_2)\right)-\mathbb{E}\left(Y\left(A_1\cap A_2\right)\right)$, with $A_1,A_2\subset\mathcal{S}$, is the second factorial moment measure of Y.

If Y is stationary and isotropic, it follows from (2) that [12, p. 58], [16, p. 219],

$$g(r) = \frac{K'(r)}{d\,\mu_L\left(B_1(\mathbf{0})\right)\,r^{d-1}}.$$

We can define g graphically by taking two concentric circles with radius r and $r+\Delta r$, where Δr is a small increment, and counting the points that fall within the ring (see Figure 7), [3, pp. 225–226].

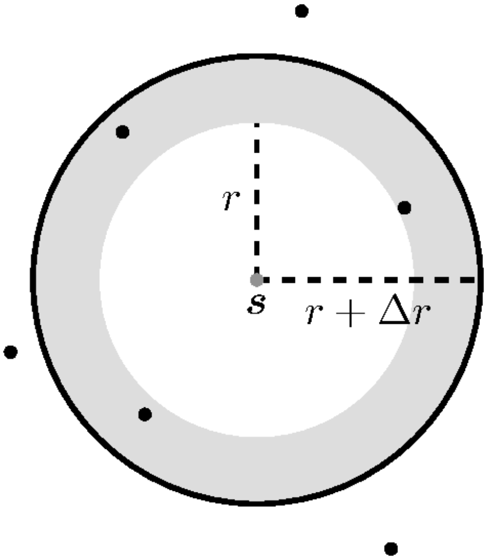

Figure 7: Pair correlation function g illustration.

If Y is stationary and isotropic, the expected number of locations in the ring is $\lambda\,K(r+\Delta r)-\lambda\,K(r)$. Dividing it by the expected value of points assuming a Poisson process, we obtain

$$\begin{aligned} g_{\Delta r}(r) &= \frac{\lambda\left(K(r+\Delta r)-K(r)\right)}{\lambda\,\mu_L\left(B_1(\mathbf{0})\right)\left(\left(r+\Delta r\right)^d - r^d\right)} \\ &= \frac{K(r+\Delta r)-K(r)}{\mu_L\left(B_1(\mathbf{0})\right)\left(\sum_{k=0}^{d}\binom{d}{k}r^{d-k}\Delta r^k - r^d\right)}. \end{aligned} \tag{3}$$

All binomial expansion components in the denominator of the second line in (3) lose significance except for $d\ r^{d-1}\Delta r$, so

$$g_{\Delta r}(r) \approx \frac{K(r+\Delta r)-K(r)}{\mu_L\left(B_1(\mathbf{0})\right)\ d\ r^{d-1}\Delta r}.$$

Taking the following limit, we get

$$\begin{aligned}\lim_{\Delta r \to 0} g_{\Delta r}(r) &\approx \lim_{\Delta r \to 0} \frac{K(r+\Delta r)-K(r)}{d\ \mu_L\left(B_1(\mathbf{0})\right)\ r^{d-1}\ \Delta r}\\ &= \frac{K'(r)}{d\ \mu_L\left(B_1(\mathbf{0})\right)\ r^{d-1}}\\ &= g(r).\end{aligned}$$

If Y is a homogeneous Poisson process on $\mathbb{R}^d$, then the pair correlation function is $g(r) = 1$.

4 Wildfires' data analysis

Conafor data are licensed for free use (see details in `https://datos.gob.mx/libreusomx`). It includes wildfire geographical coordinates and dates, as well as variables like forest type affected and severity, among other things.

4.1 Spatial analysis

This spatial analysis focuses on the F and G functions to determine whether the wildfire spatial point pattern is aggregated, complete spatial random, or regular. In addition, the intensity was estimated to support the evidence about point pattern behavior.

Plotting the spatial point pattern is a good starting point for understanding its behavior.

Figure 8 shows the spatial point pattern. The wildfires do not appear to be the result of a Poisson process.

There are multiple ways to prove if a point pattern comes from a Poisson point process (see [3, ch. 10]).

The simulation envelopes provide a formal way to decide if the spatial pattern comes from the Poisson process. It is equivalent to performing a hypothesis test. The simulation envelopes are obtained under the assumption of a Poisson process [2, pp. 98–99], [3, pp. 268–271], [5, pp. 161–163].

If the empirical curve falls within the envelope, we can conclude that the point pattern comes from a Poisson process.

Figures 9 and 10 show the estimated F and G functions, as well as the theoretical functions for the Poisson process and simulation envelopes. For this, we use the `R` package `spatstat` [4].

Clearly, the spatial point pattern does not follow the Poisson model.

In Figure 9 note that $\widehat{F}_{\text{obs}}(r) < F_{\text{theo}}(r)$, i.e., the point pattern has longer empty-space distances than a Poisson process. This suggests a clustered point pattern [2, p. 86]. While in Figure 10 we observe that $\widehat{G}_{\text{obs}}(r) > G_{\text{theo}}(r)$, i.e., the point pattern has shorter nearest-neighbour distances than a Poisson model, indicating a clustered pattern [2, p. 91].

Figure 11 depicts the estimated intensity using a Gaussian kernel with bandwidth of 17 km. It can be used to locate wildfire hotspots.

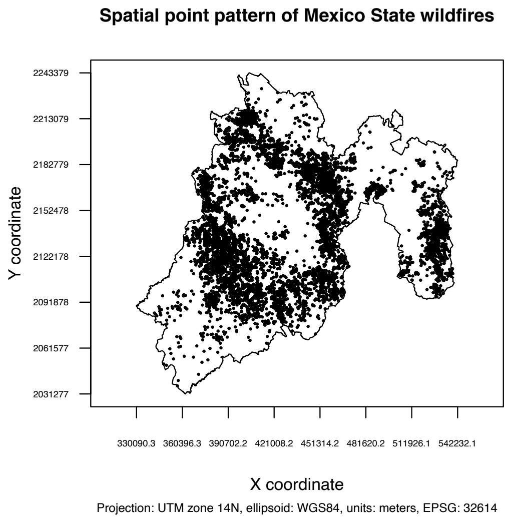

Figure 8: Spatial point pattern of Mexico State wildfires.

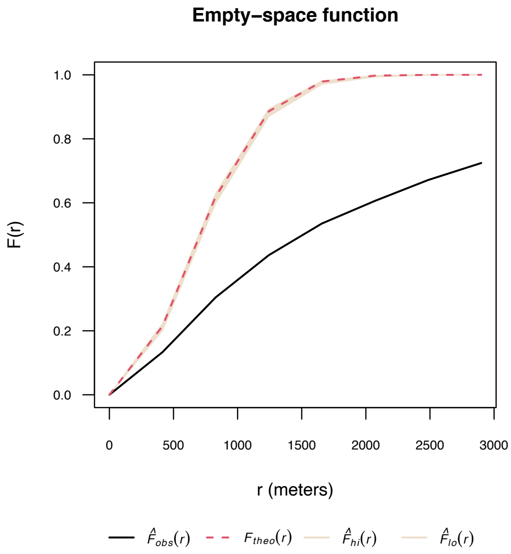

Figure 9: Estimated F function and simulation envelopes.

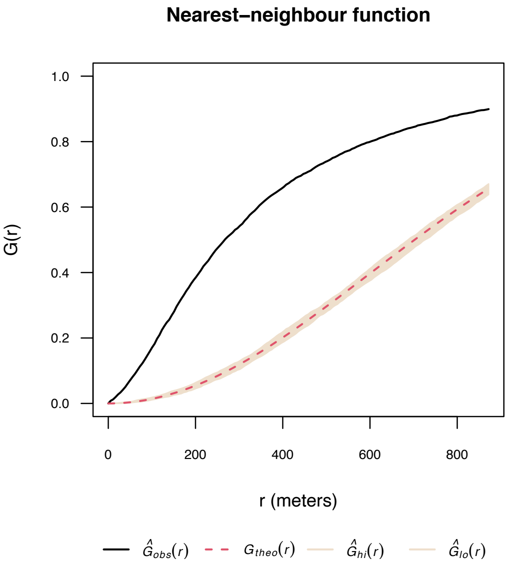

Figure 10: Estimated G function and simulation envelopes.

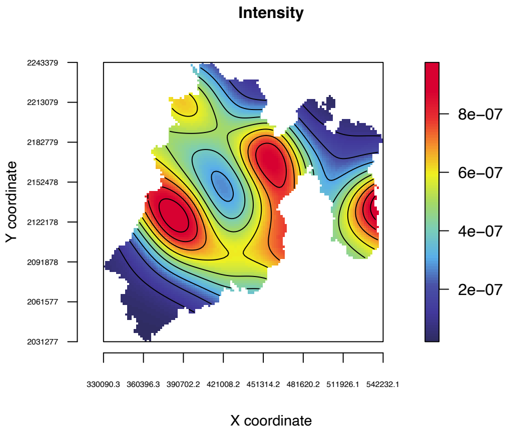

Figure 11: Estimated intensity.

4.2 Time series analysis

This time series analysis was carried out to describe the temporal behavior of wildfires. Figure 12 displays the daily number of wildfires. This immediately suggests that the wildfire time series is seasonal.

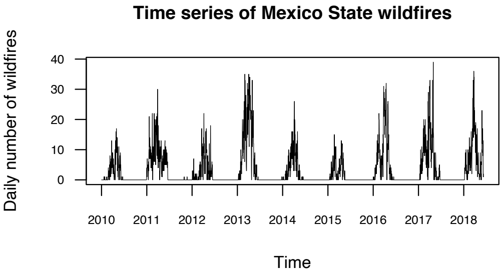

Figure 12: Time series of Mexico State wildfires.

The augmented Dickey-Fuller test is used to prove that the time series is seasonal (see details in [6, pp. 169–173]). This test is included in the `R` package `tseries` [25], where the null hypothesis is that the time series is non-stationary, against the alternative hypothesis that the time series is stationary.

Table 1 displays the results of the augmented Dickey-Fuller test for the wildfire time series, with a significance level of $\alpha = 0.05$.

Test statistic	p-value
-5.1037	< 0.01

Table 1: Augmented Dickey-Fuller test results.

4.3 Spatio-temporal analysis

To demonstrate clustering or regularity in a spatio-temporal point pattern, the space-time inhomogeneous K function (STIK) and space-time pair correlation function (STPC) can be used [15, p. 6].

On the assumption that the point process Y on $\mathbb{R}^d$ is second-order stationary, that is, their first-order and second-order properties are invariant under translations, the K function is [13, p. 45],

$$K(r) = d\,\mu_L\left(B_1(\mathbf{0})\right) \int_0^r g(z)\; z^{d-1}\mathrm{d}z. \tag{4}$$

In addition, a spatio-temporal point process is second-order intensity reweighted stationary and isotropic if its intensity function is bounded away from zero, and its g function is solely determined by (u, v), where $u = \|\boldsymbol{s}_i - \boldsymbol{s}_j\|$ and $v = |t_i - t_j|$, with $\boldsymbol{s}_i, \boldsymbol{s}_j \in \mathbb{R}^2$, $t_i, t_j \in \mathbb{R}_+$, [15, p. 3].

Let Y be a second-order intensity reweighted stationary and isotropic spatio-temporal point process with intensity λ; then, from (4), its STIK function is, [15, p. 6], [13, p. 45],

$$K_{ST}(u, v) = 2\pi \int_0^v \int_0^u g(w, z)\; w\; \mathrm{d}w\; \mathrm{d}z,$$

where $g(u, v) = \frac{\lambda_2(u,v)}{\lambda(\boldsymbol{s}_i,t_i)\;\lambda(\boldsymbol{s}_j,t_j)}$ is the spatio-temporal pair correlation function g of Y.

For any inhomogeneous spatio-temporal Poisson process with intensity bounded away from zero,

$$K_{ST}(u, v) = \pi u^2 v.$$

Figures 13 and 14 show the estimated STIK function in contour and perspective plots, respectively.

The values $\widehat{K}_{ST}(u,v) - \pi u^2 v$ were plotted in order to use them as a measure of spatiotemporal aggregation or regularity. According to [13, p. 45], $\widehat{K}_{ST}(u,v) - \pi u^2 v < 0$ indicates regularity.

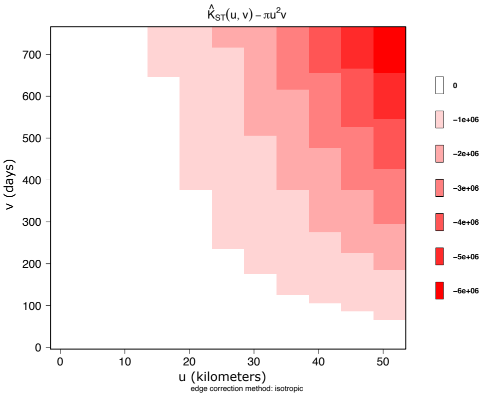

Figure 13: Estimated STIK function contour plot.

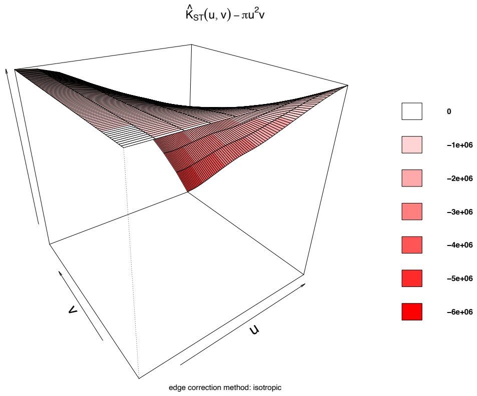

Figure 14: Estimated STIK function perspective plot.

Figures 15 and 16 illustrate estimated STPC function in contour and perspective plots, respectively.

For a spatio-temporal Poisson point process, $g(u,v) = 1$. This reference can be used to determine how much more or less likely it is that a pair of events will occur at specific locations than in a Poisson process of equal intensity [15, p. 3].

Surface behavior is regular; that is, there is yearly seasonality at distances less than 10 km, implying spatio-temporal regularity.

5 Conclusions and perspectives

The spatio-temporal point pattern of Mexico State wildfires from 2010 to 2018 tends to cluster spatially, as shown by Figures 8, 9, 10, and 11.

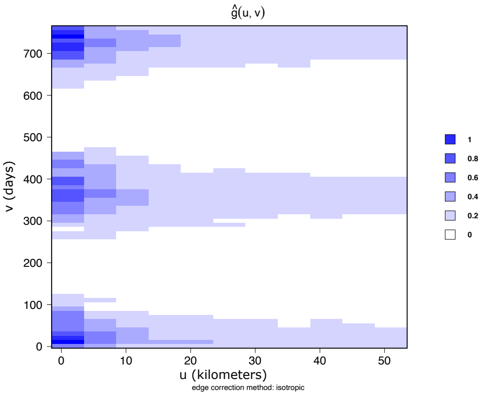

Figure 15: Estimated STPC function contour plot.

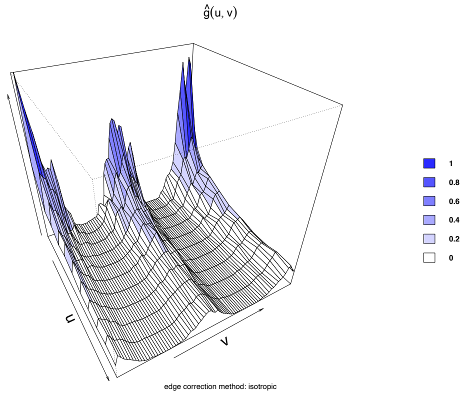

Figure 16: Estimated STPC function perspective plot.

While the temporal behavior is stationary, as illustrated in Figure 12 and Table 1, there is a yearly wildfire season during the first semester of each year.

Finally, as shown in Figures 13, 14, 15, and 16, we demonstrate that the spatio-temporal behavior is regular. This means that wildfires tend to occur in the same season and in the same areas each year. This regular spatio-temporal behavior suggests that the underlying point process is predictable in some ways.

This research could be expanded by looking into models such as spatio-temporal log-Gaussian Cox processes [24], which can be used to make spatio-temporal predictions.

Acknowledgments

The authors would like to express their gratitude to the Universidad Autónoma Chapingo.

Appendix

This analysis was performed using the statistical programming language `R` [20]. The developed code is available in the repository:
`https://github.com/LuisMunive/Spatio-temporal-point-process-analysis-of-Mexico-State-wildfires`.

References

[1] A. Baddeley, I. Bárány, and R. Schneider. Spatial point processes and their applications. *Stochastic Geometry: Lectures Given at the CIME Summer School Held in Martina Franca, Italy, September 13–18, 2004*, pages 1–75, 2007.

[2] A. Baddeley et al. Analysing spatial point patterns in R. In *Workshop notes version*, volume 3, 2008.

[3] A. Baddeley, E. Rubak, and R. Turner. *Spatial point patterns: methodology and applications with R.* CRC Press, 2015.

[4] A. Baddeley and R. Turner. spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42, 2005.

[5] R. S. Bivand, E. J. Pebesma, V. Gómez-Rubio, and E. J. Pebesma. *Applied spatial data analysis with R*, volume 747248717. Springer Science & Business Media, 2008.

[6] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting.* Springer Science & Business Media, 2016.

[7] R. L. Carrillo-García, D. A. Rodríguez-Trejo, H. Tchikoué, A. I. Monterroso-Rivas, and J. Santillan-Pérez. Análisis espacial de peligro de incendios forestales en Puebla, México. *Interciencia*, 37(9):678–683, 2012.

[8] D. Cisneros-González, G. Pérez-Verdín, M. Pompa-García, D. A. Rodríguez-Trejo, and J. M. Zúñiga-Vásquez. Spatial modeling of forest fires in Mexico: an integration of two data sources. *Bosque*, 38(3):563–574, 2017.

[9] Conafor. Historical yearly series of wildfires 2010-2018 period. Extracted from: `https://datos.gob.mx/busca/dataset/incendios-forestales/resource/5720e224-3d0c-4eed-ac65-ea7aac7d72e8`, 2018. Date: 2019-08-04.

[10] N. Cressie. *Statistics for spatial data.* John Wiley & Sons, 1991.

[11] D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure.* Springer Science & Business Media, 2007.

[12] P. J. Diggle. *Statistical analysis of spatial and spatio-temporal point patterns.* CRC Press, 2013.

[13] E. Gabriel and P. J. Diggle. Second-order analysis of inhomogeneous spatio-temporal point process data. *Statistica Neerlandica*, 63(1):43–51, 2009.

[14] E. Gabriel, F. Rodriguez-Cortes, J. Coville, J. Mateu, and J. Chadoeuf. Mapping the intensity function of a non-stationary point process in unobserved areas. *Stochastic Environmental Research and Risk Assessment*, pages 1–17, 2022.

[15] E. Gabriel, B. S. Rowlingson, and P. J. Diggle. stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. *Journal of Statistical Software*, 53:1–29, 2013.

[16] J. Illian, A. Penttinen, H. Stoyan, and D. Stoyan. *Statistical analysis and modelling of spatial point patterns*. John Wiley & Sons, 2008.

[17] J. Møller and R. P. Waagepetersen. *Statistical inference and simulation for spatial point processes*. Chapman and Hall/CRC, 2003.

[18] L. R. Munive-Hernández. Predicción espacial de incendios forestales usando aprendizaje máquina. 2021.

[19] M. Pompa-García, J. Camarero J., D. A. Rodríguez-Trejo, and D. J. Vega-Nieva. Drought and spatiotemporal variability of forest fires across Mexico. *Chinese Geographical Science*, 28(1):25–37, 2018.

[20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022.

[21] B. D. Ripley. Modelling spatial patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(2):172–192, 1977.

[22] D. A. Rodríguez-Trejo. Incendios de vegetación: su ecología, manejo e historia vol. 1. Technical report, 2014.

[23] D. A. Rodríguez-Trejo. Incendios de vegetación: su ecología, manejo e historia vol. 2. Technical report, 2015.

[24] B. M. Taylor, T. M. Davies, B. S. Rowlingson, and P. J. Diggle. lgcp: an r package for inference with spatial and spatio-temporal log-gaussian cox processes. *Journal of Statistical Software*, 52:1–40, 2013.

[25] A. Trapletti and K. Hornik. *tseries: Time Series Analysis and Computational Finance*, 2022. R package version 0.10-52.

[26] A. Villanueva-Morales. Modified pseudo-likelihood estimation for Markov random fields with Winsorized Poisson conditional distributions. 2008.

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

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

Requisitos para la colaboración en el Boletín

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

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

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

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

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

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

Sociedad Mexicana de Computación Científica y sus Aplicaciones

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

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