

BENEMÉRITA UNIVERSIDAD AUTÓNOMA
DE PUEBLA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

T E S I S

NOMBRE DE TESIS:

IDENTIFICACIÓN DE PARÁMETROS DEL MODELO SIR

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN MATEMÁTICAS APLICADAS

PRESENTA:

C. LAURA RAMÍREZ RAMÍREZ

DIRECTORES DE TESIS:

DR. JOSÉ JULIO CONDE MONES

DR. CARLOS ARTURO HERNÁNDEZ GRACIDAS

Puebla, Puebla, Enero 2024



Título: IDENTIFICACIÓN DE PARÁMETROS DEL
MODELO SIR

Estudiante: LAURA RAMÍREZ RAMÍREZ

COMITÉ

Dr. José Jacobo Oliveros Oliveros
Presidente

Dr. José Rubén Conde Sánchez
Secretario

MC. Felix Augusto Aquino Camacho
Vocal

Dr. José Julio Conde Mones
Director de Tesis

Dr. Carlos Arturo Hernández Gracidas
Codirector de Tesis

Dedicatoria ...

Dedico este trabajo de investigación con todo el corazón, amor y cariño :

- * *A Dios, por ser la base de sustento en cada día de mi vida.*

- * *A mis padres, Isaac Ramírez y Sabina Ramírez por ser mi Fortaleza y los guías de mi vida.*

- * *A mis hermanos y hermanas (Nahum, Paty, Isaac, Rocío, y José) por acompañarme en este proceso, con amor, comprensión y cariño.*

- * *A mi cuñada y sobrinos, por ser la alegría y darme motivación.*

- * *A mis amigos, familiares y conocidos, por ser parte de mi vida.*

Agradecimientos

Desde niña he planificado cada paso de mi vida, sin embargo no tomaba en cuenta las adversidades y aunque estas han demorado unos meses o quizá años mis metas, he logrado lo que me he propuesto.

Es por eso que en el presente trabajo quiero agradecerle principalmente a Dios, por convertiste en la razón de mi vida, darme salud, y vida, además de ser el inspirador y darme fuerzas para continuar en este proceso de obtener uno de los anhelos más deseados y permitirme llegar a este punto y poder compartir este logro con mi familia, **¡Dios, gracias por tanta bendición!**

A mis padres Isaac Ramírez Méndez y Sabina Ramírez Largo, es muy difícil y no creo tener las palabras adecuadas para expresarles mi total agradecimiento por todo lo que han hecho por mí durante toda mi vida; en carencia de tales palabras, sólo me queda decirles gracias. En primer lugar, gracias por darme la vida, por enseñarme cosas básicas del día a día como disfrutar de los pequeños detalles, por permitirme no tener miedo y, a la vez, tenerlo cuando se necesita, gracias al obligarme a hacer algo por mí bien y eso ha hecho de mí una buena persona y se ha encargado de impulsarme de todas las maneras posibles para alcanzar mis sueños. Gracias por darme una bonita familia y un hogar. Gracias por apoyarme en todos mis proyectos desde que inicié hasta el día de hoy, por pensar que puedo lograr todo lo que me proponga aunque yo misma no lo crea así. Gracias por trabajar sin cansancio para que a mis hermanas, hermanos y a mí no nos faltara nada. **¡Gracias por cuidar siempre de mí!**

A mis hermanos y hermanas: ¿Cómo hay personas que pueden vivir siendo hijos únicos? ¿Cómo pueden crecer sin hermanos? No sería capaz de imaginar mi vida sin ustedes. Quiero agradecerles por ser mis confidentes, por su paciencia y tolerancia, por su cariño y comprensión a pesar de las diferencias, por ayudarme en todo lo que he necesitado. Les agradezco por soportarme durante tantos años. Yo sé lo difícil que es lidiar conmigo en muchas ocasiones, en especial cuando estoy estresada. A pesar de ello, nunca me dejaron sola. El mejor regalo que Dios y la naturaleza pudo darme son ustedes, mis hermanos. **¡Gracias por apoyarme y**

estar siempre conmigo!

A mi cuñada y sobrinos por tanta alegría y motivación que me han brindado. Estoy muy feliz de que mi hermano haya conocido a una mujer tan guapa y trabajadora, que tanto le aporta a él y a toda la familia. **¡Los amo con todo el corazón!**

Agradezco a todos mis amigos y compañeros de vida que siempre han creído en mí y me han ayudado a crecer en mi formación personal y académica. Gracias por su grata compañía, por el apoyo que me han brindado y por todo lo que he aprendido de ellos y con ellos. Manifiesto mi aprecio aquellos amigos que me ayudaron a salvar un poco la carrera y un poco la vida. **¡Gracias por su granito de arena!**

De igual manera gracias a quienes de manera voluntaria e involuntaria han contribuido en mi formación.

Quiero de manera especial agradecer a mis directores de tesis, al *Dr. José Julio Conde Mones* y al *Dr. Carlos Arturo Hernández Gracidas*, por confiar en mí, para realizar este proyecto, por haber sido muy pacientes, por la inmensa ayuda que me han brindado y compartir sus conocimientos, haber sido esas personas que con sus directrices pudieron explicarme aquellos detalles para culminar mi tesis. De verdad Dr. José Julio Conde Mones y Dr. Carlos Arturo Hernández Gracidas, **muchas gracias por sus enseñanzas.**

Agradezco a mis sinodales, por el tiempo dedicado a la revisión de mi tesis.

Resumen

En este trabajo, se plantean y resuelven los problemas directo e inverso del modelo SIR. Este último, consiste en estimar los parámetros del modelo a partir de las mediciones. Por ejemplo, para el caso de COVID-19, las mediciones son los casos reportados de infectados por día. De hecho, el modelo SIR es de propósito general para el estudio de cualquier enfermedad infecciosa en una población que puede ser estudiada en tres estados o compartimentos a lo largo de un período de tiempo, por lo que puede aplicarse a la epidemia COVID-19. En este trabajo se presentan ejemplos sintéticos de los problemas directo e inverso del modelo SIR aplicado a COVID-19.

El problema inverso del modelo SIR es un problema inverso mal planteado en sentido de Hadamard, ya que puede tener más de una solución; y además, puede presentar inestabilidad numérica. Para abordar la inestabilidad numérica se utilizan técnicas de regularización, las cuales producen soluciones estables respecto a datos (mediciones) con o sin ruido. En este trabajo no se aborda el problema de la inestabilidad numérica. El objetivo general de este trabajo es resolver el problema inverso de identificación de parámetros del modelo SIR, considerando que se tienen mediciones exactas, proponiendo un método que consiste en determinar los parámetros del modelo SIR minimizando una función objetivo sujeta a restricciones.

Para abordar este problema inverso, nuestra hipótesis de investigación es que el software DIFFPAR determina mínimos locales con bastante precisión, el cual aplica un método de regularización local del algoritmo de Gauss-Newton para encontrar un mínimo local de la función objetivo propuesta en este trabajo y sujeta a restricciones. La factibilidad de DIFFPAR se mostró con ejemplos sintéticos para validar y analizar la convergencia del método, tomando diferentes puntos iniciales. Para tal fin, y como contribución de esta tesis, se desarrollaron programas computacionales en Matlab (incluidas en el Apéndice de la Tesis) para resolver tanto el problema directo como el problema inverso. Tales programas se pueden modificar y adecuarse para resolver los problemas directo e inverso de las diferentes variantes del modelo SIR, entre otros problemas de identificación de parámetros modelados por ecuaciones diferenciales ordinarias con condiciones iniciales y cuya dinámica se desarrolla a lo largo de un periodo de tiempo.

Índice general

Dedicatoria	I
Agradecimientos	II
Resumen	IV
Introducción	XI
1. Los modelos matemáticos en epidemiología	1
1.1. Uso de la modelación matemática en epidemiología	1
1.2. Epidemiología	2
1.2.1. Los primeros modelos matemáticos en epidemiología	3
1.2.2. Términos generales importantes	4
1.3. Modelos matemáticos epidemiológicos	5
1.3.1. Formas de transmisión de enfermedades	6
1.3.2. Modelos epidemiológicos clásicos	7
1.4. El modelo SIR y sus parámetros	7
1.5. Problemas inversos	11
2. Planteamiento del problema y resultados numéricos	14
2.1. Problema directo e inverso	14
2.2. Planteamiento del problema de identificación de parámetros	15
2.3. Resultados numéricos	16
2.3.1. Resultados numéricos del PD	16
2.3.2. Resultados numéricos del PI	18
2.4. Conclusión	32
A. Ecuaciones diferenciales ordinarias y métodos numéricos	35
A.1. Sistema de ecuaciones diferenciales	35
A.1.1. Sistemas de ecuaciones diferenciales ordinarias de primer orden	36
A.1.2. Ecuaciones diferenciales lineales de primer orden	37
A.2. Métodos numéricos	39
A.2.1. Solución numérica del PVI	39

<i>ÍNDICE GENERAL</i>	VI
A.2.2. Clasificación de métodos numéricos para las EDO	41
A.2.3. Métodos de Runge Kutta	42
A.2.4. Método de Dormand-Prince 5(4)	44
A.2.5. Métodos implícitos BDFs de paso variable	46
A.2.6. Solución del PVI por el método implícito de Euler	47
B. Códigos de los programas elaborados	49
B.1. Programas para los problemas directos del modelo SIR	49
B.1.1. Programa 1: Método de Runge Kutta de orden 4, usando ode45 de MATLAB.	49
B.1.2. Programa 2: Método de Euler Implícito para graficar la fun- ción objetivo g	52
B.2. Programas para el problema inverso (PI) del modelo SIR	62
B.3. Instrucciones para resolver el PI del modelo SIR aplicando DIFFPAR	63
Bibliografía	69

Índice de figuras

1.1.	Problema Directo. Fuente: Martínez, Víctor. Problemas inversos: los casi olvidados de la Matemática Educativa, 2011 [48].	12
1.2.	Problema inverso, especificación. Fuente: Martínez, Víctor. Problemas inversos: los casi olvidados de la Matemática Educativa, 2011 [48].	12
1.3.	Problema inverso, causalidad. Fuente: Martínez, Víctor. Problemas inversos: los casi olvidados de la Matemática Educativa, 2011 [48].	13
2.1.	Solución del PD del modelo SIR para $(\beta, \gamma) = (0.6, 0.1) \in \Omega$, con $I_0 = 1, S_0 = 999, R_0 = 0$, y $N = 1000$	17
2.2.	Solución del PD del modelo SIR para $(\beta, \gamma) = (0.6, 0.3) \in \Omega$, con $I_0 = 1, S_0 = 999, R_0 = 0$ y $N = 1000$	18
2.3.	Solución del PD del modelo SIR para $(\beta, \gamma) = (0.667, 0.167) \in \Omega$, con $I_0 = 1, S_0 = 999, R_0 = 0$, y $N = 1000$, aplicando el Método Implícito de Euler (MIE) en el intervalo de tiempo $[0, 30]$	19
2.4.	Gráfica de la función objetivo $g(\beta, \gamma)$, para los datos exactos de los parámetros $(\beta^*, \gamma^*) = (0.667, 0.167)$ y los valores iniciales S_0, I_0 y R_0 dados del ejemplo 2.3.3, en la región factible $\Omega = [0.1, 1.5] \times [0.05, 1]$	20
2.5.	Gráfica de las soluciones ajustadas S^{ω_n} (curva en color azul), I^{ω_n} (curva en color rojo), y R^{ω_n} (curva en color verde) del modelo SIR contra los datos de casos por día (puntos marcados con asteriscos para cada compartimento $S^{\omega^*}, I^{\omega^*}$ y R^{ω^*}), correspondientes al par de estimaciones $\omega_n = (\beta_n, \gamma_n)$ del par de parámetros $\omega^* = (\beta^*, \gamma^*)$ del PI, en la iteración $n = 52$	27
2.6.	Gráfica de la función objetivo $g(\beta, \gamma)$, para los datos exactos de los parámetros $(\beta^*, \gamma^* = (0.6, 0.1)$ y los valores iniciales S_0, I_0 y R_0 dados en el ejemplo 3, en la región factible $\Omega = [0.1, 1] \times [0.05, 1]$	28
2.7.	Gráfica de las soluciones ajustadas S^{ω_n} (curva en color azul), I^{ω_n} (curva en color rojo), y R^{ω_n} (curva en color verde), del modelo SIR contra los datos de casos por día (puntos marcados con asteriscos para cada compartimento $S^{\omega^*}, I^{\omega^*}$ y R^{ω^*}), correspondientes al par de estimaciones $\omega_n = (\beta_n, \gamma_n)$ del par de parámetros $\omega^* = (\beta^*, \gamma^*)$ del PI, en la iteración $n = 90$	29

2.8. Gráfica de la función objetivo $g(\beta, \gamma)$, para los datos exactos de los parámetros $(\beta^*, \gamma^* = (0.6, 0.3))$ y los valores iniciales S_0, I_0 y R_0 dados en el ejemplo 3, en la región factible $\Omega = [0.1, 1] \times [0.05, 1]$	30
B.1. Ventana principal de DIFFPAR	63
B.2. Subventana MODEL	64
B.3. Subventana DATA	65
B.4. Subventana para generar datos artificiales	65
B.5. Ventana principal de DIFFPAR ,para trbajar en la optimización	66
B.6. Subventana Data/Weights	66
B.7. Subventana TUNING	67
B.8. Subventana OPTIMIZATION	67
B.9. Subventana de ANALISYS	68

Índice de tablas

2.1.	Resultados numéricos (en <i>formato short e</i> de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.	22
2.2.	Resultados numéricos (en <i>formato short e</i> de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.	23
2.3.	Resultados numéricos (en <i>formato short e</i> de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.	24
2.4.	Resultados numéricos (en <i>formato short e</i> de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.	25
2.5.	Resultados numéricos (en <i>formato short e</i> de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.	26
2.6.	Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.3, 0.05)$, con una tolerancia de $1e - 3$ y un máximo de 90 iteraciones para el algoritmo de Gauss Newton implementado en DIFFPAR. Parámetros exactos $(\beta^*, \gamma^*) = (0.6, 0.1)$ del PI.	28
2.7.	Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.8, 0.3)$, con una tolerancia de $1e - 3$ y un máximo de 90 iteraciones para el algoritmo de Gauss Newton implementado en DIFFPAR. Parámetros exactos $(\beta^*, \gamma^*) = (0.6, 0.1)$ del PI.	28

2.8.	Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.85, 0.32)$, con una tolerancia de $1e - 3$ y un máximo de 90 iteraciones para el algoritmo de Gauss Newton implementado en DIFFPAR. Parámetros exactos $(\beta^*, \gamma^*) = (0.6, 0.3)$ del PI.	30
2.9.	Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.7, 0.32)$, con una tolerancia de $1e - 3$ y un máximo de 90 iteraciones para el algoritmo de Gauss Newton implementado en DIFFPAR. Parámetros exactos $(\beta^*, \gamma^*) = (0.6, 0.3)$ del PI.	30
A.1.	Coeficientes del método Dormand-Prince 5(4) (DOPRI5). Fuente: Tomado de la Tabla 5.2, página 178, Hairer, E., Nørsett, S. P., y Wanner, G., 1993, [33].	46

Introducción

En la actualidad, hay un gran interés por el estudio y aplicación de técnicas o métodos de optimización para resolver problemas inversos de identificación de características desconocidas de un medio, sistema o modelo matemático a partir del registro de las manifestaciones externas o datos medidos en la frontera exterior del medio de estudio. Entre los problemas inversos en donde se aplican los métodos de optimización tenemos: el problema inverso electroencefalográfico que consiste en la identificación de fuentes bioeléctricas superficiales y/o volumétricas en el cerebro a partir del electroencefalograma (EEG) medido sobre el cuero cabelludo [18, 19, 53], el problema inverso de conducción de calor hacia atrás en el tiempo que consiste identificar la distribución de calor y/o fuente de calor en un medio u otras características del medio a partir de datos medidos del calor o flujo de calor en un instante de tiempo final [40, 47], el problema de la tomografía de capacitancia eléctrica que consiste en determinar la permitividad de un flujo multifásico que pasa por un tubo a partir de mediciones de capacitancia en la frontera de dicho tubo, el cual puede abordarse como un problema inverso de conductividades (identificación de coeficientes o parámetros de la ecuación de Laplace) a partir de los datos de Cauchy [2, 8], el problema inverso gravimétrico en geoprospección y geodesia que consiste en determinar la distribución de masas en el interior de la Tierra, donde la información proporcionada es utilizado para estudiar tanto la estructura global de la corteza terrestre como para la prospección en áreas de extensión más reducida, y como apoyo a otras técnicas como la sísmica [28, 77]. Cabe mencionar que estos tipos de problemas inversos aplican técnicas y métodos numéricos de visualización no invasivas y que la mayoría de estas técnicas son económicas. Otra aplicación es la correspondiente a la estimación de parámetros del modelo SIR a partir de los datos de casos reportados de una población de infectados [69], el cual es uno de los casos abordados en esta tesis.

Recientemente, el modelo epidemiológico SIR y sus variantes han tenido una gran cantidad de aplicaciones para predecir la dinámica de una enfermedad infecciosa como la epidemia de COVID-19 [1, 12, 16, 29, 41, 56, 57, 60, 62, 73, 74, 76, 79], que es una enfermedad causada por un coronavirus denominado SARS-CoV-2. La Organización Mundial de la Salud (OMS) tuvo conocimiento por primera vez de este nuevo virus el 31 de diciembre de 2019, tras la notificación de un conglomerado de casos de la llamada neumonía viral, ocurridos en Wuhan (República Popular

China). El modelo SIR y sus variantes pueden proporcionar herramientas útiles y económicas para probar hipótesis sobre la forma óptima de controlar la propagación de la infección, en particular a través de la vacunación [9]. Daniel Bernoulli fue el primer matemático en proponer un modelo para describir una enfermedad infecciosa [36]; a saber, modeló la propagación de la viruela en 1760 [10]. Años después, W. O. Kermack y A. G. McKendrick, en 1927, propusieron el modelo determinista compartimental que predice el comportamiento de los brotes epidémicos, llamado modelo susceptibles, infectados y recuperados (SIR) [42]. En este trabajo, se estudia el problema inverso de identificación de parámetros del modelo SIR a partir de un conjunto de mediciones (variaciones de las poblaciones o compartimentos que componen el modelo). Cabe mencionar que las estimaciones de estos parámetros, de acuerdo a estudios previos realizados, no siempre se encuentran en la literatura médica. Estas estimaciones pueden ser consultadas por los organismos privados, públicos, gubernamentales y de salud, para determinar la tasa de reproducción básica \mathfrak{R}_0 , la cual permite explorar el impacto en la dinámica de la transmisión de enfermedades infecciosas. Además, este tipo de modelos compartimentales puede coadyuvar a identificar, cuantificar, evaluar e implementar políticas de intervención dirigidas a reducir el efecto de una epidemia a través de la reducción del impacto de los mecanismos de la transmisión de una enfermedad contagiosa (considerados en el modelo matemático).

Otros problemas prácticos, donde es importante la identificación de parámetros de ecuaciones diferenciales ordinarias con condiciones iniciales, es el crecimiento y decaimiento asociados a la dinámica poblacional que son estudiados por el modelo de Malthus y el modelo de Verhulst. El modelo de Malthus es uno de los primeros intentos para modelar el crecimiento de la población humana por medio de las matemáticas y que llevó a cabo el economista inglés Thomas Malthus en 1798 [78]. El modelo de Verhulst, o modelo logístico, introducido por Verhulst (1838) y posteriormente estudiado por R. Pearl y L. J. Reed (1920), está asociado al estudio de la dinámica de una población bajo alguna dependencia en las tasas de natalidad y mortalidad, pero se considera fuertemente la capacidad de carga del sistema (espacio natural delimitado o acondicionado) [54].

Hipótesis y objetivos de la Tesis

Hipótesis

La hipótesis de la Tesis es que *el software DIFFPAR determina los coeficientes o parámetros del modelo SIR con bastante precisión de manera local.*

Objetivo general

Resolver el problema inverso de identificación de parámetros del modelo SIR.

Objetivos específicos

1. Plantear el problema inverso del modelo SIR como un problema de optimización, el cual consiste en determinar los coeficientes del modelo minimizando una función objetivo sujeta a restricciones como se muestra en [23] o en [69].
2. Elaborar programas en MATLAB para resolver el problema directo para diferentes valores de los parámetros α y β del modelo SIR y visualizar los mínimos de la función objetivo g del problema de minimización (2.2) (propuesta en [69]) en una región factible de parámetros Ω . De hecho, este es un método gráfico para resolver el problema inverso de identificación de los parámetros del modelo SIR.
3. Elaborar ejemplos sintéticos para validar los programas computacionales realizados para resolver el problema directo, para diferentes valores de los parámetros α y β del modelo SIR.
4. A partir de la gráfica de la función objetivo g , estimar un punto inicial $\theta_0 = (\alpha_0, \beta_0, S_0, I_0, R_0)$ cercano al punto exacto $\theta^* = (\alpha^*, \beta^*, S_0, I_0, R_0)$, para resolver el problema inverso aplicando DIFFPAR, donde α^* , β^* son los parámetros desconocidos a estimar del problema inverso y $S_0 = S(0)$, $I_0 = I(0)$, $R_0 = R(0)$ son las condiciones iniciales del modelo SIR, en el tiempo inicial $t = 0$.
5. Elaborar ejemplos sintéticos para validar y analizar la convergencia del método de regularización local del algoritmo de Gauss-Newton que usa DIFFPAR para minimizar la función objetivo propuesta en [23], para diferentes puntos iniciales $\theta_0 = (\alpha_0, \beta_0, S_0, I_0, R_0)$ cercanos al punto exacto θ^* determinados en el punto anterior, así como también, para otros puntos iniciales θ_0 no cercanos al punto exacto θ^* .

Contribuciones de la Tesis

1. Se planteó el problema inverso del modelo SIR como un problema de optimización, en donde la función objetivo es el cuadrado de una norma ponderada de residuos usando el software DIFFPAR de Edsberg and Wedin (1995) [23]. Este mismo planteamiento de resolver este tipo de problemas inversos puede aplicarse a variantes del modelo SIR y en otros contextos.
2. Se desarrollaron programas computacionales para la solución del modelo SIR (problema directo), los cuales se pueden generalizar para las diferentes variantes del modelo SIR.

3. Se programó en MATLAB el método implícito de Euler para poder graficar la función objetivo g del problema de optimización (2.2) propuesta en [69].
4. Se describió con mayor detalle el método explícito de Dormand-Prince 5(4) [21] que está implementado en la función `ode45` de las nuevas versiones de MATLAB [67], el cual es conocido como el método DOPRI5 [33] y cuyo algoritmo no está descrito en MATLAB.
5. Se escribió un pequeño manual para resolver el problema inverso de identificación de parámetros del modelo SIR usando DIFFPAR.
6. Por último, parte de los resultados numéricos presentados en esta Tesis están aceptados para su publicación en las memorias del X Coloquio de Modelación Matemática en Biología, Ciencias Médicas e Ingeniería.

Finalmente, el contenido de la Tesis es el siguiente: en el Capítulo 1, se presenta un resumen sobre las formas de transmisión de enfermedades infecciosas, el modelo SIR y sus parámetros, así como algunas variantes del modelo SIR. También, se introduce la noción de un problema directo y su correspondiente problema inverso, y se da la definición de qué es un problema bien planteado en sentido de Hadamard. En el Capítulo 2, se presenta el planteamiento de los problemas directo e inverso del modelo SIR y los resultados numéricos obtenidos, así como también, se muestran las conclusiones de la tesis, y por último, se incluye un Apéndice sobre los métodos numéricos existentes para resolver las ecuaciones diferenciales ordinarias y los códigos de los programas generados para resolver los problemas directo e inverso del modelos SIR, así como una pequeña guía del uso de DIFFPAR para resolver el problema inverso del modelo SIR.

Capítulo 1

Los modelos matemáticos en epidemiología

Los modelos matemáticos coadyuvan al estudio de enfermedades infecciosas. Las enfermedades infecciosas constituyen una amenaza en la sociedad, es por esto que han ganado importancia entre la comunidad científica y la de salud el uso de modelos y métodos matemáticos para estudiar la dinámica de transmisión y control de enfermedades patógenas con el propósito de idear programas efectivos que interpreten patrones epidemiológicos. La interacción entre profesionales de matemáticas aplicadas y de salud conllevaría a realizar investigaciones y modelos que se aproximan de manera más pertinente a los factores que influyen en la dinámica de las enfermedades infecciosas. En este capítulo se describen algunos modelos epidemiológicos clásicos para clasificar, en función de su estado de salud a una población respecto de una enfermedad, en susceptibles, infectados y recuperados, entre otras clasificaciones o compartimentos de la población [52].

1.1. Uso de la modelación matemática en epidemiología

Al paso de los años, las matemáticas han sido aplicadas a una gran variedad de campos como la epidemiología. En este sentido, los modelos matemáticos son una herramienta que permite disponer del conocimiento para evaluar el comportamiento y realizar pronósticos para la toma de decisiones en el seguimiento y control de epidemias [66]. Kermack y McKendrick publicaron en 1927 un modelo para predecir el tamaño final de una epidemia, la forma en la que se propaga dicha epidemia y además, plantearon lo que se conoce como teorema del umbral, donde se asume que la población es homogénea, es decir, los individuos tienen la misma probabilidad de encontrarse e infectarse [75]. Esta afirmación no es cierta, ya que depende del modo de transmisión, los agentes infecciosos o de la población afectada, sin embargo este modelo es un primer acercamiento para describir la dinámica de una epidemia, el

cual sigue siendo empleado para describir la dinámica del COVID-19. La modelación matemática desempeña, cada vez más, un papel protagónico en la gestión de brotes y epidemias, pero también en la información de las decisiones de salud pública.

Las enfermedades infecciosas progresan dentro de las poblaciones tanto por el comportamiento del agente infeccioso como por la propia población, y estas pueden estudiarse por medio de modelos matemáticos epidemiológicos, los cuales se basan en un conjunto de suposiciones y estadísticas que se utilizan para establecer un conjunto de parámetros, que informan de cuán efectiva será la intervención. Esta información se puede usar para predecir qué intervenciones implementar o evitar, así como patrones de crecimiento y expansión. Por tanto, la modelación de epidemias debe de ser utilizada de manera diferente para cada tipo de contexto, pues se deben de tener presentes aspectos como la difusión de la enfermedad, si se es inmune o no a la enfermedad, y el medio de transmisión de la enfermedad, es decir, si el contagio es de persona a persona o por vectores. Aunque los modelos matemáticos son la única herramienta práctica que nos permite contestar preguntas acerca de que medidas de control son las más adecuadas y efectivas, no siempre es posible introducir en estos modelos todos los factores que intervienen en el proceso de propagación de una enfermedad. Al usar modelación matemática estamos traduciendo los problemas que aparecen en un determinado ámbito científico o tecnológico al lenguaje matemático, de tal forma que el análisis teórico y numérico que se hace de los mismos proporcione información para entender mejor los mecanismos que rigen el fenómeno en cuestión. Se trata pues de una herramienta de investigación que puede ser considerada como un complemento a la teoría y experimentación en la investigación científica. De hecho, la modelación matemática es la opción óptima cuando se trata de obtener conocimientos e información de experimentos [61]. Sin embargo, debemos tener en cuenta que a pesar de que los modelos epidemiológicos son beneficiosos, también tienen algunas limitaciones; por ejemplo, un modelo epidemiológico es una simplificación extrema de la realidad, no es la realidad, y aun así, con las limitaciones de estos modelos pueden ayudar a entender de manera más simple o más clara la propagación de una epidemia; además, pueden dar información para la toma de decisiones y control de la epidemia [31].

1.2. Epidemiología

El término *epidemiología* proviene del griego, *epi* que significa arriba, *demos* pueblo y *logos* estudio o tratado. El énfasis inicial de la epidemiología estuvo en el estudio de las enfermedades infecciosas consideradas epidemias, aquellas las cuales constituyen un serio problema de salud de las poblaciones. La epidemiología se encarga del estudio de los factores que causan o están asociados con la enfermedad, así como también del estudio de la incidencia, la distribución y el control de los factores determinantes en enfermedades frecuentes en humanos, con el objetivo de determi-

nar las formas de prevención y control de estas enfermedades. Una vez identificados los factores determinantes de la generación y desarrollo de la enfermedad o de la conservación del estado de salud, la epidemiología puede evaluar las necesidades de atención y recursos y medir la eficacia de las medidas implementadas. Antiguamente, el interés en epidemias ha sido una prioridad, ya que millones de personas morían por estas enfermedades infecciosas. Actualmente, el estudio de las enfermedades infecciosas sigue cobrando gran interés. La descripción cuantitativa de las epidemias se realiza mediante modelos matemáticos. Asimismo, estos permiten comparar y evaluar el impacto de las distintas medidas de control. El carácter interdisciplinario de la epidemiología está ligado a su intención de construir un conocimiento de todo el entorno o contexto en el que se desenvuelven los problemas de salud para poder comprenderlos.

1.2.1. Los primeros modelos matemáticos en epidemiología

En 1760, el matemático Daniel Bernoulli [10] presentó ante la academia real de ciencias de París un trabajo en el cual por primera vez usó un modelo matemático para estudiar la difusión de una enfermedad infecciosa en la población. El origen de este trabajo fue el siguiente: durante la primera mitad del siglo XVIII se introdujo en Europa la técnica de variolación o inoculación contra la viruela. La técnica de variolación usada desde tiempos remotos en India y China, consisten en inocular, con el material tomado de una pústula de viruela de un caso activo a un individuo susceptible de contraer la enfermedad, con la intención de producir en él un ataque ligero. Una vez que la persona se recuperaba de la enfermedad, este adquiría inmunidad permanente. Bernoulli, quien además de matemático era médico, se interesó en el problema, y para evaluar la efectividad de la técnica de variolación en esa época formuló y resolvió su famosa ecuación diferencial y evaluó los resultados en términos de las medidas de control involucradas. Durante la segunda mitad del siglo XIX, el desarrollo espectacular de la Bacteriología, debido a los trabajos de Pasteur (1822-1895), Koch (1843-1910) [51] y otros, fue quizás uno de los factores decisivos en el comienzo real de una nueva ciencia: la epidemiología teórica. La primera contribución importante a la epidemiología teórica se debió a los trabajos de Hamer en 1906 [35], quién postuló que el curso de una epidemia depende de la tasa de contactos entre individuos susceptibles e infecciosos. Esta noción se convirtió en uno de los conceptos más importantes en epidemiología matemática: la ley de acción de masas, la cual dice que la tasa a la cual una enfermedad se propaga es proporcional al número de individuos susceptibles por el número de individuos infecciosos. Su estudio fue bastante relevante debido a que fue el primero en considerar que la incidencia de una enfermedad está relacionada con las densidades de población susceptible e infecciosas. En 1911, Ronald Ross [64] formuló el principio de acción de masas para un modelo continuo, en su trabajo pionero sobre la dinámica de la malaria. Kermack y McKendrick [42], establecieron el celebrado Teorema del umbral, el cual postula que la introducción de un individuo infeccioso a una comunidad no dará lugar a un

brote epidémico, a menos que la densidad de los susceptibles en la población sobrepase cierta cantidad umbral. Los trabajos sobre modelos epidémicos de Kermack y McKendrick han tenido una gran influencia para el desarrollo posterior de modelos matemáticos. Los modelos matemáticos pueden ayudarnos a predecir el curso de una epidemia dentro de una población; pueden también ser una herramienta útil para detectar los umbrales de población más allá de los cuales existe el riesgo de una epidemia. En el contexto de una enfermedad endémica, los modelos matemáticos dan información acerca de cómo los niveles de endemidad están relacionados con factores que pueden ser controlados por la intervención de las autoridades sanitarias, como la que tuvo suceso en 1927. Longini, Ackerman y Elveback (1978) [45] usaron un modelo matemático para decidir que grupos de edad deben vacunarse primero para minimizar el costo o riesgo de muerte en una epidemia de influenza. Hethcote y Yorke (1984) [37] compararon los procedimientos para el control de la gonorrea en la población estadounidense. Para mayor información sobre los primeros modelos matemáticos epidemiológicos ver páginas 57-58 de [25].

1.2.2. Términos generales importantes

A continuación, presentamos algunos conceptos importantes necesarios para las siguientes secciones.

Definición 1.2.1. *Epidemiología* es una disciplina científica que estudia la frecuencia y distribución de los eventos de salud y de sus determinantes en las poblaciones humanas, y la aplicación de este estudio en la prevención y control de los problemas de salud [17].

La epidemiología analiza la enfermedad para desarrollar planes de prevención y de lucha, tales como la vacunación o la cuarentena. Además, intenta indicar cuál será el número total o el número máximo de infectados en un determinado momento.

Definición 1.2.2. Una *epidemia* es el aumento inusual del número de casos de una enfermedad determinada en una población específica, en un período determinado.

Definición 1.2.3. Una *pandemia* es la afección de una enfermedad infecciosa que se ha extendido por un área geográficamente extensa y que, generalmente, afecta a un gran número de personas.

Definición 1.2.4. Consideramos que es una *enfermedad endémica* si persiste todo el tiempo en una zona geográfica.

Definición 1.2.5. La *prevalencia* es el número de casos (nuevos y existentes) de una enfermedad en una población y lugar en un período de tiempo dado.

Definición 1.2.6. La *incidencia* es el número de casos nuevos de una enfermedad en una población en un período determinado.

Definición 1.2.7. La *inmunidad* es el estado de resistencia general que posee una persona, asociado con la presencia de anticuerpos o células que poseen acción específica contra el microorganismo causante de una enfermedad infecciosa o contra su toxina.

Definición 1.2.8. Las *enfermedades infecciosas emergentes* son un tipo de enfermedades infecciosas que surgen en lugares y momentos específicos y se convierten, o amenazan con convertirse, en nuevas epidemias.

Definición 1.2.9. Las *enfermedades reemergentes* se refieren al resurgimiento de enfermedades que ya habían sido aparentemente erradicadas o su incidencia fue disminuida.

Definición 1.2.10. El *período de latencia* o de *exposición* es el tiempo que transcurre desde la exposición al agente hasta el momento en que la persona puede transmitir la enfermedad.

Definición 1.2.11. El *período de incubación* es el lapso que transcurre desde la exposición inicial a un agente infeccioso hasta la presentación del primer signo o síntoma de la enfermedad que ese agente produce.

Definición 1.2.12. El *período infeccioso* corresponde al intervalo en el que la persona puede transmitir la enfermedad. Este período puede preceder a los síntomas y puede durar más que los síntomas.

Definición 1.2.13. Un *umbral epidemiológico* es el número mínimo de personas infectadas durante el desarrollo de una enfermedad el cual determinará si habrá o no epidemia.

Para mayor información, ver [17].

1.3. Modelos matemáticos epidemiológicos

Los modelos matemáticos son muy importantes para el estudio de enfermedades, pues en la mayor parte de los problemas de enfermedades infecciosas no es factible la experimentación, ya que puede ser muy costoso, peligroso, y/o incluso imposible. Un modelo matemático es una representación matemática (normalmente mediante una función, una ecuación, o un sistema de ecuaciones diferenciales ordinarias o parciales con condiciones iniciales o de contorno) de un fenómeno del mundo real, como el tamaño de la población, la esperanza de vida al nacer o la propagación de enfermedades. Dado un problema del mundo real, la primera tarea es construir un modelo matemático. Para ello, se identifican y denominan variables independientes y dependientes, y se formulan hipótesis para simplificar el fenómeno, de modo que pueda ser apreciado matemáticamente.

Existen dos grandes tipos de modelos matemáticos epidemiológicos: el determinístico y el estocástico. En los *modelos determinísticos*, también conocidos como compartimentales, se pueden controlar los factores que intervienen en el estudio del proceso o fenómeno y por tanto se pueden predecir con exactitud sus resultados. Cada compartimento viene dado por el estado en el que se encuentran los individuos. Los modelos compartimentales pueden ser estudiados en poblaciones grandes y facilitan el estudio analítico de la epidemia. El nombre de estos modelos se basa en los patrones de flujo entre los distintos estados por los que pasan los individuos. En los *modelos estocásticos* no es posible controlar los factores que intervienen en el estudio del fenómeno y, en consecuencia, no producen simples resultados únicos. Cada uno de los resultados posibles se genera con una función que les adjudica una probabilidad.

Además, en un modelo compartimental lo importante es la variación de los estados a lo largo del tiempo mediante contagio entre los individuos de una población de tamaño N que permanece constante. Por ejemplo, para un modelo de cuatro estados o compartimentos, en cada momento del tiempo t los individuos pueden encontrarse en uno de los siguientes posibles compartimentos:

- a)* Susceptibles (S). Individuos sanos que pueden contraer la enfermedad.
- b)* Expuestos (E). Individuos infectados que no pueden contagiar la enfermedad.
- c)* Infectados (I). Individuos infectados que pueden contagiar a otros.
- d)* Recuperados (R). Individuos resistentes a la enfermedad, normalmente la han superado o han sido vacunados.

1.3.1. Formas de transmisión de enfermedades

En algunas enfermedades hay varios factores que impiden que se estudien de la misma manera, uno de estos es la forma de transmisión. Por lo tanto, el conocimiento del modo de transmisión y el estado del patógeno es esencial para la aplicación de técnicas de control de enfermedades [17]. Existen dos formas de transmisión: la directa e indirecta. La *transmisión directa* se da de persona a persona y puede ocurrir por la propagación de gotitas o gotículas respiratorias, y por contacto directo, físico y sexual. La *transmisión indirecta* se da por medio de vehículos de transmisión o fómites, por intermedio de un vector y por aire. La transmisión también depende de las características de la población, como el número de habitantes, si hay inmigración, emigración, muertes y nacimientos, y de factores como la edad, el género, la comorbilidad, entre otros.

1.3.2. Modelos epidemiológicos clásicos

Los modelos compartimentales o de compartimentos se utilizan con frecuencia para describir el flujo de material en sistemas biológicos. Cada modelo contiene un número de compartimentos por los cuales circula el material con flujos de entrada y salida definidos. De estos modelos es que mediante una formulación matemática se logran reunir todos sus elementos. Dichos elementos, ya expresados en el ambiente matemático, pueden ser manipulados y obtener nuevas expresiones matemáticas. Un modelo epidemiológico de compartimentos puede definirse a partir de las clases y subclases de individuos en que se puede dividir una población afectada por una enfermedad; los flujos entre clases y subclases.

1.4. El modelo SIR y sus parámetros

Consideremos el **modelo SIR** más conocido en epidemiología, propuesto por W. O. Kermack y A. G. McKendrick en 1927 [42], que caracteriza la evolución de una epidemia que se propaga mediante contagio entre los individuos de una población de tamaño constante N . Es un modelo clásico compartimental determinístico, en donde la población total se subdivide en tres compartimentos: Susceptibles (S), Infectados (I) y Recuperados (R). El modelo cuenta con los siguientes dos parámetros:

1. *Tasa de infección o tasa de transmisión de la enfermedad β* : Nos indica el número de contactos suficientes para la transmisión de una persona por unidad de tiempo, es decir, el parámetro β es el producto del número de contactos medios por individuo y unidad de tiempo y la probabilidad de contagio en un solo contacto entre un individuo infectado y un individuo susceptible. En consecuencia, $\beta I(t)/N$ nos indica el número medio de contactos infectados por unidad de tiempo t de un individuo susceptible, mientras que $\beta S(t)I(t)/N$ es el número de casos nuevos por unidad de tiempo, donde N es el tamaño total de la población.
2. *Tasa de recuperación γ* : Nos indica la tasa por unidad de tiempo a la cual los individuos infectados se recuperan. Es decir, es la razón entre el número de recuperados y el número de infectados por unidad de tiempo. El período medio de latencia, la cual se define como el tiempo que transcurre desde la exposición al agente hasta el momento en que la persona puede transmitir la enfermedad, se calcula como la inversa de la tasa de recuperación, $L = 1/\gamma$. Es importante resaltar, que el valor de la tasa de infección o transmisión está relacionado con:
 - a) El número de contactos por unidad de tiempo.
 - b) La probabilidad de contagio.

Si bien es cierto que una parte importante de la tasa de infección β , depende de si la enfermedad es más o menos contagiosa, las medidas de prevención, como el confinamiento, el distanciamiento y la adopción de medidas de higiene y protección por parte de la población, tienden a reducir su valor.

Otro parámetro de gran interés en epidemiología es la *tasa de reproducción básica* o *número reproductivo básico* \mathcal{R}_0 que se utiliza para describir la intensidad de una enfermedad infecciosa. De forma específica, la tasa reproductiva básica es el número promedio de nuevas infecciones causadas por uno y solo un individuo infeccioso en una población totalmente susceptible. La tasa de reproducción básica, en una población donde todos son susceptibles, se calcula de la siguiente manera:

$$\mathcal{R}_0 = \frac{\beta}{\gamma}.$$

La tasa reproductiva básica nos da indicios de si la enfermedad se va a propagar o no, así como de la velocidad con que puede propagarse en una población de individuos determinada. Cuando $\mathcal{R}_0 > 1$ la enfermedad puede llegar a propagarse entre la población. Sin embargo, si $\mathcal{R}_0 < 1$ la enfermedad desaparece tras un cierto período de tiempo [46, 61]. Con los parámetros descritos anteriormente, el **modelo SIR** queda establecido por el siguiente sistema de ecuaciones diferenciales ordinarias no lineales de primer orden:

$$\begin{cases} \frac{\partial S}{\partial t} = -\beta S(t) \frac{I(t)}{N} \\ \frac{\partial I}{\partial t} = \beta S(t) \frac{I(t)}{N} - \gamma I(t), \\ \frac{\partial R}{\partial t} = \gamma I(t), \end{cases} \quad (1.1)$$

para $t \in [t_0, t_f]$, donde t_0 es el tiempo inicial y t_f es el tiempo final. Este modelo se basa, además de las mencionadas antes, en las siguientes hipótesis que pueden consultarse en [4, 14, 75]:

1. La transmisión se produce a través del contacto directo entre un individuo susceptible y un individuo infectado (transmisión horizontal), a diferencia de la transmisión vertical que se produce de madre a hijo, antes, durante o después del nacimiento.
 2. La población es homogénea, es decir, los individuos tienen la misma probabilidad de encontrarse e infectarse, y la enfermedad se transmite según la ley de acción de masas. Así, la cantidad de contactos depende solo del número de individuos en cada compartimento. Siendo la tasa de incidencia (nuevos infectados por unidad de tiempo) $\beta I(t)S(t)$ (con β siendo el coeficiente de transmisión).
 3. La tasa de transferencia de un compartimento a otro es proporcional al número de individuos del compartimento de origen. Así, la tasa de transferencia de enfermos a recuperados, es decir, la tasa de recuperación, será $\gamma I(t)$.
-

4. Las personas recuperadas no pueden reinfectarse, entonces la tasa de transferencia de recuperados a susceptibles es nula.
5. La población se considera constante e igual a N , es decir, no hay llegada de nuevos susceptibles a la población y se consideran las mismas tasas de nacimiento y muerte durante el proceso de propagación de la enfermedad, es decir, se estudia la epidemia en tiempos cortos. También, se puede considerar que no hay muertes por causas no relacionadas con la enfermedad. Los individuos que murieron a causa de la enfermedad son contabilizados como parte de la población.
6. El período de latencia, desde el momento de la exposición hasta aquel en que el individuo comienza a ser infeccioso, es lo suficientemente pequeño como para no tomarlo en cuenta, es decir, todo individuo infectado es infeccioso, no hay período de latencia. Por lo que no hay compartimento de expuestos.

A continuación, presentamos algunas de las variantes del modelo SIR más conocidas en la epidemiología (ver páginas 15-20 [28]):

1. **MODELO SI.** El modelo SI es el más simple posible. La población está formada sólo por susceptibles (S) e infectados (I), y si se contagia un individuo, la enfermedad es permanente (no hay recuperación). Consiste en un sistema de dos ecuaciones diferenciales en el que aparecen el número $S(t)$ de personas susceptibles y el número $I(t)$ de personas infectadas como variables dependientes del tiempo t . Este modelo está dado por el siguiente sistema de ecuaciones diferenciales ordinarias

$$\begin{aligned}\frac{dS}{dt} &= -\beta S(t)I(t), \\ \frac{dI}{dt} &= \beta S(t)I(t),\end{aligned}$$

con condiciones iniciales $S(0) = I_0 > 0$ e $I(0) = I_0 > 0$. Por lo tanto, si $\beta > 0$ es la tasa de contagios entonces $\beta S(t)I(t)$ es la cantidad de susceptibles que se convierten en infectados por unidad de tiempo.

2. **MODELO SIS.** Se consideran nacimientos a lo largo del tiempo, existiendo así una renovación de individuos susceptibles a la enfermedad. Los individuos mueren por la enfermedad o de forma natural. Sus ecuaciones diferenciales son

$$\begin{aligned}\frac{dS}{dt} &= -\beta S(t)I(t) + \mu(N - S(t)) + \gamma I(t), \\ \frac{dI}{dt} &= \beta S(t)I(t) - \gamma I(t) - \mu I(t),\end{aligned}$$

donde el nuevo parámetro μ es la tasa promedio de defunciones (probabilidad de que un individuo infectado muera debido a la enfermedad).

3. **MODELO SIRS.** Este es un modelo que solo es una extensión del modelo SIR, donde los individuos recuperados pierden la inmunidad y vuelven a ser susceptibles. Este modelo está dado por las siguientes ecuaciones diferenciales

$$\begin{aligned}\frac{dS}{dt} &= -\beta S(t)I(t) + fR(t), \\ \frac{dI}{dt} &= \beta S(t)I(t) - \gamma I(t) - \mu I(t), \\ \frac{dR}{dt} &= \gamma I(t) - \mu R(t) - fR(t),\end{aligned}$$

donde el nuevo parámetro f es la tasa promedio de pérdida de inmunidad en individuos recuperados.

4. **MODELO SEIR.** Al modelo SIR se le puede agregar un nuevo compartimento, una nueva clase de individuos que portan la enfermedad pero que, al hallarse en su período de incubación, no muestran síntomas y aún no pueden infectar a otros. A dichos individuos se los denomina expuestos, y el número de ellos en el tiempo t se denota mediante $E(t)$. En el caso de que un individuo no presente síntomas, pero sí pueda contagiar a otros lo contabilizamos en $I(t)$, no en $E(t)$. Este modelo se denomina SEIR, y en él hay un nuevo parámetro σ , de tal forma que $1/\sigma$ es el tiempo promedio de incubación. En el modelo SEIR, el flujo entre los distintos grupos (que de nuevo deben cumplir $S(t) + E(t) + I(t) + R(t) = N$) se puede representar mediante el siguiente sistema de ecuaciones diferenciales

$$\begin{aligned}\frac{dS}{dt} &= B - \beta S(t)I(t) - \mu S(t), \\ \frac{dE}{dt} &= \beta S(t)I(t) - (\epsilon + \mu)E(t), \\ \frac{dI}{dt} &= \epsilon E(t) - (\gamma + \mu)I(t), \\ \frac{dR}{dt} &= \gamma I(t) - \mu R(t),\end{aligned}$$

donde los nuevos parámetros son B y ϵ . Aquí B es la tasa promedio de nacimientos y $\frac{1}{\epsilon}$ es el tiempo promedio de incubación.

5. **MODELO MSEIR.** El modelo compartimental más general está compuesto por cinco compartimentos, etiquetados con las letras M , S , E , I y R . La letra M representa a los individuos con inmunidad pasiva. Esta inmunidad es temporal para algunas enfermedades y adquirida por los recién nacidos gracias a los anticuerpos que la madre les transfiere. Con la desaparición de estos anticuerpos, los individuos pasan a la clase de susceptibles (S). El compartimento S incluye a todos los individuos que pueden infectarse. Cuando un individuo
-

de la clase S tiene un contacto adecuado con un infectado (entendiendo como tal cualquier actividad que resulte en la infección del susceptible) es removido hacia el compartimento de los expuestos (E). En este, los individuos están en fase de latencia, esto es, infectados, pero aún incapaces de transmitir la enfermedad. Después de esta fase, el sujeto pasa a la clase de los infecciosos (I), pudiendo infectar a otros, período tras el cual se traslada al compartimento de los recuperados (R) y queda temporal o permanentemente inmune a la enfermedad. La clase R también incluye a los individuos que fallecen debido a la propia enfermedad. El modelo MSEIR está dado por las siguientes ecuaciones diferenciales

$$\begin{aligned}\frac{dM}{dt} &= B - \delta M(t)S(t) - \mu M(t), \\ \frac{dS}{dt} &= -\delta M(t)S(t) - \beta S(t)I(t) - \mu S(t), \\ \frac{dE}{dt} &= \beta S(t)I(t) - (\epsilon + \mu)E(t), \\ \frac{dI}{dt} &= \epsilon E(t) - (\gamma + \mu)I(t), \\ \frac{dR}{dt} &= \gamma I(t) - \mu R(t),\end{aligned}$$

donde el nuevo parámetro δ es el tiempo promedio de inmunidad temporal.

Existen otros modelos epidemiológicos compartimentales como son: Susceptibles-Expuestos-Infectados (SEI), Susceptibles-Expuestos-Infectados-Susceptibles (SEIS), entre otros (ver [28]).

Todos los modelos anteriores pueden ser resueltos de manera analítica o numérica si se conocen los valores de los parámetros de cada modelo. A este tipo de problemas se les conoce como problemas directos. El problema inverso consiste en determinar los valores de los parámetros del modelo a partir de las soluciones de dicho modelo como se muestra en el Capítulo 3. En la siguiente sección se introduce la noción de qué son un problema directo y un problema inverso, así como la definición de un problema mal planteado.

1.5. Problemas inversos

Los problemas inversos surgen de la necesidad de interpretar medidas indirectas e incompletas. Resolver un problema inverso implica la implementación de un algoritmo computacional que recupera información “útil” de los datos medidos (la palabra “útil” se puede entender mejor en el contexto de una aplicación en particular) [20]. Los problemas inversos, son más difíciles e interesantes y esto se debe, en gran parte, a que tienen múltiples soluciones o bien son insolubles, o presentan

inestabilidad numérica en el sentido de que pequeños errores en los datos medidos puede producir errores grandes en la solución estimada, y además, se presentan habitualmente en la práctica profesional de muchas carreras y profesiones [48]. Los problemas inversos son problemas que consisten en encontrar una propiedad desconocida de un objeto o de un medio a partir de las observaciones de una respuesta de este objeto o medio a una señal de prueba. En este caso, la teoría de problemas inversos proporciona una base teórica para la detección lejana y la evaluación no destructiva.

Veamos la noción de qué son un problema directo e inverso con el siguiente ejemplo: Dada una cierta enfermedad, enumerar los síntomas es un problema directo y sencillo, que ya está resuelto y se puede ver en cualquier texto especializado. En cambio, diagnosticar la enfermedad del paciente a partir de sus síntomas no siempre es sencillo y requiere un médico experimentado [48]. “Podemos decir que un problema inverso es en el cual uno conoce la respuesta, pero no conoce cuál es la pregunta” [20].

Ilustraremos esto con un diagrama para entender mejor la diferencia entre problemas directos e inversos.

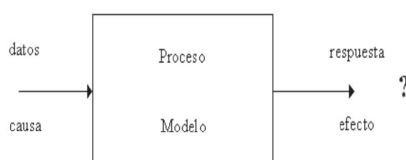


Figura 1.1: Problema Directo. Fuente: Martínez, Víctor. Problemas inversos: los casi olvidados de la Matemática Educativa, 2011 [48].

Observando la Figura 1.1 el problema directo sería calcular la consecuencia (respuesta o efecto) una vez conocidos la causa y el modelo.

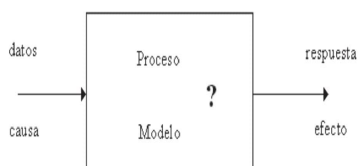


Figura 1.2: Problema inverso, especificación. Fuente: Martínez, Víctor. Problemas inversos: los casi olvidados de la Matemática Educativa, 2011 [48].

En cambio, el problema inverso se podría plantear de dos maneras. Por una parte el problema inverso que trata de calcular el modelo conociendo la consecuencia y la causa, como se muestra en la Figura 1.2, y por otro lado, el problema inverso para obtener la causa conociendo el modelo y la consecuencia, como se muestra en

la Figura 1.3.

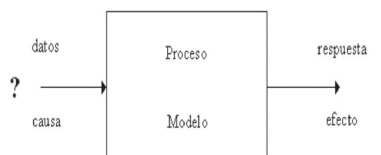


Figura 1.3: Problema inverso, causalidad. Fuente: Martínez, Víctor. Problemas inversos: los casi olvidados de la Matemática Educativa, 2011 [48].

Para el estudio tanto del problema inverso como del directo, se requiere la siguiente definición debida a Hadamard [43]. Un problema es “bien planteado” si cumple las siguientes tres condiciones:

1. **Existencia:** Debe existir al menos una solución.
2. **Unicidad:** Debe haber como máximo una solución.
3. **Estabilidad:** La solución debe depender continuamente de los datos.

Generalmente, los problemas inversos son mal planteados, mientras que los problemas directos son bien planteados. En la mayoría de los problemas inversos, no se tiene toda la información y raramente se puede responder a las tres condiciones anteriores de un problema bien planteado en sentido de Hadamard, por lo que se consideran “mal planteados”. El problema de identificación de parámetros de una ecuación diferencial es un problema mal planteado, pues puede haber más de una solución, o pequeños errores en los datos de entrada pueden producir errores grandes en la solución aproximada (ver [2]), es decir, se tiene una inestabilidad numérica. Para abordar la inestabilidad numérica se utilizan técnicas de regularización, las cuales producen soluciones estables respecto a datos medidos con o sin ruido, como son los métodos de regularización de Tikhonov y el de Lauvrentiev [43]. En esta tesis, nos concentramos en estudiar los problemas directo e inverso del modelo SIR. En particular, el problema inverso estudiado en este trabajo es mal planteado en sentido de Hadamard, ya que puede haber más de una solución o puede presentar inestabilidad numérica. Este último aspecto, no se aborda en esta tesis. Otros ejemplos de problemas mal planteados se pueden consultar en [3, 8, 18, 19, 24, 39, 40, 47, 53, 68, 77].

Capítulo 2

Planteamiento del problema y resultados numéricos

En este capítulo, se aborda el planteamiento de los problemas directo e inverso del modelo epidemiológico SIR y se presentan los resultados numéricos obtenidos.

2.1. Problema directo e inverso

Al sistema (1.1) se le deben incluir condiciones iniciales para hallar una única solución, lo que nos lleva al siguiente problema de valores iniciales (2.1), el cual es utilizado para ejecutar simulaciones numéricas del modelo SIR:

$$\left\{ \begin{array}{l} \frac{dS}{dt} = -\beta S(t) \frac{I(t)}{N}, \quad S(0) = S_0, \\ \frac{dI}{dt} = \beta S(t) \frac{I(t)}{N} - \gamma I(t), \quad I(0) = I_0, \\ \frac{dR}{dt} = \gamma I(t), \quad R(0) = R_0, \end{array} \right. \quad (2.1)$$

para todo instante de tiempo $t \in [t_0, t_f]$ con valores iniciales en el tiempo inicial $t_0 = 0 : S(0) = S_0 = N - I_0$, $I(0) = I_0$ y $R(0) = R_0$, donde N es el tamaño de la población y los parámetros β y γ son valores positivos dados. $S_0 \geq 0$, $I_0 \geq 0$, y $R_0 \geq 0$ son valores dados de las condiciones iniciales $S(0)$, $I(0)$ y $R(0)$ en el tiempo inicial $t_0 = 0$ para poder hallar una única solución del problema (2). A partir de aquí, se utilizarían métodos numéricos para resolver (2) en el intervalo de tiempo $t \in [t_0, t_f]$. De esta forma, podemos definir los problemas directo e inverso del modelo SIR.

Definición 1. (Problema Directo): El Problema Directo (PD) del modelo SIR, consiste en determinar las variaciones de las poblaciones (compartimentos) en di-

Planteamiento del problema y resultados numéricos

2.2 Planteamiento del problema de identificación de parámetros

ferentes instantes del tiempo, resolviendo el sistema de ecuaciones diferenciales ordinarias con condiciones iniciales (2.1) dados los parámetros β y γ , elegidos en un conjunto factible $\Omega \subset \mathbb{R}^2$ de parámetros de dicho modelo.

Definición 2. (Problema Inverso): El Problema Inverso (PI) del modelo SIR, consiste en determinar los parámetros β y γ de dicho modelo a partir de las variaciones de las poblaciones en diferentes instantes del tiempo, de forma que al resolver el PD para estos parámetros encontremos las mismas variaciones de las poblaciones.

En la práctica, si COVID-19 es modelado por el modelo SIR (2.1) los instantes de tiempo son medidos en días, por lo que se registra el número diario de casos infectados, lo que da las variaciones de este compartimento. La solución del PD produce datos que brindan información sobre cómo se comporta la enfermedad infecciosa. La solución del PD es necesaria para plantear y resolver el PI como se muestra a continuación.

2.2. Planteamiento del problema de identificación de parámetros

Para plantear el PI del modelo SIR, considere una pandemia causada por un nuevo virus, como el coronavirus de COVID-19, donde las tasas de transmisión β y de recuperación γ son desconocidas, y que denotaremos estas, de aquí en adelante, por β^* y γ^* , respectivamente. Suponiendo que la única información disponible es el número diario de casos infectados, se plantea el siguiente **problema de optimización** que toma en cuenta los casos diarios reportados, y que la solución de (2.1) se ajuste a esa información, para algún par de valores (β, γ) . Dado que estos parámetros impulsan la dinámica de transmisión de la epidemia, ellos serían las variables de interés en el problema de optimización. Con esto, la función objetivo se puede formular como:

$$\left\{ \begin{array}{l} \text{minimizar } g(\omega) = \sum_{i=0}^N (I_r(t_i) - I^\omega(t_i))^2, \\ \text{sujeto a } \omega \in \Omega \subset \mathbb{R}^n, \end{array} \right. \quad (2.2)$$

donde I_r es el número reportado de casos de COVID-19, I^ω es la solución de (2.1) dado un $\omega = (\beta, \gamma)$ particular de una región factible Ω también conocida como región de sentido epidemiológico. Así, se resuelve (2.1) tomando $\omega = (\beta, \gamma)$ en Ω . Después de ello, se compara $I^\omega(t_i)$ contra $I_r(t_i)$, en cada instante tiempo t_i . En los ejemplos numéricos se consideran los datos $I_r(t_i)$ generados de manera numérica para un par de valores exactos β^* y γ^* , con el objetivo de simular los datos requeridos para resolver el PI. Para esto, se resuelve primero el PD del modelo SIR, es decir, primero se resuelve el sistema (2.1) para un par de valores exactos β^* y γ^* , y se consideran

los valores $I_r(t_i)$ en cada punto t_i de una partición del intervalo $[t_0, t_f]$. Después, se resuelve el PI, es decir, se hallan estimaciones aproximadas de los parámetros β^* y γ^* . Estos datos $I_r(t_i)$ son denotados por $I(t_i)$. En este contexto, el **problema de optimización** (2.2) busca encontrar un $\omega = (\beta, \gamma)$ que minimice la distancia entre el número de casos reportados contra los que predice el modelo. Dichos problemas de optimización se pueden resolver utilizando métodos iterativos, métodos de construcción gráfica o métodos heurísticos como es la inteligencia artificial [69]. Así, el PI consiste en determinar los parámetros β^* y γ^* , minimizando la función objetivo g del problema (2.2) (ver [69]). Otra forma de encontrar una buena estimación de los parámetros desconocido $\omega = (\beta^*, \gamma^*)$ es en sentido clásico de mínimos de cuadrados, en donde la función objetivo, que denotaremos por \tilde{g} , es el cuadrado de una norma ponderada de residuos [23]. En este trabajo, para resolver el PI se minimizará la función objetivo \tilde{g} propuesta en [23], usando el software DIFFPAR de Edsberg and Wedin (1995):

$$\left\{ \begin{array}{l} \text{minimizar } \tilde{g}(\omega) = \sum_{i=0}^N w_1^i (S_r(t_i) - S^\omega(t_i))^2 + w_2^i (I_r(t_i) - I^\omega(t_i))^2 + w_3^i (R_r(t_i) - R^\omega(t_i))^2, \\ \text{sujeto a } \omega \in \Omega \subset \mathbb{R}^n, \text{ donde } \omega = (\beta, \gamma). \end{array} \right. \quad (2.3)$$

donde $w_j^i \geq 0$ son los pesos de los datos de entrada $S_r(t_i)$, $I_r(t_i)$ y $R_r(t_i)$, respectivamente, para $i = 1, 2, \dots, N$ y $j = 1, 2, 3$. Para resolver el PI se usará DIFFPAR para hallar una estimación del punto (β^*, γ^*) donde la función objetivo \tilde{g} es la dada en el **problema de optimización** (2.3).

2.3. Resultados numéricos

En esta sección, se presentan los resultados numéricos obtenidos por medio de ejemplos sintéticos, asociados a los problemas PD y PI del modelo SIR. Para obtener dichos resultados, se elaboraron programas en MATLAB para resolver el PD y se utilizó DIFFPAR (el cual es un software que se ejecuta en MATLAB) para resolver el PI. En los ejemplos numéricos, se elige el conjunto factible de pares de parámetros $\omega = (\beta, \gamma)$ a un conjunto $\Omega \subset \mathbb{R}^2$.

2.3.1. Resultados numéricos del PD

Consideremos que la epidemia de COVID-19, que es una enfermedad causada por un coronavirus SARS-CoV-2, es modelada por el modelo SIR para estudiar la dinámica de dicha enfermedad. Para ilustrar la dinámica de dicha enfermedad, se resuelve el PD para dos pares de parámetros del modelo SIR, como se muestra en los siguientes dos ejemplos.

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

Ejemplo 2.3.1. Se resuelve el PD para el par de parámetros $\omega = (\beta, \gamma) = (0.6, 0.1) \in \Omega = [0.1, 1] \times [0.05, 1] \subset \mathbb{R}^2$, con valores iniciales $I(t_0) = I_0 = 1, S(t_0) = S_0 = N - I_0 = 999$, y $R(t_0) = R_0 = 0$ en el tiempo inicial $t_0 = 0$, donde $N = 1000$. Para resolver (2.1), se aplica el método de Dormand-Prince 5(4) [21] implementada en la nueva función *ode45* de las nuevas versiones de MATLAB [67]. En versiones anteriores de MATLAB, el método de Runge Kutta de orden 4 fue implementada en la vieja función de *ode45* de MATLAB, que aquí no es aplicado para resolver el PD. De hecho, la versión de MATLAB utilizada en este trabajo es 2013a, en donde funciona bien DIFFPAR para resolver el PI. En la Figura 2.1, se presenta la gráfica de la solución numérica de (2.1) para estos valores de β y γ , en el intervalo de tiempo $[0, 61]$, considerando un transcurso de tiempo de 61 días. La solución numérica es calculada prácticamente en cada instante de tiempo.

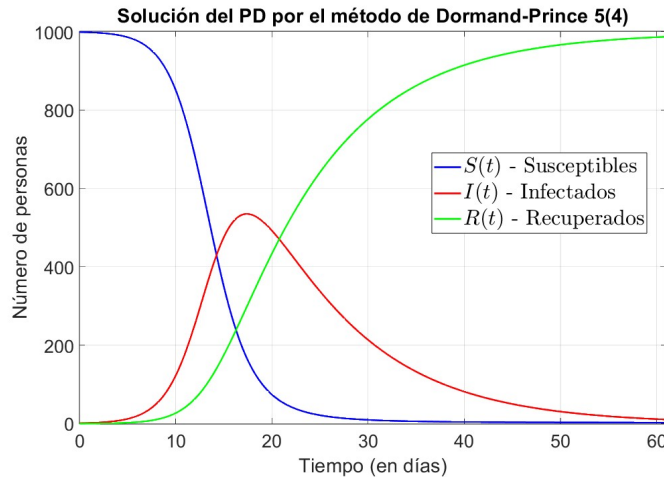


Figura 2.1: Solución del PD del modelo SIR para $(\beta, \gamma) = (0.6, 0.1) \in \Omega$, con $I_0 = 1, S_0 = 999, R_0 = 0$, y $N = 1000$.

Ejemplo 2.3.2. Se resuelve el PD para $(\beta, \gamma) = (0.6, 0.3) \in \Omega$, y los valores iniciales: $I(t_0) = I_0 = 1, S(t_0) = S_0 = N - I_0 = 999$ y $R(t_0) = R_0 = 0$ en $t_0 = 0$, con $N = 1000$. Para estos valores se obtiene la solución numérica de (2.1) mostrada en la Figura 2.2, para el intervalo de tiempo $[0, 61]$. La solución de (2.1) fue calculada por el método Dormand-Prince 5(4), implementado en la función *ode45* de MATLAB, en un mallado de más de 62 puntos del intervalo $[0, 61]$.

Las Figuras 2.1 y 2.2 muestran el comportamiento del modelo para cada par de valores de $(\beta, \gamma) \in \Omega$, con los mismos valores iniciales del modelo SIR. Obsérvese que, si la tasa de infección β es la misma y la tasa de recuperación γ aumenta, entonces el número de infectados disminuye, como se muestra en la Figura 2.2.

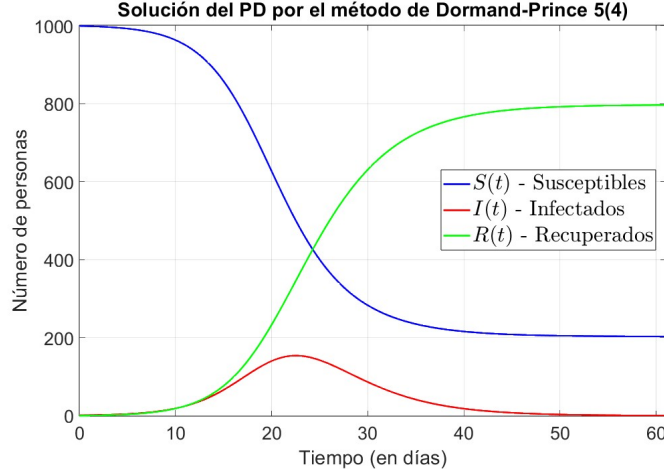


Figura 2.2: Solución del PD del modelo SIR para $(\beta, \gamma) = (0.6, 0.3) \in \Omega$, con $I_0 = 1$, $S_0 = 999$, $R_0 = 0$ y $N = 1000$.

2.3.2. Resultados numéricos del PI

Para hallar una estimación de los valores desconocidos β^* y γ^* del PI se usará DIFFPAR, que minimizará la función objetivo \tilde{g} propuesta en [69] y mostrada en el **problema de optimización (2.3)**. DIFFPAR está basado en un método de regularización local del algoritmo de Gauss-Newton, para encontrar un mínimo local del problema (2.3) desde un punto inicial dado $\theta_0 = (\beta_0, \gamma_0, S_0, I_0, R_0)$, donde $S_0 = S(0)$, $I_0 = I(0)$, y $R_0 = R(0)$ son las condiciones iniciales del modelo SIR en el tiempo inicial $t_0 = 0$. El algoritmo es globalmente convergente, es decir, converge a un mínimo local para un punto inicial no muy alejado del punto exacto desconocido $\theta^* = (\beta^*, \gamma^*, S_0^*, I_0^*, R_0^*)$ de (2.3). En este caso, el algoritmo de Gauss-Newton empieza en un punto inicial θ_0 cercano al punto exacto θ^* , en donde S_0^* , I_0^* y R_0^* son siempre constantes e iguales a las condiciones iniciales del modelo SIR $S_0 = S(0)$, $I_0 = I(0)$ y $R_0 = R(0)$, respectivamente, en cada iteración del algoritmo. Denotemos las estimaciones del par de parámetros $\omega^* = (\beta^*, \gamma^*) \in \Omega$ del PI en cada iteración del algoritmo por $\omega_n = (\beta_n, \gamma_n) \in \Omega$, para $n = 0, 1, 2, 3, \dots$, y denotemos la solución del PD, para el par de parámetros estimados $\omega_n = (\beta_n, \gamma_n)$ por S^{ω_n} , I^{ω_n} y R^{ω_n} . Para resolver los PD, existen métodos numéricos tanto para *problemas stiff* (ver el Apéndice A.2.2) como para *problemas no stiff* [23, 33, 34, 44, 67]. Para resolver los PD en cada iteración del algoritmo de Gauss-Newton DIFFPAR usa un algoritmo basado en el método de Runge-Kutta de tercer orden para problemas no stiff, y un método BDF (por sus siglas en inglés Backwards Differentiation Formula) de tercer orden para *problemas stiff*. Para nuestros ejemplos numéricos se elige el método de Runge-Kutta de tercer orden para problemas no stiff. Por otra parte, el problema de hallar el mínimo global de la función objetivo g o de \tilde{g} empezando en cualquier punto inicial θ_0 , no se aborda en este trabajo.

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

Consideremos los siguientes errores absolutos (EA) y errores relativos (ER):
 $EA(\beta_n, \beta^*) = |\beta_n - \beta^*|$, $EA(\gamma_n, \gamma^*) = |\gamma_n - \gamma^*|$, $ER(\beta_n, \beta^*) = \frac{|\beta_n - \beta^*|}{|\beta^*|}$, y
 $ER(\gamma_n, \gamma^*) = \frac{|\gamma_n - \gamma^*|}{|\gamma^*|}$ de las aproximaciones (β_n, γ_n) del PI.

Ejemplo 2.3.3. En este ejemplo se determinará cuánto afecta la partición equiespa-ciada del intervalo de tiempo en la solución del PI del modelo SIR usando DIFFPAR. Para esto se realizan varias pruebas numéricas con diferentes mallas uniformes del intervalo de tiempo, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo iterativo de DIFFPAR. Para estas pruebas numéricas, se considera que los parámetros desconocidos del PI son $(\beta^*, \gamma^*) = (0.667, 0.167)$, con condiciones iniciales $I(0) = I_0 = 1$, $S(0) = S_0 = N - I_0 = 999$, $R(0) = R_0 = 0$, y $N = 1000$, en el intervalo de tiempo $[0, 30]$. En este caso, la solución del PD para $(\beta^*, \gamma^*) = (0.667, 0.167)$ está graficada en la Figura 2.3. La gráfica de la función objetivo $g(\beta, \gamma)$ del problema de optimización (2.2) está en la Figura 2.4 para diferentes valores de β y γ , donde $(\beta, \gamma) \in \Omega = [0.1, 1] \times [0.05, 1.5] \subset \mathbb{R}^2$. Para graficar la función $g(\beta, \gamma)$ se resolvió el PD para diferentes valores de β y γ por el Método Implícito de Euler (MIE), con un tamaño de paso $h = 0.01$, y así poder evaluar $g(\beta, \gamma)$ en cada punto (β, γ) de un mallado rectangular de la región Ω mostrado en el lado derecho de la Figura 2.4. En la Figura 2.4 se pueden observar los máximos y mínimos locales de la función $g(\beta, \gamma)$ dada en el problema (2.2). Esto puede funcionar como un método gráfico para estimar los valores de los parámetros de β^* y γ^* .

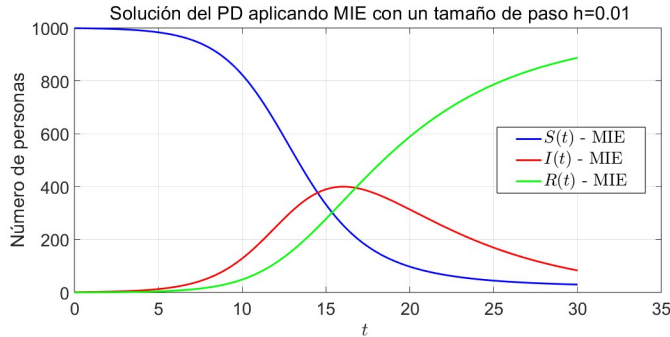


Figura 2.3: Solución del PD del modelo SIR para $(\beta, \gamma) = (0.667, 0.167) \in \Omega$, con $I_0 = 1$, $S_0 = 999$, $R_0 = 0$, y $N = 1000$, aplicando el Método Implícito de Euler (MIE) en el intervalo de tiempo $[0, 30]$.

Los resultados numéricos (en *formato short e* de MATLAB) del PI para los diferentes parámetros establecidos en el párrafo anterior aplicando el software DIFFPAR están dados en las tablas 2.1 a la 2.5 comenzando en diferentes puntos iniciales (β_0, γ_0) para el *método de Runge-Kutta* implementado en DIFFPAR, en donde se minimiza la función objetivo \tilde{g} propuesta en [23].

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

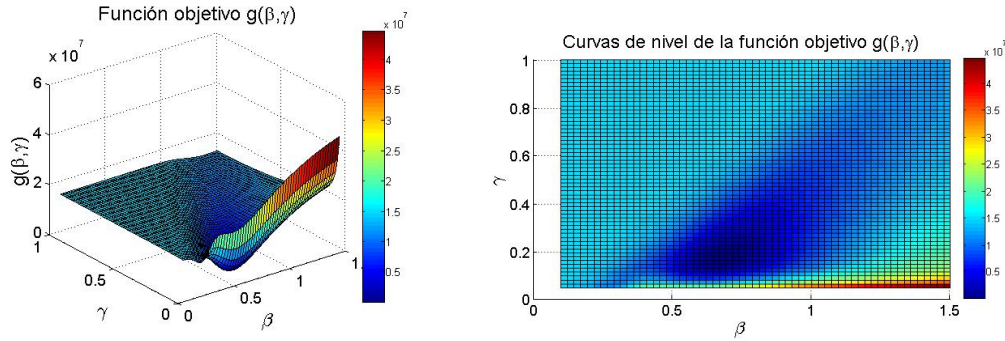


Figura 2.4: Gráfica de la función objetivo $g(\beta, \gamma)$, para los datos exactos de los parámetros $(\beta^*, \gamma^*) = (0.667, 0.167)$ y los valores iniciales S_0 , I_0 y R_0 dados del ejemplo 2.3.3, en la región factible $\Omega = [0.1, 1.5] \times [0.05, 1]$.

Los resultados numéricos de la Tabla 2.1 muestran que los ER de los parámetros estimados están alrededor del orden $1e - 2$ para un espaciado equidistante del tiempo de 30 y 50 particiones del intervalo $[0, 30]$. Para un espaciado del tiempo de 100 y 200 los ER de los parámetros estimados aumentan. Puede observarse que las mejores estimaciones de los parámetros del modelo SIR se consiguen cuando el espaciado de tiempo es de 30, donde el valor estimado se aproxima bastante bien al punto exacto $(\beta^*, \gamma^*) = (0.667, 0.167)$, con ER de $7.9010e - 03$ y $2.5269e - 02$ de los parámetros β_n y γ_n , respectivamente, en la iteración máxima $n = 25$. Para obtener una mejor aproximación de los parámetros en los espaciados equidistantes de 100 y 200 del tiempo es necesario aumentar el número de iteraciones. Por ejemplo, en el espaciado de 100 con 40 iteraciones se tiene una mejor estimación de los parámetros $\beta_n = 6.6498e - 01$ y $\gamma_n = 1.6555e - 01$, con un ER de $3.0285e - 03$ y $8.6826e - 03$ de los parámetros β_n y γ_n , respectivamente, en la iteración máxima $n = 40$ y con una tolerancia de paro de $1e - 5$. También, si cambiamos la tolerancia de paro por $1e - 3$ (con un máximo de 40 iteraciones) se obtienen aproximaciones muy cercanas a las obtenidas con una tolerancia de $1e - 5$ y un máximo de iteraciones de 40. En este caso, las estimaciones para una tolerancia de $1e - 3$ son $\beta_n = 6.6497e - 01$ y $\gamma_n = 1.6542e - 01$, con un ER de $3.0435e - 03$ y $9.4611e - 03$, respectivamente, en la iteración máxima $n = 40$. Por otro parte, cuando tenemos un espaciado equidistante de 30, si aumentamos las iteraciones máximas a 40, con una tolerancia de $1e - 5$, se obtienen las aproximaciones $\beta_n = 6.6602e - 01$ y $\gamma_n = 1.6625e - 01$, con ER de $1.4693e - 03$ y $4.4910e - 03$, respectivamente. En este mismo caso, si disminuimos la tolerancia a $1e - 3$ y con la misma iteración máxima de 40, se obtienen las aproximaciones $\beta_n = 6.6604e - 01$ y $\gamma_n = 1.6626e - 01$, con ER de $1.4393e - 03$ y $4.4311e - 03$, respectivamente. De lo anterior, para un espaciado equidistante de 30 y considerando que siempre se inicia en el mismo punto inicial $k_0 = (\beta_0, \gamma_0) = (0.8, 0.1)$ para el algoritmo iterativo de DIFFPAR, se puede observar que se obtienen mejores aproximaciones de los parámetros β^* y γ^* si aumentamos

el número máximo de iteraciones y la tolerancia de paro a $1e - 3$, obteniendo los ER del mismo orden que cuando la tolerancia de paro es de $1e - 5$. Por lo tanto, es mejor un espacio equidistante del tiempo de 30, considerando una tolerancia de paro de $1e - 3$, con el mismo número de iteraciones máximas y empezando en un mismo punto inicial del algoritmo iterativo de DIFFPAR. Algo similar muestran los resultados de la Tabla 2.2, empezando en un punto inicial diferente $k_0 = (\beta_0, \gamma_0) = (1, 0.2)$ y con diferentes espaciado equidistantes del intervalo $[0, 30]$.

Por otra parte, las tablas 2.3, 2.4 y 2.5 muestran que las estimaciones de los parámetros están alejadas del punto exacto $(\beta^*, \gamma^*) = (0.667, 0.167)$. Los resultados de estas tablas muestran que si escogemos un punto inicial bastante alejado del punto exacto $(\beta^*, \gamma^*) = (0.667, 0.167)$ el algoritmo iterativo de DIFFPAR puede arrojar parámetros muy alejados de estos parámetros exactos. Esto es consistente con el método de DIFFPAR pues, como se mencionó antes, el *método de Runge-Kutta* es un método que calcula mínimos locales de la función \tilde{g} propuesta en [23], y si elegimos un punto muy alejado del mínimo de interés, el método iterativo nos podría arrojar otro mínimo local. Así que DIFFPAR da una buena aproximación de los parámetros del modelo SIR si elegimos puntos cercanos al mínimo de interés. Cabe mencionar que en esta tesis se está trabajando con *datos exactos (casos reportados de COVID-19)*, que son generados de manera numérica o sintética, de las distribuciones de cada compartimento S , I y R en cada instante de tiempo $t \in [0, 30]$. Además, la elección del espaciado equidistante del intervalo de tiempo depende del problema de estudio abordado. Para COVID-19, los casos reportados son dados por día, así que el intervalo de tiempo está medido (dividido) en días.

A continuación, tomando en cuenta las observaciones del ejemplo anterior sobre el espaciado equidistante del intervalo de tiempo y de la tolerancia de paro, se consideran los siguientes dos ejemplos para ilustrar el PI.

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

Para un punto inicial del algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (0.8, 0.1)$, con un espaciado de 30.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	8.0000e-01	1.0000e-01	1.0530e+06	1.3300e-01	1.9940e-01	6.7000e-02	4.0120e-01
2	5.4688e-01	1.0161e-01	6.4271e+05	1.2012e-01	1.8009e-01	6.5390e-02	3.9156e-01
3	6.4736e-01	1.1641e-01	1.3892e+05	1.9640e-02	2.9445e-02	5.0590e-02	3.0293e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	6.6173e-01	1.6278e-01	9.3303e+02	1.9640e-02	2.9445e-02	4.2200e-03	0.0253
25	6.6173e-01	1.6278e-01	8.8863e+02	5.2700e-03	7.9010e-03	4.2200e-03	2.5269e-02

Para un punto inicial del algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (0.8, 0.1)$, con un espaciado de 50.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	5.4499e-01	1.0158e-01	1.7543e+06	1.2201e-01	1.8292e-01	1.0158e-01	3.9174e-01
2	6.5404e-01	1.1649e-01	1.0948e+06	1.2960e-02	1.9430e-02	1.1649e-01	3.0246e-01
3	6.5404e-01	1.2082e-01	2.4165e+05	1.2960e-02	1.9430e-02	1.2082e-01	2.7653e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	6.6290e-01	1.6176e-01	1.8402e+03	4.1000e-03	6.1469e-03	1.6176e-01	3.1377e-02
25	6.6290e-01	1.6176e-01	1.5529e+03	4.1000e-03	6.1469e-03	1.6176e-01	3.1377e-02

Para un punto inicial del algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (0.8, 0.1)$, con un espaciado de 100.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	8.0000e-01	1.0000e-01	3.5074e+06	1.3300e-01	1.9940e-01	6.7000e-02	4.0120e-01
2	5.4356e-01	1.0155e-01	2.2260e+06	1.2344e-01	1.8507e-01	6.5450e-02	3.9192e-01
3	6.5924e-01	1.1655e-01	5.0742e+05	7.7600e-03	1.1634e-02	5.0450e-02	3.0210e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	6.5924e-01	1.5790e-01	1.0018e+04	7.7600e-03	1.1634e-02	9.1000e-03	5.4491e-02
25	6.5924e-01	1.5883e-01	8.5225e+03	7.7600e-03	1.1634e-02	8.1700e-03	4.8922e-02

Para un punto inicial del algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (0.8, 0.1)$, con un espaciado de 200.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	8.0000e-01	1.0000e-01	7.0136e+06	1.3300e-01	1.9940e-01	6.7000e-02	4.0120e-01
2	5.4284e-01	1.0154e-01	4.4890e+06	1.2416e-01	1.8615e-01	6.5460e-02	3.9198e-01
3	6.6189e-01	1.2113e-01	1.0450e+06	5.1100e-03	7.6612e-03	4.5870e-02	2.7467e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	6.6271e-01	1.6264e-01	4.7590e+03	4.2900e-03	6.4318e-03	4.3600e-03	2.6108e-02
25	6.6271e-01	1.6312e-01	4.2013e+03	4.2900e-03	6.4318e-03	3.8800e-03	2.3234e-02

Tabla 2.1: Resultados numéricos (en *formato short e* de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1, 0.2)$, con un espaciado de 30.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.0000e+00	2.0000e-01	2.3147e+06	3.3300e-01	4.9925e-01	3.3000e-02	1.9760e-01
2	5.2602e-01	1.8870e-01	1.9276e+06	1.4098e-01	2.1136e-01	2.1700e-02	1.2994e-01
3	9.3455e-01	1.9870e-01	1.7045e+06	2.6755e-01	4.0112e-01	3.1700e-02	1.8982e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	6.8596e-01	1.8457e-01	1.0987e+04	1.8960e-02	2.8426e-02	1.7570e-02	1.0521e-01
25	6.8596e-01	1.8457e-01	1.0987e+04	1.8960e-02	2.8426e-02	1.7570e-02	1.0521e-01

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1, 0.2)$, con un espaciado de 50.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.0000e+00	2.0000e-01	3.8559e+06	3.3300e-01	4.9925e-01	3.3000e-02	1.9760e-01
2	5.2393e-01	1.8866e-01	3.3027e+06	1.4307e-01	2.1450e-01	2.1660e-02	1.2970e-01
3	9.4195e-01	1.9879e-01	2.9552e+06	2.7495e-01	4.1222e-01	3.1790e-02	1.9036e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	6.8434e-01	1.8208e-01	1.4882e+04	1.7340e-02	2.5997e-02	1.5080e-02	9.0299e-02
25	6.8434e-01	1.8208e-01	1.4882e+04	1.8960e-02	2.8426e-02	1.5080e-02	9.0299e-02

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1, 0.2)$, con un espaciado de 100.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.0000e+00	2.0000e-01	7.7086e+06	3.3300e-01	4.9925e-01	3.3000e-02	1.9760e-01
2	5.2235e-01	1.8862e-01	6.7420e+06	1.4465e-01	2.1687e-01	2.1620e-02	1.2946e-01
3	9.4780e-01	1.9885e-01	6.0913e+06	2.8080e-01	4.2099e-01	3.1850e-02	1.9072e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	6.8579e-01	1.8373e-01	3.4909e+04	1.8790e-02	2.8171e-02	1.6730e-02	1.0018e-01
25	6.8579e-01	1.8373e-01	3.4908e+04	1.8790e-02	2.8171e-02	1.6730e-02	1.0018e-01

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1, 0.2)$, con un espaciado de 200.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.0000e+00	2.0000e-01	1.5414e+07	3.3300e-01	4.9925e-01	3.3000e-02	1.9760e-01
2	5.2156e-01	1.8861e-01	1.3622e+07	1.4544e-01	2.1805e-01	2.1610e-02	1.2940e-01
3	9.5083e-01	1.9888e-01	1.2370e+07	2.8383e-01	4.2553e-01	3.1880e-02	1.9090e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	6.8541e-01	1.8356e-01	6.7500e+04	1.8410e-02	2.7601e-02	1.6560e-02	9.9162e-02
25	6.8541e-01	1.8356e-01	6.7498e+04	1.8410e-02	2.7601e-02	1.6560e-02	9.9162e-02

Tabla 2.2: Resultados numéricos (en *formato short e* de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1.2, 0.3)$, con un espaciado de 30.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.2000e+00	3.0000e-01	3.6927e+06	5.3300e-01	7.9910e-01	1.3300e-01	7.9641e-01
2	1.2000e+00	2.9995e-01	3.6930e+06	5.3300e-01	7.9910e-01	1.3295e-01	7.9611e-01
3	1.2000e+00	2.9989e-01	3.6933e+06	5.3300e-01	7.9910e-01	1.3289e-01	7.9575e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	1.2000e+00	2.9876e-01	3.6998e+06	5.3300e-01	7.9910e-01	1.3176e-01	7.8898e-01
25	1.2000e+00	2.9871e-01	3.7001e+06	5.3300e-01	7.9910e-01	1.3176e-01	7.8898e-01

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1.2, 0.3)$, con un espaciado de 50.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.2000e+00	3.0000e-01	6.1521e+06	5.3300e-01	7.9910e-01	1.3300e-01	7.9641e-01
2	1.2000e+00	2.9995e-01	6.1526e+06	5.3300e-01	7.9910e-01	1.3295e-01	7.9611e-01
3	1.2000e+00	2.9989e-01	6.1531e+06	5.3300e-01	7.9910e-01	1.3289e-01	7.9575e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	1.2000e+00	2.9876e-01	6.1640e+06	5.3300e-01	7.9910e-01	1.3176e-01	7.8898e-01
25	1.2000e+00	2.9870e-01	6.1646e+06	5.3300e-01	7.9910e-01	1.3170e-01	7.8862e-01

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1.2, 0.3)$, con un espaciado de 100.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.2000e+00	3.0000e-01	1.2301e+07	5.3300e-01	7.9910e-01	1.3300e-01	7.9641e-01
2	1.2000e+00	2.9995e-01	1.2302e+07	5.3300e-01	7.9910e-01	1.3295e-01	7.9611e-01
3	1.2000e+00	2.9989e-01	1.2303e+07	5.3300e-01	7.9910e-01	1.3289e-01	7.9575e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	1.2000e+00	2.9875e-01	1.2324e+07	5.3300e-01	7.9910e-01	1.3175e-01	7.8892e-01
25	1.2000e+00	2.9870e-01	1.2325e+07	5.3300e-01	7.9910e-01	1.3170e-01	7.8862e-01

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1.2, 0.3)$, con un espaciado de 200.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.2000e+00	3.0000e-01	2.4597e+07	5.3300e-01	7.9910e-01	1.3300e-01	7.9641e-01
2	1.2000e+00	2.9995e-01	2.4599e+07	5.3300e-01	7.9910e-01	1.3295e-01	7.9611e-01
3	1.2000e+00	2.9989e-01	2.4601e+07	5.3300e-01	7.9910e-01	1.3289e-01	7.9575e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	1.2000e+00	2.9875e-01	2.4645e+07	5.3300e-01	7.9910e-01	1.3175e-01	7.8892e-01
25	1.2000e+00	2.9870e-01	2.4647e+07	5.3300e-01	7.9910e-01	1.3175e-01	7.8892e-01

Tabla 2.3: Resultados numéricos (en *formato short e* de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (0.2, 0.3)$, con un espaciado de 30.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	2.0000e-01	3.0000e-01	1.1195e+07	4.6700e-01	7.0015e-01	1.3300e-01	7.9641e-01
2	2.0000e-01	3.0001e-01	1.1195e+07	4.6700e-01	7.0015e-01	1.3301e-01	7.9647e-01
3	2.0000e-01	3.0003e-01	1.1195e+07	4.6700e-01	7.0015e-01	1.3303e-01	7.9659e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	2.0000e-01	3.0030e-01	1.1195e+07	4.6700e-01	7.0015e-01	1.3330e-01	7.9820e-01
25	2.0000e-01	3.0031e-01	1.1195e+07	4.6700e-01	7.0015e-01	1.3331e-01	7.9826e-01

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (0.2, 0.3)$, con un espaciado de 50.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	2.0000e-01	3.0000e-01	1.8370e+07	4.6700e-01	7.0015e-01	1.3300e-01	7.9641e-01
2	2.0000e-01	3.0001e-01	1.8370e+07	4.6700e-01	7.0015e-01	1.3301e-01	7.9647e-01
3	2.0000e-01	3.0003e-01	1.8370e+07	4.6700e-01	7.0015e-01	1.3303e-01	7.9659e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	2.0000e-01	3.0030e-01	1.8370e+07	4.6700e-01	7.0015e-01	1.3330e-01	7.9820e-01
25	2.0000e-01	3.0031e-01	1.8370e+07	4.6700e-01	7.0015e-01	1.3331e-01	7.9826e-01

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (0.2, 0.3)$, con un espaciado de 100.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	2.0000e-01	3.0000e-01	3.6307e+07	4.6700e-01	7.0015e-01	1.3300e-01	7.9641e-01
2	2.0000e-01	3.0001e-01	3.6307e+07	4.6700e-01	7.0015e-01	1.3301e-01	7.9647e-01
3	2.0000e-01	3.0003e-01	3.6307e+07	4.6700e-01	7.0015e-01	1.3303e-01	7.9659e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	2.0000e-01	3.0030e-01	3.6307e+07	4.6700e-01	7.0015e-01	1.3330e-01	7.9820e-01
25	2.0000e-01	3.0031e-01	3.6307e+07	4.6700e-01	7.0015e-01	1.3331e-01	7.9826e-01

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (0.2, 0.3)$, con un espaciado de 200.

<i>Iter. n</i>	β_0	γ_0	<i>Residual</i>	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	2.0000e-01	3.0000e-01	7.2183e+07	4.6700e-01	7.0015e-01	1.3300e-01	7.9641e-01
2	2.0000e-01	3.0001e-01	7.2183e+07	4.6700e-01	7.0015e-01	1.3301e-01	7.9647e-01
3	2.0000e-01	3.0003e-01	7.2183e+07	4.6700e-01	7.0015e-01	1.3303e-01	7.9659e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	2.0000e-01	3.0030e-01	7.2183e+07	4.6700e-01	7.0015e-01	1.3330e-01	7.9820e-01
25	2.0000e-01	3.0031e-01	7.2183e+07	4.6700e-01	7.0015e-01	1.3331e-01	7.9826e-01

Tabla 2.4: Resultados numéricos (en *formato short e* de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1.5, 1)$, con un espaciado de 30.

Iter. n	β_0	γ_0	Residual	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.5000e+00	1.0000e+00	1.9407e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
2	1.5000e+00	1.0000e+00	1.9407e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
3	1.5000e+00	1.0000e+00	1.9408e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	1.5000e+00	1.0003e+00	1.9421e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9898e+00
25	1.5000e+00	1.0003e+00	1.9422e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9898e+00

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1.5, 1)$, con un espaciado de 50.

Iter. n	β_0	γ_0	Residual	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.5000e+00	1.0000e+00	3.1927e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
2	1.5000e+00	1.0000e+00	3.1928e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
3	1.5000e+00	1.0000e+00	3.1929e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	1.5000e+00	1.0003e+00	3.1948e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9898e+00
25	1.5000e+00	1.0003e+00	3.1949e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9898e+00

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1.5, 1)$, con un espaciado de 100.

Iter. n	β_0	γ_0	Residual	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.5000e+00	1.0000e+00	6.3229e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
2	1.5000e+00	1.0000e+00	6.3230e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
3	1.5000e+00	1.0000e+00	6.3232e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	1.5000e+00	1.0003e+00	6.3269e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9898e+00
25	1.5000e+00	1.0003e+00	6.3271e+06	8.3300e-01	1.2489e+00	8.3300e-01	4.9898e+00

Para un punto inicial de algoritmo Gauss-Newton $k_0 = (\beta_0, \gamma_0) = (1.5, 1)$, con un espaciado de 200.

Iter. n	β_0	γ_0	Residual	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
1	1.5000e+00	1.0000e+00	1.2583e+07	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
2	1.5000e+00	1.0000e+00	1.2584e+07	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
3	1.5000e+00	1.0000e+00	1.2584e+07	8.3300e-01	1.2489e+00	8.3300e-01	4.9880e+00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	1.5000e+00	1.0002e+00	1.2591e+07	8.3300e-01	1.2489e+00	8.3320e-01	4.9892e+00
25	1.5000e+00	1.0002e+00	1.2591e+07	8.3300e-01	1.2489e+00	8.3320e-01	4.9892e+00

Tabla 2.5: Resultados numéricos (en *formato short e* de MATLAB) de la estimación de los parámetros $\beta^* = 0.667$ y $\gamma^* = 0.167$ del modelo SIR usando DIFFPAR, para un espaciado igual del intervalo de tiempo de 30 (días), 50, 100 y 200, respectivamente, con una tolerancia de 10^{-5} y un máximo de 25 iteraciones para el algoritmo.

Ejemplo 2.3.4. Considerando que la dinámica de COVID-19 es estudiada por el modelo SIR, entonces los datos o casos reportados (mediciones) de infectados son registrados por día, así que el intervalo de tiempo está medido (dividido) en días. En este contexto, si los parámetros desconocidos del PI son $\omega^* = (\beta^*, \gamma^*) = (0.6, 0.1)$, con condiciones iniciales $I(0) = I_0 = 1$, $S(0) = S_0 = N - I_0 = 999$, $R(0) = R_0 = 0$, y $N = 1000$, en el intervalo de tiempo $[0, 61]$, entonces, para determinar

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

los casos registrados por día para este par de parámetros $\omega^* = (0.6, 0.1)$ se resuelve el PD correspondiente. La solución numérica del PD, es decir, la variación en el tiempo de cada uno de los compartimentos se muestra en la Figura 2.5, en donde los susceptibles S^{ω^*} están marcados con asteriscos en color magenta, los infectados I^{ω^*} están marcados con asteriscos en color morado, y los recuperados R^{ω^*} están marcados con asteriscos en color verde. En este caso, la Figura 2.6 muestra la gráfica de la función objetivo $g(\beta, \gamma)$ dada en el problema de optimización (2.2) para diferentes valores de β y γ , donde $(\beta, \gamma) \in \Omega = [0.1, 1] \times [0.05, 1] \subset \mathbb{R}^2$. En la Figura 2.6 se pueden observar los máximos y mínimos locales de la función $g(\beta, \gamma)$ dada en el problema (2.2). Esto puede funcionar como un método gráfico para estimar los valores de los parámetros de β^* y γ^* .

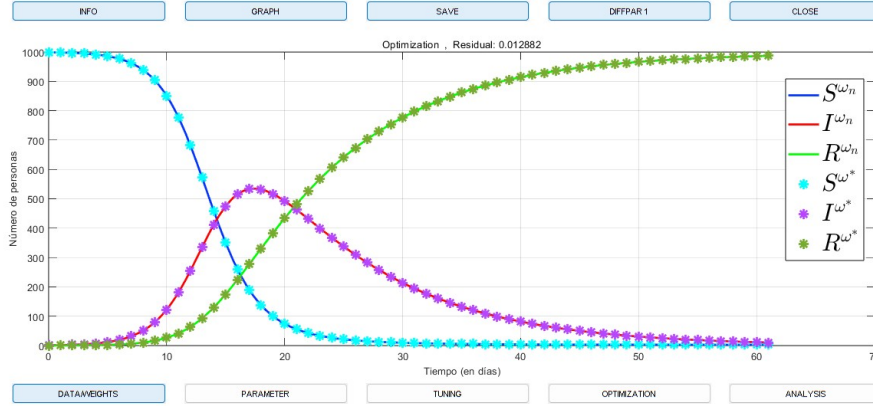


Figura 2.5: Gráfica de las soluciones ajustadas S^{ω_n} (curva en color azul), I^{ω_n} (curva en color rojo), y R^{ω_n} (curva en color verde) del modelo SIR contra los datos de casos por día (puntos marcados con asteriscos para cada compartimento S^{ω^*} , I^{ω^*} y R^{ω^*}), correspondientes al par de estimaciones $\omega_n = (\beta_n, \gamma_n)$ del par de parámetros $\omega^* = (\beta^*, \gamma^*)$ del PI, en la iteración $n = 52$.

Ahora, para resolver el PI, apliquemos el software DIFFPAR para minimizar la función objetivo \tilde{g} propuesta en [23], con una partición equiespaciada de 61 puntos del intervalo de tiempo $[0, 61]$. En este caso, se obtiene la siguiente aproximación numérica de β^* y γ^* dada en la Tabla 2.6, la cual se aproxima muy bien al punto exacto $(\beta^*, \gamma^*) = (0.6, 0.1)$, tomando como punto inicial $(\beta_0, \gamma_0) = (0.3, 0.05)$. El ajuste de las soluciones de (2.1) para $(\beta_n, \gamma_n) = (5.9999e - 01, 9.9992e - 02)$, en la iteración $n = 52$, con los datos de casos por día (puntos marcados con asteriscos para cada compartimento) se muestran en la Figura 2.5, cuyo residual es $1.2882e - 02$. En este caso, $ER(\beta_n, \beta^*) = 1.6667e - 05$ y $ER(\gamma_n, \gamma^*) = 8.0000e - 05$. En la Tabla 2.7, se muestran otros resultados que se aproximan al punto exacto $(\beta^*, \gamma^*) = (0.6, 0.1)$, tomando otro punto inicial $(\beta_0, \gamma_0) = (0.8, 0.3)$. En este caso, el

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

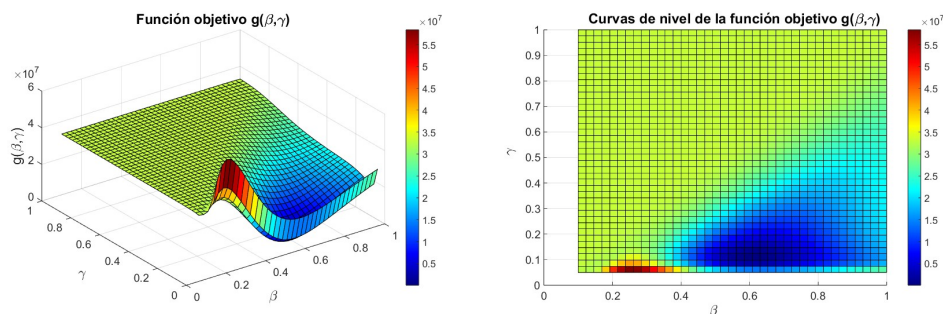


Figura 2.6: Gráfica de la función objetivo $g(\beta, \gamma)$, para los datos exactos de los parámetros $(\beta^*, \gamma^* = (0.6, 0.1))$ y los valores iniciales S_0, I_0 y R_0 dados en el ejemplo 3, en la región factible $\Omega = [0.1, 1] \times [0.05, 1]$.

algoritmo de DIFFPAR paró en la iteración 81, en donde $ER(\beta_n, \beta^*) = 2.1873e - 01$ y $ER(\gamma_n, \gamma^*) = 1.5331e + 00$.

Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.3, 0.05)$ con 52 iteraciones (Iter).							
Iter.	β_n	γ_n	Residual	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
0	3.0000e-01	5.0000e-02	1.0860e+07	3.0000e-01	0.5000e-01	5.0000e-02	5.0000e-01
1	5.0866e-01	7.4953e-02	7.1824e+05	9.1300e-02	1.5220e-01	2.5047e-02	2.5047e-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
52	5.9999e-01	9.9992e-02	1.2882e-02	1.000e-05	1.6667e-05	8.0000e-06	8.0000e-05

Tabla 2.6: Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.3, 0.05)$, con una tolerancia de $1e - 3$ y un máximo de 90 iteraciones para el algoritmo de Gauss Newton implementado en DIFFPAR. Parámetros exactos $(\beta^*, \gamma^*) = (0.6, 0.1)$ del PI.

Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.8, 0.3)$ con 81 iteraciones (Iter).							
Iter.	β_n	γ_n	Residual	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
0	8.0000e-01	3.0000e-02	1.3787e+06	2.0000e-01	3.3330e-01	7.0000e-02	7.0000e-01
1	7.3124e-01	2.8349e-01	1.3037e+06	1.3124e-01	2.1873e-01	1.8349e-01	1.8349e+00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
81	7.3124e-01	2.5331e-01	2.9216e-02	1.3124e-01	2.1873e-01	1.5331e-01	1.5331e+00

Tabla 2.7: Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.8, 0.3)$, con una tolerancia de $1e - 3$ y un máximo de 90 iteraciones para el algoritmo de Gauss Newton implementado en DIFFPAR. Parámetros exactos $(\beta^*, \gamma^*) = (0.6, 0.1)$ del PI.

Ejemplo 2.3.5. Consideremos que los parámetros desconocidos del PI son $(\beta^*, \gamma^*) = (0.6, 0.3)$, con condiciones iniciales $I(0) = I_0 = 1$, $S(0) = S_0 = N - I_0 = 999$, $R(0) = R_0 = 0$ y $N = 1000$, en el intervalo de tiempo $[0, 61]$. Los casos registrados por día para este par de parámetros $\omega^* = (0.6, 0.1)$ se hallan resolviendo el PD correspondiente. La solución numérica del PD se muestra en la Figura 2.7, en donde los susceptibles, S^{ω^*} , están marcados con asteriscos en color magenta, los infectados, I^{ω^*} , están marcados con asteriscos en color morado, y los recuperados, R^{ω^*} , están marcados con asteriscos en color verde. Para este caso, la Figura 2.8 muestra la gráfica de la función objetivo $g(\beta, \gamma)$ dada en el problema de optimización (2.2) para diferentes valores de $(\beta, \gamma) \in \Omega = [0.1, 1] \times [0.05, 1] \subset \mathbb{R}^2$, donde se pueden observar los puntos de máximos y mínimos locales de la función objetivo $g(\beta, \gamma)$.

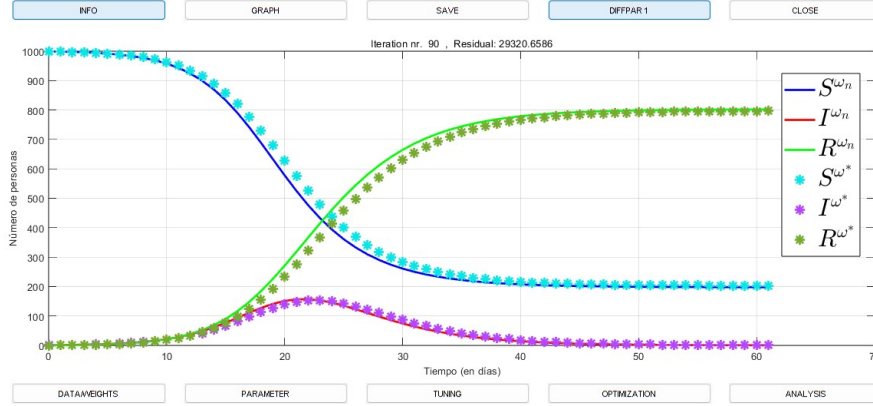


Figura 2.7: Gráfica de las soluciones ajustadas S^{ω_n} (curva en color azul), I^{ω_n} (curva en color rojo), y R^{ω_n} (curva en color verde), del modelo SIR contra los datos de casos por día (puntos marcados con asteriscos para cada compartimento S^{ω^*} , I^{ω^*} y R^{ω^*}), correspondientes al par de estimaciones $\omega_n = (\beta_n, \gamma_n)$ del par de parámetros $\omega^* = (\beta^*, \gamma^*)$ del PI, en la iteración $n = 90$.

Por último, para resolver el PI, aplicamos el software DIFFPAR para minimizar la función objetivo \tilde{g} propuesta en [23], con una partición equiespaciada de 61 puntos del intervalo de tiempo $[0, 61]$. En este caso, se obtiene la siguiente aproximación numérica de β^* y γ^* dada en la Tabla 2.8. Se observa que el punto estimado $(\beta_n, \gamma_n) = (6.2217e - 01, 3.0807e - 01)$, en la iteración máxima $n = 90$, se aproxima bien al punto exacto $(\beta, \gamma) = (0.6, 0.3)$. El ajuste de las soluciones para $(\beta_n, \gamma_n) = (6.2217e - 01, 3.0807e - 01)$ con los datos de casos por día (puntos marcados con asteriscos para cada compartimento) se muestra en la Figura 2.7, cuyo residual es $2.9321e + 04$. En este caso, $ER(\beta_n, \beta^*) = 3.6950e - 02$ y $ER(\gamma_n, \gamma^*) = 2.6900e - 02$. En la Tabla 2.9, se muestran otros resultados que se aproximan al punto exacto β^* y $\gamma^* = (0.6, 0.3)$. En este caso, el algoritmo de DIFFPAR paró en la iteración máxima de 90, donde el $ER(\beta_n, \beta^*) = 1.8900e - 02$ y $ER(\gamma_n, \gamma^*) = 4.1200e - 02$.

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

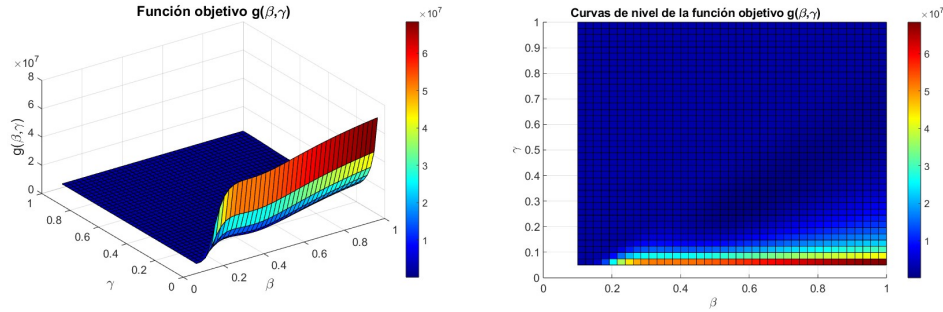


Figura 2.8: Gráfica de la función objetivo $g(\beta, \gamma)$, para los datos exactos de los parámetros $(\beta^*, \gamma^* = (0.6, 0.3))$ y los valores iniciales S_0, I_0 y R_0 dados en el ejemplo 3, en la región factible $\Omega = [0.1, 1] \times [0.05, 1]$.

Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.8, 0.3)$ con 90 iteraciones (Iter).							
Iter.	β_n	γ_n	Residual	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
0	5.0000e-01	3.2000e-01	4.0261e+06	1.0000e-01	1.6667e-01	2.0000e-02	6.6667e-02
1	8.2794e-01	3.2955e-01	3.2286e+06	2.2794e-01	3.7990e-01	2.9550e-02	9.8500e-02
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
90	6.2217e-01	3.0807e-01	2.9321e+04	2.2170e-02	3.6950e-02	8.0700e-03	2.6900e-02

Tabla 2.8: Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.85, 0.32)$, con una tolerancia de $1e - 3$ y un máximo de 90 iteraciones para el algoritmo de Gauss Newton implementado en DIFFPAR. Parámetros exactos $(\beta^*, \gamma^*) = (0.6, 0.3)$ del PI.

Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.8, 0.3)$ con 90 iteraciones (Iter).							
Iter.	β_n	γ_n	Residual	$EA(\beta_n, \beta^*)$	$ER(\beta_n, \beta^*)$	$EA(\gamma_n, \gamma^*)$	$ER(\gamma_n, \gamma^*)$
0	7.0000e-01	3.2000e-01	3.1172e-01	1.0000e-01	1.6667e-01	2.0000e-02	6.6667e-02
1	5.6778e-01	3.1172e-01	3.1172e-01	3.2220e-02	5.3700e-02	1.1720e-02	3.9067e-02
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
90	5.8866e-01	3.1236e-01	1.2561e+05	1.1340e-02	1.8900e-02	1.2360e-02	4.1200e-02

Tabla 2.9: Resultados numéricos para el punto inicial $(\beta_0, \gamma_0) = (0.7, 0.32)$, con una tolerancia de $1e - 3$ y un máximo de 90 iteraciones para el algoritmo de Gauss Newton implementado en DIFFPAR. Parámetros exactos $(\beta^*, \gamma^*) = (0.6, 0.3)$ del PI.

Finalmente, de los resultados numéricos presentados, el software DIFFPAR podrá dar una buena aproximación del mínimo de interés (β^*, γ^*) del PI si elegimos de manera adecuada la iteración máxima, la tolerancia de paro y un punto inicial

Planteamiento del problema y resultados numéricos

2.3 Resultados numéricos

no tan alejado del mínimo (global) de interés de la función objetivo \tilde{g} planteada en el problema de optimización 2.3, como se esperaba del método de Gauss-Newton implementado en DIFFPAR, el cual es un método local.

2.4. Conclusión

En este trabajo, se dio solución a los problemas PD y PI del modelo SIR. Se generaron programas en MATLAB para resolver el PD usando el Método Implícito de Euler (para graficar la función objetivo g), y el método método Dormand-Prince 5(4) implementado en la función `ode45` de las nuevas versiones de Matlab. Para el PI se utilizó el software DIFFPAR. En este caso, se generaron los programas necesarios del modelo SIR para poder aplicar DIFFPAR. De los resultados numéricos presentados para el PI de estimación de los parámetros del modelo SIR, se observa que la aplicación del software DIFFPAR es factible. Cuando se toman puntos iniciales cercanos a los datos exactos (y desconocidos para el PI), β^* y γ^* , los valores estimados por DIFFPAR son bastante aceptables. Dos puntos interesantes que no se abordaron en este trabajo, consisten en hallar el mínimo global de la función objetivo \tilde{g} dada en el problema de optimización (2.3) y utilizar datos reales.

Contribuciones

Este trabajo impacta en la solución de problemas directo e inverso de identificación de parámetros constantes de un sistema dinámico multidimensional con varios parámetros (sistema de ecuaciones diferenciales ordinarias con coeficientes que pueden depender de algunos parámetros que es necesario estimar) y cuya dinámica se desarrolla a lo largo de un periodo de tiempo, usando el software DIFFPAR, tales como el modelo SIR y sus variantes, entre otros modelos determinados por algún sistema dinámico multidimensional con varios parámetros que modelan diferentes fenómenos que aparecen en diferentes áreas de la investigación, como la biología matemática [5], ecología [32, 63] o epidemiología [14, 15] que estudian la interacción entre diferentes especies. Otro ejemplo, donde aparecen sistemas dinámicos multidimensionales son para estudiar las reacciones químicas [38], que dependen de ciertos parámetros, como la velocidad de reacción o la constante de equilibrio [58, 70]. Otra aplicación de sistemas dinámicos se encuentran en economía. En este caso, lo que interesa es predecir una situación desfavorable, como una recesión o una depresión y de esta forma planificar con anticipación para reducir el impacto de posibles efectos negativos [22, 55].

Entre las contribuciones de esta Tesis están las siguientes:

- 1) Se planteó el problema inverso del modelo SIR mediante un problema de optimización (que consiste en determinar los parámetros del modelo minimizando una función objetivo sujeta a restricciones). La ventaja de este enfoque es que puede trasladarse o plantearse en diferentes contextos, como los mencionados antes, en donde aparecen sistemas dinámicos multidimensionales con varios parámetros a

determinar [22].

2) Se desarrollaron programas en Matlab para resolver los problemas directo e inverso del modelo SIR y se presenta un pequeño manual para usar DIFFPAR (mostrados en el Apéndice). La ventaja de estos programas es que pueden ser modificados por otros usuarios para resolver los problemas directo e inverso de su interés usando DIFFPAR. Cabe señalar que DIFFPAR solo resuelve sistemas dinámicos multidimensionales con coeficientes constantes, aunque se puede aplicar el software varias veces en instantes cortos de tiempo para determinar a trozos los coeficientes del modelo SIR o de alguno de sus variantes. Además, el modelo SIR, o alguno de sus variantes, nos pueden ayudar a estudiar la dinámica de una enfermedad infecciosa en el tiempo como es la epidemia COVID-19 y desarrollar estrategias para reducir la propagación de la infección.

3) Por último, se mostró la factibilidad de aplicar DIFFPAR con ejemplos sintéticos para validar y analizar la convergencia del método Gauss-Newton (implementado en DIFFPAR) tomando diferentes puntos iniciales, usando los programas computaciones desarrollados en Matlab (que están incluidas en el Apéndice) para resolver tanto el problema directo como el problema inverso. Además, parte de los resultados numéricos presentados en esta Tesis están aceptados para su publicación en las memorias del X Coloquio de Modelación Matemática en Biología, Ciencias Médicas e Ingeniería.

Trabajos futuros

Como trabajos futuros de investigación sobre el método propuesto para resolver el PI del modelo SIR se tienen:

1. Resolver el PI usando mediciones o datos perturbados (datos registrados con error de cada compartimento).
2. Resolver el PI usando datos reales, por ejemplo de la epidemia COVID-19.
3. Resolver el PI usando otros métodos iterativos, como el método de gradiente conjugado, para hallar el mínimo de la función objetivo g o \tilde{g} del problema de optimización, y comparar los resultados obtenidos con los del método de Gauss-Newton, implementada en DIFFPAR.
4. Hallar el mínimo global de la función objetivo g o \tilde{g} , usando algún método heurístico como puede ser la inteligencia artificial (algoritmos genéticos, enjambres de partículas, etc.) en combinación con el método de Gauss-Newton implementado en DIFFPAR.

Por otra parte, es posible aplicar el método propuesto en este trabajo en otros contextos y para variantes del modelo SIR como Susceptibles-Infectados

(SI), Susceptibles-Infectados-Susceptibles (SIS), Susceptibles-Expuestos-Infectados (SEI), Susceptibles-Expuestos-Infectados-Susceptibles (SEIS), entre otros, usando DIFFPAR, adecuando los programas que requiere DIFFPAR para cada modelo, para resolver el PI correspondiente a la variante del modelo SIR de estudio.

Apéndice A

Ecuaciones diferenciales ordinarias y métodos numéricos

A.1. Sistema de ecuaciones diferenciales

En muchas ocasiones es deseable describir en términos matemáticos el comportamiento de algunos sistemas o fenómenos de la vida real, que pueden ser físicos, químicos, sociológicos, económicos, etc. La descripción matemática de un sistema o fenómeno se llama modelo matemático y se construye con ciertos objetivos, como el de entender que ocurrirá en el futuro o que pasó en el pasado. Para la realización de un modelo matemático sobre un sistema, primero hay que identificar todas las variables que ocasionan que el sistema cambie y posteriormente establecer de qué manera estas variables afectan al sistema. Es claro que cuantas más variables, que afectan al sistema, se añadan mejor se ajusta el modelo al sistema o fenómeno real. Sin embargo, el modelo será cada vez más complejo cuantas más variables entren en juego; el cual puede ser más complicado de resolver en forma analítica o numérica, en el sentido de que se puede llegar a utilizar muchos recursos y tiempo de cómputo. La mayoría de los modelos matemáticos están descritos por ecuaciones diferenciales ordinarias o parciales que a continuación se definen, así como también se presentan algunos teoremas de existencia y unicidad de las ecuaciones diferenciales ordinarias de primer orden.

Definición A.1.1. *Una ecuación diferencial es aquella en la que interviene al menos una derivada o diferencial de una función incógnita.*

Definición A.1.2. *Ecuación diferencial ordinaria (EDO) es una ecuación que contiene funciones de una sola variable independiente, y una o más de sus derivadas con respecto a dicha variable.*

Definición A.1.3. *Ecuación diferencial parcial (EDP) es una ecuación que contiene funciones de dos o más variables independientes y algunas de sus derivadas con respecto a esas variables. (ver [11].)*

Entonces las ecuaciones diferenciales pueden clasificarse como se sigue:

- Según la cantidad de variables independientes (ordinarias y parciales)
- Por el tipo de operación a la que está afectada la función incógnita (lineales y no lineales)
- Según el orden de la ecuación diferencial (ED). El orden de una ED es el mayor orden de derivación que aparece en la ecuación.

Algunos ejemplos de EDO y EDP son los siguientes.

Ejemplo A.1.1. $\frac{dy}{dx} + 5xy = e^y$, donde la función incógnita es y , que solo depende de la variable independiente x . Esta es una EDO no lineal de primer orden, en donde $\frac{dy}{dx}$ es la derivada de y con respecto de la variable x . También, para las derivadas se puede usar la notación de primas. La $\frac{dy}{dx}$ se denota por y' .

Ejemplo A.1.2. $\frac{d^2y}{dx^2} + 2xy + x^2 = 0$, que en notación de primas es $y'' + 2xy + x^2 = 0$, donde la función incógnita es y que depende de la variable independiente x . Esta es una EDO lineal de segundo orden.

Ejemplo A.1.3. $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = yx^3$, donde la función incógnita es u que depende de las variables independiente x e y . Esta es una EDP lineal de segundo orden.

A.1.1. Sistemas de ecuaciones diferenciales ordinarias de primer orden

Definición A.1.4. *Un sistema de ecuaciones diferenciales ordinarias de primer orden de dimensión m es un conjunto de m ecuaciones diferenciales de primer orden con m funciones incógnitas $y^1(t), \dots, y^m(t)$ que dependen de una sola variable independiente t y su representación viene dada por*

$$\begin{cases} \frac{dy^1(t)}{dt} = f^1(t, y^1(t), \dots, y^m(t)), \\ \vdots \\ \frac{dy^m(t)}{dt} = f^m(t, y^1(t), \dots, y^m(t)), \end{cases}$$

donde f^1, f^2, \dots, f^m son funciones de $m + 1$ variables, y las funciones incógnitas $y^1(t), \dots, y^m(t)$ dependen sólo de la variable independiente t . Este sistema puede ser escrito en su forma matricial equivalente

$$y' = f(t, y),$$

Ecuaciones diferenciales ordinarias y métodos numéricos

A.1 Sistema de ecuaciones diferenciales

donde $y = [y^1(t), y^2(t), \dots, y^m(t)]^T$, $y' = [\frac{dy^1(t)}{dt}, \frac{dy^2(t)}{dt} \dots, \frac{dy^m(t)}{dt}]^T$ y $f(t, y) = [f^1(t, y^1(t), \dots, y^m(t)), f^2(t, y^1(t), \dots, y^m(t)), \dots, f^m(t, y^1(t), \dots, y^m(t))]^T$. Aquí el superíndice T indica la traspuesta de una matriz.

A continuación, se anuncian dos teoremas de existencia y unicidad de una EDO, respectivamente.

Teorema A.1.1. *Considérese la EDO con condiciones iniciales*

$$\begin{cases} y'(t) = f(t, y(t)), \forall t \in I = [t_0, t_f] \subseteq \mathbb{R}, \\ y(t_0) = y_0, \end{cases} \quad (\text{A.1})$$

donde t_0 es llamado tiempo inicial y t_f es llamado tiempo final. Si $f(t, y)$ es continua y acotada en $I \times \mathbb{R}^m$ entonces el problema de valor inicial tiene al menos una solución en $I \times \mathbb{R}^m$.

Teorema A.1.2. *Considérese el sistema de ecuaciones de primer orden dado por A.1. Si se verifican las condiciones*

1. $f(t, y)$ es continua en $I \times \mathbb{R}^m$
2. Se verifica la condición de Lipschitz respecto de la segunda variable en $I \times \mathbb{R}^m$, es decir, existe una constante $L > 0$ tal que para todo $t \in I$ y todo $x = (x^1, \dots, x^m)$, $y = (y^1, \dots, y^m) \in \mathbb{R}^m$ se verifica que
$$|F(t, x) - F(t, y)| \leq L|x - y|.$$

Entonces el problema de valor inicial tiene una única solución en $I \times \mathbb{R}^m$.

Del Teorema A.1.2 se anuncia el siguiente.

Corolario A.1.1. *Considérese el sistema de ecuaciones de primer orden dado por A.1. Si $f(t, y)$ es continua en $I \times \mathbb{R}^m$ y para cada $i = 1, 2, \dots, m$ existen $\frac{\partial f^1(t, y)}{\partial y^i}, \frac{\partial f^2(t, y)}{\partial y^i}, \dots, \frac{\partial f^m(t, y)}{\partial y^i}$ y son continuas en $I \times \mathbb{R}^m$. Entonces el problema de valor inicial tiene una única solución en $I \times \mathbb{R}^m$.*

A.1.2. Ecuaciones diferenciales lineales de primer orden

Otras EDO de primer orden que pueden ser resueltas de forma sencilla son aquellas de tipo lineal, más concretamente.

Definición A.1.5. *Una EDO de primer orden es lineal si se escribe de la forma $a_1(t)y'(t) + a_0(t)y(t) = b(t)$, donde y, a_1, a_0 y b son funciones que solo dependen de la variable t .*

Definición A.1.6. *Una EDO lineal se dice que es homogénea cuando la función $b(t)$ de la definición anterior es cero.*

Ecuaciones diferenciales ordinarias y métodos numéricos

A.1 Sistema de ecuaciones diferenciales

Así, ejemplos de EDO lineales de primer orden homogéneas son: $\sin(x)y'(x) + \cos(x)y(x) = 0$, $e^x y'(x) + y(x) = 0$ y $(x^2 - 1)y'(x) + xy(x) = 0$, donde la función y depende solo de la variable independiente x . En algunas aplicaciones la variable x puede representar la variable del tiempo t .

A continuación, se tiene el siguiente teorema de existencia y unicidad de un sistema lineal de EDO.

Teorema A.1.3. Sean $A(t)$ una función matricial cuadrada de orden n , $b(t)$ una función vectorial, continuas en un intervalo $I \subseteq \mathbb{R}$ y sea el problema de valor inicial

$$\begin{cases} y'(t) = A(t)y(t) + b(t), \forall t \in I \subseteq \mathbb{R} \\ y(t_0) = y_0, \text{ con } t_0 \in \mathbb{R}, \end{cases}$$

donde $y = (y^1, \dots, y^n) \in \mathbb{R}^n$ es la función incógnita vectorial definida en $I \subseteq \mathbb{R}$. Entonces existe una única función vectorial $y(t)$ que es solución del problema (ver [44]).

Solución de algunas EDO lineales homogénea y no homogénea

Para resolver una EDO de la forma $a_1(x)y'(x) + a_0(x)y(x) = b(x)$, primero se resuelve la ecuación homogénea asociada $a_1(x)y'(x) + a_0(x)y(x) = 0$, donde x es la variable independiente. Esta última EDO es separable. La solución de la EDO homogénea es de la forma $y(x) = c_0 y_1(x)$ para una cierta constante c_0 . Luego, si $b(x)$ no es cero, las soluciones de la ecuación $a_1(x)y' + a_0(x)y = b(x)$, vendrán dadas de la forma $y(x) = u(x)y_1(x)$, para una cierta función $u(x)$ a determinar. Es decir, se sustituirá la constante c_0 dada en las soluciones de la EDO homogénea por la función $u(x)$. Entonces se sustituye la función $y(x) = u(x)y_1(x)$ sobre la ecuación $a_1(x)y'(x) + a_0(x)y(x) = b(x)$ y esto nos dará una ecuación de donde podremos despejar $u'(x)$. Finalmente, integrando la expresión de $u'(x)$ calculamos $u(x)$ y, por tanto, las soluciones de la EDO. En lo que sigue solo se escribirá la función $y(x)$, que depende solo de la variable independiente x , como y y $y'(x)$ como y' .

Veamos la solución de los siguientes dos ejemplos aplicando el método analítico presentado en el párrafo anterior.

Ejemplo A.1.4. La EDO $(1 + x^2)y' + xy = 0$ es lineal homogénea, por lo que se puede resolver simplemente teniendo en cuenta que es de tipo separable. La ecuación se puede escribir como

$$\frac{dy}{y} = \frac{-x dx}{1+x^2}$$

Siempre que y no se anule. De hecho, $y(x) = 0$ es solución del problema. Si se integra la ecuación anterior obtenemos $\ln |y| = -\frac{1}{2} \ln(1+x^2) + c$. O equivalentemente $y = \frac{c_0}{\sqrt{1+x^2}}$, para c_0 un número real cualquiera, que incluye la solución $y(x) = 0$ tomando $c_0 = 0$.

Ejemplo A.1.5. Para resolver la EDO de primer orden lineal $\arctg(x)y' - \frac{2}{1+x^2}y = \arctg^3(x)$ primero se debe resolver la EDO homogénea $\arctg(x)y' - \frac{2}{1+x^2}y = 0$. Esta la podemos reescribir (para $y \neq 0$) como

$$\frac{dy}{y} = \frac{2dx}{(\arctg(x))(1+x^2)}$$

Con lo que integrando obtenemos $y = c_0 \arctg^2(x)$. Por tanto, las soluciones de la EDO lineal $\arctg(x)y' - \frac{2}{1+x^2}y = \arctg^3(x)$ las obtendremos de la forma $y = u(x) \arctg^2(x)$ donde, para calcular $u(x)$, sustituimos en la ecuación, con lo que $\arctg^3(x) = \arctg(x)(u(x) \arctg^2(x))' - \frac{2}{1+x^2}(u(x) \arctg^2(x)) = u'(x) \arctg^3(x)$. Tenemos entonces que $u'(x) = 1$ o bien $u(x) = x + c$, por lo que las soluciones de la EDO inicial son $y = (x + c) \arctg^2(x)$.

Pueden consultarse otros métodos analíticos para resolver las EDO con valores iniciales en [78].

A.2. Métodos numéricos

Los métodos numéricos para resolver EDO dan aproximaciones numéricas a las soluciones de éstas. Sin embargo, no siempre una ecuación diferencial tiene una solución, y aún si la tiene, no siempre podemos expresarla en forma explícita o implícita, y en muchos casos tendremos que conformarnos con una aproximación numérica. Para afrontar este problema se recurre a los métodos numéricos que nos brindan de forma aproximada una solución de la EDO de una variable, para lo cual es necesario conocer el valor de la curva solución en un punto. Estos métodos, en lugar de dar una expresión algebraica explícita de la solución, producen una serie de pares ordenados donde la primera componente se corresponde con el valor de la variable independiente y la segunda es una aproximación de la solución en el punto.

En este capítulo se presenta un resumen de una serie de métodos numéricos de paso simple, múltiple o variable, es decir, métodos que permiten calcular el valor aproximado solo utilizando el valor aproximado obtenido en uno o varios pasos previos, de la variable independiente, para aproximar la solución del problema de valor inicial.

A.2.1. Solución numérica del PVI

Consideremos el El problema de EDO (A.1) es también conocido como Problema de Valor Inicial (PVI) de m ecuaciones diferenciales ordinarias de primer orden, donde $f = (f^1, \dots, f^m) : \mathbb{R} \times \mathbb{R}^m \rightarrow \mathbb{R}^m$, $y = (y^1, \dots, y^m) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ e $y_0 = (y_0^1(t_0), \dots, y_0^m(t_0))^T \in \mathbb{R}^m$, mientras que t, t_0 son reales. Entonces el PVI (A.1) consiste en conocer el comportamiento de $y(t)$ en un intervalo $[t_0, t_f] \subseteq \mathbb{R}$ para una condición inicial dada $y(t_0) = y_0$, donde t_0 es el tiempo inicial y t_f es el tiempo final.

Sea $\{t_N\}$ una malla arbitraria del intervalo $[t_0, t_f]$ y denotemos el tamaño de paso por $h_n = t_{n+1} - t_n$, $n = 1, 2, \dots, N$. Si tenemos un mallado uniforme el tamaño de paso se denota por $h = \frac{t_f - t_0}{N}$. Denotemos por y_i las aproximaciones numéricas (dada por algún método numérico) a la solución exacta $y(t_i)$ del PVI para $i = 1, 2, \dots, N$.

Ahora, la fórmula general de un método multipaso o de k -pasos es

$$\left. \begin{aligned} \sum_{j=0}^k \alpha_j y_{n+j} &= h \phi_f(y_{n+k}, y_{n+k-1}, \dots, y_n, t_n, h), \\ y_\mu &= \eta_\mu(h), \quad \mu = 0, 1, \dots, k-1. \end{aligned} \right\} \quad (\text{A.2})$$

donde $k \geq 1$ es el número de pasos del método, α_j son constantes que no dependen de n , $y_\mu = \eta_\mu(h)$ son las condiciones de partida del método (tales que $\lim_{h \rightarrow 0} y_\mu = y(t_\mu)$, $\mu = 0, 1, \dots, k-1$), y_{n+k} es el valor que se quiere calcular en cada iteración, mientras que y_{n+k-i} , con $1 \leq i \leq k$ son valores que se han obtenido previamente.

A continuación se tiene la definición de convergencia de un método numérico y un teorema sobre la convergencia de estos.

Definición A.2.1. (*Convergencia de un método numérico*): Un método numérico es convergente si, para todo PVI (A.1) que satisface las hipótesis del A.1.2, se obtiene que

$$\max_{0 \leq i \leq N} \|y(t_n) - y_n\| \rightarrow 0, \text{ cuando } h \rightarrow 0.$$

Teorema A.2.1. *La condición necesaria y suficiente para que el método numérico sea convergente es que sea consistente y cero-estable.*

Un método numérico es consistente y cero estable si satisface la siguientes definiciones, respectivamente.

Definición A.2.2. (*Método consistente*):

$$\text{Sea } R_{n+k} := \sum_{j=0}^k \alpha_j y(t_{n+j}) - h \phi_f(y_{n+k}, y_{n+k-1}, \dots, y_n, t_n, h). \quad (\text{A.3})$$

Si el residual R_{n+k} definido por (A.3) satisface $\lim_{h \rightarrow 0} \frac{R_{n+k}}{h} = 0$, entonces el método es consistente.

Definición A.2.3. (*Método cero estable*): Sean dos perturbaciones $\{\delta_n\}_{n=0}^N$ y $\{\delta_n^*\}_{n=0}^N$ y las respectivas soluciones perturbadas $\{y_n\}_{n=0}^N$ y $\{y_n^*\}_{n=0}^N$ del sistema perturbado:

$$\left. \begin{aligned} \sum_{j=0}^k \alpha_j y_{n+j} &= h [\phi_f(y_{n+k}, y_{n+k-1}, \dots, y_n, t_n, h) + \delta_{n+k}], \\ y_\mu &= \eta_\mu(h) + \delta_\mu, \quad \mu = 0, 1, \dots, k-1. \end{aligned} \right\}$$

Entonces se dice que el método numérico (A.2) es cero-estable si existen una constantes S y h_0 tal que $\forall h \in (0, h_0]$, $\|y_n - y_n^*\| \leq S\varepsilon$, $0 \leq n \leq N$, siempre que $\|\delta_n - \delta_n^*\| \leq \varepsilon$, $0 \leq n \leq N$.

A.2.2. Clasificación de métodos numéricos para las EDO

En esta sección se enlistan varios tipos de métodos que tienen interesantes propiedades que integran con buenos resultados una clase muy abundante de ecuaciones diferenciales ordinarias como son los problemas stiff. Actualmente, todavía no hay una definición exacta de que es un problema stiff, aunque históricamente se han realizado muchos esfuerzos para separar la clase stiff de la clase no-stiff. En el libro de Lambert [44] se presenta una amplia discusión sobre ambas clases de problemas. Se puede decir que la clase stiff es de alguna manera “difusa”, ya que se pasa de la clase no-stiff a la clase stiff de forma gradual y no hay una frontera clara que separe ambas clases. Así, es corriente que los especialistas digan que un problema es “midly” stiff [59]. Hay problemas de valor inicial que son de tipo stiff en una parte del intervalo de integración siendo no-stiff en otra parte del mismo, intercalándose así zonas no-stiff con zonas stiff. Un ejemplo de este tipo de problemas es el oscilador no lineal de van der Pol, ver páginas 4, 21-24 de [34]. Los especialistas en análisis numéricos con amplia experiencia computacional definen a los problemas stiff como aquellos donde los integradores numéricos basados en métodos explícitos no funcionan satisfactoriamente, es decir, para una precisión determinada, los métodos explícitos (como los métodos Runge-Kutta) necesitan tomar pasos mucho más pequeños que lo que la regularidad de la solución demanda para integrarlos [34].

Algunos de los métodos numéricos para resolver los problemas stiff y no stiff son los siguientes.

Métodos explícitos:

- Métodos de Runge-Kutta de un paso y multipaso, con tamaño de paso constante h [44].
- Métodos predictor-corrector:
 - 1) Método de Milne-Simpson.
 - 2) Método de Adams-Bashforth-Moulton de cuarto orden [13, 44].
- Métodos de Runge-Kutta de paso variable, también denominados de pares encajados:
 - 1) Método de Runge-Kutta-Fehlberg que usa fórmulas Runge-Kutta de orden 4 y 5.
 - 2) Método de Dormand-Prince 5(4) [13, 33, 44].

Métodos implícitos:

- Fórmulas de diferenciación regresiva, en inglés Backward Differentiation Formulas (BDFs), los cuales son métodos multipaso utilizados para resolver problemas stiff [34, 44].

- Fórmulas de diferenciación numérica, en inglés Numerical Differentiation Formulas (NDFs) [44, 67].
- Exponential fitting o métodos adaptados cuando $y(t) \in \text{gen}\{1, \dots, t^{\eta-1}, e^{\lambda t}, \dots, x^{\gamma-1}e^{\lambda t}\}$ [49].

A continuación, se describen algunos de estos métodos numéricos. En especial se describe el Método explícito de Dormand-Prince 5(4) que está implementado en la función `ode45` de las nuevas versiones de MATLAB, así como también, el método implícito de Euler, los cuales son usados para resolver los PD del modelos SIR.

A.2.3. Métodos de Runge Kutta

Los métodos de Runge Kutta tienen el error local de truncamiento del mismo orden que los métodos de Taylor, pero prescinden del cálculo y evaluación de las derivadas de la función $f(t, y)$. Para hallar una solución aproximada del PVI dado por (A.1), consideremos una malla t_0, t_1, \dots, t_N de paso h , donde $t_0 = t_0$ y $t_f = t_N$. En estos puntos es donde se va a obtener la aproximación de la solución.

En esencia, los métodos de Runge-Kutta son generalizaciones de la fórmula básica de Euler $y_{i+1} = hf(t_i, y_i)$ en los que el valor de la función f se reemplaza por un promedio ponderado de valores de f en el intervalo $t_i \leq t \leq t_{i+1}$, es decir,

$$y_{i+1} = y_i + h \underbrace{(w_1 k_1 + w_2 k_3 + \dots + w_m k_m)}_{\text{promedio ponderado}}$$

En esta expresión las ponderaciones $w_i, i = 1, \dots, m$ son constantes para las que en general se pide que su suma sea igual a 1, es decir, $w_1 + w_2 + \dots + w_m = 1$, y cada k_j es la función f evaluada en un punto seleccionado (t, y) para el cual $t_i \leq t \leq t_{i+1}$. Se puede observar que el k_j se definen en forma recursiva en el Método Runge-Kutta de una etapa presentado adelante [78].

Se define el *orden de un método numérico multipaso* como el *orden* p del error local que hay entre la solución exacta $y(t_i)$ y la solución aproximada y_i en cada punto t_i de la malla del intervalo cerrado $[a, b]$ (ver [13, 44]).

Método clásicos de Runge-Kutta

La expresión general de un método Runge Kutta de m evaluaciones es :

$$\left\{ \begin{array}{l} cy_{n+1} = y_n + h\phi(t_n, y_n; h), \\ \phi(t, y_n; h) = \sum_{i=1}^m c_i k_i \\ k_1 = f(t, y_n) \\ k_i = f(t + ha_i, y_n + h \sum_{s=1}^{i-1} b_{is} k_s), i = 2, 3, 4, \dots, m, \\ a_i = \sum_{s=1}^{i-1} b_{is}, i = 2, 3, 4, \dots, m. \end{array} \right.$$

Un método de Runge-Kutta utiliza m -evaluaciones de la función f que son las derivadas de la solución $y(t)$ en varios puntos y después en ϕ se hace el promedio ponderado por lo que tenemos que $\sum_{i=1}^m c_i = 1$. Los valores de los coeficientes a_i , c_i , b_{is} dependen del esquema numérico elegido, dependiente de la regla de cuadratura utilizada. Los esquemas Runge-Kutta pueden ser explícitos o implícitos dependiendo de las constantes b_{is} del esquema. Si esta matriz es triangular inferior con todos los elementos de la diagonal principal iguales a cero; es decir, $b_{is} = 0$ para $s = i, \dots, m-1$, los esquemas son explícitos. Por ejemplo, para construir estos métodos se utiliza la siguiente notación

$$f_t = \frac{\partial f}{\partial t}, f_y = \frac{\partial f}{\partial y}, f_{tt} = \frac{\partial^2 f}{\partial t^2}, f_{ty} = \frac{\partial^2 f}{\partial t \partial y}, \dots$$

y además

$$F = f_t + f f_y, G = f_{tt} + 2f f_{ty} + f^2 f_{yy}$$

Con esta notación el desarrollo de ϕ_t de método Taylor quedaría

$$\phi_T(t, y; h) = f + \frac{1}{2}hF + \frac{1}{6}h^2(Ff_y + G) + O(h^3)$$

Para construir los primeros métodos de Runge-Kutta se consideran los métodos para $R = 3$, es decir

$$\begin{aligned} k_1 &= f(t, y) = f, k_2 = f(t + a_2h, y + ha_2k_1), \text{ y} \\ k_3 &= f(t + a_3h, y + h((a_3 - b_{32}k_1 + b_{32}k_2))). \end{aligned}$$

Utilizando desarrollos de Taylor

$$k_2 = f + a_2hF + \frac{1}{2}a_2h^2G + O(h^3), \text{ y } k_3 = f + a_3hF + h^2(a_2b_{32}Ff_y + \frac{1}{2}a_3^3G) + O(h^3)$$

de donde se concluye que :

$$\phi(t, y; h) = (c_1+c_2+c_3)f+h(c_2a_2+c_3a_3)F+\frac{1}{2}h^2(2c_3a_2b_{32}Ff_y+(c_2a_2^2+c_3a_3^2)G)+O(h^3).$$

Entonces para nuestro método de Runge-Kutta de una etapa, es decir, para $m = 1$ se tiene que $c_2 = c_3 = 0$, quedando la función incremento

$$\phi(t, y; h) = c_1 f + O(h^3),$$

que cuando $c_1 = 1$ es el conocido método de Euler (ver página 2-5 de [71]), esto es, $y_{i+1} = y_i + h\phi(t_i, y_i; h)$ donde $\phi(t_i, y_i; h) = f(t_i, y_i)$, para $i = 0, 1, 2, \dots, N$.

A continuación, se presentan algunos métodos de paso variable.

A.2.4. Método de Dormand-Prince 5(4)

En las nuevas versiones de MATLAB, el Método de Dormand-Prince 5(4) está implementado en la nueva función *ode45* de MATLAB [13, 67], el cual es un método similar al método de Runge-Kutta-Fehlberg [6] o también llamado RKF45 [50]. El programa *ode45* utiliza un par de fórmulas de órdenes $p = 4$ y $(p + 1) = 5$ de Dormand y Prince [21], el cual es conocido como el método DOPRI5, estimando nuevamente el error local como en el método de Runge-Kutta-Fehlberg (ver sección 5.4, página 80 de [6]).

Método de Dormand-Prince 5(4): Para calcular la aproximación y_{n+1} del siguiente paso temporal t_{n+1} de la EDO o PVI (A.1) se deben realizar los siguientes pasos [21]:

- Paso 1) Calcular los siguientes valores:

$$\begin{aligned} k_1 &= h_n f(t_n, y_n), \\ k_2 &= h_n f\left(t_n + \frac{1}{5}h_n, y_n + \frac{1}{5}k_1\right), \\ k_3 &= h_n f\left(t_n + \frac{3}{10}h_n, y_n + \frac{3}{40}k_1 + \frac{9}{40}k_2\right), \\ k_4 &= h_n f\left(t_n + \frac{4}{5}h_n, y_n + \frac{44}{45}k_1 - \frac{56}{15}k_2 + \frac{32}{9}k_3\right), \\ k_5 &= h_n f\left(t_n + \frac{8}{9}h_n, y_n + \frac{19372}{6561}k_1 - \frac{25360}{2187}k_2 + \frac{64448}{6561}k_3 - \frac{212}{729}k_4\right), \\ k_6 &= h_n f\left(t_n + h_n, y_n + \frac{9017}{3168}k_1 - \frac{355}{33}k_2 + \frac{46732}{5247}k_3 + \frac{49}{176}k_4 - \frac{5103}{18656}k_5\right), \\ k_7 &= h_n f\left(t_n + h_n, y_n + \frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6\right). \end{aligned}$$

- Paso 2) Calcular el valor del siguiente paso y_{n+1} mediante el método de Runge-Kutta de orden 4 como

$$y_{n+1} = y_n + \frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6.$$

- Paso 3) Calcular el valor del siguiente paso z_{n+1} mediante el método de Runge-Kutta de orden 5 como

$$z_{n+1} = y_n + \frac{5179}{57600}k_1 + \frac{7571}{16695}k_3 + \frac{393}{640}k_4 - \frac{92097}{339200}k_5 + \frac{187}{2100}k_6 + \frac{1}{40}k_7.$$

- Paso 4) Calcular una estimación del error local $E_{n+1} = z_{n+1} - y_{n+1}$ dada por

$$E_{n+1} = \frac{71}{57600}k_1 - \frac{71}{16695}k_3 + \frac{71}{1920}k_4 - \frac{17253}{339200}k_5 + \frac{22}{525}k_6 + \frac{1}{40}k_7.$$

Observemos que no es necesario calcular y_{n+1} , sólo el error E_{n+1} .

- Paso 5) Si se cumplen la tolerancia $|E_{n+1}|_\infty \leq \epsilon$, entonces $y_{n+1} = z_{n+1}$ y $k_1 = k_7$. En caso contrario, se ensaya con un nuevo tamaño de paso temporal h_{new} tal que

$$h_{new} = 0.9h_n \sqrt[5]{\frac{\epsilon}{|E_{n+1}|_\infty}} \quad \text{o} \quad h_{new} = 0.9h_n \sqrt[4]{\frac{\epsilon}{\left|\frac{E_{n+1}}{h_n}\right|_\infty}} \quad (\text{A.4})$$

donde h_n es la longitud del intervalo de tiempo anterior t_n , ϵ es la tolerancia de paro del algoritmo, y $|E_{n+1}|_\infty = |z_{n+1} - y_{n+1}|_\infty$ es la norma del máximo de E_{n+1} . En caso de que el número del PVI (A.1) sea 1 la norma del error local E_{n+1} es el valor absoluto de E_{n+1} . En la programación, esta nueva h_{new} se utilizará en el cálculo del siguiente paso temporal t_{n+1} , es decir, se hace $h_n = h_{new}$ y se regresa al Paso 1. También, se puede usar otra norma para calcular el error local E_{n+1} y el nuevo tamaño de paso h_{new} , los cuales están dados en la ecuación (4.11) y (4.13), página 168, de [33], donde se multiplica h_n por un factor $fac = 0.8, 0.9$ (como se muestra en (A.4)), $(0.25)^{\frac{1}{5}}$, o $(0.38)^{\frac{1}{5}}$, para que el error sea aceptable en la siguiente iteración con alta probabilidad. Además, de esta forma no se permite que h_n aumente ni disminuya demasiado rápido [33].

Observemos que este método evalúa seis veces la función f para calcular las soluciones de cuarto y quinto orden, aunque el error se controla sólo para la solución de cuarto orden y_{n+1} . Además, Dormand y Prince escogieron los coeficientes de su método para minimizar el error de la solución de quinto orden. Esta es la principal diferencia con el método de Runge-Kutta-Fehlberg, que se construyó de modo que la solución de cuarto orden tenga un error pequeño. Por esa razón, el método de Dormand–Prince 5(4) es más adecuado cuando la solución de orden superior (la solución de quinto orden z_{n+1}) se usa para continuar la integración, una práctica conocida como interpolación local [33]. La Tabla A.1 presenta los coeficientes de este método, los cuales coinciden con los valores de la Tabla 2 del artículo [21] para el orden $p = 4$. La penúltima línea de coeficientes proporciona la solución de cuarto orden y_{n+1} y la última línea la solución de quinto orden z_{n+1} .

Matriz de Butcher del método es: Dormand-Prince 5(4) (DOPRI5)

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
y_{n+1}	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
z_{n+1}	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{4}$

Tabla A.1: Coeficientes del método Dormand-Prince 5(4) (DOPRI5). Fuente: Tomado de la Tabla 5.2, página 178, Hairer, E., Nørsett, S. P., y Wanner, G., 1993, [33].

A.2.5. Métodos implícitos BDFs de paso variable

Por simplicidad, consideremos una ecuación diferencial ordinaria unidimensional con condiciones iniciales:

$$y' = f(t, y), \quad y(t_0) = y_0, \quad \text{for } t \in [t_0, t_f], \quad (\text{A.5})$$

donde t_0 es el tiempo inicial y t_f es el tiempo final. Sea $\{t_N\}$ una malla arbitraria del intervalo $[t_0, t_f]$ y denotemos el tamaño de paso por $h_n = t_n - t_{n-1}$, para $n = 1, 2, \dots, N$.

Las BDFs de paso variable se definen como sigue. Suponga que las k soluciones aproximadas y_{n-k+1}, \dots, y_n ya se han calculado, y considere el polinomio $p_k(t)$ de grado $k \geq 1$ que interpola el $k + 1$ pares (t_j, y_j) para $j = n - k + 1, \dots, n, n + 1$ (incluida la aproximación desconocida y_{n+1}) y $p_k(t)$ satisface la ecuación diferencial (A.5) en t_{n+1} , es decir,

$$p'_k(t_{n+1}) = f(t_{n+1}, p_k(t_{n+1})) = f(t_{n+1}, y_{n+1}), \quad \text{donde } p_k(t_{n+1}) = y_{n+1}. \quad (\text{A.6})$$

Usando la fórmula de interpolación de Newton tenemos

$$p_k(t) = \sum_{j=0}^k \left(\prod_{i=0}^{j-1} (t - t_{n+1-i}) \right) \nabla^j y[t_{n+1}, t_n, \dots, t_{n-j+1}], \quad (\text{A.7})$$

donde las diferencias divididas $\nabla^j y[t_n, \dots, t_{n-j}]$ son definidas recursivamente por

$$\begin{aligned} \nabla^0 y[t_{n-j}] &= y_{n-j}, \quad j = -1, 0, 1, \dots, k-1, \\ \nabla^j y[t_{n+1}, \dots, t_{n-j+1}] &= \frac{\nabla^{j-1} y[t_{n+1}, \dots, t_{n-j+2}] - \nabla^{j-1} y[t_n, \dots, t_{n-j+1}]}{t_{n+1} - t_{n-j+1}}, \end{aligned}$$

para $j = 1, 2, 3, \dots, k$.

Combinando (A.6) y (A.7) y despejando $y(t_{n+1})$, obtenemos las BDFs de tamaño de paso variable dado por

$$\sum_{j=1}^k h_n \left(\prod_{i=1}^{j-1} (t_{n+1} - t_{n+1-i}) \right) \cdot \nabla^j y[t_{n+1}, \dots, t_{n-j+1}] = h_n f(t_{n+1}, y_{n+1}),$$

donde k es el número de pasos del método.

En el caso de un tamaño de paso constante h , las BDFs de k pasos están dados por

$$\sum_{j=1}^k \frac{1}{j} \nabla^j y_{n+1} = h f(t_{n+1}, y_{n+1}),$$

que es estable para $k \leq 6$ e inestable para $k \geq 7$ (ver p. 381, Sección III.3 de [33]). Para un tamaño de paso temporal constante h , las BDFs de un paso ($k = 1$) es el método implícito de Euler:

$$y_{n+1} - y_n = h f(t_{n+1}, y_{n+1}), \quad (\text{A.8})$$

donde la nueva aproximación en el siguiente paso temporal es y_{n+1} . En la siguiente subsección se da un método para calcular el valor y_{n+1} .

A.2.6. Solución del PVI por el método implícito de Euler

En el método implícito de Euler (A.8), para un tamaño de paso temporal constante h , la nueva aproximación y_{n+1} se calcula iterativamente por el *método iterativo de Newton modificado* (ver páginas 12-13, de [44]): $y_{n+1}^{(\nu+1)} = y_{n+1}^{(\nu)} + \Delta y_{n+1}^{(\nu)}$, para $\nu = 0, 1, 2, \dots$, donde el incremento $\Delta y_{n+1}^{(\nu)}$ se calcula resolviendo el sistema lineal de ecuaciones:

$$A \Delta y_{n+1}^{(\nu)} = r_{n+1}^{(\nu)}, \quad (\text{A.9})$$

donde $A = \left(I - h \frac{\partial f}{\partial y} \right)$, $\frac{\partial f}{\partial y}$ es la matriz Jacobiana de f respecto a la variable y , de tamaño 1×1 evaluada en el punto $(t_{n+1}, y_{n+1}^{(\nu)})$, y el residual $r_{n+1}^{(\nu)} = y_n - y_{n+1}^{(\nu)} - h f(t_{n+1}, y_{n+1}^{(\nu)})$. Una fórmula análoga a (A.9) para las BDFs y NDFs pueden ser encontradas en [67]. Por lo tanto, en cada iteración del *método iterativo de Newton modificado* se tiene que

$$y_{n+1}^{(\nu+1)} = y_{n+1}^{(\nu)} - A^{-1} r_{n+1}^{(\nu)}, \quad \nu = 0, 1, 2, \dots \quad (\text{A.10})$$

El *método iterativo de Newton modificado* que fue programado en MATLAB inicia tomando $y_{n+1}^{(\nu)} = y_n$ para $\nu = 0$, y el algoritmo se detiene cuando se satisface que $r_{n+1}^{(\nu)} \leq \epsilon$, para una tolerancia dada ϵ , y se toma $y_{n+1} = y_{n+1}^{(\nu+1)}$.

Si $y, f \in \mathbb{R}^m$ ($m \geq 2$) entonces $A = \left(I - h \frac{\partial f}{\partial y}\right)$ es una matriz de tamaño $m \times m$, $\frac{\partial f}{\partial y}$ es la matriz Jacobiana de f respecto a la variable y de tamaño $m \times m$ dada por

$$\frac{\partial f}{\partial y} = \begin{bmatrix} \frac{\partial f^1}{\partial y^1} & \cdots & \frac{\partial f^1}{\partial y^m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f^m}{\partial y^1} & \cdots & \frac{\partial f^m}{\partial y^m} \end{bmatrix}$$

evaluada en el punto (t_{n+1}, y_{n+1}^v) . Además, el sistema lineal de ecuaciones (A.9) se puede resolver mediante la matriz inversa A^{-1} de A , por factorización LU o por iteración de Gauss-Seidel, entre otros métodos para resolver el sistema. La solución de los problemas directos del modelo SIR fueron resueltos aplicando el método implícito de Euler, resolviendo el sistema (A.9) usando la matriz inversa de A , en cada iteración del *método iterativo de Newton modificado*.

Observemos que para el cálculo de la matriz Jacobiana $\frac{\partial f}{\partial y}$ se requiere en el método de Euler implícito, los métodos BDFs y NDFs, y también en cada paso temporal se resuelva iterativamente un sistema de ecuaciones lineales. En este caso, el costo computacional por paso se incrementa dramáticamente; mientras que los métodos explícitos de Runge-Kutta sólo requieren un número finito de evaluaciones de la función f , con lo cual el tiempo de cómputo es menor con estos métodos. Los dos códigos explícitos de Runge-Kutta, ode23 y ode45, en versiones anteriores de MATLAB han sido reemplazados por códigos con los mismos nombres que solucionan algunas deficiencias en el diseño y aprovechan los desarrollos en la teoría y la práctica de los métodos de Runge-Kutta. La nueva versión de ode23 se basa en el par Bogacki-Shampine 3(2) y la nueva versión de ode45 se basa en el método de Dormand-Prince 5(4). Sin embargo, los métodos implícitos en términos de diferencias regresivas BDFs y NDFs son los más utilizados para resolver problemas stiff. El comando de MATLAB que utiliza estas fórmulas es ode15s. La función ode15s genera las matrices jacobianas numéricamente. Sin embargo, este solucionador a menudo se ejecuta más rápido si se proporciona la matriz Jacobiana de manera explícita [67].

Apéndice B

Códigos de los programas elaborados

B.1. Programas para los problemas directos del modelo SIR

En esta sección se presentan los códigos de los programas empleados en el Capítulo 3, en lenguaje *MATLAB*, para resolver el PD del modelo SIR. Estos programas se pueden encontrar en el siguiente link: https://drive.google.com/drive/folders/1eIZmENmdeoT5nww6yljXjn_xGg7oBa9c?usp=sharing, los cuales se puede identificar con los nombres de los archivos escritos al inicio de cada programa o subprograma.

B.1.1. Programa 1: Método de Runge Kutta de orden 4, usando ode45 de MATLAB.

Nombre del Programa 1: *Solucion_Runge_Kutta_ode45_Laura.m*

```
% Solución del modelo SIR por el Método de Runge Kutta de
orden 4, usando ode45 de Matlab.
% EDO a resolver (MODELO SIR):
% dx(t)/dt = -beta*x(t)*y(t)/N;
% dy(t)/dt = beta*x(t)*y(t)/N - gamma*y(t);
% dz(t)/dt = gamma*y(t);

fprintf('Método de Runge Kutta de orden 4, usando ode45:
\n');

% Valores de los parámetros:
beta = 0.6; % beta debe estar entre 0.1 y 1
gamma = 0.3; % gamma debe estar entre 0.05 y 1
```

Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
t_0 = 0;      % Tiempo inicial
t_f = 61;     % Tiempo final

N = 1000;    % N es la población total.

% Valores iniciales en el tiempo inicial t_0=0 son:
y_0 = 1;     %y_0 es la condición inicial de los
             % Infectados I(t)
x_0 = N - y_0;%x_0 es la condición inicial de los
             % Susceptibles S(t)
z_0 = 0;     %z_0 es la condición inicial de los
             % Recuperados R(t)

% Número de reproducción básica: R_0 = beta*N/gamma
R_0 = beta*N/gamma;

% Eleegir opciones para ejecutar ode45:
options = odeset('AbsTol',1e-8,'RelTol',1e-8,'Stats','on'
                );

% Condiciones iniciales:
Yo = [x_0; y_0; z_0];

% Intervalo de tiempo:
tspan = [t_0, t_f];

% Llamado del software ode45 para aplicar Método de Runge
% Kutta de orden 4:
[T,Y] = ode45(@(t,y)EDO_RK(t,y,beta,gamma,N),tspan,Yo,
              options);

figure(5)
plot(T,Y(:,1),'b','linewidth',2,'MarkerSize',10)
title('Solución del PD por el método de Runge Kutta de
      orden 4','FontSize',22)
xlabel('$tiempo(días)$','Interpreter','latex','FontSize',
      22)
ylabel('$número$ $de$ $personas$', 'Interpreter','latex','
      FontSize',22)
set(gca, 'FontSize',20)
hold on
grid on
xlim([0 t_f]);
```

```
ylim([-1 N+1]);
l1=legend('$S(t)$ - Susceptibles','$I(t)$ - Infectados','$R(t)$ - Recuperados');
    set(l1,'Interpreter','latex','FontSize',24)

figure(5)
plot(T,Y(:,2),'r','linewidth',2,'MarkerSize',10)
xlabel('Tiempo (en días)','Interpreter','latex','FontSize',22)
ylabel('Número de personas','Interpreter','latex','FontSize',22)
set(gca, 'FontSize',20)
title('Solución del PD por el método de Runge Kutta de orden 4','FontSize',22)
hold on
grid on
xlim([0 t_f]);
ylim([-1 N+1]);
l1=legend('$S(t)$ - Susceptibles','$I(t)$ - Infectados','$R(t)$ - Recuperados');
    set(l1,'Interpreter','latex','FontSize',24)

figure(5)
plot(T,Y(:,3),'g','linewidth',2,'MarkerSize',10)
title('Solución del PD por el método de Runge Kutta de orden 4','FontSize',22)
xlabel('Tiempo (en días)','Interpreter','latex','FontSize',22)
ylabel('Número de personas','Interpreter','latex','FontSize',22)
set(gca, 'FontSize',20)
hold on
grid on
xlim([0 t_f]);
ylim([-1 N+1]);
l1=legend('$S(t)$ - Susceptibles','$I(t)$ - Infectados','$R(t)$ - Recuperados');
    set(l1,'Interpreter','latex','FontSize',24)
```

Subprograma 1, que usa el Programa 1

Nombre del Subprograma 1: *EDO_RK.m*

```
function dy = EDO_RK(t,y,beta,gamma,N)
```

```
% ESTE PROGRAMA DEFINE EL SISTEMA DE ECUACIONES
  DIFERENCIALES DEL MODELO SIR:
% dx(t)/dt = -beta*x(t)*y(t)/N;
% dy(t)/dt = beta*x(t)*y(t)/N - gamma*y(t);
% dz(t)/dt = gamma*y(t);

% beta=0.6;
% gamma=0.3;

dy(1)=-beta*y(1)*y(2)/N;
dy(2)= (beta*y(1)*y(2)/N) -gamma*y(2);
dy(3)= gamma*y(2);

dy = [dy(1); dy(2); dy(3)];

end
```

B.1.2. Programa 2: Método de Euler Implícito para graficar la función objetivo g .

Nombre del Programa 2: *Sol_Implic_EulerMethodInverseMatrix_Laura.m*

```
% Método de Euler Implícito para resolver el modelo SIR:
% dx(t)/dt = -beta*x(t)*y(t)/N;
% dy(t)/dt = beta*x(t)*y(t)/N - gamma*y(t);
% dz(t)/dt = gamma*y(t);

fprintf('Solución del modelo SIR por el Método de Euler
  Implícito (usando la matriz inversa): \n');

% PARÁMETROS EXACTOS O COEFICIENTES EXACTOS DEL MODELO
  SIR:
beta = 0.6; % beta debe estar entre 0.1 y 1
gamma = 0.3; % gamma debe estar entre 0.05 y 1
h=1e-1; % Tama o de paso.
N = 1000; % Número total de habitantes de la población.

% Condiciones iniciales de las EDO:
x_0 = N-1; % S_0 Condición inicial de los Susceptibles
y_0 = 1; % I_0 Condición inicial de los Infectados
z_0 = 0; % R_0 Condición inicial de los Recuperados
```

Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
t_0 = 0;    % Tiempo inicial
t_f = 61;   % Tiempo final

% Condiciones iniciales:
Yo = [x_0; y_0; z_0];

y_n = Yo;

% Intervalo de tiempo:
tspan = [t_0, t_f];

x0=[x_0, y_0, z_0]; % Valores iniciales
ne=3;               % número de EDO

n =(t_f-t_0)/h; % número de iteraciones

% Inicialización del tiempo
t(1)=t_0;

% Inicialización de matriz x(i,j)
% i denota el número de ecuación diferencial
% j denota el número de iteración del método de Euler
    Implícito
for i=1:ne
    x(i,1)=x0(i);
end

num_evals_f = 0; % Número de evaluaciones de la función f
                (lado derecho del modelo SIR).

% Iteración del Método Implícito de Euler
for j=1:n

    t(j+1)=t(j)+h;

    %[y_n_mas_1_k_mas_1,num_evals_f_n_k] =
        Newton_Method_3(t_n_mas_1,beta,gamma,y_n,h,
            max_iter,epsilon,N);
    [y_n_mas_1_k_mas_1,num_evals_f_n_k] = Newton_Method_3
        (t(j+1),beta,gamma,y_n,h,400,1e-8,N);

    for i=1:ne
```

Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
% i denota el número de ecuación diferencial
% j denota el número de iteración del método de Euler
    Implícito
x(i,j+1) = y_n_mas_1_k_mas_1(i,1);

y_n(i,1) = x(i,j+1);

end

num_evals_f = num_evals_f + num_evals_f_n_k;

end

fprintf('número de iteraciones del método implícito \n de
    Euler (usando la matrix inversa): %d \n',n);

fprintf('número de evaluaciones de f: %d \n',num_evals_f)
;

fprintf('número de evaluaciones de f_y: %d \n',
    num_evals_f);

fprintf('número de soluciones de sistemas lineales: %d \n
    ',num_evals_f);

% SOLUCIONES NUMÉRICAS DE LA EDO:
S1 = x(1,:); % Susceptibles
I1 = x(2,:); % Infectados
R1 = x(3,:); % Recuperados

figure(1)
plot(t,S1,'b','linewidth',1,'MarkerSize',10)
title('Solución del modelo SIR para los valores exactos
    de \beta^{\ast} y \gamma^{\ast} con h=0.01','
    Interpreter','tex','FontSize',18')
xlabel(' $t$ ','Interpreter','latex','FontSize',22)
ylabel(' $S(t)$ ','Interpreter','latex','FontSize',22)
set(gca, 'FontSize',20)
hold on
grid on
l1=legend('$S(t)$ - IEM (inverse matrix)','$I(t)$ - IEM (
    inverse matrix)','$R(t)$ - IEM (inverse matrix)');
set(l1,'Interpreter','latex','FontSize',20)
```


Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
% % % PARÁMETROS EXACTOS O COEFICIENTES EXACTOS DEL
    MODELO SIR:
% % beta = 0.667; % beta debe estar entre 0.1 y 1
% % gamma = 0.167; % gamma debe estar entre 0.05 y 1

nn=length(beta_i);
mm=length(gamma_j);

Residual = zeros(mm,nn);

for ii=1:mm
    for jj=1:nn

% Valores de los parámetros:
beta = beta_i(ii);
gamma = gamma_j(jj);

% Condiciones iniciales de las EDO:
x_0 = N-1; % donde N número de habitantes de la población
.
y_0 = 1;
z_0 = 0;

% Condiciones iniciales:
Yo = [x_0; y_0; z_0];
y_n = Yo;

% Intervalo de tiempo:
tspan = [t_0, t_f];
% t_0 es el Tiempo inicial
% t_f es el Tiempo final

x0=[x_0, y_0, z_0]; %Valores iniciales
ne=3; % número de EDO

n =(t_f-t_0)/h; % número de iteraciones

% Inicializacion del tiempo
t(1)=t_0;

%Inicialización de matriz x(i,j)
% i denota el número de ecuación diferencial
```

Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
% j denota el número de iteración del método de Euler
for i=1:ne
    x(i,1)=x0(i);
end

num_evals_f = 0;

% Iteración del Método Implícito de Euler
for j=1:n

    t(j+1)=t(j)+h;

    %[y_n_mas_1_k_mas_1,num_evals_f_n_k] =
        Newton_Method_3(t_n_mas_1,alpha,beta,gamma,y_n,h,
            max_iter,epsilon,N);

    [y_n_mas_1_k_mas_1,num_evals_f_n_k] = Newton_Method_3
        (t(j+1),beta,gamma,y_n,h,400,1e-8,N);

    for i=1:ne
        % i denota el número de ecuación diferencial
        % j denota el número de iteración del método de Euler
        x(i,j+1) = y_n_mas_1_k_mas_1(i,1);

        y_n(i,1) = x(i,j+1);

    end

    num_evals_f = num_evals_f + num_evals_f_n_k;

end

% SOLUCIONES NUMÉRICAS DEL MODELO SIR:
S2 = x(1,:);
I2 = x(2,:);
R2 = x(3,:);

Residual2 = 0;

for r=1:n % numero de particiones del intervalo del
    tiempo [t_0, t_f].
```

Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
Residual2 = Residual2 + (I1(1,r)-I2(1,r)).^2; %
    Residual2 es la función g(beta(jj),gamma(ii)).

end

Residual(jj,ii) = Residual2; % para graficar el
    funcional g(beta,gamma) para los diferentes valores de
    beta(jj) y gamma(ii).

    end
end

%%%%% GRÁFICA LA FUNCIÓN OBJETIVO g(beta,gamma) %%%%%

[A,B]=meshgrid(beta_i,gamma_j);

figure(2)
surf(A,B,Residual)
colormap jet
title('Función objetivo g(\beta,\gamma)','Interpreter','tex',
    'FontSize',18)
xlabel('$\beta$','Interpreter','latex','FontSize',18)
ylabel('$\gamma$','Interpreter','latex','FontSize',18)
zlabel('g(\beta,\gamma)','Interpreter','tex','FontSize',
    18)
set(gca,'FontSize',14)

figure(3)
surf(A,B,Residual)
colormap jet
title('Curvas de nivel de la función objetivo g(\beta,\gamma)',
    'Interpreter','tex','FontSize',18)
xlabel('$\beta$','Interpreter','latex','FontSize',18)
ylabel('$\gamma$','Interpreter','latex','FontSize',18)
zlabel('g(\beta,\gamma)','Interpreter','tex','FontSize',
    18)
set(gca,'FontSize',14)
```

Subprograma 1, que usa el Programa 2

Nombre del Subprograma 1: *Newton_Method_3.m*

```
% PROGRAMA PARA EL MODELO SIR POR EL MÉTODO IMPLICITO DE
    EULER:
```

Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
% dx(t)/dt = -beta*x(t)*y(t)/N;
% dy(t)/dt = beta*x(t)*y(t)/N - gamma*y(t);
% dz(t)/dt = gamma*y(t);

function [y_n_mas_1_k_mas_1,num_evals_f_n_k] =
    Newton_Method_3(t_n_mas_1,beta,gamma,y_n,h,max_iter,
    epsilon,N)

% beta y gamma son los parametros del modelo SIR.

% - max_iter es el número máximo de iteraciones.
% - h es el tamaño de paso del tiempo de [t_n, t_{n+1}].
% - t_n_mas_1 es el siguiente valor del tiempo 't' después
  de t_n

Delta_k = [0; 0; 0]; % Delta_k es un punto inicial para
    el método implícito de Euler.

num_evals_f_n_k = 0;

y_n_mas_1_k = y_n; % [0, 0, 0];

for k=1:max_iter

    [Dfx]=Jacobiano_3x3(y_n_mas_1_k,beta,gamma,N); %
        matriz Jacobiana de f evaluada en x_n

    [fx]=edo3por3(y_n_mas_1_k,beta,gamma,N); % función f
        evaluada en el punto x_n

    r_n_mas_1_k = y_n - y_n_mas_1_k + h.*fx; % Residual
        r_{n+1}^{(k)}

    num_evals_f_n_k = num_evals_f_n_k + 1;

    Matriz_n = eye(3,3)-h.*Dfx;
    deter_k = det(Matriz_n); % Se calcula el determinante
        de Matriz_n

    if deter_k ~= 0

        Delta_k_mas_1 = inv(Matriz_n)*r_n_mas_1_k;
```

Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
error_k = max(abs(Delta_k_mas_1 - Delta_k));

if error_k < epsilon,

    y_n_mas_1_k_mas_1 = Delta_k_mas_1 + y_n_mas_1_k
        ;

    break,
else

    y_n_mas_1_k_mas_1 = Delta_k_mas_1 + y_n_mas_1_k;

    y_n_mas_1_k = y_n_mas_1_k_mas_1; % Esta instrucción
        es para evaluar el jacobiano de f en el punto t_
        {n+1}.

Delta_k = Delta_k_mas_1;

end
end
    if deter_k == 0
fprintf('\n El determinante del sistema de ecuaciones
        es cero. \n');
    end
end
end
```

Subsubprograma 1, que usa el Subprograma 1 (del Programa 2)

Nombre del Subsubprograma 1: *Jacobiano_3x3.m*

```
% Cálculo de la matriz Jacobiana del segundo miembro de
    la EDO (modelo SIR):
% dx(t)/dt = -beta*x(t)*y(t)/N;
% dy(t)/dt = beta*x(t)*y(t)/N - gamma*y(t);
% dz(t)/dt = gamma*y(t);

function [J]=Jacobiano_3x3(y,beta,gamma,N)

% % Valores de los parámetros:
% beta=0.6;
% gamma=0.3;

% La funcion x(t) se denota por y(1)
```

Códigos de los programas elaborados
B.1 Programas para los problemas directos del modelo SIR

```
% La funcion y(t) se denota por y(2)
% La funcion z(t) se denota por y(3)

% J es la matriz Jacobiana de f (segundo miembro del
  modelo SIR):
J=[  -beta*y(2)/N,          -beta*y(1)/N,
      0;
      beta*y(2)/N,        (beta*y(1)/N) - gamma,
      0;
      0,                  gamma,
                        0];
```

end

Subsubprograma 2, que usa el Subprograma 1 (del Programa 2)

Nombre del Subsubprograma 2: *edo3por3.m*

```
% Cálculo de la matriz f(t,x(t),y(t),z(t)) del segundo
  miembro del modelo SIR:
% dx(t)/dt = -beta*x(t)*y(t)/N;
% dy(t)/dt = beta*x(t)*y(t)/N - gamma*y(t);
% dz(t)/dt = gamma*y(t);

function [fx]=edo3por3(x,beta,gamma,N)

% % Valores de los parámetros:
% beta=0.6;
% gamma=0.3;

yprime(1) = -beta.*(x(1).*x(2)./N);
yprime(2) = beta.*(x(1).*x(2)./N) - gamma.*x(2);
yprime(3) = gamma.*x(2);

fx=[yprime(1); yprime(2); yprime(3)]; % fx es la
  matriz f(t,x(t),y(t),z(t)) del segundo miembro del
  modelo SIR.

end
```

B.2. Programas para el problema inverso (PI) del modelo SIR

En esta sección se presentan los códigos de los programas empleados en el Capítulo 3, en lenguaje *MATLAB*, para resolver el PI del modelo SIR. Estos programas se pueden encontrar en la carpeta del software DIFFPAR y que puede bajar en el siguiente link: https://drive.google.com/drive/folders/1Qm6ET01NH4Q_zjub4dfk3VnE10LwcVdb?usp=sharing, los cuales se puede identificar con los nombres de los archivos escritos al inicio de cada programa. Además, para que DIFFPAR funcione bien se debe instalar la versión 2013a o 2013b de MATLAB.

Nombre del Programa 1 que usa DIFFPAR: *SIR_m.m*

```
function yprime=SIR_m(t,y,k);

% Modelo SIR, donde
% k(1) es el parámetro beta
% k(2) es el parámetro gamma
% y(1) Número de susceptibles (S)
% y(2) Número de infectados (I)
% y(3) Número de recuperados (R)

% N número de individuos en la población
N=1000;

yprime=[
           -k(1)*y(1)*y(2)/N;
(k(1)*y(1)*y(2)/N) - k(2)*y(2);
           k(2)*y(2)];

end
```

Nombre del Programa 2 que usa DIFFPAR: *SIR_d.m*

```
function [fy,fk]=SIR_d(t,y,k);

% Jacobianos fy y df/dk del modelo SIR, donde
% k(1) es el parámetro beta
% k(2) es el parámetro gamma
% y(1) Número de susceptibles (S)
% y(2) Número de infectados (I)
% y(3) Número de recuperados (R)

% N número de individuos en la población
N=1000;
```

```
f y = [ -k(1)*y(2)/N,          -k(1)*y(1)/N,          0;
        k(1)*y(2)/N,         k(1)*y(1)/N - k(2),        0;
        0,                    k(2),                    0];
```

```
f k = [-y(1)*y(2)/N,          0;
        y(1)*y(2)/N,         -y(2);
        0,                    y(1)];
```

end

B.3. Instrucciones para resolver el PI del modelo SIR aplicando DIFFPAR

En esta sección se presenta una guía rápida para el usuario de DIFFPAR, basado en The Graphical User Interface of DIFFPAR desarrollado por Wikström (1997) [72] y del artículo de Edsberg and Wedin (1995) [23]. Hasta el momento, el software más completo que se dispone para resolver el problema inverso del modelo SIR es el paquete DIFFPAR. DIFFPAR está basado en un método de regularización local del algoritmo de Gauss-Newton, para encontrar un mínimo local desde un punto inicial dado. En este apéndice se da una explicación breve de cómo usar las funciones para la estimación de parámetros. La función principal es el *dp_gui.m*, para ejecutar esta función se escribe *dp_gui* en la ventana de comandos de MATLAB y se abre la ventana principal como se encuentra en la Figura B.1, que es donde se trabajara todo. Los botones importantes a utilizar son los que se encuentran en la parte inferior de nuestra ventana, como lo son MODEL, DATA, DIFFPAR, PARÁMETROS.

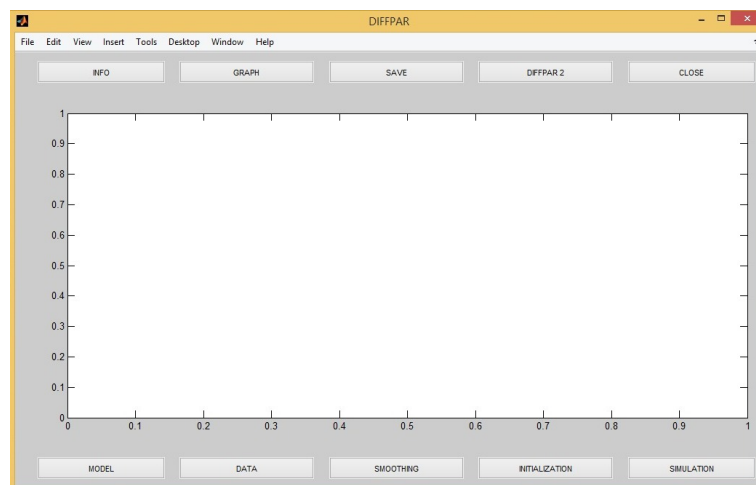


Figura B.1: Ventana principal de DIFFPAR

Para resolver el problema inverso de identificación de los parámetros β y γ del

modelo SIR se siguen los siguientes pasos:

Paso 1: Pulsamos el botón MODEL de la Figura B.1, se abrirá una ventana como se muestra en la Figura B.2, ahí es donde se especifica la función de la EDO que se va a trabajar (el modelo está programado en el archivo SIR_m.m), después se especifica el Jacobiano del modelo (en nuestro caso esta programado en el archivo SIR_d.m). Finalmente se especifica el número de variables de estado, es decir el número de componentes de la variable dependiente y que aparecen en el modelo, y el número de componentes del parámetro k ; en el caso del modelo SIR son 3 las variables dependientes (Susceptibles S , Infectados I y Recueperados R), y 2 los coeficientes del modelo SIR (tasa de infección β y tasa de recuperación γ), respectivamente. Una vez especificado todos los datos que se requieren se debe cerrar la ventana MODEL, pulsando el botón CLOSE.

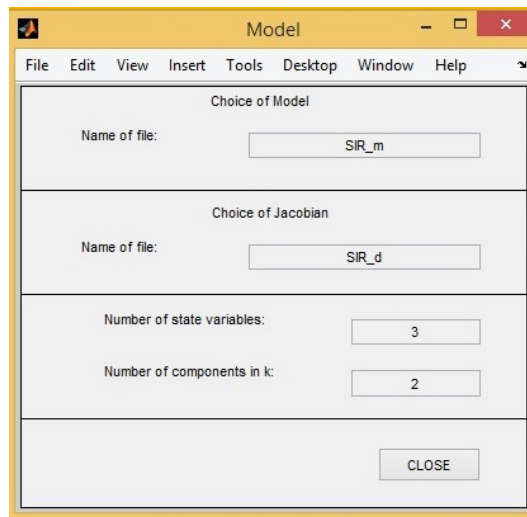


Figura B.2: Subventana MODEL

En nuestro caso vamos a generar datos artificiales, para eso hay que pulsar el botón *generation of data* de la Figura B.3, se mostrara una ventana como la que se muestra en la Figura B.4, ahí se generan los datos artificiales. Comenzamos indicando que nuestra ecuación diferencial a resolver es no stiff. En seguida especificamos los parámetros, para caso particular, por ejemplo para el modelo SIR, $k = (\beta, \gamma) = (0.667, 0.167)$ y las condiciones iniciales del modelo $y_0 = (S_0, I_0, R_0) = (999, 1, 0)$, también especificamos el intervalo de tiempo que se requiere $[t_0, t_{end}] = [0, 30]$. Para el tamaño de paso inicial h_0 (in %) = 5 y tolerancia tol (in %) = 0.1 del método se pueden dejar los que da DIFFPAR por default, en este caso, ambos en el tanto por ciento indicado. Ahora para generar la discretizacion del tiempo inicial al tiempo final, seleccionamos el botón de *equally spaced* de la Figura B.4 e indicamos el número de subintervalos en que se va a dividir el intervalo de tiempo $[t_0, t_{end}] = [0, 30]$, en nuestro caso es 30. Finalmente, generamos los datos presionando en STAR

Códigos de los programas elaborados

B.3 Instrucciones para resolver el PI del modelo SIR aplicando DIFFPAR

GENERATION de la Figura B.4 y los resultados de la solución numérica del modelo SIR se mostrarán en una gráfica en la ventana principal de la Figura B.1.

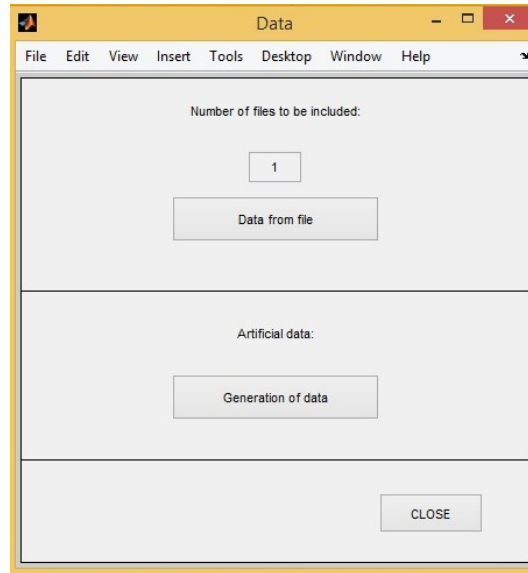


Figura B.3: Subventana DATA

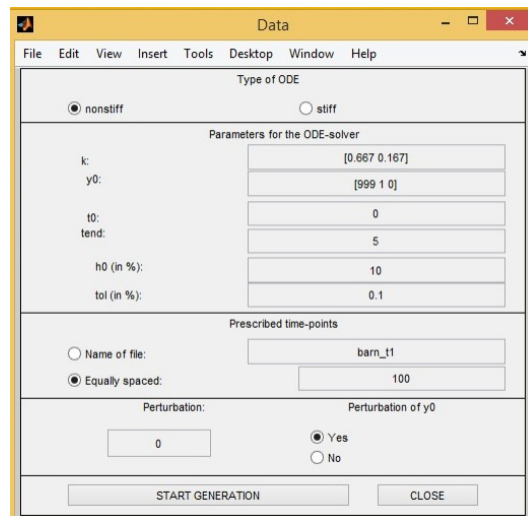


Figura B.4: Subventana para generar datos artificiales

Paso 2: Pulsaremos el botón DIFFPAR 2 de la Figura B.1, entonces la ventana principal cambia como se muestra en la Figura B.5.

Paso 3: Elegimos el botón DATA/WEIGHTS aquí se eligen los tipos de datos y pesos para la optimización, en este caso lo dejaremos tal cual está en la Figura

Códigos de los programas elaborados

B.3 Instrucciones para resolver el PI del modelo SIR aplicando DIFFPAR

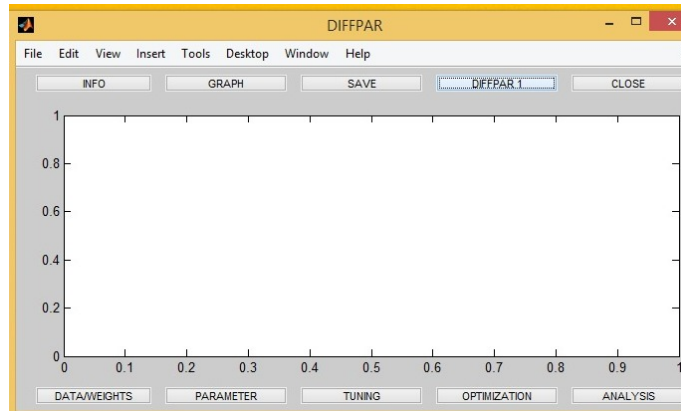


Figura B.5: Ventana principal de DIFFPAR ,para trbajar en la optimización

B.6, solo pulsaremos en el botón *Generate date*.

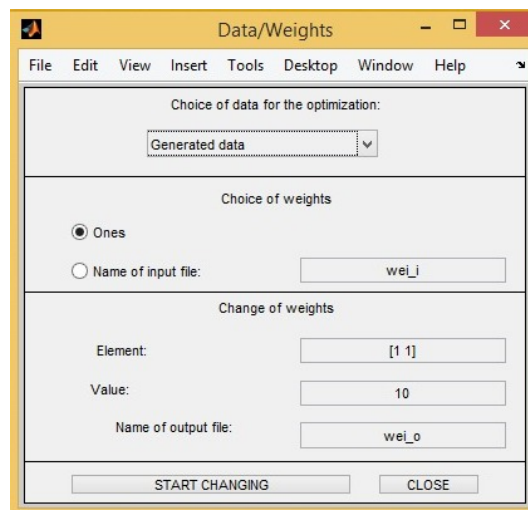


Figura B.6: Subventana Data/Weights

Paso 4: En TUNING se especifican algunos parámetros importantes para el proceso de optimización, por ejemplo, el tipo de EDO a resolver (stiff o no stiff), factores de tolerancia para resolver el EDO y para la optimización, y un número máximo de iteraciones (véase la Figura B.7). Es posible elegir la cantidad y tipo de información que se recibirá durante la iteración, y la posibilidad de graficar los resultados de cada iteración en la ventana principal de la Figura B.1.

Paso 5: Aquí pulsaremos en el botón de OPTIMIZATION (véase la Figura B.1.), pues es aquí donde se inicia el proceso de iteración para resolver el problema de mínimos cuadrados. En esta versión solamente hay una alternativa para el algoritmo de optimización, que es un método tipo *Gauss-newton+Discrete Trust-Region*,

Códigos de los programas elaborados

B.3 Instrucciones para resolver el PI del modelo SIR aplicando DIFFPAR

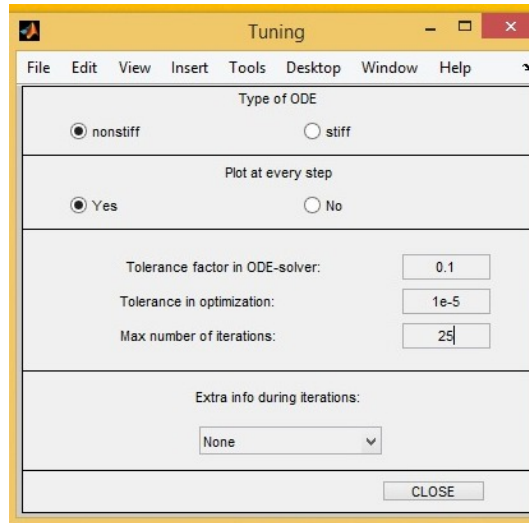


Figura B.7: Subventana TUNING

se inicia el proceso de optimización al pulsar el botón START OPTIMIZATION.

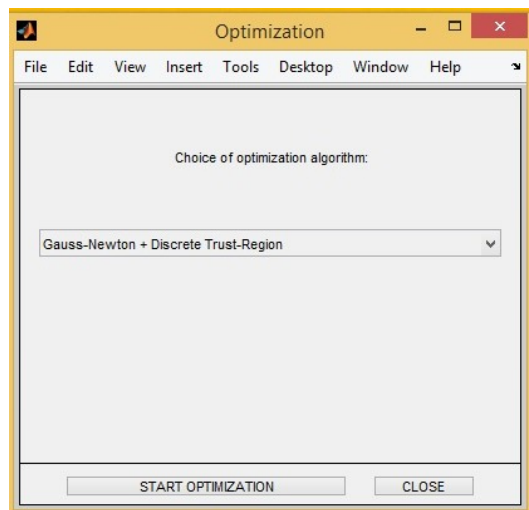


Figura B.8: Subventana OPTIMIZATION

Paso 6: Pulsamos en ANALYSIS es donde se puede obtener mayor información sobre los resultados finales del proceso de optimización y/o de análisis (véase la Figura B.1). El software DIFFPAR da información compacta de cada iteración (valores de los parámetros, tamaño del cuadrado del residual y la norma del gradiente), escritos sobre la ventana principal de MATLAB.

Códigos de los programas elaborados

B.3 Instrucciones para resolver el PI del modelo SIR aplicando DIFFPAR

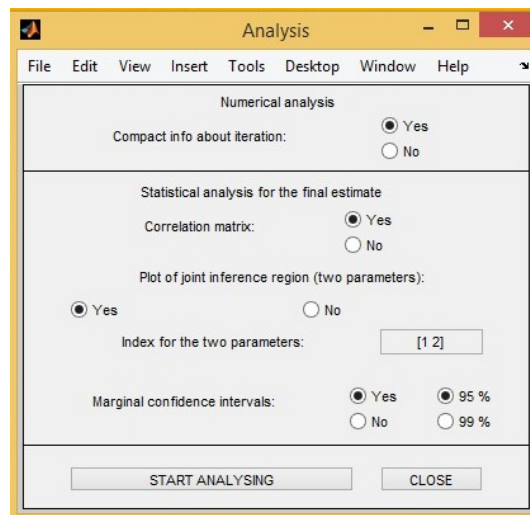


Figura B.9: Subventana de ANALISYS

Bibliografía

- [1] Abbasi, Z., Shafieirad, M., Amiri Mehra, A. H., & Zamani, I. (2023). Vaccination and isolation based control design of the COVID-19 pandemic based on adaptive neuro fuzzy inference system optimized with the genetic algorithm. *Evolving Systems*, vol. 14, no 3, p. 413-435.
- [2] Alcalá Vallejo Armando (2017). Un algoritmo de inversión basado en ecuaciones integrales para resolver el problema inverso de tomografía de capacitancia eléctrica para flujos bifásicos. Tesis de Maestría en Modelación Matemática. Universidad Tecnológica de la Mixteca. Huajuapán de León Oaxaca.
- [3] Algreto-Badillo, I., Conde-Mones J. J., Hernández-Gracidas, C. A., Morín-Castillo, M. M., Oliveros-Oliveros, J. J., & Feregrino-Urbe, C. (2020). An FPGA-based analysis of trade-offs in the presence of ill-conditioning and different precision levels in computations. *Plos one*, vol. 15. no. 6, e0234293.
- [4] Arguindegui Gabriela Alejandra (2021). Modelos Compartimentales SIR-SUS Enfoques Determinista y Estocástico. Máster en Estadística Aplicada; Facultad de Ciencias de la Universidad de Granada.
- [5] Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J. A., & Blom, J. G. (2009). Systems biology: parameter estimation for biochemical models. *The FEBS journal*, 276(4), 886-902.
- [6] Atkinson, K., Han, W., and Stewart, D. E. (2011). Numerical solution of ordinary differential equations. John Wiley & Sons.
- [7] Ávila-Pozos, R., & Rangel-Zúñiga, D. (2017). Revisión histórica sobre la Epidemiología Matemática. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, 2017, vol. 4, no. 8
- [8] Bacca Moreno Jorge Luis (2013). Un instrumento virtual para tomografía por capacitancia eléctrica. Tesis de Maestría en la Especialidad de Electrónica del Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla, Puebla, México.
- [9] Béraud, G. (2018). Mathematical models and vaccination strategies. *Vaccine*, Elsevier; vol. 36, no. 36, p. 5366-5372.

- [10] Bernoulli, D. (1760). Essai d'une nouvelle analyse de la mortalite causee par la petite verole et des avantages de l'inoculation pour la prevenir, *Memoirs of Mathematics and Physics of the Royal Academy of Sciences*, Paris.
- [11] Bertossi, V. I., Pastorelli, S. P., & Casco, E. S. (2019). Ecuaciones diferenciales ordinarias: libro digital interactivo.
- [12] Bhattar, S., et al. (2023). A new fractional mathematical model to study the impact of vaccination on COVID-19 outbreaks. *Decision Analytics Journal*, vol. 6, p. 100156.
- [13] Butcher, J. C. (2008). Numerical methods for ordinary differential equations. Second Edition, John Wiley & Sons.
- [14] Brauer, F., Castillo-Chávez, C., De La Pava, E., Barley, K., Castillo-Garsow, C. W., Chowell, D., Espinosa, B., Moreno, V., Barley, K., & Moreno, V. M. (2015). Modelos de la propagación de enfermedades infecciosas (Vol. 1). Cali, Colombia: Universidad Autónoma de Occidente.
- [15] Brauer, F., Castillo-Chavez, C., & Feng, Z. (2019). *Mathematical Models in Epidemiology* (Vol 32), Springer, Berlin, Germany.
- [16] Cavalleri, F., Irisarri, M., Bittar, G., Cuello, G., Pérez, M., & Aleman, A. (2020). Modelos epidemiológicos en la pandemia por SARS-CoV-2: concepto, aplicaciones y alcance. *Revista Uruguaya de Medicina Interna*, vol. 5, no. 2, p. 4-8.
- [17] COVID-19 Glosario sobre brotes y epidemias (2020). Un recurso para periodista y comunicadores. Organización Panamericana de la Salud (OPS); Organización Mundial de la Salud.
- [18] Conde Mones, J. J., Estrada Aguayo, E. R., Oliveros Oliveros, J. J., Hernández Gracidas, C. A., & Morín Castillo, M. M. (2021). Stable Identification of Sources Located on Interface of Nonhomogeneous Media. *Mathematics*, vol. 9, no. 16, 1932.
- [19] Conde Mones, J. J., Hernández Gracidas, C. A., Morín Castillo, M. M., Oliveros Oliveros, J. J., & Juárez Valencia, L. H. (2022). Stable Numerical Identification of Sources in Non-Homogeneous Media. *Mathematics*, vol. 10, no. 15, 2726.
- [20] Colmenarejo Pérez, Lucía (2021). Problemas inverso y redes neuronales. Trabajo de fin de grado en Matemáticas; grado en matemáticas del Departamento de Matemáticas, Facultad de Ciencias Universidad Autónoma de Madrid de la Universidad Autónoma de Madrid.

- [21] Dormand, J. R., & Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, vol. 6, no. 1, p. 19-26.
- [22] Dufera, T. T., Seboka, Y. C., & Fresneda Portillo, C. (2022). Parameter Estimation for Dynamical Systems Using a Deep Neural Network. *Applied Computational Intelligence and Soft Computing*.
- [23] Edsberg, L.; & Wedin, P. (1995). Numerical tools for parameter estimation in ODE-systems. *Optimization Methods and Software*, vol. 6, no. 3, p. 193-217.
- [24] El Badia, A., & Farah, M. (2006). Identification of dipole sources in an elliptic equation from boundary measurements: application to the inverse EEG problem. *Journal of Inverse and Ill-Posed Problems*, vol. 14, no. 4, p. 331-353.
- [25] Esteva, L., Gómez, G., Hernández, J., & Zepbda, M. (1991). *Matemáticas y epidemiología*. *Ciencias*, vol. 24, p. 57-63.
- [26] Fresnadillo Martínez, M. J., García-Sánchez, E., García-Merino, E., del Rey, Á. M., & García-Sánchez, J. E. (2013). Modelización matemática de la propagación de enfermedades infecciosas: de dónde venimos y hacia dónde vamos. *Revista Española de Quimioterapia*, vol 26, no. 2.
- [27] García Pallero José Luis (2009). *Técnicas geodésicas y riesgos naturales: el problema inverso de la gravimetría*. Tesis de Doctorado. Universidad Complutense de Madrid.
- [28] García Rovira, L. (2017). *Modelos matemáticos compartimentales en epidemiología*. Facultad de Farmacia Universidad la Laguna; Universidad la Laguna.
- [29] Ghostine, R., Gharamti, M., Hassrouny, S., & Hoteit, I. (2021). An extended SEIR model with vaccination for forecasting the COVID-19 pandemic in Saudi Arabia using an ensemble Kalman filter. *Mathematics*, vol. 9, no. 6, p. 636.
- [30] González Rodríguez, Benito J., Hernández Abreu Domingo, Jiménez Paiz Mateo M., Marrero Rodríguez M. Isabel & Sanabria García Alejandro (2013). *Ecuaciones diferenciales ordinarias de primer orden del departamento de Análisis Matemático*, Universidad de La Laguna.
- [31] Gonzales Arroyo María, 2017. *Modelización y simulación en epidemiología*. Trabajo de fin de grado, grado en matemáticas de la Facultad de Ciencias Matemáticas Universidad Complutense de Madrid.
- [32] Gonze, D., Coyte, K. Z., Lahti, L., & Faust, K. (2018). Microbial communities as dynamical systems. *Current opinion in microbiology*, vol. 44, p. 41-49.

- [33] Hairer, E., Nørsett, S. P., & Wanner, G. (1993). Solving ordinary differential equations I, Nonstiff problems. Springer-Vlg.
- [34] Hairer, E., & Wanner, G. (1996). Solving ordinary differential equations. II, volume 14 of Springer Series in Computational Mathematics.
- [35] Hammer, W. H. (1906). Epidemic disease in England. *Lancet*, I, p. 733–754.
- [36] Harko, T. Lobo, F. S., & Mak, M. (2014). Exact analytical solutions of the Susceptible-Infected-Recovered (SIR) epidemic model and of the SIR model with equal death and birth rates. *Applied Mathematics and Computation*, vol. 236, p. 184-194.
- [37] Hethcote, H. W., & Yorke, J. A. (1984). *Gonorrhoea transmission dynamics and control* (Vol. 56). Springer.
- [38] Higham, D. J. (2008). Modeling and simulating chemical reactions. *SIAM review*, vol. 50, no 2, p. 347-368.
- [39] Iijima, K. (2004). Numerical solution of backward heat conduction problems by a high order lattice-free finite difference method. *Journal of the Chinese Institute of Engineers*, vol. 27, no. 4, p. 611-620.
- [40] Jourhmane, M.; Mera, N.S. (2002). An Iterative Algorithm for the Backward Heat Conduction Problem Based on Variable Relaxation Factors. *Inverse Problems in Engineering*, vol. 10, no. 4, p. 293–308.
- [41] Juárez, J., Sánchez, A., Díaz, R., & Altamirano, L.(2020). Application of an Artificial Neural Network to Classify countries with COVID-19 adjusting to SIR model. *En Memorias del Congreso Nacional de Ingeniería Biomédica*. p. 454-461.
- [42] Kermack, W. O. & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, vol. 115, no. 772, p. 700-721.
- [43] Kirsch, A. (1996), *An introduction to the mathematical theory of inverse problems*. Springer Verlag.
- [44] Lambert, J. D. (1992). *Numerical methods for ordinary differential systems*. New York: Wiley.
- [45] Longini Jr, I. M., Ackerman, E., & Elveback, L. R. (1978). An optimization model for influenza A epidemics. *Mathematical Biosciences*, 1978, vol. 38, no 1-2, p. 141-157.

- [46] Martín Barroso, V. (2020). Una breve introducción al modelo SIR aplicado al caso del Covid-19. Instituto Complutense de Estudios Internacionales.
- [47] Martín García J. A., Gutiérrez Cabeza J. M. & Corz Rodríguez A. (2009). Two-dimensional non-linear inverse heat conduction problem based on the singular value decomposition, *International Journal of Thermal Sciences*, vol. 48, no. 6, p. 1081-1093.
- [48] Martínez, V. (2011). Problemas inversos: los casi olvidados de la Matemática Educativa.
- [49] Martín Vaquero, Jesús, 2005. Métodos Exponential Fitting y Adaptados para problemas stiff. Tesis de Doctorado en Ciencias Matemáticas; Departamento de Matemáticas y Física Aplicadas y Ciencias de la Naturaleza de la Universidad Rey Juan Carlos, Madrid, España.
- [50] Mathews, J. H., & Fink, K. D. (2000). Métodos Numéricos con Matlab. Madrid: Prentice Hall.
- [51] Mazzáfero, V. E. (2021). Salud pública: desde la molécula a la cultura. Universidad del Aconcagua.
- [52] Montesinos-López, O. A., & Hernández-Suárez, C. M. (2007). Modelos matemáticos para enfermedades infecciosas. *Salud pública de México*, vol. 49, no. 3, 218-226.
- [53] Morín M., Netzahualcoyotl, C., Oliveros, J., Conde, J., & Juárez, L. (2019). Stable identification of sources located on separation interfaces of two different homogeneous media. *Ad. Differ. Equ. Control Process*, vol. 20, p. 53-97.
- [54] Moya L.M. & Rojas E. (2020). Ecuaciones diferenciales ordinarias, técnicas de resolución. Universidad Nacional de Colombia, p. 85-89.
- [55] Novotná, V., & Štěpánková, V. (2015). Parameter estimation for dynamic model of the financial system. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, vol. 63, no 6, p. 2051-2055.
- [56] Ortega-Lenis, D., Arango-Londoño, D., Muñoz, E., Cuartas, D. E., Caicedo, D., Mena, J., Torres, M., & Mendez, F. (2023). Predicciones de un modelo SEIR para casos de COVID-19 en Cali, Colombia. *Revista de Salud Pública*, vol. 22, p. 132-137.
- [57] Ortigoza, G., Lorandi, A., & Neri, I. (2020). Simulación Numérica y Modelación Matemática de la propagación del Covid 19 en el estado de Veracruz. *Revista Mexicana de Medicina Forense y Ciencias de la Salud*, vol. 5, no. 3, p. 21-37.

- [58] Peifer, M., & Timmer, J. (2007). Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting. *IET systems biology*, vol. 1, no 2, p. 78-88.
- [59] Pérez Rodríguez Maria Soledad (2000). Integración de problemas stiff a través de métodos Runge-Kutta. Tesis Doctoral. Universidad de La Laguna, España.
- [60] Peter, O. J., Panigoro, H. S., Abidemi, A., Ojo, M. M., & Oguntolu, F. A. (2023). Mathematical model of COVID-19 pandemic with double dose vaccination. *Acta biotheoretica*, vol. 71, no. 2, p. 9.
- [61] Pliego Pliego Emilene Carmelita (2011). Modelos epidemiológicos de enfermedades virales infecciosas. Tesis de Licenciatura en Matemáticas; Licenciatura en Matemáticas de la Facultad de Ciencias Físico Matemáticas de la Benemérita Universidad Autónoma de Puebla, México.
- [62] Qiu, Z., Sun, Y., He, X., Wei, J., Zhou, R., Bai, J., & Du, S. (2022). Application of genetic algorithm combined with improved SEIR model in predicting the epidemic trend of COVID-19, China. *Scientific Reports*, vol. 12, no 1, p. 8910.
- [63] Qureshi, S., & Yusuf, A. (2019). Mathematical modeling for the impacts of deforestation on wildlife species using Caputo differential operator. *Chaos, Solitons & Fractals*, Vol. 126, p. 32-40.
- [64] Ross, R. (1911). *The prevention of Malaria*, Murray, 2nd ed., London.
- [65] San Martin Moreno, J., Tomeo Perucha, V., & Uña Juárez, I. (2005). *Métodos matemáticos: ampliación de matemáticas para ciencias e ingeniería*. Ediciones Paraninfo, SA.
- [66] Sanz Garayalde Iranzu (2016). Modelos epidemiológicos basados en ecuaciones diferenciales. Tesis de grado en Matemáticas; trabajo fin de grado en Matemáticas de la Facultad de Ciencias y tecnología en la universidad de la rioja.
- [67] Shampine, L. F., & Reichelt, M. W. (1997). The matlab ode suite. *SIAM journal on scientific computing*, vol. 18, no. 1, p. 1-22.
- [68] Shidfar, A., & Zakeri, A. (2005). A numerical technique for backward inverse heat conduction problems in one-dimensional space. *Applied Mathematics and Computation*, vol. 171, no. 2, p. 1016-1024.
- [69] Spataru, Daiana (2021). Using a genetic algorithm for parameter estimation in a modified SEIR model of COVID-19 spread in Ontario. Tesis Doctoral. University of Guelph.
- [70] Tjoa, I. B., & Biegler, L. T. (1991). Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial & Engineering Chemistry Research*, vol. 30, no 2, p. 376-385.

- [71] Vadillo, F., Métodos de un paso. Métodos Runge-Kutta (Notas). Departamento de Matemática Aplicada y Estadísticas de la Universidad del País Vasco. Recibido por los editores el 28 de marzo de 2019.
- [72] Wikström, G. (1997). PAPER IV: The Graphical User Interface of DIFFPAR. Department of Computing Science, Umeå University, Sweden.
- [73] Xu, Z., Wu, B., & Topcu, U. (2021). Control strategies for COVID-19 epidemic with vaccination, shield immunity and quarantine: A metric temporal logic approach. *Plos one*, vol. 16, no. 3, p. e0247660.
- [74] Yarsky, P. (2021). Using a genetic algorithm to fit parameters of a COVID-19 SEIR model for US states. *Mathematics and Computers in Simulation*, vol. 185, p. 687-695.
- [75] Zarate Siordia Luis Alberto, 2012. Modelación de enfermedades infecciosas con información geográfica. Tesis de Maestría en Ciencias Matemáticas Aplicadas e Industriales de la Universidad Autónoma Metropolitana Iztapalapa.
- [76] Zelenkov, Y., & Reshettsov, I. (2023). Analysis of the COVID-19 pandemic using a compartmental model with time-varying parameters fitted by a genetic algorithm. *Expert Systems with Applications*, vol. 224, p. 120034.
- [77] Zidarov, D. (1990). *Inverse Gravimetric Problem in Geoprospecting and Geodesy*, 2nd ed.; Elsevier: Amsterdam, The Netherlands.
- [78] Zill, D. G., & Cullen, M. R., (2009). *Ecuaciones diferenciales con problemas de valores en la frontera*. Séptima Edición. Cengage Learning.
- [79] Zisad, S. N., Hossain, M. S., Hossain, M. S., & Andersson, K. (2021). An integrated neural network and SEIR model to predict Covid-19. *Algorithms*, vol. 14, no. 3, p. 94.